



Universidad de Valladolid

Facultad de Ciencias

Trabajo de Fin de Grado

Junio 2021

Grado en Matemáticas

**Integración numérica de algunos modelos de
taxis-reacción-difusión relacionados con el
crecimiento de tumores**

Autor: Gabriel Navarrete Manso

Tutora: María Paz Calvo Cabrero

*A mi abuelo, por enseñarme que a través del esfuerzo
un hombre puede lograr todo lo que se proponga.*

Índice

INTRODUCCIÓN	1
1. MÉTODOS RUNGE-KUTTA LINEALMENTE IMPLÍCITOS	3
1.1. DEFINICIÓN DEL PROBLEMA Y MÉTODOS	3
1.2. ORDEN DEL MÉTODO	4
1.2.1. PRIMERAS CONDICIONES DE ORDEN	4
1.2.2. TEORÍA GENERAL	9
1.2.3. OBTENCIÓN DE LAS CONDICIONES DE ORDEN	12
1.3. ESTABILIDAD LINEAL	17
1.4. UN MÉTODO RKLI DE PASO VARIABLE	20
2. MODELOS DE CRECIMIENTO DE TUMORES	23
2.1. ANGIOGÉNESIS	23
2.2. INVASIÓN TUMORAL	25
3. RESOLUCIÓN NUMÉRICA DE MODELOS UNIDIMENSIONALES	26
3.1. ANGIOGÉNESIS	26
3.1.1. DISCRETIZACIÓN ESPACIAL	27
3.1.2. LIMITADORES DE FLUJO	29
3.1.3. RESULTADOS NUMÉRICOS	30
3.2. INVASIÓN TUMORAL	36
3.2.1. DISCRETIZACIÓN ESPACIAL	36
3.2.2. RESULTADOS NUMÉRICOS	39
BIBLIOGRAFÍA	41
A. PROGRAMAS DE MATLAB	42
A.1. MODELO DE ANGIOGÉNESIS	42
A.2. MODELO DE INVASIÓN	49

INTRODUCCIÓN

En este trabajo, dentro del contexto de la simulación numérica de ecuaciones de advección-reacción-difusión, se aborda el estudio de métodos Runge-Kutta linealmente implícitos para la integración temporal de los sistemas diferenciales resultantes tras la discretización espacial de dichas ecuaciones.

En la primera sección se plantean los métodos Runge-Kutta linealmente implícitos, se formaliza la correspondiente teoría general del orden y se lleva a cabo un estudio de su estabilidad lineal proponiendo una ecuación test adecuada. También se presenta el par encajado de métodos que se va a emplear en los experimentos numéricos y se detallan los pasos necesarios para implementar el método con paso variable.

En la segunda sección se describen los procesos biológicos que tienen lugar durante el desarrollo de los tumores en dos contextos distintos. Esto nos permite comprender las ecuaciones empleadas para modelar dichos fenómenos, así como entender e interpretar correctamente los resultados obtenidos, no solo desde el punto de vista matemático.

En la tercera sección se detallan los pormenores de la discretización espacial de las ecuaciones de los modelos, y se muestran las dificultades que han ido apareciendo, junto con las técnicas implementadas para solventarlas. Finalmente, dentro de esta sección se presentan los resultados numéricos obtenidos y se describen las características matemáticas y biológicas de las soluciones halladas.

Por último, en el apéndice se pueden ver los códigos que resuelven cada uno de los problemas planteados, implementados en Matlab.

1. MÉTODOS RUNGE-KUTTA LINEALMENTE IMPLÍCITOS

1.1. DEFINICIÓN DEL PROBLEMA Y MÉTODOS

Consideramos el siguiente sistema de ecuaciones diferenciales ordinarias

$$\begin{cases} \frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}) = L\mathbf{y} + \mathbf{N}(t, \mathbf{y}), \\ \mathbf{y}(t_0) = \mathbf{y}_0, \end{cases} \quad (1)$$

donde $\mathbf{y} : \mathbb{R} \rightarrow \mathbb{R}^d$, $\mathbf{N} : \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ son funciones suficientemente regulares y $L \in \mathcal{M}_{d \times d}$ es una matriz.

En general, encontramos sistemas diferenciales como (1) tras discretizar en espacio ecuaciones en derivadas parciales de advección-reacción-difusión. En ese caso, L se obtiene de la discretización espacial de los términos difusivos (y tendrá, por tanto, autovalores negativos, con radio espectral tendiendo a $+\infty$ con el cuadrado del diámetro de la red espacial) y \mathbf{N} de la de los términos de advección, reacción y fuente.

Para integrar numéricamente el problema se pueden usar los ya conocidos métodos Runge-Kutta. Sin embargo, en este trabajo vamos a estudiar métodos Runge-Kutta linealmente implícitos que, como veremos, aportan algunas ventajas sobre los métodos Runge-Kutta clásicos. Para integrar la parte lineal, es lógico emplear un método implícito que tenga una amplia región de estabilidad; en cambio, para la parte no lineal, usaremos un método explícito que no aumente sobremanera el coste computacional. Siguiendo estos requerimientos y conocida una aproximación \mathbf{y}_n a la solución de (1) en tiempo t_n , las ecuaciones que permiten describir el paso $t_n \rightarrow t_{n+1} = t_n + h$, donde h es la longitud de paso, son

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{y}_n, \\ \mathbf{Y}_i &= \mathbf{y}_n + h \left[\sum_{j=2}^i a_{ij} L \mathbf{Y}_j + \sum_{j=1}^{i-1} \hat{a}_{ij} \mathbf{N}(t_n + hc_j, \mathbf{Y}_j) \right], \quad 2 \leq i \leq s+1, \\ \mathbf{y}_{n+1} &= \mathbf{y}_n + h \left[\sum_{i=2}^{s+1} b_i L \mathbf{Y}_i + \sum_{i=1}^{s+1} \hat{b}_i \mathbf{N}(t_n + hc_i, \mathbf{Y}_i) \right]. \end{aligned} \quad (2)$$

El tablero de Butcher con los coeficientes de este método Runge-Kutta linealmente implícito (RKLI de ahora en adelante) es

0	0					0					
c_2	0	γ				\hat{a}_{21}	0				
c_3	0	a_{32}	γ			\hat{a}_{31}	\hat{a}_{32}	0			
\vdots	\vdots	\vdots	\ddots	\ddots		\vdots	\vdots	\ddots	\ddots		
c_{s+1}	0	$a_{s+1,2}$	\dots	$a_{s+1,s}$	γ	$\hat{a}_{s+1,1}$	$\hat{a}_{s+1,2}$	\dots	$\hat{a}_{s+1,s}$	0	
	0	b_2	\dots	b_s	γ	\hat{b}_1	\hat{b}_2	\dots	\hat{b}_s	\hat{b}_{s+1}	

Cuadro 1: Tablero de Butcher del método RK linealmente implícito.

El tablero izquierdo corresponde a un método RK diagonalmente implícito que, como hemos mencionado antes, se utilizará para el término lineal, mientras que el tablero derecho describe un método RK explícito, el cual usaremos para los términos no lineales.

A lo largo de todo el trabajo vamos a suponer que se satisfacen las condiciones siguientes

$$c_i = \sum_{j=1}^i a_{ij} = \sum_{j=1}^{i-1} \hat{a}_{ij}, \quad 1 \leq i \leq s+1, \quad (3)$$

$$b_i = \hat{b}_i, \quad 1 \leq i \leq s+1. \quad (4)$$

Gracias a estos dos grupos de condiciones vamos a poder simplificar nuestro estudio. Para empezar, al estudiar las condiciones de orden que deben satisfacer estos métodos, gracias a (3) podemos limitarnos a considerar sistemas autónomos

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}) = L\mathbf{y} + \mathbf{N}(\mathbf{y}), \quad (5)$$

aunque se puedan utilizar después para sistemas más generales como (1). Además, estas dos condiciones nos serán de gran utilidad en próximas secciones, donde redactaremos una teoría general del orden.

1.2. ORDEN DEL MÉTODO

En esta sección nos centraremos en el estudio del orden de los métodos RK linealmente implícitos. Comenzaremos con un cálculo directo de las condiciones de orden. Posteriormente, haremos un análisis más profundo de la teoría, donde mostraremos una metodología más eficiente para calcular las mencionadas condiciones. Finalmente, emplearemos el método desarrollado para recuperar las condiciones calculadas en primera instancia, evidenciando la simplicidad en los nuevos cálculos.

1.2.1. PRIMERAS CONDICIONES DE ORDEN

A continuación, vamos a desarrollar en serie de Taylor las soluciones exacta y numérica para así deducir las condiciones de orden de las que venimos hablando.

Podremos comprobar como la dificultad de los cálculos aumenta notablemente con el orden de las derivadas, entendiéndose como necesaria la búsqueda de otro procedimiento para deducir los resultados generales.

Comenzaremos desarrollando la solución exacta

$$\mathbf{y}(t_0 + h) = \mathbf{y}(t_0) + h\mathbf{y}'(t_0) + \frac{h^2}{2}\mathbf{y}''(t_0) + \frac{h^3}{3!}\mathbf{y}'''(t_0) + \frac{h^4}{4!}\mathbf{y}^{(iv)}(t_0) + O(h^5) \quad (6)$$

Cuando calculemos las derivadas respecto de \mathbf{y} , realmente tendremos que derivar parcialmente con respecto a cada componente y_i , pero esto no vamos a escribirlo explícitamente para así aligerar la notación. Tampoco especificaremos la dependencia de \mathbf{f} y de \mathbf{N} respecto de \mathbf{y} , por el mismo motivo.

Empezamos a calcular las derivadas, teniendo en cuenta (5)

$$\begin{aligned} \mathbf{y}'' = (\mathbf{y}')' &= L\mathbf{f} + \mathbf{N}_y\mathbf{f} = \\ &= L^2\mathbf{y} + L\mathbf{N} + \mathbf{N}_yL + \mathbf{N}_y\mathbf{N}. \end{aligned} \quad (7)$$

Calculando la tercera derivada obtenemos

$$\begin{aligned} \mathbf{y}''' = (\mathbf{y}'')' &= L^2\mathbf{f} + \mathbf{N}_{yy}[\mathbf{f}, \mathbf{N}] + \mathbf{N}_y\mathbf{N}_y\mathbf{f} + L\mathbf{N}_y\mathbf{f} + \mathbf{N}_{yy}[\mathbf{f}, L] + \mathbf{N}_yL\mathbf{f} \\ &= L^3\mathbf{y} + L^2\mathbf{N} + 2\mathbf{N}_{yy}[L, \mathbf{N}] + \mathbf{N}_{yy}[\mathbf{N}, \mathbf{N}] + \mathbf{N}_y\mathbf{N}_yL\mathbf{y} + \mathbf{N}_y\mathbf{N}_y\mathbf{N} \\ &\quad + L\mathbf{N}_yL\mathbf{y} + L\mathbf{N}_y\mathbf{N} + \mathbf{N}_{yy}[L\mathbf{y}, L] + \mathbf{N}_yL^2\mathbf{y} + \mathbf{N}_yL\mathbf{N}. \end{aligned}$$

Finalmente, vamos a calcular la derivada cuarta

$$\begin{aligned} \mathbf{y}^{iv} = (\mathbf{y}''')' &= L^3\mathbf{f} + L^2\mathbf{N}_y\mathbf{f} + \mathbf{N}_{yyy}[\mathbf{f}, \mathbf{N}, \mathbf{N}] + 2\mathbf{N}_{yy}[\mathbf{N}_y\mathbf{f}, \mathbf{N}] \\ &\quad + 2\mathbf{N}_{yy}[L\mathbf{f}, \mathbf{N}] + \mathbf{N}_{yyy}[\mathbf{f}, L\mathbf{y}, \mathbf{N}] + \mathbf{N}_{yy}[\mathbf{f}, \mathbf{N}_y\mathbf{N}] \\ &\quad + \mathbf{N}_y\mathbf{N}_{yy}[\mathbf{f}, \mathbf{N}] + \mathbf{N}_y\mathbf{N}_y\mathbf{N}_y\mathbf{f} + \mathbf{N}_{yy}[\mathbf{f}, \mathbf{N}_yL] + \mathbf{N}_y\mathbf{N}_{yy}[\mathbf{f}, L\mathbf{y}] \\ &\quad + \mathbf{N}_y\mathbf{N}_yL\mathbf{f} + L\mathbf{N}_{yy}[\mathbf{f}, L\mathbf{y}] + L\mathbf{N}_yL\mathbf{f} + L\mathbf{N}_{yy}[\mathbf{f}, \mathbf{N}] + L\mathbf{N}_y\mathbf{N}_y\mathbf{f} \\ &\quad + \mathbf{N}_{yyy}[\mathbf{f}, L\mathbf{y}, L\mathbf{y}] + 2\mathbf{N}_{yy}[L\mathbf{f}, L\mathbf{y}] + \mathbf{N}_{yyy}[\mathbf{f}, \mathbf{N}, L\mathbf{y}] \\ &\quad + 2\mathbf{N}_{yy}[\mathbf{N}_y\mathbf{f}, L\mathbf{y}] + \mathbf{N}_{yy}[\mathbf{N}, L\mathbf{f}] + \mathbf{N}_{yy}[\mathbf{f}, L^2\mathbf{y}] + \mathbf{N}_yL^2\mathbf{f} \\ &\quad + \mathbf{N}_{yy}[\mathbf{f}, L\mathbf{N}] + \mathbf{N}_yL\mathbf{N}_y\mathbf{f} \end{aligned}$$

Continuando con este proceso iríamos obteniendo los distintos términos del desarrollo de Taylor de la solución exacta. En la siguiente sección mostraremos una expresión general para este desarrollo.

Ahora tenemos que repetir este procedimiento con la solución numérica. Una vez hechos los dos cálculos, simplemente tendremos que ir igualando los términos de ambos desarrollos que se correspondan a derivadas del mismo orden para deducir las condiciones de orden.

Recordemos que las ecuaciones del método RK linealmente implícito con el que estamos trabajando son las que se especifican en (2).

Haremos el desarrollo de la solución numérica tras el primer paso

$$\mathbf{y}_1 = \mathbf{y}_0 + h \left[\sum_{i=2}^{s+1} b_i L \mathbf{Y}_i + \sum_{i=1}^{s+1} \hat{b}_i \mathbf{N}(\mathbf{Y}_i) \right].$$

Desarrollando en serie de Taylor

$$\mathbf{y}_1 = \mathbf{y}_0 + h \frac{d\mathbf{y}_1}{dh} \Big|_{h=0} + \frac{h^2}{2} \frac{d^2\mathbf{y}_1}{dh^2} \Big|_{h=0} + \frac{h^3}{3!} \frac{d^3\mathbf{y}_1}{dh^3} \Big|_{h=0} + \frac{h^4}{4!} \frac{d^4\mathbf{y}_1}{dh^4} \Big|_{h=0} + O(h^5).$$

Vamos a calcular los diferentes términos de esta expresión teniendo en cuenta, ahora sí, (3) y (4), pues de esta forma los resultados se simplificarán bastante. Lo primero que tenemos, sin más que fijarnos en la forma del método que estamos manejando, es que

$$\mathbf{Y}_i \Big|_{h=0} = \mathbf{y}_0.$$

Ahora calculamos la primera derivada

$$\begin{aligned} \frac{d\mathbf{y}_1}{dh} &= \sum_{i=2}^{s+1} b_i L \mathbf{Y}_i + \sum_{i=2}^{s+1} b_i \mathbf{N}(\mathbf{Y}_i) + h \left[\sum_{i=2}^{s+1} b_i L \frac{d\mathbf{Y}_i}{dh} + \sum_{i=2}^{s+1} b_i \mathbf{N}_y(\mathbf{Y}_i) \frac{d\mathbf{Y}_i}{dh} \right] \\ &\stackrel{h=0}{=} \sum_{i=2}^{s+1} b_i L \mathbf{y}_0 + \sum_{i=2}^{s+1} b_i \mathbf{N}(\mathbf{y}_0). \end{aligned}$$

Si comparamos esta expresión con la de la primera derivada de la solución exacta e imponemos que sean iguales, obtenemos la condición de orden 1, que es

$$\sum_{i=2}^{s+1} b_i = 1. \quad (8)$$

Veamos cual es la expresión de la derivada segunda de la solución numérica

$$\begin{aligned} \frac{d^2\mathbf{y}_1}{dh^2} &= 2 \sum_{i=2}^{s+1} b_i L \frac{d\mathbf{Y}_i}{dh} + 2 \sum_{i=2}^{s+1} b_i \mathbf{N}_y(\mathbf{Y}_i) \frac{d\mathbf{Y}_i}{dh} \\ &\quad + h \left[2 \sum_{i=2}^{s+1} b_i L \frac{d^2\mathbf{Y}_i}{dh^2} + \sum_{i=2}^{s+1} b_i \left(\mathbf{N}_{yy}(\mathbf{Y}_i) \left[\frac{d\mathbf{Y}_i}{dh}, \frac{d\mathbf{Y}_i}{dh} \right] + \mathbf{N}_y(\mathbf{Y}_i) \frac{d^2\mathbf{Y}_i}{dh^2} \right) \right] \\ &\stackrel{h=0}{=} 2 \sum_{i=2}^{s+1} b_i L \frac{d\mathbf{Y}_i}{dh} \Big|_{h=0} + 2 \sum_{i=2}^{s+1} b_i \mathbf{N}_y(\mathbf{y}_0) \frac{d\mathbf{Y}_i}{dh} \Big|_{h=0}. \end{aligned}$$

Para tener esta expresión completamente desarrollada, hemos de calcular las derivadas de los \mathbf{Y}_i

$$\begin{aligned}
\frac{d\mathbf{Y}_i}{dh} &= \sum_{j=2}^i a_{ij} L \mathbf{Y}_j + \sum_{j=1}^{i-1} \hat{a}_{ij} \mathbf{N}(\mathbf{Y}_j) \\
&\quad + h \left[\sum_{j=2}^i a_{ij} L \frac{d\mathbf{Y}_j}{dh} + \sum_{j=1}^{i-1} \hat{a}_{ij} \mathbf{N}_y(\mathbf{Y}_j) \frac{d\mathbf{Y}_j}{dh} \right] \\
&\stackrel{h=0}{=} \sum_{j=2}^i a_{ij} L \mathbf{y}_0 + \sum_{j=1}^{i-1} \hat{a}_{ij} \mathbf{N}(\mathbf{y}_0) = c_i L \mathbf{y}_0 + c_i \mathbf{N}(\mathbf{y}_0), \quad (9)
\end{aligned}$$

donde hemos utilizado (3). Una vez hecho este cálculo, ya tenemos la expresión completa de la derivada segunda

$$\begin{aligned}
\left. \frac{d^2 \mathbf{y}_1}{dh^2} \right|_{h=0} &= 2 \sum_{i=2}^{s+1} b_i c_i L^2 \mathbf{y}_0 + 2 \sum_{i=2}^{s+1} b_i c_i L \mathbf{N}(\mathbf{y}_0) \\
&\quad + 2 \sum_{i=2}^{s+1} b_i c_i L \mathbf{y}_0 \mathbf{N}_y(\mathbf{y}_0) + 2 \sum_{i=2}^{s+1} b_i c_i \mathbf{N}(\mathbf{y}_0) \mathbf{N}_y(\mathbf{y}_0). \quad (10)
\end{aligned}$$

Si comparamos esta expresión con la que obtuvimos para la solución exacta (7) e imponemos que sean iguales, entonces deducimos la condición de orden 2

$$\sum_{i=2}^{s+1} b_i c_i = \frac{1}{2}. \quad (11)$$

Notemos que, aunque en (7) y en (10) aparecen 4 sumandos bien diferenciados, el coeficiente que aparece en todos ellos es el mismo, dando lugar a una única condición de orden 2.

