

APLICACIONES DEL MÉTODO LDG A SISTEMAS QUÍMICOS Y BIOLÓGICOS CON ESTRUCTURAS DE TURING

Por

Iván Yessel Henríquez Rivera

Tesis sometida en cumplimiento parcial de los requerimientos para el grado de

MAESTRÍA EN CIENCIAS

en

MATEMÁTICAS (APLICADA)

UNIVERSIDAD DE PUERTO RICO
RECINTO UNIVERSITARIO DE MAYAGÜEZ

2013

Aprobada por:

Lev Steinberg, Ph.D
Miembro, Comité Graduado

Fecha

Gustavo Gutierrez, Ph.D
Miembro, Comité Graduado

Fecha

Paul Castillo, Ph.D
Presidente, Comité Graduado

Fecha

Pedro I. Rivera Vega, Ph.D
Representante de Estudios Graduados

Fecha

Omar Colón, Ph.D
Director de Departamento

Fecha

Resumen de Disertación Presentado a Escuela Graduada
de la Universidad de Puerto Rico como requisito parcial de los
Requerimientos para el grado de Maestría en Ciencias

APLICACIONES DEL MÉTODO LDG A SISTEMAS QUÍMICOS Y BIOLÓGICOS CON ESTRUCTURAS DE TURING

Por

Iván Yessel Henríquez Rivera

2013

Consejero: Paul E. Castillo
Departamento: Matemática

En su trabajo pionero sobre morfogénesis [34] en 1952, A. Turing propuso un modelo teórico de dos especies químicas en el cual la evolución de un estado estacionario heterogéneo (estructura de Turing) desde un estado estacionario homogéneo es debido a la difusión. Desde la primera evidencia física de la existencia de tales estructuras de Turing el modelamiento teórico y diseño experimental de sistemas biológicos y químicos con estructuras de Turing han sido testigos de un progreso considerable. Típicamente tales modelos son matemáticamente formulados como un sistema de ecuaciones no lineales de reacción difusión

$$\frac{\partial}{\partial t} \mathbf{c}(t, \mathbf{x}) = (D \otimes \Delta) \mathbf{c}(t, \mathbf{x}) + \mathbf{r}(\mathbf{c}(t, \mathbf{x})), \quad (1)$$

donde \mathbf{c} es un vector en \mathbb{R}^p que representa la concentración de p especies químicas, D_p es la matriz de coeficientes de difusión la cual generalmente se asume ser diagonal y constante; y \mathbf{r}_p es el operador de reacción que acopla el sistema. La naturaleza no lineal de las reacciones químicas hace difícil (de no ser imposible) expresar la solución del sistema con una fórmula exacta. Esto ha motivado el interés en el desarrollo de técnicas computacionales para la simulación numérica de tales sistemas de

ecuaciones diferenciales en derivadas parciales.

En este trabajo se realiza una comparación de dos versiones del método *Local discontinuous Galerkin* (LDG) para la discretización en espacio: la clásica y la de disipación mínima; la integración en tiempo es llevada a cabo mediante métodos de descomposición de operador o un método IMEX. Usando un método de Galerkin discontinuo para la discretización en espacio el sistema global no lineal es desacoplado en cada paso de tiempo en un conjunto de pequeños sistemas no lineales que pueden ser resueltos en paralelo. Un marco computacional fue implementado para discretizaciones espaciales de alto orden en 2D para sistemas no lineales de reacción difusión con un número arbitrario de especies. La comparación es llevada a cabo usando un sistema lineal de reacción difusión, el modelo de Schnakenberg y un modelo de pigmentación de pieles de animales.

Abstract of Disertation Presented to the Graduate School
of the University of Puerto Rico in Partial Fulfillment of the
Requirements for the Degree of Master of Sciences

**APPLICATIONS OF THE LDG METHOD TO CHEMICAL AND
BIOLOGICAL SYSTEMS WITH TURING STRUCTURES**

By

Iván Yessel Henríquez Rivera

December 2013

Chair: Paul E. Castillo

Major Department: Mathematical Sciences

In the pioneering work on morphogenesis [34] in 1952, A. Turing proposed a theoretical model consisting of two chemical species in which the evolution of a spatially stable heterogeneous steady state (Turing structures) from a homogeneous stationary state is driven by diffusion. Since the first physical evidence of the existence of such structures theoretical modeling and experimental design of chemical and biological systems revealing Turing structures has witnessed considerable progress. Typically such models are mathematically formulated as a system of non-linear reaction diffusion equations

$$\frac{\partial}{\partial t} \mathbf{c}(t, \mathbf{x}) = (D \otimes \Delta) \mathbf{c}(t, \mathbf{x}) + \mathbf{r}(\mathbf{c}(t, \mathbf{x})), \quad (2)$$

where \mathbf{c} is a vector in \mathbb{R}^p representing the concentration of p chemical species, D_p is a matrix of diffusion coefficients which is generally assumed diagonal and constant; and \mathbf{r}_p couples the system with p chemical reactions. The non-linearity nature of chemical kinetics makes it difficult (if not impossible) to express the solution of the system in closed form. This motivated an increase of interest in the development of computational techniques for the numerical simulation of such systems of partial

differential equations.

In this work a numerical comparison of two versions of the Local Discontinuous Galerkin (LDG) method for the spatial discretization: the classical and the minimal dissipation LDG; coupled with an operator splitting or an IMEX time stepping technique is performed. By using a discontinuous Galerkin discretization the global non-linear system of equations is decoupled at each time step into a set of small non-linear systems which can be solved in parallel. A computational framework was implemented for high order discretizations in 2D for non-linear reaction diffusion systems with an arbitrary number of species. The comparison is carried out using a linear reaction diffusion, the classical Schnakenberg model problem and a coat pigmentation model.

Copyright © 2013

por

Iván Yessel Henríquez Rivera

A Dios, y a mi familia

AGRADECIMIENTOS

A Dios quien ha sido mi más significativo apoyo en todos los aspectos de mi vida.

Al Dr. Paul Castillo por su excelente asesoramiento, enseñanza, y por su valiosa amistad en todo tiempo.

Al Dr. Julio Barragán, y su familia, por su apoyo, consejos y por su valiosa amistad.

Al Departamento de Matemática de la Universidad de Puerto Rico en Mayagüez por brindarme esta gran oportunidad.

A mis amigos de Puerto Rico, Costa Rica, Colombia, Perú y Honduras por su ayuda, compañía y buenos deseos.

A mis padres, y mis hermanos, por su cariño, ayuda y apoyo incondicional en todo momento.

A mi hijo, Santiago, por ser mi mayor alegría y motivación.

Índice general

	<u>página</u>
RESUMEN EN ESPAÑOL	II
ABSTRACT ENGLISH	IV
AGRADECIMIENTOS	VIII
LISTA DE TABLAS	XI
LISTA DE FIGURAS	XIII
LISTA DE ABREVIATURAS	XV
LISTA DE SIMBOLOS	XVI
1. Introducción	1
1.1. Sistemas de reacción-difusión	1
1.2. Solución de los sistemas de reacción-difusión	2
1.3. Objetivo de la tesis	5
2. Método LDG	7
2.1. Método LDG original	7
2.2. Método LDG de mínima disipación	10
2.3. Reacción para p especies	13
3. Integración en tiempo	15
3.1. Métodos de descomposición de operador	16
3.2. Métodos IMEX	19
4. Aspectos computacionales	21
4.1. Rutinas de discretización en espacio	22
4.2. Rutinas de discretización en tiempo	23
4.3. Solución de sistemas lineales	29
4.4. Solución de sistemas no lineales	30
5. Experimentos numéricos	34
5.1. Problema escalar lineal	34
5.1.1. Precisión	36
5.1.2. Alto orden en espacio	37

5.1.3.	Estabilidad	40
5.2.	Modelo de Schnakenberg	43
5.2.1.	Precisión	44
5.3.	Modelo de pigmentación	55
5.3.1.	Precisión	55
6.	Conclusiones y trabajo a futuro	64
6.1.	Conclusiones	64
6.2.	Trabajos a futuro	67

LISTA DE TABLAS

<u>Tabla</u>	<u>pagina</u>
2-1. Error $\ u - u_h \ _0$ para $p = 1$	12
2-2. Error $\ u - u_h \ _0$ para $p = 2$	12
2-3. Error $\ u - u_h \ _0$ para $p = 3$	12
4-1. Interfaz de la clase <i>LDG2D</i>	24
4-2. Interfaz de la clase <i>DGReaction</i>	25
4-3. Interfaz de la clase abstracta <i>Operator</i>	27
4-4. Interfaz de la clase <i>SSOSI</i>	28
4-5. Interfaz de la clase BlockCSR.	31
4-6. Interfaz de la clase PCGM.	32
4-7. Interfaz de la clase Newton.	33
5-1. Descripción de mallas para el problema lineal escalar de reacción difusión.	35
5-2. Error L^2 con polinomios de grado $p = 1$. $\lambda = -1.0$, $\sigma = 1/2\pi^2$, $T = 1.0$, $\tau \approx 11h_{min}$	37
5-3. Error L^2 con polinomios de grado $p = 1$. $\lambda = 0.9$, $\sigma = 1/2\pi^2$, $T = 1.0$, $\tau \approx 11h_{min}$	38
5-4. Error L^2 , en $T = 1.0$, para discretizaciones espaciales de grados P1, P2, P3 y P4. En cada caso la malla fue escogida de tal manera que los errores sean cercanos. El resto de los parámetros son $\tau = 2.5e-03$, $\sigma = 1/(2\pi^2)$ y $\lambda = -1.0$	39
5-5. Tiempos de ejecución (en segundos) correspondientes a la Tabla 5-4.	39
5-6. Convergencia con polinomios de grado $p=2$ en espacio	41
5-7. Convergencia con polinomios de grado $p=3$ en espacio	42
5-8. Tiempos de ejecución (en segundos) con polinomios de grado $p = 2$, correspondientes a la Tabla 5-6	43

5–9. Tiempos de ejecución (en segundos) con polinomios de grado $p = 3$, correspondientes a la Tabla 5–7	44
5–10. Reacción lineal y rígida: 2048 celdas con $\tau = 0.25$	44
5–11. Reacción lineal y rígida: 8192 celdas con $\tau = 0.25$	45
5–12. Descripción de mallas para el modelo de Schnakenberg.	45
5–13. Tiempos de cómputo (en segundos) para discretizaciones espaciales con el LDG-0.	50
5–14. Tiempos de cómputo (en segundos) para discretizaciones espaciales con el LDG-1.	50
5–15. Tiempos de cómputo (en segundos) para discretizaciones espaciales con el MDLDG-1.	50
5–16. Descripción de mallas para el modelo de pigmentación.	57
5–17. Tiempos de cómputo (en segundos) para discretizaciones espaciales con el LDG-0.	57
5–18. Tiempos de cómputo (en segundos) para discretizaciones espaciales con el LDG-1.	57
5–19. Tiempos de cómputo (en segundos) para discretizaciones espaciales con el MDLDG-1.	58

LISTA DE FIGURAS

<u>Figura</u>	<u>pagina</u>
1-1. Definición de una estructura de Turing	2
2-1. Elección de β en el MDLDG. El parámetro de estabilidad es nulo en todas las aristas excepto en Γ_+	10
2-2. Número de condicionamiento espectral $\kappa(\eta)$ para $M - \Delta$. De izquierda a derecha con polinomios de grado $p = 1, 2, 3$	14
4-1. Esquema simplificado del marco computacional.	22
4-2. Clases que heredan de <i>Operator</i>	29
4-3. Diagrama simplificado del esquema CSR por bloques	30
5-1. Malla inicial m_0 (izquierda) y sus dos refinamientos sucesivos m_1 (centro) y m_2 (derecha).	35
5-2. Mallas no estructuradas. De izquierda a derecha: mallas m_1 , m_3 y m_5	45
5-3. Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 1$. En todos los casos se observa convergencia en tiempo de segundo orden.	47
5-4. Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 2$. En todos los casos se observa convergencia en tiempo de segundo orden.	48
5-5. Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 3$. En todos los casos se observa convergencia en tiempo de segundo orden.	49
5-6. Evolución en tiempo de la concentración c_1 con los métodos SS-OS con reacción explícita y MDLDG-1.	51
5-7. Evolución en tiempo de la concentración c_1 con los métodos TR-OS con reacción explícita y MDLDG-1.	52
5-8. Evolución en tiempo de la concentración c_1 con los métodos IMEX-BDF con reacción explícita y MDLDG-1.	53

5–9. Evolución en tiempo de la concentración c_1 con los métodos IMEX-RK con reacción explícita y MDLDG-1.	54
5–10. Mallas no estructuradas. De izquierda a derecha: mallas m_1 , m_3 y m_5	56
5–11. Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 1$. En todos los casos se observa convergencia en tiempo de segundo orden.	59
5–12. Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 2$. En todos los casos se observa convergencia en tiempo de segundo orden.	60
5–13. Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 3$. En todos los casos se observa convergencia en tiempo de segundo orden.	61
5–14. Patrón para un tiempo final $T = 3000$ formado por la concentración del conmutador c_3 usando los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con discretización en espacio por el MDLDG-1.	62
5–15. Patrón para un tiempo final $T = 3000$ formado por la concentración del conmutador c_3 usando los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con discretización en espacio por el MDLDG-1.	63

LISTA DE ABREVIATURAS

RD	Reacción-Difusión.
DG	Discontinuous Galerkin.
MGD	Método de Galerkin Discontinuo.
LDG	Local Discontinuous Galerkin.
MD-LDG	Minimal Dissipation Local Discontinuous Galerkin
DO	Descomposición de Operador.
SS-OS	Descomposición simétrica de Strang con Crank-Nicolson.
TR-OS	Descomposición simétrica de Strang Trapezoidal.
IMEX-RK	Runge-Kutta implícito-explicito.
IMEX-BDF	BDF implícito-explicito.
BDF	Backward Differentiation Formula
MDO	Método de Descomposición de Operador.
IMEX	Implícito-Explícito.
CSR	Compressed Sparse Row.

LISTA DE SIMBOLOS

\mathbb{R}	Conjunto de los números reales.
Ω	Dominio acotado en \mathbb{R}^2 ó \mathbb{R}^3 .
$\partial\Omega$	Frontera de un dominio Ω .
Δ	Operador Laplaciano.
Δ_h	Discretización en espacio del Laplaciano.
Δ_h^{LDG}	Discretización en espacio del Laplaciano por el método LDG.
Δ_h^{mdLDG}	Discretización en espacio del Laplaciano por el método MD-LDG.
\mathbf{r}	Operador de reacción multicomponente.
\mathbf{r}_h	Discretización del operador de reacción multicomponente.
D	Matriz de coeficientes de difusión para cada especie química.
\mathbf{c}	Función vectorial representando p especies químicas: $\mathbf{c} = (c_1, \dots, c_p)^T$.
\mathbf{C}_h	Aproximación de la función vectorial \mathbf{c} .
$\mathbb{P}_{p_T}(T)$	espacio local de polinomios de grado, a lo más $p_T \geq 1$, en la celda T .
\widehat{c}_h	Flujo numérico de c_h .
$\widehat{\mathbf{q}}_h$	Flujo numérico de \mathbf{q}_h .
$s(c_h)$	Término de estabilización.

Capítulo 1

INTRODUCCIÓN

1.1. Sistemas de reacción-difusión

Los sistemas de reacción-difusión (RD) son una excelente idealización de muchos procesos que involucran la interacción de múltiples especies químicas o biológicas moviéndose aleatoriamente en un medio físico. Estos sistemas se describen matemáticamente mediante la siguiente ecuación diferencial, la cual se cumple para todo $(t, \mathbf{x}) \in [0, T] \times \Omega$:

$$\frac{\partial}{\partial t} \mathbf{c}(t, \mathbf{x}) = (D \otimes \Delta) \mathbf{c}(t, \mathbf{x}) + \mathbf{r}(\mathbf{c}(t, \mathbf{x})), \quad (1.1)$$

donde, $\mathbf{c}(t, \mathbf{x}) = (c_1, \dots, c_p)^T$ es una función vectorial, y cada componente, $c_i(t, \cdot)$, representa la distribución espacial de la i -ésima especie en el dominio abierto y acotado Ω contenido, ya sea en \mathbb{R}^2 o \mathbb{R}^3 ; en el tiempo $t \in (0, T]$. El término $(D \otimes \Delta)$ es el operador de difusión multicomponente, donde $D \in \mathbb{R}^{p \times p}$ es la matriz de coeficientes de difusión y Δ es el operador Laplaciano para una función escalar. Por lo general, estos modelos asumen que D es una matriz diagonal de coeficientes constantes cuyas entradas en la diagonal principal son positivas. La interacción (química o biológica) está determinada por el operador de reacción multicomponente, $\mathbf{r}(\mathbf{c}) = (r_1(\mathbf{c}), \dots, r_p(\mathbf{c}))^T$, el cual suele ser por lo general no lineal.

En su trabajo pionero sobre morfogénesis [34], A. Turing demostró la posibilidad de convertir algunos de los estados homogéneos estables de un sistema químico en

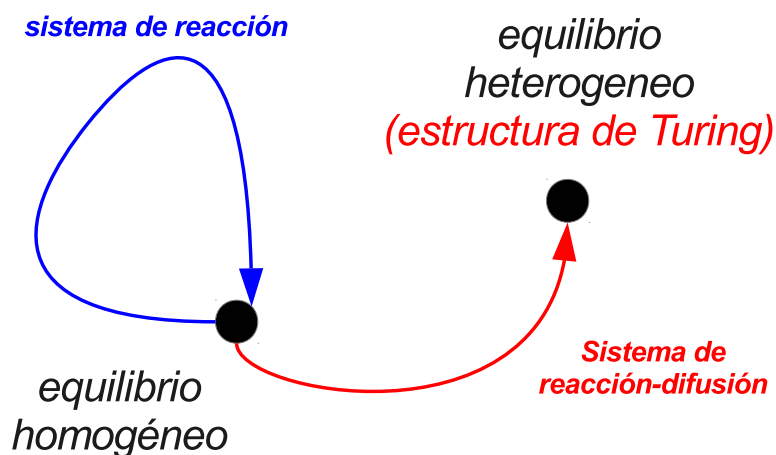


Figura 1–1: Definición de una estructura de Turing

estados inestables mediante un simple proceso de difusión. Dicha inestabilidad puede evolucionar a un estado estacionario espacial heterogéneo, mejor conocido como *estructura de Turing*, véase la Figura 1–1. Desde la primera evidencia física sobre la existencia de tales estructuras [4], el modelamiento teórico y diseño experimental de sistemas químicos que presentan estructuras de Turing han sido testigos de un progreso considerable. Una revisión sobre aplicaciones de modelos de reacción-difusión a sistemas químicos puede ser encontrada en [25, 27] y más recientemente en [36].

En biología, la aceptación de mecanismos de Turing para explicar los patrones espacialmente organizados en sistemas pluricelulares ha sido lentamente revitalizada en la última década. Varios modelos de reacción-difusión han sido propuestos recientemente para predecir una amplia variedad de mecanismos morfológicos en la formación de órganos y tejidos. Una interesante y extensa colección de modelos aplicados a sistemas biológicos puede ser encontrada en el artículo clásico [22] y más recientemente en [26].

1.2. Solución de los sistemas de reacción-difusión

La solución de sistemas RD, así como otras ecuaciones diferenciales parciales

(EDPs) no lineales con geometrías irregulares, es un problema complicado de resolver. Salvo en algunos casos particulares, es muy raro (casi imposible) encontrar una expresión analítica explícita para la representación de la solución. Por lo que en la práctica, es obligatorio el uso de métodos numéricos para obtener una aproximación de dicha solución. La búsqueda y evaluación de nuevas y mejores técnicas numéricas para dichos sistemas es motivada por el acelerado desarrollo de nuevos modelos cada vez más complejos y realistas.

Para obtener una aproximación de la solución de estos sistemas se procede primero por la discretización en espacio, obteniendo un sistema de ecuaciones diferenciales ordinarias en tiempo. Este sistema puede ser no lineal. Luego, se resuelve este sistema empleando un esquema de discretización en tiempo. Este marco de aproximación se conoce como *método de las líneas* y es muy popular debido a su simpleza y flexibilidad para combinar diferentes discretizaciones en espacio y tiempo. Sin embargo, es bien sabido que ciertas combinaciones pueden ser mejores que otras, eso dependerá de la naturaleza del problema en cuestión.

