



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

# Mecánica de fluidos computacional: reconstrucción de la solución para el cálculo de flujos y métodos *multigrid*

**Apellidos y nombre:** García-Cuevas González, Luis Miguel ([luiga12@mot.upv.es](mailto:luiga12@mot.upv.es))<sup>1</sup>  
Gil Megías, Antonio ([angime@mot.upv.es](mailto:angime@mot.upv.es))<sup>1</sup>  
Navarro García, Roberto ([ronagar@mot.upv.es](mailto:ronagar@mot.upv.es))<sup>1</sup>  
Quintero Igeño, Pedro Manuel ([pedquiig@mot.upv.es](mailto:pedquiig@mot.upv.es))<sup>1</sup>

**Departamento/Centro:** <sup>1</sup>Departamento de Máquinas y Motores Térmicos  
Universitat Politècnica de València

## Índice general

<b>1. Resumen de las ideas clave</b>	<b>2</b>
<b>2. Objetivos</b>	<b>2</b>
<b>3. Introducción</b>	<b>2</b>
<b>4. Desarrollo</b>	<b>3</b>
4.1. Reconstrucción de la solución y cálculo de flujos . . . . .	3
4.1.1. <i>Upwind de primer orden</i> . . . . .	3
4.1.2. <i>CDS</i> . . . . .	4
4.1.3. <i>Upwind de segundo orden o Linear Upwind Differencing (LUD)</i> . . . . .	4
4.1.4. <i>QUICK</i> . . . . .	5
4.1.5. Esquemas de <i>alta resolución</i> . . . . .	6
4.2. Resolución de los sistemas linealizados de ecuaciones y métodos <i>multigrid</i> . . . . .	6
<b>5. Cierre</b>	<b>8</b>
<b>Bibliografía</b>	<b>10</b>

## 1 Resumen de las ideas clave

En este artículo vamos a presentar cómo se puede reconstruir la solución para calcular los flujos entre celdas en el método de volúmenes finitos para la resolución de problemas de mecánica de fluidos computacional. Además, introduciremos los métodos *multigrid*, con sus ventajas para la resolución de grandes sistemas de ecuaciones linealizados como los encontrados en los problemas de mecánica de fluidos computacional.

## 2 Objetivos

Tras leer detenidamente este documento, el lector ha de ser capaz de:

- Describir los principales métodos de reconstrucción de la solución para el cálculo de los flujos entre celdas.
- Seleccionar el método para el cálculo de flujos en función de criterios objetivos como el número de Péclét.
- Estimar la complejidad numérica de la resolución directa de los grandes sistemas de ecuaciones linealizadas que se dan en los problemas de mecánica de fluidos computacional.
- Describir las limitaciones principales de los métodos iterativos de resolución de los sistemas de ecuaciones anteriormente mencionados.
- Describir los métodos *multigrid* principales y cómo solucionan el problema de la resolución de grandes sistemas de ecuaciones linealizadas.

## 3 Introducción

En el método de los volúmenes finitos aplicado a la mecánica de fluidos computacional se utilizan diversos algoritmos y aproximaciones para resolver las ecuaciones de Navier-Stokes. Entre otros, se encuentran:

- La forma de propagar la solución en el tiempo, usando métodos de integración temporal estándar como los métodos explícitos multipaso o los métodos implícitos para la resolución de ecuaciones rígidas.
- La aproximación para resolver el problema de la turbulencia: con simulación numérica directa (*Direct Numerical Simulation*, DNS), realizando un filtrado espacial (*Large Eddy Simulation*, LES) o realizando un promediado temporal a la Reynolds (*Reynolds-Averaged Navier-Stokes Equations*, RANS), así como combinaciones de las mismas *Detached Eddy Simulation*, DES, por ejemplo).
- En los casos en los que la turbulencia no se resuelve por completo hasta la escala de Kolmogorov, la forma en la que se modelan las escalas no resueltas, con el caso más extendido de los modelos basados en la hipótesis de viscosidad turbulenta de Boussinesq.
- La aproximación para la resolución de las ecuaciones, partiendo del límite incompresible con los métodos de corrección de presión como el PISO o el SIMPLE o los métodos acoplados de resolución basados en densidad que resuelven de forma aproximada de una manera u otra el problema de Riemann entre celdas (métodos FDM o FVM como derivados de Roe o los tipo AUSM).