Por último, haremos el cálculo de la derivada tercera. Aquí aparecerán condiciones cruzadas, con coeficientes pertenecientes a ambos tableros de (1)

$$\begin{aligned}
\frac{d^3 \mathbf{y}_1}{dh^3} &= 3 \sum_{i=2}^{s+1} b_i L \frac{d^2 \mathbf{Y}_i}{dh^2} + 3 \sum_{i=2}^{s+1} b_i \left(\mathbf{N}_{yy}(\mathbf{Y}_i) \left[\frac{d\mathbf{Y}_i}{dh}, \frac{d\mathbf{Y}_i}{dh} \right] + \mathbf{N}_y(\mathbf{Y}_i) \frac{d^2 \mathbf{Y}_i}{dh^2} \right) \\
&\quad + h \left[\sum_{i=2}^{s+1} b_i L \frac{d^3 \mathbf{Y}_i}{dh^3} + \sum_{i=2}^{s+1} b_i \left(\mathbf{N}_{yyy}(\mathbf{Y}_i) \left[\frac{d\mathbf{Y}_i}{dh}, \frac{d\mathbf{Y}_i}{dh}, \frac{d\mathbf{Y}_i}{dh} \right] \right. \right. \\
&\quad \left. \left. + 3 \mathbf{N}_{yy}(\mathbf{Y}_i) \left[\frac{d\mathbf{Y}_i}{dh}, \frac{d^2 \mathbf{Y}_i}{dh^2} \right] + \mathbf{N}_y(\mathbf{Y}_i) \frac{d^3 \mathbf{Y}_i}{dh^3} \right) \right].
\end{aligned}$$

Como se va a evaluar en $h = 0$, lo que hay dentro del corchete no nos interesa, pero aun así necesitamos conocer la derivada segunda de los \mathbf{Y}_i .

$$\begin{aligned} \frac{d^2 \mathbf{Y}_i}{dh^2} &= 2 \sum_{j=2}^i a_{ij} L \frac{d\mathbf{Y}_j}{dh} + 2 \sum_{j=1}^{i-1} \hat{a}_{ij} \mathbf{N}_y(\mathbf{Y}_j) \frac{d\mathbf{Y}_j}{dh} + h \left[\sum_{j=2}^i a_{ij} L \frac{d^2 \mathbf{Y}_j}{dh^2} \right. \\ &\quad \left. + \sum_{j=1}^{i-1} \hat{a}_{ij} \left(\mathbf{N}_{yy}(\mathbf{Y}_j) \left[\frac{d\mathbf{Y}_j}{dh}, \frac{d\mathbf{Y}_j}{dh} \right] + \mathbf{N}_y(\mathbf{Y}_j) \frac{d^2 \mathbf{Y}_j}{dh^2} \right) \right]. \end{aligned}$$

Evaluando en $h = 0$ y utilizando (9) obtenemos

$$\begin{aligned} \frac{d^2 \mathbf{Y}_i}{dh^2} \Big|_{h=0} &= 2 \sum_{j=2}^i a_{ij} L \left(\sum_{k=2}^j a_{jk} L \mathbf{y}_0 + \sum_{k=1}^{j-1} \hat{a}_{jk} \mathbf{N}(\mathbf{y}_0) \right) \\ &\quad + 2 \sum_{j=1}^{i-1} \hat{a}_{ij} \mathbf{N}_y(\mathbf{y}_0) \left(\sum_{k=2}^j a_{jk} L \mathbf{y}_0 + \sum_{k=1}^{j-1} \hat{a}_{jk} \mathbf{N}(\mathbf{y}_0) \right) \\ &= 2 \sum_{j=2}^i a_{ij} c_j L^2 \mathbf{y}_0 + 2 \sum_{j=2}^i a_{ij} c_j L \mathbf{N}(\mathbf{y}_0) \\ &\quad + 2 \sum_{j=2}^{i-1} \hat{a}_{ij} c_j L \mathbf{y}_0 \mathbf{N}_y(\mathbf{y}_0) + 2 \sum_{j=2}^{i-1} \hat{a}_{ij} c_j \mathbf{N}(\mathbf{y}_0) \mathbf{N}_y(\mathbf{y}_0). \end{aligned}$$

Llevando todo esto a la expresión de la derivada tercera de \mathbf{y}_1 se llega a

$$\begin{aligned} \frac{d^3 \mathbf{y}_1}{dh^3} \Big|_{h=0} &= 3 \sum_{i=2}^{s+1} b_i L \left(2 \sum_{j=2}^i a_{ij} c_j L^2 \mathbf{y}_0 + 2 \sum_{j=2}^i a_{ij} c_j L \mathbf{N}(\mathbf{y}_0) \right. \\ &\quad \left. + 2 \sum_{j=2}^{i-1} \hat{a}_{ij} c_j L \mathbf{y}_0 \mathbf{N}_y(\mathbf{y}_0) + 2 \sum_{j=2}^{i-1} \hat{a}_{ij} c_j \mathbf{N}(\mathbf{y}_0) \mathbf{N}_y(\mathbf{y}_0) \right) \\ &\quad + 3 \sum_{i=2}^{s+1} b_i \left[\mathbf{N}_{yy}(\mathbf{y}_0) (c_i L \mathbf{y}_0 + c_i \mathbf{N}(\mathbf{y}_0))^2 \right. \\ &\quad \left. + \mathbf{N}_y(\mathbf{Y}_i) \left(2 \sum_{j=2}^i a_{ij} c_j L^2 \mathbf{y}_0 + 2 \sum_{j=2}^i a_{ij} c_j L \mathbf{N}(\mathbf{y}_0) \right) \right. \\ &\quad \left. + 2 \sum_{j=2}^{i-1} \hat{a}_{ij} c_j L \mathbf{y}_0 \mathbf{N}_y(\mathbf{y}_0) + 2 \sum_{j=2}^{i-1} \hat{a}_{ij} c_j \mathbf{N}(\mathbf{y}_0) \mathbf{N}_y(\mathbf{y}_0) \right) \Big]. \end{aligned}$$

Si desarrollamos los paréntesis, comparamos con la expresión de la derivada tercera de la solución exacta e imponemos que ambos desarrollos coincidan, obtenemos 3 condiciones que deben satisfacer los coeficientes del método

$$\begin{aligned} \sum_{i=2}^{s+1} \sum_{j=2}^i b_i a_{ij} c_j &= \frac{1}{6}, \\ \sum_{i=2}^{s+1} \sum_{j=1}^{i-1} b_i \hat{a}_{ij} c_j &= \frac{1}{6}, \\ \sum_{i=1}^{s+1} b_i c_i^2 &= \frac{1}{3}. \end{aligned}$$

Después de haber hecho estos cálculos, queda patente la necesidad de emplear alguna herramienta que nos permita obtener estos resultados de una manera más sistemática.

1.2.2. TEORÍA GENERAL

El objetivo de esta sección es redactar una expresión general para las derivadas de las soluciones de (5), tanto exacta como numérica, en términos de las llamadas diferenciales elementales. Así, tendremos la forma de deducir las condiciones de orden para cada caso sin necesidad de realizar todos los cálculos mostrados en las páginas anteriores. A la hora de exponer esta teoría, vamos a considerar el caso de una dimensión, para facilitar la notación, pero los resultados son válidos para problemas en \mathbb{R}^d .

El problema con el que estamos trabajando es, particularizado a una dimensión, el siguiente

$$\frac{dy}{dt} = f(y) = Ly + N(y). \quad (12)$$

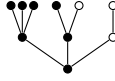
Calcular las derivadas de la solución exacta y de la numérica para obtener los desarrollos de Taylor es un trabajo arduo, como hemos podido comprobar anteriormente. Para simplificar este procedimiento introduciremos una familia adecuada de árboles con raíz que permitan una representación más compacta de las derivadas sucesivas de y . Más concretamente, puesto que $f(y)$ tiene dos partes bien diferenciadas (una lineal, Ly , y otra no lineal, $N(y)$), trabajaremos con árboles bicolor con raíz; el color blanco irá asociado a la parte lineal, Ly , mientras que el negro irá asociado a la parte no lineal, $N(y)$. Al conjunto de los árboles bicolor con raíz de orden l (con l nodos) lo denotaremos por \mathcal{T}_l , y al conjunto de árboles bicolor de cualquier orden por \mathcal{T} . Vamos a dar una definición general

Definición 1. *Un árbol bicolor con raíz de orden l es un grafo conexo y sin ciclos con l vértices o nodos, que pueden ser blancos o negros, y un cierto número de lados conectando cada vértice con un único vértice del nivel inferior, de tal forma que en el primer nivel hay un único nodo que se denomina raíz.*

Al árbol con un único vértice blanco lo denotaremos por τ_t y al árbol con un único nodo negro por τ_r , la letra t la usaremos para referirnos a árboles con raíz blanca y la letra r para los árboles con raíz negra.

Cualquier elemento de \mathcal{T} se puede representar dibujando los vértices y los lados que los unen, como veremos en un ejemplo a continuación; también se pueden representar con la notación recurrente $v = [t_1, \dots, t_n, r_1, \dots, r_m]_\xi$, cuya interpretación es la siguiente: t_1, \dots, t_n son n elementos de \mathcal{T} con raíz blanca, r_1, \dots, r_m son m elementos de \mathcal{T} con raíz negra; la raíz de v es blanca si $\xi = t$ o negra si $\xi = r$, y de ella parten $n + m$ ramas que la unen con las n raíces (blancas) de los árboles t_1, \dots, t_n y con las m raíces (negras) de los árboles r_1, \dots, r_m .

Veámoslo con un ejemplo.



Cuadro 2: Ejemplo de árbol bicolor con raíz.

Este árbol lo podemos representar como $u = [r_1, r_2, t_3]_r$, donde r_1 y r_2 indican que la primera y segunda rama empiezan con un nodo negro y t_3 que la tercera lo hace con uno blanco. El subíndice r del corchete indica que la raíz del árbol es de color negro. Esta descomposición se puede continuar, hasta que se deja todo expresado en función de los árboles con un único nodo, τ_r y τ_t . De esta forma, el árbol del ejemplo que hemos puesto lo podemos escribir como $u = [[\tau_r, \tau_r, \tau_r]_r, [\tau_r, \tau_t]_r, [\tau_t]_t]_r$.

Veamos como podemos emplear esto para expresar los desarrollos de Taylor. Vamos a introducir el concepto de diferencial elemental.

Definición 2. La diferencial elemental para el problema (12) asociada a un árbol con raíz $v = [t_1, \dots, t_n, r_1, \dots, r_m]_\xi$ se denota por $F(v)(y)$ y se define de forma recursiva como

$$\begin{aligned} F(\emptyset)(y) &= y, \\ F(\tau_t)(y) &= Ly, \\ F(\tau_r)(y) &= N(y), \\ F(v)(y) &= N_{i_1, \dots, i_n, j_1, \dots, j_m} [F(t_1)(y), \dots, F(r_1)(y), \dots], \quad \text{si } \xi = r, \\ F(v)(y) &= L \cdot F(t_1)(y), \quad \text{si } \xi = t, n = 1 \text{ y } m = 0, \\ F(v)(y) &= L \cdot F(r_1)(y), \quad \text{si } \xi = t, n = 0 \text{ y } m = 1, \\ F(v)(y) &= 0, \quad \text{si } \xi = t \text{ y } m + n > 1. \end{aligned}$$

Con esta definición podemos construir la diferencial elemental asociada al árbol del ejemplo. Cada rama que sale de la raíz implica una derivada, por tanto, las tres ramas que salen de la raíz de color negro se traducen en N_{yyy} . Actuando de la misma manera con el resto de ramas del árbol, tenemos lo siguiente

$$F(u)(y) = N_{yyy} [L^2y, N_{yy} [Ly, N], N_{yyy} [N, N, N]].$$

A la vista de la definición anterior y del ejemplo, es claro que solo aportan diferenciales elementales no nulas los árboles bicolor en los que cada nodo blanco tiene a lo sumo una rama que parte de él, ya que las derivadas de orden mayor que uno del término Ly son idénticamente nulas. Por ello denotamos por \mathcal{T}^* al subconjunto de \mathcal{T} formado por los árboles con esta propiedad, y por \mathcal{T}_l^* a los que, además de cumplir la propiedad, tienen l nodos.

Una vez hecho todo este desarrollo, vamos a mostrar dos resultados fundamentales cuya demostración excede los objetivos de este trabajo. Un estudio más exhaustivo se realiza en el artículo [1].

Teorema 1. *Para el problema (1), la expresión general del desarrollo de Taylor de la solución exacta es la siguiente*

$$y(t_0 + h) = y_0 + \sum_{k=1}^{\infty} \frac{h^k}{k!} \left(\sum_{v \in \mathcal{T}_k^*} \alpha(v) F(v)(y_0) \right),$$

donde $\alpha(v)$ denota el número de etiquetados monótonos del árbol v .

Aquí conviene mencionar la correspondencia que existe entre derivar una diferencial elemental general y añadir una nueva rama al árbol asociado a dicha diferencial elemental partiendo de alguno de los nodos de dicho árbol. De este modo, de cada árbol de orden l obtendremos varios árboles de orden $l + 1$, que refleja el hecho de que la derivada de una diferencial elemental de orden l es una combinación lineal de diferenciales elementales de orden $l + 1$.

Antes de enunciar el siguiente teorema, vamos a introducir dos nuevas funciones. Siendo $\rho(v)$ el orden del árbol v , definimos

Definición 3. *La función densidad asociada a un árbol con raíz*

$$v = [t_1, \dots, t_n, r_1, \dots, r_m]_{\xi}$$

se denota por $\gamma(v)$ y se define de forma recursiva como

$$\gamma(v) = \rho(v) \cdot \gamma(t_1) \cdots \gamma(t_n) \cdot \gamma(r_1) \cdots \gamma(r_m).$$

Notemos que la función densidad no depende del color de la raíz del árbol.

Definición 4. *La función peso elemental asociada a un árbol con raíz $v = [t_1, \dots, t_n, r_1, \dots, r_m]_{\xi}$ se denota por $\Phi(v)$ y se define de forma recursiva como*

$$\begin{aligned} \Phi(v) &= \sum_{i=1}^{s+1} b_i \Phi_i(v), \quad \text{si } \xi = t; & \Phi(v) &= \sum_{i=1}^{s+1} \hat{b}_i \Phi_i(v), \quad \text{si } \xi = r, \\ \Phi_i(v) &= \sum_{\substack{j_1, \dots, j_n=1 \\ l_1, \dots, l_m=1}}^{s+1} a_{ij_1} \Phi_{j_1}(t_1) \cdots a_{ij_n} \Phi_{j_n}(t_n) \cdot \hat{a}_{il_1} \Phi_{l_1}(r_1) \cdots \hat{a}_{il_m} \Phi_{l_m}(r_m), \\ \Phi_i(\tau_r) &= 1, \\ \Phi_i(\tau_t) &= 1. \end{aligned}$$

A la vista de estas definiciones, podemos calcular las expresiones de $\gamma(v)$ y $\Phi(v)$ para el árbol del ejemplo (ver Cuadro 2)

$$\begin{aligned}\gamma(u) &= 10 \cdot 4 \cdot 3 \cdot 2 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 240, \\ \Phi(u) &= \sum_{i=1}^{s+1} \hat{b}_i \sum_{j_1, j_2, j_3=1}^{s+1} \hat{a}_{ij_3} c_{j_3}^3 \hat{a}_{ij_2} c_{j_2}^2 a_{ij_1} c_{j_1}.\end{aligned}$$

Teorema 2. *La expresión general del desarrollo de Taylor de la solución numérica del problema (1) tras aplicarle el método Runge-Kutta linealmente implícito (2) es la siguiente*

$$y_1 = y_0 + \sum_{k=1}^{\infty} \frac{h^k}{k!} \left(\sum_{v \in \mathcal{T}_k^*} \alpha(v) [\Phi(v) \cdot \gamma(v)] F(v)(y_0) \right),$$

donde $\alpha(v)$ y $F(v)(y_0)$ son como en el Teorema 1, y $\gamma(v)$ y $\Phi(v)$ están dados por las definiciones anteriores.