Una de las mayores dificultades en la solución de los sistemas RD para estructuras de Turing es la discretización en tiempo. El uso de un método explícito es poco eficiente debido a la excesiva restricción en el tamaño de paso en tiempo impuesta por la rigidez de, al menos, el término de difusión. Por otro lado, el uso de un método completamente implícito requeriría resolver, en cada paso de tiempo, un sistema simultáneo de gran tamaño de ecuaciones no lineales. Una de las mayores desventajas de este procedimiento es el tener que resolver, en cada iteración de Newton, sistemas lineales cuya matriz asociada, el Jacobiano de dicho sistema no lineal, podría ser una matriz no simétrica. En estos casos, los métodos de *descomposición de operador* (DO) son ideales, ya que permiten desacoplar los términos

de difusión y reacción durante el proceso de integración en tiempo. Así, se puede aplicar un método implícito para la difusión y un método explícito para la reacción. Además, si para la discretización espacial se utilizó un método discontinuo, entonces el problema de reacción es local en cada celda de la malla. Otra alternativa similar a los métodos de descomposición de operador son los métodos *implícito-explícito* (IMEX) donde los términos rígidos se integran de manera implícita y los términos menos rígidos se integran de manera explícita. Así, los métodos IMEX son ideales en sistemas RD donde la reacción es considerablemente menos rígida que la difusión.

La discretización espacial presenta una amplia gama de opciones, las cuales se pueden agrupar esencialmente en tres grandes categorías: la primera consiste de métodos basados en diferencias finitas. Estos son los más fáciles de implementar y han sido previamente investigados para la formación de patrones [29]. Recientemente un nuevo método ha sido propuesto para la simulación de patrones de Turing en electrodeposición química [32]. Una importante observación señalada por varios autores [21, 23] es que los esquemas de diferencia finita pueden influenciar la solución numérica en dominios que crecen continuamente. La segunda categoría que goza de mucha popularidad debido a su flexibilidad para manejar geometrías complejas es el método de elemento finito. Aplicaciones de esta técnica a problemas no lineales de reacción-difusión pueden encontrarse en [23, 24] y en sus referencias. Finalmente en la tercera categoría se encuentran los métodos conocidos como métodos Galerkin discontinuos (MGD), los cuales han ganado mucha popularidad en las últimas décadas debido a sus propiedades de aproximación y su gran versatilidad y flexibilidad computacional. En estos métodos no se impone continuidad entre celdas adyacentes por lo que el grado de aproximación puede ser variable en cada celda y pueden utilizarse mallas arbitrarias, particularmente, mallas con nodos colgantes; haciendo que el método sea ideal para refinamiento-*hp*. Otra propiedad interesante es que la matriz

de masa es diagonal por bloques, por lo tanto fácil de invertir. Esto es conveniente en los métodos de integración en tiempo explícitos para problemas transitorios ya que se evita la técnica de agrupamiento conocida en inglés como “*mass lumping*” (utilizada comúnmente en el método de Elemento Finito continuo). La aplicación del método de Volúmen Finito, el cual puede ser visto como un método GD de bajo orden, combinado con una técnica de descomposición, fue reportado por primera vez en [35] para la simulación de un modelo de quimiotaxis. Recientemente, en [17] el método Interior Penalty fue usado para un modelo de quimiotaxis y haptotaxis y en [37] se aplicó un nuevo método GD para sistemas de reacción-difusión en problemas de desarrollo biológico.

1.3. Objetivo de la tesis

En este trabajo se considera el método *Local Discontinuous Galerkin* (LDG) originalmente propuesto por Cockburn y Shu en [16] para problemas transitorios no lineales de convección-difusión. El operador de difusión es discretizado espacialmente usando dos versiones: el LDG *original* presentado en [10] para un problema modelo elíptico y una versión más reciente, llamada *Minimal Dissipation* LDG o (MD-LDG), analizada en [14]. La convergencia del método LDG aplicado a diferentes tipos de ecuaciones ha sido presentada en diversos trabajos previos. Un estimado de error *a priori* para la versión *h* y *hp* del LDG aplicado a un problema transitorio lineal de convección-difusión se encuentra en [5, 10] para una dimensión y en [13] para varias dimensiones. La preferencia del uso del LDG sobre otros métodos GD está basada en la comparación numérica y teórica realizada en [6] para un problema modelo elíptico. Ahí se probó teóricamente que el método LDG es más estable y los experimentos numéricos mostraron una mejor precisión para este método.

El presente trabajo tiene como propósito fundamental realizar una comparación numérica del rendimiento de ambas versiones del método LDG aplicadas a problemas RD. Se consideran aspectos tales como calidad en la precisión numérica para diferentes tipos de problemas modelo, costo computacional, tiempo de ensamblado etc. Por otro lado también se comparan dos esquemas de integración en tiempo: uno basado en descomposición de operadores y otro basado en IMEX.

Para lograr el objetivo principal se desarrolló un marco computacional en 2D para resolver sistemas de reacción-difusión con reacción no lineal sujetas a condiciones de frontera de tipo Dirichlet y Neumann. El marco permite resolver sistemas con un número arbitrario de especies químicas y aproximaciones espaciales de alto orden. El código es orientado a objetos y se utilizó el lenguaje de programación C++ para la implementación de las diferentes clases abstractas.

Capítulo 2

MÉTODO LDG

Siguiendo el marco del método de las líneas, en este capítulo se discute la discretización espacial de los sistemas RD. Particularmente, se considera el método LDG y su variante de disipación mínima (MD-LDG). El sistema semidiscreto resultante para p especies químicas tendrá la forma

$$(I_p \otimes M)\dot{\mathbf{C}}_h = (D \otimes \Delta_h)\mathbf{C}_h + \mathbf{r}_h(\mathbf{C}_h), \quad t \in (0, T], \quad (2.1)$$

donde “ \otimes ” denota el producto *Kronecker*, I_p es la matriz identidad de orden p , M es la matriz de masa correspondiente a una especie, Δ_h es la matriz que corresponde a la discretización espacial del operador Laplaciano y $\mathbf{r}_h(\mathbf{C}_h)$ es la discretización espacial del operador de reacción $\mathbf{r}(\mathbf{c})$.

En las siguientes secciones se discutirá el proceso de discretización en espacio por el LDG para obtener los operadores discretos Δ_h y \mathbf{r}_h .

2.1. Método LDG original

Por simplicidad, se describirá el proceso de discretización para un sistema de reacción difusión con una especie, en el dominio abierto y acotado $\Omega \subset \mathbb{R}^2$. Considere la ecuación

$$c_t = d\Delta c + r(c), \quad \text{en } \mathbb{R}^+ \times \Omega,$$

donde d es una constante positiva. En la frontera, $\partial\Omega$, la función c satisface condiciones de flujo cero.

Introduciendo una nueva variable auxiliar $\mathbf{q} = -d\nabla c$, se obtiene el siguiente sistema de primer orden

$$\begin{cases} d^{-1}\mathbf{q} + \nabla c = 0 \\ c_t + \nabla \circ \mathbf{q} - r(c) = 0 \end{cases}$$

Para una celda T , se denota por $\mathbb{P}_{p_T}(T)$ al espacio de polinomios de grado, a lo más p_T con $p_T \geq 1$. Se introducen las funciones $c_h(\cdot, t) \in \mathbb{P}_{p_T}(T)$ y $\mathbf{q}_h(\cdot, t) \in \mathbb{P}_{p_T}(T)^2$ las cuales son aproximaciones de $c(\cdot, t)$ y $\mathbf{q}(\cdot, t)$. Multiplicando por funciones de prueba $(\mathbf{v}(\cdot, t), w(\cdot, t)) \in \mathbb{P}_{p_T}(T)^2 \times \mathbb{P}_{p_T}(T)$ el problema de elemento finito en la celda T se escribe

$$\begin{aligned} \int_T \sigma^{-1} \mathbf{q}_h \cdot \mathbf{v} + \oint_{\partial T} \widehat{c}_h \mathbf{v} \cdot \vec{n}_T - \int_T c_h \nabla \circ \mathbf{v} &= 0, \\ \int_T w \frac{\partial c_h}{\partial t} + \oint_{\partial T} w \widehat{\widehat{\mathbf{q}}}_h \cdot \vec{n}_T - \int_T \mathbf{q}_h \cdot \nabla w + \oint_{\partial T} w s(c_h) \cdot \vec{n}_T - \int_T w r(c_h) &= 0, \end{aligned}$$

donde *los flujos numéricos* \widehat{c}_h y $\widehat{\widehat{\mathbf{q}}}_h$ son aproximaciones de c y \mathbf{q} sobre ∂T , respectivamente; y $s(\cdot)$ es un término de estabilización, el cual garantiza la solución discreta para el problema estacionario [9]. Sea e una arista compartida por las celdas T_1 y T_2 ; y sean c_h^i y \mathbf{q}_h^i las aproximaciones de c y \mathbf{q} sobre la celda T_i . Sobre la arista e , los flujos numéricos son definidos como la combinación convexa de los valores de c_h y \mathbf{q}_h en las celdas T_1 y T_2 :

$$\begin{cases} \widehat{c}_h &= (1 - \beta_e) c_h^1 + \beta_e c_h^2 \\ \widehat{\widehat{\mathbf{q}}}_h &= \beta_e \mathbf{q}_h^1 + (1 - \beta_e) \mathbf{q}_h^2 \end{cases} \quad (2.2)$$

donde β_e es un parámetro que depende de la arista. Sobre la frontera, simplemente se define $\widehat{c}_h = c_h$ y $\widehat{\widehat{\mathbf{q}}}_h \cdot \vec{n}_T = \mathbf{q}_h \cdot \vec{n}_T = 0$. En la celda T_1 el término de estabilización

se define como

$$s(c_h) = \frac{\eta_e}{h_e} (c_h^1 - c_h^2) \vec{n}_T$$

donde η_e es un parámetro positivo que depende de la arista. El análisis teórico del parámetro de estabilización η_e se discute a fondo en [9, 28] mientras que las implicaciones computacionales de la elección del parámetro β son discutidas en [8, 15]. El problema discreto se reduce a un sistema no lineal de ecuaciones diferenciales ordinarias de la forma

$$\begin{aligned} D\mathbf{Q}_h + B_1 C_h &= 0 \\ M \frac{dC_h}{dt} + B_2 \mathbf{Q}_h + S_\eta C_h - r_h(C_h) &= 0 \end{aligned}$$

Usando la definición particular de los flujos numéricos, (ecuación 2.2), un simple cálculo revela que $B_2 = -B_1^T$, véase [9] para una descripción detallada de la forma bilineal. Debido a que la matriz D es diagonal por bloques, el almacenamiento físico de la variable auxiliar \mathbf{Q}_h no se requiere en la implementación, contrario a lo mencionado en [37]. Finalmente, escribiendo $B_1 = B_\beta$ se obtiene el siguiente sistema reducido que involucra solamente la variable C_h :

$$M \frac{dC_h}{dt} - \Delta_h^{LDG} C_h - r_h(C_h) = 0 \quad (2.3)$$

donde

$$-\Delta_h^{LDG} = S_\eta + B_\beta^T D^{-1} B_\beta. \quad (2.4)$$

Recientemente, en [11], se presentó un procedimiento de ensamblado rápido para $-\Delta_h^{LDG}$ que no requiere el almacenamiento de las matrices globales B_β , D^{-1} y S_η . Otros detalles sobre la implementación del método LDG han sido ampliamente discutidos en varios artículos. En [8], una descripción detallada de las estructuras de datos eficientes para la adaptividad en h usando mallas no conformes con un número arbitrario, pero fijo, de nodos colgantes fue presentado. En [7] se propusieron algunas

heurísticas para reducir la conectividad entre las celdas.

2.2. Método LDG de mínima disipación

En problemas transitorios el término de estabilización del método LDG, al igual que el de otros métodos discontinuos, produce extra disipación no deseada que puede deteriorar la precisión del método. Para evitar este problema, o al menos minimizarlo, Cockburn y Dong [14] propusieron un nuevo método, *Minimal Dissipation* LDG o MD-LDG el cual se deriva de una simple modificación del método LDG original.

Los valores β en cada arista definen la conectividad entre celdas para la matriz B_β . En el MDLDG los valores de β se escogen de acuerdo a un vector dirección \vec{d} , la estabilización se impone exclusivamente en un subconjunto de las aristas de la frontera, Γ_+ , véase la Figura 2-1.

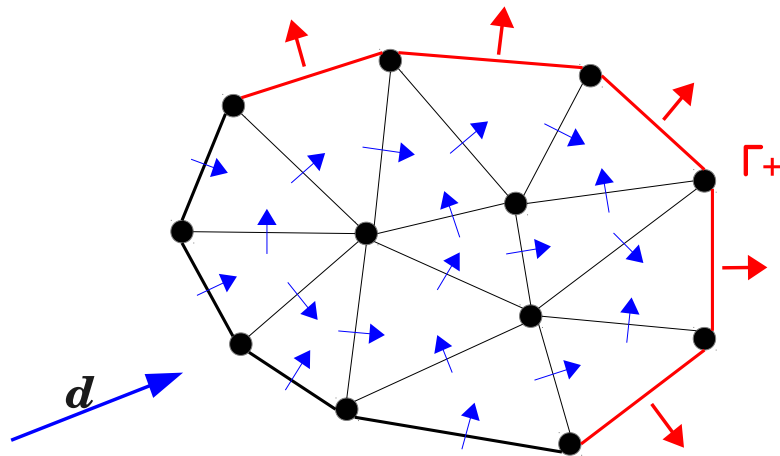


Figura 2-1: Elección de β en el MDLDG. El parámetro de estabilidad es nulo en todas las aristas excepto en Γ_+ .

Para un problema con condiciones de borde tipo Neumann, la estabilización puede ser eliminada completamente. En este caso particular la discretización del

operador Laplaciano se reduce a

$$-\Delta_h^{mdLDG} = B_\beta^T D^{-1} B_\beta. \quad (2.5)$$

Para evaluar la precisión de los métodos se considera el siguiente problema estacionario de reacción-difusión en el cuadrado unitario

$$-\Delta u + u = f,$$

sujeto a condiciones de flujo cero en el borde. Se asume como solución exacta

$$u(x, y) = 5000e^{-10[(x-1/2)^2+(y-1/2)^2]} [(x^2 - x)(y^2 - y)]^2.$$

Los cálculos se realizarán sobre una secuencia de mallas estructuradas, (m_1, m_2, \dots) . La primera malla posee dos triángulos rectángulos. El resto de las mallas se obtiene por refinamientos globales. Se consideran algunas elecciones particulares de β . En el LDG-0, se establece $\beta_e = 0.5$ para todas las aristas interiores. Esta configuración corresponde a la conectividad completa entre las celdas, la cual incluye los vecinos inmediatos de la celdas y los vecinos inmediatos de éstos. En el LDG-1 y LDG-2, β es calculado como sigue: sea e una arista compartida por las celdas T_1 y T_2 . Dado un vector dirección \vec{d} , se define $\beta_e = 0$ ($\hat{c}_h = c_h^1$) si $\vec{n}_T \cdot \vec{d} > 0$ y $\beta_e = 1$ ($\hat{c}_h = c_h^2$) de otra forma. En el LDG-1 y MDLDG-1, $\vec{d} = (-1, 1)^T$ satisface una conectividad reducida en algunas mallas particulares; y finalmente, en LDG-2 y MDLDG-2 se establece $\vec{d} = (2, 1)^T$ el cual produce una conectividad intermedia.

En las Tablas 2-1, 2-2 y 2-3 se compara el error L^2 para aproximaciones con polinomios de grado $p = 1, 2, 3$ sobre las mallas m_3, \dots, m_7 . Para todos los métodos LDG se selecciona $\eta = 1$. En todos los casos se observa convergencia de orden $p+1$ lo cual concuerda con la teoría. No se observa una diferencia sustancial en la precisión.

LDG-0	LDG-1	LDG-2	MDLDG-1	MDLDG-2
2.72e-01	2.62e-01	2.53e-01	2.41e-01	2.39e-01
7.60e-02	6.79e-02	6.57e-02	6.18e-02	6.16e-02
1.97e-02	1.71e-02	1.66e-02	1.55e-02	1.55e-02
4.96e-03	4.29e-03	4.16e-03	3.89e-03	3.89e-03
1.24e-03	1.07e-03	1.04e-03	9.72e-04	9.73e-04

Tabla 2-1: Error $\|u - u_h\|_0$ para $p = 1$.

LDG-0	LDG-1	LDG-2	MDLDG-1	MDLDG-2
2.10e-02	2.38e-02	2.46e-02	2.37e-02	2.59e-02
2.46e-03	3.03e-03	3.14e-03	3.02e-03	3.32e-03
3.00e-04	3.80e-04	3.94e-04	3.79e-04	4.18e-04
3.73e-05	4.75e-05	4.94e-05	4.75e-05	5.24e-05
4.65e-06	5.94e-06	6.17e-06	5.93e-06	6.55e-06

Tabla 2-2: Error $\|u - u_h\|_0$ para $p = 2$.

LDG-0	LDG-1	LDG-2	MDLDG-1	MDLDG-2
2.38e-03	2.50e-03	2.61e-03	2.49e-03	2.73e-03
1.57e-04	1.57e-04	1.66e-04	1.57e-04	1.75e-04
9.88e-06	9.84e-06	1.04e-05	9.85e-06	1.10e-05
6.19e-07	6.15e-07	6.54e-07	6.15e-07	6.88e-07
3.87e-08	3.84e-08	4.09e-08	3.84e-08	4.30e-08

Tabla 2-3: Error $\|u - u_h\|_0$ para $p = 3$.

Debido a que el propósito es evaluar el desempeño de los métodos para la ecuación modelo, es importante comparar el condicionamiento espectral, $\kappa(\eta)$, de matrices de la forma $M - \alpha\Delta_h$ con condiciones de borde homogéneas tipo Neumann.

En la figura 2-2 se muestra el comportamiento $\kappa(\eta)$ de la matriz $M - \Delta_h$ para una malla estructurada con 2048 celdas y varios grados de polinomio. El comportamiento asintótico de el condicionamiento espectral es el mismo para todas las elecciones de β en el método LDG. Sin embargo, se observa que se obtiene una discretización más estable con el LDG-0. Para una conectividad reducida el condicionamiento de los métodos MDLDG y LDG es bastante similar cuando $\eta \rightarrow 0$.

2.3. Reacción para p especies

El operador (potencialmente no lineal) de reacción , $\mathbf{r} : [0, T] \times \Omega \longrightarrow \mathbb{R}^p$, acopla las p ecuaciones del sistema de reacción-difusión. Este operador no involucra derivadas espaciales, por consiguiente, el proceso de discretización en espacio por un método discontinuo puede realizarse localmente en cada celda de la malla.

De la formulación del problema de elemento finito discontinuo en la celda T , se deduce que el operador discreto de reacción se obtiene por la expresión

$$\mathbf{r}_h(t, \mathbf{C}_h^T) = \left[\int_T \mathbf{r}(t, \mathbf{C}_h^T) \cdot \boldsymbol{\varphi}_n \right]_n ,$$

siendo $\{\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_s\}$ una base para $[\mathbb{P}_{pT}(T)^p]$, con $s = \dim [\mathbb{P}_{pT}(T)^p]$. La variable \mathbf{C}_h^T representa los grados de libertad de \mathbf{C}_h que solamente están asociados a la celda T .

En la solución de un sistema no lineal de ecuaciones puede ser necesario el cómputo del Jacobiano. Este puede obtenerse, para la celda T , por la expresión

$$\partial_{\mathbf{c}} \mathbf{r}_h(t, \mathbf{C}_h^T) = \left[\int_T (\partial_{\mathbf{c}} \mathbf{r}(t, \mathbf{C}_h^T) \boldsymbol{\varphi}_m) \cdot \boldsymbol{\varphi}_n \right]_{n,m} ,$$

donde $\partial_{\mathbf{c}} \mathbf{r}(t, \mathbf{c})$ es el Jacobiano de \mathbf{r} respecto a la variable \mathbf{c} .

