Euro. Jnl of Applied Mathematics (2018), *vol.* 29, *pp.* 450–469.
doi:10.1017/S0956792517000249

Application of a generalized finite difference method to mould filling process

E. O. RESÉNDIZ-FLORES[1], J. KUHNERT[2] and F. R. SAUCEDO-ZENDEJO[3]

[1]*Division of Postgraduate Studies and Research, Department of Metal-Mechanical Engineering, The Technological Institute of Saltillo, Blvd. V. Carranza 2400 Col. Tecnológico C.P. 25280, Saltillo Coahuila, MX*
email: eresendiz@itsaltillo.edu.mx
[2]*Fraunhofer-Institut für Techno-und Wirtschaftsmathematik, Fraunhofer-Platz-1, 67663 Kaiserslautern, Germany*
email: joerg.kuhnert@itwm.fraunhofer.de
[3]*Division of Postgraduate Studies and Research, The Technological Institute of Saltillo, Blvd. V. Carranza 2400 Col. Tecnológico C.P. 25280, Saltillo Coahuila, MX*
email: feliks@live.com.mx

(*Received 22 May 2017; revised 12 July 2017; accepted 31 July 2017; first published online 24 August 2017*)

This paper proposes the use of a generalized finite difference method for the numerical simulation of free surface single phase flows during mould filling process which are common in some industrial processes particularly in the area of metal casting. A novel and efficient idea for the computation of the normal vectors for free surface flows is introduced and presented for the first time. The incompressible Navier–Stokes equations are numerically solved by the well-known Chorin's projection method. After we showed the main ideas behind the meshless approach, some numerical results in two and three dimensions are presented corresponding to mould filling process simulation.

Key words: 97M50, 76M25, 35Q35, 35R35

1 Introduction

Metal casting processes are one of the oldest techniques used to manufacture a wide range of metal components with specific dimensions and shapes for different applications. These processes start with a poured process which is simply to fill a mould with molten material which takes the shape thereof and then is allowed to solidify [33].

Mould filling is one of the most important factors that determine the quality of casting end products. During metal filling, free surface flow which initially enters into the mould can be divided into streams and these, at the same time, could be partially sprayed therein due to its high geometric complexity resulting in a set of defects in the end product [4]. One of such defect is mainly due to the generation of additional free surfaces that leads to the appearance of secondary phase layers (for example, oxide layers when the casting process takes place in ambient atmosphere).

In order to obtain homogeneous casting components with an acceptably low amount of defects, it is required to make a proper design process in which the various mould filling gates and the positions of vents assure an optimum filling flow. Improvements in product quality and process productivity can be achieved by improving the design of

casting. These include the development of more effective control of mould filling and mould thermal performance [12].

Numerical simulation offers a powerful alternative at an attractive cost to study the effectiveness of the different designs and mould filling processes. In the last decades, several numerical methods and software packages have been developed to simulate and analyse the metal flow during mould filling. These methods and packages are based on traditional methods such as Finite Element Method [13,16,25,31] and Finite Volume Method [1,19], and generally they use the Volume of Fluid Method [17] to track free surfaces. However, these methods become both computational and economically expensive to the real-world problems of today [26].

Meshless methods have been developed as an alternative to mesh-based methods and they are able to overcome some of the limitations presented in traditional methods [2,28,32]. One of the first meshless methods that has been developed was the Smoothed-Particle Hydrodynamics (SPH). This Lagrangian method was originally proposed to solve borderless problems in gas dynamics of astrophysical interest [15,27]. Since that time, the SPH has proved to be a very effective method to predicting complex fluid flow in injection moulding process and casting of components having highly complex three-dimensional shapes. The SPH also has proved to be suitable for simulating mechanical processes such as metal forming and extrusion forging due to its ability to model the behaviour of complex-free surfaces and their ability to tolerate high levels of deformation as well as tracking deformation history [3,5–12,20].

Nonetheless, since the development of the original version, SPH suffered instability, inconsistency and difficulties in proper treatment of the boundary conditions so that over the next years, many improvements were incorporated to the original SPH formulation [26,28] and several alternative meshless methods have been proposed [14].

A Lagrangian truly meshless approach that can overcome some of the problems in SPH formulation and especially those related to the treatment of the boundary conditions is the Finite Pointset Method (FPM) developed by Kuhnert [21], at the Fraunhhofer Institute for Industrial Mathematics (ITWM by its name in German) in Kaiserslautern, Germany, it has been continuously studied and applied to problems in fluid dynamics [22,38,39]. FPM uses a weighted least squares (LS) interpolation scheme to approximate the field variables and their derivatives at each point of the domain. Some of the advantages of this method is its relatively simple numerical implementation and the boundary conditions can be implemented in a natural way just prescribing them on those points placed on the boundary [38].

FPM has shown to be far superior to traditional mesh-based methods due to its Lagrangian nature for problems involving rapidly changing flow domains with respect to time, multiphase or free surface flows [18,23,36,39–41]. This makes it especially attractive for processes simulation in particular for the metal casting industry where free surface flows with rapidly changing flow domains over time are taking place. Thus, as an attempt to develop simulation software for mould filling processes in the metal casting industry, we propose in this work the application of the generalized finite difference method, called FPM, developed by Kuhnert to this particular field of research. To the authors knowledge, this is the first time that the presented original idea for the normal vector computation as well as some numerical results in 3D for a mould filling process simulation using this meshless approach are reported in the scientific literature.

The structure of the paper is as follows: Section 2 presents the governing partial differential equations for fluid flow and the well-known Chorin's projection method for the numerical solution of the corresponding PDEs. Section 3 shortly describe the basic ideas behind FPM. Section 4 deals with the free surface velocities derivatives. Section 5 presents a completely new and original idea for the computation of normal vectors on the free surface boundaries. The numerical simulations are presented in Section 6 with the corresponding results followed by some conclusions given in last section.

2 Mathematical model

The considered mathematical model for incompressible low Reynolds number flow are the Navier–Stokes equations in Lagrangian form which read

$$\frac{D\mathbf{x}}{Dt} = \mathbf{v}, \tag{2.1a}$$

$$\frac{D\mathbf{v}}{Dt} = \frac{-1}{\rho}\nabla p + \mu \Delta \mathbf{v} + \mathbf{g}, \tag{2.1b}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{2.1c}$$

where D/Dt denotes the material derivative, $\mathbf{x}$ represents the trajectories of material points, $\mathbf{v}$ is the fluid velocity field, p is the fluid pressure, ρ denotes the fluid density, μ the dynamic viscosity and $\mathbf{g}$ the gravitational vector field.

Different boundary conditions must be considered in the case of mould filling processes, in particular, we imposed no-slip and slip conditions on mould walls and inlet channel walls, respectively, while constant inlet velocity condition on the inlet surface has been fixed.

The boundary condition on the free surface is defined as the jump ($[\cdot]$) on the interface between two fluids thus the surface stress boundary reads

$$[\tau \cdot \mathbf{n} - p\mathbf{n}] = \sigma\kappa\mathbf{n}, \tag{2.2}$$

where τ is the viscous stress tensor, σ is the fluid surface tension, κ denotes the curvature on the interface and $\mathbf{n}$ is the unit normal vector on the interface. This can be represented under certain assumptions as follows:

$$p - \mathbf{n} \cdot \tau \cdot \mathbf{n} = p_0 + \sigma\kappa, \tag{2.3a}$$

$$\mathbf{t} \cdot \tau \cdot \mathbf{n} = 0, \tag{2.3b}$$

with p_0 the fixed pressure corresponding of a fluid in contact with the free surface.

