



Estimación experimental del orden de convergencia de un esquema numérico

Modelos y simulación de flujos e instalaciones

Escuela de Ingeniería y Arquitectura (EINA), Universidad de Zaragoza, Zaragoza (España)

Adrián Navas Montilla, Pilar García Navarro
anavas@unizar.es

3 de julio de 2019

1 Objetivos

- Conocer las formas de evaluar el orden de convergencia de un esquema numérico.
- Aplicar la técnica a algunos casos particulares.
- Comprender la utilidad de los esquemas numéricos de alto orden en términos de eficiencia computacional.

2 Introducción

Consideremos un problema de evolución dado por una ecuación en derivadas parciales y denotado por

$$\mathcal{L}(u) = 0 \quad (1)$$

donde \mathcal{L} es un operador diferencial que actúa sobre la variable u . De aquí en adelante, el análisis se centrará en el estudio de la ecuación de transporte lineal

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \quad (2)$$

cuyo operador diferencial se puede escribir como

$$\mathcal{L} \equiv \frac{\partial}{\partial t} + a \frac{\partial}{\partial x} \quad (3)$$

En la ecuación (2), $u \in \mathcal{C} \subseteq \mathbb{R}$ es la variable transportada con velocidad $a \in \mathbb{R}^+$, donde \mathcal{C} es el subconjunto de valores relevantes de u . La coordenada temporal y espacial se define como t y x , respectivamente. La solución, $u = u(x, t)$, está definida en el dominio espacio-temporal $\Omega \times [0, T]$, donde $T \in \mathbb{R}^+$ es el tiempo de simulación y $\Omega = [x_a, x_b]$ es el dominio espacial, con $x_a, x_b \in \mathbb{R}$.

La solución exacta de la ecuación (2) en el punto $x_i = i\Delta x$ y en el tiempo $t^n = n\Delta t$ se denotará por $u(x_i, t^n)$, mientras que la solución numérica se denotará por u_i^n . Esta última se obtendrá mediante la aplicación de un esquema numérico, denotado por

$$\mathcal{N}(u_i^n) = 0 \quad (4)$$

donde \mathcal{N} es un operador que actúa sobre la variable discreta u_i^n . Por ejemplo, si escogemos el esquema upwind de primer orden en combinación con el método de Euler para la resolución de la ecuación (2), tendremos que

$$\mathcal{N}(u_i^n) = \frac{u_i^{n+1} - u_i^n}{\Delta t} + a \left(\frac{u_i^n - u_{i-1}^n}{\Delta x} \right) \quad (5)$$

Por otro lado, también es de utilidad definir la solución exacta de la ecuación discreta, que se denotará por \bar{u}_i^n y que usará para definir el concepto de estabilidad de un esquema numérico.

Existen tres propiedades fundamentales que un esquema numérico debe cumplir para proporcionar una solución aproximada aceptable a una ecuación en derivadas parciales [1]. Estas propiedades son:

1. Consistencia: un esquema numérico es consistente si cuando Δx y Δt tienden a cero, dicho esquema tiende a la ecuación diferencial que aproxima. La diferencia entre el esquema numérico y la ecuación diferencial se denomina *error de truncamiento* y se denota como

$$\epsilon_T = \mathcal{N}(u_i^n) - \mathcal{L}(u_i^n) \quad (6)$$

Este error de truncamiento será de la forma $\epsilon_T = \mathcal{O}(\Delta x^p, \Delta t^q)$, donde p y q los exponentes de los términos de menor orden que aparecen en la diferencia en (6) y representan el orden del esquema numérico. Un esquema numérico será consistente si ϵ_T tiende a cero cuando Δx y Δt tienden a cero. Dicho de otro modo, p y q deberán ser mayores o iguales a 1.

Por ejemplo, el error de truncamiento para el esquema numérico upwind de primer orden en la ecuación (5) viene dado por

$$\epsilon_T = \frac{a\Delta x}{2} (1 - \nu) \partial_{xx} u + \mathcal{O}(\Delta x^2, \Delta t^2), \quad (7)$$

Como el menor exponente del término de error en (7) es el de Δx (con $p = 1$), diremos que la ecuación (5) proporciona una aproximación de primer orden. El procedimiento detallado para obtener el error de truncamiento se muestra en [1].

