Journal of Computational Physics 250 (2013) 178–205

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

An enhanced Immersed Structural Potential Method for fluid–structure interaction

A.J. Gil *, A. Arranz Carreño, J. Bonet, O. Hassan

Civil and Computational Engineering Centre, College of Engineering, Swansea University, Singleton Park, Swansea SA2 8PP, United Kingdom

ARTICLE INFO

Article history:
Received 20 December 2011
Received in revised form 20 April 2013
Accepted 8 May 2013
Available online 24 May 2013

Keywords:
Fluid–structure interaction
Immersed Boundary Method
Immersed Structural Potential Method

ABSTRACT

Within the group of immersed boundary methods employed for the numerical simulation of fluid–structure interaction problems, the Immersed Structural Potential Method (ISPM) was recently introduced (Gil et al., 2010) [1] in order to overcome some of the shortcomings of existing immersed methodologies. In the ISPM, an incompressible immersed solid is modelled as a deviatoric strain energy functional whose spatial gradient defines a fluid–structure interaction force field in the Navier–Stokes equations used to resolve the underlying incompressible Newtonian viscous fluid. In this paper, two enhancements of the methodology are presented. First, the introduction of a new family of spline-based kernel functions for the transfer of information between both physics. In contrast to classical IBM kernels, these new kernels are shown not to introduce spurious oscillations in the solution. Second, the use of tensorised Gaussian quadrature rules that allow for accurate and efficient numerical integration of the immersed structural potential. A series of numerical examples will be presented in order to demonstrate the capabilities of the enhanced methodology and to draw some key comparisons against other existing immersed methodologies in terms of accuracy, preservation of the incompressibility constraint and computational speed.

1. Introduction

The Immersed Boundary Method (IBM) was initially introduced by Peskin [2] with the purpose of studying flow patterns around heart valves. In this fluid–structure interaction (FSI) method, the interaction of an immersed structure within a fluid is modelled by means of a body force field obtained from an assemblage of simplified spring elements mimicking the behaviour of the continuum structure. This force field is then solved within the Navier–Stokes equations representing the underlying incompressible Newtonian viscous fluid.

Partitioned boundary-fitted methods are the well-established alternative to immersed methodologies. These are based on the Arbitrary Lagrangian Eulerian approach [3–5] and require mesh updating algorithms for the movement of the referential fluid mesh. Furthermore, in some cases, where the deformations are extreme and even topological changes emerge, the introduction of an adaptive remeshing technique becomes necessary, leading to prohibitively expensive numerical simulations, especially for three dimensional scenarios. In these cases, immersed methods become extremely competitive from the computational point of view.

* Corresponding author. Tel.: +44 (0)1792602552.
E-mail address: a.j.gil@swansea.ac.uk (A.J. Gil).
URL: http://www.swansea.ac.uk/staff/academic/engineering/gilantonio (A.J. Gil).

0021-9991
http://dx.doi.org/10.1016/j.jcp.2013.05.011

From a methodological point of view, immersed methods can all be framed within the so-called Fictitious Domain (FD) philosophy, introduced by Glowinski et al. in [6] for the resolution of boundary value problems in complex geometrical settings. According to [7], three different FD methods can be identified in the available literature. Firstly, non-body force based schemes [8], where the presence of an immersed solid is established by enforcing strongly that the velocity of the surrounding fluid matches that of the immersed structure at the interface (i.e. Dirichlet Boundary Conditions (BCs) on the fluid); in turn, the movement of the immersed solid is obtained after the equation of motion of the solid is solved subjected to the force field imposed by the surrounding fluid domain (i.e. Neumann BCs on the solid). Secondly, body force based Distributed Lagrange Multiplier (DLM) methods, originally developed in [9] for the simulation of particulate flows with immersed rigid particles and subsequently extended in [10,7] to model deformable solids. In this family of methods, the no-slip velocity constraint at the interface between the physics is imposed as an equation for the Lagrange multiplier defined on the solid boundary. This Lagrange multiplier can be physically understood as a FSI body force field imposed on the background fluid. Thirdly, body force based non-DLM methods, where the no-slip velocity constraint is imposed strongly on the solid (i.e. Dirichlet BCs on the solid) and a FSI body force is then evaluated and applied to the surrounding fluid (i.e. Neumann BCs on the fluid). This FSI body force is obtained from the spatial integration of the stress field which emerges as a result of the deformation experienced by the immersed structure.

As pointed in [9], the IBM introduced by Peskin [2,11,14] and Peskin and McQueen [12,13] belongs to the class of body force based non-DLM FD methods described above. In recent years, there has been a considerable revival of this particular type of immersed approaches for FSI problems, in particular, for biomedical applications [15]. The methodology is no longer restricted to modelling simplified fibre-like solids as in [16–18]. More realistic continuum-like structures can also be analysed due to enhancements introduced in the original methodology which lead to the Extended Immersed Boundary Method (EIBM) originally presented in [19] and the Immersed Finite Element Method (IFEM) pioneered in [20]. In these approaches, the immersed structure is modelled by means of the Finite Element Method, where standard piecewise polynomial shape functions are used to compute the deformation gradient tensor within the immersed structure domain. The nodal displacement field of the structure Finite Element mesh is computed after time integration of the nodal velocity field, which is obtained after suitable interpolation of the velocity field from the background fluid domain with the help of kernel functions defined on the fluid [19,15,21–24]. Both in the EIBM and the IFEM, the force field at the nodes of the deformable solid Finite Element mesh is computed as a result of the spatial integration of the stress tensor through the use of the spatial gradient of standard nodal Finite Element shape functions.

In Ref. [1], the authors introduced the Immersed Structural Potential Method (ISPM) where the immersed structure is modelled by means of a deviatoric strain energy functional. From the spatial discretisation point of view, the structure is modelled as a collection of integration points which deform according to the kinematics defined by the background fluid. In this case, no auxiliary Finite Element mesh is utilised to compute the deformation of the immersed structure. On the contrary, the kinematics of the immersed structure is recovered by means of suitable kernel functions defined on the background fluid domain, used to compute the velocity field and the spatial gradient velocity tensor directly at an integration point level. A structure preserving time integration scheme is then used to compute the deformation gradient tensor at every integration point. This methodology is in line with but distinct from approaches such as the Material Point Method (MPM) [25,26] or the force-projection method presented in [27].

After introducing the continuum immersed formulation from a variationally consistent point of view, this paper aims to establish a comparison between the ISPM and other alternative continuum immersed methodologies, specifically the EIBM or the IFEM [19,20,15,21,23] where the immersed structure is represented in the discrete setting by means of the Finite Element Method. Aspects such as the influence of the interpolation methodology on the structural stresses obtained, structure preserving features of the time–space integrators of each methodology (i.e. conservation of the incompressibility constraint) and speed of computation of the non-linear coupled algorithm, will be discussed in order to emphasise the advantages of our new ISPM approach.

In the Immersed Interface Method (IIM) introduced by LeVeque and Li [28], the spatial discretisation of the fluid equations are re-written to minimise the local truncation error of the discrete equations, formulated in this case by using the Finite Difference Method. The correct implementation of the normal and tangential jumps at the physics interface enables the capturing of the sharp interface. However, this technique is not without computational difficulties, discussed in depth in [27].

Alternative immersed methods (e.g. IBM, IFEM, ISPM, EIBM, MPM) rely upon the introduction of interpolating kernel functions which are used to transfer information (i.e. velocities and FSI forces) between the immersed structure and the background fluid domain. The introduction of these kernel functions leads inevitably to the smearing of the solution variables around the interface region, which some authors have addressed by introducing telescopic refinement strategies [29]. Peskin introduced in [30,12] and revisited in [14] a methodology for the definition of suitable kernel functions to be used in conjunction with the IBM. However, these IBM kernel functions are not necessarily optimal when used in conjunction with alternative immersed methodologies such as the ISPM, where not only the kernel functions but also their spatial gradient must be evaluated as part of the computation of the FSI forces.

In [27], the authors employed piecewise linear finite elements to describe the kinematics of the immersed solid (i.e. for the computation of the deformation gradient tensor). As remarked in [27], the lack of regularity of these isoparametric interpolants (discontinuous derivatives across element interfaces) can lead to force singularities in the computation of the FSI forces. The direct use of this approach would result in the appearance of numerical oscillations in the unknown fields

[22] which can lead to numerical instabilities [27] and, subsequently, lack of robustness of the overall technique. In [27], the authors utilise a posteriori Zienkiewicz–Zhu recovery methodology in the structure domain in order to remove these spurious oscillations from the solution field.

Similar numerical oscillations to those described above also emerge in the ISPM when utilising classical IBM kernels due to their lack of regularity (with discontinuous second derivatives). Furthermore, it is important to remark that the immersed structure stresses are captured in the Lagrangian description and hence, in order to compute them accurately, it is important to ensure that these spurious oscillations are not introduced via the kernel interpolation functions. In this paper, the authors have specifically designed a new family of kernel functions which do not introduce these spurious oscillations. The kernel functions are obtained by taking into account discrete reproducibility conditions as originally introduced by Peskin [14] (in our case, tailor-made for Cartesian staggered grids) and regularity requirements to prevent the appearance of spurious oscillations when computing derivatives. A Maple computer program has been developed to obtain explicit expressions for the new kernels.

However, it is also certain that the regularity requirement when designing the kernel function comes at a price, which can be the widening of the kernel support. The authors have experimented with alternative kernels, with different supports (e.g. $[-3/2,3/2],[-2,2],[-3,3]\ldots$) and regularity conditions (e.g. $\mathcal{C}^2,\mathcal{C}^3,\mathcal{C}^4\ldots$), and have elected to use the one showing the most improved results in terms of absence of spurious oscillations, faster convergence of the fixed point iteration scheme and improved error convergence after using high order quadrature rules whilst maintaining comparable smearing effects to those shown by classical kernel functions.

This paper is broken down into the following sections. In Section 2, the governing equations of the problem are presented in a variational format and the underlying incompressible fluid is discretised, taking advantage of a very efficient low order finite volume scheme set in a Cartesian staggered mesh. Section 3 revisits the formulation of the immersed structural energy potential. In Section 4, the methodology used to compute the immersed solid deformation gradient tensor is re-written to highlight its qualities and allow for an easy theoretical comparison with alternative continuum immersed methodologies, as is presented in Section 5. Section 6 briefly summarises the algorithms in the form of a flowchart for both the ISPM and an alternative continuum immersed methodology. Section 7 starts by reviewing the construction procedure originally proposed by Peskin [11] to define IBM kernel functions. The method is then generalised for any branch-based kernel. Subsequently, a new family of spline-branch-based kernel polynomials is introduced, highlighting some aspects in respect to their Fourier decomposition and their use in conjunction with new quadrature rules. Section 8 presents a range of numerical results in order to demonstrate the aspects described in previous sections of the paper. Finally, in Section 9, some concluding remarks will be made.

2. Governing equations

Let us consider the motion of a continuum defined by means of a mapping $\boldsymbol{\phi}$ established between a reference or material configuration $\boldsymbol{X}\in\Omega_0\subset\mathbb{R}^2$ and a spatial or current configuration $\boldsymbol{x}\in\Omega\subset\mathbb{R}^2$ at time t, namely $\boldsymbol{x}(t)=\boldsymbol{\phi}(\boldsymbol{X},t)$. The deformation gradient tensor $\boldsymbol{F}$ is defined as the material gradient of the spatial position as,

$$\boldsymbol{F}=\nabla_0\boldsymbol{x}=\frac{\partial\boldsymbol{x}}{\partial\boldsymbol{X}},\quad J=\det\boldsymbol{F} \tag{1}$$

where J is the Jacobian of the transformation. In addition, the velocity $\boldsymbol{u}=[u,v]^{\mathrm{T}}$ of the continuum is computed as $\boldsymbol{u}(\boldsymbol{X},t)=\frac{\partial\boldsymbol{x}}{\partial t}$. The conservation of linear momentum for an arbitrary spatial volume Ω is expressed in integral form as,

$$\frac{D}{Dt}\int_\Omega\rho\boldsymbol{u}\,\mathrm{d}v=\int_\Omega\boldsymbol{g}\,\mathrm{d}v+\int_{\partial\Omega}\boldsymbol{t}\,\mathrm{d}a \tag{2}$$

where $\boldsymbol{g}$ denotes an external volume force field per unit of spatial volume and $\boldsymbol{t}=\boldsymbol{\sigma}\boldsymbol{n}$ is the traction vector associated to the Cauchy stress tensor $\boldsymbol{\sigma}$ and an element of area da in the boundary of the current configuration $\partial\Omega$ with outward unit normal $\boldsymbol{n}$. Decomposition of the stress tensor $\boldsymbol{\sigma}$ into its volumetric $-p\boldsymbol{I}$ and deviatoric $\boldsymbol{\sigma}'$ components renders,

$$\frac{D}{Dt}\int_\Omega\rho\boldsymbol{u}\,\mathrm{d}v=\int_\Omega\boldsymbol{g}\,\mathrm{d}v-\int_{\partial\Omega}p\boldsymbol{n}\,\mathrm{d}a+\int_{\partial\Omega}\boldsymbol{\sigma}'\boldsymbol{n}\,\mathrm{d}a \tag{3}$$

Application of the Reynolds' transport theorem (Lie derivative) and the consideration of a Newtonian μ-viscous incompressible continuum leads to,

$$\int_\Omega\frac{\partial}{\partial t}(\rho\boldsymbol{u})\,\mathrm{d}v+\int_{\partial\Omega}(\rho\boldsymbol{u}\otimes\boldsymbol{u}+p\boldsymbol{I}-\mu\nabla\boldsymbol{u})\cdot\boldsymbol{n}\,\mathrm{d}a=\int_\Omega\boldsymbol{g}\,\mathrm{d}v \tag{4}$$

which represents the conservation of linear momentum for an incompressible Newtonian viscous continuum in an integral format. Within the framework of low order Finite Volume schemes [31,32], Ω can be regarded as a control volume where the above vector equation (4) can be re-interpreted according to an Eulerian variational formulation as the following weak form,

$$\delta\boldsymbol{W}^{\Omega}(\boldsymbol{\phi},\delta\boldsymbol{u})=\delta\boldsymbol{W}^{\Omega}_{iner}(\boldsymbol{\phi},\delta\boldsymbol{u})+\delta\boldsymbol{W}^{\Omega}_{int}(\boldsymbol{\phi},\delta\boldsymbol{u})-\delta\boldsymbol{W}^{\Omega}_{ext}(\boldsymbol{\phi},\delta\boldsymbol{u})=0 \tag{5}$$

$$\delta \boldsymbol{W}^{\Omega}_{iner}(\boldsymbol{\phi}, \delta \boldsymbol{u}) = \int_{\Omega} \delta \boldsymbol{u} \cdot \frac{\partial}{\partial t}(\rho \boldsymbol{u})\, \mathrm{d}v + \int_{\partial\Omega} \delta \boldsymbol{u} \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) \cdot \boldsymbol{n}\, \mathrm{d}a \tag{6}$$

$$\delta \boldsymbol{W}^{\Omega}_{int}(\boldsymbol{\phi}, \delta \boldsymbol{u}) = \int_{\partial\Omega} \delta \boldsymbol{u} \cdot (p\boldsymbol{I} - \mu \nabla \boldsymbol{u}) \cdot \boldsymbol{n}\, \mathrm{d}a \tag{7}$$

$$\delta \boldsymbol{W}^{\Omega}_{ext}(\boldsymbol{\phi}, \delta \boldsymbol{u}) = \int_{\Omega} \delta \boldsymbol{u} \cdot \boldsymbol{g}\, \mathrm{d}v \tag{8}$$

where $\delta \boldsymbol{u} = [\delta u, \delta v]^{\mathrm{T}}$ is an arbitrary piecewise constant virtual velocity field with support $\overline{\Omega} = \Omega \bigcup \partial\Omega$. Particularising for the case of a Cartesian staggered mesh, let $\Omega_{u^{A_x}}$ and $\Omega_{v^{A_y}}$ be the control volumes associated with the Cartesian components of the velocity u^{A_x} and v^{A_y}, respectively, with an arrangement similar to that of a Marker And Cell (MAC) grid [33] – see Fig. 1(a) and (b). Here, A_x and A_y denote fluid cell edges perpendicular to the *ox* and *oy* Cartesian axes, respectively, and u^{A_x} and v^{A_y} their corresponding normal edge velocities, for which the above weak form (5) can be split into,

$$\delta W^{\Omega_{u^{A_x}}}(\boldsymbol{\phi}, \delta u^{A_x}) = \int_{\Omega_{u^{A_x}}} \delta u^{A_x} \frac{\partial}{\partial t}(\rho u)\, \mathrm{d}v + \int_{\partial\Omega_{u^{A_x}}} \delta u^{A_x} \boldsymbol{\mathcal{F}}_u \cdot \boldsymbol{n}\, \mathrm{d}a - \int_{\Omega_{u^{A_x}}} \delta u^{A_x} \boldsymbol{g} \cdot \boldsymbol{e}_x\, \mathrm{d}v = 0 \tag{9}$$

$$\delta W^{\Omega_{v^{A_y}}}(\boldsymbol{\phi}, \delta v^{A_y}) = \int_{\Omega_{v^{A_y}}} \delta v^{A_y} \frac{\partial}{\partial t}(\rho v)\, \mathrm{d}v + \int_{\partial\Omega_{v^{A_y}}} \delta v^{A_y} \boldsymbol{\mathcal{F}}_v \cdot \boldsymbol{n}\, \mathrm{d}a - \int_{\Omega_{v^{A_y}}} \delta v^{A_y} \boldsymbol{g} \cdot \boldsymbol{e}_y\, \mathrm{d}v = 0 \tag{10}$$

where $\{\boldsymbol{e}_x, \boldsymbol{e}_y\}$ is the standard Cartesian basis. Eqs. (9) and (10) represent the conservation of linear momentum variables ρu and ρv in a variational integral form with $\mathcal{F}_u$ and $\mathcal{F}_v$ their corresponding interface fluxes, namely,

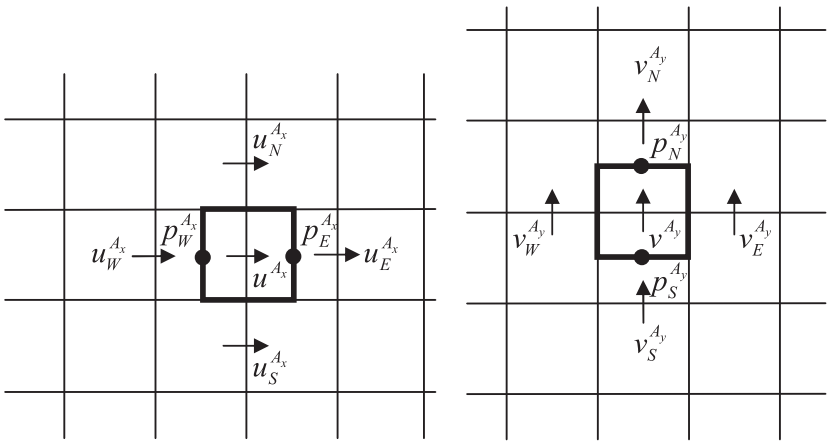
$$\mathcal{F}_u = \rho u \boldsymbol{u} + p\boldsymbol{e}_x - \mu \nabla u, \quad \mathcal{F}_v = \rho v \boldsymbol{u} + p\boldsymbol{e}_y - \mu \nabla v \tag{11}$$

The convective components of the numerical fluxes, that is $\rho u \boldsymbol{u}$ and $\rho v \boldsymbol{u}$, are obtained using a stabilised convective approximation, such as SMART [34], HLPA [35], VONOS [36] or QUICK [37], which minimise numerical diffusion, avoid the creation of spurious oscillations and reduce the total variation of the solution by accounting for the transportive nature of the fluid [38].

