

Using DC PSE operator discretization in Eulerian meshless collocation methods improves their robustness in complex geometries



George C. Bourantas^{a,b,d,1}, Bevan L. Cheeseman^{a,b,d,1}, Rajesh Ramaswamy^c, Ivo F. Sbalzarini^{a,b,d,*}

^a Chair of Scientific Computing for Systems Biology, Faculty of Computer Science, TU Dresden, 01069 Dresden, Germany

^b Max Planck Institute of Molecular Cell Biology and Genetics, Pfotenhauerstr. 108, 01307 Dresden, Germany

^c Max Planck Institute for the Physics of Complex Systems, Nöthnitzerstr. 38, 01187 Dresden, Germany

^d MOSAIC Group, Center for Systems Biology Dresden, Pfotenhauerstr. 108, 01307 Dresden, Germany

ARTICLE INFO

Article history:

Received 18 May 2015

Revised 27 November 2015

Accepted 8 June 2016

Available online 16 June 2016

Keywords:

Incompressible steady state

DC PSE

MLS

Meshless point collocation

Particle methods

Complex geometry

Velocity-correction

Discretization correction

ABSTRACT

Many fluid-dynamics applications require solutions in complex geometries. In these cases, mesh generation can be a difficult and computationally expensive task for mesh-based methods. This is alleviated in meshless methods by relaxing the neighborhood relations between nodes. Meshless methods, however, often face issues computing numerically robust local operators, especially for the irregular node configurations required to effectively resolve complex geometries. Here we address this issue by using Discretization-Corrected Particle Strength Exchange (DC PSE) operator discretization in a strong-form Eulerian collocation meshless solver. We use the solver to compute steady-state solutions of incompressible, laminar flow problems in standard benchmarks and multiple complex-geometry problems in 2D with a velocity-correction method in the Eulerian framework. We verify that the solver produces stable and accurate results across all benchmark problems. We find that DC PSE operator discretization is more robust to varying node configurations than Moving Least Squares (MLS). In addition, we find that in more challenging complex geometries, the solver using MLS operator discretization fails to converge, whereas DC PSE operators provide robust solutions without node adjustment.

© 2016 The Author(s). Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license.

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

1. Introduction

Solving fluid dynamics equations in complex geometries is essential across several branches of science and engineering. Examples of applications that necessitate fluid-dynamics simulations in complex geometries include gas turbine combustors [46,49], turbulent flow past a landing gear [29], physics of plasma processing devices [36], swimming of fish-like organisms [26,66], and physics of active matter in biological morphogenesis [16,30,32,55]. In such applications, the efficacy of mesh-based numerical methods, such as Finite Difference Methods (FDM), Finite Volume Methods (FVM), and Finite Element Methods (FEM) [17,70], is limited by the quality of the mesh used to discretize the spatial domain. De-

spite developments, mesh generation in complex irregular geometries still presents a significant bottleneck for mesh-based methods and can be the most computationally demanding part of a simulation [3]. Furthermore, for irregular 3D geometries there is a lack of fully automated mesh generators with hexahedral elements, with most methods still requiring manual mesh correction [40]. Motivated by these issues, a large family of numerical point-collocation schemes, called *meshless methods* or *particle methods*, have been formulated [8,18,33,37,38,50,53,65]. Meshless methods can eliminate the need for a mesh by relaxing the requirement for explicit neighborhood relationships, or configurations, between nodes.

In meshless methods, the spatial domain is discretized by a set of nodes arbitrarily distributed without any interconnectivity. Resolution refinement, therefore becomes a relatively straightforward procedure of selectively adding nodes where more resolution is required in the domain. Since the introduction of Smoothed Particle Hydrodynamics (SPH) [28,45], meshless methods proliferated with developments such as the element-free Galerkin (EFG) method [9],

* Corresponding author.

E-mail addresses: george.bourantas@gmail.com (G.C. Bourantas), cheesema@mpi-cbg.de (B.L. Cheeseman), rjesh@pks.mpg.de (R. Ramaswamy), ivos@mpi-cbg.de (I.F. Sbalzarini).

¹ Joint first authors; contributed equally to this work.

diffuse element method (DEM) [52], partition of unity finite element method (PUFEM) [7], *hp*-clouds method [21], finite point method [54], local boundary integral equation (LBIE) method [69], meshless local Petrov-Galerkin (MLPG) method [4,5], reproducing kernel particle methods (RKPM) [42,43], and Particle Strength Exchange (PSE) [23]. Specifically, in the field of fluid dynamics, meshless methods have been developed based on the weak (Galerkin) [1,39,44,48,62,68], and strong (collocation) [10–13,47,51] formulations. Strong-form methods largely rely on Moving Least Squares (MLS) [35,63] and Radial Basis Functions (RBFs) [15] for the approximation/interpolation of the unknown field functions and their derivatives. These methods do not face the same consistency issues as earlier meshless methods, such as SPH [41]. However, despite their success in producing accurate numerical results across benchmark problems, strong-form methods still pose challenges regarding robustness of operator computation across different node configurations, especially when dealing with irregular geometries [31]. Therefore, there is still a need for research into finding meshless numerical schemes that combine the ease of node generation of meshless methods with robust operator computation in complex and irregular geometries.

Strong-form meshless methods, also known as meshless collocation methods, usually require the inversion of a matrix, often termed the *Moment Matrix* or *Coefficient Matrix*, in order to compute the kernel weights of the discrete operators. This matrix depends on the local spatial distribution of nodes in the operator support. The condition number of this matrix limits the accuracy of the inversion and may dominate the global error [31]. Therefore, strong-form collocation methods in the Eulerian framework have been found to fail to converge under irregular node distributions in complex geometries [14,31,40]. This is induced by the numerical errors in the kernel weights and is not related to von-Neumann-type stability. Jin et al. [31] proposed the use of positivity conditions for the discretized operator, which, when satisfied, ensure a low condition number of the coefficient matrix in the Finite Cloud Method (FCM). For RBF, Schaback [58] proved an uncertainty principle, stating that there is a trade-off between having a low condition number for the operator and the operator's order of accuracy. Using the MLS method, Bourantas et al. [14] found that the condition number of the moment matrix for a given node distribution anti-correlates with the stability of the numerical scheme, with complex geometries and irregular node configurations often leading to high condition numbers and hence non-convergent solutions.

The DC PSE method [59–61] was originally formulated to overcome consistency issues of weak-form PSE operators [19,20,23] on irregular node distributions. The method is formulated by establishing a set of discrete moment conditions over a local distribution of nodes that, when satisfied, insure the approximation of a given differential operators with a specified convergence order. In the strong form [56] this results in a formulation similar to other collocation methods, such as MLS or RPKM, with the required inversion of a matrix comparable to the moment matrix or coefficient matrix of other schemes. The DC PSE method is general, in that similar discrete moment conditions must be satisfied by all meshless schemes to ensure convergence. For the strong-form operators investigated here, the DC PSE formulation is similar to that of the independently developed Differential Reproducing Kernels (DRK) [67] for solid mechanics applications. The DC PSE method has previously been used in 2D and 3D Lagrangian particle methods for linear advection-diffusion problems [56,60,61]. In Lagrangian methods, irregular node distributions occur due to advection of the particles. Simulating flows in a purely Lagrangian framework is hampered by the all-against-all N -body interaction. This is avoided in hybrid particle-mesh methods and in Eulerian meshless formulations.

Here, we use DC PSE operator discretization in a standard Eulerian meshless collocation solver. We show that this improves the robustness of the method, leading to coefficient matrices with small condition numbers. We also present a novel compact formulation of strong-form DC PSE operators using the Vandermonde system. This is the first time that DC PSE operators are used to solve flow problems. We investigate the robustness of the DC PSE method in the Eulerian framework by computing steady-state incompressible flows using the velocity-correction method [22] in irregular 2D geometries. Accuracy and robustness are assessed in well-known benchmark problems where we compare with established results [2,24,25,27,34,64]. Without loss of generality, we focus on 2D problems since they are easier to visualize and analyze. The robustness of the local linear systems is independent of the dimensionality of the domain, as it is always a tensor of second order that only depends on the neighboring nodes. We focus on the robustness of DC PSE operators and the resulting numerical scheme in several irregular geometries. Specifically, we use examples of flow behind a cylinder, flow single and multiple stenosed arteries, and a fully irregular bifurcating flow problem. For each problem, we test both regular and irregular node distributions. To assess the robustness of the DC PSE operators we provide maximum condition numbers of the moment matrices for both DC PSE and MLS operators in comparison. Furthermore, we compare the results obtained in each test problem with those from the MLS method [14], as a meshless benchmark.

The rest of this paper is structured as follows: In Section 2, we revisit DC PSE for operator discretization and provide a compact and cogent formulation of collocation DC PSE. In Section 3, we review the governing flow equations in their velocity-vorticity formulation along with the velocity-correction method. In Section 4, we present the numerical benchmark examples, illustrating the accuracy and robustness of the proposed scheme in simple and complex geometries. Finally, in Section 5, we present our conclusions.

2. DC PSE

DC PSE [60] was originally formulated as a correction of the Particle Strength Exchange method (PSE) [20,20,23] on irregular particle distributions. PSE is used for the evaluation of spatial derivatives of any degree of a (sufficiently smooth) function discretized over scattered collocation points. The PSE operators can be derived in two steps: First by constructing an integral operator that satisfies continuous moment conditions, ensuring it approximates a spatial derivative to a defined order of accuracy in the continuous domain. Second, by discretizing the integral operator over the particle positions using the mid-point quadrature method. A drawback from this two-step procedure is the introduction of an *overlap condition* for the PSE operators [60]. The overlap condition requires that for the operator to be consistent, the inter-particle spacing h and the width ϵ of the operator kernel have to satisfy $h \leq a\epsilon^q$, $0 < a < 1$, $q > 1$, where q depends on the order of the operator. This results in a large number of particles being required for small kernel sizes ϵ . For the DC PSE method, the overlap condition can be relaxed by directly satisfying discrete moment conditions over the collocation points, requiring only that $h = a\epsilon$, independent of the order of the operator and for any $a > 0$.

The discrete moment conditions for DC PSE can be thought of as a direct analogue to the continuous moment conditions for PSE. The DC PSE operators therefore overcome the limitation in PSE that the error resulting from operator discretization (i.e., the quadrature error) dominates the overall order of accuracy of the operator as prescribed by the moment conditions.

Here we briefly review the DC PSE operators for strong-form formulations (i.e. neglecting collocation point volumes) in 2D and

how to construct them. For simplicity, we present the DC PSE formulation in 2D, with standard kernel functions and form. However, the formulation of the DC PSE operators in n dimensions with arbitrary kernel functions is straight forward and the reader is directed to [59–61] for the original formulations and in-depth analysis.

We begin by considering a differential operator, of arbitrary order, for a sufficiently smooth field $f(\mathbf{x}) = f(x, y)$ at point $\mathbf{x}_p = \{x_p, y_p\}$ on a particular collocation point set

$$\mathbf{D}^{m,n} f(\mathbf{x}_p) = \left. \frac{\partial^{m+n}}{\partial x^m \partial y^n} f(x, y) \right|_{x=x_p, y=y_p} \quad (1)$$

where m and n are integers that determine the order of the differential operator.

The DC PSE operator for the spatial derivative $\mathbf{D}^{m,n} f(\mathbf{x}_p)$ is:

$$\mathbf{Q}^{m,n} f(\mathbf{x}_p) = \frac{1}{\epsilon(\mathbf{x}_p)^{m+n}} \sum_{\mathbf{x}_q \in \mathcal{N}(\mathbf{x}_p)} (f(\mathbf{x}_q) \pm f(\mathbf{x}_p)) \eta \left(\frac{\mathbf{x}_p - \mathbf{x}_q}{\epsilon(\mathbf{x}_p)} \right), \quad (2)$$

where $\epsilon(\mathbf{x})$ is a spatially dependent scaling or resolution function, $\eta(\mathbf{x}, \epsilon(\mathbf{x}))$ a kernel function,² and $\mathcal{N}(\mathbf{x}_p)$ is the set of points in the support of the kernel function. The sign in Eq. (2) is positive for $m+n$ odd, and negative if even. The form of the operator in Eq. (2), including the sign change, is chosen to match with [60] for similarity to the original PSE operators [23]. However, the DC PSE formulation outlined here can also be applied in general to operators written in the form $\sum_{\mathbf{x}_q \in \mathcal{N}(\mathbf{x})} f(\mathbf{x}_q) \eta(\mathbf{x}_q - \mathbf{x}_p)$.

We want to construct the DC PSE operators so that as we decrease the average spacing between nodes, $h(\mathbf{x}_p) \rightarrow 0$, the operator converges to the spatial derivative $\mathbf{D}^{m,n} f(\mathbf{x}_p)$ with an asymptotic rate r :

$$\mathbf{Q}^{m,n} f(\mathbf{x}_p) = \mathbf{D}^{m,n} f(\mathbf{x}_p) + \mathcal{O}(h(\mathbf{x}_p)^r), \quad (3)$$

where it is convenient to explicitly define the component-wise average neighbor spacing as $h(\mathbf{x}_p) = \frac{1}{N} \sum_{\mathbf{x}_q \in \mathcal{N}(\mathbf{x}_p)} (|x_p - x_q| + |y_p - y_q|)$, where N is the number of nodes in the support of \mathbf{x}_p .

