

Capítulo 7

Resultados

En este capítulo vamos a describir los resultados de los cálculos que se realizan en el programa.

7.1. Estado

El programa principal gestiona la evolución temporal del campo de velocidades en un cubo, el algoritmo se puede ver en la figura 6.2 y en las ecuaciones 4.42 y 4.48 se indica los procedimientos seguidos para integrar. Para poder hacer otro tipo de cálculos o para poder continuar la simulación en otro momento, se guarda un archivo con los datos necesarios para describir el estado del sistema. Este archivo se genera en el directorio desde el que se ejecuta el programa y tiene como nombre *MPI.Estado-NXX.dat* donde *XX* hay que sustituirlo por el valor de *N* de la simulación. En el cuadro 7.1 se puede ver la estructura del archivo:

Variable	Tipo	Descripción
N	Entero	Número de divisiones del cubo.
N3D	Entero	Tamaño de las matrices tridimensionales.
N3D_DIV	Entero	Factor de escala para normalizar las transformadas.
N3D_FREC	Entero	Tamaño de las matrices en frecuencia.
K_FILTRO	Real	Valor de k máximo que deja pasar el filtro esférico.
Kcuadrado_FILTRO	Real	Cuadrado del valor anterior.
K_VISCOSIDAD	Real	Valor de k por debajo del cual se introduce energía.
K_VISCOSIDAD_CUAD	Real	Cuadrado del valor anterior.
VISCOSIDAD	Real	Valor de ν .
VISCOSIDADNEG	Real	Último valor calculado de la viscosidad negativa.
EPSILON	Real	Último valor calculado de ϵ .
ETA	Real	Último valor calculado de η .
KETA_ESTACIONARIA	Real	$k_{max}\eta$ al que el PID intenta estabilizar el sistema.
TIEMPOTOTAL	Real	Tiempo total de la simulación.
dt	Real	Último paso de integración utilizado.
FORZADO	Entero	Variable que indica si se está usando el forzado.
CALCULAINVARIANTES	Entero	Variable que indica si se calculan los invariantes.
PID.error	Real	Último valor calculado del error en el PID para alcanzar el $k_{max}\eta$ deseado.
PID.deriv_error	Real	Último valor calculado de la derivada del error en el PID para alcanzar el $k_{max}\eta$ deseado.

Variable	Tipo	Descripción
PID.int_error	Real	Último valor calculado de la integral del error en el PID para alcanzar el $k_{max}\eta$ deseado.
PID.error_ant	Real	Valor del error en el paso anterior.
PID.K_P	Real	Constante de proporcionalidad del PID.
PID.K_I	Real	Constante de derivativa del PID.
PID.K_D	Real	Constante de Integral del PID.
Espectro.u_prima_cuadrado	Real	Valor de u'^2 .
Espectro.EPSILON	Real	Valor de ϵ calculado a partir del espectro.
Espectro.L	Real	Valor de L calculado a partir del espectro.
Imagen.alto	Entero	Alto en píxeles de las imágenes que se generan.
Imagen.ancho	Entero	Ancho en píxeles de las imágenes que se generan.
Imagen.Divisiones_X	Entero	Número de décadas de k que se muestran en la gráfica del espectro.
Imagen.Divisiones_Y	Entero	Número de décadas de $E(k)$ que se muestran en la gráfica del espectro.
Imagen.MARGEN_X	Entero	Tamaño en píxeles del margen a los lados de las imágenes.
Imagen.MARGEN_Y	Entero	Tamaño en píxeles del margen superior e inferior de las imágenes.
Imagen.Puntos_Por_Division_X	Entero	Resolución en píxeles de cada década de k .
Imagen.Puntos_Por_Division_Y	Entero	Resolución en píxeles de cada década de $E(k)$.
Imagen.KETA_MAX	Real	Valor máximo que se muestra en la gráfica de $k_{max}\eta$.
Imagen.KETA_MIN	Real	Valor mínimo que se muestra en la gráfica de $k_{max}\eta$.
VX.Frec	Real	Componente X de velocidad en el instante actual.
VY.Frec	Real	Componente Y de velocidad en el instante actual.
VZ.Frec	Real	Componente Z de velocidad en el instante actual.
VVORTX.Frec	Real	Componente X de $\left(\overbrace{\mathbf{v} \times \boldsymbol{\omega}}\right)_k$ en el instante actual.
VVORTY.Frec	Real	Componente Y de $\left(\overbrace{\mathbf{v} \times \boldsymbol{\omega}}\right)_k$ en el instante actual.
VVORTZ.Frec	Real	Componente Z de $\left(\overbrace{\mathbf{v} \times \boldsymbol{\omega}}\right)_k$ en el instante actual.
VX_nuevo.Frec	Real	Componente X de velocidad en el paso siguiente.
VY_nuevo.Frec	Real	Componente Y de velocidad en el paso siguiente.
VZ_nuevo.Frec	Real	Componente Z de velocidad en el paso siguiente.
VVORTX_ant.Frec	Real	Componente X de $\left(\overbrace{\mathbf{v} \times \boldsymbol{\omega}}\right)_k$ en el instante anterior.
VVORTY_ant.Frec	Real	Componente Y de $\left(\overbrace{\mathbf{v} \times \boldsymbol{\omega}}\right)_k$ en el instante anterior.
VVORTZ_ant.Frec	Real	Componente Z de $\left(\overbrace{\mathbf{v} \times \boldsymbol{\omega}}\right)_k$ en el instante anterior.