Now, let $\mathbf{x}$ be a set of numerical points/particles representing a finite part of the fluid and distributed on the entire fluid domain of interest, these particles carry all the relevant information as velocity, density, pressure, etc. These particles obey the fluid dynamics and need to be updated as time evolves. Since such particles obey the incompressible Navier–Stokes equations, the first-order Chorin's projection method is implemented for numerically solving and this is shortly presented for completeness [38] and [42].

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

Let $\mathbf{x}^k$ and $\mathbf{v}^k$ be initially given. They denote the particles positions and velocity field evaluated at these particles at time t^k, respectively. The Chorin's projection method is of first-order accuracy in time and it consist of the following two main steps:

- Update particle positions and temporal fluid velocities at time t^{k+1}

$$\mathbf{x}^{k+1} = \mathbf{x}^k + dt\mathbf{v}^k, \tag{2.4a}$$

$$\tilde{\mathbf{v}} = \mathbf{v}^k + dt\Delta\mathbf{v}^k + dt\mathbf{g}. \tag{2.4b}$$

- Correct fluid velocity

$$\mathbf{v}^{k+1} = \tilde{\mathbf{v}} - dt\nabla p^{k+1}, \tag{2.5}$$

with the constraint $\nabla\cdot\mathbf{v}^{k+1} = 0$. When using the free divergence constraint, the following Poisson equation arises:

$$\Delta p^{n+1} = \frac{\nabla\cdot\tilde{\mathbf{v}}}{dt}. \tag{2.6}$$

Please note that the pressure gradient update ∇p^{k+1} is obtained when solving equation (2.6).

3 The FPM main ideas

In this section, we describe the main ideas of FPM proposed by [21]. The FPM is a member of the family of the LS methods and it is closely related to the finite point method by Oñate et al. [29] and [30]. Although they are very similar, they are not identical. The main difference is that finite point method of Oñate uses polynomial basis and the FPM method uses Taylor series which allows to compute, by an LS approach, the function and its derivatives values that naturally appear as unknown coefficients in the series.

The method is based on the so-called moving LS procedure which is shortly described next, following [24]:

Let Ω be a given domain with boundary $\partial\Omega$ and suppose that the set of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ is distributed with corresponding function values $f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)$. The problem is to find an approximate value of f at some arbitrary location $f(\mathbf{x})$. Thus, the following procedure can be applied:

Define the approximation to $f(\mathbf{x})$ as

$$\tilde{f}(\mathbf{x}) = \sum_{k=1}^{m} p_k(\mathbf{x})b_k(\mathbf{x}) = \mathbf{p}^t(\mathbf{x})\mathbf{b}(\mathbf{x}), \tag{3.1}$$

whose local version reads

$$\tilde{f}(\mathbf{x}, \bar{\mathbf{x}}) = \sum_{k=1}^{m} p_k(\bar{\mathbf{x}})b_k(\mathbf{x}) = \mathbf{p}^t(\bar{\mathbf{x}})\mathbf{b}(\mathbf{x}), \tag{3.2}$$

where $p_k(\mathbf{x})$ denotes a set of linear independent functions, in particular, they can be linear monomials, m is the number of basis functions and $b_k(\mathbf{x})$ are the corresponding coefficients

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

that need to be computed. The idea of this local version is to approximate a continuous field by patches of polynomial surfaces where these are fitted to the data points in the sense of LS using only local information for each computational point.

Consider also a weight function w which should satisfy several properties of a reproducing kernel, in particular, the normalization condition, the Dirac-delta function and compact support properties. Moreover, different weight functions have been used in the literature and the most common ones are the cubic spline and the Gaussian functions, being the last one chosen to be used in this paper which reads

$$\mathrm{w}(\mathbf{x}-\mathbf{x}_i)=\begin{cases} e^{-\gamma\|\mathbf{x}-\mathbf{x}_i\|^2/h^2}, & \text{if } \frac{\|\mathbf{x}-\mathbf{x}_i\|}{h}\leqslant 1 \\ 0 & else \end{cases}, \tag{3.3}$$

where h is the smoothing length and γ is a positive constant.

Now, minimize the quadratic form

$$J=\sum_{j=1}^{n}\mathrm{w}(\mathbf{x},\mathbf{x}_j)e_j^2, \tag{3.4a}$$

$$=\sum_{j=1}^{n}\mathrm{w}(\mathbf{x},\mathbf{x}_j)\left(\sum_{k=1}^{m}p_k(\mathbf{x}_j)b_k(\mathbf{x})-f(\mathbf{x}_j)\right)^2, \tag{3.4b}$$

in order to get the optimal coefficients

$$\mathbf{b}=(P^tWP)^{-1}(P^tW)\mathbf{f}, \tag{3.5}$$

where

$$P=\begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_{n_p}) & p_2(\mathbf{x}_{n_p}) & \cdots & p_m(\mathbf{x}_{n_p}) \end{bmatrix}, \tag{3.6}$$

$$W=\begin{bmatrix} \mathrm{w}(\mathbf{x},\mathbf{x}_1) & 0 & \cdots & 0 \\ 0 & \mathrm{w}(\mathbf{x},\mathbf{x}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathrm{w}(\mathbf{x},\mathbf{x}_{n_p}) \end{bmatrix}, \tag{3.7}$$

$$\mathbf{f}=[f(\mathbf{x}_1),f(\mathbf{x}_2),\ldots,f(\mathbf{x}_{n_p})^t=[f_1,f_2,\ldots,f_{n_p}]^t, \tag{3.8}$$

$$\mathbf{b}(\mathbf{x})=[b_1(\mathbf{x}),b_2(\mathbf{x}),\ldots,b_m(\mathbf{x})]^t, \tag{3.9}$$

n_p denotes the number of neighbour points $\mathbf{x}_j$ of $\mathbf{x}$ and $\mathrm{w}(\mathbf{x},\mathbf{x}_j)$ denotes a weight function with compact support.

Once $\mathbf{b}$ is known, the function approximation at point $\mathbf{x}$ reads

$$\tilde{f}(\mathbf{x})=\sum_{k=1}^{m}p_k(\mathbf{x})b_k(\mathbf{x})=\mathbf{p}^t(\mathbf{x})(P^tWP)^{-1}(P^tW)\mathbf{f}=\Phi(\mathbf{x})\mathbf{f}. \tag{3.10}$$

