Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 217 (2006) 680–692
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Computing a null divergence velocity field using smoothed particle hydrodynamics

F. Colin [a], R. Egli [b,*], F.Y. Lin [b]

[a] Department of Mathematics and Computer Science, Laurentian University, 935 Ramsey Lake Road, Sudbury, Ont., Canada P3E 2C6
[b] Département d'informatique, Université de Sherbrooke, Sherbrooke, Que., Canada J1K 2R1

Received 9 June 2005; received in revised form 16 November 2005; accepted 13 January 2006
Available online 24 February 2006

Abstract

A new use of smoothed particle hydrodynamics (SPH) in fluid simulation is presented: an algorithm solving the Helmholtz–Hodge decomposition using SPH in order to find the null divergence velocity field for incompressible flow simulation. Accordingly, a new version of the Laplacian for a vector field is proposed here. In order to improve the accuracy of the SPH technique, the paper also presents some test problems for understanding the limitations of different kinds of gradient and Laplacian approximation formulas.

PACS: 47.11+j; 02.70.Ns

MSC: 65N22; 76M28

Keywords: Smoothed particle hydrodynamics (SPH); Incompressible flows; Helmholtz–Hodge decomposition; Meshfree particle method

1. Introduction

Smoothed particle hydrodynamics (SPH) was first invented simultaneously by Lucy [13] and Gingold and Monaghan [7] to solve astrophysical problems. It has since been used to study a range of astrophysical topics such as galaxy formation, star formation, supernovas, solar system formation, tidal disruption of stars by massive black holes, stellar collisions, etc.

SPH is a meshfree method (MFM) based on the Lagrangian description. It is not the first generation of MFM: many other types exist, such as the finite point method, the diffuse element method, etc. But its particular combination of adaptivity and Lagrangian and particle nature makes SPH very popular among MFMs. Today, the SPH method is being used in many areas, such as astronomy, computational fluids, and even solid mechanics. For decades, it has been expected to be better than the other traditional numerical

* Corresponding author. Tel.: +1 819 821 8000; fax: +1 819 821 8200.
E-mail address: Richard.Egli@USherbrooke.ca (R. Egli).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.01.021

methods for many applications in computational fluid dynamics (CFD). Because of its wide range of applications in different areas, SPH has enjoyed considerable progress in terms of accuracy, stability and extensibility since it was first proposed.

SPH can be used for various purposes. Monaghan [14] explains more clearly the properties and criteria of the formulations used in SPH and gives examples of applications in astronomy and fluid dynamics. We can see that it can also be used to describe deformable bodies, as in the work of Desbrun and Gascuel [5]. Also, different smoothing kernels and approximation methods for field functions in governing Navier–Stokes equations can be used in SPH methods. For example, Monaghan [14] and Desbrun and Gascuel [5] used the same damping function for diffusion force and suggested a symmetric formula for pressure force. Other authors, such as Yoon et al. [20], Premože et al. [17] and Koshizuka et al. [10], have suggested another smoothing kernel for density estimation and used the deterministic particle interaction models for the gradient and Laplacian terms in the Navier–Stokes equations.

There are various applications of SPH focused on fluid dynamics-related areas, for example incompressible flow, elastic flow, multi-phase flow, supersonic flow, shock simulation, heat transfer, mass flow, explosion phenomena, metal forming fractures, etc. [12,14]. One crucial step in simulating an incompressible fluid is ensuring that the continuity equation is satisfied. One way of doing this is to represent the incompressibility of the fluid by the constancy of its density. In an interesting series of articles by Koshizuka and his co-workers [10,11,20], the authors keep the particle number density constant by introducing a corrective pressure obtained by solving a Poisson equation. Idelsohn et al. [9] subsequently generalized this procedure to Lagrangian formulations. Other authors (for instance Morris et al. [15]) obtain the pressure mentioned above from what are called state equations. It should also be noted that Foster and Metaxas [6] adopted a similar approach, but within the framework of grid-based methods. A second way of satisfying the continuity equation is by ensuring that the divergence of the velocity fields remains null. This is the approach favored, for instance, by Cummins and Rudman [4], who developed an SPH projection operator based on the solution of a pressure Poisson equation, thereby allowing them to obtain a null divergence velocity field. In the context of grid-based methods, Stam [18] used the Helmholtz–Hodge decomposition, again with the goal of achieving a null divergence vector field. Here, we present an SPH solution of the Helmholtz–Hodge decomposition for a null divergence velocity field in an incompressible fluid simulation. At the same time, we propose a new version of the Laplacian formula for smoothing kernels instead of the basic one. Some advantages and disadvantages of different gradient and Laplacian approximation formulas used in SPH are discussed.

2. SPH fundamentals

SPH is an approximation method for particle systems. As stated in [14], we can estimate a function f at position $\mathbf{r}$ by using smoothing kernels W to approximate in a local neighborhood within distance h as follows:

$$f(\mathbf{r}) = \sum_{j=1}^{n} m_j \frac{f_j}{\rho_j} W(\mathbf{r} - \mathbf{r_j}, h), \tag{1}$$

where m_j is the mass, $\mathbf{r_j}$ is the position, ρ_j is the density and f_j is the quantity f for neighbor particle j, respectively. Here, n is the number of neighbor particles with $|\mathbf{r} - \mathbf{r_j}| \leqslant h$. To shorten the equation, we omit the n in all summations. When $\mathbf{r} = \mathbf{r_i}$, $f(\mathbf{r})$ is denoted by f_i.

Thus, we estimate the density ρ_i for a particle i at location $\mathbf{r_i}$ by

$$\rho_i = \sum m_j \frac{\rho_j}{\rho_j} W(\mathbf{r_i} - \mathbf{r_j}, h), \tag{2}$$

that is,

$$\rho_i = \sum m_j W(\mathbf{r_i} - \mathbf{r_j}, h), \tag{3}$$

where j denotes the index of the neighbor particle (this notation is used throughout).