3. Immersed structural potential

Let us consider an incompressible deformable solid fully immersed within the surrounding incompressible viscous fluid. Following a numerical immersed strategy [2,11,14], the solid can be modelled as a Helmholtz's free energy density functional Ψ^s whose spatial gradient defines a fluid–structure interaction force field which is regarded as an external source term by the background viscous fluid [1]. In addition, if the background fluid is incompressible and ensuring that the numerical scheme used to solve the background fluid is divergence-free velocity preserving, only the deviatoric component of the solid stress will need to be taken into consideration [1]. Hence, the homogeneous (distortional) component of the solid energy density functional, namely $\widehat{\Psi}^s$ will be defined in terms of the isochoric component of the strain [39–42].

For spatial semi-discretisation purposes, the solid domain is modelled in a Lagrangian manner as a collection of integration points a_p immersed within the fluid, moving from an initial position $\boldsymbol{X}^{a_p}$ to the spatial position $\boldsymbol{x}^{a_p}$ at time instant t,



(a) Control volume $\Omega_{u^{A_x}}$ for velocity u^{A_x} (b) Control volume $\Omega_{v^{A_y}}$ for velocity v^{A_y}

Fig. 1. Control volumes for staggered Cartesian velocity field u and v.

through the deformation gradient tensor $\boldsymbol{F}$ defined by the motion of the surrounding continuum (i.e. non-slip condition). To describe the constitutive behaviour of the structure and within the context of hyperelasticity, a potential energy functional Π^s is then introduced as,

$$\Pi^s(\boldsymbol{\phi}) = \int_{\Omega_0^s} \widehat{\Psi}^s(\boldsymbol{\phi})\, \mathrm{d}V \simeq \sum_{a_p} \widehat{\Psi}^s(\boldsymbol{\phi}^{a_p}) W^{a_p} \tag{12}$$

where $\widehat{\Psi}^s$ is the stored strain energy density functional per unit of undeformed volume Ω_0^s and W^{a_p} is the material or Lagrangian weight associated with a structure integration point a_p. With the purpose of distinguishing the surrounding fluid phase from the immersed solid phase, a superindex $(.)^s$ will be employed when referring to the latter.

It is important to remark that the integration points' parameters (i.e. spatial location $\boldsymbol{X}^{a_p}$ and associated tributary weight W^{a_p}) do not necessarily need to be evaluated from discrete non-overlapping elements after an initial tessellation has been carried out in the solid domain Ω_0^s (in the sense of a Finite Element approach). Alternatively, $\boldsymbol{X}^{a_p}$ and W^{a_p} can be obtained directly from the use of optimal high order Gaussian quadrature rules. Crucially, the latter approach can ensure accuracy of quadrature of the immersed potential $\widehat{\Psi}^s$ and improve speed of computation.

The velocity of the deformable immersed solid can be obtained after suitable definition of an interpolation operator which enables to transfer information from the background Eulerian fluid to the Lagrangian solid. Specifically, the velocity $\boldsymbol{u}$ at any integration point a_p currently at $\boldsymbol{x}^{a_p}$ can be evaluated as follows,

$$\boldsymbol{u}^{a_p} = [u^{a_p}, v^{a_p}]^{\mathrm{T}}, \quad \boldsymbol{u}^{a_p} = \mathcal{I}(\boldsymbol{u})(\boldsymbol{x}^{a_p}) = \left[\sum_{A_x} u^{A_x} \varphi^{A_x}(\boldsymbol{x}^{a_p}), \sum_{A_y} u^{A_y} \varphi^{A_y}(\boldsymbol{x}^{a_p}) \right]^{\mathrm{T}} \tag{13}$$

where

$$\varphi^{A_x}(\boldsymbol{x}) = \varphi(\boldsymbol{x} - \boldsymbol{x}^{A_x}), \quad \varphi^{A_y}(\boldsymbol{x}) = \varphi(\boldsymbol{x} - \boldsymbol{x}^{A_y}) \tag{14}$$

are centred at fluid cell edges A_x and A_y, defined by the spatial position $\boldsymbol{x}^{A_x}$ and $\boldsymbol{x}^{A_y}$, mid-points of their respective fluid cell edges. For a Cartesian Eulerian mesh, it is convenient to formulate these interpolating functions by means of a tensor product expansion as follows,

$$\varphi(\boldsymbol{x}) = \frac{1}{\Delta x \Delta y} \phi\left(\frac{x}{\Delta x}\right) \phi\left(\frac{y}{\Delta y}\right), \quad \boldsymbol{x} = [x, y]^{\mathrm{T}} \tag{15}$$

where ϕ is a one dimensional kernel function [14,43]. Similarly, a virtual velocity field vector $\delta\boldsymbol{u}^s = [\delta u^s, \delta v^s]^{\mathrm{T}}$ evaluated at a structure integration point a_p can also be described as,

$$\delta\boldsymbol{u}^{a_p} = [\delta u^{a_p}, \delta v^{a_p}]^{\mathrm{T}}, \quad \delta\boldsymbol{u}^{a_p} = \mathcal{I}(\delta\boldsymbol{u})(\boldsymbol{x}^{a_p}) = \left[\sum_{A_x} \delta u^{A_x} \varphi^{A_x}(\boldsymbol{x}^{a_p}), \sum_{A_y} \delta u^{A_y} \varphi^{A_y}(\boldsymbol{x}^{a_p}) \right]^{\mathrm{T}} \tag{16}$$

where a consistent interpolating methodology is employed as in Eq. (13), to ensure conservation of the overall scheme. The internal virtual work formulated in the case of the immersed solid domain is defined as the directional derivative of the Helmholtz's free energy functional with respect to a virtual velocity field vector [40,39] as follows,

$$\delta W^s_{int}(\boldsymbol{\phi}, \delta\boldsymbol{u}^s) = \int_{\Omega_0^s} \boldsymbol{\tau}'^s : \nabla\delta\boldsymbol{u}^s \, \mathrm{d}V \simeq \sum_{a_p} W^{a_p} \boldsymbol{\tau}'^{s,a_p} : \nabla\delta\boldsymbol{u}^{a_p} \tag{17}$$

where $\boldsymbol{\tau}'^s$ is the Kirchhoff stress tensor. The evaluation of the above formula (17) requires the computation of the spatial gradient of the virtual velocity at integration point a_p, which can be easily obtained by making use of the spatial gradient of the interpolating kernel functions defined above,

$$\nabla\delta\boldsymbol{u}^{a_p} = \nabla\mathcal{I}(\delta\boldsymbol{u})(\boldsymbol{x}^{a_p}) = \left[\sum_{A_x} \delta u^{A_x} \nabla\varphi^{A_x}(\boldsymbol{x}^{a_p}), \sum_{A_y} \delta v^{A_y} \nabla\varphi^{A_y}(\boldsymbol{x}^{a_p}) \right]^{\mathrm{T}} \tag{18}$$

After re-writing the Kirchhoff stress tensor in the form $\boldsymbol{\tau}'^s = [\boldsymbol{\tau}'^s_x, \boldsymbol{\tau}'^s_y]^{\mathrm{T}}$, Eq. (17) can be reformulated in a continuum manner as,

$$\delta W^s_{int}(\boldsymbol{\phi}, \delta\boldsymbol{u}^s) = \int_{\Omega_0^s} (\boldsymbol{\tau}'^s_x \cdot \nabla\mathcal{I}(\delta u)(\boldsymbol{x}^s) + \boldsymbol{\tau}'^s_y \cdot \nabla\mathcal{I}(\delta v)(\boldsymbol{x}^s)) \, \mathrm{d}V \tag{19}$$

Substitution of Eq. (18) back into (19) yields in a discrete manner,

$$\delta W^s_{int}(\boldsymbol{\phi}, \delta\boldsymbol{u}^s) = \sum_{A_x} \delta u^{A_x} f_x^{A_x} + \sum_{A_y} \delta v^{A_y} f_y^{A_y} \tag{20}$$

where,

$$f_x^{A_x} = \int_{\Omega_0^s} \boldsymbol{\tau}_x'^s \cdot \nabla \varphi^{A_x}(\boldsymbol{x}^s)\, \mathrm{d}V \simeq \sum_{a_p} W^{a_p} \boldsymbol{\tau}_x'^{s,a_p} \cdot \nabla \varphi^{A_x}(\boldsymbol{x}^{a_p}) \tag{21}$$

$$f_y^{A_y} = \int_{\Omega_0^s} \boldsymbol{\tau}_y'^s \cdot \nabla \varphi^{A_y}(\boldsymbol{x}^s)\, \mathrm{d}V \simeq \sum_{a_p} W^{a_p} \boldsymbol{\tau}_y'^{s,a_p} \cdot \nabla \varphi^{A_y}(\boldsymbol{x}^{a_p}) \tag{22}$$

In order to guarantee conservation of the scheme, and after comparing Eqs. (9), (10) and (20), it transpires that,

$$g_x^{A_x} = \frac{f_x^{A_x}}{|\Omega_{u^{A_x}}|}, \quad g_y^{A_y} = \frac{f_y^{A_y}}{|\Omega_{v^{A_y}}|} \tag{23}$$

where $g_x^{A_x}$ and $g_y^{A_y}$ represent the fluid–structure interaction force per unit volume which must be applied at the fluid cell edges A_x and A_y, respectively. As can observed from equations (13), (16) and (18), both the velocity vector and the velocity gradient tensor are sampled directly at integration points a_p within the solid domain. In addition, the Kirchhoff stress tensor is directly integrated back to the background viscous fluid (21) and (22). This methodology, similar to the Material Point Method [25,26,44] or the force projection method presented in [27], differs from existing continuum immersed methodologies [19,15,21,27,23] where an auxiliary Finite Element mesh is used to describe the kinematics of the immersed solid domain with the use of standard piecewise polynomials. In the ISPM, both entities describing the kinematics of the solid, namely velocity $\boldsymbol{v}$ and spatial velocity gradient tensor $\boldsymbol{l}$, are evaluated directly onto the integration points by means of the kernel functions set on the background fluid mesh, which removes the need for utilising an intermediate Finite Element mesh to model the solid and leads to a variationally consistent scheme.

4. Evaluation of the immersed solid deformation gradient tensor $\boldsymbol{F}$

A key ingredient for the evaluation of the stress tensor $\boldsymbol{\tau}'^s$ (21) and (22) is the deformation gradient tensor $\boldsymbol{F}$ at any location within the immersed continuum. In the ISPM [1], a new methodology for the evaluation of $\boldsymbol{F}$ is suggested in contrast to alternative immersed methodologies [19,15,21,23]. First, the spatial velocity gradient tensor $\boldsymbol{l}$ must be evaluated at every integration point a_p, for which the following interpolation operator must be recalled (18):

$$\boldsymbol{l}^{a_p} = \nabla \boldsymbol{u}^{a_p} = \nabla \mathcal{I}(\boldsymbol{u})(\boldsymbol{x}^{a_p}) = \left[\sum_{A_x} u^{A_x} \nabla \varphi^{A_x}(\boldsymbol{x}^{a_p}), \sum_{A_y} v^{A_y} \nabla \varphi^{A_y}(\boldsymbol{x}^{a_p}) \right]^{\mathrm{T}} \tag{24}$$

where the spatial gradient of the kernel functions $\{\nabla \varphi^{A_x}, \nabla \varphi^{A_x}\}$ can be explicitly computed. Second, a time integration scheme is proposed for the tensor system of kinematic differential equations:

$$\dot{\boldsymbol{F}} = \boldsymbol{l}\boldsymbol{F} = (\boldsymbol{d} + \boldsymbol{w})\boldsymbol{F} \tag{25}$$

where $\boldsymbol{d}$ and $\boldsymbol{w}$ are the strain rate tensor and the vorticity tensor, respectively. An iterative explicit time integration scheme can be employed with the purpose of obtaining $\boldsymbol{F}$ at time instant $n+1$ and iteration $k+1$ as follows,

$$\boldsymbol{F}_{k+1}^{n+1} = e^{\Delta t \boldsymbol{l}_{k+1}^{n+1}} \boldsymbol{F}^n \tag{26}$$

Alternative time integration schemes based upon the Magnus' expansion [45] or a generalised exponential map θ-method [46] could also be used. Following [1], a pseudo-polar decomposition is computed as,

$$e^{\Delta t \boldsymbol{l}} = e^{\Delta t(\boldsymbol{d}+\boldsymbol{w})} \simeq e^{\Delta t \boldsymbol{w}} e^{\Delta t \boldsymbol{d}} \tag{27}$$

To compute the first factor in (27), which can be seen as an approximation to an increment in rotation $\Delta \boldsymbol{R}$, we use Hughes–Winget's update [47],

$$e^{\Delta t \boldsymbol{w}} \simeq \Delta \boldsymbol{R} = \left[\boldsymbol{I} - \frac{1}{2}\Delta t \boldsymbol{w} \right]^{-1} \left[\boldsymbol{I} + \frac{1}{2}\Delta t \boldsymbol{w} \right] \tag{28}$$

and given the symmetry of $\boldsymbol{d}$, and hence its normality, we use the spectral decomposition theorem for the second factor, the increment in stretch $\Delta \boldsymbol{U}$,

$$e^{\Delta t \boldsymbol{d}} = \Delta \boldsymbol{U} = \sum_i e^{\lambda_i \Delta t} \boldsymbol{n}_i \otimes \boldsymbol{n}_i \tag{29}$$

where λ_i and $\boldsymbol{n}_i$ are the eigenvalues and eigenvectors of $\boldsymbol{d}$, respectively. It is of prime importance to ensure the incompressibility of the solid to avoid numerical instability of the scheme. Two issues must finally be emphasised: first, the velocity gradient tensor $\boldsymbol{l}$ calculated at an integration point a_p followed from the interpolation procedure depicted in Eq. (24) must be traceless and second, the time-integration algorithm used to solve the tensor system of differential equations (26) must respect this property, namely $J = \det \boldsymbol{F} = 1$. Suitable projection of $\boldsymbol{l}$ in conjunction with the use of the scheme described above [1] guarantees that $J = \det \boldsymbol{F} = 1$ when evaluating the immersed structure stresses. The evaluation of $\boldsymbol{F}$ from the time

integration of $\boldsymbol{l}$ and not from the gradient of the spatial coordinates $\boldsymbol{x}$ enables the ISPM to be understood as a mixed method where both $\boldsymbol{x}$ and $\boldsymbol{F}$ are unknown variables of the immersed structure problem.

In the ISPM, structure volume conservation errors will be introduced by the computation of the spatial velocity gradient tensor $\boldsymbol{l}$ at every structure integration point when using the interpolation operator $\boldsymbol{l}^{a_p} = \nabla \boldsymbol{u}^{a_p} = \nabla \mathcal{I}(\boldsymbol{u})(\boldsymbol{x}^{a_p})$. The use of a fractional step approach in conjunction with a Fast Fourier Transform (FFT) technique in the background fluid enables the satisfaction of the divergence free velocity field to machine accuracy in the fluid cell centres. A linear projection of the tensor $\boldsymbol{l}$ is carried out to the vector space tr $\boldsymbol{l} = 0$ in order to eliminate the error brought about by the kernel interpolation.

The construction of the time integration of the deformation gradient tensor $\dot{\boldsymbol{F}} = \boldsymbol{l}\boldsymbol{F}$ preserves the condition $J = \det \boldsymbol{F} = 1$. Whilst this condition is satisfied in the ISPM, given the fact that an auxiliary solid mesh is not used in this methodology, it is not possible to ensure that the nodes of this auxiliary solid mesh will move in a way that preserves the total volume of this mesh, as it is proposed in Ref. [27]. However, in order to corroborate that this type of more restrictive incompressibility is preserved, an ad hoc Delaunay mesh reconstruction based on the position of the solid integration points has been carried out and the preservation of this ad hoc volume mesh will be shown below to be excellent.

