

# Towards pseudo-spectral incompressible smoothed particle hydrodynamics (ISPH)<sup>☆</sup>

G. Fourtakas<sup>\*</sup>, B.D. Rogers, A.M.A. Nasar

Department of Mechanical, Aerospace & Civil Engineering, Faculty of Science and Engineering, The University of Manchester, Manchester, M13 9PL, UK

## ARTICLE INFO

### Article history:

Received 12 February 2021  
Received in revised form 12 April 2021  
Accepted 6 May 2021  
Available online 17 May 2021

### Keywords:

Incompressible smoothed particle hydrodynamics (ISPH)  
Pseudo-spectral methods  
High-order schemes  
Fast Fourier transforms  
Pressure Poisson solver  
Fast Poisson solver

## ABSTRACT

In this paper a pseudo-spectral incompressible smoothed particle hydrodynamics (FFT-ISPH) solver is presented. While the solution of the linear system arising from the pressure Poisson equation in physical space using iterative solvers in incompressible SPH is a viable solution, it is widely accepted that the computational cost of solving the pressure Poisson equation by iterative solvers is excessive and affects negatively on the efficiency of the solver. The proposed scheme is an intermediate between a fully spectral and a standard incompressible SPH solver. Herein, the solution of the pressure Poisson equation is performed in spectral space whereas the discretisation and time integration are performed in physical space. This results in a second and higher-order scheme with the classical first-order projection and a third-order Runge-Kutta time integration scheme, respectively. A detailed performance analysis shows gains in computational cost of two orders of magnitude. Further, it is demonstrated that the solution of the pressure Poisson equation in spectral space is independent of the number of neighbouring nodes, in contrast to the iterative solver whose cost increases by a factor of three with smoothing length to particle size ratio. Periodic, bounded, and mixed boundary conditions test cases have been used to demonstrate the applicability, accuracy and robustness of the scheme with second and fourth-order convergence rates. The scheme opens an avenue for simulations of high-order incompressible flows in SPH where filtering operations in the frequency domain can be performed straightforwardly.

© 2021 Elsevier B.V. All rights reserved.

## 1. Introduction

Smoothed particle hydrodynamics (SPH) was originally developed for astrophysical problems in the late seventies [1]. In the last three decades it has been widely adopted in the field of computational fluid mechanics [2] as an alternative meshless discretisation scheme that uses the Lagrangian description of motion and in particular computational fluid dynamics for a variety of applications [3–6]. An early review of the scheme can be found in Monaghan [7]. More recent reviews (see, [2,8–10]) demonstrate the rapid development and maturity of the scheme over the last couple of decades. SPH is now considered to be a mainstream method for free-surface flows [11,12] and multi-phase flows [13] due to its meshless nature which allows for straightforward tracking of multiple continua. Early adopters of SPH used the weakly compressible variant (WCSPH) [7] which is fully explicit and relates the pressure field to the density field through an equation of state. The WCSPH method is widely adopted to date as it is easy to implement

and offers good stability characteristics. However, the explicit and weakly compressible nature of the scheme translates to small permissible time step size and non-physical acoustic waves which SPH practitioners attempt to filter from the solution [14–16]. On the other hand and since the work of Cummins and Rudman [17], the strictly incompressible SPH (ISPH) scheme is gaining increasing attention due to the smooth and accurate pressure field that do not require corrections or filtering, and the larger time steps.

In ISPH the pressure Poisson equation is assembled using the SPH discrete operators involving a local summation over neighbouring nodes within a support radius [17]. The most commonly used Laplacian operation is the so-called Morris operator [18] that results in a matrix for the linear system which must be inverted for the solution to be obtained. This is a common procedure applied to mesh-based [19] and mesh-less [20] schemes, although the computational cost in SPH is much larger than mesh-based schemes due to the number of neighbouring nodes in the locally compact support being larger than traditional or high-order compactly supported finite differences. This results in a large matrix which is more densely populated sparse matrix as demonstrated later in Section 5.2. Typically, the solution of the pressure Poisson equation occupies most of the computational time when using

<sup>☆</sup> The review of this paper was arranged by Prof. N.S. Scott.

<sup>\*</sup> Corresponding author.

E-mail address: georgios.fourtakas@manchester.ac.uk (G. Fourtakas).

iterative solvers [21] also demonstrated herein in Section 5.2. This makes the ISPH scheme impractical from a computational cost perspective for large engineering applications and is mainly restricted to academic flows. The aim of this work is to produce an efficient and high-order accurate solver where we can operate directly on different wave numbers or frequencies in the spectral domain.

The performance bottleneck of ISPH is the solution of the pressure Poisson equation. The SPH discrete operators and formalism is ideally suited to hardware acceleration in message parsing interface (MPI) [21] and co-processors such as graphics processing units [22] due to the  $n$ -body structure of the discrete operators which are mostly compute intensive. Herein, we propose a second and high-order pseudo-spectral strictly incompressible smoothed particle hydrodynamics methodology using fast Fourier transforms (FFT-ISPH) for the solution of the pressure Poisson equation that offers large gains in computational efficiency. Indeed, Fourier and Chebyshev discretisation schemes [23] have been used widely in the past for either spectral or pseudo-spectral compactly supported high-order schemes especially for large eddy simulation (LES) and direct numerical simulation (DNS) solvers [24] due to their computational efficiency [25]. Furthermore, solving the equations in the frequency domain rather than the physical domain offers unique advantages over conventional SPH approaches since operations that require filtering based on specific frequencies or wave numbers can be easily performed.

To the best of the authors' knowledge this is the first time such an approach has been demonstrated in SPH to this extent. Nevertheless, the authors acknowledge the work by Robinson [26] and Dehnen and Aly [27] who analysed the Fourier transforms of SPH kernels and Violeau, et al. [28] who used spectral analysis to examine the conditioning of the Laplacian operator for the pressure Poisson equation and demonstrated that the SPH Laplacian operator has a high condition number similar to finite differences for bounded and unbounded domains. In this work, and as proof of concept, we will use the Eulerian variant of SPH as demonstrated by Lind and Stansby [29] which is a high-order scheme directly related to the smoothing kernel characteristics with arbitrary even order of spatial accuracy. We reserve the non-equispaced variant in Eulerian-Lagrangian framework [30] for future work.

This paper is organised as follows; in Section 2 the general numerical framework used in this paper is presented for the continuous governing equations, the time stepping schemes employed in this work and the SPH formalism. This is followed by Section 3, which includes the spatial discretisation of the advection and viscous terms in SPH formalism, pressure gradients and a consideration of accuracy for SPH. After this overview, the solution methods of the pressure Poisson equation are presented followed by the software implementation and performance tests. An extensive list of test cases that includes an unbounded, bounded and mixed boundary conditions is presented in Section 6. In the final Section 7, the conclusions and future work are presented.

## 2. The numerical method framework

### 2.1. Governing equations

The governing equations employed in the paper are the incompressible Navier-Stokes equations which in continuous form read,

$$\nabla \cdot \mathbf{u} = 0, \quad (2.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -\left((\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{\rho} \nabla P\right) + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \quad (2.2)$$

where  $\mathbf{u}(\mathbf{x}, t)$  and  $p(\mathbf{x}, t)$  is the velocity and pressure field respectively for a constant reference density  $\rho$  at time  $t \in \mathbb{R}$  with a kinematic molecular viscosity of  $\nu$  and a forcing field  $\mathbf{f}(\mathbf{x}, t)$  with

$\mathbf{x} \in \mathbb{R}^d$ , for  $d = 1, \dots, 3$ . It is noted that the advection term in Eq. (2.2) is not in the skew-symmetric form which reduces the aliasing errors originating from the real eigenvalues, customarily used in spectral schemes [31]. The reasoning is the absence of sharp discontinuities which are often associated with spectral blocking.

### 2.2. Time stepping

The time integration scheme is the classical semi-implicit fractional projection step method of Cummins and Rudman [17] which reads,

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \mathbf{F}^n, \quad (2.3)$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = \nabla \tilde{p}^{n+1}, \quad (2.4)$$

with

$$\mathbf{F}^n = -(\mathbf{u}^n \cdot \nabla \mathbf{u}^n) + \nu \nabla^2 \mathbf{u}^n + \mathbf{f}, \quad (2.5)$$

where the pressure is integrated in time over a sub-step  $n \in \mathbb{N}$  as

$$\tilde{p}^{n+1} = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} p \, dt. \quad (2.6)$$

The rationale for the low-order time integration scheme employed in this work is to produce comparable results with the SPH literature [3,30,32] and reduce the computational cost of the projection step to relatively manageable levels. Nevertheless, the widely used higher-order, low storage, Runge-Kutta (RK) schemes [33] may be employed as demonstrated in Sections 5 and 6.

The numerical solution is bounded by a CFL condition which reads,

$$\Delta t = C \frac{\Delta x}{u_{\max}}, \quad (2.7)$$

where  $C = 0.1$  is the Courant number,  $u_{\max}$  is the maximum velocity magnitude of the system, and  $\Delta x$  is the nodal or particle spacing.

### 2.3. SPH formalism

In this section, the basic SPH principles are recalled and it is assumed that the reader is familiar with the scheme. For extensive reviews the reader is directed to [2,34] and [35].

The core principle of SPH is the convolution of a sufficiently smooth function  $f(\mathbf{x})$  with a kernel function  $W : \mathbb{R}^d \rightarrow \mathbb{R}$  in a domain  $\Omega$ , defined as

$$\langle f(\mathbf{x}) \rangle := \int_{\Omega_{\mathbf{x}}} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}', \quad (2.8)$$

over  $\mathbf{x}' \in \mathbb{R}^d$  defined locally as  $\Omega \cap \Omega_{\mathbf{x}}$  i.e., a compact support at  $\mathbf{x}$ , with a characteristic length  $h \in \mathbb{R}^+$ . Therefore,  $h$  defines the radius of influence (the size of the support domain) of the weighting kernel function  $W$ . The brackets on the left-hand side of Eq. (2.8) denotes an SPH interpolant and will be dropped henceforth for simplicity. The smoothing kernel  $W$  is of the general form,

$$W(\mathbf{x}, h) := \frac{1}{h^d} \omega(|q|), \quad (2.9)$$

with  $q = |\mathbf{x}|/h$ ,

$$\int_{\Omega_{\mathbf{x}}} W(\mathbf{x}, h) d\mathbf{x} = 1, \quad (2.10)$$

and compact support defined by

$$W(\mathbf{x}, h) = 0 \quad \text{for } |\mathbf{x}| \geq kh, \quad k \in \mathbb{R}^+, \quad (2.11)$$

$\omega$  is a monotonic decreasing sufficiently smooth function with

$$\lim_{h \rightarrow 0} W(\mathbf{x}, h) = \delta(\mathbf{x}), \quad (2.12)$$

where  $\delta$  is the Dirac delta function [34]. Several Gaussian shape kernels have been proposed in the literature [27]. Herein, the quintic spline kernel [18] with a  $3h$  compact support and the  $G^4$  Gaussian kernel [29] with a  $4h$  compact support has been employed to demonstrate second and higher-order convergence respectively.

