

Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura



Tesis Doctoral

**Integración numérica por cuantificación de
estados: Generalización de algoritmos y
herramientas de análisis con aplicaciones.**

Mariana Bergonzi

Director: Dr. Ernesto Kofman
Co-Director: Dr. Rodrigo Castro

Miembros del jurado: Dr. María Soledad Aronna
Dr. César Pairetti
Dr. Mario Storti

*Tesis presentada en la Facultad de Ciencias Exactas, Ingeniería y Agrimensura, en
cumplimiento parcial de los requisitos para optar al título de*

Doctora en Ingeniería

Diciembre 2025

Certifico que el trabajo incluido en esta tesis es el resultado de tareas de investigación originales y que no ha sido presentado para optar a un título de postgrado en ninguna otra Universidad o Institución.

A handwritten signature in black ink, consisting of a stylized 'M' and 'B' intertwined.

Mariana Bergonzi

Agradecimientos

En primer lugar agradezco a mi director Ernesto Kofman, que además de sorprender cada día con su inteligencia es una excelente persona. Sin dudas con su pasión por cada cosa que hace, su generosidad, su humildad y la consistencia que muestra entre teoría y práctica, dichos y hechos, palabras y acciones, motiva a seguir su ejemplo (aunque a veces sea difícil de alcanzar).

En segundo lugar quiero agradecer a mi co-director Rodrigo Castro, que, a pesar de la distancia geográfica, siempre estuvo presente a lo largo de esta etapa.

También agradezco a mi mamá Inés y mi papá Pedro, quienes con su ejemplo me transmitieron, enseñaron y mostraron los valores más importantes de la vida. Lograron que el amor, la incondicionalidad, la generosidad y el apoyo siempre estuvieran y estén presentes en mi vida.

Le agradezco a mi compañero de vida Gustavo, con quién compartimos muchas experiencias que nos hicieron aprender y crecer juntos. Gracias por bancarme y ayudarme en este camino que decidí emprender, con todo lo que eso significa.

Les agradezco a los amigos y amigas que me dio el Cifa, sin dudas en este instituto hay muchas personas hermosas, con las que comparto muchos valores y hacen que la rutina sea más divertida, a pesar de los momentos difíciles que estamos atravesando.

No quiero dejar afuera al resto de mi familia, Zuly, Luli, Drun, Poli, Ani, Gra, Reven, Oli y Chacho quienes forman parte importante de mi vida.

Y ya con lágrimas en los ojos, agradezco a la vida, a la educación pública y al CONICET por permitirme transitar este bello camino lleno de aprendizajes, porque aprender es algo que me apasiona y disfruto enormemente.

Resumen

En esta tesis, presentamos nuevas herramientas teóricas para estimar costos computacionales derivados de la simulación de diferentes modelos a través de métodos de integración numérica. Esto incluye la introducción el concepto de *homogeneidad de la actividad* en las soluciones de ecuaciones diferenciales ordinarias (ODEs), el cual caracterizamos mediante una métrica denominada *Factor de Homogeneidad*. Mostramos que dicha métrica puede relacionarse con la conveniencia de utilizar esquemas clásicos de integración numérica o métodos basados en cuantificación de estados, tales como los algoritmos Quantized State System (QSS).

Las nociones desarrolladas, que también se extienden a sistemas que presentan discontinuidades, proporcionan un fundamento teórico que corrobora lo que intuitivamente se observa en la práctica: que los métodos QSS ofrecen ventajas cuando la actividad es heterogénea, los sistemas son dispersos y/o se producen discontinuidades frecuentes.

También proponemos una metodología para desarrollar nuevos algoritmos de integración numérica para ecuaciones diferenciales ordinarias basados en la cuantificación de estados, generalizando las nociones de los métodos Linearly Implicit Quantized State Systems (LIQSS). A partir de esta formulación, diseñamos dos nuevas subfamilias de algoritmos que mejoran el desempeño de los métodos LIQSS actuales, preservando al mismo tiempo sus propiedades de estabilidad, su cota de error global y su capacidad de manejo eficiente de eventos.

Finalmente, aplicamos los conceptos y algoritmos desarrollados a tres casos de estudio: un modelo de advección-difusión-reacción, una red neuronal y un modelo epidemiológico. Comparamos las predicciones teóricas con resultados empíricos obtenidos mediante simulaciones que emplean distintos métodos de integración numérica (clásicos, QSS originales y nuevos), confirmando que la métrica propuesta identifica de manera consistente la estrategia de integración que resulta más eficiente en la práctica. En estos resultados también resultó evidente la ventaja obtenida de los nuevos algoritmos propuestos con respecto tanto a los métodos originales de QSS, como a los métodos clásicos de integración numérica.

Abstract

In this thesis, we present new theoretical tools for estimating the computational costs arising from the simulation of different models using numerical integration methods. This includes the introduction of the concept of *activity homogeneity* in the solutions of ordinary differential equations (ODEs), which we characterize through a metric called the *Homogeneity Factor*. We show that this metric can be related to the suitability of using either classical numerical integration schemes or state-quantization-based methods such as Quantized State System (QSS) algorithms.

The notions developed here, which are also extended to systems exhibiting discontinuities, provide a theoretical foundation that supports what is commonly observed in practice: QSS methods offer advantages when activity is heterogeneous, systems are sparse, and/or discontinuities occur frequently.

We also propose a methodology for developing new numerical integration algorithms for ODEs based on state quantization, generalizing the framework of Linearly Implicit Quantized State Systems (LIQSS). Building on this idea, we design two new subfamilies of algorithms that improve the performance of current LIQSS methods while preserving their properties of stability, global error bounds, and efficient event handling.

Finally, we apply the concepts and algorithms developed in this work to three case studies: an advection–diffusion–reaction model, a neural network, and an epidemiological model. Theoretical predictions are compared with empirical results obtained through simulations using different numerical integration methods (classical methods, original QSS algorithms, and the newly proposed ones), confirming that the proposed metric consistently identifies the integration strategy that is most efficient in practice. These results also highlight the clear advantages achieved by the new algorithms compared with both the original QSS methods and classical numerical integration schemes.

Índice general

1. Introducción	14
1.1. Contexto y Motivación	15
1.2. Aportes de la Tesis	17
1.2.1. Discretización temporal vs. cuantificación de estados	17
1.2.2. Propuesta de métodos que se acerquen al costo computacional mínimo	18
1.2.3. Aplicaciones	19
1.3. Contribuciones Originales y Publicaciones	19
1.4. Organización de la Tesis	23
2. Preliminares	26
2.1. Clasificación de Modelos Matemáticos	26
2.2. Métodos Clásicos de Integración Numérica	27
2.2.1. Sistemas de tiempo continuo y ODEs	27
2.2.2. Métodos de Euler	28
2.2.3. Métodos monopaso	29
2.2.4. Métodos multipaso	30
2.2.5. PDEs y el Método de Líneas	30
2.2.6. Solvers de ODEs y herramientas de software	31
2.2.7. Limitaciones de los Métodos Clásicos de Integración Numérica	31
2.3. Métodos de Quantized State Systems	34
2.3.1. Origen de los métodos QSS basados en cuantificación de estados	34
2.3.2. Método de QSS1	34
2.3.3. Métodos QSS de Orden Superior	36
2.3.4. Ventajas Prácticas de QSS	37
2.3.5. Propiedades Teóricas de QSS	38
2.3.6. Métodos de QSS Linealmente Implícitos	39
2.3.7. Control de Error Relativo	43
2.3.8. Solver de QSS	43
2.4. El Concepto de Actividad	44

3. Homogeneidad de la Actividad	46
3.1. Ejemplo Motivador	47
3.2. Actividad instantánea, local y global	49
3.3. Relación entre actividad y mínimo costo computacional	50
3.3.1. Costos computacionales en métodos basados en cuantificación de estados	51
3.3.2. Costos computacionales en métodos basados en discretización temporal	52
3.4. Factor de Homogeneidad	53
3.4.1. Costos Computacionales y Factor de Homogeneidad	54
3.5. Ejemplo de aplicación	56
3.5.1. Actividades Locales, Globales y Factor de Homogeneidad	57
3.5.2. Número de Pasos y Actualizaciones	58
3.5.3. Factor de Homogeneidad y Actualizaciones de Estado	59
3.6. Extensión a Sistemas con Discontinuidades	59
4. Algoritmos de LIQSS Generalizados	63
4.1. Formulación Básica	63
4.2. Derivación de los Métodos LIQSS	66
4.2.1. Método LIQSS de primer orden	67
4.2.2. Método LIQSS de segundo orden	68
4.2.3. Método LIQSS de tercer orden	69
4.2.4. Política de Actualización del Estado Cuantificado	69
4.3. Método LIQSS de Chebyshev	70
4.3.1. Maximización del tamaño de paso	70
4.3.2. Método LIQSS de Chebyshev de primer orden	72
4.3.3. Método LIQSS de Chebyshev de segundo orden	73
4.3.4. Método LIQSS de Chebyshev de tercer orden	74
4.4. Implementación de los nuevos métodos	75
4.5. Comportamiento de los métodos LIQSS, eLIQSS y CheQSS	76
4.6. Propiedades de los métodos eLIQSS y CheQSS	78
5. Aplicación a un Modelo de Advección-Difusión-Reacción	79
5.1. Descripción del Modelo	80
5.2. Análisis Basado en Actividad	81
5.2.1. Actividades Locales, Globales y Factor de Homogeneidad	81
5.2.2. Número mínimo de pasos y actualizaciones	83
5.2.3. Factor de homogeneidad y actualizaciones de estado	84
5.2.4. Variación de parámetros	85
5.3. Eficiencia de los métodos	87

6. Aplicación a una Red Neuronal Pulsante	91
6.1. Modelos de neuronas y redes neuronales	91
6.2. Modelo utilizado	92
6.3. Análisis Basado en Actividad	93
6.4. Eficiencia de los métodos	97
7. Aplicación a un Modelo Epidemiológico Multiescala	100
7.1. Modelos epidemiológicos	100
7.2. Modelo Epidemiológico Multiagente Propuesto	101
7.2.1. Parámetros elegidos	104
7.3. Análisis Basado en Actividad	105
7.4. Eficiencia de los métodos	107
8. Conclusiones y trabajo futuro	110
8.1. Conclusiones	110
8.2. Trabajo futuro	112
A. Demostración del Teorema 1	120
A.1. Resultados auxiliares	120
A.2. Demostración del Teorema 1	122

Índice de figuras

2.1. Trayectorias típicas de un estado y su estado cuantificado en QSS1	35
2.2. Trayectorias típicas de un estado y su estado cuantificado en QSS2	37
2.3. Trayectorias típicas de un estado y su estado cuantificado en QSS3	37
2.4. Trayectorias típicas de estado y estado cuantificado en LIQSS1	39
2.5. Trayectorias del estado y del estado cuantificado en LIQSS2.	42
2.6. Trayectorias del estado y del estado cuantificado en LIQSS3.	42
3.1. Trayectorias del sistema ADR para distintos valores de parámetros.	48
3.2. Actividad instantánea (local y global) del modelo ADR para los diferentes casos	57
4.1. Trayectorias del estado y del estado cuantificado de diferentes métodos LIQSS (detalle) en la simulación de la Ec. (4.43).	77
4.2. Diferencia entre las trayectorias del estado y del estado cuantificado para dife- rentes métodos LIQSS (detalle) en la simulación de la Ec. (4.43).	78
5.1. Trayectorias de estado y actividades instantáneas (local y global) del modelo ADR	82
5.2. Trayectorias y actividades para distintos parámetros del modelo ADR.	85
5.3. Trayectorias del modelo ADR con $N = 100$	88
5.4. Errores y tiempos de simulación resultantes de los distintos algoritmos en la simulación del modelo ADR	89
6.1. Subconjunto de las trayectorias de estado para el modelo SNN considerando una corriente de entrada constante para cada neurona.	94
6.2. Subconjunto de trayectorias de estado para el modelo SNN considerando que cada neurona tiene como entrada un tren de spikes.	97
6.3. Tiempos de simulación según diferentes tolerancias de los distintos algoritmos en la simulación del modelo SNN	99
7.1. Algunas trayectorias de estados del modelo de huésped	105
7.2. Concentración de carga viral en el ambiente compartido	106

7.3. Curvas de contagio en el modelo epidemiológico para $N = 100$	106
7.4. Tiempos de simulación según el tamaño de la población de los distintos algoritmos en la simulación del modelo epidemiológico	109

Índice de tablas

3.1. Número de pasos requeridos para simular el modelo ADR.	49
3.2. Resumen de las definiciones de actividad.	50
3.3. Factores de Homogeneidad - Modelo ADR con $N = 4$	58
3.4. Cota inferior y número real de pasos para distintos métodos numéricos - Modelo ADR	59
3.5. Relación entre las actualizaciones de estado de algoritmos LIQSS y métodos clásicos de integración numérica.	59
4.1. Número de pasos y sus cotas inferiores en la simulación de la Ec. (4.43). . . .	76
5.1. Actividades locales de orden 1 a 3 para los estados del modelo ADR	81
5.2. Factores de Homogeneidad - Modelo ADR con $N = 10$	82
5.3. Cota inferior y número real de pasos utilizando diferentes métodos numéricos. Modelo ADR $N = 10$	83
5.4. Relación entre las actualizaciones de estado de los distintos algoritmos LIQSS y los métodos clásicos de integración numérica	84
5.5. Factores de Homogeneidad - Modelos ADR con variación de parámetros. . . .	86
5.6. Pasos y actualizaciones de simulación y sus cotas inferiores teóricas. Modelo ADR con variación de parámetros	86
5.7. Número de pasos, tiempo de CPU y error en la simulación del modelo ADR variando las tolerancias.	88
6.1. Parámetros del modelo SNN.	93
6.2. Número de discontinuidades (spikes) y número de pasos (cotas inferiores y valores observados) en distintos escenarios	95
6.3. Relación teórica y simulada entre los costos computacionales de un algoritmo de cuantificación de estados y uno de tiempo discreto. Modelo SNN	96
6.4. Número de pasos, tiempo de CPU y error en la simulación del modelo SNN variando el quantum y los valores de tolerancia.	98

7.1. Factores de Homogeneidad - Modelo epidemiológico	106
7.2. Número mínimo teórico y real de pasos y actualizaciones en el modelo epidemiológico	107
7.3. Número de pasos y tiempo de CPU en la simulación del modelo epidemiológico variando el tamaño de población.	108

Listado de Siglas

ADR	Advección–Difusión–Reacción
BDF	Backward Difference Formula (Fórmula de diferencia hacia atrás)
BE	Backward Euler (Euler hacia atrás)
CheQSS	Chebyshev QSS
DAE	Differential Algebraic Equations (Ecuaciones diferenciales algebraicas)
FE	Forward Euler (Euler hacia adelante)
HF	Homogeneity Factor (Factor de Homogeneidad)
LIF	Leaky Integrate and Fire (Modelo de integración y disparo con fuga)
LIQSS	Linearly Implicit Quantized State System (QSS Linealmente Implícitos)
MAE	Mean Absolute Error (Error Medio Absoluto)
MOL	Method Of Lines (Método de Líneas)
ODE	Ordinary Differential Equations (Ecuaciones diferenciales Ordinarias)
PDE	Partial Differential Equations (Ecuaciones Diferenciales Parciales)
QSS	Quantized State System (Sistemas de Estados Cuantificados)
RK	Runge-Kutta
TR	Trapezoidal Rule (Regla Trapezoidal)

Capítulo 1

Introducción

Los modelos matemáticos constituyen una herramienta fundamental para comprender, predecir y controlar el comportamiento de sistemas presentes en prácticamente todas las áreas de la ciencia y la ingeniería. Mediante la formulación de relaciones cuantitativas entre las variables relevantes, estos modelos permiten analizar fenómenos complejos, evaluar escenarios alternativos y diseñar estrategias de intervención o control. En particular, los *modelos dinámicos*, que describen la evolución temporal de un sistema, resultan esenciales cuando el interés está puesto en estudiar procesos que cambian en el tiempo, como por ejemplo sistemas biológicos, sistemas sociales, procesos físico-químicos, sistemas mecánicos, o redes neuronales. Según las características propias del sistema en estudio, existen distintas maneras de representar mediante distintos tipos de modelos matemáticos las dinámicas del mismo.

Si bien en algunos casos es posible obtener las soluciones exactas de estos modelos, en la mayoría de las aplicaciones de interés esto no es factible y es necesario recurrir a *métodos numéricos* que permitan obtener soluciones aproximadas mediante simulaciones computacionales. En este proceso, el costo computacional asociado a la integración numérica se convierte en un aspecto crítico, especialmente cuando los modelos son de gran escala (cuando está compuesto por un número elevado de variables de estado), presentan rigidez (cuando coexisten dinámicas muy lentas y muy rápidas y estas últimas requieren pasos de simulación muy pequeños para mantener la estabilidad) o incluyen discontinuidades frecuentes.

Los métodos de integración numérica pueden clasificarse en dos grandes grupos, según el enfoque que utilicen para realizar las aproximaciones: un grupo (clásico) se basa en la discretización de las variables en el tiempo, mientras que el otro enfoque (de desarrollo reciente) se basa en la cuantificación de los estados.

Cada uno de estos enfoques presenta ventajas y desventajas dependiendo de la naturaleza del sistema a simular y esto motiva una pregunta central:

¿Cuándo es más conveniente, en términos de costo computacional, utilizar discre-

tización temporal y cuándo cuantificación de estados?

Responder esta pregunta requiere un criterio cuantitativo que permita comparar ambos enfoques bajo condiciones generales. Para ello, en este trabajo se introduce una herramienta basada en el análisis de la *actividad* de las trayectorias del sistema, un concepto originalmente definido para medir la tasa de cambio de una señal continua. Esta herramienta incluye la definición de una serie magnitudes que están directamente relacionadas con el número mínimo de pasos teóricos necesarios para completar una simulación tanto para los métodos basados en cuantificación de estados como para los métodos basados en discretización temporal. A partir de ello se define el *Factor de Homogeneidad*, un indicador que permite identificar qué enfoque resultará más eficiente para un sistema dado.

Una vez establecido un criterio para determinar qué tipo de enfoque es en teoría más *barato*, surge una pregunta complementaria:

¿Cómo diseñar un algoritmo capaz de operar lo más cerca posible de este costo computacional mínimo?

En el proceso de búsqueda de respuestas a esta pregunta, surgió la necesidad de realizar una generalización de la formulación de los métodos linealmente implícitos basados en cuantificación de estados, lo que permitió no sólo reinterpretarlos bajo un marco unificado sino también proponer nuevas variantes optimizadas.

1.1. Contexto y Motivación

Como se mencionó anteriormente, los modelos matemáticos se utilizan para estudiar el comportamiento de sistemas en muchas áreas de la ciencia y la ingeniería. Dependiendo de cómo evolucionan en el tiempo las variables del sistema en estudio, se elige la manera más adecuada de representar matemáticamente la dinámica del mismo [69, 70].

Los *modelos continuos* de parámetros concentrados, en los cuales las variables evolucionan de forma continua en el tiempo, se describen mediante ecuaciones diferenciales ordinarias (ODEs) o algebraico diferenciales (DAEs). Los *modelos de tiempo discreto*, por su parte, actualizan el estado en instantes separados mediante ecuaciones en diferencias. Finalmente, en los *modelos basados en eventos* las variables se actualizan en instantes de tiempo como resultado de reglas lógicas, umbrales o interacciones que producen discontinuidades. Muchas aplicaciones reales presentan simultáneamente estos tres tipos de comportamiento, combinando partes continuas gobernadas por ODEs con dinámicas discretas que introducen discontinuidades en

las trayectorias, dando lugar a los denominados *sistemas híbridos*.

Para la mayoría de los modelos que contienen dinámicas continuas representadas por ODEs, resulta imposible obtener soluciones analíticas y es por ello que es necesario recurrir a *métodos de integración numérica* para aproximar la evolución de sus estados en el tiempo [32, 64, 19].

Dado que los sistemas híbridos incluyen dinámicas continuas, su simulación requiere la utilización de métodos de integración numérica que, debido a la interacción entre las partes continuas y discretas, suelen implicar altos costos computacionales. En muchas aplicaciones, estos modelos son el resultado de la interconexión de una gran cantidad de subsistemas híbridos simples, conformando redes complejas. Entre las aplicaciones mencionadas se incluyen modelos de redes inteligentes (*smart grids*), redes neuronales pulsantes (SNN, por *Spiking Neural Networks*), modelos biológicos multicelulares y sistemas multiagente, entre otros. En estos casos, al desafío inherente de la interacción entre las ODEs y la parte discreta se suma la gran escala del problema y la alta frecuencia de discontinuidades. En algunos casos la dinámica continua además es rígida, lo que obliga a utilizar métodos implícitos que agregan un alto costo computacional adicional. En consecuencia, el problema de simulación requiere de una gran cantidad de cálculos y se torna excesivamente costoso.

Por estos motivos, la simulación numérica de modelos de gran escala discontinuos y eventualmente rígidos es un problema abierto, que requiere soluciones que incluyan la mejora y el desarrollo de nuevas técnicas y herramientas de simulación que resulten eficientes en problemas que actualmente no son tratables en tiempos razonables.

Tradicionalmente, la aproximación numérica de ODEs se realiza mediante técnicas de *discretización temporal*, donde todos los estados del sistema se actualizan sincrónicamente en una serie de instantes a lo largo de la simulación. Métodos como Euler, Runge–Kutta y solvers como DOPRI o DASSL siguen este enfoque [19], que resulta conveniente cuando la dinámica es relativamente homogénea entre las variables.

Sin embargo, estos métodos enfrentan dificultades importantes ante la presencia de discontinuidades. Los pasos que atraviesan discontinuidades suelen introducir errores inadmisibles, por lo que deben evitarse mediante algoritmos de *detección y manejo de cruces por cero*. Estos algoritmos requieren localizar con precisión el instante del evento, de modo de avanzar la simulación hasta ese momento. Luego, la integración puede reiniciarse a partir de la situación posterior a la ocurrencia del evento [19]. Este proceso implica un elevado costo computacional debido a la necesidad de iteraciones adicionales y pasos de integración cada vez más pequeños, dificultades que se agravan cuando el modelo es de gran tamaño, ya que aumenta tanto el costo de cada paso como la frecuencia de discontinuidades, volviendo muchas veces impracticable la simulación.

Los métodos de *Quantized State Systems* (QSS) [47], basados en cuantificación de estados, constituyen una alternativa conceptualmente distinta. En lugar de avanzar el tiempo

globalmente, los métodos QSS actualizan individualmente una variable de estado cuando esta experimenta un cambio suficientemente grande. Este enfoque asincrónico y orientado a eventos es muy ventajoso cuando en cierto momento la actividad de un modelo está localizada en una pequeña parte del mismo (unas pocas variables de estado experimentan grandes cambios, mientras las otras no sufren modificaciones o solo cambian en menor medida) [69] ya que realizan cálculos únicamente en las variables activas. Además, el manejo de discontinuidades se simplifica de manera notable [43]: una discontinuidad local provoca simplemente un evento local, sin necesidad de reiniciar la simulación completa. Por estas razones, los métodos QSS han mostrado ventajas significativas en la simulación de sistemas híbridos como modelos de convertidores de electrónica de potencia [52], redes neuronales [12], sistemas multiagentes [54], modelos biológicos [1, 62, 22], entre otros. Estas ventajas se extienden a ciertos problemas puramente continuos donde las variables evolucionan de manera heterogénea, como en ciertos problemas de advección-difusión-reacción [4].

En estudios dentro de estos dominios, las simulaciones basadas en QSS suelen alcanzar velocidades de cómputo que superan en más de un orden de magnitud a las obtenidas por los integradores clásicos más eficientes. Uno de los objetivos centrales de esta Tesis es potenciar estas ventajas mediante **refinamientos teóricos, nuevas formulaciones y optimizaciones de algoritmos**.

1.2. Aportes de la Tesis

Esta tesis aborda dos cuestiones relacionadas con la eficiencia en simulaciones de sistemas puramente continuos o con presencia de discontinuidades. En primer lugar, establece un marco teórico que permite identificar, para un sistema dinámico general, qué enfoque de integración numérica (discretización temporal o cuantificación de estados) resulta más eficiente. Segundo, propone una optimización de algoritmos que permite acercar el rendimiento real de los métodos LIQSS al costo computacional teórico mínimo, mejorando sustancialmente su eficiencia. Todos estos conceptos son luego abordados en el contexto de tres aplicaciones.

1.2.1. Discretización temporal vs. cuantificación de estados

La conveniencia de utilizar métodos basados en discretización temporal o métodos basados en cuantificación de estados depende, en gran medida, de la naturaleza del sistema. Como mencionamos anteriormente, en sistemas con dinámica homogénea suele resultar más eficiente actualizar todas las variables de manera simultánea. En cambio, cuando la dinámica es heterogénea y localizada, es más eficiente actualizar únicamente las variables activas.

Esto conduce a preguntarse qué tan homogéneo debe ser el comportamiento de un sistema para que sea más conveniente utilizar discretización temporal en la simulación, y cuánta hete-

rogeneidad debe presentar para que resulte más eficiente el enfoque basado en cuantificación de estados.

Para establecer un criterio cuantitativo que permita comparar ambos enfoques bajo condiciones generales, es necesario considerar cómo opera cada tipo de algoritmo. Los métodos clásicos de simulación para sistemas continuos descritos por ecuaciones diferenciales ordinarias, al basarse en técnicas de discretización temporal [32, 33, 19], avanzan el estado global del sistema mediante actualizaciones sincrónicas en instantes discretos para todas las variables, independientemente de las dinámicas localizadas que puedan existir. Por otro lado, los métodos basados en cuantificación de estado, como los métodos de QSS [41, 19, 51], operan realizando actualizaciones asincrónicas sobre variables individuales del estado únicamente cuando se producen cambios suficientemente significativos.

Basado en este análisis, surge el primer aporte de esta Tesis, que consiste en extender el concepto de *actividad* [18], originalmente definido para medir la tasa de cambio de una señal continua, distinguiendo entre la *actividad local* asociada a cada variable individual y la *actividad global* asociada al estado completo. Mostramos que estas magnitudes están directamente relacionadas con el número teórico mínimo de pasos necesarios para los métodos QSS y para los métodos de discretización temporal, respectivamente.

Luego, utilizando las expresiones obtenidas para la actividad local y global, caracterizamos la *homogeneidad de la actividad* mediante la definición del *Factor de Homogeneidad*, un indicador que permite identificar de forma cuantitativa cuál de los dos paradigmas de simulación resulta más eficiente (en teoría) para un sistema dado. Este análisis constituye también un **corroboración teórica** de numerosos resultados empíricos publicados que comparan métodos QSS con algoritmos numéricos clásicos, donde la conveniencia de los QSS está asociada a combinaciones de actividad heterogénea, dispersión y presencia de discontinuidades frecuentes.

1.2.2. Propuesta de métodos que se acerquen al costo computacional mínimo

Al estimar teóricamente el límite inferior de la cantidad de pasos requeridos por un método numérico, surge la pregunta de qué características debe tener un algoritmo para lograr operar lo más cerca posible de este costo computacional mínimo.

Para responder esta pregunta, nos enfocamos en los algoritmos de QSS debido a las ventajas que presentan en gran cantidad de sistemas híbridos de gran escala, con alto costo computacional. La familia de métodos QSS está compuesta por los algoritmos explícitos (QSS) y linealmente implícitos (LIQSS) de primer, segundo y tercer orden [47, 42, 44, 51]. Estos últimos son capaces de simular eficientemente ciertos tipos de sistemas rígidos (otro aspecto que, como mencionamos, aumenta el costo computacional).

El segundo aporte de esta tesis consiste en desarrollar una nueva metodología para diseñar métodos LIQSS basada en la construcción explícita de los polinomios que determinan la diferencia entre el estado y su versión cuantificada. Esta metodología permite recuperar la formulación original de los métodos LIQSS, mejorarla sistemáticamente y proponer una nueva familia de métodos basados en polinomios de Chebyshev, denominados CheQSS. Bajo supuestos ideales, estos métodos maximizan el tamaño de paso admisible para una tolerancia dada, acercándose al límite teórico mínimo derivado del análisis de actividad.

1.2.3. Aplicaciones

Para corroborar la eficacia de los conceptos de actividad y de los nuevos métodos de LIQSS desarrollados, seleccionamos tres modelos con distintas características que implican una simulación con alto costo computacional. El primero de ellos es un modelo continuo de advección–difusión–reacción unidimensional discretizado espacialmente que presenta rigidez. El segundo es una red neuronal pulsante de gran escala que presenta un gran número de discontinuidades. El tercero es un modelo epidemiológico basado en agentes, de gran escala, con rigidez y con discontinuidades.

1.3. Contribuciones Originales y Publicaciones

A continuación se detallan los trabajos publicados tanto en revistas científicas como en distintas conferencias nacionales. En primer lugar se mencionan los artículos que conforman el contenido de esta tesis y luego se nombran otros trabajos adicionales que también fueron realizados a lo largo del doctorado.

Los primeros trabajos estuvieron relacionados al estudio de la eficiencia de distintos métodos de la familia QSS en sistemas complejos, de gran escala, con combinación de dinámicas continuas y discretas. En el trabajo titulado *Modelado y Simulación Eficiente de Sistemas Epidemiológicos Multiescala*, presentado en los congresos MACI 2023 [11] y RPIC 2021 [10], el sistema elegido para modelar y simular fue un modelo epidemiológico basado en agentes que combina la dinámica biológica de una infección viral en los huéspedes con la transmisión del virus entre individuos de una población mediante un ambiente compartido, produciendo así un modelo multiescala y rígido. Observamos que, en primer lugar, el modelo es capaz de reproducir evoluciones típicas de la propagación de una epidemia en una población y permite estudiar el impacto de parámetros biológicos y comportamientos sociales en las curvas de contagio.

Desde el punto de vista computacional, se simulieron distintos escenarios para analizar los costos computacionales de diferentes algoritmos de integración numérica basados en dos distintos enfoques: uno basado en discretización temporal y otro basados en cuantificación de

estados. Luego del análisis de los resultados obtenidos se comprobó la conveniencia de utilizar métodos de integración por cuantificación en este tipo de modelo.

Otro artículo que forma parte de esta Tesis es un estudio vinculado a la eficiencia computacional de los métodos QSS en la simulación de redes neuronales pulsantes, publicado en el *Journal of Simulation* bajo el título *Quantized based simulation of spiking neurons: theoretical properties and performance analysis* [12]. En el mismo comenzamos a utilizar los conceptos existentes de actividad para estimar teóricamente una cota del costo computacional de la simulación del modelo de una neurona mediante métodos basados en cuantificación de estados y contrastar estas estimaciones con los resultados obtenidos en distintos experimentos de simulación. Además, realizamos un análisis teórico en el que presentamos cotas de error para la simulación de este tipo específico de sistemas con los métodos de QSS. Corroboramos estos resultados mediante la simulación de este modelo bajo diversas condiciones.

Luego extendimos el estudio a redes neuronales, comparamos los errores obtenidos de distintos métodos numéricos y observamos cómo escalan estos algoritmos con el tamaño de la red y su conectividad. Los resultados obtenidos muestran que los algoritmos QSS, sin ningún tipo de optimización o especialización, logran aproximaciones precisas con menor costo computacional, incluso en redes de gran tamaño y con un alto nivel de conectividad. El desempeño de estos algoritmos fue superado luego por los algoritmos de LIQSS optimizados desarrollados posteriormente en el contexto de esta Tesis.

Tanto el análisis del estado del arte como parte de las aplicaciones que forman parte de esta Tesis fueron publicados en el artículo *Discrete-event simulation of continuous-time systems: evolution and state of the art of quantized state system methods* [17] en la revista *Simulation*. Este artículo reúne los orígenes, los resultados principales y los avances recientes de la simulación de eventos discretos aplicada a sistemas de tiempo continuo. En el trabajo se muestra de qué manera, a partir de los enfoques tempranos que intentaron representar dinámicas de tiempo continuo dentro del marco formal del Discrete Event System Specification (DEVS), se llegó a la formalización de la familia de algoritmos de integración numérica Quantized State System. Además, se describen los algoritmos QSS, sus propiedades, sus extensiones hasta el momento y las principales herramientas de software que los implementan. El trabajo incluye varios ejemplos de simulación que ilustran sus características y ventajas principales mediante comparaciones con los *solvers* de simulación de tiempo continuo más avanzados del estado del arte.

Uno de los aportes principales de esta Tesis fue publicado en el artículo *Activity Homogeneity: A Measure for Comparing Time Discretization and State Quantization in ODE Simulation* [7] en la revista *Simulation*. En este trabajo introducimos el concepto de *homogeneidad de la actividad* en las soluciones de ecuaciones diferenciales ordinarias (ODEs) y lo caracterizamos mediante una métrica denominada *Factor de Homogeneidad*. Este indicador

cuantifica el grado de similitud en la evolución temporal de las variables de estado del sistema. Mostramos que el valor de esta medida está directamente vinculado con estimaciones de las cotas teóricas del número mínimo de pasos requeridos por métodos clásicos de integración numérica y por métodos basados en cuantificación de estados, como los algoritmos Quantized State System, permitiendo identificar cuál de los enfoques se espera que presente un mejor desempeño.

Las nociones desarrolladas, que también se extienden a sistemas que presentan discontinuidades, proporcionan un argumento teórico que corrobora observaciones de trabajos previos, indicando que los métodos QSS ofrecen ventajas cuando la actividad es heterogénea, los sistemas son ralos (sparse) y/o ocurren discontinuidades frecuentes.

Los conceptos se aplicaron a dos casos de estudio presentes en esta tesis: un modelo de advección-difusión-reacción y una red neuronal pulsante. Las predicciones teóricas se comparan con resultados empíricos obtenidos mediante simulaciones utilizando diferentes métodos de integración numérica, confirmando que la métrica propuesta identifica de manera consistente la estrategia de integración que resulta más eficiente computacionalmente en la práctica.

Dos versiones resumidas preliminares de este trabajo fueron presentadas en dos congresos nacionales: MACI 2025 [9] y JAIIO 2025 [8] bajo los títulos *¿Discretización temporal o de estados? Conveniencia según el concepto de actividad homogénea* y *Discretización Temporal vs Cuantificación de Estados: Homogeneidad de la Actividad y Discontinuidades*, respectivamente.

Otro aporte principal de esta Tesis es el artículo titulado *On General Linearly Implicit Quantized State System Methods*, que se encuentra en revisión en la revista *Computational and Applied Mathematics*. En este trabajo se propone una metodología para desarrollar nuevos algoritmos de integración numérica para ecuaciones diferenciales ordinarias basados en la cuantificación de estados, generalizando las nociones de los métodos Linearly Implicit Quantized State Systems (LIQSS). A partir de esta idea, se diseñan dos subfamilias novedosas de algoritmos que mejoran el rendimiento de los métodos LIQSS actuales, preservando a su vez sus propiedades en cuanto a estabilidad, cota global del error y al manejo eficiente de discontinuidades. Se estudian las características de los nuevos algoritmos en dos ejemplos de aplicación presentes en esta tesis y también se analizan las ventajas respecto de los algoritmos clásicos de integración numérica.

Para completar, nombraré los siguientes trabajos, que fueron realizados a lo largo de esta carrera, pero cuyo contenido no aparece en esta tesis.

El artículo *Quantized State Simulation of Stochastic Differential Equations* fue aceptado recientemente en la revista *Computational and Applied Mathematics* [57]. Este trabajo explora el uso de los métodos QSS para la simulación de Ecuaciones Diferenciales Estocásticas (SDEs). Con ese propósito, se propone una extensión de estos algoritmos en la cual el proceso de

Wiener que gobierna la dinámica se muestrea a intervalos regulares, mientras que los estados se actualizan de manera asincrónica cuando satisfacen las condiciones de umbral correspondientes al método QSS utilizado.

Mostramos que los esquemas resultantes producen trayectorias que convergen a las soluciones reales de las SDEs cuando tanto el intervalo de muestreo h como los parámetros de cuantificación de los estados tienden a cero. Además, demostramos que, en casos lineales estables e invariantes en el tiempo, la norma esperada del error está globalmente acotada por una función lineal de la cuantificación y del intervalo de muestreo h , lo que evidencia que el esquema propuesto preserva la estabilidad práctica independientemente de la elección de estos parámetros. También presentamos experimentos de simulación que ilustran algunas ventajas potenciales del esquema.

Previamente, atravesados por el contexto social, habíamos trabajado en un modelo simple epidemiológico de Covid, publicando el artículo *Discrete-Time Modeling of COVID-19 Propagation in Argentina with Explicit Delays* [13] en la revista *Computing in Science & Engineering*. En dicho trabajo presentamos un nuevo modelo determinista de tiempo discreto y por compartimentos para COVID-19 que tiene en cuenta explícitamente los retrasos relevantes asociados a las distintas etapas de la enfermedad, su diagnóstico y el sistema de reporte, permitiendo además representar la presencia de casos importados.

Además de desarrollar las ecuaciones del modelo, describimos un mecanismo automático de ajuste de parámetros utilizando datos oficiales sobre la propagación del virus en Argentina. El resultado refleja de manera consistente el comportamiento de la enfermedad respecto de los tiempos característicos –latencia, período infeccioso, reporte de casos (confirmados y fallecidos)– y permite detectar automáticamente cambios en el número reproductivo y en el factor de mortalidad.

También analizamos la capacidad predictiva del modelo y presentamos resultados de simulación bajo diferentes escenarios futuros. Incluimos una aplicación del modelo en un esquema de control en lazo cerrado, donde la presencia explícita de retrasos desempeña un papel clave para proyectar dinámicas más realistas que las de los modelos clásicos en tiempo continuo.

Una versión resumida de este trabajo titulada *Modelado con Retardos Explícitos de la Propagación de COVID-19 en Argentina* se presentó en los congresos de AADECA 2020 [5] y AUGM 2020 [6].

Por último presentamos, también bajo influencia del contexto regional, un trabajo titulado *Modelado Orientado a Objetos de Sistemas Poblacionales y Epidemiológicos en Tiempo Discreto* que fue presentado en las JAIIO 2025 [40], en el que presentamos una librería básica orientada a objetos que permite representar modelos epidemiológicos compartimentales de tiempo discreto con retardos explícitos. Los modelos pueden diseñarse con distintas complejidades: poblaciones variables (con nacimientos y muertes) o constantes; puede incluir entrada

de personas ajenas a la población inicial y también salidas. La librería también brinda la posibilidad de representar de manera muy simple epidemias transmitidas por un vector.

Además de la descripción de la librería, el presente trabajo muestra modelos de ejemplo y resultados simulación en un modelo de dengue que tiene en cuenta fenómenos meteorológicos que se ajustan a datos reales.

1.4. Organización de la Tesis

En primer lugar, se presentan las temáticas y los conceptos fundamentales en los que se basa el resto del trabajo. Los mismos se encuentran en el Capítulo 2, en el que se detallan definiciones correspondientes a sistemas continuos e híbridos. Además se introducen los métodos clásicos de integración numérica, incluyendo sus principales usos y problemáticas existentes. Luego, se presentan los métodos de QSS, explicando sus características y las ventajas que presentan. Finalmente, se describe el concepto de actividad, el cual será de suma utilidad en el desarrollo de las herramientas presentadas más adelante.

A partir del Capítulo 3, todo el contenido presentado corresponde a investigaciones originales. Dicho capítulo comienza con un ejemplo motivador que permite introducir la problemática que se busca abordar. A continuación, se presentan los conceptos de actividad instantánea, local y global, así como su relación con el costo computacional inherente a la simulación de un sistema dado mediante un determinado algoritmo de integración numérica. En este marco, se determina el número mínimo teórico de pasos necesarios para completar la simulación, el cual constituye un indicador fundamental del costo computacional esperado, si bien el costo total también depende del costo asociado a cada paso de integración.

