Bernal, Francisco; Heryudono, Alfa; Larsson, Elisabeth. (2017). Numerical Solution of the Viscous Flow Past a Cylinder with a Non-global Yet Spectrally Convergent Meshless Collocation Method. In: Bittencourt M., Dumont N., Hesthaven J. (eds). *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016: Selected Papers from the ICOSAHOM conference, June 27-July 1, 2016, Rio de Janeiro, Brazil.* (Lecture Notes in Computational Science and Engineering, 119). Springer. Pp. 495-507.

DOI: https://doi.org/10.1007/978-3-319-65870-4_35

Numerical solution of the viscous flow past a cylinder with a non-global yet spectrally convergent meshless collocation method (REVISED)

Francisco Bernal, Alfa Heryudono and Elisabeth Larsson

Abstract The flow of a viscous fluid past a cylinder is a classical problem in fluid-structure interaction and a benchmark for numerical methods in computational fluid dynamics. We solve it with the recently introduced radial basis function-based partition of unity method (RBF-PUM), which is a spectrally convergent collocation meshless scheme well suited to this kind of problem. The resulting discrete system of nonlinear equations is tackled with a trust-region algorithm, whose performance is much enhanced by the analytic Jacobian which is provided alongside. Preliminary results up to $Re = 60$ with just 1292 nodes are shown.

1 Introduction

The steady flow of a viscous fluid past a fixed, perpendicular cylinder is one of the simplest nontrivial problems in fluid dynamics. Moreover, it furnishes a relevant model for fluid-structure interaction (for instance, bridge pillars or the drillpipe of an oil plattform, see Figure 1). The problem was already tackled by Stokes [1], who was unable to find an analytical solution; a situation which—to the best of our knowledge—persists today. Consequently, it has been mostly studied numerically, with a special interest in the structure of the wake (the recirculating region immediatly downstream the cylinder,

Francisco Bernal
CMAP, École Polytechnique, Paris (France) e-mail: francisco.bernal@polytechnique.edu

Alfa R.H. Heryudono
Department of Mathematics, University of Massachusetts Dartmouth (USA) e-mail: aheryudono@umassd.edu

Elisabeth Larsson
Dept. of Information Technology, Uppsala University, Box 337, SE-751 05 Uppsala, (Sweden) e-mail: elisabeth.larsson@it.uu.se

shown in Figure 1) and the drag coefficient, as the Reynolds number (*Re*) grows.

Fig. 1 (Left) Highly symmetric flow past a cylinder. (Right) An oil rig.

The physical problem becomes unstable from $Re \approx 40$ onwards, when small perturbations in the symmetry of the incoming flow lead to an asymmetric vortex structure in the wake (the well-known Von Karman vortex street). In highly controlled laboratory conditions, it is possible to delay the onset of the physical instability until a higher value of *Re*, resulting in the symmetric, but unstable, pattern shown on Figure 1 (left).

From a numerical point of view, it is certainly possible to enforce symmetric conditions to arbitrary *Re* (unlike with the physical problem). However, as *Re* grows, the discretized equations become more and more ill-conditioned and it is challenging to keep the numerical simulation stable. (The roundoff errors appearing during the iterations of the nonlinear mathematical problem act in a similar way as flow perturbations and are prone to pick up the physical instability.) Another source of difficulty is derived from the fact that the computational domain must be very large compared with the area of interest (the wake), in order to enforce the far field boundary conditions (BCs) at a finite distance. These and other aspects are discussed in detail by Fornberg, who solved the flow problem up to $Re = 600$ [2] (with *Re* based on the cylinder diameter).

In this paper, we present a novel approach based on three ingredients. First, instead of the streamfunction/ vorticity formulation, we follow [3] in using the natural variables and apply a transformation of the unbounded domain into a finite rectangle (Section 2). Second, we discretize the resulting equations according to the recently introduced Radial Basis Functions-based Partition of Unity method (RBF-PUM) [4]. RBF-PUM has all the advantages of RBF collocation—such as spectral accuracy for smooth functions and flexibility in the choice of discretization—while being able to tackle much larger problems (Section 3). Third, in Section 4 the analytical Jacobian of the nonlinear algebraic system is derived, which is critical for convergence. This idea was introduced in [5], allowing highly nonlinear elliptic problems to

be solved with RBF meshless methods. Preliminary results up to $Re = 60$ are presented and briefly discussed in Section 5, and Section 6 concludes the paper with pointers to future work.