In a discrete form and by dropping the brackets for brevity, Eq. (2.8) is defined as

$$f(\mathbf{x}_i) := \sum_{j \in P} f(\mathbf{x}_j) W(\mathbf{x}_i - \mathbf{x}_j, h) \Delta x_j^d, \quad (2.13)$$

with subscripts  $i, j \in P = \{1, \dots, N\}$  denoting the interpolating and neighbouring nodes (or particles in SPH formalism) with a particle volume  $V_j = \Delta x_j^d$ . The continuous and discrete form of a gradient in SPH take the following forms,

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} := \int_{\Omega_{\mathbf{x}}} f(\mathbf{x}') \frac{\partial W}{\partial \mathbf{x}}(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}', \quad (2.14)$$

$$\frac{\partial f(\mathbf{x}_i)}{\partial \mathbf{x}} := \sum_{j \in P} f(\mathbf{x}_j) \frac{\partial W}{\partial \mathbf{x}}(\mathbf{x}_i - \mathbf{x}_j, h) V_j, \quad \forall i \in P, \quad (2.15)$$

respectively, by employing the convolution identity,

$$\frac{\partial f}{\partial \mathbf{x}} * W = f * \frac{\partial W}{\partial \mathbf{x}}. \quad (2.16)$$

### 3. Spatial discretisation

#### 3.1. Advection and viscous term

Consider a 2-D domain of dimensions  $L_x \times L_y$  discretised on a Cartesian particle distribution of  $n = n_x \times n_y$  particles with a spatial distribution of  $(x, y) = ((i-1)\Delta x, (j-1)\Delta y)$ ,  $i, j = \{1, \dots, N\}$ . Employing the discrete gradient of Eq. (2.15) and using the shorthand  $\mathbf{f}_i = f(\mathbf{x}_i)$ , the advection term of Eq. (2.2) can be written as [30]

$$\mathbf{u} \cdot \nabla \mathbf{u}_i = \sum_j \mathbf{u}_{ij} \cdot (\mathbf{u}_{ij} \otimes \nabla W_{ij}) V_j, \quad \forall i, \quad (3.1)$$

with  $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h)$  and  $(\cdot)_{ij} = (\cdot)_i - (\cdot)_j$  for a general function, i.e.,  $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ .

The Laplacian term of Eq. (2.2) may use similar forms as per Eq. (2.15) by employing the second derivative of the kernel function. However, such terms have proven to be unstable due to the characteristics of the kernel especially with particle distributions in a lattice which is non-Cartesian (i.e., non-homogeneous particle distributions). In addition, the checkerboard effect associated with aliasing of the operator is severe due to the collocated nature of the scheme and therefore, it is common practice to use an over-dissipative operator which uses a finite difference scheme to approximate the first derivative [18] as shown below,

$$\nu \nabla^2 u \Big|_i = \sum_j m_j \frac{2(\nu_i + \nu_j) \mathbf{x}_{ij} \cdot \nabla W_{ij}}{\rho_j (x_{ij}^2)} \mathbf{u}_{ij}, \quad \forall i. \quad (3.2)$$

#### 3.2. Pressure gradient

There are several forms of gradient operators in SPH. In Eq. (2.15) the gradient operator does not strictly satisfy a zero gradient over a constant field function for a non-Cartesian particle distribution. Another form of the SPH gradient operator may be written as

$$\nabla f^{(\pm)} \Big|_i = \sum_j^N (f_j \pm f_i) \nabla W_{ij} V_j, \quad \forall i. \quad (3.3)$$

The positive gradient operator of Eq. (3.3) has enhanced conservation properties but also does not satisfy a zero gradient over a constant field [36], whereas the negative form guarantees a zero gradient and is suitable for the approximation of  $\nabla \bar{p}$ . Numerical experiments [29] have demonstrated the advantages of the negative form in incompressible fluid flows and therefore is preferred and will be denoted as  $\nabla^{(-)} f$  henceforth. Similarly, it is possible to construct a divergence operator from Eq. (3.3).

#### 3.3. A consideration on accuracy

The theoretical convergence of SPH has been discussed in the recent work of Franz and Wendland [38] with application to barotropic flows. Additionally, the accuracy and consistency characteristics of the SPH scheme over equispaced and non-equispaced discretised domains has also been examined in the past [37]. Herein, we take advantage of the formulation of Lind and Stansby [29] which demonstrated that higher than second-order convergence rates in SPH can be achieved based on the kernel characteristics with arbitrary even order of spatial accuracy. This is achieved by constructing symmetric kernels using high-order terms such as

$$f(\mathbf{x}) = \sum_{k=0}^n A_k f^{(k)}(\mathbf{x}) + O(h^n), \quad (3.4)$$

$$A_k = \frac{(-1)^k h^k}{k!} \int_{\Omega_{\mathbf{x}}} \left( \frac{\mathbf{x} - \mathbf{x}'}{h} \right)^k W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}', \quad (3.5)$$

to the  $n$ -th order for  $(k)$  derivatives of a field function. For  $k = 2$  a positive kernel can be constructed with the odd term  $k = 1$  being zero. By using higher-order terms in Eq. (3.4), even terms from the Taylor series expansion are removed and lower-order odd error terms cancel due to kernel symmetry thus, arbitrary even higher than second-order accurate kernels can be constructed. Although these high-order kernels are compactly supported, they are partially negative [39] and therefore are not monotonically decreasing. Partially negative kernels are not preferred for non-equispaced SPH discretisation. **Herein, we use partially negative high-order kernel in the form of the  $G^4$  Gaussian kernel to achieve fourth-order spatial accuracy.**

### 4. Solution methods for the pressure Poisson equation

#### 4.1. Spatial discretisation of the pressure Poisson equation

In a semi-implicit fractional projection step method, incompressibility is enforced by projecting the intermediate velocity to a divergence free space by solving a Poisson equation of the general form of

$$\nabla^2 \phi = g, \quad (4.1)$$

where  $g$  is the source term. It is generally accepted that the computational cost of the inversion of the Poisson equation using iterative linear solvers is the most computationally intensive part of a

incompressible solver [21] especially for high-order schemes that require iterative techniques to achieve second-order and higher-order time integration (i.e. RK schemes). Two different techniques employed in this paper for the solution of Eq. (4.1) are discussed below.

#### 4.2. Linear solver in space domain

By recasting Eq. (4.1), the pressure Poisson equation takes the following form,

$$\nabla \cdot \nabla \tilde{p}^{n+1} = \frac{1}{\Delta t} \nabla^{(-)} \cdot \mathbf{u}^*, \quad (4.2)$$

for an incompressible fluid. The pressure Poisson equation can be discretised by Eqs. (3.2) and (3.3) for the Laplacian of pressure and the divergence of velocity as follows,

$$\sum_j^N V_j \frac{2(\tilde{p}_i^{n+1} - \tilde{p}_j^{n+1}) \mathbf{x}_{ij} \cdot \nabla W_{ij}}{(x_{ij}^2)} = \frac{1}{\Delta t} \sum_j^n (\mathbf{u}_j^* - \mathbf{u}_i^*) \cdot \nabla W_{ij} V_j, \quad (4.3)$$

which forms a linear system of  $A(i, j)X(i) = B(i)$  with the following entries for the off diagonals terms,

$$A(i, j) = -2V_j \frac{\mathbf{x}_{ij} \cdot \nabla W_{ij}}{\rho_j (x_{ij}^2)}, \quad (4.4)$$

with diagonals terms,

$$A(i, i) = -\sum_j^n A(i, j), \quad (4.5)$$

and the divergence of fractional velocity

$$B(i) = \frac{1}{\Delta t} \sum_j^n (\mathbf{u}_j^* - \mathbf{u}_{ji}^*) \cdot \nabla W_{ij} V_j, \quad (4.6)$$

where the system can be solved implicitly to obtain the time averaged pressure,

$$\tilde{p}^{n+1}(i) = X(i). \quad (4.7)$$

Herein, a stabilised biconjugate gradient solver (Bi-CGSTAB) with Jacobi preconditioner is employed [40,41] although, more sophisticated approaches have been used the past to scale the solver over MPI with GMRES and HYPRE BoomerAMG with considerable success [21].

#### 4.3. Pseudo-spectral methods

Although iterative linear solvers have been used traditionally in SPH, fast Fourier transforms (FFTs) have lower computational cost than their iterative linear solvers counterparts [42] and the inversion of the pressure Poisson equation in Fourier space is a viable alternative that has not been explored by SPH practitioners to date.

Consider Eq. (4.1) in a 1-D domain  $\Omega \subseteq \mathbb{R}$  of interval  $[0, L]$  for a sufficiently smooth source term  $g$  and a uniform particle distribution of

$$x_i = i \frac{L}{n}, \quad i = 0 \dots n, \quad (4.8)$$

in a **periodic domain**,

$$f(x) = (f(x) + L). \quad (4.9)$$

The discrete Fourier transform (DFT) of  $f$  may therefore be written as

$$\hat{f}_k = \frac{1}{n} \sum_{i=0}^{n-1} f_i e^{-2\pi i k i / n}, \quad \text{for } k = 0, \dots, n-1, \quad (4.10)$$

where  $\iota = \sqrt{-1}$  and  $2\pi k/L$  denotes the wave number. The inverse discrete Fourier transform (IDFT) is

$$f_i = \sum_{k=0}^{n-1} \hat{f}_k e^{+2\pi i k i / n}. \quad (4.11)$$

Eq. (4.10) computes the discrete Fourier coefficients of  $f_i$  with the discrete Fourier eigenvectors whereas, (4.11) computes the inverse onto the values at a point  $f_i$ . Since  $f \in \mathbb{R}$ , the Fourier series required in the solution of the pressure Poisson equation is a real-to-complex (real-to-complex) fast Fourier transform which is symmetric (Hermitian symmetry) and thus only  $n/2+1$  complex outputs are required. Further, Eqs. (4.10) and (4.11) are said to have an arithmetic complexity of  $\mathcal{O}(n \log(n))$  in 1-D.

Periodic boundary conditions are of academic interest to the solution of incompressible flows and two more boundary conditions (BCs) are necessary for more general application, Dirichlet and Neumann BCs. Each corresponds to real sine or cosine basis functions respectively (real-to-real). There exist sixteen FFT combinations for real even and odd symmetrical data with the most common and applicable to collocated schemes the discrete sine and cosine of first kind.

