AIAA 2009-3546

19th AIAA Computational Fluid Dynamics Conference, June 22–25, San Antonio, Texas

Panel-free boundary conditions for viscous vortex methods

Christopher D. Cooper *

Dept. of Mechanical Engineering, Universidad Técnica F. Santa María, Casilla 110-V, Valparaíso, Chile

L. A. Barba †

Department of Mathematics, University of Bristol, Bristol, United Kingdom BS8 1TW

The Lagrangian vortex particle method for solving the Navier-Stokes equations is essentially a meshfree method. For the satisfaction of the boundary conditions on a submerged body, however, the need to generate a mesh (on the boundary) remains, as the only known method is based on a panel-type formulation. Here, we propose an alternative method for satisfaction of the boundary conditions which is meshfree, and based on a radial basis function collocation approach. Radial basis functions are used both to calculate the vortex sheet on the body which cancels the slip velocity, and to solve the diffusion equation which transfers the vorticity from the solid wall to the fluid domain. Proof-of-concept calculations are presented on the impulsively-started cylinder at Reynolds numbers 200 and 1000. These results demonstrate the feasibility of the concept, and further investigations are underway to assess overall accuracy and improve computational efficiency.

Nomenclature

α	Coefficients of an RBF expansion of a function
β	Coefficients of RBF decomposition for solving the diffusion equation
δt	Time step length
$\mathbf{x}$	Spatial position
t	Time
$\boldsymbol{\omega}$	Vorticity
γ	Vortex sheet strength
Γ	Circulation
$\hat{n}$	Unit vector normal to the body surface
$\hat{s}$	Unit vector tangential to the body surface
$\mathbb{K}$	Biot-Savart kernel
$\mathbf{U}_\infty$	Free stream velocity
$\mathbf{u}$	Velocity
$\mathbf{u}_{slip}$	Slip velocity at boundary
ν	Viscosity
Ω	Rotational velocity of body
ϕ	Radial basis function
ψ	Stream function
ρ	Eigenfunction of the first term in the spectral decomposition of the kernel
σ	Core size of the particles, or shape parameter of the Gaussian radial basis function
A_B	Cross-sectional area of the body, perpendicular to the flow
L	Perimeter of body

*Undergraduate student.

†AIAA Member. Corresponding author. New address: Dept. of Mechanical Engineering, Boston University, Boston, MA 02215 USA +1-617-353-3883, labarba@bu.edu.

I. Introduction

ONE of the various ways to solve the Navier-Stokes equations is to use its velocity-vorticity formulation. This is found taking the curl of the momentum equation and considering an incompressible fluid for which $\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0$. Thus, the vorticity transport equation is obtained, which is the governing equation in vortex methods:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \mathbf{u} + \nu \Delta \boldsymbol{\omega} \tag{1}$$

To solve this equation, a Lagrangian method may be used. In fact, if we look at the 2D inviscid case, Eq. (1) turns into $\frac{D\omega}{Dt} = 0$, where $\frac{D}{Dt}$ stands for the material derivative, making a Lagrangian method seem ideal. In the vortex particle method, the domain is discretized into moving computational nodes, interpreted as vortex particles. The computational nodes are identified by: a position vector, $\mathbf{x}_i$; a strength vector of circulation (vorticity$\times$volume); and a core size, σ. The discretized vorticity field is expressed as the sum of the vorticities of the vortex particles in the following way:

$$\boldsymbol{\omega}(\mathbf{x}, t) \approx \boldsymbol{\omega}_\sigma(\mathbf{x}, t) = \sum_{i=1}^{N} \boldsymbol{\Gamma}_i(t) \zeta_\sigma \left(\mathbf{x} - \mathbf{x}_i(t)\right), \tag{2}$$

where $\boldsymbol{\Gamma}_i$ corresponds to the vector circulation strength of particle i (scalar in 2D). The characteristic distribution of vorticity at each node, ζ_σ , is often a Gaussian distribution, which for the two dimensional case can be written as:

$$\zeta_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-|\mathbf{x}|^2}{2\sigma^2}\right). \tag{3}$$

The vortex particles are assumed to convect without deformation with the local velocity, which is calculated by means of the Biot-Savart law:

$$\begin{aligned} \mathbf{u}(\mathbf{x}, t) &= \int (\nabla \times \mathbb{G})(\mathbf{x} - \mathbf{x}')\omega(\mathbf{x}', t) d\mathbf{x}' \\ &= \int \mathbb{K}(\mathbf{x} - \mathbf{x}')\omega(\mathbf{x}', t) d\mathbf{x}' = (\mathbb{K} * \omega)(\mathbf{x}, t) \end{aligned} \tag{4}$$

Here, $\mathbb{K} = \nabla \times \mathbb{G}$ is known as the Biot-Savart kernel, $\mathbb{G}$ is the Green's function for the Poisson equation, and $*$ represents convolution. However, as potential flow is irrotational, it cannot be represented with Eq. (4). When the velocity field has an irrotational component, it is added to Eq. (4) and we write:

$$\mathbf{u}(\mathbf{x}, t) = (\mathbb{K} * \omega)(\mathbf{x}, t) + \mathbf{U}_\infty \tag{5}$$

where $\mathbf{U}_\infty$ is the free stream velocity. It is worth noting that the potential flow added is not the solution of a homogeneous Laplacian equation considering the presence of a solid body, but only of the free-stream velocity.

Finally, the Lagrangian formulation of the (viscous) vortex method in two dimensions can be expressed in the following system of equations:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{u}(\mathbf{x}_i, t) = (\mathbb{K} * \omega)(\mathbf{x}_i, t) + \mathbf{U}_\infty \tag{6}$$

$$\frac{d\omega}{dt} = \nu \Delta \omega + \text{B.C.} \tag{7}$$

where B.C. stands for boundary conditions. This pair of equations is solved at every time step. In order to take the diffusion into account, several viscous scheme are available in vortex methods which solve Eq. (7) either approximately or exactly. The most popular ones are the particle strength exchange (PSE), random

vortex method and core spreading. More details of the various schemes can be found in the book by Cottet and Koumoutsakos.[1] In the present work, the core spreading viscous scheme has been used.