2 Transformed Navier-Stokes equations

A fixed, infinite circular cylinder of radius $a > 0$ is immersed in a fluid of kinematic viscosity $\nu > 0$, which flows steadily perpendicularly to the cylinder with far-field velocity $U > 0$ and far-field pressure P_0. The Reynolds number (based on the diameter) is $Re = 2aU/\nu$. By symmetry, the problem is two-dimensional, with the obstacle being the circular section. Let (x, y) be a Cartesian dimensionless frame (with $a = 1$) centred at the axis; (r, φ) polar coordinates (where $\varphi = 0$ marks the direction of advance of the flow); $\mathbf{u} = (u, v)$ the dimensionless velocity field (with $U = 1$); P the pressure and $p = P - P_0$. The steady Navier-Stokes equations are then given by

$$\frac{Re}{2}(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla^2\mathbf{u} - \nabla p, \qquad \nabla\cdot\mathbf{u} = 0.$$

They are supplemented with five boundary conditions:

$$\begin{aligned} u(r=\infty,\varphi) &= 1, & v(r=\infty,\varphi) &= 0, & &\text{(unperturbed flow)}\\ u(r=1,\varphi) &= 0, & v(r=1,\varphi) &= 0, & &\text{(non-slip condition)}\\ & & p(r=\infty,\varphi) &= 0. & &\text{(unperturbed pressure)} \end{aligned}$$

Instead of taking a large finite domain and enforcing the BCs far away from the cylinder, the infinite domain is compressed into the rectangle $[0,1]\times[0,2\pi]$ via the following transformation [3]:

$$\xi = 1 - 1/r \qquad (\text{ such that } \xi(r=1) = 0 \text{ and } \xi(r=\infty) = 1).$$

Moreover, $\partial/\partial r = (1-\xi)^2\partial/\partial\xi$, so that the unit vectors point in the same direction: $\mathbf{i}_r = \mathbf{i}_\xi$. Denoting as $\dot\xi$ and $\dot\varphi$ the components of the fluid velocity in the new coordinates, the velocity field is transformed as

$$\mathbf{u} = \dot\xi\mathbf{i}_r + (1-\xi)\dot\varphi\mathbf{i}_\varphi.$$

(Note that the dot notation is intuitive for velocities and useful so as not to overload the notation, but recall that the problem is steady and involves no time derivatives.) We also introduce the following notation: $\partial^2_{\xi\varphi}\dot\varphi = \partial^2\dot\varphi/\partial\xi\partial\varphi$, $\partial_\varphi p = \partial p/\partial\varphi$, etc. The Navier-Stokes equations in the variables (ξ, φ) are

$$\mathcal{W}_1 = \mathcal{W}_2 = \mathcal{W}_3 = 0, \tag{1}$$

where:

$$
\begin{aligned}
\mathcal{W}_1\left(\xi,\dot{\xi},\dot{\varphi},\partial_\xi\dot{\xi},\partial_\varphi\dot{\xi},\partial_\varphi\dot{\varphi},\partial_\xi p,\partial^2_{\xi\xi}\dot{\xi},\partial^2_{\varphi\varphi}\dot{\xi}\right) &= \frac{Re}{2}\left[(1-\xi)\dot{\xi}\partial_\xi\dot{\xi}+\dot{\varphi}\partial_\varphi\dot{\xi}-\dot{\varphi}^2\right]+\\
(1-\xi)\partial_\xi p-(1-\xi)^3\partial^2_{\xi\xi}\dot{\xi}-(1-\xi)\partial^2_{\varphi\varphi}\dot{\xi}+(1-\xi)^2\partial_\xi\dot{\xi}+2(1-\xi)\partial_\varphi\dot{\varphi}+(1-\xi)\dot{\xi},&\\
\mathcal{W}_2\left(\xi,\dot{\xi},\dot{\varphi},\partial_\varphi\dot{\xi},\partial_\varphi\dot{\varphi},\partial_\xi\dot{\varphi},\partial_\varphi p,\partial^2_{\xi\xi}\dot{\varphi},\partial^2_{\varphi\varphi}\dot{\varphi}\right) &= \frac{Re}{2}\left[(1-\xi)\dot{\xi}\partial_\xi\dot{\varphi}+\dot{\varphi}\partial_\varphi\dot{\varphi}+\dot{\xi}\dot{\varphi}\right]+\\
\partial_\varphi p-(1-\xi)^3\partial^2_{\xi\xi}\dot{\varphi}-(1-\xi)\partial^2_{\varphi\varphi}\dot{\varphi}+(1-\xi)^2\partial_\xi\dot{\varphi}-2(1-\xi)\partial_\varphi\dot{\xi}+(1-\xi)\dot{\varphi},&\\
\mathcal{W}_3\left(\xi,\dot{\xi},\partial_\xi\dot{\xi},\partial_\varphi\dot{\varphi}\right) = (1-\xi)\partial_\xi\dot{\xi}+\partial_\varphi\dot{\varphi}+\dot{\xi}.&
\end{aligned}
\tag{2}
$$

Remark. The last two terms in [3, formula (7)] are seemingly wrong.

Moreover, since the problem is symmetric along the x axis, only $0 \le \varphi \le \pi$ needs to be considered. (The boundary conditions along the x axis are now of reflecting type to enforce the symmetry.) The transformed BCs are sketched in Figure 2.