2.1. Choice of a good kernel

The first criterion in choosing a kernel to meet the requirements of SPH is that it must estimate the density of the particle system to maximum accuracy. The kernel should be smooth, symmetric [8,12] and satisfy the following equation:

$$int_{\Omega} W(\mathbf{r}, h)\mathrm{d}r = 1, \tag{4}$$

where W is the kernel function, h is the smoothing length and the integration is taken over all space. More generally, in order to ensure the consistency of the SPH approximations, as mentioned in [12, Chapter 3], a kernel must verify the following conditions:

$$\begin{cases} M_0 = \int_{\Omega} W(x - x', h)\mathrm{d}x' = 1, \\ M_1 = \int_{\Omega} (x - x') W(x - x', h)\mathrm{d}x' = 0, \\ M_2 = \int_{\Omega} (x - x')^2 W(x - x', h)\mathrm{d}x' = 0, \\ \qquad\qquad \vdots \\ M_n = \int_{\Omega} (x - x')^n W(x - x', h)\mathrm{d}x' = 0, \end{cases} \tag{5}$$

also

$$\begin{cases} W(x - x', h)|_s = 0, \\ W'(x - x', h)|_s = 0, \end{cases} \tag{6}$$

in order to ensure that the first two derivatives of the function can be exactly approximated to nth order accuracy. If only Eq. (5) is satisfied, it means that the function can be approximated to nth order accuracy. Since the highest derivative of the field function in the governing equations is of the second order, with the condition mentioned above, the 2D poly6 kernel chosen for use in this paper is

$$W_{\mathrm{poly6}}(\mathbf{r}, h) = \frac{4}{\pi h^8} \begin{cases} (h^2 - \mathbf{r}^2)^3, & 0 \leqslant \mathbf{r} \leqslant h, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The 3D expression of poly6 has been presented by Müller et al. [16]. It can be verified that the kernel poly6 fulfills Eq. (5) for at least M_0 and M_1 and also Eq. (6) which is the minimum requirement of consistency for considering second partial derivatives (see [12]).

2.2. Gradient and Laplacian

In SPH, the derivatives of a function f can be obtained by using the derivatives of the smoothing kernel. The following expressions are the gradient and Laplacian of the function f obtained in this way:

$$\nabla f_i = \sum_j m_j \left(\frac{f_j}{\rho_j} \right) \nabla W(\mathbf{r_i} - \mathbf{r_j}, h), \tag{8}$$

$$\nabla^2 f_i = \sum_j m_j \left(\frac{f_j}{\rho_j} \right) \nabla^2 W(\mathbf{r_i} - \mathbf{r_j}, h). \tag{9}$$

In this paper, we call these the basic gradient approximation formula (BGAF) Eq. (8) and the basic Laplacian approximation formula Eq. (9). Eq. (9) can be validated by Green's formula when the kernel W is of class C^2 and $\nabla W = 0$ on $\partial\Omega$.

2.2.1. Choosing the gradient formula

There are other ways than BGAF to compute the gradient value for a function f.

The first one is derived by using the derivative of a product [14]. We have

$$\nabla(\rho f) = \rho\nabla f + f\nabla\rho, \tag{10}$$

which can be rewritten as

$$\rho\nabla f = \nabla(\rho f) - f\nabla\rho. \tag{11}$$

This yields the following expression:

$$\rho_i\nabla f_i = \sum_j m_j(f_j - f_i)\nabla W(\mathbf{r_i} - \mathbf{r_j}, h). \tag{12}$$

Thus the gradient of the function f can be approximated as

$$\nabla f_i = \left(\frac{1}{\rho_i}\right)\sum_j m_j(f_j - f_i)\nabla W(\mathbf{r_i} - \mathbf{r_j}, h). \tag{13}$$

In this paper, we can call Eq. (13) the difference gradient approximation formula (DGAF).

To deal with symmetric forces in order to fulfill Newton's second law with two particles, some authors [5] consider another form of gradient by using the derivative of a quotient function. The derivative of a quotient of functions is given by

$$\nabla\left(\frac{f}{\rho}\right) = \frac{\rho\nabla f - f\nabla\rho}{\rho^2}, \tag{14}$$

i.e.

$$\frac{\nabla f}{\rho} = \nabla\left(\frac{f}{\rho}\right) + \frac{f\nabla\rho}{\rho^2}. \tag{15}$$

We can obtain the formula

$$\frac{\nabla f_i}{\rho_i} = \sum_j m_j\left(\frac{f_i}{\rho_i^2} + \frac{f_j}{\rho_j^2}\right)\nabla W(\mathbf{r_i} - \mathbf{r_j}, h). \tag{16}$$

That is

$$\nabla f_i = \rho_i\sum_j m_j\left(\frac{f_i}{\rho_i^2} + \frac{f_j}{\rho_j^2}\right)\nabla W(\mathbf{r_i} - \mathbf{r_j}, h). \tag{17}$$

In this paper, we call Eq. (17) the symmetric gradient approximation formula (SGAF).

In order to understand the differences in the gradient approximation given by Eqs. (8), (13) and (17), we used the three formulas to approximate the gradient value for the same function f. Obviously, the function f and the function $f + C$ (where C is a constant) have the same gradient. But when approximating the gradient, as shown below, the results showed that only the difference gradient formula is insensitive to C.

The first example we consider is a plane $f(x, y) = C$. The gradient of this plane should be 0. In our experiments, however, the three different forms of gradient approximation behave differently, as shown in Fig. 1. DGAF obtained the most accurate results (in fact, exact results in this case), while the other two formulas had much greater errors. We also observed that SGAF had a larger error than BGAF. The error increased as the value of the constant C increased, except for DGAF.

