Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 298 (2016) 29–57
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A spectral element formulation of the immersed boundary method for Newtonian fluids☆

C.F. Rowlatt, T.N. Phillips*

School of Mathematics, Cardiff University, Cardiff, CF24 4AG, United Kingdom

Received 26 September 2014; received in revised form 18 August 2015; accepted 20 August 2015
Available online 28 September 2015

Abstract

A spectral element formulation of the immersed boundary method (IBM) is presented. The spectral element formulation (SE-IBM) is a generalisation of the finite element immersed boundary method (FE-IBM) based on high-order approximations of the fluid variables. Several schemes for tracking the movement of the immersed boundary are considered and a semi-implicit Euler scheme is shown to offer advantages in terms of accuracy and efficiency. High-order spectral element approximations provide improved area conservation properties of the IBM due to the incompressibility constraint being more accurately satisfied. Superior orders of convergence are obtained for SE-IBM compared with FE-IBM in both L^2 and H^1 norms. The area conservation and convergence properties of the scheme are demonstrated on a series of benchmark problems.

MSC: 65M70; 74F10

Keywords: Immersed boundary method; Spectral element; Newtonian; Fluid–structure interaction

1. Introduction

In a classical formulation of a fluid–structure interaction problem, the fluid and structure are treated separately where the fluid is solved on a time-dependent domain and coupled to the structure equations using appropriately chosen interface conditions. The fluid–structure system of equations is then solved computationally using either a partitioned approach or a monolithic approach. A monolithic approach involves solving a single non-linear system of equations for both the fluid and the structure. A partitioned approach involves two systems of equations which are solved separately and then coupled together by interface conditions. A common approach in the literature is to formulate the fluid equations using the Arbitrary–Lagrangian–Eulerian (ALE) technique (see e.g. [1,2]). One of the major drawbacks of the classical approach is the computational time required — remeshing is often needed as the computational domain for the fluid equations is time-dependent. ALE was introduced to overcome the difficulties

☆ All data for this research are openly available at http://dx.doi.org/10.17035/d.2015.100114.
* Corresponding author.
E-mail addresses: RowlattC@Cardiff.ac.uk (C.F. Rowlatt), PhillipsTN@cf.ac.uk (T.N. Phillips).

http://dx.doi.org/10.1016/j.cma.2015.08.010

caused by the reconstruction of the mesh in time. In a fluid–structure interaction problem, the fluid is considered in an ALE formulation and the structure in a Lagrangian formulation [2]. The ALE formulation introduces an additional frame of reference, called the referential frame or configuration, which tracks the motion of the mesh. The classical approach to fluid–structure interaction problems using an ALE formulation is quite complex, particularly the interface conditions which have to be formulated using the so-called ALE map. Additionally, an ALE formulation can be computationally expensive when large deformations are considered. An alternative approach was introduced by Peskin and is called the immersed boundary method (IBM).

The immersed boundary method (IBM), proposed by Peskin for studying flow patterns around heart valves [3], has been applied to a wide range of problems including arterial blood flow [4], modelling of the cochlea [5], modelling of red blood cells in Poiseuille flow [6] and flows involving suspended particles [7]. A comprehensive list of applications can be found in [8]. The IBM is both a mathematical formulation and a numerical scheme for fluid–structure interaction problems. As mentioned above, in a classical fluid–structure interaction problem, the fluid and the structure are considered separately and then coupled together via some suitable jump conditions. In the IBM however, the structure – which is usually immersed in a Newtonian fluid – is viewed as being part of the surrounding fluid. This means that only a single equation of motion needs to be solved (i.e. a one-phase formulation). Additionally, the IBM allows the immersed structure to move freely over the underlying fluid mesh, alleviating the need for the remeshing required in a classical formulation.

The IBM replaces the immersed structure with an Eulerian force distribution. This Eulerian force distribution is calculated by *spreading* a Lagrangian force density to the underlying fluid using the Dirac delta distribution. The position of the immersed structure is then automatically tracked in an *interpolation* phase, where the local fluid velocity is interpolated onto the immersed structure using the Dirac delta distribution. For numerical computations, a smoothed approximation of the delta distribution is required and the same approximation must be used for both the spreading and the interpolation phases.

The original IBM proposed by Peskin [3] is based on a couple of assumptions: the immersed structure is fibrous and the viscosity is constant throughout the computational domain. While the first assumption may be physically realistic in certain cases, the second assumption is in general not desirable. The immersed finite element method (IFEM) proposed by Zhang et al. [9] used finite elements for both the fluid and the immersed structure. Using finite elements for the structure alleviates the first assumption in the original IBM and allows for a more physically realistic representation of a thick immersed structure. Additionally, IFEM used the Reproducing Kernel Particle Method (RKPM) to construct an approximation to the Dirac delta distribution. The approximation used in the original IBM is C^1 continuous. However, the approximation constructed from RKPM is C^N continuous as RKPM allows the exact reconstruction of polynomials of degree N. The finite element immersed boundary method (FE-IBM), proposed by Boffi and Gastaldi [10], also overcomes the first assumption of the original IBM and this is the method adopted in this paper. Like IFEM, the FE-IBM uses finite elements for both the fluid and the immersed structure. However the key difference between the two methods is that FE-IBM does not numerically approximate the Dirac delta distribution. Instead, the interaction is governed within a weak formulation using the action of the delta distribution on a test function and using the sifting property of the delta function. Both the IFEM and FE-IBM, suffer from the limitation of constant viscosity throughout the computational domain.

In this paper, we apply a high-order method to the FE-IBM, which we call the spectral element immersed boundary method (SE-IBM). The aim of using a high-order method is to improve the accuracy of the spreading and interpolation phases and thus improve the order of convergence of the velocity and pressure variables.

This paper is constructed as follows: Sections 2 and 3 are concerned with the fluid equations and the derivation of the FE-IBM. Section 4 discusses the spatial discretisation and Section 5 summarises the temporal stability properties of the SE-IBM. Section 6 illustrates the area conservation of the SE-IBM and finally Section 7 illustrates the application of the SE-IBM to some well known benchmark problems. Section 8 presents our conclusions and discusses avenues for future work.

2. Newtonian fluid

Let $\Omega_t^f, t \in (t_0, T]$ (where t_0 and T are the initial and final times respectively), be the time-dependent fluid domain. The equations governing the motion of an incompressible fluid flow can be characterised by the incompressible

Navier–Stokes equations

$$\rho \frac{D\boldsymbol{u}}{Dt} = -\nabla p + \eta_s \nabla^2 \boldsymbol{u} + \boldsymbol{f} \qquad \text{in } \Omega_t^f \tag{1a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega_t^f \tag{1b}$$

$$\boldsymbol{u} = \boldsymbol{u}_0 \qquad \text{in } \Omega_0^f \tag{1c}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \partial\Omega_t^f \tag{1d}$$

where ρ is the density, η_s is the constant fluid viscosity, $\boldsymbol{u}$ is the velocity field, p is the pressure, $\boldsymbol{f}$ is an Eulerian force density, $\boldsymbol{u}_0$ is the initial velocity field (at time $t = t_0$) and $\boldsymbol{u}_D$ is the Dirichlet boundary conditions.

In this article, we assume that inertia is negligible (so-called creeping flow). In such a situation, the Navier–Stokes equations (1) reduce to the steady Stokes equations:

$$-\eta_s \nabla^2 \boldsymbol{u} + \nabla p = \boldsymbol{f} \qquad \text{in } \Omega_t^f \tag{2a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega_t^f \tag{2b}$$

$$\boldsymbol{u} = \boldsymbol{u}_0 \qquad \text{in } \Omega_0^f \tag{2c}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \partial\Omega_t^f. \tag{2d}$$

Additionally, in Section 5 we consider the unsteady Stokes equations, which are a linearisation of the Navier–Stokes equations, and are given by:

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} - \eta_s \nabla^2 \boldsymbol{u} + \nabla p = \boldsymbol{f} \qquad \text{in } \Omega_t^f \tag{3a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega_t^f \tag{3b}$$

$$\boldsymbol{u} = \boldsymbol{u}_0 \qquad \text{in } \Omega_0^f \tag{3c}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \partial\Omega_t^f. \tag{3d}$$

3. Finite element immersed boundary method

The need to construct an approximation to the delta function is a disadvantage of the original IB method [3,8,11]. Even though the work of Zhang et al. [9] allows for a higher-order approximation, it still requires several additional steps: the construction of the window function, the construction of the correction function and the construction of the domain of influence. Boffi et al. [10] defined the delta function variationally through its action on a test function. This removed the need to construct a suitably regularised approximation to the delta function and naturally allowed for a finite element type approach. This idea was later extended by the same authors to allow for thick immersed materials [12].

Let $\Omega_r \in \mathbb{R}^m$ be the reference configuration of both the fluid and the immersed structure, whose domains are denoted by $\Omega_r^f \subseteq \Omega_r$ and $\Omega_r^s \subseteq \Omega_r$ respectively, such that $\bar{\Omega}_r = \bar{\Omega}_r^f \cup \bar{\Omega}_r^s$, where $\Omega_r^f \in \mathbb{R}^m$ and $\Omega_r^s \in \mathbb{R}^d$, $d \leq m$. In the usual continuum mechanical way, we define a sequence of configurations of our domain in time with the set $\Omega := \{\Omega_t : t \in (t_0, T]\}$ where Ω_t denotes the configuration at time t, such that $\bar{\Omega}_t = \bar{\Omega}_t^f \cup \bar{\Omega}_t^s$ where $\Omega_t^f \subseteq \Omega_t$ and $\Omega_t^s \subseteq \Omega_t$ denote the fluid and structure domains at time t, respectively. Define the pointwise map $\boldsymbol{X}\colon \Omega_r \times (t_0, T] \to \Omega$ from the reference configuration to a current configuration. Following Boffi et al. [12], we let $\boldsymbol{s} \in \Omega_r$ denote the reference (or Lagrangian) variable, which labels a material point in Ω_r, and $\boldsymbol{x} \in \Omega_t$ the Eulerian variable. Therefore, for each material point $\boldsymbol{s} \in \Omega_r$ at time t, there exists $\boldsymbol{x} = \boldsymbol{X}(\boldsymbol{s}, t) \in \Omega_t$. The map $\boldsymbol{X}$ is used to denote the Cartesian position of a structure particle in the Eulerian domain. In this paper the computational domain does not change in time so we write $\Omega \equiv \Omega_t, \forall t \in (t_0, T]$ even though Ω_t^f and Ω_t^s can change with time.

The structure is considered to be part of the fluid where additional forces (and/or mass) act. Boffi et al. [12] assumed a Maxwell model for the immersed structure where the fluid component (dashpot) was assumed to be of the same form

as the surrounding fluid so that the Cauchy stress tensor can be written as:

$$\boldsymbol{\sigma} := \begin{cases} \boldsymbol{\sigma}_f & \text{in } \Omega_t^f \\ \boldsymbol{\sigma}_f + \boldsymbol{\sigma}_s & \text{in } \Omega_t^s \end{cases} \tag{4}$$

where $\boldsymbol{\sigma}_f = -p\boldsymbol{I} + 2\eta_s \dot{\boldsymbol{\gamma}}$ is the Cauchy stress tensor for a Newtonian fluid and $\dot{\boldsymbol{\gamma}} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$ is the rate-of-strain tensor. Applying, the principle of virtual work to the momentum equation, using integration by parts and separating the region $\mathcal{P}_t$ into its fluid and structure parts, Boffi et al. [12] showed that:

$$\begin{aligned} \int_{\mathcal{P}_t} (\nabla \cdot \boldsymbol{\sigma}) \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} = &- \int_{\mathcal{P}_t} \boldsymbol{\sigma}_f : \nabla \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \int_{\partial \mathcal{P}_t} (\boldsymbol{n}^T \cdot \boldsymbol{\sigma}_f) \cdot \boldsymbol{v} \, \mathrm{d}a \\ &- \int_{\mathcal{P}_r \cap \Omega_r^s} \mathbb{P} : \nabla_s \boldsymbol{v}(\boldsymbol{X}(\boldsymbol{s}, t)) \, \mathrm{d}\boldsymbol{s} + \int_{\partial \mathcal{P}_r \cap \Omega_r^s} (\boldsymbol{N}^T \cdot \mathbb{P}) \cdot \boldsymbol{v}(\boldsymbol{X}(\boldsymbol{s}, t)) \, \mathrm{d}A \end{aligned} \tag{5}$$

where $\boldsymbol{b}$ is the body force, $\boldsymbol{v}$ is the virtual displacement, $\mathcal{P}_t := \boldsymbol{X}(\mathcal{P}_r, t)$ is an arbitrary region of our current configuration $\mathcal{P}_t \subseteq \Omega_t$, integrals over $\mathcal{P}_t \cap \Omega_t^s$ have been mapped to the reference configuration $\mathcal{P}_r \cap \Omega_r^s$, $\partial \mathcal{P}_t$ is the boundary of $\mathcal{P}_t$, $\boldsymbol{n}$ is the unit outward normal to the boundary of $\mathcal{P}_t$, $\boldsymbol{N}$ is the outward normal to the boundary of $\mathcal{P}_r$ defined by $\boldsymbol{n}\mathrm{d}a = J\mathbb{F}^{-T}\boldsymbol{N}\mathrm{d}A$, where $\mathrm{d}a$ is an element of area in current configuration and $\mathrm{d}A$ is an element of area in reference configuration, and finally, $\mathbb{P}$ is the first Piola–Kirchhoff stress tensor defined by $\mathbb{P}(\boldsymbol{s}, t) = J\boldsymbol{\sigma}(\boldsymbol{X}(\boldsymbol{s}, t), t)\mathbb{F}^{-T}(\boldsymbol{s}, t)$, where $J = \det \mathbb{F}(\boldsymbol{s}, t)$ and $\mathbb{F}(\boldsymbol{s}, t) := \nabla_s \boldsymbol{X}(\boldsymbol{s}, t)$ is the deformation gradient tensor. The sifting property of the delta function is defined by:

$$\int_{\mathcal{P}_t} \boldsymbol{v}(\boldsymbol{x}) \boldsymbol{\delta}(\boldsymbol{x} - \boldsymbol{X}(\boldsymbol{s}, t)) \, \mathrm{d}\boldsymbol{x} = \begin{cases} \boldsymbol{v}(\boldsymbol{X}(\boldsymbol{s}, t)) & \text{if } \boldsymbol{X}(\boldsymbol{s}, t) \in \mathcal{P}_t \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Boffi et al. [12] defined the delta function variationally by choosing $\boldsymbol{v}(\boldsymbol{x})$ to be the virtual displacement used in the principle of virtual work (5). Circumventing the need to discretise the delta function explicitly means that a regularised approximation is not required. Substituting the above, interchanging the order of integration and employing some simple identities yields the interaction terms:

$$\boldsymbol{f}(\boldsymbol{x}, t) = \int_{\mathcal{P}_r} (\nabla_s \cdot \mathbb{P}) \delta(\boldsymbol{x} - \boldsymbol{X}(\boldsymbol{s}, t)) \, \mathrm{d}\boldsymbol{s}, \qquad \boldsymbol{t}(\boldsymbol{x}, t) = -\int_{\partial \mathcal{P}_r} (\boldsymbol{N}^T \cdot \mathbb{P}) \delta(\boldsymbol{x} - \boldsymbol{X}(\boldsymbol{s}, t)) \, \mathrm{d}A.$$

Since we are neglecting inertia, the full system of equations written in strong form is given by (2) with $\boldsymbol{f}$ replaced by $\boldsymbol{f} + \boldsymbol{t}$.

The FE-IBM has fewer additional components when compared to IFEM [9]; the IFEM requires the construction of the discrete delta function, the so-called domain of influence, which is the set of fluid nodal points in a certain region around a given structure marker particle, and the location of the structure marker particles within the Eulerian mesh. In contrast to the IFEM [9], the FE-IBM does not require an approximation to the delta function to be constructed, the domain of influence is simply an element of the Eulerian mesh and so the only additional computational work involves locating the structure marker particles.

4. Spatial discretisation

In this paper the spectral element method (SEM) [13] has been applied to the fluid variables in the IBM to improve the accuracy of the spreading and interpolation phases. It is well-known that the SEM *should* perform better than traditional finite elements both in terms of accuracy and efficiency provided the solution is sufficiently regular and the tolerance is sufficiently small.

