

Modeling Inviscid Flow with the Point-Vortex Method

Amélie Chiasson David¹

Directed Reading Program

¹*Department of Physics, McGill University*

June 22, 2025



Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 3 |
| 2 | Preliminaries | 4 |
| 2.1 | The Incompressible Euler Equations | 4 |
| 2.2 | Vorticity | 8 |
| 3 | Point-Vortex Method | 8 |
| 3.1 | Derivation | 9 |
| 3.2 | Convergence to Euler | 11 |
| 3.3 | Boundary Conditions | 11 |
| 4 | Computational Methods | 14 |
| 4.1 | Numerical Integration: RK4 | 14 |
| 4.2 | Solving for the Point-Vortex Velocity | 16 |
| 4.3 | Point-Vortices in a Circle | 17 |
| 4.4 | Dual Vortex Initial Condition | 18 |
| 4.5 | Lamb-Chaplygin Dipole in a Cylindrical Boundary | 20 |
| 4.6 | Future Improvements: Fast Multipole Method | 23 |
| 5 | Conclusion | 23 |
| 6 | Acknowledgments | 23 |

1 Introduction

The Euler equations are among the most well-known systems of partial differential equations (PDEs), governing the dynamics of inviscid flows—fluids idealized as having no internal friction. These equations play a fundamental role in modeling a wide range of physical phenomena, from atmospheric dynamics to aerodynamic simulations, and are central to many areas of applied mathematics and natural science.

However, solving the Euler equations numerically presents significant challenges due to their nonlinear and often singular behavior. To manage this complexity, numerical analysts frequently employ discretization techniques that approximate the continuous system while preserving essential physical properties. One such approach is the Point-Vortex Method, which models the fluid as a collection of n discrete vortices. This transforms the original PDE into a system of coupled ordinary differential equations (ODEs), offering a more tractable and computationally efficient framework for studying fluid dynamics.

This exploratory paper presents an overview of the Point-Vortex Method in 2D, from derivation to applied computation. Section 2 covers the preliminaries, offering an intuitive derivation of the incompressible Euler equations, which are the version of the equations that will be utilized in this report. It also introduces the concept of vorticity, an essential tool in dealing with the Point-Vortex Method. Section 3 provides a derivation of this method, as well as a discussion on the convergence to the actual Euler equations, which turns out to be in the weak sense. Boundary conditions are detailed at the end of this section via the infamous Green's functions. Finally, Section 4 goes into the nitty gritty of the computation, starting from the entirety of \mathbb{R}^2 to a simple closed cylindrical domain. Interesting initial conditions, such as a dual vortex system and the Lamb-Chaplygin dipole, are also presented. This section is concluded with a short discussion on how to reduce the computational cost of Point-Vortex Methods via Fast Multipole Methods.

This paper also aims to serve as a user-friendly introduction for those beginning to explore numerical methods for PDEs, offering simplified explanations and computational techniques through the study of an equation fundamental to our physical world.

Keywords: Fluid, incompressible Euler equations, vorticity, Runge-Kutta Method, Point-Vortex Method, Vortex, Green's Functions, Lamb-Chaplygin dipole, Fast Multipole Method.

2 Preliminaries

2.1 The Incompressible Euler Equations

The Euler equations are used to study the dynamics of a fluid, assuming no viscosity or thermal conductivity is in play. In this paper, we will be looking at the incompressible case, which can be defined as follows: For a homogeneous fluid, the incompressible flows are solutions of the first-order system of PDEs

$$\begin{cases} \mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p \\ \operatorname{div} \mathbf{u} = 0 \end{cases} \quad (1)$$

where $\mathbf{u}(\mathbf{x}, t) = (u_1, \dots, u_N)$ is the velocity field, u_t is the time derivative, and $p(x, t)$ is a scalar pressure due to there being an isotropic normal force exerted by the fluid at every point. We have also assumed that there is no body force such as gravity.