5. Alternative immersed methodology

In alternative immersed methodologies [19,15,21,23], the deformable solid is modelled as a Finite Element Method (FEM) mesh. This way, an interpolation-spreading strategy based upon kernel functions is established exclusively between the background Eulerian fluid cell edges and the immersed nodes of the Finite Element solid mesh. The deformation gradient tensor $\boldsymbol{F}$ is then obtained within the immersed continuum by using standard piecewise polynomial FEM shape functions. Equivalent internal forces are computed at the nodes of the FEM mesh after suitable numerical quadrature of the stress tensor evaluated at Gauss points. The equivalent nodal forces are then spread back to the Eulerian fluid mesh by means of kernel functions ensuring conservation of the scheme. Specifically, the velocity field $\boldsymbol{u}^s$ at the nodes a of the FEM mesh at time step $n+1$ and iteration $k+1$ is obtained after suitable interpolation from the background Eulerian grid as,

$$\boldsymbol{u}^{a,n+1}_{k+1} = \mathcal{I}(\boldsymbol{u}^{n+1}_{k+1})(\boldsymbol{x}^{a,n+1}_{k}) \tag{30}$$

The spatial position of every FEM node a is usually time integrated according to the trapezium rule,

$$\boldsymbol{x}^{a,n+1}_{k+1} = \boldsymbol{x}^{a,n} + \frac{\Delta t}{2}(\boldsymbol{u}^{a,n} + \boldsymbol{u}^{a,n+1}_{k+1}) \tag{31}$$

Making use of the definition of the interpolation operator, the above formula (31) can be re-written as,

$$\boldsymbol{x}^{a,n+1}_{k+1} = \boldsymbol{x}^{a,n} + \frac{\Delta t}{2}[\mathcal{I}(\boldsymbol{u}^{n})(\boldsymbol{x}^{a,n}) + \mathcal{I}(\boldsymbol{u}^{n+1}_{k+1})(\boldsymbol{x}^{a,n+1}_{k})] \tag{32}$$

Following a standard displacement based formulation, the deformation gradient tensor $\boldsymbol{F}$ is evaluated at every Gauss point g_p for every FEM element as,

$$\boldsymbol{F}^{g_p,n+1}_{k+1} = \sum_a \boldsymbol{x}^{a,n+1}_{k+1} \otimes \nabla_0 N^a_{g_p} \tag{33}$$

or as it is customary in a FEM formulation,

$$\boldsymbol{F}^{g_p,n+1}_{k+1} = \sum_a \boldsymbol{x}^{a,n+1}_{k+1} \otimes \left[\sum_b \nabla_\xi N^b_{g_p} \otimes \boldsymbol{X}^b\right]^{-1} \nabla_\xi N^a_{g_p} \tag{34}$$

where N^a are the standard FEM interpolation functions and ∇_ξ symbolises the operator gradient with respect to the isoparametric coordinates in the parent domain. Above formulae (33) and (34) can also be viewed as an alternative integration scheme for the tensor differential equation $\dot{\boldsymbol{F}} = \boldsymbol{l}\boldsymbol{F}$. Indeed, substitution of Eq. (32) into (33) renders after expansion,

$$\boldsymbol{F}^{g_p,n+1}_{k+1} = \sum_a \boldsymbol{x}^{a,n} \otimes \nabla_0 N^a_{g_p} + \frac{\Delta t}{2}\sum_a \mathcal{I}(\boldsymbol{u}^{n})(\boldsymbol{x}^{a,n}) \otimes \nabla_0 N^a_{g_p} + \frac{\Delta t}{2}\sum_a \mathcal{I}(\boldsymbol{u}^{n+1}_{k+1})(\boldsymbol{x}^{a,n+1}_{k}) \otimes \nabla_0 N^a_{g_p} \tag{35}$$

The above formula (35) can be re-written in terms of the spatial gradient of the velocity field $\boldsymbol{l}$ as,

$$\boldsymbol{F}^{g_p,n+1}_{k+1} = \boldsymbol{F}^{g_p,n} + \frac{\Delta t}{2}\boldsymbol{l}^{g_p,n}\boldsymbol{F}^{g_p,n} + \frac{\Delta t}{2}\boldsymbol{l}^{g_p,n+1}_{k+1}\boldsymbol{F}^{g_p,n+1}_{k+1} \tag{36}$$

where the following relationships have been employed.

$$\nabla_0 N^a_{g_p} = \boldsymbol{F}^{\mathrm{T}} \nabla N^a_{g_p}, \quad \boldsymbol{l}^{g_p} = \sum_a \mathcal{I}(\boldsymbol{u})(\boldsymbol{x}^a) \otimes \nabla N^a_{g_p} \tag{37}$$

Finally, expression (36) can be re-written as,

$$\boldsymbol{F}_{k+1}^{g_p,n+1} = \left[\boldsymbol{I} - \frac{\Delta t}{2}\boldsymbol{l}_{k+1}^{g_p,n+1}\right]^{-1}\left[\boldsymbol{I} + \frac{\Delta t}{2}\boldsymbol{l}^{g_p,n}\right]\boldsymbol{F}^{g_p,n} \tag{38}$$

where it can be observed that the deformation gradient tensor is advanced in time after time-integration of the spatial velocity gradient tensor. It is also worth noting that the time integration of $\dot{\boldsymbol{F}} = \boldsymbol{l}\boldsymbol{F}$ is carried out via an explicit scheme of the form $\boldsymbol{F}_{k+1}^{n+1} = \boldsymbol{E}_h\boldsymbol{F}^n$ where,

$$\boldsymbol{E}_h = \left[\boldsymbol{I} - \frac{\Delta t}{2}\boldsymbol{l}_{k+1}^{n+1}\right]^{-1}\left[\boldsymbol{I} + \frac{\Delta t}{2}\boldsymbol{l}^{n}\right] \tag{39}$$

It is easy to show that $\det \boldsymbol{E}_h \neq 1$, thus proving $\boldsymbol{E}_h$ is an incompressibility non-preserving time integration approximation to the exponential $e^{\Delta t\boldsymbol{l}}$. This issue is overcome by the ISPM [1] presented in the previous section or by employing correction constraints such as that presented in [27].

Once the deformation gradient tensor $\boldsymbol{F}$ is known, a constitutive law can then be evaluated to determine the deviatoric Kirchhoff stress tensor in order to compute the equivalent nodal forces at mesh nodes as,

$$\boldsymbol{f}^a = \int_{\Omega_0^s} \boldsymbol{\tau}'^s : \nabla N^a \, \mathrm{d}V \approx \sum_{g_p} J^{g_p}\boldsymbol{\tau}'^{s,g_p} : \nabla N_{g_p}^a \tag{40}$$

where J^{g_p} is the product of the Jacobian and the weight associated with the Gauss point g_p. The spreading of nodal forces $\boldsymbol{f}^a = [f_x^a, f_y^a]^{\mathrm{T}}$ onto the Eulerian fluid can be carried out using operators $\mathcal{S}_x$ and $\mathcal{S}_y$ suitably defined to ensure the conservation of the scheme,

$$f_x^{A_y} = \mathcal{S}_x(\boldsymbol{f})(\boldsymbol{x}^{A_y}) = \sum_a f_x^a \varphi^a(\boldsymbol{x}^{A_y}) \tag{41}$$

$$f_y^{A_x} = \mathcal{S}_y(\boldsymbol{f})(\boldsymbol{x}^{A_x}) = \sum_a f_y^a \varphi^a(\boldsymbol{x}^{A_x}) \tag{42}$$

Formula (23) is finally needed to compute the fluid–structure force per unit volume which must be applied at the fluid cell edges for solution of the Navier–Stokes equations.

6. Algorithmic comparison of the Immersed Structural Potential Method (ISPM) with alternative immersed methodologies

Two flowcharts are displayed in Figs. 2 and 3 to explain the various computational steps needed in both the ISPM and by using an alternative immersed methodology, respectively. In both algorithms, for the sake of simplicity, a fixed point iteration scheme is used to guarantee full coupling between the fluid phase and the solid phase within each time step. A residual norm is computed between the two latest available force terms at iterations k and $k+1$, whereby satisfaction of the coupling between both phases is assessed, leading to either a new iteration $k+2$ or a new time step $n+2$.

In Fig. 3, the evaluation of the deformation gradient tensor $\boldsymbol{F}$ at every Gauss point g_p requires two interpolation stages; first, interpolation of the velocity field $\{u^{A_x}, v^{A_y}\}$ from the fluid mesh to the solid FEM nodes $\boldsymbol{v}^a$ and, second, computation of $\boldsymbol{F}^{g_p}$ by using the gradient of the FEM shape functions $\nabla_0 N_{g_p}^a$. Similarly, the computation of the immersed forces $\{f_x^{A_x}, f_y^{A_y}\}$ at every fluid cell edge is carried out after two spatial integration steps; first, the deviatoric Kirchhoff stress tensor is integrated at every Gauss point $\boldsymbol{\tau}'^{g_p}$ to obtain equivalent internal FEM nodal forces $\boldsymbol{f}^a$ and, second, the forces are then spread to the fluid cell edges.

7. Construction of an improved kernel function for the ISPM

In this section, the construction process of two of the interpolating kernel functions present in the IBM literature [30,43] will be revisited. The analysis will motivate the discussion of their properties and will also prompt the construction of alternative kernels more suitable for use in conjunction with the ISPM, where the evaluation of the gradient of the kernel functions is also necessary for the computation of the velocity gradient tensor $\boldsymbol{l}$ at the immersed structure integration points.

7.1. Kernel construction procedure proposed by Peskin

Since the inception of the IBM [2], it has been crucial to construct one-dimensional kernel functions ϕ. In the original IBM, a cosine type of kernel [2] is employed due to its low computational cost and simplicity. In [30] a new kernel is introduced, setting out some of the key mathematical properties which are necessary for kernels to satisfy when being used with the IBM. We will review here these properties and comment on the limitations and potentials of this method with the purpose of extending this approach when employed in conjunction with the ISPM. The key ingredients introduced by Peskin for the construction of this kernel are as follows [30,14],

- Fixed point iteration, while $\|\boldsymbol{f}_k^{n+1} - \boldsymbol{f}_{k-1}^{n+1}\|/\|\boldsymbol{f}^n\| > tolerance$
  - Compute solid immersed forces
    - Loop over integration points a_p
      1. Interpolate velocity field at structure integration points a_p

$$\boldsymbol{u}_{k+1}^{a_p,n+1} = \mathcal{I}\left(\boldsymbol{u}_{k+1}^{n+1}\right)\left(\boldsymbol{x}_k^{a_p,n+1}\right)$$

      2. Interpolate velocity gradient tensor at structure integration points a_p

$$\boldsymbol{l}_{k+1}^{a_p,n+1} = \left[\boldsymbol{\nabla}\mathcal{I}\left(\boldsymbol{u}_{k+1}^{n+1}\right)\left(\boldsymbol{x}_k^{a_p,n+1}\right)\right]$$

      3. Integrate at $\dot{\boldsymbol{F}} = \boldsymbol{l}\boldsymbol{F}$ according to

$$\boldsymbol{F}_{k+1}^{a_p,n+1} = e^{\left(\Delta t \boldsymbol{l}_{k+1}^{a_p,n+1}\right)}\boldsymbol{F}^{a_p,n}$$

      4. Evaluate constitutive law to obtain deviatoric Kirchhoff stress

$$\boldsymbol{\tau}'^{s,a_p,n+1}_{k+1} = \mathcal{G}(\boldsymbol{F}_{k+1}^{a_p,n+1}, \boldsymbol{l}_{k+1}^{a_p,n+1})$$

      5. Spatial integration of Kirchhoff stress to obtain forces at fluid cell edges

$$f_{i,k+1}^{A_i,n+1} = \sum_{a_p} W^{a_p}\boldsymbol{\tau}_i'^{s,a_p} \cdot \boldsymbol{\nabla}\varphi^{A_i}(\boldsymbol{x}^{a_p}), \quad i = x, y$$

      6. Add inertia and gravity forces if required at fluid cell edges

$$f_{i,k+1}^{ine,A_i,n+1} = \sum_{a_p} W^{a_p}\Delta\rho^{a_p}\left(g_i - \frac{D(\boldsymbol{u}^{a_p}\cdot\boldsymbol{e}_i)}{Dt}\bigg|_{k+1}^{n+1}\right)\varphi^{A_i}\left(\boldsymbol{x}_k^{a_p,n+1}\right), \quad i = x, y$$

      7. Update spatial position of integration points a_p

$$\boldsymbol{x}_{k+1}^{a_p,n+1} = \boldsymbol{x}^{a_p,n} + \frac{1}{2}\Delta t\left(\boldsymbol{u}_{k+1}^{a_p,n+1} + \boldsymbol{u}^{a_p,n}\right)$$

  - Solve Navier-Stokes equations to obtain p_{k+2}^{n+1}, $u_{k+1}^{A_x,n+1}$, $v_{k+1}^{A_y,n+1}$

Fig. 2. Flowchart of the Immersed Structural Potential Method.

1. ϕ is continuous
2. $\text{supp}(\phi) = [-2, 2]$
3. $\sum_j j^k \phi(r - j) = r^k, \quad \forall r \in \mathbb{R},\ k = 0, 1$
4. $\sum_j \phi(r - j)^2 = C, \quad \forall r \in \mathbb{R}$

The first property is highly desirable; however, for use with the ISPM it is necessary to impose higher regularity on the resulting kernel in order to approximate derivative fields accurately. The second property merely requires the resulting kernel to have a compact support, which is extremely useful from the computational point of view in order to reduce the bandwidth of the resulting scheme. The third property is usually referred to as the discrete reproducibility conditions of zeroth and first order, which will be denoted as R_0 and R_1 (i.e. R_k in general for order k discrete reproducibility), respectively. The fourth property was introduced with the purpose of establishing an upper bound on the translation dependence of the kernel in the original IBM, where C is a constant yet to be defined. A complete translation invariant approximation would be ideal, however, as is shown in [30,14] this condition is incompatible with the second property mentioned above (i.e. compact support).

The practical procedure for the determination of this kernel function ϕ is as follows. Firstly, the problem is reduced to the interval $[0, 1]$. In doing so, condition 3 above can be re-written in the following manner,

$$\sum_j j^k \phi_j(r) = r^k, \quad \forall r \in [0, 1],\ k = 0, 1 \tag{43}$$

where the branches ϕ_j are defined as,

- Fixed point iteration, while $\|\boldsymbol{f}_k^{n+1} - \boldsymbol{f}_{k-1}^{n+1}\|/\|\boldsymbol{f}^n\| > tolerance$
  - Compute solid immersed forces
    1. Interpolate velocity field at FEM mesh nodes a:

$$\boldsymbol{u}_{k+1}^{a,n+1} = \mathcal{I}\left(\boldsymbol{u}_{k+1}^{n+1}\right)\left(\boldsymbol{x}_k^{a,n+1}\right)$$

    2. Time integration of the spatial position for node a:

$$\boldsymbol{x}_{k+1}^{a,n+1} = \boldsymbol{x}^{a,n} + \frac{1}{2}\left(\boldsymbol{u}_{k+1}^{a,n+1} + \boldsymbol{u}^{a,n}\right)$$

    3. Loop over elements e
       · Loop over Gauss points g_p for element e
       i. Compute deformation gradient $\boldsymbol{F}$ at Gauss point g_p:

$$\boldsymbol{F}_{k+1}^{g_p,n+1} = \sum_a \boldsymbol{x}_{k+1}^{a,n+1} \otimes \boldsymbol{\nabla}_0 N_{g_p}^a$$

       ii. Compute velocity gradient $\boldsymbol{l}$ at Gauss point g_p:

$$\boldsymbol{l}_{k+1}^{g_p,n+1} = \sum_a \boldsymbol{u}_{k+1}^{a,n+1} \otimes \boldsymbol{\nabla} N_{g_p}^a$$

       iii. Evaluate constitutive law to obtain deviatoric Kirchhoff stress:

$$\boldsymbol{\tau}'^{s,g_p,n+1}_{k+1} = \mathcal{G}(\boldsymbol{F}_{k+1}^{g_p,n+1}, \boldsymbol{l}_{k+1}^{g_p,n+1})$$

       iv. Spatial integration of Kirchhoff stress to obtain forces at each node a:

$$\boldsymbol{f}_{k+1}^{a,n+1} = \sum_{g_p} J^{g_p} \boldsymbol{\tau}'^{s,g_p} \boldsymbol{\nabla} N_{g_p}^a$$

       v. Add inertia and gravity forces if required at each node a:

$$\boldsymbol{f}_{k+1}^{ine,a,n+1} = \sum_{g_p} J^{g_p} \Delta\rho^{g_p} \left(\boldsymbol{g} - \sum_b N_{g_p}^a N_{g_p}^b \frac{D}{Dt}\left(\boldsymbol{u}_{k+1}^{b,n+1}\right)\right)$$

    4. Spread nodal forces back onto the fluid cell edges:

$$f_i^{A_y} = \mathcal{S}_i(\boldsymbol{f})(\boldsymbol{x}^{A_y}), \quad i = x, y$$

  - Solve Navier-Stokes equations to obtain p_{k+2}^{n+1}, $u_{k+1}^{A_x,n+1}$, $v_{k+1}^{A_y,n+1}$

Fig. 3. Flowchart of alternative immersed methodology.

$$\phi_j(r) = \phi(r - j), \quad r \in [0, 1] \tag{44}$$

Using condition 2 above, it is clear that only four branches remain to be determined, namely ϕ_{-1}, ϕ_0, ϕ_1 and ϕ_2. Our aim is to express all branches as a function of the central branch ϕ_0 and then determine the latter in order to obtain the final solution sought ϕ. Unfortunately, note that condition 3 is equivalent to a system of two linear equations involving the four branches and a polynomial of degree 1. In [30,14], this is resolved by replacing condition 3 for the case of $k = 0$ by an even/odd reproducibility condition, gaining an extra equation, as follows,

$$\phi_0(r) + \phi_2(r) = 1/2; \quad \phi_{-1}(r) + \phi_1(r) = 1/2 \tag{45}$$

Thus, the resulting three equations for the four unknown branches are,

$$\phi_0(r) + \phi_2(r) = \frac{1}{2}; \quad \phi_{-1}(r) + \phi_1(r) = \frac{1}{2}; \quad -\phi_{-1}(r) + \phi_1(r) + 2\,\phi_2(r) = r \tag{46}$$

which enables all branches to be expressed in terms of the central branch ϕ_0 to yield,

$$\phi_{-1}(r)=\frac{3}{4}-\frac{r}{2}-\phi_0(r);\quad \phi_1(r)=-\frac{1}{4}+\frac{r}{2}+\phi_0(r);\quad \phi_2(r)=-\phi_0(r)+\frac{1}{2} \tag{47}$$

Substitution of above formulae (47) into the condition 4 above, yields a quadratic equation involving only the central branch ϕ_0 as,

$$\left(\frac{3}{4}-\frac{r}{2}-\phi_0(r)\right)^2+(\phi_0(r))^2+\left(-\frac{1}{4}+\frac{r}{2}+\phi_0(r)\right)^2+\left(-\phi_0(r)+\frac{1}{2}\right)^2=C \tag{48}$$

resulting in,

$$\phi_0(r)=-\frac{r}{4}+\frac{3}{8}\pm\frac{\sqrt{-4r^2+4r-5+16C}}{8} \tag{49}$$

where the constant C remains to be determined. By using the even reproducibility condition for $r=0$, it yields,

$$\phi_0(0)+\phi_{-2}(0)=\frac{1}{2} \tag{50}$$

which implies $\phi_0(0)=\frac{1}{2}$ (as $\phi_{-2}(0)=0$ by construction). Substituting this value of $\phi_0(0)$ into expression (48) leads to a constant $C=3/8$. Finally, a closed expression for the kernel ϕ is reached as follows,

$$\phi(r)=\begin{cases}0 & r\leqslant -2\\ \frac{r}{4}+\frac{5}{8}-\frac{\sqrt{-4r^2-12r-7}}{8} & -2<r\leqslant -1\\ \frac{3}{8}+\frac{r}{4}+\frac{\sqrt{-4r^2-4r+1}}{8} & -1<r\leqslant 0\\ -\frac{r}{4}+\frac{3}{8}+\frac{\sqrt{-4r^2+4r+1}}{8} & 0<r\leqslant 1\\ \frac{5}{8}-\frac{r}{4}-\frac{\sqrt{-4r^2+12r-7}}{8} & 1<r\leqslant 2\\ 0 & r>2\end{cases} \tag{51}$$

As pointed out in [14], this kernel used by Peskin (51) can be very well represented by the original cosine type kernel function [2,14]. Nonetheless, the latter does not satisfy the discrete reproducibility condition of first order (i.e. condition 3 above for $k=1$) and will not be considered hereafter.