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

If the base functions $p_i(\mathbf{x})$ are defined as follows:

$$\mathbf{p}^t = [1, \Delta x_j, \Delta y_j, \Delta z_j, \ldots], \tag{3.11}$$

which in the particular case of a second-order approximation, the vector takes the following form:

$$\mathbf{p}^t = \left[1, \Delta x_j, \Delta y_j, \Delta z_j, \frac{1}{2}\Delta x_j^2, \Delta x_j \Delta y_j, \Delta x_j \Delta z_j, \frac{1}{2}\Delta y_j^2, \Delta y_j \Delta z_j, \frac{1}{2}\Delta z_j^2\right], \tag{3.12}$$

where $\Delta x_j = x_j - x$, $\Delta y_j = y_j - y$ and $\Delta z_j = z_j - z$ for $j = 1, \ldots, n_p$. Thus, considering the general approximation (3.11), the following equivalent representation is obtained:

$$\tilde{f}(\mathbf{x}) \simeq \sum_{k=1}^{m} p_k(\mathbf{x}) b_k(\mathbf{x}) = f(\mathbf{x}_j) + \nabla f(\mathbf{x}_j) \cdot \Delta \mathbf{x}_j + \ldots, \tag{3.13}$$

which implies that under this representation, the new vector of unknown coefficients becomes

$$\mathbf{b}(\mathbf{x}) = [f(\mathbf{x}), \partial_x f(\mathbf{x}), \partial_y f(\mathbf{x}), \ldots]^t. \tag{3.14}$$

In this way, we automatically get the values of a function and its derivatives at points $\mathbf{x}$ which are needed for the computation of the pressure through (3.14) and for the approximation of the velocity and pressure spatial derivatives in (2.4) and (2.5). We refer to [37] for a more explicit presentation of the FPM method applied to the Poisson equation.

4 Velocities derivatives on free surface points

During FPM implementation, the efficient numerical approximation of velocities derivatives should be carefully considered in particular those on free surface particles since (2.2) explicitly couples the components of velocity field. Therefore, we need to consider the Taylor series expansion of $\mathbf{v} = (u, v, w)$ around (x, y, z) coupled with the boundary condition on this surface (2.2). This can be rewritten in matrix form as

$$\mathbf{e} = P\mathbf{b} - \mathbf{f},$$

with

$$P = \begin{pmatrix} P_{xyz} & 0 & 0 \\ 0 & P_{xyz} & 0 \\ 0 & 0 & P_{xyz} \\ \xi_1 & \xi_2 & \xi_3 \end{pmatrix}, \tag{4.1}$$

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

where

$$P_{xyz} = \begin{pmatrix} dx_1 & dy_1 & dz_1 & \frac{1}{2}dx_1^2 & dx_1dy_1 & dx_1dz_1 & \frac{1}{2}dy_1^2 & dy_1dz_1 & \frac{1}{2}dz_1^2 \\ dx_2 & dy_2 & dz_2 & \frac{1}{2}dx_2^2 & dx_2dy_2 & dx_2dz_2 & \frac{1}{2}dy_2^2 & dy_2dz_2 & \frac{1}{2}dz_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ dx_n & dy_n & dz_n & \frac{1}{2}dx_n^2 & dx_ndy_n & dx_ndz_n & \frac{1}{2}dy_n^2 & dy_ndz_n & \frac{1}{2}dz_n^2 \end{pmatrix}, \tag{4.2}$$

$$\xi_1 = \begin{pmatrix} 2\mu n_x^2 & 2\mu n_x n_y & 2\mu n_x n_z & 0 & 0 & 0 & 0 & 0 & 0 \\ 2s_x n_x & s_y n_x + s_x n_y & s_z n_x + s_x n_z & 0 & 0 & 0 & 0 & 0 & 0 \\ 2t_x n_x & t_y n_x + t_x n_y & t_z n_x + t_x n_z & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.3}$$

$$\xi_2 = \begin{pmatrix} 2\mu n_y n_x & 2\mu n_y^2 & 2\mu n_y n_z & 0 & 0 & 0 & 0 & 0 & 0 \\ s_y n_x + s_x n_y & 2s_y n_y & s_z n_y + s_y n_z & 0 & 0 & 0 & 0 & 0 & 0 \\ t_y n_x + t_x n_y & 2t_y n_y & t_z n_y + t_y n_z & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.4}$$

$$\xi_3 = \begin{pmatrix} 2\mu n_z n_x & 2\mu n_z n_y & 2\mu n_z^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_z n_x + s_x n_z & s_y n_z + s_z n_y & 2s_z n_z & 0 & 0 & 0 & 0 & 0 & 0 \\ t_z n_x + t_x n_z & t_y n_z + t_z n_y & 2t_z n_z & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.5}$$

$$\mathbf{b} = (u_x, u_y, u_z, u_{xx}, u_{xy}, u_{xz}, u_{yy}, u_{yz}, u_{zz}, v_x, v_y, v_z, v_{xx}, v_{xy}, v_{xz}, v_{yy}, v_{yz}, v_{zz}, \dots \\ w_x, w_y, w_z, w_{xx}, w_{xy}, w_{xz}, w_{yy}, w_{yz}, w_{zz})^t,$$

and

$$\mathbf{f} = (u_1 - u, u_2 - u, \dots, u_n - u, v_1 - v, v_2 - v, \dots, v_n - v, w_1 - w, w_2 - w, \dots, w_n - w, \\ p - p_0 + \sigma\kappa, 0, 0)^t.$$

$\mathbf{n} = (n_x, n_y, n_z)^t$ is a normal vector, while $\mathbf{s} = (s_x, s_y, s_z)^t$ and $\mathbf{t} = (t_x, t_y, t_z)^t$ denote tangential vectors to the free surface. Therefore, unlike the procedure for inner particles, from the expressions for $\mathbf{b}$ and P, it can be seen that at least 27 neighbouring particles are needed for the computation of the velocity derivatives for free surface points.

5 Computation of normal vector on free surface

For the numerical computation of free surface normal vectors, some details are reported for the two-dimensional case in [38] and [42]. Considering these ideas in three dimensions, we can solve the following model for the sphere:

$$x^2 + y^2 + z^2 + Dx + Ey + Fz + G = 0,$$

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

once the constants are determined, the centre and radius of the sphere can be computed

$$(x_c, y_c, z_c) = (-D/2, -E/2, -F/2), \tag{5.1a}$$

$$R = \frac{1}{2}\sqrt{D^2 + E^2 + F^2 - 4G}, \tag{5.1b}$$

hence the free surface curvature and unit normal vector read

$$\kappa = \frac{1}{R}, \tag{5.2a}$$

$$\mathbf{n} = (\mathbf{x} - \mathbf{x}_c)\kappa. \tag{5.2b}$$

Note that this must be done for every particle on the free surface flow.

Another more efficient and elegant approach to compute the boundary normal vector on the free surface flow proposed by the second author is for the first time reported as follows:

We are interested in computing the outward normal vector $\mathbf{n}$ at the point $\mathbf{x}_i$ on the free surface. Thus, let $\mathbf{x}_j$ be one of the N_b free surface particles in a neighbourhood defined by a weight function ω_{ij}. The weight function ω_{ij} can be quite arbitrary; however, in this work, we consider a Gaussian weight function. Then, the angle between the boundary normal and the vector $\mathbf{x}_j - \mathbf{x}_i$ is given by

$$\cos\alpha_{ij} = \frac{\mathbf{n}\cdot\left(\mathbf{x}_j - \mathbf{x}_i\right)}{\|\mathbf{x}_j - \mathbf{x}_i\|\,\|\mathbf{n}\|}., \tag{5.3}$$

moreover, the following quadratic form arises

$$\cos^2\alpha_{ij} = \frac{\mathbf{n}^T\left(\mathbf{x}_j - \mathbf{x}_i\right)\left(\mathbf{x}_j - \mathbf{x}_i\right)^T\mathbf{n}}{\|\mathbf{x}_j - \mathbf{x}_i\|^2}, \tag{5.4}$$

if all the boundary particles in the neighbourhood of $\mathbf{x}_i$ are considered, we have

$$\sum_j \cos^2\alpha_{ij}\omega_{ij} = \mathbf{n}^T\sum_j\left(\frac{\left(\mathbf{x}_j - \mathbf{x}_i\right)\left(\mathbf{x}_j - \mathbf{x}_i\right)^T\omega_{ij}}{\|\mathbf{x}_j - \mathbf{x}_i\|^2}\right)\mathbf{n} \tag{5.5}$$

defining

$$C = \sum_j\left(\frac{\left(\mathbf{x}_j - \mathbf{x}_i\right)\left(\mathbf{x}_j - \mathbf{x}_i\right)^T\omega_{ij}}{\|\mathbf{x}_j - \mathbf{x}_i\|^2}\right). \tag{5.6}$$

