Computers & Fluids 89 (2014) 53–65

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Comparative accuracy and performance assessment of the finite point method in compressible flow problems

Enrique Ortega [a,*], Eugenio Oñate [a], Sergio Idelsohn [a,b], Roberto Flores [a]

[a] International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnica de Cataluña, Edificio C1, Campus Norte, UPC, Gran Capitán, s/n, 08034 Barcelona, Spain
[b] Institucio Catalana de Recerca i Estudis Avancats (ICREA), Cataluña, Spain

ARTICLE INFO

Article history:
Received 25 April 2013
Received in revised form 12 September 2013
Accepted 21 October 2013
Available online 30 October 2013

Keywords:
Meshless
Finite point method
Computational cost
Accuracy
Compressible flow
FEM comparison

ABSTRACT

A comparative assessment of the Finite Point Method (FPM) is presented. Using a wing-fuselage configuration under transonic inviscid flow conditions as reference test case, the performance of the FPM flow solver is compared with an equivalent edge-based Finite Element (FEM) implementation. Efficiency issues have discouraged practical application of meshless methods in the past. Thus, a simplification of the basic FPM technique is proposed in order to reduce the performance gap with respect to classical grid-based algorithms. A comparative evaluation of the accuracy, computational cost and parallel performance of the meshless implementation is carried out with the objective to assess the level of maturity of the technique and identify improvements still required to tackle practical applications. The results obtained show accuracy and performance of the core algorithm comparable to a conventional FEM implementation, thus removing a major obstacle for further developments of the FPM.

1. Introduction

Numerous meshless techniques have been developed to solve flow problems in the last two decades (see [1] for a comparative analysis of some popular approaches). From the first applications in the early 1990s to the latest, meshless techniques have been successfully tested in a growing array of problems, and many potential advantages over conventional discretization approaches have been revealed. However, while most of the efforts focused on new and improved developments, little progress has been done in comparative studies with conventional methods to evaluate accuracy, numerical implementation issues, computational efficiency, robustness and other aspects of practical interest. This is not a minor issue because very often the complexity, low efficiency and lack of robustness found in meshless implementations negate the specific advantages of the approach (e.g. simplified model preparation and discretization, easy implementation of adaptivity and domain deformation, etc.). Therefore, a satisfactory performance of the basic technique is also required so that meshless advantages can be exploited efficiently.

With this in mind, and with the purpose to gain some insight into the real capabilities of meshless approach, a technique known as Finite Point Method (FPM) [2–4] is compared in this work with an equivalent classical Finite Element Method (FEM). The comparative assessment is based on the inviscid solution of a three-dimensional compressible aerodynamics problem and focuses, primarily, on determining if the FPM technique could be competitive in terms of accuracy and computational cost. In order to carry out the comparative study, an unstructured edge-based FEM solver named PUMI [5] is employed as a reference. This choice is motivated by the fact that PUMI and the adopted FPM solver follow very similar solution strategies (apart from, of course, the intrinsic differences due to the spatial discretization methods). This allows for a more direct comparison of the core algorithm properties eliminating effects arising from specific implementation choices.

This paper is organized as follows. The flow solution methodologies adopted in the FPM and PUMI solvers are discussed and compared in Section 2 and a simplification of the meshless technique, aimed to improve its efficiency, is presented in Section 3. Then, a comparative assessment is carried out in Section 4 to evaluate the solution accuracy, computational cost and parallel performance of the FPM in relation to the FEM solver. Finally, some considerations about point discretization in FPM are given in Section 5 and the most relevant conclusions of this work are outlined in Section 6.

2. Flow solution approaches in FPM and FEM

The strategies adopted in FPM and PUMI to solve the flow equations follow similar lines. However, the numerical

* Corresponding author. Tel.: +34 93 401 0808.
E-mail address: eortega@cimne.upc.edu (E. Ortega).

0045-7930/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compfluid.2013.10.024

implementations differ as a consequence of specific features of the spatial discretizations. These differences bring about advantages and disadvantages of one method with respect to the other, concerning both the computational requirements and the properties of the discrete schemes thus derived. Hence, both solution approaches are analyzed and compared in this section to identify meshless implementation issues affecting the FPM competitiveness. The flow equations to be solved are recalled first.