7.2. Generalisation of the branch-based approach

The above approach for constructing a kernel prompts the following generalisation. The objective is to construct a kernel function of a certain order of reproducibility m and support $[-L,L]$, where L is a rational number. By introducing $\lfloor x\rfloor$ as the largest integer not greater than x, we can write $L=\frac{q}{r}=\lfloor L\rfloor+\frac{s}{t}$ where $q,r,s,t\in\mathbb{N}$ and $s\leqslant t$ and we can then explore a larger family of branches formulated as follows,

$$\phi_j^k(r)=\phi(r-j),\qquad r\in\left[\frac{k}{t},\frac{k+1}{t}\right],\quad k=0,\ldots,t-1 \tag{52}$$

where each branch is defined in t possible subintervals of $[0,1]$. For example, for the other classical IBM kernel function, the so-called Roma–Peskin kernel [43], the following parameters were employed $t=2$, $L=\frac{3}{2}=1+\frac{1}{2}$ ($s=1$, $t=2$). For discrete reproducibility t systems of $m+1$ equations can be considered as,

$$\sum_j j^n\phi_j^k(r)=r^n;\qquad n=0,\ldots,m;\quad k=0,\ldots,t-1 \tag{53}$$

For system k, we restrict ourselves to the branches satisfying $\left[\frac{k}{t},\frac{k+1}{t}\right]\subset[0,1]$. As the support of $\phi_j^k(r)=\phi(r-j)$ is $[-L+j,L+j]$, only the branches j such that $[-L+j,L+j]\bigcap\left(\frac{k}{t},\frac{k+1}{t}\right)\neq\emptyset$ need to be considered, which is equivalent to satisfying one of the following two inequalities,

$$\begin{aligned}&\frac{k-s}{t}-\lfloor L\rfloor<j<\frac{k+s}{t}+\lfloor L\rfloor\\ &\frac{k+1-s}{t}-\lfloor L\rfloor<j<\frac{k+1+s}{t}+\lfloor L\rfloor\end{aligned} \tag{54}$$

It can be shown that the above inequalities (54) involve from $2\lfloor L\rfloor$ up to $2\lfloor L\rfloor+2$ branches. The objective is to express for each system k, all branches (i.e. unknowns) involved in that system as a linear combination of the central branch ϕ_0^k. For this to happen the rank of the reproducibility system must be one less than the number of branches in that system. In this case, all remaining branches will be of the form

$$\phi_j^k(r)=\alpha\phi_0^k(r)+P_j^k(r) \tag{55}$$

where $\alpha \in \mathbb{R}$ and P_j^k is a polynomial of degree m in r. The advantage of this approach is that it allows the problem of determining the kernel to be split into two subsequent stages.

The first stage involves the solution of k linear systems for the order of reproducibility sought. Once solved, all branches will be written in the above form as functions of the main branches ϕ_0^k. The second stage is the determination of these main branches, subject to whatever constraints are appropriate for the problem. In the case of the two classical IBM kernels derived above [30,43], the main branches ϕ_0^k are the solutions of a quadratic equation. In general, for the branch-based approach for order of reproducibility m, the main branch would be the linear combination of a polynomial of degree m and a square root of a polynomial of degree $2m$.

A graphical representation of Peskin's classical IBM kernel [30] is displayed in Fig. 4(a) along with its first and second derivatives. The lack of regularity of the kernel is clearly observed (discontinuous second derivative) and similar discontinuities are also observed with alternative IBM kernels, such as the Roma–Peskin kernel [43]. Non-optimal results are obtained when these kernel functions are utilised in conjunction with continuum immersed methods (e.g. IFEM, ISPM, EIBM, MPM) where $\boldsymbol{F}$ is evaluated at a continuum level. These effects are even more severe in the ISPM, or similar techniques such as the MPM [25,26,44] or the force projection method presented in [27], due to the use of the gradient of the kernel functions (or piecewise linear shape functions equivalent) to determine the structural stresses. In the ISPM, this effect is clearly observable in stress contour plots and prevents the correct evaluation of the solid stresses. These spurious oscillations, also present in the kinematic fields $\boldsymbol{l}$ and $\boldsymbol{F}$ emerge from the artificial lack of regularity introduced by the use of the standard IBM kernels (see Fig. 5).

It is worth noting that if more sophisticated formulations are considered involving the modelling of an immersed membrane-shell solid domain where the evaluation of a curvature tensor is required, up to second derivatives of the kernel function might be required for the accurate capture of the kinematics of the solid.

7.3. Spline-branch-based new kernel

In this section, a new spline-based kernel, tailor-made to be used in conjunction with the ISPM, is introduced. Following the branch-based approach outlined in the previous section, an appropriate general polynomial will be utilised for each of the selected branches. Crucially, this will remove the spurious oscillations present in the structure stress field in comparison with existing IBM kernels available in the literature. It will also be shown later in Section 8.3 that the use of this new kernel will allow for the use of high-order quadrature rules to integrate the energy potential, which will lead to faster convergence in the computation of FSI forces.

This new spline-based kernel function which satisfies first order reproducibility conditions and yields excellent results was derived considering a wider support of $[-3,3]$. While this consideration increases the computational cost associated with the evaluation of the kernel, particularly for three dimensional problems, it is not necessarily the case in terms of the overall computation time, as this depends on the number of iterations required for the convergence of the fixed point iteration algorithm. In the authors' experience, this number of iterations has been always lower for the kernels introduced in this section than when using the standard IBM kernels presented above. In this case, seventh degree polynomial branches will be employed. As before, the equations required for discrete reproducibility of first order must be fulfilled, namely,

$$-2\,\phi_{-2}(r) - \phi_{-1}(r) + \phi_1(r) + 2\,\phi_2(r) + 3\,\phi_3(r) = r \tag{56a}$$

$$\phi_{-2}(r) + \phi_{-1}(r) + \phi_0(r) + \phi_1(r) + \phi_2(r) + \phi_3(r) = 1 \tag{56b}$$

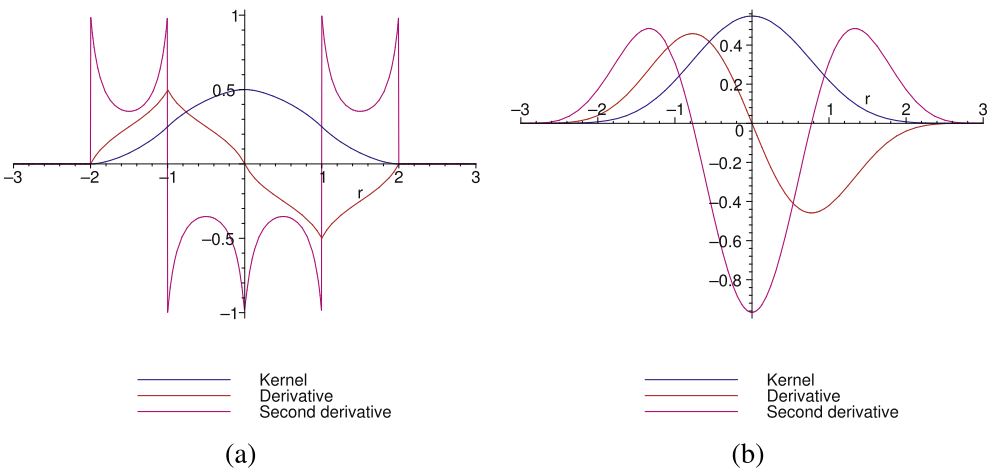


Fig. 4. (a) Peskin's ($\mathcal{C}^1$) kernel, support $[-2,2]$ and (b) proposed new $\mathcal{C}^5$ kernel with first order of discrete reproducibility, support $[-3,3]$. Note the jumps in second derivative for Peskin's kernel.

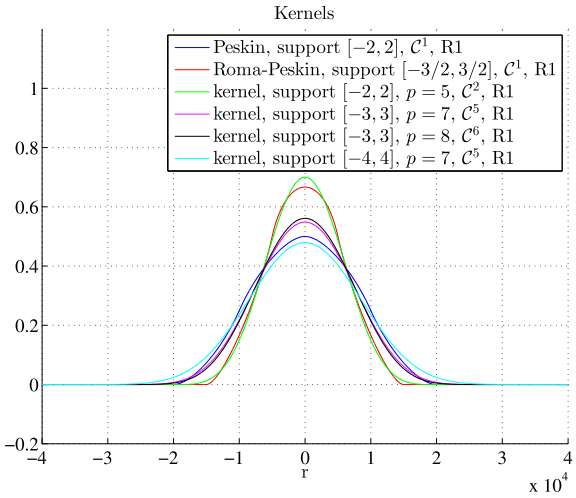


Fig. 5. Alternative IBM and spline-based kernels.

These are two equations involving six unknowns, leading to four indeterminate branches ϕ_0, ϕ_1, ϕ_2 and ϕ_3, while the remaining two can be expressed as follows,

$$\phi_{-2}(r) = -r + 2\,\phi_1(r) + 3\,\phi_2(r) + 4\,\phi_3(r) + \phi_0(r) - 1 \tag{57a}$$

$$\phi_{-1}(r) = r - 3\,\phi_1(r) - 4\,\phi_2(r) - 5\,\phi_3(r) - 2\,\phi_0(r) + 2 \tag{57b}$$

Thus, the expression for a general kernel of support $[-3,3]$ will be of the form,

$$\begin{cases} 0 & r \leqslant -3 \\ \phi_3(r+3) & -3 < r \leqslant -2 \\ \phi_2(r+2) & -2 < r \leqslant -1 \\ \phi_1(r+1) & -1 < r \leqslant 0 \\ \phi_0(r) & 0 < r \leqslant 1 \\ r + 1 - 3\,\phi_1(r-1) - 4\,\phi_2(r-1) - 5\,\phi_3(r-1) - 2\,\phi_0(r-1) & 1 < r \leqslant 2 \\ -r + 1 + 2\,\phi_1(r-2) + 3\,\phi_2(r-2) + 4\,\phi_3(r-2) + \phi_0(r-2) & 2 < r \leqslant 3 \\ 0 & 3 < r \end{cases} \tag{58}$$

Substituting each of the unknown branches by a general polynomial of degree seven and imposing $\mathcal{C}^6$ regularity at $r = 0, \pm 1$ and $\mathcal{C}^5$ regularity at $r = \pm 2, \pm 3$ we arrive at the following unique kernel,

$$\begin{cases} 0 & r \leqslant -3 \\ -\frac{29}{7560} r^7 - \frac{5}{72} r^6 - \frac{21}{40} r^5 - \frac{17}{8} r^4 - \frac{39}{8} r^3 - \frac{243}{40} r^2 - \frac{27}{8} r - \frac{81}{280} & -3 < r \leqslant -2 \\ \frac{17}{1512} r^7 + \frac{1}{8} r^6 + \frac{13}{24} r^5 + \frac{79}{72} r^4 + \frac{65}{72} r^3 + \frac{7}{120} r^2 + \frac{13}{72} r + \frac{1447}{2520} & -2 < r \leqslant -1 \\ -\frac{11}{756} r^7 - \frac{1}{18} r^6 + \frac{7}{36} r^4 - \frac{29}{60} r^2 + \frac{691}{1260} & -1 < r \leqslant 0 \\ \frac{11}{756} r^7 - \frac{1}{18} r^6 + \frac{7}{36} r^4 - \frac{29}{60} r^2 + \frac{691}{1260} & 0 < r \leqslant 1 \\ -\frac{17}{1512} r^7 + \frac{1}{8} r^6 - \frac{13}{24} r^5 + \frac{79}{72} r^4 - \frac{65}{72} r^3 + \frac{7}{120} r^2 - \frac{13}{72} r + \frac{1447}{2520} & 1 < r \leqslant 2 \\ \frac{29}{7560} r^7 - \frac{5}{72} r^6 + \frac{21}{40} r^5 - \frac{17}{8} r^4 + \frac{39}{8} r^3 - \frac{243}{40} r^2 + \frac{27}{8} r - \frac{81}{280} & 2 < r \leqslant 3 \\ 0 & 3 < r \end{cases} \tag{59}$$

Fig. 4(b) displays the graph of this new proposed kernel along with its first and second derivatives. As it can be observed, the new kernel remains an even function without negative tails and it shows higher regularity than that of Peskin's kernel [30].

The spline-branch-based approach was extensively used in conjunction with the symbolic algebraic computer package `Maple` [48] leading to the definition of an entire family of new kernel functions with desirable regularity properties, yet satisfying discrete reproducibility conditions. For the numerical examples presented in this paper, the kernel formulated above (59) was preferred due to its accuracy whilst still remaining computationally efficient.

The new spline-based kernel functions are characterised by their support, the order of regularity ($\mathcal{C}^p$) and the order of reproducibility (R_1). The spline-based kernels (with support $[-2,2]$, $[-3,3]$ and $[-4,4]$) have a much faster decaying Fourier transform than the non-spline-based kernel functions, which translates to quicker dampening of spurious high frequency modes present in the solution. This is particularly relevant when trying to approximate derivatives using these kernels. With

Peskin's [30] and Roma–Peskin's [43] kernels, as the high frequencies still carry an important component, very high oscillations appear in the approximation of magnitudes such as the spatial velocity gradient tensor $\boldsymbol{l}$ and, hence, in the structure stress field. In general, if a function $f(x)$ and its derivatives up to the $(k-1)$-th order are continuous and $d^k f(x)/dx^k$ is piecewise continuous, then the Fourier coefficients c_n decrease as $1/n^k$, so that $\lim_{n\to\infty} n^k c_n = 0$ [49].

Widening of the support whilst respecting the regularity assumption (i.e. support $[-4,4]$, $p=7$, $\mathcal{C}^5$), results in a more favourable spectrum. However, the computational cost associated with the evaluation of the kernel starts to become prohibitive, especially in three dimensional applications. Analogously, increase in the regularity order of the interpolating polynomial while maintaining the same support might be thought of as desirable; for instance, achieving $\mathcal{C}^6$ regularity with support $[-3,3]$ requires order $p=8$. However, the corresponding kernel has a slightly less favourable spectrum for very high frequencies.

Table 1 displays a summary of the most relevant kernel functions described in this section. After extensive numerical experimentation, the best performing kernel for the ISPM (i.e. removing spurious oscillations in the structure stress field) was found to be the one with support $[-3,3]$, $p=7$ and first order of reproducibility, which will be used in subsequent numerical examples. The procedure presented for the construction of spline-based kernels shares some common elements with *B*-splines [50,51], but the key difference between the two approaches is the satisfaction of the discrete reproducibility conditions, that *B*-splines do not meet.

To conclude this section, a final remark on numerical quadrature is vital, especially in connection with the new spline-based kernel functions defined above. A key aspect of the success of immersed methodologies is the accurate description of the immersed structural domain. In the case of the ISPM, this relies upon the accurate spatial integration of the immersed structural potential and, crucial to this is the quadrature rule employed as well as the number of integration points used. In the original IBM or in alternative immersed methodologies constructed on the basis of the Finite Element Method [19,15,21,23], a sufficiently high degree of discretisation of the immersed structure mesh is vital to ensure the accurate computation of interaction between the physics of the problem. Existing references [14,20] propose a heuristic criterion of two discrete structural elements per each underlying discrete fluid element.

In the original introduction of the ISPM [1], a simple procedure was employed for the construction of a low-order quadrature rule for an arbitrary immersed solid domain. Given a solid domain Ω_0^s, a classical simplicial mesh is constructed with any of the many existing mesh generator computer packages, and the corresponding dual mesh is considered (see Fig. 6). Using this approach, the integration points are taken as the nodes of the original mesh, and the integration weights as the tributary areas/volumes of the corresponding dual element to that integration point.