Therefore, we need to find a kernel function $\eta(\mathbf{x})$ and a scaling relation $\epsilon(\mathbf{x}_p)$ that satisfy Eq. (3). To achieve this, we replace the terms $f(\mathbf{x}_q)$ in Eq. (2) with their Taylor series expansions around the point \mathbf{x}_p . This substitution gives:

$$\begin{aligned} \mathbf{Q}^{m,n} f(\mathbf{x}_p) &= \frac{1}{\epsilon(\mathbf{x}_p)^{m+n}} \sum_{\mathbf{x}_q \in \mathcal{N}(\mathbf{x}_p)} \left(\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \frac{(x_p - x_q)^i (y_p - y_q)^j (-1)^{i+j}}{i! j!} \right. \\ &\quad \left. \times \mathbf{D}^{i,j} f(\mathbf{x}_p) \pm f(\mathbf{x}_p) \right) \eta \left(\frac{\mathbf{x}_p - \mathbf{x}_q}{\epsilon(\mathbf{x}_p)} \right). \end{aligned} \quad (4)$$

It is convenient to re-write Eq. (4) in the form:

$$\begin{aligned} \mathbf{Q}^{m,n} f(\mathbf{x}_p) &= \left(\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \frac{\epsilon(\mathbf{x}_p)^{i+j-m-n} (-1)^{i+j}}{i! j!} \mathbf{D}^{i,j} f(\mathbf{x}_p) \mathbf{Z}^{i,j}(\mathbf{x}_p) \right) \\ &\quad \pm \mathbf{Z}^{0,0}(\mathbf{x}_p) \epsilon(\mathbf{x}_p)^{-m-n} f(\mathbf{x}_p), \end{aligned} \quad (5)$$

where

$$\mathbf{Z}^{i,j}(\mathbf{x}_p) = \sum_{\mathbf{x}_q \in \mathcal{N}(\mathbf{x}_p)} \frac{(x_p - x_q)^i (y_p - y_q)^j}{\epsilon(\mathbf{x}_p)^{i+j}} \eta \left(\frac{\mathbf{x}_p - \mathbf{x}_q}{\epsilon(\mathbf{x}_p)} \right) \quad (6)$$

are the *discrete moments* of η . Now if we restrict the scaling parameter $\epsilon(\mathbf{x}_p)$ to converge at the same rate as the average spacing

between points $h(\mathbf{x}_p)$, that is

$$\frac{h(\mathbf{x}_p)}{\epsilon(\mathbf{x}_p)} \in \mathcal{O}(1), \quad (7)$$

then we find that the discrete moments $\mathbf{Z}^{i,j}$ are $\mathcal{O}(1)$ as $h(\mathbf{x}_p) \rightarrow 0$ and $\epsilon(\mathbf{x}_p) \rightarrow 0$. This is because the terms $\frac{(x_p - x_q)^i (y_p - y_q)^j (-1)^{i+j}}{\epsilon(\mathbf{x}_p)^{i+j}}$ are $\mathcal{O}(1)$ from the scaling relation and definition of $h(\mathbf{x}_p)$. Further, the second term $\eta \left(\frac{\mathbf{x}_p - \mathbf{x}_q}{\epsilon(\mathbf{x}_p)} \right)$ is $\mathcal{O}(1)$, through normalization of the function argument. These qualities are the motivation for the form of the normalized kernel function. Therefore, the scaling behavior of Eq. (5) is determined solely by the $\epsilon(\mathbf{x}_p)^{i+j-m-n}$ term with the smallest power and non-zero coefficient. Note that Eq. (7) is a much looser constraint on the average spacing of nodes when compared to the overlap condition $h \leq a\epsilon^q$, $0 < a < 1$, $q > 1$ for the PSE method [60].

Given Eq. (7), the convergence rate r of the DC PSE operator $\mathbf{Q}^{m,n}$ (Eqs. 3 and 5) is determined by the coefficients of the terms $\epsilon(\mathbf{x}_p)^{i+j-m-n}$ in Eq. (5). The coefficient of $\epsilon(\mathbf{x}_p)^{i+j-m-n}$ in Eq. (3) is required to be 1 when $i = m$ and $j = n$, and 0 when $i + j - m - n < r$. This results in the following set of conditions for the discrete moments,

$$\mathbf{Z}^{i,j}(\mathbf{x}_p) = \begin{cases} i! j! (-1)^{i+j} & i = m, j = n \\ 0 & \alpha_{\min} \leq i + j < r + m + n \\ < \infty & \text{otherwise} \end{cases} \quad (8)$$

where α_{\min} is 1 if $m+n$ is even and 0 if odd. This is due to the zeroth moment $\mathbf{Z}^{0,0}$ canceling out for odd $i+j$. The choice of the factor $\epsilon(\mathbf{x}_p)^{-m-n}$ in Eq. (2) acts as to simplify the expression of the moment conditions.

For the kernel function $\eta(\mathbf{x})$ to be able to satisfy the l conditions given in Eq. (8) for arbitrary neighborhood node distributions, the operator must have l degrees of freedom. This leads to the requirement that the support $\mathcal{N}(\mathbf{x})$ of the kernel function has to include at least l nodes. In this paper, as in Schrader et al. and Reboux et al. [56,60], we use kernel functions of the form

$$\eta(\mathbf{x}) = \begin{cases} \sum_{i,j}^{i+j < r+m+n} a_{i,j} x^i y^j e^{-x^2 - y^2} & \sqrt{x^2 + y^2} < r_c \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

This is a monomial basis multiplied by an exponential window function, where r_c sets the kernel support and the $a_{i,j}$ are scalars to be determined to satisfy the moment conditions in Eq. (8). The cut-off radius r_c should be set to include at least l collocation nodes in the support $\mathcal{N}(\mathbf{x})$. In this paper, r_c is set implicitly by using the $l-1$ nearest neighbors of each node. Alternatively, adaptive methods can be used [56].

We direct the reader to [61] for the exploration of more general kernel choices and the impact of the various parameters on the computational cost. If $\alpha_{\min} = 1$, the $a_{0,0}$ coefficient is a free parameter and can be used to increase the numerical robustness of solving the linear system of equations for the remaining $a_{i,j}$ [59].

To construct the operator $\mathbf{Q}^{m,n} f(\mathbf{x}_p)$ at node \mathbf{x}_p , the coefficients are found by solving a linear system of equations from Eqs. (9) and (8). We outline now a convenient novel way of formulating the operator and the linear system ([59]). With our choice of kernel function we have,

$$\begin{aligned} \mathbf{Q}^{m,n} f(\mathbf{x}_p) &= \frac{1}{\epsilon(\mathbf{x}_p)^{m+n}} \sum_{\mathbf{x}_q \in \mathcal{N}(\mathbf{x}_p)} (f(\mathbf{x}_q) \pm f(\mathbf{x}_p)) \mathbf{p} \left(\frac{\mathbf{x}_p - \mathbf{x}_q}{\epsilon(\mathbf{x}_p)} \right) \\ &\quad \times \mathbf{a}^T(\mathbf{x}_p) e^{-\frac{(x_p - x_q)^2 + (y_p - y_q)^2}{\epsilon(\mathbf{x}_p)^2}}, \end{aligned} \quad (10)$$

² The original weak-form formulation includes a particle volume v_p and a dimension-dependent normalization factor for the particle volume $\epsilon(\mathbf{x}_p)^{-d}$, where d is the number of spatial dimensions, providing a normalization of the integration length, area, or volume for the particle. As we consider strong-form formulations both are omitted here.

where $\mathbf{p}(\mathbf{x}) = \{p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_l(\mathbf{x})\}$ and $\mathbf{a}(\mathbf{x})$ are vectors of the terms in the monomial basis and of their coefficients in Eq. (9), respectively.

Using this formulation, the operator system becomes straightforward. For example, if we set $r = 2$ and approximate the first spatial derivative in the x direction, $D^{1,0}$, we have $l = 6$ moment conditions and the monomial basis is $\mathbf{p}(x, y) = \{1, x, y, yx, x^2, y^2\}$. The linear system for the kernel coefficients then is:

$$\mathbf{A}(\mathbf{x}_p) \mathbf{a}^T(\mathbf{x}_p) = \mathbf{b}, \quad (11)$$

where

$$\mathbf{A}(\mathbf{x}_p) = \mathbf{B}(\mathbf{x}_p)^T \mathbf{B}(\mathbf{x}_p) \in \mathbb{R}^{l \times l} \quad (12)$$

$$\mathbf{B}(\mathbf{x}_p) = \mathbf{E}(\mathbf{x}_p)^T \mathbf{V}(\mathbf{x}_p) \in \mathbb{R}^{k \times l} \quad (13)$$

$$\mathbf{b} = (-1)^{m+n} \mathbf{D}^{m,n} \mathbf{p}(\mathbf{x})|_{x=0} \in \mathbb{R}^{l \times 1}. \quad (14)$$

The scalar number $k \geq l$ is the number of nodes in the support of the operator, l the number of moment conditions to be satisfied, and $\mathbf{V}(\mathbf{x}_p)$ the Vandermonde matrix constructed from the monomial basis $\mathbf{p}(\mathbf{x}_p)$. $\mathbf{E}(\mathbf{x}_p)$ is a diagonal matrix containing the square roots of the values of the exponential window function at the neighboring nodes in the operator support. Further, for node \mathbf{x}_p we define $\{\mathbf{z}_q(\mathbf{x}_p)\}_{q=1}^k = \{\mathbf{x}_p - \mathbf{x}_q\}_{\mathbf{x}_q \in \mathcal{N}(\mathbf{x}_p)}$, the set of vectors pointing to \mathbf{x}_p from all neighboring nodes \mathbf{x}_q in the support of \mathbf{x}_p . So then explicitly

$$\mathbf{V}(\mathbf{x}_p) = \begin{pmatrix} p_1\left(\frac{z_1(\mathbf{x}_p)}{\epsilon(\mathbf{x}_p)}\right) & p_2\left(\frac{z_1(\mathbf{x}_p)}{\epsilon(\mathbf{x}_p)}\right) & \cdots & p_l\left(\frac{z_1(\mathbf{x}_p)}{\epsilon(\mathbf{x}_p)}\right) \\ p_1\left(\frac{z_2(\mathbf{x}_p)}{\epsilon(\mathbf{x}_p)}\right) & p_2\left(\frac{z_2(\mathbf{x}_p)}{\epsilon(\mathbf{x}_p)}\right) & \cdots & p_l\left(\frac{z_2(\mathbf{x}_p)}{\epsilon(\mathbf{x}_p)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ p_1\left(\frac{z_k(\mathbf{x}_p)}{\epsilon(\mathbf{x}_p)}\right) & p_2\left(\frac{z_k(\mathbf{x}_p)}{\epsilon(\mathbf{x}_p)}\right) & \cdots & p_l\left(\frac{z_k(\mathbf{x}_p)}{\epsilon(\mathbf{x}_p)}\right) \end{pmatrix} \in \mathbb{R}^{k \times l} \quad (15)$$

$$\mathbf{E}(\mathbf{x}_p) = \text{diag} \left(\left\{ e^{-\frac{|\mathbf{z}_q(\mathbf{x}_p)|^2}{2\epsilon(\mathbf{x}_p)^2}} \right\}_{q=1}^k \right) \in \mathbb{R}^{k \times k}. \quad (16)$$

Once the matrix $\mathbf{A}(\mathbf{x}_p)$ is constructed at each node \mathbf{x}_p , the linear systems can be solved for the coefficients $\mathbf{a}(\mathbf{x}_p)$ used in the DC PSE operators at each node as in Eq. (10). The matrix $\mathbf{A}(\mathbf{x}_p)$ only depends on the number of moment conditions l and the local distribution of nodes in $\mathcal{N}(\mathbf{x}_p)$. Therefore, if the system in Eq. (11) is solved using a decomposition (such as LU), of $\mathbf{A}(\mathbf{x}_p)$. This form can be re-used for multiple right-hand sides, i.e., for different differential operators (albeit with different convergence rates r). The matrix \mathbf{A} has an analogue in MLS where it is often called *moment matrix*. This contains information about the spatial distribution of the collocation nodes around the center point \mathbf{x}_p . The invertibility of \mathbf{A} depends entirely on that of the Vandermonde matrix \mathbf{V} , due to \mathbf{E} being a diagonal matrix with non-zero entries. The condition number of \mathbf{A} depends on both \mathbf{V} and \mathbf{E} and determines the robustness of the numerical inversion.

3. Governing equations at steady state

The governing equations express conservation of mass and linear momentum. In their velocity-pressure formulation ($u - v - p$) at steady state, they are:

Continuity equation

$$\nabla \cdot \mathbf{u} = 0, \quad (17)$$

Momentum equation

$$\rho((\mathbf{u} \cdot \nabla) \mathbf{u}) = -\nabla p + \mu \nabla^2 \mathbf{u}, \quad (18)$$

where \mathbf{u} is the fluid velocity, p is the fluid pressure, ρ the density, and μ the dynamic viscosity.

Since for Eulerian primitive-variables formulations pressure boundary conditions are hard to define, the governing equations can be rewritten in velocity-vorticity formulation, which can be extended to 3D studies, by taking the curl of the momentum equation and accounting for the continuity equation. A vector Poisson equation relates the velocity and vorticity fields. Eqs. (17) and (18) then become:

$$\nabla^2 \mathbf{u} = -\nabla \times \boldsymbol{\omega}, \quad (19)$$

$$\mathbf{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \mathbf{u} + \frac{1}{\text{Re}} \nabla^2 \boldsymbol{\omega}, \quad (20)$$

where $\boldsymbol{\omega}$ is the vorticity. We seek a solution of Eqs. (19) and (20) in the spatial domain Ω that satisfies the boundary conditions

$$\mathbf{u} = \mathbf{u}_{\partial\Omega},$$

$$\boldsymbol{\omega} = (\nabla \times \mathbf{u})_{\partial\Omega}. \quad (21)$$

Since we consider the steady-state case in two-dimensional domains, the governing equations in a Cartesian coordinate system simplify to:

$$\nabla^2 u = -\frac{\partial \omega}{\partial y}, \quad (22)$$

$$\nabla^2 v = \frac{\partial \omega}{\partial x}, \quad (23)$$

$$u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \frac{1}{\text{Re}} \nabla^2 \omega, \quad (24)$$

where $\mathbf{u} = (u, v)$ and $\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$ is a scalar.