Por tanto, comparando los desarrollos de Taylor de las soluciones exacta y numérica podemos deducir las condiciones de orden, que enunciamos ahora

Teorema 3. *Un método Runge-Kutta linealmente implícito es de orden k si y solo si los desarrollos de Taylor de las soluciones exacta y numérica coinciden hasta los términos de orden k , o equivalentemente, si y solo si*

$$\Phi(v) = \frac{1}{\gamma(v)} \quad \text{para todo } v \in \bigcup_{j=1}^k \mathcal{T}_j^* \quad (13)$$

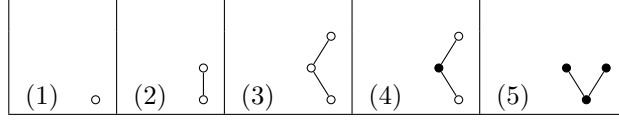
Así pues, hemos podido comprobar que gracias a los árboles podemos hallar las condiciones de orden sin necesidad de realizar todos los cálculos de la anterior subsección.

En este momento es necesario mencionar la importancia de las condiciones (3) y (4) en lo que viene a continuación. Según dichas condiciones, dos árboles que solo difieran en el color de la raíz, así como dos árboles que solo difieran en el color de los nodos terminales darán lugar a las mismas condiciones de orden (13). Esto implica que el número de árboles de cada orden que tendremos que considerar será notablemente menor, simplificando el procedimiento.

1.2.3. OBTENCIÓN DE LAS CONDICIONES DE ORDEN

En este apartado vamos a poner en práctica lo expuesto en la sección de teoría general. Calcularemos las condiciones de orden 1, 2 y 3 a partir de la teoría de los árboles bicolor con raíz para recuperar las expresiones que obtuvimos previamente. Finalmente, deduciremos las condiciones de orden 4.

Los árboles bicolor con raíz de \mathcal{T}^* de orden 1, 2 y 3 que generan condiciones de orden independientes son los que se muestran en la siguiente tabla



Cuadro 3: Árboles de orden 1, 2 y 3

Como se ha comentado antes, en situación normal, aquí aparecerían más árboles, pero estamos aplicando ya las condiciones (3) y (4), por lo que el estudio se reduce a los que vemos en la tabla.

Comenzamos por el árbol (1), que no es otro que τ_t . Observamos que $\gamma(u) = 1$ y que la función $\Phi(u)$ es

$$\Phi(u) = \sum_{i=1}^{s+1} b_i \Phi_i(\tau_t) = \sum_{i=1}^{s+1} b_i,$$

luego la condición (13) de orden 1 es la que obtuvimos en (8)

Prosigamos con el árbol (2), $u = [\tau_t]_t$. En este caso $\gamma(u) = 2 \cdot 1 = 2$, por lo que ha de cumplirse

$$\Phi(u) = \frac{1}{2}.$$

Para la función peso, $\Phi(u)$, tenemos

$$\begin{aligned} \Phi(u) &= \sum_{i=1}^{s+1} b_i \Phi_i([\tau_t]_t), \\ \Phi_i([\tau_t]_t) &= \sum_{j=1}^{s+1} a_{ij} \Phi_j(\tau_t) = c_i, \end{aligned}$$

donde se ha utilizado la hipótesis simplificadora (4). La condición de orden 2

$$\sum_{i=1}^{s+1} b_i c_i = \frac{1}{2},$$

es la obtenida en (11).

El árbol (3), $u = [[\tau_t]_t]_t$, cumple que $\gamma(u) = 3 \cdot 2 \cdot 1 = 6$, y su función peso $\Phi(u)$ es la siguiente

$$\begin{aligned}\Phi(u) &= \sum_{i=1}^{s+1} b_i \Phi_i([\tau_t]_t), \\ \Phi_i([\tau_t]_t) &= \sum_{j=1}^{s+1} a_{ij} \Phi_j([\tau_t]_t), \\ \Phi_j([\tau_t]_t) &= \sum_{k=1}^{s+1} a_{jk} \Phi_k(\tau_t) = c_j.\end{aligned}$$

Sustituyendo regresivamente obtenemos la primera condición de orden 3

$$\sum_{i,j=1}^{s+1} b_i a_{ij} c_j = \frac{1}{6}.$$

Para el árbol (4), $u = [[\tau_t]_r]_t$, la función densidad es la misma (no depende del color de los nodos) y en el peso elemental el coeficiente a_{ij} , que es el asociado al único nodo negro, tiene que ser reemplazado por \hat{a}_{ij} al estar asociado al método explícito con el que se tratan los términos no lineales. La segunda condición de orden 3 es entonces

$$\sum_{i,j=1}^{s+1} b_i \hat{a}_{ij} c_j = \frac{1}{6}.$$

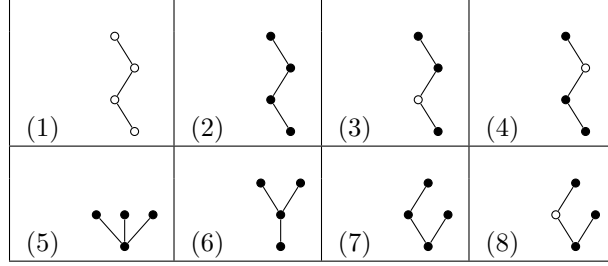
Vamos a deducir la última condición de orden 3, la del árbol $u = [\tau_r, \tau_r]_r$, el cual cumple que $\gamma(u) = 3 \cdot 1 \cdot 1 = 3$. Su peso elemental está dado por

$$\begin{aligned}\Phi(u) &= \sum_{i=1}^{s+1} b_i \Phi_i([\tau_r, \tau_r]_r), \\ \Phi_i([\tau_r, \tau_r]_r) &= \sum_{j_1, j_2=1}^{s+1} \hat{a}_{ij_1} \Phi_{j_1}(\tau_r) \hat{a}_{ij_2} \Phi_{j_2}(\tau_r) = c_i^2,\end{aligned}$$

y la tercera condición de orden 3 es

$$\sum_{i=1}^{s+1} b_i c_i^2 = \frac{1}{3}.$$

Nos disponemos a calcular ahora las condiciones de orden 4. Tras eliminar los árboles de \mathcal{T}_4^* que solo difieren en el color de la raíz o de los nodos terminales, los árboles bicolor con raíz de orden 4 que hay que considerar son los que aparecen en la siguiente tabla



Cuadro 4: Árboles de orden 4

Comenzamos con el primer árbol, $u = [[[\tau_t]_t]_t]_t$. Tenemos que $\gamma(u) = 4 \cdot 3 \cdot 2 \cdot 1 = 24$. Por tanto, la condición que se ha de cumplir es

$$\Phi(u) = \frac{1}{24}.$$

Vamos a calcular $\Phi(u)$ de forma recursiva

$$\begin{aligned} \Phi(u) &= \sum_{i=1}^{s+1} b_i \Phi_i([\tau_t]_t), \\ \Phi_i([\tau_t]_t) &= \sum_{j=1}^{s+1} a_{ij} \Phi_j([\tau_t]_t), \\ \Phi_j([\tau_t]_t) &= \sum_{k=1}^{s+1} a_{jk} \Phi_k([\tau_t]_t), \\ \Phi_k([\tau_t]_t) &= \sum_{l=1}^{s+1} a_{kl} \Phi_l(\tau_t) = c_k. \end{aligned}$$

Ahora no tenemos más que sustituir regresivamente para llegar a la condición

$$\Phi([\tau_t]_t) = \sum_{i,j,k=1}^{s+1} b_i a_{ij} a_{jk} c_k = \frac{1}{24}.$$

Para calcular las condiciones asociadas a los árboles (2), (3) y (4), no tenemos más que repetir el mismo procedimiento, con la diferencia de que cuando los nodos sean negros, hay que emplear los coeficientes del método explícito, \hat{a}_{ij} , en vez de los del implícito, a_{ij} . No obstante, en el caso de la raíz y de los nodos finales, aunque sean de color negro, por (3) y (4) podemos seguir utilizando los coeficientes b_i y c_i . Así pues, las condiciones correspondientes a los árboles (2), (3) y (4) son, respectivamente

$$\begin{aligned}\Phi(\llbracket[\tau_r]_r\rrbracket_r) &= \sum_{i,j,k=1}^{s+1} b_i \hat{a}_{ij} \hat{a}_{jk} c_k = \frac{1}{24}, \\ \Phi(\llbracket[\tau_r]_t\rrbracket_r) &= \sum_{i,j,k=1}^{s+1} b_i a_{ij} \hat{a}_{jk} c_k = \frac{1}{24}, \\ \Phi(\llbracket[\tau_r]_t\rrbracket_t) &= \sum_{i,j,k=1}^{s+1} b_i \hat{a}_{ij} a_{jk} c_k = \frac{1}{24}.\end{aligned}$$

El cálculo de la condición de orden asociada al árbol número (5), $u = [\tau_r, \tau_r, \tau_r]_r$, es ligeramente distinto, pues no está asociado a ningún árbol con la misma estructura pero con distinto color en los nodos. En este caso se tiene

$$\begin{aligned}\gamma(u) &= 4 \cdot 1 \cdot 1 \cdot 1 = 4, \\ \Phi(u) &= \sum_{i=1}^{s+1} b_i \Phi_i([\tau_r, \tau_r, \tau_r]_r), \\ \Phi_i([\tau_r, \tau_r, \tau_r]_r) &= \sum_{j_1, j_2, j_3=1}^{s+1} \hat{a}_{ij_1} \Phi_{j_1}(\tau_r) \hat{a}_{ij_2} \Phi_{j_2}(\tau_r) \hat{a}_{ij_3} \Phi_{j_3}(\tau_r) = c_i^3,\end{aligned}$$

donde, una vez más, hemos empleado (4). Sustituyendo recursivamente obtenemos

$$\Phi(u) = \sum_{i=1}^{s+1} b_i c_i^3 = \frac{1}{4}.$$

Veamos la condición asociada al árbol (6), $u = \llbracket[\tau_r, \tau_r]_r\rrbracket_r$. Se tiene que

$$\begin{aligned}\gamma(u) &= 4 \cdot 3 \cdot 1 \cdot 1 = 12, \\ \Phi(u) &= \sum_{i=1}^{s+1} b_i \Phi_i(\llbracket[\tau_r, \tau_r]_r\rrbracket_r), \\ \Phi_i(\llbracket[\tau_r, \tau_r]_r\rrbracket_r) &= \sum_{j=1}^{s+1} \hat{a}_{ij} \Phi_j([\tau_r, \tau_r]_r), \\ \Phi_j([\tau_r, \tau_r]_r) &= \sum_{k_1, k_2=1}^{s+1} \hat{a}_{jk_1} \Phi_{k_1}(\tau_r) \hat{a}_{jk_2} \Phi_{k_2}(\tau_r) = c_j^2.\end{aligned}$$

por lo que la condición de orden 4 asociada a este árbol es

$$\Phi(u) = \sum_{i=1}^{s+1} b_i \hat{a}_{ij} c_j^2 = \frac{1}{12}.$$

Vamos a calcular, por último, la condición asociada al árbol (7), ya que los cálculos para el árbol (8) son idénticos al diferir ambos árboles únicamente en el color de un nodo. Para (7) tenemos

$$\begin{aligned}\gamma(u) &= 4 \cdot 2 \cdot 1 \cdot 1 = 8, \\ \Phi(u) &= \sum_{i=1}^{s+1} b_i \Phi_i([\tau_r]_r, \tau_r)_r, \\ \Phi_i([\tau_r]_r, \tau_r)_r &= \sum_{j_1, j_2=1}^{s+1} \hat{a}_{ij_1} \Phi_{j_1}([\tau_r]_r) \hat{a}_{ij_2} \Phi_{j_2}(\tau_r), \\ \Phi_{j_1}(u_{j_1}) &= \sum_{k=1}^{s+1} \hat{a}_{j_1 k} \Phi_k(\tau_r) = c_{j_1},\end{aligned}$$

que llevan a la condición

$$\Phi(u) = \sum_{i=1}^{s+1} b_i c_i \hat{a}_{ij} c_j = \frac{1}{8}.$$

Para el último árbol, (8), se tiene la condición de orden análoga

$$\Phi([\tau_r]_t, \tau_r)_r = \sum_{i=1}^{s+1} b_i c_i a_{ij} c_j = \frac{1}{8}.$$

Se ha podido observar con claridad la utilidad de los árboles a la hora de calcular las condiciones de orden. Sin apenas dificultad se deducen de forma recursiva todas ellas, sin necesidad de realizar derivadas ni de manejar expresiones largas.

1.3. ESTABILIDAD LINEAL

Una vez elaborada la teoría del orden para métodos Runge-Kutta linealmente implícitos (2), estudiamos en esta sección sus propiedades de estabilidad. Para hacer este estudio vamos a considerar como modelo la ecuación lineal de advección-difusión siguiente

$$u_t = \nu u_{xx} + a u_x, \quad \nu, a > 0, \quad (14)$$

con condiciones periódicas en $[0, 2\pi]$, puesto que estamos interesados en utilizar los métodos RKLI para integrar las ecuaciones resultantes tras la discretización espacial de ecuaciones de advección-reacción-difusión.

Si para (14) consideramos los modos de Fourier

$$\phi_k(x) = e^{ikx}, \quad k \in \mathbb{Z},$$

la condición inicial $u(x, 0) = \sum_{k \in \mathbb{Z}} \alpha_k \phi_k(x)$, y buscamos soluciones de la forma

$$u(x, t) = \sum_{k \in \mathbb{Z}} \alpha_k y_k(t) e^{ikx}, \quad (15)$$

llevando (15) a la ecuación (14) obtenemos que para cada $k \in \mathbb{Z}$, $y_k(t)$ tiene que ser la solución de la ecuación diferencial

$$y'_k(t) = (-\nu k^2 + iak)y_k(t),$$

que es de la forma

$$y' = \lambda y + i\mu y, \quad (16)$$

donde $\lambda = -(\nu/a^2)\mu^2$ y $\mu/a = k \in \mathbb{Z}$. La ecuación (16) es para los métodos Runge-Kutta linealmente implícitos el análogo a la ecuación test escalar $y' = \lambda y$ que se utiliza habitualmente para estudiar la estabilidad lineal de los métodos Runge-Kutta.

A continuación probamos un resultado que permite escribir el resultado de aplicar un método RKLI a (16) para determinar la región de estabilidad del método.

Teorema 4. *El método Runge-Kutta linealmente implícito (1) aplicado a (16) con paso h resulta en*

$$y_{n+1} = R(z, iw)y_n,$$

donde $R(z, iw)$ define la función de estabilidad del método, cuya expresión es

$$R(z, iw) = \frac{\det(\text{Id} - zA - iw\hat{A} + (z + iw)\mathbf{b}^T)}{\det(\text{Id} - zA - iw\hat{A})}.$$

Demostración. Para demostrar este teorema, vamos a aplicar al problema (16) el método RK linealmente implícito con el que venimos trabajando hasta ahora

$$\begin{aligned} Y_1 &= y_n, \\ Y_k &= y_n + h \left(\sum_{j=1}^k a_{kj} \lambda Y_j + i \sum_{j=1}^{k-1} \hat{a}_{kj} \mu Y_j \right), \quad 2 \leq k \leq s, \\ y_{n+1} &= y_n + h \left[\sum_{k=1}^{s+1} b_k (\lambda + i\mu) Y_k \right]. \end{aligned}$$

Llevando al lado izquierdo de la ecuación el término asociado a a_{kk} , logramos tener un sistema lineal cuyo lado derecho se puede calcular de manera explícita

$$(1 - h\lambda a_{kk})Y_k = y_n + h \sum_{j=1}^{k-1} (\lambda a_{kj} + i\mu \hat{a}_{kj}) Y_j.$$

Finalmente, despejamos y_n de la ecuación, obteniendo

$$(1 - h\lambda a_{kk})Y_k - h \sum_{j=1}^{k-1} (\lambda a_{kj} + i\mu \hat{a}_{kj}) Y_j = y_n.$$

Denotamos ahora $z = h\lambda$ y $w = h\mu$. El sistema que tenemos, escrito en forma matricial, es el siguiente

$$\left(Id - zA - iw\hat{A} \right) \mathbf{Y} = y_n \mathbf{e}, \quad (17)$$

donde \mathbf{e} denota el vector con todos sus elementos iguales a 1.

Ahora vamos a añadir al sistema la ecuación que define el nuevo paso

$$y_{n+1} - (z + iw) \sum_{k=1}^{s+1} b_k Y_k = y_n. \quad (18)$$

Definimos las siguientes matrices:

$$H_1 = \left(\begin{array}{c|c} Id - zA - iw\hat{A} & y_n \mathbf{e} \\ \hline -(z + iw) \mathbf{b}^T & y_n \end{array} \right), \quad H_2 = \left(\begin{array}{c|c} Id - zA - iw\hat{A} & \mathbf{0} \\ \hline -(z + iw) \mathbf{b}^T & 1 \end{array} \right).$$

Por tanto, aplicando la regla de Cramer al sistema lineal formado por las ecuaciones (17) y (18) tenemos que

$$y_{n+1} = \frac{\det(H_1)}{\det(H_2)}.$$

Ahora vamos a operar sobre las filas de H_1 . A cada fila le restamos la última, obteniendo lo siguiente

$$H'_1 = \left(\begin{array}{c|c} Id - zA - iw\hat{A} + (z + iw) \mathbf{b}^T & \mathbf{0} \\ \hline -(z + iw) \mathbf{b}^T & y_n \end{array} \right).$$

Por tanto, el cociente dado por la regla de Cramer es

$$y_{n+1} = \frac{\det(H'_1)}{\det(H_2)} = \frac{\det(Id - zA - iw\hat{A} + (z + iw) \mathbf{b}^T)}{\det(Id - zA - iw\hat{A})} y_n = R(z, iw) y_n$$

□

Vamos a analizar ahora los requerimientos necesarios para que el método linealmente implícito que estamos manejando tenga unas buenas propiedades de estabilidad cuando se utiliza para integrar en tiempo las ecuaciones obtenidas tras discretizar en espacio la ecuación (14).