Pero hay algunos aspectos más a tener en cuenta: tales son la forma en la que se realiza la reconstrucción de la solución para estimar los flujos entre celdas y el método con el que se resuelven, al final, los enormes sistemas de ecuaciones linealizadas que se obtienen.

En la [sección 4](#) se van a plantear estas últimas cuestiones. En la [sección 5](#) se resumen las principales ideas descritas en este documento.

## 4 Desarrollo

En esta sección se mostrarán primero, los principales métodos para llevar a cabo la reconstrucción para calcular los flujos entre celdas, en la [subsección 4.1](#). Por último, se plantearán los principales métodos utilizados para resolver los sistemas de ecuaciones que se obtienen en los problemas de mecánica de fluidos computacional, en la [subsección 4.2](#).

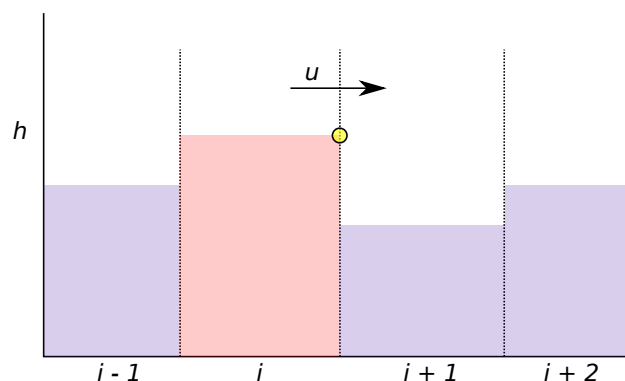
### 4.1 Reconstrucción de la solución y cálculo de flujos

Como sabemos, la resolución mediante volúmenes finitos solo obtiene el valor en nivel medio del vector de estado en cada celda computacional: el valor medio de la densidad en la celda, el valor medio de la energía interna en la celda... Y es necesario calcular los flujos entre celdas. Para hacerlo, hace falta realizar una reconstrucción de la solución y extrapolarla hasta las caras de las celdas, para poder calcular esos flujos.

#### 4.1.1 Upwind de primer orden

Los flujos convectivos tienen una direccionalidad clara: si dos celdas están conectadas y la velocidad va desde la primera a la segunda, es de esperar que el transporte de masa, energía, especies... se produzca de la primera a la segunda. En primera aproximación, se puede suponer que el valor del vector de estado en cada celda es uniforme e igual a su valor medio, como se ve en la [figura 1](#). A este tipo de reconstrucción se le llama *método de Godunov*.

Para el transporte de entalpía en el ejemplo de la figura, el flujo de la misma será directamente igual al caudal de la interfaz multiplicado por la entalpía interna específica media de la primera celda, ya que la velocidad va de la primera a la segunda.

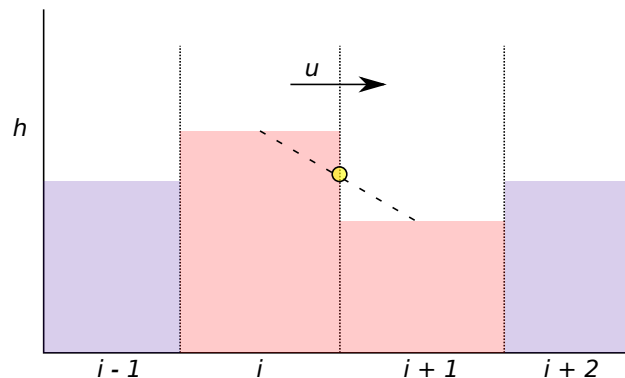


**Figura 1:** Esquema de un método *upwind* de primer orden. Para la cara entre las celdas  $i$  e  $i + 1$ , como la velocidad  $u$  entre las celdas  $i$  e  $i + 1$  va de la primera a la segunda, el flujo de entalpía  $h$  se hace con la de la celda  $i$ . Aunque afecta a dos celdas, de forma efectiva solo se transporta el vector de estado de una de las dos. La matriz del sistema quedará muy compacta.