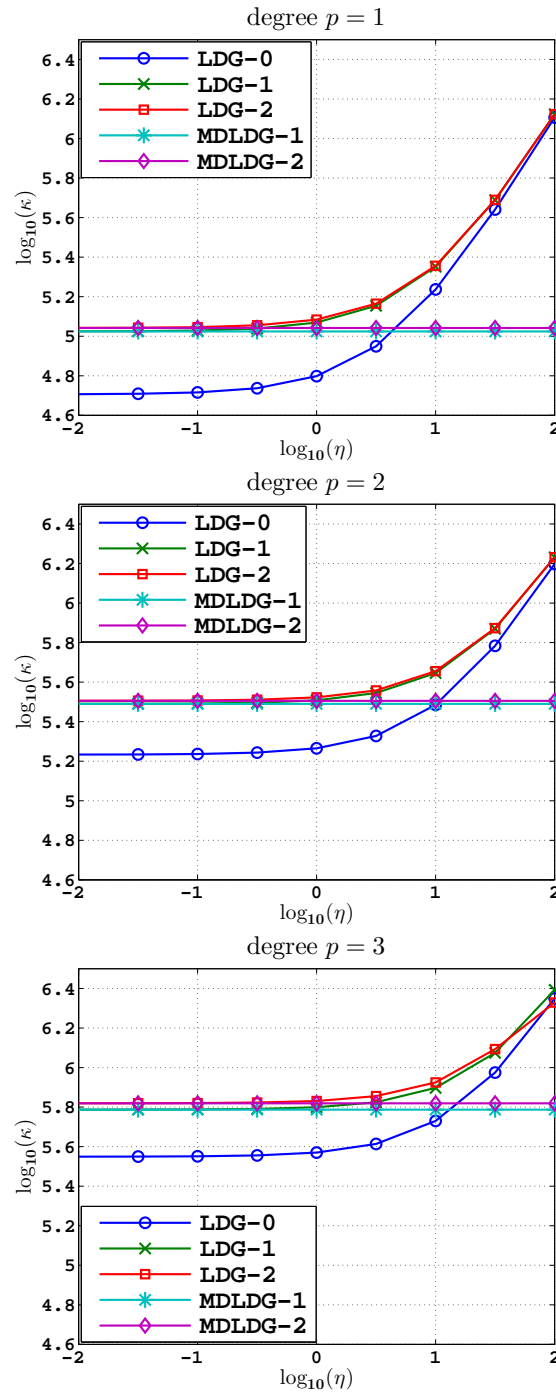


Figura 2-2: Número de condicionamiento espectral $\kappa(\eta)$ para $M - \Delta$. De izquierda a derecha con polinomios de grado $p = 1, 2, 3$.

Capítulo 3

INTEGRACIÓN EN TIEMPO

La integración en tiempo de sistemas de reacción-difusión se puede realizar de diferentes maneras. Dependiendo de la naturaleza física del problema ciertas clases de métodos pueden ser más apropiados que otros. A continuación se mencionan algunas de las clases de métodos comúnmente utilizados:

1. **métodos explícitos.** Al menos en principio, estos métodos no involucran la solución de sistemas algebraicos, por lo tanto requieren poco costo computacional. Sin embargo, no se recomiendan en problemas rígidos debido a que sufren una severa restricción en el tamaño de paso para alcanzar estabilidad.
2. **métodos implícitos.** Estos métodos son utilizados en problemas rígidos ya que pueden ser estables con tamaños de paso grandes. Por otro lado, involucran la solución de un problema algebraico lo cual puede ser muy costoso en problemas no lineales. En sistemas de reacción-difusión, los Jacobianos de dichos sistemas pueden ser no simétricos. Esto hecho supone una desventaja si el problema no lineal es de gran tamaño.
3. **métodos descomposición de operador.** Estos métodos consisten en el computo de una serie de subpasos para avanzar de t_n a t_{n+1} . Cada subpaso está asociado con un término del problema original. Estos métodos son ideales en problemas que poseen términos de diferente naturaleza (como los términos de reacción y difusión) ya que permiten el uso de la técnica numérica más apropiada para realizar cada subpaso. En sistemas de reacción-difusión, el uso de un método discontinuo puede ser benéfico ya que desacopla la

reacción espacialmente. Es importante tener en cuenta que cada subpaso no necesariamente corresponde a un paso del problema físico. Este hecho puede ser perjudicial en problemas donde las condiciones de borde son importantes.

4. **métodos implícito-explicito (IMEX).** Estos métodos son una combinación de métodos implícitos y explícitos. Son ideales en problemas de reacción-difusión, donde la difusión es lineal y rígida, y la reacción es no-lineal y relativamente menos rígida.

En este trabajo sólo se consideran los métodos de descomposición de operador y los métodos IMEX ya que son ideales para la integración en tiempo de sistemas de reacción-difusión.

3.1. Métodos de descomposición de operador

En los métodos de descomposición de operador, la solución de un problema complejo es aproximada por la solución de una secuencia de subproblemas más simples. Para ilustrar la idea, considere un sistema de ecuaciones diferenciales ordinarias de la forma

$$u'(t) = f_1(t, u(t)) + f_2(t, u(t)), \quad t > 0,$$

donde $u(t)$ es una función de \mathbb{R} en \mathbb{R}^m . Si $t_*, \tau_* \in \mathbb{R}^+$ y $w_* \in \mathbb{R}^m$ son datos arbitrarios, entonces para cada suboperador, f_i , se define un operador solución $S_{\{\tau_*, t_*\}}^i(w_*) = v_i(t_* + \tau_*)$, donde la función v_i es la solución del problema

$$\begin{cases} v_i'(t) = f_i(t, v_i(t)), & t_* < t < t_* + \tau_*, \\ v_i(t_*) = w_*, \end{cases} \quad (3.1)$$

El método de descomposición más simple se obtiene por la siguiente fórmula

$$u_{n+1} = (S_{\{\tau, t_n\}}^2 \circ S_{\{\tau, t_n\}}^1) (u_n), \quad (3.2)$$

donde $u_n \approx u(t_n)$ y $\tau = t_{n+1} - t_n > 0$. En general, este método introduce un error de descomposición. Si la solución es suficientemente suave, el método alcanza precisión de primer orden. Sin embargo, bajo ciertas condiciones sobre los suboperadores, f_i , el error de descomposición puede anularse, por lo que $u_n = u(t_n)$.

Un método de descomposición de segundo orden en tiempo se obtiene aplicando dos veces la fórmula (3.2), primero de manera directa en el intervalo $(t_n, t_n + \tau/2)$ y luego en orden inverso en el intervalo $(t_n + \tau/2, t_n + \tau)$,

$$\begin{aligned} u_{n+1} &= (S_{\{\tau/2, t_n\}}^1 \circ S_{\{\tau/2, t_n\}}^2) \circ (S_{\{\tau/2, t_n + \tau/2\}}^2 \circ S_{\{\tau/2, t_n + \tau/2\}}^1) (u_n) \\ &= (S_{\{\tau/2, t_n\}}^1 \circ S_{\{\tau, t_n\}}^2 \circ S_{\{\tau/2, t_n + \tau/2\}}^1) (u_n) \end{aligned} \quad (3.3)$$

donde $u_n \approx u(t_n)$ y $\tau = t_{n+1} - t_n > 0$. Similarmente, existe un error de descomposición el cual puede anularse bajo ciertas condiciones en los suboperadores, f_i . Este método es mejor conocido como *descomposición simétrica de Strang* [33].

La precisión y estabilidad final de los métodos de descomposición de operador está sujeta a la precisión y estabilidad de los métodos utilizados en cada subproblema. Métodos de Runge-Kutta de alto orden suficientemente estables y precisos respecto al método de descomposición pueden ser usados con un tamaño de paso de tiempo variable, sin embargo, el costo computacional puede ser elevado. Aquí se consideran algunos métodos basados en la descomposición simétrica de Strang donde los pasos intermedios se resuelven por un sólo paso de los métodos de Euler y del trapecio, [18, 19, 37].

Para los sistemas de reacción-difusión considerados en este trabajo la descomposición simétrica de Strang tiene la forma

$$u_{n+1} = \left(\mathcal{D}_{\tau/2} \circ \mathcal{R}_{\tau} \circ \mathcal{D}_{\tau/2} \right) (u_n),$$

donde $\mathcal{D}_{\tau^*}(\cdot)$ y $\mathcal{R}_{\tau^*}(\cdot)$ denotan la evolución en tiempo, con tamaño de paso τ^* , de la difusión y reacción, respectivamente.

La descomposición simétrica de Strang con Cranck-Nicolson consiste en resolver cada subproblema por el método del trapecio implícito (TI):

$$u_{n+1} = \left(\overset{TI}{\mathcal{D}_{\tau/2}} \circ \overset{TI}{\mathcal{R}_{\tau}} \circ \overset{TI}{\mathcal{D}_{\tau/2}} \right) (u_n) \quad (3.4)$$

Cuando la reacción no es rígida el paso de reacción puede realizarse por el método del trapecio explícito (TE) lo cual ayuda a reducir el costo computacional,

$$u_{n+1} = \left(\overset{TI}{\mathcal{D}_{\tau/2}} \circ \overset{TE}{\mathcal{R}_{\tau}} \circ \overset{TI}{\mathcal{D}_{\tau/2}} \right) (u_n) \quad (3.5)$$

En ambos casos se hará referencia a este método como (**SS-OS**) especificando si la reacción es implícita o explícita.

Por otro lado, la descomposición simétrica de Strang Trapezoidal (**TR-OS**) consiste en una combinación del método de Euler explícito (EE) e implícito (EI):

$$\begin{aligned} u_{n+1} &= \left(\overset{EE}{\mathcal{D}_{\tau/2}} \circ \overset{EE}{\mathcal{R}_{\tau/2}} \circ \overset{EI}{\mathcal{R}_{\tau/2}} \circ \overset{EI}{\mathcal{D}_{\tau/2}} \right) (u_n) \\ &= \left(\overset{EE}{\mathcal{D}_{\tau/2}} \circ \overset{TI}{\mathcal{R}_{\tau}} \circ \overset{EI}{\mathcal{D}_{\tau/2}} \right) (u_n) \end{aligned} \quad (3.6)$$

Observe que la expresión $\overset{EE}{\mathcal{R}_{\tau/2}} \circ \overset{EI}{\mathcal{R}_{\tau/2}}$ equivale a $\overset{TI}{\mathcal{R}_{\tau}}$. Nuevamente, cuando la reacción no es rígida se puede utilizar el método del trapecio explícito para reducir el costo

computacional:

$$u_{n+1} = \left(\begin{smallmatrix} EE & TE & EI \\ \mathcal{D}_{\tau/2} & \mathcal{R}_\tau & \mathcal{D}_{\tau/2} \end{smallmatrix} \right) (u_n) \quad (3.7)$$

En ambos casos se hará referencia a este método como **(TR-OS)** especificando si la reacción es implícita o explícita.

Todos estos métodos tienen segundo orden de convergencia. Una simple inspección revela que los métodos TR-OS requieren menos costo computacional que los métodos SS-OS. El método de mayor costo es el SS-OS con reacción implícita ya que requiere el uso de un método implícito en cada subproblema. El método TR-OS con reacción explícita es el de menor costo ya que solamente resuelve un subproblema de difusión por un método implícito.

Los métodos multipasos no son recomendados para la solución de los pasos intermedios ya que requieren información anterior de cada subproblema la cual no está naturalmente disponible en los métodos de descomposición de operador.

3.2. Métodos IMEX

Los métodos IMEX son una combinación de métodos implícitos y explícitos. Para ilustrar esta idea, considere el siguiente sistema de ecuaciones diferenciales ordinarias,

$$u'(t) = g(t, u(t)) + f(t, u(t)), \quad t > 0, \quad (3.8)$$

donde $u(t)$ es una función de \mathbb{R} en \mathbb{R}^m . Suponga que f es un operador lineal y rígido, ideal para el uso de un método implícito; mientras g es un operador no lineal y no rígido, ideal para un método explícito.

Integrando (3.8) desde t_n a t_{n+1} , con $\tau = t_{n+1} - t_n > 0$, respecto a la variable t , se obtiene

$$u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} g(t, u(t)) dt + \int_{t_n}^{t_{n+1}} f(t, u(t)) dt.$$

El método de Euler IMEX se obtiene aproximando la primera integral por la fórmula de cuadratura $\tau g(t_{n+1}, u(t_{n+1}))$, mientras que la segunda integral por la fórmula $\tau f(t_n, u(t_n))$. Esto conduce al esquema

$$u_{n+1} = u_n + \tau g(t_{n+1}, u_{n+1}) + \tau f(t_n, u_n).$$

donde $u_n \approx u(t_n)$. Luego, para calcular $u_{n+1} \approx u(t_{n+1})$ es necesario resolver un sistema lineal (ya que g es lineal). Este método tiene precisión de primer orden.

Un método IMEX Runge-Kutta de segundo orden, descrito en [19], se obtiene a partir de la siguiente fórmula

$$\begin{aligned} u^* &= u_n + \frac{\tau}{2} (g(t_n, u_n) + g(t_{n+1}, u^*)) + \tau f(t_n, u_n) \\ u_{n+1} &= u_n + \frac{\tau}{2} (g(t_n, u_n) + g(t_{n+1}, u^*)) + \frac{\tau}{2} (f(t_n, u_n) + f(t_{n+1}, u^*)) \end{aligned}$$

Este esquema será denotado como **IMEX-RK**.

Un método IMEX multipasos de segundo orden, introducido en [3], se obtiene a partir de la siguiente fórmula

$$3u_{n+1} - 4u_n + u_{n-1} = 2\tau (g(t_{n+1}, u_{n+1}) + 2f(t_n, u_n) - f(t_{n-1}, u_{n-1}))$$

Este esquema será denotado como **IMEX-BDF**.

Capítulo 4

ASPECTOS COMPUTACIONALES

En este capítulo se discuten los principales aspectos computacionales del marco para la solución de problemas transitorios de reacción-difusión en 2D con un número arbitrario de especies químicas. El código es orientado a objetos, de manera que cada rutina puede ser extendida y modificada fácilmente. El código fue escrito en el lenguaje C++.

Actualmente, el código funciona sobre mallas conformes con celdas triangulares. El grado de polinomio es arbitrario, pero se elige de manera uniforme para todas las celdas. La base para el espacio de aproximación puede ser escogida entre la base Lagrangiana y la base canónica.

La discretización en tiempo se puede realizar por dos clases de métodos: (i) los métodos de descomposición de operador y (ii) los métodos IMEX. Hasta el momento sólo ciertos métodos populares de orden dos han sido considerados en ambas clases de métodos.

El marco computacional fue desarrollado con el propósito de estudiar los sistemas de reacción-difusión que presentan estructuras de Turing. El número de especies puede ser arbitrario, las condiciones de borde pueden ser de tipo Dirichlet o Neumann, y el operador de reacción puede ser no lineal.

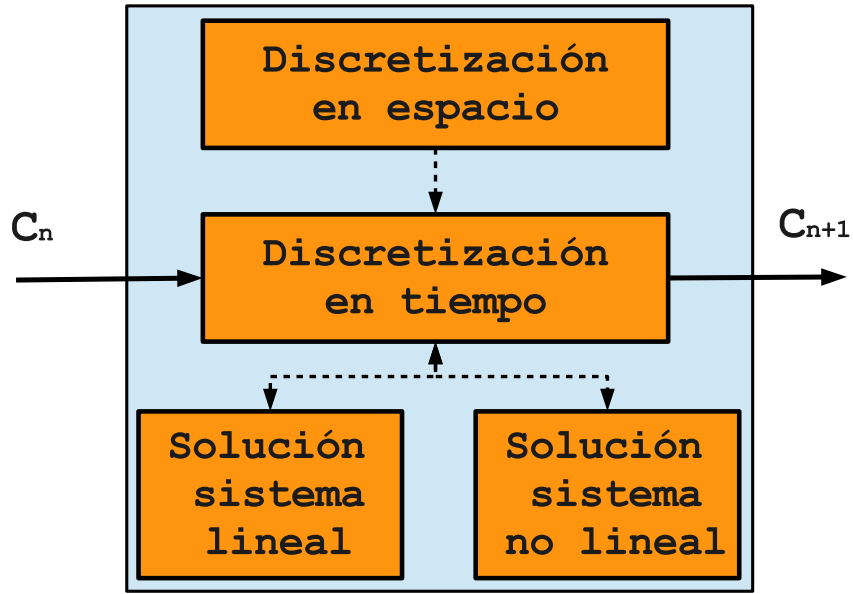


Figura 4-1: Esquema simplificado del marco computacional.

Como el código es orientado a objetos, muchos de los componentes pueden ser extendidos y modificados fácilmente. Por ejemplo, otros tipos de bases y otros métodos de integración en tiempo pueden ser incorporados al código en el futuro.

Para una descripción simple del marco computacional, los componentes básicos del código pueden clasificarse de acuerdo a la función que cumplen, véase la Figura 4-1.

4.1. Rutinas de discretización en espacio

Las rutinas de discretización en espacio son un conjunto de clases cuya finalidad es proveer la funcionalidad básica de los operadores discretos de reacción y difusión, entre otros. Dentro de las principales clases se pueden mencionar:

- **Diffusion2D**: Contiene toda la información relacionada al operador escalar de difusión como las condiciones de borde y el tensor de difusión.

- **Reaction:** Esta clase contiene el conjunto de funciones (componentes) que conforman el operador de reacción. También contiene las funciones que conforman el jacobiano del operador de reacción.
- **LDG2D:** Dado un objeto de tipo *Diffusion2D*, esta clase se encarga de la discretización espacial del operador escalar de difusión por los métodos LDG y MDLDG. Esta clase se encarga del ensamblado de la matriz Δ_h , del vector fuente y de las condiciones de borde. También permite ensamblar, invertir y evaluar la matriz de masa, M . La interfaz de esta clase se ha resumido en la Tabla 4-1.
- **DGReaction:** Este objeto es la discretización espacial del operador de reacción descrito por un objeto tipo *Reaction*. Permite evaluar los operadores \mathbf{r}_h y $\partial_{\mathbf{c}}\mathbf{r}_h$ de manera local, en la celda de referencia, y de manera global, en toda la malla. La interfaz de esta clase se ha resumido en la Tabla 4-2.

4.2. Rutinas de discretización en tiempo

La finalidad de las rutinas de discretización en tiempo es proveer la funcionalidad necesaria para la integración en tiempo del sistema semi-discreto.

Un método de integración en tiempo puede interpretarse como un operador S de la forma

$$u_{n+1} = S(t_n, u_n; \tau),$$

Clase *LDG2D*

```

class LDG2D
{
    public:
        LDG2D();
        ~LDG2D();
        void setup(const Basis2D & basis);
        BlockCSR * getFastSchurComplementAssembly(
            const Diffusion2D & diff ,
            const Mesh2D & mesh
        ) const;
        void getSource(
            double fh[],
            const Diffusion2D & diff,
            const Mesh2D & mesh, double t,
            double alpha=1.0
        ) const;
        void getBoundaryConditions(
            double gh[],
            const Diffusion2D & diff,
            const Mesh2D & mesh,
            double t,
            double alpha=1.0
        ) const;
        :
    private:
        :
};

```

Tabla 4–1: Interfaz de la clase *LDG2D*.

Clase *DGReaction*

```

class DGReaction
{
    public:
        DGReaction();
        ~DGReaction();
        void setup(const Basis2D & basisU);
        void evalGlobally(
            double Y[],
            const double X[],
            const Reaction & reac,
            const Mesh2D & mesh,
            double t,
            double alpha=1.0
        ) const;
        MATRIX * evalJacobianGlobally(
            const double X[],
            const Reaction & reac,
            const DGFES2D & dgSpace,
            double t,
            double alpha
        ) const;
        void evalLocally(
            double Y[],
            const double X[],
            const Reaction & reac,
            double t,
            double alpha=1.0
        ) const;
        void evalJacobianLocally(
            double JacMat[],
            const double X[], const Reaction & reac,
            double t,
            double alpha=1.0
        ) const;
        :
    private:
        :
};

```

Tabla 4–2: Interfaz de la clase *DGReaction*.

siendo $\tau = t_{n+1} - t_n > 0$, el tamaño de paso en tiempo. Note que para un método multipasos el operador S dependerá de cierta información del pasado, pero por simplicidad de la notación no se escribe.