For clarity, the divergence operator is defined as

$$\operatorname{div} \mathbf{u} = \sum_{j=1}^N \frac{\partial u_j}{\partial x_j},$$

the gradient operator as

$$\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_N} \right)^T,$$

and the Laplacian operator as

$$\Delta = \sum_{j=1}^N \frac{\partial^2}{\partial x_j^2}.$$

Derivation: To derive the incompressible Euler equations, we follow closely from the derivation provided in *Partial Differential Equations: A First Course* by Prof. Rustum Choksi [Cho22]. This will be done in two dimensions, i.e., $\mathbf{u} = (u_1(x, y, t), u_2(x, y, t))$. We introduce $\rho(x, y, t)$, which is the mass density with units of mass per length squared. The derivation will arise from two physical systems: (1) conservation of mass and (2) conservation of linear momentum, which is Newton's second law.

(1) Conservation of Mass. Let $\Omega \in \mathbb{R}^2$ be our domain and $W \subseteq \Omega$ be where the fluid is located. The mass of the fluid at time t is given by

$$m_W(t) := \iint_W \rho(x, y, t) dx dy.$$

Assuming enough smoothness for $\rho(x, y, t)$ we take a time derivative inside the integrals

$$\frac{dm_W(t)}{dt} = \iint_W \rho_t(x, y, t) dx dy.$$

Now, notice that the mass of the fluid can change in time t by either entering or leaving through the boundary ∂W . This change in mass translates to the following equivalence

$$\iint_W \rho_t(x, y, t) dx dy = - \int_{\partial W} \rho \mathbf{u} \cdot \mathbf{n} dL$$

where the right-hand side is the net amount of fluid leaving by the boundary since $\rho \mathbf{u} \cdot \mathbf{n}$ is the fluid flux and \mathbf{n} denotes the outer unit normal. dL is the differential since we are performing a line integral over the closed line boundary. Now, by the Divergence Theorem, we have

$$\begin{aligned} \int_{\partial W} \rho \mathbf{u} \cdot \mathbf{n} dL &= \iint_W \operatorname{div}(\rho \mathbf{u}) dx dy \\ \implies \iint_W (\rho(x, y, t) + \operatorname{div}(\rho \mathbf{u})) dx dy &= 0 \end{aligned}$$

The above is true for any piece W . Since dx and dy have not been specified, they are arbitrary¹ and hence we can finally write

$$\boxed{\rho_t + \operatorname{div}(\rho \mathbf{u}) = 0} \quad (2)$$

Equation 2 is what we know as a continuity equation (or transport equation), which describes the transport of some quantity. Here, that would be the density of the fluid in time. As $\mathbf{u}(x, y, t)$ is unknown, we must find it in what follows.

(2) Conservation of Linear Momentum. Recall that Newton's Second Law states that the rate of change of linear momentum must be balanced by the net forces. To compute the rate of change with respect to time, we must consider not only a time derivative but the **material derivative**². Consider again the part of the fluid in W at time t . Similarly to part (1), we look at the total rate of change of linear momentum in the x direction, which is given by

$$\frac{d}{dt} \iint_W \rho u_1 dx dy = \iint_W \frac{\partial(\rho u_1)}{\partial t} dx dy + \int_W (\rho u_1) \mathbf{u} \cdot \mathbf{n} dL$$

where we have taken into account the changes in the linear momentum due to fluid entering and escaping through ∂W with the second term on the right-hand side. Applying the Divergence Theorem yields

¹Or more rigorously by the IPW Theorem (See [Cho22]).

²For those who are not familiar, this basically arises from vector calculus when taking a total derivative of a vector-valued function. Let $\mathbf{A}(t)$ be the acceleration vector at time t and $\mathbf{a}(x, y, t)$ be the spatial acceleration. Then $\mathbf{A}(t) = \mathbf{a}(x(t), y(t), t)$ and similarly for velocity $\mathbf{U}(t) = \mathbf{u}(x(t), y(t), t)$. We can finally deduce that $\mathbf{A}(t) = \frac{d\mathbf{U}(t)}{dt} = \dots = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}$.

$$\iint_W \left(\frac{\partial(\rho u_1)}{\partial t} + \operatorname{div}(\rho u_1 \mathbf{u}) \right) dx dy \quad (3)$$

An analogous argument can be made for the y direction, where we would use u_2 instead of u_1 . Using classic $F = ma$, we must balance the above with the total net force in the x direction acting on W . For the fluids that we study here, we can either have **body/external** forces such as gravity or **internal forces (stress)** such as pressure. In this case, we will only consider the latter as we will remain in the 'ideal fluid' regime.