4.1. Weak formulation

The spectral element method is based on solving the governing equations in their equivalent weak form. Let $\Omega \subset \mathbb{R}^2$ and define the velocity and pressure spaces, respectively, to be

$$\mathcal{V} = [H_0^1(\Omega)]^2, \qquad \mathcal{Q} = L_0^2(\Omega).$$

The weak formulation of the problem, in the absence of the immersed boundary, given in (2) is: find $(\boldsymbol{u}, p) \in \mathcal{V} \times \mathcal{Q}$ such that

$$\eta_s a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) = L(\boldsymbol{v}) \qquad \forall \boldsymbol{v} \in \mathcal{V} \tag{7a}$$

$$b(\boldsymbol{u}, q) = 0 \qquad \forall q \in \mathcal{Q} \tag{7b}$$

where

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \nabla \boldsymbol{u} \colon \nabla \boldsymbol{v} \, \mathrm{d}\Omega, \qquad b(\boldsymbol{v}, q) = \int_\Omega q \nabla \cdot \boldsymbol{v} \, \mathrm{d}\Omega, \qquad L(\boldsymbol{v}) = \langle \boldsymbol{f}, \boldsymbol{v} \rangle = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}\Omega. \tag{8}$$

The function $\boldsymbol{f}$ is assumed, at this point, to be a source term which is treated explicitly. This source term may contain body forces such as gravity and any forces due to an immersed structure via the IBM — the IBM term will be considered in Section 4.3.

4.2. Spatial discretisation

In the spatial discretisation of the weak formulation (7) using the spectral element method, it is necessary to choose conforming discrete subspaces $\mathcal{V}_N \subset \mathcal{V}$ and $\mathcal{Q}_N \subset \mathcal{Q}$. The domain Ω is divided into K uniform, non-overlapping, quadrilateral spectral elements Ω_e, $e = 1, \ldots, K$. Let $P_N(\Omega_e)$ denote the space of all polynomials on Ω_e of degree less than or equal to N and define:

$$\mathrm{P}_N(\Omega) \coloneqq \left\{ \phi \colon \phi|_{\Omega_e} \in P_N(\Omega_e) \right\}. \tag{9}$$

The velocity and pressure approximation spaces are then be defined, respectively, by

$$\mathcal{V}_N = \mathcal{V} \cap [\mathrm{P}_N(\Omega)]^2, \qquad \mathcal{Q}_N \coloneqq \mathcal{Q} \cap \mathrm{P}_{N-2}(\Omega).$$

These spaces are compatible and hence the discrete problem is well-posed [14,15]. Although the velocity approximation space does not comprise vector fields that are identically divergence free, the incompressibility constraint is enforced weakly.

The discrete weak formulation of (7) is: find $(\boldsymbol{u}_N, p_N) \in \mathcal{V}_N \times \mathcal{Q}_N$ such that

$$\eta_s a(\boldsymbol{u}_N, \boldsymbol{v}_N) + b(\boldsymbol{v}_N, p_N) = L(\boldsymbol{v}_N) \qquad \forall \boldsymbol{v}_N \in \mathcal{V}_N \tag{10a}$$

$$b(\boldsymbol{u}_N, q_N) = 0 \qquad \forall q_N \in \mathcal{Q}_N \tag{10b}$$

where the integrals in the bilinear forms (8) are computed using Gauss–Lobatto–Legendre (GLL) quadrature.

4.3. Immersed boundary

Predominantly, throughout the IB literature, the discretisation method of choice was finite differences. In the IFEM of Zhang et al. [9] and the FE-IBM of Boffi et al. [10,12], the fluid and the immersed structure are both discretised using finite elements. In this article, the fluid is discretised using spectral elements. However, we use a piecewise linear finite element approximation of an immersed membrane and a linear spectral element approximation of an immersed shell. Linear approximations are chosen to compare our results with those of Boffi et al. [12].

Analogous to Section 4.1, we require the definition of suitable function spaces for the unknown variables. For the immersed boundary terms, there is only a single unknown: $\boldsymbol{X}$, which gives the Cartesian position of the immersed structure. The function $\boldsymbol{X}$ is a map from the reference (Lagrangian) configuration of the immersed structure, Ω_r^s, to its current (Eulerian) configuration, $\Omega_t^s \subseteq \Omega$, at time t; i.e. $\boldsymbol{X} \colon \Omega_r^s \to \Omega$.

When a one-dimensional elastic curve immersed in a two-dimensional fluid is considered, a linear finite element discretisation is employed. Following the ideas of Boffi et al. [10,12] we assume that $\boldsymbol{X}$ is Lipschitz continuous, which is a stronger condition than regular continuity. Therefore, we define the function space to be:

$$\mathcal{X} = W^{1,\infty}(\Omega_r^s)^d = \left\{ \boldsymbol{Y} \in L^\infty(\Omega_r^s)^d \colon \nabla_{\boldsymbol{s}} \boldsymbol{Y} \in L^\infty(\Omega_r^s)^{d \times d} \right\} \tag{11}$$

where $L^\infty(\Omega_r^s)$ is the space of all functions that are essentially bounded on Ω_r^s. As the domain Ω_r^s is convex, the space $W^{1,\infty}(\Omega_r^s)$ is equivalent to the space of Lipschitz continuous functions on Ω_r^s (Brenner and Scott [16, p. 43]).

Consider a finite dimensional subspace $\mathcal{X}_h \subset \mathcal{X}$, where the subscript h is used to emphasise the use of finite elements rather than spectral elements. Let $\Omega^s_{r,h}$ be a subdivision of Ω^s_r into segments $[s_{j-1}, s_j]$, $j = 1, \ldots, K_b$, known as the triangulation. Let s_j, $j = 1, \ldots, K_v$, and $\Omega^s_k = [s_{k-1}, s_k]$, $k = 1, \ldots, K_b$, denote the vertices and elements of the triangulation, respectively. Let $\mathcal{X}_h$ denote the finite element space of piecewise linear functions on $\Omega^s_{r,h}$ as follows:

$$\mathcal{X}_h = \left\{ \boldsymbol{Y} \in C^0(\Omega^s_{r,h};\ \Omega)^d :\ \boldsymbol{Y}|_{\Omega^s_k} \in P_1(\Omega^s_k)^d,\ k = 1, \ldots, K_b \right\} \tag{12}$$

where $P_1(\Omega^s_k)$ denotes the set of affine polynomials of degree less than or equal to one on Ω^s_k, $k = 1, \ldots, K_b$. The finite element approximation, $\boldsymbol{X}_h$, of a function $\boldsymbol{X}$ at time $t^n = n\Delta t$ is then given by:

$$\boldsymbol{X}^n_h = \sum_{j=1}^{K_v} \boldsymbol{X}^n_j N_j(s) \tag{13}$$

where N_j are the usual piecewise linear finite element shape functions.

When a 2D elastic structure immersed in a 2D fluid is considered, linear spectral elements are used. Define the function space $\mathcal{X}$ as in (11). Following Section 4.2, it is necessary to choose a conforming discrete subspace $\mathcal{X}_N \subset \mathcal{X}$. The domain Ω^s_r is divided into K_b (approximately) uniform, non-overlapping, quadrilateral spectral elements, Ω^s_e, $e = 1, \ldots, K_b$. Let $P_{N_b}(\Omega^s_e)$ denote the space of all polynomials on Ω^s_e of degree less than or equal to N_b and let $\mathrm{P}_{N_b}(\Omega^s_r)$ be defined as in (9) for $N = N_b$. Following Section 4.2, the approximation space for the IB position is defined as:

$$\mathcal{X}_N \coloneqq \mathcal{X} \cap [\mathrm{P}_{N_b}(\Omega^s_r)]^d. \tag{14}$$

Each spectral element is mapped to the parent domain D, using a transfinite map [17]. The approximation of the IB position at time $t^n = n\Delta t$ on the element Ω^s_e is then given by:

$$\boldsymbol{X}^{e,n}_N(\xi, \eta) = \sum_{i=0}^{N_b} \sum_{j=0}^{N_b} \boldsymbol{X}^{e,n}_{i,j} h_i(\xi) h_j(\eta) \tag{15}$$

where $h_i(\xi)$, $i = 0, \ldots, N_b$, are the Lagrange interpolants [18]. Only linear polynomials are considered here so that $N_b = 1$. Higher-order approximations for the position of the immersed boundary will be considered in future.

In the implementation of the SE-IBM, the *interpolation* phase does not require the evaluation of any integrals since the delta function is treated variationally. Instead, once the location of the immersed boundary position $\boldsymbol{X}$ within the Eulerian mesh is determined, the inverse of a transfinite map [17] is used to determine the coordinates (ξ, η), within the parent domain D, corresponding to the position $\boldsymbol{X}$. Standard spectral element interpolation of the velocity field yields $\boldsymbol{u}(\boldsymbol{X}(\boldsymbol{s}, t), t)$ which is then used to update the location of the immersed boundary. Note that when we refer to the approximation of the delta function or indeed, the approximation of the sifting property of the delta function, it is the above procedure which is performed and a regularised approximation to the delta function is never constructed.

However, the *spreading* phase does require the evaluation of integrals, even though the delta function is defined variationally. In the weak form, we have

$$\langle \boldsymbol{f}, \boldsymbol{v} \rangle = -\int_{\Omega^s_r} \mathbb{P} :\ \nabla_{\boldsymbol{s}} \boldsymbol{v}(\boldsymbol{X}(\boldsymbol{s}, t))\ \mathrm{d}\boldsymbol{s} \tag{16}$$

where $\mathbb{P} = \kappa \mathbb{F} = \kappa \nabla_{\boldsymbol{s}} \boldsymbol{X}$. As mentioned above, the position $\boldsymbol{X}$ is linear and therefore, the deformation gradient tensor $\mathbb{F}$ is constant. It was shown by Boffi et al. [19], for the case of a 1D elastic structure immersed in a 2D fluid, that the integration in the IB source term does not require the use of quadrature.

However, the integration is not as simple for 2D immersed structures and quadrature rules are generally required. As shown by Boffi et al. [12], Eq. (16) can be written as

$$\langle \boldsymbol{f}, \boldsymbol{v} \rangle = -\sum_{e=1}^{K_b} \int_{\Omega^s_e} \mathbb{P}_{\alpha,i} \frac{\partial v_\alpha(\boldsymbol{X})}{\partial s_i}\ \mathrm{d}\boldsymbol{s} = -\sum_{e=1}^{K_b} \int_{\partial \Omega^s_e} \mathbb{P}_{\alpha,i} N_i v_\alpha(\boldsymbol{X})\ \mathrm{d}\boldsymbol{s}.$$

The boundary of each element is split into its four edges and the integral along each edge is computed using 1D Gauss–Lobatto–Legendre (GLL) quadrature on N_q GLL points. As $N_b = 1$, we choose $N_q > N_b$ so that the

numerical quadrature along each edge of an element, is computed with higher accuracy. This means we need to determine additional points $\boldsymbol{X}_q$, $q = 1, \ldots, N_q - 1$, which are required for the computation of $\boldsymbol{v}(\boldsymbol{X})$. Fortunately, as the immersed structure is assumed to be linear, determining the additional points $\boldsymbol{X}_q$ is straightforward and can be accomplished using a transfinite map [17] where the vertices of an edge are the extreme quadratures points $\boldsymbol{X}_0$ and $\boldsymbol{X}_{N_q}$.

5. Temporal discretisation & stability

Throughout the IBM literature it has been documented that the method can be stiff and therefore a small timestep may be required in order to maintain stability of the method [19–25]. The common approach for the discretisation of the momentum equation (1a) is the so-called Forward Euler/Backward Euler (FE/BE) method. In the FE/BE method, the viscous and pressure terms are dealt with implicitly (i.e. at time level t^{n+1}) whilst the source term is treated explicitly (i.e. at time level t^n so that $\boldsymbol{f}^{n+1} \approx \boldsymbol{f}^n$). The evolution of the immersed membrane or shell, which is governed by the differential equation:

$$\frac{\partial \boldsymbol{X}_h}{\partial t} = \boldsymbol{u}_N(\boldsymbol{X}_h(\boldsymbol{s}, t), t) = \int_\Omega \boldsymbol{u}_N(\boldsymbol{x}, t)\delta(\boldsymbol{x} - \boldsymbol{X}_h(\boldsymbol{s}, t))\, \mathrm{d}\boldsymbol{x} \tag{17}$$

is usually discretised using the semi-implicit Euler method:

$$\frac{\boldsymbol{X}_h^{n+1}(\boldsymbol{s}) - \boldsymbol{X}_h^n(\boldsymbol{s})}{\Delta t} = \boldsymbol{u}_N^{n+1}(\boldsymbol{X}_h^n(\boldsymbol{s})). \tag{18}$$

In the FE/BE method, the above evolution equation is decoupled from the momentum equation. It is clear that the method is first order in time. Typically the restriction on time step is caused by a combination of diffusive and fibre effects and is therefore most severe for cases with a large force or small viscosity [22,23]. Stockie and Wetton [22] showed that in the case of large fibre force or small viscosity the maximum allowable time step can be as small as 10^{-5}. In order to increase the maximum allowable time step, the immersed boundary terms need to be treated more implicitly. Indeed, Peskin [11] proposed using an approximate-implicit method for the immersed boundary terms. Tu and Peskin [21] compared this approximate-implicit method against explicit and implicit approaches. It was found that the explicit method, as expected, broke down spectacularly at intermediate time steps but that the approximate-implicit method performed almost as well as the fully implicit method. We note that in the fully implicit method, the source term is dealt with implicitly (i.e. at time level t^{n+1}) so that the evolution of the immersed boundary is fully coupled with the momentum equation. Unfortunately, the fully implicit method was deemed impractical due to the large computational times (the full Navier–Stokes equations has to be solved at each time step) and therefore the approximate-implicit method was deemed to be a viable alternative. However, the problem with such an approach is that it not only increases the computational time of the method, effectively destroying any advantage that would be achieved using a larger time step but also makes the method considerably more complicated to programme. Additionally, the larger value of Δt increased the amount of area/volume loss in the computation [22].

For the FE-IBM, Boffi et al. [19] compared the standard FE/BE method (that is, the source term is calculated explicitly causing the evolution equation to be decoupled from the momentum equation) against an implicit BE/BE method (that is, the fully coupled system where the source term is treated implicitly). They found that the BE/BE method (or fully implicit scheme) was unconditionally stable although completely impractical. Newren et al. [25] also considered a fully implicit formulation and found unconditional stability for the original IBM. For the FE/BE method, Boffi et al. [20] determined a CFL condition for the FE-IBM.

As a high-order spatial approximation of the fluid variables is used, a high-order temporal scheme is desirable. Due to a fully implicit scheme being computationally impractical, we consider the standard FE/BE method (that is, the source term is treated explicitly) whilst treating the immersed boundary evolution equation more implicitly. As far as we are aware, the use of a higher-order method for the evolution of the immersed boundary has not yet been considered in the IB literature. We compared the use of a semi-implicit Euler (SIME) method (18), an implicit Euler (IME) method:

$$\frac{\boldsymbol{X}_h^{n+1}(\boldsymbol{s}) - \boldsymbol{X}_h^n(\boldsymbol{s})}{\Delta t} = \boldsymbol{u}_N^{n+1}(\boldsymbol{X}_h^{n+1}(\boldsymbol{s})) \tag{19}$$

Table 1
Values for the coefficients α and β, at time levels n and $n+1$, and C for each scheme.

Scheme	α^{n+1}	β^{n+1}	α^{n}	β^{n}	C
SIME	1	1	1	1	2
IME	1	1	1	1	2
BDF3	$\frac{1}{2}$	$\frac{11}{12}$	$\frac{1}{2}$	$\frac{27}{11}$	1

and a third-order backward differentiation formula (BDF3):

$$\boldsymbol{X}_h^{n+1} - \frac{18}{11}\boldsymbol{X}_h^{n} + \frac{9}{11}\boldsymbol{X}_h^{n-1} - \frac{2}{11}\boldsymbol{X}_h^{n-2} = \frac{6\Delta t}{11}\,\boldsymbol{u}_N^{n+1}(\boldsymbol{X}_h^{n+1}(\boldsymbol{s})) \tag{20}$$

to determine the effect on stability and time step size. The implicit IME and BDF3 methods are computed using a predictor–corrector approach by iterating over an internal time-step. Note that a fourth-order Runge–Kutta is used to generate the starting values for the BDF3 method. Explicit methods are not considered since they are known to be unstable [21].