The Dirichlet boundary condition reads,

$$f = 0 \quad \text{on } \partial\Omega, \quad (4.12)$$

and the equivalent discrete sine transform of first kind (DST-I) takes the following form,

$$\hat{f}_k = 2 \sum_{i=0}^{n-1} f_i \sin[\pi(i+1)(k+1)/(n+1)], \quad (4.13)$$

for  $k = 0, \dots, n-1$ , on internal (fluid) particles with

$$x_i = (i+1) \frac{L}{n+1}, \quad i = 0 \dots n-1, \quad (4.14)$$

whereas the Neumann boundary condition,

$$\frac{\partial f}{\partial x} = 0 \quad \text{on } \partial\Omega, \quad (4.15)$$

which generally satisfies the symmetry condition  $f(x) = f(-x)$  and  $f(x+L) = f(L-x)$  uses the cosine transform of first kind (DST-I) is

$$\hat{f}_k = f_1 + (-1)^k f_{n-1} + 2 \sum_{i=0}^{n-2} f_i \cos[\pi i k / (n-1)], \quad (4.16)$$

for  $k = 0, \dots, n-1$ , that also contains boundary nodes (ghost nodes are also permitted) with

$$x_i = i \frac{L}{n-1}, \quad i = 0 \dots n-1. \quad (4.17)$$

Eq. (4.1) and to that extent the discretised Eq. (4.3) can be expressed based in the discrete transform as

$$-(\lambda_k)^2 \hat{\phi}_k = \hat{g}_k, \quad \forall k, \quad (4.18)$$

where  $\lambda_k$  are the eigenvalues associated with the base functions for the Laplacian operator. In 2-D the discrete Fourier transform can be written as

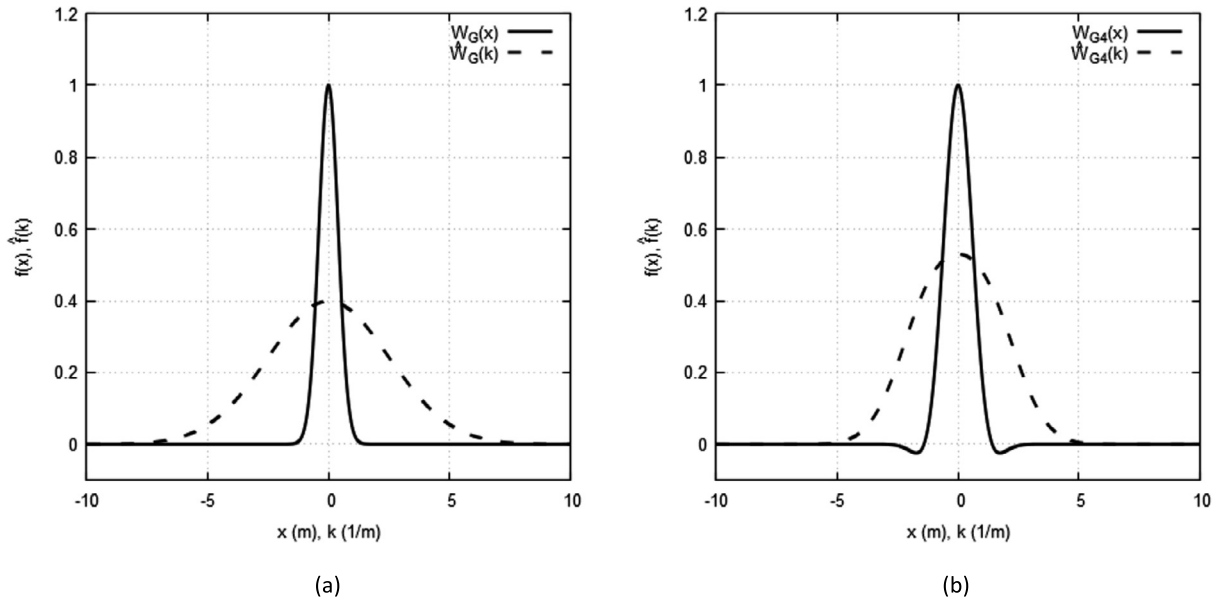


Fig. 1. Gaussian kernel (a) and G4 Gaussian (b) kernel in space and frequency domain.

**Table 1**  
Fourier transform types and corresponding eigenvalues for exact wave numbers [43].

Boundary condition	Particle arrangement	Transform type	Spectral eigenvalues
Periodic	Collocated	DFT	$\lambda_k = -\left(\frac{2\pi k}{L}\right)^2, k < [n/2] - 1$ $\lambda_k = -\left(\frac{2\pi(n-k)}{L}\right)^2, k \geq [n/2] - 1$
Dirichlet	Collocated	DST-I	$\lambda_k = -\left(\frac{\pi(k+1)}{L}\right)^2$
Dirichlet	Staggered	DST-II	$\lambda_k = -\left(\frac{\pi k}{L}\right)^2$
Neumann	Collocated	DCT-I	$\lambda_k = -\left(\frac{\pi k}{L}\right)^2$
Neumann	Staggered	DCT-II	$\lambda_k = -\left(\frac{\pi k}{L}\right)^2$

$$\hat{f}_{kl} = \frac{1}{n_x n_y} \sum_i \sum_j f_{ij} M_x(k_x x_i) M_y(l_y y_j), \quad (4.19)$$

where  $M$  is the transform type. The solution of Eq. (4.18) consists of division of the Fourier mode  $g_{kl}$  by the corresponding eigenvalues  $\Lambda_{kl} = \lambda_k^2 + \lambda_l^2$  as

$$\hat{\phi}_{kl} = \frac{\hat{g}_{kl}}{\Lambda_{kl}}, \quad (4.20)$$

and the IFFT is employed to obtain the sub-step pressure  $\tilde{p}_i^{n+1} = \text{IFFT}(\hat{\phi}_{kl})_i$ . A list of eigenvalues along with the Fourier transforms type is given in Table 1.

Modes which are singular for the second and high-order central finite difference schemes are associated with non-physical oscillations in collocated schemes. These modes with  $(\lambda_k, \lambda_l) = (0, 0)$  occurring at  $k_x = 0$  and  $k_x \Delta x = \pi$  significantly affect the pressure field from the solution of Eq. (4.20). In physical space these modes manifest as a checkerboard effect on the pressure whereas in spectral space these modes can be suppressed by skipping the corresponding null modes. To eliminate pressure checkerboarding, staggered (or partially staggered) particle arrangement may be employed with sufficient lower null modes (i.e., at  $k_x = 0$ ). In general, the use of the exact wave number and eigenvalues of Table 1 is not recommended when approximating spatial operators due to the Gibbs phenomenon arising from discontinuities in the flow and spatial domain on equispaced node distribution [44].

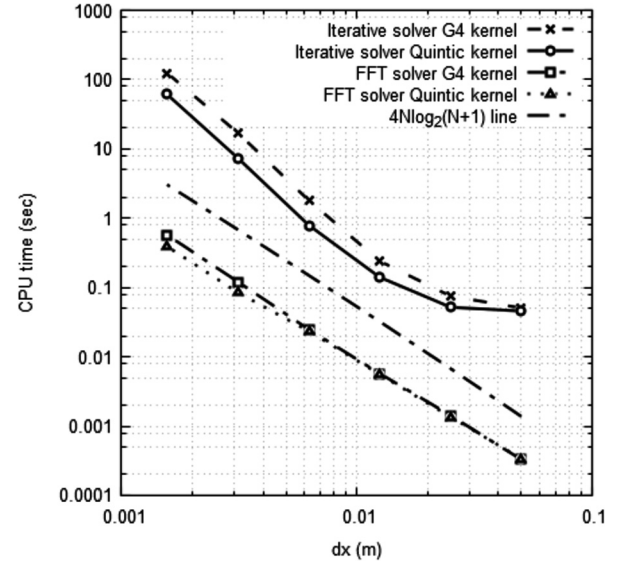


Fig. 2. Computational cost comparison of pressure Poisson equation between FFTs and iterative solver for a second-order and fourth-order kernel using six particle resolutions with  $h/dx = 2.0$ .

Herein, we use the unique advantages of the Gaussian kernel. The SPH kernel approximates the Gaussian kernel closely with the Fourier transform of the Gaussian kernel being the transformed Gaussian itself which is even and real. Indeed, the compactly supported G4 Gaussian kernel is used in our analysis with the Gaussian and G4 Gaussian kernels plotted in Fig. 1. Hence and for brevity, in this work we will use the exact wave number in the absence of sharp discontinuities such as shock waves and to ensure consistency with other parts of the SPH solver.

## 5. Software implementation

### 5.1. Fast Fourier transforms library implementation

The standard incompressible SPH solver [40] is a 2-D serial code using the Bi-CGSTAB with Jacobi preconditioner for the solution of the pressure Poisson equation using the time integration described in Eqs. (2.3) to (2.6). However, the iterative solver com-



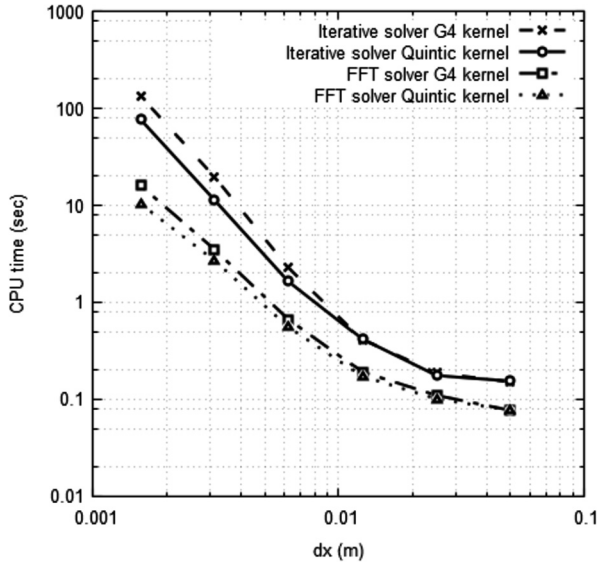


Fig. 3. Computational cost per timestep comparison between FFTs and iterative solver for a second-order and fourth-order kernel with  $h/dx = 2.0$ .

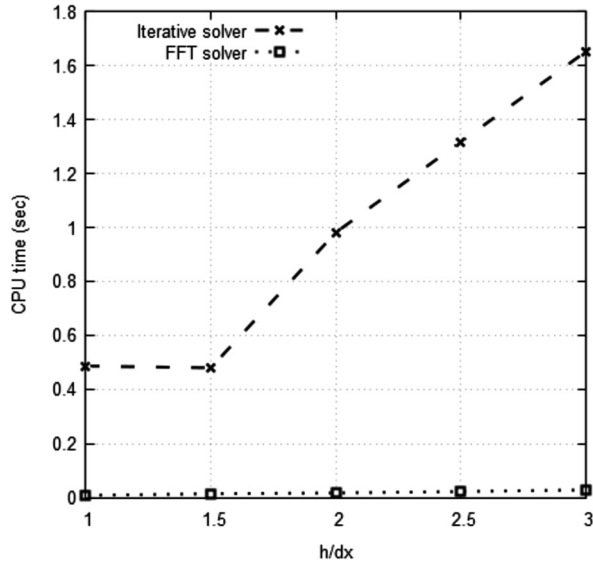


Fig. 4. Dependency of computational cost on the increase of kernel size for FFTs and iterative solver using  $160^2$  particles.

computational cost increases significantly with the number of particles and/or neighbouring particles within the kernel support domain as demonstrated below in Section 5.2. Therefore, an FFT solver is employed as a computationally efficient alternative for the solution of the pressure Poisson equation. Herein, we take advantage of the fast Poisson solver library PoisFFT [45] written in FORTRAN that also provides C++ bindings. It is based on the FFTW3 library [46] for the discrete fast Fourier transforms. The PoisFFT library is OpenMP enabled with an MPI implementation based on the PFFT library [47]. In this work, the OpenMP implementation is utilised for most test cases although extension to MPI is feasible and is reserved for future work. The PoisFFT library provides a general framework for initialisation, execution, and finalisation of the FFTW3 library with a variety of boundary conditions (periodic, Neumann, Dirichlet and mixed boundary conditions) using either the exact or modified wavenumbers with second and fourth-order central schemes for the latter two. In Table 2 a pseudo-algorithm shows the main steps of both the iterative and FFT solver im-

Table 2

ISPH pseudo-algorithm.