Let $p(x, y, t)$ be the scalar function representing the force per unit length. The net pressure force on W through the boundary ∂W is given by

$$\mathbf{F}_W := - \int_{\partial W} p \mathbf{u} dL. \quad (4)$$

The Divergence Theorem component-wise on Equation 4 yields

$$\mathbf{F}_W = - \int_{\partial W} p \mathbf{u} dL = - \iint_W \nabla p dx dy.$$

Looking at only the x component, we have

$$\mathbf{F}_W^x = - \iint_W \frac{\partial p}{\partial x} dx dy. \quad (5)$$

We now balance Equation 5 with the total rate of change in the linear momentum that was found prior (3), which gives

$$\iint_W \left(\frac{\partial(\rho u_1)}{\partial t} + \operatorname{div}(\rho u_1 \mathbf{u}) \right) dx dy = - \iint_W \frac{\partial p}{\partial x} dx dy$$

Since dx and dy are not specified, they are arbitrary, hence, the integrands must be equal, which means:

$$\begin{aligned} \frac{\partial(\rho u_1)}{\partial t} + \operatorname{div}(\rho u_1 \mathbf{u}) &= - \frac{\partial p}{\partial x} \\ \rho \left(\frac{\partial u_1}{\partial t} + \mathbf{u} \cdot \nabla u_1 \right) + u_1 \left(\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{u}) \right) &= - \frac{\partial p}{\partial x} \end{aligned}$$

where, in the second line, we have carried out the full product differentiation. We know that the second term of the left-hand side of this line is 0 due to conservation of mass, leaving us with

$$\rho \left(\frac{\partial u_1}{\partial t} + \mathbf{u} \cdot \nabla u_1 \right) = - \frac{\partial p}{\partial x}$$

Doing this for all components (x and y), we finally obtain vector-wise

$$\boxed{\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p} \quad (6)$$

where Equation 6 is the Euler equation for inviscid fluid flow!

In this paper, we are interested in the incompressible case, meaning that the fluid, which may be a gas or liquid, can change in shape but not in volume. This volume conservation requirement results in the continuity equation (2) giving $\operatorname{div} \mathbf{u} = 0$, which is due to ρ being constant in time. Setting $\rho = 1$ gives us all we need to obtain the form given by Equation 1.

Remark 1. *A consequence of incompressibility, dictated by $\operatorname{div} \mathbf{u} = 0$, is Liouville's Theorem, which states that the volume of phase space is invariant. Recall Hamilton's equations of motion:*

$$\dot{q}_i = \frac{\partial \mathcal{H}}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i}$$

where \mathcal{H} is the Hamiltonian, q_i is the generalized coordinate (usually spacial) and p_i is its conjugate momentum. By direct computation, we can show that $\operatorname{div} \mathbf{u} = 0$ implies that the volume of phase space is preserved. Let $\mathcal{H}(\mathbf{q}, \mathbf{p}) \equiv \mathcal{H}(\mathbf{z})$ where $\mathbf{z} \equiv (q_1, \dots, q_n; p_1, \dots, p_n) \in \mathbb{R}^{2n}$, and

$$\dot{\mathbf{z}} = \mathbb{J} \cdot \nabla \mathcal{H}(\mathbf{z})$$

where $\dot{\mathbf{z}}$ is the phase space velocity and \mathbb{J} is the $2n \times 2n$ symplectic matrix

$$\mathbb{J} = \begin{bmatrix} 0 & \mathbb{1} \\ -\mathbb{1} & 0 \end{bmatrix}$$

with $\mathbb{1}$ being the $n \times n$ identity matrix. Taking the divergence of $\dot{\mathbf{z}}$ yields

$$\begin{aligned} \nabla \cdot (\dot{\mathbf{z}}) &= \nabla \cdot (\mathbb{J} \cdot \nabla \mathcal{H}) \\ &= \sum_{i=1}^n \frac{\partial}{\partial q_i} \left(\frac{\partial \mathcal{H}}{\partial p_i} \right) + \frac{\partial}{\partial p_i} \left(-\frac{\partial \mathcal{H}}{\partial q_i} \right) = 0 \end{aligned}$$

Simply replace the phase space coordinates by (x, y) to obtain $\operatorname{div} \mathbf{u} = 0$.