2. Estabilidad: un esquema numérico es estable si todos los errores (incluso los errores de redondeo de máquina) permanecen acotados en el proceso de integración temporal. Si denotamos como $\bar{\epsilon}_i^n = u_i^n - \bar{u}_i^n$ la diferencia entre la solución numérica y la solución exacta

de la ecuación discreta, la condición de estabilidad se puede formular como el requisito de que $\bar{\epsilon}_i^n$ permanezca acotado para $n \rightarrow \infty$ dado un Δt [1], es decir

$$\lim_{n \rightarrow \infty} |\bar{\epsilon}_i^n| \leq K \quad (8)$$

Nótese que la condición de estabilidad solamente afecta al esquema numérico (a la ecuación discreta) y no exige ningún requisito sobre la ecuación diferencial original.

3. Convergencia: un esquema numérico es convergente si la solución numérica proporcionada por el mismo tiende a la solución analítica de la ecuación diferencial conforme Δx y Δt tienden a cero. Si definimos el error entre la solución numérica y la analítica como

$$\tilde{\epsilon}_i^n = u_i^n - u(x_i, t^n), \quad (9)$$

entonces la condición de convergencia se puede escribir como

$$\lim_{\substack{\Delta x \rightarrow 0 \\ \Delta t \rightarrow 0}} |\tilde{\epsilon}_i^n| = 0. \quad (10)$$

Cabe destacar que el error numérico es también de la forma $\tilde{\epsilon}_i^n = \mathcal{O}(\Delta x^p, \Delta t^q)$, donde p y q son los órdenes de convergencia del esquema numérico respecto al espacio y el tiempo.

Como se ha visto, las propiedades mencionadas anteriormente están relacionadas entre sí. Dicha relación se recoge en el Teorema de Equivalencia de Lax, que enuncia que *para un problema de valor inicial bien planteado y un esquema numérico consistente que lo aproxima, la estabilidad es una condición necesaria y suficiente para la convergencia*. Esto implica que para analizar un esquema numérico, será suficiente analizar la consistencia del esquema (obteniendo el error de truncamiento y orden de convergencia del mismo) así como su estabilidad.

Los métodos anteriormente referidos para la obtención del error de truncamiento y orden de convergencia son analíticos. Sin embargo, también es posible determinar dichas propiedades experimentalmente utilizando las herramientas de simulación. En la siguiente sección se detalla cómo llevar a cabo este procedimiento.

3 Determinación experimental del orden de convergencia de un esquema numérico

Para la determinación experimental del orden de convergencia se escogerá un dominio espacial Ω discretizado en N celdas, Ω_i , del mismo tamaño, Δx , cuyos centros vienen dados por x_i . Como se ha visto anteriormente, el error definido en la ecuación (9) es un error puntual, en $x = x_i$. Para la determinación experimental del orden de convergencia es necesario una medida del error que tenga en cuenta todos los puntos de la malla computacional. Para ello, se hará uso de las denominadas normas vectoriales que proporcionan un valor escalar de la magnitud de un vector. En el caso que nos ocupa, dicho vector se puede definir como el conjunto de errores puntuales en las celdas según la ecuación (9). Así, definiríamos la norma 1 de error como

$$\|\tilde{\epsilon}^n\|_1 = \sum_{i=1}^N |\tilde{\epsilon}_i^n| \Delta x, \quad (11)$$

y la norma infinito de error como

$$\|\tilde{\epsilon}^n\|_\infty = \max_{i=1, \dots, N} |\tilde{\epsilon}_i^n|. \quad (12)$$

Es importante destacar que la norma 1 de error en (11) es el equivalente discreto de la norma integral

$$\|\tilde{\epsilon}^n\|_1 = \int_{\Omega} |\tilde{\epsilon}^n(x)| dx \quad (13)$$

donde, si suponemos que conocemos la solución numérica en cualquier punto la denotaremos como, $u^n(x)$ y que el dominio se ha discretizado en N celdas Ω_i , tenemos

$$\|\tilde{\epsilon}^n\|_1 = \sum_{i=1}^N \int_{\Omega_i} |u^n(x) - u(x, t^n)| dx \quad (14)$$