El método es de primer orden: es muy difusivo y el error decrece linealmente al hacer decrecer el tamaño de las celdas. El orden de convergencia global del problema será probablemente inferior a 1 debido a otros problemas (calidad de la malla, por ejemplo).

#### 4.1.2 CDS

El esquema de diferencias centradas (*CDS*) es un esquema de segundo orden en el que se interpola linealmente la solución entre el centroide de una celda y el de la celda contigua, suponiendo que el vector de estado del centroide es igual al valor medio de la celda. Hay algunas variaciones que introducen otros términos correctores, pero el esquema siempre es principalmente el descrito, como se ve esquemáticamente en la [figura 2](#).



**Figura 2:** Esquema de un método *CDS* de segundo orden. Para la cara entre las celdas  $i$  e  $i + 1$ , el flujo de entalpía se calcula interpolando linealmente entre sus centroides, suponiendo que el valor del vector de estado en dichos centroides coincide con el valor medio conocido de cada celda. Se usa información de dos celdas, con lo que la matriz del sistema queda un poco más llena que con el *upwind* de primer orden.

Al ser un método de segundo orden, presenta menos difusión que el *upwind* de primer orden y el error es mucho más pequeño para un mismo tamaño de malla. Por el mismo motivo que antes, el orden de convergencia global del error del problema será probablemente inferior a 2.

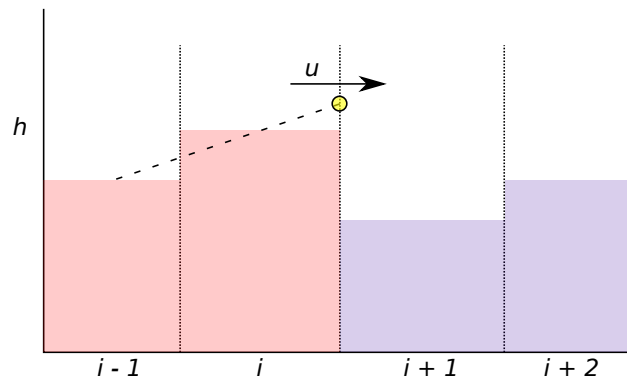
Es un método lineal: los flujos se puede expresar como combinación lineal del vector de estado original (o algunas variables derivadas del mismo). Del teorema de Godunov, es de esperar que el método sea dispersivo e introduzca oscilaciones espurias alrededor de gradientes elevados de la solución. Su uso está limitado debido a que, además, en el caso de problemas con un número de Péclet superior a 2, genera inestabilidades y divergencia en el cálculo. Es muy útil en simulaciones en las que se resuelven escalas lo suficientemente pequeñas como para que el número de Péclet sea pequeño, como son las simulaciones LES.

#### 4.1.3 Upwind de segundo orden o Linear Upwind Differencing (LUD)

En este caso, se usa información de tres celdas: las dos directamente conectadas y otra más, aguas arriba. De forma efectiva, se hace una extrapolación lineal desde los centroides de las dos celdas aguas arriba hasta el centro de la cara, como se ve en la [figura 3](#).

Es menos difusivo que el *upwind* de primer orden, aunque introduce algo de dispersión. Es menos compacto que el *CDS* (necesita información de una celda extra aguas arriba), pero no tiene sus problemas de estabilidad. Salvo que haya motivos que lo justifiquen, debería utilizarse siempre este método antes que el *upwind* de primer orden: el error es mucho menor para un tamaño de malla dado y la difusión numérica también es mucho más pequeña.