2.2 Vorticity

In order to avoid the use of the physical parameter $p(x, y, t)$, it is useful to define the **vorticity** of a fluid as $\boldsymbol{\omega} = \nabla \times \mathbf{u}$. Informally, this is a measure of the rotation of a fluid since we are taking the curl of the velocity field. Our goal is now to derive the vorticity version of the Euler equations in terms of \mathbf{u} and $\boldsymbol{\omega}$. We start by taking the curl of both sides of Equation 1

$$\begin{aligned}\nabla \times \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) &= \nabla \times (-\nabla p) \\ \frac{\partial(\nabla \times \mathbf{u})}{\partial t} + \nabla \times ((\mathbf{u} \cdot \nabla) \mathbf{u}) &= 0 \\ \boldsymbol{\omega}_t + (\mathbf{u} \cdot \nabla) \boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} &= 0\end{aligned}$$

where in the second line, the curl of the gradient of $p(x, y, t)$ gives zero, and in the third line, we take the curl of a convection term. Noting that the material derivative is defined as $\frac{D\boldsymbol{\omega}}{Dt} = \boldsymbol{\omega}_t + (\mathbf{u} \cdot \nabla) \boldsymbol{\omega}$ we get the final form of the vorticity equation

$$\frac{D\boldsymbol{\omega}}{Dt} = (\boldsymbol{\omega} \cdot \nabla) \mathbf{u}. \quad (7)$$

Physically speaking, the term $(\mathbf{u} \cdot \nabla) \boldsymbol{\omega}$ in the material derivative is known as the advection term, which describes how the vorticity $\boldsymbol{\omega}$ is carried (advected) by the velocity field \mathbf{u} , just like how temperature or dye would be transported in a fluid.

Let's now look at the 2D case of the vorticity equation, where we will get some nice cancellation and a simpler form. Let $\mathbf{u} = (u_1(x, y, t), u_2(x, y, t), 0)$ that means the only nonzero term for $\boldsymbol{\omega}$ is $\boldsymbol{\omega}_z = \frac{\partial u_2}{\partial x} - \frac{\partial u_1}{\partial y}$. Hence, the left-hand side of Equation 7 is clearly zero, and we get

$$\begin{aligned}\frac{D\boldsymbol{\omega}}{Dt} &= 0 \\ \frac{\partial \boldsymbol{\omega}}{\partial t} + \frac{\partial u_1}{\partial x} \boldsymbol{\omega} + \frac{\partial u_2}{\partial y} \boldsymbol{\omega} &= 0 \\ \boldsymbol{\omega}_t + \operatorname{div}(\boldsymbol{\omega} \mathbf{u}) &= 0\end{aligned} \quad (8)$$

where $\boldsymbol{\omega}$ is always perpendicular to the flow in 2D and can therefore be considered as a scalar field. Equation 8 will be central in Section 3.

3 Point-Vortex Method

The Euler equations are computationally challenging to solve due to their nonlinear nature. Those interested in computational mathematics often resort to what we call **Point-Vortex Methods**,

which discretize the fluid and apply Euler to each point using the vorticity. This is especially useful in engineering when looking at the motion of a fluid around a solid body. We begin with a derivation of this method, then follow with a discussion about its convergence to Euler.

3.1 Derivation

Let n be the number of point vortices making up an inviscid, incompressible fluid in \mathbb{R}^2 with positions given by $\mathbf{x}_1(\mathbf{t}), \dots, \mathbf{x}_n(\mathbf{t})$ at time t where $\mathbf{x}_i(t) = (x_i(t), y_i(t))$. The velocity for a fixed time of each point vortex is then

$$\frac{d}{dt} \mathbf{x}_i(\mathbf{t}) = \mathbf{u}_i(\mathbf{x}_i(t), t) \quad (9)$$

which is an easily solvable ODE that yields the position given the initial conditions of \mathbf{u} . We must hence find a way to solve for \mathbf{u} via Euler so it can be fed into Equation 9. Recall that $\operatorname{div} \mathbf{u} = 0$ which means that we can define what we call the stream function ψ via $u = \nabla^\perp \psi = (\psi_y, -\psi_x)$ ³. With this new formalism, we now look at the vorticity again:

$$\boldsymbol{\omega} = \operatorname{curl} \mathbf{u} = \operatorname{curl} \nabla^\perp \psi = -\psi_{yy} - \psi_{xx} = -\Delta \psi$$