In this paper, we propose a new method to enforce the boundary conditions. The standard approach used today is based on a panel method, and was developed by Koumoutsakos, Leonard and Pépin;[2] we briefly explain their method in the next section. Next, we give a short introduction to radial basis function (RBF) methods, on which we base our approach to satisfy the boundary conditions. Finally, our algorithm will be presented, and proof-of-concept calculations on the impulsively-started cylinder. We end with a discussion of remaining areas of improvement, conclusions and future work.

II. Boundary conditions for particle methods

Formulating the boundary conditions on a solid wall is notoriously problematic in vortex methods. The difficulty arises due to the absence of a vorticity boundary condition for the Navier-Stokes equation, equivalent to no-slip at the wall. The problem has been addressed by means of a model of vorticity creation at the solid wall, as described below, of which the first attempt was proposed by Chorin.[3]

When a solid is immersed in a flow, its effect can be summarized in two expressions of the boundary conditions: the flow cannot go through the solid wall, and the tangential velocity of the flow on the wall is zero. Common terminology refers to these conditions as the no-through and the no-slip boundary conditions. They are expressed in the following equations, where $\hat{n}$ and $\hat{s}$ represent unit vectors normal and tangential to the boundary, respectively:

$$\mathbf{u} \cdot \hat{n} = \frac{\partial \psi}{\partial s} = 0 \tag{8}$$

$$\mathbf{u} \cdot \hat{s} = \frac{\partial \psi}{\partial n} = 0 \tag{9}$$

In a viscous flow, the presence of a solid boundary affects the flow by forcing the fluid to decelerate to zero velocity at the wall. This effect can be explained by creation of vorticity at the solid boundary. A flow approaches the solid body with a free stream velocity and no vorticity, then, after passing around the solid body, the fluid will drift away with a non-zero vorticity distribution in the field. In this sense we can say that the solid body is a *source of vorticity*. Nowhere is this mechanism more readily apparent than in the von Kármán vortex street behind a cylinder or similar object. The concept of vorticity generation at the solid surface is the basis for the formulations for satisfaction of boundary conditions in the vorticity equation, and it is based on the ideas of Prager[4] and Lighthill.[5]

The models of vorticity generation used in vortex methods, in order to create the needed vorticity, attach a vortex sheet on the surface of the body. The equation that defines the strength of a vortex sheet that produces a velocity discontinuity of $\mathbf{u}_{slip}$, is a Fredholm equation of the second kind, written as follows:

$$\gamma(s) - \frac{1}{\pi} \oint \frac{\partial}{\partial n} [\log | \, \mathbf{x}(s) - \mathbf{x}(s') \, |] \gamma(s') \mathrm{d}s' = 2 \, \mathbf{u}_{slip} \cdot \hat{s} \tag{10}$$

Here, $\gamma(s)$ is the strength of the vortex sheet, $\mathbf{u}_{slip}$ is the velocity discontinuity created by the vortex sheet, and $\hat{s}$ is a unitary vector tangent to the solid in the direction of integration. For more details of the derivation of this equation, see Ref. 2 and Ref. 6.

As described in Ref. 7 (p. 14), in the vortex method there is a direct relation between the no-through and the no-slip boundary conditions, called the *linked boundary conditions*. Briefly, it can be shown that satisfaction of the tangential velocity condition, for a boundary enclosed in the fluid domain, implies that the normal velocity condition is also satisfied for bodies whose flow representation is irrotational. In other words, if the no-slip boundary condition is satisfied, then the no-through boundary condition is automatically satisfied. For this reason, from now on we concentrate on enforcing only the no-slip boundary condition.

In the Lagrangian vortex method, after the inviscid substep, there is a spurious velocity on the surface of the body, whose tangential component has to be cancelled out. The usual way to cancel this tangential component of the velocity is by placing a vortex sheet on the surface of the boundary, having the strength to produce the same velocity as the spurious slip, but in the opposite direction. Thus, slip is cancelled within the time step.

After a vortex sheet on the boundary is obtained to cancel the slip velocity, one has to find a way to transfer the vorticity of the vortex sheet to the nearby particles in the fluid domain. This is accomplished solving a diffusion equation with the correct boundary conditions:

$$\begin{aligned} \frac{\partial \boldsymbol{\omega}}{\partial t} - \nu \Delta \boldsymbol{\omega} &= 0 \\ \boldsymbol{\omega}(t - \delta t) &= 0 \\ \nu \frac{\partial \boldsymbol{\omega}}{\partial n} &= \frac{-\gamma(s)}{\delta t} \end{aligned} \tag{11}$$

These steps have been formulated numerically before with success by Koumoutsakos, Leonard and Pépin.[2] The technique used in that work —and from then onwards in most if not all vortex methods with boundaries— is to solve the Fredholm equation (10), and the diffusion equation (11) by means of the well-known panel method. Even though this approach has led to some excellent results on bluff body flow using vortex methods,[8] we are interested in the possibility of alternative formulations which are panel-free. The panel method is a mesh-dependent method, and because of this, it encounters difficulties when trying to solve problems with complicated geometries, which may bring serious numerical instabilities to the method. We are interested in formulating a panel-free vortex method for bounded flows to avoid such problems, and the potentially time-consuming surface mesh-generation.

One problem with the Fredholm equation of the second type in Eq. (10) is that it is singular, as it admits a non-unique solution. Hence, an extra constraint has to be imposed, which can be found with Kelvin's circulation theorem and relating the vortex sheet strength with the change of circulation of the flow. This constraint is expressed in the following equation:

$$\oint \gamma(s) \mathrm{d}s = -2A_B[\Omega(t + \delta t) - \Omega(t)] = 0 \tag{12}$$

where A_B is the area of the body perpendicular to the free-stream velocity, and Ω is the rotational velocity of the body. For simplicity we consider a body which does not rotate, thus $\Omega = 0$.