To assess stability, we use a similar definition for the energy of the system that was used by Newren et al. [25] and Boffi et al. [19]. The only difference being that, in this article, we include coefficients α^j and β^j, $j = n, n+1$, which are scheme dependent. Thus, the energy at time $t^n = n\Delta t$ is defined as:

$$E^n(\boldsymbol{u}, \boldsymbol{X}) = E(\boldsymbol{u}^n, \boldsymbol{X}^n) = \alpha^n \left\|\boldsymbol{u}^n\right\|^2_{L^2(\Omega)} + \beta^n \left(\boldsymbol{X}^n, -\kappa\frac{\partial^2 \boldsymbol{X}^n}{\partial s^2}\right)_{L^2(\Gamma)} \tag{21}$$

where the coefficients, α^n and β^n, are taken to be unity (for each time level n) in all cases except when the third-order backward differentiation formula is used. The procedure proposed by Newren et al. [25] is used to derive the energy estimates. Thus, the dependence of α and β on time level n and $n+1$ is a consequence of applying this procedure to the third-order backward differentiation formula. Note that when α^j and β^j are unity, the definition of energy in the system is identical to that used by Newren et al. [25] and Boffi et al. [19]. Although the expression for the energy (21) is for a particular choice of elastic model, it is a popular choice within the literature [3,8,11,12,25,26]. In an unconditionally stable scheme, the energy must satisfy $E^{n+1}(\boldsymbol{u}, \boldsymbol{X}) - E^n(\boldsymbol{u}, \boldsymbol{X}) \leq 0$. However, for a conditionally stable scheme, additional energies involving the time step Δt lead to estimates of the form:

$$E^{n+1}(\boldsymbol{u}, \boldsymbol{X}) - E^n(\boldsymbol{u}, \boldsymbol{X}) - C\kappa\,\Delta t\, F(\boldsymbol{u}^n, \boldsymbol{u}^{n+1}, \boldsymbol{X}^{n-2}, \boldsymbol{X}^{n-1}, \boldsymbol{X}^n, \boldsymbol{X}^{n+1}) \leq 0 \tag{22}$$

where C is a scheme-dependent constant, and the function F can depend on $\boldsymbol{u}$ and $\boldsymbol{X}$, at times t^j, $j = n-2, \ldots, n+1$. Note that the function F in (22) is scheme dependent. Also, we note that the important contribution to the estimate is the energy difference $E^{n+1}(\boldsymbol{u}, \boldsymbol{X}) - E^n(\boldsymbol{u}, \boldsymbol{X})$, as it indicates when the system is becoming unstable. Table 1 provides the values for α^{n+1}, β^{n+1}, α^n, β^n, and C for each scheme. The function F for the semi-implicit Euler (SIME) scheme is given by:

$$F = \sum_{i=1}^{K_b} \left[\frac{\boldsymbol{X}_{h,i}^{n+1} - \boldsymbol{X}_{h,i-1}^{n+1}}{\Delta s_i} - \frac{\boldsymbol{X}_{h,i}^{n} - \boldsymbol{X}_{h,i-1}^{n}}{\Delta s_i}\right] \left(\boldsymbol{u}_N^{n+1}(\boldsymbol{X}_{h,i}^n) - \boldsymbol{u}_N^{n+1}(\boldsymbol{X}_{h,i-1}^n)\right) \tag{23}$$

for the implicit Euler (IME) scheme:

$$\begin{aligned} F = &\left\{\sum_{i=1}^{K_b} \left(\frac{\boldsymbol{X}_{h,i}^{n+1} - \boldsymbol{X}_{h,i-1}^{n+1}}{\Delta s_i}\right) \left(\boldsymbol{u}_N^{n+1}(\boldsymbol{X}_{h,i}^{n+1}) - \boldsymbol{u}_N^{n+1}(\boldsymbol{X}_{h,i-1}^{n+1})\right)\right. \\ &\left. - \sum_{i=1}^{K_b} \left(\frac{\boldsymbol{X}_{h,i}^{n} - \boldsymbol{X}_{h,i-1}^{n}}{\Delta s_i}\right) \left(\boldsymbol{u}_N^{n+1}(\boldsymbol{X}_{h,i}^{n}) - \boldsymbol{u}_N^{n+1}(\boldsymbol{X}_{h,i-1}^{n})\right)\right\} \end{aligned} \tag{24}$$

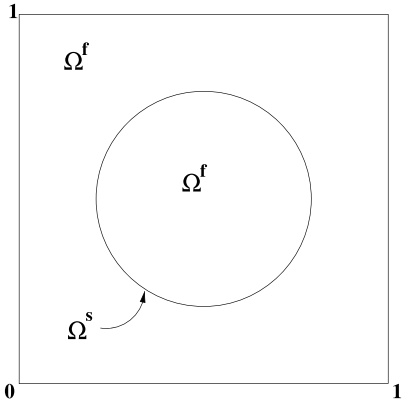


Fig. 1. Static closed membrane immersed in a fluid.

and finally for the third-order backward differentiation formula (BDF3):

$$
\begin{aligned}
F = \Bigg\{ & \sum_{i=1}^{K_b} \left(\frac{\boldsymbol{X}_{h,i}^{n+1} - \boldsymbol{X}_{h,i-1}^{n+1}}{\Delta s_i} \right) \left(\boldsymbol{u}_N^{n+1}(\boldsymbol{X}_{h,i}^{n+1}) - \boldsymbol{u}_N^{n+1}(\boldsymbol{X}_{h,i-1}^{n+1}) \right) \\
& - \sum_{i=1}^{K_b} \left(\frac{\boldsymbol{X}_{h,i}^{n} - \boldsymbol{X}_{h,i-1}^{n}}{\Delta s_i} \right) \left(\boldsymbol{u}_N^{n+1}(\boldsymbol{X}_{h,i}^{n}) - \boldsymbol{u}_N^{n+1}(\boldsymbol{X}_{h,i-1}^{n}) \right) \\
& - \frac{3}{2} \sum_{i=1}^{K_b} \frac{\left(\boldsymbol{X}_{h,i}^{n-1} - \boldsymbol{X}_{h,i-1}^{n-1} \right) \left(\boldsymbol{X}_{h,i}^{n+1} - \boldsymbol{X}_{h,i-1}^{n+1} \right)}{\Delta s_i} \\
& + \frac{1}{3} \sum_{i=1}^{K_b} \frac{\left(\boldsymbol{X}_{h,i}^{n-2} - \boldsymbol{X}_{h,i-1}^{n-2} \right) \left(\boldsymbol{X}_{h,i}^{n+1} - \boldsymbol{X}_{h,i-1}^{n+1} \right)}{\Delta s_i} \Bigg\}.
\end{aligned} \tag{25}
$$

Following Newren et al. [25], we consider the unsteady Stokes flow of an incompressible fluid (3) so that the weak form of the full discrete equations of motion is given by: find $(u_N, p_N) \in \mathcal{V}_N \times \mathcal{Q}_N$ such that

$$
\frac{1}{\Delta t} \left(\boldsymbol{u}_N^{n+1} - \boldsymbol{u}_N^{n}, \boldsymbol{v}_N \right)_{L^2(\Omega)} + \eta_s a(\boldsymbol{u}_N^{n+1}, \boldsymbol{v}_N) + b(\boldsymbol{v}_N, p_N^{n+1}) = L(\boldsymbol{v}_N) \qquad \forall \boldsymbol{v}_N \in \mathcal{V}_N \tag{26a}
$$

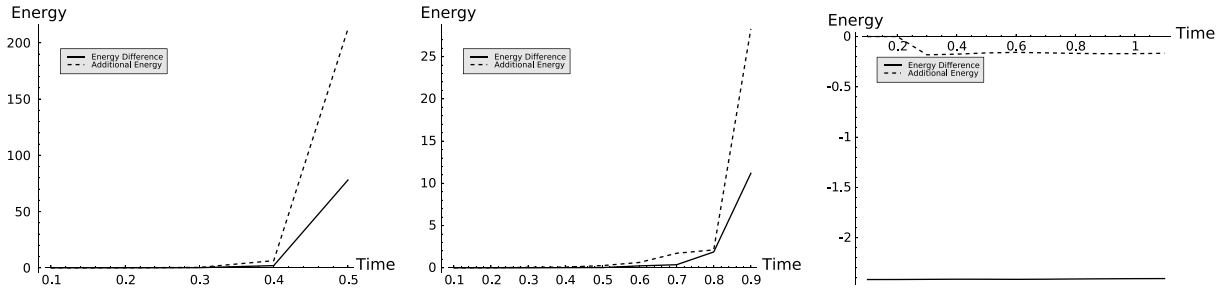
$$
b(\boldsymbol{u}_N^{n+1}, q_N) = 0 \qquad \forall q_N \in \mathcal{Q}_N \tag{26b}
$$

where $(\cdot, \cdot)_{L^2(\Omega)}$ is the L^2 inner product and we use the superscript n to denote the time level with $t^n = n\Delta t$.

Consider a closed circular membrane immersed in a Newtonian fluid. The configuration is shown in Fig. 1. Let $\Omega = [0, 1]^2$ such that $\bar{\Omega} = \bar{\Omega}_t^f \cup \bar{\Omega}_t^s$ where Ω_t^f denotes the fluid domain and Ω_t^s denotes the immersed elastic membrane at time t. As we are dealing with a membrane, we define $\Gamma_t = \Omega_t^s$. The Cartesian position of the membrane $\boldsymbol{X}$ is parameterised by arclength $s \in [0, 2\pi R]$ where $R = 0.25$ is the initial radius of the immersed circular membrane. For the time being we choose $\rho = 1$, $h_f = 1/8$, $N = 8$ and vary η_s, κ and Δt. The figures present the behaviour of both the energy difference and the additional energy (last term in (22)). Although the important quantity is the energy difference, as this indicates stability, the additional energy is included for completeness.

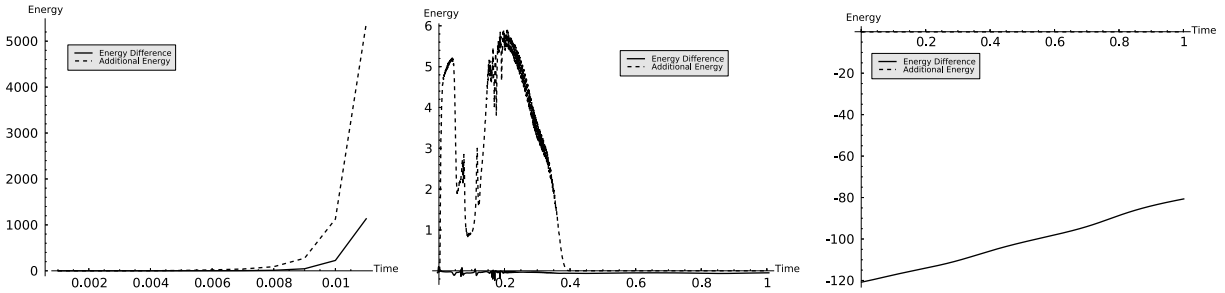
Fig. 2 provides a comparison between the SIME, IME and BDF3 schemes when $\eta_s = 1$, $\kappa = 1$ and $\Delta t = 0.1$. The IME method diverged at a slightly later time than the SIME method indicating that the use of an implicit immersed boundary evolution does have a (very) slight stabilising effect. However, the energy difference for the BDF3 scheme did not diverge, thus indicating stability. This is an improvement over the SIME and IME methods.

As the IBM is known to be unstable for large κ and small viscosity [21,22], we focus here on comparing each scheme at large κ and small viscosity values. For a more complete comparison of each scheme for a variety of values for κ and η_s, the reader is referred to Rowlatt [27]. Fig. 3 illustrates a comparison between each temporal scheme when $\Delta t = 0.001$, $\eta_s = 1$ and $\kappa = 50$. Clearly, the energy difference in the SIME method diverged quickly indicating



(a) SIME. (b) IME. (c) BDF3.

Fig. 2. Comparison of the SIME, IME and BDF3 schemes for $\kappa = 1$, $\eta_s = 1$ and $\Delta t = 0.1$.



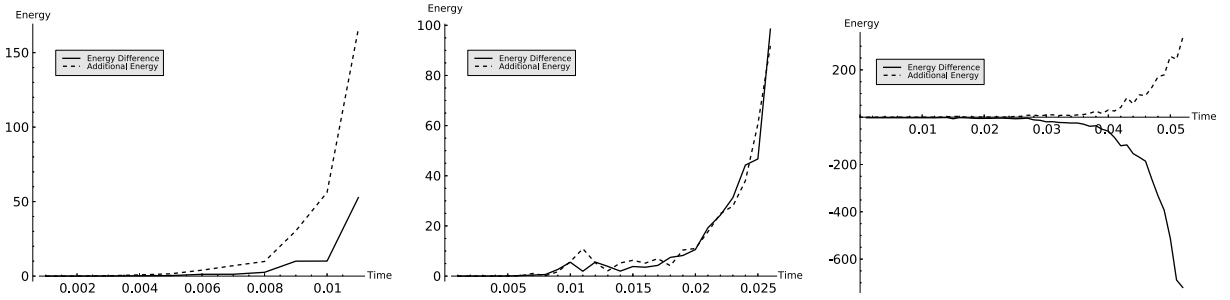
(a) SIME. (b) IME. (c) BDF3.

Fig. 3. Comparison of the SIME, IME and BDF3 schemes for $\kappa = 50$, $\eta_s = 1$.

instability. Additionally, we found that when κ was increased (but the time step length was chosen so the system remained stable) a large loss in area was seen. This was first reported by Newren et al. [25] who suggested that the area loss was a stabilising effect. The loss in area is a consequence of the large spurious velocities local to the membrane produced by large values of κ. The energy difference for the IME scheme clearly has some *noise* and we are unsure of its origins. However, it does not have an impact on the scheme and the simulation remained stable for these parameter values. Finally, the energy difference for the BDF3 scheme also did not diverge, thus indicating stability. However, the energy difference is steadily increasing and thus, it is possible that it would eventually diverge. We believe the large initial energy difference for the BDF3 scheme is caused by the need to generate starting values using a high-order one-step method.

Keeping the values of the time step length, mesh width and polynomial degree the same as above, we illustrate a comparison of the three methods for $\kappa = 1$ and $\eta_s = 0.001$. It can clearly be seen from Fig. 4 that both the SIME and IME methods are unstable. However, for the BDF3 scheme, the energy difference remained negative. Normally, this indicates stability but in this case the energy difference is diverging from around zero. As an energy difference of zero would indicate an equilibrium, this is clearly unstable behaviour.

Table 2 provides details of the dependence on the discretisation parameters on the stability characteristics of SE-IBM for the three methods. All of the schemes do not perform well for the smallest viscosity and for the largest time steps. However, it is evident that when $\kappa \leq 1$ and $\eta_s = 0.001$, larger values of Δt can be used with SIME when $N = 2$ compared with the other schemes. On the contrary, when $\kappa = 1$ and $\eta_s = 1$, larger time steps can be used with BDF3 when $N = 8$ compared with the other routines. These findings suggest that using a higher-order method for the evolution of the IB is advantageous when used in combination with a higher-order approximation of the fluid variables. Also we can see from Table 2 that BDF3 performs slightly better than the other routines when $\kappa = 50$ and $\eta_s = 1$. Table 2 indicates that when κ is fixed and η_s is increased (and also when η_s is fixed and κ is decreased) that stability generally improves. This suggests that the ratio between the viscosity and the IB stiffness is potentially more important than the individual values. Note, however, that the scales for κ and η_s are different. It is possible that an implicit treatment of the interpolation phase will allow for larger time steps for larger values of κ and that an implicit



(a) SIME. (b) IME. (c) BDF3.

Fig. 4. Comparison of the SIME, IME and BDF3 schemes for $\kappa = 1$, $\eta_s = 0.001$.

Table 2
Illustration of when a simulation for a given set of parameters has completed or stopped for each routine. ◯ means the SIME routine completed successfully, □ means the IME routine completed successfully and △ means the BDF3 routine completed successfully, × means none of the routines successfully finished.