Posteriormente, se introduce una métrica denominada Factor de Homogeneidad, junto con sus propiedades y sus posibles aplicaciones. Seguidamente, los conceptos desarrollados se aplican a un ejemplo sencillo, en el cual se contrastan los resultados teóricos obtenidos con aquellos derivados de simulaciones realizadas utilizando distintos métodos de integración numérica. Finalmente, todas las definiciones previas se extienden para contemplar sistemas con discontinuidades, ampliamente utilizados en el ámbito de la simulación.

Pasando al Capítulo 4, se encuentra el segundo desarrollo teórico de este trabajo, el cual consiste en la generalización de los algoritmos Linealmente Implícitos de QSS (LIQSS). Se comienza presentando las ideas principales y una formulación básica y luego las formulaciones particulares de los algoritmos de primer, segundo y tercer orden. Para concluir la primera parte del capítulo, se propone una mejora en la implementación de los métodos de LIQSS clásicos, llamada eLIQSS, que genera un aumento en la eficiencia que tienen estos métodos en la aproximación numérica de ODEs.

Luego, en la segunda parte del capítulo, se presenta una nueva familia de algoritmos

linealmente implícitos, aprovechando la generalización de los mismos. Esta nueva familia se desarrolló con el objetivo de aumentar aun más la eficiencia, buscando aproximar el costo computacional de un algoritmo basado en cuantificación de estados al óptimo teórico obtenido en el Capítulo 3. Para esto, se demuestra que este óptimo se alcanza cuando la diferencia entre las trayectorias de los estados y sus versiones cuantificadas la describe un polinomio de Chebyshev, y luego se diseñan algoritmos de primer, segundo y tercer orden que verifican esta propiedad. Estos algoritmos se denominan CheQSS (por Chebyshev QSS).

Cerrando este capítulo, y para lograr visualizar y comparar los algoritmos LIQSS clásico, el extendido eLIQSS y el nuevo CheQSS, se presentan los resultados de simulación en un sistema sencillo de primer orden, con las distintas implementaciones de cada uno de los métodos.

En el Capítulo 5, se realiza la aplicación detallada y exhaustiva de cada uno de los conceptos teóricos desarrollados en el tercer Capítulo a un sistema continuo. En este caso el sistema elegido es uno de advección-difusión-reacción (ADR) discretizado espacialmente por el método de líneas, para el que se calculan las actividades locales y globales, el Factor de Homogeneidad y el número teórico mínimo de pasos requeridos por cada tipo de método. Con esto se realiza una conclusión inicial de qué método resulta más conveniente utilizar para este modelo con parámetros particulares, lo que a continuación se corrobora mediante los resultados de simulación del mismo. Luego se comparan dos escenarios con variación en los parámetros (resultando uno más homogéneo y el otro más heterogéneo) y se vuelve a analizar teórica y prácticamente cada uno de ellos.

Luego, se simula el modelo utilizando los distintos algoritmos linealmente implícitos de QSS, para analizar si las nuevas implementaciones ofrecen mejoras frente al método original. Además se compara también con un método clásico de integración numérica basado en discretización temporal.

En el Capítulo 6 se presenta una red neuronal representada mediante un modelo con discontinuidades en el que se vuelven a aplicar los conceptos teóricos relacionados con la actividad. A partir de ello, se calcula el número mínimo teórico de pasos para los distintos enfoques de integración numérica, lo que se compara con resultados simulados.

Por último se simulan distintos escenarios con los métodos QSS, eLIQSS, CheQSS y un método clásico para comparar la eficiencia y errores en cada caso.

Un último ejemplo se presenta en el Capítulo 7, que consiste en un modelo epidemiológico multiagente. Este modelo suma las dificultades de los dos modelos anteriores, ya que combina rigidez (asociada a la dinámica de la infección en cada individuo) con discontinuidades (debida a la interacción de los individuos con el ambiente) y gran escala. Al igual que en los casos anteriores, se analizan en este modelo las características de actividad y se estudia el desempeño de los nuevos métodos numéricos desarrollados.

Para dar cierre a la tesis, se presentan en el Capítulo 8 las conclusiones obtenidas a lo

largo de todo el trabajo y se describen posibles ramas que quedan abiertas para continuar y profundizar los estudios hasta aquí realizados.

Capítulo 2

Preliminares

En este capítulo, presentamos conceptos básicos para comprender el contenido de la Tesis y definiciones previas en las que se basó el trabajo principal de la misma.

2.1. Clasificación de Modelos Matemáticos

Los modelos matemáticos de sistemas dinámicos pueden clasificarse, en términos generales, según la forma en la que las magnitudes evolucionan con el tiempo [70, 69]. Los modelos en los que las variables evolucionan de manera continua en el tiempo se denominan *modelos de tiempo continuo* y típicamente, bajo hipótesis de parámetros concentrados, se describen mediante ecuaciones diferenciales ordinarias. Un ejemplo de los mismos podría ser un modelo del movimiento de una masa puntual en el aire. Cuando las variables sólo pueden cambiar en determinados instantes de tiempo, se habla de *modelos de tiempo discreto*, los que se suelen representar mediante ecuaciones en diferencias. Un modelo epidemiológico que actualiza la cantidad de personas infectadas diariamente podría ser un ejemplo de este tipo de modelos. Finalmente, existen modelos en los que las variables pueden modificarse en cualquier instante de tiempo, pero sólo pueden experimentar un número finito de cambios en intervalos de tiempo finitos. Estos modelos se conocen como *modelos de eventos discretos* y un ejemplo de los mismos puede ser un modelo de cola en un supermercado, en el que el tamaño de la cola varía cada vez que un individuo arriba o parte de la misma (eventos).

Los modelos de tiempo discreto y los modelos de eventos discretos pueden simularse de forma directa. En los primeros, las mismas ecuaciones en diferencias que los constituyen establecen las reglas de actualización del estado en cada instante discreto. Los modelos de eventos discretos, en tanto, pueden formularse utilizando distintos lenguajes de representación, aunque todos ellos pueden expresarse en términos del formalismo DEVS, el cual cuenta con un simulador abstracto genérico [70, 69].

Por el contrario, los modelos de tiempo continuo no pueden simularse en general sin recurrir a algún tipo de aproximación. La forma clásica de simular este tipo de sistemas consiste en

emplear *algoritmos de integración numérica*, que, desde una perspectiva de teoría de sistemas, aproximan el comportamiento continuo mediante modelos en tiempo discreto. La literatura sobre simulación de sistemas continuos incluye diversas familias de algoritmos numéricos especializados para distintos tipos de problemas, los cuales se sustentan en una teoría bien desarrollada sobre la precisión, estabilidad y convergencia de las aproximaciones [32, 64, 20].

Por su lado, como ya mencionamos anteriormente, denominamos sistemas híbridos a los que combinan partes continuas y discretas, las primeras descritas mediante ODEs (o ecuaciones algebraico diferenciales, DAEs, en un caso más general). En muchas aplicaciones, estos modelos son el resultado de la interconexión de una gran cantidad de subsistemas híbridos simples, dando lugar a redes complejas, que como mencionamos en la introducción, debido a la interacción de las dinámicas continua y discretas, la simulación de los mismos suele tener alto costo computacional.

2.2. Métodos Clásicos de Integración Numérica

Las ecuaciones diferenciales con que se representan los sistemas de tiempo continuo modelan la tasa de cambio de sus magnitudes a través de las derivadas respecto del tiempo. Cuando dichas magnitudes presentan una distribución espacial, también aparecen derivadas respecto del espacio, dando lugar a modelos descritos por *ecuaciones diferenciales parciales* (PDE). En cambio, bajo el supuesto de parámetros concentrados, la dependencia espacial continua se ignora y sólo se consideran las derivadas temporales. En este caso, los modelos resultantes son sistemas de *ecuaciones diferenciales ordinarias* (ODEs).

En esta sección, basada principalmente en [20], se revisan los principios de la integración numérica y los algoritmos clásicos utilizados para la simulación de sistemas descritos por ODEs. En lo que respecta a las PDEs, se introduce el procedimiento conocido como *Método de las Líneas* (*Method of Lines*, MOL), que permite transformarlas en sistemas de ODEs para su posterior resolución mediante algoritmos de integración estándar.

2.2.1. Sistemas de tiempo continuo y ODEs

Un sistema de ODEs puede escribirse como

$$\begin{aligned}\dot{x}_1(t) &= f_1(x_1, x_2, \dots, x_N, t) \\ \dot{x}_2(t) &= f_2(x_1, x_2, \dots, x_N, t) \\ &\vdots \\ \dot{x}_N(t) &= f_n(x_1, x_2, \dots, x_N, t),\end{aligned}\tag{2.1}$$

donde $x_i(t)$ son las variables de estado, t el tiempo, $\dot{x}_i$ sus derivadas temporales, y $f_i : \mathbb{R}^N \times \mathbb{R} \rightarrow \mathbb{R}$ las funciones que definen la dinámica. Este sistema se expresa habitualmente en forma compacta como:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t), \quad (2.2)$$

donde $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ es el *vector de estados*.

La simulación de un modelo continuo descrito por ODEs como el de la Ec. (2.2) implica resolver el sistema a partir de una condición inicial $\mathbf{x}(t_0) = \mathbf{x}_0$. Sin embargo, las soluciones analíticas sólo pueden obtenerse en casos muy particulares, por lo que la mayoría de las simulaciones se realizan mediante *aproximaciones numéricas*, obtenidas a través de *algoritmos de integración numérica*, los cuales constituyen la base de la simulación de sistemas continuos.

2.2.2. Métodos de Euler

Dado el sistema de la Ec. (2.2) con estado inicial $\mathbf{x}(t_0) = \mathbf{x}_0$, el método de *Euler* o *Forward Euler* (FE) lo aproxima mediante:

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + h \cdot \mathbf{f}(\mathbf{x}(t_k), t_k), \quad (2.3)$$

donde $h = t_{k+1} - t_k$ es el *paso de integración*.

Un método similar es el *Backward Euler* (BE), definido por:

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + h \cdot \mathbf{f}(\mathbf{x}(t_{k+1}), t_{k+1}). \quad (2.4)$$

El método FE es *explícito*, ya que el estado $\mathbf{x}(t_{k+1})$ se obtiene directamente a partir de $\mathbf{x}(t_k)$. En este tipo de métodos, el paso de integración h debe limitarse para mantener la *estabilidad numérica*.

Considerando una EDO lineal invariante en el tiempo:

$$\dot{\mathbf{x}}(t) = A \cdot \mathbf{x}(t), \quad (2.5)$$

su solución es asintóticamente estable si todos los autovalores de A tienen parte real negativa. La aproximación FE resulta:

$$\mathbf{x}(t_{k+1}) = (I + hA) \cdot \mathbf{x}(t_k),$$

y sólo será estable si los autovalores de $(I + hA)$ se encuentran dentro del círculo unitario, lo que impone restricciones severas sobre h .

En contraste, el método BE es *implícito*, pues requiere resolver una ecuación algebraica para obtener $\mathbf{x}(t_{k+1})$ (por ejemplo, mediante la iteración de Newton). No obstante, posee

mejores propiedades de estabilidad, ya que su aproximación:

$$\mathbf{x}(t_{k+1}) = (I - hA)^{-1} \cdot \mathbf{x}(t_k),$$

preserva la estabilidad numérica para cualquier paso h , siempre que los autovalores de A tengan parte real negativa.

Los métodos de Euler son de *primer orden* de precisión, ya que aproximan el desarrollo de Taylor de la solución hasta el término de h^1 . El error local de truncamiento es proporcional a h^2 , mientras que el error global acumulado crece proporcionalmente a h .

Por lo tanto, para incrementar la precisión en un factor de 10 000, sería necesario reducir el paso h en el mismo factor, lo que implica realizar 10 000 veces más pasos. Los métodos de orden superior, en cambio, permiten alcanzar esa misma precisión con pasos mucho mayores, reduciendo así el costo computacional.

2.2.3. Métodos monopaso

Para obtener aproximaciones de orden superior, es necesario evaluar la función $\mathbf{f}(\mathbf{x}(t), t)$ en más de un punto alrededor de $\mathbf{x}(t_k)$. Los *métodos monopaso*, también conocidos como métodos de *Runge-Kutta* (RK), implementan esta idea.

Por ejemplo, el método de Heun, de segundo orden, se define como:

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(\mathbf{x}(t_k), t_k), \\ \mathbf{k}_2 &= \mathbf{f}(\mathbf{x}(t_k) + h \cdot \mathbf{k}_1, t_k + h), \\ \mathbf{x}(t_{k+1}) &= \mathbf{x}(t_k) + \frac{h}{2} \cdot (\mathbf{k}_1 + \mathbf{k}_2). \end{aligned} \tag{2.6}$$

Este algoritmo duplica aproximadamente el costo por paso respecto de Euler directo, pero permite usar pasos mucho más grandes para obtener errores globales menores, dentro de las restricciones de estabilidad propias de los métodos explícitos.

Entre los métodos RK, uno de los más utilizados es el de cuarto orden (RK4) formulado originalmente por Runge y Kutta a fines del siglo XIX, que realiza cuatro evaluaciones de $\mathbf{f}(\mathbf{x}, t)$ por paso. En la mayoría de las aplicaciones de ingeniería, donde se requiere una precisión del orden de 0.1%, el RK4 ofrece un equilibrio adecuado entre exactitud y costo computacional.

Los métodos explícitos de un paso, como Heun o RK4, comparten las limitaciones de estabilidad del método de Euler explícito. No obstante, existen variantes implícitas, como por ejemplo la *regla trapezoidal* (TR):

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \frac{h}{2} [\mathbf{f}(\mathbf{x}(t_k), t_k) + \mathbf{f}(\mathbf{x}(t_{k+1}), t_{k+1})],$$

que es de segundo orden y mantiene la estabilidad numérica en sistemas lineales para cualquier

paso h .

2.2.4. Métodos multipaso

Mientras los métodos monopaso incrementan el orden mediante evaluaciones adicionales de $\mathbf{f}(\mathbf{x}, t)$, los *métodos multipaso* lo logran utilizando información de pasos anteriores, reduciendo así el costo computacional.

Los métodos de Adams–Bashforth (explícitos) y los de *Backward Difference Formula* (BDF, implícitos) son los más representativos. Por ejemplo, la fórmula BDF de tercer orden es:

$$\mathbf{x}(t_{k+1}) = \frac{18}{11}\mathbf{x}(t_k) - \frac{9}{11}\mathbf{x}(t_{k-1}) + \frac{2}{11}\mathbf{x}(t_{k-2}) + \frac{6}{11}h \cdot \mathbf{f}(\mathbf{x}(t_{k+1}), t_{k+1}). \quad (2.7)$$

Los métodos BDF presentan excelentes propiedades de estabilidad y alcanzan altos órdenes de precisión con costos por paso similares al de Euler implícito.

2.2.5. PDEs y el Método de Líneas

La aproximación numérica de PDEs es más compleja, ya que requiere discretizar tanto el tiempo como el espacio. Existen diferentes enfoques para lograrlo; entre ellos, los métodos de diferencias finitas constituyen un marco general ampliamente utilizado [36]. En esta tesis se emplea el *Método de las Líneas* (MOL) [49], que consiste en discretizar primero las derivadas espaciales mediante aproximaciones en diferencias, transformando así la PDE en un sistema de ecuaciones diferenciales ordinarias en el tiempo.

Por ejemplo, dada la ecuación unidimensional de *advección–difusión–reacción* (ADR):

$$\frac{\partial u(t, z)}{\partial t} = -a \frac{\partial u(t, z)}{\partial z} + d \frac{\partial^2 u(t, z)}{\partial z^2} + r \cdot g(u(t, z)), \quad (2.8)$$

con parámetros a, d, r , y una función $g(u)$ conocida que, el dominio espacial se discretiza con una malla de paso Δz y puntos z_i . Las derivadas espaciales se aproximan mediante diferencias finitas:¹

$$\begin{aligned} \frac{\partial u(t, z_i)}{\partial z} &\approx \frac{u_i(t) - u_{i-1}(t)}{\Delta z}, \\ \frac{\partial^2 u(t, z_i)}{\partial z^2} &\approx \frac{u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)}{\Delta z^2}, \end{aligned} \quad (2.9)$$

donde $u_i(t) \triangleq u(z_i, t)$. Sustituyendo en la Ec. (2.8) se obtiene un conjunto de ODEs:

$$\dot{u}_i(t) = -a \frac{u_i(t) - u_{i-1}(t)}{\Delta z} + d \frac{u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)}{\Delta z^2} + r \cdot g(u_i(t)), \quad (2.10)$$

para $i = 2, \dots, N-1$, donde N es el número de puntos de la malla espacial. Las ecuaciones para $\dot{u}_1$ y $\dot{u}_N$ se pueden obtener de manera similar en función de las condiciones de borde del

¹Válido para el caso en que $a \geq 0$.

problema.

En la práctica, el número de ecuaciones resultantes puede ser muy grande, dando lugar a sistemas de gran escala.

2.2.6. Solvers de ODEs y herramientas de software

Los métodos de integración numérica suelen implementarse con algoritmos de paso variable, que ajustan automáticamente el tamaño de h para mantener una precisión prescrita. Esta adaptación se realiza estimando el error mediante la comparación de dos métodos de distinto orden.

Uno de los solvers más utilizados es el *Dormand–Prince de quinto orden* (DOPRI5), basado en dos algoritmos de Runge–Kutta de orden 4 y 5. Aunque eficiente, su carácter explícito limita el tamaño máximo del paso debido a los problemas ya mencionados de estabilidad numérica.

Para sistemas más complejos, los solvers basados en BDF, como *DASSL* y *CVODE*, ofrecen mejores propiedades de estabilidad y control adaptativo de paso y orden. Estos algoritmos, junto con DOPRI5, están disponibles como código abierto e integrados en las principales herramientas modernas de simulación.

Actualmente, el lenguaje orientado a objetos *Modelica* [29] se ha consolidado como un estándar para la representación de sistemas continuos multidominio. Herramientas como *OpenModelica*, *Dymola* y *Wolfram SystemModeler* incorporan internamente solvers como DOPRI5, DASSL o CVODE, permitiendo la simulación precisa y eficiente de modelos dinámicos continuos y mixtos.

2.2.7. Limitaciones de los Métodos Clásicos de Integración Numérica

En ciertos sistemas en tiempo continuo relativamente simples, la mayoría de los algoritmos numéricos disponibles permiten obtener resultados adecuados con costos computacionales razonables. Sin embargo, existen ODEs cuya simulación presenta dificultades adicionales y requiere un tratamiento especial. A continuación, se describen algunos de los tipos de modelos que plantean los principales desafíos desde el punto de vista numérico.

Sistemas Rígidos

En el contexto de la simulación de sistemas en tiempo continuo, se denomina *sistema rígido* (*stiff system*) a aquel que contiene simultáneamente dinámicas rápidas y lentas. Este tipo de comportamiento es muy común en aplicaciones prácticas. Los modelos *multidominio* suelen ser rígidos debido a que los distintos dominios físicos presentan habitualmente constantes de tiempo muy diferentes, que pueden variar desde microsegundos en el dominio eléctrico hasta horas en el dominio térmico. Asimismo, los sistemas que presentan un amortiguamiento elevado también pueden ser rígidos, incluso si pertenecen a un único dominio. Además, en

el problema de ADR discretizado mediante el Método de Líneas en la Ecuación (2.10), las reacciones químicas pueden evolucionar en escalas de tiempo mucho menores que las del transporte por advección-difusión, lo que constituye otra fuente típica de rigidez.

En este tipo de sistemas, la existencia de dinámicas rápidas obliga a utilizar pasos de integración muy pequeños para mantener la estabilidad numérica, mientras que la presencia de dinámicas lentas exige extender la simulación durante largos intervalos de tiempo para observar la evolución de las magnitudes lentamente variables. Como consecuencia, la simulación requiere ejecutar una cantidad extremadamente grande de pasos de integración, lo cual incrementa significativamente el costo computacional.

Una posible estrategia para mitigar este problema consiste en aumentar el tamaño del paso de integración una vez que las componentes rápidas del sistema han decaído. Sin embargo, al incrementar el paso se podría perder precisión en la solución y sólo sería posible si el algoritmo mantiene la estabilidad numérica para cualquier tamaño de paso h . Esta última propiedad la cumplen únicamente ciertos métodos implícitos, como el método de Backward Euler o las familias de métodos BDF.

Por lo tanto, los sistemas rígidos deben simularse utilizando esquemas implícitos que permitan incrementar el paso h sin introducir inestabilidad numérica. No obstante, como se mencionó previamente, este tipo de métodos presenta mayores costos computacionales, dado que requieren resolver iterativamente sistemas de ecuaciones mediante procedimientos como la iteración de Newton, lo que implica calcular la matriz Jacobiana $\partial \mathbf{f} / \partial \mathbf{x}$ e invertir una matriz en cada paso de simulación.

Sistemas de Gran Escala

Se denominan *sistemas de gran escala* a aquellos cuya variable de estado $\mathbf{x}(t)$ posee una dimensión elevada. Esta característica, aumenta aun más la demanda computacional en simulaciones con métodos de integración clásicos de problemas rígidos, ya que las iteraciones de Newton sobre sistemas de gran tamaño requieren el cálculo e inversión de matrices extensas, lo que incrementa notablemente el tiempo de simulación.

Estas dificultades pueden mitigarse mediante el uso de herramientas y algoritmos especializados para el tratamiento de matrices dispersas o sparse, que aprovechan el hecho de que, en la mayoría de los casos, los sistemas de gran escala presentan una estructura rala. Por ejemplo, el solver *CVODE BDF* permite trabajar con matrices dispersas y es capaz de simular de manera eficiente sistemas rígidos de gran dimensión.

Con frecuencia, los modelos de gran escala surgen como resultado de la discretización espacial de ecuaciones en derivadas parciales mediante técnicas como el Método de Líneas. Si bien estos enfoques cuentan con estrategias propias de reducción de orden y manejo del número de grados de libertad, los modelos resultantes suelen involucrar un gran número de

variables de estado. Por ejemplo, el modelo de la Ecuación (2.10), discretizado con un paso espacial pequeño Δz , conduce a un sistema de EDOs de gran escala, cuya simulación eficiente continúa dependiendo en gran medida del algoritmo de integración temporal utilizado.

Sistemas con Discontinuidades

La mayoría de los algoritmos de integración numérica asumen que el lado derecho de la ecuación diferencial, $\mathbf{f}(\mathbf{x}, t)$, es una función suave. Sin embargo, existen numerosos modelos de interés práctico que no cumplen esta hipótesis y presentan *discontinuidades*, que consisten en saltos finitos en $\mathbf{f}(\mathbf{x}, t)$ o en el estado $\mathbf{x}(t)$ en determinados instantes t^* , lo que conduce a dinámicas gobernadas por eventos.

Los modelos discontinuos son especialmente comunes en electrónica de potencia, donde los conmutadores (*switches*) alternan entre los estados *encendido* y *apagado* a frecuencias muy elevadas. Otro ejemplo representativo son las redes neuronales pulsantes (*spiking neural networks*), en las cuales las variables del sistema presentan saltos abruptos en el momento del disparo (*spike*).

Cuando ocurre una discontinuidad en medio de un paso de integración, entre los instantes t_k y t_{k+1} , el error numérico puede volverse muy grande e inaceptable ya que se viola la hipótesis de continuidad en la que se basan los esquemas numéricos. Para evitarlo, los algoritmos de integración deben incorporar rutinas de detección y manejo de discontinuidades.

La *detección de discontinuidades* (también denominada *detección de cruces por cero*) se realiza observando el signo de una *función de cruce por cero* provista por el usuario. Cuando el signo de esta función cambia entre un paso y el siguiente, el algoritmo interrumpe la simulación y ejecuta un procedimiento iterativo para determinar el instante exacto t^* entre t_k y t_{k+1} en el cual la función se anula. La simulación se avanza entonces hasta el tiempo t^* , donde ocurre la discontinuidad, y se reinicia la integración a partir de dicho punto.

Todo este procedimiento, conocido como *detección y manejo de discontinuidades*, añade una carga computacional considerable debido a las evaluaciones adicionales de funciones, los procedimientos iterativos de localización del evento y la posterior reinicialización del esquema de integración. En presencia de discontinuidades frecuentes, el costo total de simulación puede volverse inaceptable. Este problema se ve agravado en los sistemas de gran escala, como los que aparecen en las redes neuronales pulsantes, donde la combinación de discontinuidades y alta dimensionalidad hace que la simulación sea particularmente costosa desde el punto de vista computacional.

2.3. Métodos de Quantized State Systems

Los métodos QSS (*Quantized State System*) constituyen una familia de algoritmos de integración numérica que reemplazan la discretización temporal utilizada en los algoritmos clásicos de integración por la *cuantificación de las variables de estado* [19, 17].

2.3.1. Origen de los métodos QSS basados en cuantificación de estados

Hace algunas décadas, Bernard Zeigler comenzó a proponer distintas ideas para representar sistemas de tiempo continuo utilizando el formalismo DEVS [66, 71], lo que finalmente lo condujo a la definición de los *Sistemas Cuantificados* [68, 67]. En dichos trabajos, Zeigler mostró que un sistema continuo, cuyas variables de entrada y salida son cuantificadas, puede describirse mediante un modelo DEVS equivalente capaz de reproducir su comportamiento. Estas ideas fueron posteriormente ampliadas y exploradas por otros autores, entre ellos Giambiasi [31, 30] y Barros [2, 3].

Todos estos enfoques se centraron en el problema de representar el comportamiento de un sistema de tiempo continuo a través de un sistema de eventos discretos —más precisamente, un modelo DEVS—. Sin embargo, ninguno de ellos estableció un procedimiento sistemático y generalizado para obtener soluciones aproximadas de las ecuaciones diferenciales que gobiernan dichos sistemas, con excepción de la propuesta de Zeigler [67] de representar un *integrador cuantificado* mediante un modelo DEVS, y luego acoplar varios integradores cuantificados como aproximación de un sistema continuo completo.

No obstante, la idea de acoplar modelos DEVS de integradores cuantificados presentaba una limitación importante: este tipo de acoplamiento solía generar modelos DEVS *ilegítimos*, en el sentido de que la aproximación resultante podía producir un número infinito de eventos en un intervalo finito de tiempo. Este problema, identificado en [47], fue resuelto mediante la introducción de histéresis en la función de cuantificación, lo que dio origen a la definición del primer método de *Quantized State Systems*.

2.3.2. Método de QSS1

Dado el sistema de la Ecuación (2.2), el método de primer orden QSS1 [41] resuelve una ODE aproximada llamada Sistema de Estados Cuantificados (Quantized State System):

$$\dot{\mathbf{x}}(t) = \mathbf{f}^q(\mathbf{q}(t), t) \quad (2.11)$$

donde $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ es el vector de estados, $\mathbf{q} = [q_1, q_2, \dots, q_N]^T$ es el vector de *estados cuantificados* y $\mathbf{f}^q(\mathbf{q}, t)$ es una aproximación constante a tramos de $\mathbf{f}(\mathbf{x}, t)$ con respecto a t .

Cada estado cuantificado $q_i(t)$ está relacionado con su correspondiente estado $x_i(t)$ mediante una *función de cuantificación con histéresis*. Se dice que $x_i(t)$ y $q_i(t)$ están relacionados por

una función de cuantificación con histéresis con *quantum* ΔQ_i si $q_i(t_0) = x_i(t_0)$ y satisfacen:

$$q_i(t) = \begin{cases} q_i(t_k) & \text{si } |x_i(t) - q_i(t_k)| < \Delta Q_i, \\ x_i(t) & \text{en caso contrario.} \end{cases}$$

para $t_k < t \leq t_{k+1}$, donde t_{k+1} es el primer instante posterior a t_k tal que $|x_i(t) - q_i(t_k)| = \Delta Q_i$.

La Figura 2.1 ilustra las trayectorias de un estado y su correspondiente estado cuantificado.

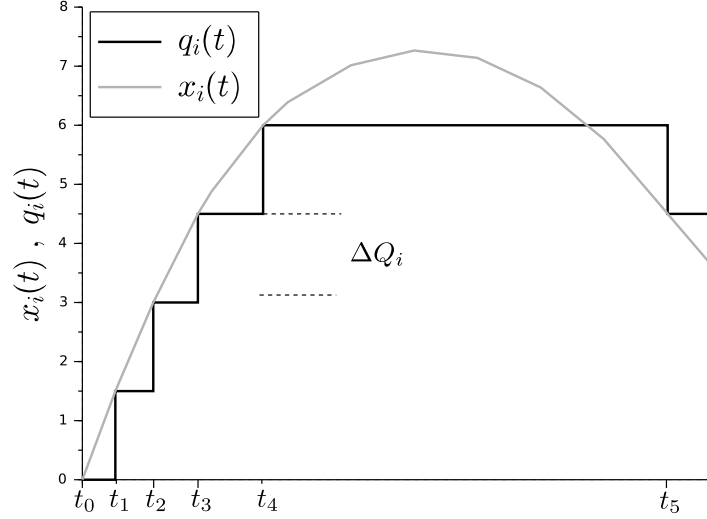


Figura 2.1: Trayectorias típicas de un estado y su estado cuantificado en QSS1

Dado que los estados cuantificados siguen trayectorias seccionalmente constantes y que la función $\mathbf{f}^q$ es seccionalmente constante respecto de t , resulta que las derivadas de los estados $\dot{x}_i(t)$, según la Ec.(2.11) siguen también trayectorias constantes a tramos. Luego, los estados $x_i(t)$ siguen trayectorias seccionalmente lineales por lo que la resolución analítica de la Ec.(2.11) es muy simple y puede realizarse por un procedimiento como el que se esboza en el Algoritmo 2.1.

Algoritmo 2.1: QSS1.

```

1 // t es el tiempo de simulación
2 // t_j es el tiempo del siguiente cambio en q_j
3 // t_j^x es el tiempo del último cambio en x_j
4 while (t < t_f) // simular hasta el tiempo final t_f
5   t = min(t_j) // avanzar tiempo de simulación
6   i = argmin(t_j) // el i-ésimo estado cuantificado cambia primero
7   e = t - t_i^x // tiempo transcurrido desde la última actualización de x_i
8   x_i = x_i + \dot{x}_i \cdot e // actualizar el valor del i-ésimo estado
9   q_i = x_i // actualizar el i-ésimo estado cuantificado
10  t_i = min(\tau > t) subject to |q_i - x_i(\tau)| = \Delta Q_i // calcular próximo cambio del i-ésimo estado cuantificado
11  for each j \in [1, n] such that \dot{x}_j depends on q_i
12    e = t - t_j^x // tiempo transcurrido desde la última actualización de x_j

```

```

13      $x_j = x_j + \dot{x}_j \cdot e$  // actualizar el valor del j-ésimo estado
14     if  $j \neq i$  then  $t_j^x = t$  // última actualización de xj
15      $\dot{x}_j = f_j(\mathbf{q}, t)$  // recalculer la derivada del j-ésimo estado
16      $t_j = \min(\tau > t)$  subject to  $|q_j - x_j(\tau)| = \Delta Q_j$  // recalculer el tiempo de
        cambio de del j-ésimo estado cuantificado
17     end for
18      $t_i^x = t$  // última actualización de xi
19 end while

```

2.3.3. Métodos QSS de Orden Superior

QSS2 es un método de segundo orden que comparte la misma definición conceptual que QSS1, excepto en que los componentes de $\mathbf{q}(t)$ se calculan de forma que sigan trayectorias lineales por tramos (en lugar de trayectorias constantes por tramos, como en QSS1). Como resultado, las derivadas de los estados $\dot{\mathbf{x}}(t)$ se representan mediante trayectorias lineales por tramos, lo que hace que los estados $\mathbf{x}(t)$ sigan trayectorias parabólicas por tramos. Formalmente, se dice que las trayectorias $x_i(t)$ y $q_i(t)$ están relacionadas por una función de cuantificación de primer orden si $q_i(t_0) = x_i(t_0)$ y si satisfacen:

$$q_i(t) = \begin{cases} q_i(t_k) + m_k(t - t_k) & \text{si } |x_i(t) - q_i(t_k)| < \Delta Q_i, \\ x_i(t) & \text{en caso contrario.} \end{cases}$$

con $t_k \leq t \leq t_{k+1}$, y la secuencia $t_0, \dots, t_k, \dots$ definida de modo que t_{k+1} sea el mínimo $t \geq t_k$ tal que:

$$|x_i(t_k) + m_k(t - t_k) - x_i(t)| = \Delta Q_i,$$

y las pendientes:

$$m_0 = 0, \quad m_k = \dot{x}_i(t_k^-), \quad k = 1, 2, \dots$$

El uso de una aproximación lineal por tramos reduce el número de pasos, ya que las trayectorias del estado y del estado cuantificado necesitan más tiempo para diferir en una cantidad ΔQ_i . Puede demostrarse fácilmente que el número de pasos en QSS2 es inversamente proporcional a la raíz cuadrada del quantum. Esto implica que, si el quantum se reduce 100 veces, el número de pasos sólo aumenta 10 veces.

El algoritmo de simulación QSS2 es muy similar al de QSS, salvo que realiza algunos cálculos adicionales para determinar y considerar las pendientes de las trayectorias. Por lo tanto, la ventaja de QSS2 frente a QSS es que permite utilizar valores de quantum más pequeños (obteniendo así errores menores) sin incrementar significativamente el número de pasos. En la Figura 2.2 se pueden observar las trayectorias típicas del estado y el estado cuantificado del algoritmo de QSS2

Esta idea fue extendida de manera análoga para desarrollar el método QSS3 [44], un método

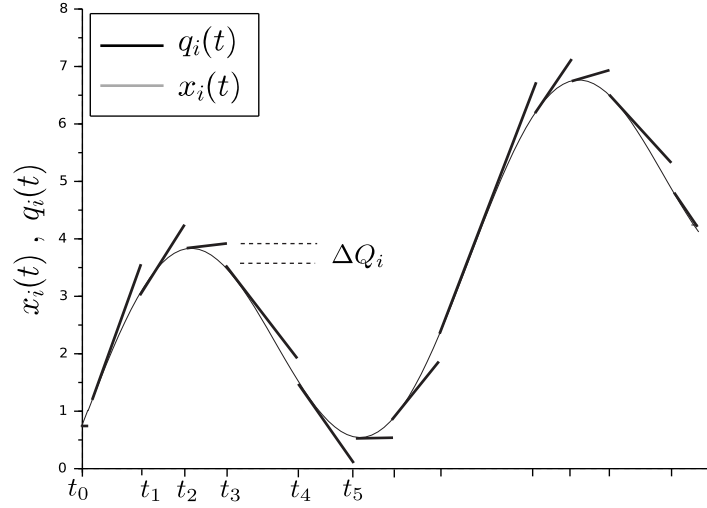


Figura 2.2: Trayectorias típicas de un estado y su estado cuantificado en QSS2

QSS de tercer orden de precisión en el cual los estados cuantificados $q_i(t)$ siguen trayectorias parabólicas por tramos, mientras que los estados siguen trayectorias cúbicas por tramos. En QSS3 el número de pasos es inversamente proporcional a la raíz cúbica del quantum, lo que permite obtener resultados más precisos con menos pasos que en QSS2. En la Figura 2.3 se pueden observar las trayectorias típicas del estado y el estado cuantificado del algoritmo de QSS3.

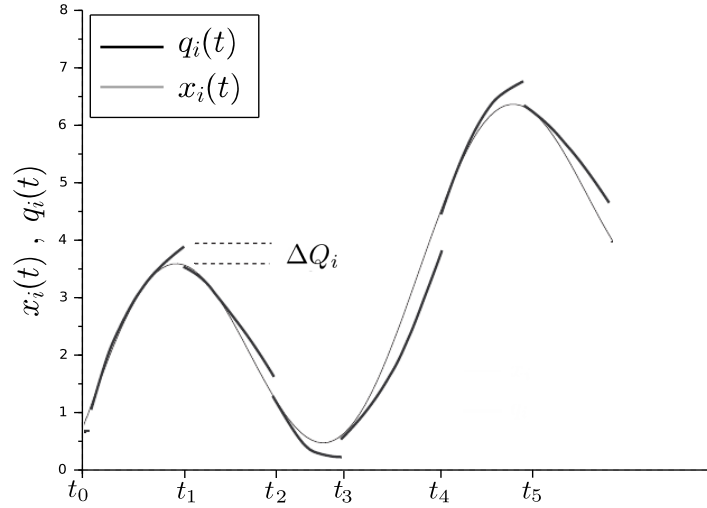


Figura 2.3: Trayectorias típicas de un estado y su estado cuantificado en QSS3

2.3.4. Ventajas Prácticas de QSS

En la práctica, los métodos QSS pueden ofrecer ciertas ventajas frente a los algoritmos tradicionales basados en la discretización temporal.

Una característica clave es la naturaleza localizada de las actualizaciones de estado: una variable de estado x_i se actualiza únicamente cuando la desviación respecto de su valor cuan-

tificado q_i excede el umbral ΔQ_i . Estas actualizaciones afectan únicamente a q_i y a aquellas derivadas de estado $\dot{x}_j = f_j^q(\mathbf{q}, t)$ para las cuales f_j^q depende explícitamente de q_i . De esta forma, los cálculos se ejecutan de manera asíncrona, sólo cuando y donde ocurren cambios significativos en el sistema.

Esta naturaleza local del cálculo en QSS conduce a ventajas prácticas, particularmente en sistemas grandes y dispersos donde los cambios dinámicos se concentran en un subconjunto limitado de variables. En tales escenarios, como los encontrados, por ejemplo, en modelos de advección–difusión–reacción que, tras una discretización espacial adecuada, presentan frentes de onda pronunciados, los métodos QSS restringen naturalmente los cálculos a las regiones donde existe actividad, reduciendo así el costo computacional total.

Además, QSS resulta especialmente adecuado para sistemas con dinámicas discontinuas. Gracias a la estructura polinómica por tramos de las trayectorias de estado, las discontinuidades pueden detectarse directamente y manejarse sin requerir procedimientos de reinicialización global.

2.3.5. Propiedades Teóricas de QSS

La Ecuación (2.11) puede reescribirse como

$$\dot{\mathbf{x}}(t) = \mathbf{f}^q(\mathbf{x}(t) + \Delta\mathbf{x}(t), t) \quad (2.12)$$

donde $\Delta\mathbf{x}(t) \triangleq \mathbf{q}(t) - \mathbf{x}(t)$ es un término de perturbación acotado por el vector de quantum $\Delta\mathbf{Q} = [\Delta Q_1, \dots, \Delta Q_n]^T$.

Por lo tanto, el uso del estado cuantificado en lugar del estado real en la aproximación es equivalente a añadir una perturbación $\Delta\mathbf{x}(t)$. Esta perturbación está acotada componente a componente por el quantum, dado que se cumple $|\Delta x_i(t)| \leq \Delta Q_i$, ya que los estados cuantificados $q_i(t)$ y los estados reales $x_i(t)$ nunca difieren entre sí en más de dicho quantum ΔQ_i .

A partir de esta observación, se establecieron las siguientes propiedades [47, 20]:

- **Convergencia:** Suponiendo que $\mathbf{f}$ satisface condiciones de Lipschitz, las soluciones de la aproximación QSS dada por la Ec. (2.11) convergen a las soluciones de la ODE original de la Ec. (2.2) cuando el quantum $\Delta\mathbf{Q} \rightarrow 0$ y $|\mathbf{f}_q(x, t) - \mathbf{f}(x, t)| \rightarrow 0$.
- **Cota Global de Error:** En sistemas lineales invariantes en el tiempo (LTI), el error global está acotado por una cantidad que depende linealmente del quantum $\Delta\mathbf{Q}$.

Una conclusión teórica importante es que QSS preserva la estabilidad numérica práctica independientemente del valor del quantum $\Delta\mathbf{Q}$ es decir que aunque el quantum aumente, esto no induce inestabilidades numéricas artificiales, sino que afecta únicamente la precisión

de la aproximación. Ésta es una característica distintiva de QSS frente a los métodos explícitos clásicos, que sólo mantienen la estabilidad para valores pequeños del paso h . Sin embargo, esto no implica que QSS sea un algoritmo adecuado para simular sistemas rígidos (*stiff*). El número de pasos que realiza QSS al simular un sistema rígido puede resultar muy grande debido a la aparición de oscilaciones espurias.