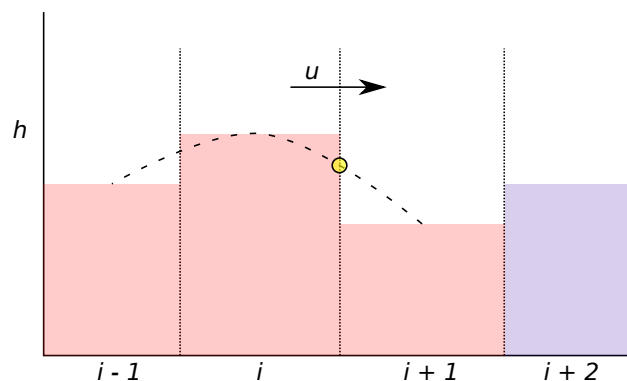
Por supuesto, el orden de convergencia global del problema será probablemente inferior a 2.



**Figura 3:** Esquema de un método *LUD* de segundo orden. Para la cara entre las celdas  $i$  e  $i + 1$ , ya que la velocidad va desde la primera a la segunda, se usa información de las celdas  $i - 1$  e  $i$ . Se extrapola linealmente desde los centroides de dichas celdas, suponiendo que el vector de estado en dichos centroides coincide con su valor medio en cada celda. Para cada par de celdas se usa información de una celda extra, por lo que la matriz del sistema queda menos compacta que en el caso del *upwind de primer orden*. Además, la celda extra de la que se saca información está más apartada, añadiendo complejidad al cálculo de conectividad.

#### 4.1.4 QUICK

En el esquema *QUICK*, *Quadratic Upstream Interpolation for Convective Kinematics*, se usan tres celdas al igual que en el *LUD*. Sin embargo, se usa un interpolador de segundo orden que pasa por el valor medio de las celdas en los centroides de las tres, incluyendo la que está aguas abajo. De este modo también se tiene en cuenta la derivada segunda de la reconstrucción y se tiene un método de tercer orden, como se ve en la [figura 4](#).



**Figura 4:** Esquema de un método *QUICK* de tercer orden. Para la cara entre las celdas  $i$  e  $i + 1$ , ya que la velocidad va desde la primera a la segunda, se usa información de las celdas  $i - 1$ ,  $i$  e  $i + 1$ . Se extrapola cuadráticamente desde los centroides de dichas celdas, suponiendo que el vector de estado en dichos centroides coincide con su valor medio en cada celda. Para cada par de celdas se usa información de una celda extra, por lo que la matriz del sistema queda menos compacta que en los caso del *upwind de primer orden*, *CDS* y *LUD*. Tiene un añadido de complejidad por conectividad como el *LUD*.

El orden de convergencia global del problema será probablemente inferior a 3.

#### 4.1.5 Esquemas de alta resolución

Es posible reducir en parte el problema de dispersión de los métodos de segundo o mayor orden de múltiples formas. Una de ellas es utilizar un *esquema de alta resolución*, en el que se calcula mediante alto orden allá donde la solución es lo suficientemente suave y se transiciona a primer orden allá donde la solución presenta cambios importantes en sus gradientes. La solución se puede ver como una media ponderada entre la de bajo orden y la de alto orden, con coeficientes de ponderación que varían de forma automática en función de la relación de gradientes sucesivos.

Cuando el esquema obtenido, incluyendo la multiplicación por los coeficientes de ponderación, se puede expresar como combinación lineal del vector de estado, se tiene un esquema lineal que, lamentablemente, no puede dejar de ser dispersivo (¡recordemos el teorema de Godunov!).

Si el esquema es no lineal, es posible (¡aunque no cualquier esquema cumple!) que sea realmente no dispersivo: son esquemas realmente *TVD* o *Total Variation Diminishing*, en los que no se introducen extremos espurios en la solución.