While the above composite prism rule is simple enough and has allowed for validation of the methodology, there exists the possibility of using more efficient approaches, such as high order quadrature rules, that produce more accurate results with fewer integration points. The introduction of such quadrature rules (e.g. tensorised Gaussian, high-order quadrature rules for *n*-simplices, mapped quadratures, etc.) is optional, but allows for the correct integration of the immersed structural potential in a more efficient manner. Note that if the immersed potential is not integrated accurately, regardless of the type of quadrature rule, it cannot be guaranteed that the presence of the immersed structure will be accurately modelled by the coupled FSI system, which might lead in turn to the so-called 'leakage' phenomenon reported in numerous references [14,20,24].

For the examples presented later in this paper, two different high order quadrature rules have been adopted. In the case of a solid rectangular domain, a composite Gauss quadrature rule is adopted, of degrees $[a,b]$ and subintervals $[m,n]$, which is defined by tensorising composite rules of degree a (uniform) with m subintervals in the x-direction and degree b with n subintervals in the y-direction (see Fig. 7(a)). In the case of a solid circular domain, a composite Gauss–Legendre quadrature rule is adopted, of degrees $[a,b]$ and subintervals $[m,n]$, this time defined with respect to a polar coordinates' parametrisation (r,θ) (see Fig. 7(b)).

For more general geometries where a FE mesh is available, it is also possible to consider high-order quadrature rules, as they would be normally used in the context of high-order FEMs [52] (see Fig. 7(c)). It is important to note that the ISPM only needs the position of the integration points and their weights, not the connectivities of the original elements that were used to define them.

Table 1
List of relevant spline-based (SB) and non-spline-based kernel functions. Degree denotes the maximum degree (if applicable) of the polynomial branches that define each kernel. Reproducibility of order k is understood at a discrete level and denoted as R_k.

Kernel name/reference	Code-name	Degree	Regularity	Reproducibility	Support
Peskin cosine [2]	k1	–	$\mathcal{C}^1$	R_0, quasi R_1	$[-2,2]$
Peskin [14]	k2	–	$\mathcal{C}^1$	R_1	$[-2,2]$
Peskin–Roma [43]	k3	–	$\mathcal{C}^1$	R_1	$[-3/2,3/2]$
SB kernel ($p=5$, $\mathcal{C}^2$, $[-2,2]$)	k11	5	$\mathcal{C}^2$	R_1	$[-2,2]$
SB kernel ($p=7$, $\mathcal{C}^5$, $[-3,3]$)	k15	7	$\mathcal{C}^5$	R_1	$[-3,3]$
SB kernel ($p=8$, $\mathcal{C}^6$, $[-3,3]$)	k16	8	$\mathcal{C}^6$	R_1	$[-3,3]$

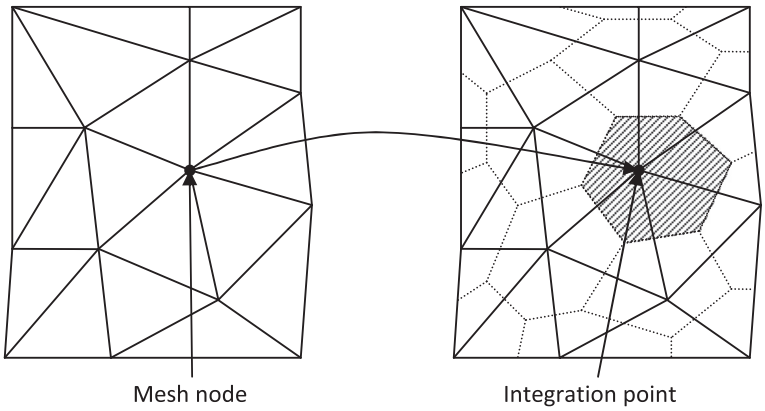


Fig. 6. Construction of dual mesh and assignment of integration point position and weight. Left: classical FEM unstructured triangular mesh. Right: corresponding dual mesh (in dashed line) and associated dual element of the integration point.

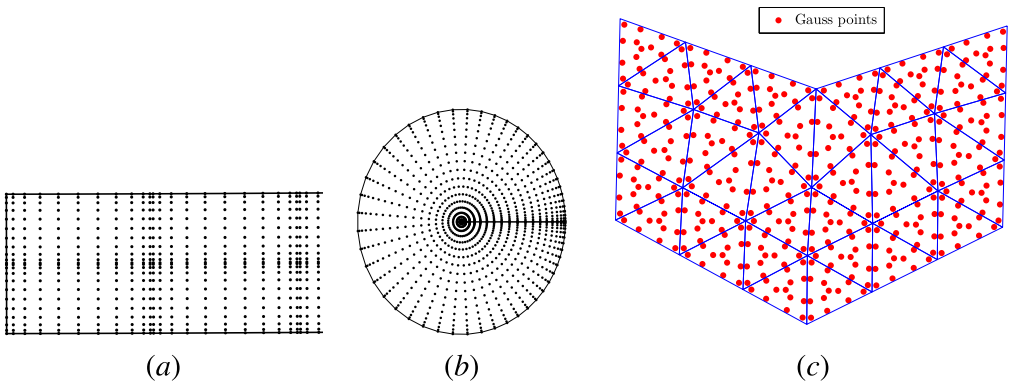


Fig. 7. (a) Composite tensorised Gaussian quadrature for a rectangle; (b) mapped (polar) Gauss–Legendre/Gauss–Lobatto quadrature for a circle; and (c) high-order (degree 7 in the case shown) quadrature rule obtained from a FE mesh.

8. Numerical results

In this section, a variety of numerical examples will be presented in order to illustrate the performance of the proposed methodology. The ISPM will be compared against alternative immersed methodologies in terms of accuracy and numerical efficiency. In addition, a comparison analysis will be drawn between classical IBM non-spline-based and the proposed spline-based kernel functions when used in conjunction with the ISPM.

8.1. Contact of a flapping leaflet in pulsatile flow

This numerical example is introduced to demonstrate that the methodology performs very well against other methods presented in the literature [53,54] within the context of biomedical applications. This problem consists of a rectangular domain filled with a Newtonian incompressible viscous fluid of properties ($\rho = 10^3$ kg/m^3, $\mu = 4 \cdot 10^{-3}$ Pa s) and a thin rectangular incompressible neo-Hookean membrane, attached to the top wall, with density $\rho_s = 10^3$ kg/m^3 and shear modulus $\mu_s = 1.2 \cdot 10^6$ Pa). Fig. 8 displays a geometrical representation of the problem.

For the top and bottom boundaries of the fluid domain, non-slip and symmetry boundary conditions are imposed, respectively. On the left boundary, a Poiseuille (half) profile inflow with pulsatile amplitude of the form ($u = \alpha \sin(2\pi t/T)$, $\alpha = 0.2$, $T = 1$) is considered, whereas on the right boundary, an outflow condition is imposed. This choice of boundary conditions pretends to simulate the closure and total inversion of a two dimensional mitral valve with no chordae tendinae under sinusoidal pulsatile flow. The numerical simulation is carried out by making use of classical IBM and the new spline-based kernel functions (see Table 1) and the discretisation parameters shown in Table 2. The quadrature rule used for the ISPM is constructed from the dual mesh of the FE mesh defined in the Table. Comparisons drawn with the EIBM are performed using the same meshes.

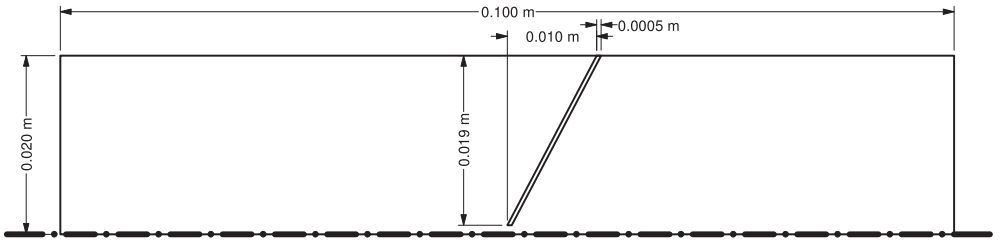


Fig. 8. Geometry for model problem 8.1.

Table 2
Discretisation parameters for problem 8.1. Nodes and elements relate to the FE mesh used to define the solid phase. For the ISPM this mesh will be used to construct a dual-mesh quadrature rule (see Section 7.3). The EIBM can use the mesh as it is. Nodal density is defined as the average number of (solid) nodes per fluid cell.

Run	(n_x, n_y)	Nodes	Elements	Δt	Nodal density
1	(128,32)	243	320	$1.875 \cdot 10^{-5}$	12.5
2	(256,64)	1206	2000	$4.6875 \cdot 10^{-6}$	15.5
3	(512,128)	4509	8000	$1.1719 \cdot 10^{-6}$	14.5
4	(1024,256)	17,017	32,000	$2.9297 \cdot 10^{-7}$	13.6

As can be seen in Fig. 9, during the first half of the sinus cycle, the fluid flows from left to right. This forces the leaflet to move towards the right (see $t = 0$ s, $t = 0.0825$ s), eventually closing the channel (at around $t = 0.1$ s). During a very short time interval, the valve is closed and the leaflet slides along the symmetry wall until it opens ($t = 0.15$ s). Due to the elastic energy stored in the deformed leaflet, this sliding motion is not smooth (i.e. snap-through response), and the leaflet pushes against the wall, vibrates and allows some fluid to slip through the gap. Past $t = 0.15$ s the leaflet fully opens, until $t = 0.25$ s, the time when the flow slows down and it partially goes back towards the left. During the second half of the cycle, an analogous behaviour can be observed, with closure, vibration and fully opened valve in reverse. In terms of the fluid dynamics of the problem (Fig. 9), no differences were observed when the analysis was carried using classical IBM kernels or the alternative EIBM.

Let us now focus on the structural membrane for the solution obtained with the finest fluid and solid discretisations (labelled run 4 in Table 2) for $t = 0.14$ s. At this stage, the membrane is fully open and highly deformed. The objective is to show the removal of spurious oscillations in solid stresses when the ISPM is used. In Figs. 10(a) and 11(a), a closeup of the principal direction associated with the maximum (signed) principal stress is displayed when using the ISPM. A smooth distribution of this vector field is achieved by means of the ISPM (specially, when used in conjunction with the new kernel 'k15'). Note that for the particular section of the membrane shown, the left part is under tension (positive normal stress) whereas the right part is under compression (negative normal stress). This explains the sudden change of orientation of ninety degrees observed in the vector plot.

The corresponding vector field obtained with the EIBM (see Figs. 10(b) and 11(b)) shows a significantly different behaviour. For the particular snapshot shown, the vector fields displays spurious oscillations, regardless of the kernel function utilised. This is clearly an undesirable unrealistic feature which prevents the capture of the smooth transition between the sections in tension and compression of the membrane.

In Fig. 12, the proposed enhancement of the ISPM (using kernel 'k15') is compared to both the EIBM using Peskin's kernel and the original ISPM using Peskin's kernel. Two different kinematic entities are chosen for the sake of comparison, namely the components F_{12} and F_{22} of the deformation gradient tensor. A series of mesh refinements are shown to illustrate convergence of all three methodologies to a common solution (only mesh refinement cases for ISPM/k15 and EIBM/k2 are included to avoid cluttering the figure). While for component F_{12} there are no substantial differences among methodologies, both the ISPM and EIBM show spurious high frequency modes when using Peskin's kernel in F_{22}. As it can be observed in the figure, the new spline-based kernel 'k15' captures the solution, while minimising the appearance of these spurious modes. Results obtained with EIBM and the new kernel 'k15' have not been included in the figure to also avoid cluttering the exposition. As it will be shown in Fig. 15 in Section 8.2, the spurious oscillations that appear in the EIBM cannot be removed by using the new kernel. Therefore the authors do not recommend using the new kernel 'k15' with the EIBM, as it will only increase the computational cost and not improve the quality of the solution.

8.2. The deformation of an elastic wall by flow

The purpose of this section is to validate solutions obtained with the ISPM in combination with the new kernel 'k15' and highlight its improved properties in terms of (i) capture of solid shear stresses when compared to alternative immersed methodologies and (ii) computational efficiency when used in conjunction with high order quadrature rules. The benchmark

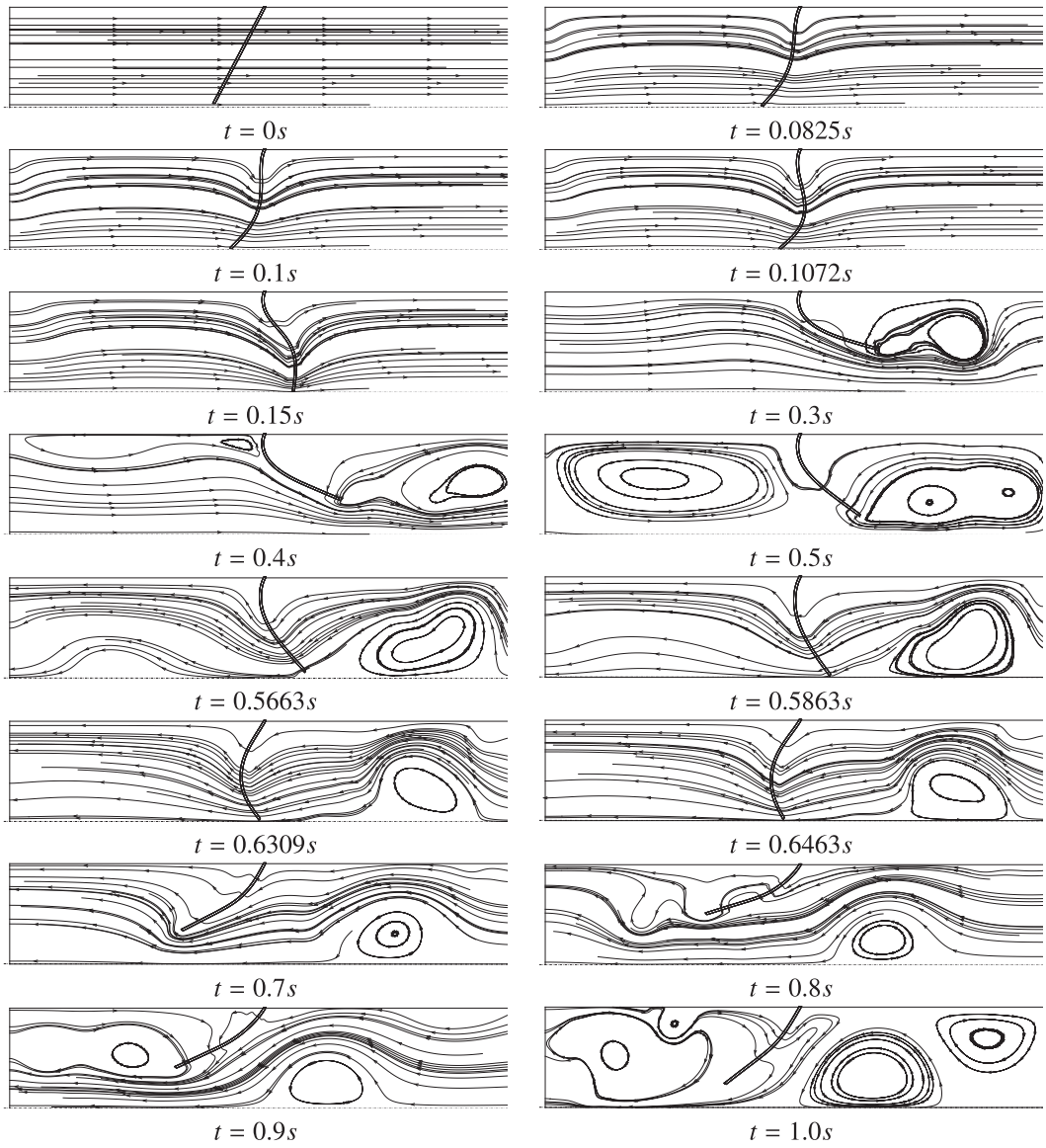


Fig. 9. Streamlines and evolution over time for model problem 8.1 using the new spline-based kernel 'k15'.

problem selected has been already analysed in the literature [27]. The computational domain is the unit square $[0,1]\times[0,1]$, where the upper half is assumed to be fluid of density $\rho = 1$ and viscosity $\mu_f = 10^{-2}$, whereas the bottom half is an elastic incompressible neo-Hookean solid with shear modulus $G = 0.25$. The solid is attached to the bottom of the domain, and the top wall of the fluid has a periodic velocity profile given by

$$u = 0, \quad v = (\cos(2\pi t) - 1)\sin(2\pi x) \tag{60}$$

Periodic boundary conditions are specified at left and right boundaries. The fluid is discretised using a Cartesian staggered grid with 256×256 cells. In all cases presented below $\Delta t = 1.5895 \cdot 10^{-4}$. The discretisation parameters of the elastic wall as well the various quadrature rules employed are covered in Table 3.

In Fig. 13 the velocity field and deformation at $t = 1.0$ s can be observed, which matches solutions present in the literature [27]. Fig. 14 shows that the pressure along the vertical line $x = 0.25$ at time $t = 0.75$ agrees well for a combination of immersed methodologies. More importantly, in the same figure, it can also be observed that the spline-based kernel 'k15' does not introduce any significant additional smoothing of the pressure jump near the interface, regardless of having a larger support. In Fig. 15, both σ_{xx} and σ_{xy} can be observed along the vertical cut $X = 0.25$ at $t = 0.75$ in the solid. As it can be observed from the figure, consistent results are obtained for σ_{xx} for the four cases compared, but a stark difference can be observed for σ_{xy}. Shear stresses show very high frequency oscillations for the EIBM, regardless of whether 'k2' or 'k15' are used. Notice

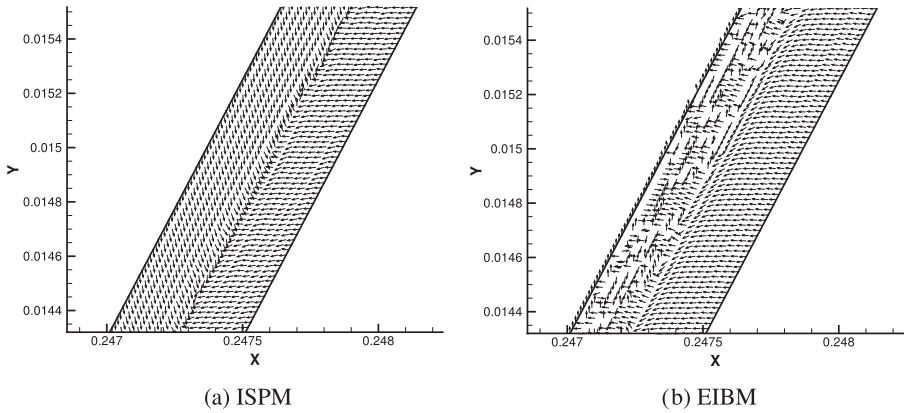


Fig. 10. Closeup of the principal direction for the maximum principal stress (as seen with respect to the material coordinates) for $t = 0.14$ s for model problem 8.1 using Peskin's kernel 'k2'.