η_s	Δt	$\kappa = 0.5$			$\kappa = 1.0$			$\kappa = 50.0$		
		$N = 2$	$N = 4$	$N = 8$	$N = 2$	$N = 4$	$N = 8$	$N = 2$	$N = 4$	$N = 8$
10^{-3}	10^{-1}	×	×	×	×	×	×	×	×	×
	10^{-2}	◯	×	×	◯	×	×	×	×	×
	10^{-3}	◯□△	◯□△	×	◯□△	◯□△	×	◯△	×	×
	10^{-4}	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	×
1.0	10^{-1}	◯□△	◯□△	◯□△	◯□△	◯□△	△	×	×	×
	10^{-2}	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	△	×	×
	10^{-3}	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	□△
	10^{-4}	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△
10.0	10^{-1}	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	△	×	×
	10^{-2}	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	□△
	10^{-3}	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△
	10^{-4}	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△	◯□△

treatment of the spreading phase will allow for larger time steps at smaller values of η_s. This is not entirely clear from Table 2 and is a subject of future research.

Although the use of an implicit method does improve the stiffness of the IBM, enabling larger time step lengths to be used, the improvement is not substantial. Similarly, although the use of a higher-order implicit method allowed larger time step lengths at higher values of polynomial degree, the improvement was not substantial. Consequently, only the semi-implicit discretisation (18) of the immersed boundary evolution equation is used in Sections 6 and 7.

6. Area conservation

The immersed boundary method is known to suffer from volume or area loss problems [12,25,26,28,29]. Peskin [3,8,11] noted that if a compatibility condition between the Eulerian and Lagrangian mesh widths is not satisfied then significant numerical diffusion takes place across the interface between the fluid and the structure. The compatibility condition is given by: $h^s \leq h_f$, where h^s is the Lagrangian mesh width and h_f the Eulerian mesh width. Boffi et al. [19] found that the optimal compatibility condition for FE-IBM was $h^s \leq h_f/2$. However, even if this condition is satisfied, volume/area loss can still be seen. The explanation given in the literature is that this is due primarily to the velocity field not being identically divergence-free and thus the incompressibility condition *not* is not satisfied accurately.

For the standard approach used in IB computations, the movement of the IB is decoupled from the solution of the momentum equation and incompressibility constraint. Thus, the velocity field *should* be discretely divergence-free if the incompressibility constraint is solved implicitly. However, the interpolated velocity, $\boldsymbol{u}(\boldsymbol{X}(\boldsymbol{s}, t), t)$, is unlikely to

be *exactly* discretely divergence-free due to the immersed boundary being unfitted. Additionally, the velocity field is most likely non-zero local to the interface due to the non-zero force which is *spread* to the surrounding fluid. Indeed, in the simple example considered below — and by many in the literature [12,21,25] – the action of the forcing term is to pull the elastic curve inwards. Therefore, without an additional constraint on the interface velocity, the immersed boundary is free to move inwards. Peskin [28] considered a modified delta function, and difference operators to improve the volume/area conservation but at an additional computational cost to the *spreading* and *interpolation* phases. Owens et al. [30] showed that increasing the number of points inside the support of the delta function, gave improved volume/area conservation. The Immersed Structural Potential Method [31] reduces the amount of area loss by reducing the number of times the spreading and interpolation operations are computed. Whilst this does improve area conservation, whether or not the area conservation is better than the FE-IBM is still an open question. The FE-IBM by-passes the need for an approximation to the delta function, thus reducing the amount of area loss due to error in the spreading and interpolation phases.

In this section, we investigate the volume/area loss of the SE-IBM considered in this article. We consider the standard model problem in which a circular static membrane is immersed in a Newtonian fluid. The problem is formulated as in Section 5 (see Fig. 1). In this geometry, the Cartesian position of the membrane, $\boldsymbol{X}$ is parameterised by arclength, $s \in [0, 2\pi R]$ where $R = 0.25$ is the initial radius of the immersed circular membrane. We use the forward Euler/backward Euler (FE/BE) discretisation of the momentum equation, which is standard in the immersed boundary literature, and use the SIME scheme (18) to move the position of the immersed membrane. The density of fluid and the structure is taken to be the same, $\rho = 1$, the viscosity of the fluid and structure $\eta_s = 1$ and the IB stiffness is taken to be $\kappa = 1$. As we are studying the effect of the spatial discretisation on the area conservation, we have taken the timestep to be sufficiently small so as to reduce the temporal error, $\Delta t = 10^{-4}$. We performed 1000 time steps and the analytical value of the area contained inside the membrane is given by 0.19635 (to 5 decimal places).

In the spatial discretisation of the immersed boundary position $\boldsymbol{X}$, the domain for the arclength parameter is divided into K_b segments, $[s_{i-1}, s_i]$, $i = 1, \ldots, K_b$, each of constant mesh width $h^s = 2\pi R/K_b$. There are two ways in which we can define the number of Lagrangian segments, K_b. The natural way of defining K_b is to define the constant fluid mesh width $h_f = 1/K_f$ where K_f is the number of fluid elements in the x or y direction. (As we are dealing with a uniform discretisation of Ω, the number of elements is the same in both the x and y directions). The value of K_b is then determined using:

$$h^s = \frac{2\pi R}{K_b} \leq \frac{h_f}{2} = \frac{1}{2K_f} \quad \Rightarrow \quad K_b \geq \frac{4\pi R}{h_f} = 4\pi R K_f. \tag{27}$$

The above condition, shows that the number of points on the immersed boundary is defined as a function of mesh width h_f, $K_b = K_b(h_f)$. As we are using a spectral approximation, we can also increase the polynomial degree. Therefore, the number of immersed boundary points must be chosen to be a function of N, $K_b = K_b(N)$. The largest distance between Gauss–Lobatto–Legendre points is approximately $1/N$. Therefore, given that there are K_f elements in the x and y directions, we assume that $h_f = 1/NK_f$ is the fluid mesh width (this is equivalent to assuming we have NK_f uniformly spaced elements in the x or y direction), i.e.

$$h^s = \frac{2\pi R}{K_b} \leq \frac{h_f}{2} = \frac{1}{2NK_f} \quad \Rightarrow \quad K_b \geq \frac{4\pi R}{h_f} = 4\pi RNK_f. \tag{28}$$

As the Gauss–Lobatto–Legendre points are not uniformly spaced, we could have chosen $h_f = 1/N^2K_f$ resulting in $K_b \geq 4\pi RN^2K_f$ which is larger than the value given above. The value given in (28) may not be the optimal choice but it is sufficient to illustrate that the value of K_b must be dependent on N when using spectral elements. As K_b is required to be an integer, we take the nearest *even* number which satisfies either (27) or (28).

The absolute error in the area and order of convergence with respect to mesh width (h-type convergence) at the end of the simulation, when $N = 2$ is fixed, are given in Table 3 when $K_b = K_b(h_f)$ and $K_b = K_b(N)$. The analytical area of the membrane is 0.19635. Approximately first-order convergence of the area with respect to mesh width is obtained for both $K_b(h_f)$ and $K_b(N)$. Although this is reasonable, we expect to achieve better order of convergence when we increase the polynomial degree.

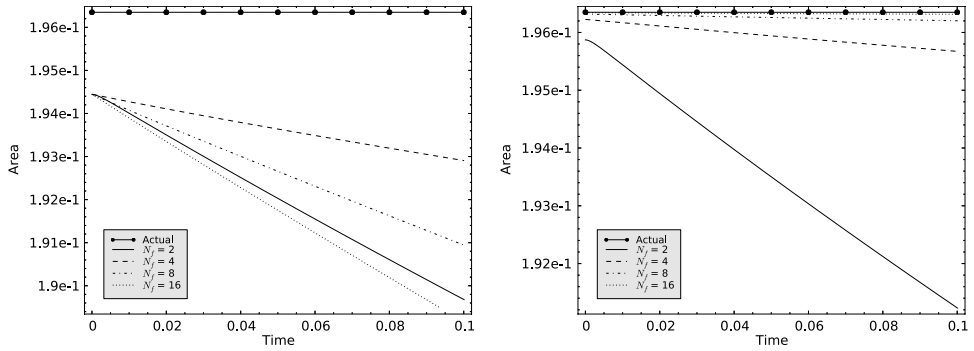
Table 4 details the absolute error in the area and order of convergence when $h_f = 1/2$ for increasing N (p-type convergence) for both $K_b = K_b(h_f)$ and $K_b(N)$. When $K_b = K_b(h_f)$ the method fails to converge with respect to N. However, approximately second order convergence is obtained when $K_b = K_b(N)$ which is an improvement on the

Table 3
Error and order of convergence of the area with respect to h_f for both $K_b(h_f)$ and $K_b(N)$ for fixed $N = 2$.

$K_b = K_b(h_f)$			$K_b = K_b(N)$		
h_f	Error	Order	h_f	Error	Order
1/2	0.04611	–	1/2	0.03402	–
1/4	0.01427	1.69273	1/4	0.00960	1.82584
1/8	0.00667	1.09619	1/8	0.00512	0.90613
1/16	0.00309	1.10877	1/16	0.00266	0.94604
1/32	0.00162	0.93103	1/32	0.00146	0.86730
1/64	0.00073	1.15752	1/64	0.00067	1.12957

Table 4
Error and order of convergence with respect to N for both $K_b = K_b(h_f)$ and $K_b(N)$ for fixed $h_f = 1/2$.

$K_b = K_b(h_f)$			$K_b = K_b(N)$		
N	Error	Order	N	Error	Order
2	0.04611	–	2	0.03402	–
4	0.02689	0.77820	4	0.00709	2.26199
8	0.02981	−0.14894	8	0.00150	2.24401
16	0.03391	−0.18563	16	0.00035	2.11087
32	0.03901	−0.20212	32	0.00008	2.06047



(a) $K_b = K_b(h_f)$. (b) $K_b = K_b(N)$.

Fig. 5. Comparison of the area conservation for different values of K_b, for increasing N when $h_f = 1/8$.

mesh width (h-type) convergence presented in Table 3. This means that the error in the area decreases with $O(1/N^2)$. In fact, second order convergence is the best we have obtained in our simulations.

Fig. 5 shows a comparison of area loss depending on the choice $K_b = K_b(h_f)$ or $K_b = K_b(N)$ where we fix $h_f = 1/8$ so that we have an 8×8 grid of elements. In Fig. 5(a), it can be seen that for $K_b = K_b(h_f)$, increasing the resolution of the Eulerian grid actually increases the area loss inside the circular membrane since the resolution of the Lagrangian grid has not been correspondingly improved — this is the reason for the erroneous order of convergence presented in Table 4. Fig. 5(b) confirms that better area conservation is obtained as we increase N provided $K_b = K_b(N)$ and illustrates the good order of convergence obtained in Table 4. Note that the area contained inside the polygon, which approximates the immersed boundary or membrane, is dependent on the value of K_b. The value of K_b is smaller when $K_b = K_b(h_f)$ than when $K_b = K_b(N)$ and this is the reason for the large initial error of the area in Fig. 5(a). Note that in this figure, $N_f = N$ denotes the polynomial degree used in the approximation of the fluid variables.

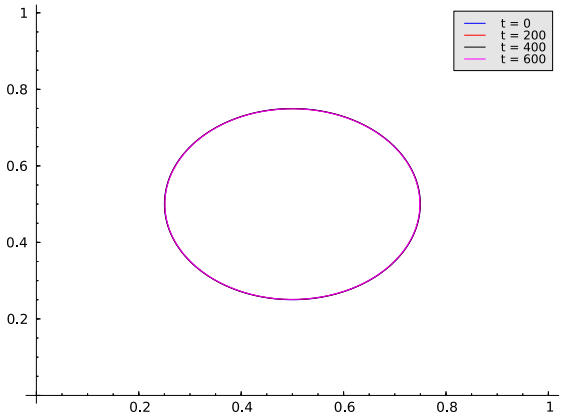


Fig. 6. IB evolution for $N = 32$ and $h_f = 1/2$.

Table 5
Value of the area at the end of the simulation for different parameter regimes.

Parameters	Final area (0.19635)	Computational time
$h_f = 1/64$, $N = 2$	0.19568	42 h
$h_f = 1/8$, $N = 8$	0.19620	3 h
$h_f = 1/2$, $N = 32$	0.19627	87 h

Fig. 6 illustrates the excellent area conservation that is achieved when high-order polynomials are used to approximate the fluid variables, even on very coarse meshes. The question is: Why do we see better order of convergence with respect to polynomial degree than with mesh width?

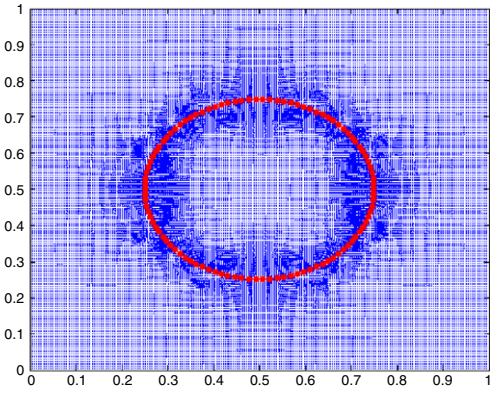
It was shown by Noutcheuwa and Owens [30] that the area loss can be improved by increasing the number of points inside the support of the delta function. In the FE-IBM, and therefore SE-IBM, there is no need to approximate the delta function. However, the spreading and interpolation phases require the discretisation of the sifting property of the delta function (see (6), for example). Hence, in the SE-IBM, the support of the sifting property of the delta function is just an element of the Eulerian fluid mesh which contains the immersed structure marker particles. Therefore, increasing the value of N increases the number of points inside an element of the mesh and hence, the support of the sifting property of the delta function. So not only do we achieve better area conservation due to the improved resolution of the interface (K_b is dependent on N) but also due to the increased number of points inside the support of the sifting property of the delta function, thus improving the accuracy of the spreading and interpolation phases.

Table 5 provides details of the area enclosed within the membrane at the end of the simulation and the computational time of the simulation, for $K_b = K_b(N)$ when $h_f = 1/64$ with $N = 2$, $h_f = 1/8$ with $N = 8$ and $h_f = 1/2$ with $N = 32$. From Table 5, it is clear that the best area conservation properties are obtained for $h_f = 1/2$ with $N = 32$ followed by $h_f = 1/8$ with $N = 8$ and $h_f = 1/64$ with $N = 2$. However, the improvement in the area loss is not significant. It is clear from Table 5, that the computational time required when $h_f = 1/64$ and $N = 2$ is significantly more than when $h_f = 1/8$ and $N = 8$. So not only does $h_f = 1/8$ with $N = 8$ give better area conservation than when $h_f = 1/64$ with $N = 2$ but it also takes a fraction of the computational time. However, the largest computational time occurred when $h_f = 1/2$ and $N = 32$. Therefore, although a very coarse mesh with a very high-order polynomial gives good area conservation, it is computationally very expensive. This suggests that using a relatively fine mesh with a medium polynomial degree is the best option.

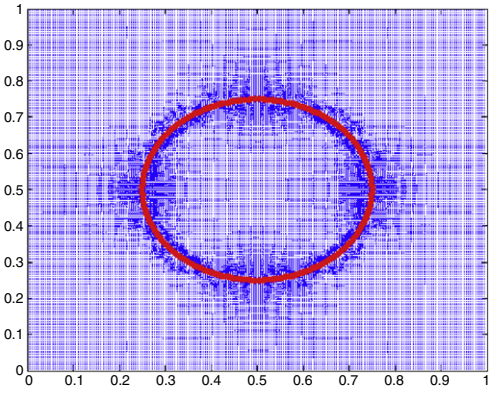
As the divergence-free constraint plays an important role in the area conservation, Table 6 details the L^2 norm of the divergence of the velocity and the order of convergence with respect to both mesh width (h-type) and polynomial degree (p-type). The order of convergence with respect to mesh width (h-type) is approximately 0.5, whereas it is approximately 1 with respect to N (p-type) when $h_f = 1/2$. The same p-type convergence behaviour was found when $h_f = 1/8$. At the beginning of this section, we mentioned that the immersed boundary force acts to pull the immersed boundary inwards and, because there is no additional constraint on the velocity on the interface, the

Table 6
Comparison of the orders of convergence for $\|\nabla \cdot \boldsymbol{u}_N\|_{L^2(\Omega)}$ with respect to h_f (with fixed $N = 2$) and N (with fixed $h_f = 1/2$).