Un *esquema de alta resolución* muy extendido es el *MUSCL* (van Leer 1979). De forma efectiva, los ponderadores también se pueden utilizar como limitadores de la extrapolación del vector de estado que se hace en el método *LUD*, reduciendo o aumentando la pendiente de la extrapolación lineal en función de la relación de gradientes entre celdas sucesivas. Esta limitación puede generar un esquema *MUSCL* lineal o no lineal.

## 4.2 Resolución de los sistemas linealizados de ecuaciones y métodos multigrid

Es habitual expresar el sistema de ecuaciones que se obtiene al calcular los flujos, los gradientes en los centros de las caras y el resto de términos del problema de mecánica de fluidos computacional de forma linealizada. El problema final consiste en resolver un sistema de gran tamaño de ecuaciones lineales. Esto se puede hacer de forma directa mediante métodos de factorización (como la factorización LU). Estos métodos tienen un coste elevado, del orden de  $\mathcal{O}(n^3)$  operaciones, siendo  $n$  el número de variables del vector de estado global del sistema. Cuando el número de celdas es grande, el número de operaciones es prohibitivamente alto.

Si la matriz del sistema está bastante vacía, se pueden usar *matrices dispersas* o matrices *sparse*. En ese caso, la información se almacena en memoria de forma distinta a la habitual, sin almacenar los términos nulos de la matriz, lo que reduce enormemente las necesidades de memoria para almacenarlas. Hay métodos de factorización adaptados a matrices *sparse*, cuyo coste depende principalmente del número de elementos no nulos de la matriz. Así, cuanto menor sea el número de puntos de los que depende la reconstrucción, menor será el número de elementos no nulos de la matriz y menor será el coste del cálculo. Esto es interesante: los métodos de alto orden para el cálculo de flujos generan matrices más llenas que los de bajo orden pero, a cambio, permiten usar menos celdas, por lo que suele ser más barato resolver con ellos.

Aun así, la resolución directa por descomposición sigue siendo muy cara incluso con *matrices dispersas*. Para acelerar el cálculo, se pueden usar métodos iterativos, en los que el número de operaciones necesarias es mucho menor por cada iteración (pues el problema se reduce a multiplicar matrices por vectores, sumas y restas, con un coste  $\mathcal{O}(n^2)$  por iteración. Además, los requisitos de memoria se reducen sustancialmente.

Los métodos iterativos, como el de Gauss Seidel, comienzan convergiendo deprisa, pero rápidamente su velocidad de convergencia se reduce hasta ser demasiado lenta. Se puede comprobar que los errores de alta frecuencia (de longitud de onda corta, de tamaño comparable a unas pocas celdas computacionales) convergen a gran velocidad, pero los errores de baja frecuencia (de

longitud de onda grande, de longitud no despreciable frente al tamaño del dominio) convergen despacio. Se puede acelerar enormemente la convergencia si se hace lo siguiente:

- Se calcula en una malla gruesa, donde la solución converge deprisa porque hay pocas celdas y todos los errores son de longitudes de onda similares al tamaño de las celdas.
- Se pasa a una malla más fina, interpolando la solución, y se vuelve a iterar hasta que la solución empieza a converger despacio. Los errores globales, de alta longitud de onda, son pequeños porque ya se redujeron en la malla anterior.
- Se pasa a una malla más fina todavía y se repite el proceso, pasando a mallas cada vez más finas hasta llegar a la inicial, donde convergen a 0 los errores del proceso iterativo de longitudes de onda más pequeñas.