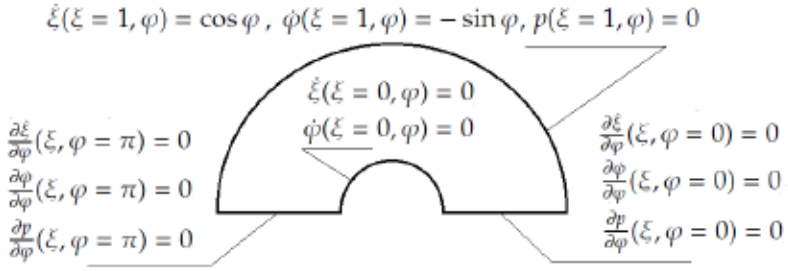

Fig. 2 Transformed BCs. The infinite anular section ($1 \le r < \infty, 0 \le \varphi \le \pi$) sketched above has been compressed into the rectangle $[0,1]\times[0,\pi]$.

3 Meshless discretization using the RBF-PUM

In this Section we briefly review the formulation described in detail in [4].

Kansa's method. Let $\mathbf{q} = (\xi,\varphi) =\in \Omega \subset \mathbb{R}^2$, where $\Omega = [0,1]\times[0,\pi]$, and let the *pointset* $\{\mathbf{q}_1,\ldots,\mathbf{q}_N\}$ be a discretization of Ω and its boundary $\partial\Omega$ into N scattered, distinct points (called *nodes*). A Radial Basis Function (RBF) approximation to the pressure is the *RBF interpolant*

$$p(\xi,\varphi) = p(\mathbf{q}) = \sum_{i=1}^{N} \lambda_i \phi_i(\|\mathbf{q}-\mathbf{q}_i\|). \tag{3}$$

Remark. For notational convenience, we use the symbols ξ, ϕ, and p both for the exact solution of equations (1) and for their RBF interpolants.

Above, $\|\cdot\|$ is the Euclidean norm, and $\phi_i(\mathbf{q})$ is the chosen RBF, which also contains a *shape parameter* $\epsilon > 0$. For instance, the Gaussian RBF

$$\phi_i(\mathbf{q}) = \exp\left[-(\epsilon\|\mathbf{q}-\mathbf{q}_i\|)^2\right].$$

The RBF coefficients $\lambda_1,\ldots,\lambda_N$ can be found by collocation. Requesting that $p(\mathbf{q})$ in (3) interpolates the nodal pressures $p(\mathbf{q}_1),\ldots,p(\mathbf{q}_N)$ leads to

$$\begin{pmatrix} p_1 \\ \vdots \\ p_N \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{q}_1) & \cdots & \phi_N(\mathbf{q}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{q}_N) & \cdots & \phi_N(\mathbf{q}_N) \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_N \end{pmatrix} \Rightarrow \lambda = [\phi]^{-1}\mathbf{p},$$

where we have introduced the notation of *nodal vectors and matrices*. Moreover, given $f_i : \Omega \mapsto \mathbb{R}$, $1 \le i \le N$, then $\mathbf{f}(\mathbf{q}) \in \mathbb{R}^N = [f_1(\mathbf{q}),\ldots,f_N(\mathbf{q}_N)]$. Note that (with a fixed ϵ) $[\phi]$ is symmetric since $[\phi]_{ij} = \phi_j(\|\mathbf{q}_i-\mathbf{q}_j\|) = \phi_i(\|\mathbf{q}_j-\mathbf{q}_i\|) = [\phi]_{ji}$. This allows to express the RBF interpolant in terms of the (unknown) nodal pressures rather than RBF coefficients,

$$p(\mathbf{q}) = \phi^T(\mathbf{q})\lambda = \phi^T(\mathbf{q})[\phi]^{-1}\mathbf{p} = \psi^T(\mathbf{q})\mathbf{p}, \tag{4}$$

where

$$\psi(\mathbf{q}) = [\phi]^{-1}\phi(\mathbf{q}) \Rightarrow \psi_i(\mathbf{q}_j) = \delta_{ij} \text{ (Kronecker's delta)}.$$

The functions $\psi_i(\mathbf{q})$ are the *cardinal basis functions* of the RBF ϕ and the pointset.

