



Revista de Matemática: Teoría y
Aplicaciones

ISSN: 1409-2433

mta.cimpa@ucr.ac.cr

Universidad de Costa Rica
Costa Rica

RUIZ V., JORGE MAURICIO; LEÓN E., DIEGO A.
UNA INTRODUCCIÓN AL MÉTODO LIBRE DE MALLA DE CONJUNTOS FINITOS DE
PUNTOS

Revista de Matemática: Teoría y Aplicaciones, vol. 23, núm. 2, julio, 2016, pp. 389-408
Universidad de Costa Rica
San José, Costa Rica

Disponible en: <http://www.redalyc.org/articulo.oa?id=45349329004>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

UNA INTRODUCCIÓN AL MÉTODO LIBRE DE MALLA DE CONJUNTOS FINITOS DE PUNTOS

AN INTRODUCTION TO THE MESHLESS FINITE POINTSET METHOD

JORGE MAURICIO RUIZ V.* DIEGO A. LEÓN E.†

*Received: 24 Aug 2015; Revised: 20 Apr 2016;
Accepted: 5 May 2016*

*Departamento de Matemáticas, Universidad Nacional de Colombia, Bogotá D.C., Colombia.
E-Mail: jmruizv@unal.edu.co

†Misma dirección que/Same address as: E-Mail: daleone@unal.edu.co

Resumen

Este trabajo propone una introducción corta y simple al método libre de malla conocido con el nombre de método de conjuntos finitos de puntos (FPM). Se describen los conceptos importantes que involucra el método como: la generación del sistema de puntos, búsqueda de puntos vecinos, aproximación de las derivadas espaciales mediante el método de mínimos cuadrados móviles y la solución de sistemas de ecuaciones diferenciales ordinarias resultantes. Como aplicación del método FPM se soluciona la ecuación viscosa y no viscosa de Burgers. Las soluciones numéricas son comparadas con la solución analítica y se realiza un análisis de convergencia del método vía experimentación numérica. Se proveen las rutinas de MATLAB de los pasos fundamentales del método FPM, que pueden ser usados para resolver problemas más complicados.

Palabras clave: métodos sin malla; mínimos cuadrados móviles; ecuación de Burgers.

Abstract

In this work we propose a short and simple introduction of the meshless method known as finite pointset method (FPM). We describe the main concepts involved in the FPM method like: the pointset generation, point neighbors search, the spatial derivatives approximation by the moving least square method and the solution of the resultant ordinary differential system. As application of the method we solve the viscous and inviscid Burgers equation. The numerical solutions are compared with the analytical solution and a convergence analysis via numerical experimentation is performed. We provide the MATLAB codes for the main steps of the FPM method, which can be used to solve more complex problems.

Keywords: meshless method; moving least square method; Burgers equation.

Mathematics Subject Classification: 65M28, 76M28, 97-01, 97N99.

1 Introducción

El método de conjuntos finitos de puntos (FPM, por sus siglas en inglés) es una técnica numérica libre de malla para el análisis y simulación de problemas de la mecánica de fluidos. Este fue desarrollado originalmente para evitar las dificultades que tiene la generación de mallas espaciales (estructuradas o no estructuradas) y su adaptación en el tiempo en la simulación numérica de fenómenos como: discontinuidades de la solución, ondas de choque, problemas de interacción entre fluido y estructura, y en problemas de superficie libre o fronteras móviles [5] [13]. Este método libre de malla puede verse como una variante del

método de diferencias finitas generalizadas (MDFG) propuesto en [3]. La diferencia está en el hecho que el método MDFG busca a partir de mallas irregulares o no estructuradas, esquemas de diferencias finitas, que serán la versión discreta de la ecuación diferencial a resolver. Mientras que, el método FPM calcula de manera directa en cada nodo de la "malla" las aproximaciones de la función y sus derivadas, sin tener que escribir explícitamente una versión de diferencias finitas de la ecuación tratada.

Pese al éxito del método de conjuntos finitos de puntos en la solución práctica de problemas complejos en mecánica de fluidos provenientes de la industria (p. ej. simulación de procesos de fabricación de envases de vidrio [8], inflado de la bolsa de aire de los automóviles, llenado y oleaje del tanque de combustible de automóviles [12]) y su implementación en el software comercial como NO-GRID points software; la literatura disponible sobre éste es muy escasa, el material expuesto no es introductorio, ni facilita al lector la implementación de los algoritmos.

El objetivo de este trabajo es presentar una introducción sucinta y simple del método de conjuntos finitos de puntos a partir de su implementación para la solución de la ecuación de Burgers. Adicionalmente proveer al lector, los códigos en MATLAB de los pasos más importantes para la implementación del método FPM que sirven de pauta para la solución de la ecuación de Burgers. Dada la documentación y lo corto de los programas, estos son fácilmente modificables o adaptables para resolver problemas más complejos. Otro objetivo es contrastar los resultados teóricos sobre el orden de convergencia del método con experimentos numéricos, que hasta nuestro conocimiento, no se presenta en la literatura.

El artículo está organizado de la siguiente manera. Inicialmente en la sección 2 se deduce de manera muy breve la ecuación de Burgers y se presenta su representación Lagrangeana. En la sección 3.1 se describe la manera de aproximar una función y las derivadas espaciales a partir de una distribución arbitraria de puntos usando el método de mínimos cuadrados móviles (MLS, por sus siglas en inglés) y sus propiedades de aproximación. Luego se explica en detalle la generación del sistema de puntos y su manipulación computacional, pilar fundamental del método (sección 3.2). El tratamiento numérico de la ecuación viscosa se presenta en la sección 3.3 y la versión no viscosa usando una adaptación de la técnica upwind en la sección 3.4. Finalmente, los resultados numéricos y conclusiones en las secciones 4 y 5.

2 La ecuación de Burgers y su formulación lagrangeana

La ecuación de Burgers es el ejemplo más sencillo de una ecuación hiperbólica no lineal. Su importancia radica en que su solución desarrolla ondas de choque o discontinuidades que ocurren en la dinámica de fluidos y que se describen por ecuaciones como lo son las de Euler y las de Navier-Stokes. Al ser una simplificación de la ecuación de Navier-Stokes, la ecuación de Burgers se ha convertido en el marco de referencia por excelencia, para probar esquemas numéricos en la solución de problemas de la mecánica de fluidos.