Se puede, incluso, empezar en la malla fina, hacer el recorrido hacia mallas gruesas y volver a las mallas finas de nuevo. O hacer varios procesos de subida y bajada. A este proceso se le llama *GMG (Geometric Multigrid)* o *FAS multigrid (Full-Approximation Storage multigrid)*. Es muy atractivo, pero requiere de múltiples mallados. Se puede simplificar más si, en vez de realizar múltiples mallados, sencillamente se agrupan términos para generar matrices del sistemas con menos términos. Esto es, al final, razonablemente equivalente a hacer *GMG*, pero es mucho más sencillo y agnóstico al problema. A esta aproximación se le llama *AMG (Algebraic Multigrid)*, y es el más utilizado en casi todos los paquetes de simulación mediante mecánica de fluidos computacional. Un esquema de ejemplo se puede ver en la [figura 5](#).

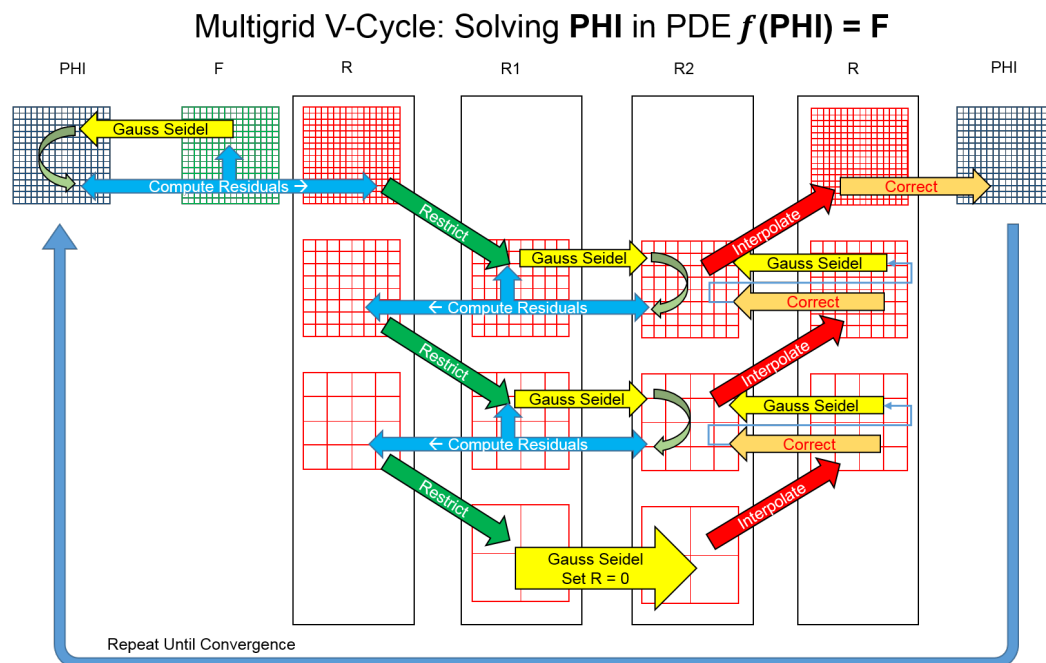


Figura 5: Esquema *multigrid*, de ansariddle 2016.

¡Los esquemas *multigrid* con *matrices dispersas* pueden llegar a tener un coste  $\mathcal{O}(n)$ !

Los ciclos principales clásicos son tres, como se ve en la [figura 6](#), ( $V$ ,  $W$  y  $F$ ), siendo cada uno de ellos más apropiado en un o u otro tipo de problema.



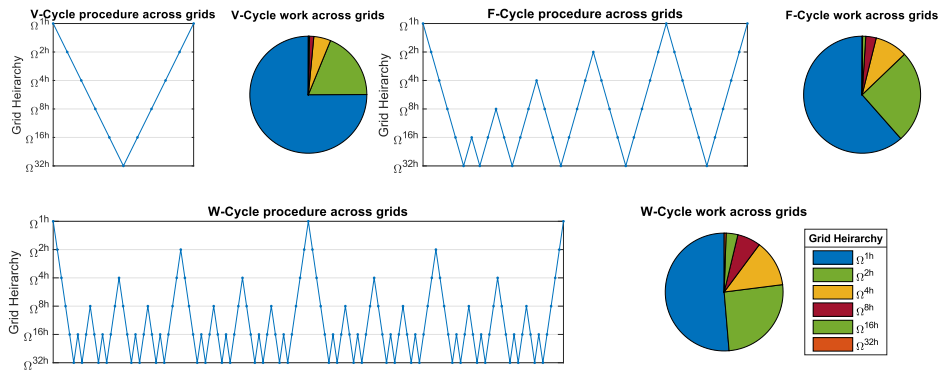


Figura 6: Ciclos en esquemas *multigrid*, de Adilnisar 2019.

