

# Trabajo Fin de Máster Máster en Ingeniería Aeroespacial

## Análisis y optimización de métodos de colocación para problemas de convección de Rayleigh-Bénard

Autor: Alberto Damas Liébana

Tutor: Miguel Pérez-Saborid Sánchez-Pastor

**Dep. Ingeniería Aeroespacial y Mecánica de Fluidos  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla**

Sevilla, 2018





Trabajo Fin de Máster  
Máster en Ingeniería Aeroespacial

**Análisis y optimización de métodos de  
colocación para problemas de convección de  
Rayleigh-Bénard**

Autor:

Alberto Damas Liébana

Tutor:

Miguel Pérez-Saborid Sánchez-Pastor

Dep. Ingeniería Aeroespacial y Mecánica de Fluidos  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2018





Trabajo Fin de Máster: Análisis y optimización de métodos de colocación para problemas de convección de Rayleigh-Bénard

Autor: Alberto Damas Liébana  
Tutor: Miguel Pérez-Saborid Sánchez-Pastor

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Resumen

---

El objetivo de este proyecto ha sido el desarrollo y optimización de métodos de resolución, implementados en *Matlab*, para problemas de convección de Rayleigh-Bénard y de convección forzada en recintos  $2D$ . Ambos problemas pueden ser tratados de la misma manera gracias a la forma en que se imponen las condiciones de contorno. Se ha preferido un enfoque didáctico, que ayude a entender la complejidad para resolver problemas dominados por las ecuaciones de Navier-Stokes, en lugar del uso de algún software tipo caja negra. Se espera que el proyecto sea capaz de hacer patente todo lo que incluye este tipo de software de manera interna. El proceso general seguido a lo largo del proyecto se ha basado en la validación y optimización de las diferentes partes que componen el método, comparando con la bibliografía existente. Posteriormente, se procede a la obtención de resultados de interés, que ayuden a entender el fenómeno en estudio.

Se ha utilizado el modelo de Boussinesq, formulado en temperatura y función de corriente, para simplificar las ecuaciones de Navier-Stokes. Se aplican estas a todos los nodos del dominio, resultando, a priori, un sistema de ecuaciones con dos incógnitas por cada nodo.

Las geometrías simuladas consisten en dominios rectangulares  $2D$ , sobre el que se han añadido dos variantes: contornos verticales u horizontales no rectos. Para ello se ha realizado un cambio de variable que permite a partir de cualquiera estos dos obtener el primero.

Se ha buscado un enfoque universal que trate las derivadas como operadores matriciales y que permita aplicar las condiciones de contorno directamente en el sistema matricial resultante. De esta forma, pueden cambiarse los métodos para la obtención de las matrices de derivadas y aplicarse directamente sin realizar ningún otro cambio. Para la obtención de tales matrices se ha hecho uso de métodos de colocación con polinomios de Lagrange como funciones interpoladoras. Para el cálculo del valor en los nodos se han usado nodos de Chebyshev (todos los de la dirección en cuestión para buscar el mínimo error) o diferencias finitas de hasta sexto orden (aplicando el algoritmo de Fornberg, [20]) con diferentes distribución de nodos. A partir de estos métodos, se obtienen matrices válidas en una dirección, que hay que extrapolar teniendo en cuenta el cambio de variable al dominio  $2D$ . Aplicando los operadores matriciales obtenidos y sustituyendo las condiciones de contorno en el sistema que gobierna el problema, se llega al sistema algebraico que ha de resolverse. Para la integración temporal se hace uso de un método semi-implícito.

Obtenidas las ecuaciones y los métodos para su resolución, en primer lugar, se realiza un análisis de estabilidad lineal en torno a un estado de equilibrio con diversas condiciones de contorno. Este se usará para entender el concepto de número de Rayleigh crítico, así como para validar y optimizar los métodos. En segundo lugar se resolverán las ecuaciones a lo largo del tiempo para diferentes condiciones de contorno, entre las que se encuentran la inclusión de un caudal.

Por último se indican posible líneas futuras para continuar el trabajo realizado en este proyecto.



# Abstract

---

The main aim of this project is the development and optimization of methods to solve Rayleigh-Bénard convection, as well as forced convection inside  $2D$  domains. Both problems can be solved in the same way thanks to how the boundary conditions are imposed. The project tries to highlight the difficulties to solve problems dominated by Navier-Stokes equations instead to using specialized software, which are usually similar to a 'black box'.

Along this project, the different parts of the methods have been validated using the bibliography as a reference. Then, they have been used to obtain several results in order to understand convection phenomenon.

Boussinesq model is applied to solve the Navier-Stokes equations, with temperature and stream function as variables. Applying these equations to all mesh nodes results in an equation system with two unknown variables per node.

The domains solved are  $2D$  rectangles. Although, vertical or horizontal boundaries can be solved as well by applying a change of variables to obtain a rectangular domain.

It is important to emphasize the universal approach to deal with derivated terms as matrix operators. This allows imposing boundary conditions directly in the matrix system resulted from applying Boussinesq model at the nodes. It does not depend on the way derivatives are calculated. In this project, a collocation method with Lagrange polynomials is used. The value at collocation points are calculated with Chebyshev nodes or using Fornberg algorithm ([20]) to obtain sixth-order finite differences. These two methods calculate matrixes for just one direction, so they have to be extrapolated to  $2D$  domain, taking into account a change of variable. Time derivative is solved by using a semi-implicit method. Applying these operators to the equation system leads to an algebraical matrix system, in wich boundary conditions are imposed by substituting the corresponding boundary equation (a row of the matrix).

First, a linear stability analysis have been made applying several boundary conditions. Critical Rayleigh number can be obtained from this, which helps to understand convection initiation. In addition, it can be used to optimize methods and finding strengths and weaknesses for each one of them. Then, multiple domains and boundary conditions are solved along time, including forced convection.

Finally, some future research lines are proposed to continue with this project.



# Índice

---

<i>Resumen</i>	I
<i>Abstract</i>	III
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación del proyecto	1
1.2. Convección natural y forzada : convección de Rayleigh-Bénard	2
1.3. Estructura del proyecto	6
<b>2. Aproximación de Boussinesq</b>	<b>9</b>
2.1. Ecuaciones generales de Navier-Stokes	9
2.2. Estado del arte	10
2.3. Aproximación básica de Boussinesq	10
<b>3. Métodos de resolución</b>	<b>15</b>
3.1. Discretización del dominio	16
3.1.1. Diferencias finitas con uso exclusivo de series de Taylor	16
3.1.2. Diferencias finitas con polinomios interpolantes de Lagrange ([20],B.5)	19
3.1.3. Método de colocación con polinomios interpolantes de Lagrange y distribución de Chebyshev (B.7)	22
3.1.4. Extensión de matrices de derivadas a dominios en dos dimensiones	23
3.1.5. Tratamiento y transformación del dominio	27
3.2. Integración temporal. Condición de Courant-Friedrich-Lewy	27
3.3. Esquema general de resolución	28
3.4. Consideraciones de carácter computacional para la optimización de los códigos	30
<b>4. Linealización y análisis de estabilidad. Número de Rayleigh crítico</b>	<b>33</b>
4.1. Número de Rayleigh crítico	34
4.2. Desarrollo de las ecuaciones linealizadas y condiciones de contorno	36
4.2.1. Condiciones de contorno	37
4.3. Resolución numérica del análisis de estabilidad	39
4.3.1. Obtención del número de Rayleigh crítico	39
4.3.2. Validación de los métodos para contornos rectos y análisis de sensibilidad de malla (B.8 y B.9)	40
Método de colocación con nodos de Chebyshev	40
Diferencias finitas	44
Comparativa de colocación con nodos de Chebyshev y con diferencias finitas	47
Problema de Rayleigh. Condiciones de contorno periódicas (B.10 y B.11)	50

4.4.	Validación de los métodos para contornos no rectos (B.12 y B.13)	53
<b>5.</b>	<b>Análisis completo de la solución</b>	<b>57</b>
5.1.	Desarrollo de las ecuaciones completas	57
5.1.1.	Condiciones de contorno	58
5.1.2.	Condiciones iniciales	60
5.2.	Validación del método semi-implícito	60
5.2.1.	Rayleigh crítico	61
5.2.2.	Evolución temporal (B.14 y B.15)	62
5.3.	Convección forzada	66
5.3.1.	Convección forzada con contornos rectos y condición de periodicidad en los laterales (B.16 y B.17)	67
	Análisis lineal para convección forzada con contornos rectos	67
	Análisis completo para convección forzada con contornos rectos	70
5.3.2.	Convección forzada con contorno inferior no recto y condiciones de periodicidad	77
<b>6.</b>	<b>Conclusiones y líneas futuras</b>	<b>83</b>
<b>Anexo A.</b>	<b>Cálculo de derivadas</b>	<b>87</b>
A.1.	Contornos horizontales no rectos	87
A.1.1.	Derivadas parciales respecto a variables $x$ y $z$	87
A.1.2.	Derivadas de $Z$ respecto a $X = x$	88
	Contorno inferior sinusoidal con longitud de onda $\lambda_b$ y amplitud $A_b$	88
	Contorno inferior basado en función gaussiana $(\mu, \sigma)$ con amplitud $A_b$	89
	Contorno superior sinusoidal con longitud de onda $\lambda_t$ y amplitud $A_t$	89
A.2.	Contornos verticales no rectos	90
A.2.1.	Derivadas parciales respecto a variables $x$ y $z$	90
A.2.2.	Derivadas de $Z$ respecto a $X = x$	91
	Contorno izquierdo sinusoidal con longitud de onda $\lambda_b$ y amplitud $A_b$	91
	Contorno izquierdo parabólico con vértice en el punto medio y amplitud en este punto $A_b$	92
	Contorno derecho sinusoidal con longitud de onda $\lambda_t$ y amplitud $A_t$	92
	Contorno derecho parabólico con vértice en el punto medio y amplitud en este punto $A_t$	92
<b>Anexo B.</b>	<b>Códigos</b>	<b>93</b>
B.1.	Definición de la geometría	93
B.2.	Cálculo de derivadas	95
B.3.	Análisis lineal	104
B.4.	Resolución del sistema completo para convección natural	115
B.5.	Resolución del sistema completo para convección forzada	118
B.6.	Funciones auxiliares	123
	<i>Índice de Figuras</i>	125
	<i>Índice de Tablas</i>	127
	<i>Índice de Códigos</i>	129
	<i>Bibliografía</i>	131



# 1 Introducción

---

## 1.1 Motivación del proyecto

En cualquier disciplina ingenieril, a medida que la complejidad de los fenómenos que se estudian aumenta, también lo hacen los modelos y métodos para obtener una solución de aquello que se quiere resolver. Los recursos disponibles varían enormemente, en función del ámbito profesional en el que se quiere estudiar dicho fenómeno (tiempo, software, hardware, etc).

En la actualidad existen herramientas con la complejidad necesaria para abordar grandes problemas ingenieriles, pero en no muchas se permite el uso sin licencia. Además, el funcionamiento interno de estas no está al alcance del usuario con toda la claridad posible (software de tipo caja negra). Esto puede suponer un gran problema tanto para nuevos usuarios, que pueden no entender las bases físicas y herramientas matemáticas detrás de los diferentes modelos, como para usuarios más avanzados, que necesitan modificar el código para conseguir sus objetivos. Concretamente, para el ámbito de la Mecánica de Fluidos, existen diversas opciones especializadas de este tipo, como pueden ser *Fluent* o *Star-CCM+*. Frente a estas, existen otras de código abierto como *OpenFoam*, que permiten que los usuarios programen todo lo que les sea necesario, accediendo a los niveles más básicos. Sin embargo, la curva de aprendizaje para usar estas herramientas puede llegar a ser excesivamente lenta en las fases iniciales, y dado el alto grado de especialización, no suele ser una opción viable para la docencia.

Este punto es una de las clave del proyecto. En el ámbito de cualquier ingeniería, la resolución de problemas no triviales puede requerir de una gran cantidad de horas lectivas. Esto hace que se acaben resolviendo solo casos triviales sin mucho interés práctico, o aplicando modelos demasiado simples para plasmar la física de un fenómeno en su totalidad. Para suplir esto no basta con mostrar resultados obtenidos con métodos complejos al alumnado, puesto que en gran parte de las ocasiones, este pierde mucha información importante o no fija las ideas al no formar parte de su resolución. Especialmente en la Mecánica de Fluidos esto es aún más claro que en otras disciplinas, ya que las ecuaciones de Navier-Stokes no tienen por lo general solución analítica, salvo que se apliquen a problemas excesivamente sencillos y de poco interés práctico.

Los objetivos del proyecto son el estudio y la optimización de problemas de convección, natural o forzada, que combina la Mecánica de Fluidos y la Transferencia de Calor. Para ello no se hará uso de herramientas comerciales específicas, sino de una genérica de cálculo numérico como *Matlab*. Este proyecto mostrará por tanto una forma de resolver problemas con cierta complejidad de una manera muy didáctica, que permitiría al alumno entender el fenómeno físico, así como programar su propio método de resolución. Esto aportaría conocimientos suficientes para comprender el esquema básico de resolución de este tipo de problemas, otorgándole las bases necesarias para poder tratar

con herramientas tipo caja negra, fundamentales para abordar fenómenos más complejos. Por otro lado, también le otorgaría un conocimiento que podría extrapolar a cualquier otro fenómeno físico, dándole métodos y esquemas para su resolución. Esto podría incluir problemas de flujo potencial alrededor de perfiles o alas, capas límite viscosa y térmica, fenómenos aeroelásticos, etc.

Existe otro plano igualmente importante en la docencia: la motivación y el interés del alumno. La posibilidad de resolver un problema de cierta complejidad desde cero es mucho más gratificante que aplicar modelos sencillos, basados únicamente en aproximaciones y sustituciones triviales, que no suponen ningún reto ni interés para el alumno.

Por otro lado, se va a hacer hincapié no solo en la resolución, sino también en la optimización. La capacidad de computación crece de manera exponencial, lo que puede llevar a pensar que la eficiencia de las herramientas no es tan importante. Sin embargo, a medida que los problemas que se quieren resolver son más complejos, también lo son los métodos, y por tanto requieren más tiempo y recursos. En definitiva, la capacidad de reducir estos recursos al mínimo es igualmente importante. Esto es muy claro en sistemas cuyos tiempos de respuesta deben ser muy pequeños o cuyo peso debe lo menor posible, factor clave en aviación. Se le dará por ello especial importancia a la optimización de los cálculos.

En resumen, el objetivo del proyecto es la resolución de fenómenos de convección, con vistas a una posible aplicación de las herramientas al ámbito docente, dando la importancia adecuada a la optimización numérica de los códigos.

## **1.2 Convección natural y forzada : convección de Rayleigh-Bénard**

Se denomina convección al mecanismo de transferencia de calor asociado al movimiento de un fluido. Si dicho movimiento no se debe a ningún factor externo, se denomina convección natural, y en caso contrario convección forzada.

El caso más usual es el de convección natural de un fluido en un campo gravitatorio con diferencia de densidad debido a gradientes térmicos, aunque también puede producirse por gradientes de concentración, por campos magnéticos o eléctricos con un fluido con carga eléctrica no nula. El primer caso es conocido como convección de Rayleigh-Bénard, y será objeto de estudio en este proyecto. Para que un fluido permanezca en reposo, debe existir no solo equilibrio mecánico sino también térmico. En efecto, si una porción de fluido tiene mayor temperatura que el fluido circundante, tiene menor densidad y por tanto, bajo la acción de un campo gravitatorio esta tenderá a subir, desplazando el fluido que se encuentra en la parte superior. Este último, al tener menor temperatura y ser más denso, descenderá, generándose así un movimiento en el seno del fluido.

Los primeros estudios acerca de la convección natural fueron realizados por Henri Bénard en torno a 1900. Realizó experimentos que consistían en una delgada capa de fluido con alta conductividad térmica sobre una placa metálica, la cual calentaba de manera uniforme. Por la parte superior, el líquido estaba libre, en contacto con el aire del ambiente. Observó que alcanzando cierta temperatura en la placa se empezaban a generar movimientos en el fluido, que pueden describirse mediante lo que se denominan como celdas o rollos de Bénard. El fluido asciende por el interior de las celdas y desciende por los límites de estas, las cuales suelen ser hexagonales (ver figuras 1.1), aunque pueden llegar a ser cuadradas o triangulares bajo ciertas condiciones. Además, mediante diferentes técnicas consiguió observar una zona de baja presión en el centro de la celda, lo que justificó con la tensión superficial del fluido en contacto con el aire.

En 1916 Lord Rayleigh publicó una teoría ([2]) que estudiaba y explicaba los experimentos realizados por Bénard pocos años antes. En esta se ponía de manifiesto los factores a tener en cuenta para concluir la existencia o no de convección natural. No solo influyen las fuerzas de flotabilidad debidas a los gradientes térmicos, que originan diferentes densidades, sino también las fuerzas

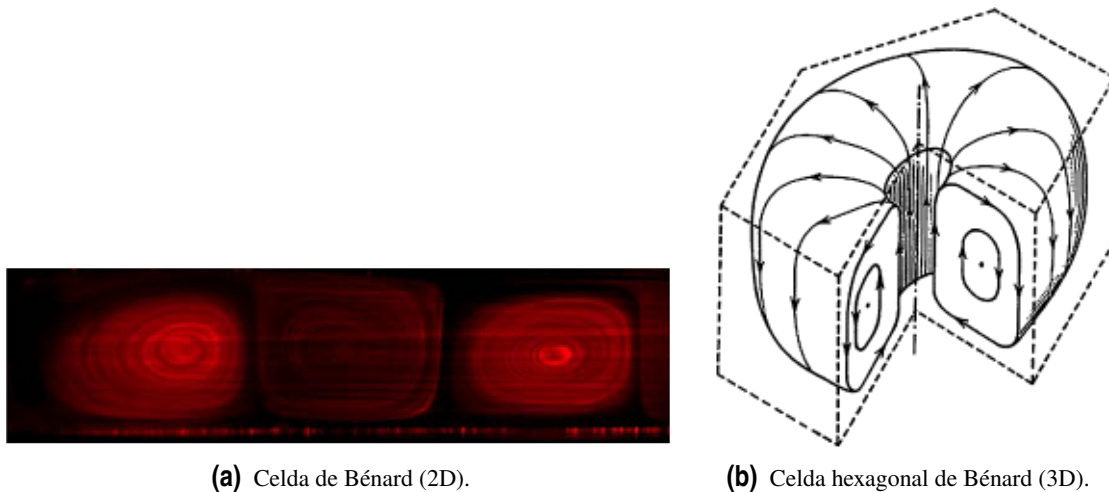


Figura 1.1 Celdas de Bénard [1].

viscosas y la conducción o difusión de calor (ver 1.2). Ambos tienden a contrarrestar el primer factor, ya sea por ser un efecto disipativo en el primer caso, o por tender a uniformizar el campo de temperaturas en el segundo. Por tanto, la existencia o no de convección natural viene determinada por un balance entre estos tres fenómenos. En el experimento realizado por Bénard, si la cantidad de calor no era lo suficientemente grande no existía convección, el único mecanismo de transferencia de calor se debía a conducción. En cambio, si esta superaba un umbral, se producía el movimiento del fluido y el inicio de la convección. El parámetro adimensional que mide la importancia relativa entre las fuerzas de flotabilidad y las fuerzas de viscosidad y conducción térmica se denomina número de Rayleigh (expresión 1.1), y su valor crítico determina para cada caso que fenómeno es predominante. En la expresión  $g$  es la gravedad,  $H$  la longitud del dominio fluido en la dirección de la gravedad,  $\beta$  el coeficiente de dilatación térmica,  $T_1$  y  $T_2$  las temperaturas máximas y mínimas respectivamente,  $\nu$  la viscosidad cinemática y  $\alpha$  la difusividad térmica.

$$Ra_c = \frac{gH^3\beta(T_1 - T_2)}{\nu\alpha} \quad (1.1)$$

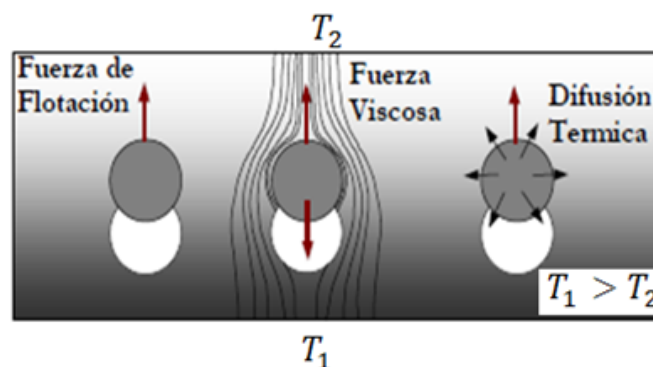


Figura 1.2 Mecanismos físicos que influyen en la convección natural.

Este mecanismo está presente en la naturaleza en multitud de fenómenos. Es de especial relevancia la convección en la meteorología ([3]), afectando en este ámbito a diferentes escalas. A escala planetaria (que incluye la global y la sinóptica) se generan tanto vientos en la superficie como verticales, y se deben a tres factores. Tanto la diferencia térmica entre la superficie terrestre y el

espacio, como el calentamiento de la tierra no homogéneo (la incidencia del sol es mayor en el ecuador), generan vientos ordenados en celdas convectivas. A estas dos causas hay que sumar la rotación de la tierra. En la imagen 1.3a y ?? se puede ver el modelo de tres celdas que incluye los tres factores (CP, CF y CH son las celdas polar, de Ferrel y de Hadley respectivamente). La celda de Ferrel, en la cual los vientos son de menor intensidad, se debe principalmente al arrastre generado por las otras dos, cuyos orígenes si que son puramente convectivos.

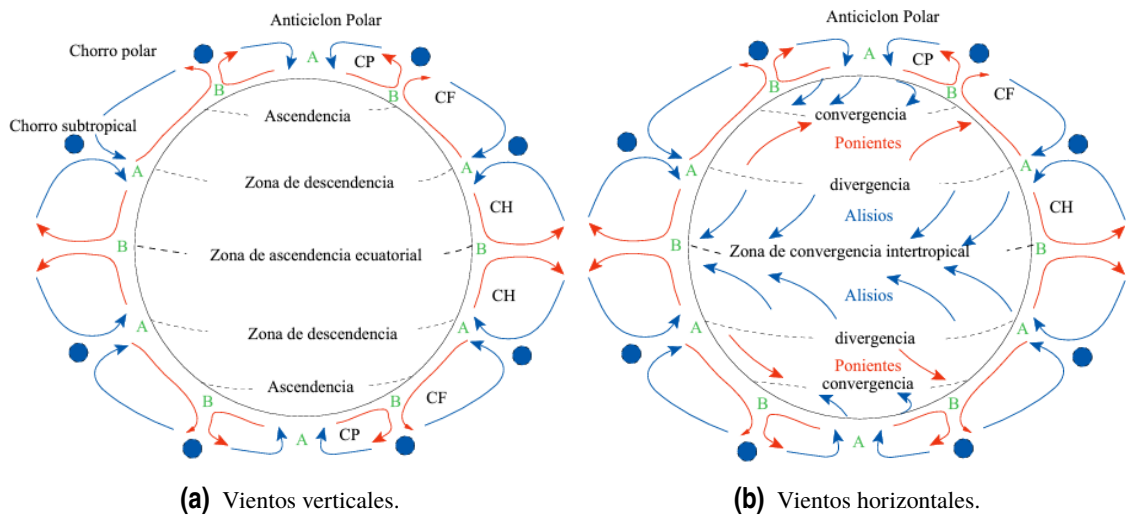


Figura 1.3 Modelo de tres celdas convectivas [3].

La convección afecta igualmente a una escala menor (mesoescala). En esta se encuentran las brisas costeras (ver 1.4, originadas por un mayor calor específico del agua, el cual provoca una acumulación de energía térmica durante el día y su liberación durante la noche) o de montaña (ver 1.5, en este caso se origina un gradiente térmico horizontal debido a la diferencia de temperatura entre la ladera y el aire adyacente). Un último ejemplo de brisa se produce en grandes ciudades, ya que la existencia de edificios permiten una mayor absorción térmica durante el día, que genera corrientes ascendentes por la noche, lo que a su vez produce vientos desde las zonas aledañas hacia el centro de la ciudad. Esto tiene relevancia en la acumulación de contaminación.

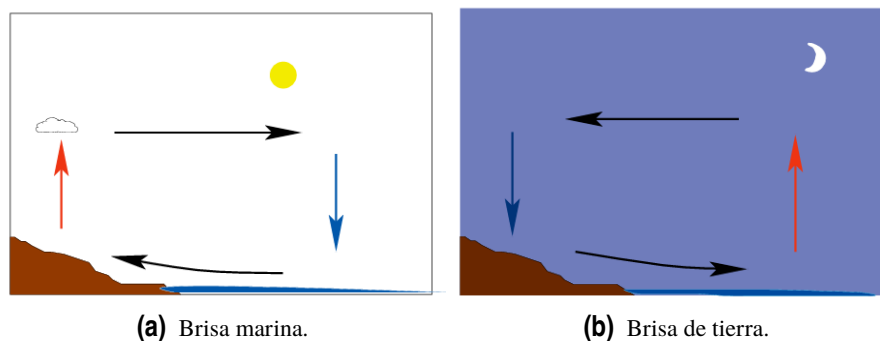
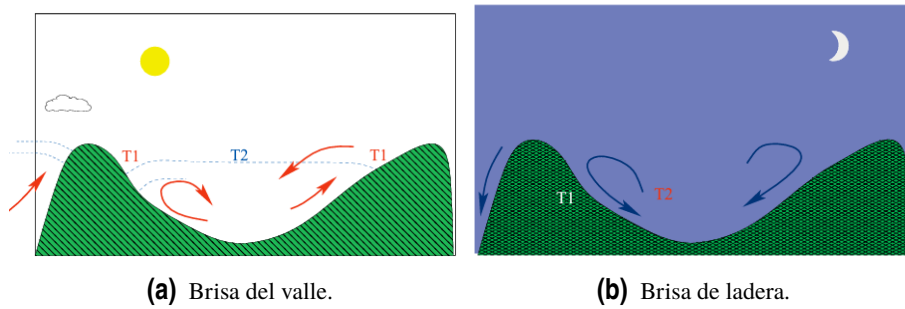


Figura 1.4 Brisas de costa [3].

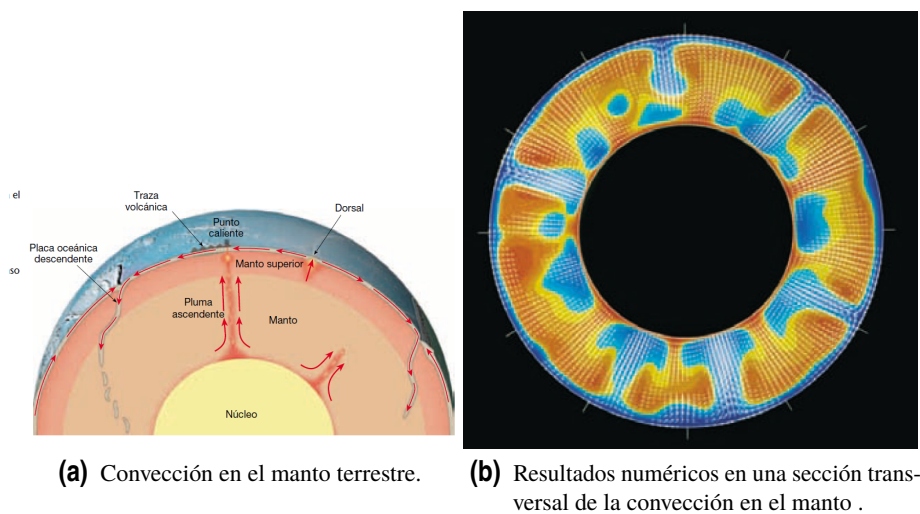
La convección también produce corrientes oceánicas, y tiene lugar en el interior de las estrellas o en el manto terrestre (ver 1.6). En 1.6b se puede ver la solución numérica de una sección transversal de este, en rojo se observan las zonas de mayor temperatura que generan corrientes ascendente



**Figura 1.5** Brisas de montaña [3].

y en azul las más frías con corrientes descendentes. Este fenómeno se produce en muy diversos ámbitos, al igual que el de convección forzada. Este último por ejemplo, tiene lugar en la circulación sanguínea para regular la temperatura corporal en animales de sangre caliente.

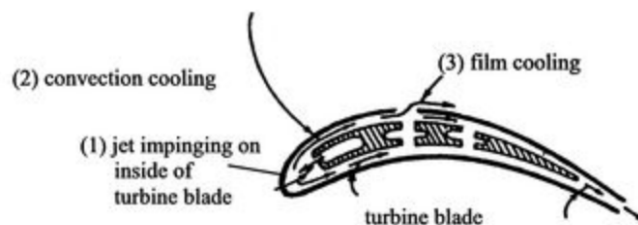
No solo es importante en la naturaleza, sino que se puede aprovechar en la industria. Claros ejemplos son los sistemas de acondicionamiento (calefacción o aire acondicionado) que se aprovechan de este fenómeno para mantener una temperatura más o menos uniforme en toda la estancia. También es de interés para refrigerar equipos electrónicos (aletas de convección), intercambiadores de calor, o en el ámbito de la aeronáutica para refrigerar turbinas de motores de aviación.



**(a)** Convección en el manto terrestre.

**(b)** Resultados numéricos en una sección transversal de la convección en el manto .

**Figura 1.6** Convección en el manto terrestre [4].



**Figura 1.7** Álabo de turbina refrigerado por convección [5].

Estos son solo algunos ejemplos que muestran la importancia de este mecanismo de transferencia de calor. Se encuentra presente en ámbitos tan complejos como el de la meteorología y tan cotidianos como calentar un cazo con agua. Por ello es importante profundizar en el funcionamiento y cuantificación de este, para entender tal cantidad de fenómenos distintos, permitiendo ello aprovechar este mecanismo al máximo.

### **1.3 Estructura del proyecto**

Una vez presentado el fenómeno que se va a estudiar se resume aquí la estructura que seguirá el proyecto. En el capítulo 2 se presenta el problema de convección desde un punto de vista histórico, en el que se resumen las aportaciones más importantes. Posteriormente se muestra el modelo de Boussinesq, señalando su rango de validez e hipótesis realizadas.

Obtenidas las ecuaciones del problema, el siguiente paso es detallar como se resolverán dichas ecuaciones en el capítulo 3. Esto incluye la discretización espacial, temporal y consideraciones respecto al cálculo matricial que se llevará a cabo. Es de especial interés el tratamiento matricial que se utilizará, ya que este es universal e independiente de la forma en que se hallen las derivadas. Sea cual sea el método de cálculo las derivadas se tratarán como operadores matriciales. Se usará un método de colocación con polinomios interpoladores de Lagrange con nodos de Chebyshev o diferencias finitas de hasta sexto orden. La diferencia entre ambos métodos es la base total o local de nodos que se usa respectivamente. A ello hay que añadir la manera en que se impondrán las condiciones de contorno ([6]), independiente también de cuales sean estas. Además, se introducirá un cambio de variable para resolver dominios con contornos no rectos en cualquiera de las dos direcciones.

En el capítulo 4 se procede a una linealización de las ecuaciones en torno al estado de equilibrio con su correspondiente descomposición y adimensionalización. Este enfoque ayudará a entender mejor la transición entre el estado de conducción y el de convección, produciéndose este último al surgir una inestabilidad. Con este análisis, el principal resultado que se obtendrá es el número de Rayleigh crítico, que determina cuando se produce tal inestabilidad, para diferentes condiciones de contorno. En base a este resultado se validarán los métodos escogidos. Para ello se comparan las soluciones obtenidas a partir de las ecuaciones linealizadas con las que se encuentran en la bibliografía. Por otra parte, se comparan los diferentes métodos usados en cuanto a tiempo y potencia de computación requeridos, obteniendo además criterios sobre que parámetros deben usarse en los métodos para que sean lo más óptimo posibles. Se resaltarán con especial interés la aplicación de condiciones de contorno periódicas con el fin de modelar el problema original de Rayleigh-Bénard. A diferencia de como suele hacerse en la bibliografía, expandiendo en senos y cosenos, en este caso se aprovecha el método de colocación imponiendo directamente que los valores de las magnitudes y sucesivas derivadas en los nodos de los contornos sean iguales. Por último se valida también el cambio de variable para modelar contornos no rectos.

Habiendo validado los métodos, en el capítulo 5 se resuelve ahora el problema completo para diferentes condiciones de contorno. Para ello se realizará otra adimensionalización a la llevada a cabo anteriormente. En primer lugar, para casos muy concretos, se obtendrá el valor de Rayleigh crítico para compararlos con los obtenidos en el capítulo 3, así como otros resultados para estudiar la importancia del paso temporal. De esta forma se valida también la discretización temporal y se obtienen criterios para su validez. Por último, se obtendrá la evolución temporal en diversos problemas de convección natural y forzada. En ellos se analizará entre otras cosas como afectan las condiciones de contorno en la aparición de la inestabilidad convectiva, la creación de las celdas de Bénard a lo largo del tiempo y flujo de calor que aparece en el contorno inferior.

Por último en el capítulo 6 se resumen las conclusiones obtenidas sobre los diferentes métodos

a lo largo del proyecto así como los resultados más básicos que ayudan a entender el fenómeno convectivo. Finalmente se proponen algunas líneas de investigación o mejoras posibles que ayudasen a mejorar los códigos.





## 2 Aproximación de Boussinesq

---

En general, las ecuaciones de Navier-Stokes deben resolverse numéricamente dada su complejidad. Además, en función del fenómeno físico que se modele en cada caso, es útil realizar hipótesis que las simplifiquen. Para fenómenos de convección, numerosos investigadores han realizado avances, desde modelos muy sencillos, hasta otros capaces de modelar fenómenos más complejos o con mayor precisión. En primer lugar se muestran las ecuaciones generales de Navier-Stokes, y a continuación se hace un resumen de las principales aportaciones en cuanto a la resolución de fenómenos convectivos. Posteriormente se detalla el modelo de Boussinesq con las hipótesis necesarias, y se reformula el problema en las variables que se van a resolver finalmente.

### 2.1 Ecuaciones generales de Navier-Stokes

Se muestran a continuación las ecuaciones generales de Navier-Stokes para un fluido homogéneo en su composición, viscoso y newtoniano (ver [7]). Este sistema de ecuaciones diferenciales en derivadas parciales servirá como punto de partida para aplicar una serie de hipótesis, con el fin de obtener unas ecuaciones simplificadas que modelen el problema de manera adecuada, conocido como modelo de Boussinesq. Los coeficientes que aparecen en las ecuaciones son función de la temperatura y la presión,  $\mu = \mu(T, p)$ ,  $\mu_v = \mu_v(T, p)$  y  $K = K(T, p)$ . Además de las ecuaciones de Navier-Stokes, son necesarias las ecuaciones de estado  $T = T(e, \rho)$  y  $p = p(e, \rho)$ .

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \quad (2.1)$$

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho \vec{v} \cdot \nabla \vec{v} = -\nabla p + \nabla \cdot \overline{\overline{\tau}} + \rho f_m \quad (2.2)$$

$$\rho \frac{\partial e}{\partial t} + \rho \vec{v} \cdot \nabla e = \nabla \cdot (K \nabla T) - p \nabla \cdot \vec{v} + \overline{\overline{\tau}} : \nabla \vec{v} + Q_r + Q_q \quad (2.3)$$

$$\text{siendo,} \quad \tau'_{ij} = 2\mu \gamma_{ij} + \left( \mu_v - \frac{2}{3}\mu \right) \nabla \cdot \vec{v} \delta_{ij} \quad (2.4)$$

$$\gamma_{ij} = \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad (2.5)$$

No se va a considerar en principio ninguna aportación de calor por unidad de tiempo (conducción, radiación o por reacción química) y por tanto  $Q_r = Q_q = 0$ .

## 2.2 Estado del arte

El modelo básico conocido como modelo de Boussinesq fue estudiado inicialmente por Oberbeck ([8],[9]), además del propio Boussinesq [10]. Este ha servido como base de numerosos modelos que amplían su validez o mejoran sus prestaciones.

La principal característica de dicho modelo es que supone un origen únicamente térmico (no mecánico) en los cambios de densidad del fluido. Además, tal variación, solo influye en el término que incluye las fuerzas gravitatorias en la ecuación de cantidad de movimiento. Esto solo es aplicable si las diferencias de temperatura son suficientemente pequeñas, y, por tanto, se pueden asumir las propiedades del fluido constantes. Por otro lado, también se supone la disipación viscosa despreciable.

Boussinesq intentó justificar su uso para un fluido compresible en fenómenos atmosféricos, argumentando que los cambios de densidad son despreciables en estos casos. En cambio, Rayleigh [2] aplicó estas suposiciones para estudiar las celdas de Rayleigh-Bénard en el caso de un líquido, siendo por tanto las hipótesis bastante razonables. Jeffreys [11] añadió más adelante un estudio para casos de movimientos infinitesimales.

Fueron Spiegel y Veronis [12] quienes, mediante un análisis de órdenes de magnitud para un gas, cuya ecuación de estado es la de un gas perfecto con propiedades constantes, obtuvieron un conjunto de ecuaciones equivalentes para fluidos compresibles. Para ello había que suponer que la altura característica del movimiento es mucho menor a la del estado de referencia. Esta suposición fue eliminada por Ogura y Phillips [13], que llegaron a un conjunto de ecuaciones (anelásticas, equivalentes a las deducidas por Batchelor [14]) que además permitían usar un paso de tiempo mayor para integrar las ecuaciones, ya que eliminaban la posibilidad de que existieran ondas acústicas. Esto fue confirmado por Charney y Ogura [15].

Mihaljan [16] supuso que la densidad varía linealmente con la temperatura, y el resto de propiedades constantes, para llegar a ecuaciones equivalentes a las de Boussinesq. Una generalización de estas, permitía una dependencia de la difusividad térmica y viscosidad con la temperatura y fue obtenida por Malkus [17]. Un gran avance fue el estudio llevado a cabo por Gray y Giorgini [18]. Ellos ampliaron el modelo para líquidos y gases permitiendo una dependencia de las propiedades, tanto con la temperatura como con la presión, dando además criterios de la validez del modelo. Se conocen como las ecuaciones de Boussinesq extendidas.

Más reciente es el estudio de Rajagopal [19], en el que deduce las ecuaciones de Boussinesq en un marco genérico partiendo de las ecuaciones de Navier-Stokes para un fluido cuya única hipótesis es la incompresibilidad mecánica.

En resumen, dado que es un fenómeno tan común en la naturaleza, suscita gran interés y han existido numerosos avances y propuestas para su modelado y resolución. Sin embargo, en su gran mayoría, parten de la aproximación básica de Boussinesq.

## 2.3 Aproximación básica de Boussinesq

Como ya se ha dicho, la aproximación de Boussinesq parte de suponer que las variaciones de densidad se deben únicamente a razones térmicas. Expandiendo inicialmente en series de Taylor la densidad en función de las variables de estado (temperatura y presión), y considerando la definición del coeficiente de dilatación térmica y de compresibilidad ( $\beta$  y  $K$ ) respectivamente resulta:

$$\frac{\rho - \rho_0}{\rho_0} = \frac{1}{\rho_0} \left( \left. \frac{\partial \rho}{\partial T} \right|_p (T - T_0) + \left. \frac{\partial \rho}{\partial p} \right|_T (p - p_0) \right) = -\beta(T - T_0) + K(p - p_0) \simeq -\beta(T - T_0) \quad (2.6)$$

Realizando un análisis de órdenes de magnitud ([6]) en las ecuaciones de cantidad de movimiento se tiene que:

$$v_0 \simeq \sqrt{gL\beta(T - T_0)} \quad (2.7)$$

$$p - p_0 \simeq \rho_0 g L \beta (T - T_0) \quad (2.8)$$

Teniendo en cuenta 2.8 se tiene que:

$$\frac{k(p - p_0)}{\beta(T - T_0)} \simeq k\rho_0 g L \quad (2.9)$$

Por tanto, cabe preguntarse cuándo este cociente es despreciable para fluidos típicos como puedan ser agua o aire a temperatura y presión ambiente.

$$\text{Aire} \Rightarrow \left\{ \begin{array}{l} k = \frac{1}{p_0} \simeq 10^{-5} \text{ m}^2/\text{N} \\ \rho_0 \simeq 1 \text{ kg/m}^3 \\ \beta \simeq \frac{1}{T_0} = \frac{1}{300} \text{ K}^{-1} \end{array} \right\} \Rightarrow \frac{k(p - p_0)}{\beta(T - T_0)} \ll 1 \quad \text{si } L \ll 10 \text{ km} \quad (2.10)$$

$$\text{Agua} \Rightarrow \left\{ \begin{array}{l} k = 0.5 \cdot 10^{-9} \text{ m}^2/\text{N} \\ \rho_0 \simeq 1000 \text{ kg/m}^3 \\ \beta \simeq 1.5 \cdot 10^{-4} \text{ K}^{-1} \end{array} \right\} \Rightarrow \frac{k(p - p_0)}{\beta(T - T_0)} \ll 1 \quad \text{si } L \ll 200 \text{ km} \quad (2.11)$$

Es decir, la densidad puede suponerse función exclusivamente de la temperatura para aire y agua en condiciones atmosféricas si  $L \ll 10$  y  $L \ll 200 \text{ km}$  respectivamente. Para que esto sea válido la diferencia de temperatura del agua no puede superar los  $100 \text{ K}$  para mantenerse líquida, y por tanto, que las propiedades dadas sigan siendo válidas. En el caso del aire, esta no puede superar  $30 \text{ K}$  típicamente, una diferencia mayor si que afecta a sus propiedades.

Realizando también un análisis de órdenes de magnitud en la ecuación de la energía se obtiene:

$$\frac{\overline{\tau'} : \nabla \vec{v}}{\rho c_p \vec{v} \cdot \nabla T} \simeq \frac{v}{c_p} \sqrt{\frac{g\beta}{L(T - T_0)}} \quad (2.12)$$

No obstante, Boussinesq justificó el modelo para una diferencia de temperatura en torno a  $10 \text{ K}$ . Acorde a esto, es razonable suponer las propiedades del fluido constantes:  $c_p$ ,  $\mu$  y  $K$ . Además, el

coeficiente de dilatación térmica  $\beta$  puede aproximarse por

$$\beta = \frac{1}{T}$$

Para los valores típicos de problema de convección natural (en el manto terrestre por ejemplo habría que añadir la disipación viscosa) este cociente es despreciable tanto para líquidos como para gases, por lo que se va a despreciar la disipación viscosa.

Tras este análisis, el sistema de ecuaciones diferenciales en derivadas parciales para dos dimensiones pasa a tener cuatro ecuaciones (continuidad, dos de cantidad de movimiento y de energía) y cuatro incógnitas ( $v_x, v_y, T$  y  $p$ ). De 2.13 a 2.16 se recoge el sistema de ecuaciones resultante con las hipótesis realizadas.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial z} = 0 \quad (2.13)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial z} = -\frac{1}{\rho_0} \frac{\partial p}{\partial x} + \frac{\mu}{\rho_0} \nabla^2 u \quad (2.14)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial z} = -\frac{1}{\rho_0} \frac{\partial p}{\partial z} - g(1 - \beta(T - T_0)) + \nu \nabla^2 v \quad (2.15)$$

$$\frac{\partial T}{\partial t} + \left( u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial z} \right) = \frac{K}{\rho_0 c_p} \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2} \right) - P \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial z} \right) \quad (2.16)$$

A continuación se expresan estas ecuaciones con una nueva variable: la función de corriente. Esta es una función escalar cuyas derivadas espaciales están relacionadas con las componentes de la velocidad.

$$u = v_x = \frac{\partial \psi}{\partial z} \quad (2.17)$$

$$v = v_z = -\frac{\partial \psi}{\partial x} \quad (2.18)$$

Expresando las ecuaciones en términos de esta nueva variable y teniendo en cuenta que las magnitudes son suficientemente regulares, el orden en que se realicen las derivadas no influye en el resultado. De esta forma, la ecuación de continuidad es idénticamente nula, y por tanto, resulta una ecuación y una incógnita menos al aplicar este cambio. Por otro lado, se deriva 2.14 respecto a  $z$  y 2.15 respecto a  $x$  y se combinan restando la segunda a la primera.

Quedan por tanto un sistema de dos ecuaciones diferenciales en derivadas parciales con dos incógnitas,  $\psi$  y  $T$ , con derivadas de hasta cuarto orden en la primera variable y de segundo en la última.

$$\frac{\partial}{\partial t} (\nabla^2 \psi) + \frac{\partial \psi}{\partial z} \frac{\partial}{\partial x} (\nabla^2 \psi) - \frac{\partial \psi}{\partial x} \frac{\partial}{\partial z} (\nabla^2 \psi) = \nu \nabla^4 \psi - g\beta \frac{\partial T}{\partial x} \quad (2.19)$$

$$\frac{\partial T}{\partial t} + \left( \frac{\partial \psi}{\partial z} \frac{\partial T}{\partial x} - \frac{\partial \psi}{\partial x} \frac{\partial T}{\partial z} \right) = \alpha \nabla^2 T \quad (2.20)$$

Estas ecuaciones, que representan la aproximación básica de Boussinesq, son las que se resolverán en los siguientes capítulos. Se resumen a continuación las hipótesis que el modelo tiene en cuenta.

- La densidad es función de la temperatura exclusivamente.
- Las variaciones de densidad son pequeñas y solo se consideran en el término de fuerzas másicas, aproximándose en el resto de términos por un valor constante.
- Las propiedades del fluido se consideran constantes dado que las variaciones de temperatura son pequeñas.
- Se desprecia la disipación viscosa.
- La velocidad en el campo fluido es mucho menor que la del sonido.
- La longitud vertical es mucho menor que la escala de altura hidrostática de presión, temperatura y densidad.

En definitiva, este modelo no es capaz de tener en cuenta una estratificación considerable de la densidad. Es recomendable el uso de un modelo más complejo como el anelástico para fenómenos atmosféricos a gran escala o para las capas más externas de las estrellas. En cambio, el modelo Boussinesq si es válido para resolver los fenómenos convectivos propios de capas no muy profundas de la atmósfera, océanos, el manto terrestre o capas intermedias de las estrellas.



### 3 Métodos de resolución

---

Los problemas que se van a resolver consisten en un sistema de ecuaciones diferenciales en derivadas parciales, siendo fundamentales los métodos numéricos que se utilizarán para la resolución de estos. Concretamente, se discutirá sobre la discretización del dominio, la forma de calcular derivadas espaciales y temporales con todos los cálculos matriciales necesarios, y por último, sobre el cambio de variable que permite resolver contornos no rectos. El objetivo es un código lo más versátil posible, que no dependa de la geometría del dominio ni de su mallado, y que se pueda modificar fácilmente para mejorar la precisión del cálculo.

El método utilizado será de colocación utilizando interpolantes de Lagrange, es decir, las funciones incógnitas se expresaran como combinación lineal de polinomios de Lagrange. Para hallar los coeficientes de estas funciones se particularizan estas y sus sucesivas derivadas, en puntos del dominio, y se introducen en las ecuaciones que gobiernan el problema y en las condiciones de contorno. Tales puntos se denominan puntos de colocación (o nodos). En este proyecto se usará un número de nodos global o local. En el primer caso se hará uso de nodos de Chebyshev, y tendrá en cuenta todos los nodos de la dirección en cuestión para el cálculo de las derivadas. En el segundo caso, se hará uso de diferencias finitas, las cuales solo tendrán en cuenta un número menor de nodos locales para el cálculo de las derivadas.

Una vez se ha discretizado el dominio, aplicando el método, el problema se reduce a resolver el valor de las funciones incógnita en los nodos. Se busca expresar este como un sistema matricial de ecuaciones algebraicas. Para ello se buscarán matrices, usando colocación con nodos de Chebyshev o diferencias finitas, que permitan expresar las derivadas espaciales de la manera adecuada (3.1).

$$f^I(\bar{x}) = \frac{\partial f}{\partial x}(\bar{x}) = \begin{bmatrix} f^I(x_1) \\ f^I(x_2) \\ \vdots \\ f^I(x_N) \end{bmatrix} = L_x^I f(\bar{x}) \quad (3.1)$$

Como ya se ha comentado, las dos principales diferencias entre ambos métodos son el carácter disperso o no de las matrices y la manera en que calcular derivadas de segundo orden o superior. En cuanto a la primera, las matrices son dispersas en caso de usar diferencias finitas (ya que las derivadas en un nodo solo dependen de los nodos circundantes), lo cual supone una gran ventaja en cuanto a tiempo y memoria requerida de computación. Sin embargo, la distribución de nodos de Chebyshev ofrece una solución óptima en cuanto a precisión, ya que le caracteriza su convergencia superalgebraica. La segunda diferencia reside en la forma de hallar las derivadas de orden dos o superior. En caso de usar nodos de Chebyshev se calcularán como producto de derivadas primera,

mientras que, en caso de usar diferencias finitas, se hallarán conjuntamente con las derivadas primeras, de la misma manera que estas.

### 3.1 Discretización del dominio

En primer lugar se va a analizar el método de colocación haciendo uso de diferencias finitas deduciendo las matrices de derivadas de 3.1 mediante series de Taylor. Los resultados obtenidos con este procedimiento presentan problemas numéricos propios de matrices de tipo Vandermonde que se encuentran mal condicionadas, mostrando los resultados inestabilidades que dependen claramente del número de nodos y la distribución de estas como se ha podido comprobar resolviendo casos muy simples. Por esta razón no se usará a lo largo del proyecto.

Un segundo método es el que se propone en [20], el cual parte de las series de Taylor y de polinomios de Lagrange. Este no presenta problemas numéricos y además está optimizado desde un punto de vista computacional. En función del número de nodos locales escogido las sucesivas derivadas tendrán una precisión mayor o menor. La precisión viene dada por la diferencia entre el número de nodos local utilizado menos uno y el orden de la derivada. Por ejemplo, las derivadas de primer orden tendrán una precisión de sexto orden en caso de utilizar siete nodos (incluido el nodo donde se calcula la derivada). Escogiendo este número de nodos la derivada de cuarto orden tendría una precisión de orden dos. Se estudiarán en el siguiente capítulo algunas distribuciones básicas de los nodos y del número de nodos locales escogidos.

A continuación se incluye el método de colocación con nodos de Chebyshev, en el que se usan todos los nodos de una dirección para calcular la derivada respectiva. La convergencia es óptima en cuanto a precisión para esta distribución, y por ello se usará esta siempre que se quieran tener en cuenta todos los nodos.

Por último se presenta el método que se aplicará para resolver contornos no rectos, que consiste en definitiva en un cambio de variable que transforma el dominio a uno con contornos rectos. El cálculo de las matrices de derivadas unidimensionales se realiza en un mallado rectangular en el dominio transformado  $X - Z$ . Posteriormente estas matrices son transformadas al dominio original  $x - z$ , y son las que se aplicarán en las ecuaciones que gobiernan el problema.

#### 3.1.1 Diferencias finitas con uso exclusivo de series de Taylor

Sea un dominio unidimensional dividido en  $N - 1$  partes y que por tanto tiene  $N$  nodos (figura 3.1). No es necesario que la malla sea uniforme, de manera que las longitudes entre nodos pueden ser diferentes ( $h_i$ ).



Figura 3.1 Dominio unidimensional.

Se busca un sistema matricial cuadrado que permita un cálculo sencillo de la matriz  $L$ . Para ello, es necesario que el número de puntos involucrados en el cálculo de la derivada en un nodo sea  $N_i$ , siendo el orden superior de la derivada a calcular  $N_i - 1$ . La deducción se va a realizar para el caso en que se quiere obtener hasta la derivada cuarta respecto a  $x$ . En este caso para calcular la derivada en cada punto de colocación habrá involucrados cinco nodos (el nodo en cuestión más cuatro nodos adyacentes). Cómo elegir los nodos que intervienen en el cálculo de la derivada se especificará más adelante.



En el código, dado que se busca que sea fácilmente modificable, se definen los siguientes parámetros y se da su valor para el ejemplo anterior.

- $N_i = 5$ , número total de puntos involucrados para el cálculo de una derivada.
- $N_d = 3$ , número de nodos involucrados a la derecha del nodo de cálculo.
- $N_l = 1$ , número de nodos involucrados a la izquierda del nodo de cálculo.

Por tanto, se empiezan planteando las siguientes series de Taylor, donde  $d_i$  puede ser positivo o negativo.

$$f(x + d_1) = f(x) + f^I(x)d_1 + f^{II}(x)\frac{d_1^2}{2} + f^{III}(x)\frac{d_1^3}{3!} + f^{IV}(x)\frac{d_1^4}{4!} + O(5) \quad (3.2)$$

$$f(x + d_2) = f(x) + f^I(x)d_2 + f^{II}(x)\frac{d_2^2}{2} + f^{III}(x)\frac{d_2^3}{3!} + f^{IV}(x)\frac{d_2^4}{4!} + O(5) \quad (3.3)$$

$$f(x + d_3) = f(x) + f^I(x)d_3 + f^{II}(x)\frac{d_3^2}{2} + f^{III}(x)\frac{d_3^3}{3!} + f^{IV}(x)\frac{d_3^4}{4!} + O(5) \quad (3.4)$$

$$f(x + d_4) = f(x) + f^I(x)d_4 + f^{II}(x)\frac{d_4^2}{2} + f^{III}(x)\frac{d_4^3}{3!} + f^{IV}(x)\frac{d_4^4}{4!} + O(5) \quad (3.5)$$

A continuación se van a expresar las series anteriores de manera matricial (3.6). Se van a suponer los términos de orden cinco o superior despreciables, siendo por tanto la precisión para derivada primera de tercer orden.

$$\begin{bmatrix} f(x + d_1) - f(x) \\ f(x + d_2) - f(x) \\ f(x + d_3) - f(x) \\ f(x + d_4) - f(x) \end{bmatrix} = \begin{bmatrix} d_1 & \frac{d_1^2}{2} & \frac{d_1^3}{3!} & \frac{d_1^4}{4!} \\ d_2 & \frac{d_2^2}{2} & \frac{d_2^3}{3!} & \frac{d_2^4}{4!} \\ d_3 & \frac{d_3^2}{2} & \frac{d_3^3}{3!} & \frac{d_3^4}{4!} \\ d_4 & \frac{d_4^2}{2} & \frac{d_4^3}{3!} & \frac{d_4^4}{4!} \end{bmatrix} \begin{bmatrix} f^I(x) \\ f^{II}(x) \\ f^{III}(x) \\ f^{IV}(x) \end{bmatrix} = \bar{d} \begin{bmatrix} f^I(x) \\ f^{II}(x) \\ f^{III}(x) \\ f^{IV}(x) \end{bmatrix} \quad (3.6)$$

Extendiendo este planteamiento para todos los nodos de la malla se puede plantear el sistema matricial 3.7.

$$\begin{bmatrix} f(x_1 + d_{11}) - f(x_1) \\ f(x_1 + d_{21}) - f(x_2) \\ f(x_1 + d_{31}) - f(x_3) \\ f(x_1 + d_{41}) - f(x_4) \\ \vdots \\ f(x_i + d_{1i}) - f(x_i) \\ f(x_i + d_{2i}) - f(x_i) \\ f(x_i + d_{3i}) - f(x_i) \\ f(x_i + d_{4i}) - f(x_i) \\ \vdots \\ f(x_N + d_{1N}) - f(x_N) \\ f(x_N + d_{2N}) - f(x_N) \\ f(x_N + d_{3N}) - f(x_N) \\ f(x_N + d_{4N}) - f(x_N) \end{bmatrix} = \begin{bmatrix} \bar{d}_1 & & & & \\ & \ddots & & & \\ & & \bar{d}_i & & \\ & & & \ddots & \\ & & & & \bar{d}_N \end{bmatrix} \begin{bmatrix} f^I(x_1) \\ f^{II}(x_1) \\ f^{III}(x_1) \\ f^{IV}(x_1) \\ \vdots \\ f^I(x_i) \\ f^{II}(x_i) \\ f^{III}(x_i) \\ f^{IV}(x_i) \\ \vdots \\ f^I(x_N) \\ f^{II}(x_N) \\ f^{III}(x_N) \\ f^{IV}(x_N) \end{bmatrix} \Rightarrow \overline{diff}_{4N \times 1} = \bar{D}_{4N \times 4N} \bar{Df}_{4N \times 1} \quad (3.7)$$

El primer miembro de la igualdad incluye, para cada nodo donde se quieren calcular las derivadas, el valor de la función en cuatro puntos adicionales. Estos cuatro puntos serán los nodos vecinos. Por tanto, este vector columna es necesario expresarlo como función de un vector columna formado por el valor de la función en los  $N$  nodos. Esto es posible hacerlo como se muestra en 3.8 para el caso en que la malla tiene cinco nodos. Para una malla genérica de  $N$  nodos y  $N_t$  a considerar, el razonamiento es el mismo. De esta manera, quedaría la relación matricial que se muestra en 3.9. Cabe señalar que se utilizarán dos nodos a la derecha y dos a la izquierda, excepto en los extremos del dominio. En este caso, se buscará la distribución más simétrica posible, completando hasta los  $N_t$  nodos por el lado interior del dominio. Por ejemplo, si se quiere calcular la derivada del nodo  $i = 2$ , se utilizará el nodo  $i = 1$  por la izquierda, y los nodos  $i = 3, 4, 5$  por la derecha.

$$\begin{bmatrix} f(x_1 + d_{11}) - f(x_1) \\ f(x_1 + d_{21}) - f(x_1) \\ f(x_1 + d_{31}) - f(x_1) \\ f(x_1 + d_{41}) - f(x_1) \\ \vdots \\ f(x_i + d_{1i}) - f(x_i) \\ f(x_i + d_{2i}) - f(x_i) \\ f(x_i + d_{3i}) - f(x_i) \\ f(x_i + d_{4i}) - f(x_i) \\ \vdots \\ f(x_N + d_{1N}) - f(x_5) \\ f(x_N + d_{2N}) - f(x_5) \\ f(x_N + d_{3N}) - f(x_5) \\ f(x_N + d_{4N}) - f(x_5) \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix} \Rightarrow \overline{diff}_{20 \times 1} = \overline{M}_{20 \times 5} \overline{f}_{5 \times 1} \quad (3.8)$$

$$\overline{diff}_{4N \times 1} = \overline{M}_{4N \times N_t} \overline{f}_{N_t \times 1} \quad (3.9)$$

Finalmente, para la obtención de las derivadas en los diferentes nodos, basta con seguir la expresión 3.10, obtenida igualando 3.9 y 3.7. Solo habría que seleccionar las componentes de  $\overline{Df}$  para obtener las matrices  $L_x^I, L_x^H, \dots, L_x^N$ .

$$\overline{M}_{4N \times N_t} \overline{f}_{N_t \times 1} = \overline{D}_{4N \times 4N} \overline{Df}_{4N \times 1} \Rightarrow \overline{Df}_{4N \times 1} = (\overline{D} \setminus \overline{M})_{4N \times N_t} \overline{f}_{N_t \times 1} \quad (3.10)$$

Como ya se ha comentado en la introducción, la matriz  $\overline{d}$  es una submatriz de la matriz de Vandermonde, y el cálculo de su inversa (en este caso el de  $\overline{D}$ ) es muy inestable numéricamente cuando  $N$  crece. Salvo para casos concretos, generalmente con  $N$  muy pequeño, Matlab genera un aviso advirtiendo de que la matriz está cerca de la singularidad y por tanto su inversa introduce grandes errores. Por ello, se ha implementado el método de la siguiente sección que será el que finalmente se utilice.