Our second example is a plane $f(x, y) = Ax + By + C$, where A, B and C are constants, meaning that the gradient for this plane is the vector (A, B). But our experiments yields the same conclusions as in the first example. This is shown in Fig. 2.

We also tried the example of $f(x, y) = Ax^2 + By^2 + C$, with the same results. We therefore conclude that DGAF is more accurate than the others in the context of translation and so we chose it as the gradient formula to use in the SPH Helmholtz–Hodge decomposition (Section 3.2). Note that in these tests for gradient approximations at particles, about ten neighbor particles were used.

2.2.2. New version of the Laplacian

As described in the previous section, we suggested DGAF as the gradient to be used in Helmholtz–Hodge decomposition. Now, an appropriate version of the Laplacian to accommodate DGAF is needed. By applying the rule of the product twice, we obtain

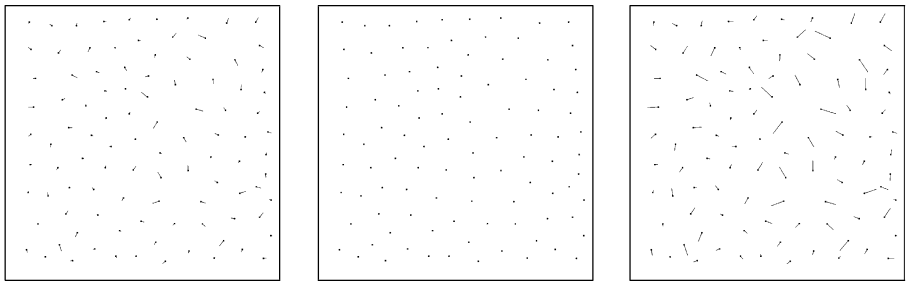

Fig. 1. Gradient approximations for the function $f(x, y) = 1$: (a) basic gradient approximation formula (BGAF); (b) difference gradient approximation formula (DGAF); (c) symmetric gradient approximation formula (SGAF).

$$\nabla^2(\rho f) = f(\nabla^2\rho) + \rho(\nabla^2 f) + 2\nabla f \cdot \nabla\rho, \tag{18}$$

i.e.

$$\rho(\nabla^2 f) = \nabla^2(\rho f) - f(\nabla^2\rho) - 2\nabla f \cdot \nabla\rho. \tag{19}$$

The corresponding SPH expression below is obtained by straightforward calculations using Eq. (12).

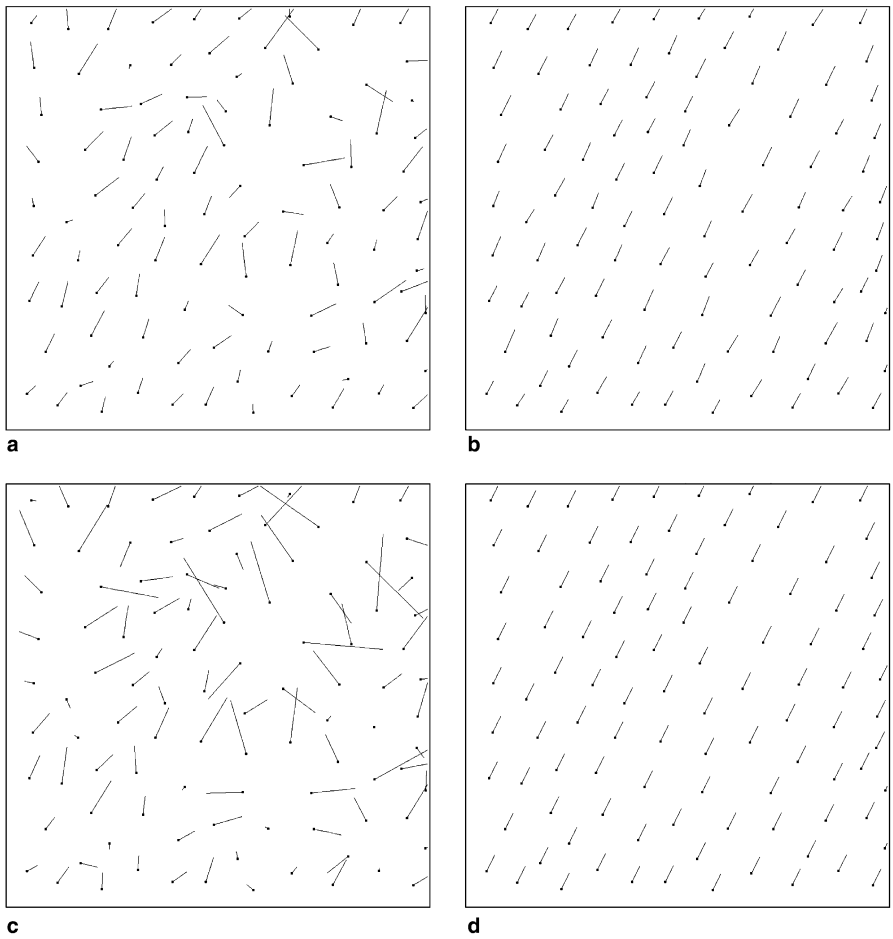

Fig. 2. Gradient approximations for the function $f(x, y) = x + 2y + 1$: (a) basic gradient approximation formula (BGAF); (b) difference gradient approximation formula (DGAF); (c) symmetric gradient approximation formula (SGAF); (d) correct gradient value.

$$\rho_i \nabla^2 f_i = \sum_j m_j (f_j - f_i) \left(\nabla^2 W(\mathbf{r_i} - \mathbf{r_j}, h) - \frac{2}{\rho_i} \nabla W(\mathbf{r_i} - \mathbf{r_j}, h) \cdot \nabla \rho_i \right). \tag{20}$$

Thus the corresponding Laplacian approximation expression for DGAF is as follows:

$$\nabla^2 f_i = \frac{1}{\rho_i} \sum_j m_j (f_j - f_i) \left(\nabla^2 W(\mathbf{r_i} - \mathbf{r_j}, h) - \frac{2}{\rho_i} \nabla W(\mathbf{r_i} - \mathbf{r_j}, h) \cdot \nabla \rho_i \right). \tag{21}$$

Like DGAF, this version of the Laplacian gives the same result for f and $f + C$, where C is a constant.