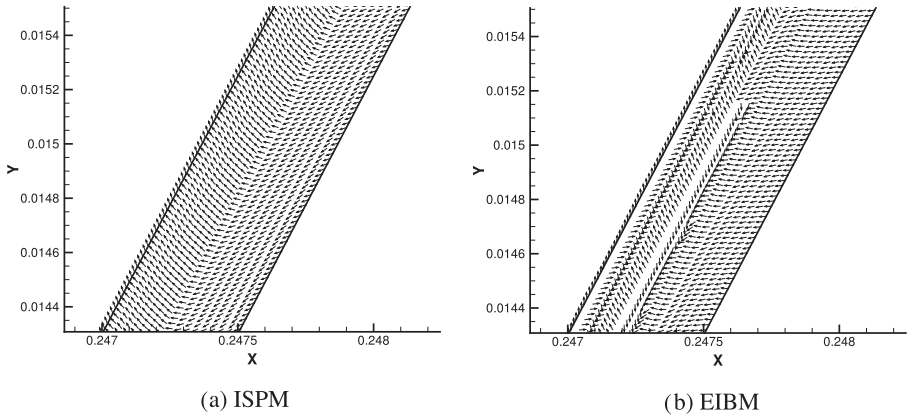


Fig. 11. Closeup of the principal direction for the maximum principal stress (as seen with respect to the material coordinates) for $t = 0.14$ s for model problem 8.1 using the new regular kernel 'k15'.

that in the case of the EIBM with kernel 'k15', even the use of a very fine solid mesh (an approximate equivalent nodal density of 9) does not remove the spurious oscillations. For the ISPM, results obtained with Peskin's kernel 'k2' show noticeable oscillations with smaller frequency. However, the combination of the new kernel 'k15' with the ISPM (even for a very coarse solid discretisation of approximate nodal density 1.5) removes all oscillations in the shear stresses while not introducing any further smoothing or smearing of the solution.

In Fig. 16, a comparison is made in terms of computational cost for solid discretisations presented above in Table 3. The average number of time steps executed per second (defined as the number of current time-step divided by elapsed wall time since start of computation) is shown as a function of the physical simulation time. Note that each time step requires a (possibly) different number of fixed point iterations for convergence, depending on the complexity of the nonlinear problem and the immersed methodology employed. This average number serves as a measure of performance. Clearly, its value at the end of the simulation multiplied by the total number of timesteps gives the total (wall) run time.

In this Fig. 16, the EIBM and ISPM (red[1] and blue lines, respectively) are compared using the same interpolating kernel 'k2' and solid discretisation (an approximate equivalent nodal density of 4). Similar comparisons have been reported by the authors in [1]. The small difference between these two cases is due to the different number of fixed point iterations required for convergence and the different cost in the evaluation of the FSI forces between the EIBM and ISPM. A third graph (green line) has been included with the use of high order quadrature rules for the ISPM with 'k15'. As it can be observed in the figure, regardless of its wider support, this new kernel in conjunction with high order quadrature is able to substantially outperform the other diagrams (i.e. EIBM and ISPM without the use of high order quadratures). The main driving factors for this save in computational time are the reduced number of fixed point iterations as well as the use of optimal quadrature rules, which reduces considerably the number of integration points required (equivalent nodal density of 1.5).

[1] For interpretation of colour in Figs. 4, 5, 7, 12–16, and 18–24, the reader is referred to the web version of this article.

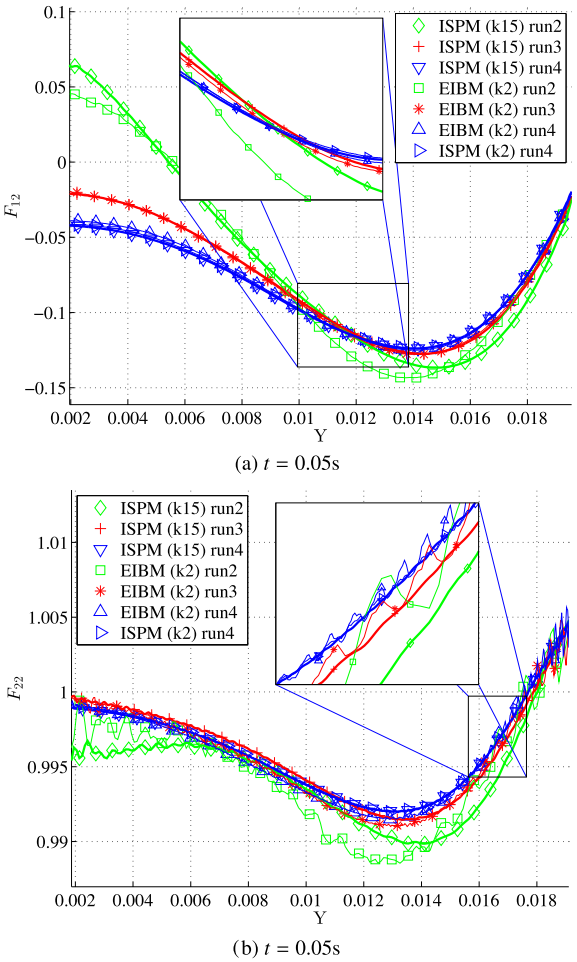


Fig. 12. Cuts along the leading edge of the membrane for: (a) F_{12} at $t = 0.05$; (b) F_{22} at $t = 0.05$ s (as seen with respect to the material coordinates) for model problem 8.1. Thick lines represent solutions obtained with the ISPM and kernel 'k15' for different mesh refinements (runs 2, 3 and 4 in increasing order of refinement). Thin lines represent corresponding solutions obtained with the EIBM and Peskin's kernel ('k2'). The solution obtained with the ISPM and Peskin's kernel and the finest mesh has also been added to allow for comparison with the original ISPM. Discretisation cases are detailed in Table 2. Note that equal line colours correspond to the same discretisation level. Convergence and consistency among solutions can be clearly observed. From the close-ups it can also be noted how the combination of the ISPM with the new kernel 'k15' removes most of the oscillations present in F, which are present in both the EIBM and ISPM while using Peskin's kernel.

Table 3
Discretisation parameters for the deformable elastic wall by flow problem (Section 8.2). In the table, CG stands for composite Gauss quadrature rule, and 'deg $[a, b]$ sub $[m, n]$' denotes a composite rule formed by tensorising composite rules of degree a (uniform) with m subintervals in the x-direction and degree b with n subintervals in the y-direction. By a uniform FEM mesh defined by n_x, n_y we understand the mesh of triangles obtained by uniform subdivision into n_x intervals in the x-direction and n_y in the y-direction and further splitting of the corresponding quadrilateral into two triangles. Such meshes can be used like that for the EIBM, but for the ISPM the corresponding quadrature rule for the dual mesh has to be computed. Note in the table that nodes are interpreted as integration nodes for the case of the ISPM and as mesh nodes for the EIBM. Element numbers are only representative for the EIBM. In the case of the ISPM, elements are only used in visualisation, not computation. Nodal density is defined as the average number of (solid) nodes per fluid cell.

Name	Nodes	Elements	Quadrature rule/mesh type	Nodal density
EIBM/k2	131,841	262,144	Uniform FEM mesh, $n_x = 2 \times 256$, $n_y = 2 \times 128$	4.0235
EIBM/k15	296,065	589,824	Uniform FEM mesh, $n_x = 3 \times 256$, $n_y = 3 \times 128$	9.035
ISPM/k2	131,841	262,144	Uniform dual-FEM mesh, $n_x = 2 \times 256$, $n_y = 2 \times 128$	4.023
ISPM/k15	50,176	99,234	CG, deg $[7, 7]$, sub $[64, 8]$	1.5313

8.3. Tensorised Gaussian quadratures

To illustrate the effect that different quadrature rules in conjunction with a given kernel function have on the quality of the numerical solution, the following two dimensional simple model problem is considered.

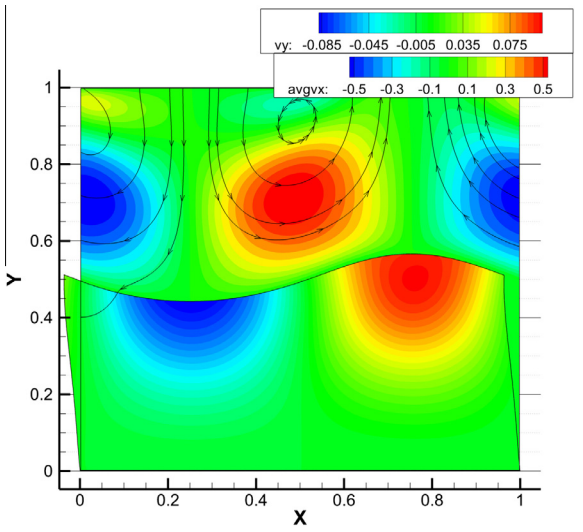


Fig. 13. Streamlines and elastic wall deformation for model problem 8.2 at $t = 1.0$ s. Colours in the fluid/solid denote horizontal/vertical components of the velocity field respectively.

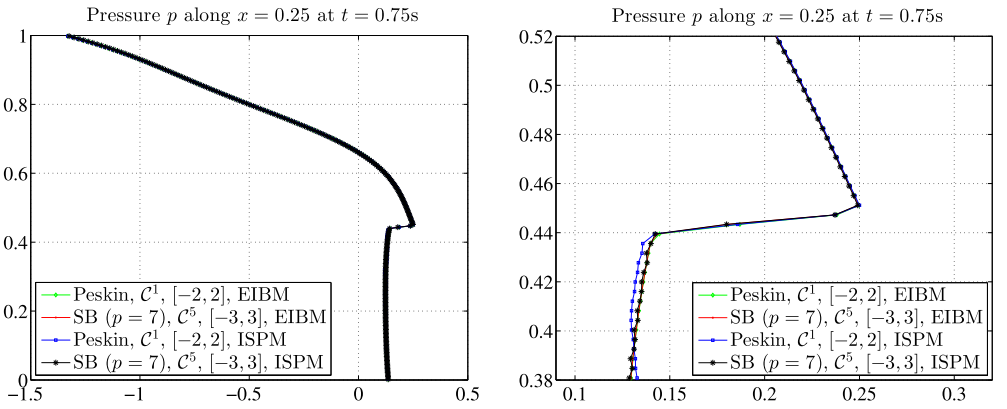


Fig. 14. Elastic wall deformed by flow; pressure along $x = 0.25$ at $t = 0.75$ s. Left: results with the ISPM/EIBM with different kernels are indistinguishable. Right: closeup of the pressure cut. As it can be observed, the new kernel k15 does not introduce significant smoothing of the jump in pressure near the interface when compared to existing immersed methodologies and kernels. In the legend, "SB ($p = 7$), $\mathcal{C}^5$, $(-3,3)$" is the spline-based kernel 'k15'.

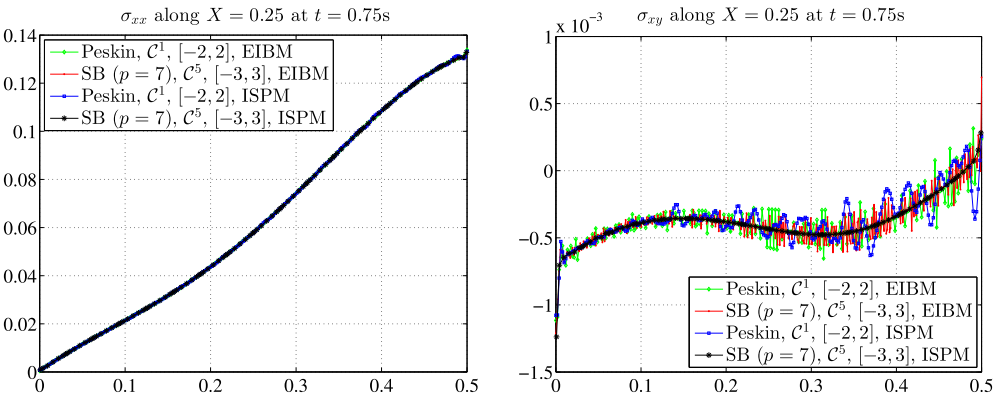


Fig. 15. Elastic wall deformed by flow; stress along $X = 0.25$ at $t = 0.75$ s. Left: σ_{xx} (σ_{yy} shows similar smooth behaviour in this case). Right: σ_{xy}. Results show that the new regular kernel k15 is able to capture a smooth shear stress field, in contrast to existing immersed methodologies. In the legend, "SB ($p = 7$), $\mathcal{C}^5$, $(-3,3)$" is the spline-based kernel 'k15'.

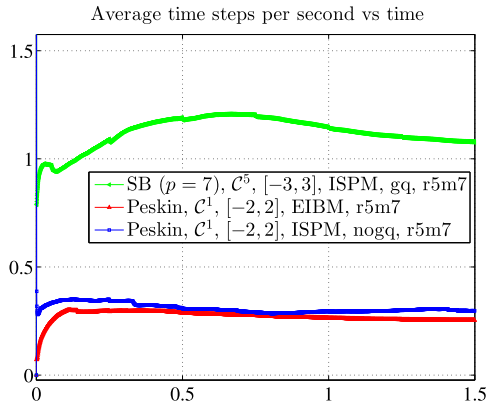


Fig. 16. Average time steps executed per second as a function of the physical simulation time. As it can be observed, the combination of the new kernel k15 is faster than alternative methodologies. Performance changes over simulation time due to the evolving complexity of the nonlinear problem. The small difference in cost between the ISPM and EIBM for Peskin's kernel is due to the different number of fixed point iterations. The new kernel k15 is more expensive per interpolation, but benefits from the use of high order quadrature rules, vastly reducing the number of integration points required to ensure accurate results, rendering the overall scheme more computationally efficient. In the legend, "SB ($p=7$), $\mathcal{C}^5$, $[-3,3]$" is the spline-based kernel 'k15', 'gq' stands for Gaussian quadrature and 'nogq' stands for standard dual-mesh quadrature rule (see Section 7.3).

Let $\Omega = [0, 0.025] \times [0, 0.02]$ m^2 be a rectangular domain filled with a Newtonian incompressible viscous fluid with properties ($\mu = 0.004$ Pa s, $\rho = 10^3$ kg/m^3). For the top and bottom boundaries of the fluid domain, non-slip and symmetry boundary conditions are imposed, respectively. Pulsatile Poiseuille inflow boundary conditions are imposed on the left boundary (half profile with unit velocity at the symmetry line $y = 0$ modulated by amplitude function $U = 0.2\sin(2\pi(t + 10^{-4}))$ m/s) and outflow boundary conditions on the right boundary. A deformable thin membrane is attached to the top wall. The membrane is modelled using a neo-Hookean material model with density $\rho_s = \rho$, and shear modulus $G = 1.2$ GPa. Fig. 17 shows a depiction of the geometry of the problem.

In this example, the error in the evaluation of the FSI forces is computed for the following numerical test cases: (a) One ISPM iteration (see Fig. 19), in order to study the effect of quadrature rule and kernel on the accuracy of computation of FSI forces; (b) 50 fully converged ISPM time steps, with fixed-point iteration until convergence to a set tolerance (see Fig. 20); and (c) 1000 fully converged ISPM time steps, in order to study longer term effects of quadrature rule and kernel, particularly when integration nodes cross kernel definition boundaries (see Fig. 21). The purpose of this numerical experiment will be to quantify the errors incurred in the computation of the FSI forces in the fluid with respect to the degree of the quadrature rule and the kernel function in use. A composite Gauss quadrature rule is adopted for the immersed solid, of degrees $[a, b]$ and subintervals $[m, n]$, as described above.

At first, we will restrict the analysis by considering a series of composite rules *aligned* with both the x- and y-fluid cells. This initial constraint requires that we have as many subintervals for the composite rule as x- and y-fluid cells are being intersected by the membrane. This will ensure that an optimal quadrature rule is used in an interval where the kernel does

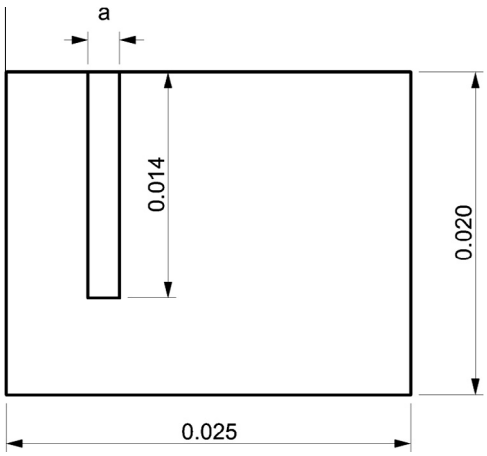


Fig. 17. Geometry for the model problem used to test the influence of different quadrature rules. The width a of the membrane is $7.77 \cdot 10^{-4}$ m for the case of aligned quadrature rules and $8.8 \cdot 10^{-4}$ m for the misaligned. All dimensions in the figure are in metres.

not change definition and also that the membrane fills a whole number of fluid cells. We then utilise in each fluid cell tensorised quadratures constructed from one dimensional Gaussian quadratures from orders 2 to 16. This will translate to an overall composite rule in the membrane with a number of integration points ranging from 1440 to 92,160. The importance of this constraint lies in the fact that the kernels used in the interpolation are piecewise functions, with changes of definition at known places. Aligned rules in this sense are those that respect the change of definition of the kernels. As it will be shown later on, this restriction can be lifted for the new spline-based kernels, providing more flexibility on the construction of quadrature rules.