Esta definición se utiliza en métodos numéricos en volúmenes finitos ya que las cantidades discretas representan promedios en las celdas. De aquí en adelante, también trabajaremos con valores promedio en las celdas y calcularemos el error como

$$\|\tilde{\epsilon}^n\|_1 = \sum_{i=1}^N |\bar{u}_i^n - \bar{u}(x_i, t^n)| \Delta x \quad (15)$$

y

$$\|\tilde{\epsilon}^n\|_{\infty} = \max_{i=1, \dots, N} |\bar{u}_i^n - \bar{u}(x_i, t^n)|. \quad (16)$$

donde la solución numérica promedio en la celda será

$$\bar{u}_i^n = \frac{1}{\Delta x} \int_{\Omega_i} u(x)^n dx \quad (17)$$

y la solución exacta promedio en la celda será

$$\bar{u}(x_i, t^n) = \frac{1}{\Delta x} \int_{\Omega_i} u(x, t^n) dx \quad (18)$$

3.1 Determinación del orden de convergencia conocida la solución analítica

Para realizar un análisis de convergencia conocida la solución analítica, es necesario calcular el error en diferentes mallas con distinto número de celdas y comparar dichos errores entre ellos. Para dos mallas $\mathcal{M}1$ y $\mathcal{M}2$ con número de celdas $N_{\mathcal{M}1}$ y $N_{\mathcal{M}2}$, y tamaño de celda $\Delta x_{\mathcal{M}1}$ y $\Delta x_{\mathcal{M}2}$, la norma vectorial de error en cada malla se denotará como $\|\tilde{\epsilon}^n\|_{(\cdot)}^{\mathcal{M}1}$ y $\|\tilde{\epsilon}^n\|_{(\cdot)}^{\mathcal{M}2}$ respectivamente. El orden global de convergencia se calculará como

$$\sigma = \lim_{\substack{\Delta x \rightarrow 0 \\ \Delta t \rightarrow 0}} \frac{\log \left(\frac{\|\tilde{\epsilon}^n\|_{(\cdot)}^{\mathcal{M}1}}{\|\tilde{\epsilon}^n\|_{(\cdot)}^{\mathcal{M}2}} \right)}{\log \left(\frac{\Delta x_{\mathcal{M}1}}{\Delta x_{\mathcal{M}2}} \right)}, \quad (19)$$

donde (\cdot) puede ser 1 o ∞ .

Es muy importante darse cuenta que en métodos numéricos de evolución temporal, el paso espacial, Δx , y el temporal, Δt , están relacionados mediante la condición de CFL

$$\Delta t = CFL \cdot \frac{\Delta x}{\lambda}, \quad (20)$$

donde λ es la velocidad de propagación de onda y CFL una constante comprendida en el intervalo $[0,1]$. Por lo tanto, para realizar un análisis de convergencia, será suficiente llevar a cabo un refinamiento espacial (hacer tender $\Delta x \rightarrow 0$). Según (20), el paso temporal también se verá automáticamente reducido al refinar en el espacio.

El orden de convergencia σ , vendrá limitado por el menor orden de la discretización espacial y temporal. Así, podemos decir que $\sigma = \min(p, q)$.

Dado que la definición en (19) incluye un límite para $\Delta x, \Delta t \rightarrow 0$, lo cual no es posible debido al coste computacional que involucraría, se propone el siguiente procedimiento para calcular el orden de convergencia experimentalmente con un número finito de simulaciones:

1. Escoger un conjunto de r mallas de cálculo, tal que el numero de celdas de cada una sea

$$N_{\mathcal{M}_j} = N_0 \cdot 2^{j-1}, \quad \text{con } j = 1, \dots, r \quad (21)$$

y N_0 el número de celdas de la malla inicial (por ejemplo, se aconseja escoger $N_0 = 20$).

2. Para las r mallas, calcular la solución numérica en un tiempo t^n y expresar su valor promedio en celda según (17). Nótese que en esquemas en volúmenes finitos, la solución numérica es un valor promedio.
3. Para las r mallas, calcular la solución analítica como promedio en celdas según (17).
4. Calcular la norma de error deseada con la solución numérica y analítica de los pasos 2 y 3 (por ejemplo, usando (15) y (16)).
5. Calcular el orden de convergencia para cada pareja de mallas con la siguiente fórmula:

$$\sigma_j = \frac{\log \left(\frac{\|\bar{\epsilon}^n\|_{(\cdot)}^{\mathcal{M}_j^{j-1}}}{\|\bar{\epsilon}^n\|_{(\cdot)}^{\mathcal{M}_j^j}} \right)}{\log(2)}, \quad \text{con } j = 2, \dots, r. \quad (22)$$

6. Comprobar si $|\sigma_r - \sigma_{r-1}| \leq C$, donde $C \ll \sigma$ es una tolerancia para comprobar que se ha llegado a la región asintótica de convergencia. Si esta condición no se cumple, se deben escoger más puntos y repetir el proceso con los nuevos puntos.

