

UN METODO DE ELEMENTOS FINITOS INCONDICIONALMENTE ESTABLE EN NORMA UNIFORME PARA RESOLVER LAS ECUACIONES DE EULER 2D

TOMAS CHACON REBOLLO

^e
IBRAHIM BLESS RANERO*

Departamento de Análisis Matemático, Universidad de Sevilla

RESUMEN

Este trabajo presenta dos algoritmos de tipo transporte e interpolación con Elementos Finitos para la resolución numérica de las ecuaciones de Euler para flujos incompresibles bidimensionales en todo el espacio \mathbb{R}^2 . En la primera versión de nuestro algoritmo, la vorticidad es discretizada mediante elementos finitos triangulares de primer grado. En la segunda, mediante elementos finitos triangulares de segundo grado. La velocidad se obtiene en ambos casos calculando exactamente el producto de convolución del núcleo de Biot-Savart con una aproximación lineal a trozos sobre cada triángulo de la vorticidad discreta. Se prueba que el primer algoritmo es incondicionalmente estable y convergente con una precisión de primer orden, en norma uniforme. Sin embargo, en la práctica la precisión alcanzada resulta escasa, debido a la difusión numérica introducida en el paso de interpolación. En el caso del segundo algoritmo, los ensayos numéricos muestran un notable incremento de la precisión, incluso para tiempos largos. Sin embargo, en este caso el algoritmo deja de ser estable en norma uniforme.

SUMMARY

We introduce two Finite Element transport-interpolation algorithms to solve the two-dimensional Euler equations in the whole \mathbb{R}^2 . In the first of these algorithms, the vorticity is discretized with triangular finite elements of degree one, and of degree two in the second one. The velocity is computed by convolution of the Biot-Savart kernel with a piecewise affine interpolate of the vorticity. We prove that the first algorithm is unconditionally stable in uniform norm, with first order accuracy. However, in practice its precision is rather low, due to the numerical diffusion introduced in the interpolation step. The second algorithm is shown numerically to produce a remarkable increase of precision, even for long integration times. However, in this case the algorithm is no longer stable in uniform norm.

* Investigación financiada parcialmente por Proyecto DGICYT 9B91-0619

Recibido: Octubre 1991

INTRODUCCION

En este trabajo, nos interesaremos por la resolución numérica de fluidos bidimensionales incompresibles y no viscosos. Estos fluidos están gobernados por las ecuaciones de Euler, que se pueden escribir como una ecuación de convección pura, para la vorticidad del flujo. Los problemas de convección no lineal aparecen frecuentemente en ingeniería. Por ello, resulta de interés el desarrollo de algoritmos de resolución numérica de los mismos.

La ecuación de convección se reduce a una familia de Ecuaciones Diferenciales Ordinarias a lo largo de las curvas características del fluido. Esto ha dado origen a los algoritmos "Lagrangianos" para las ecuaciones de Euler 2D, basados en la idea de calcular la vorticidad a lo largo de las curvas características del fluido. El primero de estos métodos fue el de "punto-vórtice", introducido por Rosenhead [18], que se basa en una discretización de la vorticidad como suma de masas de Dirac y tiene como principal ventaja, la de ser no disipativo, aunque no es estable para largos períodos de tiempo (Cf.⁹).

Al principio de los años 70, Chorin⁷, Kuwahara y Takami introdujeron la idea de discretizar la vorticidad, sustituyendo las masas de Dirac por funciones regulares que las aproximen, para estabilizar el método. Esto dio origen a métodos con mayor precisión, no disipativos y estables, que han sido estudiados ampliamente en los pasados 15 años (Cf.^{13,14,15,3,8,17}). Se trata de los métodos llamados "de Burbuja -Vórtice".

Por otra parte, también es posible construir algoritmos lagrangianos con Elementos Finitos. Bardos, Bercovier y Pironneau introdujeron en [2] un algoritmo para resolver las ecuaciones de Euler formuladas en términos de la función de corriente asociada a la vorticidad, basándose en una discretización constante a trozos de la vorticidad sobre una triangulación. En este algoritmo, la función de corriente se aproxima mediante elementos finitos conformes de primer grado. De esta forma, la velocidad discreta es constante a trozos, pero la componente normal es continua a través de los lados de la triangulación.

Este método posee la ventaja fundamental de los métodos de vórtices, al ser no disipativo. Además, es uniformemente convergente y permite la manipulación de condiciones de contorno. Sin embargo, el cálculo de las curvas características, presenta algunas dificultades computacionales, que se deben al hecho de que la velocidad no es globalmente continua.

Un algoritmo puramente lagrangiano con Elementos Finitos fue a continuación introducido en [6] por Chacón y Hou. Este algoritmo está basado en una discretización afín a trozos de la vorticidad, calculando la velocidad directamente, mediante la convolución de la vorticidad discreta con el núcleo de Biot-Savart y trasladando los vértices de la triangulación a lo largo de las curvas características del fluido, en cada paso de tiempo.

Este algoritmo es uniformemente estable, para un cierto período de tiempo, si las curvas características son calculadas con un algoritmo de segundo orden. Sin embargo, para obtener la estabilidad en largos períodos de tiempo, es necesario el uso de técnicas de remallado o un paso de tiempo muy pequeño.

Nuestro propósito en este trabajo es el de introducir una versión de tipo "Vortex-

in-Cell” del algoritmo Chacón-Hou citado arriba. La malla móvil utilizada en éste para discretizar la vorticidad es reemplazada por una malla fija, siendo necesario interpolar la vorticidad tras cada paso de tiempo. Esto produce una notable mejora de las propiedades de estabilidad del método, producida fundamentalmente por la introducción del paso de interpolación.

En la sección titulada Descripción de los Algoritmos introduciremos un primer algoritmo que utiliza esta técnica. La velocidad es calculada de la misma manera que en [6], mediante la convolución de la vorticidad discreta con el núcleo de Biot-Savart. También se introduce una variante de este algoritmo, en la cual la velocidad es calculada mediante la convolución del núcleo de Biot-Savart con un interpolado constante a trozos de la vorticidad.

En la sección titulada Análisis de Convergencia analizaremos las propiedades de convergencia y estabilidad de nuestros algoritmos. Probaremos que bajo suposiciones razonables para el operador de interpolación, ambas versiones del primer algoritmo son incondicionalmente estables y convergentes, con exactitud de primer orden en norma uniforme, para todo intervalo de tiempo finito.

La sección titulada Operadores de Interpolación está dedicada a describir algunos operadores de interpolación que cumplen las hipótesis requeridas para asegurar la convergencia. Probaremos que las interpolaciones “puntual” y “promediada” son buenas elecciones para este propósito.

En la sección titulada Propiedades de Conservación analizaremos las propiedades de conservación de nuestros algoritmos. El paso de interpolación que introducimos para calcular la vorticidad, produce un cierto aumento de la difusión numérica, debido a que los algoritmos de transporte e interpolación que conocemos no conservan el área. Daremos una demostración directa de que nuestros algoritmos conservan el área, con un error de segundo orden. Esta demostración se basa en el hecho de que las velocidades discretas son exactamente de divergencia nula.

En la sección titulada Ejemplos Numéricos, mostraremos algunos ensayos numéricos en un problema con solución analítica conocida. Este ejemplo muestra una buena correspondencia entre las predicciones teóricas y los resultados numéricos. Sin embargo, existe un alto nivel de difusión numérica.

Por último, en la sección titulada Una versión de Segundo Orden describiremos la segunda versión de nuestro algoritmo. En ella, la vorticidad se aproxima mediante elementos finitos triangulares de segundo grado. Las curvas características se discretizan hacia atrás en tiempo mediante un método no estándar de segundo orden. La velocidad se calcula convolucionando el núcleo de Biot-Savart con una aproximación afín a trozos de la vorticidad. El incremento de cálculo requerido se ve compensado por una técnica de cálculo rápido de la velocidad, que es la etapa más costosa de nuestros algoritmos. Mostraremos finalmente un ejemplo numérico en el que se aprecia una convergencia de segundo orden, manteniéndose el error prácticamente constante incluso para tiempos largos.

PLANTEAMIENTO DEL PROBLEMA

Nuestro propósito es resolver numéricamente las ecuaciones de Euler 2D para flujos incompresibles en todo el espacio \mathbb{R}^2

$$\left. \begin{aligned} u_{,t} + u \cdot \nabla u + \nabla p &= 0, & \nabla \cdot u &= 0 & \text{en } \mathbb{R}^2 \times]0, T[, \\ u(x, 0) &= u_0(x) & \text{en } \mathbb{R}^2, \\ \lim_{|x| \rightarrow \infty} u(x, t) &= 0. \end{aligned} \right\}$$

Aquí, $u(x, t)$ y $p(x, t)$ representan el campo de velocidad y la presión del fluido respectivamente, en el punto $x \in \mathbb{R}^2$ y en el instante t , $u_0(x)$ es un campo de velocidad inicial dado. Además, $[0, T]$ es el intervalo de tiempo durante el que analizamos el comportamiento del fluido y $|\cdot|$ denota la norma l^∞ en \mathbb{R}^2 .

Es conocido que el problema anterior es equivalente a la formulación “velocidad-vorticidad” de las ecuaciones de Euler con ausencia de frontera finita⁸:

$$\left. \begin{aligned} w_{,t} + u \cdot \nabla w &= 0 & \text{en } \mathbb{R}^2 \times]0, T[, \\ w(x, 0) &= w_0(x) = (\nabla \times u_0)(x), \\ u(x, t) &= \int_{\mathbb{R}^2} K(x - x') w(x', t) dx', \\ K(x, t) &= \frac{1}{2\pi} \frac{(-x_2, x_1)^t}{r^2}, \\ x &= (x_1, x_2), \quad r = \sqrt{x_1^2 + x_2^2}. \end{aligned} \right\} \tag{1}$$

La función $K(x, t)$ anterior es conocida como el núcleo 2D de Biot-Savart. Este núcleo tiene una singularidad en el origen, aunque es localmente integrable. Más aún, la convolución con K es un operador acotado de $L^p_{loc}(\mathbb{R}^2)$ en $W^{1,p}(\mathbb{R}^2)$, $1 \leq p < +\infty$. En lo que sigue, utilizaremos ampliamente esta propiedad.