Cuadro 7.1: Formato del fichero de estado.

7.2. Parámetros

Durante la ejecución del programa además de la evolución del campo de velocidades se van calculando algunos parámetros que describen características del estado. En el capítulo

2 se hace una breve introducción a los mismos, en esta sección se incluye una tabla de los valores obtenidos para distintas simulaciones:

N	32	64	128	128	128
ν	0,1	0,1	0,1	0,25	0,04
$K_{max}\eta$	1,396965	1,396395	1,466518	1,466388	1,456440
ϵ	11,60728	227,5969	3298,908	51563,77	217,0350
η	9,634238e-02	4,578344e-02	2,346429e-02	2,346221e-02	2,330304e-02
u'^2	5,320374	48,48671	291,9565	1907,654	67,51930
L	1,662018	1,767455	1,554745	1,486001	1,760550
λ	0,8809655	0,6613457	0,4341388	0,4247529	0,4308568
Re	38,33600	123,07213	265,65515	295,9327	361,6615
Re_λ	20,94586	53,85266	88,62102	84,58828	88,50889
$\frac{dt}{T_\eta}$	0,024829	0,020108	0,019825	0,024451	0,021076

Cuadro 7.2: Resultados de las simulaciones.

Como se puede ver en el cuadro 7.2 para un mismo valor de la viscosidad, aumentando N se alcanzan valores de Re mayores. Esto se debe a la siguiente relación entre los valores:

$$N \sim \frac{L}{\eta} \sim Re^{\frac{3}{4}}. \quad (7.1)$$

De la definición del número de Reynolds $Re = \frac{u'L}{\nu}$, como L es proporcional al tamaño del dominio espacial, vemos que la relación $\frac{u'}{\nu}$ permanece constante para un Re determinado. Este efecto se puede comprobar en las últimas columnas del cuadro 7.2 donde vemos tres simulaciones para $N = 128$ y distintos valores de ν . En el cuadro se comprueba que para distintos valores de ν el valor de u' se ajusta para mantener la razón aproximadamente constante lo que hace que las variaciones del valor de Re dependan muy poco de las de ν (en el cuadro las variaciones son de un 15 % aproximadamente respecto a la media). Este efecto también se puede observar en el valor de λ que operando con las ecuaciones 2.3 y 2.18 podemos expresar en función de L y la relación $\frac{u'}{\nu}$. Y consecuentemente en el valor de Re_λ que a su vez es el producto de λ y $\frac{u'}{\nu}$.

7.3. Evolución de $K_{max}\eta$

Uno de los resultados que nos ofrece el programa es una gráfica con la evolución de $K_{max}\eta$ en el último tiempo integral. Con esta gráfica podemos observar el funcionamiento del PID. En la figura 7.2 se puede ver la evolución de los instantes iniciales. Esta gráfica muestra como el PID va ajustando la inyección de energía de forma que en un tiempo integral ha conseguido que el sistema tenga un valor $K_{max}\eta$ muy próximo al deseado. El tiempo que tarda en estabilizarse el sistema en valores próximos al régimen permanente depende de las constantes del PID además de los parámetros propios de la simulación, siendo N un parámetro de gran influencia.