### 3.1.2 Diferencias finitas con polinomios interpolantes de Lagrange ([20],B.5)

Como en el caso anterior, se supone conocido el valor de una magnitud escalar cualquiera  $\phi(x)$  en  $N$  nodos. Se va a considerar un polinomio interpolador de Lagrange de esta función. No es necesario que esté basado en todos los nodos, basta con un subconjunto de ellos. El usuario definirá el mayor orden de derivada que quiere calcular,  $M$ . A su vez, establecerá el número de nodos,  $I$ , que formará el subconjunto para calcular las derivadas en un nodo (incluye el nodo donde se quiera calcular, por ello es recomendable que  $I$  sea impar). La selección de estos nodos se hará exactamente igual que en la sección anterior, escogiendo el mismo número de puntos a cada lado excepto si está lo suficientemente cerca del contorno para que esto no sea posible.

El algoritmo que se va a detallar a continuación tiene como objetivo el primer código del apéndice C de [20], el cual genera los valores de las derivadas requeridas (hasta orden  $M$ ) para varias precisiones. Para un mismo orden de derivada, a mayor número de nodos local que se utilicen, mayor será la precisión. En este caso, se denominará  $I$  al número de nodos circundantes. Este algoritmo no se ha aplicado típicamente en la literatura para resolver métodos de colocación y por ello resulta de interés realizar un estudio como este, que muestre su validez para este tipo de problemas. En este proyecto se aplicará este algoritmo y se buscarán las ventajas y desventajas respecto al método más conocido basado en nodos de Chebyshev.

Con todo ello se procede a explicar las ecuaciones a partir de las cuales se obtiene el algoritmo. El objetivo es encontrar para un punto dado  $X_0$  el valor de las derivadas, hasta orden  $M$ , de una función cuyo valor en los nodos es conocido. La idea del procedimiento reside en hacer el cálculo de manera progresiva aumentando el tamaño del subconjunto escogido, desde 1 a  $I$ . Por tanto, se va a definir un índice  $i$  cuyo valor irá de 1 a  $I$ .

El primer paso es definir un polinomio interpolador de Lagrange de la función  $\phi(x)$ , que viene dado por las expresiones 3.11 y 3.12.

$$p_i(x) = \sum_{j=1}^i f(x_j) F_{i,j}(x) \quad i = 1, 2, \dots, I \quad (3.11)$$

$$\begin{aligned} F_{i,j}(\chi + x) &= \frac{\prod_{k=1, j \neq k}^i (\chi + x - x_k)}{\prod_{k=1, j \neq k}^i (x_j - x_k)} = \\ &= \frac{(\chi + x - x_1)(\chi + x - x_2) \dots (\chi + x - x_{j-1})(\chi + x - x_{j+1}) \dots (\chi + x - x_i)}{(x_j - x_1)(x_j - x_2) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_i)} = \\ &= \begin{cases} 1 & \text{si } x = x_j \\ 0 & \text{si } x \neq x_j \end{cases} \end{aligned} \quad (3.12)$$

La derivada de orden  $k$  de la función  $\phi$  evaluada en  $x = \chi$  se puede aproximar por la expresión 3.13. Esta aproximación tendrá una mayor precisión cuanto mayor sea  $i$ .

$$\left. \frac{\partial^k p_i(x)}{\partial x^k} \right|_{x=\chi} = \sum_{j=1}^i \left. \frac{\partial^k F_{i,j}(x)}{\partial x^k} \right|_{x=\chi} f(x_j) = \sum_{j=1}^i c_{i,j,k}(\chi) f(x_j) \quad i = 1, 2, \dots, I \quad k = 1, 2, \dots, R \quad (3.13)$$

A partir de la nueva nomenclatura se puede expresar fácilmente la serie de Taylor de  $F_{i,j}(\chi)$  en función de sus derivadas  $c_{i,j,k}(\chi)$ , siendo  $k = 1, 2, \dots, R$ . El valor de  $R$  será el mínimo entre  $i$  y  $M + 1$ .

$$F_{i,j}(\chi + x) = \sum_{k=1}^R \frac{c_{i,j,k}(\chi)}{(k-1)!} x^{k-1} = \frac{c_{i,j,1}(\chi)}{0!} x^0 + \frac{c_{i,j,2}(\chi)}{1!} x^1 + \frac{c_{i,j,3}(\chi)}{2!} x^2 + \dots + \frac{c_{i,j,R}(\chi)}{(R-1)!} x^{R-1} \quad (3.14)$$

Igualando los coeficientes de las potencias de  $x$  de las expresiones 3.14 y ?? se obtienen las relaciones de 3.15 a 3.18 para un cálculo secuencial de los diferentes  $c_{i,j,k}$  y por tanto, de las derivadas de la función original en los nodos. La forma en que se realiza este cálculo es aumentando desde un solo nodo el tamaño del subconjunto  $1, 2, \dots, I$ . Para ello es interesante definir  $w_i(x) = (x_i - x_j)w_{i-1}(x)$ , válido para  $i = 2, 3, \dots, I$ , siendo  $w_1 = 1$ . Es importante señalar que aquellos coeficientes que no existan se toman iguales a cero para poder usar este método. No obstante, se han diferenciado las expresiones para el caso en que  $k = 1$ , ya que a la hora de programarlo la expresión más genérica supondría un error al intentar acceder a la componente cero de una matriz.

$$c_{i,j,k}(\chi) = \frac{1}{x_i - x_j} (x_i - \chi) c_{i-1,j,k}(\chi) \quad \text{si } i \neq j \text{ y } k = 1 \quad (3.15)$$

$$c_{i,j,k}(\chi) = \frac{1}{x_i - x_j} [(x_i - \chi) c_{i-1,j,k}(\chi) - (k-1) c_{i-1,j,k-1}(\chi)] \quad \text{si } i \neq j \text{ y } k > 1 \quad (3.16)$$

$$c_{i,i,k}(\chi) = -\frac{w_{i-2}}{w_{i-1}} (x_{i-1} - \chi) c_{i-1,i-1,k} \quad \text{si } k = 1 \quad (3.17)$$

$$c_{i,i,k}(\chi) = \frac{w_{i-2}}{w_{i-1}} [(k-1) c_{i-1,i-1,k-1} - (x_{i-1} - \chi) c_{i-1,i-1,k}] \quad \text{si } k > 1 \quad (3.18)$$

Para entender como se obtienen estas relaciones, ya que no es inmediata su obtención, se propone a continuación la particularización en caso de que  $I = 3$  y siendo  $\min(I, M+1) = I$ . Se va a omitir la dependencia del punto de evaluación de los coeficientes  $c_{i,j,k}(\chi)$  por comodidad, ya que todos ellos se calculan en este punto. En este caso, se empieza tomando un subconjunto, con  $i = 1$ , después con  $i = 2$  y finalmente con  $i = I = 3$ . Cada vez que se toma uno de estos subconjuntos,  $j$  varía de 1 a  $i$ . A su vez,  $k$  varía de 1 a  $R$ . Sabiendo los resultados que se obtienen se distinguen las siguientes dependencias:

$$\left. \begin{array}{l} c_{i-1,j,k} \\ c_{i-1,j,k-1} \end{array} \right\} \Rightarrow c_{i,j,k} \quad \text{si } i \neq j \text{ y } k \neq i \text{ y } k \neq 1 \quad (3.19)$$

$$c_{i-1,j,k} \Rightarrow c_{i,j,k} \quad \text{si } i \neq j \text{ y } k = 1 \quad (3.20)$$

$$\left. \begin{array}{l} c_{i-1,j,k} = 0 \\ c_{i-1,j,k-1} \end{array} \right\} \Rightarrow c_{i,j,k} \quad \text{si } i \neq j \text{ y } k = i \quad (3.21)$$

$$\left. \begin{array}{l} c_{i-1,i-1,k} \\ c_{i-1,i-1,k-1} \end{array} \right\} \Rightarrow c_{i,j,k} \quad \text{si } i = j \text{ y } k \neq i \text{ y } k \neq 1 \quad (3.22)$$

$$c_{i-1,i-1,k} \Rightarrow c_{i,j,k} \quad \text{si } i = j \text{ y } k = 1 \quad (3.23)$$

$$\left. \begin{array}{l} c_{i-1,i-1,k} = 0 \\ c_{i-1,i-1,k-1} \end{array} \right\} \Rightarrow c_{i,j,k} \quad \text{si } i = j \text{ y } k = i \quad (3.24)$$

Por otro lado, se van a particularizar ahora las expresiones 3.14 y 3.12 para el caso propuesto, igualándolas para obtener los diferentes coeficientes. Se parte de  $c_{1,1,1} = 1$ . Relacionando todos los coeficientes calculados en las expresiones de 3.25 a 3.30 acorde a las relaciones de precedencia descritas anteriormente de 3.19 a 3.24 se pueden verificar de 3.15 a 3.18.

El código anexo (B.5) sigue la misma nomenclatura y por tanto genera una matriz de tres dimensiones para los coeficientes  $c_{i,j,k}$  para cada punto en que se quieran calcular las derivadas ( $x = \chi$ ). El tercer índice ( $k$ ) indica el orden de la derivada, de manera que si vale 1 la derivada es de orden cero (aproximación polinómica de la función), y si vale 5 la derivada es de orden 4. Dado  $k$ , cada columna indica la precisión de los cálculos, que por lo general aumenta con el número de puntos escogidos para el cálculo (siempre que el orden de derivada  $M$  que se quiera calcular sea mayor que  $I$ ). En cualquier caso, la última columna será la de mayor precisión. En cuanto al primer índice ( $i$ ), las filas obtienen los coeficientes de cada uno de los nodos vecinos considerados. Por tanto, para obtener la matriz para un orden dado de derivada, basta con coger la columna de una precisión dada y ubicar sus valores en la fila correspondiente al nodo de cálculo, y en aquellas columnas que multipliquen a los nodos vecinos de este.

$$i = 1, \quad j = 1, \quad c_{1,1,1} = 1 \quad (3.25)$$

$$i = 2, \quad j = 1, \quad F_{2,1} = c_{2,1,1} + c_{2,1,2}x = \frac{x + \chi - x_2}{x_1 - x_2} \Rightarrow \begin{cases} c_{2,1,1} = -\frac{x_2 - \chi}{x_1 - x_2} \\ c_{2,1,2} = \frac{1}{x_1 - x_2} \end{cases} \quad (3.26)$$

$$i = 2, \quad j = 2, \quad F_{2,2} = c_{2,2,1} + c_{2,2,2}x = \frac{x + \chi - x_1}{x_1 - x_2} \Rightarrow \begin{cases} c_{2,2,1} = -\frac{x_1 - \chi}{x_1 - x_2} \\ c_{2,2,2} = \frac{1}{x_1 - x_2} \end{cases} \quad (3.27)$$

$$i = 3, \quad j = 1, \quad F_{3,1} = c_{3,1,1} + c_{3,1,2}x + c_{3,1,3}\frac{x^2}{2} = \frac{(x + \chi - x_2)(x + \chi - x_3)}{(x_1 - x_2)(x_1 - x_3)} \Rightarrow \begin{cases} c_{3,1,1} = \frac{(x_2 - \chi)(x_3 - \chi)}{(x_1 - x_2)(x_2 - x_3)} \\ c_{3,1,2} = -\frac{(x_2 - \chi) + (x_3 - \chi)}{(x_1 - x_2)(x_2 - x_3)} \\ c_{3,1,3} = \frac{1}{(x_1 - x_2)(x_2 - x_3)} \end{cases} \quad (3.28)$$

$$i = 3, \quad j = 2, \quad F_{3,2} = c_{3,2,1} + c_{3,2,2}x + c_{3,2,3}\frac{x^2}{2} = \frac{(x + \chi - x_1)(x + \chi - x_3)}{(x_2 - x_1)(x_2 - x_3)} \Rightarrow \begin{cases} c_{3,2,1} = \frac{(x_1 - \chi)(x_3 - \chi)}{(x_2 - x_1)(x_2 - x_3)} \\ c_{3,2,2} = -\frac{(x_1 - \chi) + (x_3 - \chi)}{(x_2 - x_1)(x_2 - x_3)} \\ c_{3,2,3} = \frac{1}{(x_2 - x_1)(x_2 - x_3)} \end{cases} \quad (3.29)$$

$$i = 3, \quad j = 3, \quad F_{3,3} = c_{3,3,1} + c_{3,3,2}x + c_{3,3,3}\frac{x^2}{2} = \frac{(x + \chi - x_1)(x + \chi - x_2)}{(x_3 - x_1)(x_3 - x_2)} \Rightarrow \begin{cases} c_{3,3,1} = \frac{(x_1 - \chi)(x_2 - \chi)}{(x_3 - x_1)(x_3 - x_2)} \\ c_{3,3,2} = -\frac{(x_1 - \chi) + (x_2 - \chi)}{(x_3 - x_1)(x_3 - x_2)} \\ c_{3,3,3} = \frac{1}{(x_3 - x_1)(x_3 - x_2)} \end{cases} \quad (3.30)$$

### 3.1.3 Método de colocación con polinomios interpolantes de Lagrange y distribución de Chebyshev (B.7)

En esta sección, al igual que en la anterior, se parte de los polinomios de Lagrange, aunque en este caso los polinomios incluirán todos los nodos del mallado (no como se hacía en [20]). Se ha tratado de optimizar el algoritmo de [6] reduciendo el número de bucles mediante el uso de matrices en la medida de lo posible (?). No obstante, llevando a cabo diferentes mallados se ha observado que manteniendo los bucles originales se pueden obtener órdenes de tiempo de computación similares a los obtenidos con las matrices, si y solo si se procesa en paralelo con varios procesadores. Esto, en cambio, no supone ninguna mejora sin embargo si se realiza con la opción matricial, ya que la definición de los bucles no lo permite.

En este caso, la discretización del dominio que minimiza el error tal y como demostró Chebyshev es:

$$x_i = \frac{x_{max} + x_{min}}{2} - \frac{x_{max} - x_{min}}{2} \cos\left(\frac{\pi(i-1)}{N-1}\right), \quad (3.31)$$

siendo  $N$  el número total de nodos.

Se deduce a continuación la obtención de las matrices de derivadas para una función genérica definida en un dominio de una sola dimensión. Se define el polinomio interpolante  $p(x)$  usando polinomios de Lagrange a partir de los valores conocidos ( $f_1, f_2, \dots, f_N$ ) en los puntos del dominio ( $x_1, x_2, \dots, x_N$ ) que se denominarán nodos.

$$p(x) = \sum_{i=1}^N L_i(x) f_i \quad (3.32)$$

$$L_i(x) = \prod_{j=1, j \neq i}^N \frac{x - x_j}{x_i - x_j} \quad i = 1, 2, \dots, N \quad (3.33)$$

Puede observarse que los polinomios de Lagrange en los nodos equivalen a la delta de Kronecker.

$$L_i(x_j) = \delta_{ij} \quad \forall i, j = 1, 2, \dots, N \quad (3.34)$$

Se observa que la derivada de  $p(x)$  depende de la derivada de los polinomios de Lagrange:

$$L_i^I(x) = \sum_{j=1}^N \frac{(x - x_1)(x - x_2) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_N)}{(x_i - x_1)(x_i - x_2) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_N)} \quad \forall i = 1, 2, \dots, N \quad (3.35)$$

Por tanto se obtienen ahora los valores de la derivada de la función en los nodos.

$$p^I(x) = \sum_{i=1}^N L_i^I(x) f_i \quad \Rightarrow \quad f_j^I = \sum_{i=1}^N L_i^I(x_j) f_i \quad \forall j = 1, 2, \dots, N \quad (3.36)$$

Se puede ahora repetir el proceso pero definiendo el polinomio de interpolación de la derivada de la función:

$$g(x) = f^I(x) = \sum_{i=1}^N L_i(x) f_i^I \quad (3.37)$$

Por tanto, el valor de la segunda derivada en los nodos es:

$$f_j^{II} = g^I(x_j) = \sum_{i=1}^N L_i^I(x_j) f_i^I = \sum_{i=1}^N L_i^I(x_j) \left( \sum_{k=1}^N L_k^I(x_i) f_k \right) \quad \forall j = 1, 2, \dots, N \quad (3.38)$$

Este procedimiento es válido para cualquier orden de derivada que se quiera calcular. Solo queda reformular el procedimiento para trabajar en forma matricial, lo cual es sencillo. Se definen vectores columnas formados tanto por los valores de la función en los nodos como para los de las sucesivas derivadas hasta mayor orden ( $M$ ).

$$\bar{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}, \quad \bar{f}^I = \begin{bmatrix} f_1^I \\ f_2^I \\ \vdots \\ f_N^I \end{bmatrix}, \quad \dots, \quad \bar{f}^M = \begin{bmatrix} f_1^M \\ f_2^M \\ \vdots \\ f_N^M \end{bmatrix} \quad (3.39)$$

Se define la matriz de derivada primera de los polinomios de Lagrange como sigue:

$$L^I = \begin{bmatrix} L_1^I(x_1) & L_2^I(x_1) & \dots & L_N^I(x_1) \\ L_1^I(x_2) & L_2^I(x_2) & \dots & L_N^I(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ L_1^I(x_N) & L_2^I(x_N) & \dots & L_N^I(x_N) \end{bmatrix} \quad (3.40)$$

Concluyendo, cualquier vector de incógnitas que contenga las derivada de orden  $k$  puede calcularse con las expresiones 3.41 y 3.40.

$$\bar{f}^k = L^k \bar{f} \quad k \in \mathbb{N}_{>0} \quad (3.41)$$

### 3.1.4 Extensión de matrices de derivadas a dominios en dos dimensiones

Hasta ahora se han explicado métodos para la obtención de matrices de derivadas en una sola dimensión ( $L$ ), pero dado que el dominio que se va a resolver es  $2D$ , es necesario el cálculo de nuevas matrices a partir de estas. En este apartado se supondrá un dominio rectangular en el plano  $X - Z$ . En el siguiente se introducirá el cambio de variable que permitirá resolver contornos no rectos en una u otra dirección.

El objetivo es hacer uso de un solo índice para identificar todos los nodos, y, de esta forma tener un vector de incógnitas. Este contendrá el valor de las magnitudes incógnita a resolver en los puntos de colocación. La nueva numeración se puede ver en la figura 3.2.

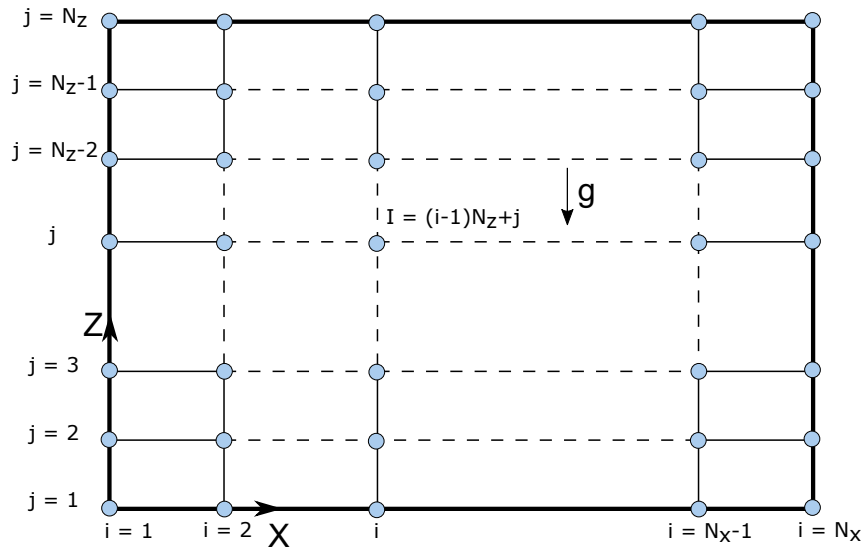


Figura 3.2 Numeración de nodos con índice único.

$$\begin{bmatrix} 1 & N_z + 1 & \cdots & N_z(N_x - 2) + 1 & N_x \\ \vdots & & \ddots & & \vdots \\ N_z & 2N_z & \cdots & N_z(N_x - 1) & N_z N_x \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 2 \\ \vdots \\ N_z - 1 \\ N_z \\ N_z + 1 \\ \vdots \\ 2N_z \\ \vdots \\ N_t = N_z N_x \end{bmatrix} \quad (3.42)$$

La numeración con un único índice permite expresar las matrices de derivadas como sigue ([6]):

$$\left. \frac{\partial \phi}{\partial X} \right|_I = \sum_{K=1}^{N_t} D_x(I, K) \phi_K \quad (3.43)$$

$$\left. \frac{\partial \phi}{\partial Z} \right|_I = \sum_{K=1}^{N_t} D_z(I, K) \phi_K \quad I, K = 1, 2, \dots, N_t \quad (3.44)$$

Como puede verse en la referencia las nuevas matrices de derivadas  $D$  pueden deducirse como se muestra a continuación. En primer lugar se muestran la interpolación de Lagrange para una función con nodos en dos dimensiones  $X$  y  $Z$  (3.45). Para ello hay que tener en cuenta las matrices obtenidas en la sección anterior, según se hayan obtenido en una u otra dimensión,  $L_X$  y  $L_Z$ . Primero se aplican los interpolantes de Lagrange en dirección  $Z$  (manteniendo  $x$  constante), y luego en dirección  $X$ .



$$\phi(X,Z) = \sum_{n=1}^{N_z} L_{Z_n}(Z) \phi(X,Z_n) = \sum_{m=1}^{N_x} \sum_{n=1}^{N_z} L_{X_m}(X) L_{Z_n}(Z) \phi(X_m, Z_n) \quad (3.45)$$

Los polinomios interpolantes se definen como anteriormente:

$$L_{X_m}(X) = \prod_{\substack{j=1 \\ j \neq m}}^{N_x} \frac{x - x_j}{x_m - x_j} \quad (3.46)$$

$$L_{Z_n}(Z) = \prod_{\substack{j=1 \\ j \neq n}}^{N_z} \frac{Z - Z_j}{Z_n - Z_j} \quad (3.47)$$

Derivando respecto a  $X$  o  $Z$  la función  $\phi(X,Z)$  en la expresión 3.45 se obtienen las expresiones 3.49, en las que aparecen las derivadas de las matrices  $L_X(X)$  y  $L_Z(Z)$ , cuyas expresiones se encuentran en la sección anterior.

$$\frac{\partial \phi(X,Z)}{\partial X} = \sum_{m=1}^{N_x} \sum_{n=1}^{N_z} L'_{X_m}(X) L_{Z_n}(Z) \phi(x_m, z_n) \quad (3.48)$$

$$\frac{\partial \phi(X,Z)}{\partial Z} = \sum_{m=1}^{N_x} \sum_{n=1}^{N_z} L_{X_m}(X) L'_Z(Z) \phi(X_m, Z_n) \quad (3.49)$$

Es fácil deducir que con la nueva numeración los dos sumatorios pueden unificarse quedando las expresiones 3.44. Por tanto, las matrices  $D_X$  y  $D_Z$  pueden obtenerse según 3.51, teniendo en cuenta que  $i = 1, 2, \dots, N_x$  y  $j = 1, 2, \dots, N_z$ .

$$D_X(I, K) = L'_{X_m}(X_i) L_{Z_n}(Z_j) \quad I = (i-1)N_z + j = 1, 2, \dots, N_t \quad (3.50)$$

$$D_Z(I, K) = L_{X_m}(X_i) L'_Z(Z_j) \quad K = (m-1)N_z + n = 1, 2, \dots, N_t \quad (3.51)$$

En [6] se muestra una forma de realizar los cálculos de las matrices  $D$ . Sin embargo, precisamente se puede aprovechar que  $L_{Z_n}(Z_j)$  y  $L_{X_m}(X_i)$  equivalen a la delta de Dirac ( $\delta_{nj}$  y  $\delta_{mi}$  respectivamente) para realizar todas las operaciones necesarias con el producto de Kronecker (en Matlab con el comando *kron*). De esta forma quedan 3.53, siendo  $I$  la matriz identidad.

$$D_X = L_X \otimes I_{N_z \times N_z} \quad (3.52)$$

$$D_Z = I_{N_x \times N_x} \otimes L_Z \quad (3.53)$$

Se muestra a continuación de manera visual la forma de la matriz  $D_Z$ . Debido a la forma de ordenar los nodos la matriz  $D_X$  es algo más compleja de plasmar de manera compacta.



plano original del problema  $x - z$  del apartado que se muestra a continuación.

### 3.1.5 Tratamiento y transformación del dominio

Como se ha mencionado en el apartado anterior, ha de introducirse un cambio de variable para poder tener en cuenta contornos no rectos en una de las direcciones. En esta sección se va a suponer que estos son los horizontales (inferior y superior) aunque la misma idea se puede aplicar en caso de que sean los verticales (códigos en B.1 y ?? respectivamente). Por comodidad solo se va a desarrollar la forma de proceder del primero aunque en los resultados se haga uso de ambos, para lo que hay que modificar el código de forma adecuada. En este caso, el cambio es el siguiente:

$$Z(x) = \frac{z - z_b(x)}{z_t(x) - z_b(x)} \tag{3.64}$$

$$X = x, \tag{3.65}$$

siendo  $z_t(x)$  y  $z_b(x)$  las curvas que definen el contorno superior e inferior respectivamente.

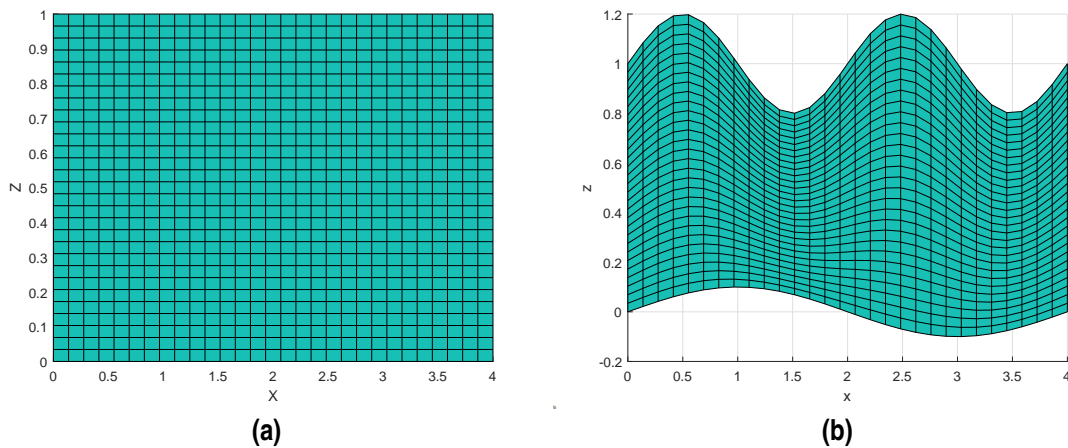


Figura 3.3 Ejemplo de malla transformada.

La razón por la cual se hace este cambio es para que la definición del mallado sea único en cada dirección, es decir, la malla para todo  $X = cte$  es la misma, e igualmente sucede para todo  $Z = cte$ . A modo de ejemplo, se muestra en la figura 3.3 una malla uniforme en  $X$  y  $Z$  (3.3a) y la correspondiente malla transformada en el dominio original  $x - z$  (3.3b).

Por último es necesario transformar las matrices halladas en el apartado anterior ( $D(X)$  o  $D(Z)$ ), las cuales se definieron en el dominio transformado  $X - Z$ , a otras que estén definidas en el dominio real del problema  $x - z$ . Para ello basta con aplicar la regla de la cadena tantas veces como sea necesario (para más detalle ver apéndices B.3 y B.4 según los contornos curvos sean los horizontales o los verticales respectivamente).

## 3.2 Integración temporal. Condición de Courant-Friedrich-Lewy

Las ecuaciones diferenciales a resolver incluyen derivadas temporales. Para su resolución se va a usar un método tipo explícito-implícito o semi-implícito (IMEX, ver [21]). Para ello la derivada

temporal se aproximará mediante la definición de derivada como se muestra en 3.67 y los términos no lineales del sistema en el instante  $t + 1$  se van a aproximar mediante los valores de la solución en el instante anterior  $t$ . De manera esquemática, el método se muestra en 3.66.

$$y(t + \delta_t) = F(y(t + \delta_t)) + G(y(t)) \quad (3.66)$$

$$\frac{\partial \phi}{\partial t} = \frac{\phi_t - \phi_{t-1}}{\delta_t} \quad (3.67)$$

Para que este método funcione correctamente el paso de tiempo debe ser lo suficientemente pequeño para recoger los diferentes términos adecuadamente, y a su vez, para que el tiempo de simulación no sea excesivo este debe ser lo suficientemente grande. Para escoger su valor es útil tener en cuenta la condición CFL (Courant-Friedrich-Lewy), la cual considera el tiempo de residencia de una partícula fluida en un volumen fluido (en este caso sería el espacio entre dos nodos en la dirección de la velocidad) y el paso de tiempo escogido. Esta condición indica que el paso de tiempo no puede ser mayor que el tiempo de residencia, ya que supondría que la solución no recoge el paso de la partícula fluida por todos los nodos. Para un dominio unidimensional se expresa como sigue:

$$C = \frac{\Delta t}{\frac{\Delta x}{v}}, \quad (3.68)$$

siendo  $\Delta x$  y  $v$  la distancia entre dos nodos y la velocidad de la partícula fluida respectivamente. Este criterio se puede extrapolar para un dominio bidimensional. Se simplificará en este caso considerando la distancia en  $x$  y  $z$  así como las componentes de velocidad por separado y tomando el máximo valor obtenido. Se va a tener en cuenta que este valor no supere la unidad, y se ha comprobado que si esto sucede el método diverge.

### 3.3 Esquema general de resolución

Se resume ahora la forma en que se resuelve el problema de manera general. Como se ha comentado anteriormente, si el dominio original tiene contornos no rectos, el primer paso es transformar según el cambio 3.65, pasando de 3.4a o 3.4b a 3.4c.

A continuación, se establece un criterio para definir los nodos de la malla en este nuevo dominio (malla lineal, polinomios de Chebyshev,...). Los nodos en el dominio original se pueden calcular fácilmente deshaciendo el cambio de variable. Se calculan las matrices de derivadas espaciales respecto a las variables transformada mediante diferencias finitas o con nodos de Chebyshev. El siguiente paso es calcular las matrices de derivadas respecto a las variables originales aplicando la regla de la cadena tantas veces como sea necesario en función del orden superior de derivada calculado inicialmente. Estas matrices son las que se aplican en las ecuaciones físicas del problema.

En cuanto a la integración en el tiempo, ya se ha comentado que se usará un método semi-implícito que tendrá en cuenta la solución del instante anterior para aproximar los términos no lineales de las ecuaciones. Por tanto, el sistema a resolver sin tener en cuenta aún las condiciones de contorno se puede ver a continuación:

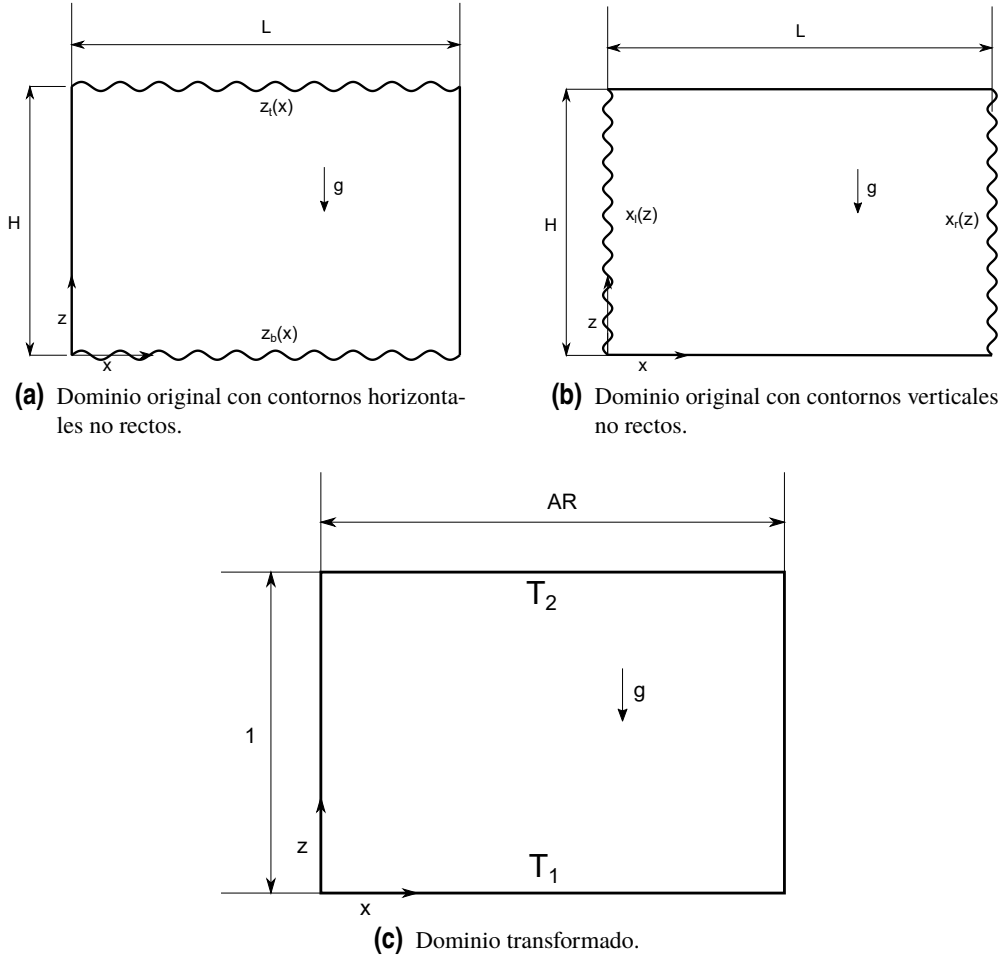


Figura 3.4 Transformación del dominio.

$$\begin{bmatrix} D_L - \delta_t \nu D_L^2 & \delta_t g \beta D_x \\ 0 & I - \delta_t D_L \end{bmatrix} \begin{bmatrix} \psi \\ T \end{bmatrix} = \begin{bmatrix} \delta_t (-D_z \psi D_x D_L \psi + D_x \psi D_z D_L \psi) \Big|_{t-1} + D_L \psi \Big|_{t-1} \\ \delta_t (-D_z \psi D_x T + D_x \psi D_z T) \Big|_{t-1} + T \Big|_{t-1} \end{bmatrix} \Rightarrow Ax = b \quad (3.69)$$

Este sistema tiene  $2N_x \times N_z$  ecuaciones y  $2N_x \times N_z$  incógnitas. Estas son los valores de la función de corriente y la temperatura en todos los nodos. Sin embargo, no todos estos valores son desconocidos. Según sean las condiciones de contorno, estos pueden ser conocidos, o en su defecto, una expresión que dependa de ellos. Por ejemplo, la derivada primera de una magnitud es nula en un contorno.

Para que el problema tenga solución, es necesario imponer tantas condiciones de contorno para cada incógnita y dirección como mayor orden de derivada exista para esa variable y dirección. Es decir, en la función de corriente aparecen derivadas en  $x$  y  $z$  de cuarto orden. Por tanto, hay que imponer cuatro condiciones de contorno en dirección  $x$  y otras tantas en  $z$  para la función de corriente. De la misma forma habrá que imponer dos más en cada dirección para la temperatura.

Se va a seguir el método propuesto en [6] para imponer las condiciones de contorno. Este consiste en sustituir las filas (ecuaciones) correspondiente a los nodos donde se quieran imponer por otras que expresen las condiciones de contorno. Para imponer más de una condición en un contorno

para una misma variable se recurre a los nodos adyacentes a estos. Se denominarán subcontornos a aquellos que quedarían al eliminar los contornos de la matriz.

Por ejemplo, si se quiere imponer un valor nulo a la función de corriente y a su derivada en la dirección normal para el contorno superior, habría que sustituir las  $N_x$  filas del sistema matricial correspondiente a los nodos de este contorno y las  $N_x - 2$  filas del subcontorno por las filas correspondientes de la matriz identidad o de  $D_z$  según se quiera imponer el valor de la variable o su derivada. A esto hay que añadir la imposición del valor deseado como término independiente en el miembro derecho de la ecuación 3.69. Este mismo procedimiento hay que aplicarlo en el caso de la temperatura para obtener finalmente el sistema matricial por el cual se resolverá el problema.

De esta forma se llega a un sistema de ecuaciones matricial algebraico que se puede resolver fácilmente en cada paso de tiempo, independientemente de la malla, la forma de obtener los operadores matriciales para calcular las derivadas o las condiciones de contorno que apliquen.

Esta forma de proceder es ciertamente ventajosa desde un punto de vista computacional ya que, una vez definido el dominio y el paso de tiempo, la matriz que multiplica al vector de incógnitas es constante. Esto supone un menor tiempo respecto al caso en que la matriz variase en cada paso de tiempo, puesto que lo más costoso es el cálculo de la inversa, o en su defecto la resolución del sistema de ecuaciones.

Como se detallará en el capítulo 5 se llega a la conclusión de que, sea la matriz dispersa o no, es decir, esté calculada con Chebyshev o con diferencias finitas, realizar una factorización  $LU$  de la matriz  $A$  previa al cálculo del primer paso de tiempo es más eficiente. Ya que la matriz  $A$  es constante, solo hay que realizar esta factorización una vez. Posteriormente, con las matrices obtenidas se resuelve el sistema (sin calcular explícitamente la inversa) mediante el comando de Matlab "\", de manera que se resuelven tres sistemas lineales en cada paso siendo la matriz de coeficientes siempre, triangular o de permutación, en lugar de un sistema lineal con una matriz llena, lo cual es mucho más eficiente.

### 3.4 Consideraciones de carácter computacional para la optimización de los códigos

En esta sección se van a dar algunas líneas generales seguidas en todos los códigos, no solo en los que se refieren al mallado. El objetivo es tener algoritmos suficientemente versátiles a la par que rápidos y eficientes, que sean capaces de realizar mallas lo más refinadas posibles sin aumentar de manera inaceptable el tiempo o la memoria de computación. Por ello, se muestran a continuación varias consideraciones que ayudan a tales fines:

- En primer lugar, es una buena práctica sopesar qué es más eficiente en cada caso: crear una variable intermedia o no hacerlo. Esto depende fundamentalmente de si el cálculo de esta variable es costoso, así como de si será necesaria un número de veces considerable, ya que supondrá un coste en cuanto a memoria requerida. En relación a esto, se ha seguido la práctica de borrar aquellas variables que no son necesarias a partir de cierto momento. Esto se ha hecho en el mallado, por ejemplo, con las matrices de derivadas respecto a  $X$  y  $Z$ , ya que una vez se han empleado para calcular las matrices en términos de  $x$  y  $z$ , por lo general, ya no son necesarias.
- Otra decisión que contribuye a una menor necesidad de memoria RAM, y también a la rapidez de la ejecución del código, es el uso de la función *bsxfun* de Matlab para genera matrices a partir de un vector y una matriz. Esto sucede a menudo al aplicar el cambio de variable en las derivadas. La alternativa sería crear una matriz de tamaño  $N_t \times N_t$  a partir del vector, que en muchos casos sería repetir este. El ahorro de memoria es enorme pero además es mucho más

eficiente en cuanto a tiempo de computación.

- Otra decisión adoptada, ha sido la creación de una función que tiene en cuenta el número de elementos distintos de cero que debería tener una matriz cuadrada, para decidir si guardarla como dispersa o llena en función de cual requiere menos memoria. Esto supone aumentar el tiempo de computación a priori, aunque se ha comprobado que el aumento no es relevante, mientras que el ahorro en términos de memoria puede ser muy importante. Además, una vez se tiene en cuenta la resolución, el hecho de tener matrices dispersas puede suponer un ahorro de tiempo muy importante si se aplican las técnicas adecuadas durante esta.
- En último lugar, de manera general se eliminan todos los bucles posibles. Esta decisión supone una gran mejora al calcular las derivadas haciendo uso de todos los nodos si se trabaja en serie.





## 4 Linealización y análisis de estabilidad. Número de Rayleigh crítico

---

Ya se tiene el conjunto de ecuaciones generales que describe el tipo de problema que se va a tratar y los métodos para modelar su física a través de estas ecuaciones. Esto incluye el cálculo de derivadas, tanto espaciales como temporales, el mallado del dominio, la aplicación de las condiciones de contorno, etc. En este capítulo se procede a estudiar la estabilidad de los sistemas partiendo de una linealización de las ecuaciones en torno a un estado de equilibrio. Como ya se comentó en la introducción el fenómeno convectivo tiene su origen en una inestabilidad que aparece a partir de un determinado valor del número de Rayleigh, diferente para cada combinación de dominio y condiciones de contorno.

En primer lugar se ahonda en el significado del número de Rayleigh crítico utilizando como fuente [7]. Dado que se va a analizar la estabilidad, o lo que es lo mismo, la existencia de convección, las condiciones de contorno que se van a imponer no provocarán par baroclínico. Esto es, la geometría del dominio y las condiciones de temperatura y velocidades impuestas no provocarán movimiento salvo que la diferencia de temperatura entre el contorno superior e inferior sea suficientemente alta. En cuanto a la geometría, para que esto se cumpla, los contornos inferior y superior deben ser rectos. En cambio, en una de las secciones, se considerarán contornos laterales variables en los que se impondrá la temperatura de equilibrio de manera que no exista par baroclínico.

A continuación se procede a desarrollar las ecuaciones linealizadas en torno al punto de equilibrio. Para esto es conveniente descomponer el campo de temperatura y de la función de corriente en dos partes: una parte debida al equilibrio y otra que se denominará de perturbación. Tras este primer paso se adimensionalizan las ecuaciones y se procesarán hasta llegar al sistema que se utilizará finalmente para estudiar la estabilidad lineal.

En este punto se explica el tipo de condiciones de contorno que se aplicará, teniendo en cuenta la descomposición de la solución realizada, y la forma en que se obtendrá el valor de Rayleigh crítico. Por último, se utilizará este resultado como base para la validación inicial de los métodos del proyecto por una serie de razones que se detallará más adelante. Además de verificar que los métodos obtienen buenos resultados, se tomarán decisiones para acotar parámetros requeridos en estos, como puede ser el número de nodos en cada dirección o la distribución de estos, así como para discernir las ventajas o inconvenientes de los diferentes métodos.

## 4.1 Número de Rayleigh crítico

Dado un sistema como el de la figura 4.1, en el que se impone una temperatura en el contorno inferior mayor que en el superior, puede o no existir convección en función de la diferencia entre los valores impuestos y del resto de condiciones de contorno.

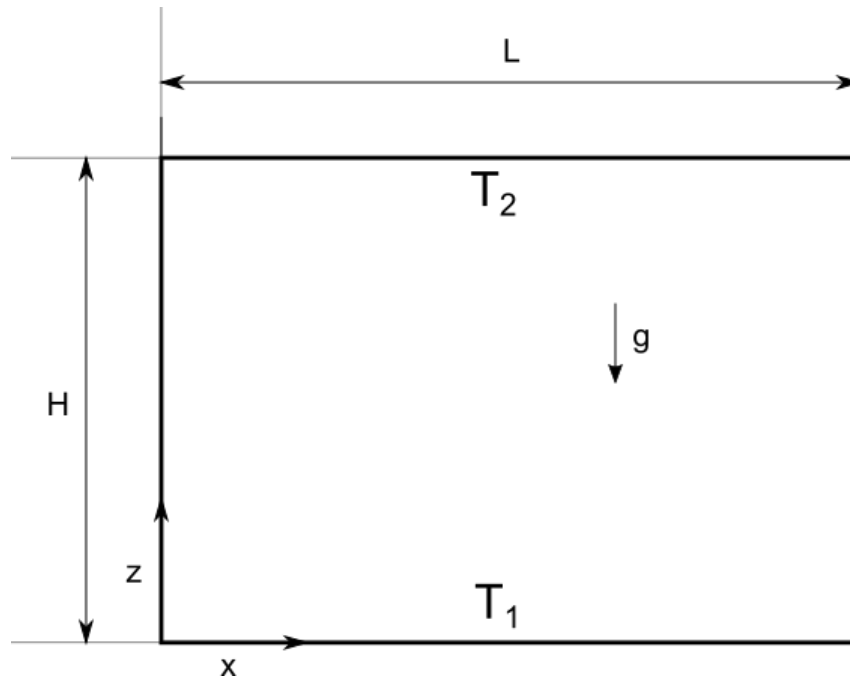


Figura 4.1 Esquema básico de convección ( $T_1 > T_2$ ).

Partiendo de la solución de equilibrio, si se aplica una pequeña perturbación al sistema, esta se amortiguará o se amplificará según el número de Rayleigh impuesto sea menor o mayor que el crítico respectivamente. En este valor subyace la importancia relativa entre los efectos disipativos y el flujo de calor existente. En [7] se puede consultar un análisis teórico para el caso de un líquido entre dos placas horizontales infinitas, o bien sobre una placa infinita de manera que el contorno superior es libre. El análisis se formula en velocidades y temperatura, imponiendo que la solución es la suma de un estado de reposo más una pequeña perturbación en las variables del problema. Para que esto sea válido, la diferencia de temperatura debe ser pequeña ( $\frac{T_1 - T_2}{T_1} \ll 1$ , siendo  $T_1$  y  $T_2$  las temperaturas de la placa inferior y la del contorno superior (placa o ambiente) respectivamente. Dado que las placas son infinitas, la solución básica y condiciones de contorno son independientes de las coordenadas  $x$  e  $y$  (la dirección  $z$  es la perpendicular al plano de la placa). Entonces, se pueden expresar las variables independientes, velocidad perpendicular a la placa y la temperatura tras reformular las ecuaciones, como se muestra en 4.2 (todas las variables son adimensionales). En este proyecto sin embargo se han formulado las velocidades en términos de la función de corriente como ya se ha mencionado.

$$v_z = e^{i(k_x x + k_y y - \omega t)} w(z) \quad (4.1)$$

$$T = e^{i(k_x x + k_y y - \omega t)} \theta(z) \quad (4.2)$$

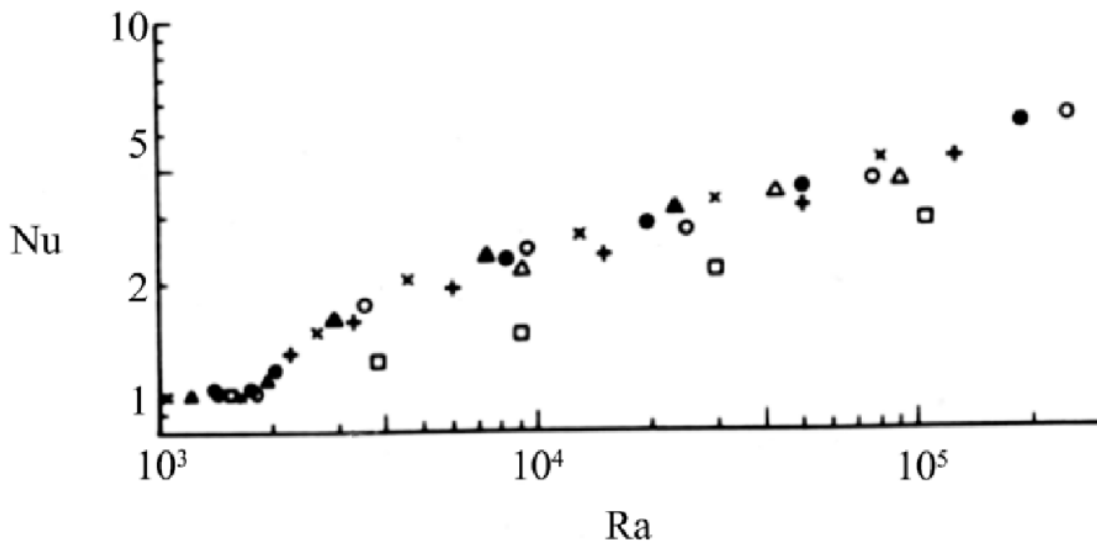
El sistema resultante depende por tanto de 3 parámetros (dado que las direcciones  $x$  e  $y$  son equivalentes  $k_x = k_y$ ):  $k$ ,  $Pr$  y  $Ra$ . Para resolver el punto en el que el sistema se hace inestable se

resuelve para  $\omega = 0$  y se reformulan las ecuaciones llegando a formular el problema en términos de  $v_z$  únicamente. En este caso, se llega a una ecuación diferencial ordinaria lineal de sexto orden y se resuelve expresando la función incógnita como la superposición lineal de tres grados de libertad:  $w(z) = C_1 w_1(z) + C_2 w_2(z) + C_3 w_3(z)$ . Imponiendo las condiciones de contorno adecuadamente se llega a un sistema algebraico con incógnitas  $C_1$ ,  $C_2$  y  $C_3$ , que tendrá solución distinta a la homogénea si y solo si el determinante del sistema es nulo.

Fijado  $Pr$ , se puede ir variando  $k$  para obtener una función  $Ra = Ra(k)$ , obteniendo una por cada modo, esto es, por cada grado de libertad dado a  $w(z)$  ( $w_1, w_2, \dots, w_n$ ). Como se puede observar en la figura 15.7 de [7], cada función  $Ra = Ra(k)$  tiene un mínimo que se da para una longitud de onda  $k$  diferente. El estado de reposo será estable frente a pequeñas perturbaciones de cualquier longitud de onda si el número de Rayleigh es menor que el mínimo de todos esos valores mínimos, siendo este el número de Rayleigh crítico. Tal y como indica la fuente, en el caso en que el contorno superior sea el impuesto por una placa rígida, el valor de Rayleigh crítico es 1708 y se da para una frecuencia  $k = 3.117$ . También ofrece el valor  $Ra_c = 1101$  para el caso con superficie superior libre. Además, se muestran formas analíticas de obtener las velocidades en el dominio.

Por último, cabe destacar la importancia del número de Rayleigh en el flujo de calor existente. El número de Nusselt indica el cociente entre el flujo de calor total y el flujo de calor por conducción, de manera que si este es unidad, la totalidad del flujo de calor se debe a conducción.

$$Nu = \frac{\left. \frac{\partial \hat{t}}{\partial \hat{z}} \right|_{\hat{z}=0}}{\left. \frac{\partial \hat{t}}{\partial \hat{z}} \right|_{\hat{z}=0, \text{conducción}}} \quad (4.3)$$



**Figura 4.2** Valores experimentales del número de Nusselt en experimentos diferentes de convección de Bénard como función del número de Rayleigh para diferentes sustancias [7].

Como puede verse en la figura 4.2, cuando el número de Rayleigh supera su valor crítico, el de Nusselt crece, esto es, la importancia del flujo de calor por convección aumenta respecto al que se produce por conducción. Para valores inferiores al crítico el número de Nusselt es unidad. Esto se aprovechará para obtener el valor crítico de Rayleigh en el siguiente capítulo.

## 4.2 Desarrollo de las ecuaciones linealizadas y condiciones de contorno

En primer lugar la temperatura se va a descomponer en dos partes: la temperatura de equilibrio y una parte de perturbación. La primera no depende del tiempo puesto que es la solución supuesto que haya equilibrio, es decir las velocidades son nulas en todo el dominio en todo momento y por tanto se cumple que  $\nabla^2 T_{eq} = 0$ . La segunda componente completa el valor de la temperatura en cada instante. De la misma forma se descompone la función de corriente aunque dado que las velocidades son nulas para el estado de equilibrio, también lo será esta.

$$T(t, x, z) = T_{eq}(x, z) + \theta(t, x, z) \quad (4.4)$$

$$\Psi(x, z, t) = \Psi_{eq}(x, z) + \phi(x, z, t) = \phi(x, z, t) \quad (4.5)$$

Para la adimensionalización se tendrán en cuenta las siguientes expresiones:

$$\hat{x} = \frac{x}{H}, \quad \hat{z} = \frac{z}{H}, \quad \hat{u} = \frac{H}{\alpha} u, \quad \hat{v} = \frac{H}{\alpha} v, \quad \hat{t} = \frac{\alpha}{H^2} t, \quad \hat{\psi} = \frac{\Psi}{\alpha}, \quad \hat{T} = \frac{T - T_2}{T_1 - T_2} \quad (4.6)$$

Se aplica esta descomposición a las ecuaciones 2.19 y 2.20, y posteriormente se desprecian los términos no lineales, obteniendo 4.7 y 4.8.

$$\frac{\partial}{\partial t} (\nabla^2 \phi) = \nu \nabla^4 \phi - g\beta \frac{\partial \theta}{\partial x} \quad (4.7)$$

$$\frac{\partial \theta}{\partial t} - \frac{\partial \phi}{\partial x} \frac{\partial T_{eq}}{\partial z} = \alpha \nabla^2 \theta \quad (4.8)$$

Se tiene en cuenta además que la temperatura de equilibrio no depende de la coordenada  $x$  en caso de que las condiciones de contorno permitan que no exista par baroclínico, siendo su expresión 4.9.

$$T_{eq} = (T_1 - T_2)(H - z) + T_2 \quad (4.9)$$

Sustituyendo 4.9 en 4.8, las variables dimensionales por las adimensionales y operando para simplificar las ecuaciones aparecen dos números adimensionales que dominan el problema: el número de Rayleigh y el número de Prandtl. Como ya se ha comentado, el primero indica la importancia de la transferencia de calor por convección respecto a la que ocurre por conducción. El segundo, es mayor cuanto mayor sea la difusión de la cantidad de movimiento frente a la de difusión de calor.

$$Ra = \frac{gH^3\beta(T_1 - T_2)}{\nu\alpha} \quad (4.10)$$

$$Pr = \frac{\nu}{\alpha} \quad (4.11)$$

$$\frac{1}{Pr} \frac{\partial(\hat{\nabla}^2 \hat{\phi})}{\partial \hat{t}} = \hat{\nabla}^4 \hat{\phi} - Ra \frac{\partial \hat{\theta}}{\partial \hat{x}} \quad (4.12)$$

$$\frac{\partial \hat{\theta}}{\partial \hat{t}} + \frac{\partial \hat{\phi}}{\partial \hat{x}} = \hat{\nabla}^2 \hat{\theta} \quad (4.13)$$

Se suponen ahora soluciones exponenciales en el tiempo del campo de temperatura y de la función de corriente, esto es:

$$\hat{\theta}(\hat{x}, \hat{z}, \hat{t}) = \hat{\theta}_0(\hat{x}, \hat{z}) e^{-s\hat{t}} \quad (4.14)$$

$$\hat{\phi}(\hat{x}, \hat{z}, \hat{t}) = \hat{\phi}_0(\hat{x}, \hat{z}) e^{-s\hat{t}} \quad (4.15)$$

Con todo ello, se llega al problema de autovalores generalizados que se muestra a continuación:

$$\begin{bmatrix} Pr\hat{\nabla}^4 & -PrRa\frac{\partial}{\partial \hat{x}} \\ -\frac{\partial}{\partial \hat{x}} & \hat{\nabla}^2 \end{bmatrix} \begin{bmatrix} \hat{\phi}_0 \\ \hat{\theta}_0 \end{bmatrix} = s \begin{bmatrix} \hat{\nabla}^2 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{\phi}_0 \\ \hat{\theta}_0 \end{bmatrix} \Rightarrow Ax = sBx \quad (4.16)$$

Para tener en cuenta las condiciones de contorno, se pueden buscar expresiones que relacionen los contornos y subcontornos con el dominio interior a estos (como se hace en [22] y [23]). Sin embargo, se ha optado por un método más directo, posible dado que se han descompuesto las variables en torno a la solución de equilibrio. Se sustituyen en las matrices  $A$  y  $B$  de 4.16 las condiciones de contorno en los diferentes nodos del contorno y subcontorno. Al resolver el problema de autovalores generalizados, se obtendrán algunos cuyo valor es unidad. Estos se deben a que en ciertas filas (contorno y subcontorno) las matrices  $A$  y  $B$  son iguales ya que se han impuesto las expresiones homogéneas de las condiciones de contorno. Por tanto, esos autovalores se obviarán cuando se busque el valor de Rayleigh crítico.

Este sistema es el que se resolverá en el análisis de estabilidad. Falta por concretar las condiciones de contorno, que dado que se ha formulado el problema en torno al equilibrio son muy sencillas, tanto para  $\hat{\theta}$  como para  $\hat{\phi}$ .

### 4.2.1 Condiciones de contorno

Tal y como se explicó en el capítulo anterior, es importante recalcar la forma en que se implementarán las condiciones de contorno (la que se realiza en [6]). El objetivo es llegar a un sistema de ecuaciones matricial algebraico que permita obtener la solución de las magnitudes incógnitas en los nodos del mallado. El producto del número de nodos por las magnitudes incógnitas (función de corriente y temperatura) determinará el número de filas de la matriz. Las condiciones de contorno se impondrán sustituyendo en la ecuación correspondiente a cada nodo (perteneciente al contorno o subcontorno como se verá más adelante) por la expresión que indique tal condición. De esta forma, se consigue un sistema de ecuaciones válido, independiente de la forma en que se hayan obtenido los operadores matriciales que representan las derivadas espaciales.