When the body surface is discretized into M boundary elements (*i.e*, panels), an overdetermined system of equations is produced, with M unknowns and $M+1$ equations. To solve this system, Koumoutsakos and Leonard[9] present a scheme that gives a well-behaved system of equations for the case of panel methods, situation that we expect to be reproduced for the scheme that we present in this paper. The final equation that we will solve is:

$$\gamma(s) - \frac{1}{\pi} \oint [\frac{\partial}{\partial n}[\log \mid \mathbf{x}(s) - \mathbf{x}(s') \mid] - \frac{\rho_1(s)}{L}]\gamma(s')\mathrm{d}s' = 2\, \mathbf{u}_{slip} \cdot \hat{s} \tag{13}$$

Here, $\rho_1(s)$ is the eigenfunction of the first term in the spectral decomposition of the kernel, and L is the perimeter of the body.

When using a panel method, the above equation is discretized in the following way:

$$\gamma(s) - \frac{1}{\pi} \sum_{i=1}^{N} \int [\frac{\partial}{\partial n}[\log \mid \mathbf{x}(s) - \mathbf{x}(s') \mid] - \frac{\rho_1(s)}{L}]\gamma(s')\mathrm{d}s' = 2\, \mathbf{u}_{slip} \cdot \hat{s} \tag{14}$$

The transfer of vorticity can also be solved using a panel method, where the strength of each particle in the fluid domain is updated by

$$\Gamma_j^{n+1} = \Gamma_j^n + \nu_j \delta t \sum_{i=1}^{M} \frac{\gamma_i}{(1 - \kappa\sqrt{\pi\nu\frac{\delta t}{2}})} \phi(\mathbf{x}, \frac{\delta t}{2}) \tag{15}$$

where κ is the curvature, and

$$\phi(\mathbf{x},\tau) = \frac{\exp\left(\frac{-y^2}{4\nu\tau}\right)}{\sqrt{4\pi\nu\tau}}\left(\text{erf}\left[\frac{d+x}{\sqrt{4\nu\tau}}\right] + \text{erf}\left[\frac{d-x}{\sqrt{4\nu\tau}}\right]\right) \tag{16}$$

with d the size of the panel. For more details, see Ref. 2.

III. Radial basis function methods

An RBF representation is used to interpolate a function when it is known only on a set of scattered points in the domain. The interpolation is obtained by representing the unknown function as a linear superposition of basis functions, as follows:

$$f(\mathbf{x}) \approx \sum_{i=1}^{N} \phi(\mid \mathbf{x} - \mathbf{x}_i \mid)\alpha_i \tag{17}$$

Here, ϕ represents the basis function, the α_i are the coefficients of the linear combination, and the $\mathbf{x}_i$ are the centers of the data in the domain. The main ingredient of this method is the solution of a system of equations for the coefficients α_i, so that Eq. (17) gives the best representation of the function $f(\mathbf{x})$. This will be accomplished when the collocation conditions are satisfied,

$$f(\mathbf{x}_j) = \sum_{i=1}^{N} \phi(\mid \mathbf{x}_j - \mathbf{x}_i \mid)\alpha_i \tag{18}$$

for the RBF centres $\mathbf{x}_i$ and the evaluation points $\mathbf{x}_j$, which here are the same. Applying Eq. (18) to every $\mathbf{x}_j$, a linear system of equations is formed, with α_i as unknowns.

There are many popular basis functions that can be used for RBF interpolation, such as multiquadrics (MQ), thin-plate splines, cubic splines or Gaussians. In this paper, a Gaussian basis function is used. The reasons for this choice are that Gaussians are easy functions to deal with, without complicated derivatives, and also it has the property of unconditional positive definitiveness. This saves some work, as otherwise a polynomial would have to be added to Eq. (17) to ensure the solvability of the system; for more details see Ref. 10. The centers of the Gaussian basis functions will be $\mathbf{x}_i$ and so we have:

$$\phi(\mathbf{x} - \mathbf{x}_i) = \exp\left(\frac{-\mid \mathbf{x} - \mathbf{x}_i \mid^2}{2\sigma^2}\right) \tag{19}$$

As was shown by Kansa,[10] RBF function representations can be used as a method in order to solve PDE's and also integral equations. An RBF expansion involves the linear combination of a function that depends only on the position of the centres, where the function is known. The only unknowns are the coefficients, which can be taken out of a differential or integral operator, for example:

$$\begin{aligned}
\nabla f &\approx \sum_{i=1}^{N} \alpha_i \nabla\phi(\mid \mathbf{x} - \mathbf{x}_i \mid) \\
\Delta f &\approx \sum_{i=1}^{N} \alpha_i \Delta\phi(\mid \mathbf{x} - \mathbf{x}_i \mid) \\
\int f \mathrm{d}x &\approx \sum_{i=1}^{N} \alpha_i \int \phi(\mid \mathbf{x} - \mathbf{x}_i \mid)
\end{aligned} \tag{20}$$

Applying the expressions in Eq. (20), at the computational nodes, to the corresponding differential or integral equation, a linear system of equations is formed from which the values of α_i are computed. Then, using Eq. (17), the actual values of the function at the nodes are finally obtained.

IV. Solving vortex boundary conditions with RBFs

A full vortex method algorithm can be summarized in the diagram given on Figure 1. The first step is the discretization of an existing initial vorticity field or, if there is no initial vorticity, of a small area around the solid body. At each time step, the velocity of the vortex particles is evaluated, then the particles are convected, and the vorticity diffused. In a fractional step formulation, boundary conditions are satisfied next, to repeat the whole process again from the velocity evaluation. Every few time steps, spatial adaptation of the particles is applied to ensure that particle overlap is maintained.