Ejercicio 1. Para comenzar, se propone explorar el siguiente código que se encuentra en un fichero llamado `singleCall.m`:

```

1  % EXECUTION OF A SINGLE SIMULATION
2  %
3  % (c) Adrian Navas Montilla, Universidad de Zaragoza, anavas@unizar.
   es
4  % This script calls the DG_RK function for a given configuration.
5  % Notation:
6  % N: number of cells
7  % L: length of the domain
8  % T: simulation time
9  % CFL: CFL number
10 % K: order of accuracy
11 % lambda: advection velocity in the case of the linear scalar
    problem
12 % equation: problem solved-> 1=linear scalar, 2=burgers
13
14 clc; clear all; close all;
15
16 %% Problem definition
17
18 L=1;
```

```

19 T=0.05;
20 CFL=0.1;
21 N=10;
22 equation=1;
23 lambda=1.0;
24
25 K=2;
26
27 %% Execution of the solver
28
29 [u_avg,u_ex] = DG_RK( N,L,T,CFL,K,lambda,equation );
30
31 %% Printing the results
32
33 x=zeros(1,length(N));
34 for n=1:N
35     x(n)=L/N/2+L/N*(n-1); %coordinates of cell centers
36     fprintf('La solucion numerica y exacta en la celda x_i=%3.3f es u
           =%7.7f y u_ex=%7.7f\n',x(n),u_avg(n),u_ex(n));
37 end

```

Este script llama a la función `DG_RK(N,L,T,CFL,K,lambda,equation)` que se encuentra en el fichero `DG_RK.m` y que contiene un algoritmo de resolución basado en el método de Galerkin Discontinuo. Este método se basa en una reconstrucción espacial de orden arbitrario (en esta función se ha implementado hasta orden 4) y una integración temporal usando un método Runge-Kutta de orden 3 (SSPRK3). Éste ultimo, al tener un orden inferior al orden espacial (en el caso de $K = 4$), limitará el máximo orden de convergencia que se puede alcanzar. La `DG_RK(N,L,T,CFL,K,lambda,equation)` función proporciona como valores de salida dos vectores, el vector `u_avg` que contiene la solución numérica promediada en las celdas y el vector `u_ex` que contiene la solución exacta promediada en las celdas. Nótese que esta función realiza los pasos 2 y 3 del procedimiento anteriormente detallado.

La función `DG_RK(N,L,T,CFL,K,lambda,equation)` resuelve tanto la ecuación lineal de transporte (`equation=1`) como la ecuación de Burgers' (`equation=2`). La solución analítica sólo se calcula para el primer caso. La condición inicial viene dada por la función

$$u(x, 0) = \cos(2\pi x) + 1, \quad (23)$$

y las condiciones de contorno se consideran periódicas, esto es $u(x = 0, t) = u(x = L, t)$.

Una vez guardado el fichero `singleCall.m` en la carpeta de trabajo, se ejecuta en la consola de Matlab de este modo:

```
>> singleCall.m
```

obteniéndose el siguiente resultado en la consola:

```

La solucion numerica y exacta en la celda x_i=0.050 es u=1.9833101 y u_ex=1.9836316
La solucion numerica y exacta en la celda x_i=0.150 es u=1.7974914 y u_ex=1.7957747
La solucion numerica y exacta en la celda x_i=0.250 es u=1.3070581 y u_ex=1.3039589
La solucion numerica y exacta en la celda x_i=0.350 es u=0.6993390 y u_ex=0.6960411
La solucion numerica y exacta en la celda x_i=0.450 es u=0.2064623 y u_ex=0.2042253
La solucion numerica y exacta en la celda x_i=0.550 es u=0.0166899 y u_ex=0.0163684
La solucion numerica y exacta en la celda x_i=0.650 es u=0.2025086 y u_ex=0.2042253
La solucion numerica y exacta en la celda x_i=0.750 es u=0.6929419 y u_ex=0.6960411
La solucion numerica y exacta en la celda x_i=0.850 es u=1.3006610 y u_ex=1.3039589
La solucion numerica y exacta en la celda x_i=0.950 es u=1.7935377 y u_ex=1.7957747

```

además de dos gráficas que muestran tanto la solución numérica puntual que depende de x (Figura 1 - izquierda), como la solución promediada en el espacio (Figura 1 - derecha). Se proponen las siguientes tareas:

- Observar como al variar K en la línea 25, la representación puntual de la solución numérica puede ser constante a trozos ($K = 1$), lineal a trozos ($K = 2$) o cuadrática a trozos ($K = 3$).
- Explorar el límite de estabilidad de los esquemas numéricos (para distintos valores de K), observando el valor de CFL a partir del cual el esquema se vuelve inestable, amplificándose las pequeñas perturbaciones numéricas.