Compared to the basic version given by Eq. (9), Eq. (21) gives a more accurate approximation for the Laplacian. Fig. 3 below shows the experimental data for the Laplacian approximation for the function $f(x, y) = x^2 + y^2$.

3. A null divergence velocity field

3.1. The Navier–Stokes equations

The classical Navier–Stokes equations for incompressible flows are the following:

$$\nabla \cdot \mathbf{v} = 0, \tag{22}$$

$$\rho \left(\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} \right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}. \tag{23}$$

The first equation ensures the conservation of mass while the second guarantees the conservation of momentum. Note that $\frac{\mathrm{D}}{\mathrm{D}t}$ is the convective derivative (also called the Lagrangian derivative or substantive derivative), $\mathbf{v}$ is the fluid velocity, ρ is the fluid density, and $\mathbf{g}$ is the gravitational acceleration. The term ∇p refers to the pressure gradient, μ is the dynamic viscosity coefficient where kinematic viscosity is defined by $\upsilon \equiv \frac{\mu}{\rho}$, and the term $\nabla^2 \mathbf{v}$ is the vector Laplacian.

Typically, after applying Eq. (23), the Helmholtz–Hodge decomposition is used to obtain a null divergence velocity field Eq. (22).

3.2. The Helmholtz–Hodge decomposition

The Helmholtz–Hodge decomposition can be described as follows: given a domain Ω and its boundary $\partial\Omega$, where $\boldsymbol{\eta}$ is the outward normal vector to the boundary and q is a scalar field, there is a unique decomposition of a C^2 vector field $\mathbf{w}$,

$$\mathbf{w} = \mathbf{v} + \nabla q, \tag{24}$$

where the divergence-free vector field is

$$\mathbf{v} = \mathbf{w} - \nabla q. \tag{25}$$

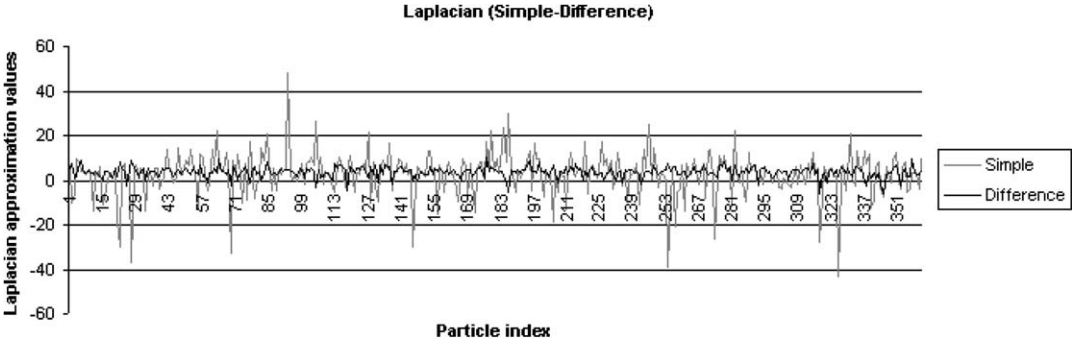

Fig. 3. Laplacian approximation values given by Eqs. (9) and (21) for the function $f(x, y) = x^2 + y^2$.

This means that the divergence of vector field $\mathbf{v}$ should be 0, i.e.

$$\nabla \cdot \mathbf{v} = 0. \tag{26}$$

Also, we have

$$\nabla \cdot \mathbf{w} = \nabla \cdot (\mathbf{v} + \nabla q) = \nabla \cdot \mathbf{v} + \nabla^2 q. \tag{27}$$

Combining Eqs. (26) and (27), we get

$$\nabla \cdot \mathbf{w} = \nabla^2 q. \tag{28}$$

When considering the boundary, one needs to solve the partial differential equation (28) with the Neumann boundary condition:

$$\begin{cases} \nabla^2 q = \nabla \cdot \mathbf{w} & \text{in } \Omega, \\ \frac{\partial q}{\partial \boldsymbol{\eta}} = 0 & \text{on } \partial\Omega. \end{cases} \tag{29}$$

Fortunately, the SPH version of this problem is very simple. It is interesting to note that the solution q is not unique, but the difference between any two solutions is a constant [19].