El método de conjuntos finitos de puntos se basa en la formulación lagrangeana del fluido en estudio. Es decir que para rastrear el movimiento del fluido, consideraremos el fluido simplemente como un sistema de partículas (volúmenes de control). Cada punto en el fluido o en el sólido se identifica como una partícula, con una posición x y una velocidad u . Estas partículas viajan junto con el fluido, midiendo las cantidades físicas del fluido que está circulando, como por ejemplo: masa, momento, energía, presión, etc.

Para deducir la ecuación de Burgers, consideremos el movimiento de una corriente de partículas en el espacio, tal que cada partícula en la posición \mathbf{x} y en el instante t tiene una velocidad constante $\mathbf{u}(\mathbf{x}(t), t)$. Si $\mathbf{x}(t) = (x(t), y(t), z(t))$ es la posición de una partícula determinada, se tiene que

$$\mathbf{u}(\mathbf{x}(t), t) = \frac{d\mathbf{x}}{dt}(t) = c, \text{ donde } c \text{ es una constante} \quad (1)$$

y la aceleración de la partícula en el fluido, es la derivada total de la velocidad con respecto al tiempo:

$$\mathbf{a}(t) = \frac{d^2}{dt^2}\mathbf{x}(t) = \frac{\partial \mathbf{u}}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial \mathbf{u}}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial \mathbf{u}}{\partial z} \frac{\partial z}{\partial t} + \frac{\partial \mathbf{u}}{\partial t} = 0. \quad (2)$$

Entonces, escribiendo como

$$\mathbf{u}(x, y, z, t) = (u_1(x, y, z, t), u_2(x, y, z, t), u_3(x, y, z, t)),$$

obtenemos la ecuación de Burgers en tres dimensiones

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \nabla \mathbf{u} = 0. \quad (3)$$

El modelo anterior describe el movimiento de la corriente de partículas y la conservación de momento en el caso ideal.

Denotando la expresión $\frac{\partial}{\partial t} + \mathbf{u} \nabla$ por $\frac{D}{Dt}$, obtenemos la formulación lagrangeana de la ecuación de Burgers

$$\frac{D\mathbf{u}}{Dt} = 0. \quad (4)$$

Esta ecuación nos dice que el fluido tiene una velocidad pero que no cambia desde el punto de vista lagrangeano. El termino $\frac{D\mathbf{u}}{Dt}$ se conoce como *derivada material* y toma en cuenta que el movimiento de un fluido depende del movimiento de las partículas en el tiempo.

Nota 2.1 En general, si θ es una cantidad que depende de la posición \mathbf{x} y el tiempo t , la expresión $\frac{D\theta}{Dt} = \frac{\partial\theta}{\partial t} + \mathbf{u}\nabla\theta$, nos dice que tan rápido esta cambiando θ por las partículas que están pasando en este mismo instante por el punto \mathbf{x} .

Una propiedad fundamental en el estudio de los fluidos es la viscosidad, ésta surge de la resistencia a la deformación del fluido, por ejemplo, si se compara el agua con la glicerina decimos que la glicerina es más viscosa que el agua. En la ecuación de Burgers la viscosidad se puede ver como la variación espacial del gradiente de velocidad. La versión viscosa de la ecuación de Burgers en formulación Lagrangeana esta dada por:

$$\frac{D\mathbf{u}}{Dt} = \varepsilon \Delta u. \quad (5)$$

En términos numéricos la viscosidad es un término de difusión, es decir, disipa el movimiento de las partículas en el fluido. La ecuación de Burgers no viscosa (3) es un caso límite del modelo con viscosidad, pues ésta permite encontrar la solución de la ecuación de Burgers como el límite de una sucesión de soluciones de la ecuación parabólica cuando ε tiende a 0.

Para fines didácticos e ilustrar de la manera más simple la implementación del método, los problemas que resolveremos en este trabajo son: la ecuación viscosa y no viscosa ($\varepsilon = 0$) de Burgers en una dimensión

$$\frac{Du}{Dt} = \varepsilon \frac{\partial^2 u}{\partial x^2} \quad (6a)$$

$$\frac{Dx}{Dt} = u \quad (6b)$$

con condición inicial

$$u(x, 0) = \begin{cases} 1 & \text{Si } -\infty < x \leq 0 \\ 1 - \frac{x}{a} & \text{Si } 0 < x < a \\ 0 & \text{Si } a \leq x < \infty. \end{cases} \quad a \in \mathbb{R}^+ \quad (6c)$$

Observemos que cada partícula se mueve de acuerdo a la ecuación (6b).

3 Descripción del método FPM

Para aplicar el método (FPM) se siguen los siguientes pasos:

1. Generación de un conjunto de puntos (partículas) en el dominio del fluido e identificación de los puntos vecinos a un punto dado.
2. Proporcionar una aproximación del campo de funciones y sus derivadas espaciales en cada punto, por el método de mínimos cuadrados móviles, a partir de la información de los puntos vecinos.
3. Resolver el problema de valor inicial (6) usando alguno de los métodos convencionales para la solución ecuaciones diferenciales ordinarias.

Si se desea resolver numéricamente un ecuación diferencial, la primera cuestión que viene a la mente es: ¿cómo aproximar las derivadas de las funciones que intervienen en la ecuación diferencial?. Pregunta que se torna más interesante en el caso en que no se utilizan mallas. Una respuesta a esta pregunta es precisamente el método de mínimos cuadrados móviles [6], sobre el cual se basa el método libre de malla de conjuntos finitos de puntos.

3.1 Aproximación y diferenciación numérica mediante MLS

El método de mínimos cuadrados móviles, consiste en realizar aproximaciones locales de una función por polinomios empleando el método de mínimos cuadrados ponderados. La naturaleza local del método permite además aproximar de manera óptima las derivadas de la función.