## 5 Cierre

A lo largo de este objeto de aprendizaje hemos visto las principales cómo se reconstruye la solución para el cálculo de flujos entre celdas y la resolución de las ecuaciones de forma eficiente mediante métodos *multigrid*. En resumen, se tiene lo siguiente:

- Para resolver un problema de mecánica de fluidos computacional mediante el método de volúmenes finitos habrá que calcular el vector de estado extrapolado desde el valor medio en la celda hasta la frontera entre celdas.
- Hay diversos métodos. Los clásicos son: *upwind*, *CDS* y *QUICK*.
- Para el *upwind*:
  - Al hacer *upwinding*, se comprueba la velocidad en la interfaz. Si la velocidad va desde la primera celda a la segunda, se usa el vector de estado de la primera. Si va de la segunda a la primera, se usa el de la segunda.
  - El método *upwind* es de primer orden, es muy estable y genera mucha difusión numérica. Puede converger más rápido que otros métodos debido a la amortiguación en las perturbaciones provocada por su alta difusividad.
  - ¡La difusividad numérica es muy grande si la dirección de la convección no coincide con los ejes principales de las celdas! Puede reducirse haciendo la malla más fina.
  - También hay *upwind* de segundo orden, que necesita información de una celda extra.
- Para el *CDS* (*Central Differencing Scheme*):
  - El vector de estado se estima en la frontera como la media entre el que se tiene en las celdas contiguas.
  - Es un esquema de segundo orden.
  - Es menos difusivo que el *upwind*.
  - El teorema de Godunov entra en juego: genera oscilaciones en la solución. Es un esquema dispersivo. Esto es más patente en zonas poco suaves del dominio.
  - Puede producir la divergencia del cálculo.
  - Solo debería usarse si el número de Péclet es menor que 2: útil para LES.

- Para el *QUICK* ('Quadratic Upstream Interpolation for Convective Kinematics'):
  - Se realiza una interpolación cuadrática para obtener el vector de estado en la frontera entre celdas.
  - Los puntos para ajustar la parábola se desplazan en la dirección desde la que se produce el transporte, realizándose *upwinding*.
  - Es un esquema de tercer orden. De nuevo, se generarán oscilaciones, siguiendo el teorema de Godunov.
  - Es menos difusivo que el *upwind*.
- Hay otros métodos (*power law*, por ejemplo). Para conseguir alto orden de convergencia y evitar las oscilaciones se pueden usar *esquemas de alta resolución no lineales* o *TVD* (*Total Variation Diminishing*). Los métodos de alta resolución no lineales suelen:
  - Obtener la solución de bajo orden (por *upwind*, por ejemplo),  $w_{lo}$ .
  - Obtener la corrección de la solución para que tenga alto orden,  $w'_{ho}$ .
  - Obtener el ratio de los gradientes sucesivos del vector de estado en todo el dominio,  $r$ .
  - Para cada punto del dominio, aplicar dicha razón de gradientes a una función llamada *función limitadora de flujo* (*flux limiter function*),  $\phi(r)$ . El valor de  $\phi(r)$  es cercano a 0 donde los gradientes cambian bruscamente y mayor (menor que 2, aunque hay excepciones) donde la solución es lo suficientemente suave.
  - El vector de estado se extrapola a las fronteras entre celdas:  $w = w_{lo} + \phi(r) \cdot w'_{ho}$ .
  - Se obtiene una solución de alto orden donde no es altamente dispersiva y se tiende a una de bajo orden (¡y difusiva!) solo muy cerca de donde no se puede usar una de segundo o mayor orden.
  - Es TVD sin violar el teorema de Godunov porque *es no lineal*.
  - Aunque el orden es elevado y se pueden evitar las oscilaciones si es no lineal, el coste por iteración es mucho mayor que en un *upwind* clásico.
  - Los métodos de alta resolución lineales son similares, más baratos que los no lineales pero no son TVD: siguen generando oscilaciones.
- Una vez se tiene la malla, el esquema para obtener los flujos, el esquema para obtener los términos fuente, el modelo de turbulencia y todo lo demás, se obtiene un sistema algebraico (matricial).
- Resolver directamente el sistema puede ser extraordinariamente caro, ya que el número de elementos será muy grande.
- Se usan métodos iterativos para obtener la solución, como el de Gauss-Seidel. Nótese que es un método iterativo para resolver el sistema algebraico. Será, en general, un bucle iterativo dentro de otros bucles iterativos: habrá que resolver el sistema de forma iterativa a cada paso de iteración de un PISO, por ejemplo.
- Los métodos iterativos para resolver sistemas lineales convergen rápido mientras el error es de alta frecuencia (errores locales), y empiezan a converger despacio a la solución cuando es de baja frecuencia (errores globales).