3.3. A system for inside particles

In order to solve system Eq. (28) in an SPH way, we use the difference version of divergence [14] and the Laplacian equation (21). For an inside particle i, we have

$$\begin{cases} (\nabla \cdot \mathbf{w})_i = \left(\frac{1}{\rho_i}\right) \sum_j m_j (\mathbf{w}_j - \mathbf{w}_i) \cdot \nabla W(\mathbf{r}_i - \mathbf{r}_j, h), \\ (\nabla^2 q)_i = \left(\frac{1}{\rho_i}\right) \sum_j m_j ((q_i - q_j) \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) - \frac{2}{\rho_i} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \cdot \nabla \rho_i), \quad j \neq i. \end{cases} \tag{30}$$

This process gives rise to the following linear system of equations:

$$\sum_j a_{ij} q_j = b_i, \tag{31}$$

where for each i,

$$\begin{cases} a_{ij} = \left(\frac{m_j}{\rho_i}\right)(\nabla^2 W(\mathbf{r_i} - \mathbf{r_j}, h) - 2/\rho_i \nabla W(\mathbf{r_i} - \mathbf{r_j}, h) \cdot \nabla \rho_i), \quad j \neq i, \\ a_{ii} = -\sum_{j \neq i} a_{ij}, \end{cases} \tag{32}$$

$$b_i = \left(\frac{1}{\rho_i}\right) \sum_j m_j (\mathbf{w_j} - \mathbf{w_i}) \cdot \nabla W(\mathbf{r_i} - \mathbf{r_j}, h). \tag{33}$$

3.4. Additional conditions for boundary particles

For boundary particles, we need to solve the following boundary condition:

$$\frac{\partial q}{\partial \boldsymbol{\eta}} = 0 \quad \text{on } \partial\Omega, \tag{34}$$

where $\boldsymbol{\eta}$ is the outward normal unit vector. Assuming that q is known for each boundary particle, according to the above condition, a SPH version of the system for boundary particles is

$$(\nabla q)_i \cdot \eta_i = 1/\rho_i \sum_j m_j (q_j - q_i)(\nabla W(\mathbf{r_i} - \mathbf{r_j}, h) \cdot \eta_i). \tag{35}$$

Thus,

$$1/\rho_i \sum_j m_j (q_j - q_i)(\nabla W(\mathbf{r_i} - \mathbf{r_j}, h) \cdot \eta_i) = 0. \tag{36}$$

By setting

$$\begin{cases} a_{ij} = \left(\frac{m_j}{\rho_i}\right)(\nabla W(\mathbf{r_i} - \mathbf{r_j}, h), h) \cdot \eta_i, \quad j \neq i, \\ a_{ii} = -\sum_{j \neq i} a_{ij}, \end{cases} \tag{37}$$

we obtain the same system as in Eq. (31)

$$\sum_j a_{ij} q_j = b_i \quad \text{for each } i, \tag{38}$$

where $b_i = 0$ for boundary particles.

Remark that the definition of a_{ii} in Sections 3.3 and 3.4 implies that the unit vector is a solution of the homogeneous system. Hence the difference between two solutions q_1 and q_2 is a scalar multiple of the unit vector ($k\mathbf{1}$ where k is a scalar). This is the discrete counterpart of the non-unicity of the solution we mentioned in Section 3.2. It justifies the use of the difference versions of the gradient (DGAF), Laplacian and divergence formulas, which are robust with respect to translation.

3.5. Solving the corresponding linear system

As noted in Sections 3.3 and 3.4, we can form a larger linear system $Aq = b$ which includes both the inside and outside particles. The upper part of squared matrix A corresponds to Eq. (32) (including the boundary particles), while the lower part corresponds to Eq. (37) (including the inside particles). To solve this linear system, we use the conjugate gradient method on $A^t Aq = A^t b$.

3.6. Boundary particle setup

In the simulation, it is preferable to have a suitable boundary that can increase the accuracy of estimation for density and other physical quantities. In setting the boundary particles, we need to obtain the approximation density ρ according to the distribution of the initial particles. (In this paper, the ρ has been taken as the density which assumed the particles are distributed evenly.) The ideal boundary width has the same value as smoothing length h. According to these two parameters, the boundary we set would have correct density ρ and a sufficient width to ensure correct distribution of the inside particles without crossing the boundary.

3.7. Sample algorithm

The computation of a null divergence velocity field can be used in an algorithm for simulating incompressible fluids as follows:

(a) Initialize particles (positions $\mathbf{p}$, velocities $\mathbf{v}$)
(b) Compute densities ρ
(c) Apply the momentum Navier–Stokes equation (23) using SPH

$$\mathbf{w} \leftarrow \mathbf{v} + \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t}\Delta t \tag{39}$$

(d) Compute the null divergence velocity field using the Helmholtz–Hodge decomposition with the Neumann boundary condition
   (I) Solve the system

$$Aq = b \tag{40}$$

   (II) Compute ∇q

$$\nabla q_i = \left(\frac{1}{\rho_i}\right) \sum_j m_j (q_j - q_i) \nabla W(\mathbf{r_i} - \mathbf{r_j}, h) \tag{41}$$

(e) Update particle velocities $\mathbf{v} \leftarrow \mathbf{w} - \nabla q$
(f) Update particle positions $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{v}\Delta t$
(g) Go to step b

4. Results

Given that we are focusing exclusively on the problem of finding a null divergence velocity field using the Helmholtz–Hodge decomposition, no particle displacement is entailed in the process. Consequently, incompressibility measures such as the one presented by Cummins and Rudman [4] do not apply here. However,

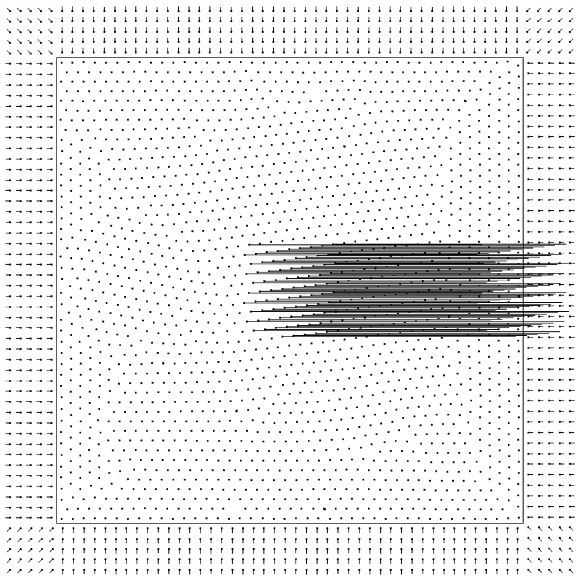

Fig. 4. Initial velocity field of example 1.