Sean $f : [a, b] \rightarrow \mathbb{R}$ una función k veces continuamente diferenciable en $[a, b]$ y $\Gamma = \{x_1 < x_2, \dots < x_N\} \subset [a, b]$ un subconjunto finito de puntos donde los valores $f(x_i)$ $i = 1, \dots, N$ son conocidos. El objetivo es aproximar los valores de la función f y sus derivadas f' y f'' en un punto $\bar{x} \in [a, b]$.

Con esta finalidad, primero definimos el conjunto de puntos vecinos $V_{\bar{x}}$ de \bar{x} como

$$V_{\bar{x}} = \{x_1, \dots, x_m\} := \{x_i \in D : |x_i - \bar{x}| < h\} \quad (m < N)$$

donde $h = \gamma \max_{i=1, \dots, N-1} |x_{i+1} - x_i|$, $1 < \gamma < 3$. Nótese que γ se escoge de esta manera para garantizar que $V_{\bar{x}}$ tenga al menos tres puntos. Al parámetro h se la llama longitud suavizante.

Segundo, puesto que $f \in C^k[a, b]$, para cualquier $x_j \in V_{\bar{x}}$ podemos escribir

$$f(x_j) = f(\bar{x}) + (x_j - \bar{x})f'(\bar{x}) + \frac{1}{2}(x_j - \bar{x})^2 f''(\bar{x}) + O(|x_j - \bar{x}|^3), \quad j = 1, \dots, m. \quad (7)$$

Este conjunto de ecuaciones forman un sistema lineal sobredeterminado de m ecuaciones y 3 incógnitas $f(\bar{x}), f'(\bar{x}), f''(\bar{x})$, el cual podemos “resolver” minimizando el error residual del sistema, es decir: Encuentre $\tilde{\mathbf{c}} \in \mathbb{R}^3$ tal que

$$\|\mathbf{f} - \mathbf{P}\tilde{\mathbf{c}}\|_{\mathcal{W}_h(\bar{x})}^2 = \min_{\mathbf{c} \in \mathbb{R}^3} \|\mathbf{f} - \mathbf{P}\mathbf{c}\|_{\mathcal{W}_h(\bar{x})}^2, \quad (8)$$

donde

$$\mathbf{f} = [f(x_1), f(x_2), \dots, f(x_m)]^T, \\ \mathbf{P} = \begin{bmatrix} 1 & (x_1 - \bar{x}) & \frac{1}{2}(x_1 - \bar{x})^2 \\ 1 & (x_2 - \bar{x}) & \frac{1}{2}(x_2 - \bar{x})^2 \\ \vdots & \ddots & \vdots \\ 1 & (x_m - \bar{x}) & \frac{1}{2}(x_m - \bar{x})^2 \end{bmatrix}$$

y $\|\cdot\|_{\mathcal{W}_h(\bar{x})}$ es la norma ponderada definida por la matriz diagonal

$$\mathbf{W} = \text{diag}(\mathcal{W}_h(x_1, \bar{x}), \mathcal{W}_h(x_2, \bar{x}), \dots, \mathcal{W}_h(x_m, \bar{x}))$$

con función peso

$$\mathcal{W}_h(x_i, \bar{x}) = \begin{cases} e^{-\alpha \frac{|x_i - \bar{x}|^2}{h^2}} & \text{si } |x_i - \bar{x}| < h \\ 0 & \text{en otro caso} \end{cases} \quad (9)$$

y α es una constante positiva, $\alpha \in [2, 8]$.

Es bien conocido que el problema de minimización (8) es equivalente a resolver las ecuaciones normales

$$\mathbf{P}^T \mathbf{W} \mathbf{P} \mathbf{c} = \mathbf{P}^T \mathbf{W} \mathbf{f} \quad (10)$$

y su solución es

$$\mathbf{c} = (\mathbf{P}^T \mathbf{W} \mathbf{P})^{-1} (\mathbf{P}^T \mathbf{W} \mathbf{f}). \quad (11)$$

Teorema 3.1 *La matriz $\mathbf{P}^T \mathbf{W} \mathbf{P}$ es invertible.*

Prueba: La no singularidad de la matriz $\mathbf{P}^T \mathbf{W} \mathbf{P}$ proviene del hecho que, ésta es una matriz simétrica definida positiva. En efecto, como \mathbf{W} es una matriz diagonal, $(\mathbf{P}^T \mathbf{W} \mathbf{P})^T = \mathbf{P}^T \mathbf{W} \mathbf{P}$. Además, para todo vector \mathbf{c} no nulo, $\mathbf{P} \mathbf{c} \neq \mathbf{0}$ luego

$$0 < \|\mathbf{P} \mathbf{c}\|_{\mathcal{W}_h(\bar{x})}^2 = (\mathbf{P} \mathbf{c})^T \mathbf{W} \mathbf{P} \mathbf{c} = \mathbf{c}^T \mathbf{P}^T \mathbf{W} \mathbf{P} \mathbf{c}, \quad \forall \mathbf{c} \neq \mathbf{0}.$$

Por otra lado, si $\mathbf{c} = \mathbf{0}$, $\mathbf{P} \mathbf{c} = \mathbf{0}$ es decir

$$c_1 + c_2(x_j - \bar{x}) + c_3 \frac{1}{2}(x_j - \bar{x})^2 = 0. \quad \text{para } j = 1, \dots, m. \quad (12)$$

Como los puntos x_j son distintos, de la ecuación (12) deducimos que x_1, x_2, \dots, x_m ($m > 2$) son raíces del polinomio cuadrático $c_1 + c_2(x - \bar{x}) + c_3 \frac{1}{2}(x - \bar{x})^2$, lo cual es absurdo. Por lo tanto la matriz $\mathbf{P}^T \mathbf{W} \mathbf{P}$ es definida positiva. ■

De este modo, el método MLS produce directamente de los valores $f(x_1), \dots, f(x_m)$ cercanos a \bar{x} las aproximaciones $\Pi f(\bar{x}) := c_1$, $\Pi f'(\bar{x}) := c_2$, $\Pi f''(\bar{x}) := c_3$ de $f(\bar{x})$, $f'(\bar{x})$ y $f''(\bar{x})$ respectivamente. El polinomio local de aproximación de f es entonces

$$p(x) = c_1 + c_2(x - \bar{x}) + c_3 \frac{1}{2}(x - \bar{x})^2.$$

Acercas del orden de aproximación que tiene el método de mínimos cuadrados móviles, en [1] se prueba bajo condiciones generales de la función de peso y la distribución de puntos, que los órdenes de aproximación de la función y sus derivadas son óptimos. Para nuestro caso el resultado de [1] se resume en el siguiente teorema.