Linear boundary value problems (BVPs) can readily be solved as follows. Let the PDE be defined by the interior operator $\mathcal{L}^{PDE}p = f$ and the BCs by the boundary operator $\mathcal{L}^{BC}p = g$. For notational convenience, let the entire BVP be described by $\mathcal{L} = h$, where $h(\mathbf{q})$ and $\mathcal{L}_\mathbf{q}$ are

$$\mathcal{L} = \begin{cases} \mathcal{L}^{PDE}, & \text{if } \mathbf{q} \in \Omega/\partial\Omega, \\ \mathcal{L}^{BC}, & \text{if } \mathbf{q} \in \partial\Omega, \end{cases} \qquad h(\mathbf{q}) = \begin{cases} f(\mathbf{q}), & \text{if } \mathbf{q} \in \Omega/\partial\Omega, \\ g(\mathbf{q}), & \text{if } \mathbf{q} \in \partial\Omega. \end{cases}$$

Applying $\mathcal{L}$ on the RBF interpolant of p yields the square linear system

$$\mathcal{L}p(\mathbf{q}) = \left[\mathcal{L}\phi_1(\mathbf{q}),\ldots,\mathcal{L}\phi_N(\mathbf{q})\right]^T[\phi]^{-1}\mathbf{p} = \mathbf{h}, \tag{5}$$

whose solution is $\mathbf{p}$, and $p(\mathbf{q})$ can be reconstructed on $\Omega \cup \partial\Omega$ by (4).

The just described algorithm is easy to code, meshfree, geometrically flexible and spectrally convergent for smooth problems (see [5]). A drawback is that the last property comes at the expense of a fully populated matrix in (5). For this reason, it is often thought that Kansa's method loses many of its advantages after a few thousand nodes. The RBF-PUM pushes N beyond that *without loss of performance*. It does so by "embedding" Kansa's method into a higher level of discretization, in order to attain a sparse matrix.

RBF-PUM. Let $\{\Omega_j\}_{j=1}^M$ such that $\Omega \subset \cup_{j=1}^N \Omega_j$ be an open *cover* of Ω satisfying a pointwise overlap condition and

$$\forall \mathbf{q} \in \Omega, \qquad \Xi(\mathbf{q}) = \{j \,|\, \mathbf{q} \in \Omega_j\} \text{ and } \#\Xi(\mathbf{q}) \leq K,$$

where # is the cardinal of the set, and K a constant independent of M. Further, let $\{w_j(\mathbf{q})\}_{j=1}^M$ be a partition of unity on Ω (i.e. $\forall \mathbf{q} \in \Omega, \sum_{j=1}^M w_j(\mathbf{q}) = 1$) subordinate to the cover. (Figure 3 shows an illustrative pointset and cover.) For w_j being C^2, this can be attained with Shepard's method, using circular patches Ω_j and Wendland's compactly supported RBF, $\phi_W(\mathbf{q})$:

$$w_j(\mathbf{q}) = \frac{\phi_{\mathbf{W}}(\mathbf{q})}{\sum\limits_{\mathbf{k}\in\Xi(\mathbf{q})} \phi_{\mathbf{W}}(\mathbf{q})} \text{ where } \phi_{\mathbf{W}}(\mathbf{q}) = \begin{cases} (1-\|\mathbf{q}\|)^4(4\|\mathbf{q}\|+1) & \text{if } 0 \leq \|\mathbf{q}\| \leq 1, \\ 0 & \text{if } \|\mathbf{q}\| > 1. \end{cases} \tag{6}$$

The solutions of $p(\mathbf{q})$ on each patch of the cover—i.e. $p_j(\mathbf{q}) = p(\mathbf{q} \in \Omega_j)$—can be "glued together" by means of the partition of unity:

$$p(\mathbf{q}) = \sum_{j\in\Xi(\mathbf{q})} w_j(\mathbf{q}) p_j(\mathbf{q}). \tag{7}$$

$p_j(\mathbf{q})$ can be expressed in terms of the nodal values on patch j, and by linearity,

$$\mathcal{L}p(\mathbf{q}) = \sum_{j\in\Xi(\mathbf{q})} \sum_{k\in\Omega_j} \mathcal{L}\big(w_j(\mathbf{q})\psi_k(\mathbf{q})\big) p_k.$$

Specifically, partial derivatives can be computed according to the formula

$$\frac{\partial^{|\alpha|}}{\partial q^\alpha} p(\mathbf{q}) = \sum_{j\in\Xi(\mathbf{q})} \sum_{k\in\Omega_j} \frac{\partial^{|\alpha|}}{\partial q^\alpha}\big[w_j(\mathbf{q})\psi_k(\mathbf{q})\big] p_k = \sum_{j\in\Xi(\mathbf{q})} \sum_{k\in\Omega_j} \left[\sum_{\beta\leq\alpha} \binom{\alpha}{\beta} \frac{\partial^{|\alpha-\beta|} w_j}{\partial q^{\alpha-\beta}} \frac{\partial^{|\beta|}\psi_k}{\partial q^\beta}\right] p_k,$$

where $\partial^{|\alpha|}/\partial^\alpha$ is the usual multi-index notation [4]. For instance, the angular derivative at a node with $\xi = \xi'$ and $\varphi = \varphi'$ is

$$\partial_\varphi p(\xi', \varphi') = \sum_{j\in\Xi(\mathbf{q}')} \sum_{k\in\Omega_j} \Big[\frac{\partial w_j}{\partial q_y}(\mathbf{q}')\psi_k(\mathbf{q}') + w_j(\mathbf{q}')\frac{\partial \psi_k}{\partial q_y}(\mathbf{q}')\Big] p_k \Rightarrow \partial_\varphi p = \partial_\varphi p(\mathbf{p}).$$