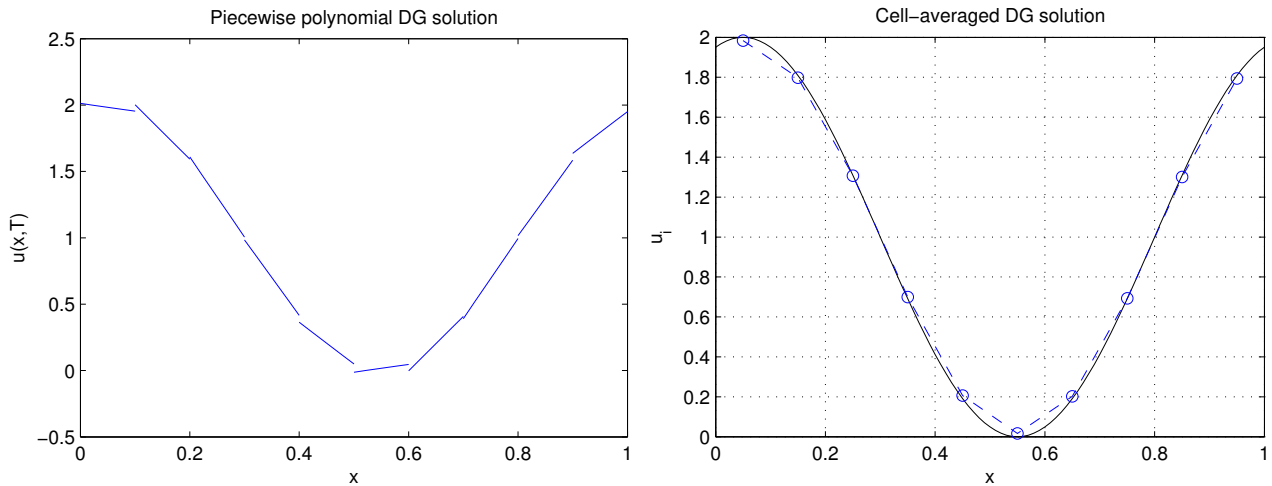


Figure 1: Representación

Ejercicio 2. En este ejercicio se propone implementar el cálculo del orden de convergencia siguiendo la metodología explicada con anterioridad. Para ello, se utilizará como plantilla el fichero `mainConvergence.m` en el que hay varios cálculos sin acabar:

```

1  % CONVERGENCE RATE TEST FOR THE DG-RK SCHEME
2  %
3  % (c) Adrian Navas Montilla, Universidad de Zaragoza, anavas@unizar.
   es
4  % This script calls the DG_RK function and carries out a convergence
5  % rate test comparing with the exact solution.
6  % Notation:
7  % N: number of cells
8  % L: length of the domain
9  % T: simulation time
10 % CFL: CFL number
11 % K: order of accuracy
12 % lambda: advection velocity in the case of the linear scalar
   problem
13 % equation: problem solved-> 1=linear scalar, 2=burgers
14
15 clc; clear all; close all;
16
17 %% Problem definition
18
19 L=1.0;
20 T=0.05;
```

```

21 CFL=0.1;
22 equation=1; %1=scalar, 2=burgers
23 lambda=1.0;
24
25 K=2; %is the order of the scheme (and number of DOFs)
26
27 %% Convergence test
28
29 %The number of cells of the grids used are defined below:
30 Narray=[ ];
31 %
32
33 L1=zeros(1,length(Narray));
34 Linf=zeros(1,length(Narray));
35 j=0;
36 for N=Narray
37     j=j+1;
38     u_avg=zeros(1,N); %initialization of the array containing the
        solution
39     u_ex=zeros(1,N); %initialization of the array containing the exact
        sol.
40     if equation==1
41         [u_avg,u_ex] = DG_RK( N,L,T,CFL,K,lambda,equation );
42     else
43         fprintf('This procedure only works if equation=1:\n');
44         pause
45     end
46     DX=L/N;
47     %calculation of the error norms
48     L1(j)=0.0;
49     Linf(j)=0.0;
50     for i=1:N
51         L1(j)= ;
52         Linf(j)= ;
53     end
54 end
55
56 %calculation of the order of convergence
57 fprintf('\n');
58 fprintf('The accuracy test yields:\n');
59 ord(1:2,1)=0.0;
60 for j=2:length(Narray)
61     ord(1,j)= ;
62     ord(2,j)= ;
63     fprintf('order L1 %2.2f and order L_inf %2.2f. \n',ord(1,j),ord(2,
        j));
64 end
65
66 %representation of the results
67 figure(4)
68 ...
69 ...