Teorema 3.2 *Sean $f \in C^{k+1}([a, b])$ y $\bar{x} \in [a, b]$. Si $\Pi f(x)$ es la aproximación por el método MLS de f en el intervalo $I_{\bar{x}} = [\bar{x} - h, \bar{x} + h]$, entonces existen constantes K_1, K_2 y K_3 independientes de h , tal que*

$$|f(\bar{x}) - \Pi f(\bar{x})| \leq K_1 \max_{\xi \in I_{\bar{x}}} |f^{k+1}(\xi)| h^{k+1} \quad (13)$$

$$|f'(\bar{x}) - \Pi f'(\bar{x})| \leq K_2 \max_{\xi \in I_{\bar{x}}} |f^{k+1}(\xi)| h^k \quad (14)$$

$$|f''(\bar{x}) - \Pi f''(\bar{x})| \leq K_3 \max_{\xi \in I_{\bar{x}}} |f^{k+1}(\xi)| h^{k-1}. \quad (15)$$

Programa 1: Método MLS.

```

1 function [d2fxo, dfxo, fxo] = mls(xo,xv,fv,h)
2 % Funcion que aproxima el valor de f(xo), f'(xo), f''(xo) por
   medio de MLS
3 % Datos de entrada:
4 % xv := vector de puntos vecinos a x0
5 % fv := f(xv) valores de la funcion en los puntos xv
6 % h := longitud suavizante del nucleo
7 % Datos de salida:
8 % fxo, dfxo d2fxo = aproximaciones de f(xo), f'(xo), f''(xo).
9 m=length(xv);
10 alpha=8;
11 k=0;
12 for j=1:m
13     d= xv(j)-xo;
14     if abs(d) <= h
15         k=k+1
16         dx(k) = d;
17         w(k)=exp(-1.0*alpha * (dx(k)/h)^2 );
18         y(k)=fv(j);
19     end
20 end
21 [dmin,i] = min(abs(dx))
22 fmin=y(i);
23 b=(y -fmin)';
24 W=diag(w);
25 M=[ones(k,1), dx', 0.5*dx'.^2];
26 c=(M'*W*M)\(M'*W*b);
27 fxo=fmin+c(1);
28 dfxo = c(2);
29 d2fxo = c(3);
30 end

```

3.2 Generación de puntos y búsqueda de los vecinos más cercanos

Antes de hacer la aproximación de f , f' y f'' en cada punto en el dominio del fluido, es necesario generar una distribución de puntos en el dominio. Esta debe ser lo suficientemente densa para garantizar que h sea suficientemente pequeño y mantener un buen orden de aproximación, pero que el número de puntos no sea tan grande que implique resolver un número grande de sistemas normales de gran tamaño. De esta manera el costo computacional del método FPM será eficiente.

En este sentido existen diferentes tipos de algoritmos para escoger los vecinos, algunos de estos se pueden encontrar en [9] y [11]. La implementación en MATLAB de la búsqueda de los vecinos más cercanos que usamos en este artículo sigue las ideas presentadas en [9].

Para empezar, los elementos requeridos para la aplicación del algoritmo son:

- El punto \bar{x} al que se le desea encontrar los vecinos.
- Un vector \mathbf{p} de posiciones de los puntos en los que se conoce el valor de la función de la que se va a hacer la aproximación en \bar{x} .

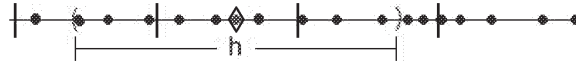


Figura 1: Bosquejo de los vecinos más cercanos. Caso unidimensional.

En el algoritmo el dominio se divide en celdas, en cada celda generamos al menos 3 puntos de manera aleatoria, de estas celdas se toman los vecinos del punto \bar{x} . El ancho de cada celda es $dx = \frac{x_{max}-x_{min}}{m}$ donde el numerador es la longitud del dominio y m es el número de celdas en las que se quiere dividir el dominio. Por lo tanto el número total de puntos en el dominio es al menos $3m$. La longitud suavizante que se escoge es: $h = 1.5dx$, esto se hace para que el sistema lineal (10) a resolver sea sobredeterminado o por lo menos cuadrado.

Si el punto \bar{x} está en la j -ésima celda, los vecinos son los puntos que se encuentran en las celdas $j - 1$, j y $j + 1$ eso quiere decir que el número mínimo de celdas en las que se debe dividir el dominio es 3.

Al tomar como ejemplo

$$\mathbf{p} = [6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17],$$

$\bar{x} = 10.5$ y $m = 4$, entonces $dx = 2.75$, $h = 4.125$ y las celdas son:

$$\begin{aligned} C_1 &= [6 \ 7 \ 8] & C_2 &= [9 \ 10 \ 11] \\ C_3 &= [12 \ 13 \ 14] & C_4 &= [15 \ 16 \ 17]. \end{aligned}$$

Después que se divide el dominio en celdas, se procede a la construcción de el vector **next** que es el vector con los índices de cada elemento de \mathbf{p} pero con ceros en cada inicio de celda y del vector **head** que es el vector con los índices de los últimos elementos de cada celda en el vector \mathbf{p} , éste se obtiene tomando los índices más grandes en los que **next** es no nulo. Continuando con el ejemplo se tiene:

$$\begin{aligned} \mathbf{next} &= [0 \ 1 \ 2 \ 0 \ 4 \ 5 \ 0 \ 7 \ 8 \ 9 \ 10 \ 11] \text{ y} \\ \mathbf{head} &= [3 \ 6 \ 9 \ 12]. \end{aligned}$$