En primer lugar, dado que aparecen derivadas espaciales de hasta cuarto orden en la función de corriente, se necesitan cuatro condiciones de contorno en cada dimensión. Los contornos se van a modelar bien como pared o bien como superficie libre dependiendo del problema. En el primer caso la velocidad es nula, mientras que en el segundo la velocidad normal al contorno y el esfuerzo tangente son nulos.

Con el objetivo de disminuir el número de operaciones se van a reformular las condiciones en velocidad. Para ello hay que distinguir si la pared es vertical u horizontal.

$$\hat{u} = \frac{\partial \hat{\phi}}{\partial \hat{z}} = 0 \quad (4.17)$$

$$\hat{v} = -\frac{\partial \hat{\phi}}{\partial \hat{x}} = 0 \quad (4.18)$$

Sea la pared horizontal, por tanto:

$$\hat{v} = -\frac{\partial \hat{\phi}}{\partial \hat{x}} = 0 \quad \text{para} \quad \hat{z} = \hat{z}_0 \quad \text{y} \quad \forall \hat{x} \Rightarrow \hat{\phi} = \hat{\phi}(\hat{z})$$

Ya que  $\hat{z} = \hat{z}_0$ , entonces el valor de la función de corriente en la pared es constante. El mismo razonamiento puede aplicarse a una pared vertical. Igualando sus valores en los vértices en común se llega a la conclusión de que la función de corriente en todos los contornos vale lo mismo, tanto si se modela como pared como si se hace como borde libre (siempre que en todos los contornos la velocidad normal a estos sea nula). Este valor es arbitrario ya que debe darse una referencia para un punto cualquiera del dominio. En este caso se impondrá un valor nulo. Por tanto las condiciones quedan como siguen:

$$\text{Pared horizontal} \begin{cases} \hat{\phi} = 0 \\ \frac{\partial \hat{\phi}}{\partial \hat{z}} = 0 \end{cases} \quad (4.19)$$

$$\text{Pared vertical} \begin{cases} \hat{\phi} = 0 \\ \frac{\partial \hat{\phi}}{\partial \hat{x}} = 0 \end{cases} \quad (4.20)$$

En segundo lugar se analiza el caso de superficie libre. En este caso la condición de velocidad normal nula se puede reformular de la misma forma que en el caso de la pared, distinguiendo si el contorno es vertical u horizontal. La otra condición se obtiene de la definición de esfuerzo tangencial:

$$\tau = \mu \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial z} \right) = \mu \left( \frac{\partial^2 \phi}{\partial z^2} - \frac{\partial^2 \phi}{\partial x^2} \right) = 0$$

También esta condición se puede reformular quedando por tanto las siguientes condiciones:

$$\text{Superficie libre horizontal} \begin{cases} \hat{\phi} = 0 \\ \frac{\partial^2 \hat{\phi}}{\partial z^2} = 0 \end{cases} \quad (4.21)$$

$$\text{Superficie libre vertical} \begin{cases} \hat{\phi} = 0 \\ \frac{\partial^2 \hat{\phi}}{\partial x^2} = 0 \end{cases} \quad (4.22)$$

En cuanto a la temperatura, el mayor orden de derivada espacial es dos y por tanto se necesitan dos condiciones de contorno en cada dimensión. Las dos condiciones más inmediatas son: desviación de temperatura nula o flujo de calor perpendicular nulo. Estas condiciones se pueden resumir en las siguientes, siendo  $\vec{n}$  el vector normal al contorno.

$$\text{Flujo de calor nulo} \Rightarrow (\hat{\nabla} \hat{\theta}) \cdot \vec{n} = 0 \quad (4.23)$$

$$\text{Desviación de temperatura nula} \Rightarrow \hat{\theta} = 0 \quad (4.24)$$

### 4.3 Resolución numérica del análisis de estabilidad

Ahora se van a obtener resultados numéricos con los métodos desarrollados anteriormente para diferentes dominios y condiciones de contorno. El primer objetivo es usar el análisis lineal de estabilidad como base para la validación de los métodos en casos sin par baroclínico, para lo cual, además, hay que encontrar criterios válidos para diferentes aspectos de estos: tipo de malla, número de nodos y paso de tiempo.

El análisis de estabilidad es más rápido que resolver en el tiempo el problema completo y requiere fijar menos parámetros, por tanto se realizará un estudio inicial sobre como afecta la malla escogida y el método de cálculo de las derivadas. En el siguiente capítulo se partirá de los resultados aquí obtenidos con el fin de fijar algunos parámetros de los métodos y se compararán algunos valores críticos calculados resolviendo el problema completo no lineal.

En primer lugar se realizará el análisis para dos casos principales: capa de fluido entre dos placas y fluido ubicado entre una placa y aire ambiente. En ambos casos se varía la relación de aspecto del dominio con el objetivo de aproximarse a los valores teóricos conocidos de placas infinitas. A continuación, se realiza el estudio para el caso en que se imponen condiciones de periodicidad en los contornos verticales, lo que ayudará a entender el caso límite de placas infinitas sin la necesidad de resolver dominios excesivamente grandes. Por último, se muestran algunos ejemplos con contornos no rectos, pero en los que tampoco existe par baroclínico debido a las condiciones impuestas, que permiten validar el cambio de variable efectuado en el capítulo anterior para modelar este tipo de dominios.

#### 4.3.1 Obtención del número de Rayleigh crítico

Para resolver el problema de autovalores generalizados se usará el comando de Matlab *eigs*. Este obtiene una matriz diagonal con los  $n$  autovalores generalizados que cumplan un criterio (mayor

o mínimo modulo, parte real o imaginaria, en torno a un valor, etc), y otra matriz con los correspondientes autovectores. Se usa este comando en lugar de *eig* para obtener solo un conjunto de autovalores dado que por lo general serán matrices de tamaño elevado y dispersas. Se usará el comando dos veces. Por un lado se buscará el autovalor con parte real más próxima a  $(0 + 1 - tol)/2$ , siendo *tol* una tolerancia dada. Por otro lado se buscará un cercano a un valor mayor que uno. De esta forma, se podrá obtener cualquier autovalor con parte real mayor que cero sin que exista la posibilidad de que el encontrado tenga valor unidad. Si uno de esos dos autovalores tiene parte real mayor que cero y es distinto de la unidad, el número de Rayleigh es superior al valor crítico.

Se aumentará el número de Rayleigh impuesto desde un valor suficientemente pequeño hasta encontrar un valor superior al crítico. Una vez acotado un intervalo que contenga a este, se aplicará un procedimiento de tipo bisección. Los autovectores correspondientes a los autovalores representan la forma de la solución pero no la amplitud de la magnitud, por lo que si se quisiera conocer la solución completa, debería procederse como se explica en el siguiente capítulo. Otra forma sería realizar una análisis de tercer orden como se hace en [23] y [22].

Se ha hecho uso de colocación con nodos de Chebyshev y diferencias finitas obtenidas por Fornberg con varios órdenes de precisión. Cabe destacar la diferencia de tiempo empleado entre ambos métodos a igual número de nodos. La única diferencia entre ambos es el cálculo de las derivadas.

#### **4.3.2 Validación de los métodos para contornos rectos y análisis de sensibilidad de malla (B.8 y B.9)**

Concretamente, se va a calcular el valor de Rayleigh crítico para dos casos diferentes: un fluido entre dos placas horizontales infinitas (la distancia entre ambas es suficientemente grande para evitar efectos adicionales debido a la tensión superficial) y un fluido sobre una placa horizontal infinita (y con límite superior libre, esto es, en contacto con la atmósfera). Se denominarán de ahora en adelante como *C1* y *C2* respectivamente. Según [7] los valores son 1708 y 1101 respectivamente.

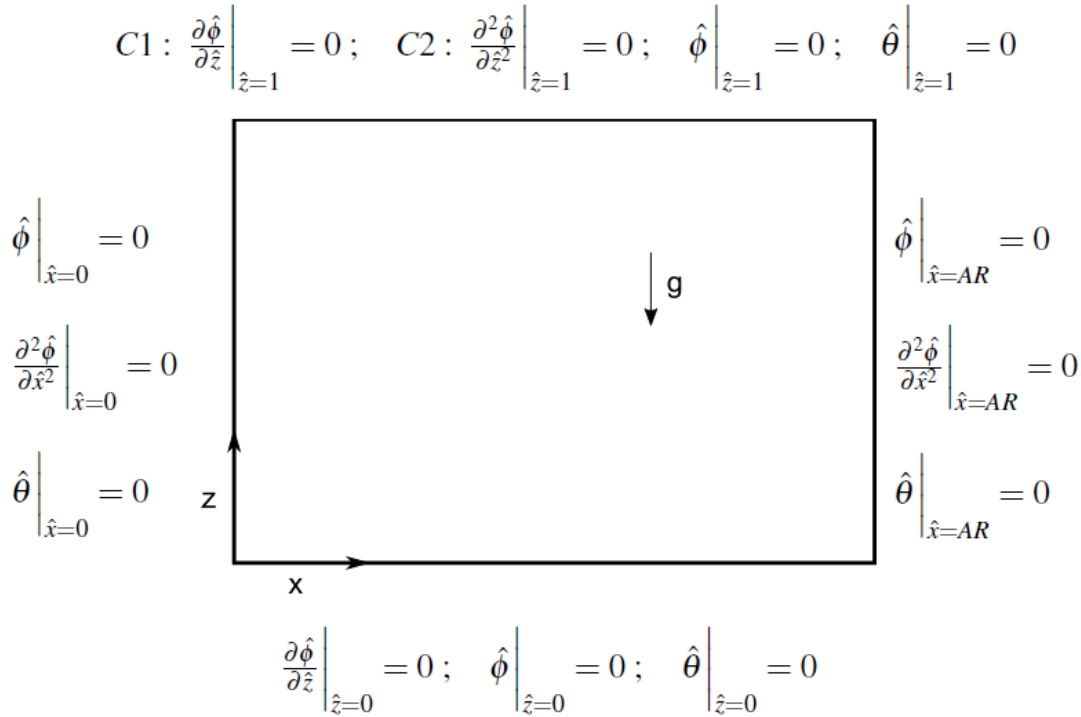
Las condiciones de contorno en la función de corriente son por tanto de pared en los contornos superior e inferior, y en los laterales se ha impuesto condiciones de superficie libre. En cuanto a la temperatura, en todos los contornos la condición impuesta es un valor nulo. Se muestra un esquema con todas ellas en 4.3', incluyendo las dos variantes en el contorno superior.

#### **Método de colocación con nodos de Chebyshev**

Para nodos de Chebyshev se ha calculado el número de Rayleigh crítico para relaciones de aspecto entre 1 y 40,  $N_z = 10, 20$  y  $30$  y  $N_x = 20, 30, \dots 100$ . Se observa en general una tendencia del valor crítico creciente a medida que la relación de aspecto disminuye. Más adelante se profundizará más en este fenómeno. Se busca una malla suficientemente precisa y a la vez que no requiera de un tiempo de computación excesivo. La bondad de la precisión de una malla se puede comprobar si los resultados no varían sustancialmente al refinar aun más dicha malla. Según este criterio, para el método de este apartado una malla con 80 y 20 elementos en las direcciones *x* y *z* es lo bastante precisa para relaciones de aspecto elevadas. Es razonable suponer que para relaciones menores la malla será también suficientemente precisa y se compararán el resto de mallas con esta. Esto deja de ser así si las celdas tienen un tamaño demasiado pequeño, provocando errores numéricos considerablemente altos. Esto es fácil de comprobar analizando la convergencia del código al calcular los autovalores.

Analizando los resultados para el caso en que el borde superior viene delimitado por una placa, se puede observar como para todas las combinaciones de alargamientos y número de nodos en *x* los resultados no varían para los valores de  $N_z$  calculados (como ejemplo ver 4.4a a 4.4b). Por tanto con el objetivo de un cálculo preciso pero a su vez rápido se va a escoger  $N_z = 10$ . En cuanto a la





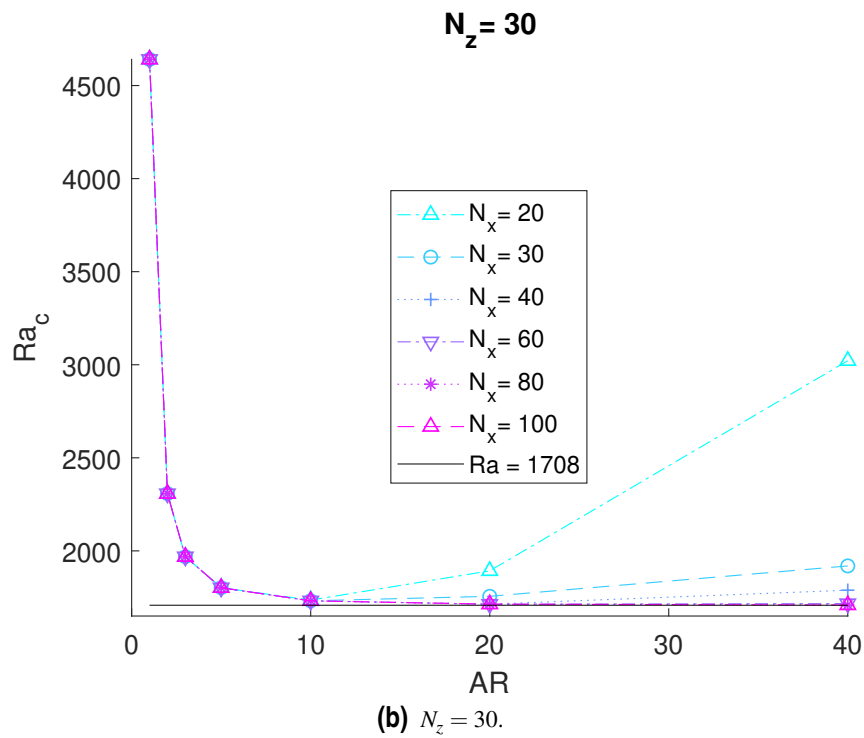
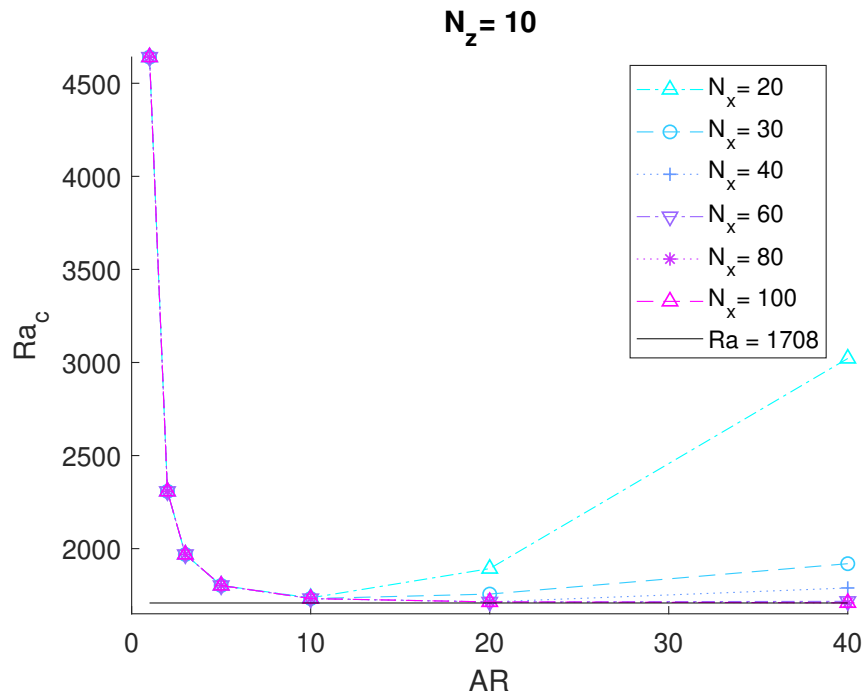
**Figura 4.3** Condiciones de contorno para análisis de sensibilidad de malla y método para C1 y C2.

dependencia con  $N_x$  es evidente que dado que la longitud adimensional en esta dirección es igual al alargamiento la precisión disminuye a medida que  $AR$  aumenta si se mantiene  $N_x$  fijo. Por tanto se va a buscar una función de la forma  $N_x(AR)$ . El criterio es el mismo que para  $N_z$ , se elige el menor número de nodos a partir del cual los resultados no varían.

- $AR = 40, N_x = 80$  y  $N_z = 10$ .
- $AR = 20, N_x = 40$  y  $N_z = 10$ .
- $AR = 10, N_x = 30$  y  $N_z = 10$ .
- $AR = 5, N_x = 20$  y  $N_z = 10$ .
- $AR = 3, N_x = 20$  y  $N_z = 10$ .
- $AR = 2, N_x = 20$  y  $N_z = 10$ .
- $AR = 1, N_x = 20$  y  $N_z = 10$ .

Por otro lado se repite el mismo procedimiento para el caso en que el borde superior está libre (ver figuras de 4.5a a 4.5b). Siguiendo el mismo razonamiento se escogen los siguientes valores:

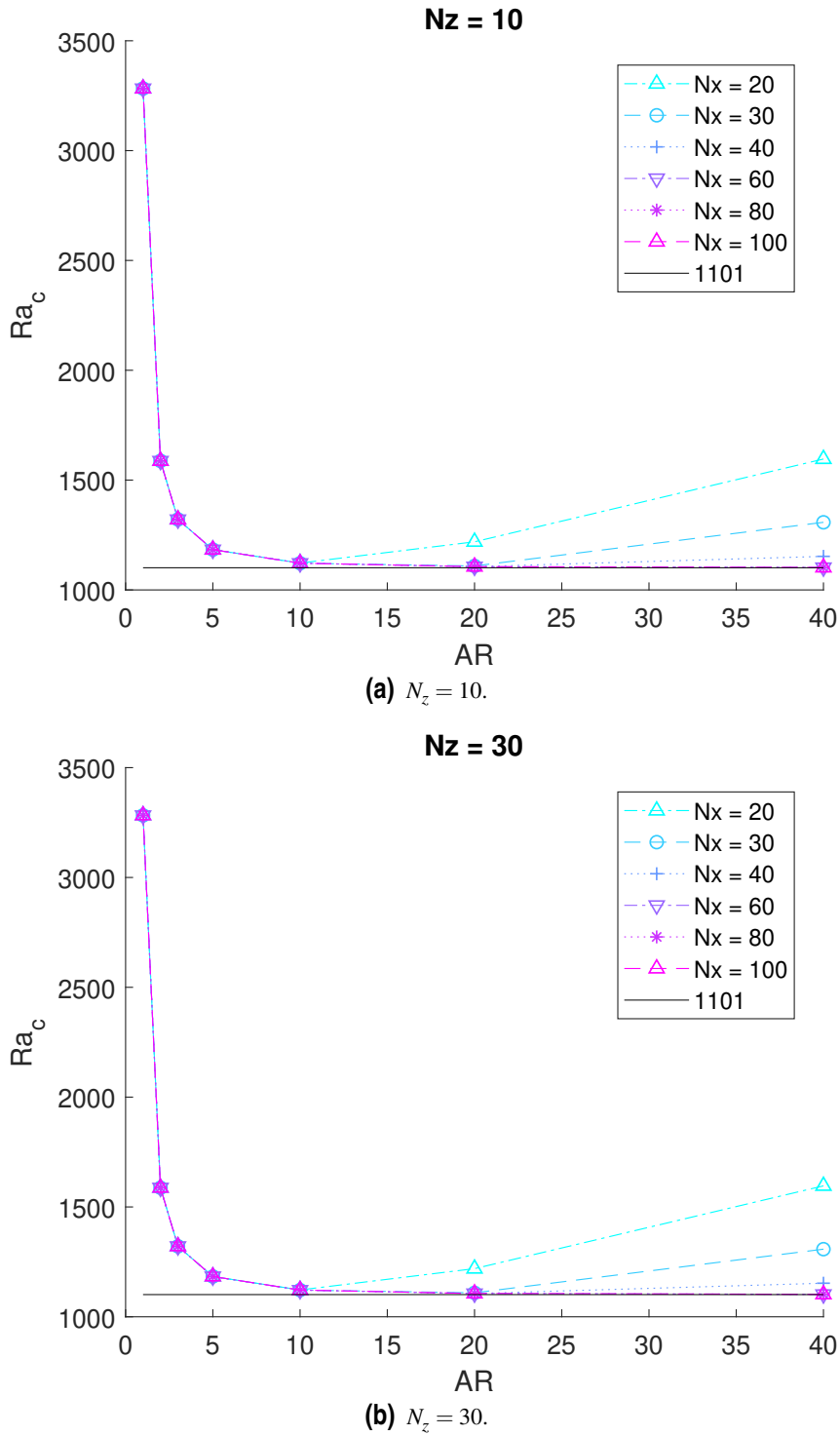
- $AR = 40, N_x = 60$  y  $N_z = 10$ .
- $AR = 20, N_x = 40$  y  $N_z = 10$ .
- $AR = 10, N_x = 30$  y  $N_z = 10$ .
- $AR = 5, N_x = 20$  y  $N_z = 10$ .
- $AR = 3, N_x = 20$  y  $N_z = 10$ .
- $AR = 2, N_x = 20$  y  $N_z = 10$ .
- $AR = 1, N_x = 20$  y  $N_z = 10$ .



**Figura 4.4** Rayleigh crítico en caso de placa en el contorno superior resuelto con nodos de Chebyshev y  $N_z = cte$ .

En ambos casos se obtienen los valores esperados para placas infinitas y cogiendo los valores más restrictivos de ambos casos se puede obtener una función polinómica de segundo orden  $N_x = N_x(AR)$ .

$$N_x = \lceil 18.82 + 0.64AR + 0.02AR^2 \rceil \quad (4.25)$$



**Figura 4.5** Rayleigh crítico en caso de contorno superior libre resuelto con nodos de Chebyshev y  $N_z = cte$ .

Los tiempos de resolución de todas las curvas mostradas son del orden de 3.4 y 3.1 horas para el caso de placa y borde libre en el contorno superior respectivamente (ver código B.8).

**Diferencias finitas**

El siguiente paso es realizar el mismo procedimiento pero utilizando las diferencias finitas calculadas según B.5. Inicialmente se han resuelto ambos casos con el mismo número de nodos, distribución de Chebyshev y utilizando cinco nodos para el cálculo de cada derivada, siendo por tanto la precisión para la derivada cuarta de orden uno. El tiempo de resolución para ambos casos es de 0.2 horas aproximadamente, mucho menor que el tiempo necesario usando nodos de Chebyshev. Se observa que la resolución de la malla, la distribución de nodos o el número de nodos circundantes usados para el cálculo de las derivadas no son adecuados para obtener los valores conocidos para alargamientos altos (ver figuras 4.6a y 4.6b).

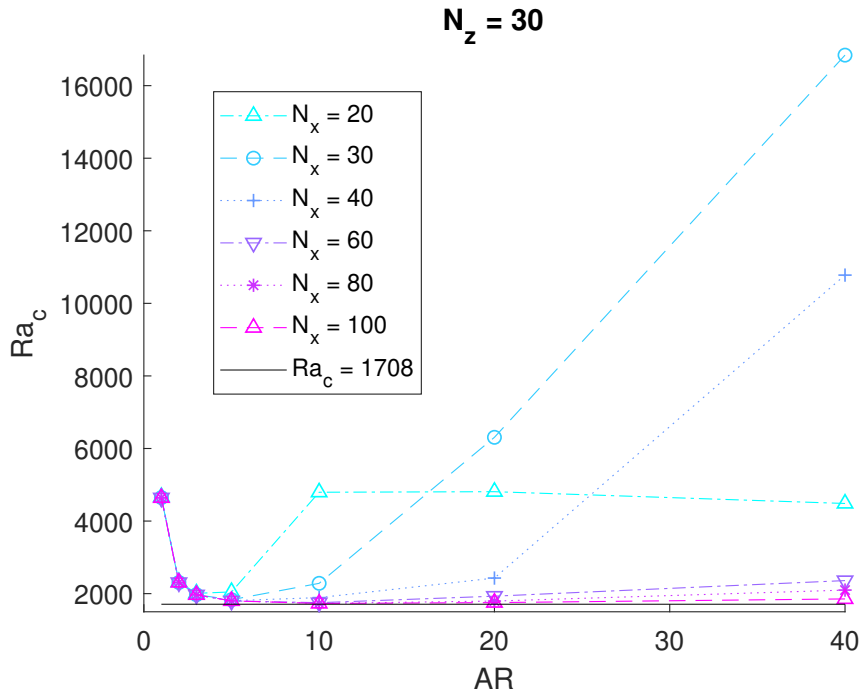
Por ello se ha optado inicialmente por aumentar el número de nodos (ver figuras de 4.7). El tiempo de resolución para los nuevos parámetros es de 1.2 horas para cada de los casos aproximadamente. Aún existiendo un número de elementos muy alto para algunas relaciones de aspecto el tiempo sigue siendo mucho menor que utilizando colocación con nodos de Chebyshev. Por otro lado se puede ver que para llegar a resolver con tanta precisión alargamientos altos es necesario aumentar el número de nodos en  $x$  de manera muy pronunciada. Además, mientras que resolviendo con Chebyshev los resultados no variaban con las diferentes mallas propuestas en  $z$ , en este caso sí que varían ligeramente, habiéndose escogido finalmente  $N_z = 30$ .

Para intentar minimizar estos efectos se realizan las mismas simulaciones pero en este caso variando la distribución de los nodos a una uniforme. Estos son similares a los obtenidos utilizando la distribución de Chebyshev tras haber refinado la malla, y el tiempo de cálculo es también muy parecido.

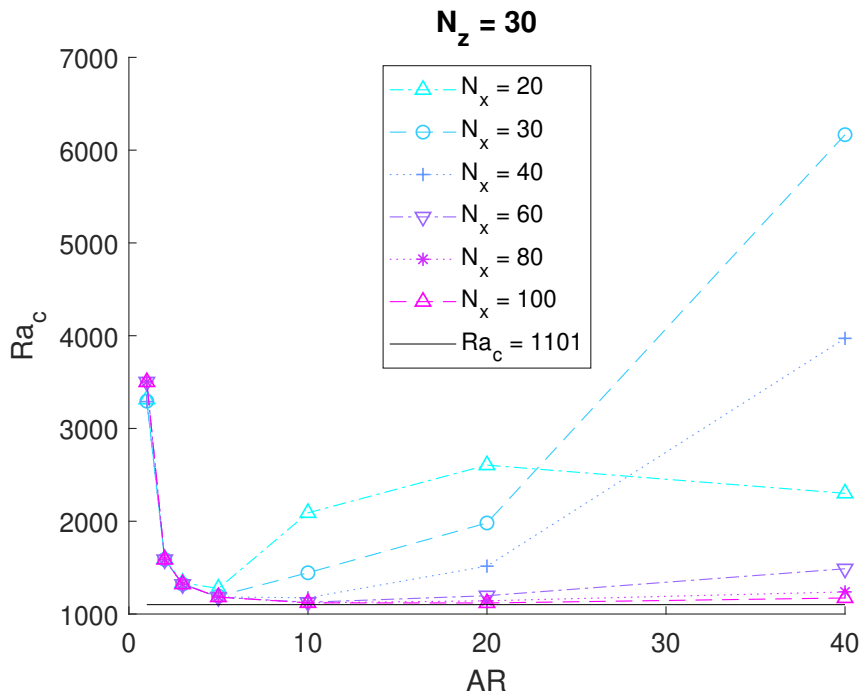
Viendo todos los resultados se puede afirmar que, resolviendo por diferencias finitas con cinco nodos circundantes, se obtienen resultados muy similares tanto con la distribución de nodos de Chebyshev como con los nodos equiespaciados. Sin embargo, la segunda opción puede ser interesante en vista a la resolución con contornos no rectos, puesto que pueden modelarse estos con mayor precisión en todo el dominio y no solo en los bordes. En cuanto al número de nodos se considera  $N_z = 30$  puesto que la variación de los resultados es mínima aumentando esta variable. Teniendo en cuenta esto se escogen los valores para  $N_x$  utilizando el mismo criterio que se usó para el método de colocación con nodos de Chebyshev, aunque en este caso pueda existir una mínima diferencia si se aumenta  $N_x$  sobre el valor escogido. Es importante tener en cuenta que en las siguientes secciones se resolverán dominios con relaciones de aspecto menores ya que no se buscará llegar al valor asintótico sino a casos más realistas que se den en la práctica. Se obtiene un polinomio para interpolar  $N_x$  como se hizo antes a partir de los resultados pero limitando el número de nodos a 400.

$$N_x = \min(400, \lceil 62 - 2.98AR + 1AR^2 \rceil) \quad (4.26)$$

Existe otra opción para aumentar la precisión de los cálculos sin aumentar el número de nodos total, y consiste en aumentar el número de nodos implicados en el cálculo de cada derivada en cada nodo. Se va a aumentar este a siete, quedando por tanto la precisión de la cuarta derivada de orden tres. En este caso las matrices siguen siendo dispersas aunque tienen un mayor número de elementos no nulos. EL tiempo de cálculo para los casos  $C1$  y  $C2$  ha sido de 0.5 horas aproximadamente aunque no se puede comparar ya que se han cambiado los parámetros. Se han simulado  $N_z = 20, 30$  y  $40$  y  $N_x$  los que se muestran en las figuras 4.8b y 4.8a. En cuanto a los resultados se aprecia una gran mejoría para relaciones de aspecto altas. Como ejemplo, para  $AR = 40$  se consiguen buenos resultados con 200 nodos mientras que para conseguir resultados similares utilizando cinco nodos hay que aumentar hasta 400 nodos. Sin embargo, para relaciones de aspecto menores que 10 los resultados son muy similares.



(a)  $N_z = 30$  para placa en el contorno superior.



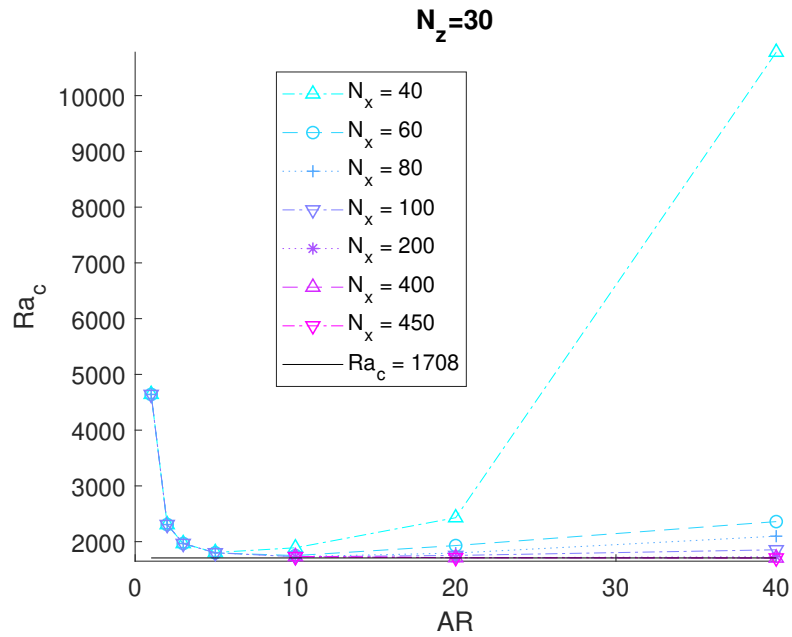
(b)  $N_z = 30$  para contorno superior libre.

**Figura 4.6** Rayleigh crítico para  $C1$  y  $C2$  calculado con diferencias finitas, distribución de nodos de Chebyshev y 5 nodos circundantes.

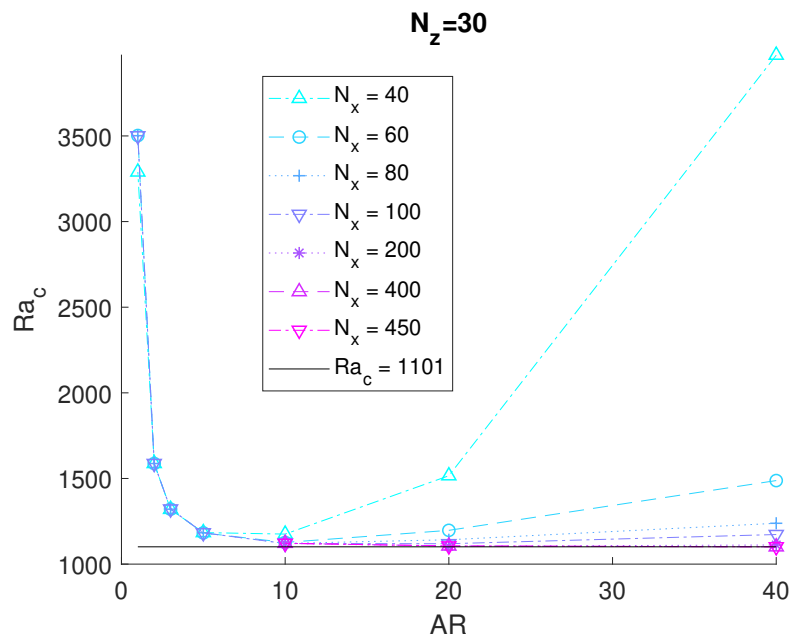
De la misma forma que se hizo antes se puede obtener el siguiente polinomio:

$$N_x = [27.31 + 5.64AR - 0.03AR^2] \tag{4.27}$$

Con el análisis hecho hasta ahora se ha comprobado que el uso de una distribución de nodos



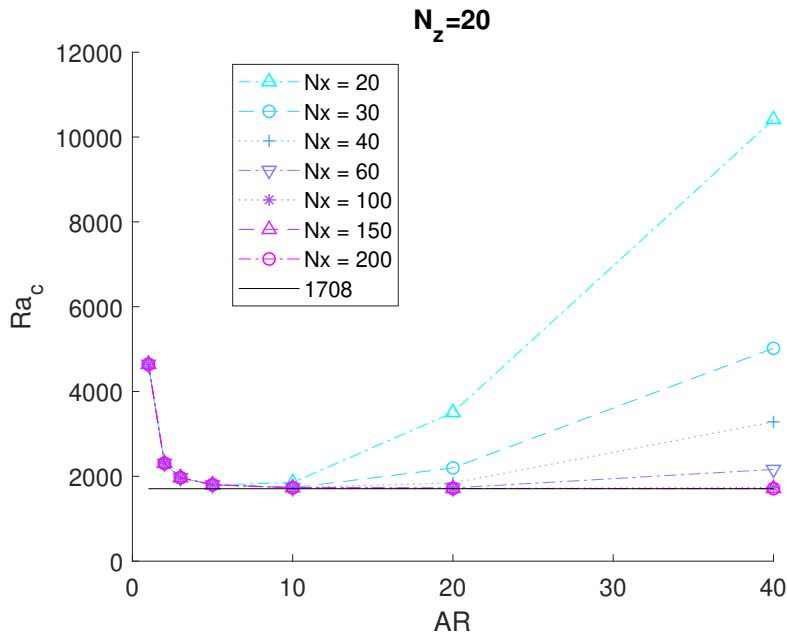
(a)  $N_z = 30$  para placa en el contorno superior.



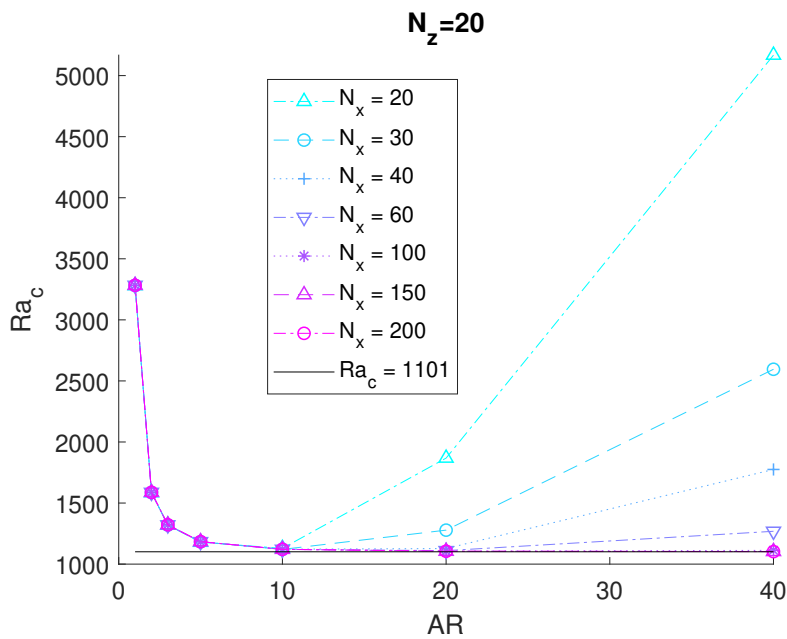
(b)  $N_z = 30$  para contorno superior libre.

**Figura 4.7** Rayleigh crítico para  $C1$  y  $C2$  calculado con diferencias finitas tras aumentar el número de nodos, distribución de nodos de Chebyshev y 5 nodos circundantes.

de Chebyshev para el método de diferencias finitas no presenta ninguna ventaja. Además, como se comprobará después, es útil que los nodos estén uniformemente distribuidos sobre todo si los contornos no van a ser rectos, para captar con fidelidad la geometría del dominio. En la siguiente sección se realizará un estudio más detallado utilizando los polinomios obtenidos en esta y para diferentes configuraciones de los métodos. Se observará la influencia de estas en los resultados de Rayleigh crítico y en los tiempos de computación necesarios.



(a)  $N_z = 20$  para placa en el contorno superior.



(b)  $N_z = 20$  para contorno superior libre.

**Figura 4.8** Rayleigh crítico para  $C1$  y  $C2$  calculado con diferencias finitas, distribución de nodos uniforme y 7 nodos circundantes.

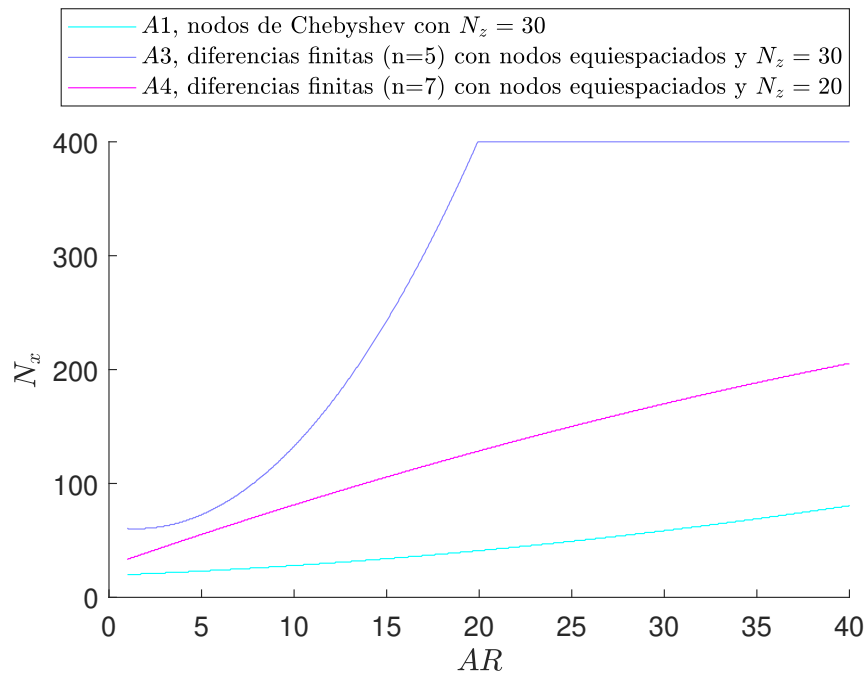
**Comparativa de colocación con nodos de Chebyshev y con diferencias finitas**

Una vez se ha estudiado la convergencia del valor de Rayleigh crítico en los dos casos propuestos en función de los parámetros de la malla, se van a comparar de manera más precisa los resultados así como el tiempo requerido para los distintos métodos y las mallas obtenidas anteriormente. En caso de buscar la evolución temporal, como se hará en el capítulo 5, de alguna variable del dominio, el tiempo de cálculo de autovalores no es interesante. Sin embargo, si es de interés el tiempo requerido para el cálculo de las matrices de derivadas, el montaje de las matrices a partir de las ecuaciones

**Tabla 4.1** Casos simulados para diferentes combinaciones de mallas y método de obtención de derivadas.

		A1	A2	A3	A4	A5
<b>Derivadas (nodos circundantes n para Dif. Fin.)</b>	x	Cheb.		Dif. Fin. (n = 5)	Dif. Fin. (n = 7)	
	z	Cheb.		Dif. Fin. (n = 5)	Dif. Fin. (n = 7)	
<b>Distribución nodal</b>	x	Cheb.		Uniforme	Uniforme	
	z	Cheb.		Uniforme	Uniforme	
<b>Número de nodos</b>	$N_x$	$\lceil 18.82 + 0.64AR + 0.02AR^2 \rceil$		$\lceil \min(400, 62 - 2.98AR + AR^2) \rceil$	$\lceil 27.31 + 5.64AR - 0.03AR^2 \rceil$	
	$N_z$	10	30	30	20	30

del problema y la factorización LU que se comentó en el esquema de resolución.



**Figura 4.9** Configuraciones de métodos y mallas.

Se muestran en 4.2 y 4.3 los resultados más significativos de cinco configuraciones de métodos y mallas para los casos C1 y C2 (pared o borde libre en el contorno superior).

En primer lugar, se observa que para todas las relaciones de aspecto las cinco configuraciones de resolución obtienen los mismos resultados, y que concretamente, en el caso en que  $AR$  es suficientemente grande, el valor es el dado en [7]. En cuanto a los tiempos de cálculo cabe resaltar que se muestran los mayores de todo el proceso de resolución. El siguiente bloque que puede llegar a requerir más tiempo es el que consiste en la formación de las matrices, incluyendo desde el cálculo de las derivadas en una dimensión a su extensión a dos dimensiones, cambio de variable incluido. En los casos más costosos (A3 y A4), este bloque no pasa de los seis segundos. Por ello se han omitido estos resultados.

Se observa el gran aumento de tiempo requerido para la factorización, para  $AR = 40$  y C1, en caso de hacerse uso de los nodos de Chebyshev y de pasar de 10 a 30 nodos en la dirección z



**Tabla 4.2** Rayleigh crítico y tiempos de computación [s] para C1 y diferentes métodos.

Factorización LU.						Promedio de cálculo de autovalores.					
AR	A1	A2	A3	A4	A5	AR	A1	A2	A3	A4	A5
<b>1</b>	0.02	0.90	7.52	0.39	1.33	<b>1</b>	0.04	0.32	0.18	0.06	0.10
<b>4</b>	0.03	1.23	9.02	1.22	4.13	<b>4</b>	0.03	0.43	0.19	0.11	0.22
<b>8</b>	0.05	1.69	32.14	3.78	11.99	<b>8</b>	0.04	0.59	0.29	0.17	0.35
<b>40</b>	1.61	54.70	1822.69	75.12	240.18	<b>40</b>	0.65	10.97	1.37	0.55	1.30

Rayleigh crítico.

AR	A1	A2	A3	A4	A5
<b>1</b>	4640.16	4640.16	4637.23	4640.16	4639.18
<b>4</b>	1854.03	1854.03	1853.05	1854.03	1854.03
<b>8</b>	1744.65	1744.65	1743.68	1744.65	1744.65
<b>40</b>	1709.50	1709.50	1708.52	1709.50	1709.50

(manteniendo fijo  $N_x$ ), multiplicándose por más de 30. Lo mismo se observa en el caso de diferencias finitas con siete nodos circundantes, al pasar de  $N_z$  20 a 30, multiplicándose por tres. En efecto, supuesta una malla de  $m \times n$  nodos, la matriz final a la cual hay que aplicar la factorización LU es  $(2m \times n) \times (2m \times n)$ . Por tanto, el hecho de poder tener pocos nodos en una dirección disminuye el tiempo de computación considerablemente.

Las diferencias están relacionadas no solo con el número de nodos en dirección  $x$  y el aumento de nodos en  $z$ , sino también con el carácter disperso o lleno de las matrices. También puede verse el alto coste de factorización en el caso de A3, que pese a tratar con matrices dispersas, el hecho de tener una alta cantidad de nodos es mucho más notorio.

Por otro lado, en cuanto al tiempo medio de cálculo de autovalores, resulta muy llamativa la gran diferencia entre tratar con matrices dispersas o llenas. Esto se observa con claridad para altas relaciones de aspecto comparando A2 con cualquiera de los casos de diferencias finitas (A3, A4 y A5), concretamente comparando con A4, en el que aún siendo el número de nodos mucho mayor, el tiempo de cálculo es aproximadamente 8 veces menor.

Por último, se puede reflexionar acerca de la precisión en el método de diferencias finitas. La precisión para cada derivada calculada, usando siete nodos circundantes en lugar de cinco, aumenta dos órdenes respecto al tamaño de la malla. Se ha comprobado que para un análisis lineal ambas pueden ofrecer los mismos resultados refinando la malla adecuadamente. No obstante, el tiempo de computación requerido es menor manteniendo un número total de nodos menor y aumentando el número de elementos distinto de ceros de las matrices, ya que en este caso, solo se añaden dos diagonales.

**Tabla 4.3** Rayleigh crítico y tiempos de computación [s] para C1 y diferentes métodos.

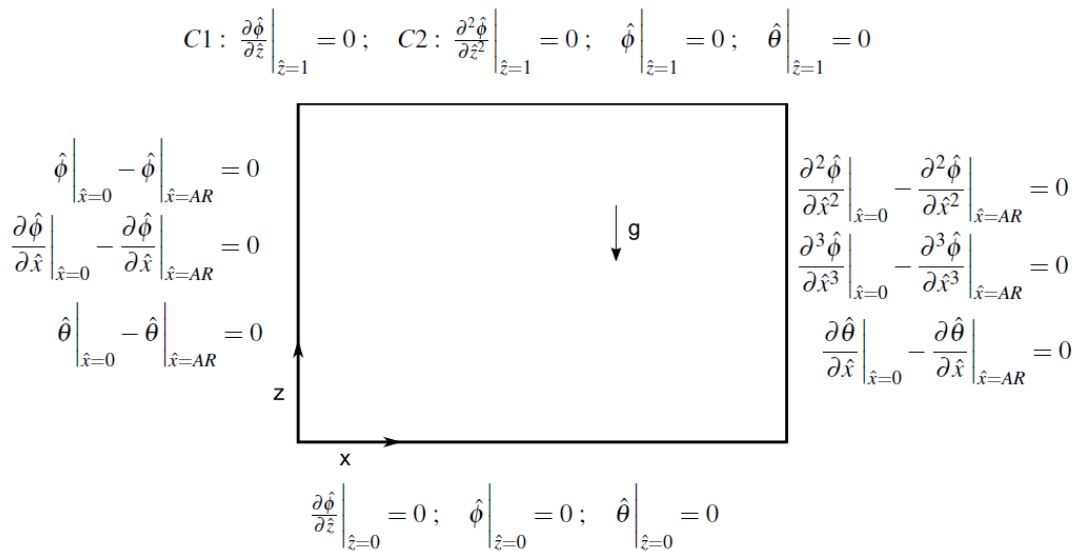
Factorización LU.						Promedio de cálculo de autovalores.					
AR	A1	A2	A3	A4	A5	AR	A1	A2	A3	A4	A5
1	0.02	0.74	7.45	0.39	1.34	1	0.02	0.31	0.21	0.06	0.13
4	0.03	1.22	9.62	1.37	4.45	4	0.03	0.48	0.18	0.12	0.25
8	0.05	1.69	32.54	3.74	11.05	8	0.03	0.69	0.34	0.20	0.44
40	1.60	54.07	1810.55	74.83	240.80	40	0.79	13.72	1.87	1.16	2.51

Rayleigh crítico.

AR	A1	A2	A3	A4	A5
1	3282.74	3282.74	3275.90	3282.74	3281.76
4	1228.05	1229.03	1226.10	1228.05	1228.05
8	1132.35	1132.35	1130.39	1132.35	1132.35
40	1102.07	1102.07	1100.12	1102.07	1102.07

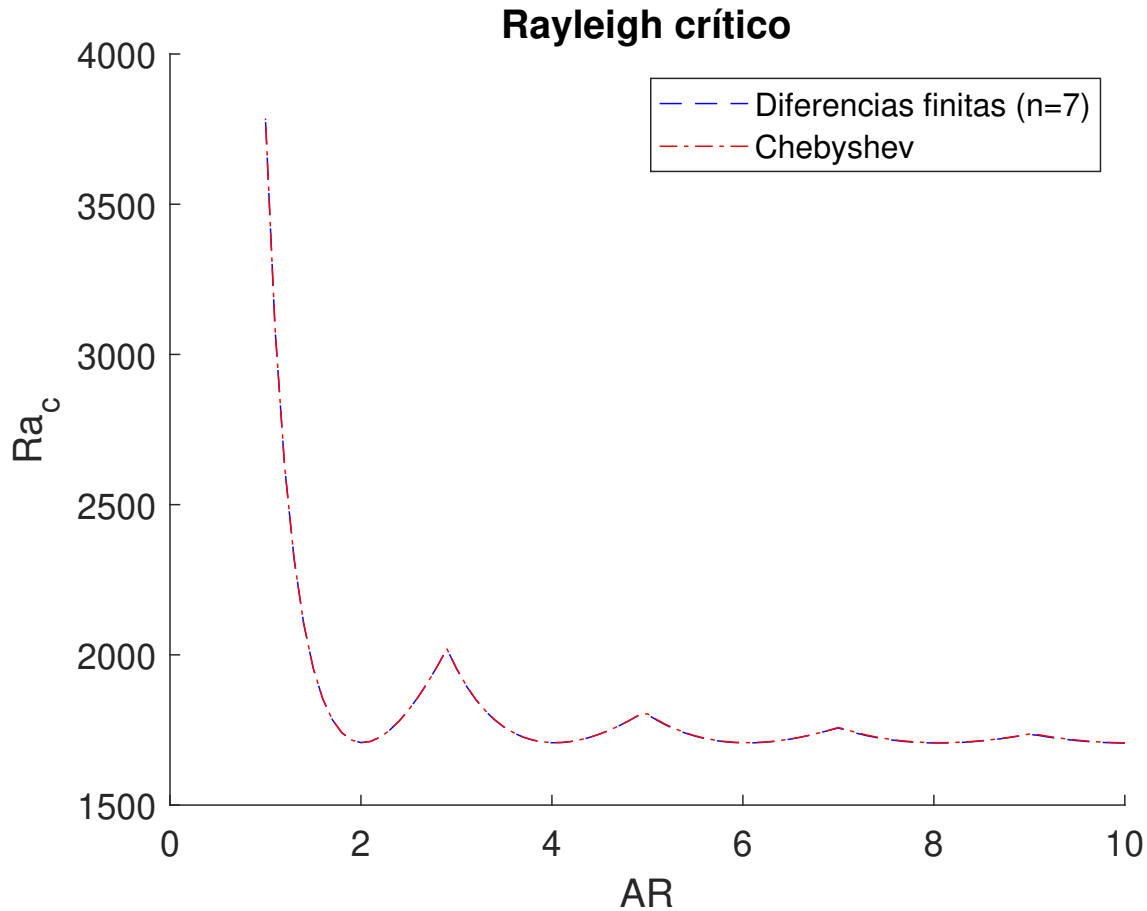
**Problema de Rayleigh. Condiciones de contorno periódicas (B.10 y B.11)**

En esta sección se va a estudiar el caso concreto en que las condiciones de contorno son exactamente las que existen en el problema de Rayleigh-Bénard. Los contornos superior e inferior se modelan como pared con temperaturas impuestas como ya se ha hecho anteriormente. En cuanto a los contornos laterales, en ellos se imponen condiciones periódicas, tantas en cada variable como el mayor orden de derivada que aparezca en las ecuaciones. Para ello se igualan tanto el valor de las magnitudes como el de las sucesivas derivadas (hasta completar las condiciones de contorno que haya que imponer para cada variable) de los contornos y subcontornos con sus respectivas parejas (nodos del contorno derecho con el izquierdo). Un esquema puede verse en 4.10.



**Figura 4.10** Condiciones de contorno de Rayleigh-Bénard.

Se han empleado dos métodos diferentes, uno de diferencias finitas con siete nodos circundantes y otro con nodos de Chebyshev (A1 y A4) respectivamente.



**Figura 4.11** Rayleigh crítico para condiciones de periodicidad en contornos verticales (C1).

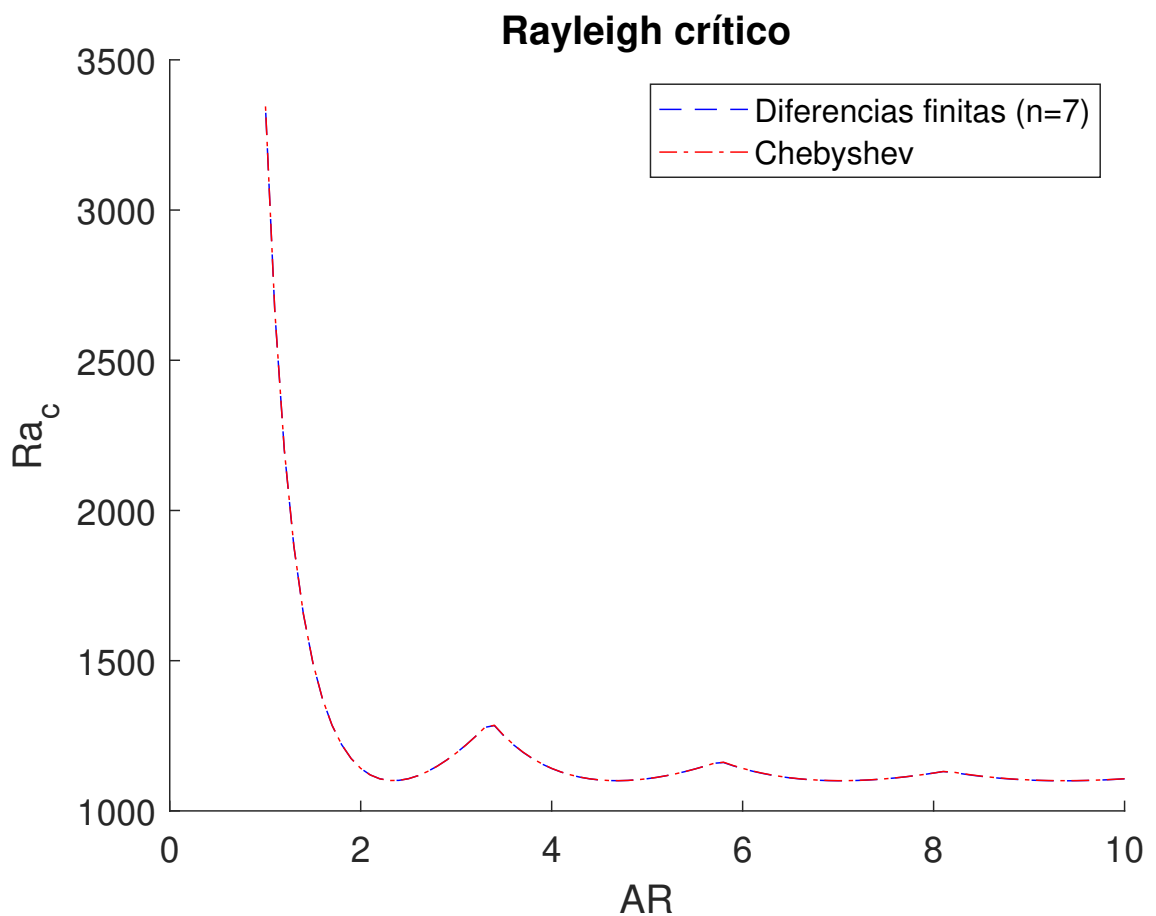
En la figura 4.11 cabría esperar una recta horizontal, sin embargo, aparecen picos cuyos valores aumentan aparentemente con cierta periodicidad, aunque estos son menos pronunciados a medida que la relación de aspecto es mayor. Los valores esperados se encuentran para relaciones de aspecto que son aproximadamente múltiplos de dos. Como pasaba para los casos C1 y C2, a medida que esta se lleva al límite, el efecto de la creación de una nueva celda respecto al resto disminuye y se tiende asintóticamente al valor esperado.