3.1. Velocity-correction method

After discretizing the governing Eqs. (17)–(24) using DC PSE, the differential operators are represented by sparse matrices of dimension $N \times N$, with N being the total number of nodes. An iterative velocity correction scheme [22], previously also used for MLS in [12], is used for numerically solving these linear equations. Initially, the Poisson equations for the velocity components (Eqs. (22)–(23)) are solved and an intermediate velocity \mathbf{u}^* is computed that, in general, does not satisfy the continuity equation, that is $\nabla \cdot \mathbf{u}^* \neq 0$. Satisfaction of the continuity equation is then accomplished by updating the velocity field with a velocity correction $\partial \mathbf{u}^{k+1}$:

$$\mathbf{u}^{k+1} = \mathbf{u}^* + \partial \mathbf{u}^{k+1}. \quad (25)$$

Without loss of generality, we assume that the velocity correction is irrotational (but not the flow itself), i.e. $\nabla \cdot \partial \mathbf{u}^{k+1} \neq 0$, then a Helmholtz potential (or correction potential) ψ^{k+1} can be defined as: $\nabla \psi^{k+1} = \partial \mathbf{u}^{k+1}$. Since the new velocity update is required to satisfy continuity, i.e. $\nabla \cdot \partial \mathbf{u}^{k+1} = 0$, the correction potential ψ^{k+1} must satisfy the Poisson-type equation

$$\nabla^2 \psi^{k+1} = -\nabla \cdot \mathbf{u}^*. \quad (26)$$

Once this Helmholtz-Poisson problem is solved, the velocity field is updated in order to satisfy continuity. The momentum equation is still satisfied because the correction is irrotational. From the updated velocity field \mathbf{u}^{k+1} we calculate the vorticity. The vorticity transport Eq. (20) is then solved using the updated vorticity boundary values. Then, the new vorticity values ω^{k+1} are used for the next iteration. The entire algorithm hence is:

- Use an initial guess for the velocity components $u^{(0)}$ and $v^{(0)}$.
- Calculate the initial vorticity field $\omega^{(0)}$ using the formula $\omega^{(0)} = \nabla \times \mathbf{u}^{(0)}$.

- Calculate the vorticity gradients (right-hand sides of Eqs. (22)–(23)) and solve the Poisson equations for the velocity components, yielding the intermediate velocity field u^* and v^* .
- Calculate the Helmholtz potential ψ using the appropriate boundary conditions and calculate the updated velocity fields.
- Solve the vorticity transport equation (Eq. (24)).
- Check for error convergence and iterate.

This is a standard method that has been used before. However, the convergence of the iterative scheme to the steady state solution depends on how the differential operators are discretized.

4. Numerical experiments

We explore the robustness of the solution when using DC PSE and MLS. First, we assess our method in terms of accuracy and validity by using two well-established benchmark problems: the lid-driven cavity flow, and the backward-facing step. Second, we explore the stability of the numerical scheme in complex geometries in four flow problems: flow behind a cylinder, single and multiple stenosed arteries, and a bifurcating artery.

The lid-driven cavity flow and the backward-facing step are standard test problems with extensive reference data available in the literature. They are frequently used to assess numerical methods for flow problems [2,24,25,27,34,64]. We present both qualitative, through figures, and quantitative solution values, through tables, in order to assess the method's accuracy and as a reference for comparison with previous works. In all cases we check the validity of the computed solutions by comparing with benchmark data from established methods. Furthermore, we perform grid-convergence studies for all problems to obtain grid-independent solutions with respect to these benchmark solutions.

A key issue in the application of meshless methods, especially for strong-form collocation methods, is their lack of numerical robustness in complex geometries [14,31,33,40,50,60,65]. Lack of convergence of the iterative solver, has been associated with node distributions with high condition numbers of the moment matrices [14,31,40], or with certain asymmetries [60]. Here, we use four flow problems in complex geometries to test our method's numerical stability and assess its robustness against node rearrangements. We validate the accuracy of the solutions by comparison with the established solvers from COMSOL and the same meshless scheme [14] using MLS, instead of DC PSE to discretize the differential operators. We use the MLS meshless scheme as a stability benchmark for the different geometries. For every problem we provide the maximum condition number of the moment matrix for the MLS operators and its equivalent for DC PSE, the maximum absolute difference of the solution, and whether either scheme had issues regarding convergence. Although the comparison of the absolute value of the condition numbers across the MLS and DC PSE methods is not meaningful, due to different prefactors, the relative variation of the condition numbers for each method by itself across node distributions does provide an indication of the sensitivity of the operator to node rearrangement. The robustness of the overall solution is also reflected in the condition number of the global linear system for the Poisson-type equations. While the condition number of the global linear system depends on the condition numbers of the local (i.e., moment matrix) systems, it also depends on many other factors that are problem dependent. Therefore, we restrict our discussion of robustness to the local linear systems. Further, we do not provide a computational cost study for the method here, as the numerical procedure of computing the DC PSE operators is identical to that when using other strong-form collocation methods, such as MLS or FCM. The computational cost is independent of the choice of operator discretization and simply reflects the

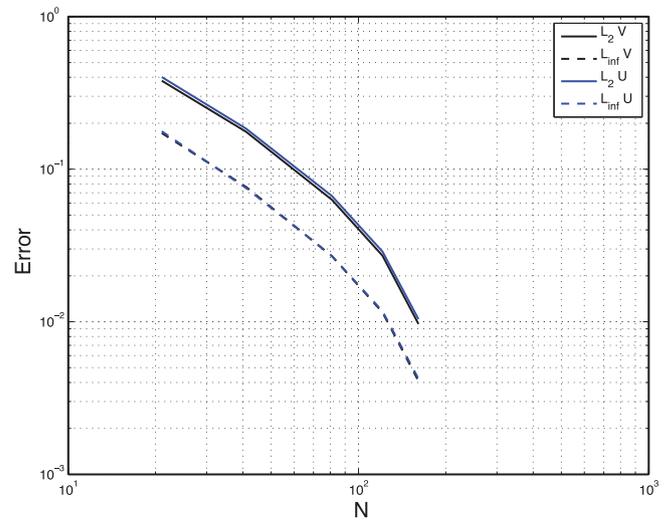


Fig. 1. Lid-driven cavity flow: Convergence for $Re = 400$ for increasing resolution: 21×21 , 41×41 , 81×81 , 121×121 , and 161×161 . The absolute L_2 and L_∞ error norms over all nodes are calculated with respect to a solution computed with 201×201 nodes, as in [27]. $N \times N$ is the total number of nodes used in the simulation, and N the number of nodes in each direction.

cost of the Eulerian solver [12]. Here we are interested in whether solutions can be obtained *at all* and how robust they are across different node configurations.

4.1. Lid-driven cavity flow

We first test our method on the lid-driven flow in a square cavity, for the domain $x \in [0, 1]$ and $y \in [0, 1]$. Lid-driven flow is a computationally demanding problem due to the multiple recirculation zones at the corners, especially for moderate and high Reynolds numbers. The problem has been intensively studied over the years, and numerical studies for $Re \leq 1000$ produce consistent results across methods. Here we present numerical solutions from our method for Reynolds numbers up to 10,000, for which the flow regime is considered to be laminar and the solution stable.

For the flow boundary conditions, we apply no slip at all cavity walls, except the top, which moves with unit velocity to drive the flow. For the DC PSE operators, $r = 2$ for first-order derivatives, and $r = 1$ for second-order derivatives. The support cutoff radius r_c of the kernel is set to $2.5h$ for interior nodes and $3.5h$ for boundary nodes, where h ($h = \frac{1}{260}$) is the inter-node spacing. To assess the convergence of our method in a mesh-refinement sense, we compute the solution with progressively higher resolution at constant $Re = 400$, reporting norms relative to the highest resolution used (201×201). We use regular Cartesian node distributions for this test. The resulting convergence curve is shown in Fig. 1, demonstrating that the scheme converges at a rate between 1.5 and 2.5.

For $Re = 1000$ and $Re = 2000$ we use a uniform node distribution of 261×261 nodes with a total of 68,121 nodes. For $Re = 10,000$ we use a uniform grid of 361×361 nodes. We assess the numerical solution of our method through both quantitative and qualitative comparisons with benchmark numerical results from the literature [24]. For qualitative evaluation of the solution, stream-lines (computed as in [24]) of the flow field are presented for $Re = 1000$ in Fig. 2, for $Re = 2500$ in Fig. 3, and for $Re = 10,000$ in Fig. 4. These figures show the formation of the counter-rotating secondary vortices that appear as the Reynolds number increases. For quantitative comparison, we present velocity values along the vertical center line for u in Table 1, and for v in Table 2, for both $Re = 1000$ and $Re = 2500$ alongside the numerical results from [24].

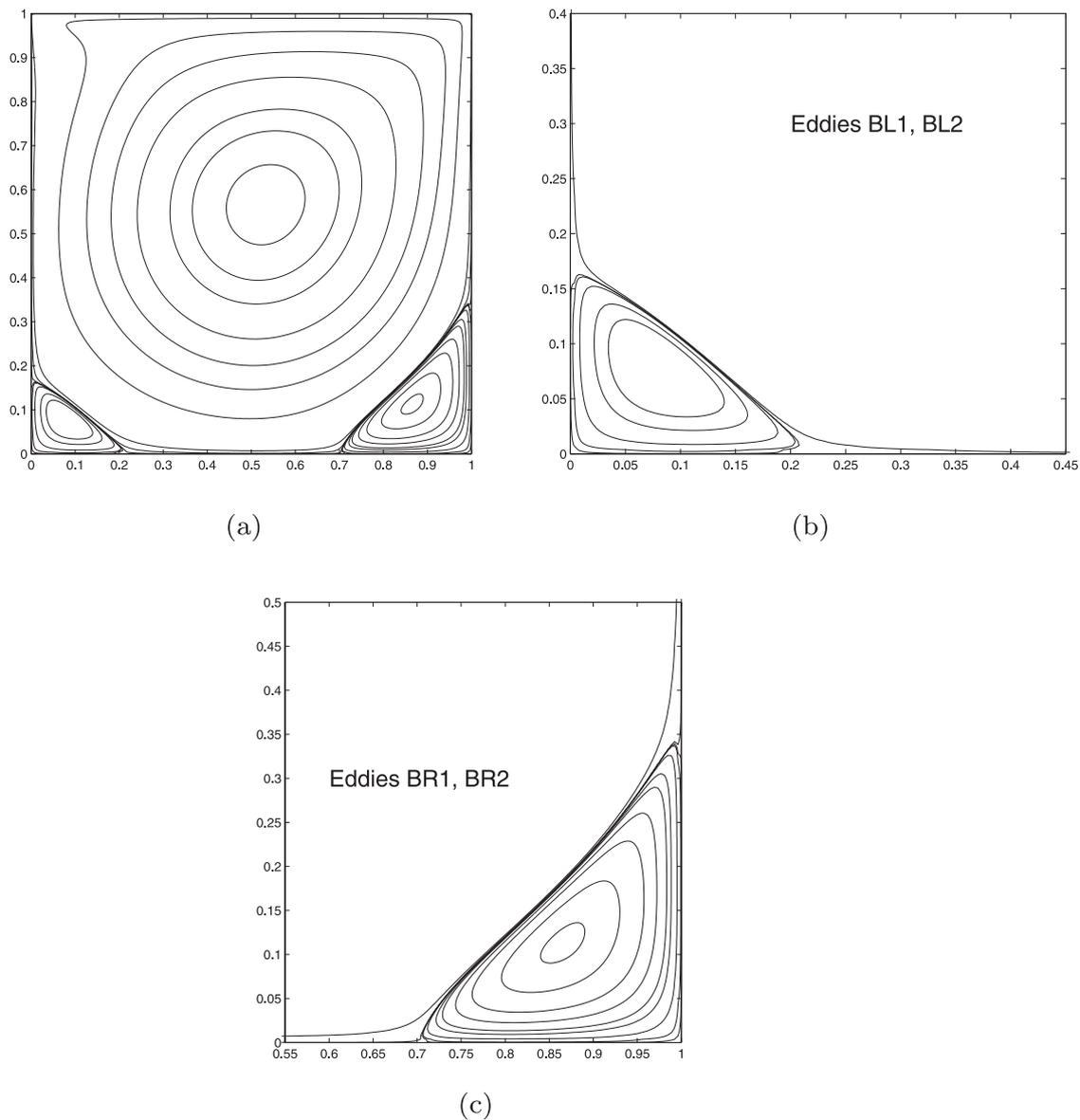


Fig. 2. Lid-driven cavity flow: Streamline patterns of primary and secondary vortices (computed as in [24]) obtained for $Re = 1000$. (a) The full domain; (b) close-up on the eddies BL1 and BL2; (c) close-up on BR1 and BR2. BL1, BL2, BR1 and BR2 are defined as in [24].

Assessing the numerical robustness of the scheme relative to MLS, we find the maximum condition number of the moment matrix for MLS and DC PSE operators on the same 261×261 node distribution to be 167 for DC PSE and 2.74×10^{13} for MLS. When comparing the solutions using MLS and DC PSE as operator discretization, the maximum absolute differences for the u and v velocity components are 5.54×10^{-2} and 5.06×10^{-2} for $Re = 1000$, and 5.96×10^{-2} and 5.34×10^{-2} for $Re = 2500$, respectively. Therefore, we find that the DC PSE scheme reproduces previous results for the lid-driven cavity flow benchmark problem in this simple geometry.

4.2. Backward-facing step

As a second benchmark problem, we consider the backward-facing step (BFS) [2]. The backward-facing step involves a channel of width H ($H = 1$) and length L ($L = 30H$), with a backward-facing step (of height $H/2$) placed at the left-most edge of the inlet ($x = 0$) where flow is assumed to be fully developed. This flow is set to have a parabolic inflow velocity profile given by

$\mathbf{u} = (12y - 24y^2, 0)$ for $y > 0.5$. Fully developed flow is also assumed at the outlet (right edge), with the velocity profile given by $\mathbf{u} = (0.75 - 3y^2, 0)$ for $0 < y < 1$.