For the unbounded vortex method, all of the algorithmic steps have already been formulated with a meshfree approach; only the satisfaction of boundary conditions has remained tied to some form of mesh (surface panels). The key of the new meshfree approach presented in this paper is in the enforcement of the boundary conditions without panels. To accomplish this, the formulation is based on a radial basis function (RBF) interpolation approach, used both to calculate the strength of the vortex sheet on the body needed to cancel the slip velocity, and to diffuse the vorticity into the fluid domain.

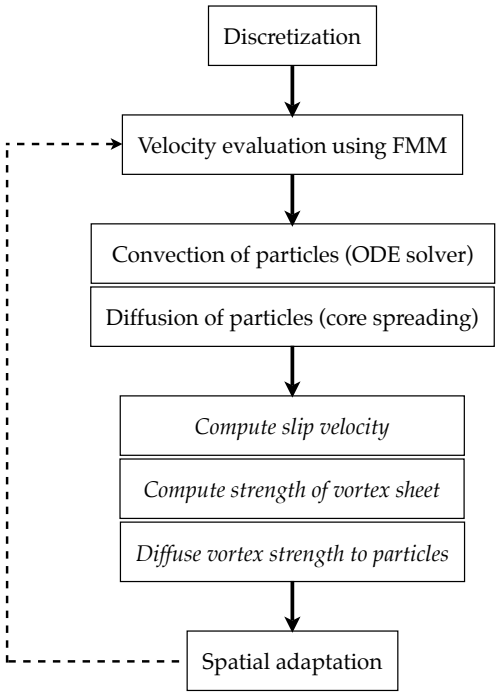


Figure 1. Flow diagram of a full vortex method algorithm, for bounded viscous flow. The satisfaction of the boundary conditions has been expressed in three sub-steps, as described in the text. Spatial adaptation is applied as needed, to preserve discretization accuracy.

IV.A. Finding the vortex sheet strength

In the conventional approach to providing solid-body boundary conditions in vortex methods, the surface is discretized by means of panels; in 3D, this usually calls for a surface triangulation algorithm. In the new method presented here, the surface is represented only with nodes or points. These will constitute the RBF centers for the computation of the vortex sheet strength, γ. The distribution of centers around the solid surface has to be such that all features of the body are well covered by points; one could place higher densities of points where there are more detailed geometric features. For the purposes of this proof-of-concept, involving the flow around a circular cylinder, it is sufficient to establish a uniform distribution of points along the surface.

The first step in the enforcement of the boundary conditions is to compute the slip velocity on the surface of the boundary, which was caused by the previous inviscid time sub-step. This value is needed at the position of the boundary nodes, and is obtained using the Biot-Savart law, as for any particle.

Having the slip velocity, it is then possible to obtain γ from Eq. (13). The value of ρ can also be solved with an RBF approach, using the same principles that will be applied when solving the Fredholm equation of the second kind, which will be described in detail below. For more details on ρ, see Ref. 11.

Let the RBF approximation of the function $\gamma(s)$ around the surface of the solid body be as follows:

$$\gamma(\mathbf{x}) \approx \sum_{i=1}^{N} \phi(\mid \mathbf{x} - \mathbf{x_i} \mid)\alpha_i \tag{21}$$

Now, equation Eq. (13) can be written in the following way

$$\sum_{i=1}^{N} \phi(\mid \mathbf{x} - \mathbf{x}_i \mid)\alpha_i - \oint \frac{1}{\pi}\frac{\partial}{\partial n}[\log \mid \mathbf{x} - \mathbf{x}' \mid] \cdot \sum_{i=1}^{N} \alpha_i \phi(\mid \mathbf{x}' - \mathbf{x}_i \mid)\mathrm{d}\mathbf{x}'$$
$$+ \oint \frac{\rho_1(x)}{L} \cdot \sum \alpha_i \phi(\mid \mathbf{x}' - \mathbf{x}_i \mid)\mathrm{d}\mathbf{x}' = \mathbf{u}_{slip} \cdot \hat{s} \tag{22}$$

Let

$$F = \frac{1}{\pi}\frac{\partial}{\partial n}[\log \mid \mathbf{x}-\mathbf{x}' \mid]\phi(\mid \mathbf{x}'-\mathbf{x}_i) \tag{23}$$

$$G = \frac{\rho_1}{L}\phi(\mid \mathbf{x}'-\mathbf{x}_i \mid) \tag{24}$$

The integral and the sum operators are commuted, leaving the coefficients α_i, our unknowns, outside of the integral:

$$\sum_{i=1}^{N}\phi(\mid \mathbf{x}-\mathbf{x}_i \mid)\alpha_i - \sum_{i=1}^{N}\alpha_i \oint F\mathrm{d}\mathbf{x}' + \sum_{i=1}^{N}\alpha_i \oint G\mathrm{d}\mathbf{x}' = \mathbf{u}_{slip}\cdot\hat{s} \tag{25}$$

The integrals can be solved numerically with any kind of quadrature, depending on the needed precision. We are left with a linear system of equations, which can be written in matrix form as follows:

$$\begin{pmatrix} \phi_{11}-\Theta_{11}+\Lambda_{11} & \phi_{12}-\Theta_{12}+\Lambda_{12} & \cdots & \phi_{1N}-\Theta_{1N}+\Lambda_{1N} \\ \phi_{21}-\Theta_{21}+\Lambda_{21} & \phi_{22}-\Theta_{22}+\Lambda_{22} & \cdots & \phi_{2N}-\Theta_{2N}+\Lambda_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N1}-\Theta_{N1}+\Lambda_{N1} & \phi_{N2}-\Theta_{N2}+\Lambda_{N2} & \cdots & \phi_{NN}-\Theta_{NN}+\Lambda_{NN} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} u^1_{slip}\cdot s_1 \\ u^2_{slip}\cdot s_2 \\ \vdots \\ u^N_{slip}\cdot s_N \end{pmatrix}$$

or, in a simplified way, as

$$[\phi_{ki}-\Theta_{ki}+\Lambda_{ki}]\cdot[\alpha_i] = [u^i_{slip}\cdot s_i] \tag{26}$$

where $\mathbf{x}=\mathbf{x}_k$ and

$$\Theta = \oint F\mathrm{d}\mathbf{x}' \tag{27}$$

$$\Lambda = \oint G\mathrm{d}\mathbf{x}' \tag{28}$$

Solving this linear system of equations, we find the coefficients of the RBF representation that best fits the function for the strength of the vortex sheet around the surface of the body, such that the slip velocity is cancelled.