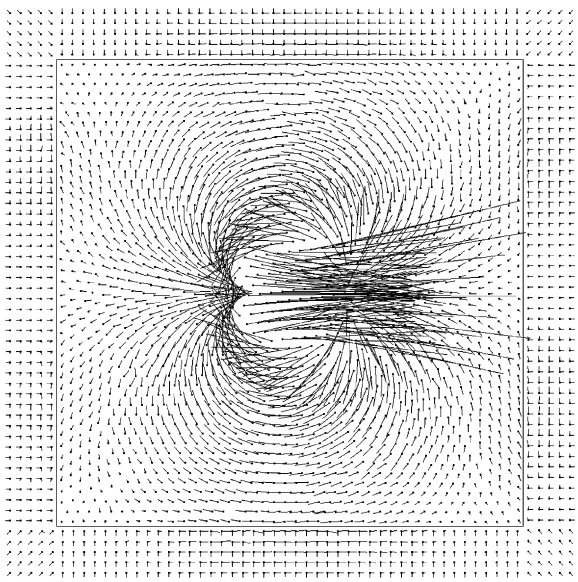

Fig. 5. Null divergence velocity field of example 1.

if the context for our method is taken into consideration, we believe the SPH formulation for the divergence of the resultant velocity field is the most appropriate measure of the fluid's incompressibility. We therefore consider the SPH divergence of the vector field for each inside particle (first equation of Eq. (30)), and then, for each of the following examples, we present the maximum divergence value and the sum of squared divergence. It should also be recalled that the Helmholtz–Hodge decomposition of a vector field is obtained by solving the discrete version of a Poisson equation, which is simply a system of linear equations. When the solution of this

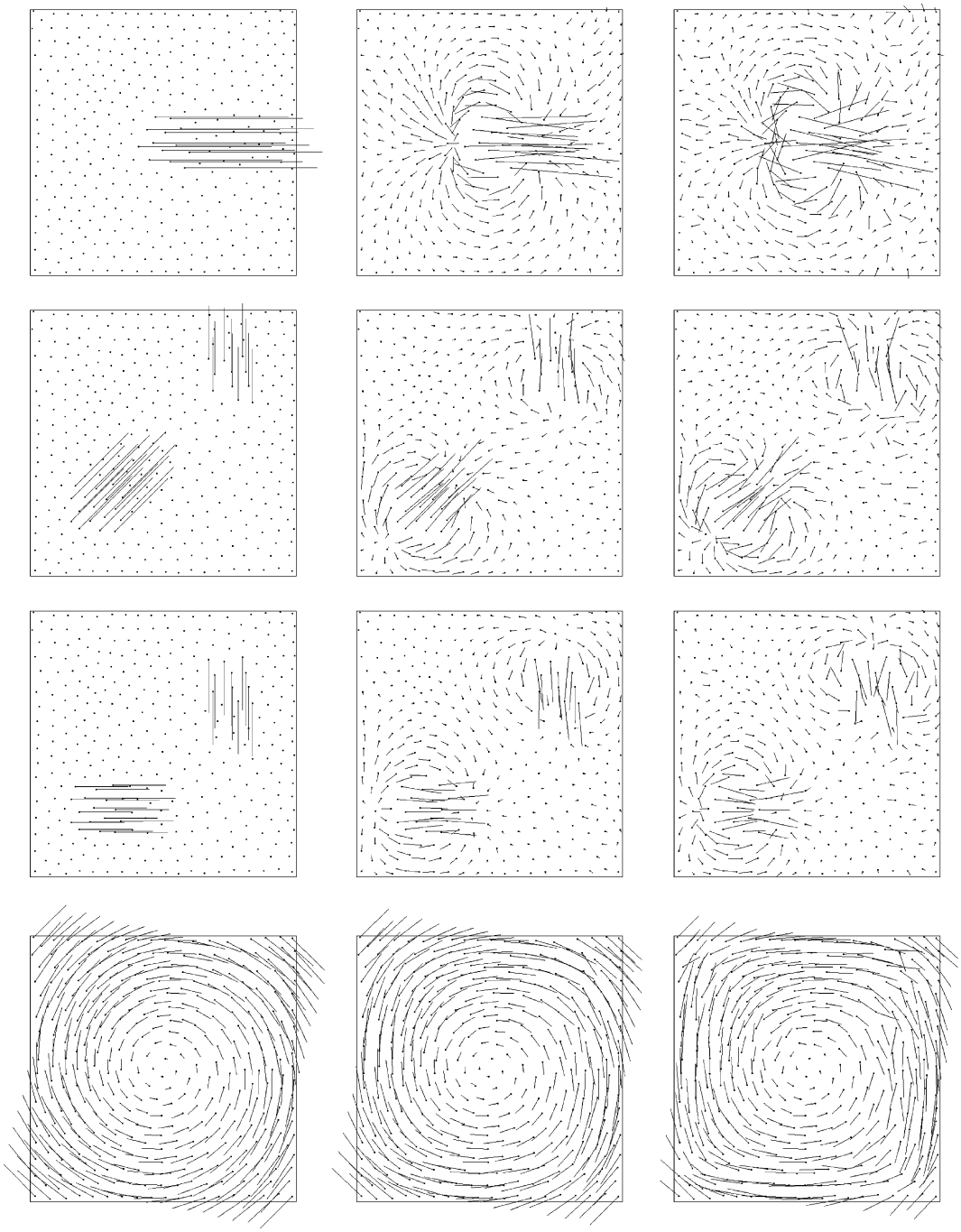

Fig. 6. The rows correspond to examples 2, 3, 4 and 5. The first column shows the initial problems; the second column, the state after the first iteration in computing the null divergence; and the last column, the state after the fifth iteration.

system is precise, if the divergence of the vector field after correction is still not null, the error arises from a difference between the SPH formulations for the Laplacian and the gradient of the divergence; such differences can occur because we are dealing with approximations. However, successive correction by reiteration of our procedure allows us to take this difference into account and obtain a vector field with the smallest possible divergence.