This means that to obtain \mathbf{u} from $\boldsymbol{\omega}$ we must compute (1) $-\Delta \psi = \boldsymbol{\omega}$ and then (2) $\mathbf{u} = \nabla^\perp \psi$. With this in hand, it does not seem like we have made our lives any easier. In fact, now we have to solve a whole Poissonian, which would be computationally tedious. We will come back to this momentarily. For now, let us define the circulation Γ_i , which is the strength of the vorticity at each point, i.e., $\Gamma_i = \boldsymbol{\omega}_i(\mathbf{x}_i(t), t)$. We are led to approximate the vorticity with

$$\boldsymbol{\omega}(\mathbf{x}, t) \approx \tilde{\boldsymbol{\omega}}(\mathbf{x}, t) = \sum_{i=1}^n \Gamma_i \delta(\mathbf{x} - \mathbf{x}_i) \quad (10)$$

where the Delta function picks out the strength of the vorticity at each point $x_i(t)$. Here is where the magic happens: if you have had experience with PDEs, you may recall the fundamental solution for the Laplacian in 2D which, in the sense of distributions, is often written as⁴:

$$\Delta G(\mathbf{x}; \mathbf{x}_\alpha) := \Delta \frac{1}{2\pi} \log |\mathbf{x} - \mathbf{x}_i| = \delta(\mathbf{x} - \mathbf{x}_i)$$

We can hence plug this into Equation 10 and take the ∇^\perp to get:

³For those familiar with differential forms, defining a scalar function ψ from the fact that $\operatorname{div} \mathbf{u} = 0$ follows from Poincaré's Lemma. Then, by uniqueness of the Poisson problem, we get the stream function ψ .

⁴Please see p.398-400 of [Cho22] for a great derivation.

$$\tilde{\omega}(\mathbf{x}, t) = \sum_{i=1}^n \Gamma_i \Delta \frac{1}{2\pi} \log |\mathbf{x} - \mathbf{x}_i| \quad (11)$$

$$\implies \tilde{\psi}(\mathbf{x}, t) = - \sum_{i=1}^n \Gamma_i \frac{1}{2\pi} \log |\mathbf{x} - \mathbf{x}_i|$$

$$\xrightarrow{\mathbf{u} = \nabla^\perp \psi} \dot{x} = - \frac{1}{2\pi} \sum_{\beta \neq \alpha}^n \Gamma_\alpha \frac{(y_\alpha - y_\beta)}{\|\mathbf{x}_\alpha - \mathbf{x}_\beta\|^2} \quad (12)$$

$$\text{and } \dot{y} = \frac{1}{2\pi} \sum_{\beta \neq \alpha}^n \Gamma_\alpha \frac{(x_\alpha - x_\beta)}{\|\mathbf{x}_\alpha - \mathbf{x}_\beta\|^2} \quad (13)$$

The set of Equations 12 and 13 are exactly what we need to solve to get the velocity field $\mathbf{u} = (\dot{\mathbf{x}}, \dot{\mathbf{y}})$ and evolve our point vortices in time. We hence went from a second-order PDE to two coupled first-order ODEs, which is much more computationally sound!

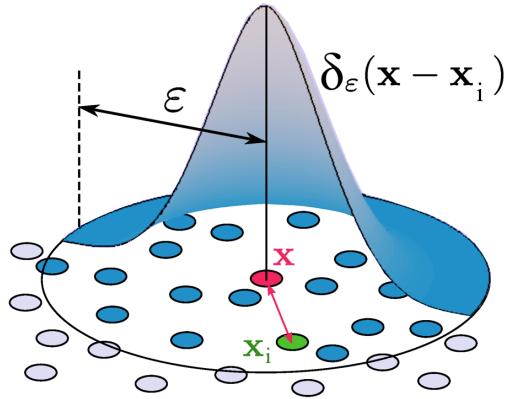


Figure 1: Visual of the Delta function going over each point vortex [MM21].