Lo primero que haremos será definir los conceptos de región de estabilidad absoluta y de A-estabilidad.

Definición 5. *El conjunto*

$$\mathcal{R}_A := \{(z, w) \in \bar{\mathbb{C}} : |R(z, iw)| < 1\} \quad (19)$$

recibe el nombre de región de estabilidad absoluta del método (16).

Definición 6. *Un método numérico para el que*

$$\mathbb{C}^- := \{(z, w) \in \mathbb{C} : z < 0\} \subset \mathcal{R}_A \quad (20)$$

se dice A-estable.

Según esta definición, hay que remarcar que nuestro método (2) no podrá ser A-estable, ya que los términos no lineales son integrados por un método explícito, el cual sabemos que tiene una región de estabilidad acotada.

Ahora bien, supongamos que para un cierto $\alpha_0 > 0$ la parábola $z = -\alpha_0 w^2$ está contenida en la región de estabilidad de (1). Puesto que los autovalores de (16) se encuentran en la parábola $z = -\left(\frac{\nu}{a^2}\right) w^2$, la restricción de paso $h < \frac{\nu}{a^2 \alpha_0}$ aseguraría la estabilidad para todos los valores de N considerados en la discretización espacial. Notemos que cuanto menor sea el valor de α_0 , mejores propiedades de estabilidad tendremos, ya que la restricción en el paso será menor. Una condición necesaria, aunque no suficiente, para que se de la propiedad previa relativa a la parábola es

$$\lim_{z \rightarrow -\infty} R(z, iw) = 0, \quad \forall w \in \mathbb{R}, \quad (21)$$

una propiedad similar a la L-estabilidad de un método RK implícito [6].

La estructura de las matrices A y \hat{A} hace que el denominador de $R(z, iw)$ sea $(1 - \gamma z)^s$, un polinomio de grado s . En cambio, el numerador es un polinomio de grado s en z y de grado $s + 1$ en w . Por tanto, para que se cumpla la igualdad anterior, (21), el coeficiente que acompaña a z^s ha de ser cero, luego la condición que necesitamos para asegurar la propiedad de estabilidad buscada es que $\hat{a}_{s+1,1} = 0$, dado que $\hat{a}_{s+1,1}$ aparece como un factor en el coeficiente del término z^s .

1.4. UN MÉTODO RKLI DE PASO VARIABLE

En los experimentos numéricos que mostraremos más adelante, vamos a emplear un método RK linealmente implícito de 4 etapas y orden 3 propuesto en [2]. Su tablero de Butcher es el siguiente

0	0				0				
λ	0	λ			λ	0			
$\frac{1+\lambda}{2}$	0	$\frac{1-\lambda}{2}$	λ		$\frac{1+\lambda}{2} - \hat{a}_{32}$	\hat{a}_{32}	0		
1	0	b_2	b_3	λ	0	$1 - \hat{a}_{43}$	\hat{a}_{43}	0	
<hr/>					<hr/>				
	0	b_2	b_3	λ	0	b_2	b_3	λ	

Cuadro 5: Método RK linealmente implícito empleado en la integración temporal.

donde λ es la raíz intermedia del polinomio $6x^3 - 18x^2 + 9x - 1 = 0$, $b_2 = -\frac{3}{2}\lambda^2 + 4\lambda - \frac{1}{4}$, $b_3 = \frac{3}{2}\lambda^2 - 5\lambda + \frac{5}{4}$, $\hat{a}_{32} = -0.35$ y $\hat{a}_{43} = \frac{\frac{1}{3} - 2\lambda^2 - 2b_3\hat{a}_{32}\lambda}{\lambda(1-\lambda)}$. Estos valores los metemos como variables en Matlab y hacemos que nos los calcule a partir del valor de λ . Aun así, queda un parámetro libre correspondiente al valor de \hat{a}_{32} , que se elegirá con el objetivo de adaptar la región de estabilidad del método al problema al que nos estamos enfrentando. La región de estabilidad del método linealmente implícito debería contener a las curvas $z = -\alpha_0 w^2$, con $\alpha_0 > 0$ lo más pequeño posible. Estas regiones las mostramos en la siguiente imagen, extraída del artículo [2].

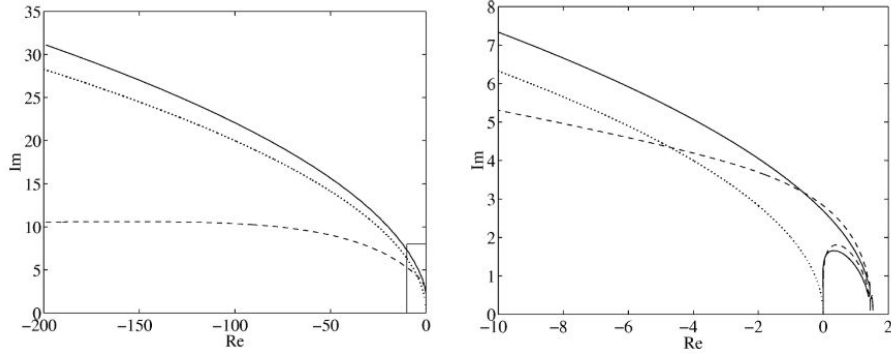


Figura 1: Regiones de estabilidad de los métodos RKLI

La función de estabilidad $R(z, iw)$ del método (5) es

$$\begin{aligned}
& \frac{1}{(1-\gamma z)^3} \left(1 + z(1-3\gamma) + z^2 \left(3\gamma^2 - 3\gamma + \frac{1}{2} \right) \right. \\
& \quad \left. + iw \left[1 + z(1-3\gamma) + z^2 \left(2\gamma^3 - 3\gamma^2 + \frac{1}{6} \right) \right] \right. \\
& \quad \left. - \frac{w^2}{2} \left[1 + z \left(2\gamma^3 - 6\gamma^2 + \frac{2}{3} \right) - 2z^2\gamma^2 (\gamma - \hat{a}_{43}\hat{a}_{32}) \right] \right. \\
& \quad \left. - \frac{iw^3}{6} [1 - 6z\gamma^2 (\gamma - 2\hat{a}_{43}\hat{a}_{32})] + w^4\gamma^2\hat{a}_{43}\hat{a}_{32} \right).
\end{aligned}$$

Llevando a cabo los calculos correspondientes (ver [2]), se deduce que el valor del parámetro libre es $\hat{a}_{32} = -0.35$ y que el valor de α_0 es 0.205693746700778.

Con estos coeficientes implementaremos el método RK linealmente implícito con paso variable. Notamos en primer lugar que, al ser linealmente implícito, para hallar cada etapa intermedia hemos de resolver un sistema lineal. Partiendo de las ecuaciones

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{y}_n, \\ \mathbf{Y}_i &= \mathbf{y}_n + h \left[\sum_{j=2}^i a_{ij} L \mathbf{Y}_j + \sum_{j=1}^{i-1} \hat{a}_{ij} \mathbf{N}(t_n + hc_j, \mathbf{Y}_j) \right], \quad 2 \leq i \leq 4, \\ \mathbf{y}_{n+1} &= \mathbf{y}_n + h \left[\sum_{i=2}^4 b_i L \mathbf{Y}_i + \sum_{i=1}^4 \hat{b}_i \mathbf{N}(t_n + hc_i, \mathbf{Y}_i) \right], \end{aligned}$$

pasando al lado izquierdo de la ecuación que define la etapa intermedia i -ésima el término correspondiente al coeficiente a_{ii} , lo que queda en el lado derecho se puede calcular de manera explícita en función de las etapas intermedias anteriores y, resolviendo un sistema lineal con matriz $Id - ha_{ii}L$ se puede obtener el valor de \mathbf{Y}_i . Las ecuaciones que hemos de implementar son, por tanto,

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{y}_n, \\ (Id - h\lambda L) \mathbf{Y}_i &= \mathbf{y}_n + h \left[\sum_{j=2}^{i-1} a_{ij} L \mathbf{Y}_j + \sum_{j=1}^{i-1} \hat{a}_{ij} \mathbf{N}(t_n + hc_j, \mathbf{Y}_j) \right], \\ \mathbf{y}_{n+1} &= \mathbf{y}_n + h \left[\sum_{i=2}^4 b_i L \mathbf{Y}_i + \sum_{i=1}^4 \hat{b}_i \mathbf{N}(t_n + hc_i, \mathbf{Y}_i) \right], \end{aligned}$$

donde $2 \leq i \leq 4$. Hay que resolver, por tanto, 3 sistemas lineales por paso, pero todos ellos con la misma matriz.

Para calcular la longitud de paso óptima en cada iteración, estimaremos el error local mediante la fórmula de segundo orden siguiente

$$\begin{aligned} \hat{\mathbf{y}}_{n+1} &= \mathbf{y}_n + h (b_2 [L \mathbf{Y}_2 + \mathbf{N}(t_n + c_2 h, \mathbf{Y}_2)] \\ &\quad + b_3 [L \mathbf{Y}_3 + \mathbf{N}(t_n + c_3 h, \mathbf{Y}_3)] + \lambda [L \hat{\mathbf{y}}_{n+1} + \mathbf{N}(t_n + h, \mathbf{Y}_4)]), \end{aligned}$$

donde \mathbf{Y}_2 , \mathbf{Y}_3 e \mathbf{Y}_4 son las etapas internas del método de orden 3 con el que estamos trabajando. Para calcular el elemento $\hat{\mathbf{y}}_{n+1}$ hemos de resolver un sistema lineal, que es el siguiente

$$\begin{aligned} (Id - h\lambda L) \hat{\mathbf{y}}_{n+1} &= \mathbf{y}_n + h (b_2 [L \mathbf{Y}_2 + \mathbf{N}(t_n + c_2 h, \mathbf{Y}_2)] \\ &\quad + b_3 [L \mathbf{Y}_3 + \mathbf{N}(t_n + c_3 h, \mathbf{Y}_3)] + \lambda \mathbf{N}(t_n + h, \mathbf{Y}_4)). \end{aligned}$$

Esta aproximación de segundo orden puede ser interpretada como la solución generada a partir de un método Runge-Kutta linealmente implícito con una etapa adicional $\mathbf{Y}_5 = \hat{\mathbf{y}}_{n+1}$ definida por los coeficientes $a_{5,i} = \hat{a}_{5,i} = b_i$, $1 \leq i \leq 3$, $a_{5,4} = \hat{a}_{5,5} = 0$, $a_{5,5} = \hat{a}_{5,4} = \gamma$.

El estimativo lo calculamos como $EST = \|\mathbf{y}_{n+1} - \hat{\mathbf{y}}_{n+1}\|_\infty$. El motivo de usar la norma infinito es que así no influye el tamaño de la malla (dimensión de los vectores). Una vez obtenido el estimativo del error, podemos calcular la longitud de paso óptima como en el caso de métodos Runge-Kutta

$$h_{opt} = h \left(\frac{TOL}{EST} \right)^{\frac{1}{3}},$$

donde TOL es la tolerancia del error que nosotros fijamos antes de ejecutar el programa y el exponente $\frac{1}{3}$ es debido a que el estimador que estamos manejando es de orden 2. A la hora de dar cada paso, hacemos que el ordenador compare el estimativo del error con la tolerancia impuesta. Pueden ocurrir dos cosas

1. Que el estimativo sea menor que la tolerancia. En este caso hemos dado un paso más pequeño de lo necesario, pero lo aceptaremos. La siguiente longitud de paso la definiremos como $0.9 \cdot h_{opt}$.
2. Que el estimativo sea mayor que la tolerancia. Esto implica que el paso dado es demasiado grande, por lo que nos vemos obligados a rechazarlo y repetir los cálculos con una longitud de paso menor, que será $0.9h_{opt}$.

Notemos que a las nuevas longitudes de paso las multiplicamos sistemáticamente por 0.9. Esto es para asegurarnos de que son suficientemente pequeñas y evitar que se rechacen demasiados pasos, lo que supondría un elevado coste computacional.

2. MODELOS DE CRECIMIENTO DE TUMORES

El objetivo de esta sección es describir de forma concisa las características de los tumores y su forma de actuar, de tal manera que podamos comprender mejor las expresiones matemáticas de los modelos, así como las condiciones de contorno impuestas.

2.1. ANGIOGÉNESIS

En primer lugar, hay que tener en cuenta que un tumor necesita de riego sanguíneo ya que, de lo contrario, no dispondrá de suministros para crecer y expandirse hacia otras regiones del organismo. Por tanto, el futuro del tumor depende de la capacidad de este de inducir la creación de nuevos vasos sanguíneos que penetren en él y lo "alimenten". Este proceso de generación de nuevos vasos sanguíneos se conoce como *angiogénesis*. Para lograr esto, es sabido (ver

[3]) que los tumores segregan un compuesto químico conocido como *tumour angiogenesis factor* (TAF).

Existen tres procesos principales relativos a las células endoteliales (las células que recubren el interior de los vasos sanguíneos) una vez se ha esparcido el TAF y que constituyen el proceso de angiogénesis

1. Se degrada la membrana base debido a enzimas segregadas por las células.
2. Se produce una migración de las células endoteliales.
3. Tiene lugar la proliferación de las células endoteliales.

El modelo matemático que vamos a emplear trata de reproducir la difusión del TAF y el efecto que tiene sobre las células endoteliales. Incorporaremos al modelo los dos últimos procesos antes descritos, por lo que este constará de dos ecuaciones y dos variables dependientes. Las dos ecuaciones nos permitirán resolver el problema de la evolución de la concentración de TAF y de la densidad de células endoteliales, que serán las dos variables del modelo.

Vamos a suponer que el tumor y los vasos sanguíneos se encuentran lo suficientemente cerca entre sí. Una vez las células endoteliales han comenzado a migrar hacia el tumor, comienzan a proliferar; es por este motivo que a la hora de modelizar el proceso consideramos las células de los extremos de los capilares como sumideros de TAF. Con todo esto que hemos comentado, tenemos que la ecuación de conservación del TAF es la siguiente

Incremento de TAF = difusión - sumideros celulares - decaimiento del químico.

Como condición inicial se escoge una función que describa cualitativamente bien el perfil de la concentración de TAF y su comportamiento en los extremos. Nosotros vamos a considerar una concentración normalizada, de tal manera que sea 1 donde se encuentra el tumor y 0 en la zona en la que se encuentran inicialmente los vasos sanguíneos.

Ahora vamos a comentar brevemente los detalles relativos a la concentración de células endoteliales. Como hemos comentado antes, lo primero que hacen estas células es destruir las paredes de la matriz que las contiene y empiezan a migrar hacia el exterior, generando brotes de vasos capilares. A una cierta distancia de los extremos de los brotes, las células empiezan a dividirse y a proliferar, por lo que contribuyen a la población de células migratorias. Existen evidencias (ver [3]) de que la respuesta de las células al TAF es de tipo *quimiotaxis* (el movimiento de las células se produce de acuerdo al gradiente de la concentración de cierta sustancia química), por lo que el flujo de células es debido tanto a la difusión como a la quimiotaxis.

Flujo = difusión + quimiotaxis.

La ecuación de conservación de la concentración de células endoteliales viene dada, por tanto, por la siguiente expresión

Incremento de células endoteliales = migración + mitosis - pérdidas.

En cuanto a las condiciones iniciales, consideramos una densidad máxima normalizada en el extremo más alejado del tumor y nula en el resto del espacio. Las condiciones de contorno concuerdan con lo expuesto anteriormente, la densidad es máxima en el extremo opuesto del tumor puesto que ahí están los vasos sanguíneos y es nula justo en el tumor, pues las células endoteliales aun no lo han alcanzado.

Así pues, teniendo en cuenta todas estas consideraciones y siguiendo lo planteado en [3] y [7], el modelo general para este proceso es el siguiente

$$\partial_t c = \delta \Delta c - \lambda c - \frac{\alpha \rho c}{\gamma + c}, \quad (22)$$

$$\partial_t \rho = \varepsilon \Delta \rho - \kappa \nabla \cdot (\rho \nabla c) + \mu \rho (1 - \rho) \max(0, c - c^*) - \beta \rho. \quad (23)$$

2.2. INVASIÓN TUMORAL

Todos los conceptos explicados en el anterior modelo relativos a las características y forma de actuar de los tumores son aplicables también en el proceso de invasión tumoral, que vamos a describir en esta sección.

En la sección anterior considerábamos el tumor estático y nos centrábamos en estudiar la evolución de la concentración de células sanguíneas, que son las responsables de que al formar las estructuras correspondientes (los vasos sanguíneos) los nutrientes alcancen el tumor y le permitan desarrollarse. En este modelo analizaremos el avance del tumor a medida que va degradando las estructuras celulares del entorno que lo rodea. Para ello, emplearemos un modelo con tres ecuaciones y tres variables dependientes, que introduciremos a continuación.

Como variable de mayor interés tendremos la densidad de células tumorales, que denotaremos por n . El avance del tumor depende de la capacidad del mismo de degradar el tejido que lo rodea. El estado de este tejido estará representado por la densidad de matriz extracelular, ECM (*extracellular matrix*), que representaremos por la variable c_1 . Para degradar el entorno, el tumor emplea una serie de enzimas denotadas por las siglas MDE (*matrix degradative enzymes*), cuya densidad vendrá recogida en la variable c_2 . Las hipótesis principales del modelo son que las células tumorales producen MDEs que degradan la ECM localmente, lo cual resulta en la producción de una serie de moléculas que ejercen atracción sobre las células tumorales, por lo que ayudan directamente a la movilidad del tumor, proceso conocido por *haptotaxis*.