Además, se han obtenido resultados recientes que establecen propiedades de convergencia de los métodos QSS cuando se aplican a ecuaciones diferenciales estocásticas [57].

2.3.6. Métodos de QSS Linealmente Implícitos

La aparición de oscilaciones espurias en los métodos explícitos de QSS al simular sistemas rígidos es análoga a lo que ocurre con los métodos clásicos explícitos de paso variable. Como consecuencia el número de pasos que realizan puede volverse inaceptablemente grande lo que provoca un alto costo computacional [20].

Con el fin de abordar dinámicas rígidas, la familia QSS fue extendida mediante el desarrollo de los algoritmos *Linearly Implicit QSS* (LIQSS) [51]. La idea fundamental de LIQSS se inspira en técnicas de integración implícita, donde la derivada del estado se evalúa en instantes futuros. Sin embargo, a diferencia de los métodos implícitos tradicionales, LIQSS evita resolver ecuaciones no lineales o invertir matrices. En cambio, aprovecha una propiedad característica de los métodos QSS: el valor futuro del estado cuantificado está completamente determinado en cada paso, ya que será $q_i(t) = q_i(t^-) \pm \Delta Q_i$.

Cuando la aproximación del método predice que la derivada futura será positiva, el estado cuantificado se fija como $q_i(t) = x_i(t) + \Delta Q_i$. En caso contrario, si se predice un valor negativo, se fija como $q_i(t) = x_i(t) - \Delta Q_i$. La Figura 2.4 muestra la trayectoria típica del estado y del estado cuantificado del algoritmo LIQSS1.

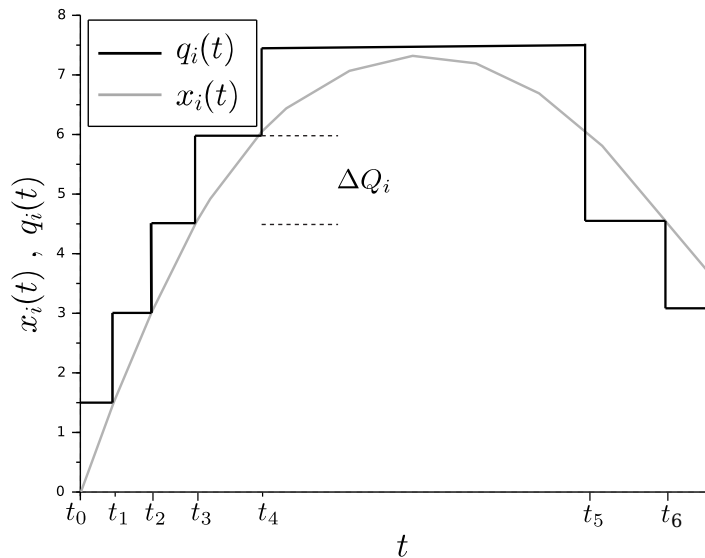


Figura 2.4: Trayectorias típicas de estado y estado cuantificado en LIQSS1

Puede suceder que elegir el incremento positivo $q_i(t) = x_i(t) + \Delta Q_i$ genere una derivada negativa del estado, es decir, $\dot{x}_i(t) < 0$, mientras que elegir el incremento negativo $q_i(t) = x_i(t) - \Delta Q_i$ dé lugar a una derivada positiva, $\dot{x}_i(t) > 0$. En tal caso, debe existir necesariamente un valor intermedio $q_i^* \in (x_i(t) - \Delta Q_i, x_i(t) + \Delta Q_i)$ para el cual la derivada sea nula, $\dot{x}_i(t) = 0$. Este valor se selecciona entonces como el estado cuantificado: $q_i(t) = q_i^*$. En esta situación denominada *equilibrio*, el algoritmo, en principio, ya no necesita realizar más pasos mientras se mantenga dicha condición.

La predicción del signo de la derivada y el cálculo de q_i^* se obtienen mediante una aproximación lineal de la dinámica dada por la ecuación (2.11). Esta aproximación toma la forma:

$$\dot{x}_i(t) = f_i^q(\mathbf{q}(t), t) = a_i(t_j^i) q_i(t) + u_i(t), \quad (2.13)$$

donde t_j^i es el instante del j -ésimo cambio en el estado cuantificado $q_i(t)$, es decir, $t_j^i \leq t < t_{j+1}^i$, y

$$a_i(t_j^i) = \left. \frac{\partial f_i}{\partial x_i} \right|_{\mathbf{q}(t_j^i)}, \quad u_i(t) = f_i^q(\mathbf{q}(t), t) - a_i(t_j^i) q_i(t). \quad (2.14)$$

Encontrar el valor de equilibrio q_i^* es la ventaja principal de LIQSS1 ya que en dicha situación un método como QSS1 provocaría oscilaciones (que en un caso rígido podrían ser muy rápidas) en torno a dicho valor.

El Algoritmo 2.2 esboza el método de LIQSS1. Como puede verse, es muy similar al Algoritmo 2.1 excepto por el cálculo de los coeficientes de la aproximación lineal y su uso en la determinación del valor de $q_i(t)$. Otro detalle es que al haber un cambio en la dirección de $\dot{x}_i$, se permite que el estado se aleje hasta $2\Delta Q_i$ de su valor cuantificado q_i (esto está ilustrado en la Fig.2.4).

Algoritmo 2.2: LIQSS1.

```

1 // t es el tiempo de simulación
2 // t_j es el tiempo del siguiente cambio en q_j
3 // t_j^x es el tiempo del último cambio en x_j
4 // a_i y u_i son los coeficientes de la aproximación lineal de la Ec.(2.14)
5 while (t < t_f) // simular hasta el tiempo final t_f
6     t = min(t_j) // avanzar el tiempo de simulación
7     i = argmin(t_j) // el i-ésimo estado cuantificado cambia primero
8     e = t - t_i^x // tiempo transcurrido desde la última actualización de x_i
9     x_i = x_i + \dot{x}_i \cdot e // actualizar el valor del i-ésimo estado
10    q_i^- = q_i // guardar el valor previo de q_i
11    \dot{x}_i^- = \dot{x}_i // guardar el valor previo de dxi/dt
12    \dot{x}_i^+ = a_i \cdot (x_i + sign(\dot{x}_i) \cdot \Delta Q_i) + u_i // estimación del próximo valor de la derivada
        del estado
13    if (\dot{x}_i^- \cdot \dot{x}_i^+ > 0) // la derivada del estado mantiene su signo
14        q_i = x_i + sign(\dot{x}_i) \cdot \Delta Q_i

```

```

15  else // el estado cambia su dirección
16     $q_i = -u_{i,i}/a_i$  // elegir  $q_i$  tal que  $dx_i/dt = 0$ 
17  end if
18   $t_i = \min(\tau > t)$  subject to  $x_i(\tau) = q_i$  // calcular el próximo instante de
    cambio del i-ésimo estado cuantificado
19  for each  $j \in [1, n]$  such that  $\dot{x}_j$  depends on  $q_i$ 
20     $e = t - t_j^x$  // tiempo transcurrido desde la última actualización de  $x_j$ 
21     $x_j = x_j + \dot{x}_j \cdot e$  // actualizar el valor del j-ésimo estado
22    if  $j \neq i$  then  $t_j^x = t$  // última actualización de  $x_j$ 
23     $\dot{x}_j = f_j(\mathbf{q}, t)$  // recalcular la derivada del j-ésimo estado
24     $t_j = \min(\tau > t)$  subject to  $x_j(\tau) = q_j$  or  $|q_j - x_j(\tau)| = 2\Delta Q_j$  // recalcular el pr
      óximo cambio del j-ésimo estado cuantificado
25  end for
26  // actualizar los coeficientes de la aproximación lineal
27   $a_i = (\dot{x}_i - \dot{x}_i^-) / (q_i - q_i^-)$  // estimación de entrada de la diagonal Jacobiana
28   $u_i = \dot{x}_i - a_i \cdot q_i$  // coeficientes afines
29   $t_i^x = t$  // última actualización de  $x_i$ 
30 end while

```

Métodos LIQSS de orden superior

Las versiones de orden superior, LIQSS2 y LIQSS3, extienden estas ideas utilizando trayectorias cuantificadas lineales y cuadráticas por partes, respectivamente. Esto permite combinar las ventajas de los métodos QSS de mayor orden con una mejor eficiencia al tratar problemas rígidos.

Cabe mencionar que la definición original de LIQSS2 presentada en [51] calculaba el estado cuantificado de modo que cada tramo lineal comenzara en un valor que verificara $q_i(t_j) = x_i(t_j) \pm \Delta Q_i$ y además coincidiera en la derivada inicial, es decir, $\dot{q}_i(t_j) = \dot{x}_i(t_j)$. De este modo, $q_i(t)$ disponía de un valor futuro de x_i pero no necesariamente de la pendiente futura. Este enfoque fue posteriormente mejorado en [22], donde se propone calcular el estado cuantificado de manera tal que se verifique $\dot{q}_i(t_j + h) = \dot{x}_i(t_j + h)$, siendo $t_j + h$ el instante del siguiente cambio en $q_i(t)$. Una idea similar se menciona para el caso de LIQSS3, aunque sin implementación práctica.

Las Figuras 2.5 y 2.6 muestran las trayectorias típicas del estado y del estado cuantificado de los algoritmos LIQSS2 (según la modificación de [22]) y LIQSS3 (según la definición original de [51]). En los tres casos (LIQSS 1, 2 y 3), las trayectorias cuantificadas se calculan de modo que se asemejan a las de los métodos QSS, pero desplazadas hacia atrás en el tiempo.

El método de LIQSS2, al igual que el de QSS2, calcula no sólo el estado cuantificado q_i sino también su derivada $\dot{q}_i$. Para esto, la aproximación lineal de la Ec.(2.14) también calcula la derivada $\dot{u}_i(t)$. De manera similar, en LIQSS3 se utiliza además la derivada segunda $\ddot{u}_i(t)$.

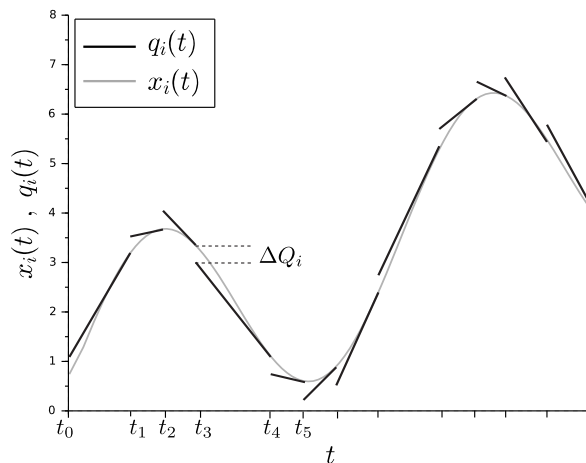


Figura 2.5: Trayectorias del estado y del estado cuantificado en LIQSS2.

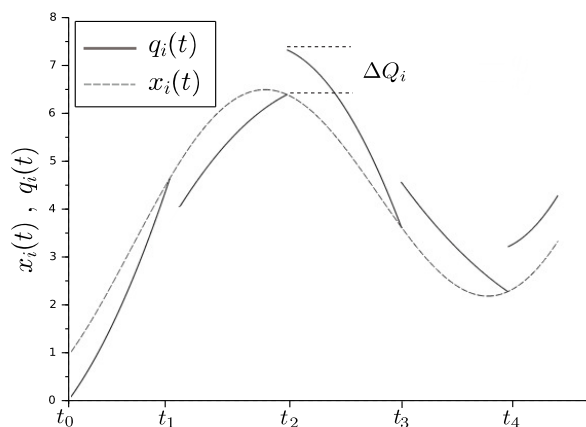


Figura 2.6: Trayectorias del estado y del estado cuantificado en LIQSS3.

Propiedades teóricas de los métodos LIQSS

En la formulación original de los métodos LIQSS, el estado cuantificado puede diferir hasta en $2\Delta Q_i$ del estado $x_i(t)$ cuando la trayectoria del estado cambia de dirección y comienza a alejarse del valor cuantificado. Este hecho implica que la perturbación inducida por la diferencia $q_i(t) - x_i(t)$ puede duplicar la correspondiente a los métodos QSS clásicos. Como consecuencia, los acotamientos teóricos del error en LIQSS son también, en el peor caso, aproximadamente el doble que los de los métodos QSS.

Limitaciones de LIQSS en la simulación de sistemas rígidos

En lo que respecta a los sistemas rígidos, los métodos LIQSS resultan adecuados cuando la rigidez se origina en la presencia de valores grandes en la diagonal principal de la matriz Jacobiana. En tales situaciones, el uso de métodos QSS explícitos puede provocar que una variación en un estado cuantificado genere un cambio significativo y de signo opuesto en su derivada. Como consecuencia, la diferencia entre q_i y x_i alcanza el valor del quantum en intervalos de tiempo muy cortos, dando lugar a oscilaciones rápidas. Este fenómeno desaparece

en los métodos LIQSS, ya que éstos calculan q_i como un valor futuro de x_i , lo que lleva a encontrar un valor de equilibrio q_i^* .

Sin embargo, la rigidez no siempre se debe exclusivamente a la presencia de valores grandes en la diagonal principal; en muchos casos también intervienen términos significativos fuera de ella. En estas situaciones, la simulación mediante LIQSS puede presentar oscilaciones entre pares de estados o incluso entre múltiples variables acopladas. Esta limitación fue parcialmente superada en [22], donde se introdujo un mecanismo para detectar oscilaciones espurias entre pares de variables y actualizar simultáneamente ambos estados cuantificados mediante un esquema linealmente implícito. Dichas ideas fueron posteriormente refinadas en [25].

Entre las aplicaciones donde los métodos LIQSS son adecuados podemos mencionar a la mayoría de los circuitos de electrónica de potencia [52] y las PDEs de Advección-Difusión-Reacción discretizadas mediante el Método de las Líneas (como en la Ec. (2.10)), siempre que el término de difusión sea pequeño en comparación con los de advección y reacción [4]. Un caso donde la rigidez no se debe a grandes términos en la diagonal de la matriz Jacobiana, sino a la estructura general de la misma, ocurre, por ejemplo, en ecuaciones dominadas por la difusión [20]. En estos casos, los métodos de LIQSS no terminan de resolver el problema de la aparición de oscilaciones espurias.

2.3.7. Control de Error Relativo

El uso de un quantum constante es equivalente a implementar un control de error absoluto en los algoritmos clásicos de tiempo discreto. Sin embargo, en numerosas aplicaciones es necesario controlar también el error relativo. Esto puede lograrse en los algoritmos QSS mediante el uso de *cuantificación logarítmica* [45], en la cual el tamaño del quantum aumenta a medida que crece el valor absoluto del estado $x_i(t)$.

Para este fin, el quantum se calcula utilizando dos parámetros, ΔQ_i^{rel} y ΔQ_i^{abs} . El primero se multiplica por el valor absoluto del estado y define la componente relativa del quantum; el segundo establece el tamaño mínimo que el quantum puede tomar cuando el estado se encuentra cerca de cero. Estos parámetros cumplen un rol análogo al de las tolerancias relativa y absoluta empleadas en los métodos clásicos de integración con paso variable.

2.3.8. Solver de QSS

El *Stand-Alone QSS Solver* [27] constituye una implementación completa y eficiente de la familia de algoritmos QSS. En este entorno, los modelos se especifican utilizando un subconjunto restringido del lenguaje Modelica [50], conocido como μ -Modelica. La herramienta traduce automáticamente los modelos a lenguaje C, codificando el correspondiente sistema de ecuaciones diferenciales ordinarias junto con las funciones de cruces por cero y las rutinas de manejo de eventos requeridas para representar dinámicas discontinuas.

Adicionalmente, el solver extrae información estructural, como las matrices de incidencia, y genera rutinas simbólicas para la evaluación de la matriz Jacobiana tanto en formato denso como disperso. El código C resultante puede luego vincularse tanto con algoritmos QSS y LIQSS de orden uno a tres, como con solvers convencionales, tales como DOPRI5 [23], DASSL [56], CVODE [21] e IDA [35].

Una ventaja importante de este solver es que proporciona información estructural completa y evaluación simbólica del Jacobiano, lo que permite que los integradores convencionales alcancen un desempeño significativamente superior en comparación con otras interfaces [46]. Además, dado que el código C resultante que se utiliza para evaluar las ecuaciones del sistema es básicamente el mismo tanto para los métodos de QSS como por los integradores clásicos, el entorno ofrece una base rigurosa para realizar comparaciones de desempeño justas entre ambos enfoques.

2.4. El Concepto de Actividad

El concepto de *actividad* asociado a señales continuas fue introducido originalmente con el propósito de cuantificar su tasa de cambio a lo largo del tiempo [39].

La definición formal de la actividad asociada a una señal continua $x_i(t)$, cuya derivada es integrable en un intervalo temporal $[t_0, t_f]$, se expresa como:

$$A_{x_i[t_0, t_f]} \triangleq \int_{t_0}^{t_f} \left| \frac{dx_i(\tau)}{d\tau} \right| d\tau \quad (2.15)$$

Esta métrica permite capturar la variación acumulada de una señal al cuantificar la distancia entre sus máximos y mínimos sucesivos. En el caso particular de una señal monótona, la expresión de la Ecuación (2.15) mide simplemente la distancia entre el valor máximo y el mínimo de la trayectoria. En cambio, para señales con múltiples cambios de pendiente, la actividad acumula dichas distancias a lo largo de los distintos tramos monótonos.

De este modo, la actividad de una señal puede relacionarse con el número de intervalos necesarios para construir una aproximación por funciones a trozos constantes. Cuando una señal continua es aproximada por un algoritmo de QSS1, la trayectoria de $q_i(t)$ resulta ser una función constante a trozos, tal como se ilustra en la Figura 2.1. En este contexto, el número de discontinuidades del estado cuantificado², puede aproximarse dividiendo la actividad de la señal por el tamaño del quantum:

$$k \approx \frac{A_{x_i[t_0, t_f]}}{\Delta Q_i} \quad (2.16)$$

Esta relación permite establecer cotas inferiores para el número de pasos que deben realizar

²Cada discontinuidad del estado cuantificado corresponde a un evento de actualización del algoritmo y, por lo tanto, al procesamiento de un nuevo paso de simulación.

los métodos QSS de primer orden al simular un sistema fijada una cierta tolerancia. Así, la actividad clásica se puede vincular directamente con los costos computacionales mínimos de QSS1.

De acuerdo con esta formulación, la actividad sólo mide distancias entre valores iniciales y finales, sin utilizar la información sobre cómo la señal alcanza dichos valores. Por ejemplo, una señal monótona que crece (o decrece) con una trayectoria lineal presenta la misma actividad que una señal monótona que comienza y termina en los mismos valores pero sigue una función del tiempo más compleja, aún cuando esta última requiere un número mayor de segmentos para ser representada con la precisión deseada si se utilizaran aproximaciones seccionalmente lineales. Por este motivo, la definición original de actividad presenta limitaciones cuando se consideran métodos de QSS de orden mayor, como por ejemplo QSS2 que genera trayectorias de salida seccionalmente lineales como muestra la Figura 2.2. Además, se sabe que en métodos de segundo orden el número de pasos depende de la raíz cuadrada del tamaño del quantum, una relación que la actividad clásica no puede capturar.

Con el objetivo de extender esta noción y permitir el análisis de aproximaciones de orden superior, la definición original fue generalizada en [18] mediante la introducción del concepto de *actividad de orden n* . Esta extensión incorpora información no sólo sobre los valores de la señal, sino también sobre sus derivadas temporales hasta un orden n dado. Formalmente, la actividad de orden n de una señal $x_i(t)$ en el intervalo $[t_0, t_f]$ se define como:

$$A_{x_i[t_0, t_f]}^{(n)} \triangleq \int_{t_0}^{t_f} \left| \frac{1}{n!} \frac{d^n x_i(\tau)}{d\tau^n} \right|^{1/n} d\tau \quad (2.17)$$

Esta formulación permite estimar el número de segmentos polinomiales (de grado hasta $n - 1$) necesarios para aproximar la señal $x_i(t)$ con un error máximo inferior a un valor predeterminado ΔQ_i dentro del intervalo de interés. En particular, el número estimado de segmentos k puede expresarse como:

$$k \approx \left(\frac{1}{\Delta Q_i} \right)^{1/n} \int_{t_0}^{t_f} \left| \frac{1}{n!} \frac{d^n x_i(\tau)}{d\tau^n} \right|^{1/n} d\tau = \frac{A_{x_i[t_0, t_f]}^{(n)}}{(\Delta Q_i)^{1/n}} \quad (2.18)$$

La estimación anterior considera que cada segmento de la aproximación polinómica se inicia con los mismos valores y derivadas (hasta orden $n - 1$) que la señal original $x_i(t)$ en el punto inicial del tramo correspondiente. Por lo tanto, esta expresión proporciona una estimación teórica del número mínimo de pasos de integración necesarios para aproximar la variable de estado $x_i(t)$ cuando se especifica un quantum ΔQ_i , con los métodos clásicos de QSS de orden n .

Capítulo 3

Homogeneidad de la Actividad

En este capítulo, nos centramos en la primera pregunta presentada en la introducción: *¿Cuándo es más conveniente, en términos de costo computacional, utilizar discretización temporal y cuándo cuantificación de estados?*.

Para esto, es importante tener en cuenta algunas ideas mencionadas en el segundo capítulo. En primer lugar, que los **métodos clásicos** de simulación para sistemas continuos descritos por ecuaciones diferenciales ordinarias (ODEs) se basan fundamentalmente en técnicas de discretización temporal [32, 33, 19]. y que estos algoritmos **avanzan el estado global del sistema computando actualizaciones sincrónicas en puntos de tiempo discretos para todas las variables**, independientemente de la dinámica local que pueda estar presente.

En segundo lugar, recordamos que un marco alternativo lo proporcionan los **métodos de QSS** [41, 19, 51], que **operan realizando actualizaciones asincrónicas sobre variables individuales del estado, únicamente cuando ocurren cambios lo suficientemente significativos**. Como consecuencia, los métodos QSS explotan de manera natural la dispersión y la localidad de la actividad en sistemas dinámicos, ofreciendo ventajas computacionales cuando los cambios de estado se encuentran limitados a un subconjunto acotado de variables, es decir, cuando la actividad [39, 53, 18] del sistema exhibe cierto grado de heterogeneidad.

Con estos conceptos en mente, en este capítulo, extendemos en primer lugar el concepto de actividad, que previamente había sido definido para medir la tasa total de cambio de una señal continua [18]. En esta extensión, proponemos una definición de *actividad instantánea* y luego distinguimos la *actividad local* de una trayectoria de estado particular de la *actividad global* del vector de estados completo. Mostramos que la actividad local está relacionada con el número de pasos requeridos por los métodos QSS, mientras que la actividad global está relacionada con el número de pasos requeridos por los algoritmos clásicos de integración numérica en tiempo discreto y mostramos como pueden utilizarse estos conceptos para derivar cotas inferiores del número de pasos requeridos en cada enfoque de integración numérica.

Luego, utilizando las expresiones obtenidas para la actividad local y global, caracterizamos

la homogeneidad de la actividad mediante la definición del *Factor de Homogeneidad*. Esta métrica cuantitativa proporciona un criterio para identificar el enfoque de simulación más adecuado, considerando las ventajas en términos de costo computacional que presentan los métodos basados en discretización temporal y los basados en cuantificación de estados.

Estos conceptos, inicialmente desarrollados para sistemas puramente continuos, se extienden luego para contemplar la presencia de discontinuidades.

Mostramos que el análisis basado en la homogeneidad de la actividad constituye la corroboración teórica de los resultados prácticos observados en la mayoría de las publicaciones que comparan QSS con algoritmos numéricos clásicos, donde la conveniencia de QSS se relaciona con combinaciones de actividad heterogénea, dispersión y presencia frecuente de discontinuidades.

3.1. Ejemplo Motivador

Consideremos el siguiente conjunto ODEs, que representa una discretización espacial de un modelo unidimensional de advección-difusión-reacción (ADR) obtenida mediante el Método de Líneas [49]:

$$\dot{u}_i(t) = -A \cdot \frac{u_i(t) - u_{i-1}(t)}{\Delta x} + D \cdot \frac{u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)}{\Delta x^2} + R \cdot (u_i(t)^2 - u_i(t)^3) \quad (3.1)$$

para $i = 2, 3, \dots, N-1$, donde A , D y R son los parámetros de advección, difusión y reacción, respectivamente y Δx representa la distancia entre los puntos contiguos de la grilla.

Los puntos inicial ($i = 1$) y final ($i = N$) de la grilla unidimensional obedecen a las siguientes ecuaciones:

$$\dot{u}_1(t) = -A \cdot \frac{u_1(t) - 1}{\Delta x} + D \cdot \frac{u_2(t) - 2u_1(t) + 1}{\Delta x^2} + R \cdot (u_1(t)^2 - u_1(t)^3) \quad (3.2)$$

$$\dot{u}_N(t) = -A \cdot \frac{u_N(t) - u_{N-1}(t)}{\Delta x} + D \cdot \frac{2u_{N-1}(t) - 2u_N(t)}{\Delta x^2} + R \cdot (u_N(t)^2 - u_N(t)^3) \quad (3.3)$$

La Figura 3.1a muestra las trayectorias de los estados obtenidas al simular el sistema con $N = 4$ y parámetros $A = 1$, $D = 1$, $R = 0.1$, longitud $L = 10$ y $\Delta x = L/(N-1)$. La Figura 3.1b muestra los resultados para un caso similar, siendo la única diferencia el valor de $R = 1000$.

Se observa que, en el primer caso, la mayor parte del tiempo hay al menos dos variables que presentan cambios simultáneos apreciables, mientras que en el segundo caso no se observan variaciones simultáneas significativas entre las variables. Diremos entonces que en el primer caso las trayectorias evolucionan de manera más *homogénea*, mientras que en el segundo la

evolución es más *heterogénea*.

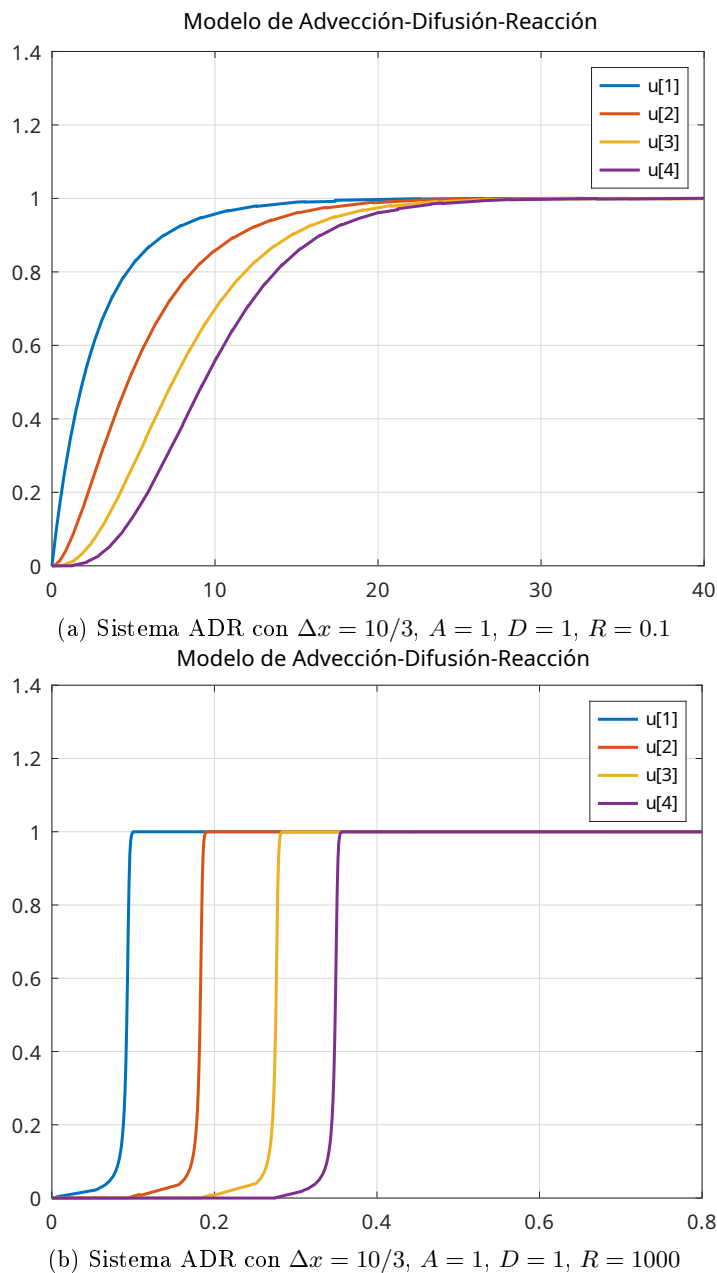


Figura 3.1: Trayectorias del sistema ADR para distintos valores de parámetros.

Luego, ambos modelos fueron simulados utilizando dos métodos de integración numérica de primer orden: una versión de paso variable del algoritmo Backward Euler (BE01), y el método LIQSS1, basado en cuantificación de estados. La Tabla 3.1 presenta el número de pasos y el número de evaluaciones escalares de funciones requeridos por cada método para completar las simulaciones, empleando una tolerancia absoluta (para BE01) y un tamaño de quantum (para LIQSS1) de 10^{-2} .¹

Como se observa en la tabla, en el primer caso ($R = 0.1$), el método BE01 completó la

¹Este valor fue seleccionado con fines ilustrativos, con el objetivo de mostrar el comportamiento de los métodos bajo un nivel de precisión comparable, sin que la elección específica del parámetro constituya el foco principal del análisis.

Caso	Pasos LIQSS1	Pasos BE01	Evals. LIQSS1	Evals. BE01
Caso 1 ($R = 0.1$)	405	196	1015	784
Caso 2 ($R = 1000$)	407	454	1020	1816

Tabla 3.1: Número de pasos requeridos para simular el modelo ADR.

simulación utilizando menos pasos que LIQSS1. Sin embargo, en el segundo caso ($R = 1000$), LIQSS1 requirió menos pasos que BE01.

Esta observación corrobora lo que se mencionó anteriormente: cuando un sistema evoluciona de forma *homogénea* (es decir, cuando las trayectorias individuales de los estados presentan tasas de cambio similares), resulta más eficiente actualizar todas las variables simultáneamente (de manera síncrona), como ocurre en los métodos clásicos de integración. Por el contrario, cuando las trayectorias de los estados presentan tasas de cambio disímiles, se dice que el sistema evoluciona de forma *heterogénea*, y en estos casos suele ser más conveniente actualizar las variables de manera independiente (es decir, de forma asincrónica), como en los métodos QSS.

Con el fin de formalizar esta afirmación, extenderemos el concepto de *actividad*, dado que su definición actual no permite cuantificar la homogeneidad de las trayectorias del sistema ni estimar el número de pasos requeridos por los algoritmos clásicos de integración.

3.2. Actividad instantánea, local y global

Considerando la definición de la *actividad de orden n* de una señal $x_i(t)$ en un intervalo dado $[t_0, t_f]$, presentada en la Sección 2.4, definimos la *actividad instantánea local de orden n* de la señal como el integrando de la Ecuación (2.17), dado por

$$a_{x_i}^{(n)}(t) \triangleq \left| \frac{1}{n!} \cdot \frac{d^n x_i(t)}{dt^n} \right|^{1/n} \quad (3.4)$$

y definimos la *actividad local de orden n* de $x_i(t)$ en el intervalo de tiempo $[t_0, t_f]$ como ²

$$A_{x_i[t_0, t_f]}^{(n)} \triangleq \int_{t_0}^{t_f} a_{x_i}^{(n)}(t) dt \quad \text{con } i = 1, 2, \dots, N \quad (3.5)$$

La expresión anterior coincide con la definición original de *actividad* presentada en [18] y puede vincularse con el número mínimo de pasos realizados por los métodos QSS.

Con el objetivo de obtener una medida relacionada con el número de pasos requeridos por los métodos clásicos en tiempo discreto, extendemos esta idea definiendo la *actividad instantánea global de orden n* como el máximo valor de la *actividad instantánea local* entre todos los estados del sistema:

²Notar que el calificativo *local* hace referencia a que la métrica se asocia a una única variable del estado $x_i(t)$.

$$a_{\mathbf{x}}^{(n)}(t) \triangleq \max_i \{a_{x_i}^{(n)}(t)\} = \max_i \left\{ \left| \frac{1}{n!} \cdot \frac{d^n x_i(t)}{dt^n} \right|^{1/n} \right\} \quad (3.6)$$

Posteriormente, definimos la *actividad global de orden n* para el vector de trayectorias de estado $\mathbf{x}(t)$ en un intervalo $[t_0, t_f]$ mediante la integración de la actividad instantánea global:³

$$A_{\mathbf{x}[t_0, t_f]}^{(n)} \triangleq \int_{t_0}^{t_f} a_{\mathbf{x}}^{(n)}(t) dt \quad (3.7)$$

La Tabla 3.2 resume entonces las distintas definiciones de actividad.

Tipo de actividad	Símbolo	Expresión
Instantánea local	$a_{x_i}^{(n)}(t)$	$\left \frac{1}{n!} \cdot \frac{d^n x_i(t)}{dt^n} \right ^{1/n}$
Local	$A_{x_i[t_0, t_f]}^{(n)}$	$\int_{t_0}^{t_f} a_{x_i}^{(n)}(t) dt$
Instantánea global	$a_{\mathbf{x}}^{(n)}(t)$	$\max_i \{a_{x_i}^{(n)}(t)\}$
Global	$A_{\mathbf{x}[t_0, t_f]}^{(n)}$	$\int_{t_0}^{t_f} a_{\mathbf{x}}^{(n)}(t) dt$

Tabla 3.2: Resumen de las definiciones de actividad.

3.3. Relación entre actividad y mínimo costo computacional

El costo computacional de la simulación de un sistema de ODEs está intrínsecamente ligado al número de pasos y cálculos requeridos por el método de integración numérica para alcanzar un nivel de precisión determinado en la aproximación de las trayectorias de estado.

El siguiente teorema, cuya demostración se presenta en el Apéndice A, sirve como base para establecer una relación entre la *actividad* de una trayectoria y el número mínimo de pasos necesarios para su obtención en simulación:

Teorema 1.

Dada una señal $x_i(t)$ expresada como un polinomio de grado n , existe un polinomio $q_i(t)$ de grado menor o igual que $n - 1$ que satisface la condición $|q_i(t) - x_i(t)| \leq \Delta Q_i$ en un intervalo $[t_j, t_j + \Delta t]$ si y sólo si

$$\Delta t \leq \frac{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}}{a_{x_i}^{(n)}(t_j)} \quad (3.8)$$

donde $a_{x_i}^{(n)}(t_j)$ representa la actividad instantánea local de orden n de la señal $x_i(t)$ en $t = t_j$.

³Para simplificar la notación, ocasionalmente se omitirá la dependencia explícita en el intervalo de tiempo $[t_0, t_f]$.

3.3.1. Costos computacionales en métodos basados en cuantificación de estados

La Ecuación (3.8), que es válida para cualquier intervalo de tiempo $[t_j, t_j + \Delta t]$, puede reescribirse como

$$\frac{a_{x_i}^{(n)}(t_j) \cdot \Delta t}{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}} \leq 1 \quad (3.9)$$

Consideremos una simulación QSS de orden n de la Ecuación (2.2) en el intervalo temporal $[t_0, t_f]$, donde el estado cuantificado $q_i(t)$ se actualiza en los instantes $t_0, t_1, \dots, t_{k-1} \leq t_f$. En cada intervalo del tipo $[t_j, t_{j+1})$, el estado cuantificado $q_i(t)$ sigue un polinomio de grado $n-1$, mientras que el estado $x_i(t)$ evoluciona según un polinomio de grado n .

Dado que la diferencia entre x_i y q_i está acotada por ΔQ_i , la Ecuación (3.9) se cumple para todo $0 \leq j \leq k-1$. Si sumamos ambos lados de la desigualdad desde $j=0$ hasta $j=k-1$, obtenemos:

$$\sum_{j=0}^{k-1} \frac{a_{x_i}^{(n)}(t_j) \cdot (t_{j+1} - t_j)}{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}} \leq \sum_{j=0}^{k-1} 1 = k \quad (3.10)$$

Como $a_{x_i}^{(n)}$ es constante en cada intervalo $[t_j, t_{j+1}]$, la suma anterior puede representarse como una integral, de modo que se tiene:

$$\frac{1}{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}} \int_{t_0}^{t_f} a_{x_i}^{(n)}(t) dt \leq k \quad (3.11)$$

Sustituyendo la integral por la definición de actividad local dada en la Ecuación (3.5), resulta:

$$\frac{A_{x_i}^{(n)}}{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}} \leq k \quad (3.12)$$

Por lo tanto, el número mínimo de segmentos del estado cuantificado $q_i(t)$ utilizados por un método QSS de orden n en el intervalo $[t_0, t_f]$ puede aproximarse mediante:

$$k_{x_i[t_0, t_f]}^{(n)}(\Delta Q_i) \triangleq \frac{A_{x_i}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q_i)^{\frac{1}{n}}} \quad (3.13)$$

Dado que los métodos QSS actualizan cada estado de manera independiente, una cota inferior para el número total de pasos requeridos para simular el sistema completo puede aproximarse por:

$$k_{\text{QS}[t_0, t_f]}^{(n)}(\Delta \mathbf{Q}) \triangleq \sum_{i=1}^N k_{x_i[t_0, t_f]}^{(n)}(\Delta Q_i) = \sum_{i=1}^N \frac{A_{x_i}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q_i)^{\frac{1}{n}}} \quad (3.14)$$

donde $\Delta \mathbf{Q} = [\Delta Q_1, \dots, \Delta Q_N]^T$.

Durante cada paso, un método QSS realiza cálculos no sólo sobre el estado x_i que cambia su valor cuantificado, sino también sobre aquellos estados x_j cuyas derivadas dependen de x_i . Denotando con r_i al número de estados cuyas derivadas dependen de x_i , el número total de actualizaciones realizadas por un método QSS de orden n puede acotarse inferiormente mediante:

$$u_{\text{QS}[t_0, t_f]}^{(n)}(\Delta \mathbf{Q}) \triangleq \sum_{i=1}^N k_{x_i[t_0, t_f]}^{(n)}(\Delta Q_i) \cdot r_i = \frac{1}{2^{\frac{2n-1}{n}} \cdot (\Delta Q)^{\frac{1}{n}}} \sum_{i=1}^N A_{x_i}^{(n)} \cdot r_i \quad (3.15)$$

donde el último paso es válido únicamente cuando $\Delta Q_i = \Delta Q$ para $i = 1, \dots, N$.

Cabe destacar que la Ecuación (3.13) proporciona una cota inferior para el número de pasos más ajustada que la obtenida en la Ecuación (2.18). La razón es que la expresión de (2.18) sólo era válida para los métodos QSS originales, en los cuales cada segmento de q_i comenzaba coincidiendo con x_i . La nueva formulación de la Ec.(3.13), en cambio, cubre el caso general. En los algoritmos LIQSS, por ejemplo, los segmentos de q_i se inician a una distancia ΔQ_i de x_i , lo que permitiría realizar pasos de mayor tamaño hasta que la distancia entre ambos vuelve a ser igual a ΔQ_i .

3.3.2. Costos computacionales en métodos basados en discretización temporal

Los algoritmos clásicos con control de paso estiman el error local restando la solución obtenida mediante un método de orden $n - 1$ de aquella obtenida con un método de orden n ⁴. Si esta estimación del error supera la tolerancia especificada, el paso se repite utilizando un tamaño de paso menor; en caso contrario, el paso se acepta (y el tamaño de paso se incrementa).