Remark 2. It turns out that the stream function ψ plays a role analogous to the Hamiltonian \mathcal{H} in phase space. As we know, the velocity field can be written as $\mathbf{u} = \nabla^\perp \psi$, which has the same structure as Hamilton's equations:

$$\dot{x} = \frac{\partial \psi}{\partial y}, \quad \dot{y} = - \frac{\partial \psi}{\partial x}.$$

This makes the motion of passive particles in the flow formally identical to trajectories in a Hamiltonian system, with the stream function ψ acting as the Hamiltonian. In this sense, streamlines are level sets of constant energy, and the incompressibility condition $\nabla \cdot \mathbf{u} = 0$ corresponds to the symplectic structure being preserved. This connection underlies many geometric and conservation properties of 2D fluid flows.

3.2 Convergence to Euler

Following this derivation, a natural question to ask is: Does the Point-Vortex Method converge to the Euler equations, and if so, in what sense? The answer is nuanced and was rigorously addressed in a foundational paper by Goodman, Hou, and Lowengrub [GHL90]. In their work, the authors establish that under appropriate smoothness and spacing conditions, the Point-Vortex Method indeed converges to the solution of the 2D incompressible Euler equations as the number of vortices $n \rightarrow \infty$.

The key insight is that if the initial vorticity is smooth and compactly supported, it can be approximated by a discrete measure using delta functions located at the vortex positions (as we did in Equation 10). Goodman et al. show that when this discretization is constructed carefully, ensuring that the vortex strengths are properly scaled and that the spacing between vortices vanishes appropriately, the velocity field generated by the point vortices converges in the weak sense to the true velocity field of the Euler solution.

Importantly, the convergence is shown in the sense of distributional solutions: the empirical vorticity of the point vortex system converges to a weak solution of the vorticity form of the Euler equations. This convergence relies on both consistency and stability.

Moreover, they prove that this method is second-order accurate in the vortex spacing, assuming the initial vorticity is C^2 . However, the convergence can break down in the presence of singularities or filamentation in the vorticity, which motivates more refined vortex-blob or regularized vortex methods in practical simulations.

Thus, while the Point-Vortex Method is a dramatic simplification of the fluid, it retains fidelity to Euler under suitable assumptions. This result gives theoretical grounding to the practical effectiveness of vortex methods in fluid dynamics, particularly in inviscid, incompressible regimes.

3.3 Boundary Conditions

We will now deal with n vortices in 2-dimensional domains with solid boundaries. This section follows closely from the explanations given in *The N-Vortex Problem: Analytic Techniques* by Paul K. Newton [New01]. We will stick with closed, simply connected regions, however, this could of course be generalized to multiply connected or unbounded domains that contain a solid region. Let $\Omega \subset \mathbb{R}^2$ be a simply connected region where our fluid, made of n point vortices, lives. The condition imposed at the boundary $\partial\Omega$ is that of no fluid penetration, i.e,

$$\mathbf{u} \cdot \mathbf{n} = 0|_{\partial\Omega} \tag{14}$$

where \mathbf{n} is the unit normal to the boundary. This condition can also be written in term of the stream function ψ . Since

$$\begin{aligned}\mathbf{u} &\equiv \nabla^\perp \psi \\ \mathbf{n} &\equiv \nabla \psi,\end{aligned}$$

then Equation 14 is immediately satisfied on any constant streamline, which we usually choose to be zero, i.e., $\psi = 0$. Equation 14 in terms of the velocity potential $\mathbf{u} = \nabla \phi$ becomes a Neumann condition

$$\mathbf{n} \cdot \nabla \phi \equiv \frac{\partial \phi}{\partial \mathbf{n}} = 0|_{\partial \Omega}.$$

It turns out that there exists a Green's function which we can construct in a domain with boundaries to help us solve the boundary value problem. In the closed, simply connected domain Ω , the Green's function of the first kind G_I can be written as

$$\nabla^2 G_I(\mathbf{x}; \mathbf{x}_\alpha) + \delta ||\mathbf{x} - \mathbf{x}_\alpha|| = 0, \quad \mathbf{x} \in \Omega \quad (15)$$

$$G_I(\mathbf{x}; \mathbf{x}_\alpha) = 0, \quad \mathbf{x} \in \partial \Omega \quad (16)$$

where the delta function represents the source located at $\mathbf{x}_\alpha \in \Omega$. $G_I(\mathbf{x}, \mathbf{x}_\alpha)$ is also known as the Dirichlet function since it can be used to solve the Dirichlet problem