Step	Iterative solver	FFT solver
1:	time step $n \leftarrow \mathbf{u}^n, \bar{p}^n$	
2:	update wall boundary velocities	
3:	compute advection term, Eq. (3.1)	
4:	compute viscous term, Eq. (3.2)	
5:	compute RHS of Eq. (4.3)	
6:	compute LHS of Eq. (4.3)	forward transform, Eq. (4.19)
7:	build linear system	solve PPE in spectral space, Eq. (4.20)
8:	solve PPE iteratively	backwards transform
9:	Obtain pressure $\bar{p}^{n+1}$	
10:	Compute pressure gradient Eq. (3.3)	
11:	Obtain velocity at $\mathbf{u}^{n+1}$	
12:	time step $n + 1 \rightarrow \mathbf{u}^{n+1}, \bar{p}^{n+1}$	

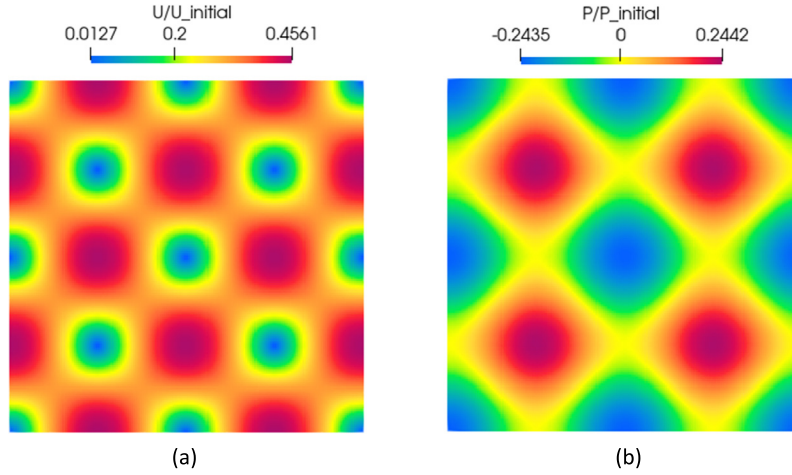
plementations where steps 6–8 show the differences between the iterative ISPH solver and the new ISPH-FFT solver.

## 5.2. Performance tests

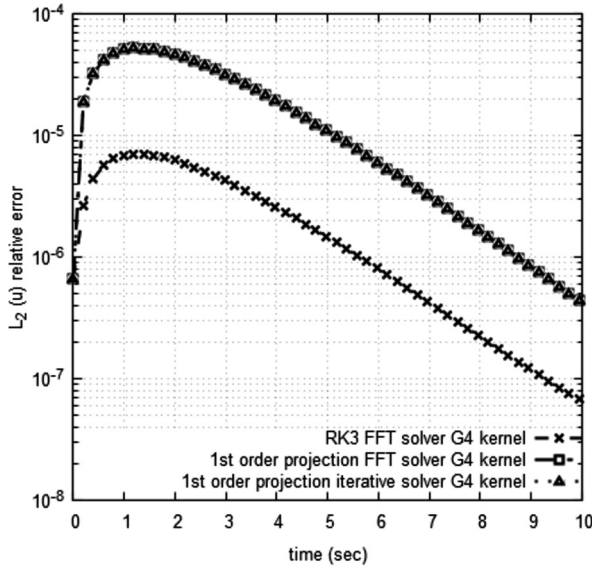
In order to provide a fair comparison between the iterative solver and the FFTs, all other parts of the SPH solver [40] remain unchanged and only the setup and solution of the pressure Poisson solver is modified. This includes the assembly of the linear system, execution of the solver (either FFTs of iterative solver) and any pressure field calculations required on each variant of pressure Poisson solver. For our performance tests, the sequential execution of the PoisFFT library and iterative solver is used as the latter is only implemented in serial execution in the current SPH solver. A comprehensive study of the OpenMP and MPI performance evaluation is demonstrated in Fuka [45]. Here, we will focus on the numerical and computational behaviour of the solvers from an SPH prospective.

Since the computational cost of boundary conditions may vary slightly, a fully 2-D periodic case is used and a performance comparison between the FFT solver and the iterative solver (IS) is conducted. The 2-D test case has dimensions of  $1 \times 1$  m with a length to particle spacings of  $L/dx = [20, 40, \dots, 640]$  resulting in 400 to 409600 particles. A detailed description of the case is given in section 6.1.1. The tests are conducted on the computational shared facility 3 (CSF3) cluster of the University of Manchester, UK, on an Intel Skylake Xeon Gold 6130 CPU @ 2.10GH processor with 6 GB of memory. Although the performance tests were conducted using serial executions the remaining runs on this paper are executed on two Skylake processors with  $2 \times 16$  cores and 6 GB of memory per core. Performance gains for the PoisFFT library are demonstrated in Fuka [45].

The first test is a direct comparison of the computational cost in CPU wall time of the pressure Poisson solvers averaged over the timesteps for a simulation of  $t = 1.0$  s. It should be noted that the error threshold criterion for the iterative solver is set to  $10^{-10}$  with a maximum of 1000 iterations as an exit criterion throughout this paper. Fig. 2 shows the comparison of the computational cost for two different kernels, namely the quintic spline and Gaussian G4 kernel. The FFTs show a speedup of two orders of magnitude for all six particle resolutions. Furthermore, the CPU times exhibit different behaviour, with the FFTs following the expected  $\mathcal{O}(4N \log_2(N+1))$  computational complexity associated with FFTs in two dimensions. On the other hand, the iterative solver shows much higher computational complexity close to  $\mathcal{O}(N^3)$ . In Fig. 3, the CPU wall-time for a timestep averaged over all timesteps for  $t = 1.0$  s, is shown for two different kernels over six particle resolutions. The FFTs show an order of magnitude reduction in computational cost. This is to be expected as the serially executed SPH solver cost, dominates over the FFTs CPU wall time as shown in Fig. 2.



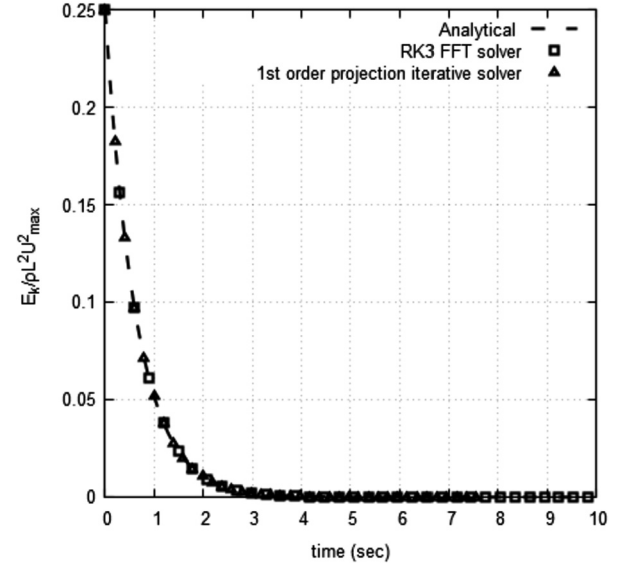
**Fig. 5.** Taylor-Green vortices velocity and pressure snapshot at  $t = 1$  s using the FFT and RK3 time integration for  $L/dx = 160$  particles and  $Re = 100$ . (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)



**Fig. 6.** Relative temporal  $L_2$  velocity error characteristics comparison between the FFT, IS solver for the RK3 and first-order projection time integration scheme for  $L/dx = 160$ .

One of the main issues encountered with iterative solvers in SPH, from a computational cost perspective, is the large number of neighbours required for the SPH interpolation. In 2-D for the above tests with a  $h/dx$  ratio of 2.0, it results in 98 and 161 neighbours for the quintic and G4 Gaussian kernel, respectively. These large numbers decrease the sparsity of the matrix which increases the computational cost. The computational cost is one of the main drawbacks of SPH as reported widely in the literature [2,35,48]. On the other hand, the FFTs do not exhibit similar behaviour since the computational cost is directly associated with the number of particles and is independent of the number of neighbours. The only dependency of the FFTs on the neighbours is a small computational overhead that occur when computing the right-hand side of Eq. (4.6) which of course is independent of the FFT calculations.

Nevertheless, since herein we compare the computational cost for setting up the linear system and solving the pressure Poisson equation, this cost is considered in the comparisons. Fig. 4 depicts a comparison of the CPU wall time for various  $h/dx$  ratios for the quintic kernel using a particle resolution of  $L/dx = 160$ . It is shown that the cost of FFTs in SPH is independent of the kernel support size (and therefore the number of neighbours) whereas the itera-



**Fig. 7.** Kinetic energy ( $E_k$ ) decay time history comparison of the RK3 FFT scheme and the first-order projection iterative solver using the Gaussian G4 kernel with  $L/dx = 160$  for  $Re = 100$ .

tive solver shows a linear increase of approximately a factor x3 in computational time with increasing kernel support radius  $h/dx$ .