Programa 2: Generación aleatoria del conjunto de puntos.

```

1 function [x, head, next, dx]=celdas(m, nin, xmin, xmax)
2 % Funcion que genera la distribucion de puntos en el dominio de
   computo
3 % Datos de entrada:
4 % Intervalo [xmin, xmax] i
5 % m : numero de celdas
6 % nin : numero de particulas en cada celda
7 % Datos de salida:
8 % x : vector de posiciones de los puntos
9 % head : vector de indices del extremo derecho de cada celda
10 % next : vector de siguientes
11 dx = (xmax-xmin)/m;
12 % Definicion del centro de la celda xmitt
13 for j=1:m
14     xmitt(j) = xmin + (j-0.5)*dx;
15 end
16 % llenado de cada celda por puntos aleatorios
17 n = 0;
18 for j=1:m
19     for i=1:nin
20         n=n+1;
21         xleft = xmitt(j) - 0.5*dx;
22         x(n) = xleft + rand()*dx;
23     end
24 end
25 x=sort(x);
26 head=zeros(m,1); % determinando las cabeceras de cada celda
27 for i=1:n
28     j = ceil((x(i)-xmin)/dx);
29     next(i) = head(j);
30     head(j) = i;
31 end

```

Ahora se ubica la celda en la que se encuentra \bar{x} , la celda es $C_j = \lceil \frac{\bar{x}-x_{min}}{m} \rceil$, de nuevo en el ejemplo $\bar{x} \in C_2$. Cuando ya se ha ubicado \bar{x} en la celda lo que resta es indexar los vecinos enumerando las posiciones de las partículas en las tres celdas de las que se van a sacar los vecinos. Para esto se usa el vector **head** pues se enumeran las partículas de cada celda hasta llegar a la cabeza de la celda respectiva. En el ejemplo los índices de los vecinos son: $[1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$.

Tomando los respectivos elementos indexados de **p** se obtienen los vecinos de \bar{x} , que en el ejemplo son: $\mathbf{x}_v = [6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14]$.

De esta manera se obtienen los vecinos más cercanos de la partícula \bar{x} . Este procedimiento se repite para cada partícula en la que se necesite hacer la aproximación por mínimos cuadrados móviles.

Programa 3: Método de búsqueda de los puntos vecinos de un punto dado.

```

1 function [xv,nb]=vecindad(p_x,xmin,m,dx,x,head,next)
2 % Programa que determina los puntos vecinos de un punto
   arbitrario p_x
3 % Datos de entrada:
4 % p_x : punto arbitrario
5 % xmin : extremo izquierdo del intervalo [xmin, xmax]
6 % dx : ancho de la celda
7 % head : vector de indices de la particula cabecera de cada
   celda
8 % next : vector de indices de cada elemento del vector de
   vecinos p
9 % Datos de salida:
10 % xv : vector de puntos vecinos de p_x
11 % nb : número de puntos vecinos
12 j = ceil((p_x - xmin)/dx);
13 nb = 0;
14 if j==1
15     for icell=j:j+2
16         k = head(icell);
17         while ( k ~= 0 )
18             nb = nb + 1;
19             list(nb) = k;
20             k = next(k);
21         end
22     end
23     elseif (1<j) & (j<m)
24         for icell=j-1:j+1
25             k = head(icell);
26             while ( k ~= 0 )
27                 nb = nb + 1;
28                 list(nb) = k;
29                 k = next(k);
30             end
31         end
32     else
33         for icell=j-2:j
34             k = head(icell);
35             while ( k ~= 0 )
36                 nb = nb + 1;
37                 list(nb) = k;
38                 k = next(k);
39             end
40         end
41     end
42     for k=1:nb
43         xv(k)=x( list(k));
44     end

```

Nota 3.1 Si la partícula \bar{x} está en alguna de las fronteras, es decir, en la primera o en la última celda, los vecinos se toman de las tres primeras celdas en el caso de estar en la primera celda o se toman de las tres últimas en el caso de estar en la última celda. La influencia de los vecinos permite tener una buena aproximación de la función en los puntos de frontera.

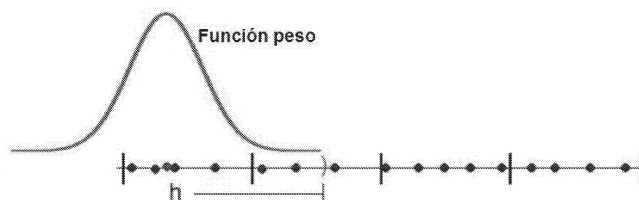


Figura 2: Influencia de los vecinos más cercanos en los puntos de frontera.

3.3 Solución de la ecuación viscosa de Burgers

Después de generar el conjunto finito de puntos $\Gamma = \{x_1 < x_2 < \dots < x_N\} \subset \mathbb{R}$, seleccionamos un tamaño de paso de tiempo Δt y definimos los tiempos discretos como $t^k = k\Delta t$ para $k = 0, 1, 2, \dots$. La aproximación de las derivadas espaciales en cada instante t^k se obtiene empleando el método (MLS) sobre el conjunto Γ (ver sección 3.1). Estas aproximaciones las denotamos por

$$\Pi \frac{\partial u_i^k}{\partial x} \approx \frac{\partial u}{\partial x}(x_i, t^k) \quad \text{y} \quad \Pi \frac{\partial^2 u_i^k}{\partial x^2} \approx \frac{\partial^2 u}{\partial x^2}(x_i, t^k).$$

Por simplicidad en la presentación de la implementación, para la integración numérica del sistema de ecuaciones diferenciales ordinarias (6) usaremos el método de Euler progresivo con paso de integración Δt

$$u_i^{k+1} = u_i^k + \Delta t \varepsilon \Pi \frac{\partial^2 u_i^k}{\partial x^2} \tag{16}$$

$$x_i^{k+1} = x_i^k + \Delta t u_i^k \tag{17}$$

para $i = 1, 2, \dots, N$ y $k = 0, 1, \dots$, donde la condición inicial es dada por u_i^0 para todo $i = 1, \dots, N$. Cabe notar que es posible utilizar otros esquemas de solución como por ejemplo métodos de Runge-Kutta.