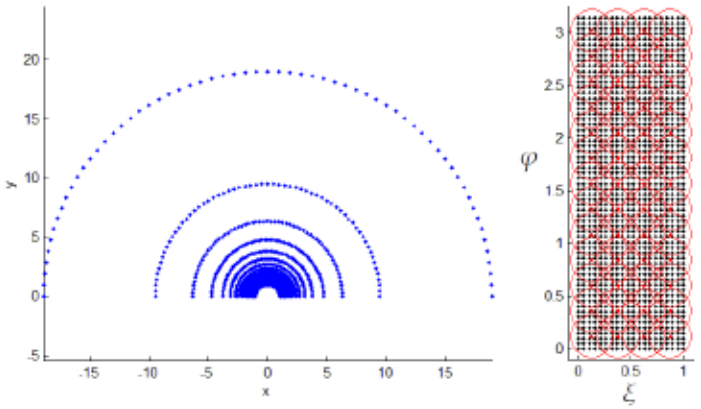

Fig. 3 PURBF discretization in the numerical example in Section 5. (Left) Nodes in real space. (Right) Nodes in the (ξ,φ) frame, with a cover overlaid.

Above, we stress the fact that partial derivatives of an RBF interpolant can be expressed as a linear combination of its nodal values, with the coefficients depending only on the discretization (i.e. pointset, cover, partition of unity and RBFs), but independent of the function being differentiated. Therefore, one can explicitly compute the matrices for the nodal vectors of the required derivatives at start, and reuse them as needed. For instance, in the previous example, calling $\left((\partial_\varphi p)_1,\ldots,(\partial_\varphi p)_N\right)^T =: \partial_\varphi \mathbf{p} : [\partial_\varphi]\mathbf{p}$, with

$$[\partial_\varphi]_{mn} = \begin{cases} \sum\limits_{j\in\Xi(\mathbf{q}_n)} \left[\frac{\partial w_j}{\partial\varphi}(\mathbf{q}_m)\psi_n(\mathbf{q}_m) + w_j(\mathbf{q}_m)\frac{\partial\psi_n}{\partial\varphi}(\mathbf{q}_m)\right] & \text{if } \Xi(\mathbf{q}_m)\cap\Xi(\mathbf{q}_m)\neq\emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Note that matrices such as $[\partial_\varphi]$ are very sparse because only entries with indices associated to nodes in overlapping patches are nonzero. Analogously, let us define the RBF interpolants of $\dot{\xi}(\xi,\varphi)$ and $\dot{\varphi}(\xi,\varphi)$ as

$$\dot{\xi}(\mathbf{q}) = \sum_{j\in\Xi(\mathbf{q})}\sum_{k\in\Omega_j}\left(w_j(\mathbf{q})\psi_k(\mathbf{q})\right)\dot{\xi}_k, \qquad \dot{\varphi}(\mathbf{q}) = \sum_{j\in\Xi(\mathbf{q})}\sum_{k\in\Omega_j}\left(w_j(\mathbf{q})\psi_k(\mathbf{q})\right)\dot{\varphi}_k.$$

Since there are 3 PDEs in (1), the RBF collocation system has $3N$ equations with $3N$ unknowns $(\dot{\xi},\dot{\varphi},\mathbf{p})$. Let us define the *system nodal vector* as

$$\mathbf{X} = (\dot{\xi}_1,\ldots,\dot{\xi}_N,\dot{\varphi}_1,\ldots,\dot{\varphi}_N,p_1,\ldots,p_N)^T. \tag{9}$$

We shall now tackle the collocation of the complete Navier-Stokes equations. Let us start with the BCs, which are linear. We shall assume, without loss of generality, that there are $N_B < N$ nodes discretizing $\partial\Omega$ and that they

are ordered first: $\mathbf{q}_i \in \partial\Omega$ iff $i \leq N_\mathrm{B}$. Moreover, the BC nodes are in turn ordered as follows: first, the set FAR of far-field nodes (with $\xi = 1$); then, the set $AXIS$ of nodes on the $x-$axis (with either $\varphi = 0$ or $\varphi = \pi$); and finally, the set CYL of nodes on the cylinder (with $\xi = 0$). For an $N \times N$ matrix $[\mathcal{L}]$ such as $[\partial_\varphi]$ in (8), let $[\mathcal{L}]_{SET \times N}$ represent the block with all the N columns and the $\#SET$ rows in the set SET. Then, collocation of the BCs in Figure 2 yields the following contiguous matrix block B of size $\#BCs \times 3N$, where $\#BCs = 3(\#FAR + \#AXIS) + 2\#CYL$ and I is the $N \times N$ identity matrix:

$$\begin{pmatrix} I_{FAR\times N} & 0 & 0 \\ 0 & I_{FAR\times N} & 0 \\ 0 & 0 & I_{FAR\times N} \\ [\partial_\varphi]_{AXIS\times N} & 0 & 0 \\ 0 & [\partial_\varphi]_{AXIS\times N} & 0 \\ 0 & 0 & [\partial_\varphi]_{AXIS\times N} \\ I_{CYL\times N} & 0 & 0 \\ 0 & I_{CYL\times N} & 0 \end{pmatrix} \mathbf{X} =: B\mathbf{X} = \begin{pmatrix} \left[\cos\varphi_1, \ldots, \cos\varphi_{\#(FAR)}\right]^T \\ \left[-\sin\varphi_1, \ldots, -\sin\varphi_{\#(FAR)}\right]^T \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{10}$$

The collocations of the nonlinear operators $\mathcal{W}_1, \mathcal{W}_2$, and $\mathcal{W}_3$ on the interior nodes follow. They are $3N - \#BCs$ nonlinear algebraic equations in $\mathbf{X}$. (For instance, a product like $\dot{\xi}\partial_\xi\dot{\xi}$ depends quadratically on $\mathbf{X}$.) We write $W(\mathbf{q}', \mathbf{X})$ to denote the collocation of a nonlinear operator $\mathcal{W}$ acting on the RBF *interpolants* at node $\mathbf{q}'$. In sum, the discretized system of collocation equations reads (note that $\mathcal{W}_3$ is also enforced on the nodes in CYL)

$$\begin{pmatrix} B\mathbf{X} \\ W_1(\mathbf{q}_{N_\mathrm{B}+1}, \mathbf{X}) \\ \vdots \\ W_1(\mathbf{q}_N, \mathbf{X}) \\ W_2(\mathbf{q}_{N_\mathrm{B}+1}, \mathbf{X}) \\ \vdots \\ W_2(\mathbf{q}_N, \mathbf{X}) \\ W_3(\mathbf{q}_{N_\mathrm{B}-\#CYL+1}, \mathbf{X}) \\ \vdots \\ W_3(\mathbf{q}_N, \mathbf{X}) \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \text{or} \qquad \begin{pmatrix} B\mathbf{X} \\ \mathbf{W}_{123}(\mathbf{X}) \end{pmatrix} = \begin{pmatrix} \cos\varphi_1 \\ \vdots \\ -\sin\varphi_{\#FAR} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{11}$$

Elimination of the BCs. The block B in (10) and (11) contains only linear equations because the BCs are linear. It is advantageous to eliminate them before solving the nonlinear equations, also shrinking the size of the system to be solved. An optimally stable way of doing so is described in [5], which involves the QR decomposition of B:

$$B^T\Pi = \begin{bmatrix} Q_1 Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix},$$

where Π is a permutation matrix, R is upper triangular, and $Q_1 \in \mathbb{R}^{3N\times \#BCs}$ and $Q_2 \in \mathbb{R}^{3N\times(3N-\#BCs)}$ are made up of orthogonal columns. Then, the solution vector $\mathbf{X}$ can be expressed in terms of a fixed vector and a smaller vector $\mathbf{Y}$ (which remains to be found) as

$$\mathbf{X} = Q_1 R^{-T}\Pi^T \mathbf{g} + Q_2 \mathbf{Y}. \tag{12}$$

After solving for $\mathbf{Y}$, the nodal values of pressure and velocity (i.e. the vector $\mathbf{X}$ in (9)) can be found according to (12), and with them the pressure and velocity anywhere in the infinite domain can be reconstructed by virtue of (6) and (7). The nonlinear system of equations to be solved is thus

$$\mathbf{W}_{123}\left(Q_1 R^{-T}\Pi^T \mathbf{g} + Q_2 \mathbf{Y}\right) = 0. \tag{13}$$

($\#PDEs := 3N - \#BCs$ nonlinear equations in $\#PDEs$ unknowns $\mathbf{Y}$).