En este modelo se consideran dos contribuciones al movimiento del tumor, una aleatoria y otra debida a la haptotaxis antes mencionada. Por tanto, la expresión que describe el flujo de las células tumorales es

$$\mathbf{J}_n = \mathbf{J}_{random} + \mathbf{J}_{hпто}$$

Para las otras variables, tenemos que las MDEs son producidas por las células tumorales y se difunden a través del medio sufriendo una descomposición

con el avance. Finalmente, la presencia de esta enzimas degrada la ECM, que suponemos sin movilidad. Todas estas consideraciones aparecen reflejadas en las ecuaciones generales del modelo propuesto en [5] y que mostramos a continuación

$$\partial_t n = \varepsilon \Delta n - \chi \nabla \cdot (n \nabla c_1), \quad (24)$$

$$\partial_t c_1 = -\eta c_2 c_1, \quad (25)$$

$$\partial_t c_2 = d_2 \Delta c_2 + \alpha n - \beta c_2. \quad (26)$$

Como comentario final, en este modelo no se ha considerado la proliferación de células tumorales, únicamente se estudia su avance. No obstante, la proliferación se podría tener en cuenta añadiendo un término adicional a la ecuación que describe la variación de la densidad de células tumorales, y las ecuaciones resultantes se resolverían de manera análoga.

3. RESOLUCIÓN NUMÉRICA DE MODELOS UNIDIMENSIONALES

3.1. ANGIOGÉNESIS

Vamos a trabajar con un modelo de angiogénesis en 1D siguiendo el empleado en [3]. Sus ecuaciones son las siguientes

$$c_t = \delta c_{xx} - \lambda c - \frac{\alpha \rho c}{\gamma + c}, \quad (27)$$

$$\rho_t = \varepsilon \rho_{xx} - \kappa (c_x \rho)_x + \mu \rho (1 - \rho) \max(0, c - c^*) - \beta \rho, \quad (28)$$

con $x \in [0, 1]$, $t \in [0, 0.7]$, y donde ρ es la concentración de células endoteliales y c es la concentración de TAF. Los valores de los diferentes parámetros son

$$\begin{array}{lll} \varepsilon = 0.001 & \delta = 1 \text{ o } 0.001 & \alpha = 10 \\ \beta = 4 & \gamma = 1 & \kappa = 0.75 \\ \lambda = 1 & \mu = 100 & c^* = 0.2 \end{array}$$

Las condiciones iniciales y de contorno se han cogido de [3] y son

$$c(x, 0) = \cos\left(\frac{\pi}{2}x\right), \quad c(0, t) = 1, \quad c(1, t) = 0. \quad (29)$$

$$\rho(x, 0) = \begin{cases} 0 & \text{si } 0 \leq x < 1, \\ 1 & \text{si } x = 1. \end{cases} \quad \rho(0, t) = 0, \quad \rho(1, t) = 1. \quad (30)$$

Lo primero que haremos será discretizar espacialmente las ecuaciones del modelo para después integrarlas temporalmente empleando el método RK linealmente implícito que describimos con anterioridad.

3.1.1. DISCRETIZACIÓN ESPACIAL

Observamos que estas ecuaciones tienen dos partes bien diferenciadas. La primera, la que involucra a la derivada segunda espacial (término de difusión), la cual sabemos que podemos discretizar de forma lineal usando diferencias centrales. La segunda parte está formada por el resto de términos que, en general, son no lineales. Notemos que, una vez discretizado el problema, el resultado encajará con lo expuesto en la parte teórica, por lo que podremos aplicar un método RK linealmente implícito para integrar temporalmente las ecuaciones resultantes.

El intervalo espacial que vamos a considerar es, por sencillez, $[0, 1]$. Para la discretización espacial de (17)-(18) vamos a utilizar diferencias finitas. Para ello, en el intervalo $[0, 1]$ consideramos una malla uniforme de $m + 1$ nodos x_j , $j = 0, \dots, m$, donde $x_0 = 0$, $x_m = 1$ y $x_j = j\Delta x$, siendo $\Delta x = \frac{1}{m}$ el diámetro de la malla.

A la hora de discretizar, denotamos por $C_j(t)$ al valor aproximado de $c(x_j, t)$ en el nodo x_j en el tiempo t , y por $R_j(t)$ al valor aproximado de $\rho(x_j, t)$ en el nodo x_j y en el tiempo t . A partir de ahora, y mientras estemos hablando sólo de la discretización espacial, omitiremos la dependencia de t por comodidad en la notación. Dicho esto, los términos de difusión se discretizan mediante diferencias centradas de segundo orden

$$c_{xx}|_{x=x_j} \approx \frac{C_{j+1} - 2C_j + C_{j-1}}{(\Delta x)^2}, \quad 1 \leq j \leq m-1, \quad (31)$$

$$\rho_{xx}|_{x=x_j} \approx \frac{R_{j+1} - 2R_j + R_{j-1}}{(\Delta x)^2}, \quad 1 \leq j \leq m-1. \quad (32)$$

La ecuación semidiscretizada para el TAF resulta ser

$$\frac{d}{dt}C_j = \delta \left(\frac{C_{j+1} - 2C_j + C_{j-1}}{(\Delta x)^2} \right) - \lambda C_j - \frac{\alpha R_j C_j}{\gamma + C_j}, \quad j = 1, \dots, m-1, \quad (33)$$

con $C_0 \equiv 1$ y $C_m \equiv 0$.

Ahora vamos a discretizar la ecuación de la concentración de células endoteliales. Para ello seguimos el desarrollo llevado a cabo en [7], ya que el término de advección $(c_x \rho)_x$ requiere un tratamiento específico

$$-(c_x \rho)_x|_{x=x_j} \approx \frac{1}{\Delta x} (a_{j-1/2} R_{j-1/2} - a_{j+1/2} R_{j+1/2}), \quad j = 1, \dots, m-1 \quad (34)$$

donde $a_{j+1/2} = \frac{1}{\Delta x} (-C_j + C_{j+1})$, $j = 0, \dots, m-1$, es una aproximación a c_x en $x = x_j + \frac{\Delta x}{2}$,

Así pues, la ecuación semi discretizada queda

$$\begin{aligned} \frac{d}{dt}R_j &= \varepsilon \left(\frac{R_{j+1} - 2R_j + R_{j-1}}{(\Delta x)^2} \right) + \frac{\kappa}{(\Delta x)^2} (a_{j-1/2}R_{j-1/2} - a_{j+1/2}R_{j+1/2}) \\ &\quad + \mu R_j(1 - R_j) \text{máx}(0, C_j - c^*) - \beta R_j, \quad 1 \leq j \leq m-1, \end{aligned} \quad (35)$$

con $R_0 \equiv 0$ y $R_m \equiv 1$.

En esta expresión nos falta hacer una pequeña modificación que viene motivada por lo siguiente: puesto que estamos trabajando con aproximaciones, puede darse el caso de que, en ocasiones, nos encontremos con algún j para el cual R_j tenga un valor negativo, aunque pequeño. En esta circunstancia pueden desarrollarse inestabilidades que colapsen el modelo. Es por ello que el término $\mu R_j(1 - R_j)$ se sustituye por $\mu |R_j|(1 - R_j)$.

Nos planteamos, por último, como resolver los elementos del término de advección con subíndice semientero. Hemos considerado tres posibilidades.

1. Usar un esquema de primer orden en el cual hagamos

$$R_{j+1/2} = R_j, \quad j = 0, \dots, m-1, \quad (36)$$

y así con todos los términos con subíndice semientero.

2. Emplear un esquema de segundo orden que consiste en un promedio de los dos valores más próximos

$$R_{j+1/2} = \frac{1}{2} (R_j + R_{j+1}), \quad j = 0, \dots, m-1. \quad (37)$$

3. Por último, tenemos un esquema de orden 3 tomando

$$R_{j+1/2} = \begin{cases} \frac{1}{6} (-R_{j-1} + 5R_j + 2R_{j+1}) & \text{si } a_{j+1/2} \geq 0, \\ \frac{1}{6} (2R_j + 5R_{j+1} - R_{j+2}) & \text{si } a_{j+1/2} < 0. \end{cases} \quad (38)$$

Notemos que el índice j varía desde 0 hasta $m-1$, por lo que en este último esquema vamos a necesitar dos nodos auxiliares, uno a cada extremo, que denotaremos por x_{-1} y x_{m+1} . Las aproximaciones de ρ en estos puntos las vamos a construir mediante extrapolación cuadrática, utilizando las aproximaciones a ρ en los tres puntos de la malla más próximos

$$\begin{aligned} R_{-1} &= 3R_0 - 3R_1 + R_2, \\ R_{m+1} &= 3R_m - 3R_{m-1} + R_{m-2}. \end{aligned}$$

Una vez hecho esto, ya tenemos resuelta la parte relativa a la discretización espacial. De esta forma, podemos plantear el problema como un sistema de $2m + 2$ ecuaciones diferenciales de primer orden acopladas cuya expresión es la siguiente

$$\frac{d}{dt} \begin{pmatrix} C(t) \\ R(t) \end{pmatrix} = L \cdot \begin{pmatrix} C(t) \\ R(t) \end{pmatrix} + N(C(t), R(t)),$$

donde $\begin{pmatrix} C(t) \\ R(t) \end{pmatrix}$ es un vector de $2m + 2$ componentes, formado por la concatenación de los $C_j(t)$ y los $R_j(t)$, con $j = 1, \dots, m + 1$. La matriz L contiene los términos de la discretización de la parte de difusión, mientras que N es un vector de longitud $2m + 2$ que contiene la discretización del resto de términos. Observamos que el formato de este sistema se ajusta al expuesto en la sección dedicada a la teoría, por lo que podemos emplear un método RK linealmente implícito para integrarlo temporalmente.

3.1.2. LIMITADORES DE FLUJO

A la hora de ejecutar el programa una vez implementadas las ecuaciones antes descritas, nos encontramos con soluciones que difieren de las esperadas. El motivo es que los esquemas de orden 2 y 3 producen soluciones con un comportamiento oscilatorio debidos a la parte de advección, lo cual es problemático cuando nos encontramos cerca del 0, ya que la aproximación podría tomar valores negativos debido a estas oscilaciones. Para que los esquemas que estamos manejando funcionen correctamente, es necesario garantizar la positividad de la discretización espacial de las ecuaciones. Para corregir este comportamiento y que no aparezcan soluciones negativas que generen inestabilidades hemos recurrido a una técnica conocida como *limitadores de flujo* (ver [7]).

La técnica de limitación de flujo consiste en sustituir los términos $a_{j+1/2}R_{j+1/2}$ por el elemento de flujo siguiente

$$f_{j+1/2}(t, R) = \begin{cases} a_{j+1/2} [R_j + \psi(\theta_j) (R_{j+1} - R_j)], & \text{si } a_{j+1/2} \geq 0, \\ a_{j+1/2} \left[R_{j+1} + \psi \left(\frac{1}{\theta_{j+1}} \right) (R_j - R_{j+1}) \right], & \text{si } a_{j+1/2} < 0, \end{cases}$$

donde

$$\theta_j = \frac{R_j - R_{j-1}}{R_{j+1} - R_j}.$$

Notemos que en la primera expresión el índice va desde 0 hasta $m - 1$, por lo que a la hora de construir la función θ seguiremos precisando de los nodos auxiliares R_{-1} y R_{m+1} que mencionamos antes. A la hora de trabajar computacionalmente con esta función hay que tener en cuenta que el denominador puede anularse, por lo que Matlab registraría el resultado como un *not a number* (NaN). Siguiendo [4], hemos solucionado este problema sumando en el numerador y en el

denominador la variable eps , que es la mínima cantidad que en Matlab satisface $1 + eps = 1$. Así, cuando la resta $R_{j+1} - R_j$ sea 0, el denominador no se anulará debido a la presencia de esta variable, haciendo que el resultado sea un número muy alto con el que el programa puede seguir trabajando, en vez de un *not a number*, que haría que el programa diese error. Por tanto, computacionalmente trabajaremos con la variable

$$\theta_j = \frac{R_j - R_{j-1} + eps}{R_{j+1} - R_j + eps}.$$

El tipo de limitador viene determinado por la expresión de la función ψ . En nuestros experimentos hemos empleado fundamentalmente dos métodos, el primero consiste en incorporar un limitador al conocido como κ -esquema (a partir de ahora llamaremos κ -esquema a la combinación del esquema y el limitador), el segundo es el llamado limitador de *Van Leer* (ver [7]). En el primer caso,

$$\psi(\theta) = \max\left(0, \min\left(1, \frac{1}{4}(1 + \kappa) + \frac{1}{4}(1 - \kappa)\theta, \theta\right)\right), \quad (39)$$

si κ toma el valor 1 tendremos el esquema de segundo orden y si κ vale $\frac{1}{3}$, tendremos el de orden 3.

Para el limitador de *Van Leer* la expresión es

$$\psi(\theta) = \frac{1}{2} \frac{\theta + |\theta|}{1 + |\theta|}, \quad (40)$$

este método solo podemos emplearlo para el esquema de orden 3.

Para corregir el esquema de orden 1 usamos lo siguiente

$$f_{j+1/2}(t, R) = \max(a_{j+1/2}, 0)R_j + \min(a_{j+1/2}, 0)R_{j+1}.$$

Para el esquema de orden 1 no mostraremos los resultados, ya que su eficiencia es muy inferior a la de los esquemas de orden 2 y 3.

Con estos limitadores logramos corregir los problemas de positividad que se nos han presentado, obteniendo así las soluciones esperadas. En la sección de resultados compararemos los diferentes esquemas con y sin limitador, para comprender mejor el porqué de la necesidad de emplear estas técnicas.

3.1.3. RESULTADOS NUMÉRICOS

En esta sección mostraremos los diferentes resultados que hemos ido obteniendo al combinar los posibles valores de los parámetros y el diámetro de la malla utilizado en la discretización espacial.

Hemos implementado las tres discretizaciones (36), (37) y (38) para los términos de advección en (27)-(28), utilizando $\Delta x = \frac{1}{50}$, $\frac{1}{100}$ y $\frac{1}{200}$, integrando las ecuaciones semidiscretas resultantes con el método RKLI implementado con paso variable y tolerancia 10^{-5} para el error local. Como en [3] y [7], se han obtenido soluciones en $t = 0.1$ (azul), 0.3 (rojo), 0.5 (naranja) y 0.7 (violeta), la relación es que 0.4 unidades de tiempo equivalen a un tiempo real de 3.5 días

(ver [3]). En las Figuras (2) y (3) se muestran los resultados obtenidos para $\Delta x = \frac{1}{50}$ con el método de orden 2 y con el método de orden 3, respectivamente. Las gráficas de la izquierda corresponden a la concentración de células endoteliales (ρ) y las de la derecha a la concentración de TAF.

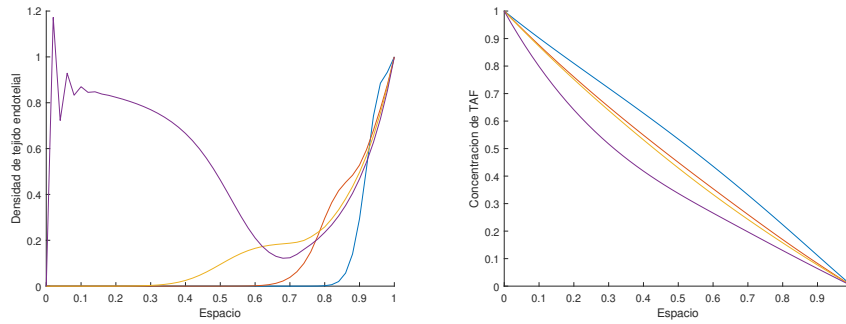


Figura 2: Método de orden 2, $\Delta x = \frac{1}{50}$, $\delta = 1$.

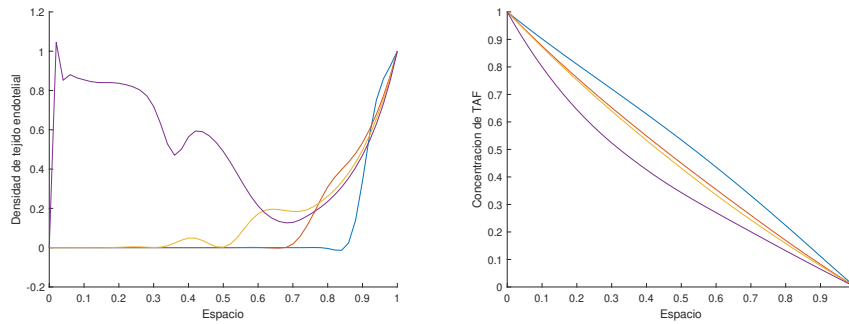


Figura 3: Método de orden 3, $\Delta x = \frac{1}{50}$, $\delta = 1$.

Las siguientes gráficas son análogas, pero con $\Delta x = \frac{1}{100}$.

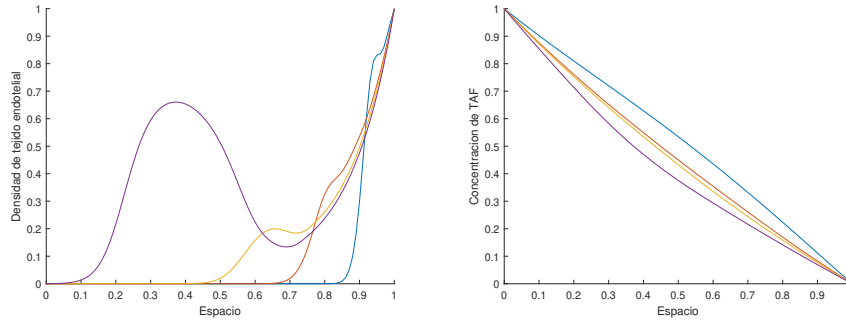


Figura 4: Método de orden 2, $\Delta x = \frac{1}{100}$, $\delta = 1$.