The BFS is considered a demanding benchmark flow problem due to the vortices formed after the step. The BFS has been studied both experimentally [2] and numerically [2,25,34,64]. The flow has been found to be stable and two-dimensional for $Re \lesssim 400$, allowing the flow to be numerically modeled in 2D and compared directly with experiments [2]. Beyond this Reynolds number, the flow is 3D and the 2D approximation is no longer valid. However, numerical results for the 2D problem for $Re > 400$ are still given in the literature as a purely numerical benchmark problem [25].

Here, we present numerical results for our method for $Re=200$, for comparison with both experimental and numerical results such as [2], and $Re = 800$ for comparison with the purely numerical benchmarks given in [25,34,64]. For the DC PSE operators, $r = 2$ for first-order derivatives, and $r = 1$ for second-order derivatives. The cut-off radius r_c of the kernel function is set to $2.5h$ for interior nodes and $3.5h$ for boundary nodes, with h being the

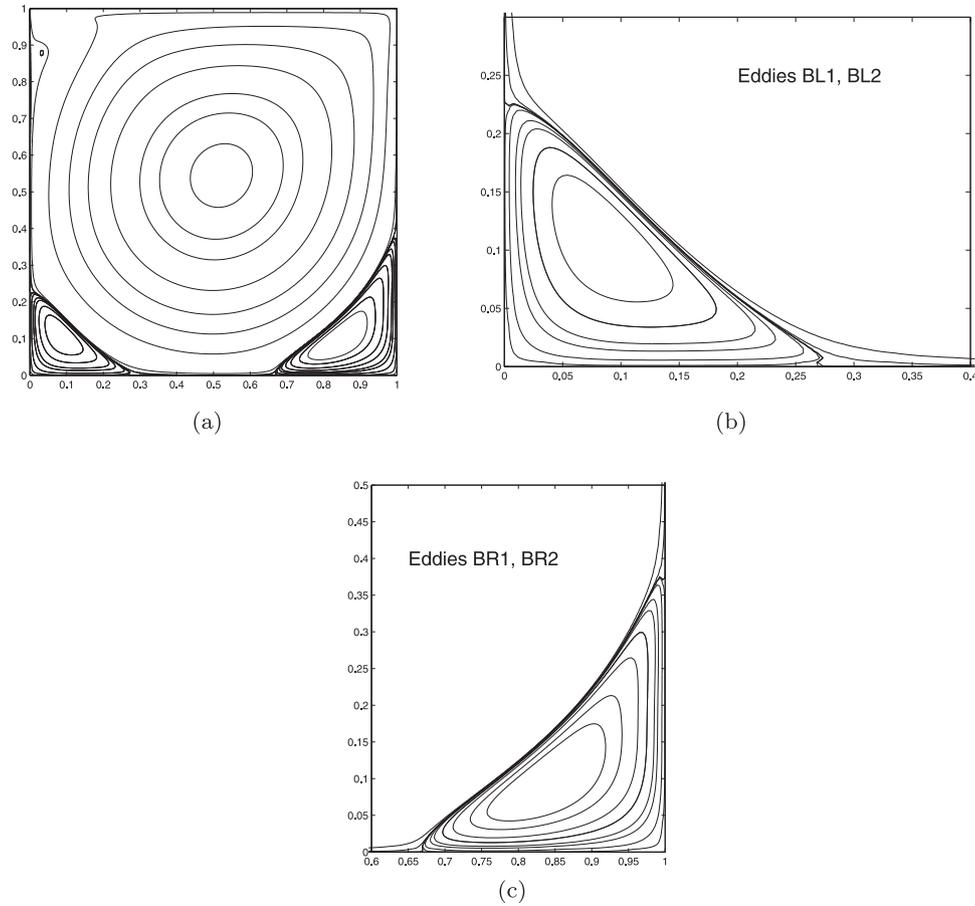


Fig. 3. Lid-driven cavity flow: Streamline patterns of primary and secondary vortices (computed as in [24]) obtained for $Re = 2500$. (a) The full domain; (b) close-up on the eddies BL1 and BL2; (c) close-up on BR1 and BR2. BL1, BL2, BR1 and BR2 are defined as in [24].

Table 1
Lid-driven cavity flow: u velocity component along the vertical line passing through the center of the cavity at $Re = 1000$ and $Re = 2500$ compared with [24].

y	Re = 1000		Re = 2500	
	DC PSE	Ref. [24]	DC PSE	Ref. [24]
1	1	1	1	1
0.99	0.8436	0.8486	0.7585	0.7704
0.98	0.6976	0.7065	0.5712	0.5924
0.97	0.5799	0.5917	0.4688	0.4971
0.96	0.4958	0.5102	0.4283	0.4607
0.95	0.4418	0.4582	0.417	0.4506
0.94	0.4098	0.4276	0.414	0.447
0.93	0.3916	0.4101	0.4107	0.4424
0.92	0.3807	0.3993	0.4052	0.4353
0.91	0.373	0.3913	0.3973	0.4256
0.9	0.3661	0.3838	0.3874	0.4141
0.5	-0.0591	-0.062	-0.0369	-0.0403
0.2	-0.3612	-0.3756	-0.3036	-0.3228
0.18	-0.3701	-0.3869	-0.3244	-0.3439
0.16	-0.3666	-0.3854	-0.3487	-0.3688
0.14	-0.349	-0.369	-0.3747	-0.3965
0.12	-0.3183	-0.3381	-0.3944	-0.42
0.1	-0.2778	-0.296	-0.3948	-0.425
0.08	-0.2316	-0.2472	-0.3652	-0.3979
0.06	-0.1825	-0.1951	-0.3069	-0.3372
0.04	-0.1299	-0.1392	-0.2307	-0.2547
0.02	-0.0702	-0.0757	-0.1362	-0.1517
0	0	0	0	0

inter-node spacing; $h = 0.05$ and $h = 0.025$ for $Re = 200$ and $Re = 800$, respectively. The kernel width ϵ is set to be equal to h . To ensure a resolution-independent numerical solution, several node configurations are tested, specifically 301×11 , 601×21 , 901×31 , and 1201×41 . This shows convergence in a mesh-refinement sense of the solution to the required precision. For $Re = 200$ and $Re = 800$, regular grid distributions of 12,621 (601×21) and 49,241 (1201×41) nodes are used, respectively.

Fig. 5 shows the streamlines and vorticity contours for $Re = 200$. The flow separates at the step corner and a vortex is formed downstream. For $Re = 200$ the reattachment length of the vortex is $L = 2.55$. Table 3 lists the reattachment length values obtained using Radial Basis Functions and the Finite Element Method from [12] along with those from our DC PSE method. The DC PSE reattachment lengths are consistent with those of established methods. Fig. 6 shows the streamlines and vorticity contours for $Re=800$. After reattachment of the upper wall eddy, the flow slowly recovers towards a fully developed Poiseuille flow. For our method, the measured separation and reattachment points at $Re = 800$ are $L_{lower} \approx 6.1$ for the lower wall separation zone, $L_{upper} \approx 5.11$ for the upper separation zone, and separation begins at $x \approx 5.19$. Comparison of the present scheme with other numerical methods for 2D computations shows good agreement, especially with respect to the lower wall separation zone. In [64] the authors used a finite difference method and predicted separation lengths of $L_{lower} \approx 6.0$ and $L_{upper} \approx 5.75$, while [64] using the FIDAP code predicted $L_{lower} \approx 5.8$ and the upper $L_{upper} \approx 4.7$. In order to validate our method, cross-channel profiles of a variety of quantities, and the equivalent data form [25], are provided at

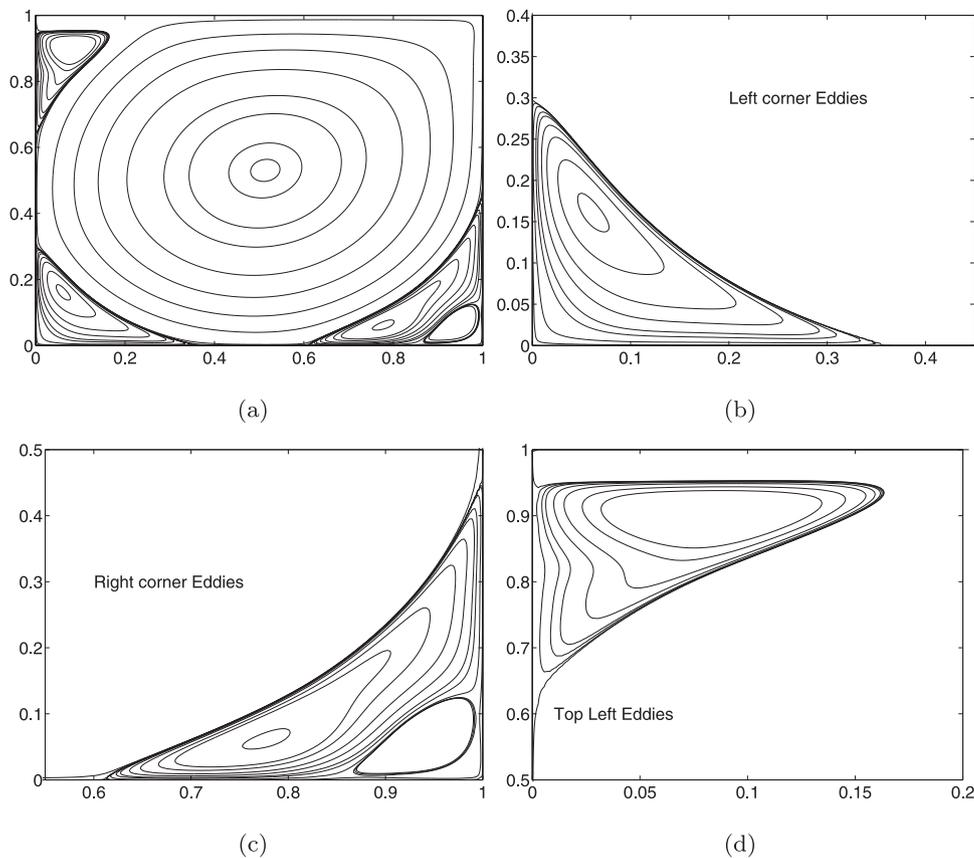


Fig. 4. Lid-driven cavity flow: Streamline patterns of primary and secondary vortices (computed as in [24]) obtained for $Re = 10,000$. (a) The full domain; (b) close-up on the eddies in bottom left corner; (c) close-up on eddies in bottom right corner, and (d) close up on eddies in top left.

$x = 7$ in Table 4, and at $x = 15$ in Table 5. These cross-channel profiles and solution comparisons for $Re = 800$ are shown in Fig. 7, providing an example of the convergence studies performed for all test cases.

Assessing the relative numerical robustness of the DC PSE scheme, we find the maximum condition number for DC PSE to be 167 and for MLS 1.01×10^9 for the same node distributions. Further, when the solution is compared to that with MLS operators, the maximum absolute differences for the u and v velocity components are 3.11×10^{-2} and 6.65×10^{-2} for $Re = 800$, and 2.1×10^{-2} and 9.1×10^{-2} for $Re = 200$, respectively. In summary, the numerical results obtained here show good agreement with the established benchmarks for the backward-facing step flow problem.

4.3. Flow past a cylinder

We consider flow in a 2D rectangular duct with a circular occlusion. This problem showcases a regular geometry with an internal obstacle. The spatial domain is $\Omega = [0, 0.1] \times [0, 0.01]$ (m) with a cylinder of radius $r = 0.0015$ (m) located at $B_r = (0.02, 0.005)$ (m). We set the kinematic viscosity $\nu = \frac{\mu}{\rho} = \frac{0.001 \text{ m}^2}{997 \text{ s}}$. For the flow boundary conditions we set no-slip conditions, $\mathbf{u} = (u, v) = (0, 0)$, for the upper and lower walls of the duct and also along the perimeter of the cylinder. For the inlet (left edge), a uniform velocity of $\mathbf{u}(0, y) = (0.01, 0) \frac{\text{m}}{\text{s}}$ is enforced, while at the outlet (right edge), a do-nothing outflow boundary condition is applied: $\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = 0$, where \mathbf{n} denotes the outward unit normal.

We present results for $Re = 30$. A total number of 25,405 nodes is used, with 45 nodes distributed on the cylinder perimeter to enforce the boundary condition there. For the DC PSE operators, $r = 2$ for first-order derivatives, and $r = 1$ for second-order derivatives. The cut-off radius r_c of the kernel function is $2.5h$ for interior nodes and $3.5h$ for boundary nodes, where h ($h = 0.02$) is the inter-node spacing.

Contour plots for the velocity components and streamlines are shown in Fig. 8. Two symmetric vortices with opposite rotation directions are formed behind the cylinder. These vortices have a recirculation length of $L = 0.00144$ m from the center of the cylinder. For solution validation, the same problem is solved using COMSOL with a total number of 72,466 degrees of freedom. The COMSOL solution is interpolated to the DC PSE node locations for direct comparison. The COMSOL solution shows similar flow profiles with a recirculation length of $L = 0.00142$ m. The maximum absolute differences in the u and v velocity components of the COMSOL and DC PSE solutions are 1.67×10^{-2} and 9.31×10^{-3} , respectively.

Checking the relative numerical robustness of the solution, the maximum condition numbers are 179 for DC PSE operators and 2.496×10^{15} for MLS. When computing the solution with MLS, the node arrangement needed to be manually refined with nodes close to degeneracy having to be redistributed for the method to converge. This manual refinement was not required when using the DC PSE operators. The maximum absolute difference between the DC PSE and MLS solutions are 4.88×10^{-4} and 5.47×10^{-4} for the u and v velocity components, respectively. In summary, we have shown that DC PSE produces consistent solutions for flow past a cylinder, and shows improved robustness with regard to node distribution than when using MLS operators.