4 Analytic Jacobian of the RBF-PUM system

In order to solve (13), we apply the trust region algorithm, which transforms a rootfinding problem (the root being the vector solution of the system) into a minimization problem for the sum-of-squares residual in $\mathbb{R}^{\#PDEs}$. The method in the context of RBF approximations is discussed in detail in [5]. Because the residual landscape is highly nonconvex, it is critical both for convergence and for speed that the analytic Jacobian (i.e. the matrix J such that $J_{ij} = \frac{\partial (W_{123})_i}{\partial Y_j}$) be available. It is given by (see [5])

$$J = \sum_{k=1} \begin{pmatrix} diag\left[\frac{\partial W_1}{\partial\left(\mathcal{L}_k \xi\right)}\right]\left[\mathcal{L}_k\right] & diag\left[\frac{\partial W_1}{\partial\left(\mathcal{L}_k \phi\right)}\right]\left[\mathcal{L}_k\right] & diag\left[\frac{\partial W_1}{\partial\left(\mathcal{L}_k p\right)}\right]\left[\mathcal{L}_k\right] \\ diag\left[\frac{\partial W_2}{\partial\left(\mathcal{L}_k \xi\right)}\right]\left[\mathcal{L}_k\right] & diag\left[\frac{\partial W_2}{\partial\left(\mathcal{L}_k \phi\right)}\right]\left[\mathcal{L}_k\right] & diag\left[\frac{\partial W_2}{\partial\left(\mathcal{L}_k p\right)}\right]\left[\mathcal{L}_k\right] \\ diag\left[\frac{\partial W_3}{\partial\left(\mathcal{L}_k \xi\right)}\right]\left[\mathcal{L}_k\right] & diag\left[\frac{\partial W_3}{\partial\left(\mathcal{L}_k \phi\right)}\right]\left[\mathcal{L}_k\right] & diag\left[\frac{\partial W_3}{\partial\left(\mathcal{L}_k p\right)}\right]\left[\mathcal{L}_k\right] \end{pmatrix} Q_2. \tag{14}$$

The diagonal matrices $diag[F]$ in (14) have diagonal entries $F(\mathbf{q}_{\#BCs+1}), \ldots, F(\mathbf{q}_N)$ (i.e. are collocated of the interior nodes of the pointset, where the PDEs apply). Formally, the sum includes all the derivatives up to second order, but note that many of them are zero. As an example, we list the nonzero Fréchet derivatives of $\mathcal{W}_3$ (check (2) for the rest):

$$\frac{\partial W_3}{\partial(\partial_\xi \dot{\xi})} = 1-\xi, \qquad \frac{\partial W_3}{\partial(\partial_\varphi \dot{\varphi})} = -1, \qquad \frac{\partial W_3}{\partial(\dot{\xi})} = -1, \qquad \frac{\partial W_3}{\partial(\xi)} = -\partial_\xi \dot{\xi}. \tag{15}$$

After collecting all of the surviving Fréchet derivatives, it turns out that the nodal matrices involved are I (the $N \times N$ identity), $[\partial^2_{\xi\xi}]$, $[\partial^2_{\varphi\varphi}]$, $[\partial_\xi]$ and $[\partial_\varphi]$. The sparsity pattern of J for the illustrative RBF-PUM discretization shown in Figure 3 is sketched in Figure 4.

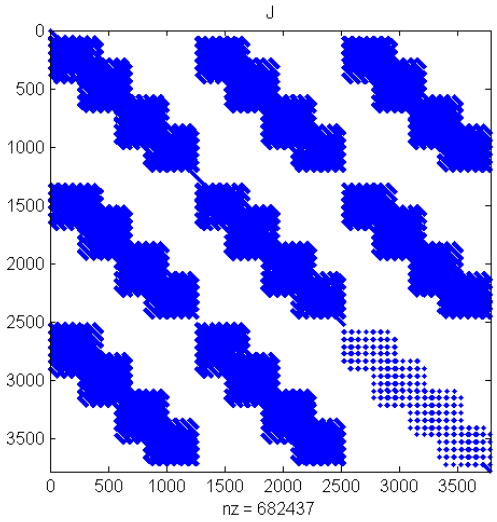

Fig. 4 Sparsity pattern of the Jacobian for the discretization in Figure 3.

The analytical Jacobian does not ensure convergence, and for highly nonlinear PDEs the Hessian is required [5]. In this paper, however, we shall not consider it. The *dogleg* method for solving the trust-region subproblem is chosen due to its better conditioning and because it is already implemented as an option in Matlab's `fsolve`. Since global convergence is missing, we use the flow solved at a smaller Re as an initial guess for the iterations. The attainable Re is ultimately limited by the quality of the interpolant, the condition number of J, and by the resolution of the wake region provided by the meshless discretization.

5 Preliminary results up to $Re = 60$

We illustrate the numerical method discussed in this paper with an example. It is preliminary because the discretization is small enough that the flow problem can be solved on a laptop, and because no effort has yet been made

to optimize the location of the collocation points, which is one of the most interesting features of meshless formulations, and a well known strategy to improve performance. Thus, there are $N = 1292$ nodes forming a grid in $[0,1]\times[0,\pi]$, corresponding to the locations in physical space shown on the left side of Figure 3 (the physical nodes for $\xi = 1$ lie at the infinity and are obviously not shown).