Después de cada paso, la diferencia entre las soluciones de los métodos de orden $n - 1$ y n puede aproximarse como la diferencia entre dos polinomios de los órdenes correspondientes [19]. Si la tolerancia de error es la misma para todos los estados y se denota como ΔQ_i , entonces el tamaño de paso Δt verificará la Ecuación (3.8) para $i = 1, \dots, N$. De esta forma, el tamaño de paso estará acotado superiormente por el componente i que imponga la menor cota para Δt , es decir:

⁴Ciertas implementaciones comparan dos métodos de orden n , pero esto es equivalente a usar un método de orden $n + 1$ ya que usando dos fórmulas de orden n la *interpolación de Richardson* [20] permite obtener una fórmula de orden $n + 1$.

$$\Delta t \leq \frac{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}}{\max_i(a_{x_i}^{(n)}(t_j))} = \frac{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}}{a_{\mathbf{x}}^{(n)}(t_j)} \quad (3.16)$$

A partir de esta ecuación, y siguiendo un razonamiento análogo al utilizado para los métodos QSS, se obtiene la siguiente expresión que aproxima una cota inferior para el número de pasos realizados por un método clásico:

$$k_{\text{DT}[t_0, t_f]}^{(n)}(\Delta Q) \triangleq \frac{A_{\mathbf{x}}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q)^{\frac{1}{n}}} \quad (3.17)$$

Dado que los algoritmos en tiempo discreto actualizan los N estados en cada paso, el número total de actualizaciones de estado puede acotarse inferiormente mediante:

$$u_{\text{DT}[t_0, t_f]}^{(n)}(\Delta Q) \triangleq N \cdot k_{\text{DT}[t_0, t_f]}^{(n)}(\Delta Q) = N \cdot \frac{A_{\mathbf{x}}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q)^{1/n}} \quad (3.18)$$

Esta formulación permite establecer una relación directa entre la actividad global del sistema y el costo computacional mínimo asociado al uso de métodos clásicos de integración numérica basados en discretización temporal.

3.4. Factor de Homogeneidad

Cuando todos los estados evolucionan con una tasa de cambio similar (es decir, de manera *homogénea*), puede notarse fácilmente que la actividad local instantánea $a_{x_i}^{(n)}(t)$ de todas las variables de estado será cercana al valor máximo de actividad local instantánea, el cual coincide con la actividad global instantánea $a_{\mathbf{x}}^{(n)}(t)$. Extendiendo este razonamiento a un intervalo de tiempo dado, se concluye que la actividad local $A_{x_i}^{(n)}$ de cada estado será próxima a la actividad global $A_{\mathbf{x}}^{(n)}$.

Por el contrario, cuando las trayectorias del sistema son *heterogéneas*, la actividad global será significativamente mayor que la mayoría de las actividades locales. En consecuencia, resulta natural definir el *Factor de Homogeneidad* (HF, por sus siglas en inglés) de orden n como:

$$H_{\mathbf{x}[t_0, t_f]}^{(n)} \triangleq \frac{\sum_{i=1}^N A_{x_i[t_0, t_f]}^{(n)}}{N \cdot A_{\mathbf{x}[t_0, t_f]}^{(n)}} \quad (3.19)$$

Del cual podemos observar lo siguiente:

- $H_{\mathbf{x}}^{(n)}$ toma valores únicamente en el intervalo $[\frac{1}{N}, 1]$.
- Un caso extremo ocurre cuando todas las variables de estado evolucionan con una actividad idéntica. En tal escenario, se cumple que $A_{x_i}^{(n)} = A_{\mathbf{x}}^{(n)}$ para todo i y, a partir de la

Ec. (3.19), resulta $H_{\mathbf{x}}^{(n)} = 1$. En este caso se alcanza la máxima homogeneidad posible.

- El otro caso extremo se presenta cuando sólo una variable de estado exhibe actividad. En tal situación, $A_{x_i}^{(n)} = A_{\mathbf{x}}^{(n)}$ para dicha variable y $A_{j[t_0, t_f]}^{(n)} = 0$ para las restantes. Así, de la Ec. (3.19) se obtiene $H_{\mathbf{x}}^{(n)} = 1/N$, lo que corresponde a la mínima homogeneidad posible.
- Una situación similar surge cuando distintas variables presentan actividad, pero no de forma simultánea (es decir, la actividad instantánea de todas excepto una variable es siempre nula). En este caso también se cumple $H_{\mathbf{x}}^{(n)} = 1/N$.
- En general, un valor de HF próximo a 1 indica una actividad homogénea, mientras que un valor cercano a $1/N$ representa una actividad heterogénea.

3.4.1. Costos Computacionales y Factor de Homogeneidad

Consideremos un sistema en el que el número de derivadas de estado que dependen de cada variable es el mismo, es decir, $r_i = r$ para $i = 1, \dots, N$ (recordando que r representa el grado de interconexión entre los estados). Supongamos además que se utiliza el mismo quantum $\Delta Q_i = \Delta Q$ para todas las variables en una simulación QSS, y que dicho valor coincide con la tolerancia de un algoritmo clásico de tiempo discreto. Entonces, utilizando las Ecs. (3.15), (3.18) y (3.19), puede relacionarse el número mínimo de actualizaciones de variables de ambos algoritmos mediante el Factor de Homogeneidad como:

$$\frac{u_{\text{QS}}^{(n)}}{u_{\text{DT}}^{(n)}} = H_{\mathbf{x}}^{(n)} \cdot r \quad (3.20)$$

A partir de esta expresión pueden extraerse las siguientes conclusiones:

- Si el producto $H_{\mathbf{x}}^{(n)} \cdot r > 1$, el límite inferior para el número de actualizaciones de estado de los métodos clásicos será menor que el de los métodos QSS. En consecuencia, se espera que los métodos clásicos sean más eficientes que los basados en cuantificación.
- Por el contrario, si $H_{\mathbf{x}}^{(n)} \cdot r < 1$, se espera que los métodos QSS sean más eficientes.
- Cuando las trayectorias del sistema son homogéneas ($H_{\mathbf{x}}^{(n)} \approx 1$), resulta $H_{\mathbf{x}}^{(n)} \cdot r > 1$ (salvo que el sistema esté totalmente desconectado, con $r = 1$). Por tanto, los algoritmos clásicos de tiempo discreto se prevén más eficientes.
- Si las trayectorias del sistema son heterogéneas ($H_{\mathbf{x}}^{(n)} \approx 1/N$), se cumple $H_{\mathbf{x}}^{(n)} \cdot r < 1$ (excepto si el sistema está completamente conectado con $r = N$). En este caso, los algoritmos basados en cuantificación resultan más convenientes.

- Cuando el sistema es muy disperso (es decir, $r \approx 1$), se obtiene $H_{\mathbf{x}}^{(n)} \cdot r < 1$ (a menos que las trayectorias sean muy homogéneas y $H_{\mathbf{x}}^{(n)} \approx 1$). En tal caso, los métodos QSS resultan convenientes.
- Cuando el sistema está fuertemente acoplado (es decir, $r \approx N$), se cumple $H_{\mathbf{x}}^{(n)} \cdot r > 1$ (salvo que las trayectorias sean muy heterogéneas y $H_{\mathbf{x}}^{(n)} \approx 1/N$). Por tanto, los algoritmos clásicos resultan más adecuados.

Estas conclusiones, que se derivan directamente de la Ec. (3.20), confirman observaciones prácticas reportadas en diversos trabajos previos acerca de la conveniencia de emplear métodos QSS en sistemas dispersos con actividad heterogénea.

Sin embargo, este análisis presenta ciertas limitaciones:

- Se asumió que cada variable de estado aparece exactamente r veces en el lado derecho de la Ec. (2.2). De todos modos, r puede interpretarse como un valor medio ponderado de los r_i . A partir de la Ec. (3.15), se tiene:

$$u_{\text{QS}[t_0, t_f]}^{(n)}(\Delta \mathbf{Q}) = \sum_{i=1}^N k_{x_i[t_0, t_f]}^{(n)}(\Delta Q_i) \cdot r^{(n)}$$

donde se define:

$$r^{(n)} \triangleq \frac{\sum_{i=1}^N k_{x_i[t_0, t_f]}^{(n)}(\Delta Q_i) \cdot r_i}{\sum_{i=1}^N k_{x_i[t_0, t_f]}^{(n)}(\Delta Q_i)}$$

es decir,

$$r^{(n)} = \sum_{i=1}^N \frac{k_{x_i[t_0, t_f]}^{(n)}(\Delta Q_i)}{k_{\text{QS}[t_0, t_f]}^{(n)}(\Delta \mathbf{Q})} \cdot r_i = \sum_{i=1}^N c_i r_i \quad (3.21)$$

donde las constantes de ponderación c_i representan la proporción de pasos ejecutados por cada variable de estado. En el caso en que $\Delta Q_i = \Delta Q$ para todo i , la Ec. (3.21) puede reescribirse como:

$$r^{(n)} = \sum_{i=1}^N \frac{A_{x_i[t_0, t_f]}^{(n)}}{\sum_{i=1}^N A_{x_i[t_0, t_f]}^{(n)}} \cdot r_i = \sum_{i=1}^N c_i r_i \quad (3.22)$$

lo que proporciona una expresión para c_i independiente del tamaño del quantum.

- Para llegar a la Ec. (3.20), asumimos que el quantum se fija como $\Delta Q_i = \Delta Q$ para todas las variables en los métodos QSS⁵. Además, consideramos ΔQ constante. En la práctica, se emplea mayormente cuantificación logarítmica, donde ΔQ_i crece con la magnitud de

⁵ Nótese que el uso de diferentes valores de ΔQ_i puede evitarse introduciendo un cambio de variables del tipo $z_i = x_i / \Delta Q_i$.

la señal $x_i(t)$. De manera análoga, en los métodos clásicos de tiempo discreto consideramos sólo la tolerancia absoluta; si tenemos en cuenta tolerancias relativas y señales con amplias variaciones de magnitud, la Ec. (3.20) podría no ser válida.

- En cualquier caso, la Ec. (3.20) relaciona los límites inferiores del número de actualizaciones de estado. El número real de pasos puede ser considerablemente mayor cuando los sistemas son rígidos, por ejemplo, y los algoritmos empleados no son adecuados para tales condiciones.
- En la práctica, el cálculo del Factor de Homogeneidad requiere conocer las actividades locales y globales, lo cual implica disponer de las trayectorias del sistema⁶. Dado que estas trayectorias sólo se conocen tras la simulación, el concepto no puede emplearse de antemano para decidir qué enfoque es más adecuado. Sin embargo, puede estimarse un valor de $H_{\mathbf{x}}^{(n)}$ para un sistema particular y utilizarlo como aproximación para otros sistemas que exhiban un comportamiento similar. Cabe destacar que la mayoría de las conclusiones anteriores no dependen del conocimiento exacto del HF.
- Una manera práctica de obtener un valor aproximado de $H_{\mathbf{x}}^{(n)}$ y decidir qué algoritmo resulta más apropiado consiste en realizar una simulación *rápida y gruesa* mediante un algoritmo QSS_n con un quantum grande, aprovechando el hecho de que estos métodos preservan la estabilidad independientemente del tamaño de ΔQ_i .
- El HF también podría calcularse durante la simulación (ya sea iniciando con un enfoque QSS o de tiempo discreto) y emplearse luego para cambiar de algoritmo según su valor. Esta idea podría conducir a un esquema similar a la selección automática de algoritmos utilizada en solvers clásicos de ODEs [55].

3.5. Ejemplo de aplicación

En esta sección presentamos la aplicación de los conceptos teóricos desarrollados previamente sobre el ejemplo motivador introducido al inicio de la sección para corroborar la correspondencia entre estos conceptos y los resultados de simulación.

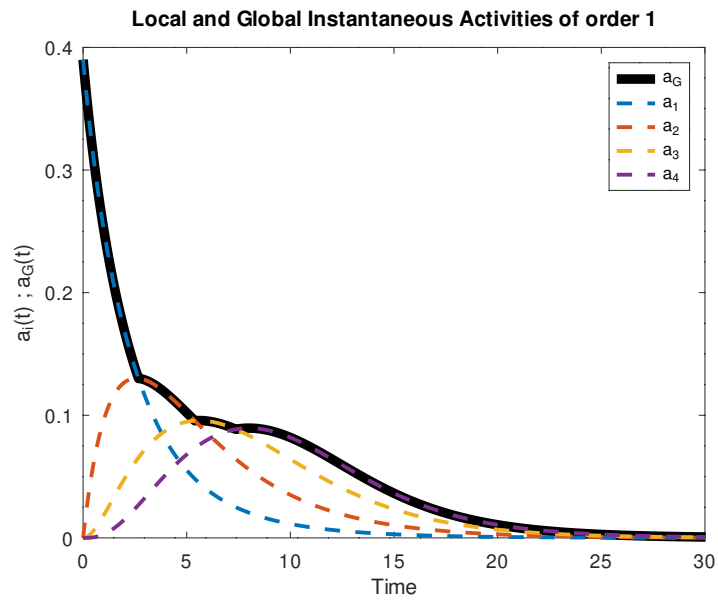
Los resultados reportados fueron obtenidos utilizando el *Stand Alone QSS Solver* [27] para los métodos QSS, mientras que los resultados de los algoritmos clásicos se obtuvieron empleando *Octave*.

⁶En la práctica, las actividades globales y locales pueden calcularse directamente a partir de los resultados de simulación aprovechando la salida densa que proporcionan los solvers.

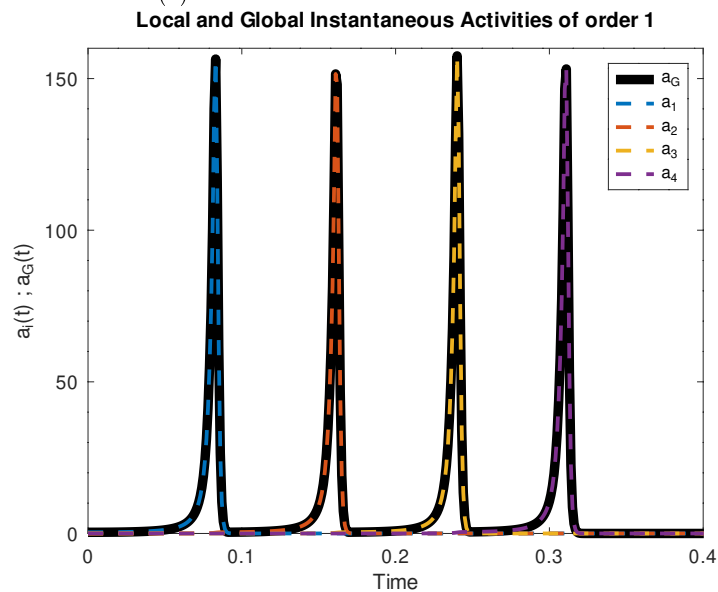
3.5.1. Actividades Locales, Globales y Factor de Homogeneidad

Dado que las trayectorias de los estados del sistema no pueden calcularse analíticamente, determinamos las actividades locales y globales a partir de la solución numérica obtenida mediante el algoritmo LIQSS2 [51] con una tolerancia muy baja ($\Delta Q = 10^{-12}$), ya que este método garantiza cotas de error global en este tipo de problemas [4].

Las Figuras 3.2a y 3.2b presentan las actividades instantáneas locales de orden 1 de cada escenario, $a_{x_i}^{(1)}$, calculadas con la Ec. (3.4). En la misma figura se muestra también las actividades instantáneas globales de orden 1, $a_x^{(1)}(t)$, que corresponden claramente al máximo de las actividades instantáneas locales en cada instante de tiempo.



(a) Actividad de orden 1 - Escenario 1



(b) Actividad de orden 1 - Escenario 2

Figura 3.2: Actividad instantánea (local y global) del modelo ADR para los diferentes casos

Integrando las actividades instantáneas $a_{x_i}^{(n)}$ en el tiempo según la Ec. (3.5), para orden $n = 1$, obtuvimos los valores de la actividad local de cada variable de estado. De forma análoga, integrando la actividad instantánea global $a_{\mathbf{x}}^{(n)}(t)$ según la Ec. (3.7), calculamos las actividades globales de orden 1. La actividad local de orden 1 de cada estado es igual a 1, ya que todas las trayectorias crecen monótonamente desde 0 hasta 1, experimentando un cambio absoluto de 1, mientras que los valores obtenidos para la actividad global de cada escenario fueron:

- Escenario 1: $A_{\mathbf{x}}^{(1)} = 1.79$
- Escenario 2: $A_{\mathbf{x}}^{(1)} = 3.98$

Con estos resultados calculamos los factores de homogeneidad $H_{\mathbf{x}}^{(n)}$ de órdenes 1 a 3 según la Ec. (3.19), cuyos valores se presentan en la Tabla 3.3. Allí también se incluye el valor del producto $H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$, utilizando el valor de $r^{(n)} = 2.5$ obtenido de acuerdo con la Ec. (3.22).

	$H_{\mathbf{x}}^{(n)}$	$H_{\mathbf{x}}^{(1)} \cdot r^{(1)}$
Escenario 1	0.56	1.40
Escenario 2	0.25	0.62

Tabla 3.3: Factores de Homogeneidad - Modelo ADR con $N = 4$.

Podemos observar que el factor de homogeneidad del primer escenario es mayor al del segundo escenario, coherentemente a lo observado en la forma de las trayectorias de las variables de estados del modelo. Además $H_{\mathbf{x}}^{(1)} \cdot r^{(1)} > 1$ en el primer escenario, lo que sugiere la conveniencia de utilizar algoritmos de discretización temporal en este caso mientras que $H_{\mathbf{x}}^{(1)} \cdot r^{(1)} < 1$ en el segundo escenario, lo que sugiere que los algoritmos QSS serían más eficientes para utilizar en este caso.

3.5.2. Número de Pasos y Actualizaciones

A partir de las actividades locales y globales estimamos las cotas inferiores para el número de pasos requeridos por los algoritmos QSS y clásicos en cada escenario, de acuerdo con las Ecs. (3.14) y (3.17), utilizando diferentes configuraciones de tolerancia. Simulamos también el modelo, empleando algoritmos LIQSS1 y una versión de paso variable de Backward Euler, BE01 (En ambos casos métodos implícitos debido a la rigidez del sistema).

Las estimaciones de las cotas inferiores y el número real de pasos realizados por los algoritmos correspondientes se reportan en la Tabla 3.4.

En todos los casos, el número de pasos realizados es mayor de las cotas inferiores calculadas. En general, los métodos requieren más pasos que el límite inferior porque los algoritmos no siguen las trayectorias óptimas del Teorema 1, en las cuales la diferencia entre las trayectorias de orden n y $n - 1$ siguen un polinomio de Chebyshev, como veremos en el Capítulo 4.

	ΔQ	LIQSS1		BE01	
		$k_{QS}^{(1)}$	Real	$k_{DT}^{(1)}$	Real
Escenario 1	10^{-2}	200	405	89	196
	10^{-3}	2000	3996	897	1897
	10^{-4}	20000	39896	8970	18891
Escenario 2	10^{-2}	200	407	200	454
	10^{-3}	2000	3998	1992	4196
	10^{-4}	20000	39965	19919	41938

Tabla 3.4: Cota inferior y número real de pasos para distintos métodos numéricos - Modelo ADR

3.5.3. Factor de Homogeneidad y Actualizaciones de Estado

Los $H_{\mathbf{x}}^{(n)}$ calculados para cada uno de los escenarios reportados en la Tabla 5.2 y su producto con el grado de conectividad $r^{(1)}$ sugieren que los métodos QSS deberían requerir menos actualizaciones de estado que los algoritmos clásicos en el Escenario 2, mientras que los métodos basados en discretización temporal serían más convenientes en el Escenario 1. Para corroborarlo, calculamos las actualizaciones de estado de los distintos algoritmos multiplicando el número de pasos de cada variable por su grado de conectividad para QSS y por el tamaño del vector de estado $N = 4$ para los métodos clásicos. Los resultados se presentan en la Tabla 3.5, donde se corrobora la conveniencia prevista de cada método en cada escenario.

	Resultados teóricos $H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$	Resultados de simulación	
		ΔQ	$\frac{u_{QSS}}{u_{DT}}$
Escenario 1	1.40	10^{-2}	1.29
		10^{-3}	1.31
		10^{-4}	1.32
Escenario 2	0.62	10^{-2}	0.56
		10^{-3}	0.56
		10^{-4}	0.56

Tabla 3.5: Relación entre las actualizaciones de estado de algoritmos LIQSS y métodos clásicos de integración numérica.

Es importante enfatizar que la relación $H_{\mathbf{x}}^{(n)} \cdot r^{(n)} < 1$ no implica que un método basado en cuantificación siempre requiera menos actualizaciones que un método basado en discretización temporal. La interpretación correcta es que la cota teórica inferior para el número de actualizaciones en los métodos basados en cuantificación es menor que la cota teórica inferior en los métodos basados en discretización temporal. Un razonamiento recíproco aplica para el caso $H_{\mathbf{x}}^{(n)} \cdot r^{(n)} > 1$.

3.6. Extensión a Sistemas con Discontinuidades

En el análisis desarrollado anteriormente asumimos que las variables de estado siguen trayectorias diferenciables. En esta sección, extenderemos dichas nociones para considerar la presencia de discontinuidades.

Con este objetivo, consideremos un sistema híbrido de la forma:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{z}(t), t) \quad (3.23)$$

donde la nueva variable $\mathbf{z}(t) \in \mathbb{R}^p$ representa el estado discreto. Diremos que ocurre una discontinuidad de tipo i , con $i \in \{1, \dots, D\}$, siempre que se cumpla la condición:

$$g_i(\mathbf{x}(t), \mathbf{z}(t), t) = 0 \quad (3.24)$$

donde $g_i : \mathbb{R}^{N+p+1} \rightarrow \mathbb{R}$ son las *funciones de cruce por cero* (*zero-crossing functions*). Una vez que se verifica la condición de cruce por cero de tipo i , los estados se actualizan de acuerdo con:

$$[\mathbf{x}(t^+), \mathbf{z}(t^+)] = H_i(\mathbf{x}(t), \mathbf{z}(t), t) \quad (3.25)$$

donde $H_i : \mathbb{R}^{N+p+1} \rightarrow \mathbb{R}^{N+p}$ es el *event handler* asociado al evento de tipo i .

Una solución de este sistema híbrido, $\mathbf{x}(t)$, verifica la Ec. (3.23) en casi todo el dominio temporal, excepto en los instantes en los que ocurre un evento, es decir, cuando alguna de las condiciones de cruce por cero de la Ec. (3.24) se cumple.

Supondremos que la discontinuidad de tipo i provoca cambios instantáneos en s_i derivadas de estado (como consecuencia del event handler H_i y de la estructura de la función $\mathbf{f}$). Consideremos además que una discontinuidad de tipo i ocurre d_i veces en el intervalo $[t_0, t_f]$, en los instantes de evento $t_1^i, t_2^i, \dots, t_{d_i}^i$. Por lo tanto, el número total de discontinuidades en dicho intervalo viene dado por $d = \sum_{i=1}^D d_i$.

Para calcular el número de pasos realizados por los algoritmos basados en cuantificación y los de tiempo discreto en este caso, analizaremos la forma en que cada uno trata las discontinuidades. En cada instante de evento t_j^i , un método QSS actualizará las s_i derivadas de estado afectadas por la discontinuidad de tipo i . De manera similar, un método clásico de tiempo discreto realizará un paso en el instante del evento y reiniciará la simulación recalculando todo el estado. En ambos casos, la presencia de una discontinuidad desencadena cálculos equivalentes a los de un paso regular.

Por lo tanto, para determinar el número total mínimo de pasos en una simulación, es necesario sumar la parte correspondiente a la integración continua (relacionada con la actividad, como en el caso continuo) y la provocada por la presencia de discontinuidades. A partir de las Ecs. (3.14) y (3.17), podemos aproximar el número total mínimo de pasos requeridos por cada tipo de método mediante las siguientes expresiones:

■ **Cuantificación de estados:**

$$k_{\text{QS}}^{(n)}(\Delta \mathbf{Q}) \triangleq \sum_{i=1}^N \frac{A_{x_i}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q_i)^{\frac{1}{n}}} + d \quad (3.26)$$

■ **Discretización temporal:**

$$k_{\text{DT}}^{(n)}(\Delta Q) \triangleq \frac{A_{\mathbf{x}}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q)^{\frac{1}{n}}} + d \quad (3.27)$$

En cuanto al número total de actualizaciones, considerando que las discontinuidades desencadenan pasos regulares en ambos casos, podemos reformular las Ecs. (3.15) y (3.18) como:

■ **Cuantificación de estados:**

$$u_{\text{QS}}^{(n)}(\Delta \mathbf{Q}) \triangleq \sum_{i=1}^N k_{x_i}^{(n)}(\Delta Q_i) \cdot r_i + \sum_{i=1}^D d_i \cdot s_i \quad (3.28)$$

■ **Discretización temporal:**

$$u_{\text{DT}}^{(n)}(\Delta Q) \triangleq N \cdot k_{\text{DT}}^{(n)}(\Delta Q) + d \cdot N \quad (3.29)$$

Asumiendo que $r_i = r$ para $i = 1, \dots, N$ (o considerando r como el promedio ponderado de los r_i) y $s_i = s$ (o el promedio de los s_i), podemos aproximar el cociente entre los números mínimos de actualizaciones requeridas por cada tipo de método como:

$$\frac{u_{\text{QS}}^{(n)}}{u_{\text{DT}}^{(n)}} = \frac{\sum_{i=1}^N (k_{x_i}^{(n)} \cdot r) + \sum_{i=1}^D (d_i \cdot s)}{N \cdot k_{\text{DT}}^{(n)} + d \cdot N} = \frac{r \cdot k_{\text{QS}}^{(n)} + d \cdot s}{N \cdot k_{\text{DT}}^{(n)} + d \cdot N} \quad (3.30)$$

A partir de esta expresión, podemos extraer las siguientes conclusiones:

- Si el número de discontinuidades es muy pequeño, los términos $d \cdot s$ y $d \cdot N$ pueden despreciarse en la Ec. (3.30), resultando:

$$\frac{u_{\text{QS}}^{(n)}}{u_{\text{DT}}^{(n)}} \approx H_{\mathbf{x}}^{(n)} \cdot r$$

En este caso, la conveniencia de utilizar métodos QSS o clásicos de tiempo discreto depende casi exclusivamente de la dinámica continua.

- Si el número de discontinuidades es muy grande en comparación con los pasos asociados a la dinámica continua, se tiene:

$$d \rightarrow \infty \implies \frac{u_{\text{QS}}^{(n)}}{u_{\text{DT}}^{(n)}} \rightarrow \frac{s}{N}$$

En este caso, se espera que los algoritmos de integración basados en cuantificación sean más eficientes.

- Si los métodos QSS resultan más convenientes para la dinámica continua, entonces:

$$\frac{r \cdot k_{\text{QS}}^{(n)}}{N \cdot k_{\text{DT}}^{(n)}} < 1 \implies \frac{r \cdot k_{\text{QS}}^{(n)} + d \cdot s}{N \cdot k_{\text{DT}}^{(n)} + d \cdot N} < 1$$

y, por lo tanto, también se espera que los métodos QSS mantengan su ventaja en presencia de discontinuidades.

Estas conclusiones, que se derivan directamente de la Ec. (3.30), corroboran las observaciones prácticas que señalan la conveniencia de utilizar métodos QSS en presencia de discontinuidades.

Es importante señalar que, en los instantes en que ocurre un evento, la actividad instantánea de los estados afectados por dicho evento, dada por la Ec. (3.4), no está bien definida, ya que las derivadas temporales correspondientes pueden no existir. Este inconveniente puede resolverse asignando un valor arbitrario (por ejemplo, 0) a la actividad instantánea en los instantes en los que ocurre un evento, ya que esto no afecta el cálculo de las actividades locales y globales, las cuales se obtienen mediante la integración de las actividades instantáneas.

Capítulo 4

Algoritmos de LIQSS Generalizados

En este capítulo nos centraremos en la segunda pregunta presentada en la introducción: *¿Cómo diseñar un algoritmo capaz de operar lo más cerca posible de este costo computacional mínimo?*

Para esto, presentamos un procedimiento para diseñar métodos LIQSS generales basado en la definición de un polinomio $p_i(t)$ que representa la diferencia entre $q_i(t)$ y $x_i(t)$. Primero mostramos que, con la elección apropiada de este polinomio, es posible reproducir los algoritmos LIQSS existentes y extenderlos para realizar pasos más largos. Luego, mostramos una forma de definir $p_i(t)$ tal que el tamaño del paso se maximice, llegando así a una nueva familia de algoritmos denominados métodos LIQSS de Chebyshev (CheQSS). Esta familia de métodos busca alcanzar el mínimo número de pasos establecidos en el análisis basado en el concepto de actividad del capítulo anterior.

4.1. Formulación Básica

Consideremos el sistema de la Ec.(2.2), la aproximación QSS de la Ec.(2.11) y la aproximación lineal de sus componentes dada por las Ecs.(2.13)–(2.14). Además, asumimos, sin pérdida de generalidad, que $t_j^i = 0$. Para simplificar la notación, escribiremos simplemente $a_i(t_j^i) = a_i(0) = a_i$.

Definimos entonces el polinomio de diferencia $p_i(t)$ como

$$p_i(t) = x_i(t) - q_i(t) \tag{4.1}$$

A partir del mismo, diseñamos los distintos métodos LIQSS de manera tal que verifiquen la Ec.(4.1) para un polinomio $p_i(t)$ propuesto. Es decir, en lugar de comenzar desde una definición para la trayectoria cuantificada $q_i(t)$, proveeremos una definición para $p_i(t)$. De esa forma, dado el polinomio de diferencia deseado $p_i(t)$ de orden n , el problema consiste en obtener la trayectoria cuantificada $q_i(t)$ de orden $n - 1$ tal que la trayectoria de estado

resultante $x_i(t)$ de orden n , obtenida luego de integrar la Ec.(2.13), verifique efectivamente la Ec.(4.1).

Para obtener los n coeficientes que definen cada segmento del polinomio $q_i(t)$, puede plantearse un sistema de ecuaciones resolviendo la Ec. (4.1) para q_i , derivando, y reemplazando $\dot{x}_i(t)$ en la aproximación lineal de la Ec.(2.13):

$$q_i(t) = x_i(t) - p_i(t) \quad (4.2a)$$

$$\dot{q}_i(t) = a_i q_i(t) + u_i(t) - \dot{p}_i(t) \quad (4.2b)$$

$$\ddot{q}_i(t) = a_i \dot{q}_i(t) + \dot{u}_i(t) - \ddot{p}_i(t) \quad (4.2c)$$

$$\ddot{\dot{q}}_i(t) = a_i \ddot{q}_i(t) + \ddot{u}_i(t) - \ddot{\dot{p}}_i(t) \quad (4.2d)$$

donde limitamos las derivadas de $q_i(t)$ hasta orden tres, ya que no desarrollaremos algoritmos de orden superior. Definimos a continuación los siguientes coeficientes auxiliares:

$$r_1 \triangleq a_i x_i(0) + u_i(0) \quad (4.3a)$$

$$r_2 \triangleq a_i^2 x_i(0) + a_i u_i(0) + \dot{u}_i(0) \quad (4.3b)$$

$$r_3 \triangleq a_i^3 x_i(0) + \ddot{u}_i(0) + a_i \dot{u}_i(0) + a_i^2 u_i(0) \quad (4.3c)$$

y

$$s_1 \triangleq a_i p_i(0) + \dot{p}_i(0) \quad (4.4a)$$

$$s_2 \triangleq a_i^2 p_i(0) + a_i \dot{p}_i(0) + \ddot{p}_i(0) \quad (4.4b)$$

$$s_3 \triangleq a_i^3 p_i(0) + a_i^2 \dot{p}_i(0) + a_i \ddot{p}_i(0) + \ddot{\dot{p}}_i(0) \quad (4.4c)$$

Notar que los coeficientes r_k y s_k sólo dependen del estado inicial $x_i(0)$, del polinomio $p_i(t)$ y de los parámetros de la aproximación lineal. Luego, reemplazando la Ec. (4.2a) en la Ec. (4.2b) obtenemos

$$\begin{aligned} \dot{q}_i(0) &= a_i(x_i(0) - p_i(0)) + u_i(0) - \dot{p}_i(0) \\ &= a_i x_i(0) + u_i(0) - a_i p_i(0) - \dot{p}_i(0) \\ &= r_1 - s_1 \end{aligned}$$

Repitiendo el mismo procedimiento para $\ddot{q}_i(t)$ y $\ddot{\dot{q}}_i(t)$ obtenemos:

$$q_i(0) = x_i(0) - p_i(0) \quad (4.6a)$$

$$\dot{q}_i(0) = r_1 - s_1 \quad (4.6b)$$

$$\ddot{q}_i(0) = r_2 - s_2 \quad (4.6c)$$

$$\ddot{\dot{q}}_i(0) = r_3 - s_3 \quad (4.6d)$$

Esto último proporciona un sistema de ecuaciones para encontrar los coeficientes del polinomio $q_i(t)$ dependiendo del polinomio de diferencia $p_i(t)$ y también de la aproximación lineal representada por a_i y $u_i(t)$.

Siguiendo los principios de los métodos QSS, $p_i(t)$ debe ser tal que $|p_i(t)| \leq \Delta Q_i$ en el intervalo $[0, t_1^i]$. Esto conduce a dos situaciones posibles:

1. Al igual que en todos los métodos LIQSS, siempre que sea posible, $p_i(t)$ debe seleccionarse como un polinomio constante de modo que $q_i(t)$ y $x_i(t)$ evolucionen en paralelo y no se requieran pasos adicionales. Esta situación de *equilibrio*, que es la que evita la aparición de oscilaciones espurias en muchos sistemas rígidos, está caracterizada por un polinomio $p_i(t) = c$ con $|c| \leq \Delta Q_i$. En este caso, puede verse fácilmente a partir de la Ec.(4.4) que

$$s_n = a_i^n c \quad (4.7)$$

y el sistema de la Ec.(4.6) resulta:

$$q_i(0) = x_i(0) - c \quad (4.8a)$$

$$\dot{q}_i(0) = r_1 - a_i c \quad (4.8b)$$

$$\ddot{q}_i(0) = r_2 - a_i^2 c \quad (4.8c)$$

$$\ddot{\dot{q}}_i(0) = r_3 - a_i^3 c \quad (4.8d)$$

Recordando que en el algoritmo de orden n se cumple $\frac{d^n q_i}{dt^n} = 0$, resulta $r_n = s_n = a_i^n c$, de donde, asumiendo que $a_i \neq 0$, se tiene

$$c = \frac{r_n}{a_i^n} \quad (4.9)$$

donde la condición $|c| \leq \Delta Q_i$ debe verificarse. Así, la solución de equilibrio para un método de orden n existe si y sólo si

$$\left| \frac{r_n}{a_i^n} \right| \leq \Delta Q_i \quad (4.10)$$

cuando $a_i \neq 0$.

En caso de que $a_i = 0$, según la Ec.(4.2), la situación de equilibrio requiere que $\frac{d^{n-1}u_i}{dt^{n-1}} = \frac{d^n q_i}{dt^n} = 0$, lo cual implica que $r_n = 0$. En tal caso, podemos tomar $q_i(t) = x_i(t)$ ya que la condición de equilibrio se verifica para cualquier valor de c .

Entonces, las trayectorias cuantificadas del estado para la condición de equilibrio pueden caracterizarse como:

$$q_i(t) = \begin{cases} x_i(t) - \frac{r_n}{a_i^n} & \text{si } |r_n| \leq |a_i|^n \Delta Q_i \wedge a_i \neq 0 \\ x_i(t) & \text{si } r_n = a_i = 0 \end{cases} \quad (4.11a)$$

$$\dot{q}_i(t) = a_i q_i(t) + u_i(t) \quad (4.11b)$$

$$\ddot{q}_i(t) = a_i \dot{q}_i(t) + \dot{u}_i(t) \quad (4.11c)$$

donde r_n está definido en la Ec.(4.3).

2. En caso de que no exista una solución de equilibrio para $p_i(t) = c$ con $|c| \leq \Delta Q_i$, definiremos $q_i(t)$ y sus derivadas correspondientes (dependiendo del orden del método) en función de la elección de $p_i(t)$.

Teniendo en cuenta estas consideraciones, desarrollaremos a continuación los algoritmos LIQSS y LIQSS de Chebyshev de orden 1 a 3.

4.2. Derivación de los Métodos LIQSS

En los métodos QSS Linealmente Implícitos (LIQSS), cada trayectoria de estado cuantificado $q_i(t)$ se calcula de manera tal que se acerque hacia el estado $x_i(t)$ partiendo desde una distancia ΔQ_i . Usando la definición de la Ecuación (4.1), un método LIQSS de orden n puede formularse tomando

$$p_i(t) = b_n(t_m - t)^n \quad (4.12)$$

donde $t_m > 0$ es el instante de tiempo en el cual $q_i(t) = x_i(t)$ y b_n es el coeficiente principal de $x_i(t)$ (el cual es desconocido, ya que depende de la elección del estado cuantificado). Notar que la Ec.(4.12) define de manera mucho más simple los métodos de LIQSS haciendo que al final del paso coincidan tanto el valor de q_i como sus derivadas con los correspondientes de x_i , algo que no se lograba en la definición original de [51] y que sólo estaba planteada, y de manera muy complicada, para LIQSS2 en [22].

Las derivadas temporales de $p_i(t)$ son

$$\begin{aligned}\dot{p}_i(t) &= -nb_n(t_m - t)^{n-1} \\ \ddot{p}_i(t) &= n(n-1)b_n(t_m - t)^{n-2} \\ \dddot{p}_i(t) &= -n(n-1)(n-2)b_n(t_m - t)^{n-3}\end{aligned}$$

A partir de estas ecuaciones, podemos reescribir

$$\begin{aligned}p_i(0) &= b_n t_m^n \\ \dot{p}_i(0) &= -n \frac{p_i(0)}{t_m} \\ \ddot{p}_i(0) &= n(n-1) \frac{p_i(0)}{t_m^2} \\ \dddot{p}_i(0) &= -n(n-1)(n-2) \frac{p_i(0)}{t_m^3}\end{aligned}$$

Entonces, a partir de la Ecuación (4.4) resulta

$$s_1 = a_i p_i(0) + \dot{p}_i(0) = a_i p_i(0) - n \frac{p_i(0)}{t_m} \quad (4.13a)$$

$$s_2 = a_i s_1 + \ddot{p}_i(0) = a_i^2 p_i(0) - a_i n \frac{p_i(0)}{t_m} + n(n-1) \frac{p_i(0)}{t_m^2} \quad (4.13b)$$

$$\begin{aligned}s_3 &= a_i s_2 + \dddot{p}_i(0) = a_i^3 p_i(0) - a_i^2 n \frac{p_i(0)}{t_m} + \\ &+ a_i n(n-1) \frac{p_i(0)}{t_m^2} - n(n-1)(n-2) \frac{p_i(0)}{t_m^3}\end{aligned} \quad (4.13c)$$

Evalutando p_i en $t = 0$, la condición

$$|p_i(0)| = |b_n| \cdot t_m^n = \Delta Q_i \quad (4.14)$$

debe verificarse, excepto en el caso de equilibrio de la Ecuación (4.11).

4.2.1. Método LIQSS de primer orden

En el método LIQSS1, tomando $n = 1$ en la Ecuación (4.13a) y recordando que $s_1 = r_1$ (lo que se obtiene de reemplazar $\dot{q}_i(0) = 0$ en la Ec. (4.6b)), resulta

$$s_1 = a_i p_i(0) - \frac{p_i(0)}{t_m(1)} = r_1$$

por lo tanto,

$$\left(\frac{r_1}{p_i(0)} - a_i \right) t_m(1) + 1 = 0$$

Recordando que $|p_i(0)| = \Delta Q_i$ y que $|r_i/a_i| \geq \Delta Q_i$ (de lo contrario se verificaría la condición de equilibrio de la Ecuación (4.10)), el signo de $p_i(0)$ debe ser opuesto al de r_1 (de otro modo

t_m sería negativo). Entonces resulta

$$p_i(0) = -\text{sign}(r_1)\Delta Q_i \quad (4.15)$$

y las ecuaciones para q_i , considerando también el caso de equilibrio de la Ecuación (4.11), son

$$q_i = \begin{cases} x_i - \frac{r_1}{a_i} & \text{si } |r_1| \leq |a_i|\Delta Q_i \wedge a_i \neq 0 \\ x_i & \text{si } r_1 = a_1 = 0 \\ x_i + \text{sign}(r_1)\Delta Q_i & \text{en caso contrario} \end{cases} \quad (4.16)$$

con $r_1 = a_i x_i + u_i$.