2.1. Governing equations

This work addresses inviscid aerodynamic problems governed by the Euler equations. These equations can be written as

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}^k}{\partial x_k}=\mathbf{0} \tag{1}$$

where $\mathbf{U}=[\rho,\rho u_i,\rho e_t]^T$ is the conservative variables vector, $\mathbf{F}^k=[\rho u_k,\rho u_i u_k+\delta_{ik}p,(\rho e_t+p)u_k]^T$ is the k-component of the advective flux vector and ρ, p, e_t and u_i denote the fluid density, pressure, total energy and velocity components, respectively. The following state relation for a perfect gas is adopted

$$p=\rho(\gamma-1)[e_t-1/2u_k u_k] \tag{2}$$

where γ is the ratio of specific heats. The initial and boundary conditions needed to solve Eq. (1) in an analysis domain Ω will be described later in Section 4.1.

2.2. The meshless FPM solver

Suppose $\varphi(\boldsymbol{x})$ is an unknown function defined in an analysis domain Ω which is discretized by a set of n points $\boldsymbol{x}_i$. Let Ω be covered by subsets of np points Ω_i (clouds of points) consisting of a point $\boldsymbol{x}_i$ called *star point* and a collection of surrounding points $\boldsymbol{x}_j$. Then, an approximation of $\varphi(\boldsymbol{x})$ in Ω_i can be stated by

$$\varphi(\boldsymbol{x})\approx\hat{\varphi}(\boldsymbol{x})=\boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{\alpha} \tag{3}$$

where $\mathbf{p}$ is a vector containing monomial basis functions and $\boldsymbol{\alpha}$ is a vector of unknown coefficients. In this work complete quadratic polynomial bases are employed (e.g. $\mathbf{p}^T(\mathbf{x})=[1,x,y,z,xy,xz,yz,x^2,y^2,z^2]$ in 3D). Assuming $np>m$ in Ω_i, the vector $\boldsymbol{\alpha}$ can be determined by minimizing the following Weighted Least-Squares (WLSQ) functional

$$J_i=\sum_{j=1}^{np}w_i(\boldsymbol{x}_j)[\hat{\varphi}_j-\varphi_j]^2=\sum_{j=1}^{np}w_i(\boldsymbol{x}_j)[\boldsymbol{p}^T(\boldsymbol{x}_j)\boldsymbol{\alpha}-\varphi_j]^2 \tag{4}$$

in which $\varphi_j=\varphi(\boldsymbol{x}_j)$ are the values of the unknown function at each point in Ω_i (nodal parameters), $\hat{\varphi}_j$ indicates approximated values at the same points and $w_i(\boldsymbol{x}_j)=w(\boldsymbol{x}_j-\boldsymbol{x}_i)$ is a compact support weighting function centered on the star point of the cloud (Fixed Least-Squares (FLS) [2]). Next, the approximation of the unknown function φ and its derivatives at the star point $\boldsymbol{x}_i$ can be expressed by

$$\hat{\varphi}_i=a_{ij}\varphi_j \quad \text{and} \quad \frac{\partial\hat{\varphi}_i}{\partial x_k}=b_{ij}^k\varphi_j \quad \forall j\in\Omega_i \tag{5}$$

where a_{ij} and b_{ij}^k are approximation (metric) coefficients obtained from the minimization of the functional (4) (see derivation details in [6]).

The semi-discrete problem in FPM results from replacing the continuous spatial variables in Eq. (1) by their discrete counterparts of Eq. (5). After some manipulation this yields [7]

$$\frac{d\widehat{\mathbf{U}}_i}{dt}=-2\sum_{j\neq i}b_{ij}^k\left[\widetilde{\mathbf{F}}_{ij}^k-\widetilde{\mathbf{F}}_i^k\right] \quad \forall j\in\Omega_i \tag{6}$$

where $\widehat{\mathbf{U}}_i=a_{ij}\mathbf{U}_j$ is a discrete approximation of the conservative variables vector at $\boldsymbol{x}_i$, $\widetilde{\mathbf{F}}_i^k=\mathbf{F}^k(\mathbf{U}_i)$ is the kth component of the advective fluxes and $\widetilde{\mathbf{F}}_{ij}$ is a numerical flux computed at the midpoint of the ray (edge) connecting $\boldsymbol{x}_i$ to point $\boldsymbol{x}_j$ of Ω_i. Following [8], the approximate Riemann solver of Roe [9] is adopted to define the numerical flux. Thus,