Se conocen resultados de existencia y regularidad de las soluciones del problema 1, por ejemplo para w_0 regular¹². Consideraremos el caso $w_0 \in C^2(\mathbb{R}^2)$ con soporte compacto. Entonces existe una solución $w \in C^2(\mathbb{R}^2 \times [0, T])$, para cualquier $T > 0$. Además, esta solución tiene soporte compacto en un tiempo cualquiera, es decir, existe una constante $R_T > 0$ tal que

$$sop(w(., t)) \subset B_T = \{x \in \mathbb{R}^2, |x| \leq R_T, \}, \quad \forall t \in [0, T]. \tag{2}$$

Observemos, que si se da el campo de velocidad u , entonces la ecuación de transporte para w en (1) puede ser integrada explícitamente. Para cada $s \in [0, T]$ fijado, consideremos la ecuación de las curvas características $X(t; s, x)$ asociadas al campo de velocidad u con origen en s :

$$\frac{dX}{dt}(t; s, x) = u(X(t; s, x), t) \text{ para } t \in [0, T], \quad X(s; s, x) = x. \tag{3}$$

entonces w es constante a lo largo de la curva $t \in [0, T] \rightarrow (X(t; s, x), t) \in \mathbb{R}^2$:

$$w(X(t; s, x), t) = w(x, s). \tag{4}$$

En particular, resulta que

$$w(x, t) = w_0(X(0; t, x)).$$

Utilizaremos este hecho en lo que sigue para construir nuestros algoritmos.

DESCRIPCION DE LOS ALGORITMOS

Sea \mathcal{T}_h una triangulación de \mathbb{R}^2 , donde h es el diámetro máximo de los triángulos de \mathcal{T}_h . Denotemos por $\{\alpha_i\}_{i \geq 1}$ los vértices de los triángulos de \mathcal{T}_h . Definamos el espacio de Elementos Finitos lineales a trozos

$$V_h = \{v_h \in C^0(\mathbb{R}^2) : v_h|_T \in P_1, \forall T \in \mathcal{T}_h\},$$

donde P_k , para $k \geq 0$ entero, es el espacio de los polinomios sobre \mathbb{R}^2 de grado $\leq k$. Una función $v_h \in V_h$, está únicamente determinada por los valores $v_i = v_h(\alpha_i), \forall i \geq 1$ (Cf.5).

Consideremos un operador de interpolación lineal

$$v_h : C^0(\mathbb{R}^2) \rightarrow V_h. \tag{5}$$

Supondremos que existe $\lambda \geq 0$ independiente de h , tal que si

$$\text{sop}(v) \subset B(0, R), \quad \text{entonces } \text{sop}(v_h) \subset B(0, R + \lambda h). \tag{6}$$

Definamos ahora dos versiones de un primer algoritmo para discretizar las ecuaciones de Euler, basados en la descripción lagrangiana (4).

Consideremos $N \geq 1$, y $\Delta t = T/N$. Llamemos $t_n = n\Delta t, 0 \leq n \leq N$, y denotemos por \tilde{w}_h^n a una aproximación de $w(\cdot, t_n)$.

Algoritmo A1

1. Inicialización:

$$\tilde{w}_h^0 = r_h w_0.$$

2. Dada $\tilde{w}_h^n \in V_h$ con soporte compacto, definimos

(a) La velocidad discreta \tilde{u}_h^n por

$$\tilde{u}_h^n(x) = K * \tilde{w}_h^n = \int_{\mathbb{R}^2} K(x - x') \tilde{w}_h^n(x'), dx' \tag{7}$$

(b) La característica discreta \tilde{T}_h^n retrocediendo con el método de Euler

$$\tilde{T}_h^n(x) = x - \Delta t \tilde{u}_h^n(x),$$

y

(c) La vorticidad discreta \tilde{w}_h^{n+1} en el instante t_{n+1} mediante transporte + interpolación,

$$\tilde{w}_h^{n+1} = r_h \circ \tilde{w}_h^n \circ \tilde{T}_h^n.$$

□

Observaciones

1. \tilde{T}_h^n es obtenida discretizando (3) hacia atrás en tiempo mediante el método de Euler, con $s = t_{n+1}$.
2. El cálculo de la velocidad discreta mediante (7) se puede hacer de la siguiente manera:

$$\tilde{u}_h^n(x) = \sum_{T \in \mathcal{T}_h} \int_T K(x - x') \tilde{w}_h^n|_T(x') dx'$$

Denotemos por $\{\varphi_i\}$ la base canónica de V_h , dada por $\varphi_i(\alpha_j) = \delta_{ij}$. Dado un triángulo $T \in \mathcal{T}_h$, denotemos por $I(T)$ el conjunto de índices $i \in \mathbb{N}$ tales que α_i es un vértice de T . Como

$$\tilde{w}_h^n|_T = \sum_{i \in I(T)} w_i \varphi_i|_T,$$

las únicas integrales a calcular son

$$\int_T K(x - x') \varphi_i(x') dx', \text{ para } i \in I(T)$$

estas integrales pueden ser expresadas analíticamente como funciones de x y programadas directamente.

Esta definición es consistente:

Sea

$$R_n = \inf\{R \geq 0 \mid \text{sop}(\tilde{w}_h^n) \subset B(0, R)\}.$$

A partir de la definición de la velocidad discreta (7) tenemos

$$|\tilde{u}_h^n(x)| \leq |\tilde{w}_h^n|_\infty \int_{B(0, R_n)} |K(x')| dx' = R_n |\tilde{w}_h^n|_\infty, \forall x \in \mathbb{R}^2 \quad (8)$$

entonces

$$|\tilde{T}_h^n(x)| \geq |x| - \Delta t R_n |\tilde{w}_h^n|_\infty. \quad (9)$$

De aquí que

$$\text{sop}(\tilde{w}_h^n \circ \tilde{T}_h^n) \subset B(0, R_n^*), \text{ donde } R_n^* = R_n(1 + \Delta t |\tilde{w}_h^n|_\infty) \quad (10)$$

y

$$R_{n+1} \leq R_n^* + \lambda h, \text{ a partir de la hipótesis hecha sobre } r_h.$$

De esta manera, \tilde{w}_h^{n+1} tiene soporte compacto y \tilde{u}_h^{n+1} puede ser correctamente definida, ya que $K \in [L^1_{loc}(\mathbb{R}^2)]^2$.

La segunda versión de nuestro primer algoritmo permite hacer un cálculo más rápido de la velocidad discreta sin pérdida de orden de convergencia, como veremos. Consideremos el espacio de las funciones constantes a trozos sobre T :

$$H_h = \{v_h : \mathbb{R}^2 \rightarrow \mathbb{R}, v_h|_T \in P_0, \forall T \in \mathcal{T}_h\}$$

Consideremos también un operador de interpolación lineal

$$s_h : C^0(\mathbb{R}^2) \rightarrow H_h.$$

Supondremos que s_h verifica las dos propiedades siguientes:

$$\text{si } \text{sop}(v) \subset B(0, R), \text{ entonces } \text{sop}(s_h v) \subset B(0, R + \lambda h), \tag{11}$$

para alguna constante $\lambda > 0$; y

$$\text{si } v \in L^\infty(\mathbb{R}^2), \text{ entonces } s_h v \in L^\infty(\mathbb{R}^2). \tag{12}$$

ahora podemos describir la segunda versión de nuestro primer algoritmo:

Algoritmo A2

Todo como en el Algoritmo A1, con (7) reemplazado por

$$\tilde{u}_h^n(x) = K * (s_h \circ \tilde{w}_h^n) = \int_{\mathbb{R}^2} K(x - x')(s_h \tilde{w}_h^n)(x') dx' \tag{13}$$

□

Esta definición es consistente:

A partir de las propiedades (11) y (12), obtenemos

$$|\tilde{u}_h^n(x)| \leq |s_h \tilde{w}_h^n|_\infty \int_{B(0, R_n + \lambda h)} |K(x')| dx' = (R_n + \lambda h) |s_h \tilde{w}_h^n|_\infty$$

entonces, como en el Algoritmo A1, esto implica que \tilde{w}_h^{n+1} tiene soporte compacto.

Observación

Para calcular la velocidad mediante (13), la única integral que se necesita es

$$\int_T K(x - x') dx', \forall T \in \mathcal{T}_h$$

que se puede obtener analíticamente, de la misma forma que en el caso de la interpolación constante a trozos.

ANALISIS DE CONVERGENCIA

Probaremos ahora la convergencia uniforme de los algoritmos A1 y A2 a la solución de (1), bajo algunas suposiciones para los operadores de interpolación r_h y s_h .

TEOREMA 1 *Supongamos que v_h verifica las siguientes propiedades:*

$$\text{Si } v \in L^\infty(\mathbb{R}^2), \text{ entonces } r_h v \in L^\infty(\mathbb{R}^2) \text{ y } |r_h v|_\infty \leq |v|_\infty, \quad (14)$$

$$|v - r_h v|_\infty < \kappa |v|_{2,\infty} h^2 \quad \forall v \in C_0^2(\mathbb{R}^2), \quad (15)$$

donde κ es una constante numérica. Supongamos también que $h = \theta \Delta t$, donde θ es una constante numérica positiva del orden de la unidad.

Entonces, el Algoritmo A1 converge uniformemente a la solución de (1). Además, la discretización es de primer orden y se cumple la siguiente estimación para el error de discretización:

$$\max_{0 \leq n \leq N} |w(\cdot, t_n) - \tilde{w}_h^n|_\infty \leq C(\Delta t + \frac{h^2}{\Delta t}), \quad (16)$$

donde C es una constante que depende únicamente de T y w_0 .

Demostración

Analizaremos separadamente la estabilidad y la consistencia. Para simplificar la notación, definamos

$$\begin{aligned} w^n(x) &= w(x, t_n), \\ \hat{T}^n(x) &= x - \Delta t u(x, t_n), \\ \hat{w}_h^{n+1} &= r_h(w^n \circ \hat{T}^n). \end{aligned}$$

Observemos que la sucesión $\{\hat{w}_h^{n+1}\}_{n \geq 1}$ es obtenida cuando un paso del Algoritmo A1 es aplicado a la solución exacta de (1).