Equation (5.5) can be expressed as

$$\sum_j \cos^2\alpha_{ij}\omega_{ij} = \mathbf{n}^T C\mathbf{n}, \tag{5.7}$$

hence, the outward normal vector $\mathbf{n}$ is automatically obtained when the following optimal problem is solved:

$$\min_{\mathbf{n}\in\mathbb{R}^3,\|\mathbf{n}\|=1}\sum_j \cos^2\alpha_{ij}\omega_{ij} = \min_{\mathbf{n}\in\mathbb{R}^3,\|\mathbf{n}\|=1}\mathbf{n}^T C\mathbf{n}, \tag{5.8}$$

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

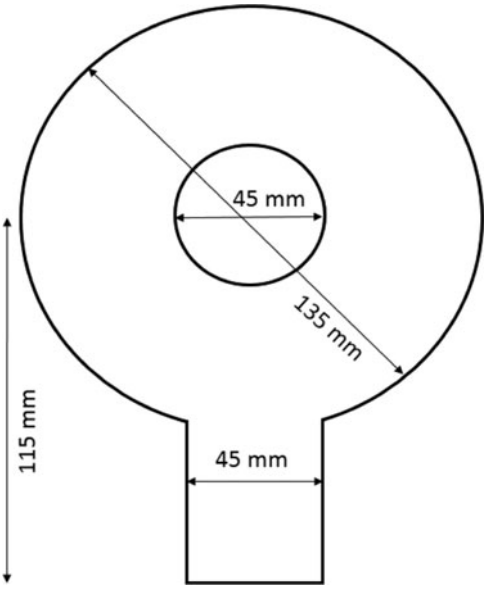

FIGURE 1. Mould geometry configuration in two dimensions.

which leads to the following eigenvalue problem:

$$C\mathbf{n} = \beta\mathbf{n}. \tag{5.9}$$

6 Numerical examples

In this section, mould filling processes in two dimensions and three dimensions are simulated using FPM.

6.1 Complex filling in two dimensions

For the reported numerical simulation in two dimensions, we have considered the following problem configuration taken from [34] and depicted in Figure 1.

We consider the filling of this mould under three different cases. The first simulation corresponds to the following fluid parameters: $\rho = 1,000\,\mathrm{kg\,m^{-3}}$, kinematic viscosity $v = 0.01\,\mathrm{m^2\,s^{-1}}$ and the inlet velocity $V = 18\,\mathrm{m\,s^{-1}}$. Under this configuration, we obtain $\mathrm{Re} = 81$. Regarding the shape function w, its parameters have been taken as $h = 3.0$ and $\gamma = 6.5$. The total number of particles used in this simulation was roughly 6,000 and the initial particle spacing for the particles was 1.5 mm.

Figure 2 shows the results of the mould filling pattern numerically predicted by FPM, the traditional SPH in [34], those obtained in [34] using the SPH_DTKGC method and the experimental results in [35] at different times. Comparisons among these results show that FPM and SPH_DTKGC methods are by far more stable than the traditional SPH method for simulating complex filling processes. Here, it can also be noted that, as opposed to FPM, SPH_DTKGC method had slight instabilities which spread on the free

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

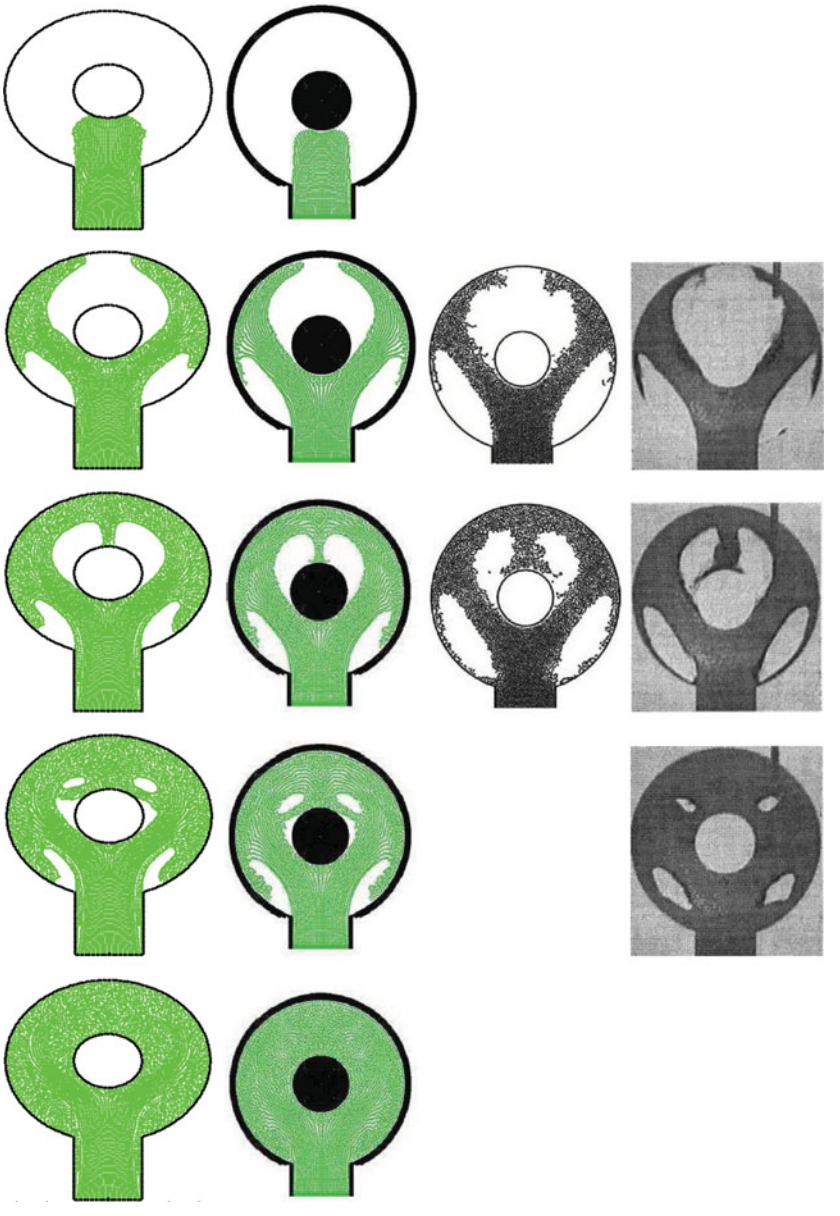

FIGURE 2. Mould filling patterns for Re = 81 at different time steps. *First column:* FPM. *Second column:* SPH_DTKGC in [34]. *Third column:* traditional SPH in [34]. *Fourth column:* the experimental pattern in [35].

surfaces. Hence, FPM method is more stable and reliable than SPH_DTKGC method to simulate complex filling processes.