```


En el código anterior aparecen varias líneas resaltadas, correspondientes a los pasos 1, 4 y 5, las cuales hay que completar. Los pasos 2 y 3 vienen dados por la función `DG_RK(N,L,T,CFL,K,lambda,equa)` como se ha mostrado en el Ejercicio 1. Para completar los pasos 1, 4 y 5, se requiere lo siguiente:

- Paso 1: Elegir las mallas de cálculo y generar un array, `Narray`, cuyos elementos contengan los números de celdas de cada una de las mallas. Para ello se utilizará la expresión en la ecuación (30).
- Paso 4: Introducir la fórmula que computa la norma 1 y la norma infinito de error (ecuaciones (15) y (16)), en el código denotadas por `L1(j)` y `Linf(j)`, donde `j` es el índice que recorre el array de mallas. Nótese que `j` toma valores entre 1 y `length(Narray)`, éste último referido como `r` en este documento (por ejemplo en la ecuación (30)).
- Paso 5: El orden de convergencia, denotado como σ_j en este documento, se almacena en las variables `ord(1,j)` y `ord(2,j)`, correspondientes a la norma 1 y norma infinito de error. Para calcular ambos, habrá que implementar la fórmula en la ecuación (22).

Ahora, se requiere representar gráficamente los resultados. Una representación logarítmica del error frente al número de celdas permite evaluar visualmente la convergencia del método numérico ya que la pendiente de la gráfica corresponderá al orden de convergencia. En la línea 67 (y siguientes), se pide realizar una representación logarítmica de $\|\tilde{\epsilon}^n\|_1$ y $\|\tilde{\epsilon}^n\|_\infty$, es decir de `L1(j)` y `Linf(j)`, en función del número de celdas. Para ello, se usará el comando `loglog` del cual se puede obtener información escribiendo en la consola:

```
>> help loglog
```

El resultado esperado debe ser similar al mostrado en la figura 2.

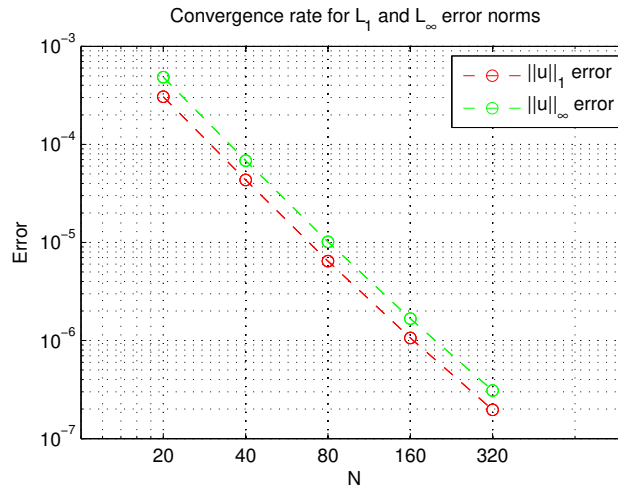


Figure 2: Representación logarítmica del error frente al numero de celdas para el esquema de orden 2.

Finalmente, el objetivo es calcular el orden de convergencia para $K = 1$, $K = 2$, $K = 3$ y $K = 4$ y comparar dichos resultados. ¿Se alcanza en todos ellos el orden esperado?. ¿Por qué en el caso de $K = 4$ el orden de convergencia no llega a 4?.