IV.B. Diffusion of vortex sheet strength

After the strength of the vortex sheet on the body is found, such that it cancels the slip velocity, it is necessary to diffuse the vorticity that was created on the surface to the surrounding fluid. This step is based on the solution of the PDE problem described by Eq. (11).

The solution of this diffusion equation will be the amount of vorticity that has to be added to the nearby particles in the fluid domain, in order to have zero velocity on the boundary surface. The vorticity will be added in terms of a change in the strength of the particles, as will be shown later. We solve this PDE, once again, with an RBF approach. An RBF approximation of the vorticity for the whole field can be written as,

$$\omega^t(x) = \sum_{i=1}^{N}\beta^t_i\phi(\mid \mathbf{x}-\mathbf{x}_i \mid) \tag{29}$$

Taking the unknown coefficients outside of the differential operator, we have:

$$\frac{\partial\omega}{\partial n} = \nabla\omega\cdot n = \sum_{i=1}^{N}\beta_i\nabla\phi(\mid \mathbf{x}-\mathbf{x}_i \mid)\cdot n$$

$$\Delta\omega = \sum_{i=1}^{N}\beta_i\Delta\phi(\mid \mathbf{x}-\mathbf{x}_i \mid) \tag{30}$$

A numerical scheme to solve the diffusion equation with an RBF method is described in Ref. 12, where the time differential operator is treated with a Crank-Nicholson scheme, which is second order accurate. The diffusion equation is discretized in the following way:

$$\frac{\omega^t - \omega^{t-\delta t}}{\delta t} = \nu(\theta\Delta\omega^t + (1-\theta)\Delta\omega^{t-\delta t}) \tag{31}$$

where θ can be any value between 0 and 1. Here $\theta = 0.5$ is used, a common choice used in the Crank-Nicholson scheme. Grouping the terms according to their position in time, we get:

$$\omega^t - \delta t\nu\theta\Delta\omega^t = \omega^{t-\delta t} + \delta t\nu(1-\theta)\Delta\omega^{t-\delta t} \tag{32}$$

Applying Eq. (29) into Eq. (32), the following expression is obtained:

$$\sum_{i=1}^{N}\beta_i^t\cdot(\phi(\mid \mathbf{x}-\mathbf{x}_i\mid) - \delta t\nu\theta\Delta\phi(\mid \mathbf{x}-\mathbf{x}_i\mid) = \sum_{i=1}^{N}\beta_i^{t-\delta t}\cdot(\phi(\mid \mathbf{x}+\mathbf{x}_i\mid) + \delta t\nu(1-\theta)\Delta\phi(\mid \mathbf{x}-\mathbf{x}_i\mid) \tag{33}$$

which can be written in matrix form when applied to every node, as follows:

$$[\phi - \delta t\nu\theta\Delta\phi][\beta^t] = [\phi + \delta t\nu(1-\theta)\Delta\phi][\beta^{t-\delta t}] \tag{34}$$

For nodes that are located on the boundaries, the Neumann boundary condition shown in Eq. (11) has to be enforced, and following Eq. (30):

$$\sum_{i=0}^{N}\beta_i^t\nabla\phi(\mid \mathbf{x}-\mathbf{x}_i\mid)\cdot n = \frac{-\gamma^t(s)}{\nu\delta t} \tag{35}$$

The value of $\beta^{t-\delta t}$ can be found using the initial condition of the diffusion equation being solved. Since this is zero everywhere, using Eq. (29) applied to every domain node, we solve the linear system of equations given, obtaining that $[\beta^{t-\delta t}]$ is a column matrix full of zeroes. Hence, the right hand side of Eq. (33), will be zero for the domain nodes, and $\frac{-\gamma^t(s)}{\nu\delta t}$ for the boundary nodes. Finally, the linear system of equations that will give the coefficients providing the best RBF representation of the diffused vorticity field is:

$$\begin{pmatrix} \nabla\phi_{11}\cdot n_1 & \nabla\phi_{12}\cdot n_1 & \cdots & \nabla\phi_{1M}\cdot n_1 \\ \vdots & \vdots & \ddots & \vdots \\ \nabla_{N1}\cdot n_N & \nabla_{N2}\cdot n_N & \cdots & \nabla_{NM}\cdot n_N \\ \phi_{(N+1)1} - \delta t\nu\theta\Delta\phi_{(N+1)1} & \phi_{(N+1)2} - \delta t\nu\theta\Delta\phi_{(N+1)2} & \cdots & \phi_{(N+1)M} - \delta t\nu\theta\Delta\phi_{(N+1)M} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{M1} - \delta t\nu\theta\Delta\phi_{M1} & \phi_{M2} - \delta t\nu\theta\Delta\phi_{M2} & \cdots & \phi_{MM} - \delta t\nu\theta\Delta\phi_{MM} \end{pmatrix} \cdot \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_N \\ \beta_{N+1} \\ \vdots \\ \beta_M \end{pmatrix} = \begin{pmatrix} \frac{\gamma_1}{\nu\delta t} \\ \vdots \\ \frac{\gamma_N}{\nu\delta t} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{36}$$