Five examples are presented. The initial condition of the testing particle system is shown in Fig. 4 for example 1; the particles are represented by dots. The inside particles and boundary particles are separated by the rectangle shown in the figure. The lines departing from boundary particles indicate the normal vectors η_i perpendicular to the boundary lines (scale of 1/100). At the corners we take the inward vector parallel to the diagonal and of same magnitude as the others on the boundary. All particles have initial velocity (0, 0), except that those in the rectangle of 0.2×0.2 in the middle have initial velocity (0.5, 0). Line segments are traced from the particles using the velocity.

The solution q was found successfully by solving the Helmholtz–Hodge decomposition with the Neumann boundary condition. We apply DGAF on q and update the velocity field $\mathbf{w}$ with Eq. (25) to obtain a new velocity field $\mathbf{v}$ which exhibits null divergence within the precision of SPH. The results shown in Fig. 5 were obtained using 2000 inside particles; the smoothing length of the kernel is 1/9. The lines perpendicular to η_i are the ∇q for the boundary particles; those departing from inside particles are the traces of velocity segments for each particle. As we can see, all the traces for the particles are well corrected and the result is convincingly good.

As shown in Fig. 6, four more examples were tested with the same conditions as example 1 (Figs. 4 and 5) but with 361 inside particles and different smoothing length for the kernel (1/5). Note that this is a small number of particles for an SPH simulation. The first row (example 2) in Fig. 6 has the initial velocity of (0.5, 0) at the center; the second row (example 3) has the initial velocity of (0.2, 0.2) at the lower left corner and (0, 0.2) at the upper right corner; the third row (example 4) has the initial velocity of (0.2, 0) at the lower left corner and (0, −0.2) at the upper right corner; the last row (example 5) has the initial velocity of $(0.25 * (\text{position}_Y - 0.5), 0.25 * (0.5 - \text{position}_X))$ [4] for all inside particles. The figure shows three iteration states, where the first column shows the initial state with initial velocities, the second column shows the result after the first iteration of the null divergence algorithm and the last column shows the result after the fifth iteration of the null divergence algorithm. Furthermore, Table 1 and Fig. 7 show how application of the null divergence algorithm reduces the divergence after each iteration. For each example, we show the maximum divergence values and the sums of squared divergence for the particles from the initial state to the fifth iteration of the algorithm. Maximum divergence is calculated by using Eq. (33) for each inside particle and selecting the maximum value of the divergences thus obtained. Sum of squared divergence is calculated by summing up the values of the squared divergence for each inside particle.

We should add that example 5 is drawn from the article of Cummins and Rudman [4], and that while direct comparison with these authors' results is impossible, both Table 1 and Fig. 7 demonstrate excellent practical results.

Table 1
Results for the five examples after successive corrections

Example	Value type	Init	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
1	Max Div	5.32	1.78	0.93	0.63	0.53	0.46
	Tot Div	1250.48	108.01	33.95	17.93	11.13	7.46
2	Max Div	1.43	0.59	0.34	0.23	0.17	0.14
	Tot Div	45.66	7.44	2.10	0.85	0.48	0.37
3	Max Div	1.06	0.49	0.28	0.17	0.14	0.13
	Tot Div	25.97	4.04	1.25	0.59	0.38	0.31
4	Max Div	0.50	0.32	0.23	0.20	0.22	0.22
	Tot Div	8.42	2.71	1.33	0.83	0.59	0.48
5	Max Div	2.81	1.37	0.76	0.52	0.39	0.32
	Tot Div	82.61	15.47	5.19	2.45	1.53	1.18

Maximum divergence (Max Div) and sum of squared divergence (Tot Div).

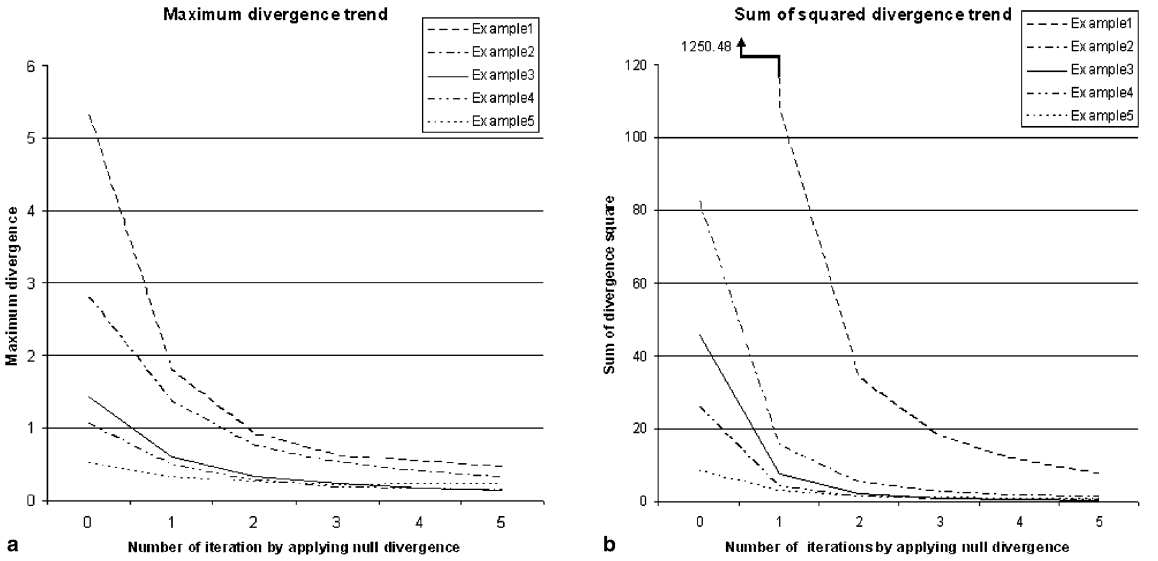

Fig. 7. Maximum divergence and sum of squared divergence: trends over five iterations: (a) maximum divergence; (b) sum of squared divergence.