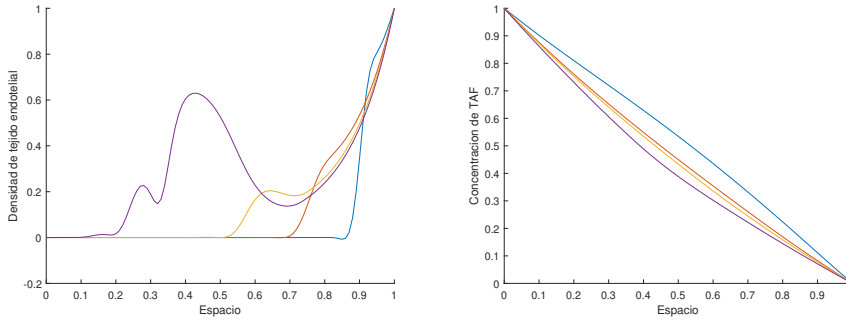


Figura 5: Método de orden 3, $\Delta x = \frac{1}{100}$, $\delta = 1$.

Comenzamos comentando los resultados del método de orden 2. Observamos que este no da problemas relativos a la positividad de la solución (solo en tiempos avanzados, de $t = 0.7$ unidades en el caso $\Delta x = \frac{1}{50}$, en donde observamos inestabilidades cerca de $x = 0$) pero, debido a su menor orden, nos vemos obligados a imponer una discretización más fina para que los resultados obtenidos sean los deseados (ver Figura 4), de lo contrario, los errores se acumulan dando lugar a una solución que difiere notablemente de la observada experimentalmente (ver Figura 2).

En cambio, si analizamos los resultados del método de orden 3, nos percatamos de que la positividad de la solución es un problema. Fijándonos en las imágenes 3 y 5 vemos que en los primeros compases ($t = 0.1$ unidades) la densidad de las células endoteliales toma valores negativos, lo que hace que el método deje de funcionar correctamente. Incluso tras refinar la discretización, en la solución nos aparece una protuberancia por delante del máximo de densidad que no se corresponde con la experimentación (ver Figura 5). Todo esto hace necesaria la aplicación de los limitadores de flujo que hemos introducido anteriormente.

Los resultados obtenidos aplicando el κ -esquema (39) son los siguientes

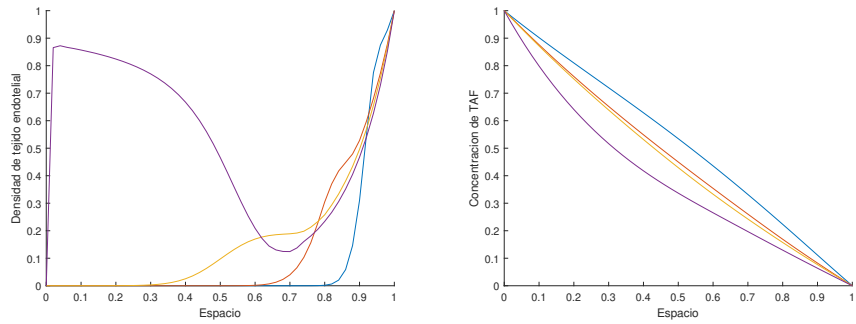


Figura 6: Método de orden 2, $\Delta x = \frac{1}{50}$, $\delta = 1$.

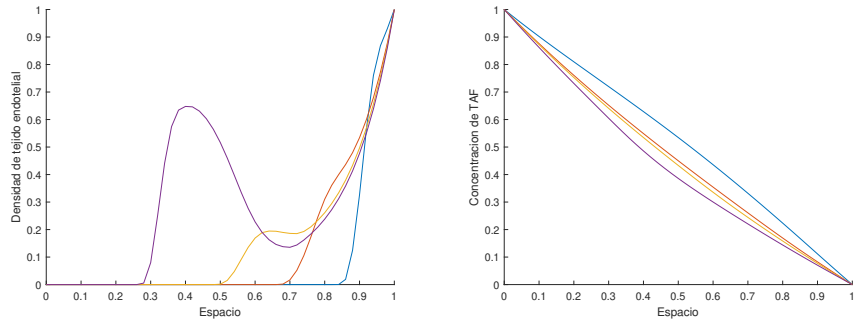


Figura 7: Método de orden 3, $\Delta x = \frac{1}{50}$, $\delta = 1$.

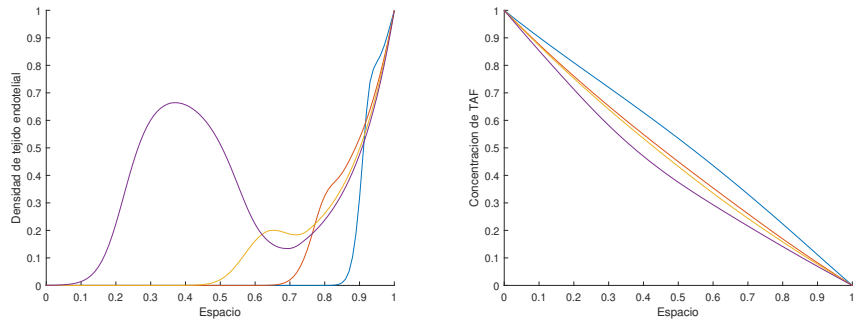


Figura 8: Método de orden 2, $\Delta x = \frac{1}{100}$, $\delta = 1$.

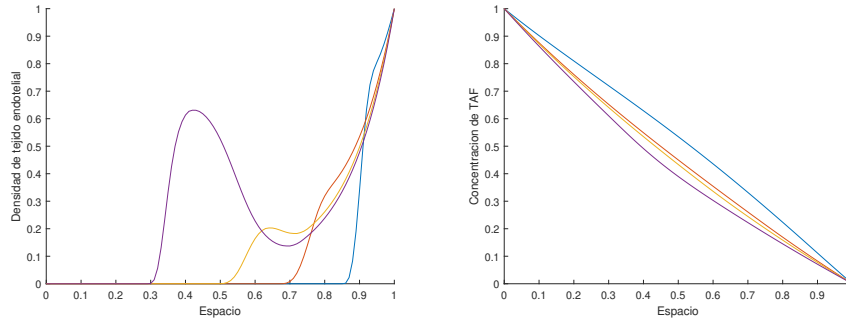


Figura 9: Método de orden 3, $\Delta x = \frac{1}{100}$, $\delta = 1$.

Las inestabilidades debidas a la no positividad de la solución han desaparecido como consecuencia del uso del limitador de flujo. En el caso del método de orden 2, los únicos errores que observamos ahora vienen dados por el menor orden y se subsanan refinando la red espacial. En el caso del método de orden 3, se puede ver que la solución no toma ya valores negativos y que no hay diferencia apreciable entre el caso $\Delta x = \frac{1}{50}$ y el $\Delta x = \frac{1}{100}$, por lo que bastaría con tomar el primero, ahorrándonos coste computacional. Nos percatamos de que las soluciones obtenidas en la Figura 8 y las obtenidas en la Figura 9 no son las mismas, esto es debido a que el método de orden 2 precisa de un diámetro de la discretización espacial menor que en el caso del método de orden 3. Ejecutando el programa para valores de Δx menores que $\frac{1}{200}$, se observa como la solución va convergiendo hacia la obtenida por el método de orden 3.

Una vez obtenidos los resultados correctos, pasamos a comentar las soluciones desde el punto de vista biológico. En cuanto a la concentración de células endoteliales, observamos que desarrolla un avance hacia la izquierda, siguiendo el gradiente de concentración de TAF que, como ya comentamos, estimula su crecimiento. A medida que pasa el tiempo, el frente se difunde para después empezar a crecer de nuevo. Esto es debido al término $\max(0, c - c^*)$, el cual es nulo hasta que se alcanza la concentración umbral de TAF, que ocurre en zonas más próximas al tumor.

Siguiendo [7], ahora vamos a resolver el mismo problema pero considerando $\delta = 0.001$, es decir, reduciendo el parámetro que da cuenta de la difusión de las células endoteliales en (27). Puesto que ya hemos mostrado los problemas que dan los métodos que no conservan la positividad de la solución, solo vamos a mostrar los resultados correspondientes a las discretizaciones espaciales que tienen el limitador de flujo incorporado. Los resultados obtenidos coinciden, como veremos, con los recogidos en [7]

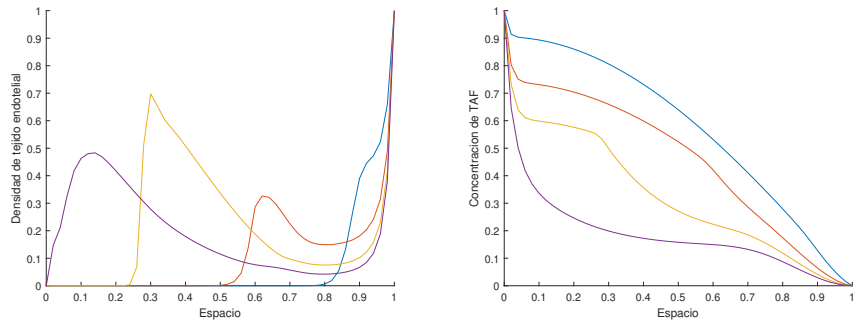


Figura 10: Método de orden 2, $\Delta x = \frac{1}{50}$, $\delta = 0.001$.

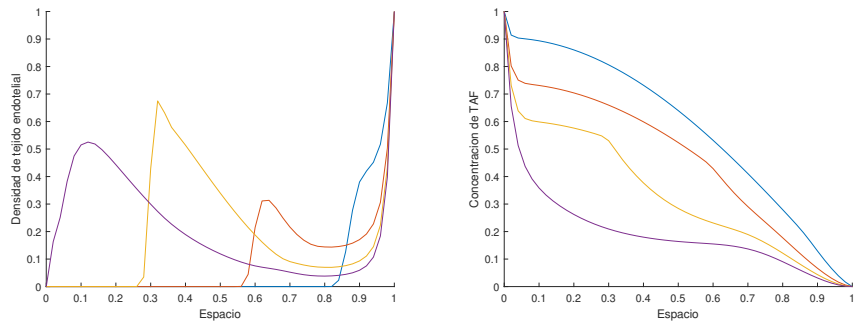


Figura 11: Método de orden 3, $\Delta x = \frac{1}{50}$, $\delta = 0.001$.

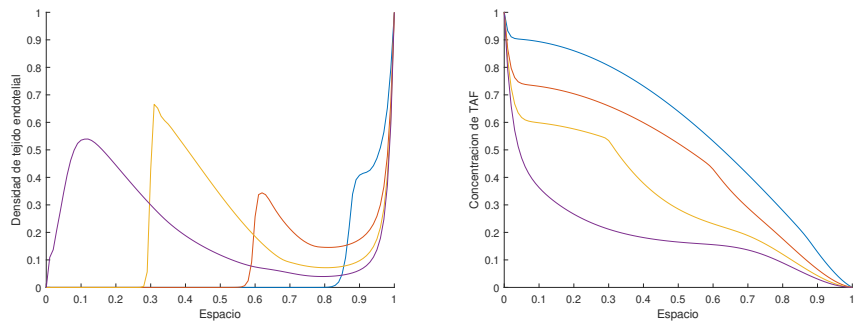


Figura 12: Método de orden 2, $\Delta x = \frac{1}{100}$, $\delta = 0.001$.

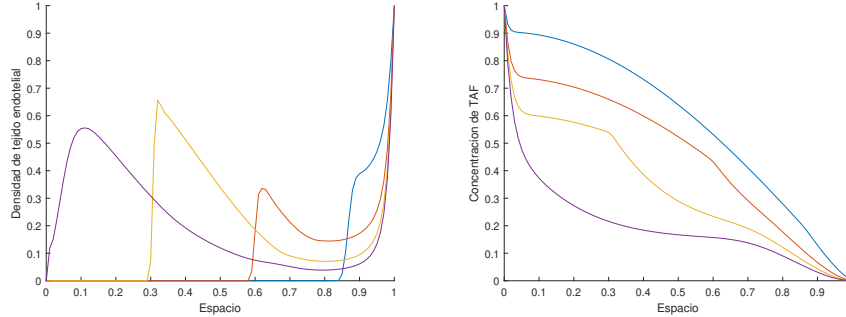


Figura 13: Método de orden 3, $\Delta x = \frac{1}{100}$, $\delta = 0.001$.

Desde el punto de vista biológico, en estos casos, al considerar un coeficiente de difusión menor, tiene lugar una mayor absorción de TAF, lo que resulta en un perfil notablemente más escarpado para la densidad de células endoteliales. El pico que observamos en el tiempo $t = 0.5$ es debido al cambio en la pendiente de la gráfica del TAF para dicho tiempo. Además, al pasar de $\Delta x = \frac{1}{50}$ a $\Delta x = \frac{1}{100}$ se observan, en particular, perfiles más verticales en $t = 0.3$ y $t = 0.5$. Por último, cabe destacar que la mayor absorción de TAF hace que el avance de las células endoteliales sea más rápido, por lo que se alcanza el tumor (situado en $x = 0$) para el tiempo $t = 0.7$, algo que no sucede cuando la difusión de células endoteliales es mayor ($\delta = 1$).

3.2. INVASIÓN TUMORAL

3.2.1. DISCRETIZACIÓN ESPACIAL

Vamos a mostrar ahora las ecuaciones que describen la evolución de las variables que mencionamos en la parte del contexto biológico en 1D. Estas ecuaciones son

$$n_t = \varepsilon n_{xx} - \chi (nc_{1x})_x, \quad (41)$$

$$c_{1t} = -\eta c_2 c_1, \quad (42)$$

$$c_{2t} = d_2 c_{2xx} + \alpha n - \beta c_2, \quad (43)$$

donde, recordemos, n denota la densidad de células tumorales, c_1 la densidad de ECM y c_2 la concentración de MDE. Vamos a comentar brevemente los términos de cada ecuación. En la ecuación (41) tenemos, en primer lugar, la difusión aleatoria y, en segundo, el término de haptotaxis. En la ecuación (42) tenemos un único término que hace referencia a la degradación de la ECM por la presencia de MDE. Por último, en la ecuación (43) tenemos, en orden, el término de difusión, el de producción y el de decaimiento.

Los valores de los parámetros de este modelo son, siguiendo [5]

$$\varepsilon = 0.001, \quad d_2 = 0.001, \quad \eta = 10, \quad \alpha = 0.1, \quad \beta = 0.$$

El valor de β normalmente se escoge igual a 0.5 pero, a la hora de resolver el problema en 1D y siguiendo la literatura, lo tomamos nulo. No obstante, hemos decidido mantenerlo en la ecuación para dejar claro que podríamos tener en cuenta el decaimiento de las MDE con tan solo añadir ese término. Por último, el parámetro χ toma los valores 0.5 y 0.005.

Ahora vamos a mostrar cuales son las condiciones iniciales y de contorno. Para las primeras, vamos a suponer que hay un nódulo de células tumorales ya presentes y que el tumor se encuentra centrado en torno a la posición $x = 0$. La densidad inicial de células tumorales se establece como

$$n(0, x) = \begin{cases} \exp\left(\frac{-x^2}{0.01}\right), & x \in [0, 0.25], \\ 0, & x \in (0.25, 1]. \end{cases}$$

Suponemos que el proceso de degradación del entorno ya se ha iniciado, por lo que tomamos el valor inicial de la densidad de ECM como

$$c_1(0, x) = 1 - 0.5n(0, x).$$

Finalmente, suponemos que la concentración inicial de MDE es proporcional a la densidad de células tumorales inicial, por lo que establecemos que

$$c_2(0, x) = 0.5n(0, x).$$

Las condiciones frontera son las de flujo nulo, es decir, la derivada de cada una de las variables en la frontera ha de ser 0. Esta condición la vamos a aplicar discretizando la derivada mediante diferencias centrales, que nos permiten tener orden 2 en la frontera. Utilizamos de nuevo, en el intervalo $[0, 1]$, una malla uniforme de $m + 1$ nodos x_j , $j = 0, \dots, m$, donde $x_0 = 0$, $x_m = 1$ y $x_j = j\Delta x$, siendo $\Delta x = \frac{1}{m}$ el diámetro de la malla. Para trabajar con las derivadas centrales, necesitamos tomar nodos auxiliares x_{-1} y x_{m+1} . Sea F_j el valor aproximado de la solución de la variable $f(x_j)$ en el nodo x_j , donde f representa a cualquiera de las tres variables dependientes de este problema. Las condiciones frontera de flujo nulo en los nodos x_0 y x_m son

$$F'_0 = \frac{F_1 - F_{-1}}{\Delta x} = 0, \quad F'_m = \frac{F_{m+1} - F_{m-1}}{\Delta x} = 0.$$

Por tanto, las condiciones frontera nos llevan a

$$F_{-1} = F_1, \quad F_{m+1} = F_{m-1}.$$

Para llevar a cabo la discretización espacial, vamos a aprovechar los desarrollos que hicimos en el modelo de angiogénesis, ya que la resolución de ambos

casos es muy similar. La diferencia más notable es el cambio en las condiciones de contorno y que ahora trabajamos con una ecuación más. Así pues, la discretización de los términos de difusión la haremos con diferencias centrales de segundo orden, como en el anterior modelo, y para el término de advección aplicaremos directamente la técnica de los limitadores de flujo, implementando el κ -esquema.