$$\begin{aligned}\nabla^2 \mathbf{u} &= 0, \quad \mathbf{x} \in \Omega \\ \mathbf{u} &= f, \quad \mathbf{x} \in \partial \Omega\end{aligned}$$

using the generalized Poisson formula

$$\mathbf{u} = \int_{\partial \Omega} \frac{\partial G_I}{\partial n} dS$$

where $\partial G_I / \partial n \equiv \mathbf{n} \cdot \nabla G_I$ is the normal derivative of G_I on $\partial \Omega$ with \mathbf{n} being the outward unit normal. Constructing G_I truly depends on the details of the boundary shape, and is usually difficult to find.

To begin, we decompose G_I into a sum of two parts

$$G_I = G + G_H^{(\alpha)}$$

where G is the Green's function for the unbounded plane, or as we recall from the beginning of this section, the fundamental solution for the Laplacian

$$G(\mathbf{x}; \mathbf{x}_\alpha) = -\frac{1}{2\pi} \log \|\mathbf{x} - \mathbf{x}_\alpha\|.$$

Since G alone does not satisfy the boundary condition bestowed by Equation 16, we must add to it a harmonic function $G_H^{(\alpha)}$ which has $\nabla^2 G_H^{(\alpha)} = 0$ as well as enforce that

$$G_H^{(\alpha)} = -G = \frac{1}{2\pi} \log \|\mathbf{x} - \mathbf{x}_\alpha\|, \quad \mathbf{x} \in \partial\Omega$$

to satisfy the boundary condition (16). Hence, in many cases, in order to solve for the vorticity given by Equation 11, we must simply replace the term in the sum by G_I , i.e,

$$\tilde{\omega} = \sum_{\beta \neq \alpha}^n \Gamma_\beta G_I(\mathbf{x}; \mathbf{x}_\alpha)$$

and take the ∇^\perp of this to get the velocity field. We are therefore required to determine the harmonic function $G_H^{(\alpha)}$. One technique often used is the Method of Images⁵, where image vortices are placed outside the domain at strategic locations to account for the normal force from the boundary. For example, in the case of one point vortex in the upper-half plane at position $\mathbf{x}_1 = (x_1, y_1)$, the image vortex would be placed at the position $\mathbf{x}_1^* = (x_1, -y_1)$. The harmonic function would simply be the fundamental solution, however, evaluated at $G(\mathbf{x}; \mathbf{x}_1^*)$ instead of $G(\mathbf{x}; \mathbf{x}_1)$.

Another example, which we will revisit towards the end of Section 4, is a set of point vortices inside or outside a circular domain. Consider a point vortex of strength Γ located at position \mathbf{x}_1 inside a circular cylinder of radius R centered at \mathbf{x}_c . We begin by placing an image vortex of strength $-\Gamma$ at the inverse point

$$\mathbf{x}_1^* = \mathbf{x}_c + \frac{(\mathbf{x}_1 - \mathbf{x}_c)R^2}{\|\mathbf{x}_1 - \mathbf{x}_c\|^2}.$$

It is easy to show that there is no radial velocity component on the circle boundary, which means that the boundary condition (14) is satisfied. All that is left is the azimuthal velocity of the vortex, given by

$$u_\theta = \frac{\Gamma}{2\pi} \cdot \frac{\|\mathbf{x}_1 - \mathbf{x}_c\|}{(R^2 - \|\mathbf{x}_1 - \mathbf{x}_c\|^2)}$$

One could also switch the roles of \mathbf{x}_1 and \mathbf{x}_1^* which would be equivalent to placing the point vortex outside of the circular cylinder. In both cases, the Green's function of the first kind is given by

⁵Other techniques include using conformal mapping, which can be very useful for more complicated domains (see Ch 3.3 of [New01]).

$$\begin{aligned}
G_I(\mathbf{x}; \mathbf{x}_1) &= -\frac{1}{2\pi} \left[\log \|\mathbf{x} - \mathbf{x}_1\| - \log \left(\frac{R}{\|\mathbf{x}_1 - \mathbf{x}_c\|} \cdot \frac{1}{\|\mathbf{x} - \mathbf{x}_i^*\|} \right) \right] \\
&\equiv G(\mathbf{x}; \mathbf{x}_1) + G_H^{(1)}(\mathbf{x})
\end{aligned} \tag{17}$$

The above equation will be especially useful for actually simulating many point vortices inside a circular domain, yielding some interesting visuals we will dissect in the next section.