## 6. Results and discussion

### 6.1. Periodic test cases

#### 6.1.1. Taylor-Green vortices

The Taylor-Green vortices are an unbounded periodic test case in a 2-D domain consisting of counter rotating vortices. The flow is considered to be incompressible with temporally decaying vortices in the presence of viscous dissipation. The analytical solution of the velocity and pressure field reads,

$$u = -\cos(2\pi x) \sin(2\pi y) e^{ct}, \quad (6.1)$$

$$v = \sin(2\pi x) \cos(2\pi y) e^{ct}, \quad (6.2)$$

$$p = \cos(2\pi x) \sin(2\pi y) e^{2ct}, \quad (6.3)$$

with  $c = -\frac{8\pi^2}{Re}$  where the  $Re = LU_{max}/\nu$  where  $L$  is defined as the horizontal length of the domain and  $U_{max}$  is the initial maxi-

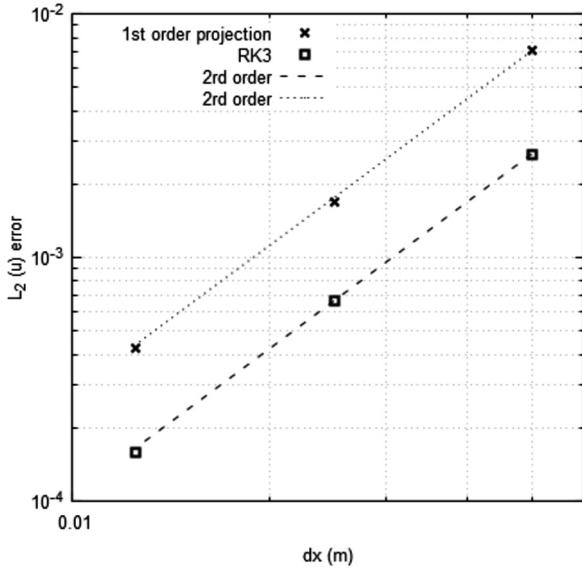


Fig. 8. Taylor-Green vortices convergence characteristics of first-order projection scheme and RK3 scheme using the quintic kernel  $t = 1.0$  s and  $Re = 100$ .

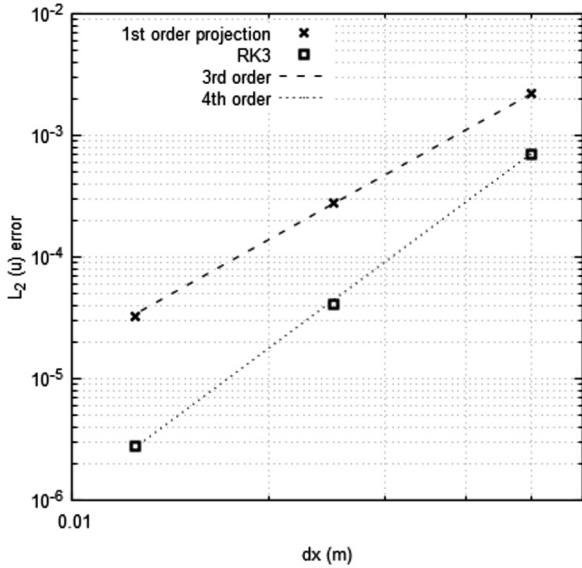


Fig. 9. Taylor-Green vortices convergence characteristics of first-order projection scheme and RK3 scheme using the fourth-order Gaussian kernel at  $t = 1.0$  s and  $Re = 100$ .

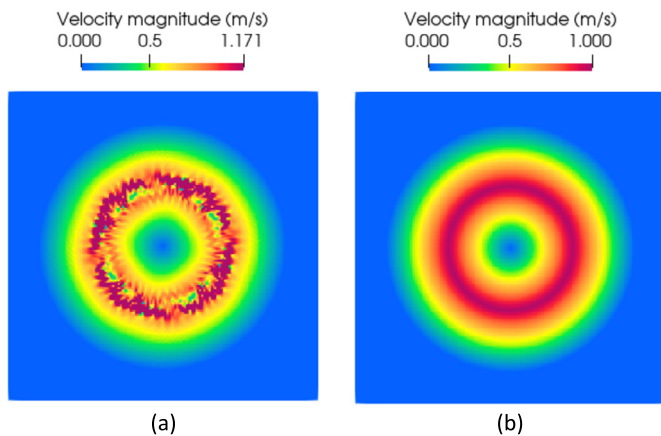


Fig. 10. Azimuthal velocity of (a) iterative solver (b) RK3 FFT solver at  $t = 1.2$  s with the G4 Gaussian kernel for  $h/dx = 1.3$  using  $L/dx = 160$ .

Table 3

CPU timings for Taylor-Green vortices at  $t = 10.0$  s for  $160^2$  particles.

Solver	FFTs	FFTs	IS	IS
Time integration	RK3	first-order	RK3	first-order
Total CPU time (s)	25029	18680	N/A	200499
CPU time per timestep (s)	1.56	1.1675	N/A	12.53

num velocity of the system. The periodic domain has dimensions of  $[-0.5, 0.5] \times [-0.5, 0.5]$  meters with a kinematic viscosity of  $\nu = 0.01$  m<sup>2</sup>/s resulting in a  $Re = 100$ .

In the absence of wall boundaries, the quintic and G4 Gaussian smoothing kernel provide a solution which is second and fourth-order spatial accurate respectively [29,39]. A third-order Runge-Kutta (RK3) scheme [33] with low storage is used to ensure the temporal error does not contaminate the spatial solution. Fig. 5 shows the velocity and pressure field at  $t = 1$  s using the FFTs solver with the RK3 time integration scheme for  $160^2$  particles and  $Re = 100$  normalised by the initial velocity and pressure, respectively, demonstrating the noise-free velocity and pressure components of the numerical scheme and the ability of FFTs to deal with the solution of the Poisson equation in periodic domains.

The following notation will be used to denote the time integration scheme and solver henceforth, where the first term refers to the time integration and the second term to the solver variant i.e., RK3 FFT refers to the Runge-Kutta (RK3) scheme using the FFT solver and so forth. To assess the temporal error, Fig. 6 shows a comparison of the relative  $L_2$  error for velocity until  $t = 10$  s between the FFT solver using the RK3 time integration scheme (RK3 FFT), the first-order projection time integration scheme of Eqs. (2.3) to (2.6) (first-order projection FFT) and the iterative solver using the first-order projection time integration scheme (first-order projection IS). In addition, the kinetic energy decay temporal evolution is shown in Fig. 7 for the RK3 FFT and first-order projection IS scheme and compared with the analytical solution.

In the RK3 FFT, the temporal error is a magnitude lower than the first-order scheme [30]. Nevertheless, the RK3 scheme comes with a large computational cost as three sweeps are required to obtain a solution which means solving the pressure Poisson equation three times.

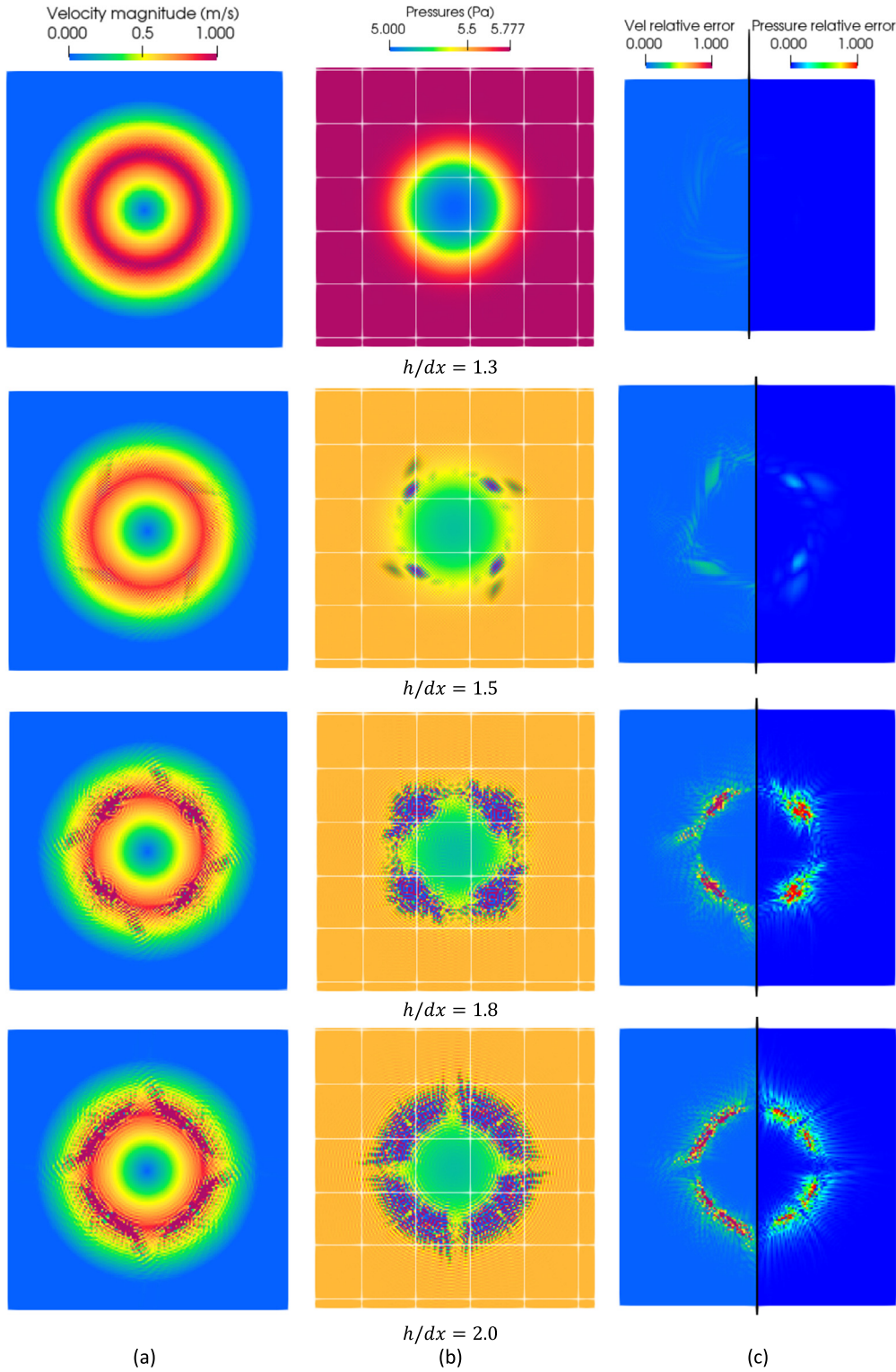
Using the iterative solver this is not practically possible due to the computational cost. Table 3 shows the computational cost of each solver variant. Notably, the RK3 FFT variant is only 33% more expensive than the first-order projection FFT solver variant and an order of magnitude faster than the first-order projection IS, similar to observations in the performance tests of Section 5.2.

Fig. 8 and Fig. 9 show the spatial convergence characteristics for the quintic and G4 Gaussian kernels using the first-order projection and RK3 time integration schemes with the FFTs pressure Poisson solver at  $t = 1$  s which is near where the maximum temporal error exists. Both time integration schemes converge with second-order behaviour for the quintic kernel. For the high-order G4 Gaussian kernel the first-order projection scheme converges with a rate of three and the RK3 with the expected convergence rate of four highlighting the importance of high-order time integration schemes in combination with high-order kernels to obtain higher-order convergence.

### 6.1.2. The Gresho vortex

The Gresho vortex test case is an incompressible, inviscid and time-independent case where the velocity depends only on the radial and centrifugal forces. The latter are balanced by the pressure gradient in an unbounded periodic domain. In the absence of dissipation, the case is ideal for testing the accuracy of the numerical scheme, the conservation of momentum and the ability of SPH to preserve pure rotation and, the stability of the FFT-ISPH scheme.