Estabilidad:

A partir de la propiedad (15) se tiene que

$$\begin{aligned} |\hat{w}_h^{n+1} - \tilde{w}_h^{n+1}|_\infty &= |r_h(w^n \circ \hat{T}^n - \tilde{w}_h^n \circ \tilde{T}_h^n)|_\infty \\ &\leq |w^n \circ \hat{T}^n - \tilde{w}_h^n \circ \tilde{T}_h^n|_\infty \end{aligned}$$

Entonces

$$|\hat{w}_h^{n+1} - \tilde{w}_h^{n+1}|_\infty \leq |w^n \circ \hat{T}^n - w_h^n \circ \tilde{T}_h^n|_\infty + |(w^n - w_h^n) \circ \tilde{T}_h^n|_\infty$$

Además, a partir de la propiedad (14) tenemos

$$|\tilde{w}_h^{n+1}|_\infty \leq |\tilde{w}_h^n \circ \tilde{T}_h^n|_\infty \leq |\tilde{w}_h^n|_\infty \leq |\tilde{w}_h^0|_\infty \leq |w_0|_\infty.$$

Entonces, de (10) se tiene que

$$R_{n+1} \leq R_n(1 + |w_0|_\infty \Delta t) + \lambda h.$$

y de aquí se obtiene que

$$\max_{0 \leq n \leq N} R_n \leq R_T^{(d)}; \text{ independientemente de } \Delta t,$$

donde

$$R_T^{(d)} = e^{|w_0|_\infty T} R_0 + \frac{\lambda}{|w_0|} \theta(e^{|w_0|_\infty T} - 1).$$

Ahora bien, de (9) sigue que si tomamos, por ejemplo, $\Delta t \leq 1$,

$$\begin{aligned} |\tilde{T}_h^n(x)| &\geq |x| - R_T^{(d)} |w_0|_\infty \Delta t \\ &\geq |x| - R_T^{(d)} |w_0|_\infty. \end{aligned}$$

Entonces,

$$\text{sop}(w^n \circ \tilde{T}_h^n) \subset B(0, R_{1T}), \quad 0 \leq t_n \leq T; \text{ con } R_{1T} = R_T + R_T^{(d)} |w_0|_\infty, \quad (17)$$

donde R_T viene dado por (2).

Además, como en (8) obtenemos

$$|u(\cdot, t)|_\infty \leq R_T |w_0|_\infty; \quad 0 \leq t \leq T.$$

De esta manera

$$|\hat{T}_h^n(x)| \geq |x| - R_T |w_0|_\infty \Delta t.$$

Consiguientemente,

$$\text{sop}(w^n \circ \hat{T}_h^n) \subset B(0, R_{2T}), \quad 0 \leq t_n \leq T, \text{ con } R_{2T} = R_T(1 + |w_0|_\infty)$$

Llamemos ahora $R_{3T} = \max(R_{1T}, R_{2T})$ y definamos

$$B_T = B(0, R_{3T}), \quad Q_T = B_T \times [0, T].$$

Entonces

$$|w^n \circ \hat{T}_h^n - w^n \circ \tilde{T}_h^n|_\infty \leq |\nabla w^n|_\infty |\hat{T}_h^n - \tilde{T}_h^n|_{L^\infty(B_T)} \leq M_T |w_0|_{1,\infty} |\hat{T}_h^n - \tilde{T}_h^n|_{L^\infty(B_T)}, \quad (18)$$

y además,

$$\begin{aligned} |\hat{T}^n(x) - \tilde{T}_h^n(x)| &\leq \Delta t \int_{B(0, 2R_{3T})} |K(x')| |w^n(x-x') - \tilde{w}_h^n(x-x')| dx' \\ &\leq 2 R_{3T} \Delta t |w^n - \tilde{w}_h^n|_\infty, \quad \forall x \in B_T : \end{aligned} \quad (19)$$

Por otra parte, como $w^n - \tilde{w}_h^n$ es afín a trozos, encontramos que

$$|(w^n - \tilde{w}_h^n) \circ \hat{T}_h^n|_\infty \leq |w^n - \tilde{w}_h^n|_\infty.$$

De esta forma, obtenemos la siguiente estimación de estabilidad:

$$|\hat{w}_h^{n+1} - \tilde{w}_h^{n+1}|_\infty \leq (1 + C_1 \Delta t) |w^n - \tilde{w}_h^n|_\infty, \quad (20)$$

con $C_1 = 2R_{3T} M_T |w_0|_{1,\infty}$.

Consistencia:

Para analizar la consistencia, comencemos por la siguiente estimación:

$$|w^{n+1} - \hat{w}_h^{n+1}|_\infty \leq |w^{n+1} - w^n \circ \hat{T}^n|_\infty + |w^n \circ \hat{T}^n - \hat{w}_h^{n+1}|_\infty. \quad (21)$$

Como $w \in C^2(\mathbb{R}^2 \times [0, T])$, entonces $\hat{T}^n \in C^3(\mathbb{R}^2)$. Entonces, de (15) obtenemos

$$|w^n \circ \hat{T}^n - \hat{w}_h^{n+1}|_\infty \leq \kappa |w^n \circ \hat{T}^n|_{2,\infty} h^2.$$

Por otra parte, a partir de la regularidad en tiempo de w , tenemos

$$\max_{0 \leq n \leq N} |w^n \circ \hat{T}^n|_{2,\infty} \leq C_2 \quad (22)$$

donde C_2 es una constante que depende únicamente de $|w|_{2,\infty}$ y T .

Estimamos ahora el primer sumando en (21). Observemos que

$$\begin{aligned} |w^{n+1}(x) - (w^n \circ \hat{T}^n)(x)| &= |w(X(t_n; t_{n+1}, x), t_n) - w(\hat{T}^n(x), t_n)| \\ &\leq |\nabla w|_\infty |X(t_n; t_{n+1}, x) - \hat{T}^n(x)|. \end{aligned}$$

De la ecuación (2) tenemos

$$\begin{aligned} X(t_n; t_{n+1}, x) &= x - \Delta t u(x, t_{n+1}) + \frac{(\Delta t)^2}{2} \frac{d^2 X}{dt^2}(\xi_n^{(1)}; t_{n+1}, x) \\ &= \hat{T}^n(x) + (\Delta t)^2 \left[\frac{1}{2} \frac{d^2 X}{dt^2}(\xi_n^{(1)}; t_{n+1}, x) - u_{,t}(x, \xi_n^{(2)}) \right] \end{aligned}$$

donde $\xi_n^{(1)}, \xi_n^{(2)} \in]t_n, t_{n+1}[$. Entonces,

$$|w^{n+1} - w^n \circ \hat{T}^n|_\infty \leq C_3(\Delta t)^2$$

donde C_3 es una constante que depende únicamente de $|w|_{2,\infty}$ y T .

Hemos probado así la siguiente estimación de consistencia:

$$|w^{n+1} - \hat{w}_h^{n+1}|_\infty \leq C_4(h^2 + (\Delta t)^2) \tag{23}$$

donde C_4 es una constante que depende únicamente de w_0 y T .

Convergencia:

Si consideramos ahora (20) y (23), tenemos

$$|w^{n+1} - \tilde{w}_h^{n+1}|_\infty \leq (1 + C_1\Delta t)|w^n - \tilde{w}_h^n|_\infty + C_4(h^2 + (\Delta t)^2).$$

Entonces,

$$|w^n - \tilde{w}_h^n|_\infty \leq e^{C_1 t_n} |w_0 - r_h w_0|_\infty + \frac{C_4}{C_1} \left(\frac{h^2}{\Delta t} + \Delta t \right) (e^{C_1 t_n} - 1), \quad 0 \leq n \leq N,$$

□

y (16) se cumple.

Volvamos ahora al Algoritmo (A2). Probaremos su convergencia uniforme bajo una suposición adicional sobre el operador de interpolación s_h :

TEOREMA 2 *Supongamos que r_h verifica las propiedades (8), (14) y (15); que s_h verifica (11)-(12) y además*

$$|s_h v|_\infty \leq |v|_\infty, \quad \forall v \in L^\infty(\mathbb{R}^2), \tag{24}$$

$$|v - s_h v|_\infty < \kappa^* |v|_{1,\infty} h \quad \forall v \in C_0^1(\mathbb{R}^2) \tag{25}$$

donde κ^* es una constante numérica. Supongamos también que $h = \theta \Delta t$, donde θ es una constante numérica positiva del orden de la unidad.

Entonces la sucesión descrita por el Algoritmo A2 verifica la estimación de error

$$\max_{0 \leq n \leq N} |w(\cdot, t_n) - \tilde{w}_h^n|_\infty \leq C^*(h + \Delta t + \frac{h^2}{\Delta t}), \tag{26}$$

donde C^* es una constante que depende únicamente de T y w_0 .

Demostración

La demostración de este teorema difiere de la demostración del Teorema 1 sólo en las estimaciones (18) y (19).

Utilizando (11) y (24) podemos deducir, de la misma forma que en el Teorema 1, que existe $R_T^* > 0$ tal que

$$\text{sop}(w^n \circ \tilde{T}_h^n) \cup \text{sop}(w^n \circ \hat{T}_h^n) \subset B_T^* = B(0, R_T^*), \quad 0 \leq t_n \leq T. \quad (27)$$

Entonces

$$|w^n \circ \tilde{T}_h^n - w^n \circ \hat{T}_h^n|_\infty \leq |\nabla w^n|_\infty |\hat{T}_h^n - \tilde{T}_h^n|_{L^\infty(B_T^*)}.$$

Ahora consideremos que si $x \in B_T^*$ se tiene que

$$\begin{aligned} |\hat{T}_h^n(x) - \tilde{T}_h^n(x)| &\leq \Delta t \int_{B(0, 2R_T^*)} |K(x')| |w^n(x - x') - (s_h w^n)(x - x')| dx' + \\ &\quad + \Delta t \int_{B(0, 2R_T^*)} |K(x')| |s_h(w^n - \tilde{w}_h^n)(x - x')| dx' \end{aligned}$$

$$\leq 2R_T^* \Delta t (|w^n - s_h w^n|_\infty + |s_h(w^n - \tilde{w}_h^n)|_\infty) \leq 2R_T^* \Delta t (\kappa^* |\nabla w^n|_\infty h + |w^n - \tilde{w}_h^n|_\infty).$$

Entonces obtenemos la siguiente estimación de estabilidad:

$$|\hat{w}_h^{n+1} - \tilde{w}_h^{n+1}|_\infty \leq (1 + C_1^* \Delta t) |w^n - \tilde{w}_h^n|_\infty + C_2^* \Delta t h, \quad (28)$$

donde C_1^* y C_2^* dependen únicamente de $|w_0|_{1,\infty}$ y T .

La consistencia se demuestra como en el caso del Teorema 1: A partir de (23) y (28), obtenemos

$$|w^{n+1} - \tilde{w}_h^{n+1}|_\infty \leq (1 + C_1^* \Delta t) |w^n - \tilde{w}_h^n|_\infty + C_3^* (\Delta t h + h^2 + (\Delta t)^2)$$

□

de donde se sigue la estimación de error (26).