Debido a las condiciones de contorno impuestas, la solución siempre contendrá un mínimo de dos rollos de Bénard, estando uno de ellos dividido en dos mitades, una cortada por el contorno izquierdo y otra por el derecho. La única solución posible imponiendo igualdad de valor en las magnitudes y de las sucesivas derivadas respecto a  $x$  en los contornos laterales es que estos corten por la mitad una celda. Para que el sistema represente realmente al problema original de Rayleigh-Bénard, la longitud de este debe ser aquella que permita que todos los rollos de convección tengan las dimensiones reales. Es decir, acorde a [7] para el fluido entre dos placas infinitas, el número de onda asociado al valor crítico es  $3.117 1/m$  y los planos de simetría entre rollos se encuentran en:

$$x = \left(m + \frac{1}{2}\right) \frac{\pi}{k} \quad m \in \mathbb{N}_{>0} \quad (4.28)$$

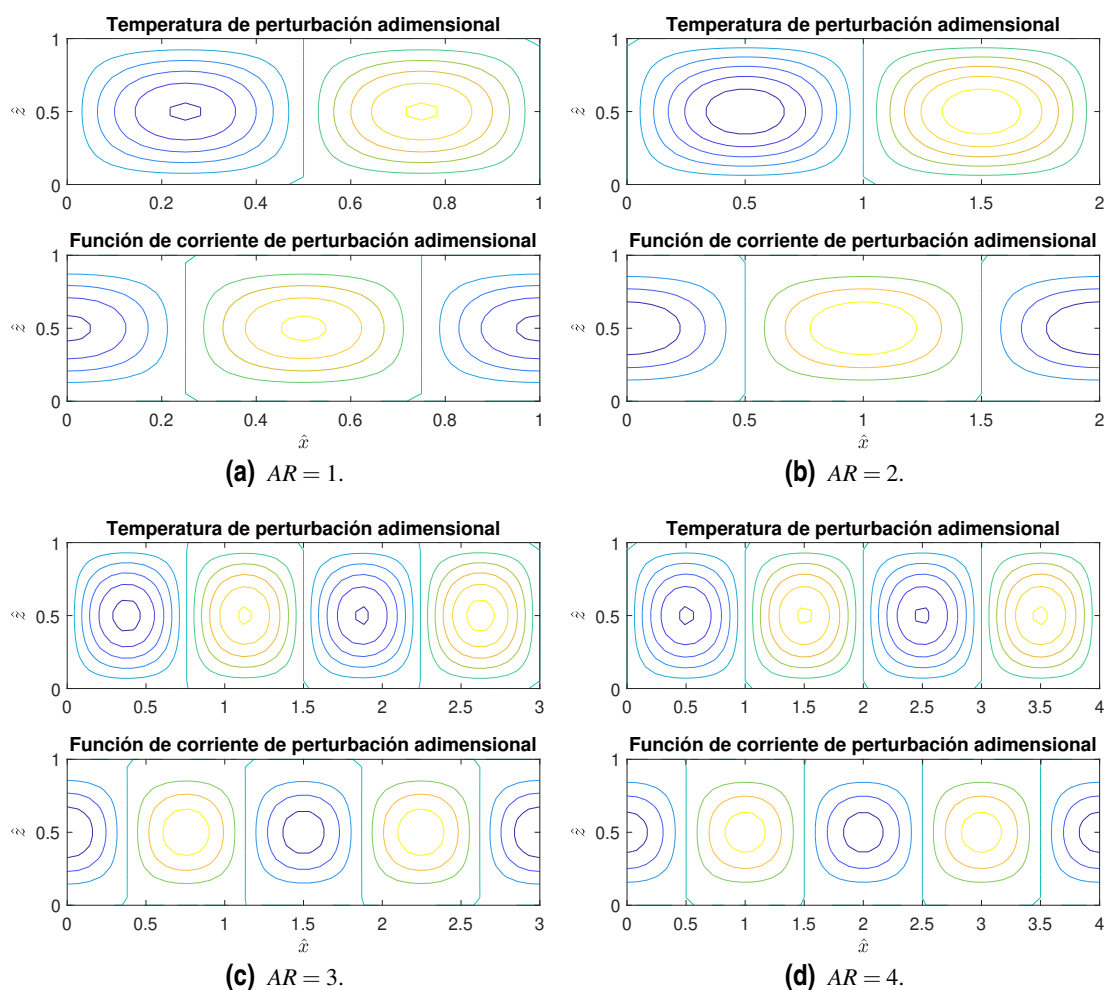
Esto es, la longitud de cada rollo es aproximadamente 1.008 metros. Esto explica que los valores críticos solo sean iguales al de placas infinitas si la relación de aspecto son aproximadamente múltiplos enteros de  $2 \cdot 1.008$ . El dominio siempre debe contener un número par de celdas (contando las dos mitades como una). Sería imposible obtener una sola celda imponiendo que el valor de la función de corriente del contorno izquierdo sea igual al del contorno derecho, ya que las celdas contiguas giran en sentido contrario, y el valor de la función de corriente no podría tener el mismo valor, excepto si fuese nulo, en cuyo caso no habría convección.

Profundizando más, existe una simetría local respecto a los máximos locales, encontrándose estos justo a la misma distancia de los valores mínimos, cuya diferencia es la existencia de dos celdas más o menos. Es decir, cuando la relación de aspecto difiere la misma cantidad de la que habría para dos celdas más o menos, el problema calculado estará lo más lejos posible del problema de Rayleigh que intenta simular. Por otro lado, se muestran a modo de ejemplo los autovectores asociados para algunas relaciones de aspecto para el caso con placa en el borde superior en las figuras 4.13a a 4.13d.



**Figura 4.12** Rayleigh crítico para condiciones de periodicidad en contornos verticales (C2).

Por último se muestra también la dependencia del valor crítico con la relación de aspecto para el caso en que el contorno superior es libre en 4.12, observándose exactamente los mismos fenómenos.



**Figura 4.13** Autovectores asociados a  $Ra_c$  para condiciones de periodicidad en contornos verticales (C1).

#### 4.4 Validación de los métodos para contornos no rectos (B.12 y B.13)

Hasta ahora no se han tenido en cuenta contornos no rectos y por tanto no se ha podido validar el cálculo de las derivadas realizando el cambio de variable.

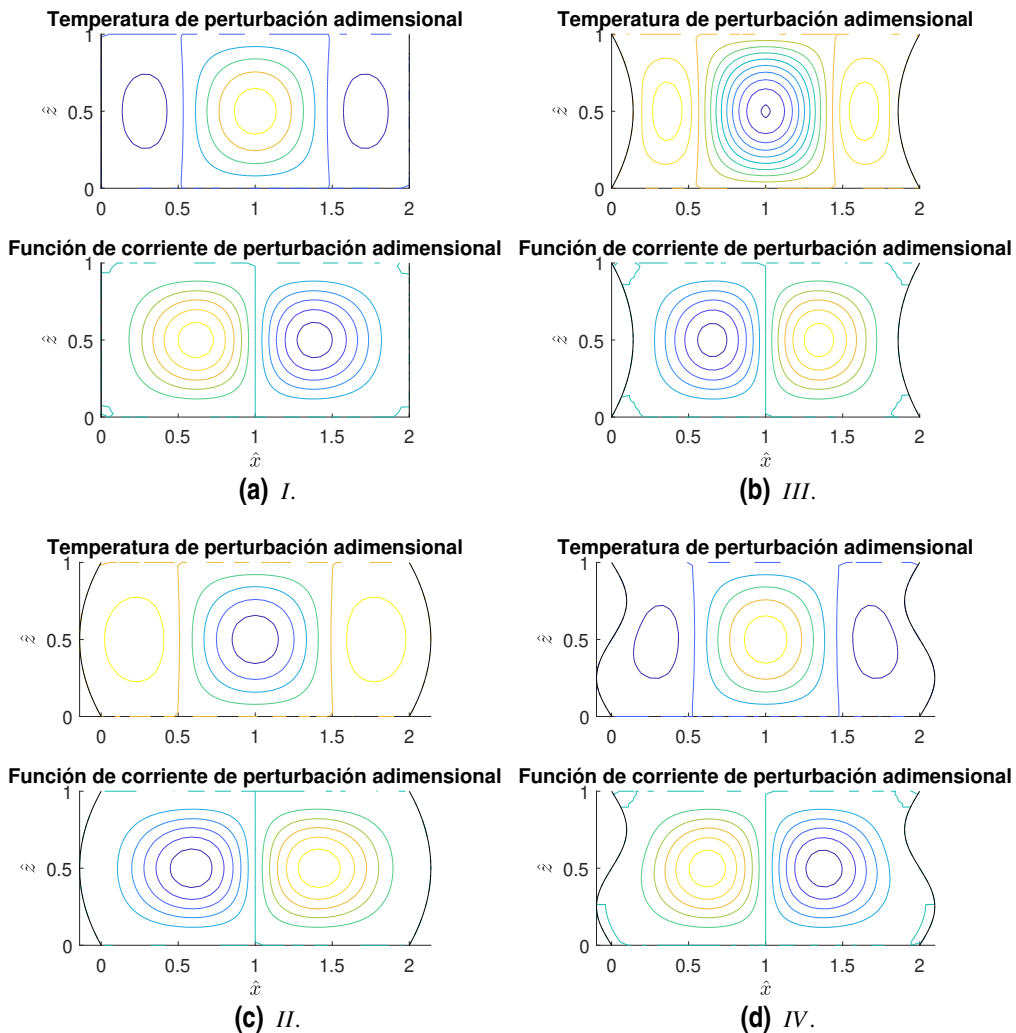
En la tabla 4.4 se observan los valores de Rayleigh crítico para diferentes contornos calculados tanto con diferencias finitas como con Chebyshev mediante el análisis lineal, y se comparan con los valores dados en [23]. Con el fin de captar la geometría con precisión el número de nodos en la dirección  $z$  para ambos métodos se ha establecido en  $N_z = 50$ . En cuanto a los nodos en dirección  $x$ , utilizando los polinomios hallados anteriormente (casos A2 y A4) se han establecido en  $N_x = 20$  y  $N_x = 39$  para Chebyshev y diferencias finitas respectivamente. En el segundo caso se han usado siete nodos circundantes y una malla equiespaciada.

Las condiciones de contorno impuestas en [23] son las mismas en todos los casos, la única diferencia es la geometría. Todos los contornos se modelan como pared, es decir,  $\hat{u} = \hat{v} = 0$  en todos ellos. En el artículo la transformación del dominio se basa en una transformación elíptica, y por tanto en los contornos la intersección con la malla es ortogonal. Esto hace que las condiciones de contorno puedan imponerse respecto a las variables  $X$  y  $Z$  del dominio transformado para imponerlas en términos de las velocidades normal y tangencial a los contornos.



**Tabla 4.4** Comparación de número de Rayleigh crítico para contornos laterales no rectos.

		Diferencias finitas (n=7)	Chebyshev	Resultados de [23]
$x_r = AR + 4Az(z - 1)$ $x_l = 4Az(z - 1)$	$A = 0$ (I)	2384.9	2384.9	2385
	$A = -0.04$ m	2538.9	2538.9	2503
	$A = -0.07$ m (II)	2707.6	2707.6	2618
	$A = 0.04$ m	2285.4	2285.5	2285
	$A = 0.07$ m (III)	2234.9	2235.0	2236
$x_r = AR + A\frac{2\pi}{\lambda}z\sin(\frac{2\pi}{\lambda}z)$ $x_l = A\frac{2\pi}{\lambda}z\sin(\frac{2\pi}{\lambda}z)$	$\lambda = 1$ m, $A = -0.021$ m	2393.6	2393.7	2394
	$\lambda = 1$ m, $A = -0.049$ m (IV)	2431.4	2431.4	2430



**Figura 4.15** Autoeigenvectores asociados a  $Ra_c$  para geometrías con contornos laterales no rectos, usando diferencias finitas.





## 5 Análisis completo de la solución

---

En el capítulo anterior se han validado algunos de los métodos desarrollados haciendo uso de un resultado fundamental como es el número de Rayleigh crítico, que determina el comienzo de la convección en el sistema. Para ello se linealizó en torno al estado de equilibrio y se supusieron soluciones exponenciales en el tiempo. Realizando un análisis de autovalores se validaron el método de diferencias finitas y el de nodos de Chebyshev, se obtuvieron criterios para escoger el número de nodos y su distribución, se estudiaron los tiempos de computación de las diferentes partes del código, y se comprobó que el cambio de variable funciona correctamente para modelar contornos no rectos.

Para empezar este capítulo, se busca validar el método semi-implícito con el que calcular las derivadas temporales, ya que en el análisis anterior no figuraban estas. Por tanto, el primer paso es obtener el sistema matricial que permitirá calcular la solución completa del problema, para lo que es útil adimensionalizar la temperatura de otra forma. Para la validación, no solo se tendrán en cuenta la visualización de los campos de temperatura y función de corriente, sino que también se observará como varía el flujo de calor en el contorno inferior con el número de Rayleigh impuesto. De esta manera se puede calcular también el valor crítico, el cual se comparará con el obtenido en el capítulo anterior. Este método no es muy utilizado en la bibliografía ya que requiere más tiempo de cálculo. Sin embargo puede servir en este caso para validar la integración temporal. Si solo se quisiera obtener el valor crítico, lo más adecuado sería proceder como en el capítulo 4. Una vez se haya validado la integración temporal, y estudiado como afecta el valor del paso temporal escogido, se resolverán casos de convección forzada, tanto con contornos rectos, como un con un contorno variable.

### 5.1 Desarrollo de las ecuaciones completas

Se parte de las ecuaciones 2.19 y 2.20 y se tienen en cuenta las mismas relaciones para adimensionalizar (4.6) que en el capítulo anterior salvo en el caso de la temperatura.

$$\hat{T} = \frac{g\beta H^3}{\nu\alpha}(T - T_0), \quad \text{siendo} \quad T_0 = \frac{T_1 + T_2}{2} \quad (5.1)$$

Para la temperatura se escogerá una referencia concreta. De esta forma, al adimensionalizar  $T_1$  y  $T_2$  se obtienen expresiones del número de Rayleigh no apareciendo este en las ecuaciones que dominan el problema como se verá a continuación. De las condiciones de contorno que se van a usar, estas son las únicas que varían respecto a las indicadas en la sección anterior. Llevando a cabo

esta adimensionalización queda el sistema de ecuaciones que sigue:

$$\frac{\partial(\hat{V}^2\hat{\psi})}{\partial\hat{t}} + \frac{\partial\hat{\psi}}{\partial\hat{z}} \frac{\partial(\hat{V}^2(\hat{\psi}))}{\partial\hat{x}} - \frac{\partial\hat{\psi}}{\partial\hat{x}} \frac{\partial(\hat{V}^2(\hat{\psi}))}{\partial\hat{z}} = Pr\hat{V}^4\hat{\psi} - Pr\frac{\partial\hat{T}}{\partial\hat{x}} \quad (5.2)$$

$$\frac{\partial\hat{T}}{\partial\hat{t}} + \frac{\partial\hat{\psi}}{\partial\hat{z}} \frac{\partial\hat{T}}{\partial\hat{x}} - \frac{\partial\hat{\psi}}{\partial\hat{x}} \frac{\partial\hat{T}}{\partial\hat{z}} = \hat{V}^2\hat{T} \quad (5.3)$$

Teniendo en cuenta el método semi-implícito y las matrices para calcular las derivadas espaciales del capítulo 3 se obtiene el sistema matricial 5.4. Cabe destacar la manera de considerar los términos no lineales que tiene el método semi-implícito, aproximándolos por sus valores en el instante de tiempo anterior.

$$\begin{bmatrix} \hat{V}^2 - \hat{\delta}_t Pr \hat{V}^4 & \hat{\delta}_t Pr \frac{\partial}{\partial \hat{x}} \\ 0 & I - \hat{\delta}_t \hat{V}^2 \end{bmatrix} \begin{bmatrix} \hat{\psi} \\ \hat{T} \end{bmatrix} = \begin{bmatrix} \hat{\delta}_t \left( \frac{\partial \hat{\psi}}{\partial \hat{z}} \frac{\partial}{\partial \hat{x}} (\hat{V}^2 \hat{\psi}) - \frac{\partial \hat{\psi}}{\partial \hat{x}} \frac{\partial}{\partial \hat{z}} (\hat{V}^2 \hat{\psi}) \right) \Big|_{t-1} + \hat{V}^2 \hat{\psi} \Big|_{t-1} \\ \hat{\delta}_t \left( \frac{\partial \hat{\psi}}{\partial \hat{z}} \frac{\partial \hat{T}}{\partial \hat{x}} - \frac{\partial \hat{\psi}}{\partial \hat{x}} \frac{\partial \hat{T}}{\partial \hat{z}} \right) \Big|_{t-1} + \hat{T} \Big|_{t-1} \end{bmatrix} \quad (5.4)$$

Para la resolución del sistema de ecuaciones a lo largo del tiempo, es conveniente realizar una factorización  $LU$  de la matriz de coeficientes. Esto supone un gran mejora en tiempo de computación, tanto si las matrices de derivadas se han obtenido con nodos de Chebyshev como si se ha hecho con diferencias finitas. El tiempo de factorización depende de si la matriz es dispersa o no, de su número de condición y de su tamaño. No obstante, el tiempo de resolución del sistema disminuye considerablemente en cada paso de tiempo, y puesto que normalmente se va a resolver un número elevado siempre compensa (en una decena de pasos temporales ya supone un ahorro de tiempo de cálculo). La idea es resolver tres sistemas lineales con matrices de coeficientes triangulares o de permutación.

$$Ax = b \quad (5.5)$$

$$x = A \setminus b \quad \text{Resolución de un sistema lineal con matriz de coeficientes general} \quad (5.6)$$

$$PA = LU \quad \text{Factorización } LU, \text{ siendo } L, P \text{ y } U \text{ dispersas} \quad (5.7)$$

$$x = U \setminus (L \setminus (Pb)) \quad \text{Resolución de tres sistemas lineales con matrices dispersas} \quad (5.8)$$

### 5.1.1 Condiciones de contorno

Para las condiciones de contorno el procedimiento es el mismo que el del capítulo anterior. Dados los órdenes superiores de derivadas que aparecen en las ecuaciones, se requieren cuatro y dos condiciones de contorno en cada dirección para la función de corriente y la temperatura respectivamente. Cabe resaltar que en el capítulo 4 se resolvían las magnitudes de perturbación respecto al estado de equilibrio. Por ello, se imponían condiciones nulas en todas las condiciones de contorno. En este caso, hay que imponer el valor que corresponda a cada una.

En esta sección, para la temperatura se impondrá el valor de esta o condiciones periódicas. En el primer caso, con la nueva adimensionalización, los valores dependen del número de Rayleigh. En el capítulo 4 este aparecía en las ecuaciones y con esta nueva forma de adimensionalizar aparece solo en las condiciones de contorno.

$$\hat{T}_1 = \frac{g\beta H^3}{\nu\alpha}(T - T_0) = \frac{Ra}{2} \quad (5.9)$$

$$\hat{T}_2 = \frac{g\beta H^3}{\nu\alpha}(T - T_0) = -\frac{Ra}{2} \quad (5.10)$$

$$\text{Solución de equilibrio} \Rightarrow \hat{T}(\hat{z}) = \hat{T}_1 + (\hat{T}_2 - \hat{T}_1)\hat{z} = \frac{Ra}{2}(1 - 2\hat{z}) \quad (5.11)$$

Para la función de corriente, las condiciones se mantienen de manera general pero en la variable completa. No obstante, cabe destacar los casos en que se impone un caudal de flujo. En estos casos, la función de corriente en las paredes sigue siendo constante, aunque en cada contorno dependerá del flujo impuesto. Se deduce a continuación el caso en que los contornos superior e inferior tienen condición de pared y temperatura  $T_2$  y  $T_1$ , y se impone un caudal y condiciones periódicas en los contornos izquierdo y derecho.

$$\begin{cases} \text{Contorno inferior} \left\{ \begin{array}{l} \hat{u} = 0 \Rightarrow \frac{\partial \hat{\psi}}{\partial \hat{z}} \Big|_{\hat{z}=0} = 0 \\ \hat{v} = 0 \Rightarrow \frac{\partial \hat{\psi}}{\partial \hat{x}} \Big|_{\hat{z}=0} = 0 \quad \forall \hat{x} \Rightarrow \hat{\psi} = \hat{\psi}(\hat{z}) \Rightarrow \hat{\psi}(\hat{z}=0) = \hat{\psi}_{inf} = 0 \\ \hat{T} \Big|_{\hat{z}=0} = \frac{Ra}{2} \end{array} \right. \\ \\ \text{Contorno superior} \left\{ \begin{array}{l} \hat{u} = 0 \Rightarrow \frac{\partial \hat{\psi}}{\partial \hat{z}} \Big|_{\hat{z}=1} = 0 \\ \hat{v} = 0 \Rightarrow \frac{\partial \hat{\psi}}{\partial \hat{x}} \Big|_{\hat{z}=1} = 0 \quad \forall \hat{x} \Rightarrow \hat{\psi} = \hat{\psi}(\hat{z}) \Rightarrow \hat{\psi}(\hat{z}=1) = \hat{\psi}_{sup} = \frac{U_\infty}{\alpha} \\ \hat{T} \Big|_{\hat{z}=1} = -\frac{Ra}{2} \end{array} \right. \\ \\ \text{Contorno izquierdo} \left\{ \begin{array}{l} \hat{\psi} \Big|_{\hat{x}=0} = \hat{\psi} \Big|_{\hat{x}=AR} \\ \frac{\partial \hat{\psi}}{\partial \hat{x}} \Big|_{\hat{x}=0} = \frac{\partial \hat{\psi}}{\partial \hat{x}} \Big|_{\hat{x}=AR} \\ \hat{T} \Big|_{\hat{x}=0} = \hat{T} \Big|_{\hat{x}=AR} \end{array} \right. \\ \\ \text{Contorno derecho} \left\{ \begin{array}{l} \frac{\partial^2 \hat{\psi}}{\partial \hat{x}^2} \Big|_{\hat{x}=0} = \frac{\partial^2 \hat{\psi}}{\partial \hat{x}^2} \Big|_{\hat{x}=AR} \\ \frac{\partial^3 \hat{\psi}}{\partial \hat{x}^3} \Big|_{\hat{x}=0} = \frac{\partial^3 \hat{\psi}}{\partial \hat{x}^3} \Big|_{\hat{x}=AR} \\ \frac{\partial \hat{T}}{\partial \hat{x}} \Big|_{\hat{x}=0} = \frac{\partial \hat{T}}{\partial \hat{x}} \Big|_{\hat{x}=AR} \end{array} \right. \end{cases} \quad (5.12)$$

### 5.1.2 Condiciones iniciales

Además de las condiciones de contorno es necesario imponer una condición inicial en cada nodo, tanto para la temperatura como para la función de corriente. A lo largo de este capítulo se han mantenido las mismas condiciones iniciales. En ambas variables se ha supuesto una condición inicial que viene dada por la que habría en equilibrio (en caso de que el resto de condiciones de contorno lo permitiesen) más una pequeña perturbación. En caso de que la geometría no sea compatible con una solución de equilibrio se supondrá que esta es la propia tras haber transformado el dominio desde el caso rectangular.

En el caso de la función de corriente la condición inicial será por tanto una pequeña perturbación cuyo valor depende de la posición si no hay caudal, y en caso de haberlo, la perturbación se suma a la función de corriente hallada en el dominio rectangular teniendo en cuenta el caudal. Para la temperatura será una función lineal con la coordenada  $\hat{z}$  que varía desde  $\hat{T}_1$  en  $\hat{z} = 0$  hasta  $\hat{T}_2$  en  $\hat{z} = H$  más otra función que depende de la posición. No se ha realizado ningún estudio sobre como afectan las condiciones iniciales o la perturbación a la evolución del sistema. Dado que las perturbaciones son pequeñas, la solución tras un tiempo suficientemente grande será independiente de estas.

## 5.2 Validación del método semi-implícito

De manera general, resolución completa requiere mayor tiempo de computación que el análisis lineal, puesto que se basa en resolver el problema durante un tiempo suficientemente grande. Al cabo de este, se observa si el sistema ha llegado a un régimen estacionario en el que poder observar el valor de las magnitudes, o en su defecto, a una solución periódica. Sin embargo, permite obtener la evolución temporal sin despreciar términos no lineales.

Como ya se ha visto, existen multitud de parámetros que influirán en la solución del problema: número y distribución de nodos en cada dirección, métodos para hallar las derivadas, paso de tiempo o perturbación inicial. A ellas habría que añadir diferentes combinaciones de las condiciones de contorno, en temperatura y velocidades, así como la geometría de estos. En primer lugar, para la validación, se hará uso de diferente configuraciones de métodos obtenidas en el capítulo anterior. Tras esta primera fase, se elegirá una sola configuración para resolver los problemas de convección forzada.

El paso de tiempo debe tener en cuenta muchas restricciones. En primer lugar, debe cumplir la condición de Courant-Friedrichs-Lewy (CFL), que asegura que su valor sea inferior al tiempo de residencia de una partícula en una celda o entre dos nodos. Esto ofrece una cota superior, y depende de la velocidad en cada celda y de su tamaño en la dirección de esta. Por tanto, depende directamente del número de Rayleigh impuesto, ya que un mayor valor en este origina mayores valores de velocidad. Como ya se ha comentado, la velocidad no solo depende del número de Rayleigh, también es función de las condiciones de contorno e iniciales impuestas. De hecho, puede llegar a ser función de la malla si esta no es lo suficientemente buena.

Por otro lado, hay que comprobar que la aproximación de los términos no lineales por sus valores en el paso de tiempo anterior es válida. Al igual que se hizo con el mallado, para comprobar esto, se usará un paso de tiempo aún menor, de manera que si los resultados no varían se considera el paso de tiempo válido. Otro enfoque posible es que la condición CFL sirva para calcular el máximo paso de tiempo para el siguiente instante. Sin embargo, esto supondría un valor variable que dependería de las velocidades en cada instante. Por tanto, no permitiría aprovechar el valor constante de la matriz de coeficientes del problema. Por ello, se prefiere hacer uso de un paso de tiempo constante, y, dado que depende de tantos factores, se buscará un valor suficientemente pequeño para que se

cumpla la condición CFL mediante prueba y error. Si esta deja de cumplirse en algún momento de la simulación, esta diverge y habría que comenzar con un paso temporal menor.

Para la validación se observará la evolución temporal de algunas magnitudes físicas, y se obtendrá el valor de Rayleigh crítico a partir del flujo de calor en el contorno inferior.

### 5.2.1 Rayleigh crítico

Para encontrar el número de Rayleigh crítico en este caso es necesario encontrar alguna variable que indique si existe convección o no. Lo más intuitivo es pensar en el campo de velocidades, de manera que si estas son distintas de cero existe movimiento y por tanto convección. Esto presenta un problema: dado que se resuelve numéricamente las velocidades nunca son cero y habría que establecer un umbral a partir del cual se consideran suficientemente altas para poder afirmar que existe convección. Sin embargo, es mucho más fácil obtener el valor crítico observando como varía el flujo de calor con el número de Rayleigh impuesto.

En el tiempo de resolución hasta llegar a un estacionario influye de manera importante la perturbación inicial. Dada la poca inercia del sistema, para valores impuestos de  $Ra$  en torno al valor crítico la pequeña perturbación inicial tardará en amortiguarse o amplificarse. En caso de no darle una perturbación inicial el tiempo para observar como evoluciona el sistema sería aún mayor. También influye la forma de la perturbación aunque esta no se ha modificado.

Una vez la solución llega a un estacionario se puede obtener el valor del flujo de calor existente en el contorno inferior. Si se resuelven soluciones estacionarias a medida que se aumenta el número de Rayleigh, el flujo de calor aumentará de manera continua, aunque existirá un punto en el que haya un cambio de pendiente tal y como se comentó en el capítulo anterior (ver 4.2). De manera equivalente se puede observar el número de Nusselt en función del valor de Rayleigh impuesto. Se distinguen dos tramos distintos: un valor unidad constante si  $Ra < Ra_c$  y una curva con pendiente mayor que cero  $Ra > Ra_c$  (ver 5.1). El caso mostrado se ha resuelto por diferencias finitas con cinco nodos circundantes, 61 y 30 nodos en las direcciones  $x$  y  $z$ , malla uniforme y relación de aspecto unidad.

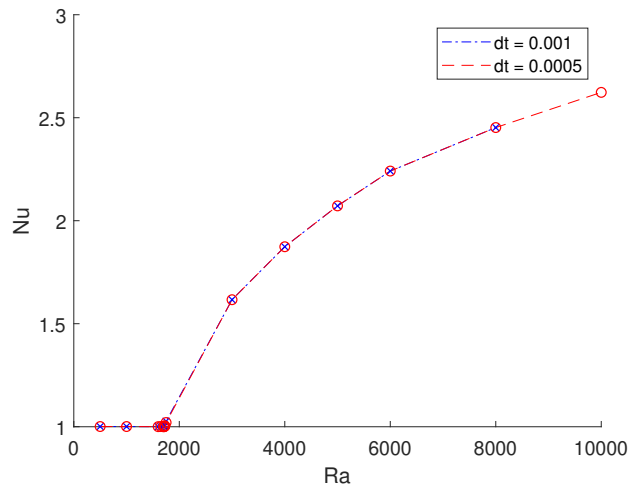
El número de Nusselt en un punto del contorno inferior, si este es recto, se define según 5.13. Para obtener el valor en todo el contorno como se ha hecho aquí, basta con integrar en él.

$$Nu = \frac{H}{(T_2 - T_1)} \frac{\partial T}{\partial z} \Big|_{x,z=0} = \frac{1}{\hat{T}_1 - \hat{T}_2} \frac{\partial \hat{T}}{\partial \hat{z}} \Big|_{\hat{x},\hat{z}=0} \quad (5.13)$$

Teniendo en cuenta esto se ha elaborado un código con la misma idea que se usaba en el estudio de estabilidad: resolver e ir aumentando el número de Rayleigh hasta obtener un valor superior al crítico y aplicar la bisección hasta obtener un intervalo suficientemente pequeño. La diferencia es que antes se comprobaba que no existieran autovalores con parte real positiva y ahora se comprueba que para todos los Rayleigh calculados por debajo del límite superior los resultados de flujo de calor definen una recta. Se ha aprovechado el conocimiento de los valores críticos de la anterior sección para definir intervalos próximos a este valor, disminuyendo así el número de iteraciones necesarias.

Se va a realizar un estudio en función del paso de tiempo impuesto. En cada instante se comprobará que este cumple la condición CFL, ya que en caso de no hacerlo, se ha visto que el método acaba divergiendo. Se ha resuelto un fluido entre dos placas para relaciones de aspecto 1 y 40. Para ello se han usado las configuraciones de métodos y mallas del capítulo anterior.

En la tabla 5.1 se muestran los resultados de Rayleigh crítico obtenidos para  $C1$  y relaciones de aspecto 1 y 40. Los resultados, para  $AR = 1$  son iguales salvo por la tolerancia permitida en el método de la bisección. Sin embargo, la diferencia es mayor si la relación de aspecto es 40.



**Figura 5.1** Nusselt en función del número de Rayleigh para el contorno inferior para fluido entre dos placas con  $AR = 40$  resuelto por diferencias finitas.

**Tabla 5.1** Comparación de Rayleigh crítico mediante resolución en el tiempo y análisis de estabilidad para C1.

		AR		
		1	40	
A1	<b>Análisis lineal</b>	4640.16	1709.5	
	<b>Resolución en el tiempo</b>	$dt = 5e - 3$	4639.8	1710.1
		$dt = 1e - 3$	4639.8	1710.1
		$dt = 5e - 4$	4639.8	1721.1
A3	<b>Análisis lineal</b>	4637.23	1708.52	
	<b>Resolución en el tiempo</b>	$dt = 5e - 3$	No cumple CFL	No cumple CFL
		$dt = 1e - 3$	4636.4	1716.8
		$dt = 5e - 4$	4636.4	1721.1
A4	<b>Análisis lineal</b>	4640.16	1709.5	
	<b>Resolución en el tiempo</b>	$dt = 5e - 3$	No cumple CFL	No cumple CFL
		$dt = 1e - 3$	4639.8	1727.3
		$dt = 5e - 4$	4639.8	1722.5

Concretamente, el método que más se acerca es el que hace uso de nodos de Chebyshev. Cabe destacar, que como se ha partido del conocimiento previo del valor crítico para que la resolución no durase un tiempo excesivamente alto, el número de Rayleigh impuesto siempre ha estado en torno al valor crítico. Por lo tanto, las velocidades nunca han estado muy lejos de ser nulas.

## 5.2.2 Evolución temporal (B.14 y B.15)

Los resultados que se muestran a continuación corresponden al caso C1 para las configuraciones A1, A3 y A4, con valores de Rayleigh entre 3000 y 8000, tres valores de paso temporal y relaciones de aspecto 1 y 5. El objetivo es observar como afecta el paso temporal escogido a la solución del sistema. Para definir en que momento se llega a un estacionario se ha tenido en cuenta la desviación típica y la media de algunas magnitudes a lo largo de un tiempo suficientemente grande.

Las configuraciones de método de obtención de derivadas son las más óptimas en cuanto a tiempo y precisión, de todas las variantes que se han analizado en el capítulo anterior. Se pretenden analizar

si existen diferencias entre ellas al resolver en el tiempo, y como afectan los dos parámetros ( $Ra$  y  $dt$ ) a ellas. Los valores de Rayleigh escogidos se deben a que estos deben provocar velocidades suficientemente altas para observar el efecto del paso temporal aplicado. Las dos relaciones de aspecto ayudarán a entender la validez del número de nodos según sea esta. Por un lado un mismo número de Rayleigh estará más lejos o más cerca del valor crítico en función de la relación de aspecto. Por otro lado, la distancia entre nodos, fijado el número de estos, varía con la relación de aspecto.

En primer lugar, se muestra en la figura 5.2a la velocidad vertical máxima para  $AR = 1$  en función del número de Rayleigh. Por un lado, se observa que existen valores de Rayleigh para los cuales los métodos de diferencias finitas no cumplen la condición CFL. Esto se debe a que existen celdas más pequeñas al tener más nodos. De hecho, este efecto es más evidente para  $AR = 5$ . Como puede verse en la figura 5.2b el número de Rayleigh para el que esto sucede es menor. Esto se debe a lo comentado anteriormente: las celdas son más pequeñas y la diferencia del número de Rayleigh respecto al crítico aumenta. De hecho, esto sucede también ahora para el método con nodos de Chebyshev, aunque para un número de Rayleigh mayor.

En segundo lugar, se aprecia una diferencia entre los resultados hallados con el método de nodos de Chebyshev y los métodos de diferencias finitas. Aunque aparece en ambas figuras, es más evidente cuanto mayor es la relación de aspecto y mayor es el número de Rayleigh. Como es lógico, si este último no es mayor que el crítico, las velocidades son nulas, y por tanto, cuanto más próximo esté a este valor menor será la diferencia entre los métodos.

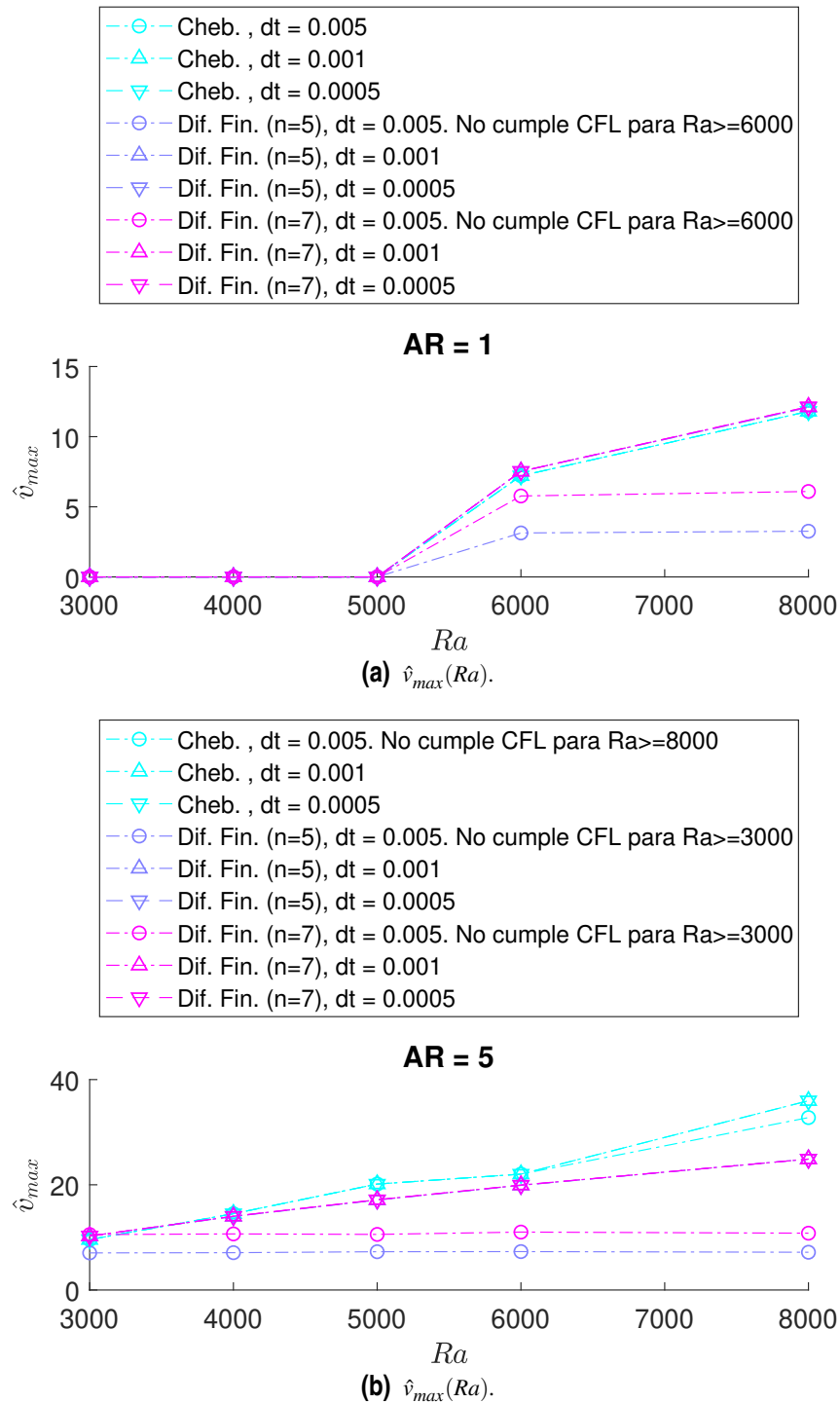
Ambos fenómenos son representativos de lo que sucede también para la velocidad horizontal, así como para las medias de ambas (calculadas como la media del valor absoluto de estas).

En cuanto al paso temporal, se puede concluir que para estos casos  $dt = 0.001$  es una buena elección, ya que los resultados no varían si este se reduce a la mitad.

Estas diferencias pueden observarse también a lo largo del tiempo. Para ello se muestran las figuras 5.3a y 5.3b, en las que se puede ver las diferencias entre los diversos métodos y pasos temporales. Se observa que con nodos de Chebyshev no converge al mismo valor que usando diferencias finitas. Se ha comprobado que aumentando el número de nodos a 41 para ambas relaciones de aspecto, este fenómeno se minimiza y las soluciones coinciden. Sin embargo, para relaciones de aspecto mayores vuelve a suceder, por lo que, se concluye que esto se debe a la distancia entre nodos.

En el caso de resolver con nodos de Chebyshev el número de nodos es menor, y además, al no estar equiespaciados, la mayor distancia entre nodos aumenta. Se deduce que esta impide que el método semi-implícito funcione correctamente, ya que en el análisis lineal esto no sucedía. Por tanto, suponer que el valor de los términos no lineales es igual a su valor en el paso de tiempo anterior no funciona correctamente si la distancia entre nodos es demasiado alta, aún cuando se cumpla la condición CFL. Es necesario señalar, que pese a esto, la estimación de valores críticos de Rayleigh observando el flujo de calor en el contorno inferior era más precisa usando el método con nodos de Chebyshev. La explicación se encontrará en que en ese el número de Rayleigh impuesto siempre estuvo muy cerca del crítico, y por tanto las velocidades eran despreciables. Por tanto, ambos resultados son perfectamente compatibles.

Esto supone una gran desventaja respecto al método de diferencias finitas. El aumento del número de nodos para solventar este problema es doblemente perjudicial para este método. Por un lado, mientras se use la distribución de nodos de Chebyshev, el aumento se concentra en los contornos, y por tanto el número de Courant aumenta de manera más pronunciada en estos, lo que implicaría tener que disminuir el paso de tiempo. Esto supone claramente un aumento de coste computacional. Por otro lado, hay que destacar que el método de Chebyshev genera matrices llenas, por lo que un aumento del número de nodos tiene mayor coste relativo en este caso que en el caso de usar las diferencias finitas. Así, se ve comprometida la mayor ventaja de este método: mayor precisión para

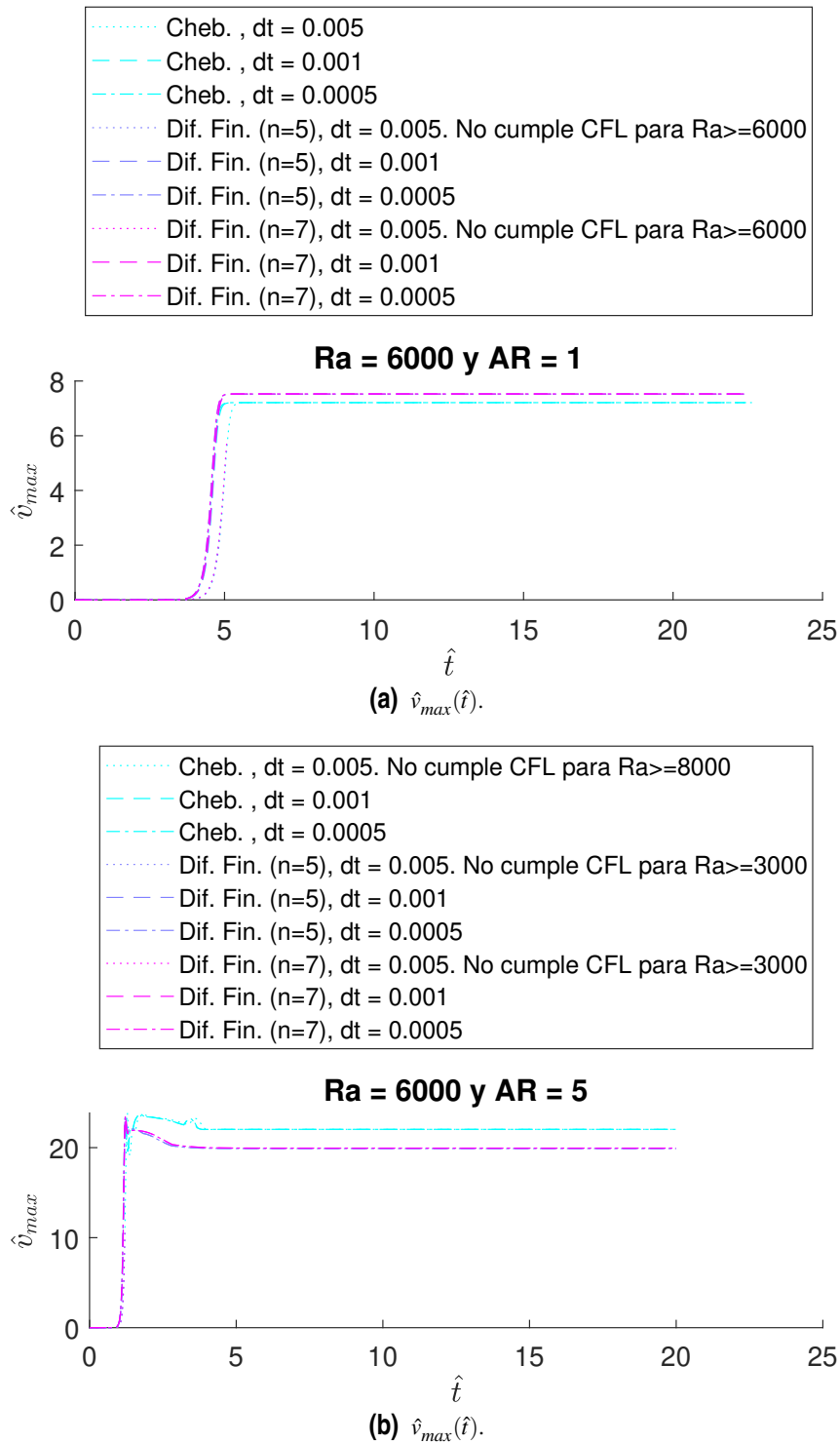


**Figura 5.2** Velocidad vertical máxima en función del número de Rayleigh, para  $AR = 1$  y  $AR = 5$ , y diferentes pasos temporales y métodos.

un mismo número de nodos.

Para concluir, queda demostrado que un paso temporal de 0.001 es suficiente para el rango de valores de las diferentes variables que se van a tener en cuenta. Esto podría dejar de ser así en el caso de convección forzada, ya que se impone una velocidad. A priori se hará uso de este valor, y en caso de que los resultados no sean razonables, o la condición CFL no se cumpla, se disminuirá este. El método usado de ahora en adelante es el de diferencias finitas con siete nodos circundantes





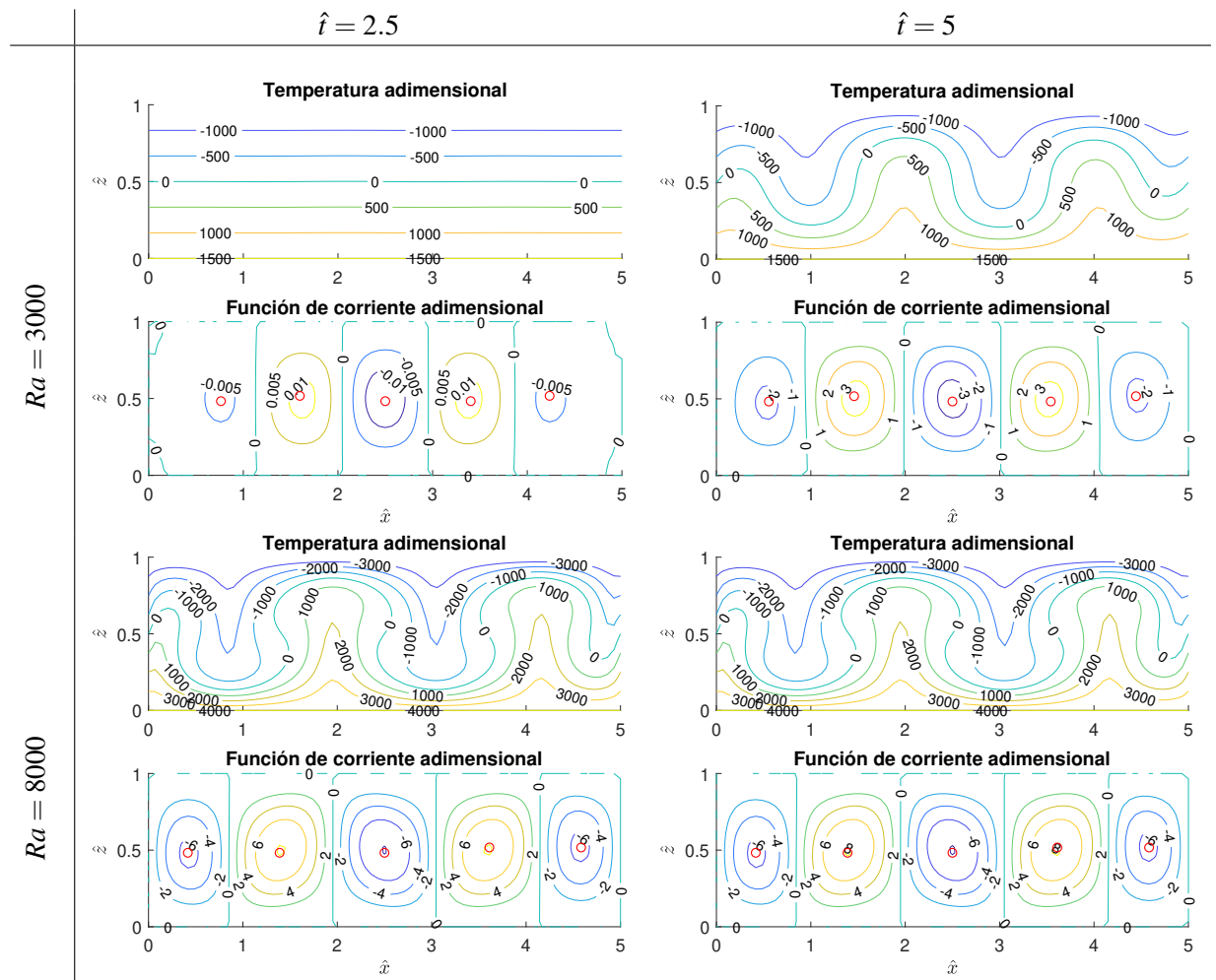
**Figura 5.3** Velocidad vertical máxima para  $Ra = 6000$ ,  $AR = 1$  y  $AR = 5$ , y diferentes pasos temporales y métodos.

y la distribución de nodos de la tabla 4.1.

Por último, se muestran las celdas de Bénard y los campos de temperatura (5.2) para una relación de aspecto igual a cinco y números de Rayleigh 3000 y 8000, en dos instantes de tiempo. se observa como el dominio se divide en celdas convectivas, simétricas respecto a los planos verticales si están se encuentran suficientemente lejos de los contornos. Además, se ha localizado el centro de estas

celdas buscando los máximos locales del valor absoluto de la función de corriente.

**Tabla 5.2** Campos de temperatura y función de corriente adimensionales para  $AR = 5$  (contornos rectos).



### 5.3 Convección forzada

Hasta ahora se han validado los métodos propuestos para resolver fenómenos de convección natural, así como el valor de Rayleigh que se ha de imponer para que estos se produzcan. Además, se ha estudiado como afecta los diferentes parámetros a la solución (número y distribución de nodos, manera en que se calculan los operadores matriciales de derivadas, paso de tiempo, etc).

En esta sección se van a resolver y analizar algunos problemas adicionales a los planteados hasta este momento. Para ello se hará uso de diferencias finitas con siete nodos circundantes y el número de nodos fijados en el capítulo anterior. El paso de tiempo adimensional se escogerá igual a 0.001.

En primer lugar se va a realizar un estudio en función del caudal a la entrada con el fin de entender el proceso de convección forzada. En definitiva esto supone la imposición de nuevas condiciones de contorno. Estas ya se han deducido al comienzo de este capítulo.

En segundo y último lugar se realizará otro estudio también en función de una velocidad de entrada, pero en el que se tiene en cuenta diferentes formas en el contorno inferior.

En ambos casos se realizará en primera instancia un análisis lineal para ver como influye el caudal y la deformación del contorno inferior en el número de Rayleigh crítico. A partir de los resultados de este se resolverán números de Rayleigh representativos para ambos problemas mediante el análisis completo.

### 5.3.1 Convección forzada con contornos rectos y condición de periodicidad en los laterales (B.16 y B.17)

Se va a considerar un caudal de fluido a lo largo de un conducto suficientemente largo. Se supondrá que el problema es de dos dimensiones, por tanto el conducto vendrá definido por un contorno superior y uno inferior. El superior estará a una temperatura  $T_2$  menor que  $T_1$ , a la que se encuentra el inferior.

Suponiendo que el conducto tiene una longitud suficientemente larga, el flujo estará desarrollado en un tramo intermedio, y no dependerá de las condiciones a la entrada ni a la salida en este tramo. En este caso, las condiciones de contorno en los laterales podrán suponerse periódicas.

En los contornos laterales se van a imponer condiciones de periodicidad tal y como se hizo en el capítulo anterior, tanto para la temperatura como para la función de corriente. En los contornos superior e inferior se impondrá las condiciones en temperatura  $T_2$  y  $T_1$  respectivamente, y condiciones de pared para las velocidades.

Como ya se dedujo al principio del capítulo, la diferencia de la función de corriente entre el contorno superior e inferior es el caudal. Además, por comodidad se establecerá la referencia de esta en el contorno inferior. Un esquema con todas las condiciones de contorno se muestra en 5.7.

$$q = \psi(z = H) - \psi(z = 0) = \psi(z = H) = \psi_{sup} \quad (5.14)$$

Se ha resuelto usando diferencias finitas con siete nodos circundantes y 50 y 20 nodos para las direcciones  $x$  y  $z$  respectivamente.

#### Análisis lineal para convección forzada con contornos rectos

El procedimiento es el mismo que se siguió en el capítulo 4. En primer lugar se separan las magnitudes físicas a resolver en dos partes: una que corresponde al estado de equilibrio y otra de perturbación (ver expresiones 5.16).

$$T = T_{eq} + \theta \quad (5.15)$$

$$\Psi = \Psi_{eq} + \phi \quad (5.16)$$

$$T_{eq} = T_1 + (T_2 - T_1) \frac{z}{H} \quad (5.17)$$

$$\Psi_{eq} = \psi_{sup} \frac{z}{H} \quad (5.18)$$

Se aplica esta descomposición en las ecuaciones generales de modelo de Boussinesq, se sustituyen las expresiones debidas al equilibrio (5.18) en ellas y se desprecian los términos no lineales. En este caso, aparecen nuevos sumandos al descomponerse términos que eran no lineales anteriormente en uno con parte lineal y otra no lineal. Por último se adimensionaliza de la misma forma que se hizo anteriormente y se supone una solución exponencial en el tiempo para ambas variables incógnita,

resultando el sistema 5.19.

$$\begin{bmatrix} Pr\hat{\nabla}^4 - \hat{\psi}_{sup} \frac{\partial}{\partial \hat{x}} \hat{\nabla}^2 & -PrRa \frac{\partial}{\partial \hat{x}} \\ -\frac{\partial}{\partial \hat{x}} & \hat{\nabla}^2 - \hat{\psi}_{sup} \frac{\partial}{\partial \hat{x}} \end{bmatrix} \begin{bmatrix} \hat{\phi}_0 \\ \hat{\theta}_0 \end{bmatrix} = s \begin{bmatrix} \hat{\nabla}^2 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{\phi}_0 \\ \hat{\theta}_0 \end{bmatrix} \Rightarrow Ax = sBx \quad (5.19)$$

Las condiciones de contorno, al ser de perturbación, vuelven a imponerse todas nulas excepto en los laterales, donde se imponen periódicas (ver 5.4).

$$C1: \left. \frac{\partial \hat{\phi}}{\partial \hat{z}} \right|_{\hat{z}=1} = 0; \quad C2: \left. \frac{\partial^2 \hat{\phi}}{\partial \hat{z}^2} \right|_{\hat{z}=1} = 0; \quad \hat{\phi} \Big|_{\hat{z}=1} = 0; \quad \hat{\theta} \Big|_{\hat{z}=1} = 0$$

$$\begin{aligned} \hat{\phi} \Big|_{\hat{x}=0} - \hat{\phi} \Big|_{\hat{x}=AR} &= 0 \\ \frac{\partial \hat{\phi}}{\partial \hat{x}} \Big|_{\hat{x}=0} - \frac{\partial \hat{\phi}}{\partial \hat{x}} \Big|_{\hat{x}=AR} &= 0 \\ \hat{\theta} \Big|_{\hat{x}=0} - \hat{\theta} \Big|_{\hat{x}=AR} &= 0 \\ \frac{\partial \hat{\phi}}{\partial \hat{z}} \Big|_{\hat{z}=0} &= 0; \quad \hat{\phi} \Big|_{\hat{z}=0} = 0; \quad \hat{\theta} \Big|_{\hat{z}=0} = 0 \\ \frac{\partial^2 \hat{\phi}}{\partial \hat{x}^2} \Big|_{\hat{x}=0} - \frac{\partial^2 \hat{\phi}}{\partial \hat{x}^2} \Big|_{\hat{x}=AR} &= 0 \\ \frac{\partial^3 \hat{\phi}}{\partial \hat{x}^3} \Big|_{\hat{x}=0} - \frac{\partial^3 \hat{\phi}}{\partial \hat{x}^3} \Big|_{\hat{x}=AR} &= 0 \\ \frac{\partial \hat{\theta}}{\partial \hat{x}} \Big|_{\hat{x}=0} - \frac{\partial \hat{\theta}}{\partial \hat{x}} \Big|_{\hat{x}=AR} &= 0 \end{aligned}$$

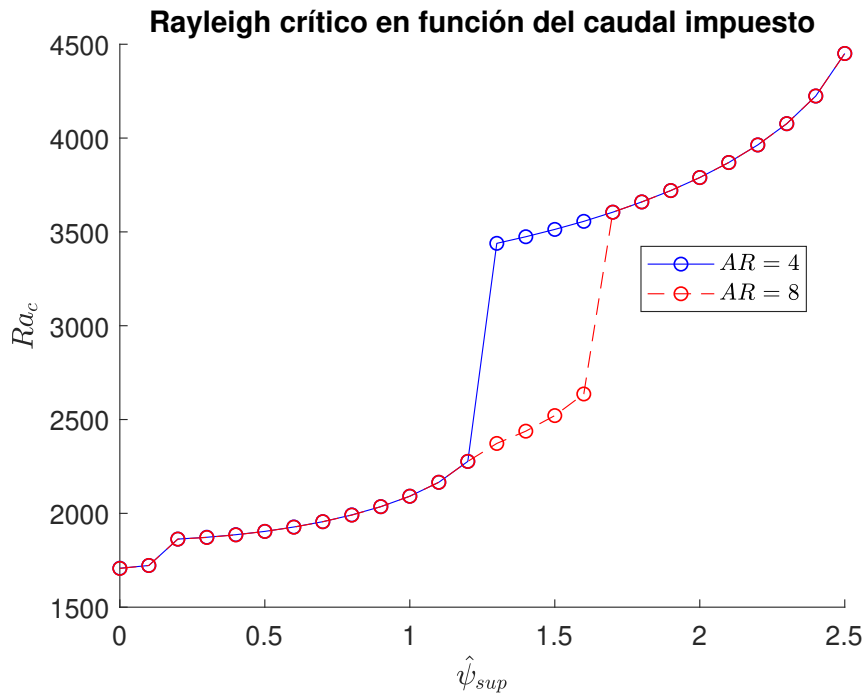
**Figura 5.4** Condiciones de contorno del análisis lineal para para convección forzada y contornos rectos.

En la figura 5.5 se muestra como afecta el caudal al Rayleigh crítico para dos relaciones de aspecto ( $AR = 4$  y  $AR = 8$ ). En primer lugar se observa, como es lógico, que el valor crítico para caudal nulo coincide con el hallado en el capítulo 4 para condiciones periódicas. No solo eso, sino que al tener una longitud que es múltiplo de la propia de las celdas de Rayleigh, este coincide con el valor para un dominio infinito en la dirección  $x$ .

Por otro lado, se observa un aumento del número del valor crítico al introducir un mayor caudal. Esto se debe a que la diferencia de temperaturas no es capaz de crear la inestabilidad si existe un fluido en movimiento introduciendo energía cinética en una dirección. Cualquier inestabilidad que pudiese crear la diferencia de temperatura puede verse amortiguada por el fluido que está introduciéndose.

Las curvas para dos relaciones de aspecto son esencialmente iguales salvo las zonas donde el Rayleigh crítico aumenta de manera pronunciada. El comienzo de esta zona se desplaza a valores superiores de  $\hat{\psi}_{sup}$  al aumentar la relación de aspecto. De hecho, para  $\hat{\psi}_{sup} = 5$ , el valor de Rayleigh obtenido para  $AR = 4$  es de  $1.8754 \cdot 10^6$ , mientras que para  $AR = 8$   $1.4341 \cdot 10^4$ , dos órdenes de magnitud menor. Es decir, en el primer caso el salto ya se ha producido, y en el segundo aun no.