Figure 4 shows the sparsity pattern of the RBF-PUM Jacobian. We report the wake structure for growing Re between $Re = 0.1$ and $Re = 60$. As reported in [2], recirculation starts at about $Re = 40$—see Figure 5. After $Re = 60$, the discretization of the wake is inadequate and trust-region algorithm ceases to converge to a root, which is revealed by the fact that iterations stall at a value $\mathbf{X}_\infty$ for which $J(\mathbf{X}_\infty)$ is numerically singular (see [5] for details). In order to proceed further, a denser pointset with nodes more concentrated in the wake region is necessary, which in turn calls for a more powerful computer than a laptop; it is thus left as future work. It is, however, remarkable that $Re = 60$ can be attained with fewer than 1300 collocation nodes.

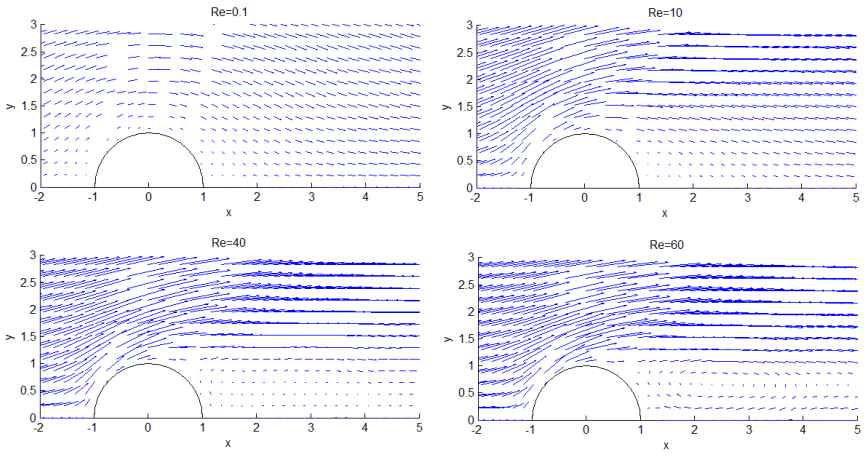

Fig. 5 Velocity field close to cylinder for increasing $Re = \{0.1, 10, 40, 60\}$.

6 Conclusions and future work

RBF-PUM is a promising spectral, meshless method that combines the flexibility and simplicity of RBF collocation, with the possibility of tackling much larger problems than before (in terms of the discretization size), thanks to the resulting sparse structure. RBF-PUM is currently being investigated along several directions. Tailored preconditioners [6] and parallel implementations

[7] have also been already proposed. In this paper, we have tested RBF-PUM on a benchmark problem in fluid dymamics. This is a challenging test due to the far-field BCs, as well as the numerical instability and nonlinearity compounding fast with growing *Re*. We have presented some numerical results from a preliminary, straightforward laptop implementation. By refining the discretization to the limit of the computational resources, we were able to solve the flow up to $Re \approx 60$, when the first eddies appear—qualitatively matching the expected flow pattern.

In order to proceed further, we plan to implement the method presented here on a parallel computer. We expect that by refining the discretization—adaptively if necessary—it will be possible to resolve the finer eddies appearing at higher *Re*. We also plan to enhance the convergence of the trust-region algorithm for the nonlinear system by incorporating Hessian information—a strategy which was deemed critical for highly nonlinear problems in [5]. With adequate computational resources, the related problem—albeit three-dimensional—of viscous flow past a sphere could be tackled with a very similar approach. Finally, by calculating and comparing the numerical drag coefficient with the experimental values over a wider range of *Re*, it will be possible to assess the convergence rate of the method for this problem.

Acknowledgements F. Bernal acknowledges support from FCT grant SFRH/BPD/79986/2011 and INESC-ID. A. Heryudono is partially supported by NSF Grant DMS 1552238.

References

1. Stokes, G.G.: Trans. Camb. Phil. Soc. 9 (1851).
2. Fornberg, B: Steady viscous flow past a circular cylinder up to Reynolds number 600. J. Comput. Phys. **61**, 297-320 (1985).
3. Mandujano F., Peralta-Fabi, R: On the viscous steady flow around a circular cylinder. Revista Mexicana de Física **51**(1), 8799 (2005).
4. Safdari-Vaighani, A., Heryudono A., Larsson, E.: A radial basis function partition of unity collocation method for convection-diffusion equations arising in financial applications. J. Sci. Comput. **64**, 341-367 (2015).
5. Bernal, F.: Trust-region methods for nonlinear elliptic equations with radial basis functions. Comput. Math. Appl. **72**(7), 1743-1763 (2016).
6. Heryudono, A., Larsson, E., Ramage, A., Von Sydow, L.: Preconditioning for Radial Basis Function Partition of Unity Methods. J. Sci. Comput **67**(3) 1089-1109 (2016).
7. Tominec, I.: Parallel localized radial basis function methods for the shallow water equations on the sphere. MSc thesis, Technische Universitaet Muenchen (2017).