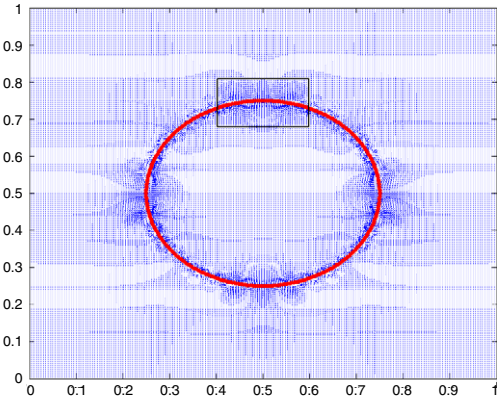
$N = 2$			$h_f = 1/2$		
h_f	$\|\nabla \cdot \boldsymbol{u}_N\|_{L^2}$	Order	N	$\|\nabla \cdot \boldsymbol{u}_N\|_{L^2}$	Order
1/2	1.09677	–	2	1.09677	–
1/4	0.52110	1.07363	4	0.43301	1.34078
1/8	0.37694	0.46723	8	0.18885	1.19718
1/16	0.26759	0.49433	16	0.08830	1.09679
1/32	0.20385	0.39247	32	0.04220	1.06528
1/64	0.13654	0.57819			



(a) Velocity vector for $h_f = 1/16$.



(b) Velocity vector for $h_f = 1/32$.

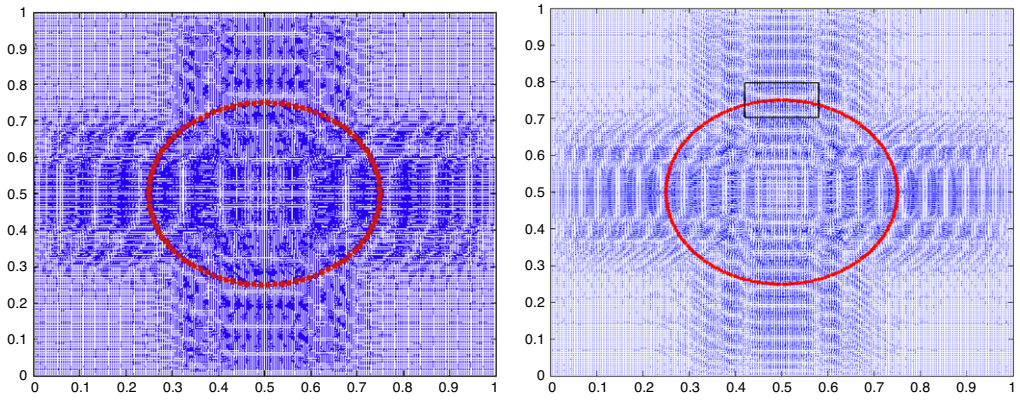


(c) Velocity vector for $h_f = 1/64$.

Fig. 7. Vector plot of the velocity for decreasing mesh width when $N = 2$ and $K_b = K_b(N)$.

immersed boundary moves freely with the non-zero local velocity. We investigate what happens to the velocity field when the polynomial degree is increased and compare this with the usual finite element approach of decreasing the mesh width.

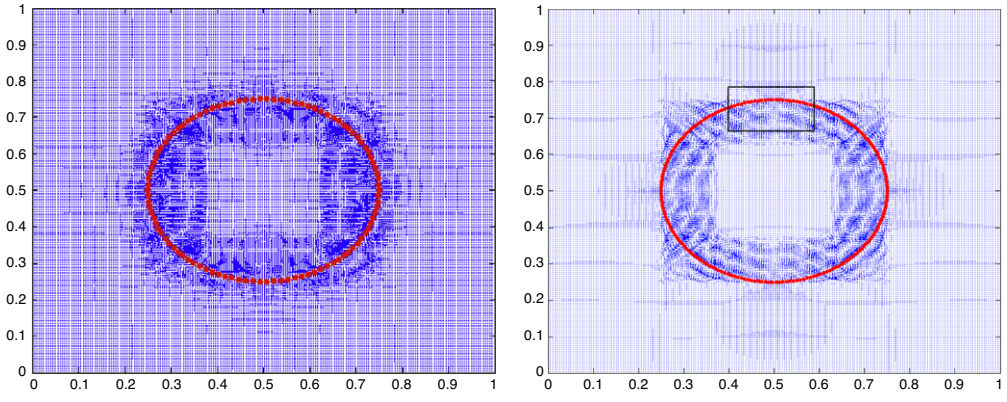
Fig. 7 illustrates the velocity field for $N = 2$ and $h_f = 1/16, 1/32$ and $1/64$ when $K_b = K_b(N)$. Fig. 7 shows that as the mesh width decreases, the velocity field becomes more local. This is physically realistic as, analytically, there should not be any spurious velocities except in the region where the structure resides. We notice that the general *pattern* does not change. In other words, the velocity is not necessarily becoming more *resolved* it is merely becoming more local or more *refined*.



(a) Velocity vector for $N = 16$.

(b) Velocity vector for $N = 32$.

Fig. 8. Vector plot of the velocity for $h_f = 1/2$ and increasing N.



(a) Velocity vector for $N = 4$.

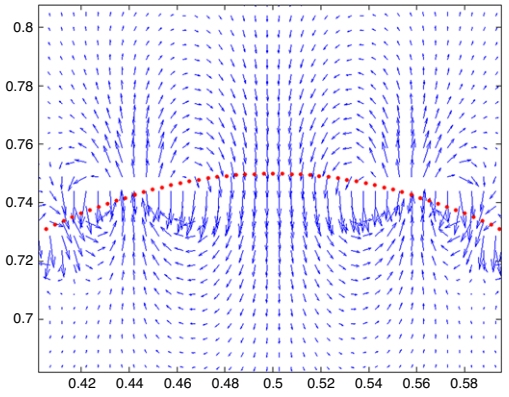
(b) Velocity vector for $N = 8$.

Fig. 9. Vector plot of the velocity for $h_f = 1/8$ and increasing N.

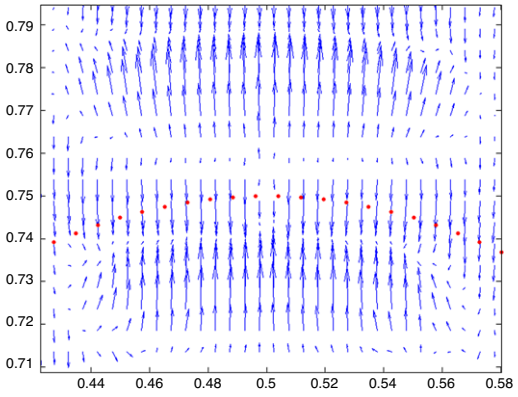
Correspondingly, Fig. 8 illustrates the velocity field for $h_f = 1/2$ with $N = 16, 32$ when $K_b = K_b(N)$. Clearly as the value of N is increased, the velocity field is not becoming more local, instead its magnitude is decreasing and a distinct *pattern* begins to emerge. This is the difference between refinement and resolution. As N is increased, the velocity field is becoming more *resolved* whilst when the mesh width is decreased the velocity field becomes more *refined*. Fig. 9 illustrates the velocity field for $h_f = 1/8$ with $N = 4, 8$ and $K_b = K_b(N)$. Once again as N is increased, a distinct *pattern* is formed in the velocity field. Comparing Figs. 7(c), 8(b) and 9(b) shows three distinct patterns for the velocity field. In order to compare these patterns more closely we zoom in to the region contained in the black square in those three plots (see Fig. 10).

The velocity vector is far more localised for the smaller mesh width. In Fig. 10(a) the velocity can be seen to be pushing through the immersed boundary towards the centre of the circle. Whilst there is no *leakage* here (the velocity field acts to push the immersed boundary towards the centre of the circle) the majority of the motion along the direction of the axes is directed towards the centre. There are small recirculatory regions on the diagonal. In Fig. 10(b) and (c) the velocity vectors inside the immersed boundary in the zoomed plots have larger magnitudes. However, this is positive in this context. We notice, that although there is still motion towards the centre of the circle, there are also velocity vectors pushing outwards from the centre of the circle. From the plots, it appears that these velocity vectors are of equal magnitude to the ones pushing inwards. This is a consequence of the incompressibility constraint being better satisfied when a high-order polynomial is used to represent the velocity field.

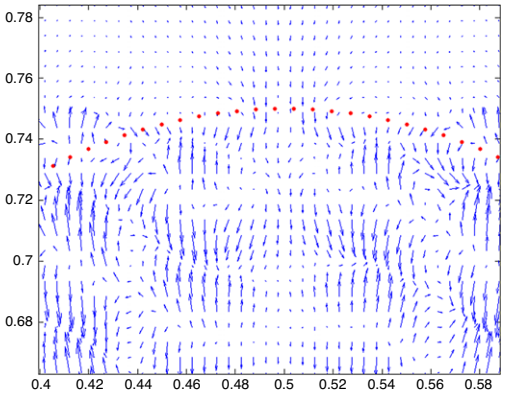
Physically, the immersed membrane has a tendency to push inwards. However, the immersed membrane cannot collapse towards the centre because the fluid inside the membrane is incompressible and thus a pressure is



(a) $h_f = 1/64$, $N = 2$.



(b) $h_f = 1/2$, $N = 32$.



(c) $h_f = 1/8$, $N = 8$.

Fig. 10. Zoomed plots of the velocity vector.

generated inside the membrane. Therefore, as the IBM is an unfitted method, numerically we require something pushing outwards from the centre to prevent the collapse of the immersed membrane. This is precisely what happens when SE-IBM is used. Whilst it is physically realistic to have a local force (because the force distribution for the membrane should only be non-zero in the region where the immersed membrane resides), the incompressibility constraint is not as accurately enforced. This is another argument in favour of the spectral/hp method [18]. Peskin [28] suggested modifying the stencil used in the finite difference computations, as well as the discrete delta function, in order to improve the area (2D)/volume (3D) loss problems. Whilst this succeeded, the method increased the amount of computational time spent in the spreading and interpolation phases. The results given above were all obtained using the SIME scheme (18) to move the membrane. We also considered a Velocity Correction Scheme (VCS) which preserves the area inside the membrane up to the order of $O((\Delta t)^2)$. The derivation of the VCS is given in Appendix A. We found that the h-type (mesh width) convergence of the area contained in the membrane had an order of approximately 2 for both $K_b = K_b(h_f)$ and $K_b(N)$. This is an improvement on the order that was inferred from Table 3. The p-type (polynomial degree) convergence when $h_f = 1/2$ also had an order of 2. The VCS only corrects the velocity for the movement of the IB and therefore, the actual velocity field is unchanged and because the derivation of the VCS is based on the SIME scheme, the velocity field plots do not change.

Finally, we note that in the immersed boundary method, the location of the interface is automatically tracked by:

$$\frac{\partial \boldsymbol{X}}{\partial t} = \boldsymbol{u}(\boldsymbol{X}(\boldsymbol{s}, t), t) \tag{29}$$

which enforces the no-slip condition at every point $\boldsymbol{X}$ by assuming the interface moves with the local fluid velocity. Clearly, if $\boldsymbol{u} = 0$ on the approximated immersed boundary, then there will be no movement. However, this can only

be accomplished if the no-slip boundary condition, $\boldsymbol{u}|_\Gamma = 0$, is enforced *strongly*. Due to the immersed boundary being unfitted, this is a non-trivial task. Lew and Guscaglia [32] introduced a discontinuous Galerkin approach which allowed boundary conditions, on an unfitted interface, to be enforced *strongly* with the aim of alleviating problems with volume/area conservation. Also, the enforcement of Dirichlet boundary conditions on the interface is problem dependent.

7. Benchmark problems for a Newtonian fluid

In this section we consider the p-type convergence (polynomial degree) using the same benchmark problems that were considered by Boffi et al. [12]: the static closed membrane introduced in Section 5, an oscillating closed membrane, a static closed shell and an oscillating closed shell.

7.1. Static closed membrane

This example was considered earlier in Sections 5 and 6 and an example of the geometry is given in Fig. 1. The immersed membrane, which is parameterised by arc-length $s \in [0, 2\pi R]$, has the initial configuration:

$$\boldsymbol{X}(s,0) = \begin{pmatrix} R\cos\left(\frac{s}{R}\right) + 0.5 \\ R\sin\left(\frac{s}{R}\right) + 0.5 \end{pmatrix} \tag{30}$$

$\forall s \in [0, 2\pi R]$. The velocity is chosen to satisfy homogeneous Dirichlet boundary conditions, $\boldsymbol{u}_D = \boldsymbol{0}$ on $\partial\Omega$, with zero initial conditions $\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0 = \boldsymbol{0}$. Fortunately, due to the simplicity of the example, an analytical solution exists [12]:

$$\boldsymbol{u}(\boldsymbol{x}, t) = 0 \qquad \forall \boldsymbol{x} \in \Omega,\ \forall t \in [0, T] \tag{31a}$$

$$p(\boldsymbol{x}, t) = \begin{cases} \kappa\left(\frac{1}{R} - \pi R\right) & |\boldsymbol{x} - \boldsymbol{c}| \le R \\ -\kappa\pi R & |\boldsymbol{x} - \boldsymbol{c}| > R \end{cases} \qquad \forall t \in [0, T] \tag{31b}$$

where $\boldsymbol{c} = (0.5, 0.5)$ is the centre of the circle. Boffi et al. [12] considered a hyperelastic fibre satisfying a neo-Hookean model. Therefore, the expression for the Piola–Kirchhoff stress tensor, $\mathbb{P}_s$, was found from the strain energy density function for the fibre so that

$$\mathbb{P}_s = \kappa\frac{\partial \boldsymbol{X}}{\partial s} = \kappa\mathbb{F} \tag{32}$$

where $\mathbb{F}$ is the deformation gradient tensor. Thus, the duality pairing $\langle \boldsymbol{f}, \boldsymbol{v}\rangle$ is given by

$$\langle \boldsymbol{f}, \boldsymbol{v}\rangle = -\kappa\int_0^{2\pi R} \frac{\partial \boldsymbol{X}}{\partial s} \cdot \frac{\partial \boldsymbol{v}(\boldsymbol{X}(s,t))}{\partial s}\, \mathrm{d}s. \tag{33}$$

Incidentally, the same expression can be obtained by considering the variational formulation of the original immersed boundary method, i.e.

$$\boldsymbol{F}(s,t) = \kappa\frac{\partial^2 \boldsymbol{X}}{\partial s^2} \tag{34}$$

followed by integration by parts in the weak formulation. Interestingly, this implies that the representation of the Lagrangian force density is the same whether we consider a Hookean spring law (as is done with the original IBM [3]) or a neo-Hookean hyperelastic material (as in [12]).

The parameters used in this example are as follows: $\eta_s = 1$, $R = 0.25$, $\kappa = 1$, $\Delta t = 0.005$ and the simulation was run for 600 time steps (until $T = 3$). Note that we consider $K_b = K_b(N)$ as defined in (28). The analytical solution quoted above, is a solution of the Stokes problem.

Table 7 details the error after a single timestep and orders of convergence for the velocity and pressure solutions. We can infer that for a mesh width $h_f = 1/8$, the order of convergence for the velocity with respect to the L^2 and H^1 norms is approximately 2 and 1, respectively. The orders of convergence for the velocity are 0.5 higher than those

Table 7
Orders of convergence, after a single timestep, with respect to N when $h_f = 1/8$ for an immersed membrane.

N	$\|\boldsymbol{u}-\boldsymbol{u}_N\|_{L^2(\Omega)}$	Order	$\|\boldsymbol{u}-\boldsymbol{u}_N\|_{H^1(\Omega)}$	Order	$\|p-p_N\|_{L^2(\Omega)}$	Order
2	0.01720	–	0.45144	–	0.70314	–
4	0.00206	3.06120	0.15411	1.55056	0.33923	1.05155
8	0.00042	2.30616	0.07032	1.13189	0.22521	0.59099
16	0.00010	2.12344	0.03505	1.00452	0.14737	0.61186

Table 8
Orders of convergence, after a single timestep, with respect to N when $h_f = 1/2$ for an immersed membrane.