Observación

No son necesarias hipótesis de regularidad de \mathcal{T}_h para asegurar la estabilidad uniformemente incondicional del algoritmo A1. La estabilidad incondicional en L^2 es una característica de los algoritmos de transporte-proyección aplicados a la resolución de la ecuación de transporte^{16,2}. En nuestro caso, también se obtiene la estabilidad en norma uniforme. Esta se debe fundamentalmente al hecho de que el operador de interpolación que consideramos, no incrementa la norma uniforme.

OPERADORES DE INTERPOLACION

Nos interesaremos por dos tipos de operadores de interpolación r_h , que satisfacen las hipótesis (6), (14) y (15): La interpolación puntual y la interpolación promediada.

Interpolación puntual:

El operador de interpolación “puntual” más simple sobre V_h está definido como sigue:

$$(r_h^1 v)(\alpha_i) = v(\alpha_i), \forall v \in C^0(\mathbb{R}^2)$$

Este operador verifica (6) (con $\lambda = 1$), (14) y (15) para cualquier triangulación \mathcal{T}_h . La única hipótesis que tenemos que hacer sobre la triangulación, es que \mathcal{T}_h constituye una partición de \mathbb{R}^2 .

Interpolación promediada:

Por simplicidad, consideraremos únicamente triángulos regulares, tal como sigue: dados $h_1, h_2 > 0$, los vértices son definidos por

$$\alpha_i = (i_1 h_1, i_2 h_2), \forall i = (i_1, i_2) \in \mathbf{Z},$$

y los triángulos son aquellos de vértices

$$\begin{cases} \alpha_i, \alpha_{(i_1+1, i_2)}, \alpha_{(i_1+1, i_2+1)}, \\ \alpha_i, \alpha_{(i_1, i_2+1)}, \alpha_{(i_1+1, i_2+1)}, \end{cases} \quad \forall i = (i_1, i_2) \in \mathbf{Z}.$$

Para $i \in \mathbf{Z}$, definiremos el rectángulo Q_i por

$$Q_i = [\alpha_{i1} - \frac{h_1}{2}, \alpha_{i1} + \frac{h_1}{2}] \times [\alpha_{i2} - \frac{h_2}{2}, \alpha_{i2} + \frac{h_2}{2}].$$

Nuestro operador de interpolación r_h^2 está definido ahora por

$$(r_h^2 v)(\alpha_i) = \frac{1}{h_1 h_2} \int_{Q_i} v(x) dx, \forall i \in \mathbf{Z}. \tag{29}$$

Nuevamente el operador r_h^2 verifica (6), (14) y (15). La exactitud de segundo orden de r_h^2 se debe fundamentalmente a las simetrías de nuestra triangulación particular. Además, este tipo de interpolación ofrece propiedades de estabilidad excelentes, lo cual se manifiesta en el siguiente resultado:

LEMA 1. *El operador de interpolación r_h^2 verifica las siguientes propiedades:*

1. Si $v \in L^1(\mathbb{R}^2)$, entonces $r_h^2 v \in L^1(\mathbb{R}^2)$ y

$$\int_{\mathbb{R}^2} (r_h^2 v)(x) dx = \int_{\mathbb{R}^2} v(x) dx$$
2. Si $v \in L^2(\mathbb{R}^2)$, entonces $r_h^2 v \in L^2(\mathbb{R}^2)$ y

$$|r_h^2 v|_2 \leq |v|_2$$
3. Si $v \in W^{1,\infty}(\mathbb{R}^2)$, entonces $r_h^2 v \in W^{1,\infty}(\mathbb{R}^2)$ y

$$|\nabla(r_h^2 v)|_\infty \leq |\nabla v|_\infty$$

□

El operador s_h

Tenemos también una gran libertad para elegir el operador s_h . Desde un punto de vista práctico, lo más fácil es tomar un promedio punto a punto:

$$(s_h v)|_T = \frac{1}{3} \sum_{i \in I(T)} v(\alpha_i), \quad \forall T \in \mathcal{T}_h.$$

Una propiedad interesante de esta interpolación, es que

$$(s_h v_h)|_T = \frac{1}{|T|} \int_T v_h(x) dx, \quad \text{si } v_h \in V_h.$$

Otra posibilidad, es tomar un punto α_T para cada $T \in \mathcal{T}_h$ y definir

$$(s_h v)|_T = v(\alpha_T).$$

Ambas elecciones verifican (11) y (12) (con $\lambda = 1$), (24) y (25).

PROPIEDADES DE CONSERVACION

Las Ecuaciones de Euler bidimensionales conservan en tiempo diferentes magnitudes promediadas relacionadas con la velocidad. Esto ocurre, por ejemplo, con la velocidad media, la energía cinética media, la entropía (norma L^2 de la vorticidad), etc. En particular, la circulación total

$$\gamma(t) = \int_{\mathbb{R}^2} w(x, t) dt$$

se conserva. Los métodos de tipo “burbuja-vórtice” verifican una propiedad de conservación de un equivalente discreto de la circulación. Para que un método de vórtices con elementos finitos sea conservativo en este sentido, debe verificar dos condiciones:

1. El operador de interpolación r_h conserva la circulación.
2. La aplicación de flujo discreta \tilde{T}_h^n preserva el área.

La primera condición se puede verificar por muchos operadores, por ejemplo la proyección ortogonal de L^2 sobre V_h utilizada en [2]. También, cualquier operador de interpolación promediada sobre celdas, como el definido por (29), satisface esta condición (ver Lema 1). Sin embargo, para verificar (2) necesitamos resolver exactamente las ecuaciones características (3) para una velocidad discreta de divergencia exactamente nula. Esto ha sido resuelto en [2] para velocidades constantes a trozos, con componente normal continua a lo largo de los lados de la triangulación. Este parece ser el único caso en el cual la condición (2) es satisfecha.

Consiguientemente, nuestros algoritmos no son conservativos. Sin embargo, los resultados de convergencia aseguran una propiedad de conservación aproximada: En efecto,

$$\begin{aligned} \left| \int_{\mathbb{R}^2} (w^n(x) - \tilde{w}_h^n(x)) dx \right| &\leq c_T |w^n - \tilde{w}_h^n|_\infty \\ &\leq c_T h, \quad \text{si } \Delta t = \theta h. \end{aligned} \tag{30}$$

Tal propiedad puede ser deducida de un resultado de convergencia en normas L^p , con p finito. Sin embargo, la estabilidad en norma uniforme hace posible hacer un análisis directo de esta propiedad:

TEOREMA 3 *Supongamos que las hipótesis de los teoremas 1 y 2 se verifican. Entonces, se cumplen las siguientes proposiciones:*

1. *Sea \tilde{T}_h^n la aplicación Lagrangiana discreta correspondiente a cualquiera de los algoritmos A1 o A2. Entonces, para $1 \leq p < +\infty$ existe una constante C_p que depende únicamente de T y w_0 tal que*

$$\max_{0 \leq n \leq N} |\det(\nabla \tilde{T}_h^n) - 1|_p \leq C_p (\Delta t)^2$$

2. *Supongamos que \tilde{T}_h^n corresponde al algoritmo A1 con el operador de interpolación r_h^2 definido en (29). Entonces, existe un tiempo T_0 , tal que si $0 < T < T_0$ y Δt es suficientemente pequeño, entonces la aplicación de flujo discreta \tilde{T}_h^n es invertible, y*

$$\max_{0 \leq n \leq N} |\det(\nabla \tilde{T}_h^n) - 1|_\infty \leq C_\infty (\Delta t)^2,$$

donde C_∞ es una constante que depende únicamente de T y w_0 .

Demostración

La demostración se basa en el hecho de que las velocidades discretas \tilde{u}_h^n definidas por (7) o (13), verifican $\tilde{u}_h^n \in C^1(\mathbb{R}^2)$ y

$$\nabla \cdot \tilde{u}_h^n = 0.$$

Entonces,

$$\det(\nabla \tilde{T}_h^n) = 1 + (\Delta t)^2 \det(\nabla \tilde{u}_h^n) \in \mathbb{R}^2.$$

Por tanto, bastará con encontrar cotas uniformes en L^p para $\nabla \tilde{u}_h^n$.

1. Recordemos que K es un operador acotado de $L^p_{loc}(\mathbb{R}^2)$ en $W^{1,p}(\mathbb{R}^2)$, $1 \leq p < +\infty$. Entonces, en el caso del algoritmo A2,

$$|\nabla \tilde{u}_h^n|_p \leq c_p |s_h \tilde{w}_h^n|_p$$

donde la constante c_p puede depender únicamente de T y w_0 .

A partir de (11), (12), (24) y (27), obtenemos

$$|\nabla \tilde{u}_h^n|_p \leq c_p |s_h \tilde{w}_h^n|_\infty \leq c_p |\tilde{w}_h^n|_\infty \leq c_p |w_0|_\infty. \quad (31)$$

En el caso del algoritmo A1, la segunda desigualdad en (31) se satisface directamente.

2. Sea R_{1T} definido por (17) y definamos las constanes

$$T_0 = \frac{1}{2C_1 |\nabla w^0|_\infty}, \quad C_0 = \frac{|\nabla w_0|_\infty}{1 - T/T_0}, \quad C_1 = R_{1T}.$$

Mostraremos recursivamente que si $0 < T < T_0$ y Δt es suficientemente pequeño, entonces

$$|\nabla \tilde{u}_h^n|_\infty \leq C_1(C_0 + 1) \quad \text{y} \quad |\nabla \tilde{w}_h^n|_\infty \leq C_0 + 1. \quad (32)$$

Consideremos primero que, a partir de (7) y del Lema 1,

$$|\nabla \tilde{u}_h^n|_\infty \leq C_1 |\nabla [r_h \circ \tilde{w}_h^n \circ \tilde{T}_h^n]|_\infty \leq C_1 |\nabla (\tilde{w}_h^n \circ \tilde{T}_h^n)|_\infty. \quad (33)$$

Si $n = 0$, entonces (32) se cumple inmediatamente. Supongamos que (32) es cierto para $0, 1, \dots, n$. Mostraremos que lo sigue siendo para $n + 1$.