5. Conclusion

One advantage of the method proposed here is that it will be possible in future work to consider an incompressible fluid of variable density, as Almgren et al. [1], for instance, have done. Indeed, equations to model such fluids require a null divergence velocity field despite the fact that the density is not constant, making it impossible to apply the methods favored by Koshizuka and his co-workers [10,11,20] or by Morris et al. [15]. To our knowledge, no SPH approach yet exists for problems of this kind.

SPH is a meshfree particle method which is attractive for its adaptivity. Although it may produce larger error for some specific problems than some other methods, its accuracy can be improved by various correction schemes [12]. As presented in this paper in Sections 2.1 and 2.2, we can use a suitable kernel, gradient approximation formula and Laplacian approximation formula to help us to obtain more accurate simulation results, as well as setting the boundary particles to avoid boundary effects.

In this study, we have achieved a different way to obtain a null divergence velocity field by using a new algorithm – a pure SPH solution of the Helmholtz–Hodge decomposition. The discussion for gradient approximation formulas will also supply a valuable reference for subsequent work.

Finally, an upcoming stage in the development of our technique would be to incorporate the corrections to the SPH method presented by Bonet and Kulasegaram in [3]. We should mention that there are many other possible corrections; among other work on this topic, we refer the reader to the paper by Belytschko et al. [2] and the references cited there.

Acknowledgements

The authors are very grateful to the reviewers for their comments and remarks, which have resulted in significant improvements to this article. The research of the first author was partially supported by a start-up grant from Laurentian University. The research of the second author was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada. We also thank Professor J.P. Dussault for his helpful suggestions on numerical aspects of this research.

References

[1] A.S. Almgren, J.B. Bell, P. Collella, et al., A conservative adaptative projection method for the variable density incompressible Navier–Stokes equations, J. Comput. Phys. 142 (1998) 1–46.

[2] T. Belytschko, Y. Krongauz, J. Dolbow, et al., On the completeness of meshfree particle methods, Int. J. Numer. Methods Engrg. 43 (1998) 785–819.
[3] J. Bonet, S. Kulasegaram, A simplified approach to enhance the performance of smooth particle hydrodynamics methods, Appl. Math. Comput. 126 (2002) 133–155.
[4] S.J. Cummins, M. Rudman, An SPH projection method, J. Comput. Phys. 152 (1999) 584–607.
[5] M. Desbrun, M.P. Gascuel, Smoothed particles: a new paradigm for animating highly deformable bodies, Proceedings of Eurographics Workshop on Animation and Simulation (1996) 61–76.
[6] N. Foster, D. Metaxas, Realistic animation of liquids, Graphical Models Image Process. 58 (5) (1996) 471–483.
[7] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Monthly Notices Roy. Astronom. Soc. 181 (1977) 375–398.
[8] R.A. Gingold, J.J. Monaghan, Kernel estimates as a basis for general particle methods in hydrodynamics, J. Comput. Phys. 46 (1982) 429–453.
[9] S.R. Idelsohn, M.A. Storti, E. Oñate, Lagrangian formulations to solve free surface incompressible inviscid fluid flows, Comput. Methods Appl. Mech. Engrg. 191 (2001) 583–593.
[10] S. Koshizuka, H. Tamako, Y. Oka, A particle method for incompressible viscous flow with fluid fragmentation, Comput. Fluid Dynam. J. 4 (1995) 29–46.
[11] S. Koshizuka, A. Nobe, Y. Oka, Numerical analysis of breaking waves using the moving particle semi-implicit method, Int. J. Numer. Methods Fluids 26 (1998) 751–769.
[12] G.R. Liu, M.B. Liu, Smoothed Particle Hydrodynamics – A Meshfree Particle Method, World Scientific, Singapore, 2003.
[13] L.B. Lucy, A numerical approach to the testing of the fission hypothesis, Astron. J. 82 (1977) 1013–1024.
[14] J.J. Monaghan, Smoothed particle hydrodynamics, Annu. Rev. Astron. Astrophys. (1992) 543–574.
[15] J.P. Morris, P.J. Fox, Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, J. Comput. Phys. 136 (1997) 214–226.
[16] M. Muller, D. Charypar, M. Gross, Particle-based fluid simulation for interactive application, in: EuroGraphics/SIGGRAPH Symposium on Computer Animation, 2003, pp. 154–159.
[17] S. Premože, T. Tasdizen, J. Bigler, A. Lefohn, R.T. Whitaker, Particle-based simulation of fluids, Eurographics, Computer Graphics forum (2003) 401–410.
[18] J. Stam, Stable fluids, in: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 1999, pp. 121–128.
[19] F. Treves, Basic Linear Partial Differential Equations, Academic Press, New York, 1975.
[20] H.Y. Yoon, S. Koshizuka, Y. Oka, Particle-gridless hybrid method for incompressible flows, Int. J. Numer. Methods Fluids 30 (1999) 407–424.