3.2 Determinación del orden de convergencia cuando la solución analítica es desconocida

Hay ecuaciones en las que la solución analítica es muy difícil de obtener o incluso imposible. En estos casos, no podemos obtener un error numérico comparando la solución numérica con

la analítica y por tanto no podemos usar el procedimiento anterior para evaluar el orden de convergencia del esquema numérico. Sin embargo, es posible utilizar un procedimiento similar en el que se calcula el error entre soluciones numéricas en mallas distintas. Este método se basa en el *principio de la doble malla*, cuya idea es que si calculamos la solución numérica en dos mallas $\mathcal{M}1$ y $\mathcal{M}2$, dicha solución se puede expresar como

$$\{\bar{u}_i^n\}_{\mathcal{M}1} = \bar{u}(x_i, t^n) + \{\tilde{\epsilon}_i^n\}_{\mathcal{M}1}, \quad \text{con } i = 1, \dots, N_{\mathcal{M}1} \quad (24)$$

$$\{\bar{u}_k^n\}_{\mathcal{M}2} = \bar{u}(x_k, t^n) + \{\tilde{\epsilon}_k^n\}_{\mathcal{M}2}, \quad \text{con } k = 1, \dots, N_{\mathcal{M}2} \quad (25)$$

donde $\{\tilde{\epsilon}_i^n\}_{\mathcal{M}1}$ y $\{\tilde{\epsilon}_k^n\}_{\mathcal{M}2}$ es el error entre la solución numérica promedio en la celda y la solución exacta promedio en la celda. Dichos errores serán del orden $\mathcal{O}(\Delta x^p, \Delta t^q)$. Para facilitar el procedimiento, se considerará que el ratio entre el número de celdas de dos mallas consecutivas, en este caso $\mathcal{M}1$ y $\mathcal{M}2$, será 2, es decir $N_{\mathcal{M}2} = 2N_{\mathcal{M}1}$. La diferencia entre (24) y (25) se calcula de la siguiente manera

$$\{\tilde{\epsilon}_i^n\}_{1,2} \equiv 0.5 \left(\{\bar{u}_{2i}^n\}_{\mathcal{M}2} + \{\bar{u}_{2i-1}^n\}_{\mathcal{M}2} \right) - \{\bar{u}_i^n\}_{\mathcal{M}1}, \quad \text{con } i = 1, \dots, N_{\mathcal{M}1} \quad (26)$$

donde vemos que la parte correspondiente a la integral de la solución exacta se anula y nos queda un término que es de la forma $\{\tilde{\epsilon}_i^n\}_{1,2} = \mathcal{O}(\Delta x^p, \Delta t^q)$. Al igual que en el apartado anterior, es posible definir una norma de error para cuantificar la magnitud del error en todo el dominio computacional. En este caso también definiremos la norma 1 y norma infinito de error, asociada a dos mallas j y $j+1$, como

$$\|\tilde{\epsilon}^n\|_{1,j} = \sum_{i=1}^N \left| \{\tilde{\epsilon}_i^n\}_{j,j+1} \right| \Delta x, \quad \text{con } j = 1, \dots, r-1 \quad (27)$$

y

$$\|\tilde{\epsilon}^n\|_{\infty,j} = \max_{i=1,\dots,N} \left| \{\tilde{\epsilon}_i^n\}_{j,j+1} \right|, \quad \text{con } j = 1, \dots, r-1 \quad (28)$$

donde r es el número de mallas utilizadas.

El orden de convergencia se aproximará con

$$\sigma_j = \lim_{\substack{\Delta x \rightarrow 0 \\ \Delta t \rightarrow 0}} \frac{\log \left(\frac{\|\tilde{\epsilon}^n\|_{(\cdot),j}}{\|\tilde{\epsilon}^n\|_{(\cdot),j+1}} \right)}{\log(2)}, \quad \text{con } j = 1, \dots, r-2 \quad (29)$$

donde (\cdot) puede ser 1 o ∞ .

El procedimiento para calcular el orden de convergencia cuando no se dispone de solución exacta se resume en las siguientes instrucciones:

1. Escoger un conjunto de r mallas de cálculo, tal que el numero de celdas de cada una sea

$$N_{\mathcal{M}j} = N_0 \cdot 2^{j-1}, \quad \text{con } j = 1, \dots, r \quad (30)$$

y N_0 el número de celdas de la malla inicial (por ejemplo, se aconseja escoger $N_0 = 20$).