Para poder entender mejor a que se debe ese aumento pronunciado del valor crítico es útil calcularlo en función de la relación de aspecto, de la misma forma en que se hizo en el capítulo 4



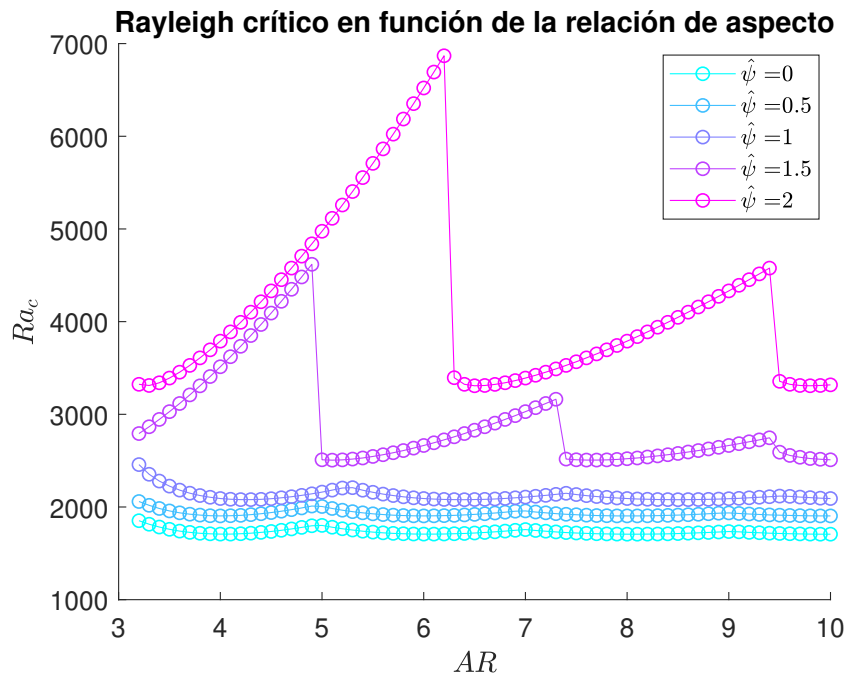
**Figura 5.5** Dependencia de Rayleigh crítico en función del caudal para  $AR = 4$  y  $AR = 8$ .

para condiciones periódicas. Los resultados se muestran en 5.6. En el caso de convección natural, cuando la relación de aspecto se encontraba a igual distancia de dos mínimos locales, existía un máximo, ya que para esa relación de aspecto, no se consigue modelar el conducto infinito. En este caso, el caudal introduce una asimetría, y por tanto este fenómeno no aparece de la misma forma, sino que aparecen discontinuidades muy próximas a los mínimos.

A partir de estos mínimos se pueden obtener las relaciones de aspecto que permiten que el problema sea lo más parecido a uno periódico. Para ello solo hay que buscar los mínimos locales. Cuando se analizaba para el caso de convección natural estas longitudes eran además la longitud entre rollos. En este caso, se han hallado estas longitudes y son:

- $\hat{\psi} = 0 \Rightarrow \hat{L} = 2$
- $\hat{\psi} = 0.5 \Rightarrow \hat{L} = 2.05$
- $\hat{\psi} = 1 \Rightarrow \hat{L} = 2.1$
- $\hat{\psi} = 1.5 \Rightarrow \hat{L} = 2.25$
- $\hat{\psi} = 2 \Rightarrow \hat{L} = 3.2$

A lo largo de la sección se obtendrán resultados para  $AR = 4$  aunque como se ha comprobado, no modele bien el problema del conducto infinito.



**Figura 5.6** Rayleigh crítico en función de la relación de aspecto para diferentes caudales adimensionales (contornos rectos).

### Análisis completo para convección forzada con contornos rectos

Tras haber realizado el análisis de estabilidad se procede a la resolución completa del sistema. En este caso, como se dedujo al comienzo del capítulo, las condiciones de contorno son las que se muestran en la figura 5.7. Se ha resuelto el problema durante un tiempo lo suficientemente grande para observar si la solución llega a un estacionario o no, y en caso de que no lo haga si esta es periódica. El paso temporal usado para la resolución es  $1 \cdot 10^{-3}$ .

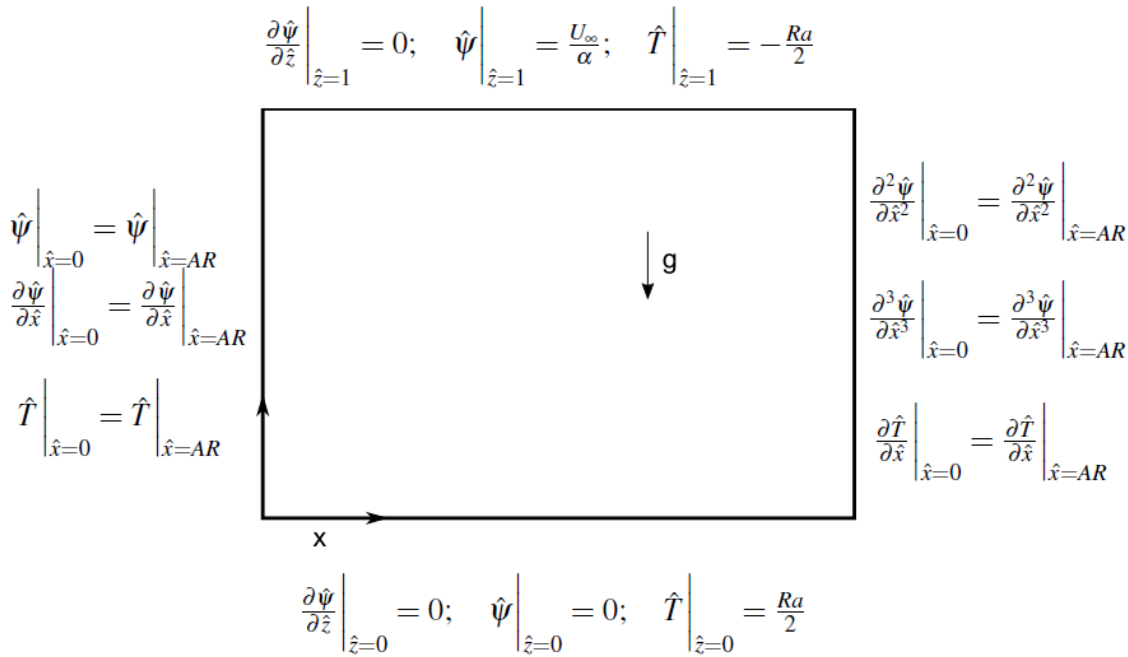
Se define el número de Nusselt en un punto cualquiera del contorno inferior como:

$$Nu(\hat{x}) = \frac{H}{(T_1 - T_2)} (\nabla T) \cdot \vec{n} \Big|_{\hat{x}} \quad (5.20)$$

Además de este, también se mostrará el número de Nusselt total del contorno inferior, es decir, la integral del anterior a lo largo del contorno.

$$Nu = \frac{Q_{\text{total}}}{Q_{\text{conducción}}} = \frac{(\nabla T) \cdot \vec{n} \Big|_{\text{total}}}{(\nabla T) \cdot \vec{n} \Big|_{\text{conducción}}} = \frac{H}{AR (T_1 - T_2)} (\nabla T) \cdot \vec{n} \Big|_{\text{total}} \quad (5.21)$$

La diferencia respecto a los casos de convección natural reside en que ahora existe un mecanismo de transferencia de energía en la dirección horizontal. En los problemas de convección natural, el movimiento que pueda aparecer en  $x$ , solo es relevante si el sistema se ha vuelto ya inestable, lo que implica el desarrollo de las celdas de Bénard. En el caso de estudio de esta sección, sin embargo, ya existe un movimiento horizontal, lo que, junto con las fuerzas de flotabilidad generadas por la diferencia de temperaturas, provoca la aparición de rollos de convección aún cuando el número de



**Figura 5.7** Condiciones de contorno para convección forzada y contornos rectos.

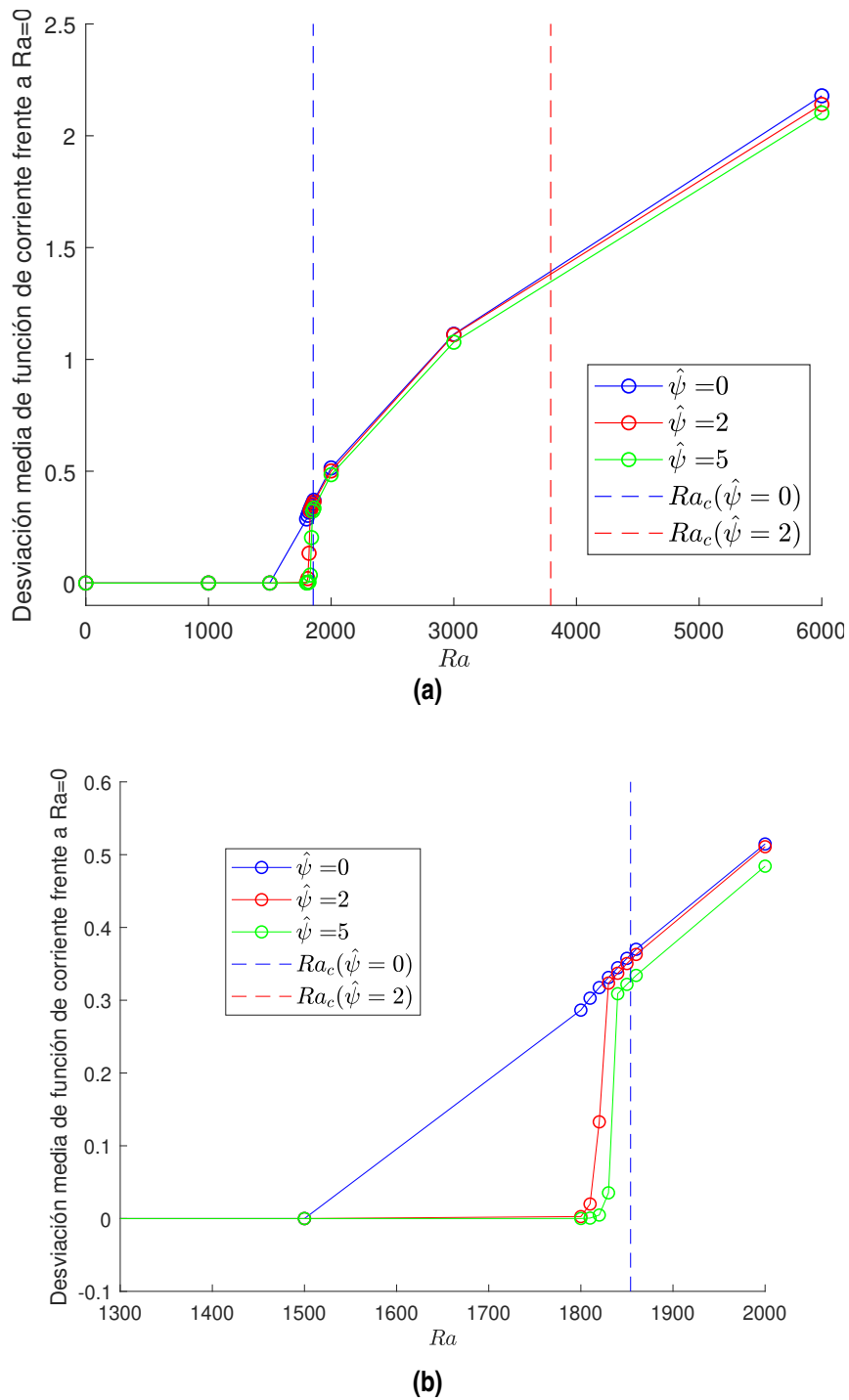
Rayleigh es inferior al crítico.

En primer lugar, se analiza lo que sucede a las líneas de corriente cuando el Rayleigh impuesto está próximo al crítico. A la vista de los resultados, estas pueden deformarse, es decir, dejar de ser horizontales, para  $Ra < Ra_c$ . Este hecho no implica sin embargo que el sistema sea inestable frente a una perturbación externa.

Para estudiar este fenómeno se ha resuelto el problema para relación de aspecto 4, tres caudales adimensionales (nulo, 2 y 5) y para varios números de Rayleigh. En primer lugar, se representa en la figuras 5.8 la desviación media de la función de corriente respecto a su valor para Rayleigh nulo. Es decir, el resultado mostrado se calcula con la expresión 5.22.

$$\text{Desviación media } \hat{\psi} = \frac{1}{N_x N_z} \sum_{i=1}^{N_x} \sum_{j=1}^{N_z} |\hat{\psi}_{(i,j)} - \hat{\psi}_{(i,j)}^{Ra=0}| \quad (5.22)$$

Se observa en primera instancia, que, por lo general, la desviación media es ligeramente inferior para mayores velocidades (o caudal, al ser la sección constante) fijado el número de Rayleigh. Esto es lógico ya que el caudal impedirá que la convección se desarrolle como lo haría si no hubiese un flujo impuesto. Por otro lado, se puede ver como en todos los casos (incluidos el de convección natural,  $\hat{\psi} = 0$ ), el campo de líneas de corriente se deforma, respecto al caso en que  $Ra = 0$  con el mismo caudal impuesto, antes de llegar a Rayleigh crítico. Esta zona en la que las líneas de corriente se deforman siendo el sistema estable, en el caso de la convección natural, es muy pequeña. Esta es la razón por la cual aún no se había detallado este fenómeno. Sin embargo, a medida que la velocidad impuesto es mayor, el tamaño de esta zona aumenta. Así, para  $\hat{\psi} = 2$ , se observa como esta viene dada por un rango de números de Rayleigh en torno a 2000. De hecho, para  $\hat{\psi} = 5$ , como ya se ha dicho antes, el valor crítico es mucho mayor, por lo que esta zona aumentaría aun más. Pese a aumentar su tamaño, también es cierto que una mayor velocidad retrasa el comienzo de la



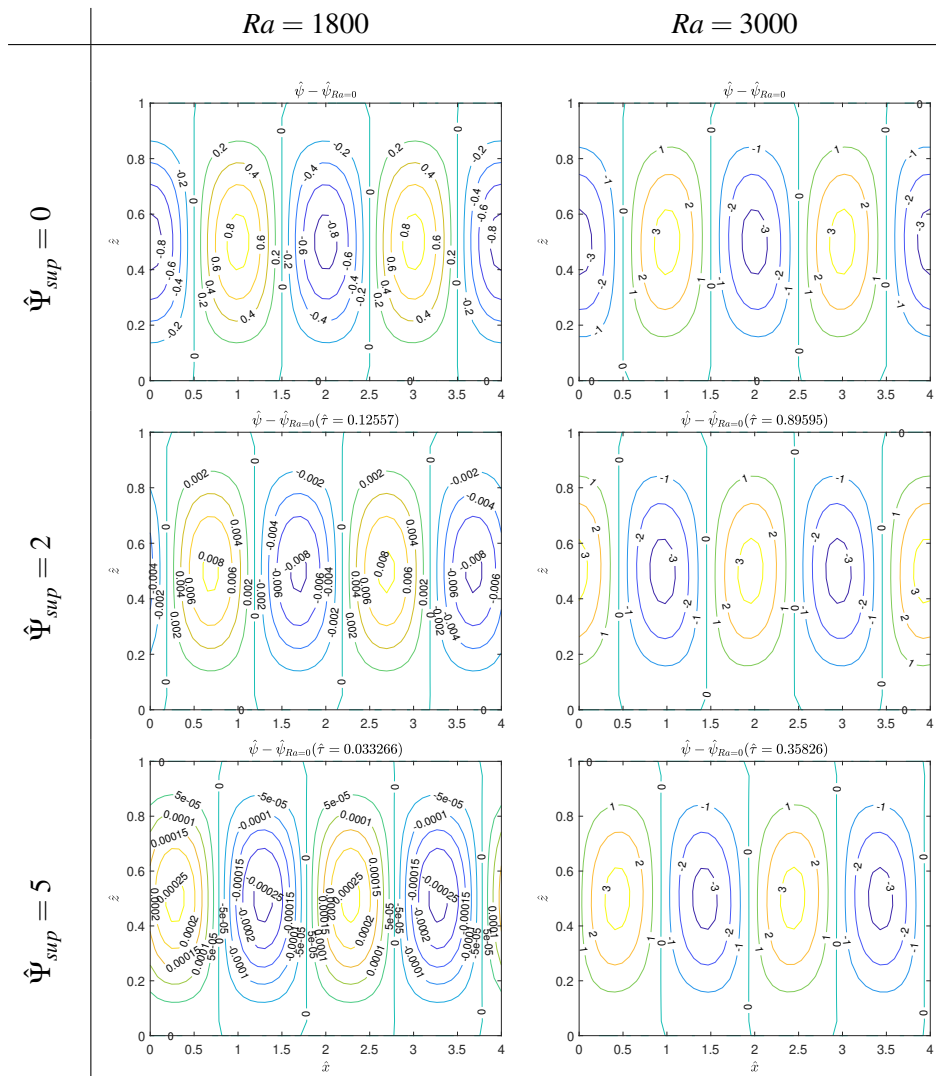
**Figura 5.8** Desviación media de la función de corriente respecto a la de Rayleigh cero.

deformación ligeramente (ver figura ampliada), siendo este efecto despreciable respecto al primero.

Otro fenómeno encontrado es que la solución no llega a un régimen estacionario, sino que esta es periódica. Para calcular el periodo se han obtenido las matrices de función de corriente y temperatura adimensional tras un tiempo suficientemente grande. A partir de ese instante, se ha obtenido la media de la diferencia respecto a estas matrices en todos los nodos. Por último, se han buscado los mínimos locales en el tiempo simulado, siendo la diferencia entre esos mínimos el periodo. En los casos en que la velocidad impuesta es nula el periodo también debe serlo, no obstante por razones numéricas el valor obtenido puede ser distinto de cero.



**Tabla 5.3** Desviación de función de corriente adimensional respecto a  $Ra = 0$  en un instante dado, para  $Ra = 1800$  y  $3000$  y  $\hat{\psi} = 0, 2$  y  $5$ , siendo  $AR = 4$  (contornos rectos).

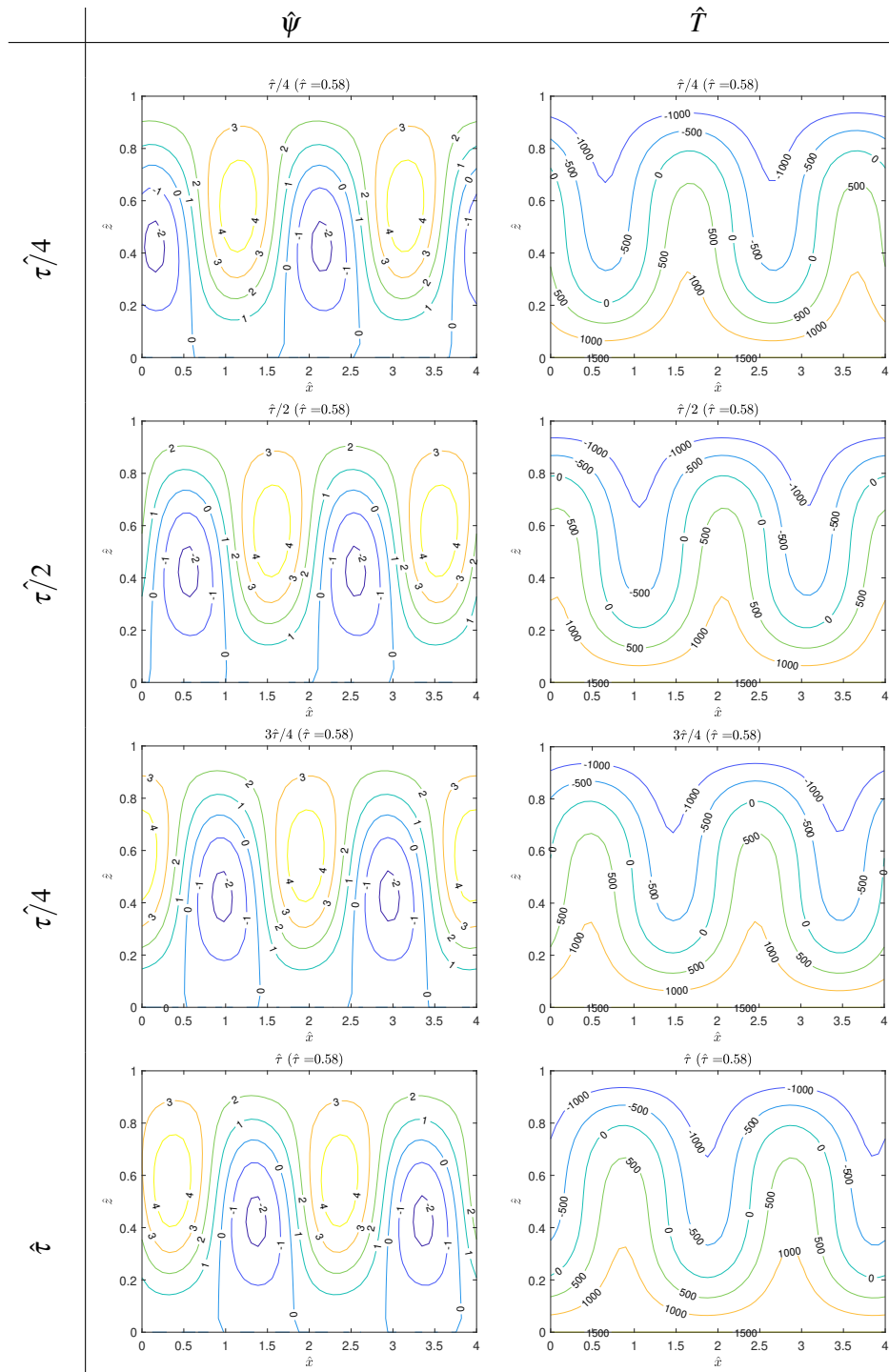


En las figuras 5.3 se muestran los campos de desviación de la función de corriente respecto al caso en que el Rayleigh es nulo para tres velocidades y dos números de Rayleigh. En ellas se muestra también el periodo calculado.

Puede observarse como para  $Ra = 1800$  la desviación de la función de corriente disminuye a medida que aumenta el caudal adimensional. Esto se debe a que este número de Rayleigh está en la zona en la que da comienzo este fenómeno, y como ya se ha comentado, la aparición de este se retrasa si se aumenta el caudal. En cambio, para  $Ra = 3000$ , cualquiera de los caudales resueltos tiene una desviación muy similar. También puede verse como en el caso de convección natural ( $\hat{\psi} = 0$ ), el campo es totalmente simétrico, siendo la solución estacionaria. Sin embargo, en los otros casos, los rollos están desplazados de manera arbitraria (se representa un instante cualquiera). Observando los periodos se puede ver que este disminuye al aumentar la velocidad impuesta. En efecto, si esta tiende a cero, los rollos no avanzan (convección natural). Es decir, los rollos avanzaran más rápido para una mayor velocidad, si está fijado el número de Rayleigh. En cambio, este último influye de manera contraria, el periodo aumenta al aumentar este si la velocidad esta fijada.

Una vez resaltada la diferencia entre el Rayleigh crítico y el proceso de deformación de las líneas de corriente, se muestran en la figura 5.4 los campos de temperatura y función de corriente a lo

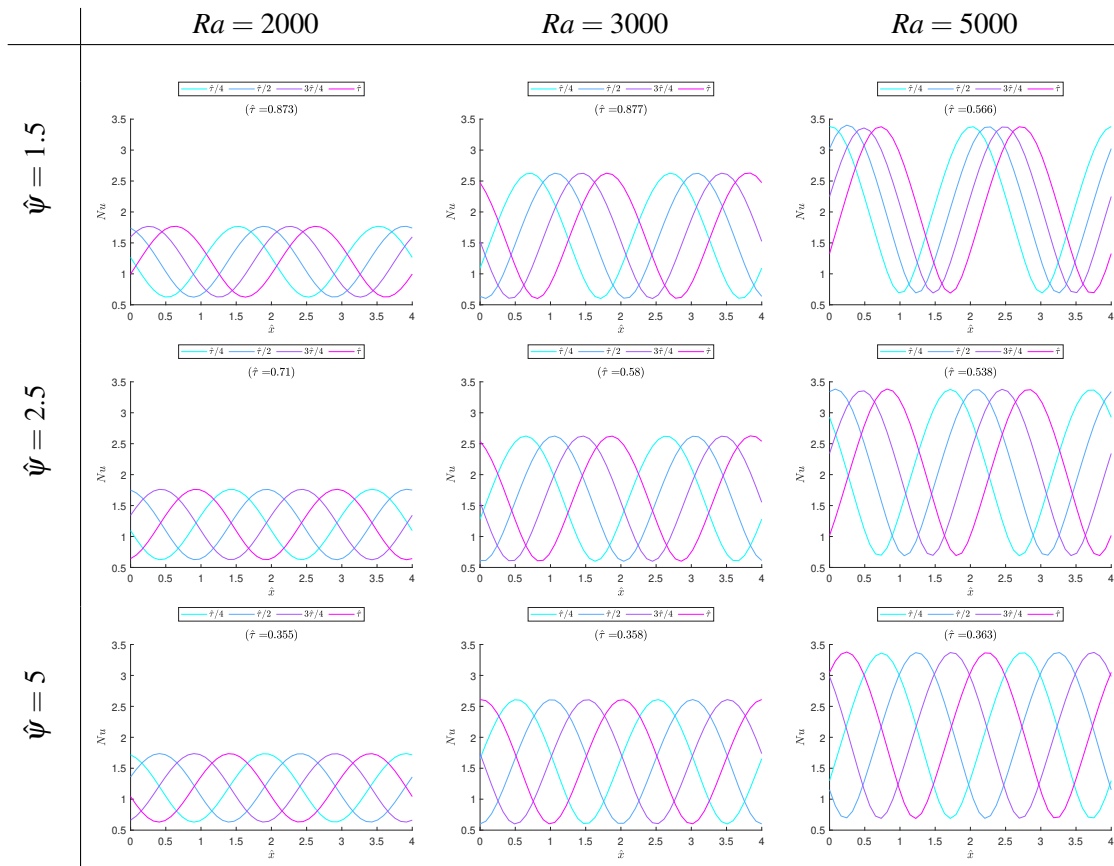
**Tabla 5.4** Campos de función de corriente y temperatura adimensionales a lo largo de un periodo, para  $Ra = 3000$ ,  $\hat{\psi} = 2.5$  y  $AR = 4$  (contornos rectos).



largo de un periodo. Los parámetros escogidos son  $Ra = 3000$  y  $\hat{\psi} = 2.5$ . Se observa perfectamente como ambos avanzan hacia la derecha. Resulta evidente que la distribución del número de Nusselt en el contorno inferior se verá afectado por ello, siguiendo por tanto también una ley periódica.

En el conjunto de figuras 5.5 se representa la distribución de número de Nusselt en el contorno inferior para varios números de Rayleigh y caudales adimensionales, siendo la relación de aspecto 4. Se mostrarán además los periodos de los diversos casos, los cuales confirman lo comentado

**Tabla 5.5** Distribución de Nusselt en el contorno inferior para diferentes caudales adimensionales, números de Rayleigh y  $AR = 4$  (contornos rectos).

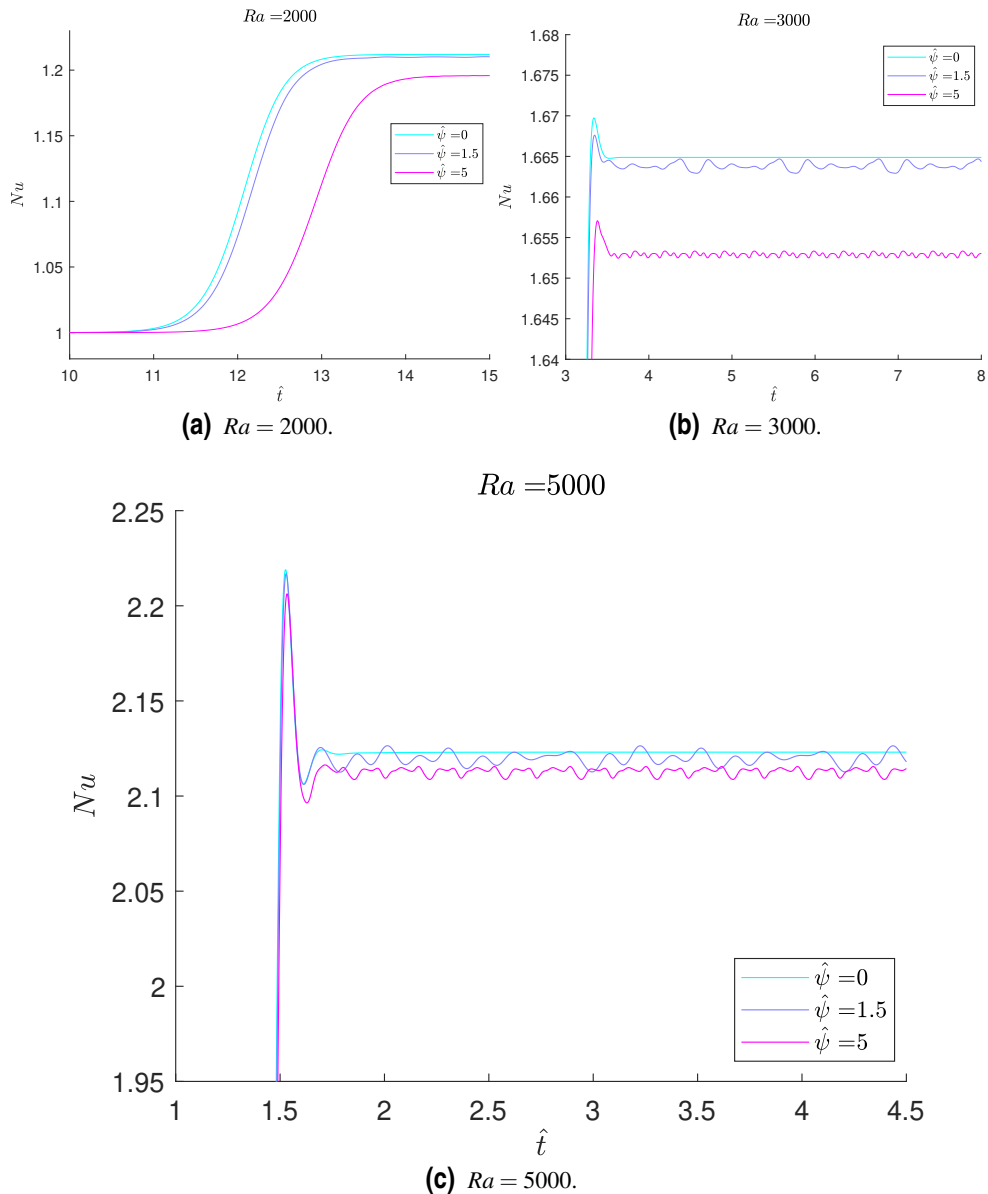


anteriormente: el periodo disminuye al aumentar el caudal o al disminuir el número de Rayleigh, manteniendo la variable la otra variable fija. Evidentemente también disminuye si se dan ambas razones de manera simultánea.

En el conjunto de figuras 5.9, puede verse el número total de Nusselt a lo largo del tiempo para diferentes números de Rayleigh y caudal adimensional. En primer lugar se observa que en todos los casos este llega a un estacionario, salvo por pequeñas fluctuaciones. Esto debe ser así debido a las condiciones periódicas impuestas. Teóricamente, si la relación de aspecto escogida pudiese contener una cantidad de rollos entera, las curvas no deberían tener ninguna fluctuación. Estas se deben a lo que sucede en los contorno, y aumentan si el caudal es mayor. Por eso, para convección natural, las fluctuaciones son inapreciables, ya que la solución es estacionaria y los rollos permanecen en la misma posición a lo largo del tiempo.

Por otro lado, se observa como el valor de Nusselt aumenta cuando lo hace el número de Rayleigh, como cabía esperar debido a la mayor deformación del campo de temperaturas. También se aprecia un ligero decremento si manteniendo el número de Rayleigh fijo se aumenta la velocidad impuesta. Esto esta directamente relacionado con la menor deformación de las líneas de corriente, y por tanto, un menor efecto en el campo de temperaturas.

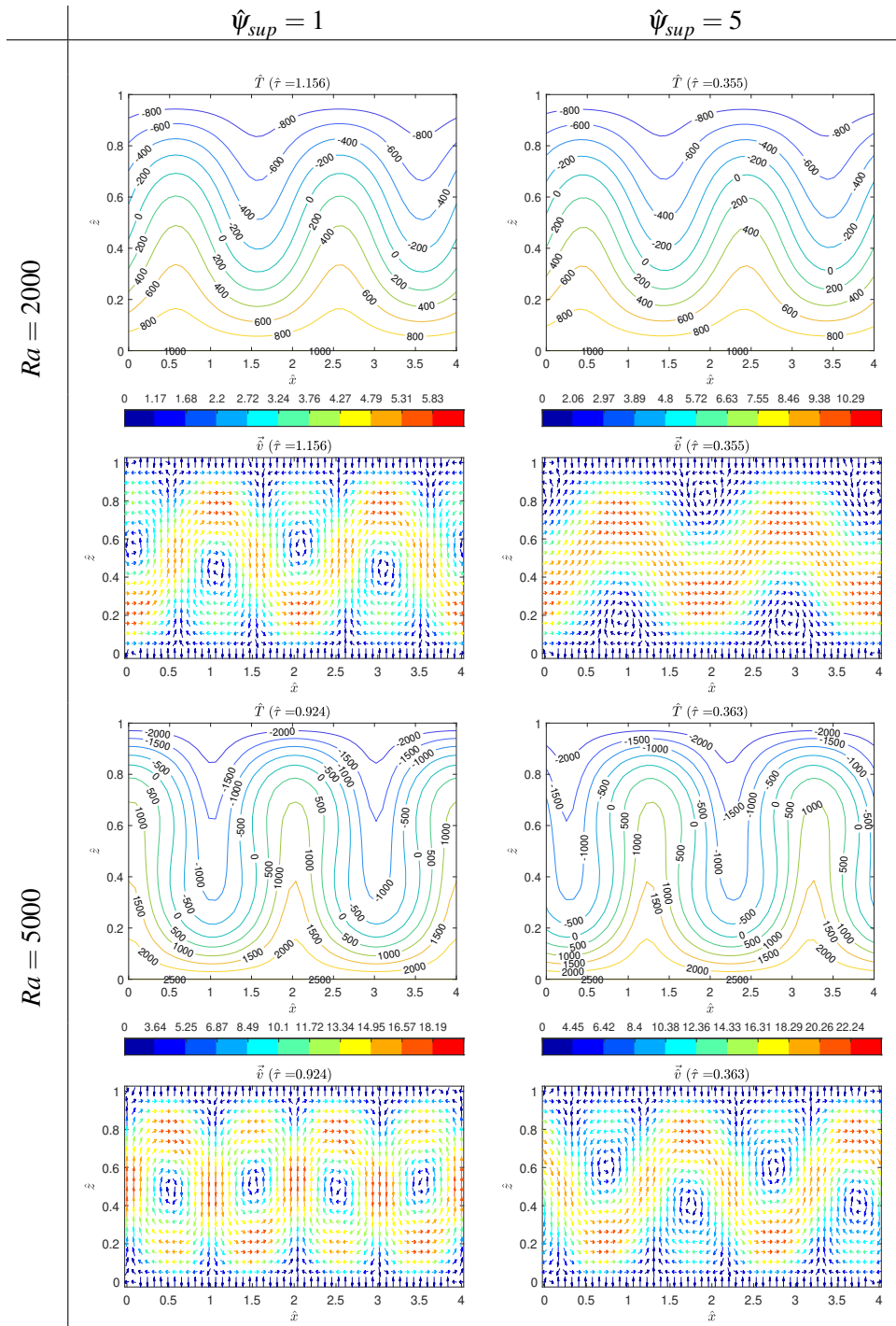
Por último se muestran los campos de temperatura y vectorial de velocidad adimensionales para dos números de Rayleigh y dos caudales adimensionales en el conjunto de figuras 5.6. Se puede ver como afectan conjuntamente los dos parámetros. Si el número de Rayleigh tiene mayor importancia que la que genera el caudal impuesto, en el campo de velocidades se ven rollos más definidos, y se reduce la zona por la que el fluido transita a lo largo de la longitud adimensional. Esto supone que el fluido debe acelerarse, además de las aceleraciones que sufre por el proceso de convección



**Figura 5.9** Número de Nusselt en el contorno inferior para diferentes números de Rayleigh y caudal adimensional, siendo  $AR = 4$  (contornos rectos).

en si, para que el caudal sea constante. Por esta razón, las velocidades máximas en los casos con  $Ra = 5000$  son muy parecidas aún cuando el caudal adimensional varía lo mismo que para los casos con  $Ra = 2000$ . En este último, la convección es mucho menor, y por tanto el fluido avanza con menos dificultad al estar los rollos menos definidos. En los campos de temperatura, por otro lado, no se aprecian grandes diferencias al cambiar el caudal impuesto. Esto pudo verse ya cuando se mostraron los números de Nusselt, tanto a lo largo de la relación de aspecto como el total, del contorno inferior.

**Tabla 5.6** Temperatura y campo vectorial de la velocidad (adimensionales) en un instante dado, para  $AR = 4$  y diferentes caudales adimensionales y números de Rayleigh (contornos rectos).



### 5.3.2 Convección forzada con contorno inferior no recto y condiciones de periodicidad

A continuación se resuelven casos de convección forzada pero añadiendo a los parámetros de estudio diferentes geometrías del contorno inferior, y manteniendo fija la relación de aspecto  $AR = 4$ . En este caso, debido al par baroclínico no existe solución de equilibrio. Se podría realizar un análisis lineal suponiendo que las amplitudes de la geometría del contorno inferior son pequeñas y que la solución de equilibrio es la que se obtendría al transformar el dominio rectangular. En este caso,

todos los sumandos de las ecuaciones de Boussinesq dependerían tanto de  $x$  como de  $z$  y por tanto las expresiones se complican bastante.

En este análisis se van a diferenciar dos casos:

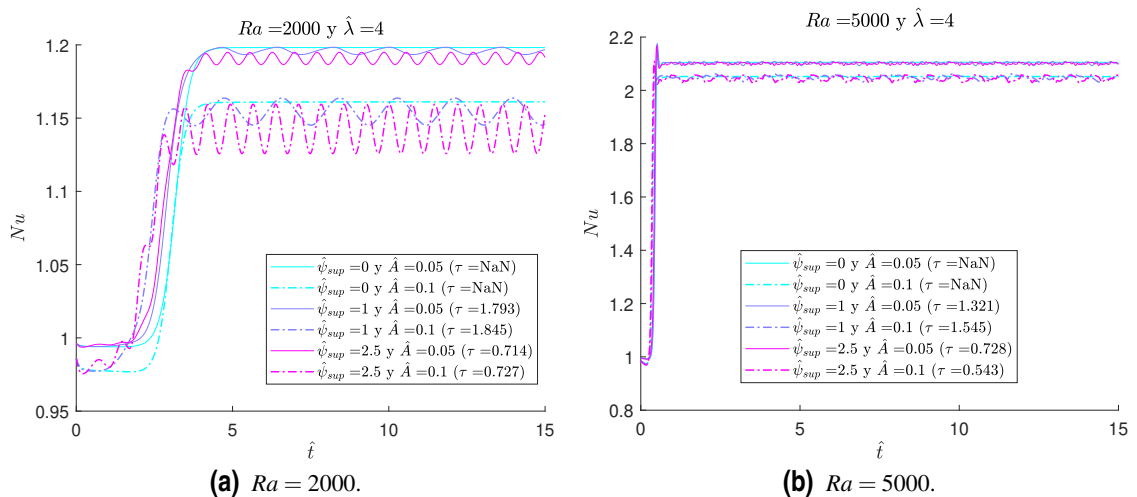
- Función sinusoidal con diferentes amplitudes ( $\hat{A}$ ) y longitud de onda ( $\hat{\lambda}$ ).
- Función gaussiana centrada en  $\mu = 0.35AR$  y con desviación típica  $\sigma = \frac{AR}{10}$ .

Las condiciones de contorno para este caso son las mismas que las impuestas para convección forzada cuando el contorno inferior era recto, la única diferencia está en que ahora  $\hat{z} = 0$  pasa a ser  $\hat{z} = \hat{z}_{inf}$ .

La resolución se ha hecho con el método de diferencias finitas con siete nodos circundantes y 50 y 20 nodos en las direcciones  $x$  y  $z$  respectivamente.

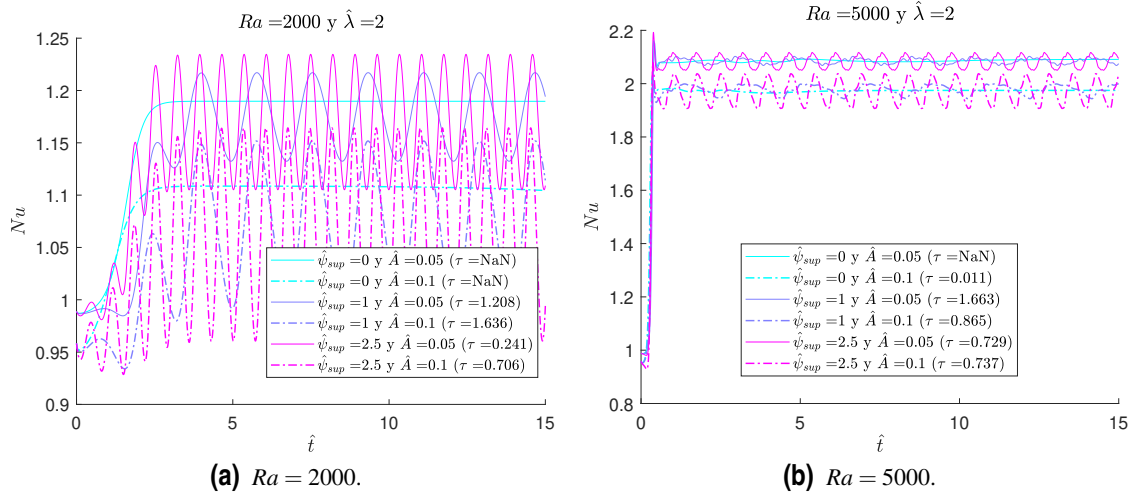
Se usarán al igual que antes las dos definiciones de Nusselt (total y puntual) a la hora de analizar los resultados. Cabe destacar que en este caso el flujo de calor normal no es el flujo vertical, y que para su cálculo se ha obtenido la normal de la curva que define el contorno inferior, con el fin de proyectar en ella el gradiente de temperaturas. Además, para obtener el calor de conducción de referencia se supone que el contorno es recto, y por tanto la definición no varía, el calor por conducción se supone igual al que habría en el estado de equilibrio si el contorno fuese recto.

En primer lugar se muestra el número de Nusselt total del contorno inferior para un contorno sinusoidal con  $\hat{\lambda} = 4$  para dos amplitudes, dos caudales adimensionales y dos números de Rayleigh (figuras 5.10a y 5.10b). En primera instancia se aprecia que el contorno induce una fluctuación en el tiempo de cierta amplitud. Esta amplitud es mayor para mayores amplitudes del contorno y para mayores velocidades (caudales). De hecho, cuando el caudal es nulo, se puede ver el caso de convección natural, en que el valor de Nusselt no varía en el contorno inferior. Un aumento del caudal adimensional, en cambio, reduce el valor medio de Nusselt. Por otro lado, un mayor Rayleigh impuesto supone un mayor valor medio de Nusselt. La amplitud de la fluctuación se mantiene del mismo orden pese al aumento del valor de Rayleigh. Vuelve a verse aquí la importancia relativa entre número de Rayleigh y caudal impuesto. Un mayor caudal reduce la deformación del campo de temperaturas.



**Figura 5.10** Nusselt en contorno inferior (sinusoidal con  $\hat{\lambda} = 4$  y dos amplitudes  $A$ ) a lo largo del tiempo en función de  $\hat{\Psi}_{sup}$  para dos números de Rayleigh.

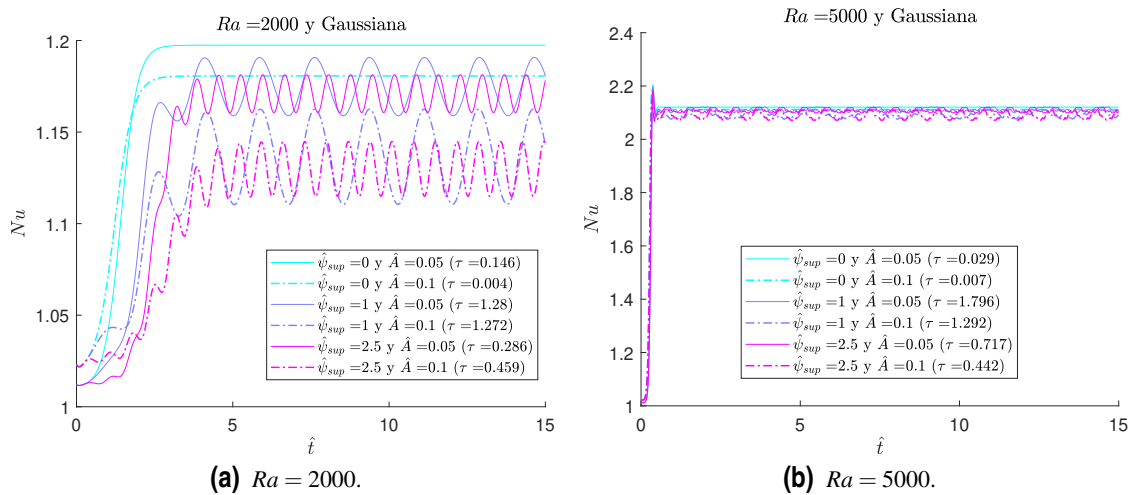
En las figuras 5.11a y 5.11b se muestra el mismo resultado pero para  $\hat{\lambda} = 2$ . En este caso las



**Figura 5.11** Nusselt en contorno inferior (sinusoidal con  $\hat{\lambda} = 2$  y dos amplitudes  $A$ ) a lo largo del tiempo en función de  $\hat{\Psi}_{sup}$  para dos números de Rayleigh.

fluctuaciones son mayores. Un contorno definido por una frecuencia menor aumenta la amplitud de las fluctuaciones del número de Nusselt en el contorno inferior, lo que a su vez supone mayores fluctuaciones en el campo de temperatura, que vendrá originado mayores velocidades.

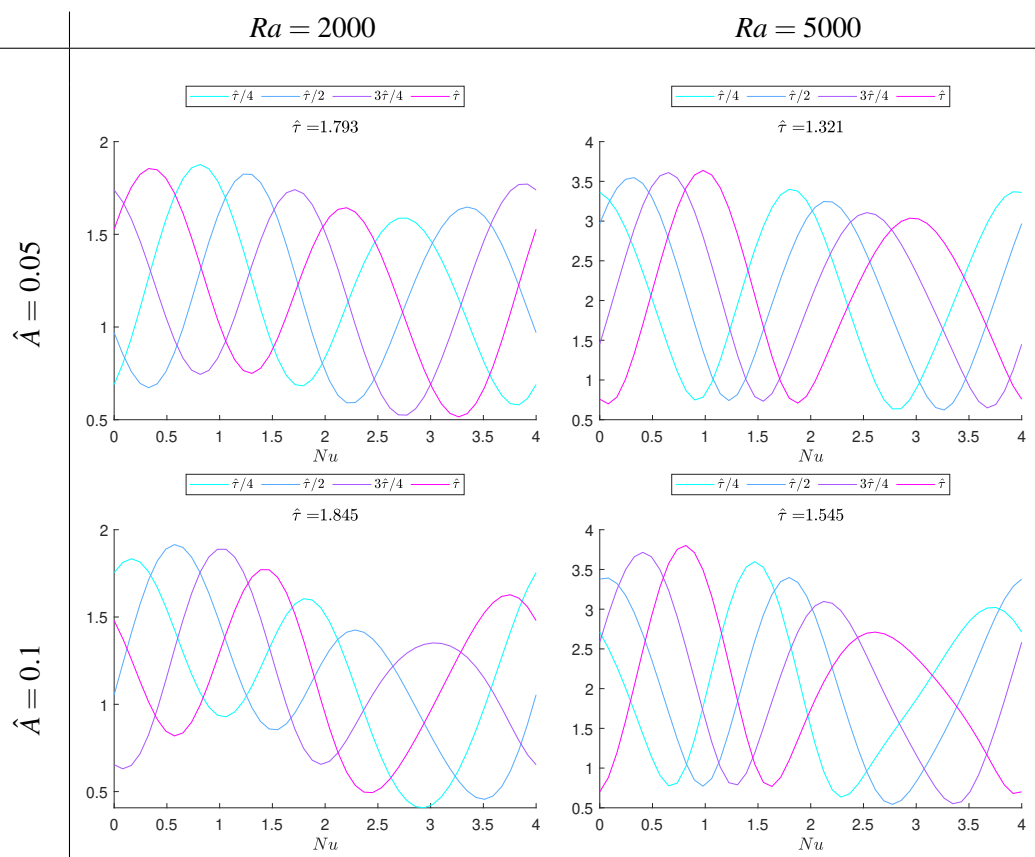
Por último se muestra el mismo resultado para el caso en que el contorno inferior esta definido por la función gaussiana (figuras 5.12a y 5.12b). En este caso, al existir solo un montículo, la amplitud de las fluctuaciones disminuye respecto al último caso y se parece más al primero.



**Figura 5.12** Nusselt en contorno inferior (definido por función gaussiana con dos amplitudes  $A$ ) a lo largo del tiempo en función de  $\hat{\Psi}_{sup}$  para dos números de Rayleigh.

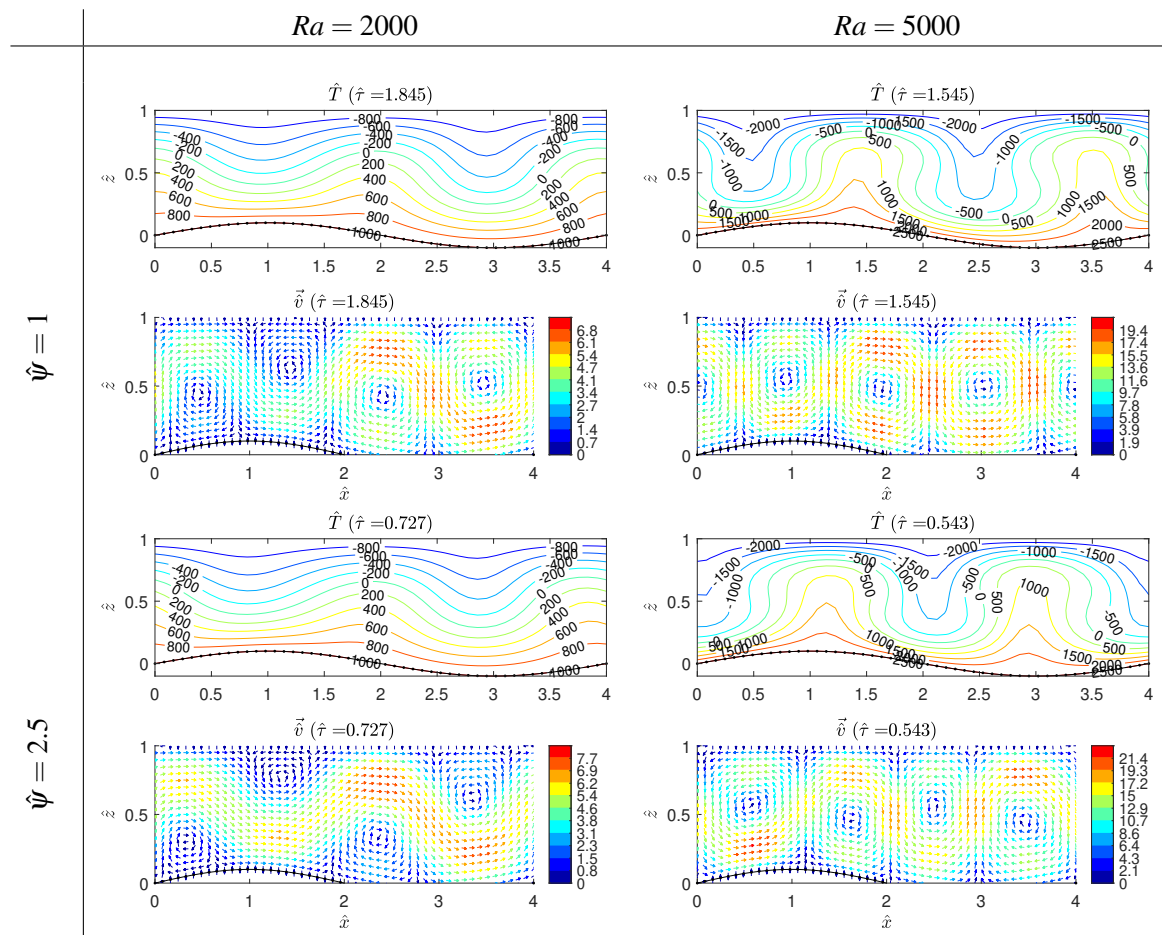
A modo de ejemplo se muestra como se distribuye el número de Nusselt en el contorno inferior a lo largo de un periodo para  $\hat{\lambda} = 4$  y  $A = 0.1$ . Por lo general se observan curvas más complejas que al tener contornos rectos. Como se ve en las imágenes 5.8 y 5.9, los montículos afectan a la estructura de las celdas convectivas, apareciendo zonas de mayor velocidad debido a la reducción de la sección de paso.

**Tabla 5.7** Distribución de Nusselt en el contorno inferior (sinusoidal con  $\hat{\lambda} = 4$  y dos amplitudes  $\hat{A}$ ) para  $\hat{\psi} = 1$  y dos números de Rayleigh.

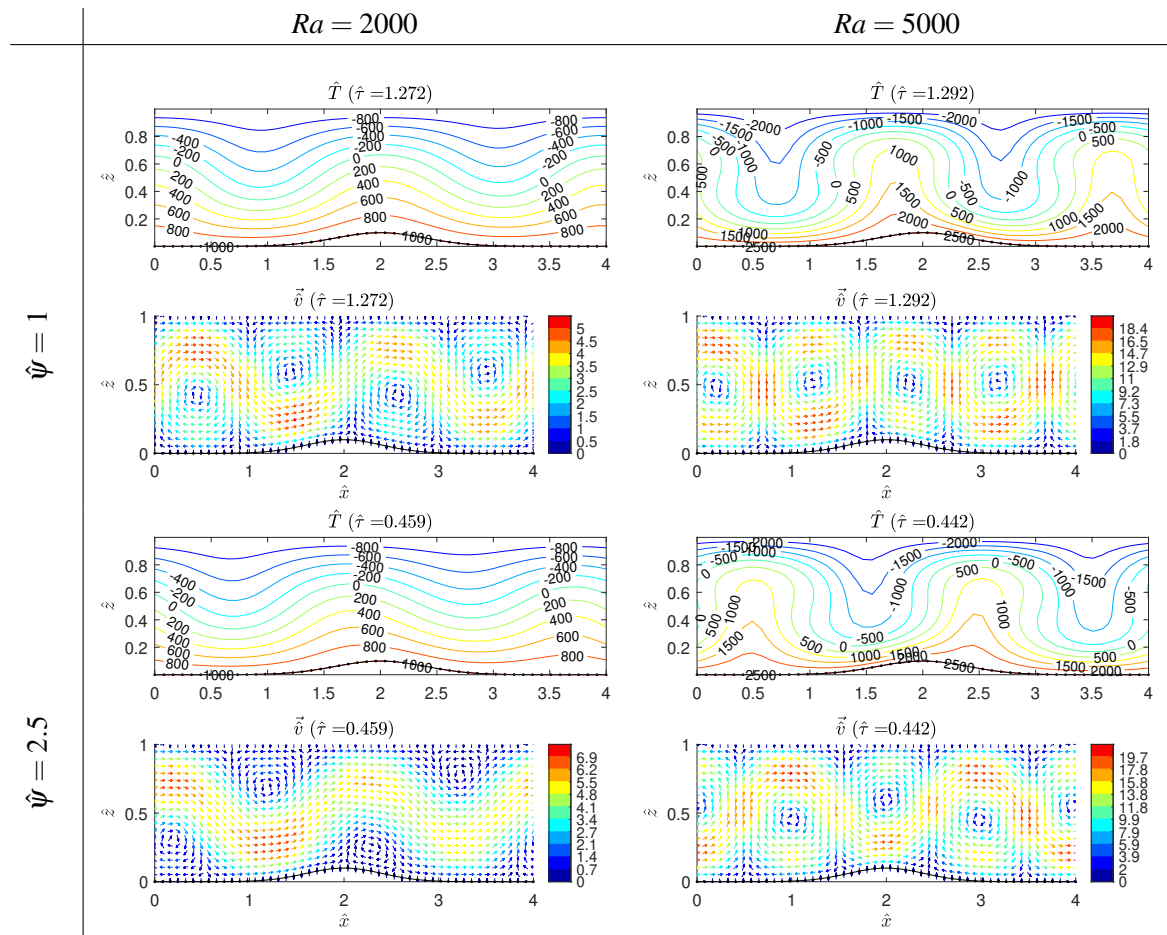




**Tabla 5.8** Velocidades y temperaturas adimensionales para contorno inferior sinusoidal con  $\hat{\lambda} = 4$  y  $\hat{A} = 0.1$ , dos caudales adimensionales y dos números de Rayleigh.



**Tabla 5.9** Velocidades y temperaturas adimensionales para contorno basado en función gaussiana,  $\hat{A} = 0.1$ , dos caudales adimensionales y dos números de Rayleigh.