$$\widetilde{\mathbf{F}}_{ij}^k=1/2\left(\widetilde{\mathbf{F}}_j^k+\widetilde{\mathbf{F}}_i^k\right)-1/2|\mathbf{A}_{\hat{n}}(\mathbf{U}_i,\mathbf{U}_j)|(\mathbf{U}_j-\mathbf{U}_i)\hat{n}_{ij}^k \tag{7}$$

where $\hat{\boldsymbol{n}}_{ij}$ is a unit vector in the direction of $\boldsymbol{l}_{ij}=\boldsymbol{x}_j-\boldsymbol{x}_i$ and $|\mathbf{A}_{\hat{n}}(\mathbf{U}_i,\mathbf{U}_j)|$ is the positive Roe matrix calculated in the same direction (see [10]). Aimed at increasing the spatial accuracy of Eq. (7), the variables $(\mathbf{U}_i,\mathbf{U}_j)$ are replaced by leftward and rightward higher-order reconstructions $\left(\mathbf{U}_i^+,\mathbf{U}_j^-\right)$ obtained by limited MUSCL extrapolation along $\boldsymbol{l}_{ij}$. The van Albada limiter is adopted (cf. [8] for implementation details).

A multi-stage time marching scheme is used to discretize Eq. (6) in time. The problem solution is advanced from a time level t^n to a level t^{n+1} by

$$\widehat{\mathbf{U}}_i^{(m)}=\widehat{\mathbf{U}}_i^{(m-1)}-\alpha_m\Delta t_i\mathbf{R}_i\left(\mathbf{U}_j^{(m-1)}\right) \quad m=1,s \tag{8}$$

where $\widehat{\mathbf{U}}_i^{(0)}=\widehat{\mathbf{U}}_i^n$ and $\widehat{\mathbf{U}}_i^{n+1}=\widehat{\mathbf{U}}_i^{(s)}$. In Eq. (8), $\mathbf{R}_i(\mathbf{U})$ denotes the residual vector (right-hand side of Eq. (6)), α_m are suitable integration coefficients [11], Δt_i is a local time increment and s is the number of integration stages. Implicit residual smoothing [12] is employed to accelerate the convergence in time of Eq. (8) and the diffusion terms in $\mathbf{R}_i$ are frozen at the first integration stage to reduce the computational cost.

It must be stressed that the FPM approximation does not interpolate nodal data (i.e. $\mathbf{U}(x_j)\neq\widehat{\mathbf{U}}(x_j)$). Thus, taking into account that $\mathbf{R}(\cdot)$ in Eq. (8) is a function of $\mathbf{U}_j$ $\boldsymbol{x}_j\in\Omega_i$, the following linear system has to be solved at each integration stage

$$\sum_{\forall j\in\Omega_i}a_{ij}\mathbf{U}_j=\widehat{\mathbf{U}}_i \quad i=1,n \tag{9}$$

The solution is obtained using a small number of Gauss–Seidel iterations with no significant additional computational cost.

2.2.1. Construction of local clouds and the numerical approximation

The procedure adopted to construct the local clouds of points follows the general lines proposed in [8]. In brief, given a point discretization bounded by a triangulation with associated geometrical data, for each star point $\mathbf{x}_i$ in Ω a set of neighboring points within a given radius is sought by using a spatial search algorithm based on bins, see for instance [13]. This initial cloud is then filtered to match boundary restrictions (if a ray from $\mathbf{x}_i$ to another point $\mathbf{x}_j$ in the local cloud pierces a boundary, the point $\mathbf{x}_j$ is discarded) and, if the number of admissible points is lower than a given threshold (about 120 in 3D), the search radius is increased and the procedure starts again. Next, the admissible points are triangulated and the first layer of Delaunay neighbors of $\mathbf{x}_i$ is retained and stored (this guarantees the necessary overlapping of the clouds throughout the domain).

The numerical approximation is computed after generating the layers of nearest neighbors. For each star point $\mathbf{x}_i$ in Ω a local cloud is initialized with its Delaunay neighbors. If the number of points is lower than $np_{\min}$ (about 30 in 3D), further points are added from an auxiliary neighbors list. This auxiliary list is constructed by adding points according to layers and ordering them by increasing distances from $\mathbf{x}_i$. Then, the minimization problem is solved and a quality check of the local approximation is performed, cf. [8]. If the test is not successful, additional points are added from the auxiliary list while $np\leqslant np_{\max}$ ($np_{\max}=50$ is enough to achieve a proper approximation in highly distorted clouds of points).

The procedure described above has proven reliable in general problems in which the data available are the coordinates of the points and a boundary grid delimiting the analysis domain.