The relative error in L_∞ norm of the FSI forces evaluated in the fluid cell edges is computed as a function of the number of integration points for both non-spline-based kernels and spline-based kernels. As there is not a closed form solution to this problem, the solutions obtained with the finest solid discretisation for each different simulation are used to compute the error norm in each different graph. As can be seen in Fig. 19, there is a clear advantage in the use of tensorised quadratures, as the error decreases rapidly. For the spline-based kernels presented here, 'k15' and 'k16', one dimensional rule of order 4 (corresponding to 5760 integration points for the membrane) or higher are sufficient to guarantee relative errors in the order of 10^{-8}. In contrast, non-spline-based kernels, 'k2' and 'k3', require rules of at least order 7 in each dimension (leading to 17,640 points) to achieve similar levels of accuracy. Roma–Peskin's kernel 'k3' requires rules of at least order 9, raising the cost to 29,160 integration points.

Note that these quadrature rules are also compatible with the changes in definition of the non-spline based kernel 'k3'. Peskin's kernel changes definition in whole multiples of the mesh size, whereas Peskin–Roma's kernel changes in half-multiples. See Fig. 18 for a pictorial representation of the changes of definition for several kernels.

Spline-based kernels 'k15' and 'k16' benefit more from the use of high-order quadrature rules than the non-spline-based kernels 'k2' and 'k3'. This is clearly due to the fact that both kernels 'k15' and 'k16' are at least $\mathcal{C}^5$, whereas kernels 'k2' and 'k3' are only $\mathcal{C}^1$. Crucially, if these quadrature rules are to be used for a structure domain of arbitrary geometry (i.e. in rectangular subdomains contained within the structure domain), then there is a clear advantage in using the newer spline-based kernels as they incur a cost that is 3–5 times smaller (merely in the number of integration points) than classical IBM non-spline-based kernels.

Let us now consider the effect of these type of quadrature rules after 50 fully converged timesteps. A full timestep in the ISPM involves a fixed point iteration to ensure coupling of both fluid and structure to a given set tolerance. In the following computations, we will consider a tolerance in the relative error in the L_∞ norm of 10^{-9}. Each iteration in the fixed-point loop will then require the evaluation of the FSI forces, which will be subjected to the integration errors described above.

Fig. 20 shows the convergence pattern after 50 fully converged timesteps, where the regularity of the new spline-based kernels poses an advantage with respect to those that lack regularity. An additional comparison with low-order (uniform spacing) quadrature rules has been included in the same figure, which highlights the vast difference in computational cost between using high-order and uniform spacing quadrature rules. Finally, Fig. 21 shows the convergence pattern after 1000 fully converged timesteps. Note that after this number of time steps the membrane has deformed considerably and the nodes used to integrate the energy potential have crossed kernel definition boundaries (see Fig. 18). As it can be observed in the graph, the new spline-based kernels are far more stable and achieve far greater accuracy than non-spline-based kernels even when the quadrature rule is unaligned. The misalignment of the quadrature rule will be the general norm as the movement of the structure through the fluid will produce changing misalignment with respect to the fluid cells. The other interesting

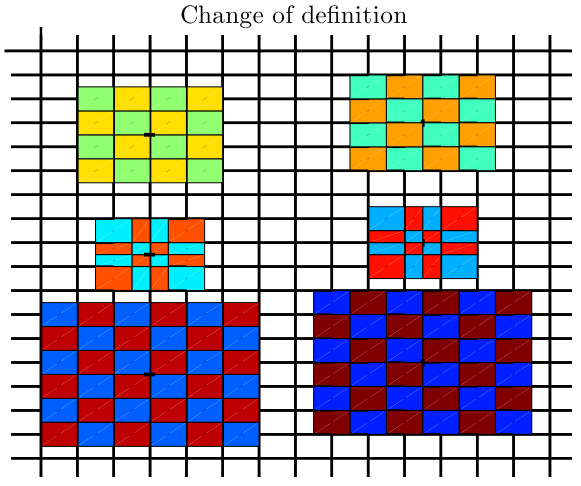


Fig. 18. Change of definition (branch) for each kernel. Each colour-filled cell denotes a region of the computational domain where the kernel is defined by the same functional branch. Boundaries between colours are therefore the locations for possible lack of regularity. Left blocks: kernels centred at vertical edges. Right blocks: kernels centred at horizontal edges. From top to bottom row: Peskin's kernel, Peskin–Roma's kernel and new kernel 'k15'.

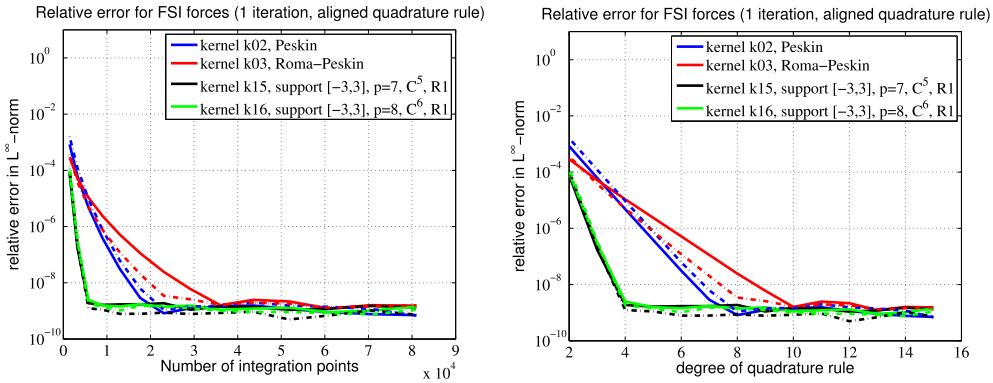


Fig. 19. Relative error in L^∞-norm in the computation of the FSI forces for model problem 8.3 for one iteration. Continuous lines denote errors in the x component and dashed lines in the y components. Left: errors with respect to the overall number of integration points in the membrane. Right: errors with respect to the degree of the Gaussian quadrature rule (in each coordinate direction).

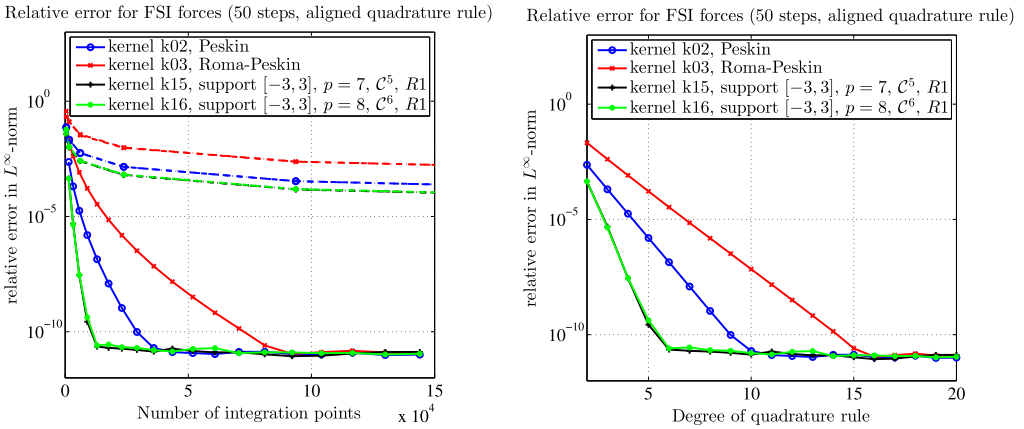


Fig. 20. Relative error in L^∞-norm in the computation of the FSI forces for model problem 8.3 for 50 timesteps. Continuous lines denote errors for the high order quadrature rules, while dashed and dotted lines represent errors when using a quadrature rule with uniform spacing and a dual-mesh based one (see Fig. 6). As it can be observed from the figure, errors in the computation of FSI forces converge substantially faster when high order quadrature rules are employed. Left: errors with respect to the overall number of integration points in the membrane. Right: errors with respect to the degree of the Gaussian quadrature rule.

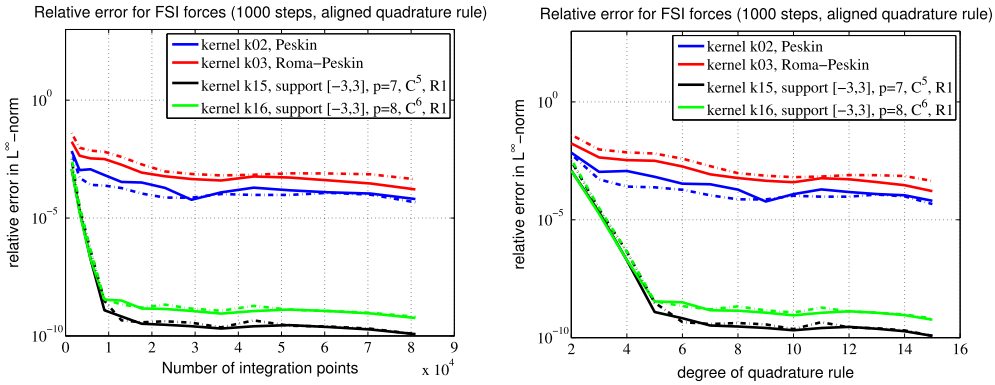


Fig. 21. Relative error in L^∞-norm in the computation of the FSI forces for model problem 8.3 for 1000 timesteps. Continuous lines denote errors in the x component and dashed lines in the y components. Left: errors with respect to the overall number of integration points in the membrane. Right: errors with respect to the degree of the Gaussian quadrature rule (in each coordinate direction).

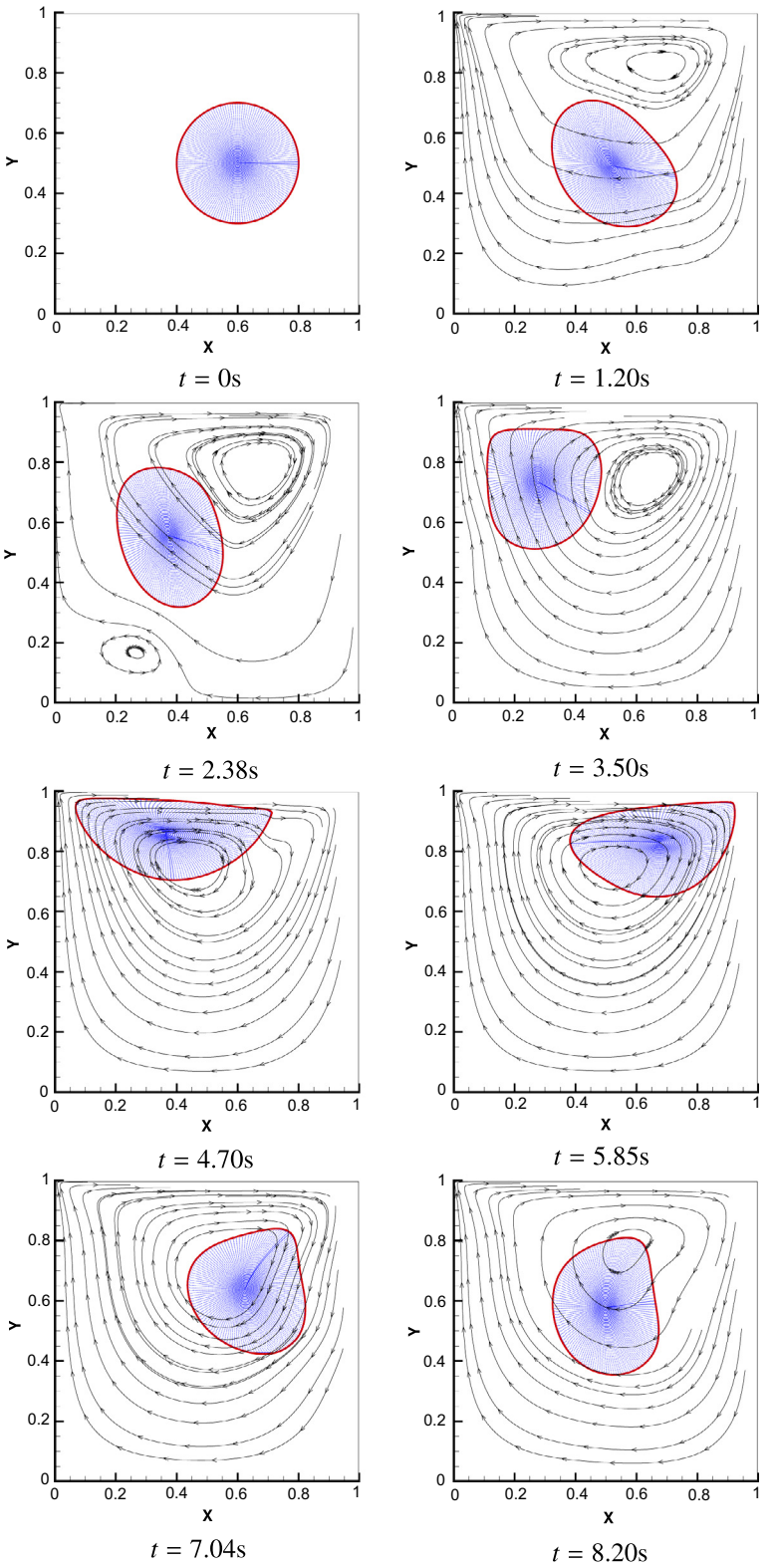


Fig. 22. Streamlines and evolution over time for model problem 8.4 using the new spline-based kernel 'k15' for $G = 0.1$ and spatial discretisation case Q1 in Table 4. Dots in blue are integration points used in the ISPM, line in red is the reference solution obtained with the EIBM and matched against the literature [27]. The small angle in blue has been added to the contour to aid visualisation of the rotation of the disc.

fact that can be observed is that there is faster decay in the error for the spline-based kernels. This is to be expected, as a higher order quadrature rule can exploit the higher regularity of the newer kernels. For arbitrarily complex geometries, including three-dimensional cases, the optimal procedure for construction of quadrature rules is an open problem, but the authors have opted for the initial tessellation of the geometry by rectangles/prisms, which can be efficiently integrated using the tensorised quadrature rules described above, and an additional mesh of n-simplices for the remaining geometry, which can be in turn integrated using high-order quadrature rules customary in the FEM.

Classical IBM non-spline-based kernels are only able to maintain the relative error in the computation of the FSI forces within a tolerance of 10^{-4}, but with a great computational expense. The spline-based kernels 'k15' and 'k16' still show excellent convergence properties.

8.4. A deformable cylinder in a lid-driven cavity

In this section we consider a common benchmark in the context of solvers for fluid–structure interaction problems [27,24,55]. The main objective of this section is not only to show that the ISPM is able to successfully solve the problem and match existing results in the literature, but that it is capable of preserving the volume of the cylinder in a similar way to alternative immersed methodologies. A combination of immersed methodologies will be applied to simulate the motion and deformation of an elastic cylinder in a $[0,1]\times[0,1]$ lid-driven cavity (with $\mu=10^{-2}$, $\rho=1$). The stress-free cylinder of radius 0.2 is initially centred at $(0.6,0.5)$, zero velocity is imposed at all walls, except the top one, where a horizontal velocity $U=1$ is applied (corresponding to $Re=100$). Two cases will be considered, the first where the cylinder is highly deformable ($G=0.1$) and the second where it is almost rigid ($G=10$). The fluid is discretised using a 128×128 Cartesian staggered grid.

The flow streamlines and solid deformation for the deformable and rigid cases using the ISPM and spline-based kernel 'k15' can be visualised in Figs. 22 (spatial discretisation case Q1 in Table 4) and 23 (spatial discretisation case Q3 in Table 4), respectively. As it can be observed from Fig. 22, for the case $G=0.1$, a considerable deformation can be observed, whereas for $G=10$, (see Fig. 23) the cylinder appears almost rigid. In both cases, an almost perfect agreement can be observed with results present in the literature [27,55] and those obtained using the EIBM (red outline in both figures). Results obtained with all the other spatial discretisation cases are almost indistinguishable. It is worth noting that the rigid case can be solved with a coarser spatial discretisation than the highly deformable one.

A thorough parametric study has been performed to quantify the dependence of overall preservation of volume on several factors, namely: time step Δt, nonlinear coupling tolerance ε, refinement of solid and type of solid discretisation. It is possible to show that both Δt and ε play a negligible role in volume preservation in both the highly deformable and rigid cylinder cases, since the largest contribution to the error in volume preservation is produced by spatial discretisation. The errors introduced by temporal integration of the position of nodes/integration points are negligible in comparison due to the small size of the time step since the scheme employed is explicit. The nonlinear coupling tolerance ε plays no role in volume preservation if chosen to be less than 10^{-5}.

A series of spatial discretisations have been considered in order to assess their influence on volume preservation. For the case of the EIBM, several meshes with various degrees and type of refinement have been chosen (see entries M1–M4 in Table 4). Note how cases M1 and M3 are almost uniform meshes (M1 finer, M3 coarser), whereas M2 and M4 have been purposefully refined near the boundary of the cylinder. For the ISPM and kernel 'k15', high order quadrature rules can be used (see entries Q2 and Q4 in Table 4). A composite mapped quadrature (polar coordinates mapping) from the square to the cylinder (circle) is considered (composite mapped Gauss–Legendre rule in both radial and angular coordinates). The combination of the ISPM with Peskin's kernel ('k2') has to resort to either mapped low-order composite quadrature rules (see entries Q1 and Q3 in Table 4) or one constructed from the dual-mesh of a FE mesh. For all immersed methodologies, the same time

Table 4
Discretisation parameters for deformable/rigid cylinder in a lid driven cavity flow (Section 8.4). In the table, CGL stands for composite Gauss–Legendre quadrature rule, 'deg $[a,b]$' denotes (uniform) degrees a and b in the parametric space, and likewise, 'sub $[m,n]$' denotes m and n subdivisions in parametric space (r,θ) for the composite rule. Note in the table that nodes are interpreted as integration nodes for the case of the ISPM and as mesh nodes for the EIBM. Element numbers are only representative for the EIBM. In the case of the ISPM, elements are only used in visualisation, not computation. Nodal density is defined as the average number of (solid) nodes per fluid cell.