Consideremos que

$$|\det(\nabla \tilde{T}_h^n(x))| \geq 1 - 2|\nabla \tilde{u}_\infty^n|^2 (\Delta t)^2.$$

Supongamos ahora que

$$\Delta t < \min\left(\frac{1}{\sqrt{2}C_1(C_0 + 1)}, \frac{2C_0 + 1}{2C_1^2 C_0^3 [e^{2(C_0+1)T} - 1]}\right) \quad (34)$$

Entonces

$$\det(\nabla \tilde{T}_h^n(x)) \neq 0, \forall x \in \mathbb{R}^2.$$

De aquí que, \tilde{T}_h^n sea globalmente inversible, y $(\tilde{T}_h^n)^{-1} \in \mathcal{C}^1(\mathbb{R}^2)$. Definamos

$$\Omega_T = (\tilde{T}_h^n)^{-1}(\text{int}(T)), \forall T \in \mathcal{T}_h,$$

donde $\text{int}(T)$ denota al interior topológico del triángulo T .

Entonces, $\partial\Omega_T = (T_h^n)^{-1}(\partial T)$ y

$$\begin{aligned} \text{área}(\partial\Omega_T) &= \int_{\partial\Omega_T} dx = \int_{\partial T} |\det[(\nabla\tilde{T}_h^n)^{-1}(y)]| dy \\ &= \int_{\partial T} \frac{1}{|\det[(\nabla\tilde{T}_h^n)(\tilde{T}_h^n(y))]|} dy \leq \frac{1}{1 - 2|\nabla\tilde{u}_h^n|_\infty^2(\Delta t)^2} \int_{\partial T} dy = 0. \end{aligned}$$

Como $\nabla(\tilde{w}_h^n \circ \tilde{T}_h^n)|_{\Omega_T} \in C^0(\Omega_T)$,

$$|\nabla(\tilde{w}_h^n \circ \tilde{T}_h^n)|_\infty = \max_{T \in \mathcal{T}_h} \sup_{x \in \Omega_T} |\nabla(\tilde{w}_h^n \circ \tilde{T}_h^n)(x)|. \tag{35}$$

Consideremos ahora que

$$\forall x \in \Omega_T, \frac{\partial}{\partial x_i}(\tilde{w}_h^n \circ \tilde{T}_h^n)(x) = \frac{\partial \tilde{w}_h^n}{\partial x_i}[(\tilde{T}_h^n)(x)] - \Delta t \sum_{j=1}^2 \frac{\partial \tilde{w}_h^n}{\partial x_j}[(\tilde{T}_h^n)(x)] \frac{\partial \tilde{u}_{hj}^n}{\partial x_i}(x).$$

Entonces

$$\sup_{x \in \Omega_T} |\nabla(\tilde{w}_h^n \circ \tilde{T}_h^n)(x)| \leq (1 + 2\Delta t |\nabla\tilde{u}_h^n|_\infty) |\nabla\tilde{w}_h^n|_\infty.$$

De (33), (35),

$$|\nabla\tilde{w}_h^{n+1}|_\infty \leq (1 + 2C_1\Delta t |\nabla\tilde{w}_h^n|_\infty) |\nabla\tilde{w}_h^n|_\infty. \tag{36}$$

Esta estimación permite acotar $|\nabla\tilde{w}_h^n|_\infty$ durante un intervalo limitado de tiempo. En efecto, definamos la sucesión

$$\tilde{\xi}_n = |\nabla\tilde{w}_h^n|_\infty, \text{ para } n = 0, 1, \dots, N.$$

Como (36) es también verdadera para $0, 1, \dots, n$, entonces la sucesión $\{\tilde{\xi}_k\}_{k=1}^n$ está acotada término a término por la sucesión $\{\xi_k\}_{k=1}^n$ definida por

$$\xi_{k+1} = (1 + 2C_1 \Delta t \xi_k)\xi_k; \quad \xi_0 = \tilde{\xi}_0 \tag{37}$$

Esta sucesión se obtiene aplicando el método de Euler progresivo a la ecuación diferencial

$$\xi'(t) = 2C_1\xi^2(t), \quad \xi(0) = \tilde{\xi}_0. \tag{38}$$

La solución de esta ecuación diferencial es

$$\xi(t) = \frac{\xi_0}{1 - 2C_1\xi_0 t},$$

que es una función creciente no-negativa, que tiende a infinito en el instante $t = \frac{1}{2C_1\xi_0} \geq T_0$.

Podemos probar ahora que

$$|\xi_n| \leq C_0 + 1, \quad \forall n = 1, 2, \dots, N. \quad (39)$$

Para ello, observemos en primer lugar que ello es cierto para ξ_0 . Supongámoslo cierto para $k = 0, 1, \dots, n$. Definamos

$$\rho_k = \xi(t_k) - \xi_k; \quad 0 \leq k \leq N.$$

Entonces, de (37) y (38) obtenemos

$$\rho_{k+1} = [1 + \Delta t(\xi(t_k) + \xi_k)]\rho_k + \frac{(\Delta t)^2}{2}\xi''(c_k), \quad \text{con } c_k \in]t_k, t_{k+1}[, \quad 0 \leq k \leq n,$$

y de aquí que

$$|\rho_{k+1}| \leq [1 + (2C_0 + 1)\Delta t]|\rho_k| + 2C_1^2 C_0^3 (\Delta t)^2, \quad 0 \leq k \leq n.$$

Con esta desigualdad y (34) llegamos a la cota deseada:

$$|\rho_{k+1}| \leq \frac{2C_1^2 C_0^3}{2C_0 + 1} [e^{(2C_0+1)t_{n+1}} - 1] \quad \text{si } \Delta t < 1.$$

Entonces obtenemos nuestro resultado:

$$|\nabla \tilde{w}_h^{n+1}|_\infty \leq \xi_{n+1} \leq |\xi(t_{n+1})| + 1 \leq C_0 + 1.$$

Observación

El hecho de que la aplicación discreta de flujo

$$y = \tilde{T}_h^n(x)$$

sea invertible con inversa uniformemente acotada globalmente puede utilizarse para dar una demostración sencilla de la convergencia del algoritmo A1 con r_h^2 en normas L^p para cortos intervalos de tiempo.

EJEMPLOS NUMERICOS

En esta sección, analizaremos los resultados de la ejecución práctica de los algoritmos anteriores en un caso test. La convergencia de primer orden predicha por el teorema 1 es confirmada numéricamente en velocidad y vorticidad.

En nuestros tests utilizamos una solución estacionaria (u, w) de las ecuaciones de Euler 2D (1). Esta solución es

$$u(x) = f(r)(-x_2, x_1),$$

$$\text{con } r = \sqrt{x_1^2 + x_2^2},$$

$$f(r) = \begin{cases} \frac{[1-(1-r^2)^8]}{16r^2} & \text{si } r \leq 1, \\ \frac{1}{16r^2} & \text{si } r > 1; \end{cases}$$

y

$$w(x) = \begin{cases} (1-r^2)^7 & \text{si } r \leq 1, \\ 0 & \text{si } r > 1. \end{cases}$$

Las funciones u y w son muy regulares. El soporte de w es el círculo unidad. El flujo correspondiente es radialmente simétrico y rota alrededor del origen, produciendo un gran gradiente normal en la velocidad tangencial en una banda estrecha alrededor del círculo $r = 0.4$. Esta solución de las ecuaciones de Euler 2D ha sido ampliamente utilizada para mostrar las propiedades de convergencia de métodos de vórtices. De esta forma, tenemos un buen test para mostrar los resultados obtenidos por la ejecución de nuestro algoritmo.

Para discretizar la vorticidad hemos utilizado triangulaciones uniformes, con una talla $h = 0.15$, $h = 0.10$ y $h = 0.05$. Si los triángulos se enumeran con cuidado, es posible encontrar el triángulo que contiene un punto $x \in \mathbb{R}^2$ dado en un número limitado de operaciones. Entonces, el paso de interpolación requerido por la actualización de la vorticidad resulta poco costoso computacionalmente. Para triangulaciones generales, este costo se ve incrementado, aunque puede ser mantenido en un número de operaciones del orden de $\frac{1}{h}$ si es programado con suficiente cuidado.

La principal dificultad práctica que encontramos cuando programamos las dos primeras versiones de nuestro algoritmo es el cálculo del soporte de la vorticidad discreta. La estimación (10) puede ser utilizada para dar un círculo C^{n+1} que contenga $\text{sop}(\tilde{w}^{n+1})$, a partir de $\text{sop}(\tilde{w}^n)$. Sin embargo, calcular \tilde{w}^{n+1} en todo C^{n+1} requeriría una gran cantidad de trabajo computacional inútil. Para obviar esta dificultad, en nuestros cálculos hemos utilizado una técnica especial para "predecir" más precisamente $\text{sop}(\tilde{w}^{n+1})$. Esta técnica debe ser descrita como sigue:

- (a) Inicialización: Aproximar $\Gamma_h^0 = \text{sop}(\tilde{w}^0)$ mediante una línea quebrada de vértices $\{\xi_i^0\}_{i=1}^M$.
- (b) Paso de tiempo:

- i. Dado el conjunto de puntos $\{\xi_i^n\}_{i=1}^M$ que describe la frontera computacional Γ_h^n del $sop(\tilde{w}^n)$, definimos los puntos

$$\xi_i^{n+1} = \xi_i^n + \Delta t \tilde{u}_h^n(\xi_i^n) \quad (\text{Método de Euler progresivo});$$

- ii. Definimos la frontera aproximada Γ_h^{n+1} del $sop(\tilde{w}^{n+1})$ como la línea quebrada de vértices $\{\xi_i^{n+1}\}_{i=1}^M$, con las mismas conexiones de Γ_h^n .
- iii. Definimos el soporte computacional de \tilde{w}^{n+1} como la unión de todos los triángulos de \mathcal{T}_h que quedan dentro de Γ_h^{n+1} , más los que se intercepten con Γ_h^{n+1} .

Aunque esta técnica introduce una cantidad adicional de error de discretización, nuestros cálculos muestran que éste sigue siendo de primer orden.

Para medir los errores hemos utilizado una seminorma de tipo L^2 . Para una función $v: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, esta seminorma se define como sigue:

$$|v|_{l^2}^2 = \sum_{\tau \in \mathcal{T}_h} \frac{\text{Area}(\tau)}{3} \sum_{j=1}^3 |v(\alpha_j^\tau)|^2,$$

donde α_j^τ es el j -ésimo vértice del triángulo τ .

El error normalizado en velocidad es definido por,

$$e_u(t_n, h) = \frac{|u^n - \tilde{u}_h^n|_{l^2}}{|u^n|_{l^2}}.$$

Con esta definición de errores, el orden de convergencia, por ejemplo en velocidad $p_u(t)$, ha sido estimado utilizando dos valores consecutivos de h :

$$p_u(t) = \frac{\log[e_u(t, h_1)/e_u(t, h_2)]}{\log(h_1/h_2)}.$$

Aunque hemos demostrado la convergencia de primer orden en norma uniforme, por razones prácticas es más conveniente calcular los errores utilizando normas l^2 ; esto permite obtener curvas de error más regulares. En cambio, las normas l_∞ producen grandes oscilaciones en las curvas que muestran los órdenes de convergencia, que desaparecen cuando utilizamos normas l^2 .