Table 2

Lid-driven cavity flow: v velocity component along the vertical line passing through the center of the cavity at $Re = 1000$ and $Re = 2500$ compared with [24].

y	Re = 1000		Re = 2500	
	DC PSE	Ref. [24]	DC PSE	Ref. [24]
1	0	0	0	0
0.985	-0.0916	-0.0973	-0.151	-0.1675
0.97	-0.2046	-0.2173	-0.3382	-0.3725
0.955	-0.3208	-0.34	-0.4786	-0.5192
0.94	-0.4183	-0.4417	-0.5254	-0.5603
0.925	-0.4811	-0.5052	-0.4995	-0.5268
0.91	-0.5041	-0.5263	-0.4499	-0.4741
0.895	-0.4942	-0.5132	-0.4078	-0.4321
0.88	-0.464	-0.4803	-0.3794	-0.4042
0.865	-0.4262	-0.4407	-0.3596	-0.3843
0.85	-0.3891	-0.4028	-0.3432	-0.3671
0.5	0.0264	0.0258	0.0165	0.016
0.15	0.3558	0.3756	0.368	0.3918
0.135	0.3498	0.3705	0.3814	0.4078
0.12	0.3394	0.3605	0.3894	0.4187
0.105	0.3249	0.346	0.3895	0.4217
0.09	0.3066	0.3273	0.38	0.4142
0.075	0.2842	0.3041	0.3602	0.395
0.06	0.2558	0.2746	0.3311	0.3649
0.045	0.2178	0.2349	0.2923	0.3238
0.03	0.1652	0.1792	0.2355	0.2633
0.015	0.0933	0.1019	0.1418	0.1607
0	0	0	0	0

4.4. Flow in stenosed arteries

As a first test case with complex geometry, we simulate fluid flow in a stenosed, or contracted, artery. We present results for two cases: a duct with an asymmetric stenosis, and a more general case with several irregular stenoses. In both problems, the length and width of the unconstrained channel is $L = 10$ and $H = 1$, respectively. At the inlet, the flow is assumed to be fully developed with the velocity having a parabolic profile. On the lower and upper walls, no-slip boundary conditions are imposed. Hence:

$$\begin{aligned}
 \text{Inflow:} & \quad (x = 0, 0 \leq y \leq 1) \\
 & \quad (u = 4y - 4y^2, v = 0) \\
 \text{Outflow:} & \quad (x = 10, 0 \leq y \leq 1) \\
 & \quad \left(\frac{\partial u}{\partial x} = 0, v = 0 \right) \\
 \text{Walls:} & \quad (y = y(x), 0 \leq x \leq 10) \\
 & \quad (u = 0, v = 0).
 \end{aligned} \tag{27}$$

These flow problems showcase the performance of our method on increasingly irregular geometries and node distributions. Therefore, we test both regular and irregular node distributions. The irregu-

lar distributions are generated by extracting the vertex positions, and ignoring the mesh neighbor properties, produced by the COMSOL mesh generator. While this is not how one would place nodes in practice, it serves as a reproducible benchmark for comparison between the MLS and DC PSE operators.

4.4.1. Single stenosis

The computational domain has a single stenosis of asymmetric shape:

$$y_{\text{lower}} = A_1 \text{sech}(B_1(x - x_1)), \quad 0 \leq x \leq 10 \tag{28}$$

$$y_{\text{upper}} = 1 - A_2 \text{sech}(B_2(x - x_2)), \quad 0 \leq x \leq 10 \tag{29}$$

where sech is the hyperbolic secant function. The positive constants A_1, A_2 control the degree of constriction of the channel, while B_1, B_2 control the length of the constricted area. The stenosis location is controlled by the constants x_1 and x_2 for the lower and upper channel walls, respectively. We use $A_1 = 0.5, A_2 = 0.4, B_1 = 6,$ and $B_2 = 4,$ with the stenosis positioned at $x_1 = 3, x_2 = 4.$ We present results for the flow problem at $Re = 200.$ A grid-independent solution was obtained by a convergence study using successively finer node distributions. For the uniform node distribution, we use a total number of 61,073 nodes with 1760 of them representing the boundary. For the irregular node distribution, 12,061 nodes are used. For the DC PSE operators, $r = 2$ for first-order derivatives, and $r = 1$ for second-order derivatives. The cutoff r_c of the kernel function is $2.5h$ for interior nodes and $4.5h$ for boundary nodes, where h is the average inter-node distance ($h = 0.0125$).

Fig. 9 shows the solution’s stream function isocontours (a) and vorticity isocontours (b) for $Re = 200.$ Two vortices of opposite sign form downstream of the stenosis, the first clock-wise and the second counter-clock-wise. For future validation, cross-channel velocity profiles at $x = 3.5$ and $x = 5$ (where the vortices are) are tabulated in Tables 6 and 7, respectively. The results are compared with a solution obtained using COMSOL. The maximum absolute error between the DC PSE and COMSOL solution are 0.14 and 0.022 and the L2 norms are 0.04494 and 0.00962 for the u and v velocity components, respectively. To assess the robustness of our method against variations in the node distribution, we interpolated the solution from the irregular nodes to the regular nodes. The maximum absolute difference between the two solutions is $1.234 \times 10^{-9}.$

For the regular node distribution, the DC PSE operators had a maximum condition number of $2.64 \times 10^3,$ whereas MLS had $1.28 \times 10^{10}.$ For the irregular node distribution, the DC PSE operators had a maximum condition number of $2.5 \times 10^2,$ MLS of $3.5 \times 10^{13}.$ The solver using the MLS operators on the irregular node distribution did not converge unless further node refinement was done manually. The maximum absolute differences between the MLS and DC PSE solutions for the u and v velocity components is 3.25×10^{-2} and $5.64 \times 10^{-2}.$

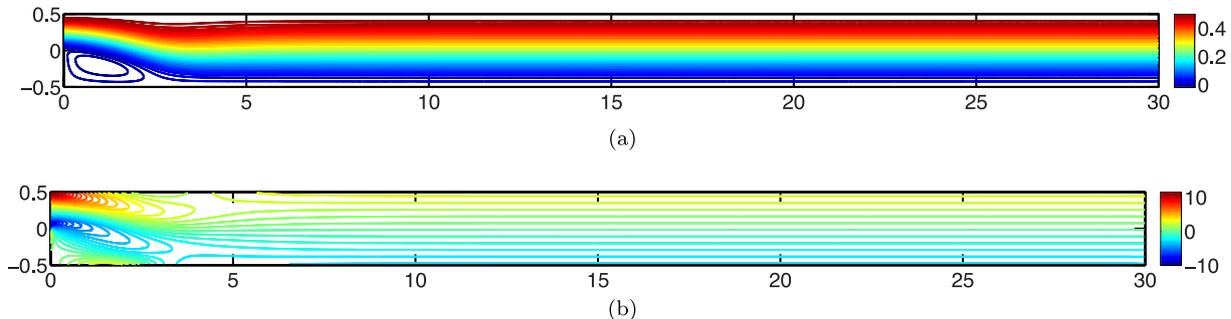


Fig. 5. Backward-facing step: (a) Stream function contours for $Re = 200$ and (b) vorticity isocontours. Stream function computed as in [25].

Table 3
Backward-facing step: Primary vortex strength and location, length of recirculation region, and comparison with RBF and FEM solutions from [12] for Re = 200.

	DC PSE	RBF	FEM
Length of recirculation	2.55	2.72	2.67
Ψ_{\min}	-0.0321	-0.0315	-0.0331
Ψ_{\min} location	(0.975, -0.200)	(1.333, -0.2167)	(1.0021, -0.2030)

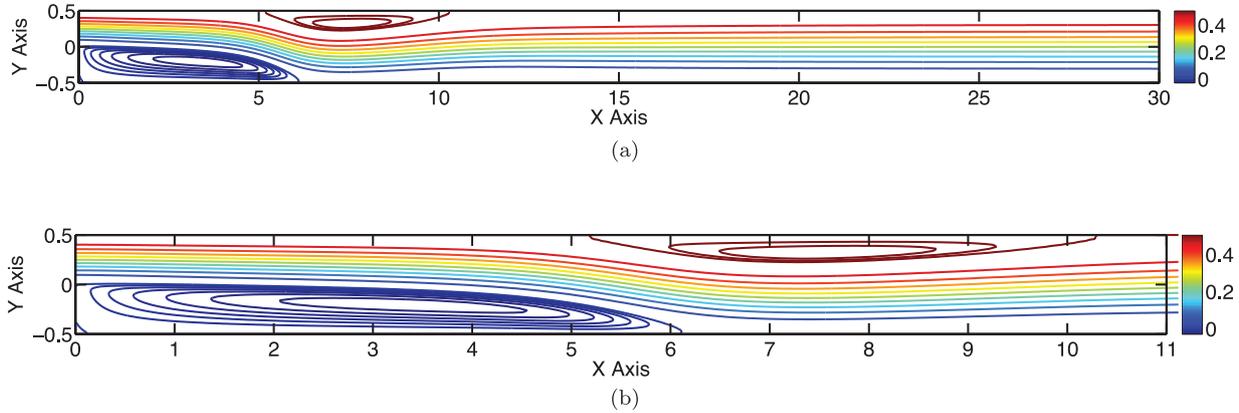


Fig. 6. Backward-facing step: (a) Stream function contours for Re = 800 and (b) vorticity isocontours. Stream function computed as in [25].

Table 4
Backward-facing step: Cross-channel profiles for the two velocity components u, v and the vorticity ω at $x = 7$ compared with the FEM solution from Ref. [25].

y	u		$v (\times 10^{-2})$		ω	
	DC PSE	FEM	DC PSE	FEM	DC PSE	FEM
0.5	0	0	0	0	-0.992	-1.034
0.45	-0.037	-0.038	-0.026	-0.027	-0.462	-0.493
0.4	-0.047	-0.049	-0.086	-0.086	0.083	0.061
0.35	-0.028	-0.032	-0.16	-0.147	0.651	0.635
0.3	0.019	0.015	-0.238	-0.193	1.256	1.237
0.25	0.097	0.092	-0.33	-0.225	1.916	1.888
0.2	0.21	0.204	-0.466	-0.268	2.622	2.588
0.15	0.357	0.349	-0.689	-0.362	3.291	3.267
0.1	0.532	0.522	-1.03	-0.544	3.746	3.751
0.05	0.719	0.709	-1.478	-0.823	3.777	3.821
0	0.893	0.885	-1.973	-1.165	3.269	3.345
-0.05	1.029	1.024	-2.427	-1.507	2.27	2.362
-0.1	1.105	1.105	-2.753	-1.778	0.958	1.046
-0.15	1.113	1.118	-2.89	-1.925	-0.445	-0.374
-0.2	1.053	1.062	-2.811	-1.917	-1.728	-1.684
-0.25	0.936	0.948	-2.522	-1.748	-2.733	-2.719
-0.3	0.778	0.792	-2.054	-1.423	-3.38	-3.392
-0.35	0.598	0.613	-1.447	-1	-3.632	-3.658
-0.4	0.415	0.428	-0.766	-0.504	-3.625	3.687
-0.45	0.226	0.232	-0.202	-0.118	-4.014	-4.132
-0.5	0	0	0	0	-5.087	5.14

4.4.2. Multiple stenoses

As a more complex case, we simulate flow past multiple stenoses in the same duct. First, the channel narrows with a symmetric stenosis close to the inlet, then recovers its width downstream. A second, more severe asymmetric stenosis follows halfway through the channel. After full recovery from this second stenosis, the channel slowly narrows towards the exit, which has a smaller diameter than the inlet. The lower and upper walls of the channel are defined by the equations:

$$y_{\text{lower}} = C_1 \left(1 - \cos \left(\frac{2\pi(x - D(x))}{E(x)} \right) \right), \quad 0 \leq x \leq 10 \quad (30)$$

$$y_{\text{upper}} = 1 - C_2(x) \left(1 - \cos \left(\frac{2\pi(x - D(x))}{E(x)} \right) \right), \quad 0 \leq x \leq 10 \quad (31)$$

where C_1 is a positive constant taken as $C_1 = 0.075$ and the piecewise-constant functions $C_2(x)$, $D(x)$, and $E(x)$ are defined as:

$$C_2(x) = \begin{cases} 0.075 & x < 3.2 \\ 0.225 & 3.2 \geq x \geq 5.8 \\ 0.075 & x > 5.8, \end{cases}$$

$$E(x) = \begin{cases} 3.2 & 0 \leq x < 3.2 \\ 2.6 & 3.2 \geq x \geq 5.8 \\ 7.2 & x > 5.8, \end{cases}$$

$$D(x) = \begin{cases} 0 & 0 \leq x < 3.2 \\ 2.6 & 3.2 \geq x \geq 5.8 \\ 7.2 & x > 5.8. \end{cases} \quad (32)$$

Table 5

Backward-facing step: Cross-channel profiles for the two velocity components u , v and the vorticity ω at $x = 15$ compared with the FEM solution from Ref. [25].

y	u		$v (\times 10^{-2})$		ω	
	DC PSE	FEM	DC PSE	FEM	DC PSE	FEM
0.5	0	0	0	0	2.027	2.027
0.45	0.101	0.101	0.021	0.021	2.011	2.013
0.4	0.202	0.202	0.072	0.072	2.019	2.023
0.35	0.303	0.304	0.14	0.14	2.051	2.058
0.3	0.407	0.408	0.209	0.207	2.08	2.09
0.25	0.511	0.512	0.263	0.26	2.063	2.075
0.2	0.611	0.613	0.292	0.288	1.947	1.959
0.15	0.703	0.704	0.288	0.283	1.694	1.703
0.1	0.778	0.779	0.252	0.245	1.292	1.298
0.05	0.83	0.831	0.186	0.18	0.761	0.761
0	0.853	0.853	0.102	0.095	0.146	0.141
-0.05	0.844	0.844	0.01	0.003	-0.49	-0.5
-0.1	0.804	0.804	-0.075	-0.081	-1.083	-1.096
-0.15	0.737	0.737	-0.143	-0.147	-1.575	-1.588
-0.2	0.649	0.649	-0.183	-0.185	-1.929	-1.939
-0.25	0.547	0.547	-0.19	-0.191	-2.134	-2.139
-0.3	0.438	0.438	-0.167	-0.166	-2.211	-2.213
-0.35	0.328	0.328	-0.12	-0.119	-2.211	-2.21
-0.4	0.218	0.218	-0.065	-0.065	-2.186	-2.184
-0.45	0.109	0.109	-0.019	-0.019	-2.174	-2.174
-0.5	0	0	0	0	-2.183	-2.185