En esta implementación la funcionalidad de un operador es incorporada mediante una clase abstracta, llamada **Operator**, véase la Tabla 4-3. Luego, cada método de integración en tiempo se deriva de la clase *Operator*, véase la Figura 4-2. Por ejemplo, en la Tabla 4-4 se ilustra la interfaz de la clase *SSOSI* para el método SS-OS con reacción implícita.

Otros operadores que se utilizan frecuentemente también son implementados como clases que se derivan de *Operator*. Por ejemplo, el operador $(D \otimes \Delta_h)$ es un objeto de la clase *MultiCompDiff* la cual se deriva de *Operator*. Otros operadores de uso frecuente son $(I \otimes M)$ y $(M - \alpha\tau\Delta_h)$, véase la Figura 4-2. Observe que estos operadores utilizan la información de las matrices M , M^{-1} y Δ_h para simular operadores multicomponentes. Así, el marco computacional sólo requiere almacenar tales matrices y el resto de los operadores son simulados. Esto reduce considerablemente el almacenamiento en memoria ya que los operadores multicomponentes no son ensamblados.

La matriz Δ_h tiene una conveniente estructura por bloques que facilita su ensamblado y almacenamiento [11]. Debido al gran número de bloques nulos en Δ_h , el uso de un esquema de compresión para matrices huecas por bloques resulta idóneo para reducir el almacenamiento en memoria.

Aquí se ha implementado la clase **BlockCSR**, véase Tabla 4-5, para el formato de almacenamiento CSR [30] por bloques, donde en principio sólo se almacena los

Clase Operator

```
class Operator
{
    public:
        Operator();
        virtual ~Operator();
        void eval(
            double Y[],
            const double X[],
            double t,
            double alpha=1.0
        ) const = 0;
        void setDimensions(
            uint dimx,
            uint dimy
        );
        virtual uint getDimensionX() const;
        virtual uint getDimensionY() const;
    private:
        :
};
```

Tabla 4-3: Interfaz de la clase abstracta *Operator*.

 Clase *SSOSI*

```

class SSOSI : public Operator
{
  public:
    SSOSI();
    virtual ~SSOSI();
    void setup(
      const MultiCompMass & M,
      const MultiCompDiff & D,
      const MultiCompReac & R
    );
    void setTimeStep(double dt);
    void eval(
      double Y[],
      const double X[],
      double t,
      double alpha=1.0
    ) const;
    void setDimensions(
      uint dimx,
      uint dimy
    );
    virtual uint getDimensionX() const;
    virtual uint getDimensionY() const;
    :
  private:
    :
};

```

 Tabla 4–4: Interfaz de la clase *SSOSI*.

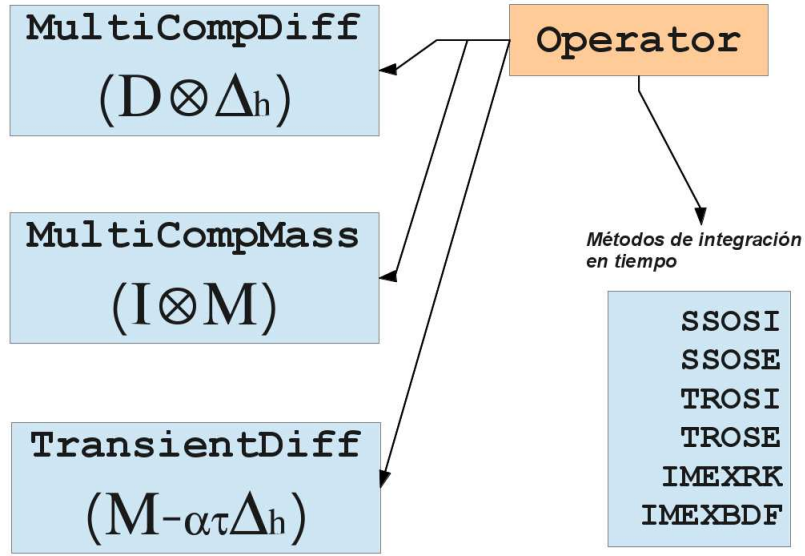


Figura 4-2: Clases que heredan de *Operator*

bloques no nulos de cada fila. Este formato resulta conveniente ya que facilita el ensamblado de la matriz Δ_h , fila por fila. La clase tiene la información necesaria para realizar de manera óptima la multiplicación matriz-vector, ya que sólo utiliza la información de los bloques no nulos.

4.3. Solución de sistemas lineales

Durante el proceso de integración en tiempo es necesario la solución de uno o más sistemas lineales multicomponentes. La matriz principal de estos sistemas tiene una estructura diagonal por bloques

$$(I \otimes M) - \alpha\tau(D \otimes \Delta_h) = \begin{bmatrix} (M - \alpha\tau d_1 \Delta_h) & & \\ & \ddots & \\ & & (M - \alpha\tau d_p \Delta_h) \end{bmatrix}$$

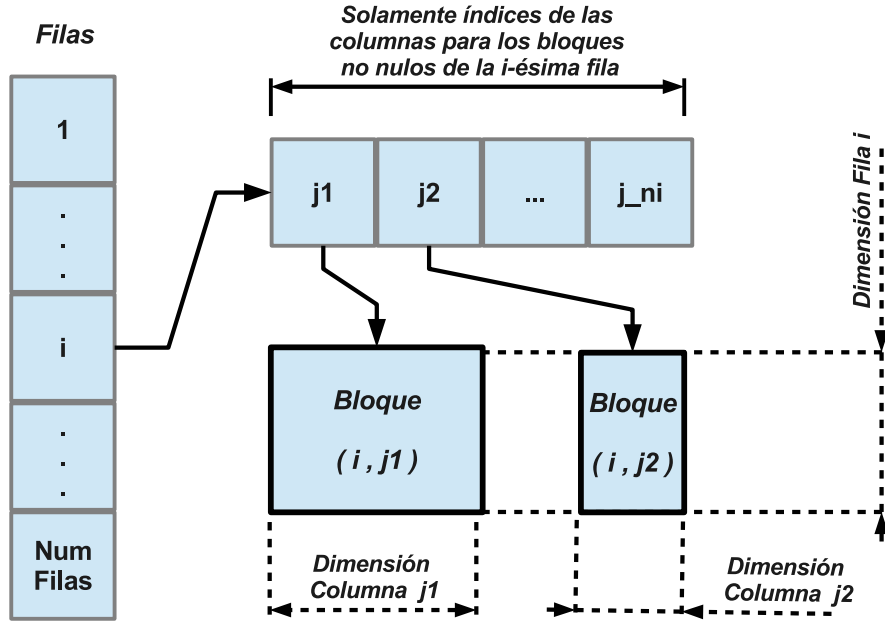


Figura 4-3: Diagrama simplificado del esquema CSR por bloques

donde α es una constante real positiva y τ es el tamaño de paso en tiempo. Luego, la solución de estos sistemas lineales consiste en resolver varios subsistemas independientes de la forma $(M - \beta\tau\Delta_h)$, siendo β una constante positiva. Esta estructura diagonal ayuda a reducir el consumo de memoria ya que solamente es necesario almacenar la matriz Δ_h . Como se mencionó anteriormente el operador $(M - \beta\tau\Delta_h)$ es un objeto de la clase *TransientDiff* que hereda de la clase *Operator*. Para cada subsistema se utiliza el método del gradiente conjugado el cual es un objeto de la clase **PCGM**, véase Tabla 4-6.

4.4. Solución de sistemas no lineales

Algunos de los métodos de descomposición de operador que fueron implementados requieren la solución de un sistema no lineal de ecuaciones el cual proviene de

Clase *BlockCSR*

```

class BlockCSR
{
    public:
        BlockCSR();
        ~BlockCSR();
        void eval(
            double Y[],
            const double X[],
            double alpha=1.0
        ) const;
        void insert(
            const uint i,
            const uint nz,
            JROW jVal,
            AROW aVal,
            uint mode = FULL_CHECK_MODE
        );
        :
    private:
        :
};

```

Tabla 4–5: Interfaz de la clase BlockCSR.

la integración en tiempo del problema de reacción. Como el operador de reacción fue discretizado por un método discontinuo, la solución de este sistema no lineal de gran tamaño puede reducirse a varios subsistemas no lineales, uno por celda, los cuales son de tamaño pequeño y para una celda T tienen la forma

$$\begin{aligned}
 \mathcal{F}_T(\mathbf{X}) &= (I_p \otimes M_T)\mathbf{X} - \alpha\tau\mathbf{r}_h(t, \mathbf{X}) - \mathbf{b} \\
 &= \mathbf{0},
 \end{aligned}$$

donde α es una constante positiva, τ es el tamaño de paso en tiempo y \mathbf{b} es un vector. La matriz M_T representa la matriz de local de masa respecto a la celda T . La definición de $\mathbf{r}_h(t, \mathbf{X})$ fue dada en la Sección 2.3. Como estos sistemas son

Clase *PCGM*

```

class PCGM
{
    public:
        PCGM();
        ~PCGM();
        void setup(
            uint maxI,
            double rTol,
            double aTol
        );
        uint solve(
            double X[],
            const double rhs[],
            const Operator & A,
            const Operator & Preconditioner,
            double t=0
        );
        :
    private:
        :
};

```

Tabla 4-6: Interfaz de la clase PCGM.

pequeños pueden ser resueltos eficientemente por el método de Newton:

$$\mathbf{X}_{n+1} = \mathbf{X}_n - J_n^{-1} \mathcal{F}_T(\mathbf{X}_n),$$

donde

$$J_n = (I_p \otimes M_T) - \alpha \tau \partial_{\mathbf{x}} \mathbf{r}_h(t, \mathbf{X}_n)$$

es el Jacobiano de $\mathcal{F}_T(\cdot)$ en \mathbf{X}_n . La definición de $\partial_{\mathbf{x}} \mathbf{r}_h(t, \mathbf{X})$ fue dada en la Sección 2.3. La interfaz de la clase Newton para resolver sistemas no lineales se muestra en la Tabla 4-7.

Clase Newton

```
class Newton
{
    public:
        Newton();
        ~Newton();
        void setup(uint maxI,double rTol,double aTol);
        uint solve(
            double Y[],
            const double X[],
            const Operator & F,
            const Operator & JacF,
            double t=0
        );
        :
    private:
        :
};
```

Tabla 4–7: Interfaz de la clase Newton.

Capítulo 5

EXPERIMENTOS NUMÉRICOS

En este capítulo se llevan a cabo una serie de experimentos con el propósito de comparar los diferentes métodos numéricos. Primero se considera un problema modelo lineal y escalar. Luego se considera el modelo de Schnakenberg [31] donde la reacción es no lineal. Este problema de dos especies químicas ha sido utilizado en diferentes trabajos para evaluar el desempeño de métodos numéricos [19, 29, 37]. Finalmente se considera un modelo no lineal que involucra tres especies químicas para la pigmentación en pieles de animales [22].

5.1. Problema escalar lineal

Como primer ejemplo para el estudio comparativo se considera el siguiente problema modelo en, $\Omega = (0, 1) \times (0, 1)$, y en el intervalo de tiempo $(0, T)$, $T > 0$,

$$\frac{\partial c}{\partial t} = \sigma \Delta c + \lambda c, \quad (5.1)$$

sujeto a condiciones de flujo cero en la frontera de Ω y a la condición inicial $c(x, y, 0) = \cos(\pi x) \cos(\pi y)$. La solución exacta para este problema es

$$c(x, y, t) = e^{(\lambda - 2\pi^2 \sigma)t} \cos(\pi x) \cos(\pi y)$$

Se desea estudiar la precisión y estabilidad de los esquemas numéricos. Para ello se construye una secuencia de mallas anidadas y estructuradas, (m_i) . La malla inicial, m_0 , consiste de dos triángulos rectángulos. El resto de las mallas se obtiene

por refinamientos uniformes de m_0 , véase la Figura 5-1. La malla m_i tiene $2 \times 4^{i-1}$ celdas con $h_{min}^i = \sqrt{2} \times 2^{-i-1}$, véase la Tabla 5-1.

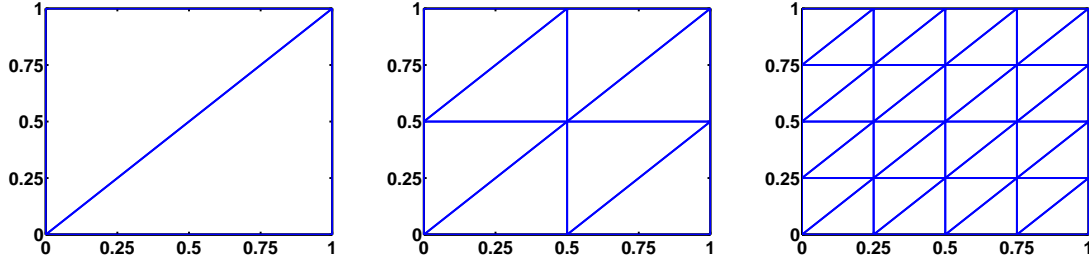


Figura 5-1: Malla inicial m_0 (izquierda) y sus dos refinamientos sucesivos m_1 (centro) y m_2 (derecha).

	h_{min}	h_{max}	# Celdas	# Nodos	# Aristas
m_1	1.41e-00	1.41e-00	2	4	5
m_2	7.07e-01	7.07e-01	8	9	16
m_3	3.54e-01	3.54e-01	32	25	56
m_4	1.77e-01	1.77e-01	128	81	208
m_5	8.84e-02	8.84e-02	512	289	800
m_6	4.42e-02	4.42e-02	2048	1089	3136
m_7	2.21e-02	2.21e-02	8192	4225	12416

Tabla 5-1: Descripción de mallas para el problema lineal escalar de reacción difusión.

Para la discretización en espacio se comparan los métodos LDG-0, LDG-1 y MDLDG-1. En el LDG-0, se establece $\beta_e = 0.5$ para todas las aristas interiores. Esta configuración corresponde a la conectividad completa entre las celdas, la cual incluye los vecinos inmediatos de la celdas y los vecinos inmediatos de estos. En el LDG-1 y MD-LDG1, β es calculado como sigue: sea e una arista compartida por las celdas T_1 y T_2 . Dado un vector dirección \vec{d} , se define $\beta_e = 0$ ($\hat{c}_h = c_h^1$) si $\vec{n}_T \cdot \vec{d} > 0$ y $\beta_e = 1$ ($\hat{c}_h = c_h^2$) de otra forma. En el LDG-1 y MDLDG-1, $\vec{d} = (-1, 1)^T$ satisface una conectividad reducida respecto a la sucesión de mallas (m_i).

Para la discretización en tiempo se consideran, y se comparan, los métodos de descomposición de operador (SS-OS y TR-OS) y los métodos IMEX (IMEX-RK y

IMEX-BDF).

5.1.1. Precisión

Los parámetros σ y λ definen la dinámica del sistema. En este experimento el interés es estudiar la precisión para el caso $\lambda < 2\pi^2\sigma$, con $\sigma > 0$, donde un simple análisis de la solución exacta revela que el sistema decae a cero. Luego, existen dos casos para la reacción: (i) la reacción contribuye a que el sistema decrezca, es decir $\lambda < 0$, y (ii) la reacción contribuye a que el sistema crezca, $\lambda > 0$.

Para estudiar el caso (i), se establecen los valores $\lambda = -1.0$ y $\sigma = 1/(2\pi^2)$. En el tiempo final $T = 1.0$ se calcula el error L^2 sobre las mallas m_4, \dots, m_7 y polinomios de grado $p = 1$, con $\tau \approx 11h_{min}$. Para reducir en lo posible que el error del método del gradiente conjugado afecte la precisión de los cálculos se utiliza una tolerancia relativa para el error de $1.0\text{e-}14$. Los resultados se muestran en la Tabla 5-2. Como era de esperarse, todos los métodos muestran segundo orden de convergencia. No se observa diferencia sustancial en la precisión entre las diferentes discretizaciones en espacio. Los métodos SS-OS y TR-OS muestran ser levemente más precisos que los métodos IMEX.

Para el caso (ii), se consideran los valores $\lambda = 0.9$ y $\sigma = 1/(2\pi^2)$. Nuevamente se calcula el error L^2 para el tiempo final $T = 1.0$ sobre las mallas m_4, \dots, m_7 y polinomios de grado $p = 1$, con $\tau \approx 11h_{min}$. Los resultados revelan convergencia de segundo orden, véase la Tabla 5-3. No se observa diferencia sustancial en la precisión entre las diferentes discretizaciones en espacio. Sin embargo, los métodos IMEX muestran ser considerablemente más precisos que los métodos de descomposición SS-OS y TR-OS. Este hecho se debe a que el paso de reacción en los métodos de

descomposición aleja considerablemente la solución del equilibrio.

discretización espacial	SS-OS		TR-OS		IMEX-BDF		IMEX-TR	
	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa
MDLDG-1	3.78e-03	-	2.62e-03	-	3.23e-02	-	1.23e-02	-
	7.60e-04	2.31	4.90e-04	2.42	7.64e-03	2.08	2.06e-03	2.58
	1.70e-04	2.16	1.04e-04	2.24	1.72e-03	2.15	4.24e-04	2.28
	4.03e-05	2.08	2.38e-05	2.13	3.93e-04	2.13	9.64e-05	2.14
LDG-1	3.76e-03	-	2.59e-03	-	3.23e-02	-	1.23e-02	-
	7.54e-04	2.32	4.84e-04	2.42	7.64e-03	2.08	2.06e-03	2.58
	1.69e-04	2.16	1.03e-04	2.24	1.72e-03	2.15	4.23e-04	2.28
	4.00e-05	2.08	2.34e-05	2.13	3.93e-04	2.13	9.61e-05	2.14
LDG-0	3.73e-03	-	2.57e-03	-	3.23e-02	-	1.23e-02	-
	7.48e-04	2.32	4.78e-04	2.43	7.63e-03	2.08	2.05e-03	2.58
	1.67e-04	2.16	1.01e-04	2.24	1.72e-03	2.15	4.21e-04	2.28
	3.96e-05	2.08	2.30e-05	2.13	3.92e-04	2.13	9.57e-05	2.14

Tabla 5-2: Error L^2 con polinomios de grado $p = 1$. $\lambda = -1.0$, $\sigma = 1/2\pi^2$, $T = 1.0$, $\tau \approx 11h_{min}$

5.1.2. Alto orden en espacio

Una simple consecuencia del análisis de error del método en términos de h sugiere que para obtener mayor precisión en espacio se debe incrementar la resolución espacial. Sin embargo, el número de grados de libertad puede ser extremadamente grande obtener la precisión deseada. Cuando la solución es de alta regularidad es más conveniente utilizar aproximaciones de alto orden sobre una malla gruesa. Esto se ilustra en el siguiente experimento.