A continuación explicamos brevemente el funcionamiento de un PID y el significado de las constantes que lo regulan. Un controlador PID es aquel que regula un parámetro controlador (en nuestro caso la *viscosidad negativa*) en función del error cometido en otro parámetro

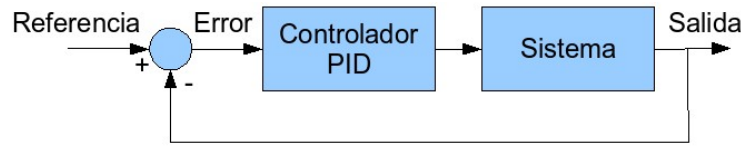


Figura 7.1: Esquema de un controlador PID.

controlado frente al resultado deseado. En la figura 7.1 se representa un esquema de su funcionamiento. Como se puede ver la entrada del controlador es una función del error $e(t)$ que se comete entre el parámetro del sistema que se controla y una referencia externa. El controlador calcula una señal de control que genera a partir de la siguiente expresión:

$$PID = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt}. \quad (7.2)$$

La influencia de un incremento en alguno de los parámetros del controlador en la respuesta del sistema se puede resumir en la siguiente tabla:

Constante	Tiempo de subida	Sobreoscilación	Tiempo de establecimiento	Error en régimen estacionario
K_P	Decrece	Crece	Afecta poco	Decrece
K_I	Decrece	Crece	Crece	Lo elimina
K_D	Afecta poco	Decrece	Decrece	Afecta poco

Cuadro 7.3: Influencia de un incremento en las constantes de un controlador PID.

Experimentalmente se ha comprobado que las constantes del PID funcionan bien en los siguientes rangos: K_P entre 50 y 100, K_I un orden de magnitud menor que K_P , por ejemplo, $K_I = 5$. El efecto de K_D se ha comprobado que no ofrece ninguna ventaja y para ahorrar coste computacional se deja a cero.

Una de las características del código, es que se puede utilizar un filtro esférico que sea más estricto de lo que deseamos simular para poder estabilizar poco a poco el sistema, en la gráfica 7.3 se puede ver un ejemplo típico de evolución del filtro esférico en el que cuando el sistema está cerca de la zona estable el filtro crece y vuelve a estabilizarse hasta que se llega al tamaño deseado. Esta opción se introduce para poder simular cubos con valores de N elevados que son más difíciles de estabilizar.

7.4. Espectro

Para calcular el espectro se utiliza la siguiente fórmula obtenida de la bibliografía

$$E(n) = \frac{1}{2} \sum_{n-0,5 < k < n+0,5} |\mathbf{v}(\mathbf{k})|^2 \quad (7.3)$$

En la figura 7.5 se representa la salida del programa, en ella se pueden ver cuatro líneas que se describen a continuación. La línea amarilla representa el espectro obtenido a partir de

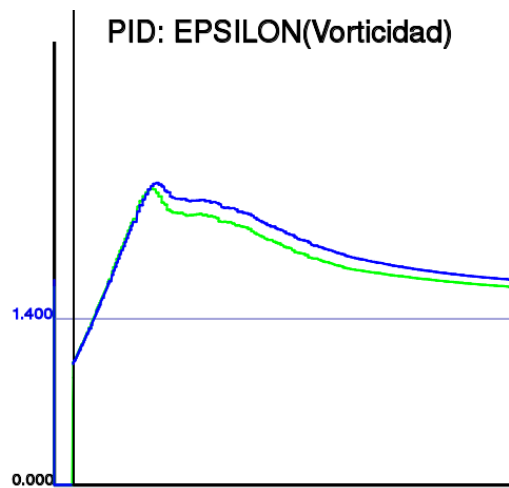


Figura 7.2: Evolución de $K_{max}\eta$ en los instantes iniciales.

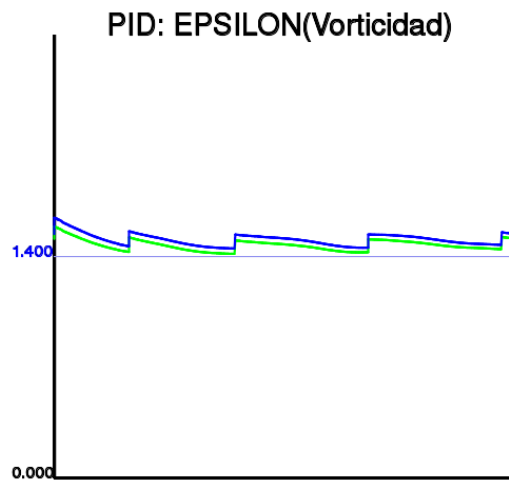


Figura 7.3: Evolución de $K_{max}\eta$. Cambios en el filtro esférico.

la expresión 2.20 con η tal que $K_{max}\eta = 1,4$ y L el real de la simulación en curso. La línea marrón representa la misma gráfica salvo que utiliza el valor real de η . La línea azul es el espectro de la simulación y la verde se corresponde con la siguiente expresión: $E(k)k^{\frac{5}{3}}\epsilon^{-\frac{2}{3}}$.