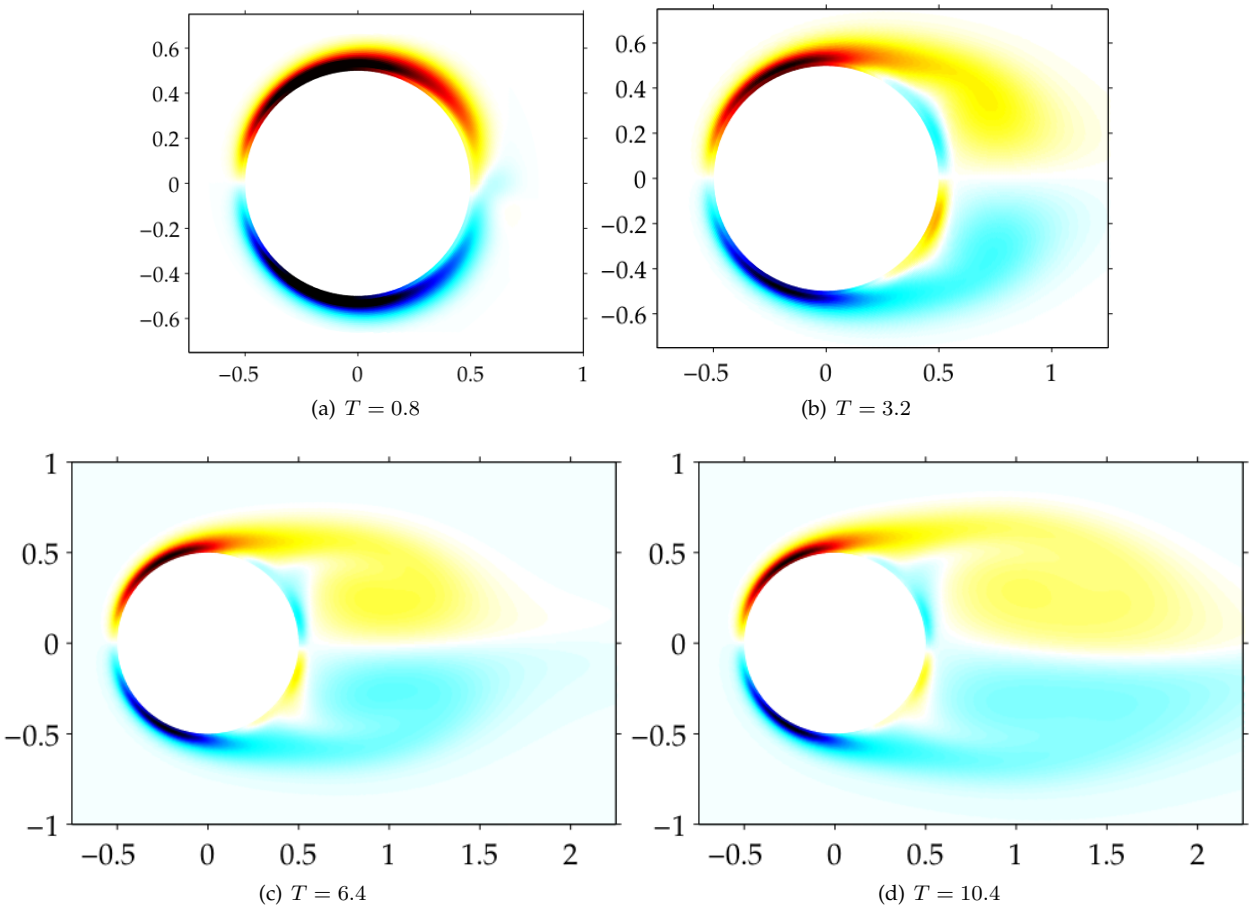
Here, we have M nodes in the whole domain, from which the first N nodes are located on the boundary (hence, $M-N$ is the number of nearby particles considered for diffusion).

Solving this system, the coefficients of the RBF expansion in Eq. (29) are obtained. Then, using Eq. (29), the vorticity field diffused from the vortex sheet is computed.

The position of the domain nodes for this RBF scheme are the positions of the existing vortex particles. However, the core size of the basis function, which is a Gaussian, does not necessarily have to be the same as the core size of the particles. This is an advantage, since the width of the basis functions, being a very important factor in the accuracy of any RBF method, can be chosen at will.

V. Proof-of-concept calculations

For bluff-body flows, the classic benchmark problem is the flow around a circular cylinder. In this case, we have tested our method with the impulsively started cylinder, using Reynolds number $Re = 200$ and $Re = 1000$. Preliminary results are shown in the figures below, consisting of a visualization of the vorticity field for the starting flow. Time is non-dimensionalized as $T = \frac{tU}{R}$, where t is the actual time, U the free-stream velocity and R the cylinder radius. As can be seen, the expected vorticity distribution for starting cylinder flow is obtained, with secondary vorticity of opposite sign appearing in the wake area.



(a) $T = 0.8$ (b) $T = 3.2$

(c) $T = 6.4$ (d) $T = 10.4$

Figure 2. Vorticity field for times indicated in a proof-of-concept calculation with $Re = 200$. Figure 2(d) has nearly 550,000 particles.

To perform these simulations, we placed particles on nodes of a polar lattice, with a separation of $h = 0.007$ for the $Re = 200$ case and $h = 0.006$ for the $Re = 1000$ case, always with an overlap ratio $\frac{h}{\sigma}$ of 1. The advection step was performed using an Adams-Bashforth second-order method with a time-step of $\delta t = 0.01$. For the first time-step and after every step of spatial adaptation, an Euler method was used with time step $\delta t = 0.002$, so as to maintain a good accuracy. A spatial adaptation processing of the particles was performed every 4 time-steps, using an RBF interpolation as developed in Ref. 13. The diameter of the cylinder and the value of free stream velocity were both 1. Simulations began with around 4,500 particles, and the last frames shown in Figure 2 and Figure 3 have reached, respectively, approximately 550,000 and 290,000 particles. The large numbers of particles in these runs are not optimal, as domain population control for the particles has not been carefully applied. This and other optimizations are under development.