Considere $\sigma = 1/(2\pi^2)$ y $\lambda = -1.0$. La discretización en espacio se obtiene por el método MD-LDG1. Se consideran diferentes combinaciones de mallas y grados de polinomio, de manera que el error L^2 en el tiempo final $T = 1.0$ sea aproximadamente el mismo. Para la discretización en tiempo se escoge un tamaño de paso $\tau = 2.5e-03$, el cual es suficientemente pequeño en comparación con el error en espacio. Para reducir en lo posible que el error del método del gradiente conjugado afecte

discretización espacial	SS-OS		TR-OS		IMEX-BDF		IMEX-TR	
	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa
MDLDG-1	1.23e-02	-	1.95e-02	-	7.65e-04	-	6.44e-04	-
	3.53e-03	1.80	5.30e-03	1.88	2.51e-04	1.61	1.72e-04	1.90
	9.47e-04	1.90	1.39e-03	1.93	7.25e-05	1.79	4.48e-05	1.94
	2.45e-04	1.95	3.56e-04	1.96	1.93e-05	1.91	1.15e-05	1.97
LDG-1	1.24e-02	-	1.96e-02	-	8.77e-04	-	7.04e-04	-
	3.57e-03	1.80	5.34e-03	1.88	2.81e-04	1.64	1.84e-04	1.94
	9.56e-04	1.90	1.40e-03	1.93	8.02e-05	1.81	4.71e-05	1.96
	2.48e-04	1.95	3.58e-04	1.97	2.12e-05	1.92	1.20e-05	1.98
LDG-0	1.26e-02	-	1.98e-02	-	1.03e-03	-	8.03e-04	-
	3.61e-03	1.80	5.38e-03	1.88	3.20e-04	1.69	2.06e-04	1.97
	9.67e-04	1.90	1.41e-03	1.93	9.02e-05	1.83	5.21e-05	1.98
	2.50e-04	1.95	3.61e-04	1.97	2.37e-05	1.93	1.31e-05	1.99

Tabla 5–3: Error L^2 con polinomios de grado $p = 1$. $\lambda = 0.9$, $\sigma = 1/2\pi^2$, $T = 1.0$, $\tau \approx 11h_{min}$

la precisión de los cálculos se utiliza una tolerancia relativa para el error de $1.0e-14$. Los resultados se presentan en la Tabla 5–4 para las combinaciones $(p = 1, m_7)$, $(p = 2, m_5)$, $(p = 3, m_3)$ y $(p = 4, m_2)$, donde los grados de libertad respectivos son 393,216, 98,304, 12,288, 1,280 y 480. En todos los casos se observa aproximadamente el mismo error. En la Tabla 5–5 se presentan los respectivos tiempos de ejecución. Es evidente que el uso de alto orden no sólo reduce el costo de almacenamiento sin sacrificar precisión, sino que además, reduce el costo computacional. Este hecho se debe, por una parte, a la cantidad de datos que se procesan. Por otro lado, el condicionamiento espectral de los sistemas lineales involucrados tiene un comportamiento asintótico de $\mathcal{O}(h^{-2})$ [6]. Tales sistemas se han resuelto por el método del gradiente conjugado pero no se ha hecho uso de un preconditionador, lo cual afecta el desempeño para mallas finas. La Tabla 5–5 también muestra que los métodos TR-OS, IMEX-BDF y IMEX-TR tienen un tiempos similares, sin embargo el método SS-OS puede requerir 1.5 veces más tiempo.

	grados de libertad	($p = 1$, m_8) 393,216	($p = 2$, m_5) 12,288	($p = 3$, m_3) 1,280	($p = 4$, m_2) 480
MD-LDG1	SS-OS	1.06e-06	8.57e-07	2.34e-06	2.68e-06
	TR-OS	1.06e-06	8.57e-07	2.34e-06	2.68e-06
	IMEX-BDF	1.03e-06	1.02e-06	2.40e-06	2.73e-06
	IMEX-RK	1.03e-06	8.66e-07	2.34e-06	2.68e-06
LDG1	SS-OS	1.15e-06	8.02e-07	2.24e-06	2.61e-06
	TR-OS	1.15e-06	8.01e-07	2.24e-06	2.61e-06
	IMEX-BDF	9.55e-07	9.72e-07	2.30e-06	2.67e-06
	IMEX-RK	1.09e-06	8.10e-07	2.24e-06	2.61e-06
LDG0	SS-OS	1.40e-06	6.06e-07	2.12e-06	2.48e-06
	TR-OS	1.42e-06	6.05e-07	2.12e-06	2.48e-06
	IMEX-BDF	1.07e-06	8.16e-07	2.19e-06	2.54e-06
	IMEX-RK	1.34e-06	6.17e-07	2.13e-06	2.48e-06

Tabla 5–4: Error L^2 , en $T = 1.0$, para discretizaciones espaciales de grados P1, P2, P3 y P4. En cada caso la malla fue escogida de tal manera que los errores sean cercanos. El resto de los parámetros son $\tau = 2.5\text{e-}03$, $\sigma = 1/(2\pi^2)$ y $\lambda = -1.0$.

	grados de libertad	($p = 1$, m_8) 393,216	($p = 2$, m_5) 12,288	($p = 3$, m_3) 1,280	($p = 4$, m_2) 480
MD-LDG1	SS-OS	9,559.21	114.46	8.81	4.64
	TR-OS	7,533.34	81.20	5.01	2.50
	IMEX-BDF	8,191.24	82.07	4.68	2.31
	IMEX-RK	7,833.75	83.25	5.14	2.54
LDG1	SS-OS	10,085.3	121.24	8.45	4.76
	TR-OS	7,906.39	84.93	4.64	2.50
	IMEX-BDF	8,617.79	84.22	4.50	2.51
	IMEX-RK	8,219.29	86.44	4.85	2.51
LDG0	SS-OS	10,093.60	143.57	8.71	4.08
	TR-OS	7,631.87	94.01	5.77	2.71
	IMEX-BDF	8,583.81	92.94	5.33	2.47
	IMEX-RK	8,063.98	96.00	5.79	2.74

Tabla 5–5: Tiempos de ejecución (en segundos) correspondientes a la Tabla 5–4.

Ahora se desea explorar la convergencia cuando se utiliza alto orden en espacio y orden dos en tiempo. Suponga que para un tiempo final, T , se tiene que $\|c - c_{\{h,\tau\}}\|_0 \leq k_1 h^{p+1} + k_2 \tau^2$. Luego, seleccionando $\tau \leq kh^{(p+1)/2}$ se espera obtener una tasa de convergencia de orden $p+1$. Para reducir en lo posible que el error del método del gradiente conjugado afecte la precisión de los cálculos se utiliza una tolerancia relativa para el error de 1.0e-14. Considere los valores de parámetros $\sigma = 1/(2\pi^2)$ y $\lambda = -1.0$. En la Tabla 5–6 se muestran los errores y las tasas de convergencia para las mallas m_3 hasta m_6 con polinomios de grado $p = 2$. Para cada malla, el tamaño de paso en tiempo se ha seleccionado como 1.25e-02, 5.00e-03, 1.5625e-03 y 6.25e-04 respectivamente, los cuales satisfacen $\tau \leq 0.2415 h^{3/2}$. Como se esperaba, se obtuvo una tasa de convergencia de orden 3 en cada caso. No hubo diferencia sustancial en la precisión.

El experimento anterior se repite nuevamente, pero con polinomios de grado $p = 3$ y con las mallas m_1 hasta m_4 . Para cada malla se escoge el tamaño de paso en tiempo como 1.25e-02, 3.125e-03, 7.8125e-04 y 1.953125e-04 respectivamente, los cuales satisfacen la condición $\tau \leq 0.025 h^2$. Los resultados se muestran en la Tabla 5–7. Como se esperaba, se obtuvo una tasa de convergencia de orden 4 en cada caso. No hubo diferencia sustancial en la precisión.

En cuanto al tiempo de ejecución, se puede observar mejores resultados para $p = 3$, pese a la severa restricción en el tamaño de paso en tiempo. Esto se debe a que el número de grados de libertad es mucho menor que para $p = 2$.

5.1.3. Estabilidad

Una análisis lineal de estabilidad en la norma L^2 demuestra que la matriz de amplificación de un métodos de descomposición de operador se reduce a la multiplicación de los factores de amplificación de los pasos intermedios. Sean $A_{\mathcal{R}}$ y $A_{\mathcal{D}}$

discretización espacial	SS-OS		TR-OS		IMEX-BDF		IMEX-TR	
	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa
MDLDG-1	5.46e-05	-	5.46e-05	-	5.57e-05	-	5.46e-05	-
	6.84e-06	2.75	6.84e-06	2.76	7.15e-06	2.72	6.82e-06	2.48
	8.56e-07	3.00	8.56e-07	3.00	8.81e-07	3.02	8.57e-07	3.00
	1.07e-07	3.00	1.07e-07	3.00	1.12e-07	3.00	1.07e-07	3.00
LDG-1	5.13e-05	-	5.14e-05	-	5.23e-05	-	5.13e-05	-
	6.41e-06	2.76	6.41e-06	2.76	6.72e-06	2.72	6.42e-06	2.75
	8.01e-07	3.00	8.01e-07	3.00	8.27e-07	3.02	8.02e-07	3.00
	1.00e-07	3.00	1.00e-07	3.00	1.06e-07	3.00	1.00e-07	3.00
LDG-0	4.03e-05	-	4.04e-05	-	4.13e-05	-	4.02e-05	-
	4.90e-06	2.80	4.90e-06	2.80	5.28e-06	2.73	4.90e-06	2.80
	6.05e-07	3.02	6.05e-07	3.02	6.38e-07	3.05	6.05e-07	3.02
	7.53e-08	3.01	7.52e-08	3.01	8.26e-08	2.95	7.53e-08	3.01

Tabla 5–6: Convergencia con polinomios de grado $p=2$ en espacio

las matrices de amplificación de los pasos de reacción y difusión respectivamente del método SS-OS con reacción explícita, véase la fórmula 3.5. Debido a que el paso de reacción se simplifica en un sistema de ecuaciones diferenciales ordinarias desacoplado, se obtiene

$$A_{\mathcal{D}} = \left(I - \frac{\tau}{4}M^{-1}\Delta_h\right)^{-1} \left(I + \frac{\tau}{4}M^{-1}\Delta_h\right) \quad \text{y} \quad A_{\mathcal{R}} = \left(1 + \lambda\tau + \frac{(\lambda\tau)^2}{2}\right) I$$

Así, la matriz de amplificación A_{SS-OS} satisface

$$A_{SS-OS} = \left(1 + \lambda\tau + \frac{(\lambda\tau)^2}{2}\right) \left(\left(I - \frac{\tau}{4}M^{-1}\Delta_h\right)^{-1} \left(I + \frac{\tau}{4}M^{-1}\Delta_h\right)\right)^2$$

Es claro que una condición suficiente para la A-estabilidad es que el radio espectral de A_{SS-OS} satisfaga $\rho(A_{SS-OS}) \leq 1$. Para problemas con poca difusión, $\sigma \ll \lambda$, el tamaño de paso en tiempo estará limitado por las restricciones de estabilidad impuestas por el método de integración explícito para el término de reacción. Debido a que la reacción está completamente desacoplada en los esquemas de descomposición de operador se puede simplemente usar una expresión más *conveniente*, integración implícita para dicho término. Aquí se considera el método del trapezoide implícito

discretización espacial	SS-OS		TR-OS		IMEX-BDF		IMEX-TR	
	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa	$\ c - c_h\ _0$	Tasa
MDLDG-1	5.74e-04	-	5.76e-04	-	5.75e-04	-	5.76e-04	-
	3.73e-05	4.00	3.73e-05	4.00	3.73e-05	3.95	3.73e-05	4.00
	2.34e-06	4.00	2.34e-06	4.00	2.34e-06	4.00	2.34e-06	4.00
	1.46e-07	3.99	1.46e-07	3.99	1.46e-07	3.99	1.46e-07	3.99
LDG-1	5.69e-04	-	5.69e-04	-	5.68e-04	-	5.69e-04	-
	3.60e-05	4.00	3.60e-05	3.99	3.60e-05	3.99	3.60e-05	4.00
	2.24e-06	4.00	2.24e-06	4.01	2.24e-06	4.01	2.24e-06	4.01
	1.39e-07	4.00	1.39e-07	4.00	1.39e-07	3.99	1.39e-07	3.99
LDG-0	5.56e-04	-	5.56e-04	-	5.54e-04	-	5.56e-04	-
	3.41e-05	4.04	3.41e-05	4.04	3.40e-05	4.03	3.41e-05	4.04
	2.12e-06	4.04	2.12e-06	4.00	2.12e-06	4.00	2.12e-06	4.00
	1.32e-07	4.00	1.32e-07	4.00	1.32e-07	4.00	1.32e-07	4.00

Tabla 5–7: Convergencia con polinomios de grado $p=3$ en espacio

para ambos métodos SS-OS y TR-OS. La matriz de amplificación es

$$\frac{1 + \tau\lambda/2}{1 - \tau\lambda/2}I$$

Observe que la discretización en espacio es por un método discontinuo, así la reacción está desacoplada a nivel de las celdas y esto hace que este paso sea altamente paralelizable.

En el siguiente experimento se establece $\lambda = -10$ y $\sigma = 1/(2\pi^2)$. Note que para este valor de λ la región de estabilidad para el método del trapecioide implícito es $(0, 0.2)$. Se considera un valor de τ el cual viola la condición de estabilidad: $\tau = 0.25$. Para reducir en lo posible que el error del método del gradiente conjugado afecte la precisión de los cálculos se utiliza una tolerancia relativa para el error de $1.0e-14$. De igual manera, la tolerancia relativa para el método de Newton fue de $1.0e-14$. En las Tablas 5–10 y 5–11 se muestra el error L^2 en el tiempo final $T = 1.0$, para mallas con 2048 y 8192 cells respectivamente y polinomios de grado $p = 1$. Todos los métodos usando un método de integración explícito para la reacción son inestables, sin embargo, los métodos de descomposición que utilizan un método implícito para

	SS-OS	TR-OS	IMEX-BDF	IMEX-RK
MDLDG-1	0.72	0.44	0.42	0.46
	8.14	5.11	4.86	5.28
	174.34	121.37	119.88	124.87
	2410.10	1635.39	1634.36	1671.67
LDG-1	0.74	0.43	0.39	0.46
	8.17	5.12	4.87	5.33
	184.69	126.26	124.03	130.33
	2380.10	1594.16	1573.48	1626.17
LDG-0	0.72	0.44	0.40	0.46
	9.37	5.74	5.42	5.46
	217.01	142.45	138.25	145.55
	2681.27	1703.89	1667.65	1741.77

Tabla 5–8: Tiempos de ejecución (en segundos) con polinomios de grado $p = 2$, correspondientes a la Tabla 5–6

la reacción alcanzan una mejor aproximación.

5.2. Modelo de Schnakenberg

El siguiente modelo, introducido en [31], es un sistema de reacción-difusión que relaciona la concentración de dos especies químicas, c_1 y c_2 :

$$\begin{cases} \frac{\partial c_1}{\partial t} &= d_1 \Delta c_1 + \kappa (a - c_1 + c_1^2 c_2) \\ \frac{\partial c_2}{\partial t} &= d_2 \Delta c_2 + \kappa (b - c_1^2 c_2) \end{cases} \quad (5.2)$$

Este sistema con reacción no lineal es un ejemplo de un modelo de activador-substrato ya que la presencia de la concentración c_1 activa la producción de ella misma por consumo de la concentración c_2 . Se ha considerado este problema ya que ha sido utilizado en otros trabajos para comparar diferentes métodos numéricos.

	SS-OS	TR-OS	IMEX-BDF	IMEX-RK
MDLDG-1	0.08	0.05	0.04	0.05
	1.22	0.85	0.75	0.87
	22.58	15.10	13.57	15.60
	480.47	354.27	331.59	363.58
LDG-1	0.07	0.04	0.04	0.05
	1.24	0.86	0.67	0.89
	21.56	13.85	12.00	15.96
	479.27	351.57	322.25	364.48
LDG-0	0.05	0.03	0.03	0.04
	1.17	0.78	0.68	0.83
	23.89	15.71	14.28	16.19
	594.86	440.33	412.14	448.94

Tabla 5–9: Tiempos de ejecución (en segundos) con polinomios de grado $p = 3$, correspondientes a la Tabla 5–7

			reacción explícita		reacción implícita	
	IMEX-BDF	IMEX-RK	SS-OS	TR-OS	SS-OS	TR-OS
MDLDG-1	4.879e+00	3.369e+00	1.279e+00	1.274e+00	1.960e-05	1.949e-05
LDG-1	4.877e+00	3.369e+00	1.277e+00	1.272e+00	1.957e-05	1.946e-05
LDG-0	4.875e+00	3.368e+00	1.276e+00	1.270e+00	1.953e-05	1.942e-05

Tabla 5–10: Reacción lineal y rígida: 2048 celdas con $\tau = 0.25$

5.2.1. Precisión

Se analiza numéricamente la precisión usando la siguiente configuración de parámetros $(d_1, d_2) = (0.05, 1.0)$, $\kappa = 100$ y $(a, b) = (0.1305, 0.7695)$; y, la misma condición inicial utilizada en [19, 37]:

$$\begin{aligned}
 c_1(0, x) &= a + b + 10^{-3} \exp \left(-100 \left((x_0 - 1/3)^2 + (x_1 - 1/2)^2 \right) \right) \\
 c_2(0, x) &= \frac{b}{(a + b)^2},
 \end{aligned}$$

la cual es una perturbación del estado estacionario homogéneo. El dominio espacial es el cuadrado unitario y sobre el borde se imponen condiciones de flujo cero.

	IMEX-BDF	IMEX-RK	reacción explícita		reacción implícita	
			SS-OS	TR-OS	SS-OS	TR-OS
MDLDG-1	4.882e+00	3.369e+00	1.280e+00	1.275e+00	1.964e-05	1.953e-05
LDG-1	4.881e+00	3.369e+00	1.280e+00	1.275e+00	1.963e-05	1.952e-05
LDG-0	4.881e+00	3.369e+00	1.280e+00	1.275e+00	1.962e-05	1.951e-05

Tabla 5–11: Reacción lineal y rígida: 8192 celdas con $\tau = 0.25$

Para los cálculos de esta sección se ha utilizado una secuencia de mallas no estructuradas (m_1, m_2, \dots) . La malla inicial m_1 posee 21 celdas triangulares. El resto de las mallas se obtiene por varios refinamientos globales de manera sucesiva, véase la Figura 5–2 y la Tabla 5–12.

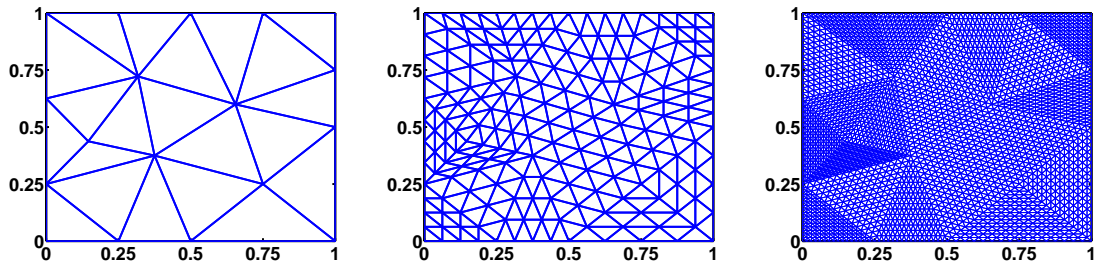


Figura 5–2: Mallas no estructuradas. De izquierda a derecha: mallas m_1 , m_3 y m_5 .

	h_{min}	h_{max}	# Celdas	# Nodos	# Aristas
m_1	3.32e-01	5.00e-01	21	18	38
m_2	1.66e-01	2.50e-01	84	56	139
m_3	8.31e-02	1.25e-01	336	195	530
m_4	4.16e-02	6.25e-02	1344	725	2068
m_5	2.08e-02	3.12e-02	5376	2793	8168

Tabla 5–12: Descripción de mallas para el modelo de Schnakenberg.

Usando un tamaño de paso en tiempo $\tau^* = 3.125 \times 10^{-5}$ la solución exacta para un tiempo final T puede ser aproximada. Luego, es posible estimar el error en

tiempo $E_\tau(x, T)$. Para reducir en lo posible que el error del método del gradiente conjugado afecte la precisión de los cálculos se utiliza una tolerancia relativa para el error de $1.0\text{e-}14$. Los resultados se muestran en las Figuras 5-3, 5-4 y 5-5 para las combinaciones de grados de polinomio y mallas ($p = 1, m_5$), ($p = 2, m_3$) y ($p = 3, m_1$). Se obtuvo una tasa de convergencia de 2 en tiempo, como lo predice la teoría, para todos los métodos de discretización en espacio y tiempo. Sin embargo, los métodos de descomposición de operadores tienen dos ordenes de magnitud con menos precisión que los métodos IMEX. Los resultados son los mismos para cada grado de polinomio.

En las Tablas 5-13, 5-14 y 5-15 se registran los tiempos de ejecución para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio. De los métodos de integración en tiempo, el TR-OS, IMEX-BDF y IMEX-RK muestran un costo computacional bastante similar para todas las discretizaciones en espacio. El SS-OS es el que muestra el mayor costo computacional, requiere aproximadamente 2 veces el tiempo de ejecución que el resto. Para las discretizaciones en espacio, no parece haber diferencia sustancial en el tiempo de computo de los métodos LDG-1 y MDLDG-1, sin embargo el método LDG-0 requiere casi 2 veces más tiempo.