**Fig. 11.** Azimuthal velocity (a) pressure field (b) and relative error (c) for  $L/dx = 160$  using four different  $h/dx$  ratios at time  $t = 2.0$  s using the RK3 FFT solver with the G4 Gaussian kernel.

Usually this challenging test case is reserved for compactly supported high-order WENO schemes due to the high accuracy and conservation properties required [49]. The analytical azimuthal velocity and pressure is given by

$$u_\phi = \begin{cases} 5r & \text{if } 0 \leq r < 0.2 \\ 2 - 5r & \text{if } 0.2 \leq r < 0.4 \\ 0 & \text{otherwise} \end{cases}, \quad (6.4)$$

and

$$P = \begin{cases} 5 + \frac{25}{2}r^2 & \text{if } 0 \leq r < 0.2 \\ 9 + \frac{25}{2}r^2 - 20r - 4\ln 0.2 + 4\ln r & \text{if } 0.2 \leq r < 0.4 \\ 3 + 4\ln 2 & \text{otherwise} \end{cases}, \quad (6.5)$$

which is dependent only on the radius  $r$  defined in the centre of a Cartesian periodic domain with dimensions  $[0, 1] \times [0, 1]$  m.

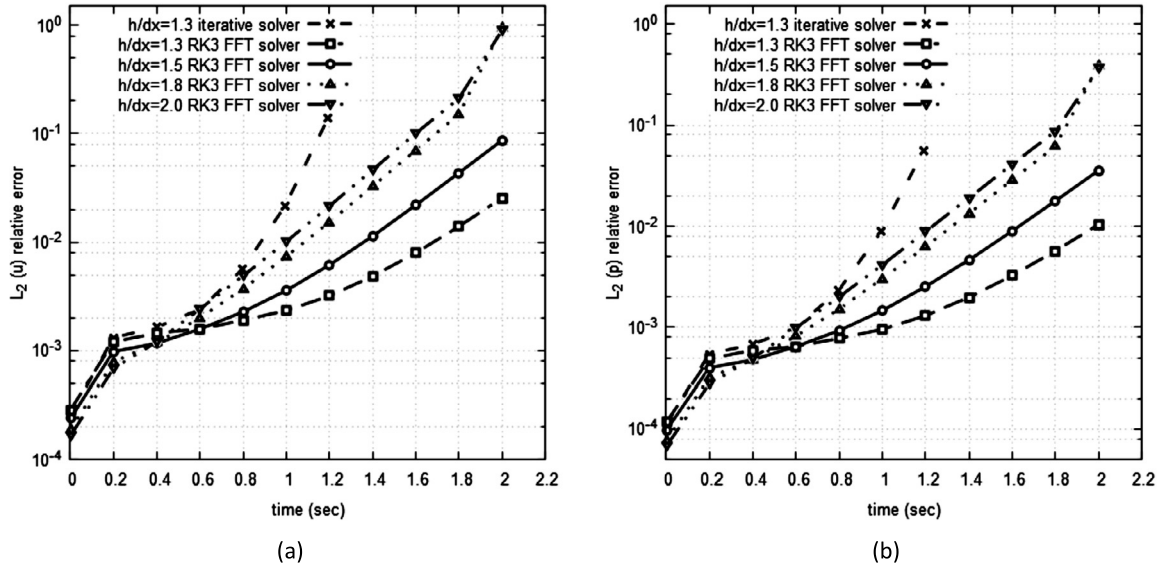


Fig. 12. Relative temporal  $L_2$  error for velocity (a) and pressure (b) using different  $h/dx$  ratios for the Gresho vortex with the G4 Gaussian kernel.

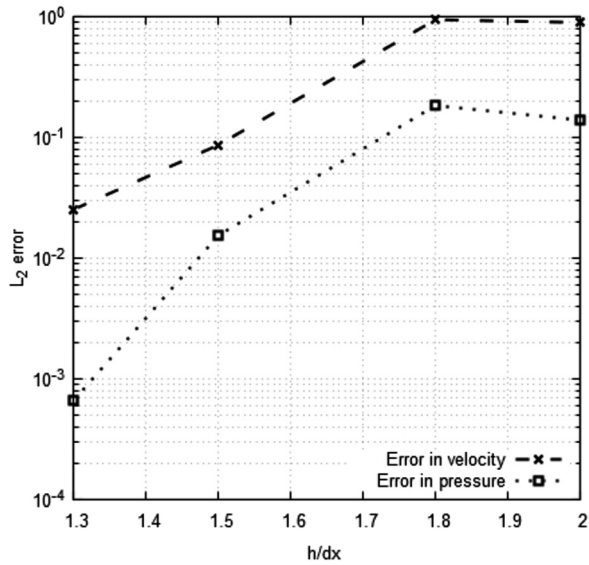


Fig. 13.  $L_2$  error for different  $h/dx$  using the RK3 FFT scheme and the G4 Gaussian kernel at  $t = 2.0$  s.

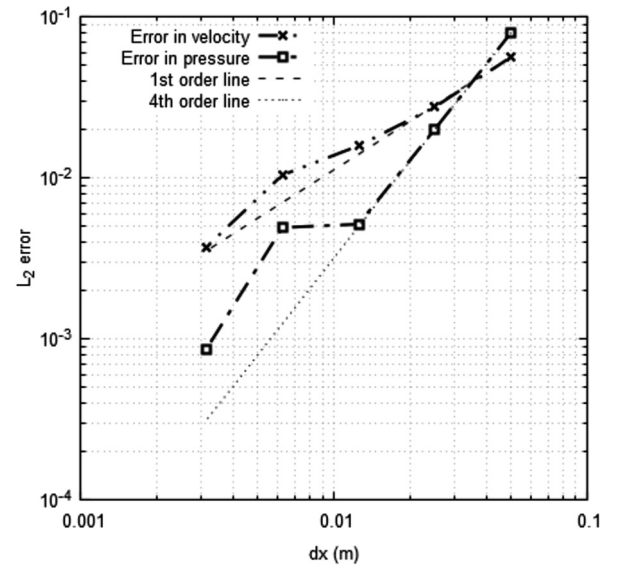


Fig. 14. Spatial convergence of the Gresho vortex for  $h/dx = 1.3$  using the RK3 FFT scheme and the G4 Gaussian kernel at  $t = 2.0$  s.

A direct comparison of the Gresho vortex using the iterative solver and the proposed RK3 FFT scheme is shown in Fig. 10 at  $t = 1.2$  s. This is the maximum attainable time the iterative solver reached before the checkerboarding effect dominated the solution causing the solver to stop. On the other hand, the RK3 FFT scheme reached  $t = 2.0$  s which is the maximum simulated time, showing the robustness of the proposed scheme. Iterative solvers usually require a Dirichlet condition to be defined in the domain to guarantee the uniqueness of the solution. Herein, the analytical pressure is prescribed as a necessary anchoring point for the iterative solver on a single particle at the corner of the domain which may have introduced numerical noise to the iterative solver. Furthermore, a higher error tolerance criterion up to  $10^{-12}$  on the iterative solver has been tested with no further improvements. Checkerboarding in ISPH (a collocated scheme) has attracted attention lately [39] as it restricts the applicability of the scheme at high orders of accuracy. For the FFT solver and by using the exact wave number, checkerboarding is avoided. Further, pressure anchoring using Dirichlet points is not required for FFTs.

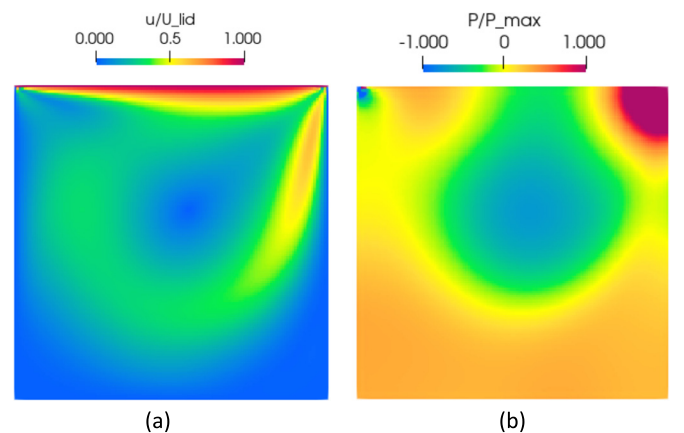
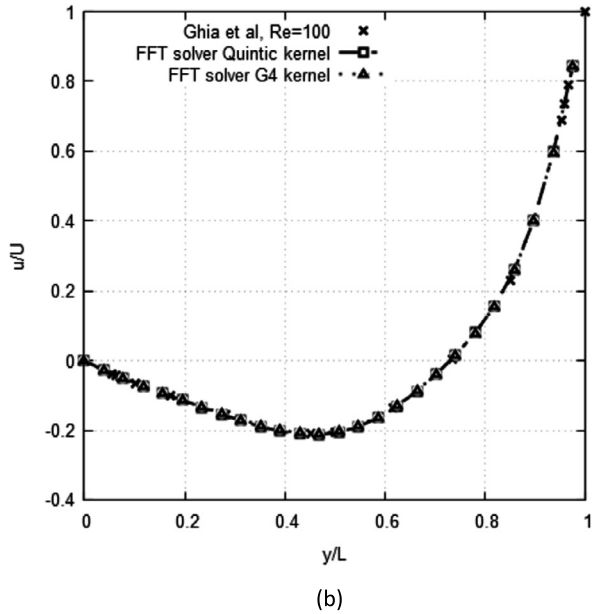
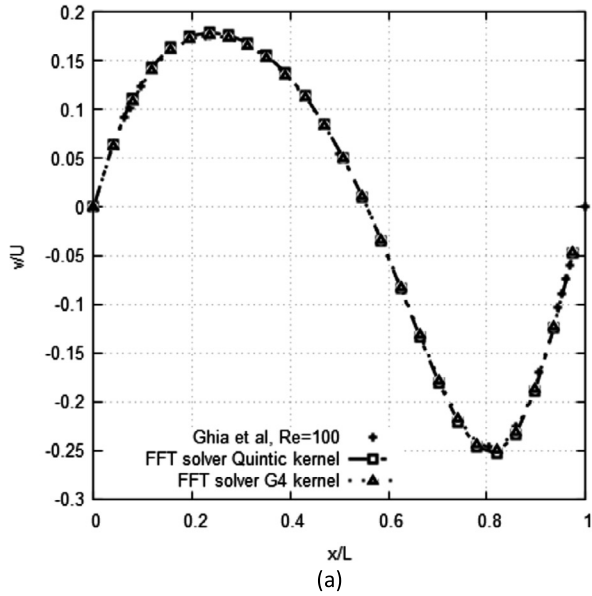
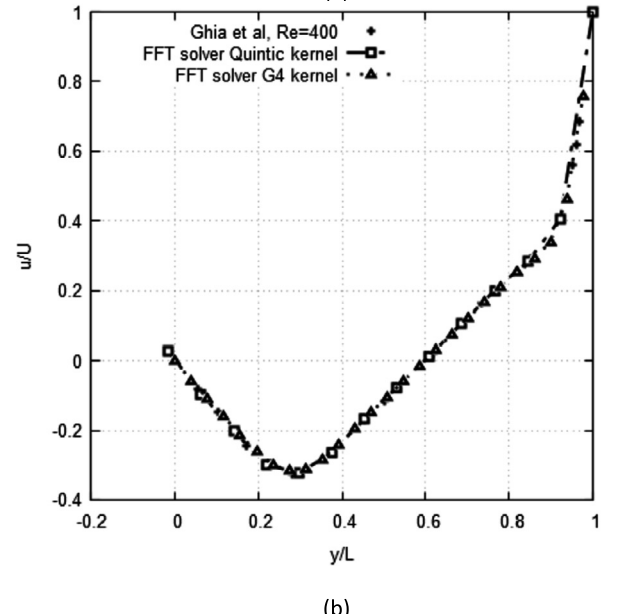
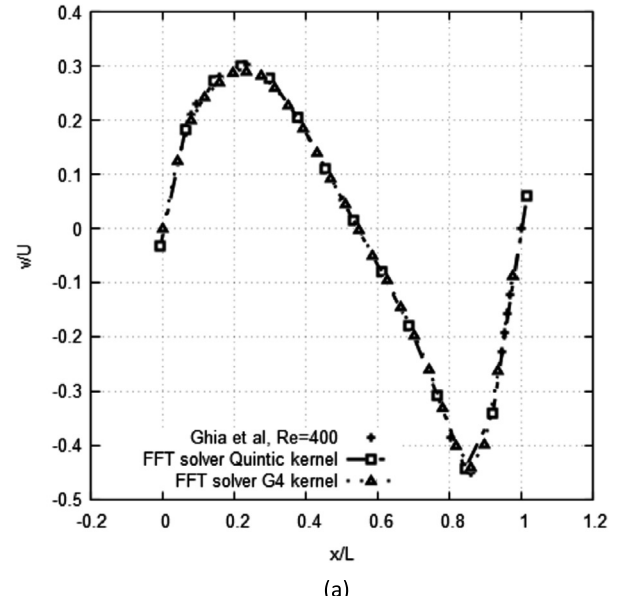