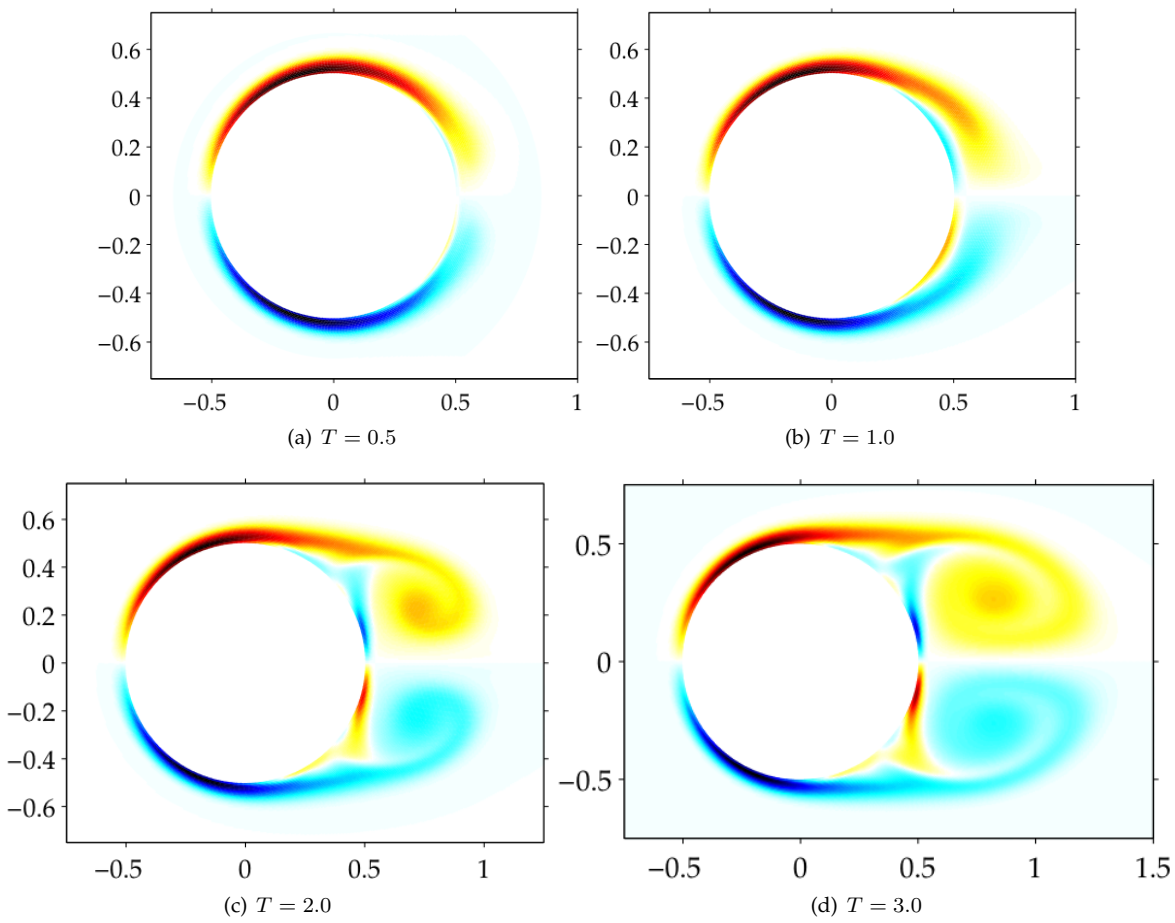


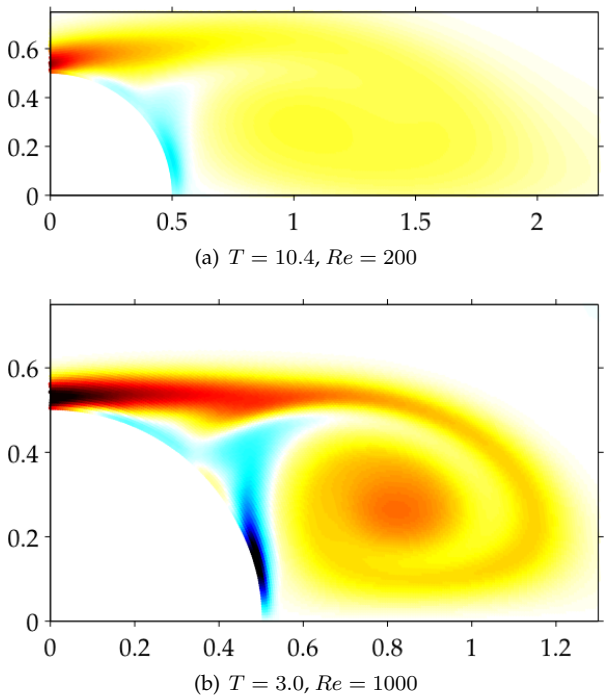
Figure 3. Vorticity field for times indicated in a proof-of-concept calculation with $Re = 1000$. Figure 3(d) has more than 250,000 particles.

VI. Implementation details

These proof-of-concept calculations were obtained using a completely serial code developed in Python, with bindings to C^{++} modules for the most computationally-intensive components: the vorticity evaluation and the linear system solver used in the RBF methods.

The linear systems for the computation of the particle circulations were solved using an $\mathcal{O}(N)$ method described in Ref. 14, which is able to handle a large number of particles. The velocity calculations were accelerated using an $\mathcal{O}(N)$ implementation of the fast multipole method (FMM), used in Ref. 15 and available online[a]. Our current implementation also involves an $\mathcal{O}(N)$ algorithm to calculate the vorticity field. As Gaussians decay fast, the far-field effect may be neglected, considering only the near-field contributions. The size of the near field is dependent on σ.

The main bottleneck of this implementation is found when solving the diffusion of the vortex sheet. As our code does not include a matrix-free solver for this stage, memory becomes an issue when considering too many nearby particles to account for diffusion. For this reason, we are not able to run this non-optimized code for higher Reynolds numbers, that require a higher resolution. We are currently working on improved implementations which will allow larger problem sizes.