Denotamos por $N_j(t)$, $C_j(t)$ y $Z_j(t)$ a las soluciones aproximadas de las variables n , c_1 y c_2 , respectivamente, en el nodo x_j en el tiempo t . Como hemos hecho anteriormente, la dependencia temporal no la especificaremos para aligerar la notación. Los términos de difusión de las ecuaciones (41) y (43) se discretizan de la siguiente manera

$$\begin{aligned} n_{xx} \Big|_{x=x_j} &= \frac{N_{j+1} - 2N_j + N_{j-1}}{(\Delta x)^2}, \quad j = 1, \dots, m-1, \\ c_{2xx} \Big|_{x=x_j} &= \frac{Z_{j+1} - 2Z_j + Z_{j-1}}{(\Delta x)^2}, \quad j = 1, \dots, m-1. \end{aligned}$$

Notemos que no hemos incorporado las ecuaciones correspondientes a los nodos frontera, x_0 y x_m , porque en esas ecuaciones hay que tener en cuenta las condiciones de contorno que hemos especificado anteriormente. Una vez incorporadas, la discretización en los puntos frontera es

$$\begin{aligned} n_{xx} \Big|_{x=x_0} &= \frac{2N_1 - 2N_0}{(\Delta x)^2}, \\ n_{xx} \Big|_{x=x_m} &= \frac{2N_{m-1} - 2N_m}{(\Delta x)^2}, \\ c_{2xx} \Big|_{x=x_0} &= \frac{2Z_1 - 2Z_0}{(\Delta x)^2}, \\ c_{2xx} \Big|_{x=x_m} &= \frac{2Z_{m-1} - 2Z_m}{(\Delta x)^2}. \end{aligned}$$

El término de advección discretizado sería

$$(nc_{1x})_x \Big|_{x=x_j} = \frac{f_{j-1/2} - f_{j+1/2}}{\Delta x}, \quad j = 0, \dots, m,$$

donde $f_{j+1/2}$ viene definido por la expresión (39).

Por último, queda discretizar los términos más simples de las ecuaciones, que son

$$\begin{aligned} \eta c_2 c_1 \Big|_{x=x_j} &= \eta Z_j C_j, \\ (\alpha n - \beta c_2) \Big|_{x=x_j} &= \alpha N_j - \beta Z_j. \end{aligned}$$

Hecho esto, podemos expresar las tres ecuaciones diferenciales discretizadas como suma de una parte lineal, que incluye la discretización de los términos de difusión de n y c_2 , y una parte no lineal incluyendo el resto de términos cuya discretización hemos detallado anteriormente.

3.2.2. RESULTADOS NUMÉRICOS

Al igual que hemos hecho para el modelo de angiogénesis, vamos a mostrar los resultados obtenidos para el caso de invasión tumoral, analizando cómo varían las soluciones dependiendo del valor de los parámetros. Como hemos mencionado anteriormente, a la hora de resolver este modelo hemos aplicado directamente los limitadores de flujo, para así obtener desde el principio los resultados correctos. Además, hemos ejecutado el método de orden 3, ya que se ha podido comprobar que los resultados son mejores, incluso con una red espacial más gruesa.

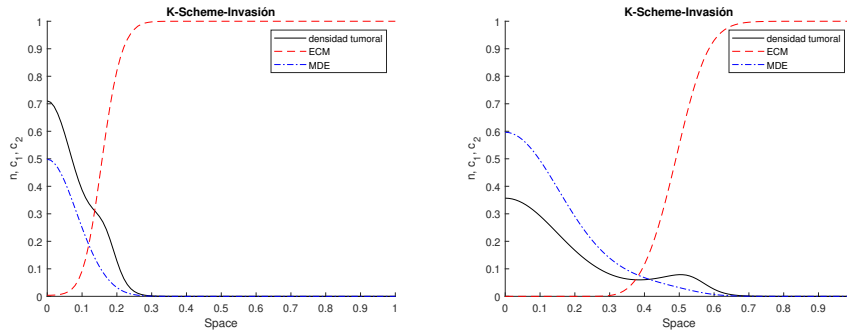


Figura 14: $\Delta x = \frac{1}{100}$, $\chi = 0.005$, $t = 1$, 10 unidades.

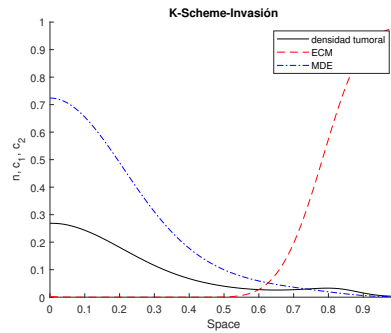


Figura 15: $\Delta x = \frac{1}{100}$, $\chi = 0.005$, $t = 20$ unidades.

En los gráficos podemos observar la evolución de la densidad de células tumorales (negro), de ECM (rojo) y de MDE (azul) en el espacio y para diferentes tiempos.

El procedimiento de actuación es el siguiente, las células tumorales segregan MDE, que son las enzimas que se encargan de la degradación del medio, de ahí que en las zonas donde estas abundan, la densidad de ECM sea baja. A medida que las MDE van degradando el medio, el tumor va avanzando, que es lo que se observa en la línea de color negro representada en las tres gráficas.

Es importante resaltar que no hemos incluido un término de generación de células tumorales, por lo que su expansión por el organismo tiene un límite. Además, tampoco hemos incluido un término de decaimiento de las MDE, por lo que estas son generadas por el tumor de forma continuada.

Otro aspecto que merece la pena comentar es que el tumor parece desarrollarse lentamente. Esto se debe a que los parámetros que dan peso a los términos de difusión y haptotaxis son comparables. Esto implica que la evolución del tumor tiene un importante componente aleatorio, es decir, que habrá células que se desplacen en el sentido de avance marcado por las MDE pero habrá otras que lo hagan en sentido opuesto debido a ese término difusivo. Estos parámetros nos permiten modelar diferentes tumores según su forma de desarrollarse.

Para completar más este último comentario, hemos ejecutado el programa modificando el valor del parámetro χ , los resultados que se obtienen en este caso pueden verse en la Figura 16.

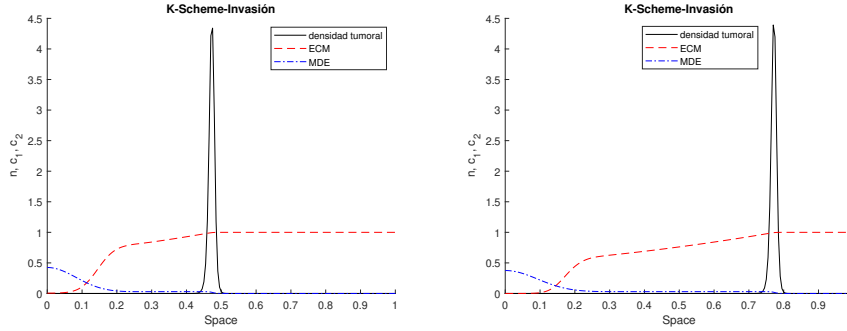


Figura 16: $\Delta x = \frac{1}{200}$, $\chi = 0.5$, $t = 1, 2$ unidades.

Como hemos comentado anteriormente, la evolución del tumor según este modelo está condicionada en gran medida por los valores de los parámetros ε y χ . En este segundo caso que nos ocupa, se ha aumentado el valor de χ hasta 0.5 unidades, lo que hace que sea dos órdenes de magnitud mayor que ε . Esto implica que el término de difusión se vuelve despreciable comparado con el de haptotaxis, por lo que en las gráficas no observamos movimiento aleatorio de las células tumorales, si no que estas se agrupan y avanzan en la misma dirección a una velocidad mucho mayor que en el otro caso, hasta tal punto que solo podemos simular hasta $t = 2$ unidades, pues para tiempos mayores el tumor alcanza la frontera y el modelo deja de ser válido. Como consecuencia, obtenemos los resultados correspondientes a un cáncer muy agresivo. De nuevo, las Figuras 14, 15 y 16 reproducen resultados que habían aparecido en la literatura [5].

BIBLIOGRAFÍA

- [1] A. L. Araújo, A. Murúa y M. Sanz-Serna. «Symplectic Methods Based on Decompositions». En: *SIAM Journal of Numerical Analysis* 34 (1997), págs. 1926-1947.
- [2] M. P. Calvo, J. de Frutos y J. Novo. «Linearly implicit Runge-Kutta methods for advection-reaction-diffusion equations». En: *Applied Numerical Mathematics* 37 (2001), págs. 535-549.
- [3] M. A. J. Chaplain y A. M. Stuart. «A model mechanism for the chemotactic response of endothelial cells to tumour angiogenesis factor». En: *IMA Journal of Mathematics Applied in Medicine and Biology* 10 (1993), págs. 149-168.
- [4] A. Gerisch y et al. «A Positive Splitting Method for Mixed Hyperbolic-Parabolic Systems». En: *Numerical Methods for Partial Differential Equations* 17 (2001), págs. 152-168.
- [5] A. Gerisch y M.A.J Chaplain. «Robust numerical methods for taxis-diffusion-reaction systems: Applications to biomedical problems». En: *Mathematical and Computer Modelling* 43 (2006), págs. 49-75.
- [6] E. Hairer y G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer, 1996.
- [7] W. Hundsdorfer y J.G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer, 2003.

A. PROGRAMAS DE MATLAB

En este apéndice mostramos los códigos de los programas de Matlab que hemos realizado para la resolución de los modelos de angiogénesis e invasión. Para el primer caso hay dos programas, el primero no tiene implementados los limitadores de flujo, el segundo lleva implementado el κ -esquema. Para el segundo modelo solo hay un programa que lleva implementadas todas las correcciones.

A.1. MODELO DE ANGIOGÉNESIS

En primer lugar, incorporamos el código sin limitadores de flujo

```
function [] = SinLimitador(ind,m,tf)
    %Definimos las variables que vamos a emplear como globales
    global e d a b g k l mu cc hh

    %Definimos la longitud de la discretizacion espacial
    hh = 1/m;

    %Definimos nuestro vector de soluciones
    sol = zeros(2*m+2,1);

    %Vamos a introducir las condiciones iniciales
    for j=1:m+1
        sol(j) = cos((pi*(j-1)*hh)/2);
    end
    sol(2*m+2) = 1;

    %Introducimos las constantes
    e = 0.001;
    d = 1;
    a = 10;
    b = 4;
    g = 1;
    k = 0.75;
    l = 1;
    mu = 100;
    cc = 0.2;

    t0 = 0;
    h0 = 1/100; %Longitud de paso temporal inicial
    tol = 1/100000; %Tolerancia del error en la solucion
    %Resolvemos el problema bajo estas condiciones
    [xx,~] = RKLinImp(@funL,@funN,t0,tf,h0,tol,sol,ind,m);
    espacio = [0:1/m:1];

    %Representamos los resultados
    figure(1)
    hold on
    plot(espacio,xx(end,m+2:2*m+2));
    xlabel('Espacio')
    ylabel('Densidad de tejido endotelial')

    figure(2)
```



```

hold on
plot(espacio,xx(end,1:m+1));
xlabel('Espacio')
ylabel('Concentracion de TAF')

end

%Funcion que implementa el metodo RKLI
function [xx,tt] = RKLinImp(funL,funN,t0,tf,h0,tol,sol,ind,m)
    %Importamos el metodo desde un programa auxiliar
    [A,Ahat,bb,~,lamb] = tableroRK();
    h = h0;
    tt(1) = t0;
    n = 1;
    xx(1,:) = sol';
    s = 3; %Numero de etapas del metodo
    L = funL(m);
    Y = zeros(s+1,2*m+2);
    I = eye(2*m+2);

    %Fijamos la condicion de parada del bucle while
    while tt(n)<tf

        %En caso de pasarnos de tf, reducimos el paso
        if tt(n)+h>tf
            h = tf - tt(n);
        end

        %Definimos la etapa inicial
        Y(1,:) = xx(n,:);
        N = funN(xx(n,:),ind,m);
        NN(1,:) = N';

        %Ejecutamos el resto de etapas
        for i=2:s+1
            Y(i,:) = (I-h*A(i,i)*L)\(xx(n,:) + ...
                h*(A(i,1:i-1)*(L*Y(1:i-1,:))' + ...
                + Ahat(i,1:i-1)*NN(1:i-1,:))');
            N = funN(Y(i,:),ind,m);
            NN(i,:) = N';
        end

        %Vamos a guardar la solucion de forma provisional, pues la
        %necesitamos para comprobar si el paso que hemos dado es lo
        %suficientemente pequeno
        xx(n+1,:) = xx(n,:);
        for i=1:s+1
            xx(n+1,:) = xx(n+1,:) + h*(bb(i)*(L*Y(i,:))' + bb(i)*NN(i,:));
        end

        %Ahora vamos a calcular el estimativo del error
        yy = ((I-h*lamb*L)\(xx(n,:) + h*(bb(2)*((L*Y(2,:))'+...
            NN(2,:))+bb(3)*((L*Y(3,:))'+NN(3,:)) + lamb*NN(4,:))')');

        %Con la norma infinito no nos influye el tamano de la malla
        %(espacio)
        EST = norm(xx(n+1,)-yy, "inf");
        hopt = h*(tol/EST)^(1/3); %Porque el orden del estimador es 2

```

```

%Distinguimos casos en funcion del estimativo del error
if EST<tol
    %En este caso, aceptamos el paso y nos quedamos con la
    %solucion que habiamos calculado antes
    tt(n+1) = tt(n) + h;
    n = n+1;
    h = 0.9*hopt;

else
    %Rechazamos el paso
    h = 0.9*hopt;
    %Al no aumentar la n, la siguiente iteracion del bucle va a
    %reescribir la anterior solucion
    continue
end
end
end

%Funcion que recoge los datos de los tableros de Butcher
function [A,Ahat,bb,c,lamb] = tableroRK()

    lamb = 4.358665215084597e-01;
    r = (-3/2)*lamb^2 + 4*lamb -1/4;
    s = (3/2)*lamb^2 - 5*lamb + 5/4;
    t = -0.35;
    u = (1/3 - 2*lamb^2 - 2*s*(-0.35)*lamb)/(lamb*(1-lamb));

    A = zeros(4);
    Ahat = zeros(4);

    A(2,2)=lamb;
    A(3,2)=(1-lamb)/2; A(3,3)=lamb;
    A(4,2)=r; A(4,3)=s; A(4,4)=lamb;

    bb = [0;r;s;lamb];

    Ahat(2,1)=lamb;
    Ahat(3,1)=((1+lamb)/2)-t; Ahat(3,2)=t;
    Ahat(4,2)=1-u; Ahat(4,3)=u;

    c = [0;lamb;(1+lamb)/2;1];
end

%Vamos a programar una funcion que nos evalúe el lado derecho del
%sistema de ecuaciones diferenciales y'=Ly+N(y)
function L = funL(m)
    global e d hh
    L = zeros(2*m+2);
    for j=2:m
        L(j,j) = -2*d;
        L(j,j-1) = d;
        L(j,j+1) = d;
        L(j+m+1,j+m+1) = -2*e;
        L(j+m+1,j+m) = e;
        L(j+m+1,j+m+2) = e;
    end
end

```

```

L = (1/hh^2)*L;
end
function N = funN(sol,ind,m)
global a b g k l mu cc hh
N = zeros(2*m+2,1);
%Generamos los coeficientes a_j+1/2
aa = zeros(m,1);
for j=1:m
    aa(j) = (1/hh)*(sol(j+1) - sol(j));
end
if ind==1
    for j=2:m
        N(j) = -1*sol(j) - (a*sol(j+m+1)*sol(j))/(g + sol(j));
        N(j+m+1) = (k/hh)*(aa(j-1)*sol(j+m) - aa(j)*sol(j+m+1)) +...
            mu*abs(sol(j+m+1))*(1-sol(j+m+1))*max(0,sol(j)-cc) -...
            b*sol(j+m+1);
    end
elseif ind==2
    for j=2:m
        N(j) = -1*sol(j) - (a*sol(j+m+1)*sol(j))/(g + sol(j));
        N(j+m+1) = (k/(2*hh))*(aa(j-1)*(sol(j+m)+sol(j+m+1)) -...
            aa(j)*(sol(j+m+1)+sol(j+m+2))) +...
            mu*abs(sol(j+m+1))*(1-sol(j+m+1))*max(0,sol(j)-cc) -...
            b*sol(j+m+1);
    end
else
    ind1 = aa>=0;
    ind2 = aa<0;
    s0 = 3*sol(m+2)-3*sol(m+3)+sol(m+4);
    sf = 3*sol(2*m+2)-3*sol(2*m+1)+sol(2*m);
    omega = (1/6)*([s0 sol(m+2:2*m)]+5*sol(m+2:2*m+1)+...
        2*sol(m+3:2*m+2)).*ind1 + (1/6)*(2*sol(m+2:2*m+1)+...
        5*sol(m+3:2*m+2)-[sol(m+4:2*m+2) sf]).*ind2;
    for j=2:m
        N(j) = -1*sol(j) - (a*sol(j+m+1)*sol(j))/(g + sol(j));
        N(j+m+1) = (k/hh)*(aa(j-1)*omega(j-1) - aa(j)*omega(j)) +...
            mu*abs(sol(j+m+1))*(1-sol(j+m+1))*max(0,sol(j)-cc) -...
            b*sol(j+m+1);
    end
end
end
end

```

Mostramos ahora el código con el limitador aplicado al κ -esquema

```

function [] = kScheme(ind,m,tf)
%Definimos las variables que vamos a emplear como globales
global e d a b g k l mu cc hh

%Definimos la longitud de la discretizacion espacial
hh = 1/m;

%Definimos nuestro vector de soluciones
sol = zeros(2*m+2,1);

%Vamos a introducir las condiciones iniciales
for j=1:m+1
    sol(j) = cos((pi*(j-1)*hh)/2);
end