N	$\|\boldsymbol{u}-\boldsymbol{u}_N\|_{L^2(\Omega)}$	Order	$\|\boldsymbol{u}-\boldsymbol{u}_N\|_{H^1(\Omega)}$	Order	$\|p-p_N\|_{L^2(\Omega)}$	Order
2	0.18909	–	1.25811	–	1.40229	–
4	0.02781	2.76525	0.47229	1.41351	0.83724	0.74407
8	0.00544	2.35471	0.20530	1.20193	0.56314	0.57214
16	0.00122	2.16191	0.09508	1.11048	0.36653	0.61957
32	0.00029	2.09135	0.04565	1.05863	0.25108	0.54577

found by Boffi et al. [12], who found orders of 1.5 and 0.5 for the L^2 and H^1 norms, respectively. However, due to the irregularity of the pressure solution, we do not expect SE-IBM to perform any better than the FE-IBM. Indeed, we find that the order of convergence for the pressure is approximately 0.5 which is the same as inferred by Boffi et al. [12].

The orders of convergence for the velocity shown in Table 7 are as expected, according to the theory. Boffi et al. [12] attributed their sub-optimal order of convergence of the velocity on the lack of regularity of the pressure. However, when we consider p-type convergence, it appears that the regularity of the pressure solution does not affect the order of convergence for velocity. To explore this further, we considered the same example on a coarser mesh. Table 8 shows the order of convergence of the velocity and pressure solutions when $h_f = 1/2$ for different values of N. It is clear that we can infer an order of 2 and 1 for the L^2 and H^1 norms of the velocity, respectively. We can also infer an order of approximately 0.5 for the L^2 norm of the pressure. Therefore, we can conclude that the lack of regularity of the pressure solution does *not* affect the convergence rate for the velocity when a high-order approximation is used.

According to Maday, Patera and Rønquist [14] (their Theorem 3.8) and Bernardi and Maday [33, p. 152] (their Theorem 7.7), for $\boldsymbol{u} \in H^m(\Omega)^2$ and $p \in H^{m-1}(\Omega)$, $m \geq 1$, the approximation error is given by:

$$\|\boldsymbol{u}-\boldsymbol{u}_N\|_{L^2(\Omega)^2} \leq cN^{-m}\left(\|\boldsymbol{u}\|_{H^m(\Omega)^2} + \|p\|_{H^{m-1}(\Omega)}\right) \tag{35a}$$

$$\|\boldsymbol{u}-\boldsymbol{u}_N\|_{H^1(\Omega)^2} \leq cN^{1-m}\left(\|\boldsymbol{u}\|_{H^m(\Omega)^2} + \|p\|_{H^{m-1}(\Omega)}\right) \tag{35b}$$

$$\|p-p_N\|_{L^2(\Omega)} \leq cN^{\frac{3}{2}-m}\left(\|\boldsymbol{u}\|_{H^m(\Omega)^2} + \|p\|_{H^{m-1}(\Omega)}\right). \tag{35c}$$

Note that the integration error of the source term has been omitted since for the case of an immersed boundary, the integration present in the source term (or spreading phase) is calculated analytically. From these results we can see that the order of convergence for the velocity if $m = 2$ is 2 and 1 for the L^2 and H^1 norms, respectively, and 0.5 for the L^2 norm of the pressure. This is in agreement with the results presented in the tables above. However, as mentioned by Boffi et al. [12], $p \notin H^1(\Omega)$. Therefore, according to the estimates given above, we would expect the order of the velocity and pressure convergence to be impaired due to the irregularity of the pressure solution. However, the condition on the regularity of the pressure given in the above error estimates is not satisfied, therefore we cannot assume that the error estimates are of the form given above. Thus, it is possible that the error of the velocity is not dependent on the regularity of the pressure. For example, we wish to find: $(\boldsymbol{u}_N, p_N) \in \mathcal{V}_N \times \mathcal{Q}_N$, where $\mathcal{V}_N$ and $\mathcal{Q}_N$ are as defined in (9), such that

$$(\nabla \boldsymbol{u}_N, \nabla \boldsymbol{v}_N)_N + (p_N, \nabla \cdot \boldsymbol{v}_N)_N = \langle \boldsymbol{f}, \boldsymbol{v}_N \rangle \qquad \forall \boldsymbol{v}_N \in \mathcal{V}_N \tag{36}$$

$$(q_N, \nabla \cdot \boldsymbol{u}_N)_N = 0 \qquad \forall q_N \in \mathcal{Q}_N. \tag{37}$$

The velocity solution $\boldsymbol{u}_N$ of the above system must belong to the space:

$$\mathcal{V}_N^{div} = \left\{ \boldsymbol{u}_N \in \mathcal{V}_N : \ (q_N, \nabla \cdot \boldsymbol{u}_N)_N \ , \ \forall q_N \in \mathcal{Q}_N \right\}. \tag{38}$$

Therefore, following Bernardi and Maday [34], $\boldsymbol{u}_N$ is a solution of the problem: find $\boldsymbol{u}_N \in \mathcal{V}_N^{div}$ such that

$$(\nabla \boldsymbol{u}_N, \nabla \boldsymbol{v}_N)_N = \langle \boldsymbol{f}, \boldsymbol{v}_N \rangle \qquad \forall \boldsymbol{v}_N \in \mathcal{V}_N^{div}. \tag{39}$$

It is clear from the above, that the error of the velocity solution is only dependent on the error of the source term (or spreading phase). Technically, the pressure belongs in a broken Sobolev space and as far as we are aware, error estimates for the spectral approximation of pressure from such a space do not currently exist. The question then is: Why do we obtain second and first order convergence rates for the L^2 and H^1 norms of the velocity when $p \notin H^1(\Omega)$?

Potentially, there are two main causes of the impaired order of convergence found by Boffi et al. [12]: Errors in the spreading and interpolation phase due to low order interpolation and the linear approximation of the curved immersed boundary. However, the convergence studies of SE-IBM carried out here are not affected by those errors. The spreading and interpolation phases are dominated by interpolation errors, therefore using higher order polynomials will reduce the interpolation error. As we saw in Section 6, improving the accuracy of the spreading and interpolation phases improves the area conservation. Additionally, we are not considering mesh width (h-type) convergence, therefore the linear approximation of the curved immersed boundary does not affect the rate of convergence with respect to polynomial degree (p-type). Li et al. [35] showed that for problems with a curved interface, obtaining optimal rates of convergence with respect to mesh width (h-type) can only be achieved if a higher order approximation of the interface is considered.

Fig. 11 compares the computed pressure solution for $h_f = 1/8$ and $N = 16$ against the analytical solution. It is clear that we have obtained good agreement with the analytical solution. Gibbs phenomenon can be seen local to the immersed membrane due to the IBM being an unfitted method, i.e. the immersed membrane is allowed to move freely over the underlying fluid mesh. Comparing Fig. 11 with those in the work of Boffi et al. [12] (their Fig. 2) shows that there is good agreement.

7.2. Oscillating closed membrane

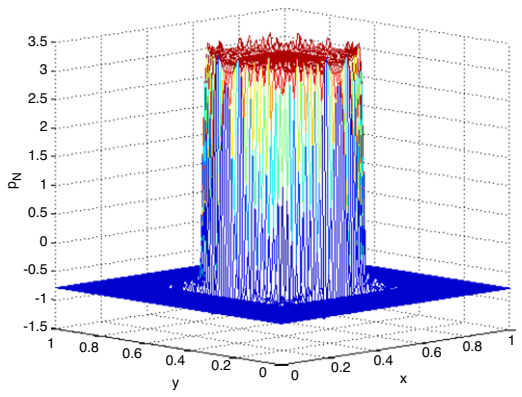
In this section, we apply the SE-IBM to an oscillating immersed membrane. Fig. 12 illustrates the movement of the immersed boundary for $h_f = 1/8$ when $N = 2$ and $N = 8$ and also the pressure solution after a single timestep. A substantial loss in area when $N = 2$ is observed as expected for such a coarse approximation. As the analytical solution for such an example is not known, we compare our results to those of Boffi et al. [12]. The computed pressure solution in Fig. 12(c), gives very good agreement with the computed solution of Boffi et al. [12] (their Fig. 3(b)). We note that there are still small oscillations close to the membrane. Although the analytical solution is not known, we expect the pressure to be discontinuous across the membrane. Therefore, these oscillations are most likely caused by Gibbs phenomenon.

7.3. Static closed shell

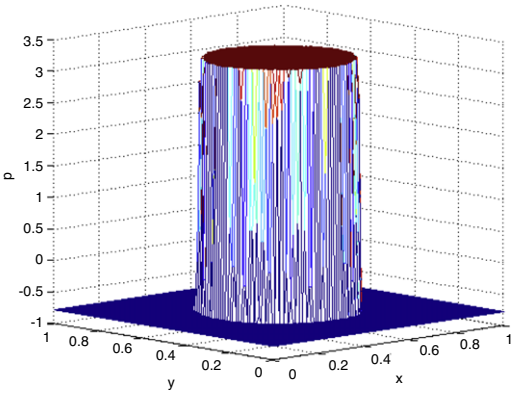
The FE-IBM, and therefore the SE-IBM, has the ability to deal with *thick* immersed structures such as a two-dimensional structure immersed in a two-dimensional fluid. The simplest immersed thick structures we can consider, are generalisations of the two examples considered in Sections 7.1 *and* 7.2 where the membrane is *thickened* to a width w. This example was considered by Boffi et al. [12]. Following their example, we define the reference configuration of the immersed structure as $\Omega_r^s = [0, 2\pi R] \times [0, w]$. We denote a material point in the reference configuration as $\boldsymbol{s} = (s, r)$ where $s \in [0, 2\pi R]$ is the arc-length parameter and $r \in [0, w]$ is the radial parameter. We define the Piola–Kirchhoff stress tensor by

$$\mathbb{P}_s = \frac{\kappa}{w} \begin{pmatrix} \dfrac{\partial X}{\partial s} & 0 \\ \dfrac{\partial Y}{\partial s} & 0 \end{pmatrix} \tag{40}$$

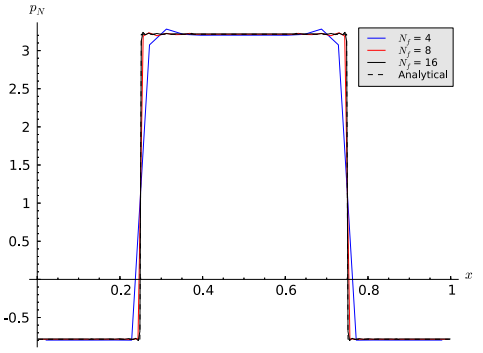
where $\boldsymbol{X} = (X, Y)$. It is evident from the above definition that we are assuming that the deformation of the structure occurs only in the tangential direction. In other words, we are assuming that the deformation is fibre-like and therefore,



(a) Computed pressure solution for $N = 16$.



(b) Analytical pressure solution.



(c) Computed pressure solution along the cut $y = 0.5$ for $N = 4, 8, 16$.

Fig. 11. Computed and analytical pressure plots, at the end of the simulation, for increasing N when $h_f = 1/8$.

this example should be equivalent to the original IBM with multiple fibres placed at certain $r \in [0, w]$. The initial configuration of the IB position is given by

$$\boldsymbol{X}(\boldsymbol{s}, 0) = \begin{pmatrix} (R + r)\cos\left(\dfrac{s}{R}\right) + 0.5 \\ (R + r)\sin\left(\dfrac{s}{R}\right) + 0.5 \end{pmatrix}. \tag{41}$$

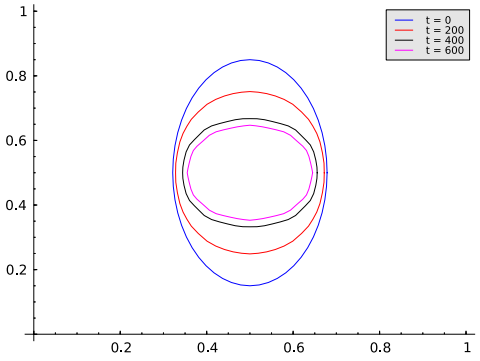
Note that is *not* the same initial configuration that was given by Boffi et al. [12]. Their initial condition implied a radial vector $\boldsymbol{r}$ of length $R(1 + w)$, whereas the analytical solution they give implied a radial vector of length $R + w$. We have modified the initial condition so that it matches the analytical solution. The analytical solution given by Boffi et al. [12] is

$$\boldsymbol{u}(\boldsymbol{x}, t) = 0 \tag{42a}$$

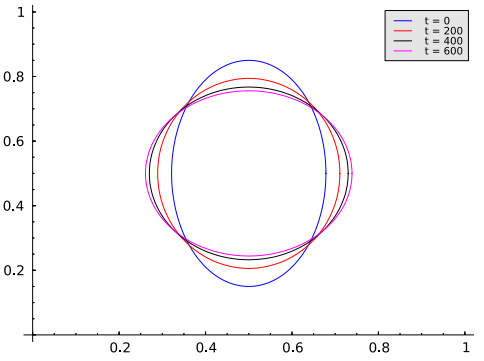
$$p(\boldsymbol{x}, t) = \begin{cases} p_0 + \dfrac{\kappa}{R} & |\boldsymbol{x} - \boldsymbol{c}| \le R \\ p_0 + \dfrac{\kappa}{w}\dfrac{1}{R}(R + w - r) & R < |\boldsymbol{x} - \boldsymbol{c}| \le R + w \\ p_0 & R + w < |\boldsymbol{x} - \boldsymbol{c}| \end{cases} \tag{42b}$$

where $r = |\boldsymbol{x} - \boldsymbol{c}|$, $\boldsymbol{c} = (0.5, 0.5)$ is the centre of the shell and the constant p_0 is chosen so that the zero mean pressure condition is satisfied. Therefore, p_0 is given by

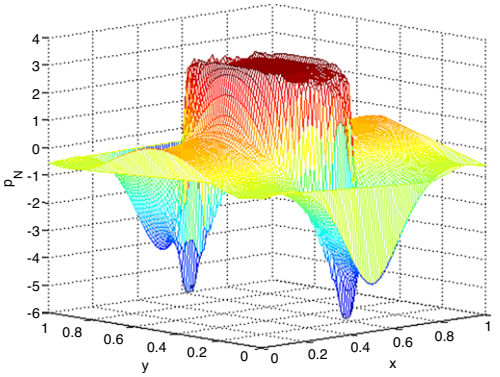
$$p_0 = \frac{\kappa\pi}{3w}\left[R^2 - \frac{(R + w)^3}{R}\right]. \tag{43}$$



(a) IB evolution for $N = 2$.



(b) IB evolution for $N = 8$.



(c) Pressure solution for $N = 8$ after a single timestep.

Fig. 12. Immersed boundary evolution for $h_f = 1/8$ and $N = 2, 8$ and the pressure solution for $h_f = 1/8$ when $N = 8$.

We note that the analytical solution for the pressure is continuous everywhere and linear in the region where the immersed structure resides. The parameters chosen for this example are the same as given in [12] and are as follows: $\eta_s = 1$, $R = 0.25$, $\kappa = 1$, $w = 0.0625$, $\Delta t = 0.005$ and the simulation was run for 600 time steps. Just as was done previously for the case of an immersed membrane, we must choose the number of structure elements so that $h^s \leq h_f/2$. In this article, we assume that the total number of elements is given by $K_b = K_b^s \times K_b^r$, where $K_b^s = K_b^s(N)$ is calculated using the relation in (28) and K_b^r is calculated using:

$$K_b^r \geq \frac{2w}{h_f} \tag{44}$$

where $h_f = 1/NK_f$. These values of K_b^s and K_b^r ensure that the decomposition of Ω_r^s is composed of *approximately* uniform quadrilateral elements. Again, the analytical solution given in (42) is the analytical solution for Stokes flow.

Table 9 details the error and orders of convergence for the velocity and pressure solutions after a single timestep for a mesh width of $h_f = 1/8$. We can infer that the order of convergence for the velocity with respect to the L^2 and H^1 norms is approximately 3 and 2, respectively. These orders of convergence are higher than those found by Boffi et al. [12] who inferred an order of 2.5 in the L^2 norm and 1.5 in the H^1 norm for the velocity. The pressure solution can be seen to have an order of approximately 2 which again is higher than the order of 1.5 inferred by Boffi et al. [12].