4.2.2. Método LIQSS de segundo orden

Tomando $n = 2$ en las Ecs. (4.13b) y usando que $s_2 = r_2$ (lo que se obtiene de reemplazar $\ddot{q}_i(0) = 0$ en la Ec. (4.6c)), resulta

$$s_2 = a_i^2 p_i(0) - 2a_i \frac{p_i(0)}{t_m} + 2 \frac{p_i(0)}{t_m^2} = r_2 \quad (4.17)$$

por lo tanto,

$$\left(\frac{r_2}{p_i(0)} - a_i^2 \right) t_m^2 + 2a_i t_m - 2 = 0 \quad (4.18)$$

Para que esta ecuación tenga una solución real positiva para t_m , el coeficiente principal debe ser positivo, lo cual sucede únicamente cuando r_2 y $p_i(0)$ tienen el mismo signo. Por ello,

$$p_i(0) = \text{sign}(r_2)\Delta Q_i \quad (4.19)$$

y los valores de q_i y $\dot{q}_i$, considerando también el caso de equilibrio de la Ecuación (4.11), son

$$q_i = \begin{cases} x_i - \frac{r_2}{a_i^2} & \text{si } |r_2| \leq |a_i|^2 \Delta Q_i \wedge a_i \neq 0 \\ x_i & \text{si } r_2 = a_i = 0 \\ x_i - \text{sign}(r_2)\Delta Q_i & \text{en caso contrario} \end{cases} \quad (4.20a)$$

$$\dot{q}_i = \begin{cases} a_i q_i + u_i & \text{si } |r_2| \leq |a_i|^2 \Delta Q_i \\ a_i q_i + u_i + 2 \frac{\text{sign}(r_2)\Delta Q_i}{t_m} & \text{en caso contrario} \end{cases} \quad (4.20b)$$

donde t_m es la única solución positiva de (4.18) con $r_2 = a_i^2 x_i + a_i u_i + \dot{u}_i$.

4.2.3. Método LIQSS de tercer orden

Tomando $n = 3$ en las Ecs. (4.13c) y usando que $s_3 = r_3$ (lo que se obtiene de reemplazar $\ddot{q}_i(0) = 0$ en la Ec. (4.6d)), resulta

$$s_3 = a_i^3 p_i(0) - 3a_i^2 \frac{p_i(0)}{t_m} + 6a_i \frac{p_i(0)}{t_m^2} - 6 \frac{p_i(0)}{t_m^3} = r_3 \quad (4.21)$$

por lo tanto,

$$\left(\frac{r_3}{p_i(0)} - a_i^3 \right) t_m^3 + 3a_i^2 t_m^2 - 6a_i t_m + 6 = 0 \quad (4.22)$$

Esta ecuación tiene solución $t_m > 0$ para todo a_i , siempre que el coeficiente principal sea negativo. Por tanto, el signo de $p_i(0)$ debe ser opuesto al de r_3 , resultando

$$p_i(0) = -\text{sign}(r_3) \Delta Q_i \quad (4.23)$$

y los valores de q_i , $\dot{q}_i$ y $\ddot{q}_i$, considerando también el caso de equilibrio de la Ecuación (4.11), son

$$q_i = \begin{cases} x_i - \frac{r_3}{a_i^3} & \text{si } |r_3| \leq |a_i|^3 \Delta Q_i \wedge a_i \neq 0 \\ x_i & \text{si } r_3 = a_i = 0 \\ x_i + \text{sign}(r_3) \Delta Q_i & \text{en caso contrario} \end{cases} \quad (4.24a)$$

$$\dot{q}_i = \begin{cases} a_i q_i + u_i & \text{si } |r_3| \leq |a_i|^3 \Delta Q_i \\ a_i q_i + u_i + 3 \frac{\text{sign}(r_3) \Delta Q_i}{t_m} & \text{en caso contrario} \end{cases} \quad (4.24b)$$

$$\ddot{q}_i = \begin{cases} a_i \dot{q}_i + \dot{u}_i & \text{si } |r_3| \leq |a_i|^3 \Delta Q_i \\ a_i \dot{q}_i + \dot{u}_i - 6 \frac{\text{sign}(r_3) \Delta Q_i}{t_m^2} & \text{en caso contrario} \end{cases} \quad (4.24c)$$

donde t_m es la única solución real y positiva de (4.22) con $r_3 = a_i^3 x_i + a_i^2 u_i + a_i \dot{u}_i + \ddot{u}_i$.

4.2.4. Política de Actualización del Estado Cuantificado

Las Ecs. (4.16), (4.20) y (4.24) proveen las fórmulas para calcular los estados cuantificados en los métodos LIQSS de primer, segundo y tercer orden. Sin embargo, no indican cómo calcular el instante de la próxima actualización del estado cuantificado. Respecto a ello, existen dos alternativas:

- En la formulación original de [51], los pasos LIQSS se realizan inmediatamente después de que el estado cuantificado q_i alcanza al estado x_i . Si la trayectoria del estado cambia de dirección (debido a un cambio en otro estado cuantificado) y comienza a alejarse de

q_i , entonces el paso se realiza cuando la diferencia alcanza $2\Delta Q_i$ (razón por la cual los algoritmos LIQSS tenían el doble de cota de error).

Para evitar esta cota de error mayor, en esta tesis proponemos realizar el paso cuando la diferencia alcanza ΔQ_i en lugar de $2\Delta Q_i$.

- Otra alternativa es realizar los pasos únicamente cuando la diferencia entre el estado y el estado cuantificado alcanza el quantum (independientemente de que $q_i(t)$ alcance o no a $x_i(t)$). De esta forma, pueden realizarse pasos más grandes sin aumentar la cota de error, ya que la diferencia entre x_i y q_i estará igualmente acotada por ΔQ_i .

Llamaremos *LIQSS extendido* (eLIQSS) a los algoritmos que utilizan esta segunda política de actualización.

4.3. Método LIQSS de Chebyshev

La idea central de los métodos LIQSS de Chebyshev (CheQSS) consiste en utilizar un polinomio de diferencia $p_i(t)$ que maximice el intervalo de tiempo durante el cual permanece acotado por $\pm\Delta Q_i$. Este principio, en teoría, permite maximizar el tiempo en que $q_i(t)$ permanece cercano a $x_i(t)$, lo cual a su vez maximiza el intervalo entre pasos sucesivos y reduce el número total de pasos requeridos.

4.3.1. Maximización del tamaño de paso

La base de los métodos CheQSS se establece mediante el resultado del Teorema 1, presentado en el Capítulo 3, que establece una cota superior para el tiempo durante el cual una trayectoria de estado $x_i(t)$ y su correspondiente trayectoria cuantificada $q_i(t)$ pueden permanecer próximas entre sí. A continuación, el siguiente teorema construye un polinomio $q_i(t)$ de grado $n - 1$ (o menor) que verifica dicha cota.

Teorema 2. *Sea $x_i(t)$ un polinomio de grado n con coeficiente principal b_n . Consideremos el polinomio*

$$q_i(t) = x_i(t) - \text{sign}(b_n) \cdot \Delta Q_i \cdot T_n\left(\frac{2t - t_m(n)}{t_m(n)}\right) \quad (4.25)$$

donde $T_n(\cdot)$ denota el polinomio de Chebyshev de grado n y $t_m(n)$ se define como

$$t_m(n) \triangleq \frac{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}}{a_{x_i}^{(n)}(0)} \quad (4.26)$$

siendo $a_{x_i}^{(n)}(0)$ la actividad local instantánea de $x_i(t)$ para $t = 0$. Entonces, $q_i(t)$ cumple:

1. Su grado es menor o igual a $n - 1$.
2. Verifica la condición $|q_i(t) - x_i(t)| \leq \Delta Q_i$ en el intervalo $[0, t_m(n)]$.

Demostración.

1. El coeficiente principal de $T_n(t)$ es 2^{n-1} . Por lo tanto, el coeficiente principal de

$$T_n\left(\frac{2t - t_m(n)}{t_m(n)}\right)$$

resulta

$$c_n = \frac{2^{n-1}2^n}{t_m(n)^n} = \frac{2^{n-1}2^n}{\left(2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}\right)^n} = \frac{(a_{x_i}^{(n)}(0))^n}{\Delta Q_i} = \frac{\left|\frac{1}{n!} \cdot \frac{d^n x_i(t)}{dt^n}\right|}{\Delta Q_i} = \frac{|b_n|}{\Delta Q_i}$$

De este modo, el coeficiente principal de

$$p_i(t) = \text{sign}(b_n) \cdot \Delta Q_i \cdot T_n\left(\frac{2t - t_m(n)}{t_m(n)}\right)$$

es $d_n = \text{sign}(b_n) \Delta Q_i c_n = \text{sign}(b_n) |b_n| = b_n$. Por lo tanto, $x_i(t)$ y $p_i(t)$ comparten el mismo coeficiente principal, y en consecuencia $q_i(t) = x_i(t) - p_i(t)$ tiene grado menor o igual a $n - 1$.

2. Como

$$t \in [0, t_m(n)] \implies \frac{2t - t_m(n)}{t_m(n)} \in [-1, 1] \implies |T_n\left(\frac{2t - t_m(n)}{t_m(n)}\right)| \leq 1$$

entonces

$$t \in [0, t_m(n)] \implies |p_i(t)| = \Delta Q_i \cdot |T_n\left(\frac{2t - t_m(n)}{t_m(n)}\right)| \leq \Delta Q_i$$

lo cual completa la prueba.

□

Notar que de acuerdo al Teorema 1, el máximo intervalo de tiempo que una sección de polinomio de $q_i(t)$ puede permanecer a una distancia no mayor a ΔQ_i de $x_i(t)$ está dado por la expresión de la Ec.(3.8). Por otro lado, el Teorema 2 establece que eligiendo $q_i(t)$ según la Ec.(4.25) dicho polinomio efectivamente permanece a una distancia no mayor a ΔQ_i durante dicho intervalo de tiempo.

Este resultado constituye la base para el diseño de los métodos LIQSS de Chebyshev. Utilizando la definición de la Ec. (4.1) y la Ec. (4.25), un método CheQSS de orden n puede formularse seleccionando

$$p_i(t) = \text{sign}(b_n) \cdot \Delta Q_i \cdot T_n\left(\frac{2t - t_m(n)}{t_m(n)}\right) \quad (4.27)$$

Debe señalarse que $x_i(t)$ depende de $q_i(t)$, por lo que ni b_n ni $t_m(n)$ son conocidos de antemano.

4.3.2. Método LIQSS de Chebyshev de primer orden

Para el método de primer orden, tomando $n = 1$ y recordando que $T_1(z) = z$, podemos escribir a partir de la Ec. (4.27):

$$\begin{aligned} p_i(t) &= \text{sign}(b_1) \Delta Q_i \cdot \frac{2t - t_m(1)}{t_m(1)} \\ \dot{p}_i(t) &= \text{sign}(b_1) \Delta Q_i \cdot \frac{2}{t_m(1)} \end{aligned}$$

Luego, evaluando en $t = 0$ resulta:

$$p_i(0) = -\text{sign}(b_1) \Delta Q_i \quad (4.29a)$$

$$\dot{p}_i(0) = -p_i(0) \frac{2}{t_m(1)} \quad (4.29b)$$

Sustituyendo en la expresión de s_1 , de la Ec. (4.4):

$$s_1 = a_i p_i(0) + \dot{p}_i(0) = a_i p_i(0) - p_i(0) \frac{2}{t_m(1)}$$

Recordando que $s_1 = r_1$, obtenemos:

$$\left(\frac{r_1}{p_i(0)} + a_i \right) t_m(1) + 2 = 0 \quad (4.30)$$

De la Ec. (4.29), podemos observar que $|p_i(0)| = \Delta Q_i$ y que $|r_1/a_i| \geq \Delta Q_i$ (de otra manera la condición de equilibrio de la Ec. (4.10) se verificaría), luego $p_i(0)$ debe tener signo opuesto al de r_1 (de otra manera $t_m(1)$ resultaría negativo). Así,

$$p_i(0) = -\text{sign}(r_1) \Delta Q_i \quad (4.31)$$

Con esto, los valores de q_i , considerando también el caso de equilibrio de la Ec. (4.11), resultan:

$$q_i = \begin{cases} x_i - \frac{r_1}{a_i}, & |r_1| \leq |a_i| \Delta Q_i \wedge a_i \neq 0, \\ x_i, & r_1 = a_i = 0, \\ x_i + \text{sign}(r_1) \Delta Q_i, & \text{en otro caso.} \end{cases} \quad (4.32)$$

con $r_1 = a_i x_i + u_i$. Notar que (4.32) coincide exactamente con (4.16); por lo tanto, eLIQSS1 y CheQSS1 son equivalentes.

4.3.3. Método LIQSS de Chebyshev de segundo orden

Para el algoritmo de segundo orden, tomando $n = 2$ y recordando que $T_2(z) = 2z^2 - 1$, obtenemos a partir de (4.27) las expresiones del polinomio $p_i(t)$ y sus derivadas de la siguiente manera

$$\begin{aligned} p_i(t) &= \text{sign}(b_2) \Delta Q_i \left[2 \left(\frac{2t - t_m(2)}{t_m(2)} \right)^2 - 1 \right] \\ \dot{p}_i(t) &= \text{sign}(b_2) \Delta Q_i \cdot 4 \frac{2t - t_m(2)}{t_m(2)} \cdot \frac{2}{t_m(2)} \\ \ddot{p}_i(t) &= \text{sign}(b_2) \Delta Q_i \cdot \frac{16}{t_m(2)^2} \end{aligned}$$

Evaluando las expresiones para $t = 0$:

$$\begin{aligned} p_i(0) &= \text{sign}(b_2) \Delta Q_i \\ \dot{p}_i(0) &= -\text{sign}(b_2) \Delta Q_i \frac{8}{t_m(2)} \\ \ddot{p}_i(0) &= \text{sign}(b_2) \Delta Q_i \frac{16}{t_m(2)^2} \end{aligned}$$

Luego, sustituyendo estas expresiones en (4.7) obtenemos:

$$\begin{aligned} s_1 &= a_i p_i(0) + \dot{p}_i(0) = a_i p_i(0) - p_i(0) \frac{8}{t_m(2)} \\ s_2 &= a_i s_1 + \ddot{p}_i(0) = a_i^2 p_i(0) - a_i p_i(0) \frac{8}{t_m(2)} + p_i(0) \frac{16}{t_m(2)^2} \end{aligned}$$

Recordando que $s_2 = r_2$, resulta:

$$\left(\frac{r_2}{p_i(0)} - a_i^2 \right) t_m(2)^2 + 8a_i t_m(2) - 16 = 0 \quad (4.35)$$

Esto es una ecuación cuadrática en $t_m(2)$, con discriminante:

$$\Delta = 64a_i^2 + 64 \left(\frac{r_2}{p_i(0)} - a_i^2 \right) = 64 \frac{r_2}{p_i(0)}$$

el cual es positivo únicamente si r_2 y $p_i(0)$ tienen el mismo signo. Por lo tanto,

$$p_i(0) = \text{sign}(r_2) \Delta Q_i$$

y las expresiones finales para q_i y $\dot{q}_i$, considerando el caso de equilibrio de la Ec. (4.11),

son:

$$q_i = \begin{cases} x_i - \frac{r_2}{a_i^2}, & |r_2| \leq |a_i|^2 \Delta Q_i \wedge a_i \neq 0, \\ x_i, & r_2 = a_i = 0, \\ x_i - \text{sign}(r_2) \Delta Q_i, & \text{en otro caso,} \end{cases} \quad (4.36a)$$

$$\dot{q}_i = \begin{cases} a_i q_i + u_i, & |r_2| \leq |a_i|^2 \Delta Q_i, \\ a_i q_i + u_i + 8 \frac{\text{sign}(r_2) \Delta Q_i}{t_m(2)}, & \text{en otro caso.} \end{cases} \quad (4.36b)$$

donde $t_m(2)$ es la única solución positiva de la Ec. (4.35) y $r_2 = a_i r_1 + \dot{u}_i$.

4.3.4. Método LIQSS de Chebyshev de tercer orden

En el algoritmo de tercer orden, tomando $n = 3$ y recordando que $T_3(z) = 4z^3 - 3z$, podemos obtener a partir de la Ec. (4.27) la expresión del polinomio $p_i(t)$ y de sus derivadas, dadas por:

$$p_i(t) = \text{sign}(b_3) \Delta Q_i \cdot \left[4 \left(\frac{2t - t_m(3)}{t_m(3)} \right)^3 - 3 \left(\frac{2t - t_m(3)}{t_m(3)} \right) \right] \quad (4.37a)$$

$$\dot{p}_i(t) = \text{sign}(b_3) \Delta Q_i \cdot \left[12 \left(\frac{2t - t_m(3)}{t_m(3)} \right)^2 - 3 \right] \cdot \frac{2}{t_m(3)} \quad (4.37b)$$

$$\ddot{p}_i(t) = \text{sign}(b_3) \Delta Q_i \cdot \frac{48}{t_m(3)} \cdot \frac{2t - t_m(3)}{t_m(3)} \cdot \frac{2}{t_m(3)} \quad (4.37c)$$

$$\ddot{\dot{p}}_i(t) = \text{sign}(b_3) \Delta Q_i \cdot \frac{96}{t_m(3)^2} \cdot \frac{2}{t_m(3)} \quad (4.37d)$$

Evaluando las Ecs. (4.37) en $t = 0$, se obtiene:

$$p_i(0) = -\text{sign}(b_3) \Delta Q_i \quad (4.38a)$$

$$\dot{p}_i(0) = \text{sign}(b_3) \Delta Q_i \frac{18}{t_m(3)} \quad (4.38b)$$

$$\ddot{p}_i(0) = -\text{sign}(b_3) \Delta Q_i \frac{96}{t_m(3)^2} \quad (4.38c)$$

$$\ddot{\dot{p}}_i(0) = \text{sign}(b_3) \Delta Q_i \frac{192}{t_m(3)^3} \quad (4.38d)$$

$$(4.38e)$$

Luego, reemplazando estas expresiones en s_1, s_2 y s_3 obtenemos:

$$\begin{aligned}
s_1 &= a_i p_i(0) + \dot{p}_i(0) = a_i p_i(0) - p_i(0) \frac{18}{t_m(2)} \\
s_2 &= a_i s_1 + \ddot{p}_i(0) = a_i^2 p_i(0) - a_i p_i(0) \frac{18}{t_m(2)} + p_i(0) \frac{96}{t_m(3)^2} \\
s_3 &= a_i s_2 + \ddot{\dot{p}}_i(0) \\
&= a_i^3 p_i(0) - a_i^2 p_i(0) \frac{18}{t_m(2)} + a_i p_i(0) \frac{96}{t_m(3)^2} - p_i(0) \frac{192}{t_m(3)^3}
\end{aligned} \tag{4.39}$$

Recordando ahora que $s_3 = r_3$, resulta:

$$\left(\frac{r_3}{p_i(0)} - a_i^3 \right) t_m(3)^3 + 18 a_i t_m(3)^2 - 96 t_m(3) + 192 = 0 \tag{4.40}$$

Esta ecuación siempre posee una solución $t_m(3) > 0$ siempre que el coeficiente principal sea negativo. Para ello, el signo de $p_i(0)$ debe ser opuesto al de r_3 , resultando:

$$p_i(0) = -\text{sign}(r_3) \Delta Q_i \tag{4.41}$$

A partir de esto, los valores de $q_i, \dot{q}_i$ y $\ddot{q}_i$, incluyendo también el caso de equilibrio de la Ec. (4.11), quedan determinados por:

$$q_i = \begin{cases} x_i - \frac{r_3}{a_i^3} & \text{si } |r_3| \leq |a_i|^3 \Delta Q_i \wedge a_i \neq 0 \\ x_i & \text{si } r_3 = a_i = 0 \\ x_i + \text{sign}(r_3) \Delta Q_i & \text{en caso contrario} \end{cases} \tag{4.42a}$$

$$\dot{q}_i = \begin{cases} a_i q_i + u_i & \text{si } |r_3| \leq |a_i|^3 \Delta Q_i \\ a_i q_i + u_i - 18 \frac{\text{sign}(r_3) \Delta Q_i}{t_m(3)} & \text{en caso contrario} \end{cases} \tag{4.42b}$$

$$\ddot{q}_i = \begin{cases} a_i \dot{q}_i + \dot{u}_i & \text{si } |r_3| \leq |a_i|^3 \Delta Q_i \\ a_i \dot{q}_i + \dot{u}_i + 96 \frac{\text{sign}(r_3) \Delta Q_i}{t_m(3)^2} & \text{en caso contrario} \end{cases} \tag{4.42c}$$

donde $t_m(3)$ es la única solución real positiva de la Ec. (4.40), con $r_3 = a_i^3 x_i + a_i^2 u_i + a_i \dot{u}_i + \ddot{u}_i$.

4.4. Implementación de los nuevos métodos

Los algoritmos eLIQSS y CheQSS fueron implementados en la herramienta Stand-Alone QSS Solver [27]. Dado que dicha herramienta cuenta con funciones separadas para definir la política de cuantificación y el procedimiento de integración temporal, la implementación de los nuevos algoritmos sólo requirió incorporar las funciones encargadas de calcular $q_i(t)$ y sus

derivadas.

Todos los nuevos métodos se encuentran incluidos en la versión actual del software, disponible en: <https://github.com/CIFASIS/qss-solver>.

4.5. Comportamiento de los métodos LIQSS, eLIQSS y CheQSS

Con el fin de ilustrar el comportamiento de los métodos LIQSS, eLIQSS y CheQSS, simulamos una ODE simple de primer orden dada por

$$\dot{x}(t) = 1 - x(t) \quad (4.43)$$

con un tiempo final $t_f = 5$ y diferentes valores para el tamaño del quantum.

La Figura 4.1 muestra las trayectorias producidas por los diferentes algoritmos utilizando $\Delta Q = 10^{-2}$. La Figura 4.2 también presenta la trayectoria de $p(t) = q(t) - x(t)$ para cada método, donde pueden apreciarse las típicas propiedades de equi-oscilación de los polinomios de Chebyshev.

La Tabla 4.1 muestra el número de pasos de integración requeridos por cada método para distintos valores del quantum, y los compara con el mínimo teórico calculado según la Ec. (3.14).

n	ΔQ	Min. Teórico	CheQSS _n	eLIQSS _n	LIQSS _n
1	10^{-2}	50	51	51	100
	10^{-3}	497	497	497	993
	10^{-4}	4965	4965	4965	9924
2	10^{-2}	5	7	9	15
	10^{-3}	15	17	23	44
	10^{-4}	46	48	67	136
3	10^{-2}	2	4	5	8
	10^{-3}	5	7	9	16
	10^{-4}	10	12	17	33

Tabla 4.1: Número de pasos y sus cotas inferiores en la simulación de la Ec. (4.43).

Podemos observar que, como era de esperar, dado que los métodos eLIQSS y CheQSS de primer orden son idénticos entre sí, ambos requieren exactamente el mismo número de pasos de simulación y coinciden con la cota mínima teórica. En comparación con el método LIQSS original de primer orden, se obtiene una mejora significativa, con aproximadamente un 50 % de reducción en el número de pasos.

Al analizar los pasos realizados por los algoritmos de segundo y tercer orden, volvemos a ver que tanto eLIQSS como CheQSS reducen el número de pasos respecto de los métodos LIQSS no extendidos. Además, el número de pasos de CheQSS es muy cercano a los valores mínimos teóricos. De hecho, realiza siempre dos pasos más que el mínimo teórico ya que necesita dos pasos para estimar correctamente los coeficientes de la aproximación lineal.

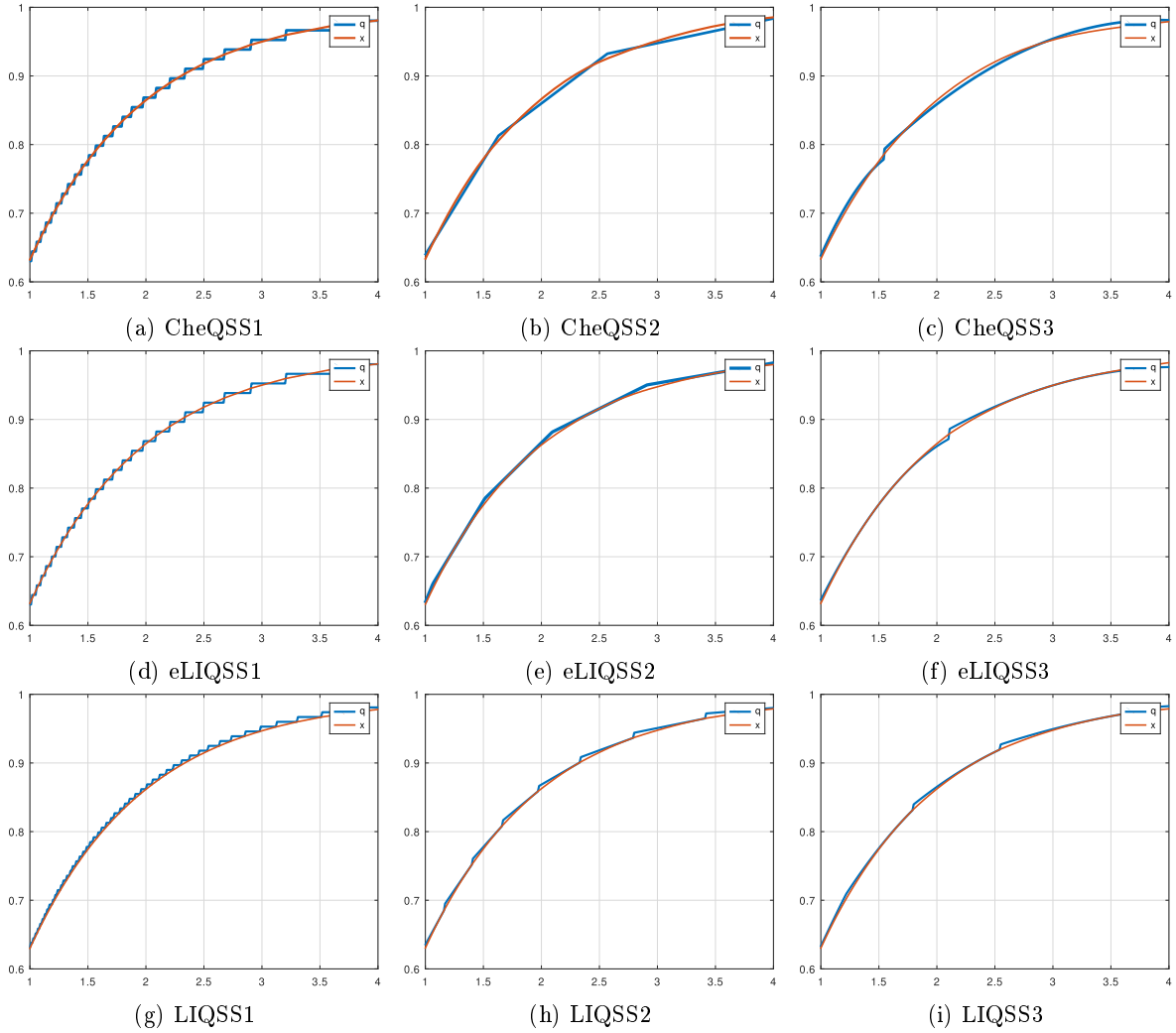


Figura 4.1: Trayectorias del estado y del estado cuantificado de diferentes métodos LIQSS (detalle) en la simulación de la Ec. (4.43).

Cabe aclarar que, en este ejemplo, las características particulares del sistema considerado permiten alcanzar los valores óptimos teóricos. Una de ellas es que el sistema es desacoplado, ya que tiene una única variable de estado. En sistemas con múltiples variables de estado, donde la derivada de un estado x_i depende de otros estados cuantificados q_j , un cambio en alguno de estos últimos puede modificar la derivada de x_i e impedir completar el paso máximo correspondiente a su aproximación.

Asimismo, el sistema del ejemplo es lineal, lo cual contribuye a que los métodos CheQSS alcancen el tamaño máximo teórico de los pasos. En sistemas no lineales, la aproximación lineal utilizada por la familia de métodos LIQSS puede introducir errores que, nuevamente, dificulten alcanzar el paso óptimo previsto por el análisis teórico.

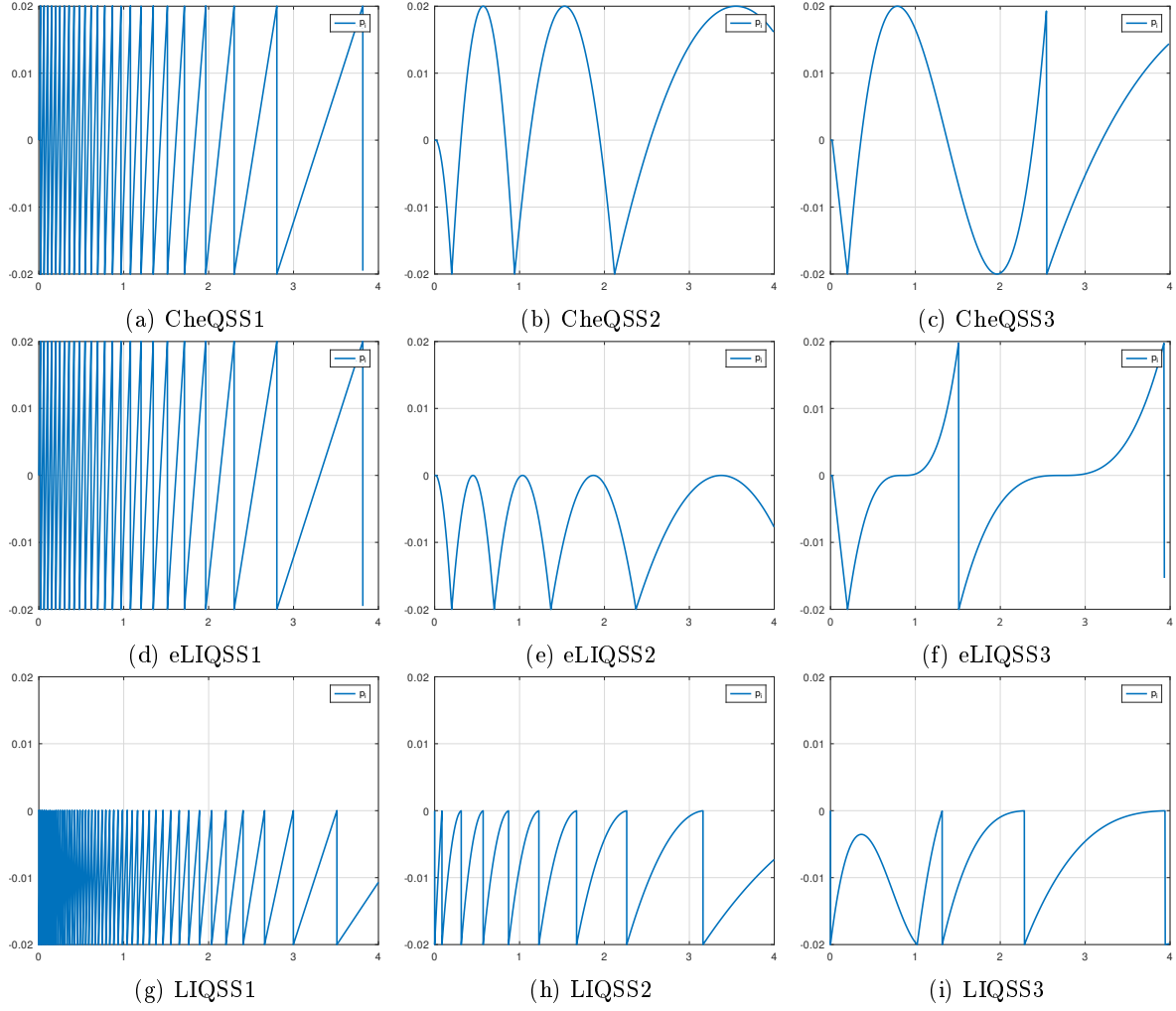


Figura 4.2: Diferencia entre las trayectorias del estado y del estado cuantificado para diferentes métodos LIQSS (detalle) en la simulación de la Ec. (4.43).

4.6. Propiedades de los métodos eLIQSS y CheQSS

Las propiedades teóricas de los métodos QSS respecto de estabilidad práctica, convergencia y cotas de error se derivan de la condición $|q_i(t) - x_i(t)| \leq \Delta Q_i$, que implica que la cuantificación del estado es equivalente a la adición de perturbaciones acotadas.

Dado que todos los algoritmos eLIQSS y CheQSS verifican la condición $|p_i(t)| = |q_i(t) - x_i(t)| \leq \Delta Q_i$ (ver Figura 4.2), todos estos métodos comparten las mismas cotas de error y propiedades de convergencia que los métodos QSS tradicionales.

Capítulo 5

Aplicación a un Modelo de Advección-Difusión-Reacción

En este capítulo aplicamos los conceptos desarrollados en los dos capítulos anteriores a un modelo de Advección-Difusión-Reacción (ADR) ante distintos escenarios resultantes de la variación de los parámetros del modelo.

En primer lugar, analizamos las nociones relacionadas con la actividad y la cantidad de pasos mínimos teóricos requeridos por los métodos que utilizan cada enfoque de simulación (discretización temporal o cuantificación de estados). En el caso de discretización temporal, comparamos los resultados teóricos con los obtenidos tras realizar simulaciones utilizando tres métodos clásicos. En el caso de cuantificación de estados utilizamos los métodos originales de LIQSS de distintos órdenes y sumamos los resultados obtenidos mediante simulaciones utilizando eLIQSS y CheQSS. Esto último permite también poder comprobar la mejora en la eficiencia de los nuevos métodos respecto de los originales.

A continuación, realizamos una variación de parámetros para poder comparar un escenario con actividad homogénea y otro con actividad más heterogénea. Para ambos escenarios, nuevamente aplicamos los conceptos desarrollados de actividad, identificamos teóricamente cuál de los enfoques de simulación resulta más conveniente y luego corroboramos esta conclusión con resultados obtenidos luego de simular ambos escenarios variando las tolerancias y los algoritmos de simulación.

Por último, comparamos con mayor profundidad los métodos originales de LIQSS con las nuevas versiones y con los métodos clásicos más eficientes, teniendo en cuenta no sólo el número de pasos realizados, sino también tiempo requerido de simulación y errores respecto a una solución de referencia.

Cabe destacar que, para las simulaciones presentes en este y en los dos capítulos siguientes, utilizamos el QSS-Solver [27] y el software Octave [24] en una computadora de escritorio con procesador Intel Core i5-10400 @ 2.90GHz, bajo el sistema operativo Ubuntu 22.04.

5.1. Descripción del Modelo

El primer caso de estudio considerado en esta Tesis está dado por el siguiente conjunto de ecuaciones diferenciales ordinarias, que corresponden a la discretización espacial de un modelo unidimensional de advección–difusión–reacción:

$$\dot{u}_i(t) = -A \cdot \frac{u_i(t) - u_{i-1}(t)}{\Delta x} + D \cdot \frac{u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)}{\Delta x^2} + R \cdot (u_i(t)^2 - u_i(t)^3) \quad (5.1)$$

para $i = 2, \dots, N-1$, junto con las siguientes ecuaciones para los extremos de la grilla ($i = 1$ y $i = N$):

$$\dot{u}_1(t) = -A \cdot \frac{u_1(t) - 1}{\Delta x} + D \cdot \frac{u_2(t) - 2u_1(t) + 1}{\Delta x^2} + R \cdot (u_1(t)^2 - u_1(t)^3) \quad (5.2)$$

$$\dot{u}_N(t) = -A \cdot \frac{u_N(t) - u_{N-1}(t)}{\Delta x} + D \cdot \frac{2u_{N-1}(t) - 2u_N(t)}{\Delta x^2} + R \cdot (u_N(t)^2 - u_N(t)^3) \quad (5.3)$$

Los modelos de ADR son muy útiles para describir diversos fenómenos físicos, tales como el transporte de especies químicas o la propagación de incendios, entre otros ejemplos. En esta Tesis, nos centramos en el análisis del desempeño de distintos métodos de simulación aplicados al sistema de EDOs resultante.

Este sistema surge de aplicar el método de líneas a la siguiente ecuación en derivadas parciales:

$$\frac{\partial u(x, t)}{\partial t} = -A \frac{\partial u(x, t)}{\partial x} + D \frac{\partial^2 u(x, t)}{\partial x^2} + R \cdot (u(x, t)^2 - u(x, t)^3) \quad (5.4)$$

donde la variable de estado $u(x, t)$ podría representar la concentración de una especie química en la posición espacial x y en el instante t , mientras que A , D y R son parámetros que corresponden a los coeficientes de advección, difusión y reacción, respectivamente.

Para obtener el sistema de EDOs, las derivadas espaciales se aproximan mediante un esquema de diferencias finitas sobre una grilla uniforme, de modo que cada variable $u_i(t)$ representa una aproximación de $u(x_i, t)$ con $x_i = i \cdot \Delta x$.

En el marco de esta Tesis, el foco del análisis está puesto en la simulación y el comportamiento numérico del sistema de EDOs discretizado, por lo que los detalles específicos del proceso de discretización espacial no constituyen el objeto de estudio y se incluyen únicamente con el propósito de contextualizar el origen físico del modelo considerado.

5.2. Análisis Basado en Actividad

En este caso consideramos los siguientes parámetros para el modelo ADR: $N = 10$; $A = 1$; $D = 0.01$; $R = 10$ y $\Delta x = 10/N$. El sistema será simulado desde $t_0 = 0$ segundos hasta $t_f = 5$ segundos.

5.2.1. Actividades Locales, Globales y Factor de Homogeneidad

Inicialmente, calculamos las actividades locales y globales a partir de la solución numérica obtenida mediante el algoritmo LIQSS2 [51], utilizando una precision muy alta ($\Delta Q_i = 10^{-12}$).

La Figura 5.1a muestra las trayectorias de los estados, mientras que la Figura 5.1b presenta las correspondientes actividades instantáneas locales de orden 1, $a_{x_i}^{(1)}$, calculadas utilizando la Ec. (3.4). En la misma figura se muestra además, la actividad instantánea global de orden 1, $a_{\mathbf{x}}^{(1)}(t)$, que corresponde claramente al máximo valor de la actividad instantánea local en cada instante de tiempo.