En la gráfica se puede observar que la zona donde se introduce la energía presenta ciertas variaciones respecto a la teórica, esto es debido al método con el que se introduce la energía en el sistema, se deja para estudios posteriores la implementación de otro tipo de sistemas.

7.5. Invariantes

Uno de los cálculos que mas tiempo de cómputo requiere es el cálculo de los invariantes y su representación gráfica, para su cálculo usamos las fórmulas que aparecen en el capítulo 2. Este cálculo da como resultado un archivo en el que se guardan las parejas de invariantes de cada punto del espacio en un instante dado, el número de datos es tan elevado que se recurre a una gráfica como la de la figura 7.6 que representa los isocontornos de probabilidad de que un punto caiga en una determinada zona de la gráfica para poder estudiarlos.

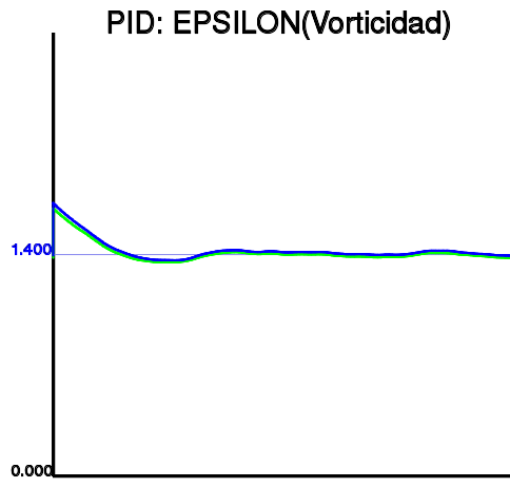


Figura 7.4: Evolución de $K_{max}\eta$. Entrada en régimen permanente.

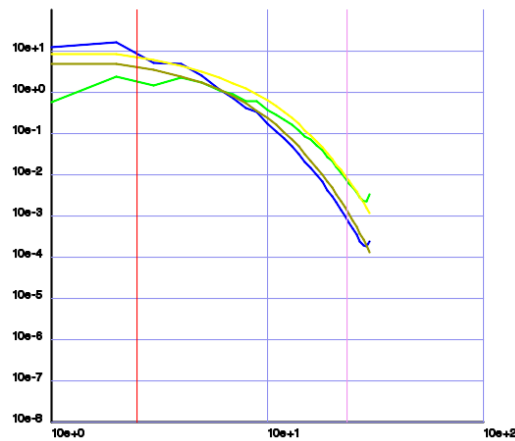


Figura 7.5: Espectro de una simulación con $N=64$ y $\nu = 0,1$.

Una representación alternativa es la de la figura 7.7 que se construye coloreando cada punto de la gráfica según el número de puntos del espacio que tienen sus invariantes en esa zona de la gráfica, de este modo si un punto de la gráfica no se corresponde con ninguna pareja de invariantes en el cubo se deja blanco. Según va correspondiendo a varios puntos del cubo se va cambiando su color de blanco a negro en varios tonos de gris, luego de negro a azul oscuro y finalmente de azul oscuro a celeste donde satura a partir de cierto número de puntos.

7.6. Errores numéricos

Para realizar una comparativa del error numérico, lo definimos como la diferencia máxima entre las componentes espectrales de la simulación y las mismas calculadas con un paso de integración de referencia mucho menor. El error relativo lo definimos como el error numérico dividido entre el valor de la simulación con el paso de integración de referencia.

En la gráfica 7.8 se pueden observar los errores numéricos relativos en función del paso integral. En esta gráfica podemos observar dos comportamientos. Para pasos grandes el error

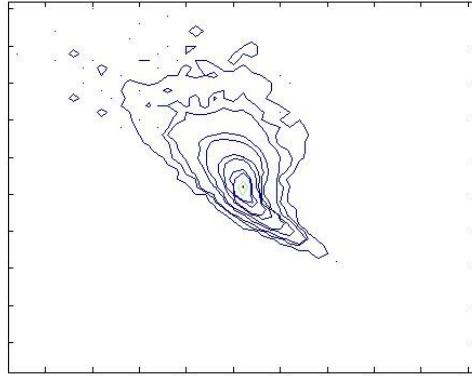


Figura 7.6: Isocontornos de la gráfica de invariantes con $N = 64$ y $\nu = 0,2$.