To ensure that we are in fact obtaining higher orders of convergence than were seen by Boffi et al. [12] we consider the order of convergence for a mesh width of $h_f = 1/2$. Table 10 details the error and orders of convergence for the velocity and pressure solutions after a single timestep. We can infer from Table 10 that we have an order of approximately 3 and 2 in the L^2 and H^1 norm of the velocity error and an order of approximately 1.5 in the L^2 norm

Table 9
Order of convergence with respect to N_f when $h_f = 1/8$ for a shell immersed in a Stokesian fluid.

N_f	$\|\boldsymbol{u} - \boldsymbol{u}_N\|_{L^2(\Omega)}$	Order	$\|\boldsymbol{u} - \boldsymbol{u}_N\|_{H^1(\Omega)}$	Order	$\|p - p_N\|_{L^2(\Omega)}$	Order
2	0.01889	–	0.47424	–	0.61513	–
4	0.00092	4.35444	0.06499	2.86740	0.09170	2.74584
8	0.00008	3.58092	0.01273	2.35195	0.02406	1.93034

Table 10
Order of convergence with respect to N_f when $h_f = 1/2$ for a shell immersed in a Stokesian fluid.

N_f	$\|\boldsymbol{u} - \boldsymbol{u}_N\|_{L^2(\Omega)}$	Order	$\|\boldsymbol{u} - \boldsymbol{u}_N\|_{H^1(\Omega)}$	Order	$\|p - p_N\|_{L^2(\Omega)}$	Order
2	0.20952	–	1.38309	–	1.73205	–
4	0.03071	2.77024	0.50879	1.44275	0.57180	1.59890
8	0.00469	2.71096	0.17514	1.53853	0.25584	1.16026
16	0.00035	3.76494	0.03091	2.50226	0.06881	1.89459

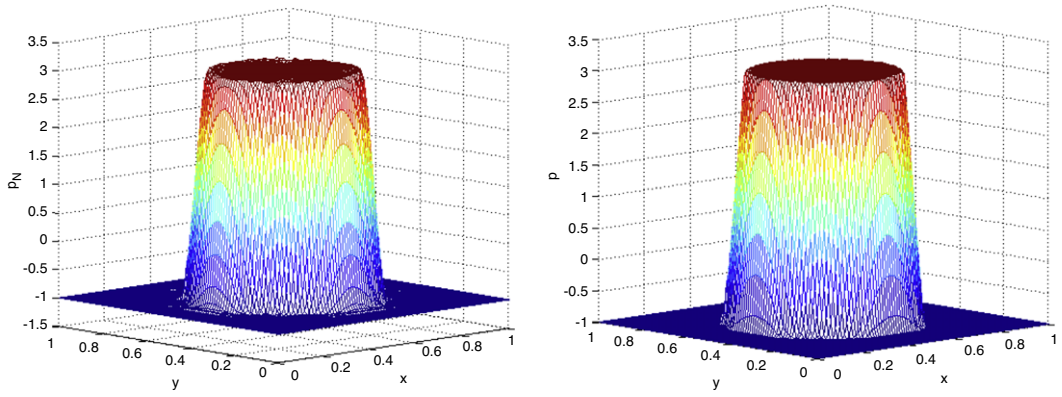
of the pressure error. The velocity order is higher than the order found by Boffi et al. [12] but the order of pressure given here is the same as that inferred by Boffi et al.

Since $p \in H^1(\Omega)$ but $p \notin H^2(\Omega)$, we can assume that p exists in an interpolation space, somewhere between H^1 and H^2. Boffi et al. [12] stated that $p \in H^{3/2-\varepsilon}(\Omega)$, $\varepsilon > 0$, and they attributed their sub-optimal orders of convergence to the pressure being from this interpolation space. Unfortunately, as far as we are aware, error estimates for the spectral approximation of a function from an interpolation space such as $H^{3/2-\varepsilon}$ do not exist. Thus, we can infer that the reason for the superior rates of convergence given above are the same as discussed in Section 7.1. For the original immersed boundary method, it has been shown in the literature [36,37] that second order convergence of the velocity with respect to mesh width (h-type) is obtained for sufficiently smooth problems. In this section, we have illustrated that higher rates of convergence can be obtained when using SE-IBM. Again, we emphasise that the majority of the error in the IBM originates in the spreading and interpolation phases; specifically, for the original IBM the largest error can be found in the approximation of the delta function. Given that the delta function is not a function, it makes more sense to approximate it distributionally (i.e. approximate its action on another function). This is the procedure employed in the IFEM [9] and the FE-IBM [10]. In the FE-IBM, and hence the SE-IBM, the action of the delta function is included through its sifting property. Polynomial interpolation is then used to construct the spreading and interpolation phases, as shown in Section 3. Therefore, as mentioned earlier, the main source of error comes from interpolation error, which is only dependent on the regularity of the function being interpolated. Thus, a function $u \in H^2$ should yield second order convergence, whilst a function $u \in H^3$ should yield third order convergence (assuming no impairment occurs from elsewhere).

Fig. 13(a) and (b) present the computed and analytical pressure solutions. It is clear that there is good agreement between the two solutions and with the results of Boffi et al. [12] (their Fig. 4(f)). We note that the solution of Boffi et al. had larger oscillations local to the shell compared to our solution.

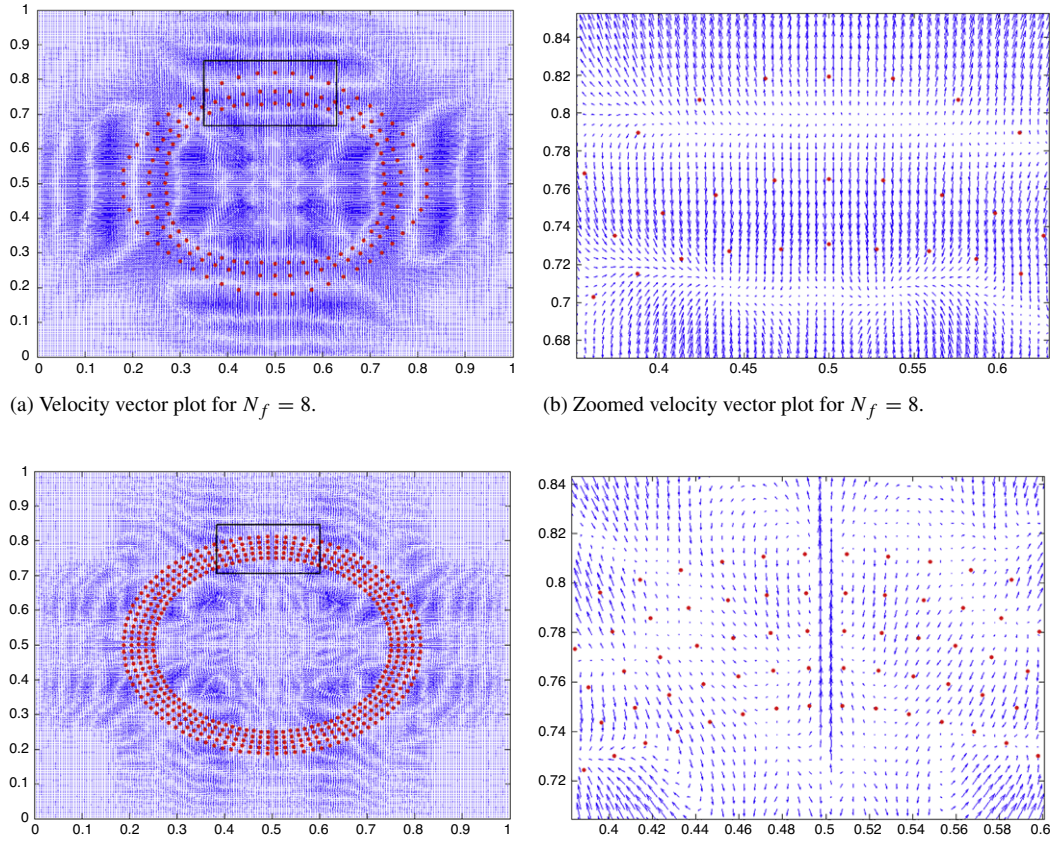
Fig. 14 presents the velocity vector at the end of the simulation when $N = 8, 16$ for $h_f = 1/2$. Just as was seen with the immersed membrane, the pattern of the velocity field around the shell is becoming more pronounced and by the time $N = 16$ it has a clear shape. Clearly, when $N_f = 8$ (Fig. 14(a)) there is separation in the immersed structure marker particles. This is clearer to see in the zoomed plots in Fig. 14(b). This is due to the nature of the local velocity field around each marker particle since each particle is allowed to move independently of its neighbours. However, when $N = 16$ (Fig. 14(c)) there is very little separation and the position of the immersed boundary marker particles are more or less uniformly arranged. This illustrates a potential weakness in the IBM, but this weakness is unlikely to occur on finer meshes or for higher values of polynomial degree.

Fig. 15 illustrates the velocity field when $N = 8$ and $h_f = 1/8$. The plot of the immersed boundary in Fig. 15 is after 200 time steps rather than 600 because the solution had reached a steady state in the velocity field and so the simulation terminated early. Just as with Fig. 14(c) we notice that the immersed boundary has remained intact and there is no separation of the *layers* of the immersed structure.



(a) Computed pressure solution. (b) Analytical pressure solution.

Fig. 13. Pressure plots, at the end of the simulation, for an immersed shell when $h_f = 1/8$ and $N = 8$.



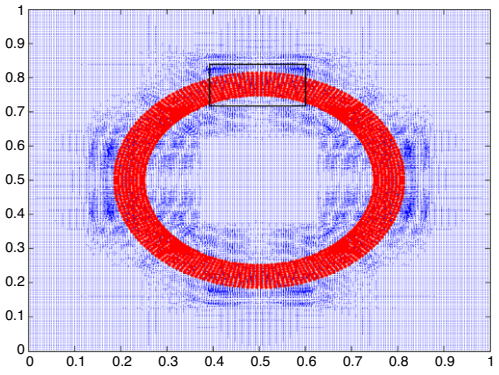
(a) Velocity vector plot for $N_f = 8$. (b) Zoomed velocity vector plot for $N_f = 8$.

(c) Velocity vector plot for $N_f = 16$. (d) Zoomed velocity vector plot for $N_f = 16$.

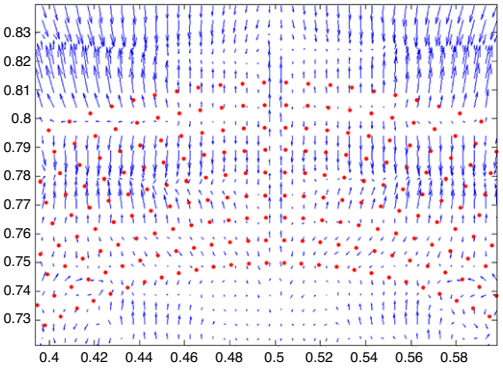
Fig. 14. Velocity vector for $h_f = 1/2$ with increasing N. The red dots on each plot are the final position of the IB marker particles. The black box is the region which is shown in the zoomed plot. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

7.4. Oscillating closed shell

In this section we apply the SE-IBM to an oscillating immersed closed shell. Fig. 16 illustrates the movement of the immersed shell and also the pressure solution after a single time step. Once again our parameters are: $h_f = 1/8$

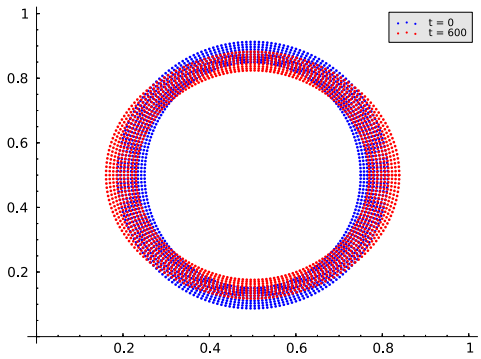


(a) Velocity vector plot for $N_f = 8$.

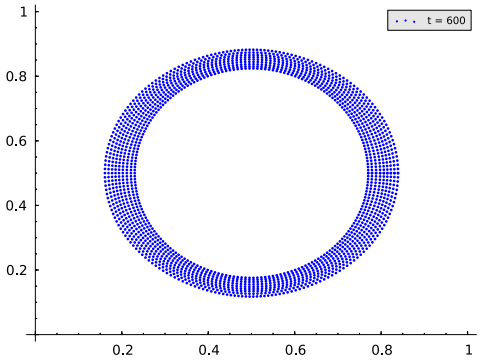


(b) Zoomed velocity vector plot for $N_f = 8$.

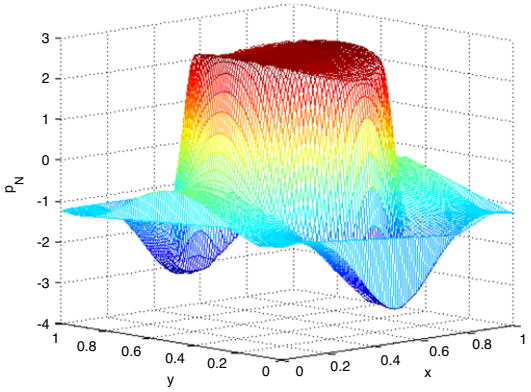
Fig. 15. Velocity vectors for $h_f = 1/8$ and $N = 8$. The red dots on each plot are the final position of the IB marker particles. The black box is the region which is shown in the zoomed plot.



(a) IB evolution for $N = 8$ at $T = 0$ and $T = 3$.



(b) IB evolution for $N = 8$ at $T = 3$.



(c) Computed pressure solution for $N = 8$ after a single timestep.

Fig. 16. Immersed shell evolution for $h_f = 1/8$ and $N = 2, 8$ and the pressure solution, after a single timestep, for $h_f = 1/8$ when $N = 8$.

and $N = 8$. The simulation was terminated at $T = 3$. It is clear from the motion of the shell that there is no separation between the *layers* of marker particles. This could be due to the number of marker particles that make up the shell or the more accurate velocity field as a result of the higher order approximation. However, it seems from Fig. 16(b) that the shell is thinner at the top and bottom than it is at the sides. This again is most likely caused by the local fluid velocity and it may be that increasing the order beyond $N = 8$ will produce a more uniform shell.

8. Conclusions and future work

In this article, the spectral element method was applied to the immersed boundary method giving rise to the spectral element immersed boundary method (SE-IBM). The aim of using a high-order approximation was to improve the accuracy of the spreading and interpolation phases, which is the main source of error in area conservation for the IBM. When a closed, circular, elastic boundary is immersed in a fluid, the use of a higher-order spectral discretisation was shown (in Section 6) to improve the accuracy of the spreading and interpolation phases and hence improved the area conservation properties of the method. Additionally, it was illustrated in Section 7, that the improved accuracy of the spreading and interpolation phases yielded superior convergence properties for the velocity field. In Section 6 (Table 5), we showed that SE-IBM could achieve better area conservation than FE-IBM in a fraction of the computational time. However, we also established that although a coarse mesh width $h_f = 1/2$ and a high polynomial degree $N = 32$ yielded very good area conservation, it was inefficient in terms of computational time. Therefore, we suggested that a medium mesh width, e.g. $h_f = 1/8$, with a medium polynomial degree, e.g. $N = 8$, is the best option as it gives a balance between good area conservation (and therefore accurate spreading and interpolation phases) and computational efficiency.

The implication of using higher-order polynomial approximations on the velocity field local to the immersed boundary was also investigated. When low order polynomials and fine mesh widths are used the velocity field is local to the membrane (or boundary) and primarily directed inwards towards the centre of the immersed circular membrane. However, when a high-order polynomial is used, the velocity field not only directs inwards but also outwards from the centre of the immersed circular membrane. We believe this is the reason for the better area conservation obtained using higher-order methods. The fluid contained inside the membrane is incompressible, and therefore the membrane should *not* be able to collapse inwards. As the IBM is an unfitted method, the satisfaction of boundary conditions on the interface between the fluid and the structure cannot be enforced strongly. Therefore, we require something inside the membrane to resist the collapse (and hence reduce the area loss) of the membrane. This is precisely the behaviour observed in the velocity field. We concluded that the use of high-order polynomials is an important asset in immersed boundary computations. Analytically, the force due to the presence of an immersed boundary should only be non-zero in the region where the immersed boundary resides. Therefore, the spectral/hp method of Karniadakis and Sherwin [18] as an appropriate scheme for immersed boundary computations.