Las Figuras 1 y 2 muestran las curvas de error en velocidad y vorticidad respectivamente, correspondientes al Algoritmo A2, con $\Delta t = h$. Se observa un crecimiento exponencial de los errores, aunque decrecen con h . En nuestro caso, este crecimiento es más rápido que en los métodos de burbuja vórtice, debido a la etapa de interpolación. Como la velocidad es más regular que la vorticidad, podemos esperar que los errores en velocidad sean más pequeños que en vorticidad. Sin embargo, vemos que sucede lo contrario. Este comportamiento aparentemente anormal se debe a pérdida de precisión cuando reinterpolamos la velocidad mediante una función constante a trozos, en el cálculo de la vorticidad con (13).

Las Figuras 3 y 4 muestran la evolución de los órdenes de convergencia calculados en velocidad y vorticidad, respectivamente. La curva correspondiente a

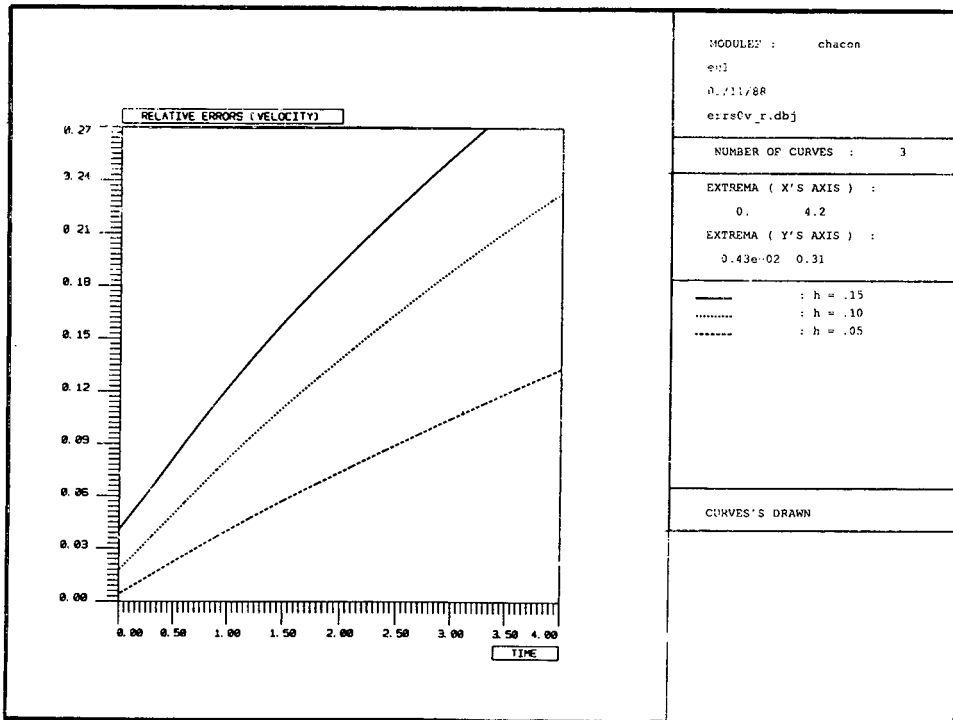


Figura 1

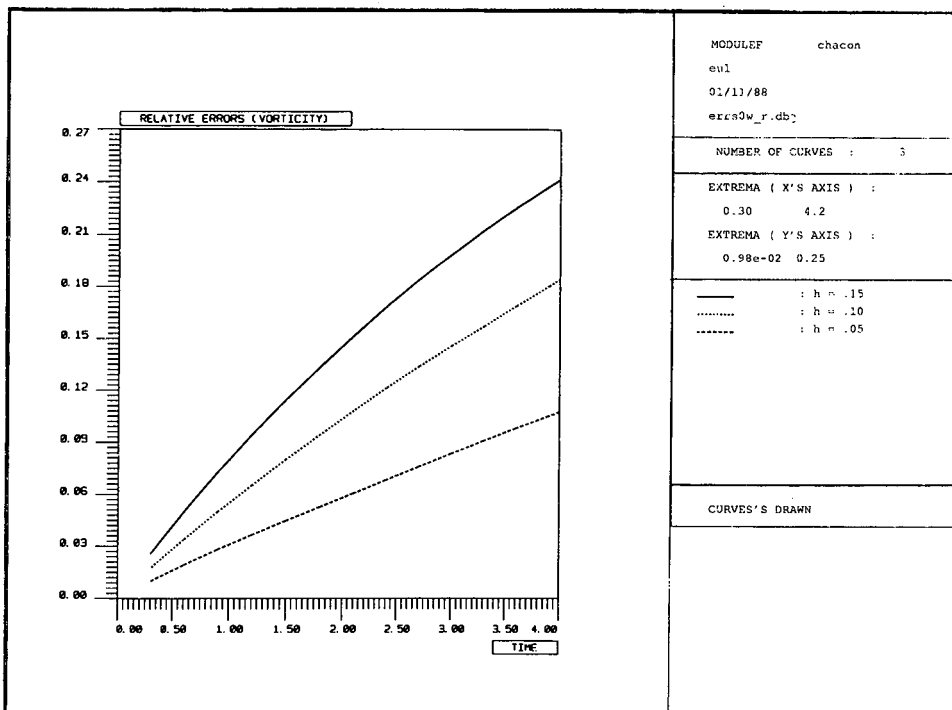


Figura 2

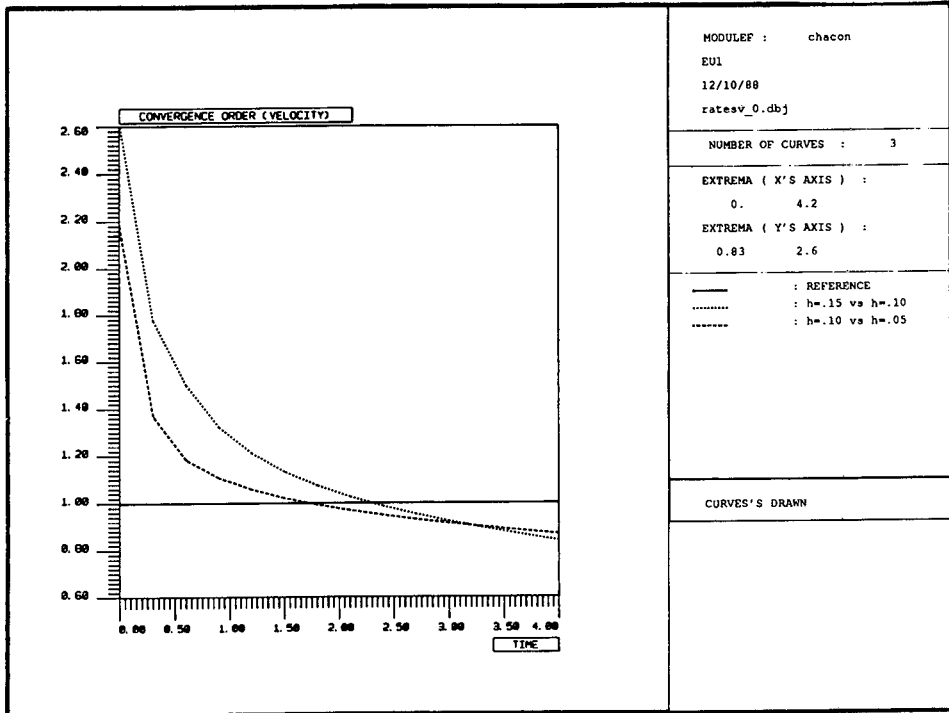


Figura 3

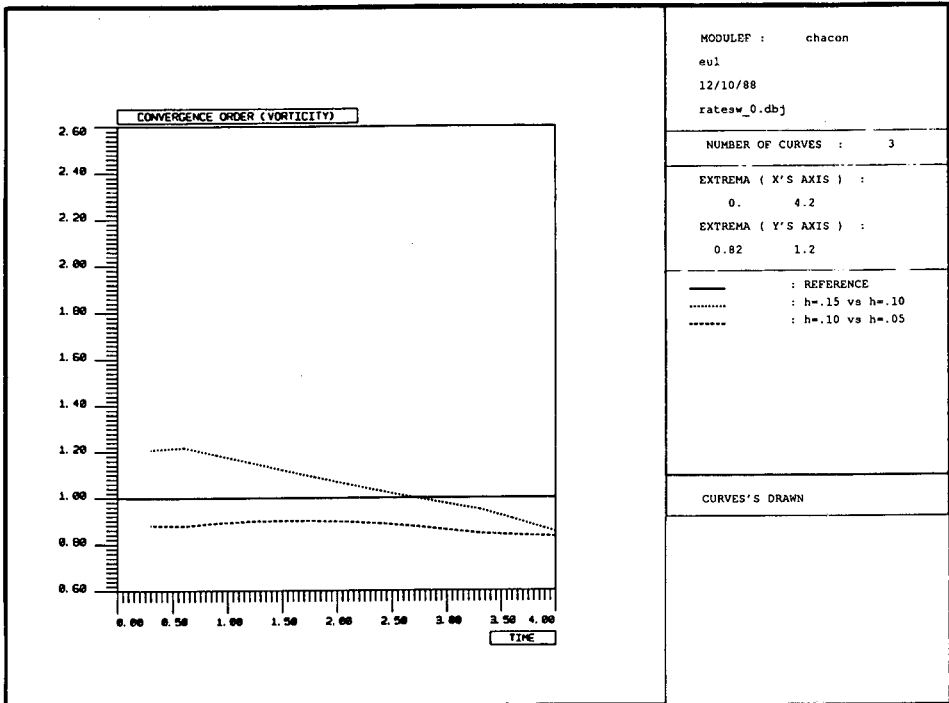


Figura 4

la velocidad muestra una exactitud de segundo orden en los momentos iniciales, debida probablemente a ciertas simetrías en las triangulaciones que producen cancelaciones. Se observa una transición suave al primer orden con el paso del tiempo. Finalmente esta exactitud se pierde progresivamente cuando el tiempo crece, como es usual en la resolución con Elementos Finitos de muchos problemas de evolución. Además, la curva correspondiente a la vorticidad muestra un comportamiento más estable en este caso. La estimación de orden en el tiempo $t = 0$ ha sido omitida, ya que nuestro error calculado es cero. Observemos que en ambos casos las curvas correspondientes a pequeños valores de h están dentro del orden teórico $p = 1$. La ejecución del Algoritmo A1 en este caso test es similar, aunque todos los errores son algo más pequeños, en concordancia con la mayor exactitud en el cálculo de velocidades.