- Los errores de baja frecuencia son equivalente a errores de alta frecuencia si la malla es más gruesa.
- Se puede acelerar la convergencia con un método *multigrid*.
- En general, un método *multigrid*:
  - Itera con una malla fina hasta que converge lentamente.
  - Interpola a una malla más gruesa.
  - Itera en la malla gruesa, donde el error de baja frecuencia se ha convertido en error de alta frecuencia, hasta que converge lentamente.
  - Vuelve a hacer la malla gruesa y a iterar varios niveles.
  - Una vez se llega a una solución convergida en la malla más gruesa, se pasa a una más fina y se vuelve a iterar.
  - Se sube de nivel hasta llegar a la malla más fina. Si en la malla más fina no se alcanza la convergencia tras pocas iteraciones, se vuelve a hacer otro ciclo.
  - Los cálculos en los niveles gruesos son mucho más rápidos que en el nivel fino.
- Esto se suele llamar *GMG (Geometric Multigrid)*. Es fácil de realizar con mallas estructuradas, pero se complica con mallas no estructuradas.
- Para tener una aproximación más general que puede ser más eficiente en mallas no estructuradas al evitar el remallado, se usa *AMG (Algebraic Multigrid)*. Los distintos niveles se obtienen directamente agrupando y promediando en la matriz del sistema.
- AMG es más complejo para mallas estáticas puramente estructuradas, donde conviene hacer GMG.
- Como referencia: algunas implementaciones de GMG también se llaman *FAS multigrid (Full-Approximation Storage multigrid)*.

## Bibliografía

- Adilnizar (2019). *MultigridWork*. Bajo licencia CC BY-SA 4.0 en Wikimedia Commons. URL: <https://upload.wikimedia.org/wikipedia/commons/5/56/MultigridWork.svg> (vid. pág. 8).
- ansariddle (2016). *Multigrid Visualization*. Bajo licencia CC BY-SA 4.0 en Wikimedia Commons. URL: [https://upload.wikimedia.org/wikipedia/commons/8/8c/Multigrid\\_Visualization.png](https://upload.wikimedia.org/wikipedia/commons/8/8c/Multigrid_Visualization.png) (vid. pág. 7).
- van Leer, Bram (jul. de 1979). "Towards the Ultimate Conservative Difference Scheme. V. A Second Order Sequel to Godunov's Method". En: *Journal of Computational Physics* 32 (1), págs. 101-136. DOI: [10.1016/0021-9991\(79\)90145-1](https://doi.org/10.1016/0021-9991(79)90145-1) (vid. pág. 6).