```

```

end
sol(2*m+2) = 1;

%Introducimos las constantes
e = 0.001;
d = 1;
a = 10;
b = 4;
g = 1;
k = 0.75;
l = 1;
mu = 100;
cc = 0.2;

t0 = 0;
h0 = 1/100; %Longitud de paso temporal inicial
tol = 1/100000; %Tolerancia del error en la solucion
%Resolvemos el problema bajo estas condiciones
[xx, ~] = RKLinImp(@funL,@funN,t0,tf,h0,tol,sol,ind,m);
espacio = [0:1/m:1];

%Representamos los resultados
figure(1)
hold on
plot(espacio,xx(end,m+2:2*m+2));
xlabel('Espacio')
ylabel('Densidad de tejido endotelial')

figure(2)
hold on
plot(espacio,xx(end,1:m+1));
xlabel('Espacio')
ylabel('Concentracion de TAF')

end

%Funcion que implementa el metodo RKLI
function [xx,tt] = RKLinImp(funL,funN,t0,tf,h0,tol,sol,ind,m)
%Importamos el metodo desde un programa auxiliar
[A,Ahat,bb,~,lamb] = tableroRK();
h = h0;
tt(1) = t0;
n = 1;
xx(1,:) = sol;
s = 3; %Numero de etapas del metodo
L = funL(m);
Y = zeros(s+1,2*m+2);
I = eye(2*m+2);

%Fijamos la condicion de parada del bucle while
while tt(n)<tf

    %En caso de pasarnos de tf, reducimos el paso
    if tt(n)+h>tf
        h = tf - tt(n);
    end
end

```

```

%Definimos la etapa inicial
Y(1,:) = xx(n,:);
N = funN(xx(n,:),ind,m);
NN(1,:) = N';

%Ejecutamos el resto de etapas
for i=2:s+1
    Y(i,:) = (I-h*A(i,i)*L)\(xx(n,:) +...
        h*(A(i,1:i-1)*(L*Y(1:i-1,:))' +...
        Ahat(i,1:i-1)*NN(1:i-1,:))');
    N = funN(Y(i,:),ind,m);
    NN(i,:) = N';
end
%Vamos a guardar la solucion de forma provisional, pues la
%necesitamos para comprobar si el paso que hemos dado es lo
%suficientemente pequeno
xx(n+1,:) = xx(n,:);
for i=1:s+1
    xx(n+1,:) = xx(n+1,:) + h*(bb(i)*(L*Y(i,:))' +...
        bb(i)*NN(i,:));
end

%Ahora vamos a calcular el estimativo del error
yy = ((I-h*lamb*L)\(xx(n,:) + h*(bb(2)*((L*Y(2,:))' +...
    NN(2,:)) + bb(3)*((L*Y(3,:))' + NN(3,:)) + lamb*NN(4,:))')');

%Con la norma infinito no nos influye el tamaño de la malla
%(espacio)
EST = norm(xx(n+1:)-yy, "inf");
hopt = h*(tol/EST)^(1/3); %Porque el orden del estimador es 2

if EST<tol
    %En este caso, aceptamos el paso y nos quedamos con la
    %solucion que habiamos calculado antes
    tt(n+1) = tt(n) + h;
    n = n+1;
    h = 0.9*hopt;

else
    %Rechazamos el paso
    h = 0.9*hopt;
    %Al no aumentar la n, la siguiente iteracion del bucle va a
    %reescribir la anterior solucion
    continue
end
end
end

%Funcion que recoge los datos de los tableros de Butcher
function [A,Ahat,bb,c,lamb] = tableroRK()

lamb = 4.358665215084597e-01;
r = (-3/2)*lamb^2 + 4*lamb -1/4;
s = (3/2)*lamb^2 - 5*lamb + 5/4;
t = -0.35;
u = (1/3 - 2*lamb^2 - 2*s*(-0.35)*lamb)/(lamb*(1-lamb));

```

```

A = zeros(4);
Ahat = zeros(4);

A(2,2)=lamb;
A(3,2)=(1-lamb)/2; A(3,3)=lamb;
A(4,2)=r; A(4,3)=s; A(4,4)=lamb;

bb = [0;r;s;lamb];

Ahat(2,1)=lamb;
Ahat(3,1)=(1+lamb)/2-t; Ahat(3,2)=t;
Ahat(4,2)=1-u; Ahat(4,3)=u;

c = [0;lamb;(1+lamb)/2;1];
end

%Vamos a programar una funcion que nos evalúe el lado derecho del sistema
%de ecuaciones diferenciales y'=Ly+N(y)
function L = funL(m)
global e d hh
L = zeros(2*m+2);
for j=2:m
    L(j,j) = -2*d;
    L(j,j-1) = d;
    L(j,j+1) = d;
    L(j+m+1,j+m+1) = -2*e;
    L(j+m+1,j+m) = e;
    L(j+m+1,j+m+2) = e;
end
L = (1/hh^2)*L;
end

function N = funN(sol,ind,m)
global a b g k l mu cc hh
N = zeros(2*m+2,1);
%Generamos los coeficientes a_j+1/2
aa = zeros(m,1);
for j=1:m
    aa(j) = (1/hh)*(sol(j+1) - sol(j));
end
if ind==1
    for j=2:m
        N(j) = -l*sol(j) - (a*sol(j+m+1)*sol(j))/(g + sol(j));
        N(j+m+1) = (k/hh)*(max(aa(j-1),0)*sol(j+m) + ...
            min(aa(j-1),0)*sol(j+m+1) - max(aa(j),0)*sol(j+m+1) - ...
            min(aa(j),0)*sol(j+m+2)) + ...
            mu*abs(sol(j+m+1))*(1-sol(j+m+1))*max(0,sol(j)-cc) - ...
            b*sol(j+m+1);
    end
elseif ind==2
    f = flux(sol,m,ind,@psi,aa);
    for j=2:m
        N(j) = -l*sol(j) - (a*sol(j+m+1)*sol(j))/(g + sol(j));
        N(j+m+1) = (k/hh)*(f(j-1) - f(j)) + ...
            mu*abs(sol(j+m+1))*(1-sol(j+m+1))*max(0,sol(j)-cc) - ...
            b*sol(j+m+1);
    end
end

```

```

else
    f = flux(sol,m,ind,@psi,aa);
    for j=2:m
        N(j) = -1*sol(j) - (a*sol(j+m+1)*sol(j))/(g + sol(j));
        N(j+m+1) = (k/hh)*(f(j-1) - f(j)) + ...
            mu*abs(sol(j+m+1))*(1-sol(j+m+1))*max(0,sol(j)-cc) -...
            b*sol(j+m+1);
    end
end

end

%Programa que calcula los f_{j+1/2} para el k-Esquema
function f = flux(sol,m,ind,psi,aa)
%Generamos los coeficientes theta_j para el k-scheme
theta = zeros(m+1,1);
s0 = 3*sol(m+2)-3*sol(m+3)+sol(m+4);
sf = 3*sol(2*m+2)-3*sol(2*m+1)+sol(2*m);
aux0 = [s0 sol(m+2:2*m+1)];
auxf = [sol(m+3:2*m+2) sf];
for j=1:m+1
    theta(j) = (sol(j+m+1)-aux0(j)+eps)/(auxf(j)-sol(j+m+1)+eps);
end

if ind==1
    psi=0;
elseif ind==2
    kappa = 1;
else
    kappa = 1/3;
end
f = zeros(m,1);
for j=1:m
    f(j) = max(aa(j),0)*(sol(j+m+1) + ...
        psi(theta(j),kappa)*(sol(j+m+2)-sol(j+m+1))) +...
        min(aa(j),0)*(sol(j+m+2) +...
        psi((1/theta(j+1)),kappa)*(sol(j+m+1)-sol(j+m+2)));
end
end

function p = psi(theta,kappa)
p = max(0,min([1 (1/4)*(1+kappa)+(1/4)*(1-kappa)*theta theta]));
end

```

A.2. MODELO DE INVASIÓN

```

function [] = kSchemeInv2(ind,m,tf)
%Definimos las variables que vamos a emplear como globales
global e hh d2 nn alpha ji

%Definimos la longitud de la discretizacion espacial
hh = 1/m;

%Definimos nuestro vector de soluciones

```

```

sol = zeros(3*m+3,1);

%Vamos a introducir las condiciones iniciales
for j=1:m+1
    if (j-1)*hh <= 0.25
        sol(j+m+1) = exp(-((j-1)*hh)^2/0.01);
    else
        continue
    end
end
for j=1:m+1
    sol(j) = 1-(1/2)*sol(j+m+1);
    sol(j+2*m+2) = (1/2)*sol(j+m+1);
end

%Introducimos las constantes
e = 0;
d2 = 0.001;
nn = 10;
alpha = 0.1;
ji = 0.005;

t0 = 0;
tf = 1; %A esta variable le damos el valor que nos interese
h0 = 1/10000; %Longitud de paso temporal inicial
tol = 1/100000000; %Tolerancia del error en la solucion
%Resolvemos el problema bajo estas condiciones
[xx,~] = RKLinImp(@funL,@funN,t0,tf,h0,tol,sol,ind,m);
espacio = [0:1/m:1];

%Representamos los resultados
figure(1)
clf
hold on
plot(espacio,xx(end,m+2:2*m+2),'-k');
xlabel('Space')
ylabel('n, c_1, c_2')
title('K-Scheme-Invasion')

plot(espacio,xx(end,1:m+1),'--r');
plot(espacio,xx(end,2*m+3:3*m+3),'-.b');
legend('densidad tumoral','ECM','MDE')
%axis([0 1 0 1])
hold off
end

%Funcion que implementa el metodo RKLI
function [xx,tt] = RKLinImp(funL,funN,t0,tf,h0,tol,sol,ind,m)
    %Importamos el metodo desde un programa auxiliar
    [A,Ahat,bb,~,lamb] = tableroRK();
    h = h0;
    tt(1) = t0;
    n = 1;
    xx(1,:) = sol';
    s = 3; %Numero de etapas del metodo
    L = funL(m);
    Y = zeros(s+1,3*m+3);

```



```

I = eye(3*m+3);

%Fijamos la condicion de parada del bucle while
while tt(n)<tf

    %En caso de pasarnos de tf, reducimos el paso
    if tt(n)+h>tf
        h = tf - tt(n);
    end

    %Definimos la etapa inicial
    Y(1,:) = xx(n,:);
    N = funN(xx(n,:),ind,m);
    NN(1,:) = N';

    %Ejecutamos el resto de etapas
    for i=2:s+1
        Y(i,:) = (I-h*A(i,i)*L)\(xx(n,:) +...
            h*(A(i,1:i-1)*(L*Y(1:i-1,:))' + ...
            Ahat(i,1:i-1)*NN(1:i-1,:))');
        N = funN(Y(i,:),ind,m);
        NN(i,:) = N';
    end
    %Vamos a guardar la solucion de forma provisional, pues la
    %necesitamos para comprobar si el paso que hemos dado es lo
    %suficientemente pequeno
    xx(n+1,:) = xx(n,:);
    for i=1:s+1
        xx(n+1,:) = xx(n+1,:) + h*(bb(i)*(L*Y(i,:))' +...
            bb(i)*NN(i,:));
    end

    %Ahora vamos a calcular el estimativo del error
    yy = ((I-h*lamb*L)\(xx(n,:) + h*(bb(2)*((L*Y(2,:))' +...
        +NN(2,:))+bb(3)*((L*Y(3,:))'+NN(3,:)) + lamb*NN(4,:))')');

    %Con la norma infinito no nos influye el tamaño de la malla
    %(espacio)
    EST = norm(xx(n+1:)-yy, "inf");
    hopt = h*(tol/EST)^(1/3); %Porque el orden del estimador es 2

    if EST<tol
        %En este caso, aceptamos el paso y nos quedamos con la
        %solucion que habiamos calculado antes
        tt(n+1) = tt(n) + h;
        n = n+1;
        h = 0.9*hopt;
    else
        %Rechazamos el paso
        h = 0.9*hopt;
        %Al no aumentar la n, la siguiente iteracion del bucle va a
        %reescribir la anterior solucion
        continue
    end
end
end
end
end

```

```

%Funcion que recoge los datos de los tableros de Butcher
function [A,Ahat,bb,c,lamb] = tableroRK()

    lamb = 4.358665215084597e-01;
    r = (-3/2)*lamb^2 + 4*lamb -1/4;
    s = (3/2)*lamb^2 - 5*lamb + 5/4;
    t = -0.35;
    u = (1/3 - 2*lamb^2 - 2*s*(-0.35)*lamb)/(lamb*(1-lamb));

    A = zeros(4);
    Ahat = zeros(4);

    A(2,2)=lamb;
    A(3,2)=(1-lamb)/2; A(3,3)=lamb;
    A(4,2)=r; A(4,3)=s; A(4,4)=lamb;

    bb = [0;r;s;lamb];

    Ahat(2,1)=lamb;
    Ahat(3,1)=(1+lamb)/2-t; Ahat(3,2)=t;
    Ahat(4,2)=1-u; Ahat(4,3)=u;

    c = [0;lamb;(1+lamb)/2;1];

end

%Vamos a programar una funcion que nos evalúe el lado derecho del sistema
%de ecuaciones diferenciales y'=Ly+N(y)
function L = funL(m)
    global e d2 hh
    L = zeros(3*m+3);
    for j=2:m
        L(j+m+1,j+m+1) = -2*e;
        L(j+m+1,j+m) = e;
        L(j+m+1,j+m+2) = e;
        L(j+2*m+2,j+2*m+2) = -2*d2;
        L(j+2*m+2,j+2*m+1) = d2;
        L(j+2*m+2,j+2*m+3) = d2;
    end
    L(m+2,m+2) = -2*e;
    L(m+2,m+3) = 2*e;
    L(2*m+2,2*m+1) = 2*e;
    L(2*m+2,2*m+2) = -2*e;
    L(2*m+3,2*m+3) = -2*d2;
    L(2*m+3,2*m+4) = 2*d2;
    L(3*m+3,3*m+2) = 2*d2;
    L(3*m+3,3*m+3) = -2*d2;
    L = (1/hh^2)*L;
end
function N = funN(sol,ind,m)
    global hh ji nn alpha
    N = zeros(3*m+3,1);
    %Generamos los coeficientes a_j+1/2
    aa = zeros(m+2,1);
    for j=1:m
        aa(j+1) = (1/hh)*(sol(j+1) - sol(j));
    end

```

```

end
aa(1) = (1/hh)*(sol(1) - sol(2));
aa(m+2) = (1/hh)*(sol(m) - sol(m+1));
if ind==1
    for j=1:m+1
        N(j) = -nn*sol(j)*sol(j+2*m+2);
        N(j+m+1) = (ji/hh)*(max(aa(j),0)*sol(j+m+1) +...
            min(aa(j),0)*sol(j+m+1) - max(aa(j+1),0)*sol(j+m+2) -...
            min(aa(j+1),0)*sol(j+m+2));
        N(j+2*m+2) = alpha*sol(j+m+1);
    end
elseif ind==2
    f = flux(sol,m,ind,@psi,aa);
    for j=1:m+1
        N(j) = -nn*sol(j)*sol(j+2*m+2);
        N(j+m+1) = (ji/hh)*(f(j) - f(j+1));
        N(j+2*m+2) = alpha*sol(j+m+1);
    end
else
    f = flux(sol,m,ind,@psi,aa);
    for j=1:m+1
        N(j) = -nn*sol(j)*sol(j+2*m+2);
        N(j+m+1) = (ji/hh)*(f(j) - f(j+1));
        N(j+2*m+2) = alpha*sol(j+m+1);
    end
end
end
end

%Programa que calcula los f_{j+1/2} para el k-Esquema
function f = flux(sol,m,ind,psi,aa)
%Generamos los coeficientes theta_j para el k-scheme
theta = zeros(m+1,1);
s0 = 4*sol(m+3)-3*sol(m+2);
sf = 4*sol(2*m+1)-3*sol(2*m+2);
aux0 = [s0 sol(m+3) sol(m+2:2*m+2)];
auxf = [sol(m+2:2*m+2) sol(2*m+1) sf];

for j=1:m+1
    theta(j+1) = (sol(j+m+1)-aux0(j+1)+eps)/(auxf(j+1)-sol(j+m+1)+eps);
end
theta(1) = (sol(m+3)-aux0(1)+eps)/(auxf(1)-sol(m+3)+eps);
theta(m+3) = (sol(2*m+1)-aux0(m+2)+eps)/(auxf(m+2)-sol(2*m+1)+eps);

if ind==1
    psi=0;
elseif ind==2
    kappa = 1;
else
    kappa = 1/3;
end
f = zeros(m+2,1);
for j=1:m
    f(j+1) = max(aa(j+1),0)*(sol(j+m+1) +...
        psi(theta(j+1),kappa)*(sol(j+m+2)-sol(j+m+1))) +...
        min(aa(j+1),0)*(sol(j+m+2) +...
        psi(1/theta(j+2),kappa)*(sol(j+m+1)-sol(j+m+2)));
end
end

```

```

f(1) = max(aa(1),0)*(sol(m+3) +...
    psi(theta(1),kappa)*(sol(m+2)-sol(m+3))) + ...
    min(aa(1),0)*(sol(m+2) + ...
    psi((1/theta(2)),kappa)*(sol(m+3)-sol(m+2)));
f(m+2) = max(aa(m+2),0)*(sol(2*m+2) + ...
    psi(theta(m+2),kappa)*(sol(2*m+1)-sol(2*m+2))) + ...
    min(aa(m+2),0)*(sol(2*m+1) + ...
    psi((1/theta(m+3)),kappa)*(sol(2*m+2)-sol(2*m+1)));
end

function p = psi(theta,kappa)
    p = max(0,min([1 (1/4)*(1+kappa)+(1/4)*(1-kappa)*theta theta]));
end

```