Fig. 15. Lid-driven cavity flow: velocity and pressure snapshot at  $t = 40$  s using the FFT and RK3 time integration with the G4 Gaussian kernel for  $160^2$  particles at  $Re = 400$ .



**Fig. 16.** Horizontal and vertical velocity profiles for two different kernels of the lid driven cavity  $Re = 100$  and comparison with Ghia, et al. [51], normalised by the lid velocity magnitude and domain length using  $160^2$  particles.



**Fig. 17.** Horizontal and vertical velocity profiles for two different kernels of the lid driven cavity  $Re = 400$  and comparison with Ghia, et al. [51], normalised by the lid velocity magnitude and domain length using  $160^2$  particles.

In the absence of any numerical error dissipation, numerical instabilities may arise especially for challenging inviscid test cases. This is demonstrated in Fig. 11 that shows the velocity, pressure field and relative error at  $t = 2.0$  s with 160 particles on the length  $L$  of the domain using the RK3 FFT scheme. Four different  $h/dx$  ratios have been tested ranging from 1.3 to 2.0. As the  $h/dx$  ratio increases, the SPH discretisation error is reduced [37] and numerical dissipation decreases resulting in numerical instabilities. This can be observed in Fig. 12 where the relative temporal  $L_2$  error is shown. Although, low  $h/dx$  ratios display higher initial errors, their error grows slower due to numerical error dissipation. These results have been repeated using the fourth central difference modified wave numbers and identical results were obtained. The  $L_2$  error behaviour of the velocity and pressure of the Gresho vortex for the different  $h/dx$  is shown in Fig. 13.

A spatial error convergence is shown in Fig. 14 demonstrating first-order convergence rate for the velocity and fourth-order for the pressure. Similar behaviour was observed by Springel [50] for

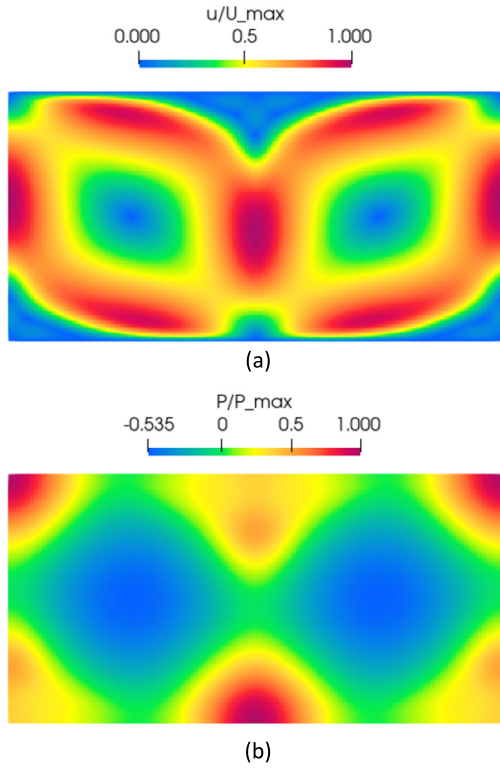
the velocity field with an order of convergence of 0.7 for velocity in the inviscid case using a Lagrangian weakly compressible SPH formulation. A possible reason for this behaviour is the effect of the non-linear advection term to the solution. The error behaviour in relation to the advection term will be examined in future work.

## 6.2. Bounded domain

Up to this point, the effectiveness of the FFT-ISPH solver and large reduction in computational cost has been demonstrated. Nevertheless, periodic cases are mostly of academic interest. Hence, a fully bounded domain test case is examined next in the form of the popular lid-driven cavity.

The domain is of dimensions  $[0, 1] \times [0, 1]$  meters with no-slip boundary conditions applied to the walls. The top boundary has an imposed horizontal wall velocity  $U_{lid} = 1$  m/s driving the flow until steady state is achieved. Two Reynolds numbers of 100 and 400 are considered.





**Fig. 18.** Velocity and pressure contours for two counter rotating vortices in a periodic channel at  $t = 2$  s for  $Re = 100$  and  $L/dx = 160$ .

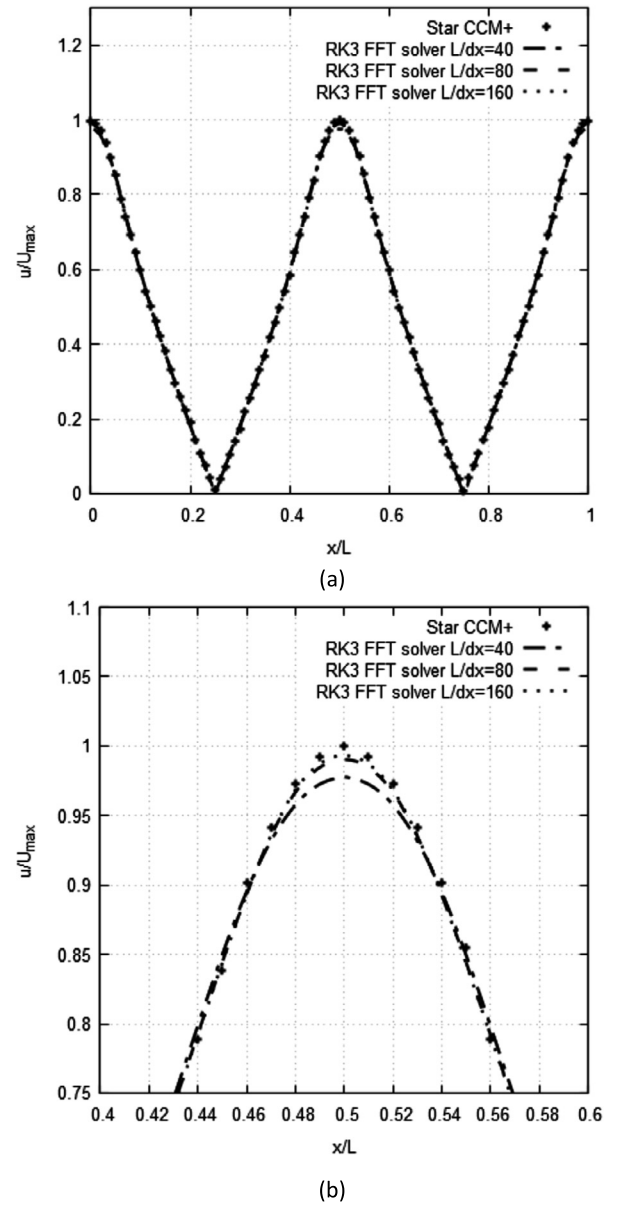
Imposing wall boundary conditions for the conventional SPH formulation, remains an open problem to the scientific community [48] and to date, no purely SPH high-order wall boundary conditions exist. Nevertheless, recent advances using a mixed SPH-finite difference approach have been successful on demonstrating high-order characteristics [39]. Our aim is the validation of the FFTs for the pressure Poisson equation and therefore for brevity, we use the straightforward mirror particle wall boundary condition approach [18] which is second-order accurate.

Fig. 15 shows the smooth velocity and pressure field obtained after  $t = 40$  s, when steady state has been reached, using the FFT RK3 scheme and the G4 Gaussian kernel with  $L/dx = 160$ . Further, the results are presented and compared with available literature [51] in Fig. 16 and Fig. 17 for two different Reynolds numbers of 100 and 400, respectively. Clearly close agreement is achieved using both kernels for the horizontal and vertical velocity profiles. Although we have used a large number of particles to test our proposed FFT RK3 solver for bounded domains, Lind and Stansby [29] have demonstrated that less than  $L/dx = 20$  are needed to obtain sufficiently good agreement in this Eulerian variant of SPH with the G4 Gaussian kernel making the scheme computationally inexpensive.

### 6.3. Mixed boundary conditions

The nature of computational fluid dynamics in practical applications may require more than one boundary condition. In the presence of mixed boundary conditions, a variety of combinations may be achieved by combining two identical boundary conditions in two spatial dimensions and a different on the third. In 2-D the third dummy dimension has a unit length.