Integrando $a_{x_i}^{(n)}$ en el tiempo, de acuerdo con la Ec. (3.5), para los órdenes $n = 1$, $n = 2$ y $n = 3$, obtuvimos los valores de la actividad local de cada variable de estado. Estos valores se presentan en la Tabla 5.1¹.

n	$A_1^{(n)}$	$A_2^{(n)}$	$A_3^{(n)}$	$A_4^{(n)}$	$A_5^{(n)}$	$A_6^{(n)}$	$A_7^{(n)}$	$A_8^{(n)}$	$A_9^{(n)}$	$A_{10}^{(n)}$
1	1	1	1	1	1	1	1	1	1	1
2	1.150	1.579	1.650	1.678	1.692	1.699	1.704	1.706	1.708	1.705
3	1.712	1.899	2.162	2.247	2.291	2.317	2.335	2.347	2.341	2.328

Tabla 5.1: Actividades locales de orden 1 a 3 para los estados del modelo ADR

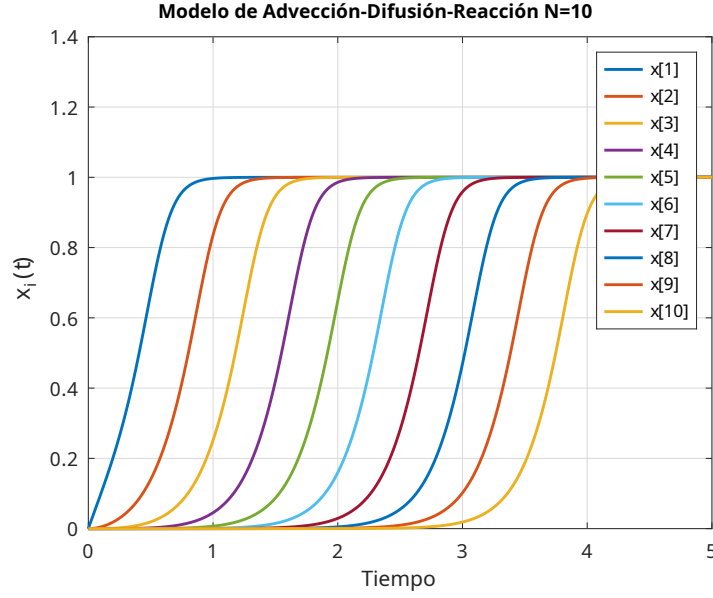
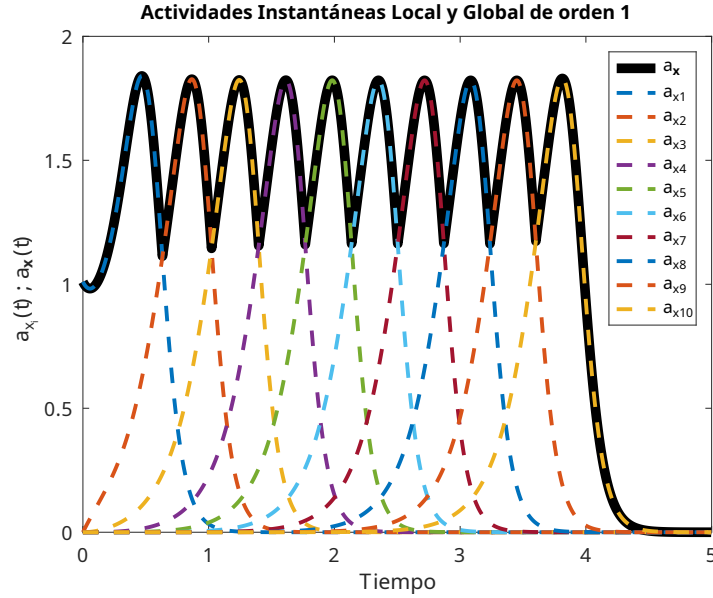
De manera análoga, integrando la actividad instantánea global $a_{\mathbf{x}}^{(n)}(t)$ según la Ec. (3.7), calculamos las actividades globales de orden 1 a 3, obteniendo los siguientes valores:

$$A_{\mathbf{x}}^{(1)} = 6.2226; A_{\mathbf{x}}^{(2)} = 6.2144; A_{\mathbf{x}}^{(3)} = 7.5833$$

A partir de estos resultados, calculamos los Factores de Homogeneidad $H_{\mathbf{x}}^{(n)}$ de órdenes 1 a 3 de acuerdo con la Ec. (3.19), cuyos valores se reportan en la Tabla 5.2. En dicha tabla también se incluyen los productos $H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$, utilizando los siguientes valores de $r^{(n)}$ obtenidos según la Ec. (3.22):

- $r^{(1)} = 2.8$,
- $r^{(2)} = 2.82$,
- $r^{(3)} = 2.82$

¹Notar que la actividad local de orden 1 de cada estado es igual a 1, ya que todas las trayectorias crecen monótonamente de 0 a 1, experimentando un cambio absoluto de magnitud unitaria.

(a) Solución numérica del sistema ADR con $N = 10$ 

(b) Actividades instantáneas locales y globales de orden 1

Figura 5.1: Trayectorias de estado y actividades instantáneas (local y global) del modelo ADR

n	$H_{\mathbf{x}}^{(n)}$	$H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$
1	0.1606	0.4497
2	0.2618	0.7395
3	0.2898	0.8161

Tabla 5.2: Factores de Homogeneidad - Modelo ADR con $N = 10$

Podemos observar que el Factor de Homogeneidad (HF) es relativamente bajo (es decir, las trayectorias del sistema presentan una actividad heterogénea) y que además $H_{\mathbf{x}}^{(n)} \cdot r^{(n)} < 1$, lo que sugiere que los algoritmos QSS serían los más eficientes en este caso.

5.2.2. Número mínimo de pasos y actualizaciones

A partir de las actividades locales y globales, estimamos las cotas inferiores para el número de pasos requeridos por los algoritmos QSS y los algoritmos clásicos, de acuerdo con las Ecs. (3.14) y (3.17), utilizando diferentes configuraciones de quantum y precisión. Asimismo, simulamos el modelo utilizando algoritmos LIQSS, eLIQSS y CheQSS de orden 1 a 3 (optamos por algoritmos linealmente implícitos debido a la rigidez del sistema) y algoritmos clásicos implícitos de paso variable de orden 1 a 3. En particular, usamos Backward Euler (BE), la Regla Trapezoidal (TR) y un método Backward Runge-Kutta de tercer orden (BRK3). En todos los casos, el tamaño de paso se controló estimando el error mediante un método de un orden inferior de aproximación, dando lugar a los métodos denominados BE01, TR12 y BRK23.

La Tabla 5.3 presenta las estimaciones de las cotas inferiores y los números reales de pasos efectuados por los algoritmos correspondientes.

ΔQ	LIQSS1		LIQSS2		LIQSS3	
	$k_{QS}^{(1)}$	Real	$k_{QS}^{(2)}$	Real	$k_{QS}^{(3)}$	Real
10^{-2}	500	1008	58	272	33	180
10^{-3}	5000	9976	182	797	70	310
10^{-4}	50000	99746	576	2388	150	607
ΔQ	eLIQSS1		eLIQSS2		eLIQSS3	
	$k_{QS}^{(1)}$	Real	$k_{QS}^{(2)}$	Real	$k_{QS}^{(3)}$	Real
10^{-2}	500	510	58	169	33	146
10^{-3}	5000	5010	182	335	70	210
10^{-4}	50000	49989	576	866	150	360
ΔQ	CheQSS1		CheQSS2		CheQSS3	
	$k_{QS}^{(1)}$	Real	$k_{QS}^{(2)}$	Real	$k_{QS}^{(3)}$	Real
10^{-2}	500	510	58	156	33	147
10^{-3}	5000	5010	182	307	70	272
10^{-4}	50000	49989	576	804	150	527
ΔQ	BE01		TR12		BRK23	
	$k_{DT}^{(1)}$	Real	$k_{DT}^{(2)}$	Real	$k_{DT}^{(3)}$	Real
10^{-2}	312	657	22	75	12	50
10^{-3}	3113	6556	70	216	24	96
10^{-4}	31130	65535	220	657	52	195

Tabla 5.3: Cota inferior y número real de pasos utilizando diferentes métodos numéricos. Modelo ADR $N = 10$

En las simulaciones realizadas con el método original LIQSS, el número de pasos ejecutados resultó superior a las cotas inferiores estimadas. Esto se debe a que los algoritmos de dicho método no siguen las trayectorias óptimas establecidas en el Teorema 1, donde la diferencia entre las trayectorias de orden n y $n - 1$ está dada por un polinomio de Chebyshev, tal como se demuestra en el Lema 1.

Por otro lado, en las simulaciones efectuadas con los métodos eLIQSS y CheQSS se observa una menor discrepancia entre el número real de pasos y la cota inferior teórica. En los

algoritmos de primer orden, la cantidad de pasos ejecutada resulta prácticamente coincidente con el límite mínimo, mientras que en los algoritmos de segundo y tercer orden persiste una diferencia. Esta diferencia se debe a que la longitud total de los pasos óptimos no siempre puede alcanzarse, ya que el sistema no es lineal y los cambios en las derivadas obligan a actualizar la variable antes de completar el largo del paso óptimo.

5.2.3. Factor de homogeneidad y actualizaciones de estado

El Factor de Homogeneidad reportado en la Tabla 5.2, así como su producto con el grado de conectividad $r^{(n)}$, sugieren que los métodos QSS deberían realizar un número menor de actualizaciones de estado que los algoritmos clásicos. Para corroborar esta hipótesis, calculamos las actualizaciones de estado reales de los distintos algoritmos multiplicando el número de pasos de cada variable por su conectividad r_i en el caso de los métodos QSS, y por el tamaño del vector de estado $N = 10$ en el caso de los métodos clásicos. Los resultados correspondientes se presentan en la Tabla 5.4.

	Resultados teóricos	Resultados de simulación			
n	$H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$	ΔQ	$\frac{u_{QSS}}{u_{DT}}$		
			LIQSS	eLIQSS	CheQSS
1	0.4497	10^{-2}	0.430	0.221	0.221
		10^{-3}	0.426	0.214	0.214
		10^{-4}	0.426	0.214	0.214
2	0.7395	10^{-2}	1,016	0.628	0.580
		10^{-3}	1,043	0.434	0.400
		10^{-4}	1,028	0.371	0.343
3	0.8161	10^{-2}	1,004	0.818	0.824
		10^{-3}	0.907	0.613	0.795
		10^{-4}	0.877	0.515	0.760

Tabla 5.4: Relación entre las actualizaciones de estado de los distintos algoritmos LIQSS y los métodos clásicos de integración numérica

En la tabla podemos observar que en el caso de los algoritmos de LIQSS de orden 1 y en los casos de mayor precisión de orden 3, se corrobora que fueron más eficientes que los algoritmos BE01 y BRK23 respectivamente. Sin embargo en el caso de LIQSS2 y la simulación de menor precisión de LIQSS3, el número de actualizaciones realizadas fue pareja para ambos algoritmos. Esto es esperable ya que los métodos de LIQSS, al terminar cada paso cuando x_i alcanza a q_i , no optimizan el tamaño del paso y de hecho este tamaño es generalmente la mitad de lo que pueden lograr los eLIQSS (que esperan a que x_i y q_i se separen nuevamente a una distancia ΔQ_i).

Es importante volver a destacar que la relación $H_{\mathbf{x}}^{(n)} \cdot r^{(n)} < 1$ no implica que un método basado en cuantificación vaya a producir siempre un número menor de actualizaciones que un método basado en discretización temporal. La interpretación correcta es que la cota inferior

teórica para el número de actualizaciones en los métodos basados en cuantificación es menor que la cota inferior correspondiente en los métodos de tiempo discreto. Un razonamiento análogo se aplica al caso $H_{\mathbf{x}}^{(n)} \cdot r^{(n)} > 1$.

Luego, en los casos de eLIQSS y CheQSS, en el que los algoritmos fueron optimizados con el objetivo de disminuir la cantidad de pasos de simulación, puede observarse que la ventaja en la práctica fue incluso mayor que la predicha por el producto $H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$.

5.2.4. Variación de parámetros

A continuación, consideramos dos escenarios extremos para el mismo modelo ADR de las Ecs. (5.1)–(5.3), utilizando dos conjuntos de parámetros diferentes:

- **Escenario 1:** $N = 10$, $R = 1000$, $D = 0.1$, $A = 1$ y $\Delta x = 10/N = 1$. Período de simulación: $(t_0 = 0, t_f = 0.6)$.
- **Escenario 2:** $N = 10$, $R = 0.1$, $D = 2$, $A = 1$ y $\Delta x = 10/N = 1$. Período de simulación: $(t_0 = 0, t_f = 25)$.

Las Figuras 5.2a y 5.2b muestran las trayectorias del sistema para ambos escenarios, mientras que las Figuras 5.2c y 5.2d presentan las actividades instantáneas y globales de orden 2 correspondientes.

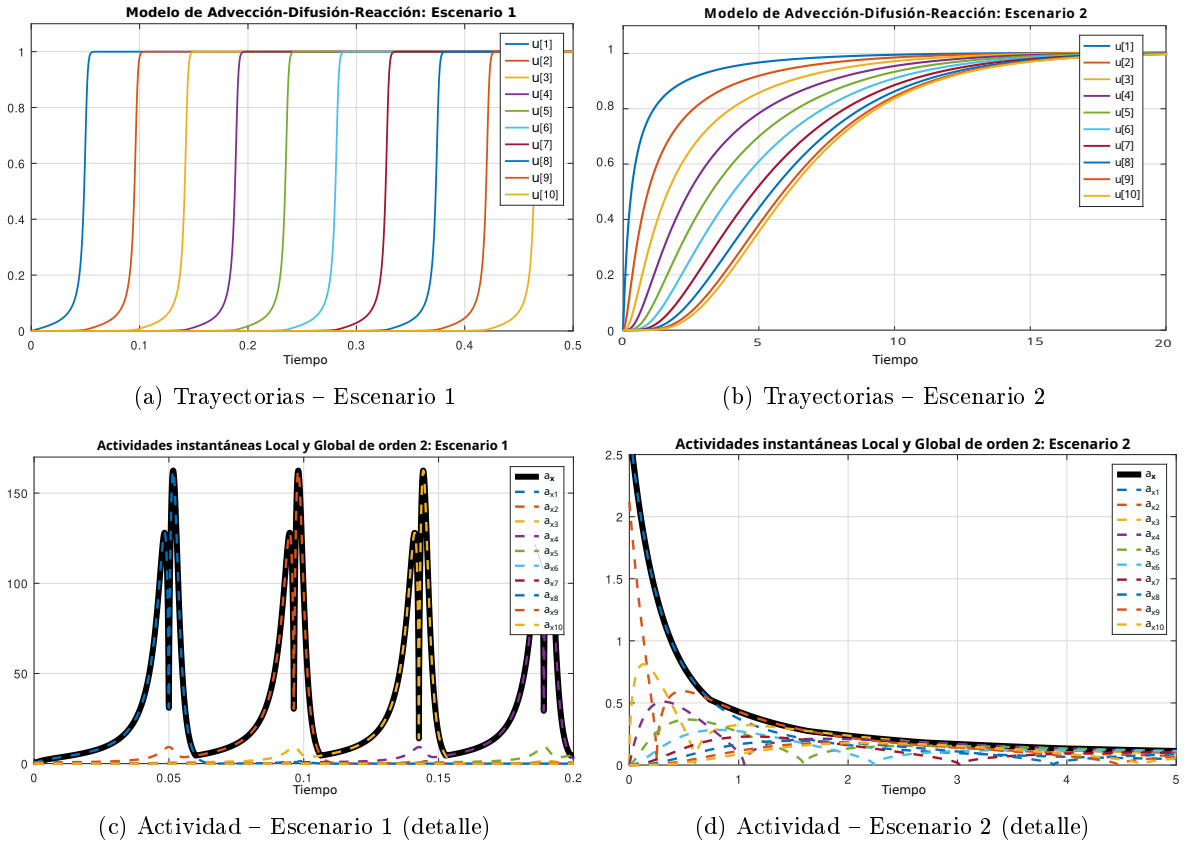


Figura 5.2: Trayectorias y actividades para distintos parámetros del modelo ADR.

En el primer caso, las trayectorias presentan una actividad heterogénea, lo cual puede observarse claramente tanto en las trayectorias de los estados (Fig. 5.2a) como en las actividades instantáneas (Fig. 5.2c). Por el contrario, en el segundo caso las trayectorias muestran actividades homogéneas.

Calculamos las actividades locales y globales de orden 2 según las Ecs. (3.5) y (3.7) y a partir de estos valores determinamos los Factores de Homogeneidad, utilizando la Ec. (3.19) y el producto $H_{\mathbf{x}}^{(2)} \cdot r^{(2)}$ (con $r^{(2)} = 2.8$ en ambos escenarios). Los resultados se presentan en la Tabla 5.5.

Escenario	$H_{\mathbf{x}}^{(2)}$	$H_{\mathbf{x}}^{(2)} \cdot r^{(2)}$
1	0.109	0.3052
2	0.567	1.5876

Tabla 5.5: Factores de Homogeneidad - Modelos ADR con variación de parámetros.

Estos resultados indican que es esperable que los métodos de QSS resulten más eficientes en el primer escenario (trayectorias heterogéneas), mientras que los métodos clásicos deberían presentar un mejor desempeño en el segundo escenario (trayectorias homogéneas).

Para corroborar esto, realizamos simulaciones de ambos escenarios para diferentes valores del quantum ΔQ , comparando los resultados de un algoritmo de discretización temporal (TR12) y de los distintos métodos de orden 2 basados en cuantificación de estados (LIQSS2, eLIQSS2 y CheQSS). La Tabla 5.6 muestra los resultados de simulación obtenidos junto con las cotas teóricas inferiores para el número de pasos y actualizaciones.

Método	Tol. abs. (ΔQ)	Escenario 1				Escenario 2			
		Pasos		Actualizaciones		Pasos		Actualizaciones	
		$k_{\text{QS}}^{(2)}$	Real	$u_{\text{QS}}^{(2)}$	Real	$k_{\text{QS}}^{(2)}$	Real	$u_{\text{QS}}^{(2)}$	Real
LIQSS2	10^{-2}	68	371	191	1039	56	611	157	1713
	10^{-3}	215	1109	602	3089	177	809	496	2271
	10^{-4}	678	3608	1899	10121	560	1569	1568	4392
eLIQSS2	10^{-2}	68	228	191	640	56	590	157	1660
	10^{-3}	215	420	602	1179	177	759	496	2126
	10^{-4}	678	1123	1899	3150	560	1331	1568	3721
CheQSS2	10^{-2}	68	207	191	590	56	588	157	1658
	10^{-3}	215	422	602	1182	177	744	496	2091
	10^{-4}	678	1062	1899	2985	560	1226	1568	3432
		$k_{\text{DT}}^{(2)}$	Real	$u_{\text{DT}}^{(2)}$	Real	$k_{\text{DT}}^{(2)}$	Real	$u_{\text{DT}}^{(2)}$	Real
TR12	10^{-2}	62	225	620	2250	10	37	100	370
	10^{-3}	196	628	1960	6280	32	102	320	1020
	10^{-4}	620	1870	6200	18700	100	305	1000	3050

Tabla 5.6: Pasos y actualizaciones de simulación y sus cotas inferiores teóricas. Modelo ADR con variación de parámetros

Estos resultados confirman las expectativas teóricas: todos los métodos de la familia de QSS muestran un mejor desempeño en el primer escenario, mientras que el algoritmo TR12 presenta un rendimiento superior en el segundo.

Una observación interesante es que, en el primer escenario, el número de actualizaciones de estado observadas con CheQSS2 es menor que la cota inferior teórica correspondiente al algoritmo de tiempo discreto $u_{DT}^{(2)}$. Esto implica que no es posible construir un algoritmo de tiempo discreto de segundo orden que realice menos actualizaciones de estado que las producidas por CheQSS2 en este caso.

Cabe destacar también, que la actividad heterogénea se asocia a valores elevados del parámetro de reacción R , que provoca que cada estado experimente un cambio abrupto de 0 a 1. Por el contrario, el comportamiento homogéneo está vinculado a valores grandes del término de difusión D . De esta manera, no resulta necesario calcular el Factor de Homogeneidad para cada conjunto de parámetros, ya que la conveniencia de emplear un enfoque QSS o clásico puede inferirse a partir de esta observación.

Por último, resulta pertinente aclarar que en esta sección se presentaron distintos experimentos con el objetivo de corroborar la correspondencia entre los resultados teóricos y prácticos únicamente para dos escenarios diferentes del modelo ADR unidimensional. A partir del análisis teórico desarrollado, podría inferirse que al considerar extensiones a dos o tres dimensiones espaciales, donde la conectividad entre las variables es mayor y, en consecuencia, también lo es el factor de conectividad $r^{(n)}$, la ventaja de los métodos QSS podría verse reducida.

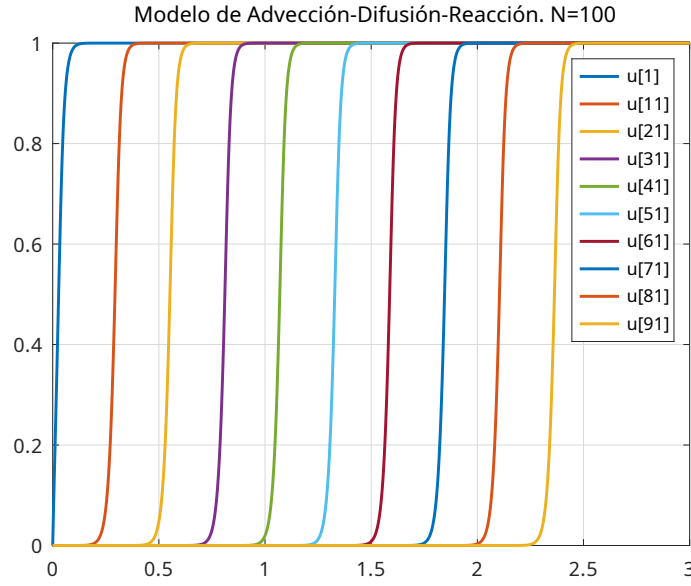
5.3. Eficiencia de los métodos

En esta sección simulamos el mismo modelo ADR descrito anteriormente desde $t_0 = 0$ hasta $t_f = 3$, considerando una $N = 100$ puntos, una longitud total $L = 10$ y un paso espacial de $\Delta x = 10/N$. Los parámetros utilizados en este caso son: $A = 1$, $D = 0.1$ y $R = 100$. La Figura 5.3 ilustra un subconjunto de las trayectorias de los estados del sistema.

Realizamos diferentes experimentos con el fin de comparar el desempeño de los nuevos algoritmos (eLIQSS y CheQSS) con el correspondiente a la definición original de LIQSS y también con el solver clásico CVODE-BDF. Entre los métodos clásicos de tiempo discreto implementados en el Stand Alone QSS solver (DOPRI, CVODE, IDA y DASSL), el solver que presentó el mejor rendimiento fue CVODE-BDF, debido a la naturaleza rígida del problema y al uso de una representación dispersa para la matriz Jacobiana.

En los diferentes experimentos variamos los valores de tolerancia utilizando los pares $(\Delta Q_{\text{rel}}, \Delta Q_{\text{abs}}) = (10^{-2}, 10^{-4}), (10^{-3}, 10^{-5}), (10^{-4}, 10^{-6})$. También modificamos el orden de los métodos QSS, analizando el comportamiento de los algoritmos de primer, segundo y tercer orden.

La Tabla 5.7 y la Figura 5.4 muestran el número total de pasos realizados por cada método y el tiempo promedio de CPU requerido para completar la simulación (calculado como la

Figura 5.3: Trayectorias del modelo ADR con $N = 100$.

media de 10 corridas independientes). Además, se incluyen dos columnas adicionales para mostrar el error de cada simulación medido contra una solución de referencia de alta precisión, obtenida mediante la simulación del modelo con CheQSS2 utilizando tolerancias muy estrictas $(\Delta Q_{\text{rel}}, \Delta Q_{\text{abs}}) = (10^{-10}, 10^{-12})$. Los errores reportados se calcularon como el valor promedio del Error Absoluto Medio (MAE por sus siglas en inglés) de cada variable de estado.

n	Rel. Tol.	Abs. Tol.	N° de pasos			Tiempo de CPU (ms)			Promedio de			MAE
			CheQSS _n	eLIQSS _n	LIQSS _n	CheQSS _n	eLIQSS _n	LIQSS _n	CheQSS _n	eLIQSS _n	LIQSS _n	
1	10^{-2}	10^{-4}	28701	28701	56464	4.4	4.6	8.8	1.8e-04	1.8e-04	2.2e-03	
	10^{-3}	10^{-5}	280812	280812	559419	41.6	41.2	85.3	2.2e-05	2.2e-05	2.3e-04	
	10^{-4}	10^{-6}	2801858	2801858	5589295	408.4	405.0	818.5	2.7e-06	2.7e-06	2.3e-05	
2	10^{-2}	10^{-4}	3173	3644	4324	1.3	1.4	1.6	3.4e-04	5.2e-04	5.9e-04	
	10^{-3}	10^{-5}	8211	9892	13009	3.0	3.4	4.6	6.8e-05	3.1e-05	5.7e-05	
	10^{-4}	10^{-6}	23510	28617	41124	7.9	9.5	13.9	8.6e-06	4.4e-06	5.8e-06	
3	10^{-2}	10^{-4}	3345	2548	5956	3.1	2.3	5.2	2.8e-04	3.7e-04	2.7e-04	
	10^{-3}	10^{-5}	5995	4012	9183	5.6	3.5	7.9	3.4e-05	3.3e-05	3.7e-05	
	10^{-4}	10^{-6}	12142	7131	16050	10.6	6.1	13.5	4.6e-06	2.1e-06	4.2e-06	
CVODE												
5	10^{-2}	10^{-4}	314			8.1			1.0e-03			
	10^{-3}	10^{-5}	414			8.2			2.4e-04			
	10^{-4}	10^{-6}	649			10.0			2.0e-05			
	10^{-5}	10^{-7}	870			15.5			2.0e-06			

Tabla 5.7: Número de pasos, tiempo de CPU y error en la simulación del modelo ADR variando las tolerancias.

A partir de estos resultados podemos extraer las siguientes observaciones:

- Tanto los métodos LIQSS extendido como los métodos LIQSS de Chebyshev requirieron, en todos los casos, menos pasos de integración y menor tiempo de simulación que el método LIQSS estándar bajo los mismos valores de tolerancia. Esta mejora se mantiene de manera consistente en todos los órdenes evaluados.

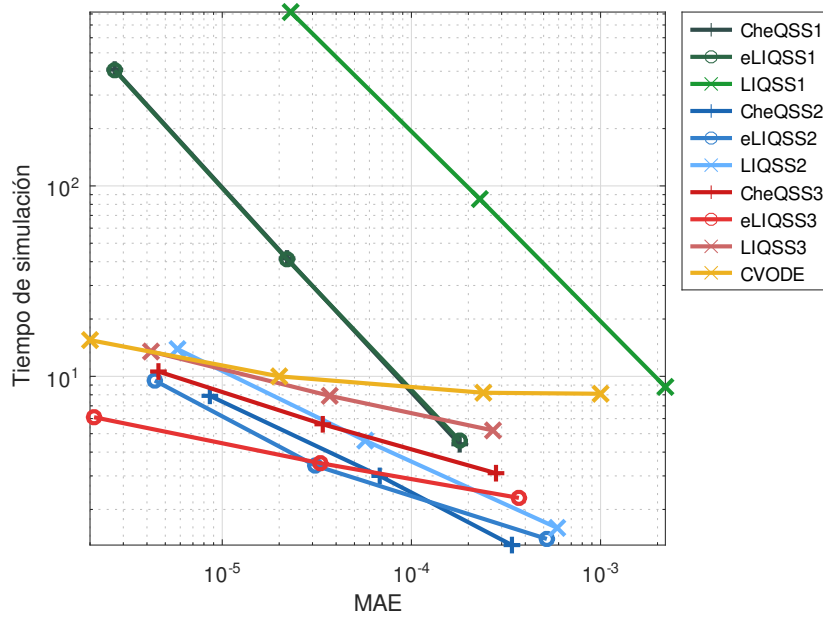


Figura 5.4: Errores y tiempos de simulación resultantes de los distintos algoritmos en la simulación del modelo ADR

- CVODE solo resultó más rápido que los algoritmos QSS de primer orden. Para los órdenes 2 y 3, todos los métodos basados en QSS superaron a CVODE en tiempo de simulación bajo configuraciones de tolerancia equivalentes.
- En cuanto a los errores, todos los métodos QSS exhibieron un comportamiento similar, con el MAE tomando valores del mismo orden de magnitud que la tolerancia absoluta ΔQ_{abs} y dos órdenes de magnitud por debajo de la tolerancia relativa ΔQ_{rel} . La única excepción es LIQSS1 original, donde los errores son un orden de magnitud mayores. Esto puede explicarse del siguiente modo: dado que $x_i(t)$ es estrictamente creciente, se sigue que $q_i(t)$ es siempre mayor que $x_i(t)$, introduciendo una perturbación siempre positiva. En los demás algoritmos, la perturbación cambia de signo a medida que avanza la simulación porque $p_i(t)$ está centrado en cero (en los algoritmos CheQSS) y/o porque las derivadas de orden superior de x_i cambian de signo (ver la solución en la Fig. 5.3).
- CVODE presentó errores similares a los de los LIQSS originales, es decir, un orden de magnitud mayores que los de eLIQSS y CheQSS ante idénticas configuraciones de la tolerancia.
- Una comparación directa entre eLIQSS y CheQSS revela lo siguiente:
 - Las versiones de primer orden se comportan exactamente igual, ya que los algoritmos son idénticos, tal como se anticipó en el Capítulo 4.
 - Para los algoritmos de orden superior (2 y 3), los errores obtenidos con ambos métodos son del mismo orden de magnitud en todos los casos evaluados. En segundo

orden, CheQSS resulta más eficiente, requiriendo menos pasos y tiempos de simulación menores aunque los errores son casi el doble que los de los eLIQSS. En tercer orden, eLIQSS realiza menos pasos y tiene menos error.

- Los métodos CheQSS fueron diseñados para realizar los pasos más largos posibles maximizando el tiempo durante el cual q_i se mantiene cercano a x_i . Sin embargo, dicho tiempo máximo solo se garantiza en sistemas lineales de primer orden. En presencia de no linealidades o cuando entre dos pasos la trayectoria del estado x_i cambia su derivada debido a un cambio en otro estado cuantificado q_j , los pasos pueden resultar más cortos. Por esta razón, en algunos escenarios eLIQSS realiza menos pasos que CheQSS.

Capítulo 6

Aplicación a una Red Neuronal Pulsante

En este capítulo aplicamos los conceptos desarrollados al modelo de una red neuronal pulsante. Para esto, comenzamos con una breve introducción sobre los distintos modelos que son mayormente utilizados para representar este tipo de neuronas y las resultantes redes neuronales. A continuación, describimos el modelo de la red neuronal elegido, sobre el que luego aplicamos los conceptos teóricos de actividad y número de pasos mínimos. Luego corroboramos las conclusiones teóricas con resultados obtenidos mediante simulaciones. Finalmente comparamos la eficiencia de los distintos métodos numéricos.

6.1. Modelos de neuronas y redes neuronales

Las neuronas pulsantes (*spiking neurons*) suelen describirse mediante sistemas de ecuaciones diferenciales ordinarias que presentan discontinuidades asociadas a los tiempos de disparo o *spikes*. Entre los diferentes modelos propuestos para representar este comportamiento, podemos mencionar el Modelo de Hodgkin–Huxley [37], desarrollado en la década de 1950, que tiene bastante precisión biológica pero un elevado costo computacional ya que cada neurona se representa mediante cuatro ecuaciones diferenciales y diez parámetros. Por esta razón, su aplicación suele restringirse a redes formadas por pocas neuronas [38]. Otro tipo de modelo, introducido en el 1907 por Lapique, es el Leaky Integrate and Fire (LIF) [48, 60], que luego tuvo diferentes extensiones [26, 38, 28]. El modelo LIF es más simplificado y se basa en una analogía eléctrica: el circuito equivalente está compuesto por una resistencia y un capacitor en paralelo, que representan la fuga y la capacitancia de la membrana, respectivamente [16]. El capacitor de la membrana se carga hasta alcanzar un umbral predefinido, momento en el cual se descarga generando un disparo o *spike*.

Al simular estos modelos mediante métodos numéricos clásicos [15] se presentan las dificultades ya mencionadas asociadas a la presencia de discontinuidades en los tiempos de disparo

y recepción de los *spikes*, ya que deben detectar cada discontinuidad, avanzar el tiempo de simulación hasta el instante correspondiente y reiniciar la integración a partir de la nueva condición [19].

Este proceso, que en general resulta computacionalmente costoso, se agrava al aumentar el número de neuronas en el modelo, ya que esto provoca que la tasa de ocurrencia de discontinuidades (es decir, los *spikes*) crezca, lo que obliga a reducir el tamaño máximo del paso de integración a medida que crece la red.

Estas limitaciones pueden verse subsanadas, como veremos en las próximas secciones, utilizando algoritmos basados en la cuantificación de estados, como los métodos de QSS.

6.2. Modelo utilizado

En este caso, utilizamos un modelo de una red neuronal pulsante (*Spiking Neural Network*, SNN), donde cada neurona se describe mediante un modelo de tipo *Leaky Integrate-and-Fire* basado en los modelos propuestos en [59, 12]. El estado de la i -ésima neurona está caracterizado por dos variables: el potencial de membrana $V_i(t)$ y la corriente sináptica $I_i^s(t)$. Su evolución temporal, en ausencia de eventos, está representada por el siguiente conjunto de ecuaciones diferenciales:

$$\begin{aligned} \frac{dI_i^s(t)}{dt} &= -\frac{I_i^s(t)}{\tau_s} \\ \frac{dV_i(t)}{dt} &= -\frac{V_i(t) - E_L}{\tau_m} + \frac{I_i^s(t)}{C_m} \end{aligned} \quad (6.1)$$

para $i = 1, \dots, N_e$, donde N_e denota el número total de neuronas en la red, E_L es el potencial de fuga, τ_m es la constante de tiempo de la membrana, C_m es la capacitancia de la membrana y τ_s es la constante de tiempo de la corriente postsináptica.

Este modelo también incorpora discontinuidades. En particular, cuando $V_i(t)$ alcanza el umbral de disparo θ , la neurona i emite una spike, su potencial se reinicia al valor V_r , y la señal de disparo se transmite a sus neuronas post-sinápticas. Este evento desencadena una actualización instantánea de las corrientes sinápticas de las neuronas receptoras. La regla de actualización depende de si la neurona que emitió el pulso es excitatoria :

$$I_j^s(t^+) \leftarrow I_j^s(t) + J_i^{\text{exc}}, \quad (6.2)$$

o inhibitoria:

$$I_j^s(t^+) \leftarrow I_j^s(t) - J_i^{\text{inh}}, \quad (6.3)$$

Los parámetros J_i^{exc} y J_i^{inh} representan las intensidades sinápticas. Para la intensidad

sináptica de cada neurona excitatoria seleccionamos aleatoriamente valores considerando una distribución uniforme en el intervalo $J \pm 0.1J$, donde J es la intensidad sináptica media. Por otro lado, para seleccionar los valores de las intensidades sinápticas inhibitorias consideramos una distribución uniforme en el intervalo $gJ \pm 0.1gJ$, utilizando un parámetro de escala $g = 5$. Este parámetro de escala está elegido para que la tasa de spikes emitidos no varíe significativamente con el nivel de conectividad de la red.

En el modelo, consideramos una red de $N_e = 1000$ neuronas, de las cuales 800 son de tipo excitatorio y las 200 restantes de tipo inhibitorio. Tenemos en cuenta un patrón de conectividad aleatoria en donde cada neurona recibe spikes provenientes de $m = 10$ conexiones sinápticas seleccionadas aleatoriamente, con un 80 % de estas conexiones originadas en neuronas excitatorias y el 20 % restante en neuronas inhibitorias.

Los estados iniciales se eligieron aleatoriamente con distribuciones uniformes $V(t = 0) \sim U[-65, -64]$ mV e $I_i^s(t = 0) \sim U[0.4, 0.5]$ nA.

Además de las spikes recibidas desde otras neuronas de la red, consideramos que cada neurona recibe una entrada independiente. En el análisis de actividad y estimación de costo computacional de la sección 6.3, consideramos una corriente constante de entrada I_e ; mientras que para la comparación de eficiencia de métodos en la sección 6.4, consideramos que cada neurona recibe un tren de pulsos de entrada independiente que sigue un proceso de Poisson con una tasa estacionaria.

Los valores y definiciones de los parámetros del modelo están resumidos en la Tabla 6.1.

Parámetros del modelo	
Valor	Descripción
$\tau_m = 10$ ms	constante de tiempo de la membrana
$\tau_r = 2$ ms	período refractario absoluto
$\tau_s = 0.5$ ms	constante de tiempo de la corriente postsináptica
$C_m = 250$ pF	capacitancia de la membrana
$V_r = -65$ mV	valor de reinicio del potencial de membrana
$\theta = -50$ mV	umbral de disparo
$E_L = -65$ mV	potencial de fuga
$J = 87.8$ pA	intensidad sináptica media
$\nu_{bg} = 8$ spikes/s	tasa promedio de spikes externas
$k_{ext} = 940$	n° de entradas por neurona

Tabla 6.1: Parámetros del modelo SNN.

A continuación, aplicamos los conceptos desarrollados previamente y comparamos las conclusiones obtenidas con los correspondientes resultados de simulación.

6.3. Análisis Basado en Actividad

En esta sección utilizaremos el modelo descrito anteriormente, considerando para cada neurona una corriente constante de entrada I_e , por lo que las ecuaciones diferenciales que

describen el comportamiento del sistema se modifican de la siguiente manera:

$$\begin{aligned}\frac{dI_i^s(t)}{dt} &= -\frac{I_i^s(t)}{\tau_s} + \frac{I_e}{\tau_s} \\ \frac{dV_i(t)}{dt} &= -\frac{V_i(t) - E_L}{\tau_m} + \frac{I_i^s(t)}{C_m}\end{aligned}\tag{6.4}$$

para $i = 1, \dots, N_e$, donde N_e es el número de neuronas en la red.

Para comenzar el análisis, resulta interesante identificar que la Ecuación (6.4) define las dinámicas continuas como en la Ecuación (3.23), las condiciones $V_i - \theta = 0$ corresponden a las funciones de cruce por cero de la Ecuación (3.24), y las Ecuaciones (6.2)–(6.3) corresponden al event handler de la Ecuación (3.25).

Con el fin de analizar de qué manera las discontinuidades afectan al número resultante de actualizaciones de estado, simulamos el modelo provocando diferentes tasas de discontinuidades mediante la variación del parámetro I_e (un valor mayor de este parámetro incrementa la velocidad con que V_s alcanza el umbral, lo que a su vez incrementa la frecuencia de ocurrencia de discontinuidades).

Realizamos simulaciones durante 0.25 segundos empleando los algoritmos QSS2, eLIQSS2, CheQSS y CVODE (con orden máximo limitado a 2) y distintos valores de tolerancia.

Las Figuras 6.1a y 6.1b muestran un subconjunto de las trayectorias de estado obtenidas de una simulación con una corriente externa constante $I_e = 375.5$ nA (este valor se encuentra próximo al mínimo necesario para que las neuronas alcancen el umbral y disparen).

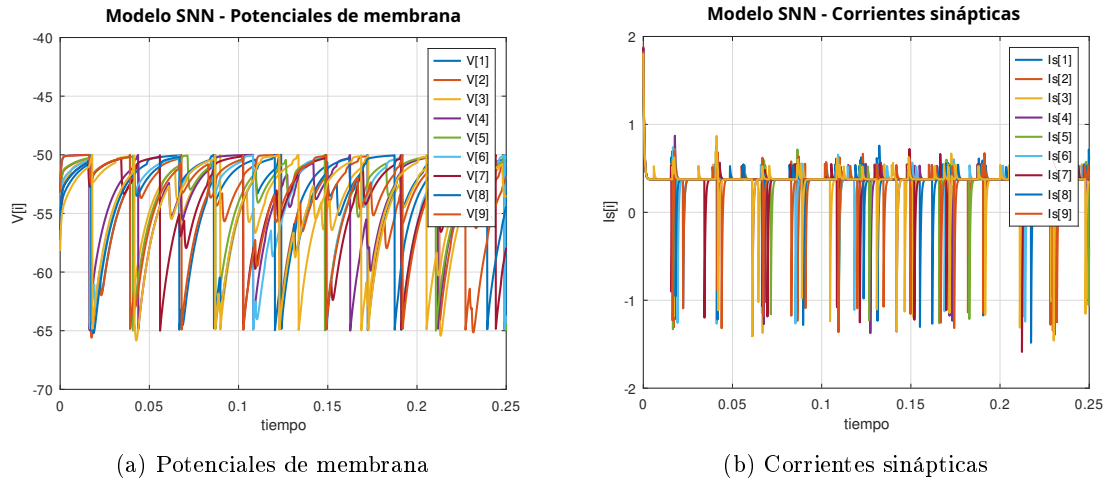


Figura 6.1: Subconjunto de las trayectorias de estado para el modelo SNN considerando una corriente de entrada constante para cada neurona.