UNA VERSION DE SEGUNDO ORDEN

Ahora nuestro propósito es describir una versión del algoritmo con segundo orden de consistencia. Si tenemos en cuenta que $T = N \Delta t$, para anular los efectos de la propagación de error al cabo de un tiempo T , convendrá exigirle a nuestro algoritmo un error de consistencia local de tercer orden en tiempo y en espacio. Para ello, es necesario en primer lugar construir un esquema con error de consistencia $O(\Delta t^3)$ para discretizar la ecuación de las curvas características (3). Sin embargo, la condición inicial está dada en el instante t_{n+1} y la velocidad únicamente es conocida en instantes anteriores. Ello hace necesario construir un esquema numérico no estándar, de tipo mixto Runge-Kutta y multipaso:

LEMA 2. *Sea*

$$\tilde{x}^{n+1} = x^{n+1} + \Delta t u(x^{n+1}, t_n). \tag{40}$$

Entonces,

$$x^n = x^{n+1} - \frac{\Delta t}{2} (3u(x^{n+1}, t_n) - u(\tilde{x}^{n+1}, t_{n-1})) \tag{41}$$

es tal que

$$|x^n - X(t_n, t_{n+1}, x^{n+1})| \leq c \Delta t^3 \tag{42}$$

□

Por otra parte, el error local en la interpolación de la vorticidad debe ser también de tercer orden en h . Ello se consigue interpolando ésta localmente mediante polinomios de segundo grado. Concretamente, la vorticidad ha sido discretizada sobre el siguiente espacio de elementos finitos:

$$W_h = \{w_h \in C^0(\mathbb{R}^2) | w_h|_T \in \mathbf{P}_2, \forall T \in \mathcal{T}_h\}. \tag{43}$$

Denotemos por $\{\mu_j\}_{j \in \mathbb{N}}$ los puntos medios de los lados de los triángulos de \mathcal{T}_h . Entonces, una función $w_h \in W_h$ queda unívocamente determinada por los valores

$$w_h(\alpha_i), \quad \forall i \in \mathbb{N} \quad \text{y} \quad w_h(\mu_j), \quad \forall j \in \mathbb{N}. \quad (44)$$

Denotaremos por $\rho_h : C^0(\mathbb{R}^2) \rightarrow W_h$ al operador que interpola los valores dados en (44).

Nuestro algoritmo puede ahora ser descrito como sigue:

Algoritmo B

Supongamos que $w_0 \in C^2(\mathbb{R}^2)$ y tiene soporte compacto. Aproximaremos $w(\cdot, t_n)$ mediante una función \tilde{w}_h^n de W_h .

(a) Inicialización:

i. Vorticidad:

$$\tilde{w}^0 = \rho_h \circ w_0, \quad (45)$$

ii. Velocidad:

$$\tilde{u}^0 = K * (r_h^1 \circ \tilde{w}_h^0), \quad (46)$$

(b) Iteración en tiempo:

Supongamos dados $\tilde{w}_h^0, \tilde{w}_h^1, \dots, \tilde{w}_h^n \in W_h$.

i. Velocidad discreta:

$$\tilde{u}_h^n = K * (r_h^1 \circ \tilde{w}_h^n), \quad (47)$$

ii. Característica discreta: \tilde{T}_h^n :

iii. Dado $x \in \mathbb{R}^2$,

$$\tilde{T}_h^n(x) = x - \frac{\Delta t}{2} [3\tilde{u}_h^n(x) - \tilde{u}_h^{n-1}(\tilde{x})]; \quad (48)$$

donde

$$\tilde{x} = x + \Delta t \tilde{u}_h^n(x).$$

iv. Vorticidad:

$$\tilde{w}_h^{n+1} = \rho_h \circ \tilde{w}_h \circ \tilde{T}_h^n. \quad (49)$$

Observación

Es necesario ejecutar el primer paso de este algoritmo con un método de segundo orden, para conservar su orden de precisión.

Observemos que en el Algoritmo B la velocidad discreta no está calculada mediante convolución del núcleo de Biot-Savart con la vorticidad discreta, sino con el interpolado afín a trozos de ésta. De esta forma se consigue una notable reducción en el número de operaciones necesarias para calcular la velocidad, sin que por ello el error de truncamiento local deje de ser de tercer orden. Esta afirmación se formaliza como sigue:

LEMA 3. *Supongamos que $w_0 \in C^3(\mathbb{R}^2)$ con soporte compacto. Definamos*

$$\hat{w}_h^n = \rho_h(w(\cdot, t_n)), \quad \hat{u}_h^n = K * (r_h \circ \hat{w}_h^n).$$

Sean $\bar{w}_h^{n+1} \in W_h$ y \bar{u}_h^{n+1} la velocidad y vorticidad calculadas mediante un paso de tiempo del Algoritmo B, a partir de los valores \hat{w}_h^n , \hat{u}_h^n y \hat{u}_h^{n-1} .

Entonces existe una constante C que depende únicamente de w_0 y T , tal que

$$|\bar{w}_h^{n+1} - w(\cdot, t_{n+1})|_{L^\infty(\mathbb{R}^2)} \leq C(h^3 + \Delta t^3);$$

$$|\bar{u}_h^{n+1} - u(\cdot, t_{n+1})|_{L^\infty(\mathbb{R}^2)} \leq C(h^3 + \Delta t^3);$$

□

De esta forma, el Algoritmo B tiene efectivamente un error de consistencia local de tercer orden. Sin embargo, este algoritmo no es ahora estable en norma uniforme. En efecto, el operador de interpolación ρ_h , en general incrementa la norma uniforme de la función interpolada.

Por otra parte, ya hemos comentado anteriormente, que en el algoritmo A1, el paso más costoso computacionalmente es el cálculo de la velocidad en los N vértices de la triangulación, que requiere en total $O(N^2)$ operaciones. Como se puede apreciar, en el algoritmo B este paso es aún más costoso, para calcular la velocidad discreta serán necesarias del orden de ocho veces este número de operaciones. Por esta razón hemos utilizado una técnica de cálculo rápido de la velocidad, que reduce este número de operaciones de $O(N^2)$ a $O(N \log^2(N))$ (Cf. [4] y [10]).

Un ejemplo numérico

En esta sección, analizaremos los resultados de la ejecución práctica del algoritmo B en el caso test empleado en la Sección 6. En él, se confirma numéricamente la convergencia de segundo orden, previsible a partir de las estimaciones de consistencia.

Para discretizar la vorticidad hemos utilizado triangulaciones uniformes, con tallas $h = \frac{1}{8}$, $h = \frac{1}{12}$ y $h = \frac{1}{16}$.

Para medir los errores hemos utilizado la misma seminorma que en la sección referente a Ejemplos Numéricos.

Las Figuras 5 y 6 muestran las curvas de error en velocidad y vorticidad respectivamente, con $\Delta t = \frac{9h}{5}$. Se observa una notable disminución de la difusión numérica, ya que el error permanece prácticamente constante.