Table 6

Single stenosis: Cross-channel velocity profiles for the velocity components u and v , the vorticity ω , and the stream function ψ at $x = 3.5$ for $Re = 200$.

y	u	v	ω	ψ
0.1	-0.1288	0.0513	1.9604	-0.0035
0.15	-0.1728	0.0881	0.0526	-0.0114
0.2	-0.1377	0.0914	-1.7952	-0.0194
0.25	-0.0362	0.0514	-3.5642	-0.024
0.3	0.1339	-0.0378	-5.7624	-0.0219
0.35	0.406	-0.1844	-8.8686	-0.0088
0.4	0.8113	-0.3842	-11.7023	0.0211
0.45	1.2787	-0.5858	-10.8888	0.0734
0.5	1.6183	-0.7093	-6.0149	0.1467
0.55	1.7404	-0.7408	-1.8799	0.2314
0.6	1.7289	-0.7265	-0.14	0.3185
0.65	1.6577	-0.6931	0.8637	0.4033
0.7	1.5362	-0.6404	2.2517	0.4834
0.75	1.3374	-0.5542	4.7004	0.5556
0.8	1.0153	-0.4161	8.2573	0.615
0.85	0.5404	-0.2198	12.2121	0.6545

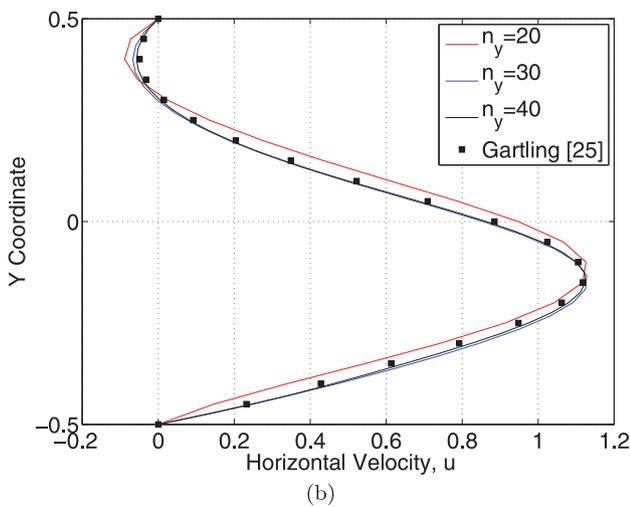
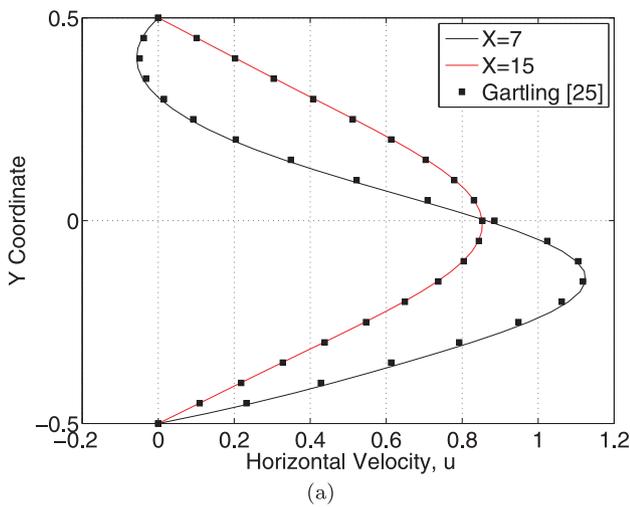


Fig. 7. Backward-facing step: (a) Horizontal velocity u profile at $x = 7$ and $x = 15$ for $Re=800$ compared to [25] and (b) grid convergence study for successfully higher numbers of y -direction nodes N_y ; u -velocity profile at $x = 7$, compared against [25].

The flow is simulated at both $Re = 200$ and $Re = 400$. For the uniform node distribution a total of 51,794 nodes are used with 1 737 nodes representing the boundary. For the irregular distribution, 88 164 nodes are used. For the DC PSE operators, $r = 2$ for first-order derivatives, and $r = 1$ for second-order derivatives. The cutoff radius r_c of the kernel is $2.5h$ for interior nodes and $4.5h$ for boundary nodes, with h being the average inter-node spacing ($h = 0.0125$).

We present the solution's velocity stream function and vorticity isocontours in Fig. 10 for both $Re = 200$ and $Re = 400$. A grid-independent solution was obtained in a convergence study using successively finer node distributions. For future validation, cross-channel velocity profiles are provided in Table 8 ($x = 3$) and Table 9 ($x = 6$) at the locations of the vortices. The results are compared with a solution obtained using COMSOL. The maximum absolute error between the DC PSE and COMSOL solutions for $Re=200$ is 0.16 and 0.044 and the L2 norms 0.0174 and 0.0027 for the u and v velocity components, respectively. For $Re = 400$ the maximum absolute errors are 0.17 and 0.042 and the L2 norms 0.0182 and 0.0029 for the u and v velocity components, respectively.

For the regular node distribution, the DC PSE operators had a maximum condition number of 2.51×10^3 (7.31×10^8 for MLS). For the irregular node distribution, the DC PSE operators had a maximum condition number of 2.5×10^2 (2.7×10^{13} for MLS). The solution using the MLS operators on the irregular node distribution did not converge without manual node curation. The maximum absolute differences in the u and v velocity components between the MLS and DC PSE solutions on the regular node distribution is 4.5×10^{-2} and 2.6×10^{-2} for $Re=200$, and 3.4×10^{-2} and 2.3×10^{-2} for $Re=400$. In summary, we find that DC PSE produces robust solutions across increasingly complex geometries without requiring manual curation on irregularly distributed nodes.

4.5. Flow in a bifurcation

As a final complex-geometry test case, we consider flow in an irregular bifurcating channel. This case models a 2D stenosed artery with a bifurcation. As boundary conditions, a uniform velocity of $u = 0.001 \frac{m}{s}$ and $v = 0$ is imposed at the inlet of the domain, while at both outlets we assume fully developed flow with $\frac{\partial u}{\partial n} = 0$. For the remaining walls, no-slip boundary conditions are applied ($u = v = 0$). The kinematic viscosity of the fluid is set to $\nu = \frac{0.001}{999} m^2 s^{-1}$. We again test both uniform and irregular node distributions.

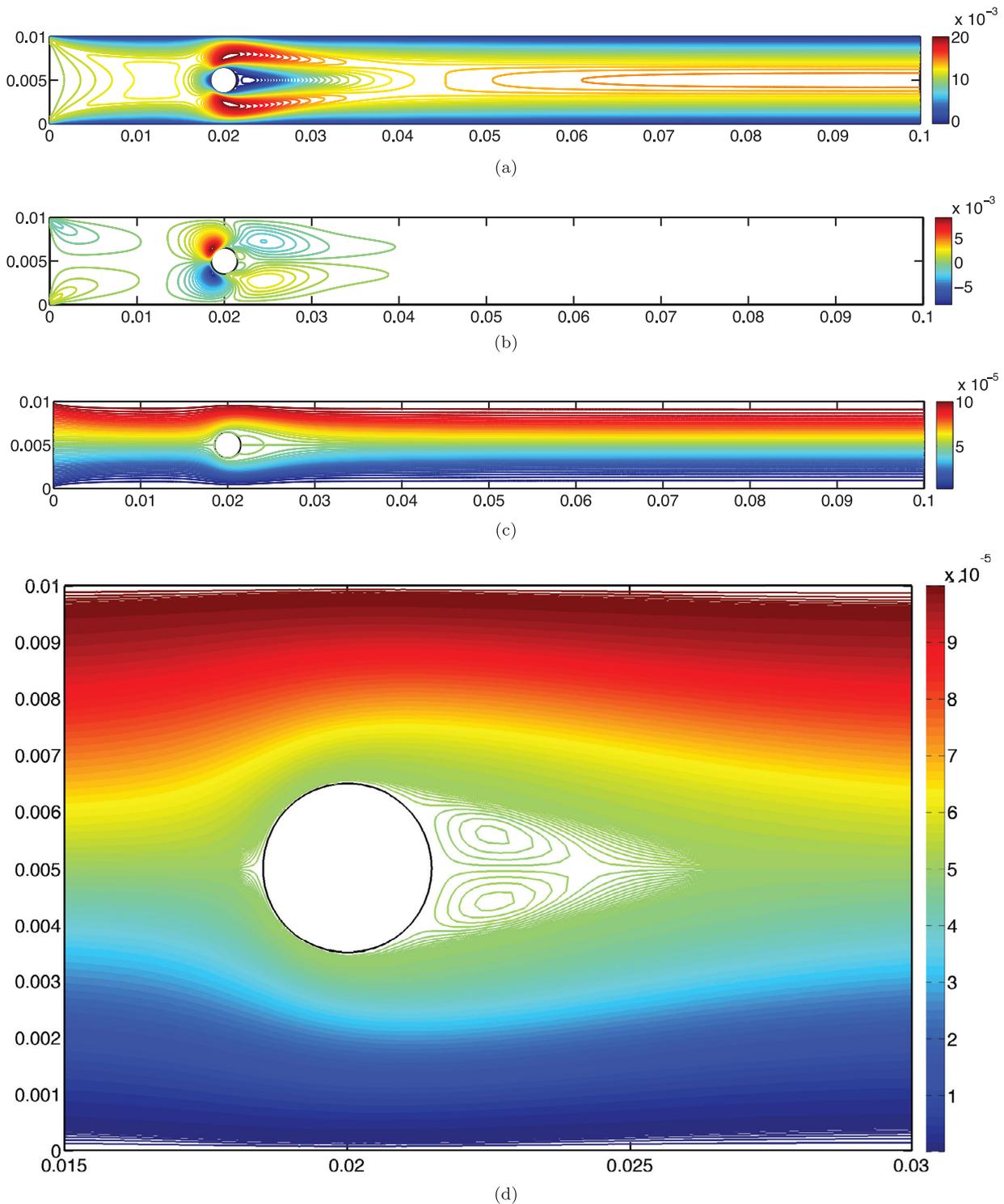


Fig. 8. Flow past a cylinder: (a) u velocity, (b) v velocity, and (c) streamlines of the flow for $Re = 30$. (d) Close-up of the streamlines in the recirculation zone behind the cylinder.

We present results for $Re=80$. For the uniform distribution, we use a total number of 25,282 nodes with 1840 nodes distributed on the boundary. For the irregular distribution, 35,421 nodes are used. For the DC PSE operators, $r = 2$ for first-order derivatives, and $r = 1$ for second-order derivatives. The cutoff r_c of the kernel is set to $2.5h$ for interior nodes and $3.5h$ for boundary nodes, with h being the average inter-node spacing ($h = 5 \times 10^{-4}$).

The numerical results for the isocontours of the u -velocity, v -velocity, and stream function are presented in Fig. 11. For validation, the results are compared against COMSOL, with the maximum absolute differences found as 4.7×10^{-2} and 2.64×10^{-2} for the u and v velocity components, respectively, interpolated from the uniform node distribution.

For the regular node distribution, the DC PSE operators had a maximum condition number of 2.5×10^2 (4.5×10^{15} for MLS). For

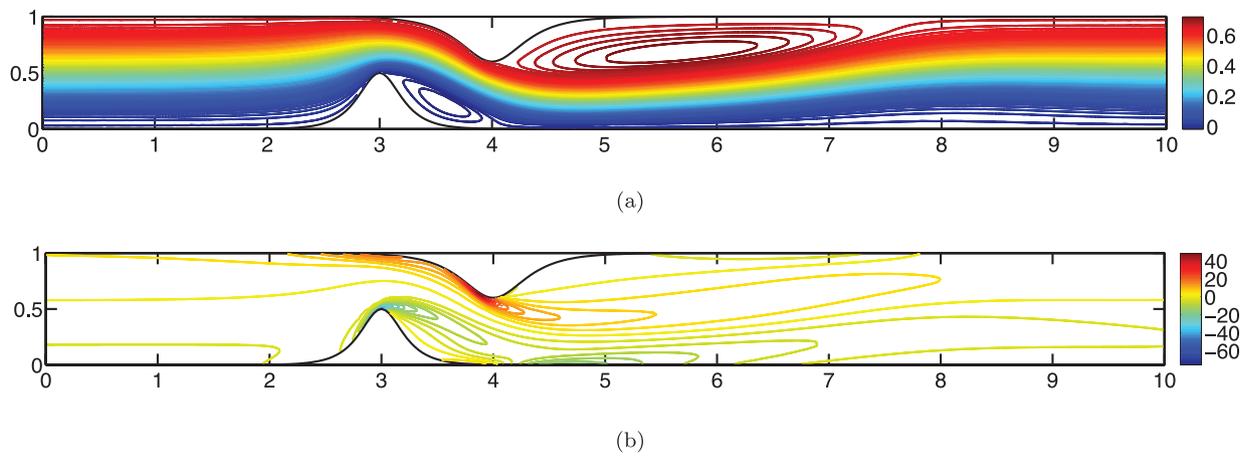


Fig. 9. Single stenosis: (a) Stream function isocontours for $Re = 200$ and (b) vorticity isocontours.