Figures 3 and 4 show the velocity and pressure profiles for this inflow predicted by FPM and SPH_DTKGC in [34]. Both methods show similar and regular pressure fields without oscillations which is very important for achieving good filling processes. Regarding

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

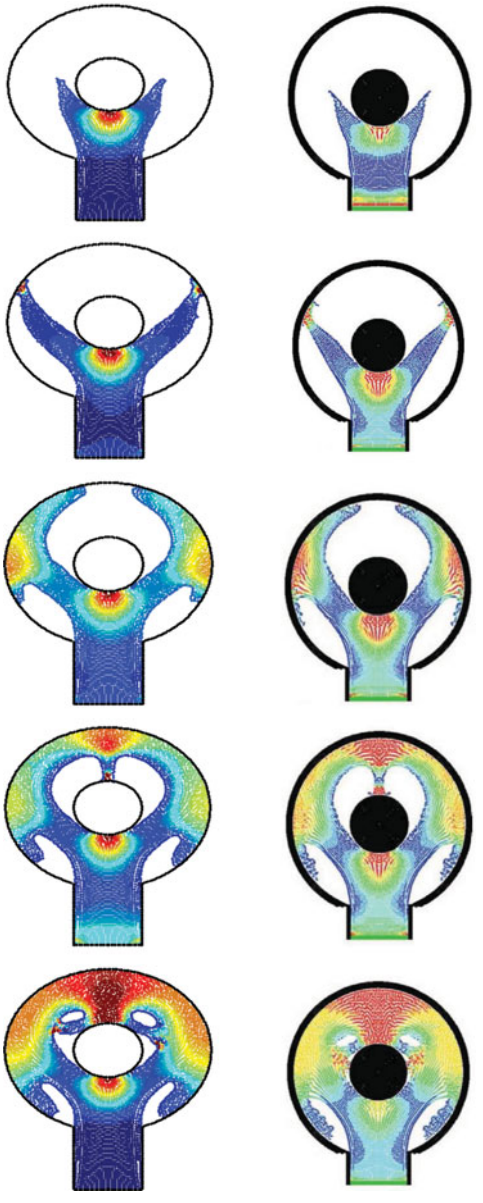

FIGURE 3. Pressure profiles for Re = 81 at different time steps. *Left column:* FPM. *Right column:* SPH_DTKGC in [34].

velocity profiles, for both methods, it can be seen that near the mould walls fluid velocity smoothly tends to be low, while speed is getting bigger in some remote regions. This pattern is directly produced by the viscosity forces between the mould walls and the fluid particles, and among fluid particles. All these results indicate that FPM is accurate and stable for simulating mould filling processes in foundry industry.

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

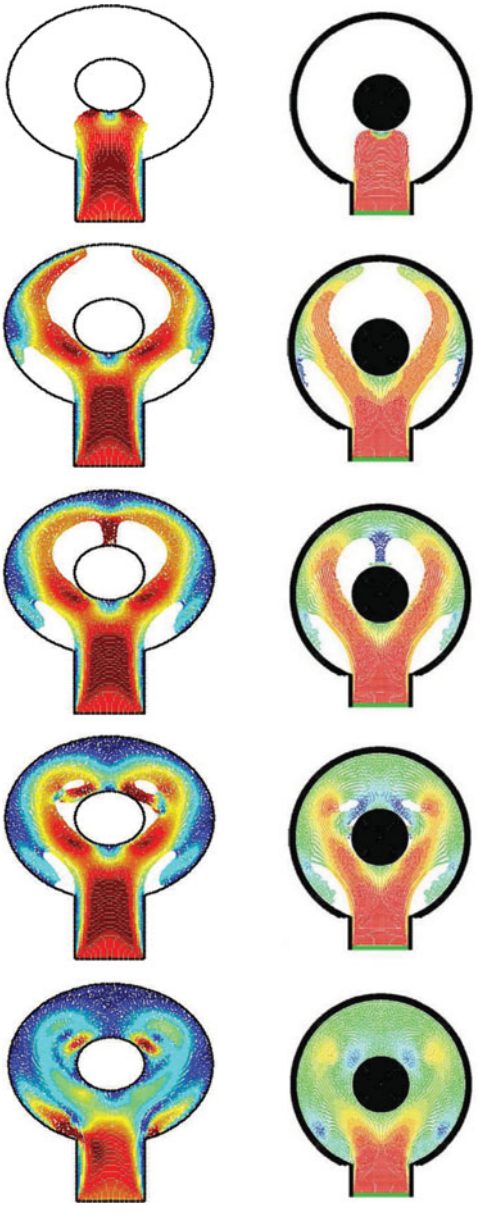

FIGURE 4. Velocity contours for Re = 81 at different time steps. *Left column:* FPM. *Right column:* SPH_DTKGC in [34].

The parameters used in [34] were taken for the second and third fluid configurations: $\rho = 1{,}000\,\mathrm{kgm}^{-3}$, $v = 0.01\,\mathrm{m}^2\,\mathrm{s}^{-1}$, $V = 10\,\mathrm{m\,s}^{-1}$ which results in Re = 45 for the second fluid, and $\rho = 1{,}000\,\mathrm{kg\,m}^{-3}$, $v = 0.1\,\mathrm{m}^2/\mathrm{s}$, $V = 2\,\mathrm{m\,s}^{-1}$ giving Re = 0.9 for the last case. In both simulations, the weighting function support was fixed to $h = 3.0$ and $\gamma = 6.5$. Now the total number of particles was roughly 6,000 with an initial particle spacing of 1.5 mm.

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

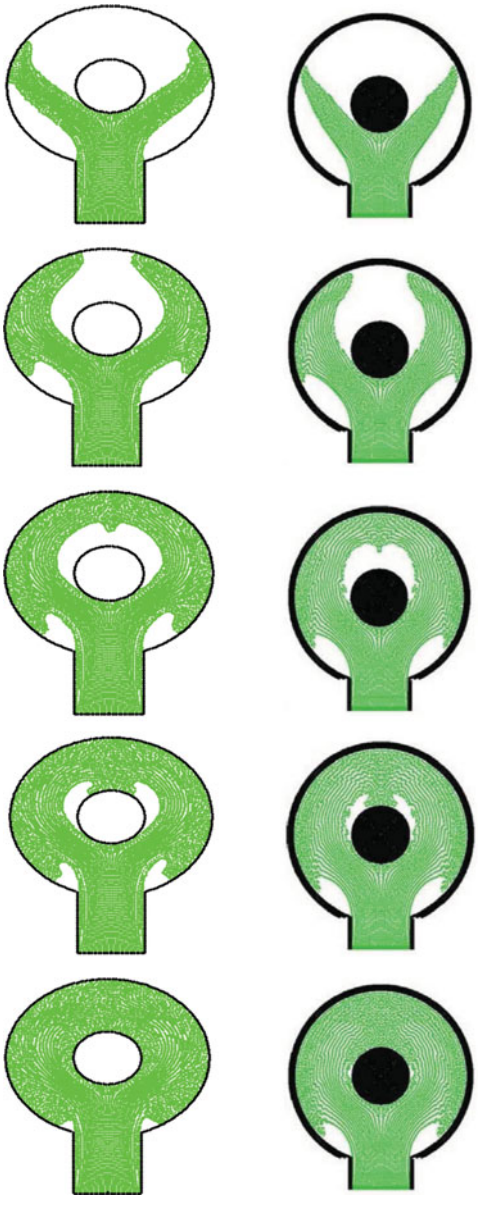

FIGURE 5. Mould filling patterns for Re = 45 at different time steps. *Left column:* FPM. *Right column:* SPH_DTKGC in [34].