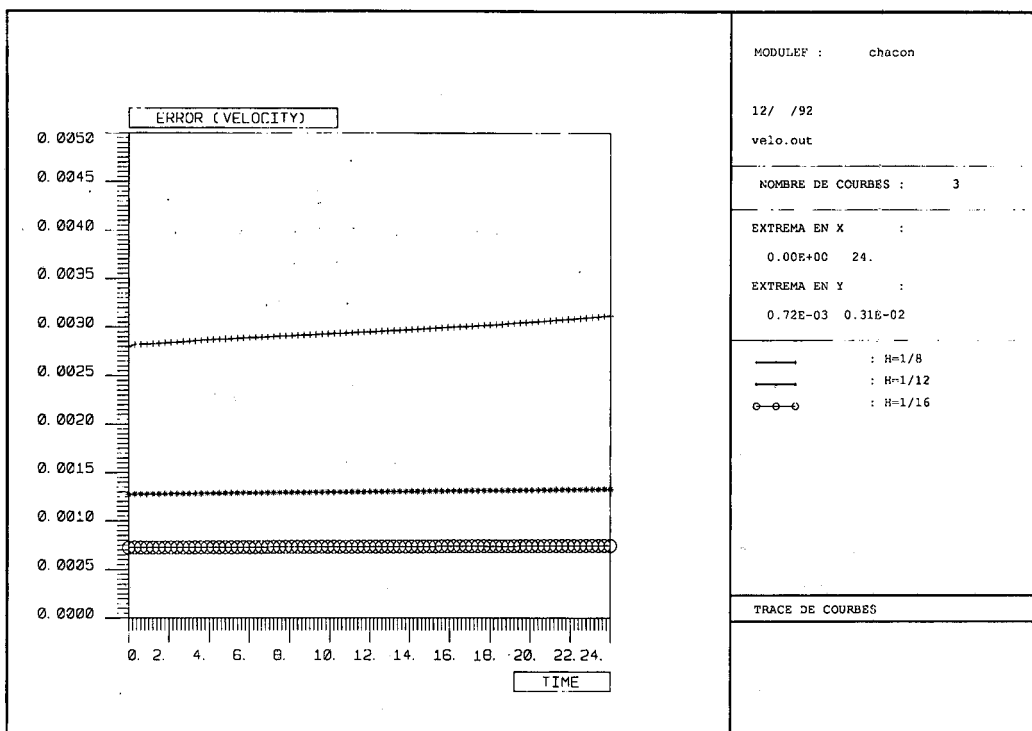


Figura 5

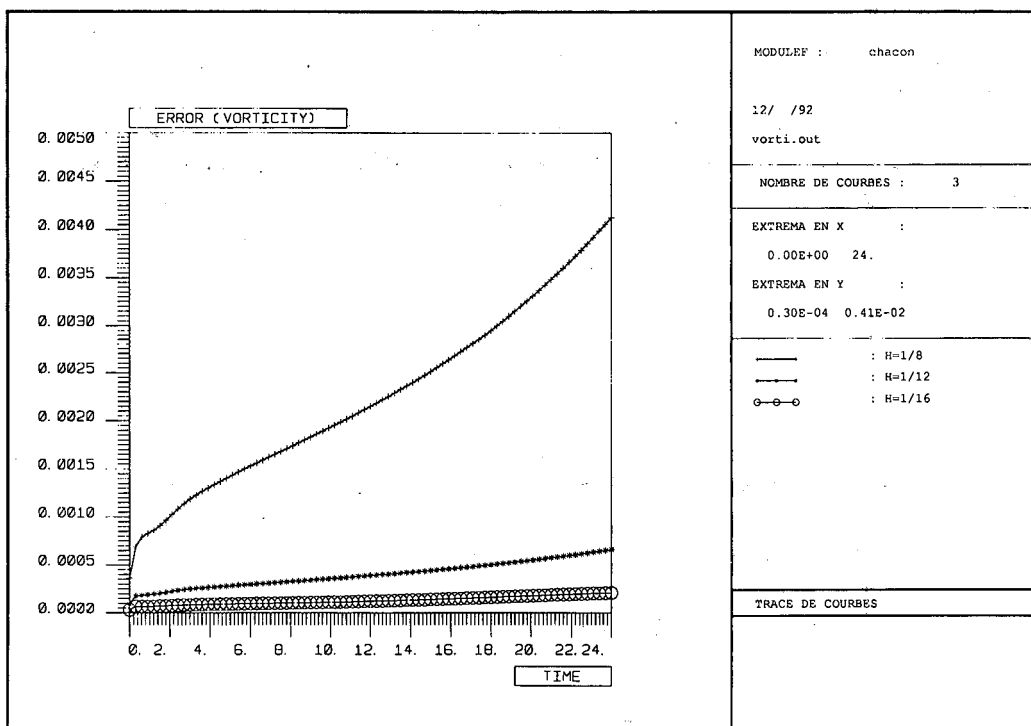


Figura 6

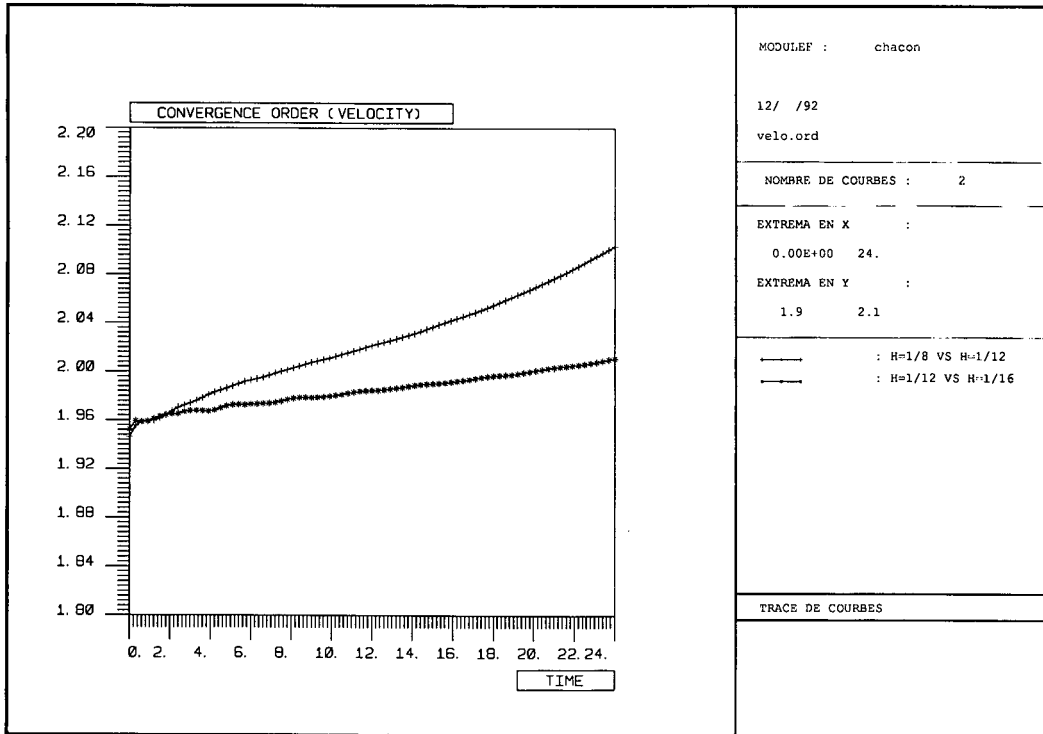


Figura 7

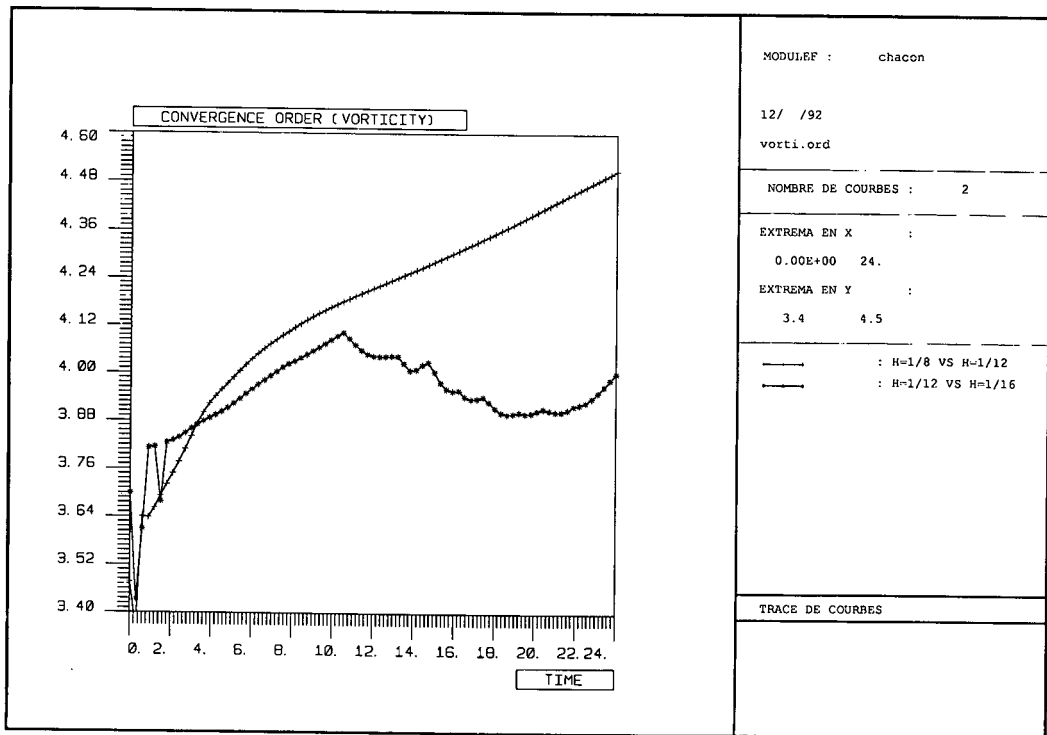


Figura 8

Las Figuras 7 y 8 muestran la evolución de los órdenes de convergencia calculados en velocidad y vorticidad, respectivamente. La curva correspondiente a la vorticidad llega a alcanzar una exactitud de cuarto orden, debido probablemente a ciertas simetría en las triangulaciones que producen cancelaciones. Para evitar perturbaciones del error, producidas por el paso de inicialización, hemos dado los valores exactos de la vorticidad en los dos instantes iniciales.

REFERENCIAS

1. C. Anderson, C. Greengard, "On vortex methods.", *SIAM J. Numer. Anal.*, **22**, no. 3, (1985).
2. C. Bardos, M. Bercovier, O. Pironneau, "The Vortex method with finite elements", *Mathematics of Computation*, Vol. **36**, no. 153, (1981).
3. T. J. Beale, A. Majda, "Vortex Methods II : High order accuracy in two and three dimensions", *Math. Comp.*, Vol. **32**, pp. 29-52, (1982).
4. I. Bless Ranero, T. Chacón Rebollo, "Sobre los algoritmos de cálculo rápido de la velocidad en métodos de vórtices", Aparecerá en 'Actas Seminario Hispano-Francés sobre Recientes Avances en Ecuaciones en Derivadas Parciales no Lineales (Sevilla, 1992)'.
5. G.P. Ciarlet, "*The Finite Element method for elliptic problems.*", North Holland, Amsterdam, (1978).
6. T. Chacón Rebollo, T. Hou, "A Lagrangian algorithm for 2D Euler equations", *Comm. Pure Appl. Maths.*, Vol. **XLIII**, pp. 735-767, (1990).
7. J.A. Chorin, "Numerical study of slightly viscous flow", *J. Fluid Mech*, Vol. **57**, pp. 786-796, (1973).
8. H.G. Cottet, "Methodes particulaires pour l'Equation d'Euler dans le plan", *These 3me Cycle Univ. Paris VI*, (1982).
9. J. Goodman, T. Hou, J. Lowengrub, "Convergence of the point-vortex method for 2D Euler equations", *Comm. Pure Appl. Maths.*, Vol. **43**, pp. 415-430, (1990).
10. L. Greengard, V. Rokhlin, "A fast algorithm for particle simulations", *J. Comp. Phys.*, Vol. **73**, pp. 325-348, (1987).
11. O. Hald, "Convergence of vortex methods II", *SIAM J. Numer. Anal.*, **16**, pp. 726-755, (1979).
12. T. Kato, "On the classical solution of the two-dimensional nonstationary Euler equations", *Arch. Rat. Mech.*, Vol. **25**, pp. 188-200, (1967).
13. A. Leonard, "Vortex methods for flow simulations", *J. Comput. Phys.*, Vol. **37**, pp. 289-335, (1980).
14. A. Leonard, "Computing three-dimensional incompressible flows with vortex elements", *Ann. Rev. Fluid Mech.*, Vol. **17**, pp. 523-559, (1985).
15. A. Majda, "Vortex Dynamics: Numerical Analysis, Scientific Computing and Mathematical Theory", *Proc. 1st. International Congress in Applied Mathematics*, Paris (1987).
16. O. Pironneau, "On the transport-diffusion algorithm and its applications to the Navier-Stokes equations", *Numer. Math.*, Vol. **38**, pp. 309-332, (1982).
17. A.P. Raviart, "An analysis of particle methods", CIME course, Como, Italy, (1983).
18. L. Rosenhead, "The point vortex approximation of a vortex sheet", *Proc. Roy. Soc., London Ser. A*, **134**, pp. 170-192, (1932).
19. O. Zienkiewicz, J. Heinrich, "The finite element method and convection problems in fluid mechanics", en *Finite Elements in Fluids 3*, R. Gallagher, ed., Wiley, New York, (1978).