Table 7

Single stenosis: Cross-channel velocity profiles for the velocity components u and v , the vorticity ω , and the stream function ψ at $x = 5.0$ for $Re = 200$.

y	u	v	ω	ψ
0.1	1.2766	0.0286	-8.8914	0.0698
0.15	1.6384	0.0404	-5.5626	0.1433
0.2	1.8575	0.0475	-2.9395	0.2312
0.25	1.9468	0.0529	-0.2065	0.3269
0.3	1.8903	0.0568	2.9325	0.4234
0.35	1.6729	0.0573	6.0851	0.5131
0.4	1.3211	0.0508	8.0499	0.5884
0.45	0.919	0.0358	8.0208	0.6444
0.5	0.5569	0.0152	6.5724	0.681
0.55	0.2782	-0.0053	4.7954	0.7016
0.6	0.0816	-0.0218	3.3005	0.7103
0.65	-0.0504	-0.0323	2.1739	0.7108
0.7	-0.1338	-0.0367	1.3018	0.7061
0.75	-0.178	-0.0356	0.5733	0.6982
0.8	-0.1887	-0.0301	-0.0669	0.6889
0.85	-0.1697	-0.0219	-0.6303	0.6798
0.9	-0.1249	-0.0128	-1.1123	0.6723
0.95	-0.0584	-0.0045	-1.5171	0.6677

Table 8

Single stenosis: Cross-channel velocity profiles for the velocity components u and v , the vorticity ω , and the stream function ψ at $x = 3.5$ for $Re = 400$.

y	u	v	ω	ψ
0.1	0.2698	-0.0211	-3.2656	0.0121
0.15	0.443	-0.0339	-3.5979	0.0299
0.2	0.626	-0.0445	-3.6445	0.0566
0.25	0.8017	-0.0518	-3.3187	0.0923
0.3	0.9522	-0.0557	-2.6839	0.1363
0.35	1.0659	-0.0566	-1.9003	0.1869
0.4	1.1392	-0.0554	-1.1149	0.2422
0.45	1.1741	-0.0524	-0.3883	0.3002
0.5	1.1733	-0.0478	0.3026	0.359
0.55	1.1374	-0.0414	1.0141	0.4169
0.6	1.0645	-0.0331	1.7742	0.4721
0.65	0.9532	-0.0233	2.5335	0.5227
0.7	0.8068	-0.0131	3.1575	0.5669
0.75	0.6363	-0.004	3.487	0.603
0.8	0.4592	0.0024	3.4329	0.6304
0.85	0.2945	0.0054	3.0258	0.6491
0.9	0.1573	0.005	2.3812	0.6603
0.95	0.0565	0.0026	1.6252	0.6655

the irregular node distribution, the DC PSE operators had a maximum condition number of 2.5×10^2 (4.5×10^{15} for MLS). The solution using MLS operators did not converge without further manual node adjustment for both the regular and irregular node distributions. DC PSE produced consistent results in all cases. In addition, the DC PSE operators result in numerically robust schemes for irregular node distributions and geometries, where standard MLS operators fail to converge due to the large condition numbers of the moment matrices.

5. Conclusions

We presented for the first time the use of DC PSE operators for Eulerian meshless collocation schemes in computational fluid mechanics. We used a velocity-correction method to numerically solve incompressible steady-state flow problems in two-dimensional geometries. We have focused on the method's robustness in complex geometries and on irregular node distributions, demonstrating competitiveness and superiority of the DC PSE approach when compared to MLS.

First, we showed that DC PSE provides accurate results in regular geometries using the benchmark fluid flow problems of the lid-driven cavity and the backward-facing step. Both numerical and

qualitative results were given, with tabular results provided for future reference. We then showed the robustness of DC PSE across four complex-geometry problems. COMSOL was used as a reference solver for the complex geometries, demonstrating that our method provides consistent results. To assess the robustness to different node configurations and relative to established strong-form collocation methods, we repeated all numerical experiments using MLS to discretize the differential operators instead of DC PSE. Across all complex geometry problems, we found favorable robustness of the numerical solutions when DC PSE operators were used. This was in contrast to the MLS scheme that had convergence issues across all four complex geometry flow problems, especially when using irregular node distributions. In the flow past a cylinder, DC PSE did not require any refinement to the node distribution. This is in contrast to MLS that required manual removal of near-degenerate nodes to render the method stable. For the stenosis flow problems, DC PSE was again observed to be robust for both regular and irregular node distributions, whereas MLS did not converge for irregular node distributions. Finally, for flow in a bifurcation only DC PSE yielded solutions.

The condition numbers of the moment matrices at each node, or the equivalent matrix in DC PSE (Eq. (14)), has been associated with the numerical robustness of meshless schemes [14,31,40,60]. For all results we have presented maximum condition numbers

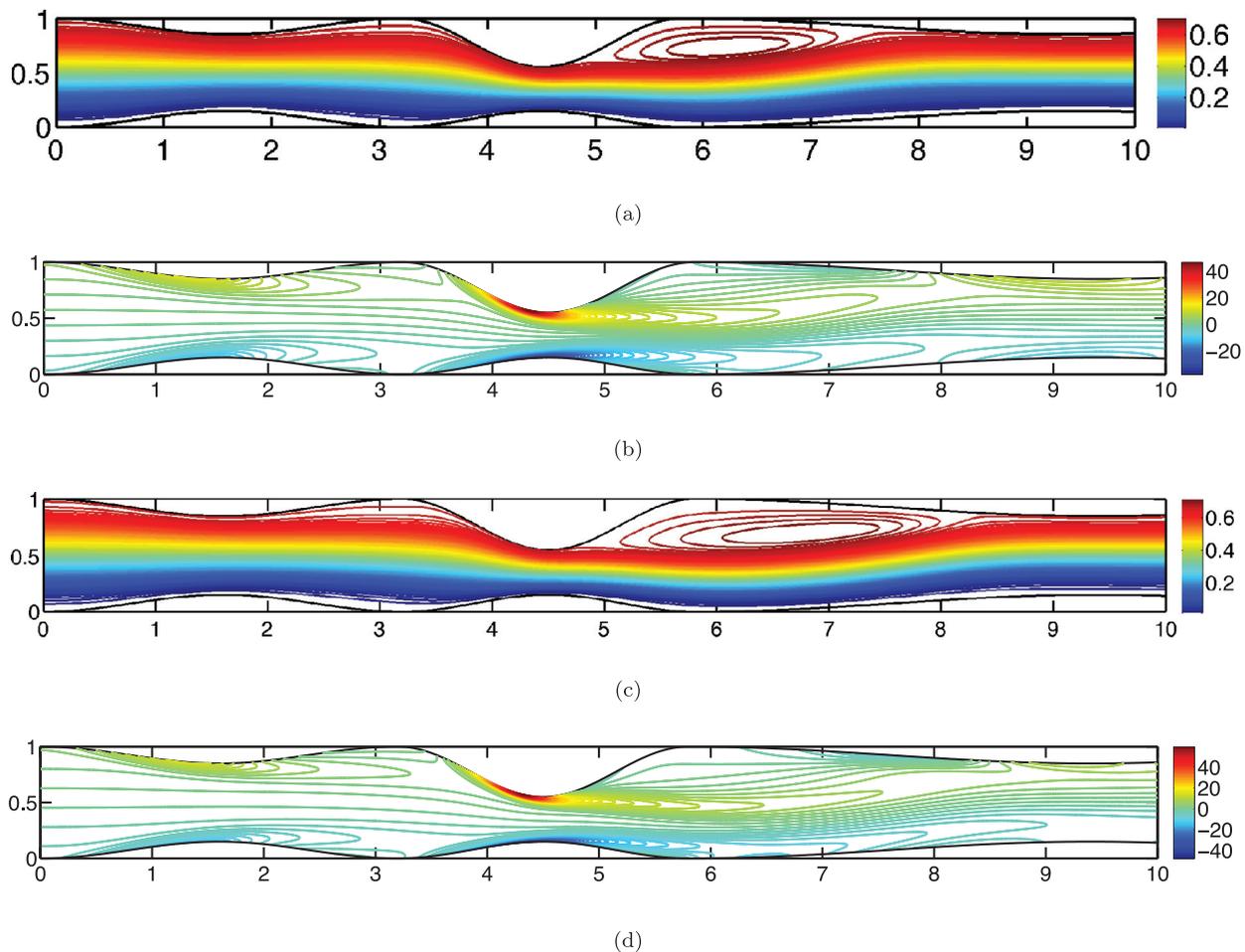


Fig. 10. Multiple stenoses: Stream function isocontours for $Re = 200$ (a) and $Re = 400$ (c), and vorticity isocontours for $Re = 200$ (b) and $Re = 400$ (d).

Table 9

Single stenosis: Cross-channel velocity profiles for the velocity components u and v , the vorticity ω , and the stream function ψ at $x = 6.0$ for $Re = 400$.

y	u	v	ω	ψ
0.1	0.5801	-0.0062	-6.7998	0.0271
0.15	0.9428	-0.0081	-7.3475	0.065
0.2	1.3055	-0.0032	-6.6267	0.1213
0.25	1.5983	0.0072	-4.5469	0.1943
0.3	1.7669	0.0195	-1.7367	0.279
0.35	1.7902	0.0306	1.2182	0.3685
0.4	1.6685	0.0387	3.9993	0.4555
0.45	1.4208	0.0417	6.1142	0.5332
0.5	1.0955	0.0379	6.9844	0.5962
0.55	0.7585	0.0276	6.5696	0.6425
0.6	0.4622	0.0139	5.4186	0.6728
0.65	0.228	0.0005	4.1244	0.6898
0.7	0.0548	-0.0095	2.9659	0.6966
0.75	-0.065	-0.0148	1.9572	0.6962
0.8	-0.1373	-0.0152	1.0273	0.6909
0.85	-0.1646	-0.0117	0.1312	0.6832
0.9	-0.1484	-0.0062	-0.7253	0.6752
0.95	-0.0915	-0.0013	-1.5138	0.669

for both DC PSE and MLS operators. Although the magnitudes of the condition numbers across the schemes can not be meaningfully compared due to prefactors, the variability and sensitivity of the condition numbers of either scheme is meaningful. We observed that the maximum condition numbers for MLS operators varied over seven orders of magnitude, whereas those of DC PSE

showed variations over only one order of magnitude. This shows that DC PSE operators are more robust. Condition numbers of 10^{12} and higher, as found for MLS, render it impossible to numerically obtain accurate solutions from the matrix.

Our results show that using DC PSE for operator discretization in meshless Eulerian collocation methods for fluid flow provides robust and accurate solutions across a range of 2D problems in both regular and irregular geometries. The robustness to node configuration of the DC PSE scheme was superior to that of MLS as an operator discretization. However, we have only presented a small subset of numerical problems with empirical results, Gaussian kernel functions, and a specific strong-form velocity-vorticity correction method. It is possible that for alternative problem formulations, window functions, and node distributions, the DC PSE and MLS operators have different performance and robustness properties. However, the results presented here are an encouraging indication, but not a final proof. In addition, we have only presented results for 2D steady-state flow problems, while the most computationally demanding applications require fluid flow solutions in complex 3D geometries. While the method presented here is extendable to transient and 3D cases, this is left for future research.

These limitations indicate that further exploration of the use of DC PSE operators for the numerical computation of solutions to systems of partial differential equations in complex geometries is needed. First steps should include the benchmarking and evaluation of accuracy and stability in both 3D and transient flow problems. The results also motivate further analytical analysis of the DC PSE operators and of their relationship to other meshless methods,

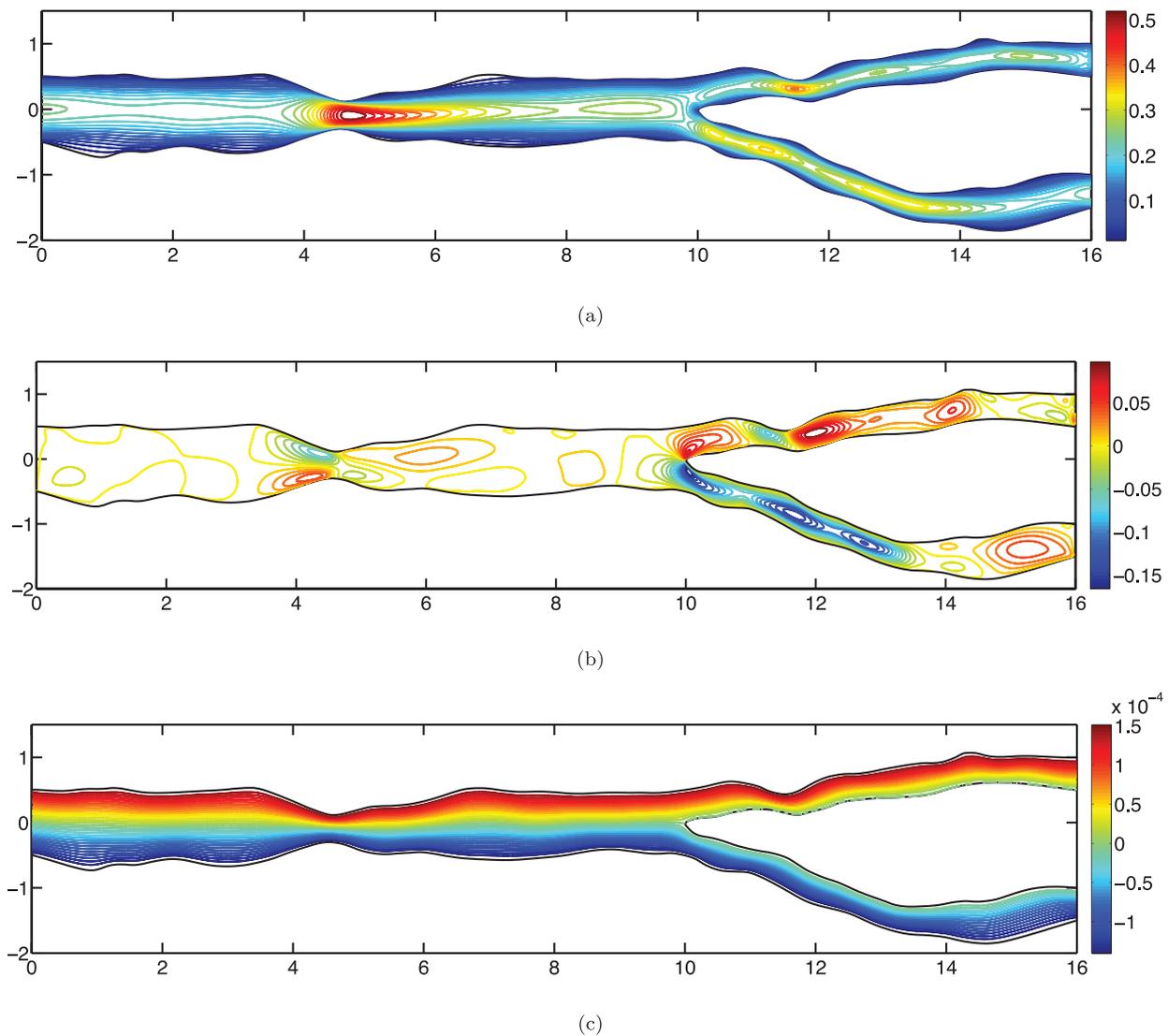


Fig. 11. Bifurcation: Isocontours for the u velocity component (a), v velocity component (b), and stream function (c) for $Re = 80$.

such as MLS and RKPM, and how this relates to their respective numerical robustness and condition numbers under varying node configurations. In addition, further research is required in order to understand how the choice of kernel function and/or window function affects the numerical robustness under different node distributions. This would help select optimal operators for different problems [61].