3.4 Solución de la ecuación no viscosa de Burgers

Para resolver la ecuación de Burgers no viscosa $\varepsilon = 0$, podemos escribir la ecuación de Burgers (3) en una dimensión de la siguiente manera:

$$\frac{\partial u}{\partial t} + \frac{1}{2}u \frac{\partial u}{\partial x} = -\frac{1}{2}u \frac{\partial u}{\partial x}.$$

Luego el correspondiente sistema de EDO a solucionar es

$$\frac{Du}{dt} = -\frac{1}{2}u \frac{\partial u}{\partial x} \quad (18)$$

$$\frac{Dx}{dt} = \frac{1}{2}u. \quad (19)$$

Antes de aproximar las derivada $\frac{\partial u}{\partial x}$, debemos observar que dada la simetría de la función peso (9), las aproximaciones de la derivadas por el método MLS, provienen implícitamente de esquemas de diferencias centrales. Ésto provoca inestabilidades en la solución numérica al momento de tratar problemas con términos convectivos como la ecuación de Burgers. Para superar este inconveniente empleamos la técnica de corrientes arriba (upwind). Por lo tanto para aproximar la primera derivada tomamos los vecinos más cercanos de cada partícula a izquierda o a derecha, dependiendo de la pendiente de las características: si la pendiente de las características es positiva la aproximación se hace con los vecinos que están atrás de la partícula y si la pendiente de las características es negativa se usan los vecinos que están adelante.

Para escoger los vecinos a izquierda o a derecha existen dos opciones: la primera consiste en seleccionar en el algoritmo de búsqueda de vecinos sólo los vecinos a derecha o a izquierda. La segunda opción es seleccionar todos los vecinos, pero en el núcleo W_h el peso de los vecinos a los que no se quiere tener en cuenta se les asigna el valor de cero. En este trabajo escogemos la segunda opción.

La discretización de la ecuación de Burgers no viscosa usando la técnica de upwind es la siguiente:

$$x_i^{k+1} = x_i^k + \frac{dt}{2} \Pi u_i^k \quad (20)$$

$$u_i^{k+1} = u_i^k - \frac{dt}{2} ((\max(u_i^k, 0) \Pi^- \frac{\partial u_i^k}{\partial x}) - (\min(u_i^k, 0) \Pi^+ \frac{\partial u_i^k}{\partial x})) \quad (21)$$

para $i = 1, 2, \dots, N$ y $k = 0, 1, 2, \dots$, donde las condiciones iniciales son dadas por u_i^0 para todo $i = 1, \dots, N$. $\Pi^- \frac{\partial u_i^k}{\partial x}$ denota la aproximación de la derivada en el punto x_i con los vecinos a izquierda y $\Pi^+ \frac{\partial u_i^k}{\partial x}$ es la aproximación de la derivada en el punto x_i con los vecinos a derecha.

Nota 3.2 *De los dos esquemas anteriores observamos que los puntos x_i se mueven con la velocidad del fluido. Como esta velocidad no la conocemos con anticipación, puede ocurrir que los puntos se acumulen o sean pocos en algunas partes del dominio ocasionando inestabilidades numéricas y aproximaciones incorrectas de soluciones discontinuas. Para evitar estas diferencias de densidad de puntos en el dominio, el primer conjunto Γ_0 de puntos (no necesariamente equidistante) lo mantenemos en memoria. Al aplicar un paso de tiempo del método de integración, los puntos de Γ_0 se mueven con su correspondientes velocidades y los nuevos valores de u son calculados en las nuevas posiciones. Entonces para mantener control sobre los puntos hacemos lo siguiente: la función u se interpola en el conjunto original de puntos Γ_0 usando el método (MLS) y las nuevas posiciones de los puntos se reemplazan por las posiciones anteriores Γ_0 . Este proceso se realiza en cada paso de tiempo del método de integración. Una manera alternativa para evitar la aglomeración o deficiencia de puntos dentro del dominio es: Si dos puntos están muy cerca, interpolamos de manera lineal el valor de la función en el punto medio entre estos y luego los dos puntos se reemplazan en Γ por dicho punto medio. En el caso que los puntos estén muy alejados, agregamos el punto medio entre éstos al conjunto Γ y el valor de la función en este punto se calcula nuevamente mediante interpolación lineal.*

4 Resultados numéricos

Para determinar la eficiencia del método de conjuntos finitos de puntos, resolveremos la ecuación viscosa de Burgers (6) con viscosidad $\varepsilon = 0.005$ y el caso no viscoso ($\varepsilon = 0$) mediante la técnica de upwind. Para ambos casos consideramos 300 puntos equidistantes y tiempos finales $t = 0, 0.25, 0.75$ y 1.0 . Dado que empleamos el método de Euler progresivo, escogemos basados en la condición CFL [4], el paso discretización en el tiempo $\Delta t \leq \frac{1}{2} \frac{\Delta x}{\max |u(x)|}$. Además presentamos un análisis del comportamiento del error de aproximación de la solución u , a medida que se aumenta el número de puntos en el dominio.

4.1 Caso 1: Ecuación viscosa de Burgers

En la Figura 3 observamos el fenómeno de distorsión de ondas no lineales cuando estas se propagan. En el instante $t = 0$, la condición inicial (6c) es tal, que el valor de u en puntos a la derecha de $x = 1$ es menor que en aquellos que están a su izquierda. Esto provoca que las partículas a la izquierda de $x = 1$ viajen más rápido que las que están a la derecha. En los instantes $t = 0.25$ y $t = 0.75$ el perfil de la onda es más pendiente hasta el punto que llega a ser vertical en $t = 1$,

formándose una onda de choque o discontinuidad. En cuanto al método FPM vemos que la solución numérica reproduce de manera suficientemente precisa la formación de la onda de choque.