Figure 5 shows the results of the mould filling patterns numerically predicted by FPM and SPH_DTKGC method in [34] for Re = 45 at different time steps. Comparisons among these results show similar filling behaviours; nevertheless, it can be seen that SPH_DTKGC approach still have slight numerical instabilities which spread on the free surfaces.

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

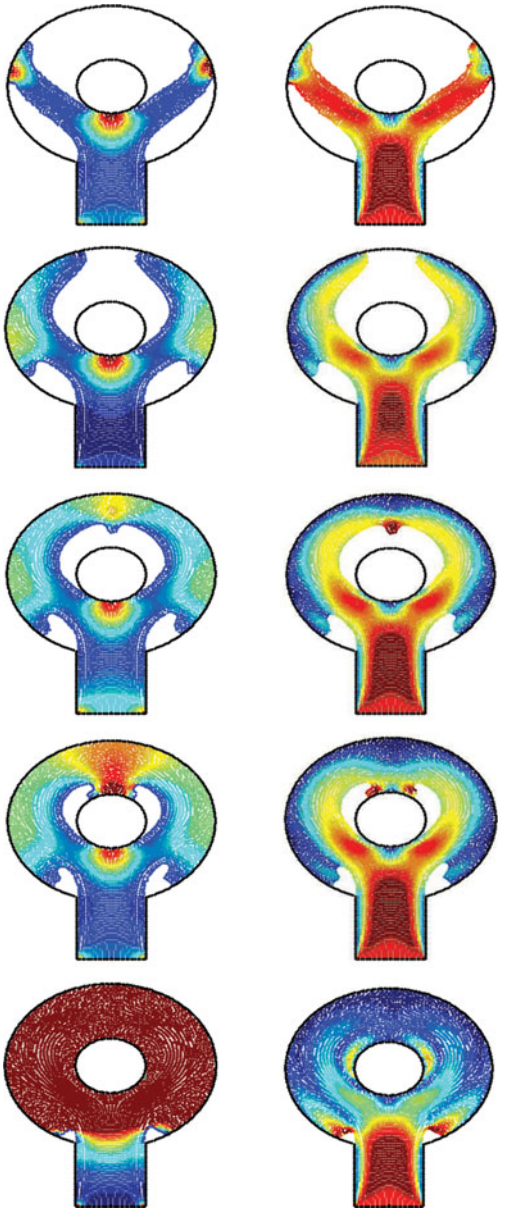

FIGURE 6. *Left column:* Pressure. *Right column:* velocity profiles for Re = 45 at different time steps (FPM).

Additionally, Figure 6 depicts the pressure and velocity profiles for this inflow predicted by FPM. It can be noted that the fluid velocity is reduced while travelling on the mould walls. This phenomenon occurs because the inflow velocity is smaller, causing that drag forces between fluid particles become more intense as it moves on the mould walls which quickly decreases the kinetic energy.

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

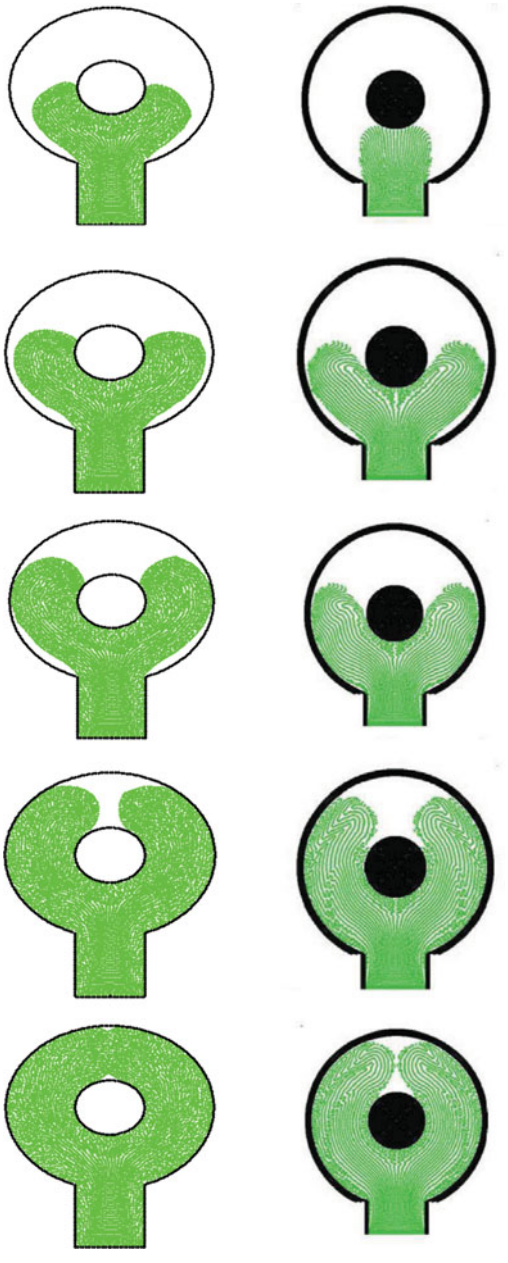

FIGURE 7. Mould filling patterns for Re = 0.9 at different time steps. *Left column:* FPM. *Right column:* SPH_DTKGC in [34].

Furthermore, Figure 7 shows the results of the filling patterns of the mould predicted by FPM and SPH_DTKGC method in [34] for Re = 0.9 at different time steps. Comparisons among these results show somewhat different filling patterns. It can be observed once again that SPH_DTKGC method shows slight numerical instabilities on free surfaces and it also was observed a high-density distribution of particles near the walls of the mould.