To demonstrate the use of mixed boundary conditions, consider a 2-D channel with domain dimensions of  $[0, 2] \times [0, 1]$  meters. Periodicity is imposed on the horizontal direction and no-slip wall boundary conditions on the vertical walls. Two counter rotating



**Fig. 19.** Horizontal velocity (a) at a centreline cross section of the domain for three particle resolutions using the FFT RK3 scheme and the G4 Gaussian kernel at  $t = 2$  s and a close-up (b) in the middle of the domain for the horizontal velocity.

vortices are initialised in the domain according to the following velocity profile,

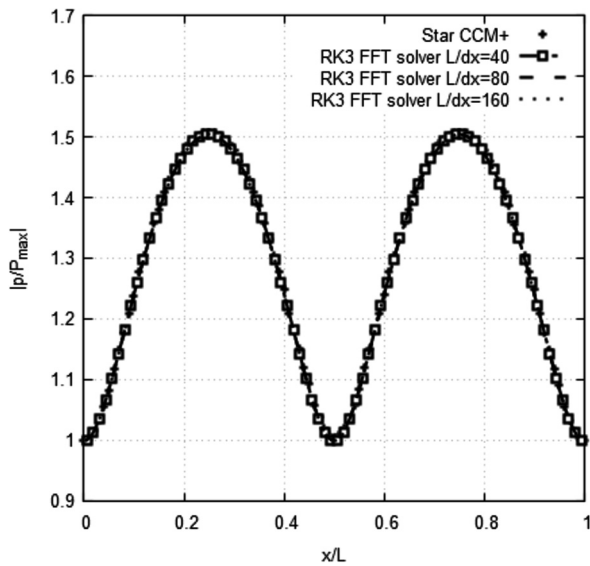
$$u = \begin{cases} (y - 0.5) & \text{if } 0 \leq x < 1 \\ (0.5 - y) & \text{if } 1 \leq x < 2 \end{cases}, \quad (6.6)$$

$$v = \begin{cases} (0.5 - x) & \text{if } 0 \leq x < 1 \\ (x - 1.5) & \text{if } 1 \leq x < 2 \end{cases}, \quad (6.7)$$

generating two symmetrical vortices in the channel. The flow is assumed to be viscous and incompressible with a Reynolds number of  $Re = 1000$ . With no external forces exerted on the fluid the counter-rotating vortices decay symmetrically about the middle of the channel. In the absence of an analytical solution, the finite volume software StarCCM+ has been taken as the reference solution for a quantitative comparison using a second-order scheme for the advection term with PISO unsteady segregated flow solver and a convergence criterion of  $10^{-6}$  using 12800 cells.

Fig. 18 shows the velocity and pressure field of the counter rotating vortices in a periodic channel using the FFT RK3 solver with





**Fig. 20.** Pressure at a centreline cross section of the domain for three particle resolutions using the FFT RK3 scheme and the G4 Gaussian kernel at  $t = 2$  s.

a G4 Gaussian kernel with symmetric vortices around the middle of the domain for the velocity and pressure, respectively demonstrating the effectiveness of FFTs in mixed boundary conditions. Fig. 19 (a) shows the quantitative comparison of the SPH results for three particle resolutions with the StarCCM+ solution for a horizontal cross section along the centreline of the domain showing the convergence agreement. A close-up is shown in Fig. 19 (b) depicts that by increasing the particle resolution the solution approaches the StarCCM+ result. In Fig. 20 the pressure comparison between SPH for three different particle resolutions and StarCCM+ is shown with excellent agreement.

## 7. Conclusions and future work

The smoothed particle hydrodynamics discretisation scheme has gained popularity in computational fluid dynamics with the incompressible SPH scheme making large advances over the last two decades. The ISPH variant is preferred over its weakly compressible form because of its smooth and accurate pressure and larger permissible time steps owing to the projection method for strictly incompressible flows. Nevertheless, inverting the linear system has been a drawback of the scheme from a computational cost perspective.

Herein a pseudo-spectral ISPH scheme has been proposed demonstrating large saving in computational cost, of two orders of magnitude. Further, it has been shown that the FFTs cost for solving the pressure Poisson equation is independent of the number of neighbours which was the major drawback in the solution of the pressure Poisson equation for ISPH. Second and fourth-order convergence rates have been demonstrated by comparing with analytical solutions using the classical first-order and higher-order low storage Runge–Kutta time integration schemes, respectively. Periodic, bounded, and mixed boundary conditions test cases have been demonstrated with highly accurate results in 2-D.

In future work, the Lagrangian and Eulerian-Lagrangian description of motion will be demonstrated which will allow for free surface flows to be stimulated efficiently and accurately using FFTs and higher-order wall boundary conditions [39]. Ultimately, this work will serve as the basis for an MPI pseudo-spectral incompressible SPH solver [21].

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors would like to acknowledge the assistance given by Research IT and the use of the Computational Shared Facility at the University of Manchester.

## References

- [1] R.A. Gingold, J.J. Monaghan, *Mon. Not. R. Astron. Soc.* 181 (1977) 375–389.
- [2] D. Violeau, B.D. Rogers, *J. Hydraul. Res.* 54 (2016) 1–26.
- [3] A. Leroy, D. Violeau, M. Ferrand, C. Kassiotis, *J. Comput. Phys.* 261 (2014) 106–129.
- [4] C. Altomare, A.J.C. Crespo, J.M. Domínguez, M. Gómez-Gesteira, T. Suzuki, T. Verwaest, *Coast. Eng.* 96 (2015) 1–12.
- [5] P.N. Sun, A. Colagrossi, D. Le Touzé, A.M. Zhang, *J. Fluids Struct.* 90 (2019) 19–42.
- [6] A. Khayyer, Y. Shimizu, H. Gotoh, K. Nagashima, *Appl. Math. Model.* (2021).
- [7] J.J. Monaghan, *Rep. Prog. Phys.* 68 (2005) 1703.
- [8] H. Gotoh, A. Khayyer, *J. Ocean Eng. Mar. Energy* 2 (2016) 251–278.
- [9] M.S. Shadloo, G. Oger, D. Le Touzé, *Comput. Fluids* 136 (2016) 11–34.
- [10] S.J. Lind, B.D. Rogers, P.K. Stansby, *Proc. R. Soc. A* 476 (2020) 20190801.
- [11] A.D. Chow, B.D. Rogers, S.J. Lind, P.K. Stansby, *Comput. Fluids* 179 (2019) 543–562.
- [12] C. Altomare, A.J.C. Crespo, B.D. Rogers, J.M. Domínguez, X. Gironella, M. Gómez-Gesteira, *Comput. Struct.* 130 (2014) 34–45.
- [13] G. Fourtakas, B.D. Rogers, *Adv. Water Resour.* 92 (2016) 186–199.
- [14] M. Antuono, A. Colagrossi, S. Marrone, *Comput. Phys. Commun.* 183 (2012) 2570–2580.
- [15] G. Fourtakas, J.M. Domínguez, R. Vacondio, B.D. Rogers, *Comput. Fluids* 190 (2019) 346–361.
- [16] M.D. Green, R. Vacondio, J. Peiró, *Comput. Fluids* 179 (2019) 632–644.
- [17] S.J. Cummins, M. Rudman, *J. Comput. Phys.* 152 (1999) 584–607.
- [18] J.P. Morris, P.J. Fox, Y. Zhu, *J. Comput. Phys.* 136 (1997) 214–226.
- [19] J.H. Ferziger, M. Perić, R.L. Street, *Computational Methods for Fluid Dynamics*, Springer, 2002.
- [20] Q.W. Ma, Y. Zhou, S. Yan, *J. Ocean Eng. Mar. Energy* 2 (2016) 279–299.
- [21] X. Guo, B.D. Rogers, S. Lind, P.K. Stansby, *Comput. Phys. Commun.* 233 (2018) 16–28.
- [22] A.J. Crespo, J.M. Domínguez, B.D. Rogers, M. Gómez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro, O. García-Feal, *Comput. Phys. Commun.* 187 (2015) 204–216.
- [23] C. Canuto, M. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York, 1988.
- [24] S. Laizet, E. Lamballais, *J. Comput. Phys.* 228 (2009) 5989–6015.
- [25] M.O. Deville, P.F. Fischer, P.F. Fischer, E. Mund, *High-Order Methods for Incompressible Fluid Flow*, Cambridge University Press, 2002.
- [26] M.J. Robinson, *Turbulence and Viscous Mixing Using Smoothed Particle Hydrodynamics*, Monash University, 2009.
- [27] W. Dehnen, H. Aly, *Mon. Not. R. Astron. Soc.* 425 (2012) 1068–1082.
- [28] D. Violeau, A. Leroy, A. Joly, A. Hérault, *Comput. Math. Appl.* 75 (2018) 3649–3662.
- [29] S.J. Lind, P.K. Stansby, *J. Comput. Phys.* 326 (2016) 290–311.
- [30] G. Fourtakas, P.K. Stansby, B.D. Rogers, S.J. Lind, *Comput. Methods Appl. Mech. Eng.* 329 (2018) 532–552.
- [31] A.G. Kravchenko, P. Moin, *J. Comput. Phys.* 131 (1997) 310–322.
- [32] S.J. Lind, R. Xu, P.K. Stansby, B.D. Rogers, *J. Comput. Phys.* 231 (2012) 1499–1523.
- [33] J. Williamson, *J. Comput. Phys.* 35 (1980) 48–56.
- [34] J.J. Monaghan, *Annu. Rev. Astron. Astrophys.* 30 (1992) 543–574.
- [35] D. Violeau, *Fluid Mechanics and the SPH Method: Theory and Applications*, Oxford University Press, 2012.
- [36] A. Colagrossi, B. Bouscasse, M. Antuono, S. Marrone, *Comput. Phys. Commun.* 183 (2012) 1641–1653.
- [37] N.J. Quinlan, M. Basa, M. Lastiwka, *Int. J. Numer. Methods Eng.* 66 (2006) 2064–2085.
- [38] T. Franz, H. Wendland, *SIAM J. Math. Anal.* 50 (2018) 4752–4784.
- [39] A.M.A. Nasar, G. Fourtakas, S.J. Lind, B.D. Rogers, P.K. Stansby, J.R.C. King, *J. Comput. Phys.* (2020) 109793.
- [40] R. Xu, P. Stansby, D. Laurence, *J. Comput. Phys.* 228 (2009) 6703–6725.
- [41] E.S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, P. Stansby, *J. Comput. Phys.* 227 (2008) 8417–8436.

- [42] A. Gholami, D. Malhotra, H. Sundar, G. Biros, arXiv preprint, arXiv:1408.6497, 2014.
- [43] U. Schumann, R.A. Sweet, J. Comput. Phys. 75 (1988) 123–137.
- [44] D. Gottlieb, C.-W. Shu, SIAM Rev. 39 (1997) 644–668.
- [45] V. Fuka, Appl. Math. Comput. 267 (2015) 356–364.
- [46] M. Frigo, S.G. Johnson, Proc. IEEE 93 (2005) 216–231.
- [47] M. Pippig, SIAM J. Sci. Comput. 35 (2013) C213–C236.
- [48] R. Vacondio, C. Altomare, M. De Leffe, X. Hu, D. Le Touzé, S. Lind, J.-C. Marongiu, S. Marrone, B.D. Rogers, A. Souto-Iglesias, Comput. Part. Mech. (2020).
- [49] W. Boscheri, M. Dumbser, D. Balsara, Int. J. Numer. Methods Fluids 76 (2014) 737–778.
- [50] V. Springel, Annu. Rev. Astron. Astrophys. 48 (2010) 391–430.
- [51] U. Ghia, K.N. Ghia, C.T. Shin, J. Comput. Phys. 48 (1982) 387–411.