## 6 Conclusiones y líneas futuras

---

A lo largo del proyecto se ha implementado una forma de resolver problemas de convección en dominios bidimensionales y con régimen laminar, resolviendo las ecuaciones de Navier-Stokes aplicando las hipótesis del modelo de Boussinesq. En la actualidad existe un gran número de herramientas que podría resolver este tipo de problemas con facilidad, aunque lo más normal es que sean tipo "caja negra", es decir, el usuario no sepa como se esta resolviendo realmente, perdiendo la perspectiva. Por ello se ha optado por implementar en *Matlab* las ecuaciones del problema y su posterior análisis. El objetivo principal es que esta sea didáctica, haciendo entender la complejidad y todos los pasos que hay que seguir hasta llegar a la solución.

Para ello se ha hecho uso de un método de colocación con el fin de llegar a un sistema matricial que permitiese obtener la solución en los nodos. Formulando las ecuaciones y condiciones de contorno se llega a la existencia de un número adimensional que domina el problema: el número de Rayleigh. Además de este aparecen otros como pueden ser el número de Prandtl.

Una de las dos principales ventajas del método utilizado es el tratamiento de las derivadas espaciales con operadores matriciales, universales e independientes al método de cálculo las derivadas. En primer lugar se obtienen operadores basados en una sola dimensión, y posteriormente mediante el producto de *Kronecker* se calculan los operadores aplicables al dominio en dos dimensiones. En este proceso se ha incluido un cambio de variable para transformar el plano coordenado de manera que permite resolver contornos no rectos en una de las dos direcciones. Además, se han implementado varias formas de obtener estos operadores, cuya diferencia es principalmente el número de nodos que se usa. En el caso de utilizar diferencias finitas las matrices son dispersas al basarse únicamente en los nodos más próximos. La otra gran ventaja del método usado en el proyecto es la forma de implementar las condiciones de contorno, basta con sustituir sus expresiones en la matriz de ecuaciones del sistema en la fila correspondiente a los nodos del contorno o subcontorno en cuestión. De esta forma se pueden imponer tantas como sean necesarias según el mayor orden de derivada que aparezca en cada variable y dirección en las ecuaciones.

En cuanto a la integración temporal se ha usado un método semi-implícito: se aplica un paso de tiempo discreto a partir de la solución en un instante y se resuelve el sistema haciendo uso del valor de algunas incógnitas en el instante de tiempo anterior. Se ha comprobado la importancia de la condición de Courant-Friedrichs-Levy (condición CFL) a la hora de escoger una paso de tiempo para lograr la convergencia de la solución.

Durante todo el proyecto la línea seguida se ha basado en un proceso de validar los diferentes métodos y buscar la optimización numérica de estos fijando algunos parámetros para los siguientes estudios, logrando a lo largo del proyecto entender el proceso de convección con mayor profundidad. En primer lugar se hizo una linealización en torno a un estado de equilibrio de las ecuaciones, que permitiesen, mediante un cálculo de autovalores y autovectores, encontrar el número de Rayleigh

crítico, que es aquel a partir del cual surge la inestabilidad que produce el inicio del proceso convectivo. Este estudio se realizó para dominios con contornos rectos y dos condiciones de contorno diferentes, que simulan a las que existen en un fluido entre dos placas infinitas (problema de Rayleigh-Bénard). Con el fin de simular estas placas infinitas se incluyó la relación de aspecto en el estudio, aumentando su valor hasta ser lo suficientemente grande. Así se pudo fijar a priori un número de nodos válido para los siguientes apartados. El valor de Rayleigh crítico fue fundamental para validar el cálculo de los operadores matriciales de derivadas, así como para realizar un estudio paramétrico en función del número y la distribución de los nodos. Este sirvió también para comparar los tiempos de computación de las diferentes partes del método. Los siguientes casos fueron la resolución de un dominio con condiciones periódicas en los contornos laterales (simulando las condiciones originales de Rayleigh-Bénard) y la resolución de un dominio con contornos laterales curvos. Realizar este análisis lineal tenía dos ventajas principalmente. Por un lado, no aparecen algunas magnitudes que si que habrá que incluir en el cálculo completo de las ecuaciones como puede ser el paso de tiempo o una posible perturbación inicial que haya que escoger. Esto reduce el número de parámetros del estudio. Por otro lado no hay que establecer un criterio de convergencia en el tiempo a partir del cual analizar el campo fluido. En cambio, un solo escalar, que puede obtenerse sin resolver las ecuaciones a lo largo del tiempo, permite una comparación con resultados numéricos de la literatura, logrando así la validación que se buscaba.

Una vez realizado el estudio de estabilidad con las ecuaciones linealizadas se tiene un criterio para escoger el número de nodos necesario, así como las ventajas y desventajas entre usar diferencias finitas (con varias distribución de nodos) o todos los nodos de Chebyshev. El siguiente paso fue resolver las ecuaciones completas a lo largo del tiempo. Tras algunas pruebas se supuso que el estado inicial del dominio sería el de equilibrio más una pequeña perturbación. Esto se ha mantenido así a lo largo del resto del proyecto, incluso para aquellos casos en los que no existe solución de equilibrio debido a las condiciones de contorno. No obstante, también se ha supuesto un estado inicial en que la temperatura es lineal con la altura para los demás casos. Observando las magnitudes del campo fluido se profundizó en la importancia del número de nodos y su distribución (variando el criterio obtenido anteriormente), llegando a la conclusión de que el uso de diferencias finitas con una malla uniforme es más estable numéricamente para el método semi-implícito propuesto, ya que es el que más control ofrece para mantener la condición CFL. Los resultados no se ven comprometidos respecto al uso de colocación con todos los nodos de Chebyshev, que aunque sea el más exacto, requiere un paso de tiempo menor al concentrarse los nodos en los contornos. También se pudo observar la creación a lo largo del tiempo de las celdas o rollos de Bénard. Además, se verificó en casos muy concretos el número de Rayleigh crítico. Para ello se observó la variación del flujo de calor en el contorno inferior en función del número de Rayleigh impuesto.

A continuación se resolvió un dominio rectangular cuyas condiciones de contorno simulaban un tramo intermedio de un conducto suficientemente largo para que las condiciones de entrada y salida no influyan en este. Para ello se imponen condiciones de periodicidad en la dirección longitudinal de este. La novedad respecto a los casos anteriores es la imposición de un caudal de flujo para resolver convección forzada. El principal resultado que se observa en este caso es el flujo de calor en el contorno inferior a través del número de Nusselt. Se observa el aumento de este cuando el número de Rayleigh es mayor al crítico y la dependencia con el caudal de flujo impuesto o la relación de aspecto. Además, al igual que sucedía al aplicar condiciones periódicas al dominio sin caudal, el valor de la relación de aspecto es fundamental para conseguir captar la estructura de un supuesto problema real que se quiera modelar. Los efectos de esta disminuyen cuanto mayor es esta, reduciendo los fenómenos locales frente a la escala global. Por último, se repite el análisis anterior pero variando la geometría del contorno inferior (en este caso se fija  $AR = 4$ ). En este caso se observa una fluctuación del número de Nusselt en el contorno inferior, lo que implica un campo de temperaturas que no solo se desplaza sino que se deforma con el tiempo debido a los

---

contornos. Queda patente en los resultados la dependencia de estas fluctuaciones con la altitud de los montículos, la frecuencia de estos, el caudal impuesto y el número de Rayleigh.

Para concluir se proponen algunas posibles líneas futuras que podrían ser de interés:

- Aplicar infinidad de condiciones de contorno de interés, incluyendo algunas que varíen a lo largo del contorno. Por ejemplo, podría resolverse el conducto de manera que existan tramos cuyas paredes sean isotermas y otros con condiciones adiabáticas.
- Modelar contornos no rectos en ambas direcciones o realizar un cambio de variable elíptico.
- Aplicar el análisis de estabilidad lineal a dominios con contornos no rectos de amplitud pequeña.
- Buscar una variación del método semi-implícito para resolver a lo largo del tiempo. Un gran avance sería permitir que el paso de tiempo fuese mayor. Para ello podría incluirse un método iterativo dentro de cada paso temporal.
- Manteniendo la estructura de resolución formular las ecuaciones en coordenadas polares, que permitan resolver con facilidad dominios circulares.
- Ampliación de las ecuaciones para resolver dominios tridimensionales.
- Partir de otros modelos de convección: variaciones de Boussinesq, modelo anelástico, modelos turbulentos, etc.



# Anexo A

## Cálculo de derivadas

---

### A.1 Contornos horizontales no rectos

#### A.1.1 Derivadas parciales respecto a variables $x$ y $z$

$$\phi(x,z) = \Phi(X,Z) \tag{A.1}$$

$$Z(x) = \frac{z - z_b(x)}{p(x)}, \quad \text{siendo } p(x) = z_t(x) - z_x(x) \tag{A.2}$$

$$\frac{\partial \phi}{\partial z} = \Phi_Z \left( \frac{1}{z_t(X) - z_b(X)} \right) \tag{A.3}$$

$$\frac{\partial^2 \phi}{\partial z^2} = \Phi_{ZZ} \left( \frac{1}{z_t(X) - z_b(X)} \right)^2 \tag{A.4}$$

$$\frac{\partial^3 \phi}{\partial z^3} = \Phi_{ZZZ} \left( \frac{1}{z_t(X) - z_b(X)} \right)^3 \tag{A.5}$$

$$\frac{\partial^4 \phi}{\partial z^4} = \Phi_{ZZZZ} \left( \frac{1}{z_t(X) - z_b(X)} \right)^4 \tag{A.6}$$

$$\frac{\partial \phi}{\partial x} = \Phi_X + Z_X \Phi_Z \tag{A.7}$$

$$\frac{\partial^2 \phi}{\partial x^2} = \Phi_{XX} + 2Z_X \Phi_{XZ} + Z_X^2 \Phi_{ZZ} + Z_{XX} \Phi_Z \tag{A.8}$$

$$\frac{\partial^3 \phi}{\partial x^3} = \Phi_{XXX} + 3Z_X \Phi_{XXZ} + 3Z_X^2 \Phi_{XZZ} + Z_X^3 \Phi_{ZZZ} + 3Z_X Z_{XX} \Phi_{ZZ} + Z_{XXX} \Phi_Z + 3Z_{XX} \Phi_{XZ} \tag{A.9}$$

$$\begin{aligned} \frac{\partial^4 \phi}{\partial x^4} = & \Phi_{XXXX} + 4Z_X \Phi_{XXXZ} + 6Z_X^2 \Phi_{XXZZ} + 4Z_X^3 \Phi_{XZZZ} + Z_X^4 \Phi_{ZZZZ} + 6Z_{XX} \Phi_{XXZ} \\ & + 12Z_X Z_{XX} \Phi_{XZZ} + 6Z_X^2 Z_{XX} \Phi_{ZZZ} + 4Z_{XXX} \Phi_{XZ} + (4Z_{XXX} Z_X + 3Z_{XX}^2) \Phi_{ZZ} + Z_{XXXX} \Phi_Z \end{aligned} \tag{A.10}$$

A.1.2 Derivadas de  $Z$  respecto a  $X = x$ 

De manera general se definen las funciones que representan los contornos superior e inferior como sigue, en función de una respectiva amplitud, longitud de onda y un desfase ( $f(t)$  y  $g(t)$  que se va a considerar salvo que se especifique lo contrario nulo).

$$Z(x) = \frac{z - z_b(x)}{p(x)}, \quad \text{siendo } p(x) = z_t(x) - z_x(x) \quad (\text{A.11})$$

$$Z_X = -\frac{Zp^I + z_b^I}{p} \quad (\text{A.12})$$

$$Z_{XX} = -\frac{Z_X p^I + Zp^{II} + z_b^{II}}{p} + \frac{p^I(Zp^I + z_b^I)}{p^2} \quad (\text{A.13})$$

$$Z_{XXX} = -\frac{Z_{XX} p^I + 2Z_X p^{II} + Zp^{III} + z_b^{III}}{p} + \frac{p^{II}(Zp^I + z_b^I) + 2p^I(Z_X p^I + Zp^{II} + z_b^{II})}{p^2} - \frac{(p^I)^2(Zp^I + z_b^I)}{p^3} \quad (\text{A.14})$$

$$Z_{XXXX} = -\frac{Z_{XXX} p^I + 3Z_{XX} p^{II} + 3Z_X p^{III} + Zp^{IV} + z_b^{IV}}{p} + \frac{p^{III}(Zp^I + z_b^I) + 3p^{II}(Z_X p^I + Zp^{II} + z_b^{II}) + p^I(Z_{XX} p^I + 2Z_X p^{II} + Zp^{III} + z_b^{III})}{p^2} - \frac{4p^I p^{II}(Zp^I + z_b^I) + 5(p^I)^2(Z_X p^I + Zp^{II} + z_b^{II})}{p^3} + \frac{3(p^I)^3(Zp^I + z_b^I)}{p^4} \quad (\text{A.15})$$

Contorno inferior sinusoidal con longitud de onda  $\lambda_b$  y amplitud  $A_b$ 

$$z_b = \text{Im} \left( A_b e^{2\pi i \left( \frac{x}{\lambda_b} - f(t) \right)} \right) \quad (\text{A.16})$$

$$z_b^I = \text{Im} \left( \left( \frac{2\pi i}{\lambda_b} \right) z_b \right) \quad (\text{A.17})$$

$$z_b^{II} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_b} \right)^2 z_b \right) \quad (\text{A.18})$$

$$z_b^{III} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_b} \right)^3 z_b \right) \quad (\text{A.19})$$

$$z_b^{IV} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_b} \right)^4 z_b \right) \quad (\text{A.20})$$



**Contorno inferior basado en función gaussiana  $(\mu, \sigma)$  con amplitud  $A_b$** 

$$z_b = A_b \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{X-\mu}{\sigma}\right)^2} \quad (\text{A.21})$$

$$z_b^I = -\frac{X-\mu}{\sigma^2} z_b(X) \quad (\text{A.22})$$

$$z_b^{II} = \left[ -\frac{1}{\sigma^2} + \left( \frac{X-\mu}{\sigma^2} \right)^2 \right] z_b(X) \quad (\text{A.23})$$

$$z_b^{III} = \left[ \left( \frac{X-\mu}{\sigma^2} \right)^3 + 3 \frac{X-\mu}{\sigma^4} \right] z_b(X) \quad (\text{A.24})$$

$$z_b^{IV} = \left[ \left( \frac{X-\mu}{\sigma^2} \right)^4 - \frac{6}{\sigma^2} \left( \frac{X-\mu}{\sigma^2} \right)^2 + \frac{3}{\sigma^4} \right] z_b(X) \quad (\text{A.25})$$

**Contorno superior sinusoidal con longitud de onda  $\lambda_t$  y amplitud  $A_t$** 

$$z_t = \text{Im} \left( A_t e^{2\pi i \left( \frac{x}{\lambda_t} - g(t) \right)} \right) \quad (\text{A.26})$$

$$z_t^I = \text{Im} \left( \left( \frac{2\pi i}{\lambda_t} \right) z_t \right) \quad (\text{A.27})$$

$$z_t^{II} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_t} \right)^2 z_t \right) \quad (\text{A.28})$$

$$z_t^{III} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_t} \right)^3 z_t \right) \quad (\text{A.29})$$

$$z_t^{IV} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_t} \right)^4 z_t \right) \quad (\text{A.30})$$

## A.2 Contornos verticales no rectos

### A.2.1 Derivadas parciales respecto a variables $x$ y $z$

$$\phi(x,z) = \Phi(X,Z) \quad (\text{A.31})$$

$$X(z) = \frac{x - x_l(z)}{p(z)} AR, \quad \text{siendo } p(z) = x_r(z) - x_l(z) \quad (\text{A.32})$$

$$\frac{\partial \phi}{\partial x} = \Phi_X \left( \frac{AR}{x_r(Z) - x_l(Z)} \right) \quad (\text{A.33})$$

$$\frac{\partial^2 \phi}{\partial x^2} = \Phi_{XX} \left( \frac{AR}{x_r(Z) - x_l(Z)} \right)^2 \quad (\text{A.34})$$

$$\frac{\partial^3 \phi}{\partial x^3} = \Phi_{XXX} \left( \frac{AR}{x_r(Z) - x_l(Z)} \right)^3 \quad (\text{A.35})$$

$$\frac{\partial^4 \phi}{\partial x^4} = \Phi_{XXXX} \left( \frac{AR}{x_r(Z) - x_l(Z)} \right)^4 \quad (\text{A.36})$$

$$\frac{\partial \phi}{\partial z} = \Phi_Z + X_Z \Phi_X \quad (\text{A.37})$$

$$\frac{\partial^2 \phi}{\partial z^2} = \Phi_{ZZ} + 2X_Z \Phi_{ZX} + X_Z^2 \Phi_{XX} + X_{ZZ} \Phi_X \quad (\text{A.38})$$

$$\frac{\partial^3 \phi}{\partial z^3} = \Phi_{ZZZ} + 3X_Z \Phi_{ZZX} + 3X_Z^2 \Phi_{ZXX} + X_Z^3 \Phi_{XXX} + 3X_Z X_{ZZ} \Phi_{XX} + X_{ZZZ} \Phi_X + 3X_{ZZ} \Phi_{ZX} \quad (\text{A.39})$$

$$\begin{aligned} \frac{\partial^4 \phi}{\partial z^4} = & \Phi_{ZZZZ} + 4X_Z \Phi_{ZZZX} + 6X_Z^2 \Phi_{ZZXX} + 4X_Z^3 \Phi_{ZXXX} + X_Z^4 \Phi_{XXXX} + 6X_{ZZ} \Phi_{ZZX} \\ & + 12X_Z X_{ZZ} \Phi_{ZXX} + 6X_Z^2 X_{ZZ} \Phi_{XXX} + 4X_{ZZZ} \Phi_{ZX} + (4X_{ZZZ} X_Z + 3X_{ZZ}^2) \Phi_{XX} + X_{ZZZZ} \Phi_X \end{aligned} \quad (\text{A.40})$$

### A.2.2 Derivadas de $Z$ respecto a $X = x$

De manera general se definen las funciones que representan los contornos superior e inferior como sigue, en función de una respectiva amplitud, longitud de onda y un desfase ( $f(t)$  y  $g(t)$  que se va a considerar salvo que se especifique lo contrario nulo).

$$X(z) = \frac{x - x_l(z)}{p(z)} AR, \quad \text{siendo } p(z) = x_r(z) - x_l(z) \quad (\text{A.41})$$

$$X_Z = -\frac{(X/AR)p^I + X_b^I}{p} \quad (\text{A.42})$$

$$X_{ZZ} = -\frac{(X_Z/AR)p^I + (X/AR)p^{II} + X_b^{II}}{p} + \frac{p^I((X/AR)p^I + X_b^I)}{p^2} \quad (\text{A.43})$$

$$\begin{aligned} X_{ZZZ} = & -\frac{(X_{ZZ}/AR)p^I + 2(X_Z/AR)p^{II} + (X/AR)p^{III} + X_b^{III}}{p} \\ & + \frac{p^{II}((X/AR)p^I + X_b^I) + 2p^I((X/AR)_Z p^I + (X/AR)p^{II} + X_b^{II})}{p^2} \\ & - \frac{(p^I)^2(Xp^I + X_b^I)}{p^3} \end{aligned} \quad (\text{A.44})$$

$$\begin{aligned} X_{ZZZZ} = & -\frac{(X_{ZZZ}/AR)p^I + 3(X_{ZZ}/AR)p^{II} + 3(X_Z/AR)p^{III} + (X/AR)p^{IV} + X_b^{IV}}{p} \\ & + \frac{p^{III}((X/AR)p^I + X_b^I) + 3p^{II}((X_Z/AR)p^I + (X/AR)p^{II} + X_b^{II})}{p^2} \\ & + \frac{p^I((X_{ZZ}/AR)p^I + 2(X_Z/AR)p^{II} + Xp^{III} + X_b^{III})}{p^2} \\ & - \frac{4p^I p^{II}((X/AR)p^I + X_b^I) + 5(p^I)^2((X_Z/AR)p^I + (X/AR)p^{II} + X_b^{II})}{p^3} \\ & + \frac{3(p^I)^3((X/AR)p^I + X_b^I)}{p^4} \end{aligned} \quad (\text{A.45})$$

### Contorno izquierdo sinusoidal con longitud de onda $\lambda_b$ y amplitud $A_b$

$$x_l = \text{Im} \left( A_b e^{2\pi i \left( \frac{z}{\lambda_b} - f(t) \right)} \right) \quad (\text{A.46})$$

$$x_l^I = \text{Im} \left( \left( \frac{2\pi i}{\lambda_b} \right) x_l \right) \quad (\text{A.47})$$

$$x_l^{II} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_b} \right)^2 x_l \right) \quad (\text{A.48})$$

$$x_l^{III} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_b} \right)^3 x_l \right) \quad (\text{A.49})$$

$$x_l^{IV} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_b} \right)^4 x_l \right) \quad (\text{A.50})$$

**Contorno izquierdo parabólico con vértice en el punto medio y amplitud en este punto  $A_b$**

$$x_l = 4A_b Z(Z - 1) \quad (\text{A.51})$$

$$x_l^I = 4A_b(2Z - 1) \quad (\text{A.52})$$

$$x_l^{II} = 8A_b \quad (\text{A.53})$$

$$x_l^{III} = 0 \quad (\text{A.54})$$

$$x_l^{IV} = 0 \quad (\text{A.55})$$

**Contorno derecho sinusoidal con longitud de onda  $\lambda_t$  y amplitud  $A_t$**

$$x_r = \text{Im} \left( A_t e^{2\pi i \left( \frac{z}{\lambda_t} - g(t) \right)} \right) \quad (\text{A.56})$$

$$x_r^I = \text{Im} \left( \left( \frac{2\pi i}{\lambda_t} \right) x_r \right) \quad (\text{A.57})$$

$$x_r^{II} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_t} \right)^2 x_r \right) \quad (\text{A.58})$$

$$x_r^{III} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_t} \right)^3 x_r \right) \quad (\text{A.59})$$

$$x_r^{IV} = \text{Im} \left( \left( \frac{2\pi i}{\lambda_t} \right)^4 x_r \right) \quad (\text{A.60})$$

**Contorno derecho parabólico con vértice en el punto medio y amplitud en este punto  $A_t$**

$$x_l = AR + 4A_t Z(Z - 1) \quad (\text{A.61})$$

$$x_l^I = 4A_t(2Z - 1) \quad (\text{A.62})$$

$$x_l^{II} = 8A_t \quad (\text{A.63})$$

$$x_l^{III} = 0 \quad (\text{A.64})$$

$$x_l^{IV} = 0 \quad (\text{A.65})$$

# Anexo B

## Códigos

### B.1 Definición de la geometría

Listing B.1 Definición de la geometría con contornos horizontales no rectos.

```
1 % Función para la creación de las matrices y vectores necesarios para el
2 % tratamiento de la geometría del dominio con contornos no rectos superior
3 % y/o inferior. Argumentos de entrada:
4 % - Nx, Nz      : número de nodos en dirección 'x' y 'z'
5 % - malla_x, malla_z : 'lin' o 'cheb' para malla uniforme o distribución de
6 %               Chebyshev
7 % - fig_dominio  : 1 para mostrar geometría del dominio
8 % - fig_malla   : 1 para mostrar mallado
9 % - H           : altura del dominio
10 % - AR          : relación de aspecto
11 % - At, Ab      : amplitud de la curva sinusoidal o factor que
12 %               multiplica a la función de densidad de probabilidad
13 %               del contorno superior o inferior
14 % - lambdat, lambdab : longitud de onda de la curva sinusoidal del contorno
15 %               superior o inferior
16 %
17 % Pueden ampliarse con:
18 %
19 % - geo         : 0 para curva sinusoidal en contorno inferior, en
20 %               caso contrario será la determinada por una función
21 %               de densidad de probabilidad de una distribución
22 %               normal fijada en el código
23 function [Xch,Zch,Xmat,Zmat,xmat,zmat,Nt,zbv,ztv,zb,zt] = ...
24     geom(Nx,Nz,malla_x,malla_z,fig_dominio,fig_malla,H,AR,At,Ab,...
25     lambdat,lambdab,geo)
26 % tic
27 Nt = Nz*Nx; % Número total de nodos
28 %%%%% Límites de z %%%%%
29 zmin = 0;
30 zmax = H;
31 zmin_uni = 0;
32 zmax_uni = 1;
33 %%%%% Límites de x %%%%%
34 xmin = 0;
35 xmax = (H-zmin)*AR+xmin;
36
37 %% Mallado en dominio transformado
38
39 % Malla en X
40 switch malla_x
41 case 'lin'
42     Xch = linspace(xmin,xmax,Nx)';
43 case 'cheb'
44     Xch = (xmax+xmin)/2 - (xmax-xmin)/2*cos(((1:Nx)-1)*pi/(Nx-1))';
45 end
46
47 % Malla en Z
48 switch malla_z
49 case 'lin'
50     Zch = linspace(zmin_uni,zmax_uni,Nz)';
51 case 'cheb'
52     Zch = (zmax_uni+zmin_uni)/2 - ...
53         (zmax_uni-zmin_uni)/2*cos(((1:Nz)-1)*pi/(Nz-1))';
54 end
55
56 [Xmat,Zmat] = meshgrid(Xch,Zch);
57
```

```

58 %% Mallado en dominio original
59 % Limite superior
60 ztv = H-At*imag(exp(2*pi*1i*(Xch/lambdat)));
61 % Limite inferior
62 if geo == 0
63     zbv = Ab*imag(exp(2*pi*1i*(Xch/lambdab)));
64     mu = NaN;
65     s = NaN;
66 else
67     mu = AR*0.35;
68     s = AR/10;
69     zbv = Ab*normpdf(Xch,mu,s);
70 end
71
72 % exp(2*pi*(x/lambda-ft*t)), los límites cambiarían con el tiempo si ft ~= 0
73
74 % Matrices de mallado de los límites inferior y superior
75 zbmat = repmat(zbv',Nz,1);
76 zb = reshape(zbmat,Nt,1);
77 ztmat = repmat(ztv',Nz,1);
78 zt = reshape(ztmat,Nt,1);
79
80 % Matrices del mallado original
81 xmat = Xmat;
82 zmat = Zmat.*(ztmat-zbmat)+zbmat;
83
84 %% Representaciones del dominio original y de su mallado
85 % Representación del dominio original
86 if fig_dominio == 1
87     figure, title('Dominio'), xlabel('x'), ylabel('z')
88     plot(Xch,zbv), hold on, plot(Xch,ztv)
89 end
90 % Representación de la malla del dominio original
91 if fig_malla == 1
92     figure,
93     surf(xmat,zmat,ones(size(xmat)));
94     view(0,90)
95 end
96 % t_geom = toc; disp(['----- t_geom = ',num2str(t_geom)])
97 end

```

Listing B.2 Definición de la geometría con contornos verticales no rectos.

```

1 % Función para la creación de las matrices y vectores necesarios para el
2 % tratamiento de la geometría del dominio con contornos no rectos izquierdo
3 % y/o derecho. Argumentos de entrada:
4 % - Nx, Nz : número de nodos en dirección 'x' y 'z'
5 % - malla_x, malla_z : 'lin' o 'cheb' para malla uniforme o distribución de
6 % Chebyshev
7 % - fig_dominio : 1 para mostrar geometría del dominio
8 % - fig_malla : 1 para mostrar mallado
9 % - H : altura del dominio
10 % - AR : relación de aspecto
11 % - At, Ab : amplitud de la curva sinusoidal del contorno
12 % derecho e izquierdo o máxima distancia si el
13 % contorno es parabólico respecto a la vertical
14 % - lambdat, lambdab : longitud de onda de la curva sinusoidal del contorno
15 % derecho e izquierdo
16 % - geo : 'par' o 'sin' según se quiera un contorno lateral
17 % parabólico o sinusoidal
18
19 function [Xch,Zch,Xmat,Zmat,xmat,zmat,Nt,xlv,xrv,xl,xr] = ...
20 geom(Nx,Nz,malla_x,malla_z,fig_dominio,...
21 fig_malla,H,AR,At,Ab,lambdat,lambdab,geo)
22
23 % tic
24
25 Nt = Nz*Nx; % Número total de nodos
26
27 %%%%% Límites de z %%%%%
28 zmin = 0;
29 zmax = H;
30 zmin_uni = 0;
31 zmax_uni = 1;
32 %%%%% Límites de x %%%%%
33 xmin = 0;
34 xmax = (H-zmin)*AR+xmin;
35
36 %% Mallado en dominio transformado
37
38 % Malla en X
39 switch malla_x
40 case 'lin'
41     Xch = linspace(xmin,xmax,Nx)';
42 case 'cheb'
43     Xch = (xmax+xmin)/2 - (xmax-xmin)/2*cos(((1:Nx)-1)*pi/(Nx-1))';
44 end
45
46 % Malla en Z
47 switch malla_z
48 case 'lin'
49     Zch = linspace(zmin_uni,zmax_uni,Nz)';

```

```

50 case 'cheb'
51     Zch = (zmax_uni+zmin_uni)/2-...
52         (zmax_uni-zmin_uni)/2*cos(((1:Nz)-1)*pi/(Nz-1));
53 end
54
55 [Xmat,Zmat] = meshgrid(Xch,Zch);
56
57 %% Mallado en dominio original
58 switch geo
59     case 'sin'
60         xrv = AR+At*imag(exp(2*pi*i*(Zch/lambdat)));
61         xlv = Ab*imag(exp(2*pi*i*(Zch/lambdab)));
62     case 'par'
63         xrv = AR+4*At*Zch.*(Zch-1);
64         xlv = 4*Ab*Zch.*(Zch-1);
65     end
66
67 % Matrices de mallado de los límites inferior y superior
68 xlm = repmat(xlv,1,Nx);
69 xrm = repmat(xrv,1,Nx);
70 xl = reshape(xlm,Nt,1);
71 xr = reshape(xrm,Nt,1);
72
73 % Matrices del mallado original
74 zmat = Zmat;
75 xmat = Xmat.*(xrmat-xlmat)/AR+xlm;
76
77 %% Representaciones del dominio original y de su mallado
78 % Representación del dominio original
79 if fig_dominio == 1
80     figure, title('Dominio'), xlabel('x'), ylabel('z')
81     plot(xlv,Zch), hold on, plot(xrv,Zch)
82 end
83 % Representación de la malla del dominio original
84 if fig_malla == 1
85     figure,
86     surf(xmat,zmat,ones(size(xmat)));
87     view(0,90)
88 end
89 % t_geom = toc; disp(['----- t_geom = ',num2str(t_geom)])
90
91 end

```

## B.2 Cálculo de derivadas

Listing B.3 Matrices de derivadas con contornos horizontales no rectos.

```

1 % Función para calcular las matrices de derivadas 'D', aplicables al
2 % dominio 2D. Opcionalmente puede incluir los vectores normal y tangencial
3 % al contorno inferior (N y DS). Argumentos de entrada:
4 % - difX, difZ : 'cheb' para calcular las matrices de derivadas
5 %               unidimensionales 'x' o 'z' de orden dos o superior como
6 %               producto de las de orden 1.
7 % - DXs, DZs : 'prod' o 'derivadas' para calcular las matrices de
8 %               derivadas de orden 2 o mayor, aplicables al dominio 2D
9 %               en variables transformadas, como producto de las de
10 %              primer orden o a partir de las matrices de derivadas
11 %              unidimensionales de mayor orden
12 % - Dxs, Dzs : 'prod' o 'cadena' para calcular las matrices de
13 %               derivadas de orden 2 o mayor, aplicables al dominio 2D
14 %               respecto a las variables originales, aplicando en todas
15 %               la regla de la cadena o solo en la primera y luego
16 %               realizando productos de esta
17 % - Xmat, Zmat : matrices de nodos en el dominio transformado 'X' y 'Z'
18 % - Nx, Nz, Nt : número de nodos en 'x' y 'z', y total
19 % - zb, zt : matrices con contornos inferior y superior
20 % - xch, zch : nodos en 'x' y 'z'
21 % - At, Ab : amplitud de la curva sinusoidal o factor que
22 %             multiplica a la función de densidad de probabilidad
23 %             del contorno superior o inferior
24 % - lambdat, lambdab : longitud de onda de la curva sinusoidal del contorno
25 %                     superior o inferior
26 %
27 % Pueden ampliarse con:
28 % - geo : 0 para curva sinusoidal en contorno inferior, en
29 %         caso contrario será la determinada por una función
30 %         de densidad de probabilidad de una distribución
31 %         normal con media 'mu' y desviación típica 's'
32 % - mu : media de la distribución
33 % - s : desviación típica de la distribución
34 % - zbv : vector con posición 'z' del contorno inferior
35 %
36
37 function [Dx,Dx2,Dx3,Dx4,Dz,Dz2,Dz3,Dz4,DL,DL2,DxDL,DzDL,N,DS] = ...
38     mat_derivadas_topbot(difX,difZ,DXs,DZs,Dxs,Dzs,...
39     Xmat,Zmat,Nx,Nz,Nt,zb,zt,xch,zch,At,Ab,lambdat,lambdab,geo,AR,mu,s,zbv)
40

```

```

41 %%%% Distribución de los nodos: malla_x y malla_y %%%%
42 % 'lin': equiespaciada
43 % 'cheb': de Chebyshev
44
45 %%%% Cálculo de las derivadas unidimensionales: difX y difZ %%%%
46 % 'cheb' : colocación con Chebyshev
47     switch difX
48     case 'cheb'
49         DXs = 'prod';
50     end
51     switch difZ
52     case 'cheb'
53         DZs = 'prod';
54     end
55 % 'difx' : dif fin. despreciando ordenes superiores a x (x=2,4,6,8)
56 % (Matriz mal condicionada, malos resultados)
57 % 'Ponberg': especificar número de nodos, NF, para el cálculo de
58 % la derivada en cada nodo (incluyendo este),
59 % y 'm' (1+orden de derivada superior)
60 NF_x = 7;
61 m_x = 7;
62 NF_z = 7;
63 m_z = 7;
64
65 %%%% Cálculo de matrices unidimensionales respecto a X y Z: DXs y DZs %%%%
66 % 'prod': derivadas orden > 1 como producto de las de primer orden
67 % 'derivadas': calculadas por diferencias finitas
68
69 %%%% Cálculo de matrices de derivadas respecto a x y z: Dxs y Dzs %%%%
70 % 'prod': por productos de derivadas primeras
71 % 'cadena': aplicando la regla de la cadena
72
73 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
74
75 % tic
76 X = reshape(Xmat,Nt,1);
77 Z = reshape(Zmat,Nt,1);
78 % t_reshape_nodos = toc; disp(['t_reshape_nodos = ',num2str(t_reshape_nodos)])
79
80 % t_derivadas_tic = tic;
81
82 % Derivadas de zb(x) y zt(x) respecto a x
83 % tic
84 if geo == 0
85     zbaux = Ab*exp(2*pi*1i*(X/lambdab));
86     zb_x = imag((2*pi*1i)/lambdab*zbaux);
87     zb_xx = imag((2*pi*1i)^2/lambdab*zbaux);
88     zb_xxx = imag((2*pi*1i)^3/lambdab*zbaux);
89     zb_xxxx = imag((2*pi*1i)^4/lambdab*zbaux);
90 else
91     Xmu = X-mu;
92     zbaux = Ab*1/s/sqrt(2*pi)*exp(-1/2*((Xmu/s).^2);
93     zb_x = -(Xmu)/s^2.*zbaux;
94     zb_xx = (-1/s^2+(Xmu/s^2).^2).*zbaux;
95     zb_xxx = -(Xmu/s^2).^3+3*Xmu/s^4).*zbaux;
96     zb_xxxx = (Xmu/s^2).^4-(Xmu/s^2).^2*6/s^2+3/s^4).*zbaux;
97 end
98 Ndx = zb_x;
99 Ndz = -1;
100 N = bsxfun(@rdivide,[Ndx,Ndz*ones(length(Ndx),1)],sqrt(Ndx.^2+Ndz.^2));
101 DS = [0;sqrt(diff(zbv).^2 + diff(xch).^2)'];
102
103
104 ztaux = At*exp(2*pi*1i*(X/lambdat));
105 zt_x = imag((2*pi*1i)/lambdat*ztaux);
106 zt_xx = imag((2*pi*1i)^2/lambdat*ztaux);
107 zt_xxx = imag((2*pi*1i)^3/lambdat*ztaux);
108 zt_xxxx = imag((2*pi*1i)^4/lambdat*ztaux);
109 % t_zb_xs = toc; disp(['t_zb_xs = ',num2str(t_zb_xs)])
110 clear X
111
112 % Derivadas de Z(x)=(z-zb(x))/(zt(x)-zb(x)) = (z-zb(x))/p(x) respecto a x
113 % tic
114 p = zt-zb;
115 p_x = zt_x-zb_x;
116 p_xx = zt_xx-zb_xx;
117 p_xxx = zt_xxx-zb_xxx;
118 p_xxxx = zt_xxxx-zb_xxxx;
119
120 Z_z = 1./p;
121 Z_X = -(Z.*p_x+zb_x).*p;
122 Z_XX = -(Z_X.*p_x+Z.*p_xx+zb_xx).*p+p_x.*(Z.*p_x+zb_x).(p.^2);
123 Z_XXX = -(Z_XX.*p_x+2*Z_X.*p_xx+Z.*p_xxx+zb_xxx).*p+...
124 p_xx.*(Z.*p_x+zb_x)+2*p_x.*(Z_X.*p_x+Z.*p_xx+zb_xx).(p.^2)+...
125 -(p_x.^2).(Z.*p_x+zb_x).(p.^3);
126 Z_XXXX = -(Z_XXX.*p_x+3*Z_XX.*p_xx+3*Z_X.*p_xxx+Z.*p_xxxx+zb_xxxx).*p+...
127 p_xxx.*(Z.*p_x+zb_x)+3*p_xx.*(Z_X.*p_x+Z.*p_xx+zb_xx)+...
128 p_x.*(Z_XX.*p_x+2*Z_X.*p_xx+Z.*p_xxx+zb_xxx).(p.^2)-...
129 (4*p_x.*p_xx.*(Z.*p_x+zb_x)+...
130 5*(p_x.^2).(Z_X.*p_x+Z.*p_xx+zb_xx).(p.^3)+...
131 3*(p_x.^3).(Z.*p_x+zb_x).(p.^4);
132
133 clear Z_v zb_x zb_xx zb_xxx zb_xxxx zb_x zb_xx zb_xxx zb_xxxx
134 % t_Z_xs = toc; disp(['t_Z_xs = ',num2str(t_Z_xs)])
135
136 % Matrices de derivadas de las variables normalizadas

```



```

137 % tic
138 switch difX
139     case 'dif2'
140         [LX,LX2] = difs2(xch);
141     case 'dif4'
142         [LX,LX2,LX3,LX4] = difs4(xch);
143     case 'dif6'
144         [LX,LX2,LX3,LX4] = difs6(xch);
145     case 'dif8'
146         [LX,LX2,LX3,LX4] = difs8(xch);
147     case 'Fornberg'
148         [~,LX,LX2,LX3,LX4] = DF_Fornberg(xch,NF_x,m_x);
149     case 'cheb'
150         [LX] = dCheby(xch,Nx);
151 end
152 % t_LXs = toc; disp(['t_LXs = ',num2str(t_LXs)])
153
154 % tic
155 switch DXs
156     case 'prod'
157         DX = kron(LX, speye(Nz));
158         DX2 = DX*DX;
159         DX3 = DX*DX2;
160         DX4 = DX*DX3;
161     case 'derivadas'
162         DX = kron(LX, speye(Nz));
163         DX2 = kron(LX2, speye(Nz));
164         DX3 = kron(LX3, speye(Nz));
165         DX4 = kron(LX4, speye(Nz));
166 end
167 DX = spfull(DX); DX2 = spfull(DX2); DX3 = spfull(DX3); DX4 = spfull(DX4);
168 clear LX LX2 LX3 LX4
169 % t_DXs = toc; disp(['t_DXs = ',num2str(t_DXs)])
170
171 % tic
172 switch difZ
173     case 'dif2'
174         [LZ,LZ2] = difs2(zch);
175     case 'dif4'
176         [LZ,LZ2,LZ3,LZ4] = difs4(zch);
177     case 'dif6'
178         [LZ,LZ2,LZ3,LZ4] = difs6(zch);
179     case 'dif8'
180         [LZ,LZ2,LZ3,LZ4] = difs8(zch);
181     case 'Fornberg'
182         [~,LZ,LZ2,LZ3,LZ4] = DF_Fornberg(zch,NF_z,m_z);
183     case 'cheb'
184         [LZ] = dCheby(zch,Nz);
185 end
186 % t_LZs = toc; disp(['t_LZs = ',num2str(t_LZs)])
187
188 % tic
189 switch DZs
190     case 'prod'
191         DZ = kron(speye(Nx), LZ);
192         DZ2 = DZ*DZ;
193         DZ3 = DZ*DZ2;
194         DZ4 = DZ*DZ3;
195     case 'derivadas'
196         DZ = kron(speye(Nx), LZ);
197         DZ2 = kron(speye(Nx), LZ2);
198         DZ3 = kron(speye(Nx), LZ3);
199         DZ4 = kron(speye(Nx), LZ4);
200 end
201 DZ = spfull(DZ); DZ2 = spfull(DZ2); DZ3 = spfull(DZ3); DZ4 = spfull(DZ4);
202 % t_DZs = toc; disp(['t_DZs = ',num2str(t_DZs)])
203 clear LZ LZ2 LZ3 LZ4
204
205 % Matrices de derivadas de las variables reales
206 % tic
207 Dz = bsxfun(@times, Z_z, DZ);
208 switch Dzs
209     case 'prod'
210         Dz2 = Dz*Dz;
211         Dz3 = Dz*Dz2;
212         Dz4 = Dz*Dz3;
213     case 'cadena'
214         Dz2 = bsxfun(@times, Z_z.^2, DZ2);
215         Dz3 = bsxfun(@times, Z_z.^3, DZ3);
216         Dz4 = bsxfun(@times, Z_z.^4, DZ4);
217 end
218 Dz = spfull(Dz); Dz2 = spfull(Dz2); Dz3 = spfull(Dz3); Dz4 = spfull(Dz4);
219 % t_Dzs = toc; disp(['t_Dzs = ', num2str(t_Dzs)])
220
221 % tic
222 Dx = DX+bsxfun(@times, Z_X, DZ); Dx = spfull(Dx);
223 switch Dxs
224     case 'prod'
225         Dx2 = Dx*Dx;
226         Dx3 = Dx*Dx2;
227         Dx4 = Dx*Dx3;
228     case 'cadena'
229         DXZ = DX*DZ;
230         DX2Z = DX*DZ2;
231         DXZ2 = DX*DZ2;
232         DX2Z2 = DX*DZ2Z;

```

```

233 DX3Z = DX*DX2Z;          DX3Z = spfull(DX3Z);
234 DXZ3 = DZ*DXZ2;          DXZ3 = spfull(DXZ3);
235
236 Dx2 = DX2+2*bsxfun(@times,Z_X,DXZ)+...
237     bsxfun(@times,Z_X.^2,DZ2)+bsxfun(@times,Z_XX,DZ);
238 % Dx2 = Dx*Dx;
239 Dx2 = spfull(Dx2);
240
241 Dx3 = DX3+bsxfun(@times,3*Z_X,DX2Z)+...
242     bsxfun(@times,3*(Z_X.^2),DXZ2)+bsxfun(@times,Z_X.^3,DZ3)+...
243     bsxfun(@times,3*Z_XX.*Z_X,DXZ2)+bsxfun(@times,Z_XXX,DZ)+...
244     bsxfun(@times,3*Z_XX,DXZ);
245 % Dx3 = Dx*Dx2;
246 Dx3 = spfull(Dx3);
247
248 Dx4 = DX4+bsxfun(@times,4*Z_X,DX3Z)+...
249     bsxfun(@times,6*(Z_X.^2),DXZ2Z)+...
250     bsxfun(@times,4*(Z_X.^3),DXZ3)+bsxfun(@times,Z_X.^4,DZ4)+...
251     bsxfun(@times,6*Z_XX,DXZ2)+bsxfun(@times,12*Z_XX.*Z_X,DXZ2)+...
252     bsxfun(@times,6*Z_XX.*(Z_X.^2),DZ3)+bsxfun(@times,4*Z_XXX,DXZ)+...
253     bsxfun(@times,4*Z_XXX.*Z_X+3.*(Z_XX.^2),DZ2)+bsxfun(@times,Z_XXXX,DZ);
254 % Dx4 = Dx*Dx3;
255 Dx4 = spfull(Dx4);
256
257 end
258 % t_Dxs = toc; disp(['t_Dxs = ',num2str(t_Dxs)])
259 clear Z_X Z_XX Z_XXX Z_XXXX DX DX2 DX3 DX4 DZ DZ2 DZ3 DZ4...
260     DX2Z DXZ2 DXZ2Z DX3Z DXZ3
261
262 % tic
263 DL =Dx2+Dz2;              DL = spfull(DL);
264 DL2 =Dx4+Dz4+2*Dx2*Dz2;  DL2 = spfull(DL2);
265 DxDL =Dx3+Dx*Dz2;        DxDL = spfull(DxDL);
266 DzDL =Dz3+Dz2*Dz3;       DzDL = spfull(DzDL);
267 % t_DLxz = toc; disp(['t_DLxz = ',num2str(t_DLxz)])
268
269 % t_derivadas = toc(t_derivadas_tic); disp(['----- t_derivadas = ',num2str(t_derivadas)])
270 % disp('-----')
271
272 % % Números de condición de las matrices
273 % condicion = {condest(Dx),condest(Dz),condest(DL),condest(DL2);...
274 %             condest(Dx2),condest(Dz2),condest(DxDL),'--';...
275 %             condest(Dx3),condest(Dz3),condest(DzDL),'--'};
276 % disp('Números de condición')
277 % disp(condicion)
278
279 % [uV,sV] = memory;
280 % sV.PhysicalMemory
281 % whos
282 end

```

Listing B.4 Matrices de derivadas con contornos verticales no rectos.

```

1 % Función para calcular las matrices de derivadas 'D', aplicables al
2 % dominio 2D. Argumentos de entrada:
3 % - difX, difZ : 'cheb' para calcular las matrices de derivadas
4 %             unidimensionales 'x' o 'z' de orden dos o superior como
5 %             producto de las de orden 1.
6 % - DXs, DZs : 'prod' o 'derivadas' para calcular las matrices de
7 %             derivadas de orden 2 o mayor, aplicables al dominio 2D
8 %             en variables transformadas, como producto de las de
9 %             primer orden o a partir de las matrices de derivadas
10 %            unidimensionales de mayor orden
11 % - Dxs, Dzs : 'prod' o 'cadena' para calcular las matrices de
12 %             derivadas de orden 2 o mayor, aplicables al dominio 2D
13 %             respecto a las variables originales, aplicando en todas
14 %             la regla de la cadena o solo en la primera y luego
15 %             realizando productos de esta
16 % - Xmat, Zmat : matrices de nodos en el dominio transformado 'X' y 'Z'
17 % - Nx, Nz, Nt : número de nodos en 'x' y 'z', y total
18 % - xl, xr : matrices con contornos izquierdo y derecho
19 % - xch, zch : nodos en 'x' y 'z'
20 % - At, Ab : amplitud de la curva sinusoidal o parabólica del
21 %             contorno derecho o izquierdo
22 % - lambdat, lambdab : longitud de onda de la curva sinusoidal del contorno
23 %             derecho o izquierdo
24 % - geo : 'sin' o 'par' según se quiera que los contornos
25 %             laterales sean sinusoidales o parabólicos
26 % - AR : relación de aspecto
27
28 function [Dx,DX,DX2,DX3,DX4,Dz,Dz2,Dz3,Dz4,DL,DL2,DxDL,DzDL] = ...
29     mat_derivadas_lr(difX,difZ,DXs,DZs,Dxs,Dzs,Xmat,Zmat,Nx,Nz,Nt,...
30     xl,xr,xch,zch,At,Ab,lambdat,lambdab,geo,AR)
31
32 %%% Distribución de los nodos: malla_x y malla_y %%%
33 % 'lin': equiespaciada
34 % 'cheb': de Chebyshev
35
36 %%% Cálculo de las derivadas unidimensionales: difX y difZ %%%
37 % 'cheb' : colocación con Chebyshev
38         switch difX
39         case 'cheb'

```

```

40         DXs = 'prod';
41     end
42     switch difZ
43     case 'cheb'
44         DZs = 'prod';
45     end
46 % 'difx' : dif fin. despreciando ordenes superiores a x (x=2,4,6,8)
47 % (Matriz mal condicionada, malos resultados)
48 % 'Fornberg': especificar número de nodos, NF, para el cálculo de
49 % la derivada en cada nodo (incluyendo este),
50 % y 'm' (1+orden de derivada superior)
51 NF_x = 7;
52 m_x = 7;
53 NF_z = 7;
54 m_z = 7;
55
56 %%% Cálculo de matrices unidimensionales respecto a X y Z: DXs y DZs %%%
57 % 'prod': derivadas orden > 1 como producto de las de primer orden
58 % 'derivadas': calculadas por diferencias finitas
59
60 %%% Cálculo de matrices de derivadas respecto a x y z: Dxs y Dzs %%%
61 % 'prod': por productos de derivadas primeras
62 % 'cadena': aplicando la regla de la cadena
63
64 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
65
66 % tic
67 X = reshape(Xmat,Nt,1);
68 Z = reshape(Zmat,Nt,1);
69 % t_reshape_nodos = toc;disp(['t_reshape_nodos = ',num2str(t_reshape_nodos)])
70
71 % t_derivadas_tic = tic;
72
73 % Derivadas de zb(x) y zt(x) respecto a x
74 % tic
75 switch geo
76 case 'sin'
77     xlaux = Ab*exp(2*pi*1i*(Z/lambdab)); %% (z = Z)
78     xl_z = imag((2*pi*1i)/lambdab*xlaux);
79     xl_zz = imag((2*pi*1i)^2/lambdab*xlaux);
80     xl_zzz = imag((2*pi*1i)^3/lambdab*xlaux);
81     xl_zzzz = imag((2*pi*1i)^4/lambdab*xlaux);
82     xraux = At*exp(2*pi*1i*(Z/lambdat));
83     xr_z = imag((2*pi*1i)/lambdat*xraux);
84     xr_zz = imag((2*pi*1i)^2/lambdat*xraux);
85     xr_zzz = imag((2*pi*1i)^3/lambdat*xraux);
86     xr_zzzz = imag((2*pi*1i)^4/lambdat*xraux);
87 case 'par'
88     xl_z = 4*Ab*(2*Z-1);
89     xl_zz = 8*Ab*ones(size(Z));
90     xl_zzz = 0*Z;
91     xl_zzzz = 0*Z;
92     xr_z = 4*At*(2*Z-1);
93     xr_zz = 8*At*ones(size(Z));
94     xr_zzz = 0*Z;
95     xr_zzzz = 0*Z;
96 end
97
98 % t_zb_xs = toc; disp(['t_zb_xs = ',num2str(t_zb_xs)])
99 clear Z
100
101 % Derivadas de Z(x)=(z-zb(x))/(zt(x)-zb(x)) = (z-zb(x))/p(x) respecto a x
102 % tic
103 p = xr-xl;
104 p_z = xr_z-xl_z;
105 p_zz = xr_zz-xl_zz;
106 p_zzz = xr_zzz-xl_zzz;
107 p_zzzz = xr_zzzz-xl_zzzz;
108
109 X_x = AR./p;
110 X_Z = -(X/AR.*p_z+xl_z)./p*AR;
111 X_ZZ = -(X_Z/AR.*p_z+X/AR.*p_zz+xl_zz)./p*AR+p_z.*(X/AR.*p_z+xl_z)./(p.^2)*AR;
112 X_ZZZ = -(X_ZZ/AR.*p_z+2*X_Z/AR.*p_zz+X/AR.*p_zzz+xl_zzz)./p*AR+...
113 (p_zz.*(X/AR.*p_z+xl_z)+2*p_z.*(X_Z/AR.*p_z+X/AR.*p_zz+xl_zz))./(p.^2)*AR+...
114 (p_z.^2).*(X/AR.*p_z+xl_z)./(p.^3)*AR;
115 X_ZZZZ = -(X_ZZZ/AR.*p_z+3*X_ZZ/AR.*p_zz+3*X_Z/AR.*p_zzz+X/AR.*p_zzzz+xl_zzzz)./p*AR+...
116 (p_zzz.*(X/AR.*p_z+xl_z)+3*p_zz.*(X_Z/AR.*p_z+X/AR.*p_zz+xl_zz)+...
117 p_z.*(X_ZZ/AR.*p_z+2*X_Z/AR.*p_zz+X/AR.*p_zzz+xl_zzz))./(p.^2)*AR+...
118 (4*p_z.*p_zz.*(X/AR.*p_z+xl_z)+...
119 5*(p_z.^2).*(X_Z/AR.*p_z+X/AR.*p_zz+xl_zz))./(p.^3)*AR+...
120 3*(p_z.^3).*(X/AR.*p_z+xl_z)./(p.^4)*AR;
121
122 clear p p_z p_zz p_zzz p_zzzz xlaux xl_z xl_zz xl_zzz xl_zzzz ...
123 xraux xr_z xr_zz xr_zzz xr_zzzz
124 % t_Z_xs = toc; disp(['t_Z_xs = ',num2str(t_Z_xs)])
125
126 % Matrices de derivadas de las variables normalizadas
127 % tic
128 switch difX
129 case 'dif2'
130     [LX,LX2] = difs2(xch);
131 case 'dif4'
132     [LX,LX2,LX3,LX4] = difs4(xch);
133 case 'dif6'
134     [LX,LX2,LX3,LX4] = difs6(xch);
135 case 'dif8'

```

```

136 [LX,LX2,LX3,LX4] = difs8(xch);
137 case 'Fornberg'
138 [~,LX,LX2,LX3,LX4] = DF_Fornberg(xch,NF_x,m_x);
139 case 'MEsp'
140 [LX] = dCheby(xch,Nx);
141 end
142 % t_LXs = toc; disp(['t_LXs = ',num2str(t_LXs)])
143
144 % tic
145 switch DXs
146 case 'prod'
147 DX = kron(LX,speye(Nz));
148 DX2 = DX*DX;
149 DX3 = DX*DX2;
150 DX4 = DX*DX3;
151 case 'derivadas'
152 DX = kron(LX,speye(Nz));
153 DX2 = kron(LX2,speye(Nz));
154 DX3 = kron(LX3,speye(Nz));
155 DX4 = kron(LX4,speye(Nz));
156 end
157 DX = spfull(DX); DX2 = spfull(DX2); DX3 = spfull(DX3); DX4 = spfull(DX4);
158 clear LX LX2 LX3 LX4
159 % t_DXs = toc; disp(['t_DXs = ',num2str(t_DXs)])
160
161 % tic
162 switch difZ
163 case 'dif2'
164 [LZ,LZ2] = difs2(zch);
165 case 'dif4'
166 [LZ,LZ2,LZ3,LZ4] = difs4(zch);
167 case 'dif6'
168 [LZ,LZ2,LZ3,LZ4] = difs6(zch);
169 case 'dif8'
170 [LZ,LZ2,LZ3,LZ4] = difs8(zch);
171 case 'Fornberg'
172 [~,LZ,LZ2,LZ3,LZ4] = DF_Fornberg(zch,NF_z,m_z);
173 case 'MEsp'
174 [LZ] = dCheby(zch,Nz);
175 end
176 % t_LZs = toc; disp(['t_LZs = ',num2str(t_LZs)])
177
178 % tic
179 switch DZs
180 case 'prod'
181 DZ = kron(speye(Nx),LZ);
182 DZ2 = DZ*DZ;
183 DZ3 = DZ*DZ2;
184 DZ4 = DZ*DZ3;
185 case 'derivadas'
186 DZ = kron(speye(Nx),LZ);
187 DZ2 = kron(speye(Nx),LZ2);
188 DZ3 = kron(speye(Nx),LZ3);
189 DZ4 = kron(speye(Nx),LZ4);
190 end
191 DZ = spfull(DZ); DZ2 = spfull(DZ2); DZ3 = spfull(DZ3); DZ4 = spfull(DZ4);
192 % t_DZs = toc; disp(['t_DZs = ',num2str(t_DZs)])
193 clear LZ LZ2 LZ3 LZ4
194
195 % Matrices de derivadas de las variables reales
196 % tic
197 Dx = bsxfun(@times,X_x,DX);
198 switch Dxs
199 case 'prod'
200 Dx2 = Dx*Dx;
201 Dx3 = Dx*Dx2;
202 Dx4 = Dx*Dx3;
203 case 'cadena'
204 Dx2 = bsxfun(@times,X_x.^2,DX2);
205 Dx3 = bsxfun(@times,X_x.^3,DX3);
206 Dx4 = bsxfun(@times,X_x.^4,DX4);
207 end
208 Dx = spfull(Dx); Dx2 = spfull(Dx2); Dx3 = spfull(Dx3); Dx4 = spfull(Dx4);
209 % t_Dxs = toc; disp(['t_Dxs = ',num2str(t_Dxs)])
210
211 % tic
212 Dz = DZ+bsxfun(@times,X_Z,DX); Dz = spfull(Dz);
213 switch Dzs
214 case 'prod'
215 Dz2 = Dz*Dz; Dz2 = spfull(Dz2);
216 Dz3 = Dz*Dz2; Dz3 = spfull(Dz3);
217 Dz4 = Dz*Dz3; Dz4 = spfull(Dz4);
218 case 'cadena'
219 DXZ = DX*DZ; DXZ = spfull(DXZ);
220 DX2Z = DX*DXZ; DX2Z = spfull(DX2Z);
221 DXZ2 = DZ*DXZ; DXZ2 = spfull(DXZ2);
222 DX2Z2 = DZ*DX2Z; DX2Z2 = spfull(DX2Z2);
223 DX3Z = DX*DX2Z; DX3Z = spfull(DX3Z);
224 DXZ3 = DZ*DXZ2; DXZ3 = spfull(DXZ3);
225
226 Dz2 = DZ2+2*bsxfun(@times,X_Z,DXZ)+...
227 bsxfun(@times,X_Z.^2,DX2)+bsxfun(@times,X_ZZ,DX);
228 % Dz2 = Dz*Dz;
229 Dz2 = spfull(Dz2);
230 %
231 Dz3 = DZ3+bsxfun(@times,3*X_Z,DXZ2)+...