The IBM is known to suffer from temporal stiffness problems at high values of the IB stiffness and low values of the fluid viscosity. Higher-order temporal discretisations of the immersed boundary evolution equation (17) were investigated. It was shown that using an implicit discretisation of the immersed boundary evolution equation allowed larger time steps to be used. However, the improvement was not substantial. Additionally, using a high-order implicit discretisation allowed larger time steps at larger values of the polynomial degree. However, once again, the improvement was not substantial. Therefore, the standard semi-implicit Euler discretisation (18) was deemed to be sufficient for all the example problems considered in this article. The examples considered here are standard within the immersed boundary literature and were considered by Boffi et al. [12] for the FE-IBM, namely a closed static membrane and shell and an oscillating closed membrane and shell. Excellent agreement was obtained with FE-IBM predictions. Furthermore, SE-IBM was shown to possess superior convergence properties for the velocity field with respect to both the L^2 and H^1 norms.

The main advantages of SE–IBM over FE-IBM are the improved accuracy of the spreading and interpolation phases – which yields better area conservation and superior convergence properties – as well as the improved computational efficiency as indicated in Table 5. However, SE-IBM has the same drawback as the FE-IBM and other immersed boundary methods; namely the assumption of constant viscosity throughout the entire computational domain. Overcoming this drawback is a future topic of research.

In future work, we will perform an extensive comparison between all of the different immersed boundary methods available as well as a classical approach. Additionally, we will investigate the order of convergence of velocity and pressure with respect to mesh width for higher-order polynomials to determine whether the sub-optimal rates of convergence reported by Boffi et al. [12] are caused by inaccuracy in the spreading or interpolation phases. We intend to apply a high-order approximation to the immersed boundary position to determine whether the linear approximation of the interface impairs the convergence of the method. Although all of the examples considered in this article are academic, the immersed boundary method is not limited to such problems. The IBM has been applied to various real-world problems such as flows with suspended particles [7] and the modelling of the cochlea [5]. Therefore, in the

future, we wish to apply the SE-IBM to more practical problems such as blood flow through the smaller blood vessels. In such an example, the blood becomes viscoelastic and in a future article we will apply the SE-IBM to this problem.

Acknowledgements

One of the authors (CFR) received funding from the Engineering and Physical Sciences Research Council of the United Kingdom to support his doctoral studies (Grant No. EP/P504864) and this is gratefully acknowledged. Part of this work was performed using the computational facilities of the Advanced Research Computing @ Cardiff (ARCCA) Division, Cardiff University. The authors would also like to thank the referees for their thorough review of this article and for their valuable comments.

Appendix. Velocity correction scheme

In this appendix, our domain $\Omega \subset \mathbb{R}^2$ is composed of two subdomains $\Omega_1 \subset \Omega$ and $\Omega_2 \subset \Omega$. The idea behind the Velocity Correction Scheme (VCS) is to ensure that $\nabla \cdot \boldsymbol{u} = 0$ in a discrete setting: i.e. $\nabla \cdot \boldsymbol{u}_N = 0$. The problem is to ensure that the area inside an enclosed membrane is conserved, i.e. for each $i = 1, 2$,

$$\int_{\Omega_i} \nabla \cdot \boldsymbol{u} \, \mathrm{d}\Omega = 0 \qquad \text{or} \qquad \int_{\partial\Omega_i} \boldsymbol{u} \cdot \boldsymbol{n} \, \mathrm{d}s = 0. \tag{A.1}$$

Discretising the boundary integral gives:

$$\int_{\partial\Omega_i} \boldsymbol{u} \cdot \boldsymbol{n} \, \mathrm{d}s \approx \sum_{j=1}^{K_b} \boldsymbol{u}_j \cdot \boldsymbol{n}_j \Delta s_j = 0 \tag{A.2}$$

where K_b is the number of segments. Suppose that we have a uniform discretisation of the interface $\partial\Omega_i$ so that the points are equally spaced. The tangent vector at the point $\boldsymbol{X}_i = (X_i, Y_i)$ is

$$\boldsymbol{t}_i = (X_{i+1} - X_{i-1}, Y_{i+1} - Y_{i-1}). \tag{A.3}$$

Thus the normal vector at the point $\boldsymbol{X}_i$ is given by:

$$\boldsymbol{n}_i = \pm \frac{(Y_{i+1} - Y_{i-1}, X_{i-1} - X_{i+1})}{|\boldsymbol{t}_i|} = \pm \frac{(Y_{i+1} - Y_{i-1}, X_{i-1} - X_{i+1})}{2\Delta s_i} \tag{A.4}$$

where $\Delta s_i = |\boldsymbol{t}_i| / 2$. To enforce (A.2), we determine the mean value of $\boldsymbol{u}_i \cdot \boldsymbol{n}_i$:

$$\mu = \frac{\sum_{i=1}^{K_b} (\boldsymbol{u}_i \cdot \boldsymbol{n}_i) \Delta s_i}{\sum_{i=1}^{K_b} \Delta s_i} \tag{A.5}$$

and define the corrected velocity by: $\hat{\boldsymbol{u}}_i = \boldsymbol{u}_i - \mu \boldsymbol{n}_i$. Let $\boldsymbol{u}_i = (u_i, v_i)$ then the area contained inside a polygon is defined as:

$$A = \frac{1}{2} \sum_{j=1}^{K_b} (X_{j-1} Y_j - X_j Y_{j-1}) = \frac{1}{4} \sum_{j=1}^{K_b} X_j (Y_{j+1} - Y_{j-1}) + \frac{1}{4} \sum_{j=1}^{K_b} Y_j (X_{j-1} - X_{j+1}). \tag{A.6}$$

Let $X_j^{n+1} = X_j^n + \Delta t \, u_j^{n+1}$ and $Y_j^{n+1} = Y_j^n + \Delta t \, v_j^{n+1}$ for $j = 1, \ldots, K_b$. Then after substituting X_j^{n+1} and Y_j^{n+1} into (A.6), the area at time t^{n+1} is given by:

$$\begin{aligned} A^{n+1} = {} & \frac{1}{4} \sum_{j=1}^{K_b} \left(X_j^n + \Delta t \, u_j^{n+1} \right) \left(Y_{j+1}^n + \Delta t \, v_{j+1}^{n+1} - Y_{j-1}^n - \Delta t \, v_{j-1}^{n+1} \right) \\ & + \frac{1}{4} \sum_{j=1}^{K_b} \left(Y_j^n + \Delta t \, v_j^{n+1} \right) \left(X_{j-1}^n + \Delta t \, u_{j-1}^{n+1} - X_{j+1}^n - \Delta t \, u_{j+1}^{n+1} \right) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{4}\sum_{j=1}^{K_b} X_j^n (Y_{j+1}^n - Y_{j-1}^n) + \frac{1}{4}\sum_{j=1}^{K_b} Y_j^n (X_{j-1}^n - X_{j+1}^n) + \frac{\Delta t}{4}\sum_{j=1}^{K_b} X_j^n (v_{j+1}^{n+1} - v_{j-1}^{n+1}) \\
&\quad + \frac{\Delta t}{4}\sum_{j=1}^{K_b} Y_j^n (u_{j-1}^{n+1} - u_{j+1}^{n+1}) + \frac{\Delta t}{4}\sum_{j=1}^{K_b} u_j^{n+1} (Y_{j+1}^n - Y_{j-1}^n) + \frac{\Delta t}{4}\sum_{j=1}^{K_b} v_j^{n+1} (X_{j-1}^n - X_{j+1}^n) \\
&\quad + \frac{\Delta t^2}{4}\sum_{j=1}^{K_b} u_j^{n+1} (v_{j+1}^{n+1} - v_{j-1}^{n+1}) + \frac{\Delta t^2}{4}\sum_{j=1}^{K_b} v_j^{n+1} (u_{j-1}^{n+1} - u_{j+1}^{n+1}) \\
&= A^n + \frac{\Delta t}{2}\sum_{j=1}^{K_b} u_j^{n+1} (Y_{j+1}^n - Y_{j-1}^n) + \frac{\Delta t}{2}\sum_{j=1}^{K_b} v_j^{n+1} (X_{j-1}^n - X_{j+1}^n) \\
&\quad + \frac{\Delta t^2}{4}\sum_{j=1}^{K_b} u_j^{n+1} (v_{j+1}^{n+1} - v_{j-1}^{n+1}) + \frac{\Delta t^2}{4}\sum_{j=1}^{K_b} v_j^{n+1} (u_{j-1}^{n+1} - u_{j+1}^{n+1}). \qquad \text{(A.7)}
\end{aligned}
$$

The new position of each point (X_j^{n+1}, Y_j^{n+1}) is found from the velocity field (u_j^{n+1}, v_j^{n+1}) and the current position (X_j^n, Y_j^n), $j = 1, \ldots, K_b$. Therefore, assuming we are using the corrected velocity to find the new position (X_j^{n+1}, Y_j^{n+1}), the velocity and normal vectors in (A.5), are evaluated at times t^{n+1} and t^n, respectively, i.e.

$$\sum_{j=1}^{K_b} \hat{\boldsymbol{u}}_j^{n+1} \cdot \boldsymbol{n}_j^n \Delta s_j = 0.$$

Substituting expressions for the normals using (A.4), we can show that

$$\frac{1}{2}\sum_{j=1}^{K_b} \hat{u}_j^{n+1} (Y_{j+1}^n - Y_{j-1}^n) + \frac{1}{2}\sum_{j=1}^{K_b} \hat{v}_j^{n+1} (X_{j-1}^n - X_{j+1}^n) = 0.$$

Therefore, the terms in A^{n+1} of $\mathcal{O}(\Delta t)$ vanish and the method preserves the area up to $\mathcal{O}(\Delta t^2)$.

References

[1] T.J.R. Hughes, W.K. Liu, T.K. Zimmerman, Lagrangian–Eulerian finite element formulation for incompressible viscous flows, Comput. Methods Appl. Mech. Engrg. 29 (1981) 329.

[2] S. Deparis, M. Discacciati, A. Quarteroni, A domain decomposition framework for fluid–structure interaction problems, in: C. Groth, D.W. Zingg (Eds.), Computational Fluid Dynamics 2004, Springer, Berlin, Heidelberg, 2004, pp. 41–58.

[3] C.S. Peskin, Flow patterns around heart valves: A numerical method, J. Comput. Phys. 10 (1972) 252–271.

[4] K.M. Arthurs, L.C. Moore, C.S. Peskin, E.B. Pitman, H.E. Layton, Modelling arteriolar flow and mass transport using the immersed boundary method, J. Comput. Phys. 147 (1998) 402–440.

[5] R.P. Beyer Jr., A computational model of the cochlea using the immersed boundary method, J. Comput. Phys. 98 (1992) 145–162.

[6] L. Shi, T.-W. Pan, R. Glowinski, Numerical simulation of lateral migration of red blood cells in Poiseuille flows, Internat. J. Numer. Methods Fluids 68 (11) (2012) 1393–1408.

[7] A.L. Fogelson, C.S. Peskin, A fast numerical method for solving the three-dimensional Stokes' equations in the presence of suspended particles, J. Comput. Phys. 79 (1988) 50–69.

[8] C.S. Peskin, The immersed boundary method, Acta Numer. (2002) 479–517.

[9] L. Zhang, A. Gerstenberger, X. Wang, W.K. Liu, Immersed finite element method, Comput. Methods Appl. Mech. Engrg. 193 (2004) 2051–2067.

[10] D. Boffi, L. Gastaldi, A finite element approach for the immersed boundary method, Comput. Struct. 81 (2003) 491–501.

[11] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.

[12] D. Boffi, L. Gastaldi, L. Heltai, C.S. Peskin, On the hyper-elastic formulation of the immersed boundary method, Comput. Methods Appl. Mech. Engrg. 197 (2008) 2210–2231.

[13] A.T. Patera, A spectral element method for fluid dynamics: Laminar flow in a channel expansion, J. Comput. Phys. 54 (1984) 468–488.

[14] Y. Maday, A.T. Patera, E.M. Rønquist, The $P_N \times P_{N-2}$ Method for the Approximation of the Stokes Problem, Internal Report 92025, Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie, Paris, 1992.

[15] Y. Maday, A.T. Patera, Spectral element methods for the incompressible Navier–Stokes equations, in: A.K. Noor, J.T. Oden (Eds.), State-of-the-Art Surveys on Computational Mechanics, ASME, 1989, pp. 71–143. Ch. 3.

[16] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, third ed., in: Texts in Applied Mathematics, Springer Science+Business Media, LLC, 2010.
[17] C.R. Schneidesch, M.O. Deville, Chebyshev collocation method and multi-domain decomposition for Navier–Stokes equations in complex curved geometries, J. Comput. Phys. 106 (1993) 234–257.
[18] G.E. Karniadakis, S. Sherwin, Spectral/hp element methods for computational fluid dynamics, in: Numerical Mathematics and Scientific Computation, Oxford University Press, 2005.
[19] D. Boffi, L. Gastaldi, L. Heltai, Numerical stability of the finite element immersed boundary method, M3AS 17 (10) (2006) 1479–1505.
[20] D. Boffi, L. Gastaldi, L. Heltai, On the CFL condition for the finite element immersed boundary method, Comput. Struct. 85 (2007) 775–783.
[21] C. Tu, C.S. Peskin, Stability and instability in the computation of flows with moving immersed boundaries: A comparison of three methods, SIAM J. Sci. Stat. Comput. 13 (6) (1992) 1361–1376.
[22] J.M. Stockie, B.R. Wetton, Analysis of stiffness in the immersed boundary method and implications of time-stepping schemes, J. Comput. Phys. 154 (1999) 41–64.
[23] J.M. Stockie, B.T.R. Wetton, Stability analysis for the immersed fiber problem, SIAM J. Appl. Math. 55 (6) (1995) 1577–1591.
[24] R.P. Beyer, R.J. Leveque, Analysis of a one-dimensional model for the immersed boundary method, SIAM J. Numer. Anal. 29 (2) (1992) 332–364.
[25] E.P. Newren, A.L. Fogelson, R.D. Guy, R.M. Kirby, Unconditionally stable discretizations of the immersed boundary equations, J. Comput. Phys. 222 (2007) 702–719.
[26] D. Boffi, N. Cavallini, F. Gardini, L. Gastaldi, Local mass conservation of Stokes finite elements, J. Sci. Comput. 52 (2012) 383–400.
[27] C.F. Rowlatt, Modelling flows of complex fluids using the immersed boundary method (Ph.D. thesis), Cardiff University, August 2014.
[28] C.S. Peskin, B.F. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, J. Comput. Phys. 105 (1993) 33–46.
[29] X.S. Wang, L.T. Zhang, W.K. Liu, On computational issues of immersed finite element methods, J. Comput. Phys. 228 (2009) 2535–2551.
[30] R.K. Noutcheuwa, R.G. Owens, A new incompressible smoothed particle hydrodynamics-immersed boundary method, Int. J. Numer. Anal. Mod. B 3 (2012) 126–167.
[31] A.J. Gil, A. Arranz Carreño, J. Bonet, O. Hassan, The immersed structural potential method for haemodynamic applications, J. Comput. Phys. 229 (2010) 8613–8641.
[32] A.J. Lew, G.C. Guscaglia, A discontinuous–Galerkin based immersed boundary method, Internat. J. Numer. Methods Engrg. 76 (2008) 427–454.
[33] C. Bernardi, Y. Maday, Approximations spectrales de problèmes aux limites elliptiques, in: Mathématiques and Applications, Springer-Verlag, 1992.
[34] C. Bernardi, Y. Maday, Uniform inf–sup conditions for the spectral discretization of the Stokes problem, M3AS 9 (3) (1999) 395–414.
[35] J. Li, J.M. Melenk, B. Wohlmuth, J. Zou, Optimal a priori estimates for higher order finite elements for elliptic interface problems, Appl. Numer. Math. 60 (2010) 19–37.
[36] B.E. Griffith, C.S. Peskin, On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems, J. Comput. Phys. 208 (2005) 75–105.
[37] Y. Mori, Convergence proof of the velocity field for a Stokes flow immersed boundary method, Comm. Pure Appl. Math. LXI (2008) 1213–1263.