En las Figuras 5-6, 5-7, 5-8 y 5-9 se ilustra la evolución en tiempo de la concentración c_1 para los diferentes métodos de integración en tiempo con un tamaño de paso de $\tau = 1.0\text{e-}3$. La discretización en espacio fue realizada por el MDLDG-1. Se puede observar que los esquemas con métodos de descomposición tienen más disipación que con los métodos IMEX.

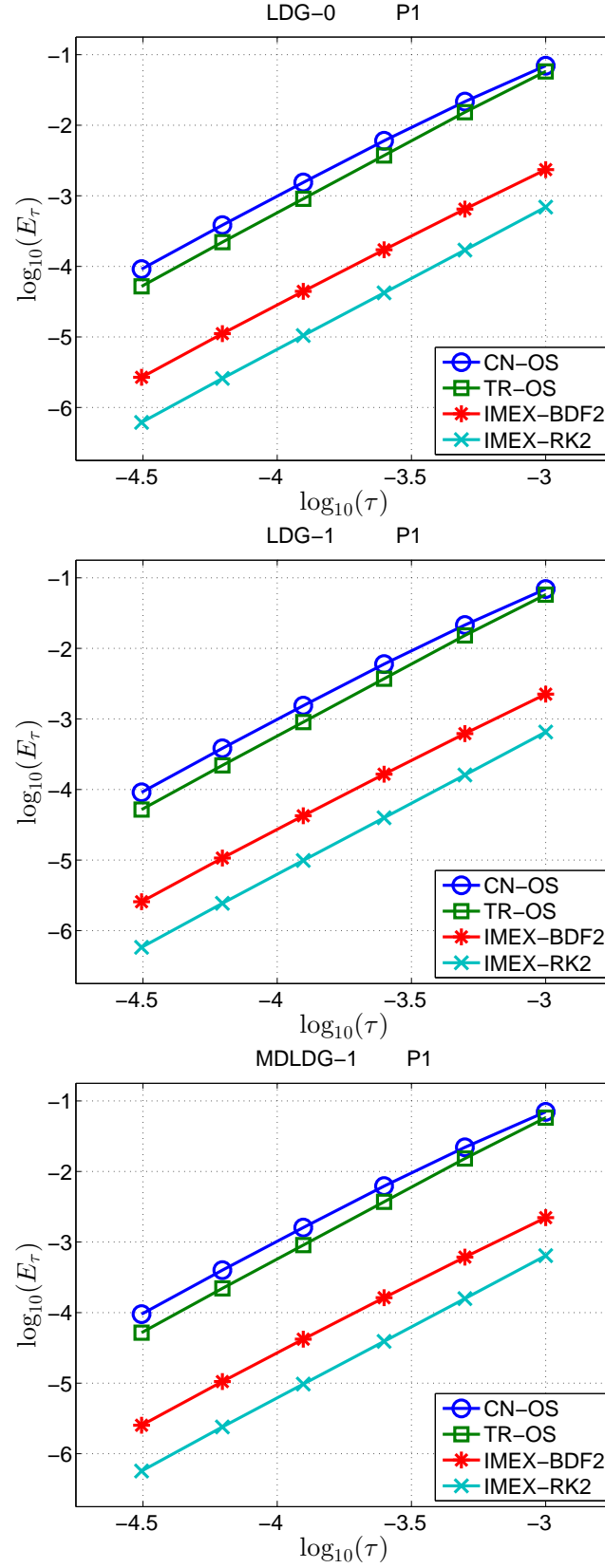


Figura 5-3: Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 1$. En todos los casos se observa convergencia en tiempo de segundo orden.

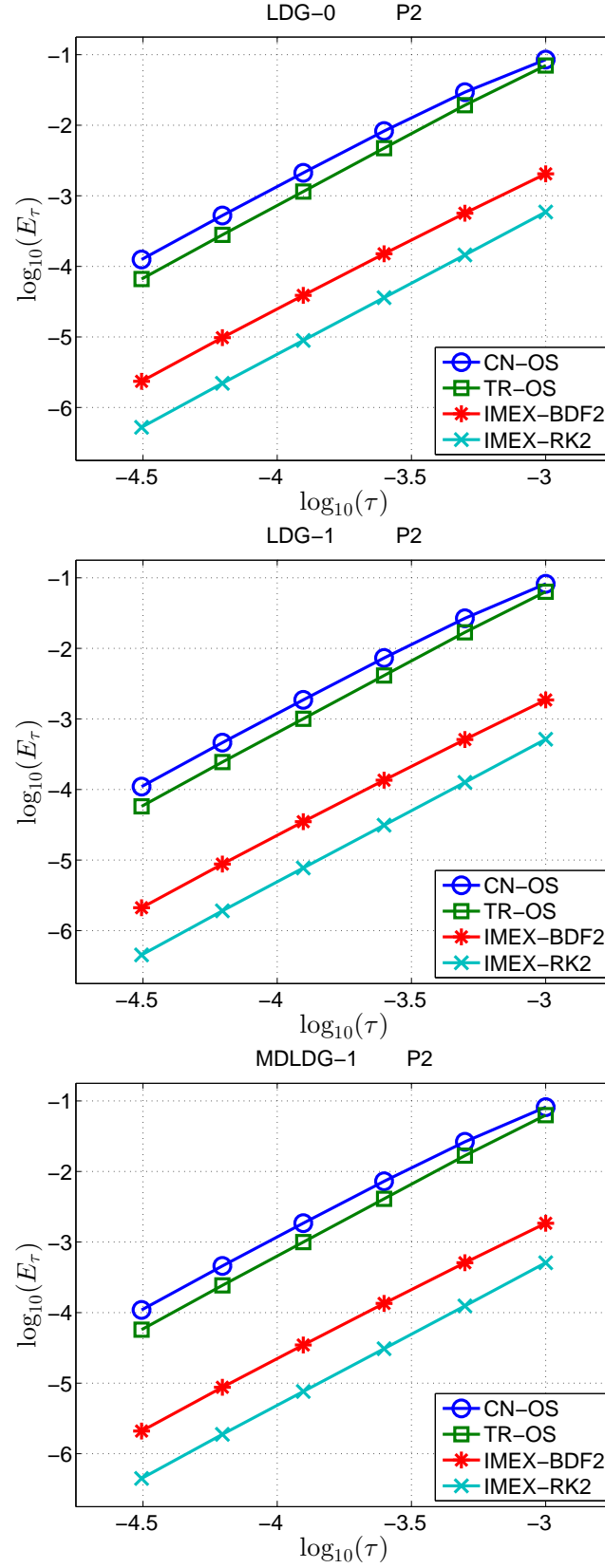


Figura 5-4: Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 2$. En todos los casos se observa convergencia en tiempo de segundo orden.

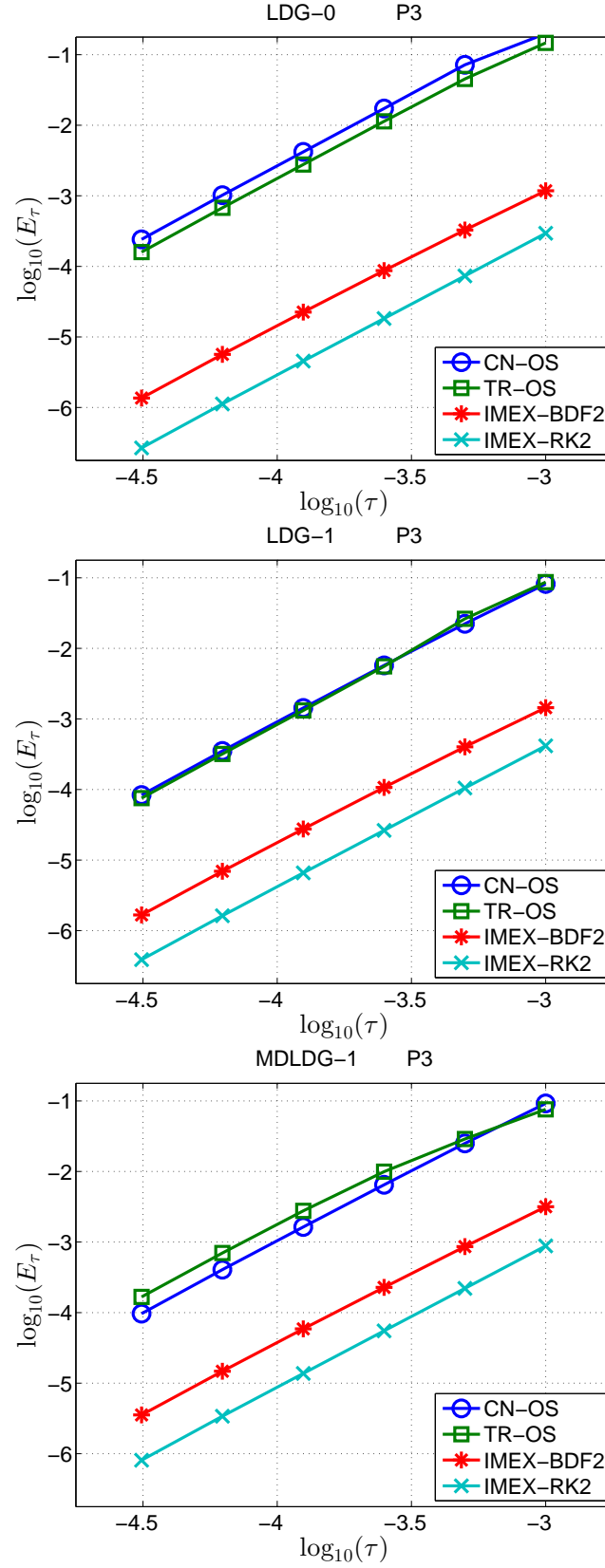


Figura 5-5: Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 3$. En todos los casos se observa convergencia en tiempo de segundo orden.

CN-OS			TR-OS			IMEX-BDF			IMEX-RK		
$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$
1622.78	98.17	9.61	1064.59	54.42	5.34	1153.35	53.20	5.28	1064.38	57.54	5.47
2571.86	204.08	20.38	1682.30	112.36	11.02	1791.75	108.19	10.88	1677.37	111.19	11.18
4207.72	420.31	43.97	2647.06	226.50	23.14	2787.57	204.24	22.16	2663.47	222.19	23.43
7152.66	908.87	90.25	4365.45	456.36	49.33	4417.50	433.91	48.73	4357.53	456.11	49.43
13317.60	1828.78	180.72	7524.01	968.60	100.14	7418.37	894.18	99.85	7568.07	969.55	101.13
25548.70	3603.23	378.39	13803.4	1982.28	202.06	13234.80	1843.20	195.04	13887.40	1945.29	200.05

Tabla 5–13: Tiempos de cómputo (en segundos) para discretizaciones espaciales con el LDG-0.

SS-OS			TR-OS			IMEX-BDF			IMEX-RK		
$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$
1240.29	68.02	6.87	825.61	39.46	4.13	896.53	37.70	3.65	827.20	40.86	4.60
1883.52	137.69	14.71	1296.84	80.08	8.47	1385.09	74.63	7.39	1293.18	82.24	9.31
2942.64	263.98	30.84	1946.53	154.21	17.09	2054.95	141.66	14.97	1948.13	157.24	18.98
4958.88	510.81	65.44	3113.86	292.34	38.39	3146.88	266.93	31.71	3111.70	299.69	41.05
8690.09	1104.72	135.79	5324.51	588.19	77.41	5170.66	518.23	65.72	5291.63	601.84	84.85
15460.7	2210.32	270.25	9452.05	1241.04	155.16	8935.13	1096.68	134.48	9387.96	1262.89	171.69

Tabla 5–14: Tiempos de cómputo (en segundos) para discretizaciones espaciales con el LDG-1.

SS-OS			TR-OS			IMEX-BDF			IMEX-RK		
$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$
1178.8	72.63	7.85	785.27	42.72	4.49	849.35	39.08	3.66	787.41	42.86	4.62
1822.15	141.13	16.48	1243.58	82.78	9.39	1328.19	75.48	7.24	1239.15	82.62	9.29
2973.51	266.29	35.29	1924.20	158.79	19.19	2027.92	142.29	14.95	1931.52	157.23	19.73
4920.77	522.90	72.33	3089.09	301.13	40.73	3137.27	268.96	31.91	3108.59	300.60	40.79
8768.49	1121.37	146.96	5249.21	599.55	82.65	5201.32	509.26	64.26	5256.87	599.01	82.60
15737.1	2252.18	295.44	9522.15	1269.13	168.37	9070.29	1094.48	131.18	9544.12	1259.60	167.02

Tabla 5–15: Tiempos de cómputo (en segundos) para discretizaciones espaciales con el MDLDG-1.

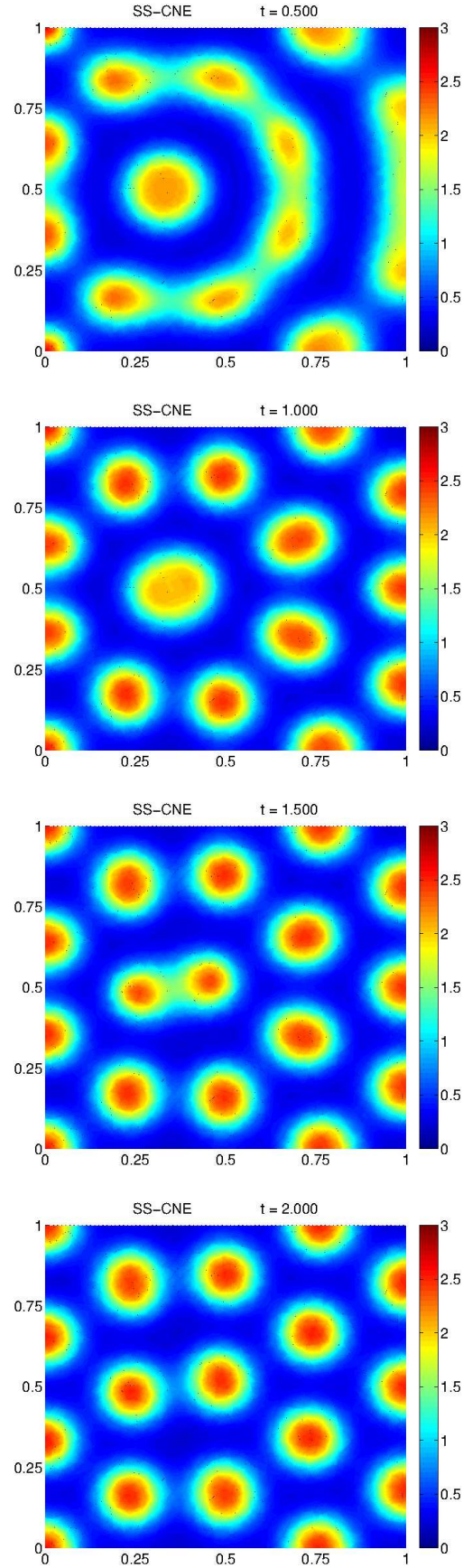


Figura 5-6: Evolución en tiempo de la concentración c_1 con los métodos SS-OS con reacción explícita y MDLDG-1.

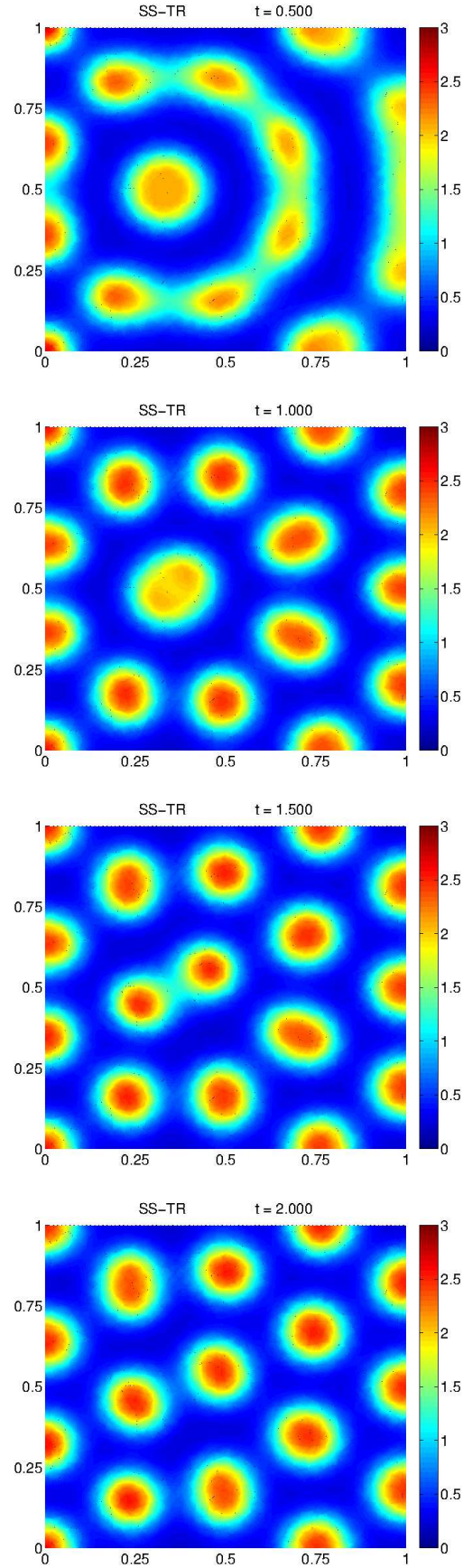


Figura 5–7: Evolución en tiempo de la concentración c_1 con los métodos TR-OS con reacción explícita y MDLDG-1.

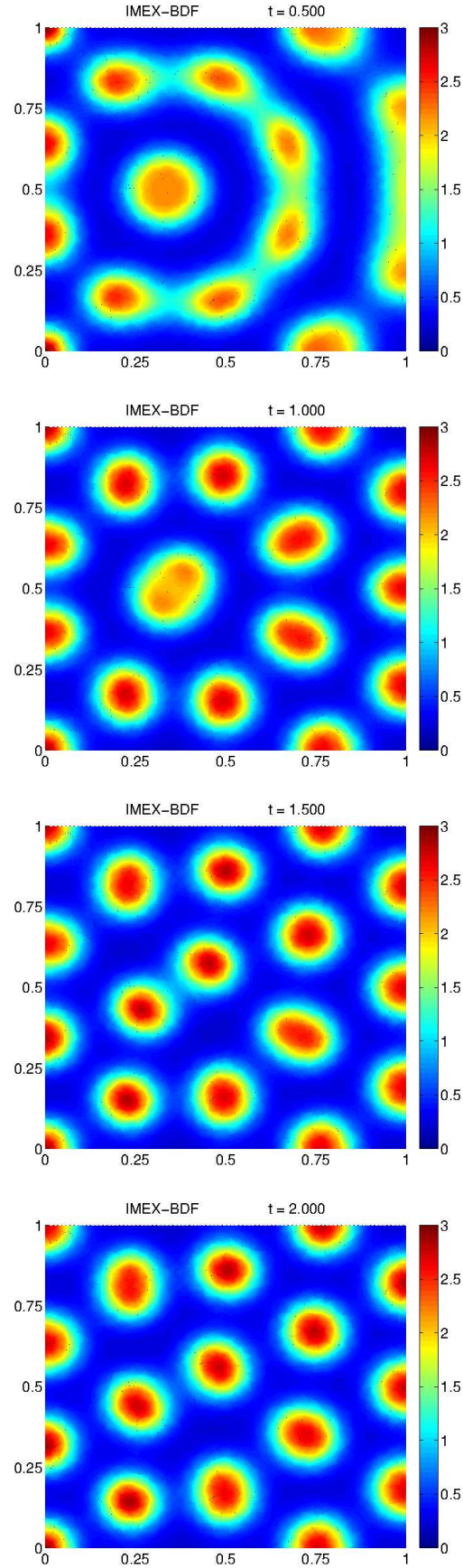


Figura 5–8: Evolución en tiempo de la concentración c_1 con los métodos IMEX-BDF con reacción explícita y MDLDG-1.