2. Para las r mallas, calcular la solución numérica en un tiempo t^n y expresar su valor promedio en celda según (17). Nótese que en esquemas en volúmenes finitos, la solución numérica es un valor promedio.
3. Para cada par de mallas, calcular la diferencia entre ellas usando la ecuación (26) en su forma genérica

$$\{\tilde{\epsilon}_i^n\}_{j,j+1} \equiv 0.5 \left(\{\bar{u}_{2i}^n\}_{\mathcal{M}2} + \{\bar{u}_{2i-1}^n\}_{\mathcal{M}2} \right) - \{\bar{u}_i^n\}_{\mathcal{M}1}, \quad \text{con } i = 1, \dots, N_{\mathcal{M}1}, \quad j = 1, \dots, r-1 \quad (31)$$

4. Calcular la norma de error deseada para los vectores de diferencias $\{\tilde{\varepsilon}_i^n\}_{j,j+1}$ calculados en el paso 3 (ecuación (31)) utilizando (27) y (28) por ejemplo.
5. Calcular el orden de convergencia con la ecuación (29).
6. Comprobar si $|\sigma_{r-2} - \sigma_{r-3}| \leq C$, donde $C \ll \sigma$ es una tolerancia para comprobar que se ha llegado a la región asintótica de convergencia. Si esta condición no se cumple, se deben escoger más puntos y repetir el proceso con los nuevos puntos.

Ejercicio 3. En este ejercicio se considera la ecuación de Burgers

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) = 0 \quad (32)$$

que se puede ver como una versión más general de la ecuación (2) en la que la velocidad $\lambda(u) = u$ es una función no lineal. La ecuación de Burgers es una ecuación no lineal que modela diversos fenómenos como tráfico urbano o dinámica de gases. Para este último caso, la ecuación (32) se deriva de las ecuaciones de Navier-Stokes para la conservación de momento

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot (\nabla \mathbf{u}) \right) - \nu \nabla^2 \mathbf{u} = \rho \mathbf{f}_m - \nabla p \quad (33)$$

donde ρ representa la viscosidad, \mathbf{u} el vector velocidad, ν la viscosidad cinemática, \mathbf{f}_m las fuerzas másicas y p la presión del gas. En un caso 1D en el que se desprecia el gradiente de presión, las fuerzas volumétricas y la viscosidad, se obtiene (32).

La obtención de una solución analítica para ecuación (32) no es trivial. Por lo tanto, es un buen ejemplo para motivar el uso del procedimiento de cálculo del orden de convergencia utilizando el principio de la doble malla. Para ello, se utilizará como plantilla el fichero `mainNonlinear.m` que está incompleto:

```

1  %   CONVERGENCE RATE TEST FOR THE DG-RK SCHEME
2  %
3  %   (c) Adrian Navas Montilla, Universidad de Zaragoza, anavas@unizar.
    es
4  %   This script calls the DG_RK function and carries out a convergence
5  %   rate test using the two-grid methodology.
6  %   Notation:
7  %   N: number of cells
8  %   L: length of the domain
9  %   T: simulation time
10 %   CFL: CFL number
11 %   K: order of accuracy
12 %   lambda: advection velocity in the case of the linear scalar
    problem
13 %   equation: problem solved-> 1=linear scalar, 2=burgers
14
15 clear all; close all;
16
17 %% Problem definition
18
19 %The problem configuration is defined here
20 L=1;
21 T=0.05;
22 CFL=0.1;
```

```

23 equation=2; %1=scalar, 2=burgers
24 lambda=1.0;
25
26 K=3; %is the order and number of DOFs
27
28 %% Convergence test
29
30 %The grids used are defined here:
31 Narray=[20 40 80 160 320];
32 %
33
34 L1=zeros(1,length(Narray));
35 Linf=zeros(1,length(Narray));
36 j=0;
37 for N=Narray
38     j=j+1;
39     u_avg=zeros(1,N);
40
41     [u_avg,~] = DG_RK( N,L,T,CFL,K,lambda,equation );
42
43     L1(j)=0.0;
44     Linf(j)=0.0;
45     for i=1:N/2
46
47     end
48 end

```

Como se observa, ahora `equation=2`, lo que implica el cálculo de la ecuación de Burgers. El objetivo de este ejercicio es utilizar la estructura propuesta en este script (si se desea, se puede empezar desde cero) para implementar el procedimiento de cálculo descrito anteriormente en la sección 3.2. Se presentarán resultados para $K = 1$, $K = 2$ y $K = 3$.

Referencias

- [1] C. Hirsch. *Numerical computation of internal and external flows: Computational methods for inviscid and viscous flows*. 1990.