```

```

232     bsxfun(@times,3*(X_Z.^2),DX2Z)+bsxfun(@times,X_Z.^3,DX3)+...
233     bsxfun(@times,3*X_ZZ.*X_Z,DX2)+bsxfun(@times,X_ZZZ,DX)+...
234     bsxfun(@times,3*X_ZZ,DXZ);
235     % Dz3 = Dz*Dz2;
236     Dz3 = spfull(Dz3);
237
238     Dz4 = Dz4+bsxfun(@times,4*X_Z,DXZ3)+...
239     bsxfun(@times,6*(X_Z.^2),DX2Z2)+...
240     bsxfun(@times,4*(X_Z.^3),DX3Z)+bsxfun(@times,X_Z.^4,DX4)+...
241     bsxfun(@times,6*X_ZZ,DXZ2)+bsxfun(@times,12*X_ZZ.*X_Z,DX2Z)+...
242     bsxfun(@times,6*X_ZZ.*(X_Z.^2),DX3)+bsxfun(@times,4*X_ZZZ,DXZ)+...
243     bsxfun(@times,4*X_ZZZ.*X_Z+3.*(X_ZZ.^2),DX2)+bsxfun(@times,X_ZZZZ,DX);
244     % Dz4 = Dz*Dz3;
245     Dz4 = spfull(Dz4);
246
247 end
248 % t_Dxs = toc; disp(['t_Dxs = ',num2str(t_Dxs)])
249 clear X_Z X_ZZ X_ZZZ X_ZZZZ DX2 DX3 DX4 DZ DZ2 DZ3 DZ4...
250       DX2Z DXZ2 DXZ2Z DX3Z DXZ3
251
252 % tic
253 DL = Dx2+Dz2;           DL = spfull(DL);
254 DL2 = Dx4+Dz4+2*Dx2*Dz2; DL2 = spfull(DL2);
255 DxDL = Dx3+Dx*Dz2;     DxDL = spfull(DxDL);
256 DzDL = Dz*Dx2+Dz3;     DzDL = spfull(DzDL);
257 % t_DLxz = toc; disp(['t_DLxz = ',num2str(t_DLxz)])
258
259 % t_derivadas = toc(t_derivadas_tic); disp(['----- t_derivadas = ',num2str(t_derivadas)])
260 % disp('-----')
261
262 % % Números de condición de las matrices
263 % condicion = {condest(Dx),condest(Dz),condest(DL),condest(DL2);...
264 %             condest(Dx2),condest(Dz2),condest(DxDL),'-';...
265 %             condest(Dx3),condest(Dz3),condest(DzDL),'-'};
266 % disp('Números de condición')
267 % disp(condicion)
268
269 % [uV,sV] = memory;
270 % sV.PhysicalMemory
271 % whos
272 end

```

### Listing B.5 Derivadas con algoritmo de Fornberg ([20]).

```

1  % Función para calcular las matrices de derivadas unidimensionales 'L'
2  % según el algoritmo de Fornberg. Los argumentos de entrada son:
3  % - xch: nodos del dominio
4  % - NF : número de nodos incluidos en el cálculo de la derivada en un
5  %       nodo (incluyendo este). Orden de precisión para la derivada
6  %       de orden k: será (NF-1-k)
7  % - m : 1 + orden de derivada superior deseado
8  function [D0,D1,D2,D3,D4] = DF_Fornberg(xch,NF,m)
9  N = length(xch);
10 Nvi = floor(NF/2);
11 Nvd = ceil(NF/2);
12 % Algoritmo recursivo (A Practical Guide to Pseudospectral Methods (1996)
13 % Página 178 (Algoritmo para el cálculo de derivadas en los nodos)
14 % Página 18 (Algoritmo genérico rápido)
15 D0 = zeros(N,N);
16 D1 = zeros(N,N);
17 D2 = zeros(N,N);
18 D3 = zeros(N,N);
19 D4 = zeros(N,N);
20 for u = 1:N
21     % Selección de nodos para el cálculo de las derivadas en x(i)
22     if u<=Nvi
23         t = 1:NF;
24     elseif u > (N-Nvd)
25         t = (N-NF+1):N;
26     else
27         t = (u-Nvi):(u+Nvd-1);
28     end
29     c = zeros(NF,NF,m);
30     % Los subíndices empiezan en 1, no en cero
31     alpha = 1;
32     c(1,1,1) = 1;
33     xI = xch(u);
34     x = xch(t);
35     ximI = (x(1)-xI);
36     % Fornberg, 1988 American Mathematical Society, página 700
37     for ii = 2:(NF) % i = 1 solo conlleva c(1,1,1) resuelto ya
38         beta = 1;
39         MN = min(ii,m);
40         C5 = ximI;
41         ximI = (x(ii)-xI);
42         for j = 1:(ii-1) % Triangular inferior
43             ximj = (x(ii)-x(j));
44             beta = beta*ximj;
45             c(j,ii,1) = ximI*c(j,ii-1,1)/ximj;
46             for k = 2:MN
47                 c(j,ii,k) = (ximI*c(j,ii-1,k)-(k-1)*c(j,ii-1,k-1))/ximj;
48             end

```

```

49     end
50     c(ii,ii,1) = -alpha*05*c(ii-1,ii-1,1)/beta;
51     for k = 2:MN
52         c(ii,ii,k) = ...
53             alpha*((k-1)*c(ii-1,ii-1,k-1)-05*c(ii-1,ii-1,k))/beta;
54     end
55     alpha = beta;
56     end
57     D0(u,t) = c(:,MN,1)';
58     D1(u,t) = c(:,MN,2)';
59     D2(u,t) = c(:,MN,3)';
60     D3(u,t) = c(:,MN,4)';
61     D4(u,t) = c(:,MN,5)';
62     end
63 end

```

### Listing B.6 Derivadas con diferencias finitas.

```

1  % Función para calcular las matrices de derivadas unidimensionales 'L'
2  % usando únicamente series de Taylor, despreciando términos de orden
3  % superior a 8.
4  % El único argumento de entrada es el vector de nodos xch
5  function [Dx, Dx2, Dx3, Dx4, Dx5, Dx6, Dx7, Dx8] = difs8(xch)
6
7  N = length(xch);
8
9  % Número total de puntos involucrados para cada derivada.
10 % Dado que se busca un sistema cuadrado Nt debe ser igual al mayor orden de
11 % derivada que se busca más 1.
12 Nt = 9;
13 % Número de puntos vecinos que se usan para el cálculo referido a cada
14 % punto
15 Nv = Nt-1;
16 % Número de puntos a la izquierda
17 Nvi = Nv/2;
18 % Número de puntos a la derecha
19 Nvd = Nv-Nvi;
20
21 % kv será el vector de nodos contiguos que se usarán para calcular la
22 % derivada en cada punto. Por tanto tendrá Nv componentes. En el caso de
23 % que el punto este en los extremos, los puntos adyacentes estarán a ambos
24 % lados, pudiendo estar todos a la derecha o a la izquierda.
25
26 % Trabajar con una matriz dispersa asignando valores a diferentes
27 % posiciones una vez creada no es eficiente. Se hará creando la matriz una
28 % vez se tienen todos los valores con sus respectivos índices.
29 for i = 1:N
30     % Habrá que distinguir según la fila de la matriz, de manera que queden
31     % Nvi y Nvd a cada lado, o el punto donde se está calculando esté cerca
32     % de los límites. En estos casos habrá que coger los que haya a
33     % izquierda o derecha (kvi o kvd puntos) y completar por el lado
34     % contrario (Nv-kvi y Nv-kvd).
35     if i<=Nvi
36
37         kv = [1:(i-1) (i+1):Nt];
38
39     elseif i >= (N-Nvd)
40
41         kv = [(N-Nt+1):(i-1) (i+1):N];
42
43     else
44
45         kv = [(i-Nvi):(i-1) (i+1):(i+Nvd)];
46
47     end
48
49 % Se evalúan los puntos x. Queda una matriz donde cada fila i
50 % contiene los puntos que se usan para el cálculo del nodo i.
51 xk = xch(kv);
52 h = xk-xch(i);
53 index_iaux = repmat((1+(i-1)*Nv):(1+(i-1)*Nv+Nv-1),Nv,1);
54 index_i(1,(1+(i-1)*Nv^2):(1+(i-1)*Nv^2+Nv^2-1)) = index_iaux(:)';
55 index_j(1,(1+(i-1)*Nv^2):(1+(i-1)*Nv^2+Nv^2-1)) = ...
56     repmat((1+(i-1)*Nv):(1+(i-1)*Nv+Nv-1),1,Nv);
57 Haux1 = [h' h'.^2/2 h'.^3/6 h'.^4/24 h'.^5/120 h'.^6/720 ...
58     h'.^7/5040 h'.^8/40320];
59 % Se convierte esta matriz a un vector para poder crear la matriz
60 % sparse con los índices.
61 Haux1 = Haux1';
62 Haux2(1,(1+(i-1)*Nv^2):(1+(i-1)*Nv^2+Nv^2-1)) = Haux1(:)';
63
64 % % Guardamos los valores de h, (f(x+h)-f(x))
65 % % hi((1+(i-1)*Nv):(1+(i-1)*Nv+Nv-1),1) = h';
66
67 % Se busca una matriz que sea función únicamente del valor de la función en
68 % los nodos. Para ello, hay que expresar diff en función de estos.
69 index_ii(1,(1+(i-1)*2*Nv):(1+(i-1)*2*Nv+2*Nv-1)) = ...
70     repmat((1+(i-1)*Nv):(1+(i-1)*Nv+Nv-1),1,2);
71 index_jj(1,(1+(i-1)*2*Nv):(1+(i-1)*2*Nv+2*Nv-1)) = [i*ones(1,Nv) kv];
72 M_diff_faux(1,(1+(i-1)*2*Nv):(1+(i-1)*2*Nv+2*Nv-1)) = [-ones(1,Nv) ones(1,Nv)];
73
74 end

```

```

75
76 % Se crea H y se calcula su inversa, siendo H la matriz tal que:
77 % [f11 f21 f31 f41 f21 f22 f32 f42 ... f1N f2N f3N f4N]' =
78 % = H [f11 f111 f1111 f1V1...f1N f1IN f1IIN f1VN]'
79 H = sparse(index_i,index_j,Haux2);
80 M_diff_f = sparse(index_ii,index_jj,M_diff_faux);
81
82 % Queda, diff = M_diff_f*f = H*df. Por tanto:
83 % df = H\M_diff_f*f
84
85 DD = H\M_diff_f;
86
87 % % Para intentar solucionar problemas de condicionamiento
88 % % Caso 1: con pinv y sumando una constante a la diagonal de la
89 % % matriz,cuanto mayor es esta, menos realista es la solución pero mas
90 % % estable
91 % % DD = pinv(full(H+0.001*speye(size(H))*max(max(H))))*M_diff_f;
92 % % Caso 2: bicg(A,b,tol,maxit) (b debe ser un vector, en este caso no vale)
93 % % DD2 = H\M_diff_f;
94 % % tol=1e-6;
95 % % maxit=10000;
96 % % DD2 = zeros(N*(Nt-1),N);
97 % % for kk = 1:length(N)
98 % % DD(:,kk) = bicg(H,M_diff_f(:,kk),tol,maxit);
99 % % end
100 % % max(max(DD2-DD))
101 % % Caso 3: X = lsqminnorm(A,B) Solo esta en matlab 2017b
102 % % DD = lsqminnorm(H,M_diff_f);
103 % % Caso 4: mldivide(A) Es lo mismo que A\ pero usa algoritmos diferentes si
104 % % es dispersa por ejemplo
105 % % DD = mldivide(H,M_diff_f);
106
107 % rcond mas eficiente pero menos preciso que cond(es una estimacion)
108 rcond_difs = rcond(full(H))
109
110
111 % Reordenando la matriz se obtiene Dx, Dx2, Dx3 y Dx4
112 Dx = DD(1:Nv:end,:);
113 Dx2 = DD(2:Nv:end,:);
114 Dx3 = DD(3:Nv:end,:);
115 Dx4 = DD(4:Nv:end,:);
116 Dx5 = DD(5:Nv:end,:);
117 Dx6 = DD(6:Nv:end,:);
118 Dx7 = DD(7:Nv:end,:);
119 Dx8 = DD(8:Nv:end,:);
120
121 end

```

Listing B.7 Derivadas con método de colocación con nodos de Chebyshev.

```

1 % Función para calcular la matriz de derivadas unidimensionales 'L' usando
2 % el método de colocación con todos los nodos de Chebyshev del dominio xch
3 % Se ha reducido el número de bucles consiguiendo reducir el tiempo de
4 % ejecución de manera considerable a menos que se trabaje en paralelo,
5 % en cuyo caso el tiempo de ejecución en ambos casos es del mismo
6 % orden.
7 function [Lp]=dCheby(xch)
8 Np = length(xch);
9 Lp = NaN(Np);
10 t_Denp_bloque = 0;
11 % t_Denp_bucle = 0;
12 % t_sumk_bucle = 0;
13 t_sumk_bloque = 0;
14 for m=1:Np
15 %% Método 1: Por bloques
16 t_Denp_bloque_tic = tic;
17 Denp = prod(xch(m)-xch(xch~=xch(m)));
18 t_Denp_bloque = t_Denp_bloque+toc(t_Denp_bloque_tic);
19 t_sumk_bloque_tic = tic;
20 ind = 1:Np;
21 for k = 1:Np
22 sumk = 0;
23 for u = 1:Np
24 if u ~= m
25 indice = ((ind ~= m) & (ind ~= u));
26 produ = prod(xch(k)-xch(indice));
27 sumk = sumk+produ;
28 end
29 end
30 Lp(k,m)=sumk/Denp;
31 end
32 t_sumk_bloque = t_sumk_bloque+toc(t_sumk_bloque_tic);
33 %% Método 2: Con bucles
34 % % t_Denp_bucle_tic = tic;
35 % % Denp = 1;
36 % % for j = 1:Np
37 % % if j ~= m
38 % % Denp = Denp*(xp(m)-xp(j));
39 % % end
40 % % end
41 % % t_Denp_bucle = t_Denp_bucle+toc(t_Denp_bucle_tic);
42 % % t_sumk_bucle_tic = tic;

```





```

63         difRa = difRa1;
64         else
65             difRa = difRa2;
66         end
67         Ra = Ra + difRa;
68     end
69     Racinf = Ra-2*difRa;
70     Racsup = Ra-difRa;
71     var_Rac2 = 0;
72     while var_Rac2 == 0
73         Ra = (Racinf+Racsup)/2;
74         [var_Rac,flag1,flag2] = calculo_estab(Nx,Nz,Pr,H,AR,At,Ab,...
75             lambdat,lambdab,Ra,malla_x,malla_z,difX,difZ,Dxs,...
76             Dzs,Dxs,Dzs);
77         if flag1 == 1 || flag2 == 1
78             flag2(ii,jj,k) = 1;
79             disp('----2----')
80         end
81         if var_Rac == 1
82             Racsup = Ra;
83         else
84             Racinf = Ra;
85         end
86         if (Racsup-Racinf)<1
87             var_Rac2 = 1;
88         end
89     end
90     Rac(ii,jj,k) = (Racsup+Racinf)/2;
91     end %%%%%%%%%%%
92 end
93 end
94 disp(k)
95 end
96 t_Rac_malla = toc
97 save('DF_topbotwall_latT0_y_tau0')
98 %%
99 %%% Figuras separadas Nxv cte
100 cont = 0;
101 linea = {'-', 'o', 'x', 'v', 'k', 'x', 'v', 'k'};
102 sh = {'-', 'o', 'x', 'v', 'k', 'x', 'v', 'k'};
103 colores1 = cool(length(Nzv));
104 for ii = 1:length(Nxv)
105     leyenda = {};
106     figure(100+ii), xlabel('AR [-]'), ylabel('Ra_c [-]'), hold on
107     cont = cont+1;
108     for jj = 1:length(Nzv)
109         Ra_AR(:,1) = Rac(ii,jj,:);
110         plot(ARv,Ra_AR,[sh{jj} linea{jj}], 'color', colores1(jj,:))
111         leyenda = [leyenda; 'N_z= ', num2str(Nzv(jj))];
112     end
113     plot(ARv,1708*ones(length(ARv),1), 'k', 'linewidth', 0.1)
114     leyenda = [leyenda; 'Ra_c = 1708'];
115     legend(leyenda)
116     title(['N_x= ', num2str(Nxv(ii))])
117     figure(100+ii), print('-depsc2', ['Nx', num2str(Nxv(ii)), '_1708_DF'], '.eps')
118     saveas (figure(100+ii), ['Nx', num2str(Nxv(ii)), '_1708_DF'], '.fig')
119 end
120
121 %%% Figuras separadas Nz cte
122 cont = 0;
123 linea = {'-', 'o', 'x', 'v', 'k', 'x', 'v', 'k', 'x', 'v', 'k'};
124 sh = {'-', 'o', 'x', 'v', 'k', 'x', 'v', 'k', 'x', 'v', 'k'};
125 colores1 = cool(length(Nxv));
126 for jj = 1:length(Nzv)
127     leyenda = {};
128     figure(200+jj), xlabel('AR [-]'), ylabel('Ra_c [-]'), hold on
129     cont = cont+1;
130     for ii = 1:length(Nxv)
131         Ra_AR(:,1) = Rac(ii,jj,:);
132         plot(ARv,Ra_AR,[sh{ii} linea{ii}], 'color', colores1(ii,:))
133         leyenda = [leyenda; 'N_x= ', num2str(Nxv(ii))];
134     end
135     plot(ARv,1708*ones(length(ARv),1), 'k', 'linewidth', 0.1)
136     leyenda = [leyenda; 'Ra_c = 1708'];
137     legend(leyenda)
138     title(['N_z= ', num2str(Nzv(jj))])
139     figure(200+jj), print('-depsc2', ['Nz', num2str(Nzv(jj)), '_1708_DF'], '.eps')
140     saveas (figure(200+jj), ['Nz', num2str(Nzv(jj)), '_1708_DF'], '.fig')
141 end
142
143 %%% Polinomio interpolador
144 % clear all
145 % Nzs = 30;
146 % Nxs = [80 40 30 30 30 30 30];
147 % ARv = [40 20 10 5 3 2 1 0];
148 % p = polyfit(ARv,Nxs,2);
149 % ARs = linspace(1,40,100);
150 % NxMEsp = ceil(polyval(p,ARs));
151 % figure(300)
152 % plot(ARs,NxMEsp, '-')
153 % xlabel('AR [-]'), ylabel('N_x')

```

Listing B.9 Función análisis lineal asociada a B.8.

```

1  % Función que determina si hay autovalores con parte real mayor que uno,
2  % ignorando aquellos que son unidad debido a la forma de imponer las
3  % condiciones de contorno.
4  function [var_Rac,flag1,flag2] = calculo_estab(Nx,Nz,Pr,H,AR,At,Ab,...
5  lambdat,lambdab,Ra,malla_x,malla_z,difX,difZ,Dxs,Dzs,Dxs,Dzs)
6  %%% Representación del dominio y la malla (1 es sí) %%%
7  fig_dominio = 0;
8  fig_malla = 0;
9  %%% Distribución de los nodos: malla_x y malla_z %%%
10 % 'lin': equiespaciada
11 % 'cheb': de Chebyshev
12 %%% Cálculo de las derivadas unidimensionales: difX y difZ %%%
13 % 'cheb': colocación con Chebyshev
14     switch difX
15     case 'cheb'
16         Dxs = 'prod';
17     end
18     switch difZ
19     case 'cheb'
20         Dzs = 'prod';
21     end
22 % 'difx' : dif fin. despreciando ordenes superiores a x (x=2,4,6,8)
23 % (Matriz mal condicionada, malos resultados)
24 % 'Ponberg': especificar número de nodos, NF, para el cálculo de
25 % la derivada en cada nodo (incluyendo este),
26 % y 'm' (1+orden de derivada superior)
27     NF_x = 5; % El orden de precisión es NF-1.
28     m_x = 5;
29     NF_z = 5;
30     m_z = 5;
31
32 %%% Cálculo de matrices unidimensionales respecto a X y Z: Dxs y Dzs %%%
33 % 'prod': derivadas orden > 1 como producto de las de primer orden
34 % 'derivadas': calculadas por diferencias finitas
35
36 %%% Cálculo de matrices de derivadas respecto a x y z: Dxs y Dzs %%%
37 % 'prod': por productos de derivadas primeras
38 % 'cadena': aplicando la regla de la cadena
39
40 % tic
41 % Geometría del dominio
42 [xch,zch,Xmat,Zmat,~,~,Dt,~,~,zb,zt] = geom(...
43     Nx,Nz,malla_x,malla_z,fig_dominio,fig_malla,H,AR,At,Ab,lambdat,lambdab);
44
45 % Derivadas
46 [Dx,Dx2,~,~,Dz,~,~,DL,DL2,~,~] = mat_derivadas_topbot(...
47     difX,difZ,Dxs,Dzs,Dxs,Dzs,NF_x,NF_z,m_x,m_z,Xmat,Zmat,Nx,Nz, Dt, zb, zt,...
48     xch,zch,At,Ab,lambdat,lambdab);
49
50 % Contornos y subcontornos
51 Id = speye(Nt);
52 c_sup = (Nz:Nz:(Nx*Nz))'; % c. Superior
53 c_inf = (1:Nz:(Nx*(Nx-1)+1))'; % c. Inferior
54 c_der = ((Nx-1)*Nz+1):(Nx*Nz))'; % c. Derecho
55 c_izq = (1:Nz)'; % c. Izquierdo
56 c_M_h = [c_inf;c_sup]; % c. Horizontales
57 c_M_v = [c_izq;c_der]; % c. Verticales
58 c_M = [c_inf;c_sup;c_izq;c_der]; % Matriz contornos
59 sub_c_sup = ((2*Nz-1):Nz:((Nx-1)*Nz-1))'; % Subc. Superior
60 sub_c_inf = ((Nz+2):Nz:(Nx*(Nx-2)+2))'; % Subc. Inferior
61 sub_c_izq = ((Nz+2):(2*Nz-1))'; % Subc. Izquierdo
62 sub_c_der = ((Nz*(Nx-2)+2):(Nz*(Nx-1)-1))'; % Subc. Derecho
63 sub_c_h = [sub_c_inf;sub_c_sup]; % Subc. Horizontales
64 sub_c_v = [sub_c_izq;sub_c_der]; % Subc. Verticales
65 sub_c_M = [sub_c_inf;sub_c_sup;sub_c_izq;sub_c_der]; % Matriz subcontornos
66
67 % Sistema matricial (gravedad en dirección '-z')
68 % Psi = Psi0*e^(st)
69 % T = T0*e^(st)
70 % x = [Psi;T]
71 % Px = sQx
72
73 %%% Variables %%%
74
75 % T: desviación de temperatura respecto al equilibrio
76 % Psi: desviación de la función de corriente respecto al equilibrio
77
78 %%% Ecuaciones adimensionales %%%
79
80 % P = [ P_APSI P_AT
81 % P_BPSI P_BT ]
82 % Q = [ Q_APSI Q_AT [ DL 0
83 % Q_BPSI Q_BT ] = 0 1 ]
84
85 P_APSI = Pr*DL2; P_APSI = spfull(P_APSI);
86 P_AT = -Pr*Ra*Dx; P_AT = spfull(P_AT);
87 P_BPSI = -Dx; P_BPSI = spfull(P_BPSI);
88 P_BT = DL; P_BT = spfull(P_BT);
89
90 Q_APSI = DL; Q_APSI = spfull(Q_APSI);
91 Q_AT = zeros(Nt); Q_AT = spfull(Q_AT);
92 Q_BPSI = zeros(Nt); Q_BPSI = spfull(Q_BPSI);
93 Q_BT = Id; Q_BT = spfull(Q_BT);
94
95 %%% Condiciones de contorno %%%
96 % T impuesta (nula) en C. Verticales y Horizontales

```

```

97     P_BT(c_M,:) = Id(c_M,:);
98     P_BPSI(c_M,:) = 0;
99     Q_BT(c_M,:) = Id(c_M,:);
100    Q_BPSI(c_M,:) = 0;
101    % Psi impuesto (nula) en C. Verticales y Horizontales
102    P_APSI(c_M,:) = Id(c_M,:);
103    P_AT(c_M,:) = 0;
104    Q_APSI(c_M,:) = Id(c_M,:);
105    Q_AT(c_M,:) = 0;
106    % Condición común a los SUBcontornos
107    P_AT(sub_c_M,:) = 0;
108    Q_AT(sub_c_M,:) = 0;
109    % Velocidades normales nulas en SUBcontornos superior e inferior
110    P_APSI(sub_c_sup,:) = Dz(sub_c_sup+1,:);
111    P_APSI(sub_c_inf,:) = Dz(sub_c_inf-1,:);
112    Q_APSI(sub_c_sup,:) = Dz(sub_c_sup+1,:);
113    Q_APSI(sub_c_inf,:) = Dz(sub_c_inf-1,:);
114    % Esfuerzo cortante en SUBcontornos izquierdo y derecho nulo
115    P_APSI(sub_c_izq,:) = Dx2(sub_c_izq-Nz,:);
116    P_APSI(sub_c_der,:) = Dx2(sub_c_der+Nz,:);
117    Q_APSI(sub_c_izq,:) = Dx2(sub_c_izq-Nz,:);
118    Q_APSI(sub_c_der,:) = Dx2(sub_c_der+Nz,:);
119    % Dispensa o llena
120    P_APSI = spfull(P_APSI); P_AT = spfull(P_AT);
121    P_BPSI = spfull(P_BPSI); P_BT = spfull(P_BT);
122
123    Q_APSI = spfull(Q_APSI); Q_AT = spfull(Q_AT);
124    Q_BPSI = spfull(Q_BPSI); Q_BT = spfull(Q_BT);
125
126    %%% Construcción de las matrices del problema %%%
127    Asyst = [P_APSI P_AT; P_BPSI P_BT]; Asyst = spfull(Asyst);
128    Qsyst = [Q_APSI Q_AT; Q_BPSI Q_BT]; Qsyst = spfull(Qsyst);
129
130    % toc
131    %% Autovalores
132    %%% Se pueden calcular hasta 200 autovalores como máximo a la vez.
133    %%% Por tanto, para evitar el caso en que un autovalor pase a ser mayor que
134    %%% 1 y existan más de 200 con valor idénticamente uno, se calculan dos
135    %%% veces:
136    %%% - Autovalores próximos a cero, que aseguren encontrar aquellos que
137    %%% se encuentren entre 0 y 1.
138    %%% - Autovalores próximos a 2 (o cualquier número mayor que 1), que
139    %%% aseguren encontrar cualquier autovalor mayor que 1.
140    toleig = 1e-2;
141    % tic
142    [~,D1,flag1] = eigs(Asyst,Qsyst,1,(0+(1-toleig))/2,'tol',1e-14,'FailureTreatment','drop');
143    s = 0;
144    while isempty(diag(D1)) && s <= 100
145        s = s+1;
146        [~,D1,flag1] = eigs(Asyst,Qsyst,1,(0+(1-toleig))/2+s/100,'tol',1e-14,'FailureTreatment','drop');
147        if s >= 100
148            flag1 = 1;
149            D1 = 1e16;
150        end
151    end
152    % toc
153    % tic
154    % Es mas rapido buscar un valor que el máximo autovalor. Se supone que el
155    % máximo no estará por encima de 9, de manera que si lo fuese no se
156    % obtendría ya que se obtendría un uno
157    [~,D2,flag2] = eigs(Asyst,Qsyst,1,5,'tol',1e-14,'FailureTreatment','drop');
158    s = 5;
159    while isempty(diag(D2)) && s < 100
160        s = s+1;
161        [~,D2,flag2] = eigs(Asyst,Qsyst,1,s,'tol',1e-14,'FailureTreatment','drop');
162        if s >= 100
163            flag2 = 1;
164            D2 = 1e16;
165        end
166    end
167    % toc
168    av1 = diag(D1);
169    av2 = diag(D2);
170    % Si suman 1 (true) es porque hay algun autovalor con parte real positiva
171    % sin tener en cuenta los que estan en torno a 1
172
173    if sum(real(av1) > 0 & (real(av1) <= (1-toleig) | real(av1) >= (1+toleig))) || sum(av2>1+toleig) > 0
174        var_Rac = 1;
175    else
176        var_Rac = 0;
177    end
178
179    end

```

Listing B.10 Estudio lineal para condiciones laterales periódicas y contorno superior libre.

```

1 clear all, close all, clc
2 tic
3 %% Dominio rectangular con gravedad en dirección "-z"
4 % Valores constantes
5 H = 1; % Altura (m)
6 alpha = 0.2216e-4; % Difusividad térmica (m^2/s)

```

```

7 nu = 1.684e-5; % Viscosidad cinemática (m^2/s)
8 Pr = nu/alpha; % Numero de Prandtl
9 g = 9.81; % Gravedad (m^2/s)
10 difRa = 500;
11
12 % Parámetros de la malla
13 malla_x = 'lin'; % 'cheb', 'lin', 'parsim'
14 malla_z = 'lin'; % 'cheb', 'lin', 'parsim'
15 difX = 'Fornberg'; % 'cheb', 'dif2', 'dif4', 'dif6', 'dif8', 'Fornberg'
16 difZ = 'Fornberg'; % 'cheb', 'dif2', 'dif4', 'dif6', 'dif8', 'Fornberg'
17 DXs = 'derivadas'; % 'prod' (por defecto si difZ 'cheb') o 'derivadas'
18 DZs = 'derivadas'; % 'prod' (por defecto si difZ 'cheb') o 'derivadas'
19 Dxs = 'cadena'; % 'prod' o 'cadena'
20 Dzs = 'cadena'; % 'prod' o 'cadena'
21
22 % Variables del bucle
23 ARv = 10:-0.1:1; % Alargamientos del dominio
24 switch difX
25 case 'MEsp'
26 N xv = ceil(18.8227+0.6415*ARv+0.0222*ARv.^2)';
27 Nz = 10;
28 case 'Fornberg'
29 N xv = ceil(27.31+5.64*ARv-0.03*ARv.^2)';
30 Nz = 20;
31 end
32
33 %% Variables auxiliares para el bucle
34 Rac = NaN(length(ARv),1);
35 flag1 = NaN(length(ARv),1);
36 flag2 = NaN(length(ARv),1);
37 V = NaN(2*max(N xv)*Nz,length(ARv));
38 xv = NaN(max(N xv)*Nz,length(ARv));
39 zv = NaN(max(N xv)*Nz,length(ARv));
40 k = 0;
41 At = 0;
42 Ab = 0;
43 lambdat = 1;
44 lambdab = 1;
45 %% Resolución
46 for AR = ARv
47 k = k+1;
48 Nx = N xv(k,1);
49 Nt = Nx*Nz;
50 Ra = 100;
51 var_Rac = 0;
52 while var_Rac == 0
53 [var_Rac,~,f1,f2,~,~] = calculo_estab(Nx,Nz,Pr,H,AR,At,Ab,...
54 lambdat,lambdab,Ra,malla_x,malla_z,difX,difZ,DXs,DZs,Dxs,Dzs);
55
56 if f1 == 1 || f2 == 1
57 flag1(k,1) = 1;
58 disp('----1----')
59 end
60 Ra = Ra + difRa;
61 end
62 Racinf = Ra-2*difRa;
63 Racsup = Ra-difRa;
64 var_Rac2 = 0;
65 while var_Rac2 == 0
66 Ra = (Racinf+Racsup)/2;
67 [var_Rac,Vaux,f1,f2,xv(1:Nt,k),zv(1:Nt,k)] = ...
68 calculo_estab(Nx,Nz,Pr,H,AR,At,Ab,lambdat,lambdab,Ra,...
69 malla_x,malla_z,difX,difZ,DXs,DZs,Dxs,Dzs);
70 if f1 == 1 || f2 == 1
71 flag2(ii,jj,k) = 1;
72 disp('----2----')
73 end
74 if var_Rac == 1
75 Racsup = Ra;
76 V(1:2*Nt,k) = Vaux;
77 else
78 Racinf = Ra;
79 end
80 if (Racsup-Racinf)<0.1
81 var_Rac2 = 1;
82 end
83 end
84 Rac(k,1) = (Racsup+Racinf)/2;
85 disp([num2str(Racinf),'-',num2str(Racsup)])
86 end
87 t_Rac_malla = toc
88 save('topfree_periodic_estab_DF')
89
90 ARplot = 1:4;
91 for AR = ARplot
92 k = find(ARv == AR)
93 Nx = N xv(k);
94 Nt = Nx*Nz;
95 Psimat = reshape(V(1:Nt,k),Nz,Nx);
96 Tmat = reshape(V((Nt+1):(2*Nt),k),Nz,Nx);
97 xmat = reshape(xv(1:Nt,k),Nz,Nx);
98 zmat = reshape(zv(1:Nt,k),Nz,Nx);
99 figure(k)
100 subplot(2,1,1), contour(xmat,zmat,Tmat)
101 ylabel('$\hat{z}$','interpreter','latex')
102 title('Temperatura de perturbación adimensional')

```

```

103 set(gca,'FontSize',12)
104 set(gcf,'renderer','Painters')
105 subplot(2,1,2),contour(xmat,zmat,Psimat)
106 xlabel('\hat{x}', 'interpreter','latex')
107 ylabel('\hat{z}', 'interpreter','latex')
108 title('Función de corriente de perturbación adimensional')
109 set(gca,'FontSize',12)
110 set(gcf,'renderer','Painters')
111 print('-depsc2',['topfree_periodic_estab_AR_',num2str(AR),'.eps'])
112 saveas (figure(k),['topfree_periodic_estab_AR_',num2str(AR),'.fig'])
113 pause(1e-16)
114 end
115
116 figure(100),title('Rayleigh critico')
117 hold on
118 plot(ARv,Rac,'--b')
119 xlabel('AR'),ylabel('Ra_c')
120 legend('Diferencias finitas (n=7)', 'Chebyshev')
121 set(gca,'FontSize',12)
122 set(gcf,'renderer','Painters')
123 print('-depsc2',['topfree_periodic_estab_Rac.eps'])
124 saveas (figure(100), 'topfree_periodic_estab_Rac.fig')

```

### Listing B.11 Función análisis lineal asociada a B.10.

```

1 % Función que determina si hay autovalores con parte real mayor que uno,
2 % ignorando aquellos que son unidad debido a la forma de imponer las
3 % condiciones de contorno. Además, obtiene el autovector asociado a ese
4 % autovalor si existe.
5 function [var_Rac,V,flag1,flag2,xv,zv] = calculo_estab(Nx,Nz,Pr,H,AR,...
6 At,Ab,lambda_t,lambda_b,Ra,malla_x,malla_z,difX,difZ,DXs,DZs,Dxs,Dzs)
7 %%% Representación del dominio y la malla (1 es si) %%%
8 fig_dominio = 0;
9 fig_malla = 0;
10 %%% Cálculo de las derivadas unidimensionales: difX y difZ %%%
11 switch difX
12 case 'cheb'
13 DXs = 'prod';
14 end
15 switch difZ
16 case 'cheb'
17 DZs = 'prod';
18 end
19
20 NF_x = 7; % El orden de precisión es NF-1.
21 m_x = 7;
22 NF_z = 7;
23 m_z = 7;
24 % tic
25 %% Geometría del dominio
26 [xch,zch,Xmat,Zmat,xmat,zmat,Nt,~,~,zb,zt] = geom(...
27 Nx,Nz,malla_x,malla_z,fig_dominio,fig_malla,H,AR,At,Ab,lambda_t,lambda_b);
28 xv = reshape(xmat,Nt,1);
29 zv = reshape(zmat,Nt,1);
30 %% Derivadas
31 [Dx,Dx2,Dx3,~,Dz,Dz2,~,~,DL,DL2,~,~] = mat_derivadas_topbot(difX,difZ,...
32 DXs,DZs,Dxs,Dzs,NF_x,NF_z,m_x,m_z,...
33 Xmat,Zmat,Nx,Nz,Nt,zb,zt,xch,zch,At,Ab,lambda_t,lambda_b);
34
35 %% Contornos y subcontornos
36 Id = speye(Nt);
37 c_sup = (Nz:Nz:(Nx*Nz))'; % c. Superior
38 c_inf = (1:Nz:(Nz*(Nx-1)+1))'; % c. Inferior
39 c_der = ((Nx-1)*Nz+1):(Nx*Nz))'; % c. Derecho
40 c_izq = (1:Nz)'; % c. Izquierdo
41 c_M = [c_inf;c_sup;c_izq;c_der]; % Matriz contornos
42 c_M_h = [c_inf;c_sup];
43 c_M_v = [c_izq;c_der];
44 sub_c_sup = ((2*Nz-1):Nz:(Nx-1)*Nz-1))'; % Subc. Superior
45 sub_c_inf = ((Nz+2):Nz:(Nx-2)+2))'; % Subc. Inferior
46 sub_c_izq = ((Nz+2):(2*Nz-1))'; % Subc. Izquierdo
47 sub_c_der = ((Nz*(Nx-2)+2):(Nz*(Nx-1)-1))'; % Subc. Derecho
48 sub_c_M = [sub_c_inf;sub_c_sup;sub_c_izq;sub_c_der]; % Matriz subcontornos
49 sub_c_h = [sub_c_inf;sub_c_sup];
50
51 %% Sistema matricial (gravedad en dirección '-z')
52 % Psi = Psi0*e^(st)
53 % T = T0*e^(st)
54 % x = [Psi;T]
55 % Px = sQx
56
57 %%% Variables %%%
58
59 % T: desviación de temperatura respecto al equilibrio
60 % Psi: desviación de la función de corriente respecto al equilibrio
61
62 %%% Ecuaciones adimensionales %%%
63
64 % P = [ P_APSI P_AT
65 % P_BPSI P_BT ]
66 % Q = [ Q_APSI Q_AT [ DL 0
67 % Q_BPSI Q_BT ] = 0 1 ]

```

```

68
69 P_APSI = Pr*DL2; P_APSI = spfull(P_APSI);
70 P_AT = -Pr*Ra*Dx; P_AT = spfull(P_AT);
71 P_BPSI = -Dx; P_BPSI = spfull(P_BPSI);
72 P_BT = DL; P_BT = spfull(P_BT);
73
74 Q_APSI = DL; Q_APSI = spfull(Q_APSI);
75 Q_AT = zeros(Nt); Q_AT = spfull(Q_AT);
76 Q_BPSI = zeros(Nt); Q_BPSI = spfull(Q_BPSI);
77 Q_BT = Id; Q_BT = spfull(Q_BT);
78
79 %%%% Condiciones de contorno %%%%
80 % Contornos verticales
81 P_AT(c_M_v,:) = 0;
82 P_BPSI(c_M_v,:) = 0;
83 Q_AT(c_M_v,:) = 0;
84 Q_BPSI(c_M_v,:) = 0;
85 % Contorno izquierdo
86 P_APSI(c_izq,:) = Id(c_izq,:) - Id(c_der,:);
87 P_BT(c_izq,:) = Id(c_izq,:) - Id(c_der,:);
88 Q_APSI(c_izq,:) = Id(c_izq,:) - Id(c_der,:);
89 Q_BT(c_izq,:) = Id(c_izq,:) - Id(c_der,:);
90 % Contorno derecho
91 P_APSI(c_der,:) = Dx(c_der,:) - Dx(c_izq,:);
92 P_BT(c_der,:) = Dx(c_der,:) - Dx(c_izq,:);
93 Q_APSI(c_der,:) = Dx(c_der,:) - Dx(c_izq,:);
94 Q_BT(c_der,:) = Dx(c_der,:) - Dx(c_izq,:);
95 % SUBcontorno izquierdo
96 P_APSI(sub_c_izq,:) = Dx2(sub_c_izq-Nz,:) - Dx2(sub_c_der+Nz,:);
97 P_AT(sub_c_izq,:) = 0;
98 Q_APSI(sub_c_izq,:) = Dx2(sub_c_izq-Nz,:) - Dx2(sub_c_der+Nz,:);
99 Q_AT(sub_c_izq,:) = 0;
100 % SUBcontorno derecho
101 P_APSI(sub_c_der,:) = Dx3(sub_c_der+Nz,:) - Dx3(sub_c_izq-Nz,:);
102 P_AT(sub_c_der,:) = 0;
103 Q_APSI(sub_c_der,:) = Dx3(sub_c_der+Nz,:) - Dx3(sub_c_izq-Nz,:);
104 Q_AT(sub_c_der,:) = 0;
105 % Contornos horizontales
106 % Contorno superior e inferior
107 P_APSI(c_M_h,:) = Id(c_M_h,:);
108 P_AT(c_M_h,:) = 0;
109 P_BPSI(c_M_h,:) = 0;
110 P_BT(c_M_h,:) = Id(c_M_h,:);
111 Q_APSI(c_M_h,:) = Id(c_M_h,:);
112 Q_AT(c_M_h,:) = 0;
113 Q_BPSI(c_M_h,:) = 0;
114 Q_BT(c_M_h,:) = Id(c_M_h,:);
115 % SUBcontornos horizontales
116 P_AT(sub_c_h,:) = 0;
117 Q_AT(sub_c_h,:) = 0;
118 % Velocidades normales nulas en SUBcontorno inferior
119 P_APSI(sub_c_inf,:) = Dz(sub_c_inf-1,:);
120 Q_APSI(sub_c_inf,:) = Dz(sub_c_inf-1,:);
121 %%% Condición en velocidades en contorno superior
122 P_APSI(sub_c_sup,:) = Dz2(sub_c_sup+1,:);
123 Q_APSI(sub_c_sup,:) = Dz2(sub_c_sup+1,:);
124
125 % Dispersa o llena
126 P_APSI = spfull(P_APSI); P_AT = spfull(P_AT);
127 P_BPSI = spfull(P_BPSI); P_BT = spfull(P_BT);
128
129 Q_APSI = spfull(Q_APSI); Q_AT = spfull(Q_AT);
130 Q_BPSI = spfull(Q_BPSI); Q_BT = spfull(Q_BT);
131
132 %%%% Construcción de las matrices del problema %%%%
133 Asyst = [P_APSI P_AT; P_BPSI P_BT]; Asyst = spfull(Asyst);
134 Qsyst = [Q_APSI Q_AT; Q_BPSI Q_BT]; Qsyst = spfull(Qsyst);
135
136 % toc
137 %%% Autovalores
138 %%% Se pueden calcular hasta 200 autovalores como máximo a la vez.
139 %%% Por tanto, para evitar el caso en que un autovalor pase a ser mayor que
140 %%% 1 y existan más de 200 con valor idénticamente uno, se calculan dos
141 %%% veces:
142 %%% - Autovalores próximos a cero, que aseguren encontrar aquellos que
143 %%% se encuentren entre 0 y 1.
144 %%% - Autovalores próximos a 2 (o cualquier número mayor que 1), que
145 %%% aseguren encontrar cualquier autovalor mayor que 1.
146 toleig = 1e-2;
147 % tic
148 [V1,D1,flag1] = eigs(Asyst,Qsyst,1,(0+(1-toleig))/2,'tol',1e-14,'FailureTreatment','drop');
149 s = 0;
150 while isempty(diag(D1)) && s <= 100
151 s = s+1;
152 [V1,D1,flag1] = eigs(Asyst,Qsyst,1,(0+(1-toleig))/2+s/100,'tol',1e-14,'FailureTreatment','drop');
153 if s >= 100
154 flag1 = 1;
155 D1 = 1e16;
156 end
157 end
158 % toc
159 % tic
160 % Es mas rapido buscar un valor que el máximo autovalor. Se supone que el
161 % máximo no estará por encima de 9, de manera que si lo fuese no se
162 % obtendría ya que se obtendría un uno
163 [V2,D2,flag2] = eigs(Asyst,Qsyst,1,5,'tol',1e-14,'FailureTreatment','drop');

```

```

164 s = 5;
165 while isempty(diag(D2)) && s < 100
166     s = s+1;
167     [V2,D2,flag2] = eigs(Asyst,Qsyst,1,s,'tol',1e-14,'FailureTreatment','drop');
168     if s >= 100
169         flag2 = 1;
170         D2 = 1e16;
171     end
172 end
173 % toc
174 av1 = diag(D1);
175 av2 = diag(D2);
176 % Si suman 1 (true) es porque hay algun autovalor con parte real positiva
177 % sin tener en cuenta los que estan en torno a 1
178
179 if sum(real(av1) > 0 & (real(av1) <= (1-toleig) | real(av1) >= (1+toleig))) || sum(real(av2)>1+toleig) > 0
180     var_Rac = 1;
181     if sum(real(av2)>1+toleig) > 0
182         ind = find(real(av2)>1+toleig);
183         V = V2(:,real(av2)>1+toleig);
184     else
185         ind = find(sum(real(av1) > 0 & (real(av1) <= (1-toleig) | real(av1) >= (1+toleig))));
186         V = V1(:,sum(real(av1) > 0 & (real(av1) <= (1-toleig) | real(av1) >= (1+toleig))));
187     end
188     V = mean(V,2);
189 else
190     var_Rac = 0;
191     V = NaN(2*Nt,1);
192 end
193 end

```

Listing B.12 Estudio lineal para contornos laterales no rectos.

```

1 clear all, close all
2 tic
3 %% Dominio rectangular con gravedad en dirección "-z"
4 % Valores constantes
5 H = 1; % Altura (m)
6 alpha = 0.2216e-4; % Difusividad térmica (m^2/s)
7 nu = 1.684e-5; % Viscosidad cinemática (m^2/s)
8 Pr = nu/alpha; % Numero de Prandtl
9 g = 9.81; % Gravedad (m^2/s)
10 difRa = 1000;
11
12 % Parámetros de la malla
13 malla_x = 'lin'; % 'cheb', 'lin', 'parsim'
14 malla_z = 'lin'; % 'cheb', 'lin', 'parsim'
15 difX = 'Fornberg'; % 'cheb', 'dif2', 'dif4', 'dif6', 'dif8', 'Fornberg'
16 difZ = 'Fornberg'; % 'cheb', 'dif2', 'dif4', 'dif6', 'dif8', 'Fornberg'
17 Dxs = 'derivadas'; % 'prod' (por defecto si difX 'cheb') o 'derivadas'
18 Dzs = 'derivadas'; % 'prod' (por defecto si difZ 'cheb') o 'derivadas'
19 Dxs = 'cadena'; % 'prod' o 'cadena'
20 Dzs = 'cadena'; % 'prod' o 'cadena'
21
22 %% Definición de los casos
23 bc_psi_top = 'wall';
24 bc_T_v = 'T0';
25 AR = 2;
26 lambdav = [0 0 0 0 1 1];
27 Av = [0 -0.04 -0.07 0.04 0.07 -0.021 -0.049]*AR;
28 %% Variables auxiliares para el bucle
29 switch difX
30     case 'cheb'
31         Nx = ceil(18.8227+0.6415*AR+0.0222*AR^2);
32     case 'Fornberg'
33         Nx = ceil(15.43+4.02*AR-0.05*AR^2)';
34 end
35 switch difZ
36     case 'cheb'
37         Nz = 30;
38     case 'Fornberg'
39         Nz = 30;
40 end
41 Nt = Nx*Nz;
42 m = length(Av);
43 Rac = NaN(m,1);
44 flag1 = NaN(m,1);
45 flag2 = NaN(m,1);
46 V = NaN(2*Nt,m);
47 xv = NaN(Nt,m);
48 zv = NaN(Nt,m);
49
50 %% % Representación de los dominios
51 %% for k = 1:m
52 %%     lambdab = lambdav(k);
53 %%     lambdat = lambdav(k);
54 %%     Ab = Av(k);
55 %%     At = Av(k);
56 %%     fig_dominio = 1;
57 %%     fig_malla = 1;
58 %%     [xch,zch,xmat,zmat,xmat,zmat,Nt,zbv,ztv,zb,zt] = geom(...
59 %%     Nx,Nz,malla_x,malla_z,fig_dominio,fig_malla,H,AR,At,Ab,lambdat,lambdab);

```

```

60 %% axis equal
61 %% close all
62 %% end
63 %% Resolución
64 for k = 1:m
65     lambdab = lambdav(k);
66     lambdat = lambdav(k);
67     if lambdav(k) == 0
68         tipo_contorno = 'par';
69     else
70         tipo_contorno = 'sin';
71     end
72     Ab = Av(k);
73     At = -Av(k);
74     Ra = 100;
75     var_Rac = 0;
76     while var_Rac == 0
77         [var_Rac,~,f1,f2,~,~] = calculo_estab(bc_psi_top,...
78             bc_T_v,Nx,Nz,Pr,H,AR,At,Ab,lambdat,lambdab,tipo_contorno,...
79             Ra,malla_x,malla_z,difX,difZ,Dxs,Dzs,...
80             DZs,Dxs,Dzs);
81         if f1 == 1 || f2 == 1
82             flag1(k,1) = 1;
83             disp('----1----')
84         end
85         Ra = Ra + difRa;
86     end
87     Racinf = Ra-2*difRa;
88     Racsup = Ra-difRa;
89     var_Rac2 = 0;
90     while var_Rac2 == 0
91         Ra = (Racinf+Racsup)/2;
92         [var_Rac,Vaux,f1,f2,xv(1:Nt,k),zv(1:Nt,k)] = calculo_estab(...
93             bc_psi_top,bc_T_v,Nx,Nz,Pr,H,AR,At,Ab,lambdat,lambdab,...
94             tipo_contorno,Ra,malla_x,malla_z,difX,difZ,Dxs,Dzs,Dzs);
95         if f1 == 1 || f2 == 1
96             flag2(ii,jj,k) = 1;
97             disp('----2----')
98         end
99         if var_Rac == 1
100             Racsup = Ra;
101             V(1:2*Nt,k) = Vaux;
102         else
103             Racinf = Ra;
104         end
105         if (Racsup-Racinf)<0.1
106             var_Rac2 = 1;
107         end
108     end
109     Rac(k,1) = (Racsup+Racinf)/2;
110     disp([num2str(Racinf),'-',num2str(Racsup)])
111 end
112 t_Rac_malla = toc
113 save('topwall_def_T0_estab')
114
115 %% Resultados
116 for k = 1:m
117     k
118     lambdab = lambdav(k);
119     lambdat = lambdav(k);
120     Ab = Av(k);
121     At = -Av(k);
122     Psimat = reshape(V(1:Nt,k),Nz,Nx);
123     Tmat = reshape(V((Nt+1):(2*Nt),k),Nz,Nx);
124     xmat = reshape(xv(1:Nt,k),Nz,Nx);
125     zmat = reshape(zv(1:Nt,k),Nz,Nx);
126     figure(k)
127     subplot(2,1,1),contour(xmat,zmat,Tmat)
128     ylabel('$\hat{t}$','interpreter','latex')
129     title('Temperatura de perturbación adimensional')
130     axis equal
131     set(gca,'FontSize',12)
132     set(gcf,'renderer','Painters')
133     subplot(2,1,2),contour(xmat,zmat,Psimat)
134     xlabel('$\hat{x}$','interpreter','latex')
135     ylabel('$\hat{z}$','interpreter','latex')
136     title('Función de corriente de perturbación adimensional')
137     axis equal
138     Ataux = floor(1000*At);
139     set(gca,'FontSize',12)
140     set(gcf,'renderer','Painters')
141     print('-depsc2',[ 'topwall_T0_AR'...
142         ,num2str(AR),'_',num2str(Ataux),'_',num2str(lambdat),'_eps'])
143     saveas(figure(k),[ 'topwall_T0_AR'...
144         ,num2str(AR),'_',num2str(Ataux),'_',num2str(lambdat),'_fig'])
145     pause(1e-16)
146 end

```

Listing B.13 Función análisis lineal asociada a B.12.

```

1 % Función que determina si hay autovalores con parte real mayor que uno,
2 % ignorando aquellos que son unidad debido a la forma de imponer las

```



```

3 % condiciones de contorno. Además, obtiene el autovector asociado a ese
4 % autovalor si existe.
5 function [var_Rac,V,flag1,flag2,xv,zv] = calculo_estab(bc_psi_top,bc_T_v,...
6     Nx,Nz,Pr,H,AR,At,Ab,lambdat,lambdab,tipo_contorno,...
7     Ra,malla_x,malla_z,difX,difZ,DXs,DZs,Dxs,Dzs)
8 %%% Representación del dominio y la malla (1 es si) %%%
9 fig_dominio = 0;
10 fig_malla = 0;
11 %%% Cálculo de las derivadas unidimensionales: difX y difZ %%%
12 switch difX
13     case 'cheb'
14         DXs = 'prod';
15     end
16 switch difZ
17     case 'cheb'
18         DZs = 'prod';
19     end
20 % tic
21 %% Geometría del dominio
22 [xch,zch,Xmat,Zmat,xmat,zmat,Nt,~,~,x1,xr] = geom_lr(...
23     Nx,Nz,malla_x,malla_z,fig_dominio,fig_malla,H,AR,...
24     At,Ab,lambdat,lambdab,tipo_contorno);
25 xv = reshape(xmat,Nt,1);
26 zv = reshape(zmat,Nt,1);
27 %% Derivadas
28 [Dx,~,~,Dz,Dz2,~,~,DL,DL2,~,~] = mat_derivadas_lr(difX,difZ,DXs,DZs,...
29     Dxs,Dzs,Xmat,Zmat,Nx,Nz,Nt,x1,xr,xch,zch,...
30     At,Ab,lambdat,lambdab,tipo_contorno,AR);
31
32 %% Contornos y subcontornos
33 Id = speye(Nt);
34 c_sup = (Nz:Nz:(Nx*Nz))'; % c. Superior
35 c_inf = (1:Nz:(Nz*(Nx-1)+1))'; % c. Inferior
36 c_der = ((Nx-1)*Nz+1):(Nx*Nz)'; % c. Derecho
37 c_izq = (1:Nz)'; % c. Izquierdo
38 c_M = [c_inf;c_sup;c_izq;c_der]; % Matriz contornos
39 c_M_h = [c_inf;c_sup];
40 c_M_v = [c_izq;c_der];
41 sub_c_sup = ((2*Nz-1):Nz:(Nx-1)*Nz-1))'; % Subc. Superior
42 sub_c_inf = ((Nz+2):Nz:(Nz*(Nx-2)+2))'; % Subc. Inferior
43 sub_c_izq = ((Nz+2):(2*Nz-1))'; % Subc. Izquierdo
44 sub_c_der = ((Nz*(Nx-2)+2):(Nz*(Nx-1)-1))'; % Subc. Derecho
45 sub_c_M = [sub_c_inf;sub_c_sup;sub_c_izq;sub_c_der]; % Matriz subcontornos
46 %% Sistema matricial (gravidad en dirección '-z')
47 % Psi = Psi0*e^(st)
48 % T = T0*e^(st)
49 % x = [Psi;T]
50 % Px = sQx
51
52 %%% Variables %%%
53
54 % T: desviación de temperatura respecto al equilibrio
55 % Psi: desviación de la función de corriente respecto al equilibrio
56
57 %%% Ecuaciones adimensionales %%%
58
59 % P = [ P_APSI P_AT
60 %     P_BPSI P_BT ]
61 % Q = [ Q_APSI Q_AT [ DL 0
62 %     Q_BPSI Q_BT ] = 0 1 ]
63
64 P_APSI = Pr*DL2; P_APSI = spfull(P_APSI);
65 P_AT = -Pr*Ra*Dx; P_AT = spfull(P_AT);
66 P_BPSI = -Dx; P_BPSI = spfull(P_BPSI);
67 P_BT = DL; P_BT = spfull(P_BT);
68
69 Q_APSI = DL; Q_APSI = spfull(Q_APSI);
70 Q_AT = zeros(Nt); Q_AT = spfull(Q_AT);
71 Q_BPSI = zeros(Nt); Q_BPSI = spfull(Q_BPSI);
72 Q_BT = Id; Q_BT = spfull(Q_BT);
73
74 %%% Condiciones de contorno %%%
75
76 % Psi impuesto (nula) en C. Verticales y Horizontales
77 P_APSI(c_M,:) = Id(c_M,:);
78 P_AT(c_M,:) = 0;
79 Q_APSI(c_M,:) = Id(c_M,:);
80 Q_AT(c_M,:) = 0;
81 % Condición común a los SUBcontornos
82 P_AT(sub_c_M,:) = 0;
83 Q_AT(sub_c_M,:) = 0;
84 % Velocidades normales nulas en SUBcontorno derecho e izquierdo
85 P_APSI(sub_c_der,:) = Dx(sub_c_der+Nz,:);
86 Q_APSI(sub_c_der,:) = Dx(sub_c_der+Nz,:);
87 P_APSI(sub_c_izq,:) = Dx(sub_c_izq-Nz,:);
88 Q_APSI(sub_c_izq,:) = Dx(sub_c_izq-Nz,:);
89 % Velocidades normales nulas en SUBcontorno inferior
90 P_APSI(sub_c_inf,:) = Dz(sub_c_inf-1,:);
91 Q_APSI(sub_c_inf,:) = Dz(sub_c_inf-1,:);
92 %% Condición en velocidades en contorno superior
93 switch bc_psi_top
94     case 'wall'
95         % Velocidades normales nulas en SUBcontorno superior
96         P_APSI(sub_c_sup,:) = Dz(sub_c_sup+1,:);
97         Q_APSI(sub_c_sup,:) = Dz(sub_c_sup+1,:);
98     case 'free'