Name	Nodes	Elements	Quadrature rule/mesh type	Nodal density
Q1	7500	14,602	CGL, polar mapping, deg $[1,1]$, sub $[50,150]$	3.64
Q2	10,000	19,502	CGL, polar mapping, deg $[5,5]$, sub $[10,40]$	4.857
Q3	2700	5162	CGL, polar mapping, deg $[1,1]$, sub $[30,90]$	1.3114
Q4	3600	6902	CGL, polar mapping, deg $[3,3]$, sub $[10,40]$	1.7485
M1	13,023	25,623	FEM mesh, 210 contour points, $h_{max}=4\cdot10^{-3}$	6.3262
M2	5587	10,441	FEM mesh, 719 contour points, $h_{max}=5\cdot10^{-3}$	2.7137
M3	3329	6405	FEM mesh, 126 contour points, $h_{max}=8\cdot10^{-3}$	1.6176
M4	5798	10,589	FEM mesh, 503 contour points, $h_{max}=8\cdot10^{-3}$	2.8162

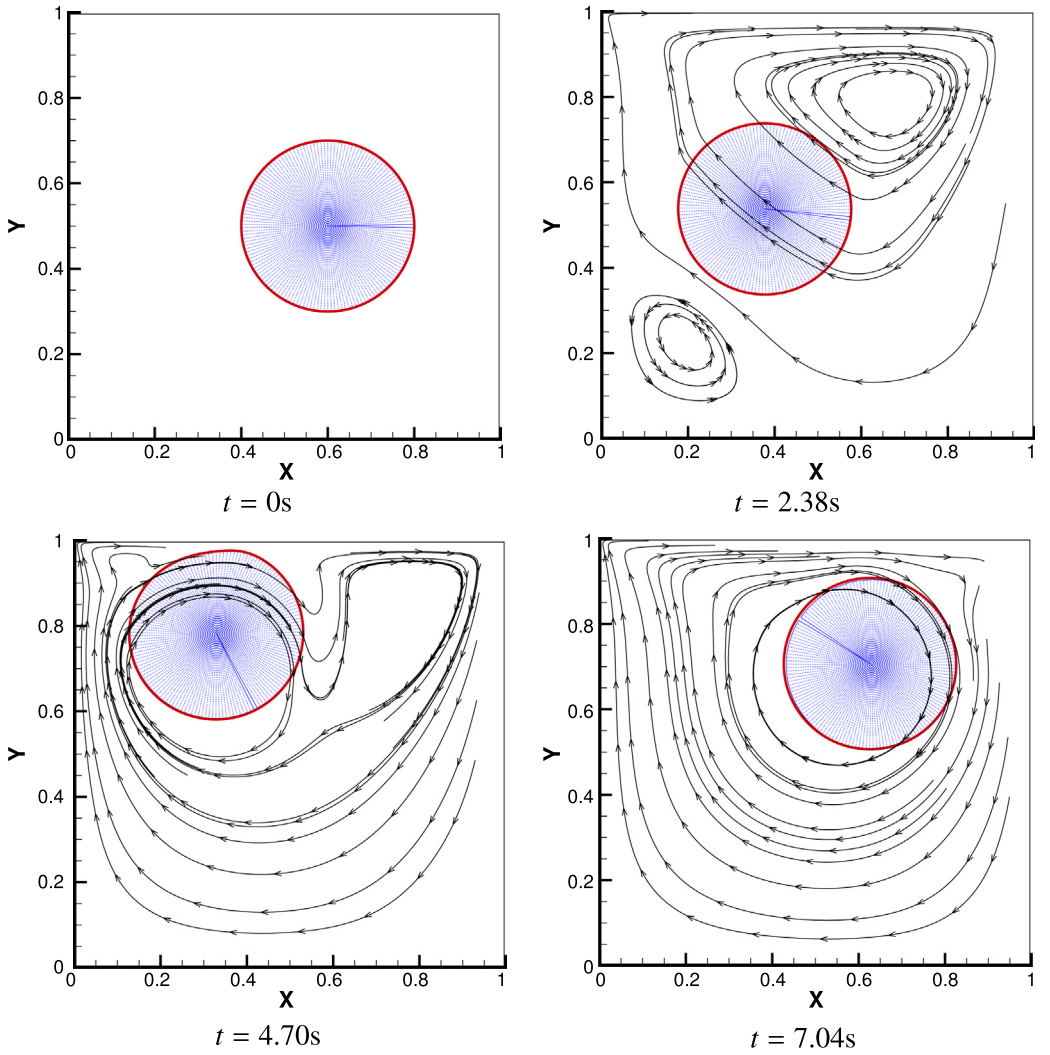
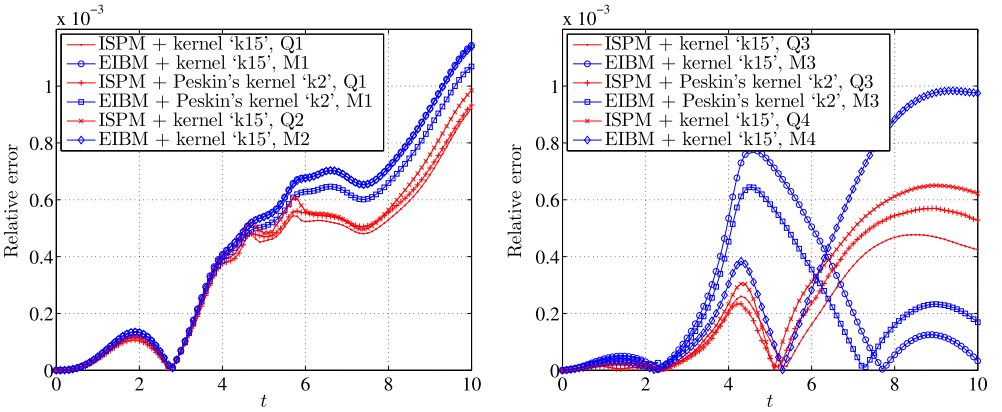


Fig. 23. Streamlines and evolution over time for model problem 8.4 using the new spline-based kernel 'k15' for $G = 10$ and spatial discretisation case Q3 in Table 4. Dots in blue are integration points used in the ISPM, line in red is reference solution obtained with the EIBM and matched against the literature [27]. The small angle in blue has been added to the contour to aid visualisation of the rotation of the disc.



(a) Deformable cylinder $G = 0.1$ (b) Rigid cylinder $G = 10$

Fig. 24. Volume conservation for model problem 8.4 using different immersed methodologies. See Table 4 for the details of spatial discretisation.

integration scheme is used with $\Delta t = 3.1789 \cdot 10^{-4}$ and nonlinear coupling tolerance $\varepsilon = 10^{-7}$, a choice that is optimal (in the sense of minimising the total number of iterations) and produces independent volume preservation results from Δt and ε.

In Fig. 24, the evolution over time of the relative error in volume conservation for the cylinder is depicted. In the case of the ISPM, quadrature points have been deliberately placed on the boundary of the solid domain in order to calculate in an ad hoc manner (using an auxiliary Delaunay mesh which connects these integration points) the volume of the immersed structure. A comparison is made between the ISPM and EIBM with Peskin's original kernel and the new spline-based kernel 'k15' for both methodologies and several spatial discretisations, as it has been remarked earlier that neither Δt nor ε have any significant influence in the above curves. As it can be observed for both cases, the ISPM preserves the volume of the structure in an order of magnitude similar to that of the EIBM for this particular problem. The comparison with the EIBM in combination with the new kernel 'k15' has been included just for completeness. It is important to remark that this kernel has not been designed for the EIBM, and as it has been shown in other examples: the EIBM does not benefit from it.

9. Conclusions

In this paper, the Immersed Structural Potential Method recently introduced in [1] is further enhanced and optimised by means of two new updates. First, a key improvement is the introduction of a new family of spline-based kernel functions satisfying prescribed reproducibility conditions but with improved order of regularity. These new kernels show excellent numerical results by removing spurious oscillations within the solution field (and its spatial derivatives) in comparison with those obtained through alternative classical non-spline based IBM kernel functions. Second, this new family of kernels can take full advantage of the use of highly efficient high-order quadrature rules for the numerical integration of the immersed structural potential, rendering optimal convergence pattern (even for low order of quadrature). The methodology has also been presented from a complete variational point of view (i.e. both fluid and structure) making reference to existing immersed methodologies, in order to draw some comparisons in terms of accuracy, preservation of the incompressibility constraint and computational speed. In the authors' opinion, a more efficient coupling algorithm, replacing the existing fixed point iteration method, is the final aspect necessary to optimise the performance of the methodology. This new coupling methodology within the context of three dimensional applications is the objective of a subsequent publication.

Acknowledgements

This work was financially supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through the Grant EP/F03010X, whose support is greatly acknowledged.

References

[1] A.J. Gil, A. Arranz Carreño, J. Bonet, O. Hassan, The Immersed Structural Potential Method for haemodynamic applications, Journal of Computational Physics 229 (22) (2010) 8613–8641.
[2] C.S. Peskin, Flow patterns around heart values: a numerical method, Journal of Computational Physics 10 (1972) 252–271.
[3] T.E. Tezduyar, M. Behr, J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces. The deforming-spatial-domain/space time procedure: I. The concept and the preliminary numerical tests, Computer Methods in Applied Mechanics and Engineering 94 (1992) 339–351.
[4] W. Dettmer, D. Perić, A computational framework for fluid-structure interaction: finite element formulation and applications, Computer Methods in Applied Mechanics and Engineering 195 (2006) 5754–5779. John H. Argyris Memorial Issue. Part II.
[5] C. Wood, A.J. Gil, O. Hassan, J. Bonet, Partitioned block-Gauss–Seidel coupling for dynamic fluid–structure interaction, Computers and Structures (2008), http://dx.doi.org/10.1016/j.compstruc.2008.08.005.
[6] R. Glowinski, T.-W. Pan, J. Périaux, A fictitious domain method for Dirichlet problems and applications, Computer Methods in Applied Mechanics and Engineering 111 (1994) 283–303.
[7] Z. Yu, A DLM/FD method for fluid/flexible-body interactions, Journal of Computational Physics 207 (2005) 1–27.
[8] D.J.J. Farnell, T. David, D.C. Barton, Numerical simulations of a filament in a flowing soap film, International Journal for Numerical Methods in Fluids 44 (3) (2004) 313–330.
[9] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, International Journal of Multiphase Flow 25 (1999) 755–794.
[10] F.P.T. Baaijens, A fictitious domain/mortar element method for fluid–structure interaction, International Journal for Numerical Methods in Fluids 35 (7) (2001) 743–761.
[11] C.S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (1977) 220–252.
[12] C.S. Peskin, D.M. McQueen, A three-dimensional computational method for blood flow in the heart. I. Immersed elastic fibers in a viscous incompressible fluid, Journal of Computational Physics 81 (1989) 372–405.
[13] C.S. Peskin, D.M. McQueen, A three-dimensional computational method for blood flow in the heart. II. Contractile fibers, Journal of Computational Physics 82 (1989) 289–297.
[14] C.S. Peskin, The immersed boundary method, Acta Numerica 11 (479–517) (2002).
[15] W.K. Liu, Y. Liu, D. Farrell, L. Zhang, X.S. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, J. Lee, J. Hong, X. Chen, H. Hsu, Immersed finite element method and its applications to biological systems, Computer Methods in Applied Mechanics and Engineering 195 (2006) 1722–1749.
[16] P.N. Watton, X.Y. Luo, X. Wang, G.M. Bernacca, P. Molloy, D.J. Wheatley, Dynamic modelling of prosthetic chorded mitral valves using the immersed boundary method, Journal of Biomechanics 40 (3) (2007) 613–626.
[17] Y.R. Yamada, C.S. Peskin, A look-ahead model for the elongation dynamics of transcription, Biophysical Journal 96 (8) (2009) 3015–3031.
[18] B.E. Griffith, X. Luo, D. McQueen, C. Peskin, Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method, Journal of Applied Mechanics 1 (2009) 137–177.
[19] X. Wang, W.K. Liu, Extended immersed boundary method using FEM and RKPM, Computer Methods in Applied Mechanics and Engineering 193 (12–14) (2004) 1305–1321. Meshfree Methods: Recent Advances and New Applications.

[20] L. Zhang, A. Gerstenberger, X. Wang, W.K. Liu, Immersed finite element method, Computer Methods in Applied Mechanics and Engineering 193 (21–22) (2004) 2051–2067. Flow Simulation and Modeling.
[21] W.K. Liu, D.W. Kim, S. Tang, Mathematical foundations of the immersed finite element method, Computational Mechanics 39 (3) (2007) 211–222.
[22] H. Wang, J. Chessa, W.K. Liu, T. Belytschko, The immersed/fictitious element method for fluid–structure interaction: volumetric consistency, compressibility and thin members, International Journal for Numerical Methods in Engineering 74 (1) (2008) 32–55.
[23] X.S. Wang, L.T. Zhang, W.K. Liu, On computational issues of immersed finite element methods, Journal of Computational Physics 228 (7) (2009) 2535–2551.
[24] X. Wang, L. Zhang, Interpolation functions in the immersed boundary and finite element methods, Computational Mechanics 45 (2010) 321–334, http://dx.doi.org/10.1007/s00466-009-0449-5.
[25] D. Sulsky, J.U. Brackbill, A numerical method for suspension flow, Journal of Computational Physics 96 (2) (1991) 339–368.
[26] D. Sulsky, Z. Chen, H.L. Schreyer, A particle method for history-dependent materials, Computer Methods in Applied Mechanics and Engineering 118 (1–2) (1994) 179–196.
[27] H. Zhao, J.B. Freund, R.D. Moser, A fixed-mesh method for incompressible flow–structure systems with finite solid deformations, Journal of Computational Physics 227 (2008) 3114–3140.
[28] R.J. LeVeque, Z. Li, Immersed interface method for Stokes flow with elastic boundaries on surface tension, SIAM Journal on Scientific Computing 18 (3) (1997) 709–735.
[29] B.E. Griffith, R.D. Hornung, D.M. McQueen, C.S. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, Journal of Computational Physics 223 (1) (2007) 10–49.
[30] C.S. Peskin, D.M. McQueen, Fluid dynamics of the heart and its valves, in: H.G. Othmer, F.R. Adler, M.A. Lewis, J.C. Dallon (Eds.), Case Studies in Mathematical Modeling – Ecology, Physiology, and Cell Biology, Prentice-Hall, Inc., 1997, pp. 309–337 (Chapter 14).
[31] J.D. Anderson, Computational Fluid Dynamics: The Basics with Applications, McGraw-Hill, 1995.
[32] C. Hirsch, Numerical Computation of Internal and External Flows: Computational Methods for Inviscid and Viscous Flows, vol. II, Wiley, and Sons, 1988.
[33] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free interface, The Physics of Fluids 8 (1965) 2182–2189.
[34] P.H. Gaskell, A.K.C. Lau, Curvature-compensated convective transport: SMART, a new boundedness- preserving transport algorithm, International Journal for Numerical Methods in Fluids 8 (6) (1988) 617–641.
[35] J. Zhu, On the higher-order bounded discretization schemes for finite volume computations of incompressible flows, Computer Methods in Applied Mechanics and Engineering 98 (3) (1992) 345–360.
[36] A. Varonos, G. Bergeles, Development and assessment of a variable-order non-oscillatory scheme for convection term discretization, International Journal for Numerical Methods in Fluids 26 (1) (1998) 1–16.
[37] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Computer Methods in Applied Mechanics and Engineering 19 (1) (1979) 59–98.
[38] H.K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics: The Finite Volume Method, second ed., Pearson Education Limited, 2007.
[39] J. Bonet, R.D. Wood, Nonlinear Continuum Mechanics for Finite Element Analysis, second ed., Cambridge University Press, 2008.
[40] G.A. Holzapfel, Nonlinear Solid Mechanics, Wiley, 2000.
[41] R.W. Ogden, Non-linear Elastic Deformations, Dover, 1997.
[42] T. Belytschko, W.K. Liu, B. Moran, Nonlinear Finite Elements for Continua and Structures, John Wiley and Sons, 2000.
[43] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, Journal of Computational Physics 153 (2) (1999) 509–534.
[44] A.R. York II, D. Sulsky, H.L. Schreyer, Fluid–membrane interaction based on the material point method, International Journal for Numerical Methods in Engineering 48 (6) (2000) 901–924.
[45] W. Magnus, On the exponential solution of differential equations for a linear operator, Communications on Pure and Applied Mathematics 7 (4) (1954) 649–673.
[46] E. de Souza Neto, D. Perić, D.R.J. Owen, Computational Methods for Plasticity: Theory and Applications, Wiley, 2008.
[47] T.J.R. Hughes, J. Winget, Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis, International Journal for Numerical Methods in Engineering 15 (12) (1980) 1862–1867.
[48] Maplesoft, Maple User Manual, 2011.
[49] R.J. Beerends, H.G. ter Morsche, J.C. van den Berg, J.C. van de Vrie, Fourier and Laplace Transforms, Cambridge University Press, 2003.
[50] L. Piegl, W. Tiller, The NURBS Book, Monographs in Visual Communication, Springer, New York, 1995.
[51] G. Farin, Curves and Surfaces for Computer Aided Geometric Design, second ed., Academic, New York, 1990.
[52] P. Solin, K. Segeth, I. Dolezel, Higher-Order Finite Element Methods, Chapman and Hall/CRC, New York, NY, 2003.
[53] R. van Loon, P.D. Anderson, F.N. van de Vosse, A fluid–structure interaction method with solid-rigid contact for heart valve dynamics, Journal of Computational Physics 217 (2) (2006) 806–823.
[54] R. van Loon, A 3D method for modelling the fluid–structure interaction of heart valves, Ph.D. thesis, Eindhoven University of Technology, 2005.
[55] K. Sugiyama, S. Ii, S. Takeuchi, S. Takagi, Y. Matsumoto, Full Eulerian simulations of biconcave neo-Hookean particles in a Poiseuille flow, Computational Mechanics 46 (1) (2010) 147–157.