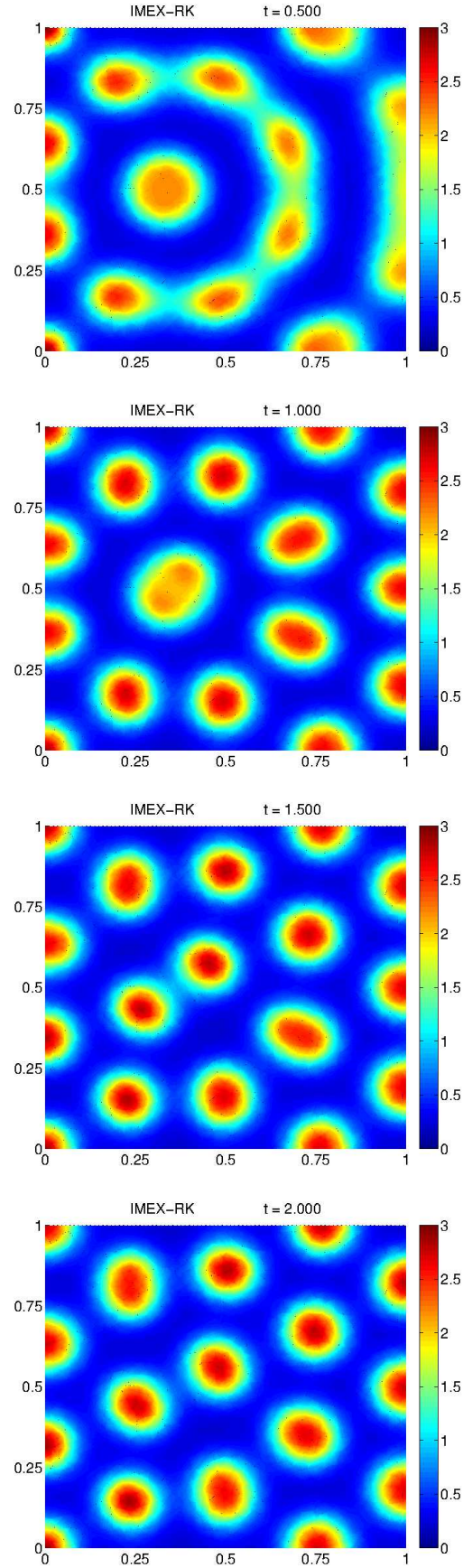


Figura 5–9: Evolución en tiempo de la concentración c_1 con los métodos IMEX-RK con reacción explícita y MDLDG-1.

5.3. Modelo de pigmentación

La pigmentación en los mamíferos se debe a un tipo de células llamadas *melanocitos* las cuales son las responsables de la producción de *melanina* (pigmento que da color al pelo). Se cree que la producción (o no producción) de melanina depende de la concentración de ciertos morfógenos. El siguiente sistema de reacción-difusión, obtenido de [22], modela la pigmentación en pieles de diferentes animales

$$\begin{cases} \frac{\partial c_1}{\partial t} &= d_1 \Delta c_1 + \rho_a \left[\frac{c_1^2 c_2}{1 + \kappa_1 c_1^2} - c_1 \right] \\ \frac{\partial c_2}{\partial t} &= d_2 \Delta c_2 + \frac{\sigma_2}{1 + \kappa_2 c_3} - \frac{\rho_2 c_1^2 c_2}{1 + \kappa_1 c_1^2} - \mu_2 c_2 \\ \frac{\partial c_3}{\partial t} &= \rho_3 \frac{c_3^2}{1 + \kappa_3 c_3^2} - \mu_3 c_3 + \sigma_3 c_1 \end{cases} \quad .$$

Este sistema es una combinación de un modelo activador-substrato (c_1, c_2) y un conmutador c_3 el cual se activa de 0 a 1 con la presencia del activador c_1 . Los melanocitos que producen melanina serían aquellos donde $c_3 = 1$, mientras los que no producen donde $c_3 = 0$. Las concentraciones de c_1 y c_2 son los morfógenos encargados de activar la producción de c_3 para la producción de la melanina. Inicialmente $c_3 = 0$, $c_1 = 0$ y $c_2 = c_2^0$ en todas partes, excepto en ciertos puntos aleatorios donde $c_1 = c_1^0$.

5.3.1. Precisión

Para los cálculos de esta sección se ha utilizado una secuencia de mallas no estructuradas (m_1, m_2, \dots) . La malla inicial m_1 posee 20 celdas triangulares. El resto de las mallas se obtiene por varios refinamientos globales de manera sucesiva, véase

la Figura 5-10 y la Tabla 5-16.

Se han utilizado los siguientes valores de parámetros obtenidos de [22]: $(d_1, d_2) = (0.015, 0.03)$, $(\rho_1, \rho_2, \rho_3) = (0.025, 0.0025, 0.03)$, $(\mu_2, \mu_3) = (0.00075, 0.003)$, $(\sigma_2, \sigma_3) = (0.00225, 0.00015)$ y $(\kappa_1, \kappa_2, \kappa_3) = (0.1, 20.0, 22.0)$. Se impusieron condiciones de borde de flujo cero. Las condiciones iniciales son $c_1 = 0$, $c_2 = 3$ y $c_3 = 0$ en todas partes, excepto en ciertos puntos aleatorios donde $c_1 = 5$.

Usando un tamaño de paso en tiempo $\tau^* = 3.125 \times 10^{-5}$ la solución exacta para un tiempo final $T = 1.0$ puede ser aproximada. Luego, es posible estimar el error en tiempo $E_\tau(x, T)$. Para reducir en lo posible que el error del método del gradiente conjugado afecte la precisión de los cálculos se utiliza una tolerancia relativa para el error de $1.0e-14$. Los resultados se muestran en la Figuras , y para las combinaciones de grados de polinomio y mallas $(p = 1, m_5)$, $(p = 2, m_3)$ y $(p = 3, m_1)$. Para cada método de integración en tiempo se obtuvo una tasa de convergencia de orden 2 para todas las discretizaciones en espacio. Se observa que los métodos TR-OS, IMEX-BDF y IMEX-RK tienen una precisión muy similar, mientras que el SS-OS considerablemente menos preciso.

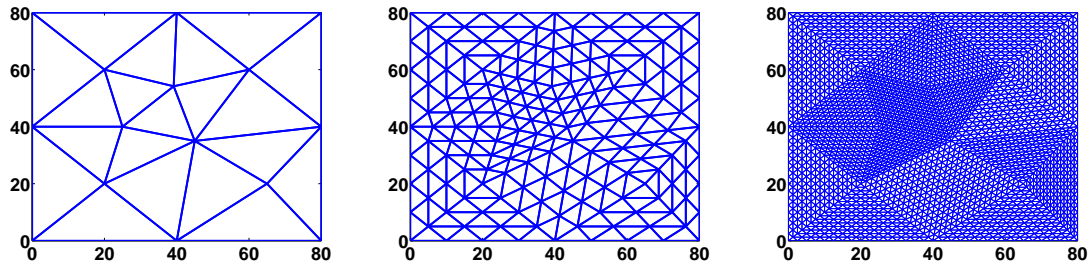


Figura 5-10: Mallas no estructuradas. De izquierda a derecha: mallas m_1 , m_3 y m_5 .

En las Tablas 5-17, 5-18 y 5-19 se registran los tiempos de ejecución para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en

	h_{min}	h_{max}	# Celdas	# Nodos	# Aristas
m_1	2.06e+01	4.00e+01	20	15	34
m_2	1.03e+01	2.00e+01	80	49	128
m_3	5.15e+00	1.00e+01	320	177	496
m_4	2.58e+00	5.00e+00	1280	673	1952
m_5	1.29e+00	2.50e+00	5120	2625	7744

Tabla 5–16: Descripción de mallas para el modelo de pigmentación.

espacio. De los métodos de integración en tiempo, el TR-OS, IMEX-BDF y IMEX-RK muestran un costo computacional bastante similar para todas las discretizaciones en espacio. El SS-OS es el que muestra el mayor costo computacional, requiere aproximadamente 2 veces el tiempo de ejecución que el resto. Para las discretizaciones en espacio, no parece haber diferencia sustancial en el tiempo de computo de los métodos LDG-1 y MDLDG-1, sin embargo el método LDG-0 requiere casi 2 veces más tiempo.

SS-OS			TR-OS			IMEX-BDF			IMEX-RK		
$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$
751.38	114.94	13.21	416.09	68.64	8.22	434.05	72.48	9.44	489.65	82.39	10.11
1499.94	227.85	26.21	847.65	137.43	16.34	885.57	140.84	18.66	995.12	161.23	20.08
3031.84	432.04	52.21	1684.26	265.02	32.78	1766.47	274.41	37.52	1987.85	310.97	39.74
5706.25	858.83	103.51	3354.67	508.69	64.44	3538.77	530.91	73.78	4018.05	607.57	79.21
10455.70	1678.40	205.64	6258.28	1020.48	128.38	6537.64	1061.91	146.76	7308.56	1229.13	157.41
20575.70	3254.38	407.64	11741.60	1985.89	293.30	12269.20	2104.72	293.30	14207.60	2396.77	315.47

Tabla 5–17: Tiempos de cómputo (en segundos) para discretizaciones espaciales con el LDG-0.

SS-OS			TR-OS			IMEX-BDF			IMEX-RK		
$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$
443.10	66.65	11.68	270.70	41.60	7.46	260.87	38.80	6.75	303.01	47.37	8.33
880.16	141.23	23.12	531.93	88.49	15.00	515.15	82.92	13.30	603.91	100.37	16.53
1690.99	288.86	46.19	1040.49	178.49	29.91	1001.64	166.27	26.39	1179.01	201.77	31.76
3228.30	579.78	91.06	2006.73	369.44	59.01	1926.58	343.84	53.15	2285.88	417.20	65.51
6257.39	1122.11	181.42	3841.86	723.36	122.46	3687.56	677.04	104.81	4465.10	823.68	130.37
13096.90	2190.03	359.53	7680.39	1416.03	236.33	7291.91	1330.41	208.00	9011.06	1606.03	258.71

Tabla 5–18: Tiempos de cómputo (en segundos) para discretizaciones espaciales con el LDG-1.

SS-OS			TR-OS			IMEX-BDF			IMEX-RK		
$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$	$p = 1$	$p = 2$	$p = 3$
431.24	67.43	11.67	262.06	41.97	7.78	252.73	39.07	6.65	296.07	48.55	8.98
872.88	137.74	22.98	527.19	87.11	14.94	512.04	81.56	13.23	604.65	99.94	17.95
1705.33	289.56	45.68	1042.27	177.25	29.32	1001.51	163.36	26.08	1200.32	206.76	35.47
3330.29	574.82	90.84	2068.58	366.19	58.33	1957.96	340.60	52.00	2374.00	423.81	70.94
6348.64	1116.04	180.37	4026.54	718.03	116.08	3797.02	672.78	104.43	4567.69	831.83	141.41
12913.90	2187.45	362.20	7859.09	1398.82	230.72	7361.95	1317.81	205.82	9028.16	1637.30	281.22

Tabla 5–19: Tiempos de cómputo (en segundos) para discretizaciones espaciales con el MDLDG-1.

En la Figura 5–14 se ilustra el patrón final obtenido para concentración del conmutador y para los diferentes métodos de integración en tiempo con un tamaño de paso de $\tau = 1.0$. La discretización en espacio fue realizada por el MDLDG-1.

Variando los parámetros se puede obtener un patrón bastante diferente. Por ejemplo los valores obtenidos de [22] se utilizan para generar una pigmentación similar a la del leopardo: $(d_1, d_2) = (0.01, 0.1)$, $(\rho_1, \rho_2, \rho_3) = (0.05, 0.0035, 0.03)$, $(\mu_2, \mu_3) = (0.003, 0.003)$, $(\sigma_2, \sigma_3) = (0.0075, 0.00007)$ y $(\kappa_1, \kappa_2, \kappa_3) = (0.5, 0.3, 22.0)$ con condiciones de borde de flujo cero y condiciones iniciales $c_1 = 0$, $c_2 = 2.5$ y $c_3 = 0$ en todas partes, excepto en los mismos puntos, del experimento anterior, donde $c_1 = 2$. En la Figura 5–15 se ilustra el patrón final obtenido para concentración del conmutador y para los diferentes métodos de integración en tiempo con un tamaño de paso de $\tau = 1.0$. La discretización en espacio fue realizada por el MDLDG-1.

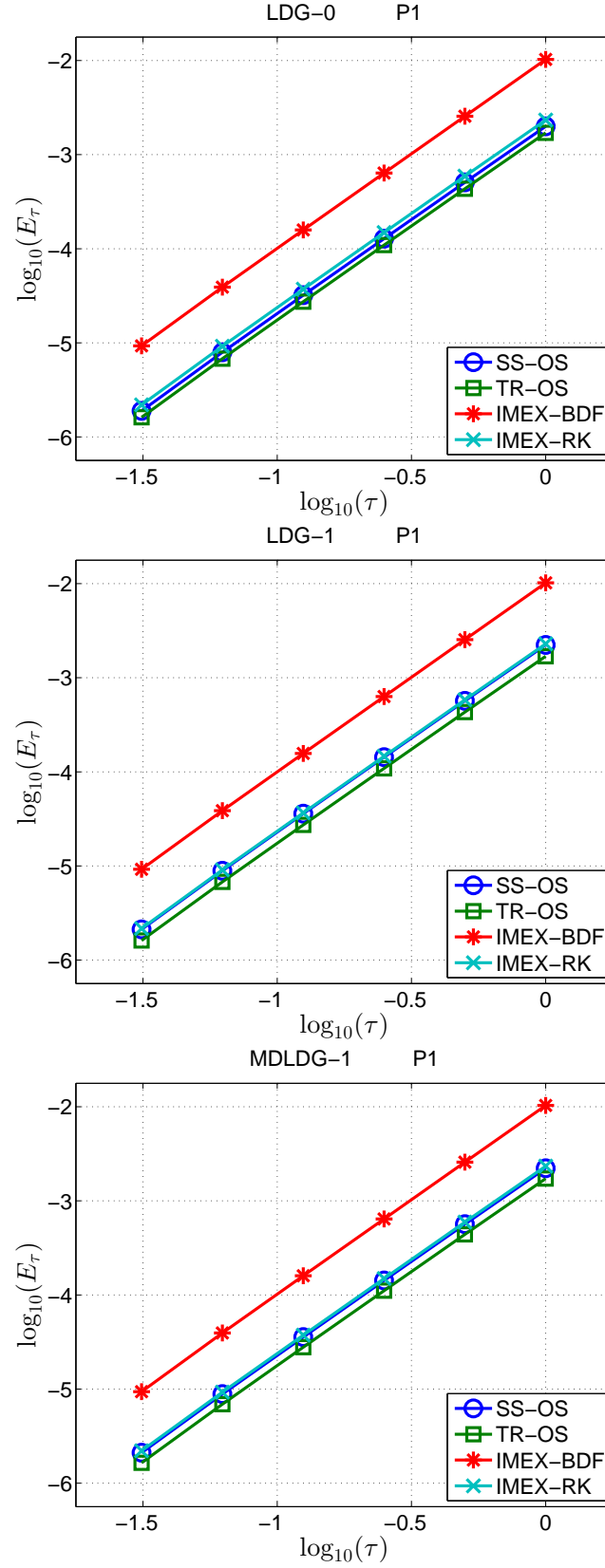


Figura 5-11: Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 1$. En todos los casos se observa convergencia en tiempo de segundo orden.

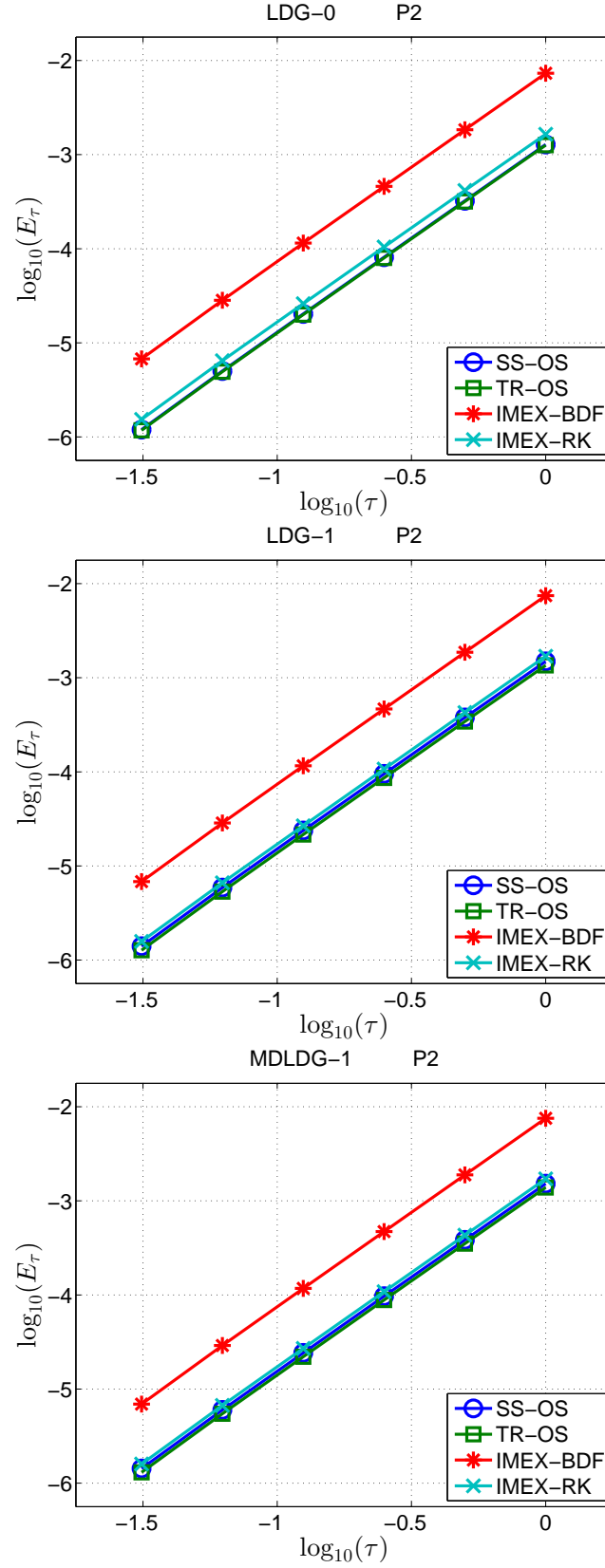


Figura 5-12: Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 2$. En todos los casos se observa convergencia en tiempo de segundo orden.

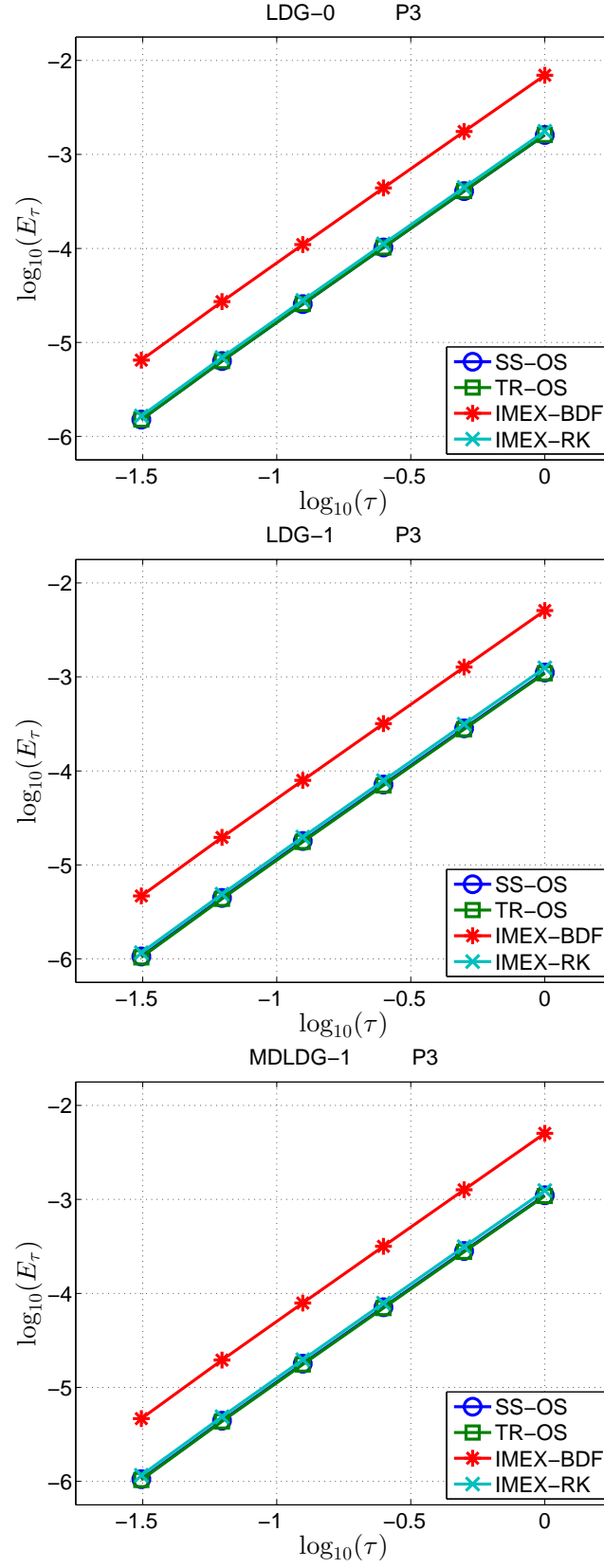


Figura 5-13: Error E_τ en $T = 1$ para los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con diferentes discretizaciones en espacio con polinomios de grado $p = 3$. En todos los casos se observa convergencia en tiempo de segundo orden.