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

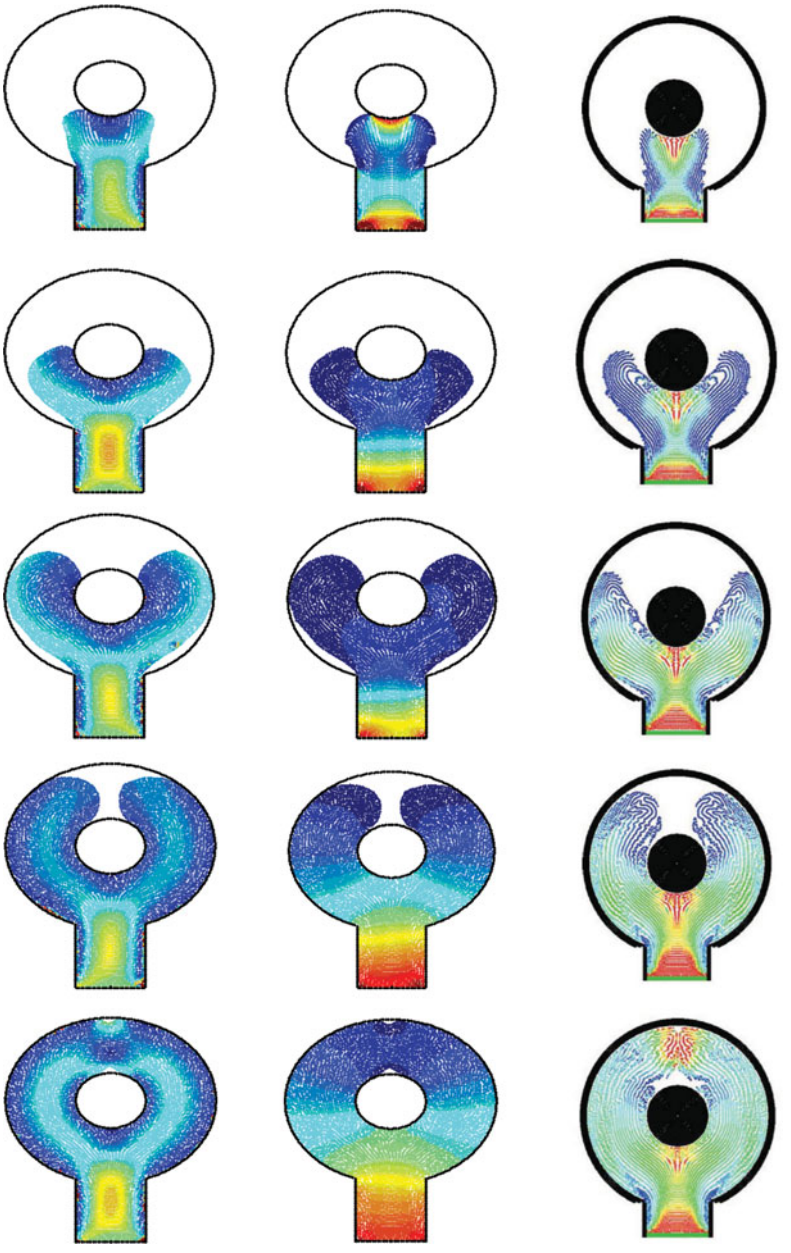

FIGURE 8. *Left column:* Velocity. *Centre column:* pressure profiles predicted by FPM. *Right column:* pressure profile predicted by SPH_DTKGC in [34] for $\mathrm{Re} = 0.9$ at different time steps.

This high density could be caused by the treatment given to the solid boundaries in this method. In contrast, the results of the simulation for the same time steps using FPM did not exhibit these irregularities on free surfaces or a high density of particles near the walls.

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

In order to determine the source of the differences between both filling patterns, Figure 8 report the velocity and pressure profiles predicted by FPM and the pressure profiles predicted by SPH_DTKGC in [34] for this case at different time steps. Comparisons among these results show stable but different pressure profiles. Since this fluid has a high viscosity and the inflow was slow, it is natural to expect roughly uniform pressure profiles with drag forces spread over mould walls. The results predicted by SPH_DTKGC in [34] seem to be higher for a flow under such conditions, causing the fluid to behave like in previous cases in the first steps. Coupled with this, in pressure profiles, it can be observed further instabilities on free surfaces and some irregularities in the domain. For FPM, otherwise, it can be observed roughly uniform pressure profiles which cause the fluid to travel over the mould walls in all steps and no instability on free surfaces or irregularity in the domain were observed. The velocity profiles predicted by FPM confirm this behaviour. Therefore, FPM works well to simulate complex filling processes.

6.2 Complex filling in 3D

For the numerical simulation in 3D, we have considered an extension of the configuration used in previous subsection with a thickness of 14.4 mm.

For this simulation, the following fluid parameters were considered: $\rho = 1\,\mathrm{kg\,m^{-3}}$, kinematic viscosity $\nu = 0.00019\,\mathrm{m^2\,s^{-1}}$ and the inlet velocity $V = 0.18\,\mathrm{m\,s^{-1}}$. Under this configuration, we obtain $\mathrm{Re} = 210$. The support of the weighting function has been fixed to $h = 3.0$ and $\gamma = 6.5$. The total number of particles used in this simulation was roughly 75,000 and the initial particle spacing was 1.6 mm.

Figure 9 shows the filling pattern of this mould predicted by FPM at different times. This shows an excellent filling pattern in 3D when compared with discussed results of previous subsection for the same geometry in two dimensions. The observed differences between three-dinensional and two-dimensional filling patterns for the same fluid are mainly due to the fact that in 3D configuration, the fluid is exposed to more drag forces. Considering these numerical results, we can conclude that FPM can be used to simulate the mould filling process in more realistic and complex 3D geometries.

7 Conclusions and future work

We have successfully implemented and reported for the first time and to the authors knowledge, the discussed version of the meshless method for unsteady three-dimensional fluid flow problems for mould filling process which can be used in the area of metal casting. Based on the numerical performance, we can conclude that FPM is other meshless method that can be used to successfully solve more complex free surface fluid problems. The biggest advantage of this approach over other meshless methods is that it is a real full meshfree method, since it does not need to compute any numerical quadratures, thus a background mesh arrangement for Gaussian points is not needed, therefore, it is a feasible and much less involved method for real problems solving. As future work, we are interested in several aspects as the implementation of higher order numerical schemes for solving in time and the coupling of a mathematical model for thermal energy

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

FIGURE 9. 3D Mould filling patterns for Re = 210 at different time steps predicted by FPM.

transfer and solidification as well as some additional physics which lead to multiphase flows.

References

[1] BASIC, H., DEMIRDZIC, I. & MUZAFERIJA, S. (2005) Finite volume method for simulation of extrusion processes. *Int. J. Numer. Methods Eng.* **62**(4), 475–494.

[2] BELYTSCHKO, T., KRONGAUZ, Y., ORGAN, D., FLEMING, M. & KRYSL, P. (1996) Meshless methods: An overview and recent developments. *Comput. Methods Appl. Mech. Engrg.* **139**(1–4), 3–47.

[3] BURUCHENKO, S. K. (2016) Three-dimensional simulation of tsunami run up around conical island using smoothed particle hydrodynamics. *IOP Conference Series: Earth and Environ. Sci.* **44**(3), 032026.

[4] CAMPBELL, J. (2003) Castings. *Advanced Materials Research*, 2nd ed. Butterworth-Heinemann, Great Britain.

[5] CLEARY, P. W., HA, J. & AHUJA, V. (2000) High pressure die casting simulation using smoothed particle hydrodynamics. *Int. J. Cast. Met. Res.* **12**(6), 335–355.

https://doi.org/10.1017/S0956792517000249 Published online by Cambridge University Press

[6] Cleary, P. W. & Ha, J. (2000) Three dimensional modelling of high pressure die casting. *Int. J. Cast. Met. Res.* **12**(6), 357–365.

[7] Cleary, P. W., Ha, J., Prakash, M. & Nguyen, T. (2006) 3D SPH flow predictions and validation for high pressure die casting of automotive components. *Appl. Math. Model.* **30**(11), 1406–1427.

[8] Cleary, P. W., Prakash, M. & Ha, J. (2006) Novel applications of smoothed particle hydrodynamics (SPH) in metal forming. *J. Mater. Process. Technol.* **177**(1–3), 41–48.