Para calcular las actividades de orden 2 para los distintos escenarios utilizamos los resultados obtenidos de simulaciones con alta precisión. Posteriormente, determinamos las cotas inferiores teóricas del número de pasos requeridos por cada método ($k_{QS}^{(2)}$ y $k_{DT}^{(2)}$) para completar la simulación, siguiendo las Ecs. (3.26) y (3.27), las cuales contemplan la presencia de

discontinuidades. Estos resultados se presentan en la Tabla 6.2, junto con el número real de pasos de simulación ejecutados por cada método numérico.

I_e/τ_s	Spikes	Abs Tol.	Métodos QSS				Método de DT	
			$k_{QS}^{(2)}$	Real			$k_{DT}^{(2)}$	Real CVODE ₂
				QSS2	eLIQSS2	CheQSS2		
751	4,741	10^{-1}	$1.15 \cdot 10^5$	$1.24 \cdot 10^6$	$1.46 \cdot 10^5$	$1.64 \cdot 10^5$	$5.47 \cdot 10^3$	$9.57 \cdot 10^3$
		10^{-2}	$3.55 \cdot 10^5$	$1.61 \cdot 10^6$	$4.80 \cdot 10^5$	$3.61 \cdot 10^5$	$7.04 \cdot 10^3$	$1.08 \cdot 10^4$
		10^{-3}	$1.11 \cdot 10^6$	$3.38 \cdot 10^6$	$1.54 \cdot 10^6$	$1.26 \cdot 10^6$	$1.20 \cdot 10^4$	$1.28 \cdot 10^4$
800	7,701	10^{-1}	$2.10 \cdot 10^5$	$1.22 \cdot 10^6$	$2.61 \cdot 10^5$	$2.17 \cdot 10^5$	$8.98 \cdot 10^3$	$1.52 \cdot 10^4$
		10^{-2}	$6.49 \cdot 10^5$	$1.85 \cdot 10^6$	$7.80 \cdot 10^5$	$6.49 \cdot 10^5$	$1.17 \cdot 10^4$	$1.69 \cdot 10^4$
		10^{-3}	$2.03 \cdot 10^6$	$4.96 \cdot 10^6$	$2.40 \cdot 10^6$	$1.99 \cdot 10^6$	$2.05 \cdot 10^4$	$2.08 \cdot 10^4$
1000	14,531	10^{-1}	$3.05 \cdot 10^5$	$1.22 \cdot 10^6$	$3.73 \cdot 10^5$	$3.18 \cdot 10^5$	$1.60 \cdot 10^4$	$2.85 \cdot 10^4$
		10^{-2}	$9.33 \cdot 10^5$	$2.48 \cdot 10^6$	$1.18 \cdot 10^6$	$9.68 \cdot 10^5$	$1.92 \cdot 10^4$	$3.00 \cdot 10^4$
		10^{-3}	$2.92 \cdot 10^6$	$7.36 \cdot 10^6$	$3.63 \cdot 10^6$	$3.01 \cdot 10^6$	$2.92 \cdot 10^4$	$3.55 \cdot 10^4$

Tabla 6.2: Número de discontinuidades (spikes) y número de pasos (cotas inferiores y valores observados) en distintos escenarios

Además, calculamos el Factor de Homogeneidad $H_{\mathbf{x}}^{(2)}$, el producto $H_{\mathbf{x}}^{(2)} \cdot r^{(2)}$ y las cotas inferiores para el número de actualizaciones requeridas por cada enfoque, siguiendo las Ecs. (3.28) y (3.29).

Para obtener las cotas inferiores de $u_{QS}^{(2)}$, determinamos previamente el valor de $r^{(2)}$ en cada escenario, dado que la conectividad no es constante (la variable $V_i(t)$ aparece en una sola ecuación de estado, mientras que $I_i^s(t)$ aparece en dos). Obtuvimos los siguientes valores:

- Escenario 1 ($I_e/\tau_s = 751$): $r^{(2)} = 1.16$
- Escenario 2 ($I_e/\tau_s = 800$): $r^{(2)} = 1.22$
- Escenario 3 ($I_e/\tau_s = 1000$): $r^{(2)} = 1.19$

Fijamos además $s = 11$, ya que cada spike afecta 11 derivadas de estado (correspondientes a las $m = 10$ neuronas que reciben el evento y la neurona que dispara).

La Tabla 6.3 muestra el Factor de Homogeneidad, la razón entre las cotas teóricas inferiores para las actualizaciones $u_{QS}^{(2)}/u_{DT}^{(2)}$ y la razón observada entre las actualizaciones en la simulación. Además, se incluye la razón entre los tiempos de CPU de ambas simulaciones.

A partir de los resultados de las Tablas 6.2 y 6.3, podemos realizar las siguientes observaciones:

- El modelo tiene $N = 2000$ variables de estado (dos por neurona), por lo que el rango teórico de $H_{\mathbf{x}}^{(n)}$ es $[\frac{1}{2000}, 1]$. En los tres escenarios, el Factor de Homogeneidad es significativamente menor que 1, lo cual es coherente con la actividad heterogénea de las trayectorias.
- CVODE2 realiza menos pasos que todos los métodos de QSS. Sin embargo, cada paso de CVODE2 implica cálculos sobre todo el vector de estado (de tamaño $N = 2000$), mientras

I_e/τ_s	ΔQ	Resultados teóricos			Resultados de simulación					
		$H_{\mathbf{x}}^{(2)}$	$H_{\mathbf{x}}^{(n)} \cdot r^{(2)}$	$\frac{u_{\text{QS}}^{(2)}}{u_{\text{DT}}^{(2)}}$	N° actualizaciones			Tiempo de Simulación		
					$\frac{u_{\text{QSS2}}}{u_{\text{CVODE2}}}$	$\frac{u_{\text{eLIQSS2}}}{u_{\text{CVODE2}}}$	$\frac{u_{\text{CheQSS2}}}{u_{\text{CVODE2}}}$	$\frac{t_{\text{QSS2}}}{t_{\text{CVODE2}}}$	$\frac{t_{\text{eLIQSS2}}}{t_{\text{CVODE2}}}$	$\frac{t_{\text{CheQSS2}}}{t_{\text{CVODE2}}}$
751	10^{-1}	0.076	0.088	0.017	0,102	0,012	0,013	0,0074	0,0020	0,0020
	10^{-2}			0.033	0,112	0,031	0,023	0,0083	0,0041	0,0033
	10^{-3}			0.056	0,187	0,081	0,065	0,0151	0,0113	0,0093
800	10^{-1}	0.079	0.096	0.018	0,064	0,014	0,011	0,0044	0,0020	0,0016
	10^{-2}			0.037	0,082	0,033	0,027	0,0062	0,0044	0,0035
	10^{-3}			0.062	0,167	0,080	0,064	0,0133	0,0098	0,0085
1000	10^{-1}	0.099	0.118	0.016	0,035	0,011	0,010	0,0026	0,0015	0,0014
	10^{-2}			0.033	0,061	0,029	0,023	0,0047	0,0036	0,0032
	10^{-3}			0.062	0,145	0,072	0,058	0,0109	0,0085	0,0074

Tabla 6.3: Relación teórica y simulada entre los costos computacionales de un algoritmo de cuantificación de estados y uno de tiempo discreto. Modelo SNN

que QSS2, eLIQSS2 y CheQSS2 sólo involucran cálculos sobre, como máximo, $r = 2$ estados (en pasos continuos) o $s = 11$ estados (cuando se produce una discontinuidad).

- En concordancia con la observación anterior, el número de actualizaciones realizadas por CVODE2 es entre 5 y 28 veces mayor que el de QSS2; entre 12 y 90 veces mayor que el de eLIQSS2 y entre 15 y 104 veces mayor que el de CheQSS2.
- El número de pasos realizados por todos los algoritmos es cercano a sus respectivas cotas inferiores teóricas. En consecuencia, la razón entre las cotas inferiores de actualizaciones $u_{\text{QS}}^{(2)}/u_{\text{DT}}^{(2)}$ se aproxima a la observada en la simulación (sobre todo en los casos de eLIQSS2 y CheQSS2).
- El hecho de que $H_{\mathbf{x}}^{(n)} \cdot r^{(2)} < 1$ implica que, incluso sin discontinuidades, los métodos QSS deberían ser más eficientes en este modelo. La presencia de discontinuidades incrementa aún más la eficiencia de los métodos basados en cuantificación, como se deduce del hecho de que $u_{\text{QS}}^{(2)}/u_{\text{DT}}^{(2)} \ll H_{\mathbf{x}}^{(n)} \cdot r^{(2)}$.
- La relación entre los tiempos de CPU muestra ventajas aún mayores para los algoritmos QSS. Mientras que el número de actualizaciones es entre 5 y 104 veces menor en QSS, las simulaciones resultan entre 15 y 650 veces más rápidas. Esto se debe principalmente a que CVODE es un algoritmo implícito que realiza iteraciones en cada paso temporal. Aunque las iteraciones se realizan sobre un sistema lineal y con una representación sparse del Jacobiano, esto implica un mayor costo computacional en comparación con los algoritmos de QSS (que en los casos de LIQSS también son implícitos, pero sólo resuelven ecuaciones lineales escalares derivadas de la aproximación lineal local de la dinámica dada por la Ec. (2.13), ya que las actualizaciones se realizan de forma local sobre variables individuales, evitando la resolución de sistemas lineales acoplados).

6.4. Eficiencia de los métodos

Para evaluar la eficiencia de los distintos métodos, utilizamos nuevamente el modelo descrito en las Secciones 6.2, pero en lugar de tener en cuenta una corriente constante de entrada I_e de cada neurona, consideramos que cada neurona recibe un tren de pulsos de entrada independiente, generado por un proceso de Poisson con tasa estacionaria $\nu_{ext} = k_{ext} \cdot \nu_{bg}$ (los valores y definiciones de k_{ext} y ν_{bg} pueden encontrarse en la Tabla 6.1), los que se obtienen mediante un generador de números pseudoaleatorios¹ con semillas prefijadas, de modo que las distintas corridas de simulación puedan compararse bajo condiciones idénticas. De esta forma, cada neurona recibe tanto su propio tren de *spikes* de entrada, como los *spikes* provenientes de sus conexiones sinápticas entrantes, descritas en la Sección 6.2.

Nuevamente consideramos una red de $N_e = 1000$ neuronas, de las cuales 800 son excitatorias y las restantes 200 inhibitorias y cada una recibe señales desde $m = 10$ conexiones sinápticas elegidas al azar, de las cuales el 80 % provienen de neuronas excitatorias y el 20 % restante de neuronas inhibitorias.

Las Figuras 6.2a y 6.2b ilustran un subconjunto de las trayectorias de estado del sistema considerando este tren de spikes en lugar de una corriente de entrada constante.

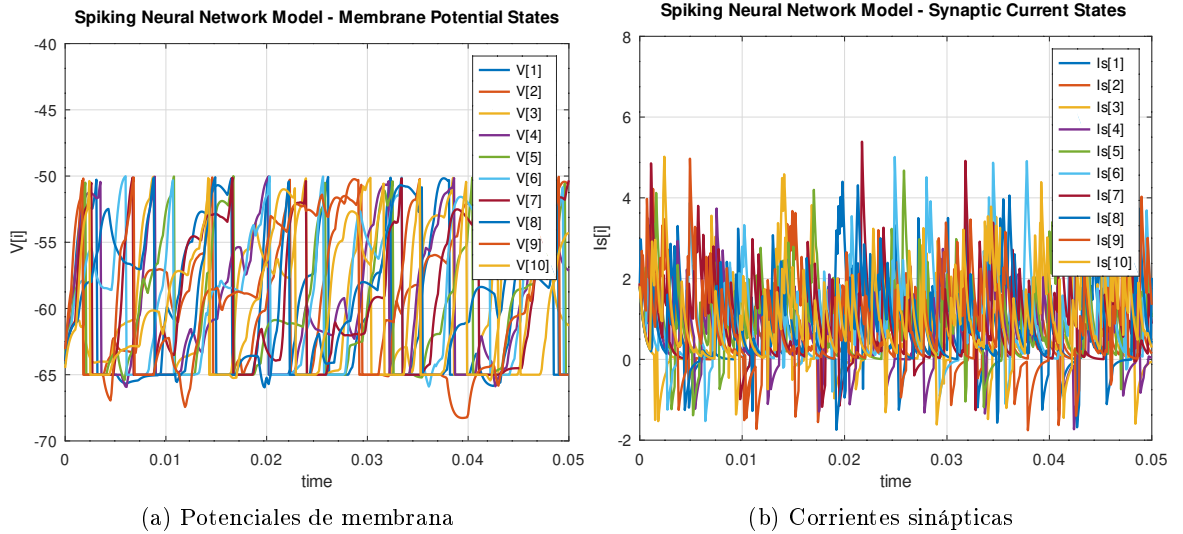


Figura 6.2: Subconjunto de trayectorias de estado para el modelo SNN considerando que cada neurona tiene como entrada un tren de spikes.

Simulamos el sistema durante un intervalo de 0.05s utilizando distintos métodos QSS: CheQSS, eLIQSS y QSS, además del método clásico en tiempo discreto DOPRI. Dado que este problema no es stiff, DOPRI mostró el mejor rendimiento entre los algoritmos clásicos de tiempo discreto. Por esta misma razón, dentro de los métodos QSS previamente definidos utilizamos QSS en lugar de LIQSS. Las simulaciones se realizaron para distintos órdenes de los métodos QSS (de 1 a 3) y empleando diferentes configuraciones de tolerancias.

¹ El paquete QSS Solvers utiliza la biblioteca estándar de C (`stdlib`) para la generación de secuencias pseudoaleatorias.

Dado que el sistema incorpora componentes estocásticos, seleccionamos 10 semillas aleatorias distintas y realizamos 10 simulaciones para cada combinación de método numérico, orden del método y valores de tolerancia. Luego calculamos el número promedio de pasos y el tiempo promedio de simulación entre estas 10 corridas, tal como se reporta en la Tabla 6.4 y en la Figura 6.3.

Para evaluar la precisión de las simulaciones teniendo en cuenta el régimen caótico de las soluciones y que pequeñas desviaciones en las trayectorias pueden generar diferencias significativas en los tiempos y cantidades de disparos, utilizamos como métrica de error la diferencia en el número total de disparos emitidos por las neuronas durante la simulación.

Para obtener una solución de referencia, realizamos una simulación adicional de alta precisión (utilizando CheQSS3 con $\Delta Q = 10^{-10}$), de la cual tomamos el número total de spikes emitidos como referencia. Luego, para cada semilla aleatoria y configuración de simulación, calculamos el número total de spikes y lo comparamos con el valor de referencia.

Para cuantificar el error de simulación, calculamos el Error Relativo Medio (MRE) entre el número de disparos producidos en cada corrida y el número de referencia, según:

$$\text{MRE} = \frac{1}{10} \sum_{i=1}^{10} \frac{|S_i^{\text{ref}} - S_i|}{S_i^{\text{ref}}}, \quad (6.5)$$

donde S_i denota la cantidad de disparos obtenida en la i -ésima simulación y S_i^{ref} el número de referencia correspondiente. Los valores resultantes de MRE se reportan en las últimas columnas de la Tabla 6.4.

n	Tol. Abs.	N° de pasos			Tiempo de CPU (ms)			Error relativo medio en número de spikes (%)		
		CheQSS _n	eLIQSS _n	QSS _n	CheQSS _n	eLIQSS _n	QSS _n	CheQSS _n	eLIQSS _n	QSS _n
2	10^{-1}	222420	248389	487716	119.0	129.1	191.9	1.84E-03	1.40E-03	2.22E-03
	10^{-2}	576976	676553	342798	244.6	279.5	453.0	4.73E-04	3.97E-04	3.03E-04
	10^{-3}	1713686	2026289	4080886	642.1	748.1	1296.8	3.77E-05	3.75E-05	3.77E-05
	10^{-4}	5265527	6284049	2603478	1877.9	2205.9	3851.6	0	0	0
	10^{-5}	16300337	19749735	39542559	5592.9	6748.6	10908.7	0	0	0
3	10^{-1}	156299	178947	471965	156.3	165.2	312.8	2.05E-03	1.31E-03	8.93E-04
	10^{-2}	234559	311722	695625	204.4	244.9	426.9	3.80E-04	5.11E-04	3.60E-04
	10^{-3}	411516	537240	167814	317.4	383.7	657.2	1.13E-04	0	0
	10^{-4}	776038	1063599	2270492	542.9	699.7	1211.8	0	0	0
	10^{-5}	1558635	2220378	4625227	1040.5	1401.2	2400.7	0	0	0
DOPRI										
5	10^{-1}	59925			7823.2			3.79E-05		
	10^{-2}	60185			7752.9			1.88E-05		
	10^{-3}	60277			7627.9			0		
	10^{-4}	60314			7550.2			0		
	10^{-5}	60329			7487.7			0		

Tabla 6.4: Número de pasos, tiempo de CPU y error en la simulación del modelo SNN variando el quantum y los valores de tolerancia.

A partir de los resultados obtenidos, podemos remarcar las siguientes observaciones:

- En todos los casos, los nuevos métodos (CheQSSy eLIQSS) resultaron más eficientes que

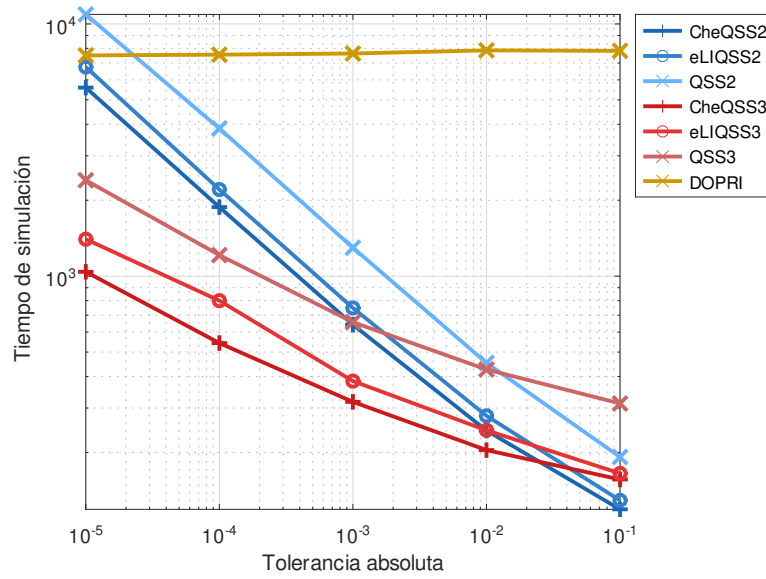


Figura 6.3: Tiempos de simulación según diferentes tolerancias de los distintos algoritmos en la simulación del modelo SNN

los QSS, como se evidencia por la menor cantidad de pasos y los menores tiempos de simulación.

- Si bien DOPRI requiere menos pasos, cada paso es computacionalmente más costoso, lo que resulta en tiempos de simulación significativamente mayores. Esto se debe a que, como método clásico en tiempo discreto, DOPRI actualiza todas las variables de estado en cada paso, mientras que en los métodos basados en cuantificación de estados sólo se actualizan aquellas variables que presentan cambios significativos.
- Los errores son similares entre los tres métodos de cuantificación de estado y tienden a cero a medida que se incrementa la precisión.
- Los errores de DOPRI tienden rápidamente a cero y el número de pasos es casi independiente de la tolerancia. Esto se debe a que el tamaño de paso está limitado por el tiempo entre eventos (correspondientes a disparos neuronales o entradas). El pequeño tiempo entre eventos genera tamaños de paso reducidos, lo que conduce a soluciones de alta precisión. Sin embargo, el tiempo de CPU es aproximadamente un orden de magnitud mayor que el de los métodos QSS de tercer orden que alcanzan, en la práctica, la misma precisión.
- Al comparar CheQSS y eLIQSS entre sí, CheQSS requiere consistentemente menos recursos computacionales: en todas las configuraciones evaluadas, eLIQSS utiliza entre un 10 % y un 40 % más de pasos y entre un 8 % y un 35 % más de tiempo de simulación. En cuanto a la precisión, excepto en el caso de tercer orden con $\Delta Q_{\text{abs}} = 10^{-3}$, ambos métodos producen errores muy similares bajo las mismas configuraciones de tolerancia.

Capítulo 7

Aplicación a un Modelo Epidemiológico Multiescala

En este capítulo aplicamos los conceptos desarrollados a un sistema que combina gran escala, rigidez y presencia de discontinuidades. El caso de estudio corresponde a un modelo epidemiológico basado en agentes que integra la dinámica biológica propia de cada individuo (escala microscópica) con dinámicas socioambientales, tales como la evolución de los contagios a nivel poblacional debido a la concentración de partículas infectivas en el aire de un ambiente compartido (escala macroscópica).

7.1. Modelos epidemiológicos

El modelado matemático de la propagación de enfermedades infecciosas en poblaciones humanas admite una amplia variedad de representaciones y puede abordarse a distintas escalas [63]. Algunos enfoques resultan adecuados para describir la propagación de un virus a nivel poblacional, como los modelos de tipo SIR [65, 14], mientras que otros se centran en el nivel individual, modelando la dinámica del proceso biológico interno de la infección viral de cada huésped [58, 34]. En la mayoría de estos modelos se introduce el concepto de compartimentos, que representan las fracciones de una población (ya sea de individuos o de células según el tipo de modelo) que se encuentran en alguno de los estados relevantes del proceso infeccioso, tales como susceptibles, infectados o recuperados. La elección de la técnica de modelado suele depender tanto de la información disponible para la estimación de parámetros como de la pregunta específica que se desea responder.

Si bien los modelos de tipo SIR y sus extensiones han demostrado ser herramientas eficaces para analizar la evolución global de una epidemia, su formulación agregada limita la posibilidad de capturar con detalle la dinámica de la infección a nivel individual, así como la influencia explícita de factores sociales y ambientales. En particular, estos modelos no permiten representar de manera directa la variabilidad entre individuos en los parámetros biológicos ni las

condiciones específicas bajo las cuales se producen los contactos y los contagios. La incorporación explícita de estas características requiere modelos más detallados, capaces de describir tanto la evolución interna de cada huésped como las dinámicas sociales y ambientales que condicionan las interacciones. Sin embargo, este mayor nivel de detalle suele venir acompañado de un incremento significativo en la complejidad del modelo y, en consecuencia, de un aumento sustancial del costo computacional asociado a su simulación.

Con el objetivo de poner a prueba los conceptos teóricos presentados y los métodos QSS desarrollados en esta tesis, se adopta un modelo epidemiológico basado en agentes, en el cual una población (susceptible de contraer una infección viral) habita un ambiente compartido del que los individuos pueden entrar y salir, y en el que pueden producirse contagios. El modelo presenta rigidez debido a las características de las ecuaciones diferenciales que describen la dinámica de la infección a nivel del huésped. Asimismo, es de gran escala, ya que involucra un elevado número de individuos, cuya evolución se modela mediante conjuntos de ecuaciones diferenciales ordinarias, y exhibe discontinuidades como consecuencia de la dinámica de entrada y salida de los agentes del ambiente compartido.

Este enfoque resulta útil, ya que permite alcanzar un mayor nivel de detalle que los modelos SIR formulados a nivel agregado de la población, al posibilitar la asociación entre la variabilidad de los parámetros individuales de cada agente y su impacto en las curvas de contagio a nivel poblacional. Sin embargo, como mencionamos anteriormente, este mayor nivel de detalle en el modelo suele traducirse en costos computacionales elevados. Por este motivo, en este capítulo comparamos el desempeño de los métodos de integración numérica clásicos con el de los métodos QSS, con el objetivo de evaluar en qué medida estos últimos permiten mitigar dicho inconveniente.

7.2. Modelo Epidemiológico Multiagente Propuesto

El último ejemplo elegido es un modelo epidemiológico multiagente que combina la dinámica biológica de una infección viral por SARS-CoV-2 en los huéspedes con la transmisión del virus entre individuos (agentes) dentro de una población, a través de un ambiente compartido.

El modelo se basa en un modelo de huésped, como el presentado en [58], que considera tanto la dinámica de replicación viral como la respuesta inmune, y se caracteriza por el siguiente conjunto de ODEs, donde para acortar la notación omitimos la dependencia temporal de los

estados:

$$\dot{V} = \gamma_V \cdot I - (\gamma_{VA} \cdot S \cdot A - \gamma_{VH} \cdot H - \alpha_V) \cdot V - \frac{a_{VI} \cdot V}{1 + a_{V2} \cdot V} \quad (7.1a)$$

$$\dot{H} = b_{HD} \cdot D \cdot (H + R) + a_R \cdot R - \gamma_{HV} \cdot V \cdot H - b_{HD} \cdot F \cdot H \quad (7.1b)$$

$$\dot{L} = \gamma_{HV} V H - \gamma_{LI} L \quad (7.1c)$$

$$\dot{I} = \gamma_{LI} L - b_{IE} E I - a_I I \quad (7.1d)$$

$$\dot{M} = (b_{MD} \cdot D + b_{MV} \cdot V) \cdot (1 - M) - a_M \cdot M \quad (7.1e)$$

$$\dot{F} = b_F \cdot M + c_F \cdot I - b_{FH} \cdot H \cdot F - a_F \cdot F \quad (7.1f)$$

$$\dot{R} = b_{HF} \cdot F \cdot H - a_R \cdot R \quad (7.1g)$$

$$\dot{E} = b_{EM} \cdot M \cdot E - b_{EI} \cdot I \cdot E + a_E \cdot (1 - E) \quad (7.1h)$$

$$\dot{P} = b_{PM} \cdot M \cdot P + a_P \cdot (1 - P) \quad (7.1i)$$

$$\dot{A} = b_A \cdot P - \gamma_{AV} \cdot S \cdot A \cdot V - \gamma_A \cdot A \quad (7.1j)$$

$$\dot{S} = r \cdot P \cdot (1 - S), \quad (7.1k)$$

En este modelo, la variable V representa la carga viral medida en cantidad de partículas virales en un dado volumen. El valor de V está escalado y vale 1 por cada 10^{10} partículas virales en $1ml$ de fluido celular. Las variables H (Sanas), L (Latentes), I (Infectadas), D (Muertas) y R (Resistentes) representan el ciclo de vida celular de las células que pueden ser infectadas por partículas virales. Dado que se mantiene una ley de conservación entre estas poblaciones de células, D puede calcularse directamente como una variable de salida $D = 1 - H - R - I$. La cantidad de células presentadoras de antígeno están representadas por la variable M . La variable F representa la cantidad de interferón α y β producido tanto por las células presentadoras de antígeno como por las células infectadas. Las variables E y P representan la concentración de células efectoras y células de plasma, respectivamente, y A representa la concentración de anticuerpos. Por último la variable S , representa la capacidad de los anticuerpos por detectar o reconocer los antígenos del virus y puede ser pensada como una medida de afinidad de unión entre los anticuerpos y el virus.

El modelo considera una población de N_p individuos, que pueden eventualmente contraer, desarrollar y transmitir una infección viral.

Este modelo combina la dinámica de infección de cada agente con la dinámica de la concentración viral en un ambiente compartido, con el cual los individuos intercambian partículas virales. De este modo, permite asociar parámetros biológicos, sociales y ambientales con la propagación de la enfermedad en la población. Para representar este fenómeno, se combinan los siguientes submodelos:

- **Modelo de huésped:** para representar la dinámica del i -ésimo huésped, quién poten-

cialmente podrá desarrollar una infección viral, utilizamos, como indicamos previamente, el modelo de la Ec.(7.1), donde reemplazamos la Ec.(7.1a) por la siguiente:

$$\dot{V}_i = \gamma_V \cdot I - (\gamma_{VA} \cdot S_i \cdot A_i - \gamma_{VH} \cdot H_i - \alpha_V) \cdot V_i - \frac{a_{V1} \cdot V_i}{1 + a_{V2} \cdot V_i} + r_i(t) \cdot V_L \quad (7.2)$$

donde V_L representa la concentración de carga viral en el ambiente compartido (por ejemplo, una habitación) y $r_i(t)$ es un parámetro que indica el nivel de recepción de carga viral de una persona. En los instantes de tiempo t en los que la persona se encuentre fuera del ambiente, tomamos $r_i(t) = 0$. Cuando la persona esté dentro del ambiente $r_i(t)$ asume valores positivos ($r_i(t) > 0$) que permiten modelar distintas situaciones como por ejemplo, uso del barbijo y la calidad del mismo.

Notar que el modelo además permite representar distintas condiciones del sistema inmunológico de las personas tanto cambiando parámetros, como definiendo condiciones iniciales para la memoria inmune $S_i(t)$ y/o para el nivel de anticuerpos $A_i(t)$.

- **Modelo del ambiente:** Representamos la evolución de la concentración de carga viral V_L en el ambiente mediante la siguiente ecuación:

$$\dot{V}_L(t) = V_{pob}(t) - \delta \cdot V_L(t) \quad (7.3)$$

donde δ representa la tasa característica de decaimiento de la concentración de carga viral en el ambiente. Este parámetro depende de las características propias del ambiente (volumen de la habitación, ventilación, etc.) y de la viabilidad del virus con el tiempo. Por último, V_{pob} es la entrada de carga viral hacia el ambiente aportada por la población de individuos presentes, y se calcula de la siguiente manera:

$$V_{pob}(t) = \sum_{i=1}^N e_i(t) \cdot V_i(t) \quad (7.4)$$

donde $e_i(t)$ es el factor de transmisión de carga viral hacia el ambiente del i -ésimo individuo. De manera similar al factor de recepción $r_i(t)$, cuando la persona está fuera del ambiente se toma $e_i(t) = 0$, y adopta valores positivos ($e_i(t) > 0$) cuando se encuentra en el interior del mismo dependiendo del uso del barbijo y su calidad, de la actividad que esté realizando, etc.

- **Clasificación de los Individuos:** Para analizar la evolución de los contagios a nivel poblacional, clasificamos los individuos en tres categorías:

- Susceptibles: individuos con nivel de anticuerpos y memoria inmune por debajo de cierto umbral.
- Infectados: individuos cuya carga viral superó cierto umbral dentro de un intervalo

de tiempo reciente.

- Recuperados: individuos que estuvieron infectados y mantienen respuesta inmune (y por lo tanto no están incluidos en los dos grupos anteriores).

7.2.1. Parámetros elegidos

Para el análisis basado en actividad y las simulaciones del modelo consideramos una población de N_p personas, de las cuales un 5 % están inicialmente infectadas con una carga viral inicial $V_i(0) = 10$, mientras que el 95 % restante es susceptible.

Los parámetros de la dinámica en huésped son idénticos a los de la referencia [58].

Para cada individuo consideramos valores aleatorios (distribuidos uniformemente en un intervalo dado) para: **(a)** los niveles de recepción y transmisión en el ambiente ($r_i(t)$ y $e_i(t)$), **(b)** una secuencia aleatoria de instantes de entrada y salida del ambiente, y **(c)** un valor inicial de concentración de anticuerpos $A_i(0)$.

Los siguientes parámetros están asociados al comportamiento social y los elegimos de manera tal de obtener un escenario en donde se contagia alrededor de la mitad de la población:

(1) $\delta = 100$: es la velocidad de dispersión de la carga viral del ambiente. Esto implica que en aproximadamente 1 hora se dispersa prácticamente toda la carga viral.

(2) $\alpha = 0,5$: es la proporción de personas que tienen bajo intercambio con el ambiente ($r_i(t) = e_i(t) \in [0.04, 0.14]$, aleatorio con distribución uniforme) y el resto alto intercambio ($r_i(t) = e_i(t) \in [0.14, 0.4]$ aleatorios con distribución uniforme). Esto podría asociarse al uso de barbijo y/o a las actividades realizadas.

(3) $t_{max} = 2$: es el tiempo de permanencia diario máximo en el ambiente compartido.

(4) $p = 0\%$ es el porcentaje de la población inmunizada previamente.

El modelo μ -Modelica utilizado en las simulaciones se encuentra disponible en https://fceia.unr.edu.ar/~kofman/files/paper_Simulation2023.zip.

En las Figuras 7.1a - 7.1d se observan algunas trayectorias típicas de algunas variables de estado del modelo de huésped, como la carga viral, memoria inmunitaria, porcentaje de células infectadas y porcentaje de células muertas a lo largo de una infección.

En estas imágenes ya se puede reconocer heterogeneidad entre las variables de estado correspondientes a un mismo individuo. A este hecho hay que sumarle además, que los procesos de infección de cada integrante de la población comienzan en distintos momentos a lo largo de todo el período de simulación.

Luego, la Figura 7.2, muestra la evolución de la concentración de la carga viral en el ambiente compartido en una simulación con $N_p = 100$ personas.

El hecho de que los individuos entren y salgan del ambiente compartido genera discontinuidades en el sistema, lo que sumado a la heterogeneidad, lleva a sospechar que los métodos

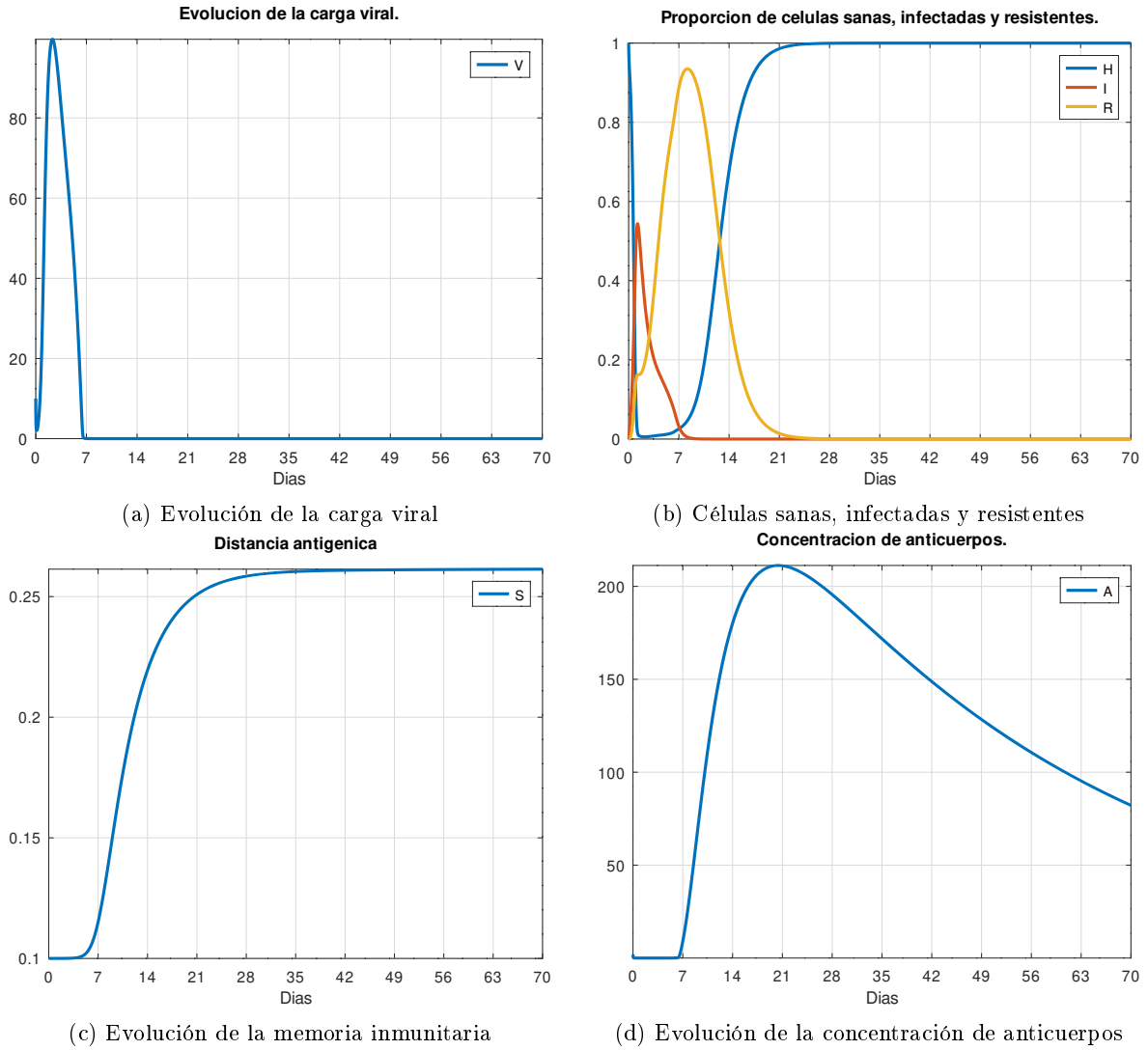


Figura 7.1: Algunas trayectorias de estados del modelo de huésped

de QSS pueden ser más eficaces para simular este sistema que los métodos clásicos de discretización temporal, lo cual vamos a analizar en las siguientes secciones.

Por último, en la Figura 7.3 se muestran las curvas de contagio resultantes del comportamiento de las personas a lo largo de toda la simulación. En la misma se puede observar que con estos parámetros, alrededor de un 60% de las personas resultaron infectadas.

7.3. Análisis Basado en Actividad

En esta sección aplicamos los conceptos teóricos desarrollados previamente para calcular la actividad local y global de este sistema y luego obtener el factor de homogeneidad. Para esto, repetimos el procedimiento descrito en las secciones anteriores y obtuvimos los valores de la actividad local y global utilizando los resultados de una simulación de alta precisión ($\Delta Q_{\text{rel}} = 10^{-7}$, $\Delta Q_{\text{abs}} = 10^{-10}$) durante 70 segundos realizada con LIQSS2, para luego calcular los HF

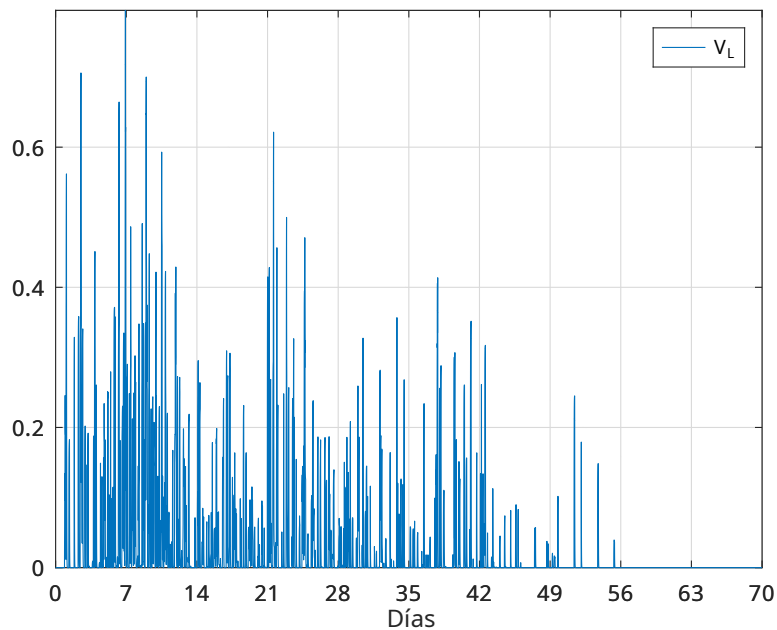
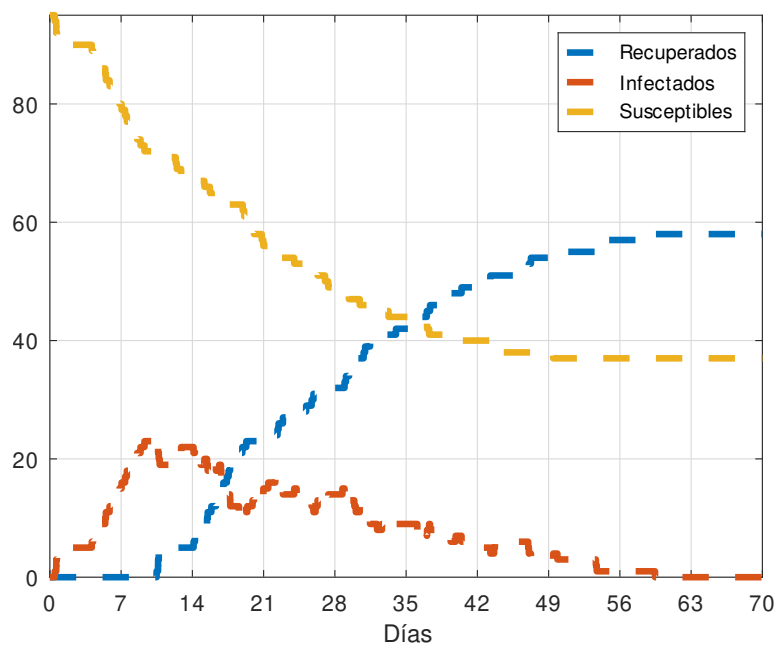


Figura 7.2: Concentración de carga viral en el ambiente compartido

Figura 7.3: Curvas de contagio en el modelo epidemiológico para $N = 100$

de órdenes 1 a 3.