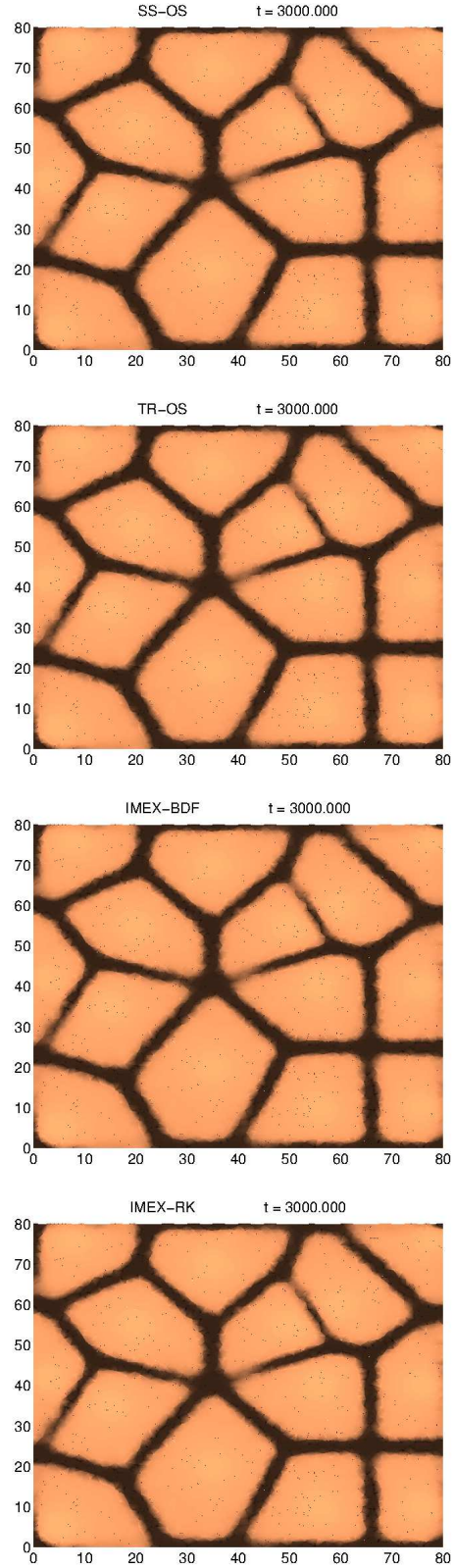


Figura 5–14: Patrón para un tiempo final $T = 3000$ formado por la concentración del conmutador c_3 usando los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con discretización en espacio por el MDLDG-1.

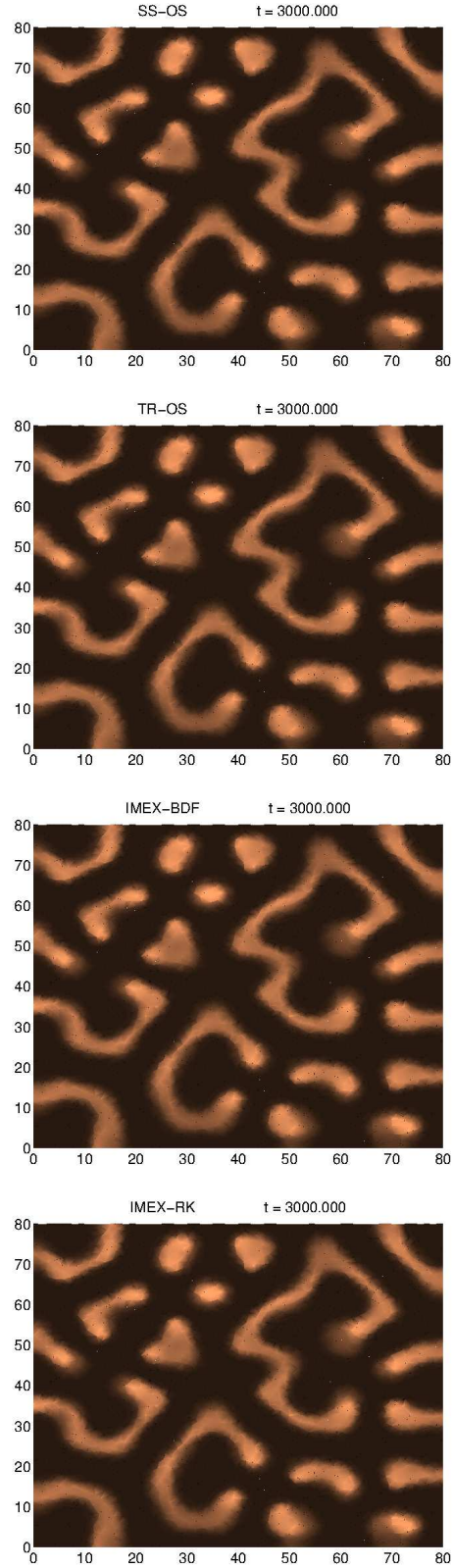


Figura 5–15: Patrón para un tiempo final $T = 3000$ formado por la concentración del conmutador c_3 usando los métodos SS-OS, TR-OS, IMEX-BDF y IMEX-RK con discretización en espacio por el MDLDG-1.

Capítulo 6

CONCLUSIONES Y TRABAJO A FUTURO

6.1. Conclusiones

Un marco computacional para la solución numérica de sistemas no lineales de reacción-difusión en 2D con múltiples especies químicas fue implementado. El número de especies químicas es arbitrario. Se asume que los coeficientes de difusión de cada especie química son constantes e independientes entre sí, es decir, la matriz de difusión es diagonal. Por otro lado, el operador de reacción que acopla el sistema puede ser no lineal.

Un marco computacional con estas características puede ser aplicado en el estudio de diferentes procesos químicos, físicos y biológicos. Para ilustrar alguna de estas aplicaciones se han considerado dos modelos de reacción-difusión. El primero es el bien estudiado modelo de Schnakenberg utilizado en el modelamiento de diferentes procesos químicos y biológicos. El segundo es un modelo para la pigmentación de pieles de animales. En ambos sistemas la formación de estructuras de Turing es importante.

Para la discretización en espacio el marco computacional utiliza el método Local discontinuous Galerkin. La elección de este método se debe a diferentes factores. Primero, en muchas aplicaciones es importante captar suficiente complejidad de la geometría del dominio espacial. Además, el método permite aproximaciones de alto orden lo cual es ideal en problemas con soluciones con alta regularidad ya que el

número de grados de libertad puede ser reducido y la eficiencia puede ser incrementada. Segundo, la discretización del operador de reacción puede realizarse de manera independiente en cada celda de la malla lo cual es de gran ventaja cuando se utiliza un método de descomposición de operador para la integración en tiempo ya que cuando la reacción es rígida no se necesita resolver un sistema algebraico no lineal de gran tamaño sino varios sistemas no lineales locales en cada celda. Además, este proceso puede ser paralelizado directamente. Tercero, en [37] se argumentó que no era recomendado el uso del LDG en sistemas de reacción difusión no lineales debido a que era necesario introducir una nueva variable auxiliar la cual incrementa el número de grados de libertad. Sin embargo, aquí se ha utilizado un algoritmo propuesto en [11] donde no se requiere una nueva variable auxiliar para la implementación del LDG. Finalmente, uno de los intereses es establecer una comparación entre el LDG en su versión clásica y el LDG en su versión de disipación mínima. Los experimentos realizados en este trabajo sugieren que no hay diferencia sustancial entre ambos métodos para problemas no lineales de reacción difusión, sin embargo, en modelos donde las condiciones de borde son de tipo Neumann la versión con disipación mínima no requiere el ensamblado de la matriz de estabilidad.

Para la discretización en tiempo dos tipos de métodos de integración en tiempo fueron implementados. El primer tipo son métodos de descomposición de operador los cuales son ideales en problemas no lineales de reacción difusión ya que permiten desacoplar ambos términos. Se recalca que como la discretización en espacio se realizó por un método discontinuo, entonces el paso de reacción es local en cada celda de la malla. Así, cuando la reacción es rígida no es necesario resolver un sistema algebraico no lineal global en toda la malla, sino uno local en cada celda. Para resolver estos pequeños sistemas no lineales se utilizó el método de Newton, y para los sistemas lineales asociados a la iteración de Newton se utilizó LAPACK

[1]. Particularmente se implementaron dos métodos basados en la descomposición simétrica de Strang. Ambos métodos tienen precisión de orden dos en tiempo. El segundo tipo son los métodos implícito-explicito los cuales son ideales en sistemas de reacción difusión con reacción no rígida ya que en estos métodos la difusión se puede integrar de manera implícita y la reacción de manera explícita. Particularmente se consideraron dos métodos IMEX de orden dos. Uno está basado en un método BDF, mientras que el otro está basado en un método de Runge-Kutta. En base a los experimentos realizados en este trabajo se concluye que no se puede decidir en cual de ambos tipos de métodos es más preciso. En el modelo de Schnakenberg se observó que los métodos IMEX son 2 ordenes de magnitud más precisos que los métodos de descomposición. En el modelo de pigmentación en pieles de animales no se observó una diferencia sustancial en la precisión de ambos tipos de métodos. En un problema lineal y escalar de reacción difusión se observó que si la reacción tiene el mismo punto de equilibrio que el sistema de reacción-difusión entonces no hay diferencia sustancial entre ambos tipos de métodos, pero en caso contrario los métodos IMEX pueden ser considerablemente más precisos. Para el mismo problema lineal y escalar se verifica que cuando la reacción es rígida es recomendable el uso de los métodos de descomposición ya que los métodos IMEX muestran ser inestables en tal caso. En cuanto al costo computacional se observa aproximadamente el mismo costo computacional, excepto en el método de descomposición SS-OS que muestra el mayor costo con un factor de costo de alrededor de 1.5 respecto a los otros métodos, lo cual se debe a que es necesario la solución de un sistema lineal adicional que el resto.

El marco computacional fue implementado en el lenguaje C++. La implementación es orientada a objetos, lo cual facilita su extensión y mantenimiento. Por ejemplo, otros tipos de bases de polinomios pueden ser incorporados mediante una simple herencia de una clase abstracta para las bases. Otros métodos de integración

en tiempo pueden ser incorporados como una clase que hereda de la clase abstracta **Operator**.

6.2. Trabajos a futuro

Precondicionamiento: Los esquemas numéricos utilizados en el marco computacional requieren la solución de sistemas lineales de gran tamaño en cada paso de tiempo. Para tomar ventaja del patrón del gran número de entradas no cero se utiliza una técnica iterativa, particularmente el método del gradiente conjugado ya que los sistemas son simétricos definidos positivos. Luego, el número de iteraciones necesarias para la convergencia depende del condicionamiento del sistema. Una técnica de precondicionamiento sobre estos sistemas lineales puede ayudar a reducir el número de iteraciones y, por lo tanto, el tiempo de cómputo. En este trabajo no se evaluó el efecto de una técnica de precondicionamiento. Para una posible técnica de precondicionamiento de estos sistemas, véase [11, 12].

Alto orden en tiempo: Actualmente, los métodos de integración en tiempo implementados tienen segundo orden de precisión. Para métodos de descomposición de operador y métodos IMEX con orden de precisión mayor que dos, véase [2, 3, 19, 20].

Implementación en paralelo: Muchas rutinas del marco computacional pueden ser totalmente paralelizadas. Por ejemplo, los cálculos sobre el operador de reacción son paralelizables a nivel de las celdas de la malla, la solución del sistema lineal puede ser paralelizado a nivel de cada especie química. Una implementación en paralelo del marco computacional queda como trabajo a futuro.

Bibliografía

- [1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users'Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.
- [2] U. Ascher, S. Ruuth, and R. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2):151–167, 1997.
- [3] U. Ascher, S. Ruuth, and B. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Num. Anal.*, 32(3):797–823, 1995.
- [4] V. Castets, E. Dulos, J. Boissonade, and P. D. Kepper. Experimental evidence of a sustained standing Turing-type nonequilibrium chemical pattern. *Phys. Rev. Lett.*, 64:2953–2956, 1990.
- [5] P. Castillo. An optimal error estimate for the Local Discontinuous Galerkin method. In B. Cockburn, G. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, volume 11 of *Lecture Notes in Computational Science and Engineering*, pages 285–290. Springer Verlag, February 2000.
- [6] P. Castillo. Performance of discontinuous Galerkin methods for elliptic PDEs. *SIAM J. Sci. Comput.*, 24(2):524–547, 2002.
- [7] P. Castillo. Computational aspects of the Local Discontinuous Galerkin method for non conforming meshes in 2D. *Rev. Int. Métodos Numér. Cál. Diseño Ing.*, 26(4):283–303, 2010.

- [8] P. Castillo. Stencil reduction algorithms for the Local Discontinuous Galerkin method. *Internat. J. Numer. Methods Engrg.*, 81(12):1475–1491, 2010.
- [9] P. Castillo, B. Cockburn, I. Perugia, and D. Schötzau. An a priori error analysis of the Local Discontinuous Galerkin method for elliptic problems. *SIAM J. Num. Anal.*, 38(5):1676–1706, 2000.
- [10] P. Castillo, B. Cockburn, D. Schötzau, and C. Schwab. An optimal a priori error estimate for the *hp*-version of the local discontinuous Galerkin method for convection-diffusion problems. *Math. Comp.*, 71(238):455–478, 2001.
- [11] P. Castillo and F. Sequeira. Computational aspects of the Local Discontinuous Galerkin method on unstructured grids in three dimensions. *Mathematical and Computer Modelling*, 57(9–10):2279 – 2288, 2013.
- [12] P. Castillo and E. Velázquez. A numerical study of a semi-algebraic multilevel preconditioner for the Local Discontinuous Galerkin method. *Internat. J. Numer. Methods Engrg.*, 74(2):255–268, 2008.
- [13] B. Cockburn and C. Dawson. Some extensions of the Local Discontinuous Galerkin method for convection diffusion equations in multidimensions. In J. Whiteman, editor, *Proceedings of the 10th Conference on the Mathematics of Finite Elements and Applications*, pages 225–238, 2000.
- [14] B. Cockburn and B. Dong. An analysis of the minimal dissipation Local Discontinuous Galerkin method for convection diffusion problems. *SIAM J. Sci. Comput.*, 32:233–262, 2007.
- [15] B. Cockburn, G. Kanschat, I. Perugia, and D. Schötzau. Superconvergence of the Local Discontinuous Galerkin method for elliptic problems on Cartesian grids. *SIAM J. Num. Anal.*, 39(1):264–285, 2001.
- [16] B. Cockburn and C. Shu. The Local Discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Num. Anal.*, 35(6):2440–2463, 1998.

- [17] Y. Epshteyn. Discontinuous Galerkin methods for the chemotaxis and haptotaxis models. *J. Comput. Applied. Math.*, 224(1):168 – 181, 2009.
- [18] W. Hundsdorfer. Trapezoidal and midpoint splittings for initial-boundary value problems. *Math. Comp.*, 67(223):1047–1062, 1998.
- [19] W. Hundsdorfer and J. G. Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations*, volume 33. Springer, 2007.
- [20] H. Jia and K. Li. A third accurate operator splitting method. *Mathematical and Computer Modelling*, 53(1):387–396, 2011.
- [21] A. Kassam and L. Trefethen. Fourth-order time stepping for stiff PDEs. *SIAM J. Sci. Comput.*, 26:1214–1233, 2005.
- [22] A. Koch and H. Meinhardt. Biological pattern formation: from basic mechanisms to complex structures. *Reviews of Modern Physics*, 66(4):1481, 1994.
- [23] A. Madzvamuse. Time-stepping schemes for moving grid finite elements applied to reaction-diffusion systems on fixed and growing domains. *J. Comput. Phys.*, 214:239–263, 2006.
- [24] A. Madzvamuse and P. Maini. Velocity-induced numerical solutions of reaction-diffusion systems on continuously growing domains. *J. Comput. Phys.*, 225:100–119, 2007.
- [25] P. Maini, K. Painter, and H. N. P. Chau. Spatial pattern formation in chemical and biological systems. *Journal of the Chemical Society, Faraday Transactions*, 93(20):3601–3610, 1997.
- [26] L. Marcon and J. Sharpe. Turing patterns in development: what about the horse part? *Current Opinion in Genetics and Development*, 22(6):578 – 584, 2012.
- [27] A. Mikhailov and K. Showalter. Control of waves, patterns and turbulence in chemical systems. *Physics Reports*, 425(2–3):79 – 194, 2006.

- [28] I. Perugia and D. Schötzau. An hp-analysis of the Local Discontinuous Galerkin method for diffusion problems. *J. Scientific Computing.*, 17(1-4):561–571, 2002.
- [29] S. Ruuth. Implicit-explicit methods for reaction-diffusion problems in pattern formation. *Journal of Mathematical Biology*, 34(2):148–176, 1995.
- [30] Y. Saad. *Iterative methods for sparse linear systems*. Siam, 2003.
- [31] J. Schnakenberg. Simple chemical reaction systems with limit cycle behaviour. *Journal of theoretical biology*, 81(3):389–400, 1979.
- [32] I. Sgura, B. Bozzini, and D. Lacitignola. Numerical approximation of Turing patterns in electrodeposition by ADI methods. *J. Comput. Applied. Math.*, 236:4132–4147, 2012.
- [33] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Num. Anal.*, 5(3):506–517, 1968.
- [34] A. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, 1952.
- [35] R. Tyson, L. Stern, and R. LeVeque. Fractional step methods applied to a chemotaxis model. *Journal of mathematical biology*, 41(5):455–475, 2000.
- [36] V. Vanag and I. Epstein. Design and control of patterns in reaction-diffusion systems. *Chaos*, 18(2):11, 2008.
- [37] J. Zhu, Y. Zhang, S. Newman, and M. Alber. Application of discontinuous galerkin methods for reaction-diffusion systems in developmental biology. *J. Scientific Computing.*, 40(1-3):391–418, 2009.

APLICACIONES DEL MÉTODO LDG A SISTEMAS QUÍMICOS Y BIOLÓGICOS CON ESTRUCTURAS DE TURING

Iván Yessel Henríquez Rivera

ivan.henriquez@upr.edu

Departamento de Matemática

Consejero: Paul E. Castillo

Grado: Maestría en Ciencias

Fecha de Graduacion: 2013

Este trabajo está dedicado a la solución numérica de sistemas no lineales de reacción difusión. Estos sistemas son importantes en el modelamiento de sistemas químicos y biológicos donde se observa la formación de patrones. Se implementó un marco computacional para la solución eficiente de sistemas no lineales de reacción difusión 2D, donde el número de especies químicas es arbitrario. Utilizando el método *Local Discontinuous Galerkin* (LDG) el marco permite aproximaciones espaciales de orden arbitrario. Para la integración en tiempo el usuario puede seleccionar entre métodos de *descomposición de operador* (DO) o métodos *implícito-explicito* (IMEX). Estas combinaciones de discretizaciones en espacio y tiempo permiten resolver los sistemas no lineales de reacción difusión en forma eficiente. Finalmente, se presentan una serie de experimentos numéricos para comparar y evaluar la eficiencia de los métodos e ilustrar las aplicaciones de marco computacional.