(a) $T = 10.4$, $Re = 200$

(b) $T = 3.0$, $Re = 1000$

Figure 4. Zoom-in to the vorticity field in the wake, with a slightly enhanced color map.

The calculations shown, which reach numbers or particles up to 550,000, were carried out using one core of a laptop computer (2.1GHz Intel Core 2 Duo processor in a MacBook).

VII. Conclusions

Motivated by the goal of producing a purely mesh-free method for the simulation of unsteady flows using vortex methods, we have developed a panel-free formulation for the enforcement of the boundary conditions. The new formulation is based on a radial basis function (RBF) collocation method, taking advantage of the use of RBF representations applied to integral equations and PDEs. As a proof-of-concept, we solved a classic benchmark problem: the impulsively started circular cylinder, with Reynolds numbers 200 and 1000. Apart from avoiding generating a mesh and the numerical instabilities related to the panels, the implementation described in this work is relatively easy to understand and code.

We have presented here some preliminary results, which were produced with a prototype code developed in Python. As these were run on desktop or laptop computers, the results take some considerable amount of time. We are now working on increasing the efficiency of the algorithm components, which would allow us to make calculations with higher Reynolds numbers (requiring smaller particle sizes, and thus larger N) and to longer times. In particular, a matrix-free implementation of the vortex sheet diffusion would allow us to work with higher resolution, and a scheme for particle population control would help keep problem sizes more manageable. Once these improvements have been implemented and tested, we will develop a full implementation in parallel, capable of simulating physically interesting unsteady flow problems.

For a production-level implementation, we count on major software library components that are currently being developed in our research group. The fast calculation of the velocity of N vortex particles using the fast multipole method has already been implemented in parallel, and released as an open source

[a] http://code.google.com/p/pyfmm/

library[b]. The $\mathcal{O}(N)$ solution of an RBF interpolation problem with Gaussians of small spread has been implemented in Python,[14] but the parallel implementation is underway. Therefore, with the new mesh-free formulation of the boundary conditions presented here, we will soon have capability for simulation of highly unsteady vortical flows of physical interest, with a fully mesh-free method.

ACKNOWLEDGEMENTS. The authors would like to thank Felipe A. Cruz for help with the FMM code in Python and Rio Yokota for help with the RBF solver. CDC acknowledges support from the SCAT project via EuropeAid contract II-0537-FC-FA, see www.scat-alfa.eu, for an extended research visit to Bristol during which this work was carried out. LAB acknowledges support from EPSRC under grant contract EP/E033083/1 and additional travel support from Boston University College of Engineering.

References

[1]Cottet, G.-H., Koumoutsakos, P., and Ould Salihi, M. L., "Vortex methods with spatially varying cores," *J. Comp. Phys.*, Vol. 162, 2000, pp. 164–185.

[2]Koumoutsakos, P., Leonard, A., and Pépin, F., "Boundary Conditions for Viscous Vortex Methods," *J. Comp. Phys.*, Vol. 113, 1994, pp. 52–61.

[3]Chorin, A. J., "Numerical study of slightly viscous flow," *J. Fluid Mech.*, Vol. 57, 1973, pp. 785–796.

[4]Präger, W., "Die Druckverteilung an Körpern in ebener Potential strömung," *Physik. Zeitschr.*, Vol. XXIX, 1928, pp. 865.

[5]Lighthill, M. J., "Introduction: Boundary Layer Theory," *Laminar Boundary Layers*, edited by L. Rosenhead, Oxford University Press, 1963, pp. 46–113.

[6]Cottet, G.-H. and Koumoutsakos, P., *Vortex Methods. Theory and Practice*, Cambridge University Press, 2000.

[7]Shiels, D., *Simulation of Controlled Bluff Body Flow with a Viscous Vortex Method*, PhD thesis, California Institute of Technology, 1998.

[8]Ploumhans, P., Winckelmans, G. S., Salmon, J. K., Leonard, A., and Warren, M. S., "Vortex methods for direct numerical simulation of three-dimensional bluff body flows: Application to the Sphere at Re=300, 500 and 1000." *J. Comp. Phys.*, Vol. 178, 2002, pp. 427–463.

[9]Koumoutsakos, P. and Leonard, A., "High-resolution simulations of the flow around an impulsively started cylinder using vortex methods," *J. Fluid Mech.*, Vol. 296, 1995, pp. 1–38.

[10]Kansa, E. J., "Multiquadrics —A Scattered data approximation scheme with applications to Computational Fluid-Dynamics, II. Solutions to parabolic, hyperbolic and elliptic partial differential equations," *Computers Math. Applic.*, Vol. 19, No. 8/9, 1990, pp. 147–161.

[11]Koumoutsakos, P. and Leonard, A., "Improved boundary integral method for inviscid boundary condition applications," *AIAA J.*, Vol. 31, No. 2, 1993, pp. 401–404.

[12]Zerroukat, M., Power, H., and Chen, C. S., "A numerical method for heat transfer problems using collocation and radial basis functions," *Int. J. Numer. Meth. Engng.*, Vol. 42, No. 7, August 1998, pp. 1263 – 1278.

[13]Barba, L. A., Leonard, A., and Allen, C. B., "Advances in viscous vortex methods — meshless spatial adaption based on radial basis function interpolation," *Int. J. Num. Meth. Fluids*, Vol. 47, No. 5, 2005, pp. 387–421.

[14]Torres, C. E. and Barba, L. A., "Fast radial basis function interpolation with Gaussians by localization and iteration," *J. Comp. Phys.*, Vol. 228, 2009, pp. 4976–4999.

[15]Cruz, F. A. and Barba, L. A., "Characterization of the accuracy of the Fast Multipole Method in particle simulations," *Int. J. Num. Meth. Eng.*, 2009, Published online, doi:10.1002/nme.2611.

[b]Access code and preprints at http://barbagroup.bu.edu/