[9] Cleary, P. W., Ha, J., Prakash, M. & Nguyen, T. (2010) Short shots and industrial case studies: Understanding fluid flow and solidification in high pressure die casting. *Appl. Math. Modelling* **34**(8), 2018–2033.

[10] Cleary, P. W. (2010) Extension of SPH to predict feeding, freezing and defect creation in low pressure die casting. *Appl. Math. Model.* **34**(11), 3189–3201.

[11] Cleary, P. W., Ha, J., Prakash, M., Sinnott, M. D., Rudman, M. & Das, R. (2011) Large scale simulation of industrial, engineering and geophysical flows using particle methods. *Comput. Methods Appl. Sci.* **25**, 89–111.

[12] Cleary, P. W., Ha, J., Prakash, M., Alguine, V. & Nguyen, T. (2002) Flow modelling in casting processes. *Appl. Math. Model.* **26**(2), 171–190.

[13] Dhatt, G., Gao, D. M. and Cheikh, A. B. (1990) A finite element simulation of metal flow in moulds. *Int. J. Num. Meth. Engrg.* **30**(4), 821–831.

[14] Fang, J. & Parriaux, A. (2008) A regularized Lagrangian finite point method for the simulation of incompressible viscous flows. *J. Comput. Phys.* **227**(20), 8894–8908.

[15] Gingold, R. A. & Monaghan, J. J. (1977) Smoothed particle hydrodynamics: Theory and applications to non-spherical stars. *Mon. Not. R. Astron. Soc.* **181**(3), 375–389.

[16] Hetu, J. F., Gao, D. M., Kabanemi, K. K., Bergeron, S., Nguyen, K. T. & Loong, C. A. (1998) Numerical modeling of casting processes. *Adv. Perform. Mate.* **5**(1–2), 65–82.

[17] Hirt, C. W. & Nichols, B.D. (1981) Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* **39**(1), 201–225.

[18] Jefferies, A., Kuhnert, J., Aschenbrenner, L. & Giffhorn, U. (2015) Finite pointset method for the simulation of a vehicle travelling through a body of water. *Lecture Notes in Comput. Sci. Eng.* **100**, 205–221.

[19] Kermanpur, A., Mahmoudi, S. & Hajipour, A. (2008) Numerical simulation of metal flow and solidification in the multi-cavity casting moulds of automotive components. *J. Mater. Process. Technol.* **206**(1–3), 62–68.

[20] Kopysov, S. P., Tonkov, L. E., Chernova, A. A. & Sarmakeeva, A. S. (2015) Modelling of the incompressible liquid flow interaction with barriers using VOF and SPH methods. *J. Vestn. Udmurtsk. Univ. Mat. Mekh. Komp. Nauki* **25**(3), 405–420.

[21] Kuhnert, J. (1999) *General smoothed particle hydrodynamics.* PhD. Thesis. Technische Universität Kaiserslautern, Germany.

[22] Kuhnert, J. (2003) An upwind finite pointset method (FPM) for compressible euler and navier-stokes equations. *Lecture Notes in Comput. Sci. Eng.* **26**, 239–249.

[23] Kuhnert, J. & Ostermann, I. (2014) The finite pointset method (FPM) and an application in soil mechanics. *Lecture Notes in Earth Syst. Sci.*, 815–818.

[24] Wawreńczuk, A., Kuhnert, J. & Siedow, N. (2007) FPM computations of glass cooling with radiation. *Comput. Methods Appl. Mech. Engrg.* **196**(45), 4656–4671.

[25] Lewis, R. W. & Ravindran, K. (2000) Finite element simulation of metal casting. *Int. J. Numer. Methods Eng.* **47**(1–3), 29–59.

[26] Liu, G. R. (2009) *Mesh Free Methods: Moving Beyond the Finite Element Method*, 2nd ed., CRC Press, USA.

[27] Lucy, L. B. (1977) A numerical approach to the testing of the fission hypothesis. *Astron. J.* **82**, 1013–1024.

[28] Nguyen, V. P., Rabczuk, T., Bordas, S. & Duflot, M. (2008) Meshless methods: A review and computer implementation aspects. *Math. Comput. Simul.* **79**(3), 763–813.

[29] Oñate, E., Idelsohn, S., Zienkiewics, O. & Taylor, R. (1996) A finite point method in computational mechanics. Applications to convective transport and fluid flow. *Int. J. Numer. Methods Eng.* **39**(22), 3839–3866.

[30] Oñate, E., Idelsohn, S., Zienkiewics, O., Taylor, R. & Sacco, S. (1996) A stabilized finite point method for analysis of fluid mechanics problems. *Comput. Methdos Appl. Mech. Engrg.* **139**(1–4), 315–346.

[31] Parka, J. S., Kimb, S. M., Kimc, M. S. & Lee, W. I. (2005) Finite element analysis of flow and heat transfer with moving free surface using fixed grid system. *Int. J. Comput. Fluid. Dyn.* **19**(3), 263–276.

[32] Perminov, V. A., Rein, T. S. & Karabtcev, S. N. (2015) NEM and MFEM simulation of interaction between time-dependent waves and obstacles. *IOP Conf. Series: Mater. Sci. Eng.* **81**(1), 012099.

[33] Ramana, T. V. (1996) Metal Casting: Principles and Practice, 1St ed., New Age International (P) Ltd, India.

[34] Ren, J., Ouyang, J., Jiang, T. & Li, Q. (2011) Simulation of complex filling process based on the generalized Newtonian fuid model using a corrected SPH scheme. *Comput. Mech.* **49**(5), 643–665.

[35] Schmid, M. & Klein, F. (1995) Fluid flow in die cavities – experimental and numerical simulation. In: *NADCA* 18. *International Die Casting Congress and Exposition*, 93–99.

[36] Suchde, P., Kuhnert, J., Schröder, S. & Klar, A. (2017) A flux conserving meshfree method for conservation laws. *Int. J. Numer. Methods Eng.*.

[37] Tiwari, S. & Kuhnert, J. (2001) Grid free method for solving the Poisson equation. *Berichte des Fraunhofer ITWM* **25**.

[38] Tiwari, S. & Kuhnert, J. (2002) Finite pointset method based on the projection method for simulations of the incompressible Navier–Stokes equations. *Springer LNCSE: Meshfree methods for Partial Differential Equations* **26**, 373–387.

[39] Tiwari, S. & Kuhnert, J. (2003) Particle method for simulation of free surface flows. In: *Hyperbolic Problems: Theory, Numerics, Applications*, Springer, Berlin, Heidelberg, 889–898.

[40] Tiwari, S., Antonov, S., Hietel, D., Kuhnert, J., Olawsky, F. & Wegener, R. (2006) A meshfree method for simulations of interactions between fluids and flexible structures. *Lecture Notes in Comput. Sci. Eng.* **57**, 249–264.

[41] Tiwari, S. & Kuhnert, J. (2007) Modeling of two-phase flows with surface tension by finite pointset method (FPM). *J. Comput. Appl. Math.* **203**(2), 376–386.

[42] Tiwari, S. & Kuhnert, J. (2002) A meshfree method for incompressible fluid flows with incorporated surface tension. *Revue Europeenne des Elements* **11**(7–8), 965–987.