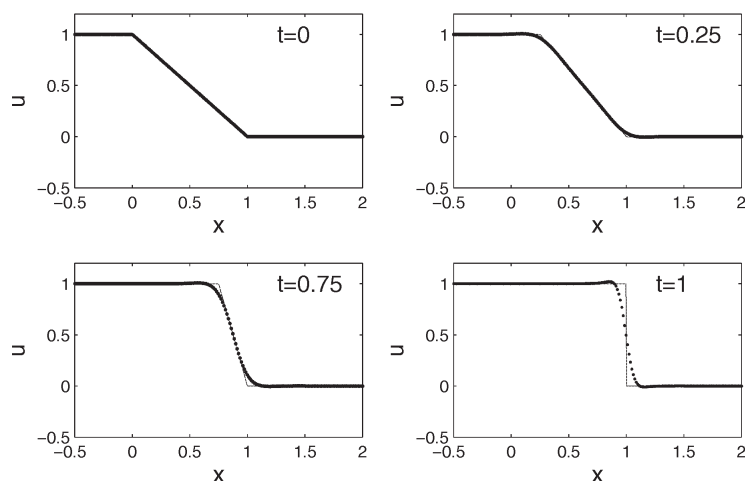


Figura 3: Solución de la ecuación no viscosa de Burgers ($\varepsilon = 0.005$) con valor inicial (6c) por el método FPM. Solución numérica curva punteada. Solución exacta curva sólida.

Es de notar que la introducción del término de viscosidad $\varepsilon \frac{\partial^2 u}{\partial x^2}$ tiene un efecto suavizador sobre la solución y juega un papel importante en la estabilización de la solución numérica en regiones donde se presenta un gradiente muy brusco (ver Figura 4). Lo anterior se aprecia en la pequeña oscilación al momento de reproducir la discontinuidad de la solución. Además en la Figura 4 apreciamos el efecto de hacer $\varepsilon \rightarrow 0$, donde el método intenta reproducir la onda de choque.

4.2 Caso 2: Ecuación no viscosa de Burgers

La Figura 5 muestra la solución de la ecuación no viscosa de Burgers usando la técnica de upwind. Observamos que el esquema localiza el choque y presenta un comportamiento conveniente, similar a la solución de la ecuación viscosa, pero obteniendo perfiles más bruscos de la onda de choque. Observemos también que debido a la simetría de la función de peso, la interpolación de la solución sobre la distribución fija de puntos Γ_0 , ocasiona las pequeñas oscilaciones que se observan antes y después del choque (Figura 5, instante $t = 1$).

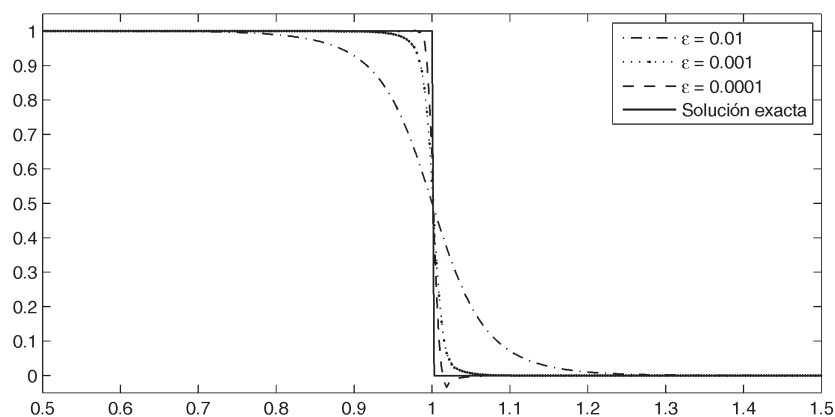


Figura 4: Solución de la ecuación viscosa de Burgers usando 750 puntos para diferentes viscosidades $\varepsilon = 0.01, 0.001, 0.0001$ en $t = 1$. Solución exacta curva sólida.

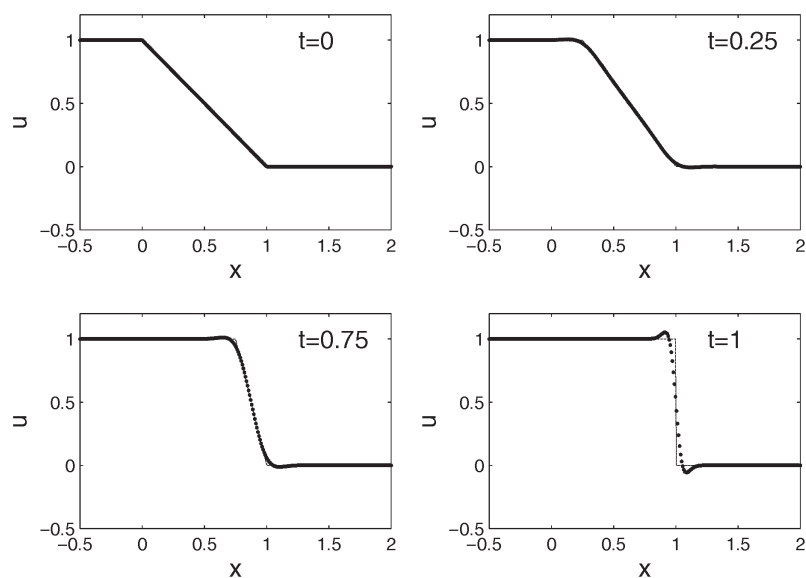


Figura 5: Solución de la ecuación de Burgers no viscosa con valor inicial (6c) mediante el método FPM usando upwind. Solución numérica curva punteada. Solución exacta curva sólida.

4.3 Orden de convergencia

Con el fin de estudiar las propiedades de convergencia del método FPM con respecto al número de puntos N en el dominio, calculamos los errores de aproximación entre la solución exacta del problema (6)

$$u(x, t) = \begin{cases} 1 & \text{si } x \leq t \\ \frac{a-x}{a-t} & \text{si } t < x < a \\ 0 & \text{si } x \geq a \end{cases} \quad a \in \mathbb{R}^+ \quad (22)$$

y la solución numérica u_i^k en la norma $L_1(\mathbb{R})$ restringida sobre el conjunto de puntos $\Gamma := \{x_1, \dots, x_N\}$. Por simplicidad consideraremos que los puntos de Γ son equidistantes. Podemos definir el error de aproximación en el instante fijo t_k como

$$e(h) := h_{max} \sum_{i=1}^N |u(x_i, t_k) - u_i^k|$$

donde $h_{max} = \max_{i=1, \dots, N} |x_i - x_{i-1}|$. Puesto que esperamos que el método tenga un orden de aproximación p , entonces $e(h) = Ch_{max}^p$ cuando $h \rightarrow 0$. Para encontrar el orden de aproximación p , consideremos las aproximaciones en dos conjuntos de puntos Γ_1 y Γ_2 con distancia máxima entre sus puntos h_1 y h_2 respectivamente, entonces p puede estimarse como