La Tabla 7.1, muestra los resultados obtenidos, que corroboran el comportamiento heterogéneo observado en las trayectorias del sistema.

n	$H_{\mathbf{x}}^{(n)}$	$H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$
1	0.017	0.057
2	0.029	0.094
3	0.060	0.187

Tabla 7.1: Factores de Homogeneidad - Modelo epidemiológico

A partir de los valores de las actividades locales y globales también calculamos las cotas teóricas mínimas para el número de pasos y actualizaciones siguiendo las Ecuaciones (3.27), (3.26), (3.29) y (3.28). Para estas estimaciones, consideramos un valor de tolerancia $\Delta Q_{\text{abs}} = 10^{-7}$ tanto para los métodos de QSS, como para CVODE. Los valores obtenidos se pueden observar en la tercera columna de la Tabla 7.2.

Luego, simulamos el modelo con los diferentes métodos numéricos de la familia QSS (LIQSS, eLIQSS y CheQSS) y con CVODE. En todos los casos utilizamos métodos de orden $n = 2$ y $n = 3$, ya que por el tamaño y complejidad del modelo, los métodos de primer orden resultan muy lentos. Para CVODE, al igual que en la sección anterior, las soluciones de orden 2 y 3 fueron obtenidas limitando el orden máximo del método.

Para concluir este análisis, comparamos el número de pasos ejecutados efectivamente en cada simulación con los resultados teóricos previamente obtenidos, lo que presentamos en la Tabla 7.2.

Abs Tol.	n	Pasos de simulación					
		Métodos QSS				Método de DT	
		$k_{\text{QS}}^{(2)}$	Real			$k_{\text{DT}}^{(2)}$	Real CVODE $_n$
			LIQSS $_n$	eLIQSS $_n$	CheQSS $_n$		
10^{-7}	2	$1,71 \cdot 10^6$	$6,16 \cdot 10^6$	$3,04 \cdot 10^6$	$2,57 \cdot 10^6$	$6,76 \cdot 10^4$	$8,78 \cdot 10^4$
	3	$2,84 \cdot 10^5$	$7,77 \cdot 10^5$	$4,33 \cdot 10^5$	$5,01 \cdot 10^5$	$1,82 \cdot 10^4$	$7,83 \cdot 10^4$
		Actualizaciones					
		Métodos QSS				Método de DT	
		$u_{\text{QS}}^{(2)}$	Real			$u_{\text{DT}}^{(2)}$	Real CVODE $_n$
			LIQSS $_n$	eLIQSS $_n$	CheQSS $_n$		
10^{-7}	2	$6,50 \cdot 10^6$	$2,07 \cdot 10^7$	$1,05 \cdot 10^7$	$9,07 \cdot 10^6$	$8,99 \cdot 10^7$	$1,06 \cdot 10^8$
	3	$1,04 \cdot 10^6$	$2,91 \cdot 10^6$	$1,60 \cdot 10^6$	$1,90 \cdot 10^6$	$3,56 \cdot 10^7$	$9,41 \cdot 10^7$

Tabla 7.2: Número mínimo teórico y real de pasos y actualizaciones en el modelo epidemiológico

Los resultados muestran una vez más que las estimaciones teóricas del número mínimo de pasos y actualizaciones, tanto para los métodos de QSS, como para los métodos clásicos, se encuentran en el mismo orden de magnitud que los números de pasos obtenidos en las simulaciones.

Además, observando que el número de actualizaciones de todos los métodos de QSS resultaron menores al número de actualizaciones de CVODE, podemos corroborar una vez más la conclusión obtenida mediante el factor de homogeneidad, ya que el valor calculado de $H_{\mathbf{x}}^{(n)} \cdot \mathbf{r}^{(n)}$ fue menor a 1 para $n = 2$ y $n = 3$, lo que indica una ventaja de QSS para simular este modelo.

7.4. Eficiencia de los métodos

Con el objetivo de comparar el costo computacional requerido por métodos basados en discretización temporal y por algoritmos basados en cuantificación de estados, simulamos el modelo epidemiológico propuesto pero esta vez teniendo en cuenta variaciones en el tamaño

de la población N_p .

Para esto, utilizamos los métodos de DOPRI, CVODE, LIQSS2, eLIQSS2 y CheQSS2 con una tolerancia relativa $\Delta Q_{\text{rel}} = 10^{-3}$ y una tolerancia absoluta $\Delta Q_{\text{abs}} = 10^{-6}$. Dado que el modelo de huésped es rígido, elegimos, dentro de los algoritmos de QSS, los linealmente implícitos (LIQSS $_n$ originales) para comparar con los nuevos algoritmos desarrollados (eLIQSS $_n$ y CheQSS $_n$). Por otro lado, dentro de los métodos clásicos de discretización temporal, elegimos como algoritmo implícito a CVODE (el uso de una matriz jacobiana sparse lo hace el más adecuado) y también utilizamos DOPRI, que es un método explícito, pero que puede llegar a ser menos costoso cuando el modelo es muy grande debido a que no tiene que invertir matrices en su ejecución.

La Tabla 7.3 y la Figura 7.4 resumen los número de pasos y los tiempos (en milisegundos) requeridos por cada algoritmo para completar la simulación en función del tamaño de la población.

n	N_p	N° de pasos			Tiempo de CPU (ms)		
		CheQSS $_n$	eLIQSS $_n$	LIQSS $_n$	CheQSS $_n$	eLIQSS $_n$	LIQSS $_n$
2	50	5,66E+05	6,71E+05	1,50E+06	332	390	803
	100	1,31E+06	1,47E+06	3,19E+06	867	973	1933
	500	6,52E+06	7,36E+06	1,61E+07	7478	8406	15629
	1000	1,32E+07	1,48E+07	3,22E+07	20240	22868	38519
3	50	2,59E+05	2,42E+05	5,01E+05	361	353	688
	100	5,61E+05	5,16E+05	1,05E+06	842	819	1592
	500	2,85E+06	2,64E+06	5,26E+06	6695	6580	12161
	1000	5,78E+06	5,35E+06	1,05E+07	18100	17552	31491
		CVODE					
5	50	2,34E+04			1768		
	100	3,82E+04			6523		
	500	1,50E+05			146074		
	1000	2,88E+05			590287		
		DOPRI					
5	50	3,06E+05			3024		
	100	3,15E+05			6264		
	500	3,65E+05			41512		
	1000	4,32E+05			112026		

Tabla 7.3: Número de pasos y tiempo de CPU en la simulación del modelo epidemiológico variando el tamaño de población.

De los resultados podemos obtener las siguientes conclusiones:

- En todos los casos resulta conveniente simular con algoritmos de QSS, aún sin limitar los órdenes de los métodos clásicos. Resultando entre 2 y 18 veces más rápidas las simulaciones con LIQSS que con los métodos clásicos y entre 5 y 33 veces más rápidas utilizando los nuevos algoritmos eLIQSS y CheQSS.
- CVODE presenta un mejor desempeño que DOPRI para un número reducido de individuos, ya que DOPRI requiere pasos muy pequeños debido a la rigidez del problema. Sin

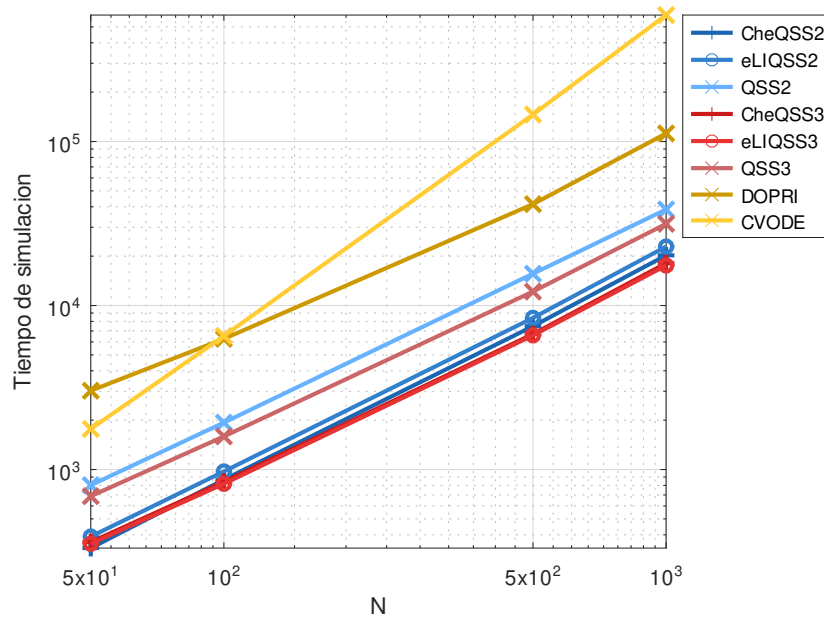


Figura 7.4: Tiempos de simulación según el tamaño de la población de los distintos algoritmos en la simulación del modelo epidemiológico

embargo, a medida que aumenta el tamaño del problema, CVODE se vuelve considerablemente más lento debido a la necesidad de invertir matrices.

- El número de pasos tanto en CVODE como en DOPRI crece en menor medida que los tiempos de simulación. Esto puede deberse a dos motivos, en primer lugar a medida que aumenta N_p , cada paso de simulación se hace más costoso al crecer la cantidad de actualizaciones que se hacen en cada uno. Al haber tantos eventos, gran parte de los pasos son debido a los mismos y son los que rigen el tamaño del paso, más que las actualizaciones de estado de variables continuas.
- Dentro de los algoritmos de QSS, nuevamente, tanto eLIQSS como CheQSS mejoraron la eficiencia del LIQSS original de manera similar, consiguiendo disminuir en general a la mitad los tiempos de simulación y el número de pasos realizados.

Capítulo 8

Conclusiones y trabajo futuro

8.1. Conclusiones

En esta tesis presentamos, en primer lugar, el desarrollo de nuevas herramientas y propiedades teóricas orientadas a estimar el costo computacional de simulaciones que utilizan diferentes enfoques de integración numérica (discretización temporal o cuantificación de estados) para aproximar numéricamente las trayectorias de los estados de un sistema en estudio.

Con este objetivo, comenzamos por extender el concepto de actividad para señales en tiempo continuo, introduciendo las definiciones de actividad instantánea, actividad local y actividad global. Mientras que la definición original de actividad estaba ligada al número mínimo de pasos requeridos por un método QSS para simular un sistema, estas nuevas métricas permiten además establecer relaciones con el número mínimo de pasos necesarios para que los algoritmos clásicos de integración numérica completen una simulación. En ese marco, desarrollamos expresiones que vinculan la actividad local y global con las cotas inferiores del número de pasos requeridos tanto por QSS como por los métodos clásicos.

Estas expresiones no sólo son válidas para los métodos QSS originales, sino también para formulaciones más generales, en las que las variables cuantificadas no necesariamente comienzan con el mismo valor ni con la misma derivada que las variables de estado correspondientes. Como consecuencia, las nuevas cotas inferiores obtenidas resultan más pequeñas que las originales, y a su vez menores que la cantidad de iteraciones reportadas por los métodos QSS originales, lo que sugiere la posibilidad de diseñar nuevos métodos capaces de realizar pasos de integración más largos.

A partir de estos desarrollos, introducimos formalmente el concepto de homogeneidad de la actividad para las trayectorias del sistema y caracterizamos este concepto mediante la definición de un *Factor de Homogeneidad*. Esta métrica toma valores cercanos a 1 cuando las trayectorias son homogéneas, y valores cercanos a $1/N$ cuando son heterogéneas (donde N es la dimensión del vector de estados). Utilizando esta herramienta, proporcionamos una demos-

tracción formal de la idea de que los métodos QSS tienden a presentar un mejor desempeño que los métodos clásicos cuando la actividad es heterogénea y los sistemas son dispersos, que luego corroboramos en casos de aplicación.

Además, extendimos estos resultados a sistemas que presentan discontinuidades, estableciendo cotas inferiores para el número de pasos y actualizaciones de estado tanto en simulaciones basadas en cuantificación como en simulaciones clásicas de tiempo discreto. Este análisis respalda formalmente la ventaja del uso de algoritmos QSS en sistemas con discontinuidades frecuentes.

En segundo lugar, a partir de la observación de que las cotas inferiores derivadas del concepto de actividad sugieren la posibilidad de realizar pasos de integración más largos, presentamos dos propuestas de mejoras de algoritmos LIQSS que buscan optimizar la cantidad de pasos necesarios para realizar una simulación. Para lograr esto, presentamos un procedimiento para desarrollar nuevos algoritmos LIQSS a partir del diseño del polinomio de diferencia $p_i(t) = x_i(t) - q_i(t)$. Utilizando este procedimiento, rediseñamos primero los métodos LIQSS originales de orden 1 a 3, obteniendo además una versión extendida (eLIQSS) que permite realizar pasos de integración más grandes. Con el objetivo de maximizar el tamaño del paso, diseñamos luego los algoritmos LIQSS basados en polinomios de Chebyshev (CheQSS) de orden 1 a 3. Dado que el polinomio de diferencia satisface la condición $|p_i(t)| = |x_i(t) - q_i(t)| \leq \Delta Q_i$, los nuevos algoritmos propuestos heredan todas las propiedades teóricas de los métodos QSS en cuanto a convergencia y cotas de error global.

Finalmente, estudiamos tres ejemplos, sobre los que aplicamos todos los resultados teóricos presentados. Los sistemas propuestos fueron:

- Un modelo continuo de una ecuación de advección-difusión-reacción unidimensional discretizada en el espacio que presenta rigidez.
- Un modelo con discontinuidades de una red neuronal pulsante de gran escala.
- Un modelo epidemiológico basado en agentes con discontinuidades, con rigidez y de gran escala.

En todos los casos realizamos un análisis exhaustivo, comenzando por la aplicación de las métricas propuestas de actividad, cota teórica mínima de pasos y factor de homogeneidad, bajo diferentes configuraciones de parámetros y tolerancias. Posteriormente, validamos las conclusiones teóricas mediante la realización de un amplio conjunto de experimentos de simulación. Para esto utilizamos en algunos casos el software Octave (para implementar algoritmos de discretización temporal de órdenes bajos como el BE01, TR12 y RK34) y el *Stand Alone QSS Solver* para utilizar métodos basados en discretización temporal como CVODE y DOPRI y distintos métodos de QSS. Para poder probar los nuevos métodos propuestos de eLIQSS y

CheQSS y comparar su eficiencia, los mismos fueron implementados en el *Stand Alone QSS Solver*.

Los resultados de estos experimentos corroboraron las conclusiones derivadas del análisis teórico de la actividad y del factor de homogeneidad de cada sistema. Pudimos observar que los métodos QSS permiten simular modelos con discontinuidades de gran tamaño con alta exactitud y costos computacionales muy bajos, aún sin modificaciones ni especialización de los algoritmos. Más aún, los nuevos algoritmos propuestos superaron el desempeño de los métodos QSS existentes y de los solvers clásicos de tiempo discreto.

8.2. Trabajo futuro

Un primer objetivo consiste en extender las definiciones de actividad y cálculo de la cota mínima de pasos, para incorporar el uso de tolerancias relativas, que en los métodos QSS se implementa mediante un tamaño de quantum ΔQ_i que varía en el tiempo de manera proporcional a la magnitud de la señal $x_i(t)$ [45].

Otra línea futura de trabajo consiste en el desarrollo de un algoritmo automático de selección de solver que calcule el factor de homogeneidad a medida que avanza la simulación y utilice su valor para decidir qué enfoque de integración numérica se espera que tenga un mejor desempeño. Pueden existir sistemas que tengan comportamiento más homogéneo durante un período de tiempo y más heterogéneo en otro, por lo que sería conveniente utilizar distintos enfoques en cada momento. Con esta motivación, se puede pensar en diseñar una estrategia de simulación en donde sea posible iterar entre ambos enfoques: el HF podría calcularse a medida que avanza la simulación (ya sea iniciando con un enfoque QSS o de tiempo discreto) y emplearse luego para cambiar de algoritmo según su valor, lo que podría conducir a un esquema similar a la selección automática de algoritmos utilizada en los solvers clásicos de ODEs [55].

Otra posibilidad interesante es extender los conceptos de actividad en modelos con aleatoriedad representados mediante ecuaciones diferenciales estocásticas, donde hay resultados prometedores respecto del uso de algoritmos de QSS [57].

La generalización de métodos de LIQSS presentada en esta tesis, abre distintas líneas para investigaciones futuras. El procedimiento de diseño puede emplearse para desarrollar nuevos algoritmos QSS con propiedades específicas. Además, un estudio más profundo del desempeño de los nuevos métodos en diferentes aplicaciones donde los métodos QSS suelen resultar convenientes sería necesario para, eventualmente, reemplazar los algoritmos LIQSS tradicionales por sus nuevas variantes eLIQSS y CheQSS.

Finalmente, como pudimos comprobar en la implementación de los métodos CheQSS, si bien el algoritmo busca optimizar la longitud de cada paso de simulación, este puede verse

interrumpido ante cambios en la derivada de una variable de estado provocados por la actualización de otras variables del sistema. Se puede pensar entonces, en alguna formulación de nuevos algoritmos, que busquen mejorar esta limitación.

Bibliografía

- [1] Rodrigo Assar and David J Sherman, *Implementing biological hybrid systems: Allowing composition and avoiding stiffness*, Applied Mathematics and Computation **223** (2013), 167–179.
- [2] Fernando J Barros, *Towards a theory of continuous flow models*, International Journal of General Systems **31** (2002), no. 1, 29–40.
- [3] Fernando J Barros and Bernard P Zeigler, *Model interoperability in the discrete event paradigm: Representation of continuous models*, Modeling and Simulation: Theory and Practice: A Memorial Volume for Professor Walter J. Karplus (1927–2001) (2003), 103–126.
- [4] Federico Bergero, Joaquín Fernández, Ernesto Kofman, and Margarita Portapila, *Time discretization versus state quantization in the simulation of a one-dimensional advection–diffusion–reaction equation*, Simulation **92** (2016), no. 1, 47–61.
- [5] Mariana Bergonzi, Rodrigo Castro, and Ernesto Kofman, *Modelado con retardos explícitos de la propagación de covid-19 en argentina*, Congreso Argentino de Control Automático, 2020.
- [6] ———, *Modelado con retardos explícitos de la propagación de covid-19 en argentina*, Encuentro Virtual de Jóvenes Investigadores organizado por la Asociación de Universidades Grupo Montevideo, 2020.
- [7] ———, *Activity homogeneity: A measure for comparing time discretization and state quantization in ode simulation*, SIMULATION (2025), in Press.
- [8] ———, *Discretización temporal vs cuantificación de estados: Homogeneidad de la actividad y discontinuidades*, Jornadas Argentinas de Informática, 2025.
- [9] ———, *¿discretización temporal o de estados? conveniencia según el concepto de actividad homogénea*, X Congreso de Matemática Aplicada, Computacional e Industrial, 2025.

- [10] Mariana Bergonzi, Rodrigo Castro, Esteban Lanzarotti, and Ernesto Kofman, *Modelado y simulación eficiente de modelos epidemiológicos multiescala*, Reunión de Trabajo en Procesamiento de la Información y Control, 2021.
- [11] ———, *Modelado y simulación eficiente de modelos epidemiológicos multiescala*, IX Congreso de Matemática Aplicada, Computacional e Industrial, 2023.
- [12] Mariana Bergonzi, Joaquín Fernández, Rodrigo Castro, Alexandre Muzy, and Ernesto Kofman, *Quantization-based simulation of spiking neurons: theoretical properties and performance analysis*, Journal of Simulation **18** (2024), no. 5, 789–812.
- [13] Mariana Bergonzi, Ezequiel Pecker-Marcosig, Ernesto Kofman, and Rodrigo Castro, *Discrete-time modeling of covid-19 propagation in argentina with explicit delays*, Computing in Science & Engineering **23** (2020), no. 1, 35–45.
- [14] ———, *Discrete-Time Modeling of COVID-19 Propagation in Argentina with Explicit Delays*, Computing in Science & Engineering **23** (2021), no. 1, 35–45.
- [15] Romain Brette, Michelle Rudolph, Ted Carnevale, Michael Hines, David Beeman, James M Bower, Markus Diesmann, Abigail Morrison, Philip H Goodman, Frederick C Harris, et al., *Simulation of networks of spiking neurons: a review of tools and strategies*, Journal of computational neuroscience **23** (2007), no. 3, 349–398.
- [16] Anthony N Burkitt, *A review of the integrate-and-fire neuron model: I. homogeneous synaptic input*, Biological cybernetics **95** (2006), no. 1, 1–19.
- [17] Rodrigo Castro, Mariana Bergonzi, Ezequiel Pecker Marcosig, Joaquín Fernández, and Ernesto Kofman, *Discrete-event simulation of continuous-time systems: evolution and state of the art of quantized state system methods*, Simulation **100** (2024), no. 6, 613–638.
- [18] Rodrigo Castro and Ernesto Kofman, *Activity of order n in continuous systems*, Simulation **91** (2015), no. 4, 337–348.
- [19] F.E. Cellier and E. Kofman, *Continuous System Simulation*, Springer, New York, 2006.
- [20] François E Cellier and Ernesto Kofman, *Continuous system simulation*, Springer Science & Business Media, 2006.
- [21] Scott D Cohen, Alan C Hindmarsh, and Paul F Dubois, *Cvode, a stiff/nonstiff ode solver in c*, Computers in physics **10** (1996), no. 2, 138–143.
- [22] Franco Di Pietro, Gustavo Migoni, and Ernesto Kofman, *Improving linearly implicit quantized state system methods*, Simulation **95** (2019), no. 2, 127–144.

- [23] John R Dormand and Peter J Prince, *A family of embedded runge-kutta formulae*, Journal of computational and applied mathematics **6** (1980), no. 1, 19–26.
- [24] John Wesley Eaton, David Bateman, Søren Hauberg, et al., *Gnu octave*, Network theory London, 1997.
- [25] Elmongi Elbellili, Philippe Blondeel, Daan Huybrechts, and Ben Lauwens, *Improving the modified linearly implicit quantized state system methods: step size, cycle detection, and linear approximation coefficient*, SIMULATION (2025), 00375497251351941.
- [26] Bard Ermentrout, *Type I membranes, phase resetting curves, and synchrony*, Neural computation **8** (1996), no. 5, 979–1001.
- [27] Joaquín Fernández and Ernesto Kofman, *A stand-alone quantized state system solver for continuous system simulation*, Simulation **90** (2014), no. 7, 782–799.
- [28] Nicolas Fourcaud-Trocmé, David Hansel, Carl Van Vreeswijk, and Nicolas Brunel, *How spike generation mechanisms determine the neuronal response to fluctuating inputs*, Journal of neuroscience **23** (2003), no. 37, 11628–11640.
- [29] Peter Fritzson and Vadim Engelson, *Modelica - A unified object-oriented language for system modeling and simulation*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1998.
- [30] Norbert Giambiasi and Jean Claude Carmona, *Generalized discrete event abstraction of continuous systems: GDEVs formalism*, Simulation Modelling Practice and Theory **14** (2006), no. 1, 47–70.
- [31] Norbert Giambiasi, Bruno Escude, and Sumit Ghosh, *GDEVs: A generalized discrete event specification for accurate modeling of dynamic systems*, Proceedings 5th International Symposium on Autonomous Decentralized Systems, IEEE, 2001, pp. 464–469.
- [32] Ernst Hairer, Syvert P Nørsett, and Gerhard Wanner, *Solving ordinary differential equations i. nonstiff problems*, Springer series in computational mathematics, 1993.
- [33] Ernst Hairer and Gerhard Wanner, *Solving ordinary differential equations. ii: Stiff and differential-algebraic problems*, Springer-Verlag, 1996.
- [34] Baris Hancioglu, David Swigon, and Gilles Clermont, *A dynamical model of human immune response to influenza a virus infection*, Journal of theoretical biology **246** (2007), no. 1, 70–86.
- [35] Alan C Hindmarsh, Peter N Brown, Keith E Grant, Steven L Lee, Radu Serban, Dan E Shumaker, and Carol S Woodward, *Sundials: Suite of nonlinear and differential/algebraic*

- equation solvers*, ACM Transactions on Mathematical Software (TOMS) **31** (2005), no. 3, 363–396.
- [36] Charles Hirsch, *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*, Elsevier, 2007.
- [37] Alan L Hodgkin and Andrew F Huxley, *A quantitative description of membrane current and its application to conduction and excitation in nerve*, The Journal of physiology **117** (1952), no. 4, 500.
- [38] EM Izhikevich, *Which model to use for cortical spiking neurons*, IEEE Transactions on Neural Networks **14** (2003), 1569–1572.
- [39] Rajanikanth Jammalamadaka, *Activity characterization of spatial models: Application to the discrete event solution of partial differential equations*, Master’s thesis, University of Arizona, Tucson, Arizona, USA (2003).
- [40] Camila Koatz, Mariana Bergonzi, and Ernesto Kofman, *Modelado orientado a objetos de sistemas poblacionales y epidemiológicos en tiempo discreto*, Jornadas Argentinas de Informática, 2025.
- [41] E. Kofman and S. Junco, *Quantized State Systems. A DEVS Approach for Continuous System Simulation*, Transactions of SCS **18** (2001), no. 3, 123–132.
- [42] Ernesto Kofman, *A second-order approximation for devs simulation of continuous systems*, Simulation **78** (2002), no. 2, 76–89.
- [43] ———, *Discrete event simulation of hybrid systems*, SIAM Journal on Scientific Computing **25** (2004), no. 5, 1771–1797.
- [44] ———, *A third order discrete event method for continuous system simulation*, Latin American applied research **36** (2006), no. 2, 101–108.
- [45] ———, *Relative error control in quantization based integration*, Latin American applied research **39** (2009), no. 3, 231–237.
- [46] Ernesto Kofman, Joaquín Fernández, and Denise Marzorati, *Compact sparse symbolic jacobian computation in large systems of odes*, Applied Mathematics and Computation **403** (2021), 126181.
- [47] Ernesto Kofman and Sergio Junco, *Quantized-state systems: a DEVS approach for continuous system simulation*, Transactions of The Society for Modeling and Simulation International **18** (2001), no. 3, 123–132.

- [48] LM Lapicque, *Recherches quantitatives sur l'excitation électrique des nerfs*, J. Physiol. Paris. **9** (1907), 620–635.
- [49] Randall J LeVeque, *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, SIAM, 2007.
- [50] Sven Erik Mattsson, Hilding Elmqvist, and Martin Otter, *Physical system modeling with modelica*, Control engineering practice **6** (1998), no. 4, 501–510.
- [51] Gustavo Migoni, Mario Bortolotto, Ernesto Kofman, and François E Cellier, *Linearly implicit quantization-based integration methods for stiff ordinary differential equations*, Simulation Modelling Practice and Theory **35** (2013), 118–136.
- [52] Gustavo Migoni, Ernesto Kofman, Federico Bergero, and Joaquín Fernández, *Quantization-based simulation of switched mode power supplies*, Simulation **91** (2015), no. 4, 320–336.
- [53] Alexandre Muzy, Rajanikanth Jammalamadaka, Bernard P Zeigler, and James J Nutaro, *The activity-tracking paradigm in discrete-event modeling and simulation: The case of spatially continuous distributed systems*, Simulation **87** (2011), no. 5, 449–464.
- [54] Petr Novak, Petr Kadera, and Manuel Wimmer, *Agent-based modeling and simulation of hybrid cyber-physical systems*, 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), IEEE, 2017, pp. 1–8.
- [55] Linda Petzold, *Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations*, SIAM journal on scientific and statistical computing **4** (1983), no. 1, 136–148.
- [56] Linda R Petzold, *Description of dassl: a differential/algebraic system solver*, Tech. report, Sandia National Labs., Livermore, CA (USA), 1982.
- [57] Noelia Pizzi, Mariana Bergonzi, and Ernesto Kofman, *Quantized state system methods for stochastic differential equations*, Computational and Applied Mathematics (2025), in Press.
- [58] Mehrshad Sadria and Anita T Layton, *Modeling within-host SARS-CoV-2 infection dynamics and potential treatments*, Viruses **13** (2021), no. 6, 1141.
- [59] Maximilian Schmidt, Rembrandt Bakker, Kelly Shen, Gleb Bezgin, Markus Diesmann, and Sacha Jennifer van Albada, *A multi-scale layer-resolved spiking network model of resting-state dynamics in macaque visual cortical areas*, PLOS Computational Biology **14** (2018), no. 10, e1006359.

- [60] Richard B Stein, *Some models of neuronal variability*, Biophysical journal **7** (1967), no. 1, 37–68.
- [61] Endre Süli and David F Mayers, *An introduction to numerical analysis*, Cambridge university press, 2003.
- [62] Gerard Olivar Tost and Olga Vasilieva, *Analysis, modelling, optimization, and numerical techniques: Icamí, san andrés island, colombia, november 2013*, vol. 121, Springer, 2015.
- [63] Emilia Vynnycky and Richard White, *An introduction to infectious disease modelling*, OUP oxford, 2010.
- [64] Gerhard Wanner and Ernst Hairer, *Solving ordinary differential equations ii*, vol. 375, Springer Berlin Heidelberg New York, 1996.
- [65] Joseph T Wu, Kathy Leung, Mary Bushman, Nishant Kishore, Rene Niehus, Pablo M de Salazar, Benjamin J Cowling, Marc Lipsitch, and Gabriel M Leung, *Estimating clinical severity of covid-19 from the transmission dynamics in wuhan, china*, Nature Medicine (2020), 1–5.
- [66] Bernard P Zeigler, *DEVS representation of dynamical systems: Event-based intelligent control*, Proceedings of the IEEE **77** (1989), no. 1, 72–80.
- [67] Bernard P. Zeigler, *DEVS theory of quantized systems*, Advanced simulation technology thrust DARPA contract (1998).
- [68] Bernard P Zeigler and Jong Sik Lee, *Theory of quantized systems: formal basis for DEV-S/HLA distributed simulation environment*, Enabling Technology for Simulation Science II, vol. 3369, SPIE, 1998, pp. 49–58.
- [69] Bernard P Zeigler, Alexandre Muzy, and Ernesto Kofman, *Theory of modeling and simulation: discrete event & iterative system computational foundations*, Academic press, 2018.
- [70] Bernard P Zeigler, Herbert Praehofer, and Tag Gon Kim, *Theory of modeling and simulation*, Academic press, 2000.
- [71] Bernard P Zeigler, Hae Sang Song, Tag Gon Kim, and Herbert Praehofer, *DEVS framework for modelling, simulation, analysis, and design of hybrid systems*, Hybrid Systems II 2, Springer, 1995, pp. 529–551.

Apéndice A

Demostración del Teorema 1

La demostración del Teorema 1 requiere algunos resultados auxiliares que se presentan a continuación.

A.1. Resultados auxiliares

Denotando T_n al polinomio de Chebyshev de grado n , la siguiente propiedad es un corolario del Teorema 8.6 de [61].

Corolario 1 (Corolario 8.1 de [61]). *Sea $n \geq 0$. Entre todos los polinomios mónicos¹ de grado $n + 1$, los polinomios $2^{-n}T_{n+1}$ y $-2^{-n}T_{n+1}$ son los que poseen la menor norma infinito en el intervalo $[-1, 1]$.*

A partir de este resultado, se enuncia el siguiente lema, que será utilizado en la demostración del teorema principal.

Lema 1. *Sea $\mathcal{P}_n$ la familia de polinomios mónicos de la forma $p(t) = t^n + a_{n-1}t^{n-1} + \dots + a_0$ con $a_i \in \mathbb{R}$. Definimos el tiempo de salida de $p(t) \in \mathcal{P}_n$ como*

$$t_{\text{out}}(p) \triangleq \inf\{t : t \geq 0 \wedge |p(t)| > 1\} \quad (\text{A.1})$$

y sea

$$t_{\text{máx}}(n) \triangleq 2^{\frac{2n-1}{n}} \quad (\text{A.2})$$

Entonces:

(i) *Existe el siguiente máximo:*

$$t_m(n) \triangleq \max_{p \in \mathcal{P}_n} t_{\text{out}}(p) \quad (\text{A.3})$$

que se denomina tiempo máximo de permanencia.

¹Un polinomio mónico es un polinomio no nulo cuyo coeficiente del término de mayor grado (coeficiente principal) es exactamente 1.

(ii) Para todo $n \geq 1$, se cumple que

$$t_m(n) = t_{\max}(n) \quad (\text{A.4})$$

y el polinomio

$$p_n^*(t) \triangleq T_n \left(\frac{2t - t_{\max}(n)}{t_{\max}(n)} \right) \quad (\text{A.5})$$

verifica $t_{\text{out}}(p_n^*) = t_m(n)$

Demostración. Notar primero que $p_n^*(t) \in \mathcal{P}_n$, ya que el coeficiente principal de $T_n(\tau)$ es 2^{n-1} y, por tanto, el coeficiente principal de $p_n^*(t)$ se obtiene como

$$\lim_{t \rightarrow \infty} \frac{p_n^*(t)}{t^n} = \lim_{t \rightarrow \infty} \frac{T_n \left(\frac{2t - t_{\max}(n)}{t_{\max}(n)} \right)}{t^n} = \frac{2^n}{t_{\max}(n)^n} \lim_{t \rightarrow \infty} \frac{T_n(t)}{t^n} = \frac{2^n}{2^{2n-1}} 2^{n-1} = 1$$

Asimismo, $t_{\text{out}}(p_n^*(t)) = t_{\max}(n)$ dado que

$$t \in [0, t_{\max}(n)] \iff \frac{2t - t_{\max}(n)}{t_{\max}(n)} \in [-1, 1]$$

y también

$$T_n(\tau) \in [-1, 1] \iff \tau \in [-1, 1]$$

Por lo tanto, tomando $\tau = \frac{2t - t_{\max}(n)}{t_{\max}(n)}$ y combinando las dos relaciones anteriores, resulta que

$$p_n^*(t) \in [-1, 1] \iff t \in [0, t_{\max}(n)]$$

Para completar la demostración, supongamos por contradicción que existe un polinomio $\hat{p}_n(t) \in \mathcal{P}_n$ tal que $t_{\text{out}}(\hat{p}_n) > t_{\max}(n)$. Entonces, definimos el siguiente polinomio:

$$\tilde{p}_n(\tau) = \hat{p}_n \left(\frac{t_{\text{out}}(\hat{p}_n)(\tau + 1)}{2} \right) \left(\frac{2}{t_{\text{out}}(\hat{p}_n)} \right)^n$$

Notar que cuando $\tau \in [-1, 1]$, el polinomio $\hat{p}_n$ se evalúa en $[0, t_{\text{out}}(\hat{p}_n)]$ verificando $|\hat{p}_n| \leq 1$, y entonces $\tilde{p}_n(\tau)$ satisface:

$$|\tilde{p}_n(\tau)| \leq \left(\frac{2}{t_{\text{out}}(\hat{p}_n)} \right)^n < \left(\frac{2}{t_{\max}(n)} \right)^n$$

para $\tau \in [-1, 1]$. Denotando $\|p(\tau)\|_\infty$ a la norma infinito de un polinomio $p(\tau)$ en el intervalo $[-1, 1]$, y reemplazando el valor de $t_{\max}(n)$, podemos reescribir:

$$\|\tilde{p}_n(\tau)\|_\infty < \left(\frac{2}{2^{\frac{2n-1}{n}}}} \right)^n = 2^{-(n-1)} \quad (\text{A.6})$$

Observar también que la norma infinito del polinomio mónico con mínima norma infinito,

según el Corolario 8.1 de [61], es

$$\|2^{-(n-1)}T_n(\tau)\|_\infty = 2^{-(n-1)} \quad (\text{A.7})$$

Dado que $\tilde{p}_n(\tau)$ es un polinomio mónico (su coeficiente principal es 1), las expresiones de las Ecs. (A.6) y (A.7) contradicen el resultado del corolario mencionado.

De esta forma, la demostración queda completa, ya que $t_{\text{out}}(p_n^*(t)) = t_{\text{máx}}(n)$ y no existe ningún polinomio en $\mathcal{P}_n$ con un valor mayor de t_{out} . $\square$

A.2. Demostración del Teorema 1

Teniendo en cuenta los resultados obtenidos en el Lema 1, presentamos la demostración del Teorema 1:

Demostración. Sea b_n el coeficiente principal de $x_i(t)$. Definimos el polinomio:

$$p(t) \triangleq \text{sign}(b_n)(x_i(t) - q_i(t)) \quad (\text{A.8})$$

Notar que $p(t)$ es un polinomio de orden n , cuyo coeficiente principal es $|b_n|$. A partir de $p(t)$ definimos el siguiente polinomio:

$$\hat{p}(t) \triangleq \frac{1}{\Delta Q_i} \cdot p \left(\left(\frac{\Delta Q_i}{|b_n|} \right)^{\frac{1}{n}} \cdot t + t_j \right) \quad (\text{A.9})$$

Observar que:

- El coeficiente principal de $\hat{p}(t)$ es 1. En consecuencia, $\hat{p}(t) \in \mathcal{P}_n$ (donde el conjunto $\mathcal{P}_n$ es el definido en el Lema 1).
- Al elegir adecuadamente $q_i(t)$ en la Ec. (A.8), puede construirse cualquier polinomio $\hat{p}(t)$ perteneciente al conjunto $\mathcal{P}_n$.
- Por lo tanto, según el Lema 1, existe un polinomio $q_i(t)$ para el cual el tiempo de permanencia que satisface la condición $|\hat{p}(t)| \leq 1$ es $t_{\text{máx}}(n)$, definido en la Ec. (A.3). Además, no existe ningún polinomio $q_i(t)$ cuyo tiempo de permanencia sea mayor.

Notar también que $|p(t)| \leq \Delta Q_i$ en $[t_j, t_j + \Delta t] \Leftrightarrow |\hat{p}(t)| \leq 1$ en $\left[0, \left(\frac{|b_n|}{\Delta Q_i}\right)^{\frac{1}{n}} \Delta t\right]$.

Por lo tanto, existirá un polinomio $q_i(t)$ que verifique $|p(t)| \leq \Delta Q_i$ en $[t_j, t_j + \Delta t]$ si y sólo si

$$\left(\frac{|b_n|}{\Delta Q_i}\right)^{\frac{1}{n}} \Delta t \leq t_{\text{máx}} = 2^{\frac{2n-1}{n}} \quad (\text{A.10})$$

o, de manera equivalente:

$$\Delta t \leq 2^{\frac{2n-1}{n}} \left(\frac{\Delta Q_i}{|b_n|} \right)^{\frac{1}{n}} \quad (\text{A.11})$$

Por otro lado, la actividad instantánea local de $x_i(t)$ en t_j es

$$a_{x_i}^{(n)}(t_j) = \left| \frac{\frac{d^n x_i(\tau)}{d\tau^n}}{n!} \right|^{1/n} = \left(\frac{|b_n| \cdot n!}{n!} \right)^{\frac{1}{n}} = |b_n|^{\frac{1}{n}} \quad (\text{A.12})$$

y reemplazando esta expresión en la Ec. (A.11) obtenemos:

$$\Delta t \leq 2^{\frac{2n-1}{n}} \cdot \frac{\Delta Q_i^{\frac{1}{n}}}{a_{x_i}^{(n)}(t_j)} \quad (\text{A.13})$$

lo que completa la demostración. □