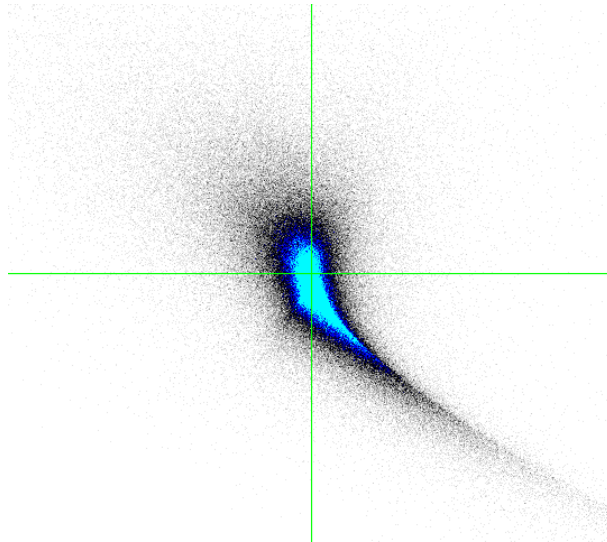


Figura 7.7: Ejemplo de invariantes con $N=64$ y $\nu = 0,2$.

relativo se hace enorme y vuelve los cálculos inservibles y para pasos muy pequeños los errores de redondeo hacen que la disminución del paso integral no redunde en precisión. Es por ello que el código utiliza un sistema que va ajustando el paso integral para mantenerse en un rango en el que los errores relativos sean suficientemente bajos y el número de pasos integrales sea el menor posible. Este rango se ha tomado partiendo de los resultados de la bibliografía en la que se indica que hay que mantener $\frac{dt}{T_\eta} \leq 0,025$ para que los errores relativos se mantengan bajos, por tanto se ha establecido el rango $0,015 \leq \frac{dt}{T_\eta} \leq 0,025$. Experimentalmente se ha comprobado que salir de ese rango no ofrece ventajas si se hace por abajo y vuelve el sistema inestable así como los cálculos erróneos para N suficientemente grande si se hace por arriba. La gráfica 7.8 muestra una comparativa realizada con un experimento con $N = 64$ y precisión simple, para pasos pequeños donde predominan los errores de redondeo se puede disminuir el error usando precisión doble en los cálculos numéricos.

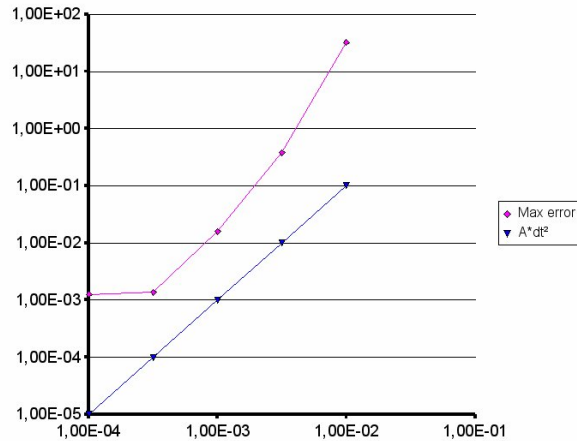


Figura 7.8: Comparativa del error relativo en función del tamaño del paso integral. Simulación con precisión simple, $N = 64$ y $\nu = 0,1$.

7.7. Paralelización

Los resultados de la paralelización pueden apreciarse en las gráficas 7.9 y 7.10. En estas gráficas se representa el tiempo necesario para realizar un mismo experimento en función del número de procesadores y su inverso que nos da una idea de la velocidad. El experimento de prueba consiste en la ejecución de mil pasos integrales con $N = 128$. En las gráficas se puede observar que el aumento del rendimiento no coincide con el número de procesadores (caso ideal), esto es debido a distintos motivos como las pérdidas por las comunicaciones y la parte del código que no se paraleliza. Por otra parte también se observa como para un número de procesadores potencia de 2 se aprecia un incremento del rendimiento relativo, esto es debido a la librería fftw que está optimizada para que el número de procesadores sea potencia de 2.