We believe that DC PSE operators have the potential for providing a robust paradigm for numerically solving fluid flow problems in complex geometries. In addition, DC PSE meshless methods can utilize the infrastructure provided by software libraries like the parallel particle-mesh library (PPM) [57] along with its domain-specific language (PPML) [6] to perform scalable parallel simulations. This would enable highly resolved stable numerical simulations of fluid flow problems in complex geometries.

Acknowledgments

This work was supported by the Max Planck Society, and by funding from the German Federal Ministry of Education and Research (BMBF) under funding code 031A099. R.R. further acknowledges a research fellowship award from the Alexander von Humboldt Foundation sponsored by the German Federal Foreign

Office and the German Federal Ministry of Education and Research (BMBF). B.L.C. acknowledges the DIGS-BB fellowship award. G.B. was funded by the German Research Foundation (DFG) via the Cluster of Excellence “Center for Advancing Electronics Dresden (cfaed)”, grant number DFG-EXC-1056.

References

- [1] Arefmanesh A, Najafi M, Abdi H. Meshless local Petrov-Galerkin method with unity test function for non-isothermal fluid flow. *CMES-Comput Model Eng Sci* 2008;25:9–25.
- [2] Armaly BF, Durst F, Pereira JCF, Schonung B. Experimental and theoretical investigation of backward-facing step flow. *J Fluid Mech* 1983;172:473–96.
- [3] Atluri S.N., Shen S.P. The Meshless Local Petrov-Galerkin (MLPG) method. Encino, USA: Tech Science Press; 2002.
- [4] Atluri SN, Zhu T. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Comput Mech* 1998a;22:117–27.
- [5] Atluri SN, Zhu TL. A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation. *Comput Model Simul Eng* 1998b;3(3):187–96.
- [6] Awile O, Mitrović M, Reboux S, Sbalzarini IF. A domain-specific programming language for particle simulations on distributed-memory parallel computers. In: Proc. III Intl. Conference on Particle-based Methods (PARTICLES). Stuttgart, Germany; 2013. p. p52.
- [7] Babuska I, Melenk JM. The partition of unity method. *Int J Numer Methods Eng* 1997;40:727–58.
- [8] Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Comput Methods Appl Mech Eng* 1996;139:3–47.

- [9] Belytschko T, Lu YY, Gu L. Element-free galerkin methods. *Int J Numer Methods Eng* 1994;37:229–56.
- [10] Bourantas GC, Loukopoulos VC. A meshless scheme for incompressible fluid flow using a velocity-pressure correction method. *Comput Fluids* 2013;88:189–99.
- [11] Bourantas GC, Petsi AJ, Skouras ED, Burganos VN. Meshless point collocation for the numerical solution of Navier–Stokes flow equations inside an evaporating sessile droplet. *Eng Anal Boundary Elem* 2012;36:240–7.
- [12] Bourantas GC, Skouras ED, Loukopoulos VC, Nikiforidis GC. Meshfree point collocation schemes for 2D steady state incompressible Navier–Stokes equations in velocity-vorticity formulation for high values of Reynolds number. *CMES–Comput Model Eng Sci* 2010b;59:31–63.
- [13] Bourantas GC, Skouras ED, Loukopoulos VC, Nikiforidis VC. Numerical solution of non-isothermal fluid flows using local radial basis functions (LRBF) interpolation and a velocity-correction method. *CMES–Comput Model Eng Sci* 2010a;64:187–212.
- [14] Bourantas GC, Skouras ED, Nikiforidis GC. Adaptive support domain implementation on the moving least squares approximation for Mfree methods applied on elliptic and parabolic PDE problems using strong-form description. *CMES–Comput Model Eng Sci* 2009;43:1–25.
- [15] Broomhead DS, Lowe D. Radial basis functions, multi-variable functional interpolation and adaptive networks. *Tech. Rep. DTIC Document*; 1988.
- [16] Cartwright JHE, Piro O, Tuval I. Fluid dynamics in developmental biology: moving fluids that shape ontogeny. *HFSP J* 2009;3(2):77–93.
- [17] Chung T. *Computational Fluid Dynamics*. 2nd. NY USA: Cambridge University Press; 2010.
- [18] Cottet G-H, Koumoutsakos P. *Vortex Methods – Theory and Practice*. New York: Cambridge University Press; 2000.
- [19] Degond P, Mas-Gallic S. The weighted particle method for convection-diffusion equations. Part 1: the case of an isotropic viscosity. *Math Comput* 1989a;53(188):485–507.
- [20] Degond P, Mas-Gallic S. The weighted particle method for convection-diffusion equations. Part 2: the anisotropic case. *Math Comput* 1989b;53(188):509–25.
- [21] Duarte CA, Oden JT. An *h-p* adaptive method using clouds. *Comput Methods Appl Mech Eng* 1996;139:237–62.
- [22] El Zahab Z, Divo E, Kassab AJ. A localized collocation meshless method (LCMM) for incompressible flows CFD modeling with applications to transient hemodynamics. *Eng Anal Boundary Elem* 2009;33(8):1045–61.
- [23] Eldredge JD, Leonard A, Colonius T. A general deterministic treatment of derivatives in particle methods. *J Comput Phys* 2002;180(2):686–709.
- [24] Erturk E, Corke TC, Gökçöl C. Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. *Int J Numer Methods Fluids* 2005;48(7):747–74.
- [25] Gartling DK. A test problem for outflow boundary conditions – flow over a backward-facing step. *Int J Numer Methods Fluids* 1990;11:953–67.
- [26] Gazzola M, Chatelain P, Van Rees WM, Koumoutsakos P. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *J Comput Phys* 2011;230(19):7093–114.
- [27] Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J Comput Phys* 1982;48(3):387–411. [http://dx.doi.org/10.1016/0021-9991\(82\)90058-4](http://dx.doi.org/10.1016/0021-9991(82)90058-4).
- [28] Gingold RA, Monaghan JJ. Kernel estimates as a basis for general particle methods in hydrodynamics. *J Comput Phys* 1982;46:429–53.
- [29] Hoffman J, Jansson J, de Abreu RV, Degirmenci NC, Jansson N, Müller K, et al. Unicorn: parallel adaptive finite element simulation of turbulent flow and fluid–structure interaction for deforming domains and complex geometry. *Comput Fluids* 2013;80:310–19.
- [30] Howard J, Grill SW, Bois JS. Turing's next steps: the mechanochemical basis of morphogenesis. *Nat Rev Mol Cell Biol* 2011;12:392–8.
- [31] Jin X, Li G, Aluru NR. Positivity conditions in meshless collocation methods. *Comput Methods Appl Mech Eng* 2004;193:1171–202.
- [32] Jülicher F, Kruse K, Prost J, Joanny J-F. Active behavior of the cytoskeleton. *Phys Rep* 2007;449:3–28.
- [33] Kennett DJ, Timme S, Angulo J, Badcock KJ. An implicit meshless method for application in computational fluid dynamics. *Int J Numer Meth Fluids* 2013;71:1007–28.
- [34] Kim J, Moin P. Application of a fractional-step method to incompressible Navier-Stokes equations. *J Comput Phys* 1985;59:308–23.
- [35] Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods. *Math Comput* 1981;37(155):141–58.
- [36] Lapenta G, linoya F, Brackbill JU. Particle-in-cell simulation of glow discharges in complex geometries. *IEEE Trans Plasma Sci* 1995;23(4):769–79.
- [37] Li S, Liu WK. Meshfree and particle methods and their applications. *Appl Mecha Rev* 2002;55(1):1–34.
- [38] Li S, Liu WK. *Meshfree Particle Methods*. Springer; 2004.
- [39] Lin H, Atluri SN. The meshless local Petrov-Galerkin (MLPG) method for solving incompressible Navier-Stokes equation. *CMES–Comput Model Eng Sci* 2001;2:117–42.
- [40] Liu G. *Mesh Free Methods, Moving beyond the Finite Elements Method*. CRC Press; 2002.
- [41] Liu G-R. *Meshfree Methods: Moving Beyond the Finite Element Method*. Taylor & Francis; 2009.
- [42] Liu WK, Chen Y, Uras RA, Chang CT. Generalized multiple scale reproducing kernel particle methods. *Comput Methods Appl Mech Eng* 1996;139(1):91–157.
- [43] Liu WK, Jun S, Zhang YF. Reproducing kernel particle methods. *Int J Numer Methods Fluids* 1995;20(8–9):1081–106.
- [44] Loukopoulos VC, Bourantas GC. MLPG6 for the solution of incompressible flow equations. *CMES–Comput Model Eng Sci* 2012;88:531–58.
- [45] Lucy LB. A numerical approach to the testing the fission hypothesis. *Astron J* 1977;82:1013–24.
- [46] Mahesh K, Stanantescu G, Moin P. A numerical method for large-eddy simulation in complex geometries. *J Comput Phys* 2004;197(1):215–40.
- [47] Mai-Duy N, Mai-Cao L, Tran-Cong T. Computation of transient viscous flows using indirect radial basis function networks. *CMES–Comput Model Eng Sci* 2007;18:59–78.
- [48] Mohammadi MH. Stabilized meshless local Petrov-Galerkin (MLPG) method for incompressible viscous fluid flows. *CMES–Comput Model Eng Sci* 2008;29:75–94.
- [49] Moin P, Apte SV. Large-eddy simulation of realistic gas turbine combustors. *AIAA J* 2006;44(4):698–708.
- [50] Morinishi K. An upwind finite difference scheme for meshless solvers. *J Comput Phys* 2003;189:1–23.
- [51] Mramor K, Vertnik R, Sarler B. Simulation of natural convection influenced by magnetic field with explicit local radial basis function collocation method. *CMES–Comput Model Eng Sci* 2013;92:327–52.
- [52] Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. *Comput Mech* 1992;10:307–18.
- [53] Nguyen VP, Rabczuk T, Bordas S, Duflo M. Meshless methods: a review and computer implementation aspects. *Math Comput Simul* 2008;79(3):763–813.
- [54] Onate E, Idelsohn S, Zienkiewicz O, Taylor R. A finite point method in computational mechanics. applications to convective transport and fluid flow. *Int J Numer Methods Eng* 1996;39:3839–66.
- [55] Ramaswamy R, Bourantas G, Jülicher F, Sbalzarini IF. A hybrid particle-mesh method for incompressible active polar viscous gels. *J Comput Phys* 2015;291:334–61.
- [56] Reboux S, Schrader B, Sbalzarini IF. A self-organizing lagrangian particle method for adaptive-resolution advection-diffusion simulations. *J Comput Phys* 2012;231(9):3623–46.
- [57] Sbalzarini IF, Walther JH, Bergdorf M, Hieber SE, Kotsalis EM, Koumoutsakos P. PPM – a highly efficient parallel particle-mesh library for the simulation of continuum systems. *J Comput Phys* 2006;215(2):566–88.
- [58] Schaback R. Error estimates and condition numbers for radial basis function interpolation. *Adv Comput Math* 1995;3:251–64.
- [59] Schrader B. Discretization-corrected PSE operators for adaptive multiresolution particle methods. *Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 19566, 2011; 2011. Ph.D. thesis*.
- [60] Schrader B, Reboux S, Sbalzarini IF. Discretization correction of general integral PSE operators for particle methods. *J Comput Phys* 2010;229:4159–82.
- [61] Schrader B, Reboux S, Sbalzarini IF. Choosing the best kernel: Performance models for diffusion operators in particle methods. *SIAM J Sci Comput* 2012;34(3):A1607–34.
- [62] Sellountos EJ, Sequeira A. An advanced meshless LBIE/RBF method for solving two-dimensional incompressible fluid flows. *Comput Mech* 2008;92:327–52.
- [63] Shepard D. A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings of the 1968 23rd ACM National Conference*. ACM; 1968. p. 517–24.
- [64] Sohn J. Evaluation of fidap on some classical laminar and turbulent. *Int J Numer Methods Fluids* 1988;8:149–1490.
- [65] Sridar D, Balakrishnan N. An implicit gridless type solver for the Navier-Stokes equations on unstructured meshes. *CFD J* 2009;9:1.
- [66] Triantafyllou MS, Triantafyllou GS, Yue DKP. Hydrodynamics of fishlike swimming. *Annu Rev Fluid Mech* 2000;32(1):33–53.
- [67] Wang Y-M, Chen S-M, Wu C-P. A meshless collocation method based on the differential reproducing kernel interpolation. *Comput Mech* 2010;45(6):585–606.
- [68] Wu YL, Liu GR, Gu YT. Application of meshless local Petrov-Galerkin (MLPG) approach to simulation of incompressible flow. *Numer Heat Transf, Part B* 2005;48:459–75.
- [69] Zhu T, Zhang JD, Atluri SN. A local boundary integral equation (LBIE) method in computational mechanics and a meshless discretization approach. *Comput Mech* 1998;22:117–27.
- [70] Zienkiewicz OC, Taylor RL, Nithiarasu P. *The Finite Element Method for Fluid Dynamics*. 6th. McGraw-Hill; 2005.