```

```

99 % Esfuerzo cortante nulo en SUBcontorno superior
100 P_APSI(sub_c_sup,:) = Dz2(sub_c_sup+1,:);
101 Q_APSI(sub_c_sup,:) = Dz2(sub_c_sup+1,:);
102 end
103
104 %%% Condición en temperatura
105 switch bc_T_v
106 case 'Q0'
107     % T impuesta (nula) en C. Horizontales
108     P_BT(c_M_h,:) = Id(c_M_h,:);
109     P_BPSI(c_M_h,:) = 0;
110     Q_BT(c_M_h,:) = Id(c_M_h,:);
111     Q_BPSI(c_M_h,:) = 0;
112     % Q impuesto (nulo) en C. Verticales
113     P_BT(c_M_v,:) = Dz(c_M_v,:);
114     P_BPSI(c_M_v,:) = 0;
115     Q_BT(c_M_v,:) = Dz(c_M_v,:);
116     Q_BPSI(c_M_v,:) = 0;
117 case 'T0'
118     % T impuesta (nula) en C. Verticales y Horizontales
119     P_BT(c_M,:) = Id(c_M,:);
120     P_BPSI(c_M,:) = 0;
121     Q_BT(c_M,:) = Id(c_M,:);
122     Q_BPSI(c_M,:) = 0;
123 end
124
125 % Dispersa o llena
126 P_APSI = spfull(P_APSI); P_AT = spfull(P_AT);
127 P_BPSI = spfull(P_BPSI); P_BT = spfull(P_BT);
128
129 Q_APSI = spfull(Q_APSI); Q_AT = spfull(Q_AT);
130 Q_BPSI = spfull(Q_BPSI); Q_BT = spfull(Q_BT);
131
132 %%% Construcción de las matrices del problema %%%
133 Asyst = [P_APSI P_AT; P_BPSI P_BT]; Asyst = spfull(Asyst);
134 Qsyst = [Q_APSI Q_AT; Q_BPSI Q_BT]; Qsyst = spfull(Qsyst);
135
136 % toc
137 %%% Autovalores
138 %%% Se pueden calcular hasta 200 autovalores como máximo a la vez.
139 %%% Por tanto, para evitar el caso en que un autovalor pase a ser mayor que
140 %%% 1 y existan más de 200 con valor idénticamente uno, se calculan dos
141 %%% veces:
142 %%% - Autovalores próximos a cero, que aseguren encontrar aquellos que
143 %%% se encuentren entre 0 y 1.
144 %%% - Autovalores próximos a 2 (o cualquier número mayor que 1), que
145 %%% aseguren encontrar cualquier autovalor mayor que 1.
146 toleig = 1e-4;
147 % tic
148 [V1,D1,flag1] = eigs(Asyst,Qsyst,1,(0+(1-toleig))/2,'tol',1e-14,'FailureTreatment','drop');
149 s = 0;
150 while isempty(diag(D1)) && s <= 100
151     s = s+1;
152     [V1,D1,flag1] = eigs(Asyst,Qsyst,1,(0+(1-toleig))/2+s/100,'tol',1e-14,'FailureTreatment','drop');
153     if s >= 100
154         flag1 = 1;
155         D1 = 1e16;
156     end
157 end
158 % toc
159 % tic
160 % Es mas rapido buscar un valor que el máximo autovalor. Se supone que el
161 % máximo no estará por encima de 9, de manera que si lo fuese no se
162 % obtendría ya que se obtendría un uno
163 [V2,D2,flag2] = eigs(Asyst,Qsyst,1,5,'tol',1e-14,'FailureTreatment','drop');
164 s = 5;
165 while isempty(diag(D2)) && s < 100
166     s = s+1;
167     [V2,D2,flag2] = eigs(Asyst,Qsyst,1,s,'tol',1e-14,'FailureTreatment','drop');
168     if s >= 100
169         flag2 = 1;
170         D2 = 1e16;
171     end
172 end
173 % toc
174 av1 = diag(D1);
175 av2 = diag(D2);
176 % Si suman 1 (true) es porque hay algun autovalor con parte real positiva
177 % sin tener en cuenta los que estan en torno a 1
178
179 if sum(real(av1) > 0 & (real(av1) <= (1-toleig) | real(av1) >= (1+toleig))) || sum(real(av2)>(1+toleig)) > 0
180     var_Rac = 1;
181     if sum(real(av2)>(1+toleig)) > 0
182         ind = find(real(av2)>(1+toleig));
183         V = V2(:,real(av2)>(1+toleig));
184     else
185         ind = find(sum(real(av1) > 0 & (real(av1) <= (1-toleig) | real(av1) >= (1+toleig))));
186         V = V1(:,sum(real(av1) > 0 & (real(av1) <= (1-toleig) | real(av1) >= (1+toleig))));
187     end
188     V = mean(V,2);
189     if sum(sum(abs(imag(V)))) > 0
190         pause
191     end
192 else
193     var_Rac = 0;
194     V = NaN(2*Nt,1);

```

```
195 end
196 end
```

## B.4 Resolución del sistema completo para convección natural

Listing B.14 Resolución completa en el tiempo del sistema.

```
1 clear all,close all
2 t_th1 = 10;
3 t_th2 = 10;
4 dtv = [5e-3 1e-3 5e-4];
5 dtv2 = [5e-3 1e-3 5e-4 1e-4];
6 Rav = [3e3 4e3 5e3 6e3 8e3 10e3];
7 m = length(dtv);
8 n = length(Rav);
9 %% Diferencias finitas con nodos de Chebyshev
10 malla_x = 'cheb'; % 'cheb', 'lin'
11 malla_z = 'cheb'; % 'cheb', 'lin'
12 difX = 'Fornberg'; % 'cheb', 'dif2', 'dif4', 'dif6', 'dif8', 'Fornberg'
13 difZ = 'Fornberg'; % 'cheb', 'dif2', 'dif4', 'dif6', 'dif8', 'Fornberg'
14 Nz = 30;
15 k = 0;
16 Qt_DF = NaN(1e6,n,m);
17 Qv_DF = NaN(1,n,m);
18 tf_DF = NaN(1,n,m);
19 ntf_DF = NaN(1,n,m);
20 flagC_DF = NaN(1,n,m);
21 tmax_x_DF = NaN(1e6,n,m);
22 tmax_z_DF = NaN(1e6,n,m);
23 umed_DF = NaN(1e6,n,m);
24 vmed_DF = NaN(1e6,n,m);
25 maxu_DF = NaN(1e6,n,m);
26 maxv_DF = NaN(1e6,n,m);
27 for dt = dtv
28     k = k + 1
29     [tmax_x_DF(:,:,k),tmax_z_DF(:,:,k),umed_DF(:,:,k),vmed_DF(:,:,k),...
30     maxu_DF(:,:,k),maxv_DF(:,:,k),Qt_DF(:,:,k),Qv_DF(1, :,k),tf_DF(1, :,k),ntf_DF(1, :,k),...
31     flagC_DF(1, :,k)] = ...
32     fun_Ra_dt(Rav,Nz,AR,dt,t_th1,t_th2,malla_x,malla_z,difX,difZ);
33 end
34 save('Ra_dt_1708_AR1')
```

Listing B.15 Función resolución completa asociada a B.14.

```
1 %% Función que resuelve el sistema a lo largo del tiempo hasta cumplir un
2 %% criterio de convergencia basado en dos parámetros: t_th1 y t_th2
3 function [tmax_x,tmax_z,umed,vmed,maxu,maxv,Qt,Qv,tf,ntf,flagC] = ...
4     fun_Ra_dt(Rav,Nz,AR,dt,t_th1,t_th2,malla_x,malla_z,difX,difZ)
5 %% Dominio rectangular con gravedad en dirección "-z"
6 %% Valores constantes
7 T2fis = 298.15; % Temperatura inferior
8 H = 1; % Altura (m)
9 alpha = 0.2216e-4; % Difusividad térmica (m^2/s)
10 nu = 1.684e-5; % Viscosidad cinemática (m^2/s)
11 Pr = nu/alpha; % Numero de Prandtl
12 g = 9.81; % Gravedad (m^2/s)
13 DIs = 'derivadas'; % 'prod' (por defecto si difZ 'cheb') o 'derivadas'
14 DZs = 'derivadas'; % 'prod' (por defecto si difZ 'cheb') o 'derivadas'
15 Dxs = 'cadena'; % 'prod' o 'cadena'
16 Dzs = 'cadena'; % 'prod' o 'cadena'
17
18 %% Parámetros
19 %% Geometría de los límites
20 At = 0; Ab = 0;
21 lambdat = 1; lambdab = 1;
22
23 switch difX
24     case 'cheb'
25         Nx = ceil(18.8227+0.6415*AR+0.0222*AR^2);
26         Nx = 41;
27     case 'Fornberg'
28         Nx = min(400,ceil(62-2.98*AR+AR^2));
29 end
30 %%% Representación del dominio y la malla (1 es sí) %%%
31 fig_dominio = 0;
32 fig_malla = 0;
33
34 %% Geometría del dominio
35 [xch,zch,Xmat,Zmat,xmat,zmat,Nt,~,~,zb,zt] = geom(...
36     Nx,Nz,malla_x,malla_z,fig_dominio,fig_malla,H,AR,At,Ab,lambdat,lambdab);
37
38 %% Derivadas
```

```

39 [Dx,Dx2,~,~,Dz,~,~,DL,DL2,DxDL,DzDL] = mat_derivadas_topbot(...
40 difX,difZ,DXs,DZs,Dxs,Dzs,Xmat,Zmat,Nx,Nz,Nt,zb,zt,...
41 xch,zch,At,Ab,lambdat,lambdab);
42 %% Sistema y condiciones de contorno
43 Id = speye(Nt);
44 c_sup = (Nz:Nz:(Nx*Nz))'; % c. Superior
45 c_inf = (1:Nz:(Nz*(Nx-1)+1))'; % c. Inferior
46 c_der = ((Nx-1)*Nz+1):(Nx*Nz)'; % c. Derecho
47 c_izq = (1:Nz)'; % c. Izquierdo
48 c_M = [c_inf;c_sup;c_izq;c_der]; % Matriz contornos
49 sub_c_sup = ((2*Nz-1):Nz:(Nx-1)*Nz-1)'; % Subc. Superior
50 sub_c_inf = ((Nz+2):Nz:(Nz*(Nx-2)+2))'; % Subc. Inferior
51 sub_c_izq = ((Nz+2):(2*Nz-1))'; % Subc. Izquierdo
52 sub_c_der = ((Nz*(Nx-2)+2):(Nz*(Nx-1)-1))'; % Subc. Derecho
53 sub_c_M = [sub_c_inf;sub_c_sup;sub_c_izq;sub_c_der]; % Matriz subcontornos
54 m_sup = length(c_sup); m_inf = length(c_inf);
55
56 %%% Ecuaciones adimensionales %%%
57 % Asyst = [ APSI AT
58 % BPSI BT ]
59
60 APSI = DL-dt*Pr*DL2; APSI = spfull(APSI);
61 AT = dt*Pr*Dx; AT = spfull(AT);
62 BPSI = zeros(Nt,Nt); BPSI = spfull(BPSI);
63 BT = Id-dt*DL; BT = spfull(BT);
64
65 %%% Condiciones de contorno %%%
66 % T impuesta (nula) en C. Verticales y Horizontales
67 BT(c_M,:) = Id(c_M,:);
68 BPSI(c_M,:) = 0;
69
70 % Psi impuesto (nula) en C. Verticales y Horizontales
71 APSI(c_M,:) = Id(c_M,:);
72 AT(c_M,:) = 0;
73 % Condición común a los SUBcontornos
74 AT(sub_c_M,:) = 0;
75 % Velocidades normales nulas en SUBcontornos superior e inferior
76 APSI(sub_c_sup,:) = Dz(sub_c_sup+1,:);
77 APSI(sub_c_inf,:) = Dz(sub_c_inf-1,:);
78 % Esfuerzo cortante en SUBcontornos izquierdo y derecho nulo
79 APSI(sub_c_izq,:) = Dx2(sub_c_izq-Nz,:);
80 APSI(sub_c_der,:) = Dx2(sub_c_der+Nz,:);
81
82 % Dispersa o llena
83 APSI = spfull(APSI); AT = spfull(AT);
84 BPSI = spfull(BPSI); BT = spfull(BT);
85
86 %%% Construcción de la matriz del problema y factorización %%%
87 Asyst = [APSI AT; BPSI BT]; Asyst = spfull(Asyst);
88 [L_lu,U_lu,P_lu] = lu(Asyst);
89
90 %% Resolución para varios Ra
91 u1 = 0;
92 n = length(Rav);
93 t = NaN(1e6,n);
94 Cx = NaN(1e6,n);
95 Cz = NaN(1e6,n);
96 tmax_x = NaN(1e6,n);
97 tmax_z = NaN(1e6,n);
98 umed = NaN(1e6,n);
99 vmed = NaN(1e6,n);
100 maxu = NaN(1e6,n);
101 maxv = NaN(1e6,n);
102 Qt = NaN(1e6,n);
103 Psit = NaN(1e6,n);
104 CVPsi = NaN(1e6,n);
105 CVQ = NaN(1e6,n);
106 stdQ = NaN(1e6,n);
107 stdPsi = NaN(1e6,n);
108 flagC = NaN(n,1);
109 Qv = NaN(n,1);
110 tf = NaN(n,1);
111 ntf = NaN(n,1);
112
113 for Ra = Rav
114 tic
115 u1 = u1 + 1;
116
117 %%% Solución de equilibrio %%%
118 Asyst_lp = DL;
119 Asyst_lp(c_M,:) = Id(c_M,:); Asyst_lp = spfull(Asyst_lp);
120 T2 = -Ra/2; % Temp adimensional en c.superior
121 T1 = Ra/2; % Temp adimensional en c.inferior
122 FBC_T_eq = zeros(Nt,1);
123 FBC_izq = interp1(linspace(zmat(1,1),zmat(end,1),Nz),...
124 linspace(T1,T2,Nz),zmat(1:end,1));
125 FBC_der = interp1(linspace(zmat(1,Nx),zmat(end,Nx),Nz),...
126 linspace(T1,T2,Nz),zmat(1:end,Nx));
127 FBC_T_eq(c_M,1) = [T1*ones(m_inf,1);T2*ones(m_sup,1);FBC_izq;FBC_der];
128 Teq = Asyst_lp\FBC_T_eq;
129
130 %%% Perturbación inicial %%%
131 Psiper = 1e-8*(Xmat.^2).*(max(max(Xmat))-...
132 Xmat).^2).*(Zmat.^2).*(max(max(Zmat))-Zmat).^2);
133 Tper = 1e-8*sin(3*pi*Zmat);
134

```

```

135 Psiper = reshape(Psiper,Nt,1);
136 Tper = reshape(Tper,Nt,1);
137 %%% Solución en el instante inicial %%%
138 T = Teq + Tper;
139 Psi = Psiper;
140
141 %%% Condiciones de contorno %%%
142 FBC_T = ones(Nt,1);
143 FBC_T(c_M,1) = 0;
144 FBC_Psi = ones(Nt,1);
145 FBC_Psi(c_M,1) = 0;
146 FBC_Psi(sub_c_M,1) = 0;
147 dfxmat = diff(xmat)';
148 dfzmat = diff(zmat);
149
150 nt = 0;
151 criterio = 0;
152 while criterio == 0
153     nt = nt + 1;
154     if nt == 1
155         t(1,u1) = 0;
156     else
157         t(nt,u1) = t(nt-1,u1) + dt;
158     end
159     u_v = Dz*Psi; % v_x (vector)
160     v_v = -Dx*Psi; % v_z (vector)
161     bNLPsi(1:Nt,1) = (-u_v.*(DxDL*Psi)-v_v.*(DzDL*Psi))*dt;
162     bNLT(1:Nt,1) = (-u_v.*(Dx*T)-v_v.*(Dz*T))*dt;
163     bLinPsi(1:Nt,1) = DL*Psi;
164     bLinT(1:Nt,1) = T;
165     bsyst(1:2*Nt,1) = [FBC_Psi.*(bNLPsi+bLinPsi); ...
166                     FBC_T.*(bNLT+bLinT)+(1-FBC_T).*FBC_T_eq];
167
168 %%% Resolución del sistema %%%
169 bn = U_lu\L_lu\ (P_lu*bsyst);
170
171 Psi = bn(1:Nt,1);
172 T = bn((Nt+1):end,1);
173
174 %%% Se comprueba que el paso de tiempo es adecuado %%%
175 umat = reshape(u_v,Nz,Nx);
176 vmat = reshape(v_v,Nz,Nx);
177 Courant_x = dt*abs(umat(:,2:end))./dfxmat;
178 Cx(nt,u1) = max(max(Courant_x));
179 Courant_z = dt*abs(vmat(2:end,:))./dfzmat;
180 Cz(nt,u1) = max(max(Courant_z));
181 tmax_x(nt,u1) = min(min(0.9*dfxmat./abs(umat(:,2:end)))));
182 maxu(nt,u1) = max(max(abs(umat)));
183 umed(nt,u1) = trapz(zch,trapz(xch,abs(umat),2))/AR/1;
184 tmax_z(nt,u1) = min(min(0.9*dfzmat./abs(vmat(2:end,:)))));
185 maxv(nt,u1) = max(max(abs(vmat)));
186 vmed(nt,u1) = trapz(zch,trapz(xch,abs(vmat),2))/AR/1;
187
188 %%% Resultados
189 Psimat = reshape(Psi,Nz,Nx);
190 Tmat = reshape(T,Nz,Nx);
191 Tifis = T2fis*((2*g*H^3+Ra*nu*alpha)/(2*g*H^3-Ra*nu*alpha)); % Kelvin
192 T0 = (Tifis+T2fis)/2;
193 Tfis = T0+nu*alpha*T0/g/H^3*T;
194 Tfismat = reshape(Tfis,Nz,Nx);
195 Qfield = reshape(Dz*Tfis,Nz,Nx);
196 Qx = Qfield(1,1:end);
197 Qt(nt,u1) = Q;
198 Psit(nt,u1) = trapz(zch,trapz(xch,abs(Psimat),2));
199
200 %%% Criterio de convergencia %%%
201
202 if t(nt,u1) > t_th1
203     m1 = (nt-t_th2/dt):nt;
204     CVPsi(nt,u1) = std(Psit(m1,u1))/mean(Psit(m1,u1));
205     CVQ(nt,u1) = std(Qt(m1,u1))/mean(Qt(m1,u1));
206     stdQ(nt,u1) = std(Qt(m1,u1));
207     stdPsi(nt,u1) = std(Psit(m1,u1));
208 end
209 if t(nt,u1) > t_th1
210     if (abs(CVPsi((nt-t_th2/dt):nt,u1)) < 0.5 | Psit((nt-t_th2/dt):nt,u1) < 1e-3 & abs(CVQ(nt,u1)) < 0.5
211         criterio = 1;
212         % plot(t(1:nt,u1),maxv(1:nt,u1))
213         % figure(u1)
214         % subplot(4,1,1),plot(1:nt,Psit(1:nt,u1))
215         % subplot(4,1,2),plot(1:nt,Qt(1:nt,u1))
216         % subplot(4,1,3),plot(1:nt,CVPsi(1:nt,u1))
217         % subplot(4,1,4),plot(1:nt,CVQ(1:nt,u1)),pause(1e-15)
218         if Psit((nt-t_th2/dt):nt,u1) < 1
219             disp('Función de corriente prácticamente cero')
220         else
221             disp('Estacionario')
222         end
223     end
224 end
225
226 %%% Courant %%%
227 if Cx(nt,u1) >= 1 || Cz(nt,u1) >= 1 || isnan(Cx(nt,u1)) || isnan(Cz(nt,u1)) || isnan(Psit(nt,u1))
228     disp(['Cx = ',num2str(Cx(nt,u1))])
229     disp(['Cz = ',num2str(Cz(nt,u1))])
230     flagC(u1) = 1;

```

```

231     disp(num2str(flagC(u1)))
232     criterio = 1;
233 end
234 %%%% Rollos de Bénard %%%%
235 aux2 = NaN(size(Psimat)+2);
236 aux2(2:(end-1),2:(end-1)) = abs(Psimat);
237 max1 = islocalmax(aux2,1,'FlatSelection','center');
238 max2 = islocalmax(aux2,2,'FlatSelection','center');
239 max1 = max1(2:(end-1),2:(end-1));
240 max2 = max2(2:(end-1),2:(end-1));
241 pos1 = find(max1==1);
242 pos2 = find(max2==1);
243 cnt = intersect(pos1,pos2);
244 n_B = length(cnt);
245
246 %%%% Representación de los campos %%%%
247
248 if (nt == 501) || (nt == 1001) ||(nt == 2501) || (nt == 5001) ||...
249     (nt == 7501) || (nt == 10001) || criterio == 1
250 figure(101)
251 subplot(2,1,1), hold on, xlabel('x [-]'),ylabel('z [-]')
252 plot(xch,zbv,'k.-')
253 contour(xmat,zmat,Tmat,'ShowText','on')
254 title('Campo de temperatura [-]'), hold off
255 subplot(2,1,2), hold on, xlabel('x [-]'),ylabel('z [-]')
256 plot(xch,zbv,'k.-'), hold on
257 contour(xmat,zmat,Psimat,'ShowText','on')
258 plot(xmat(cnt),zmat(cnt),'or')
259 plot(xmat(pos1),zmat(pos1),'xb')
260 plot(xmat(pos2),zmat(pos2),'or')
261 title('Función de corriente [-]'), hold off
262 pause(1e-16)
263 print('-depsc2',['rol_Ben',malla_x,malla_z,difX,difZ,num2str(AR),'_',num2str(t(nt,u1)),'_',num2str(Ra),num2str(criterio),'_eps
264     '])
265 saveas(figure(101),['rol_Ben',malla_x,malla_z,difX,difZ,num2str(AR),'_',num2str(t(nt,u1)),'_',num2str(Ra),num2str(criterio),'_
266     'fig'])
267 close all
268 end
269 close all
270 Qv(u1,1) = 0;
271 tf(u1,1) = t(nt,u1);
272 ntf(u1,1) = nt;
273 disp(num2str(t(nt,u1)))
274 end
275
276 end

```

## B.5 Resolución del sistema completo para convección forzada

**Listing B.16** Resolución completa en el tiempo del sistema para convección forzada.

```

1 clear all, close all, cloc
2 t_th1 = 20;
3 t_th2 = 20;
4 dt = 1e-3;
5
6 ARv = [4 8]; m = length(ARv);
7 Psisupv = [0 1 2.5 5]; m2 = length(Psisupv);
8 Rav = [1500 2500 5000]; n = length(Rav);
9 malla_x = 'lin'; % 'cheb', 'lin'
10 malla_z = 'lin'; % 'cheb', 'lin'
11 difX = 'Fornberg'; % 'cheb', 'dif2', 'dif4', 'dif6', 'dif8', 'Fornberg'
12 difZ = 'Fornberg'; % 'cheb', 'dif2', 'dif4', 'dif6', 'dif8', 'Fornberg'
13 Nz = 30;
14 Qt = NaN(1e6,n,m,m2);
15 Qv = NaN(1,n,m,m2);
16 Nut = NaN(1e6,n,m,m2);
17 Nuv = NaN(1,n,m,m2);
18 Qt_s = NaN(1e6,n,m,m2);
19 Qv_s = NaN(1,n,m,m2);
20 Nut_s = NaN(1e6,n,m,m2);
21 Nuv_s = NaN(1,n,m,m2);
22 tf = NaN(1,n,m,m2);
23 ntf = NaN(1,n,m,m2);
24 periodo = NaN(1,n,m,m2);
25 flagC = NaN(1,n,m,m2);
26 ntv = NaN(1,n,m,m2);
27 tmax_x = NaN(1e6,n,m,m2);
28 tmax_z = NaN(1e6,n,m,m2);
29 umed = NaN(1e6,n,m,m2);
30 vmed = NaN(1e6,n,m,m2);
31 maxu = NaN(1e6,n,m,m2);
32 maxv = NaN(1e6,n,m,m2);
33 Nux_s = NaN(5e3,4e2,n,m,m2);
34 Nxv = NaN(m,1);

```

```

35 xmat = NaN(Nz,400,m);
36 zmat = NaN(Nz,400,m);
37 geo = 0; lambdab = 1; Ab = 0;
38 k = 0;
39 for AR = ARv
40     k = k + 1
41     p = 0;
42     for Psisup = Psisupv
43         p = p + 1
44         [tmax_x(:, :, k, p), tmax_z(:, :, k, p), umed(:, :, k, p), vmed(:, :, k, p), ...
45         maxu(:, :, k, p), maxv(:, :, k, p), Qt(:, :, k, p), Qv(1, :, k, p), ...
46         Nut(:, :, k, p), Nuv(1, :, k, p), Qt_s(:, :, k, p), Qv_s(1, :, k, p), ...
47         Nut_s(:, :, k, p), Nuv_s(1, :, k, p), Nux_s(:, :, k, p), xmat(:, :, k), ...
48         tf(1, :, k, p), ntf(1, :, k, p), periodo(1, :, k, p), flagC(1, :, k, p), ...
49         ntv(1, :, k, p), Nxv(k, 1)] = ...
50         fun_Ra_dt(Psisup, lambdab, Ab, Rav, Nz, AR, dt, t_th1, t_th2, ...
51         malla_x, malla_z, difX, difZ, geo);
52     end
53 end
54 save('resultados')

```

Listing B.17 Función resolución completa asociada a B.16.

```

1  % Función para resolver el sistema en el tiempo aplicando condiciones de
2  % periodicidad en los contornos laterales e imponiendo un caudal de flujo.
3  % Existe un criterio de convergencia y se calcula el periodo en caso de que
4  % la solución sea periódica.
5  function [tmax_x, tmax_z, umed, vmed, maxu, maxv, Qt, Qv, Nut, Nuv, Qt_s, Qv_s, ...
6  Nut_s, Nuv_s, Nux_s, xmatv, zmatv, tf, ntf, periodo, flagC, ntv, Nx] = fun_Ra_dt(...
7  Psisup, lambdab, Ab, Rav, Nz, AR, dt, t_th1, t_th2, malla_x, malla_z, difX, difZ, geo)
8  %% Dominio rectangular con gravedad en dirección "-z"
9  %% Valores constantes
10 T2fis = 298.15; % Temperatura inferior
11 H = 1; % Altura (m)
12 alpha = 0.2216e-4; % Difusividad térmica (m^2/s)
13 nu = 1.684e-5; % Viscosidad cinemática (m^2/s)
14 Pr = nu/alpha; % Numero de Prandtl
15 g = 9.81; % Gravedad (m^2/s)
16 DXs = 'derivadas'; % 'prod' (por defecto si difZ 'cheb') o 'derivadas'
17 DZs = 'derivadas'; % 'prod' (por defecto si difZ 'cheb') o 'derivadas'
18 Dxs = 'cadena'; % 'prod' o 'cadena'
19 Dzs = 'cadena'; % 'prod' o 'cadena'
20
21 %% Parámetros
22 % Geometría de los límites
23 At = 0;
24 lambdat = 1;
25
26 switch difX
27     case 'cheb'
28         % Nx = ceil(18.8227+0.6415*AR+0.0222*AR^2);
29         Nx = 41;
30     case 'Fornberg'
31         Nx = min(400, ceil(62-2.98*AR+AR^2));
32 end
33
34 %%% Representación del dominio y la malla (1 es sí) %%%
35 fig_dominio = 0;
36 fig_malla = 0;
37
38 %% Geometría del dominio
39 [xch, zch, Xmat, Zmat, xmat, zmat, Nt, zbv, ~, zb, zt, mu, s] = geom(...
40 Nx, Nz, malla_x, malla_z, fig_dominio, fig_malla, H, AR, At, Ab, lambdat, lambdab, geo);
41 xmatv = [xmat, NaN(30, 400-length(xmat(1, :)))];
42 zmatv = [zmat, NaN(30, 400-length(xmat(1, :)))];
43 %% Derivadas
44 [Dx, Dx2, Dx3, Dx4, Dz, ~, ~, ~, DL, DL2, DxDL, DzDL, N, DS] = ...
45 mat_derivadas_topbot(difX, difZ, DXs, DZs, Dxs, Dzs, Xmat, Zmat, Nx, Nz, Nt, ...
46 zb, zt, xch, zch, At, Ab, lambdat, lambdab, geo, AR, mu, s, zbv);
47
48 Id = speye(Nt);
49 c_sup = (Nz:Nz:(Nx*Nz))'; % c. Superior
50 c_inf = (1:Nz:(Nz*(Nx-1)+1))'; % c. Inferior
51 c_der = (((Nx-1)*Nz+1):(Nx*Nz))'; % c. Derecho
52 c_izq = (1:Nz)'; % c. Izquierdo
53 c_M = [c_inf; c_sup; c_izq; c_der]; % Matriz contornos
54 sub_c_sup = ((2*Nz-1):Nz:(Nx-1)*Nz-1))'; % Subc. Superior
55 sub_c_inf = ((Nz+2):Nz:(Nz*(Nx-2)+2))'; % Subc. Inferior
56 sub_c_izq = ((Nz+2):(2*Nz-1))'; % Subc. Izquierdo
57 sub_c_der = ((Nz*(Nx-2)+2):(Nz*(Nx-1)-1))'; % Subc. Derecho
58 sub_c_M = [sub_c_inf; sub_c_sup; sub_c_izq; sub_c_der]; % Matriz subcontornos
59 m_sup = length(c_sup); m_inf = length(c_inf);
60 m_der = length(c_der); m_izq = length(c_izq);
61
62 %%% Ecuaciones adimensionales %%%
63 % Asyst = [ APSI AT
64 % BPSI BT ]
65
66 APSI = DL-dt*Pr*DL2; APSI = spfull(APSI);
67 AT = dt*Pr*Dx; AT = spfull(AT);
68 BPSI = zeros(Nt, Nt); BPSI = spfull(BPSI);
69 BT = Id-dt*DL; BT = spfull(BT);

```

```

70
71 %%% Condiciones de contorno %%%
72 % T impuesta (nula) en C. Verticales y Horizontales
73 BT(c_izq,:) = Dx(c_der,:) - Dx(c_izq,:);
74 BT(c_der,:) = Dx2(c_der,:) - Dx2(c_izq,:);
75 BT(c_inf,:) = Id(c_inf,:);
76 BT(c_sup,:) = Id(c_sup,:);
77 BPSI(c_M,:) = 0;
78 % Psi impuesto en C. Verticales y Horizontales
79 APSI(c_izq,:) = Dx(c_der,:) - Dx(c_izq,:);
80 APSI(c_der,:) = Dx2(c_der,:) - Dx2(c_izq,:);
81 APSI(c_inf,:) = Id(c_inf,:);
82 APSI(c_sup,:) = Id(c_sup,:);
83 AT(c_M,:) = 0;
84 % Condición común a los SUBcontornos
85 AT(sub_c_M,:) = 0;
86 % Velocidades normales nulas en SUBcontornos superior e inferior
87 APSI(sub_c_sup,:) = Dz(sub_c_sup+1,:);
88 APSI(sub_c_inf,:) = Dz(sub_c_inf-1,:);
89 % Esfuerzo cortante en SUBcontornos izquierdo y derecho nulo
90 APSI(sub_c_izq,:) = Dx3(sub_c_izq-Nz,:) - Dx3(sub_c_der+Nz,:);
91 APSI(sub_c_der,:) = Dx4(sub_c_der+Nz,:) - Dx4(sub_c_izq-Nz,:);
92 % Dispersa o llena
93 APSI = spfull(APSI); AT = spfull(AT);
94 BPSI = spfull(BPSI); BT = spfull(BT);
95
96 %%% Construcción de la matriz del problema y factorización %%%
97 Asyst = [APSI AT; BPSI BT]; Asyst = spfull(Asyst);
98 [L_lu,U_lu,P_lu] = lu(Asyst);
99
100 u1 = 0;
101 n = length(Rav);
102 t = NaN(1e6,n);
103 Cx = NaN(1e6,n);
104 Cz = NaN(1e6,n);
105 tmax_x = NaN(1e6,n);
106 tmax_z = NaN(1e6,n);
107 umed = NaN(1e6,n);
108 vmed = NaN(1e6,n);
109 maxu = NaN(1e6,n);
110 maxv = NaN(1e6,n);
111 Psit = NaN(1e6,n);
112 CVPsi = NaN(1e6,n);
113 CVNu = NaN(1e6,n);
114 stdQ = NaN(1e6,n);
115 stdPsi = NaN(1e6,n);
116 flagC = NaN(n,1);
117 ntv = NaN(n,1);
118 periodo = NaN(n,1);
119 Qt = NaN(1e6,n);
120 Qt_s = NaN(1e6,n);
121 Qv = NaN(n,1);
122 Qv_s = NaN(n,1);
123 Qx = NaN(400,n);
124 Nux_s = NaN(5e3,400,n);
125 Nut = NaN(1e6,n);
126 Nut_s = NaN(1e6,n);
127 Nuv = NaN(n,1);
128 Nuv_s = NaN(n,1);
129 tf = NaN(n,1);
130 ntf = NaN(n,1);
131 for Ra = Rav
132 tic
133 u1 = u1 + 1;
134 %%% Solución de equilibrio %%%
135 Asyst_lp = DL;
136 Asyst_lp(c_M,:) = Id(c_M,:); Asyst_lp = spfull(Asyst_lp);
137 T2 = -Ra/2; % Temp adimensional en c.superior
138 T1 = Ra/2; % Temp adimensional en c.inferior
139 FBC_T_eq = zeros(Nt,1);
140 FBC_T_eq(c_inf,1) = T1*ones(m_inf,1);
141 FBC_T_eq(c_sup,1) = T2*ones(m_sup,1);
142 Teq = Asyst_lp\FBC_T_eq;
143 %%% Perturbación inicial %%%
144 Psiper = 1e-8*(Xmat.^2).*(max(max(Xmat))-...
145 Xmat).^2).*(Zmat.^2).*(max(max(Zmat))-Zmat).^2);
146 Tper = 1e-8*sin(3*pi*Zmat);
147 Psiper = reshape(Psiper,Nt,1);
148 Tper = reshape(Tper,Nt,1);
149 %%% Solución en el instante inicial %%%
150 T = Teq + Tper;
151 Psi = Psiper;
152 %%% Condiciones de contorno %%%
153 FBC_T = ones(Nt,1);
154 FBC_T(c_M,1) = 0;
155 FBC_Psi = ones(Nt,1);
156 FBC_Psi(c_M,1) = 0;
157 FBC_Psi(sub_c_M,1) = 0;
158 Psi_BC = zeros(Nt,1);
159 Psi_BC(c_sup,1) = Psisup;
160 dfxmat = diff(xmat)';
161 dfzmat = diff(zmat);
162
163 nt = 0;
164 criterio = 0;
165 while criterio == 0

```



```

166     nt = nt + 1;
167     if nt == 1
168         t(1,u1) = 0;
169     else
170         t(nt,u1) = t(nt-1,u1) + dt;
171     end
172     u_v = Dz*Psi; % v_x (vector)
173     v_v = -Dx*Psi; % v_z (vector)
174     bNLPsi(1:Nt,1) = (-u_v.*(DxDL*Psi)-v_v.*(DzDL*Psi))*dt;
175     bNLT(1:Nt,1) = (-u_v.*(Dx*T)-v_v.*(Dz*T))*dt;
176     bLinPsi(1:Nt,1) = DL*Psi;
177     bLinT(1:Nt,1) = T;
178     bsyst(1:2*Nt,1) = [FBC_Psi.*(bNLPsi+bLinPsi)+(1-FBC_Psi).*(Psi_BC); ...
179                     FBC_T.*(bNLT+bLinT)+(1-FBC_T).*FBC_T_eq];
180
181     %%% Resolución del sistema %%%
182     bn = U_lu\ (L_lu\ (P_lu*bsyst));
183
184     Psi = bn(1:Nt,1);
185     T = bn((Nt+1):end,1);
186
187     %%% Se comprueba que el paso de tiempo es adecuado %%%
188     umat = reshape(u_v,Nz,Nx);
189     vmat = reshape(v_v,Nz,Nx);
190     Courant_x = dt*abs(umat(:,2:end))./dfxmat;
191     Cx(nt,u1) = max(max(Courant_x));
192     Courant_z = dt*abs(vmat(2:end,:))./dfzmat;
193     Cz(nt,u1) = max(max(Courant_z));
194     tmax_x(nt,u1) = min(min(0.9*dfxmat./abs(umat(:,2:end))));
195     maxu(nt,u1) = max(max(abs(umat)));
196     umed(nt,u1) = trapz(zch,trapz(xch,abs(umat),2))/AR/1;
197     tmax_z(nt,u1) = min(min(0.9*dfzmat./abs(vmat(2:end,:))));
198     maxv(nt,u1) = max(max(abs(vmat)));
199     vmed(nt,u1) = trapz(zch,trapz(xch,abs(vmat),2))/AR/1;
200
201     %% Resultados
202     Psimat = reshape(Psi,Nz,Nx);
203     Tmat = reshape(T,Nz,Nx);
204     Tfis = T2fis*(2*g*H^3+Ra*nu*alpha)/(2*g*H^3-Ra*nu*alpha); % Kelvin
205     T0 = (Tfis+T2fis)/2;
206     Tfis = T0+nu*alpha*T0/g/H^3*T;
207     gradQ = [Dx*Tfis, Dz*Tfis];
208     Qnorm = dot(gradQ',N')';
209     Qfield = reshape(Qnorm,Nz,Nx);
210     Qx(1:Nx,u1) = Qfield(1,1:end);
211     Q = abs(dot(DS,Qx(1:Nx,u1)));
212     Qt(nt,u1) = Q;
213     Qt_s(nt,u1) = Q/sum(DS);
214     Nut(nt,u1) = Q/(Tfis-T2fis)/AR/H;
215     Nut_s(nt,u1) = Qt_s(nt,u1)/(Tfis-T2fis)/H;
216     Nux_s(rem(nt-1,5000)+1,1:Nx,u1) = Qx(1:Nx,u1)/(Tfis-T2fis)/H;
217     Psit(nt,u1) = trapz(zch,trapz(xch,abs(Psimat),2));
218
219     %%% Criterio de convergencia %%%
220     if t(nt,u1) > t_th1
221         m1 = (nt-t_th2/dt):nt;
222         CVPsi(nt,u1) = std(Psit(m1,u1))/mean(Psit(m1,u1));
223         CVNu(nt,u1) = std(Nut(m1,u1))/mean(Nut(m1,u1));
224         stdQ(nt,u1) = std(Nut(m1,u1));
225         stdPsi(nt,u1) = std(Psit(m1,u1));
226     end
227     if t(nt,u1) > t_th1
228         if (abs(CVPsi((nt-t_th2/dt):nt,u1)) < 1 | Psit((nt-t_th2/dt):nt,u1) < 1e-3 & abs(CVNu((nt-t_th2/dt):nt,u1)) < 0.5
229             criterio = 1;
230             % %
231             plot(t(1:nt,u1),maxv(1:nt,u1))
232             figure(u1)
233             subplot(4,1,1),plot(1:nt,Psit(1:nt,u1))
234             subplot(4,1,2),plot(1:nt,Qt(1:nt,u1))
235             subplot(4,1,3),plot(1:nt,CVPsi(1:nt,u1))
236             subplot(4,1,4),plot(1:nt,CVNu(1:nt,u1)),pause(1e-15)
237             if Psit((nt-t_th2/dt):nt,u1) < 1
238                 disp('Velocidades practicamente cero')
239             else
240                 disp('Estacionario')
241             end
242         end
243     end
244
245     %%% Courant %%%
246     if Cx(nt,u1) >= 1 || Cz(nt,u1) >= 1 || isnan(Cx(nt,u1)) || isnan(Cz(nt,u1)) || isnan(Psit(nt,u1))
247         disp(['Cx = ',num2str(Cx(nt,u1))])
248         disp(['Cz = ',num2str(Cz(nt,u1))])
249         flagC(u1) = 1;
250         disp(num2str(flagC(u1)))
251         criterio = 1;
252     end
253
254     %%% Rollos de Bénard %%%
255     aux2 = NaN(size(Psimat)+2);
256     aux2(2:(end-1),2:(end-1)) = abs(Psimat);
257     max1 = islocalmax(aux2,1,'FlatSelection','center');
258     max2 = islocalmax(aux2,2,'FlatSelection','center');
259     max1 = max1(2:(end-1),2:(end-1));
260     max2 = max2(2:(end-1),2:(end-1));
261     pos1 = find(max1==1);
262     pos2 = find(max2==1);
263     cnt = intersect(pos1,pos2);

```

```

262 % % n_B = length(cnt);
263
264 %%% Representación de los campos %%%
265 if criterio == 1
266 figure(101)
267 subplot(2,1,1)
268 contour(xmat, zmat, Tmat, 'ShowText', 'on')
269 hold on,
270 plot(xch, zbv, 'k.-')
271 hold off
272 title('Campo de temperatura adimensional [-]'), xlabel('x [-]'), ylabel('z [-]')
273 subplot(2,1,2)
274 contour(xmat, zmat, Psimat, 'ShowText', 'on')
275 hold on,
276 plot(xch, zbv, 'k.-')
277 hold off
278 title('Función de corriente adimensional [-]'), xlabel('x [-]'), ylabel('z [-]')
279 pause(1e-16)
280 print('-depsc2', ['TPsigrad_Psisup_', num2str(Psisup), '_AR_', num2str(AR), '__', ...
281 num2str(t(nt, u1)), '__', num2str(Ra), '_lambdab_', num2str(lambdab), ...
282 '_Ab_', num2str(Ab), '__', num2str(criterio), '_eps'])
283 saveas('figure(101)', ['TPsigrad_Psisup_', num2str(Psisup), '_AR_', num2str(AR), '__', ...
284 num2str(t(nt, u1)), '__', num2str(Ra), '_lambdab_', num2str(lambdab), ...
285 '_Ab_', num2str(Ab), '__', num2str(criterio), '_fig'])
286 pause(1e-16)
287 figure(201)
288 xlabel('x [-]'), ylabel('z [-]')
289 quiver(xmat, zmat, umat, vmat, 'b')
290 axis([-0.1 AR+0.1 -0.1 1.1]), xlabel('x [-]'), ylabel('z [-]')
291 title('Campo de velocidades adimensional [-]'), hold off
292 print('-depsc2', ['vgrad_Psisup_', num2str(Psisup), '_AR_', num2str(AR), '__', ...
293 num2str(t(nt, u1)), '__', num2str(Ra), '_lambdab_', num2str(lambdab), ...
294 '_Ab_', num2str(Ab), '__', num2str(criterio), '_eps'])
295 saveas('figure(201)', ['vgrad_Psisup_', num2str(Psisup), '_AR_', num2str(AR), '__', ...
296 num2str(t(nt, u1)), '__', num2str(Ra), '_lambdab_', num2str(lambdab), ...
297 '_Ab_', num2str(Ab), '__', num2str(criterio), '_fig'])
298 pause(1e-16)
299 figure(301)
300 subplot(2,1,1)
301 plot(t(1:nt, u1), Nut(1:nt, u1), 'b'), hold on
302 xlabel('t [-]'), ylabel('Nu [-]')
303 subplot(2,1,2)
304 plot(xch, Nuv_s(rem(nt-1, 5000)+1, 1:Nx, u1), 'b')
305 hold on,
306 plot(xch, zbv, 'k.-')
307 hold off
308 xlabel('x [-]'), ylabel('Nu [-]')
309 print('-depsc2', ['Nu_Psisup_', num2str(Psisup), '_AR_', num2str(AR), '__', ...
310 num2str(t(nt, u1)), '__', num2str(Ra), '_lambdab_', num2str(lambdab), ...
311 '_Ab_', num2str(Ab), '__', num2str(criterio), '_eps'])
312 saveas('figure(301)', ['Nu_Psisup_', num2str(Psisup), '_AR_', num2str(AR), '__', ...
313 num2str(t(nt, u1)), '__', num2str(Ra), '_lambdab_', num2str(lambdab), ...
314 '_Ab_', num2str(Ab), '__', num2str(criterio), '_fig'])
315 pause(1e-16)
316 close all
317 end
318
319 end
320 close all
321 ntv(u1, 1) = nt;
322 tf(u1, 1) = t(nt, u1);
323 ntf(u1, 1) = nt;
324 disp(num2str(t(nt, u1)))
325 if flagC(u1) == 1
326 [local_min_Nu, t_local_min_Nu] = ...
327 findpeaks(-Nut((nt-t_th2/dt):nt, u1), t((nt-t_th2/dt):nt, u1));
328 [unicos, ~, ic] = unique(-round(local_min_Nu, 4));
329 unicos = -unicos;
330 a_counts = accumarray(ic, 1);
331 [~, ind_min_counts] = max(a_counts);
332 t_periodico = t_local_min_Nu(round(local_min_Nu, 4) == unicos(ind_min_counts));
333 if isempty(local_min_Nu)
334 periodo(u1, 1) = NaN;
335 else
336 if abs(std(local_min_Nu)/mean(local_min_Nu)) <= 0.5
337 periodo(u1, 1) = mean(diff(t_periodico));
338 tT = (nt-2*ceil(periodo(u1, 1)/dt)):nt;
339 Nuv(u1, 1) = mean(Qt(tT, u1))/(T1fis-T2fis)/H;
340 Nuv_s(u1, 1) = mean(Qt_s(tT, u1))/(T1fis-T2fis)/H;
341 Qv(u1, 1) = mean(Qt(tT, u1));
342 Qv_s(u1, 1) = mean(Qt_s(tT, u1));
343 else
344 periodo(u1, 1) = NaN;
345 Qv(u1, 1) = Q;
346 Qv_s(u1, 1) = Q/sum(DS);
347 Nuv(u1, 1) = Q/(T1fis-T2fis)/H;
348 Nuv_s(u1, 1) = Q/(T1fis-T2fis)/H/sum(DS);
349 end
350 end
351 end
352
353 end
354
355 end

```

## B.6 Funciones auxiliares

Listing B.18 Matriz dispersa o llena.

```
1 % Comprueba las dimensiones y componentes distintas de cero de una matriz
2 % para evaluar que ocupa menos memoria, si guardarla como dispersa o como
3 % matriz llena
4 % N debe ser una matriz cuadrada
5 function[M] = spfull(N)
6 %find obtiene los índices de los elementos distintos de cero de una matriz
7 %en un vector
8 [m,~] = size(N);
9 s = length(find(N));
10 if s > (m^2-m-1)/2 && issparse(N)==1
11     M = full(N);
12 elseif s < (m^2-m-1)/2 && issparse(N)~=1
13     M = sparse(N);
14 else
15     M = N;
16 end
17
18 end
```



# Índice de Figuras

---

1.1.	Celdas de Bénard [1]	3
1.2.	Mecanismos físicos que influyen en la convección natural	3
1.3.	Modelo de tres celdas convectivas [3]	4
1.4.	Brisas de costa [3]	4
1.5.	Brisas de montaña [3]	5
1.6.	Convección en el manto terrestre [4]	5
1.7.	Álabe de turbina refrigerado por convección [5]	5
3.1.	Dominio unidimensional	16
3.2.	Numeración de nodos con índice único	24
3.3.	Ejemplo de malla transformada	27
3.4.	Transformación del dominio	29
4.1.	Esquema básico de convección ( $T_1 > T_2$ )	34
4.2.	Valores experimentales del número de Nusselt en experimentos diferentes de convección de Bénard como función del número de Rayleigh para diferentes sustancias [7]	35
4.3.	Condiciones de contorno para análisis de sensibilidad de malla y método para $C1$ y $C2$	41
4.4.	Rayleigh crítico en caso de placa en el contorno superior resuelto con nodos de Chebyshev y $N_z = cte$	42
4.5.	Rayleigh crítico en caso de contorno superior libre resuelto con nodos de Chebyshev y $N_z = cte$	43
4.6.	Rayleigh crítico para $C1$ y $C2$ calculado con diferencias finitas, distribución de nodos de Chebyshev y 5 nodos circundantes	45
4.7.	Rayleigh crítico para $C1$ y $C2$ calculado con diferencias finitas tras aumentar el número de nodos, distribución de nodos de Chebyshev y 5 nodos circundantes	46
4.8.	Rayleigh crítico para $C1$ y $C2$ calculado con diferencias finitas, distribución de nodos uniforme y 7 nodos circundantes	47
4.9.	Configuraciones de métodos y mallas	48
4.10.	Condiciones de contorno de Rayleigh-Bénard	50
4.11.	Rayleigh crítico para condiciones de periodicidad en contornos verticales ( $C1$ )	51
4.12.	Rayleigh crítico para condiciones de periodicidad en contornos verticales ( $C2$ )	52
4.13.	Autovectores asociados a $Ra_c$ para condiciones de periodicidad en contornos verticales ( $C1$ )	53
4.14.	Condiciones de contorno para la validación del cambio de variable	54
4.15.	Autovectores asociados a $Ra_c$ para geometrías con contornos laterales no rectos, usando diferencias finitas	55

5.1.	Nusselt en función del número de Rayleigh para el contorno inferior para fluido entre dos placas con $AR = 40$ resuelto por diferencias finitas	62
5.2.	Velocidad vertical máxima en función del número de Rayleigh, para $AR = 1$ y $AR = 5$ , y diferentes pasos temporales y métodos	64
5.3.	Velocidad vertical máxima para $Ra = 6000$ , $AR = 1$ y $AR = 5$ , y diferentes pasos temporales y métodos	65
5.4.	Condiciones de contorno del análisis lineal para convección forzada y contornos rectos	68
5.5.	Dependencia de Rayleigh crítico en función del caudal para $AR = 4$ y $AR = 8$	69
5.6.	Rayleigh crítico en función de la relación de aspecto para diferentes caudales adimensionales (contornos rectos)	70
5.7.	Condiciones de contorno para convección forzada y contornos rectos	71
5.8.	Desviación media de la función de corriente respecto a la de Rayleigh cero	72
5.9.	Número de Nusselt en el contorno inferior para diferentes números de Rayleigh y caudal adimensional, siendo $AR = 4$ (contornos rectos)	76
5.10.	Nusselt en contorno inferior (sinusoidal con $\hat{\lambda} = 4$ y dos amplitudes $A$ ) a lo largo del tiempo en función de $\hat{\Psi}_{sup}$ para dos números de Rayleigh	78
5.11.	Nusselt en contorno inferior (sinusoidal con $\hat{\lambda} = 2$ y dos amplitudes $A$ ) a lo largo del tiempo en función de $\hat{\Psi}_{sup}$ para dos números de Rayleigh	79
5.12.	Nusselt en contorno inferior (definido por función gaussiana con dos amplitudes $A$ ) a lo largo del tiempo en función de $\hat{\Psi}_{sup}$ para dos números de Rayleigh	79

# Índice de Tablas

---

4.1.	Casos simulados para diferentes combinaciones de mallas y método de obtención de derivadas	48
4.2.	Rayleigh crítico y tiempos de computación [s] para $C1$ y diferentes métodos	49
4.3.	Rayleigh crítico y tiempos de computación [s] para $C1$ y diferentes métodos	50
4.4.	Comparación de número de Rayleigh crítico para contornos laterales no rectos	55
5.1.	Comparación de Rayleigh crítico mediante resolución en el tiempo y análisis de estabilidad para $C1$	62
5.2.	Campos de temperatura y función de corriente adimensionales para $AR = 5$ (contornos rectos)	66
5.3.	Desviación de función de corriente adimensional respecto a $Ra = 0$ en un instante dado, para $Ra = 1800$ y $3000$ y $\hat{\psi} = 0, 2$ y $5$ , siendo $AR = 4$ (contornos rectos)	73
5.4.	Campos de función de corriente y temperatura adimensionales a lo largo de un periodo, para $Ra = 3000$ , $\hat{\psi} = 2.5$ y $AR = 4$ (contornos rectos)	74
5.5.	Distribución de Nusselt en el contorno inferior para diferentes caudales adimensionales, números de Rayleigh y $AR = 4$ (contornos rectos)	75
5.6.	Temperatura y campo vectorial de la velocidad (adimensionales) en un instante dado, para $AR = 4$ y diferentes caudales adimensionales y números de Rayleigh (contornos rectos)	77
5.7.	Distribución de Nusselt en el contorno inferior (sinusoidal con $\hat{\lambda} = 4$ y dos amplitudes $\hat{A}$ ) para $\hat{\psi} = 1$ y dos números de Rayleigh	80
5.8.	Velocidades y temperaturas adimensionales para contorno inferior sinusoidal con $\hat{\lambda} = 4$ y $\hat{A} = 0.1$ , dos caudales adimensionales y dos números de Rayleigh	81
5.9.	Velocidades y temperaturas adimensionales para contorno basado en función gaussiana, $\hat{A} = 0.1$ , dos caudales adimensionales y dos números de Rayleigh	82





# Índice de Códigos

---

B.1.	Definición de la geometría con contornos horizontales no rectos	93
B.2.	Definición de la geometría con contornos verticales no rectos	94
B.3.	Matrices de derivadas con contornos horizontales no rectos	95
B.4.	Matrices de derivadas con contornos verticales no rectos	98
B.5.	Derivadas con algoritmo de Fornberg ([20])	101
B.6.	Derivadas con diferencias finitas	102
B.7.	Derivadas con método de colocación con nodos de Chebyshev	103
B.8.	Estudio lineal respecto a número de nodos y relación de aspecto para fluido entre dos placas horizontales	104
B.9.	Función análisis lineal asociada a B.8	105
B.10.	Estudio lineal para condiciones laterales periódicas y contorno superior libre	107
B.11.	Función análisis lineal asociada a B.10	109
B.12.	Estudio lineal para contornos laterales no rectos	111
B.13.	Función análisis lineal asociada a B.12	112
B.14.	Resolución completa en el tiempo del sistema	115
B.15.	Función resolución completa asociada a B.14	115
B.16.	Resolución completa en el tiempo del sistema para convección forzada	118
B.17.	Función resolución completa asociada a B.16	119
B.18.	Matriz dispersa o llena	123



# Bibliografía

---

- [1] J. Wesfreid. Henri Bénard: Thermal convection and vortex shedding. *Comptes Rendus Mécanique*, 06 2017.
- [2] Lord Rayleigh O.M. F.R.S. On convection currents in a horizontal layer of fluid, when the higher temperature is on the under side. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 32(192):529–546, 1916.
- [3] I. Zuñiga López y E. Crespo del Arco. *Meteorología y climatología*. UNED, 2010.
- [4] E.J. Tarbuck. *Ciencias de la tierra: una introducción a la geología física*. Pearson Educación, 2005.
- [5] J.L. Kerrebrock. *Aircraft Engines and Gas Turbines*. Pe Men, 1977.
- [6] P. J. Ruiz Contreras. *El Método de Colocación para el problema de convección de Rayleigh-Bernard [Recurso electrónico]: Proyecto Fin de Carrera*. 2013.
- [7] A. Barrero Ripoll y M. Pérez-Saborid Sánchez-Pastor. *Fundamentos y Aplicaciones de la Mecánica de Fluidos*. MCGraw-Hill, 2005.
- [8] A. Oberbeck. Ueber die wärmeleitung der flüssigkeiten bei berücksichtigung der strömungen infolge von temperaturdifferenzen. *Annalen der Physik*, 243(6):271–292, 1879.
- [9] A. Oberbeck. Über die bewegungserscheinungen der atmosphäre. *Sitz-Ber. Akad. Wiss., Berlin*, 383:1129–1138, 1888.
- [10] J. Boussinesq. *Theorie analytique de la Chaleur*. Gauthier-Villars, 1903.
- [11] H. Jeffreys. The instability of a compressible fluid heated below. *Mathematical Proceedings of the Cambridge Philosophical Society*, 26(2):170–172, 1930.
- [12] E. A. Spiegel and G. Veronis. On the boussinesq approximation for a compressible fluid. *Astrophys. J.*, 131:442–447, 03 1960.
- [13] Y. Ogura and N. A. Phillips. Scale analysis of deep and shallow convection in the atmosphere. *J. Atmos. Sci.*, 19, 03 1962.
- [14] G. K. Batchelor. The condition for dynamical similarity of motions of a frictionless perfect gas atmosphere. *Quarterly Journal of the Royal Meteorological Society*, 79, 1953.
- [15] J. G. Charney and Y. Ogura. A numerical model for thermal convection in the atmosphere. *Journal of the Meteorological Society of Japan. Ser. II*, 38(6):19a–19a, 1960.
- [16] J. M. Mihaljan. A rigorous exposition of the boussinesq approximations applicable to a thin layer of fluid. *The Astrophysical Journal*, 136:1126, 10 1962.
- [17] W. V. R. Malkus and M. C. Thayer. Notes on the 1966 summer study program in geophysical fluid dynamics at the woods hole oceanographic institution. 01 2019.

- [18] D. Gray and A. Giorgini. The validity of the boussinesq approximation for liquids and gases. *International Journal of Heat and Mass Transfer*, 19:545–551, 05 1976.
- [19] M. Ruzicka K. R. Rajagopal and A. R. Srinivasa. On the oberbeck-boussinesq approximation. *Mathematical Models and Methods in Applied Sciences*, 06, 11 2011.
- [20] B. Fornberg. *A Practical Guide to Pseudospectral Methods*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1996.
- [21] S. Ruuth U. Ascher and R. Spiteri. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25:151–167, 11 1997.
- [22] H. M. Park and D. H. Ryu. A solution method of nonlinear convective stability problems in finite domains. *Journal of Computational Physics*, 170:141–160, 06 2001.
- [23] H. M. Park and Y. M. Heo. Rayleigh–bénard convection in two-dimensional arbitrary finite domains. *International Journal of Thermal Sciences*, 45:697–705, 07 2006.
- [24] Andrew Roberts. [urlhttps://es.mathworks.com/matlabcentral/fileexchange/17582-ncquiverref](https://es.mathworks.com/matlabcentral/fileexchange/17582-ncquiverref).
- [25] M. Satoh. *Atmospheric Circulation Dynamics and General Circulation Models / M. Satoh*. 01 2004.
- [26] C. Y. Chow. *An Introduction to Computational Fluid Mechanics by Example*. 04 2011.
- [27] Y. L. Lin. *Mesoscale Dynamics*. 08 2013.
- [28] R. E. Kelly and D. Pal. Thermal convection with spatially periodic boundary conditions - resonant wavelength excitation. *Journal of Fluid Mechanics*, 86:433 – 456, 06 1978.
- [29] A. Passerini and G. Thäter. Boussinesq-type approximation for second-grade fluids. *International Journal of Non-Linear Mechanics*, 40:821–831, 07 2005.
- [30] M. Ruzicka Y. Kagei and G. Thäter. Natural convection with dissipative heating. *Communications in Mathematical Physics*, 214:287–313, 11 2000.
- [31] K. Krettenauer and U. Schumann. Direct numerical simulations of thermal convection over a wavy surface. *Meteorological and Atmospheric Physics*, 41:165–179, 09 1989.