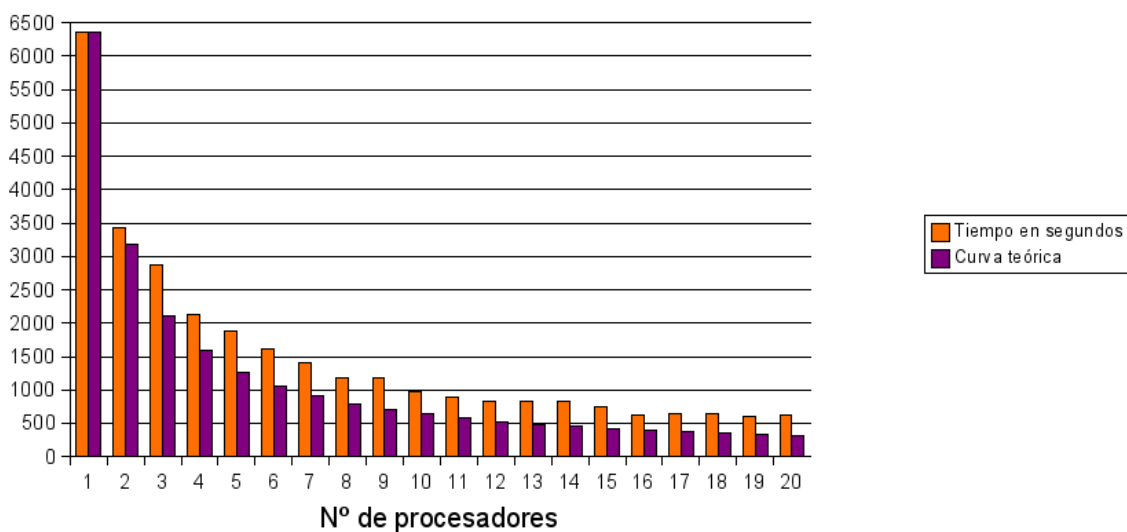


Figura 7.9: Tiempo empleado en una misma tarea en función del número de procesadores.

Se puede comprobar que a partir de cierto número de procesadores no hay un aumento

significativo del rendimiento e incluso si sigue aumentando el número empeora. Esto que en la gráfica 7.10 ocurre para 16 procesadores es un fenómeno habitual en las simulaciones en paralelo conocido como ley de Amdahl y se debe a la proporción de los intercambios de información frente al tiempo de uso de cada procesador. Así para un número pequeño de procesadores éstos consumen la mayor parte del tiempo en procesar los algoritmos pero según va aumentando el número de procesadores el intercambio de información entre los mismos aumenta hasta hacerse apreciable. Por tanto determinar el número óptimo de procesadores necesarios para una determinada simulación permite ahorrar recursos tanto económica como computacionalmente hablando.

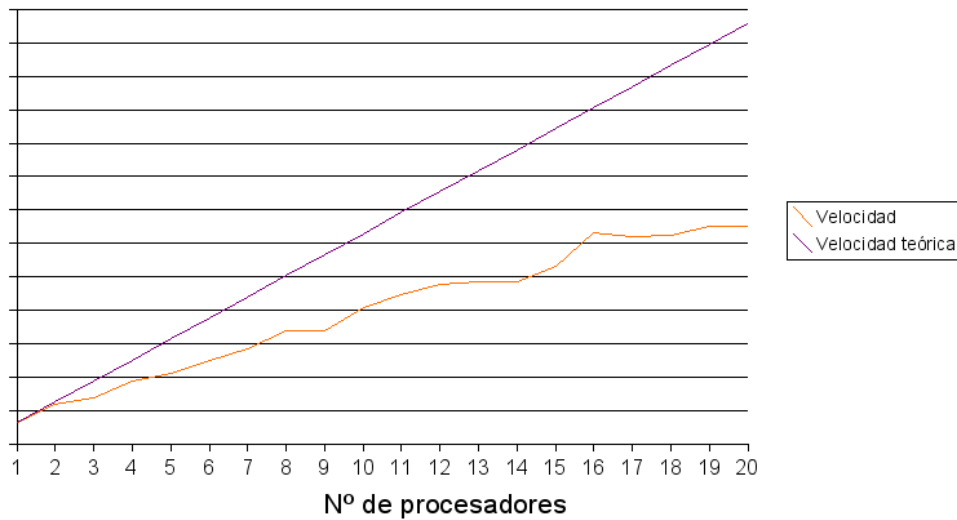


Figura 7.10: Velocidad de ejecución.