$$p \approx \frac{\log(e(h_1)/e(h_2))}{\log(h_1/h_2)}.$$

La Tabla 1 muestra para varios números de puntos en el dominio $N = 60, 120, 240, 480, 960$, y 1920 , el error de aproximación por u_i^k y los órdenes de convergencia cuando $t_k = 0.8$. Se observa de la tabla que al considerar viscosidad, el método es de primer orden de convergencia. Esto es consecuencia de la aproximación de la segunda derivada de la función por el método MLS (Ver teorema 3.2). Por otra parte, al usar la técnica upwind tenemos un orden de convergencia mayor al del método donde se introduce el término de viscosidad (Ver tabla 1). Claramente esta mejora en la aproximación se debe que para este esquema solo necesitamos aproximar la función u y su primera derivada (Ver teorema 3.2).

Ahora bien, si se desea mejorar el orden de aproximación podemos truncar el polinomio de Taylor después de los términos de tercer orden o de mayor orden. Así obtenemos un orden más o mayor en la aproximación. Sin embargo, esto tiene el inconveniente que el sistema lineal es muy mal condicionado y no obtendremos buenos resultados.

N	h	Viscosidad		Upwind	
		Error	Orden	Error	Orden
60	0.0424	0.0227	-	0.1784	-
120	0.0220	0.0095	1.2366	0.0652	1.4344
240	0.0105	0.0041	1.1969	0.0124	2.3697
480	0.0052	0.0019	1.1439	0.0041	1.6077
960	0.0026	0.0009	1.1337	0.0011	1.8714
1920	0.0013	0.0004	1.1003	0.0004	1.3710

Tabla 1: Errores de aproximación en la norma $L_1(\mathbb{R})$ y órdenes de aproximación del método FPM.

5 Conclusiones

El método libre de malla de conjuntos finitos de puntos es la combinación de varias técnicas numéricas y algorítmicas, aparentemente cada una sin relación, pero que al enlazarlas de manera acertada se aprecia la aplicación de los métodos numéricos en sí mismos. El método FPM puede catalogarse como un método semidiscreto puesto que dada una ecuación diferencial parcial, las derivadas espaciales son discretizadas para obtener un sistema de ecuaciones diferenciales ordinarias (EDO), el cual puede ser resuelto por algún método convecional para EDO. En este trabajo empleamos un esquema numérico explícito a pesar que éste es condicionalmente estable y nos obliga a tomar pequeños pasos de discretización en el tiempo. Sin embargo la ventaja está en no tener que resolver grandes problemas no lineales que aparecen si usamos esquemas implícitos. Los cálculos numéricos muestran que el método FPM es eficiente en la reproducción de ondas de choque y tiene tasas de convergencia óptimas en la norma $L_1(\mathbb{R})$.

Un pilar importante del método FPM y que le da la esencia de ser un método libre de malla, es resolver el problema de identificar los puntos vecinos a uno dado. Es a partir de estos vecinos que se realizan las aproximaciones de Taylor para construir los sistemas lineales correspondientes a cada punto. Este proceso se repite en cada punto del dominio computacional y en cada paso de tiempo. Si no se usa un algoritmo eficiente en la búsqueda de vecinos, el método FPM será ineficiente desde el punto de vista de costo computacional.

Se espera que con esta introducción al método FPM, se tenga la posibilidad de aprender más acerca del FPM y aplicarlo a problemas más complejos de la mecánica de fluidos.

Referencias

- [1] Armentano, M.G.; DurÃ¡n, R.G. (2001) “Error estimates for moving least square approximations”, *Applied Numerical Mathematics* **37**(3) : 397–416.
- [2] Dilts, G.A. (1999) “Moving least squares particle hydrodynamics I, consistency and stability”, *Int. J. Numer. Meth. Engng* **44**(8): 1115–1155.
- [3] Jensen, P.S. (1972) “Finite difference techniques for variable grids”, *Computers & Structures* **2**(1-2): 17–29.
- [4] Kröner, D. (1997) *Numerical Schemes for Conservation Laws*. Wiley Teubner. New-York.
- [5] Kuhnert, J. (1999) *General Smoothed Particle Hydrodynamics*. Ph.D. Thesis, Kaiserslautern University, Kaiserslautern, Germany.
- [6] Lancaster, P.; Salkauskas, K. (1981) “Surfaces generated by moving least square methods”, *Math of Comp* **37**(155): 141–158.
- [7] Mirzaei, D.; Schaback, R.; Dehghan, M. (2012) “On generalized moving least squares and diffuse derivatives”, *IMA J Numer Anal* **32**: 983–1000.
- [8] Moeller, A. (2014) “Influence of the counter blow air flow during container glass blow and blow process”, in: *Proceedings 12th ESG (European Glass Society)*, 21–24 September 2014, Parma. Italy.
- [9] Morris, J.P. (1995) “An overview of the method of smoothed particle hydrodynamics”, Preprint, Arbeitsgruppe Technomathematik, Technische Universität Kaiserslautern, Germany.
- [10] Oñate, E.; Idelsohn, S.; Zienkiewicz, O.C.; Taylor, R.L. (1996) “A finite point method in computational mechanics. Applications to convective transport and fluid flow”, *International Journal for Numerical Methods in Engineering* **39**(22): 3839–3866.
- [11] Paiva, A.; Petronetto, F.; Tavares, G.; Lewiner T. (2009) “Simulação de Fluidos sem Malha: Uma introdução ao método SPH”, in: *27 Colóquio Brasileiro de Matemática*, IMPA. Rio de Janeiro.
- [12] Prätzel-Wolters, D. (2014) “Jahresberichte Fraunhofer Institut für Technologie und Wirtschaftsmathematik ITWM”, Kerner Druck GmbH, Kaiserslautern.
- [13] Tiwari, S.; Manservigi, S. (2002) “Modelling incompressible Navier-Stokes Flows by LSQ-SPH”, *The Nepali Math Sc. Report* **20**(2): 1–23.