4 Computational Methods

Setting up the computation for the Point-Vortex Method can seem daunting, however, it turns out that all simplifies quite beautifully. Let n be the number of point vortices and $\mathbf{x}_i = (x_i, y_i)$ be the position array for each point. More specifically, $x_i = (x_1, \dots, x_n)$ and $y_i = (y_1, \dots, y_n)$ for $i \in \{1, 2, \dots, n\}$. Given initial conditions on the positions of the points, we will compute the velocities using the set of Equations 12 and 13. These velocities will then be plugged into a numerical integrator, which will be used to evolve the particle's positions. These steps are repeated for a given number of iterations and time step.

4.1 Numerical Integration: RK4

What is meant by numerical integration is not so much computing an integral, but solving an ODE. Here, we look for a method to solve differential equations of the form $dx/dt = u(x, t)$ and $x(t_0) = x_0$. There are many ways to go about this, which often stem from the well-known forward Euler; however, we will busy ourselves with the Runge Kutta methods, specifically (RK4).

In order to outline this method, we will begin our discussion with the first-order version. When dealing with numerical integration, we often want to predict the form of the solution by taking a small step h (in our case, a time step into the future). For RK1, this is accomplished by computing the Taylor expansion x about t_0 :

$$\begin{aligned}
x(t) &= x(t_0) + x'(t_0)t + x''(t_0)\frac{t^2}{2} + \dots \\
x(h) &= x(t_0) + x'(t_0)h + x''(t_0)\frac{h^2}{2} + \dots
\end{aligned}$$

To first order, we can hence approximate the solution at time step $h > 0$ to be

$$x(h) \approx x^*(h) = x(t_0) + k_1 h$$

where $k_1 = x'(t_0)$, i.e, the slope or velocity for the Point-Vortex Method. For RK2, we use the result k_1 to generate an approximation at $y(t_0 + h/2)$, and then determine k_2 from this estimation. This coefficient basically gives us the slope at $t_0 + h/2$. The third-order version follows a similar scheme to the prior. In the context of our Point-Vortex problem, we are solving for the system of equations yielding the x and y coordinates of the points given the velocity. Hence, the RK4 scheme used for n time steps is what follows:

$$\begin{aligned} k_1 &= \mathbf{u}(t_n, \mathbf{x}_n) \\ k_2 &= \mathbf{u}\left(t_n + \frac{h}{2}, \mathbf{x}_n + k_1 \frac{h}{2}\right) \\ k_3 &= \mathbf{u}\left(t_n + \frac{h}{2}, \mathbf{x}_n + k_2 \frac{h}{2}\right) \\ k_4 &= \mathbf{u}(t_n + h, \mathbf{x}_n + k_3 h) \end{aligned}$$

More specifically, k_1 is the slope at the beginning of the interval (using Euler's method), k_2 is the slope at the midpoint of the interval, k_3 is similar to k_2 , and k_4 is the slope at the end of the interval (see the left of Figure 2). We now take a weighted average to compute an estimation for $\mathbf{x}(t_{n+1}) \approx \mathbf{x}_{n+1}$, adding this to the current position \mathbf{x}_n we are at

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

where the division by 6 is because we are technically adding six slopes together. Note that the fact that we are using \mathbf{u} to calculate every coefficient indicates we will need to call our function that computes the velocity of our points every time.

It is important to take note that when **order** is mentioned in Runge-Kutta methods, this makes reference not to the number of coefficients used, but the order of the error made in approximating the solution. For RK4, the local truncation error is of $O(h^5)$, while the total accumulation error is of $O(h^4)$ ⁶.

Using RK4 is not a requirement. We could have easily used another numerical integration method, such as midpoint or Euler. However, RK4 comes with its perks; it offers a good tradeoff between accuracy and step size. As mentioned, the global truncation error is of $O(h^4)$, hence it can be used with a relatively large step h without the computation time being too long. The Point-Vortex Method is governed by nonlinear ODEs, which are sensitive to small changes, so this is a must for enhanced precision.

⁶The **local truncation error** is the error caused by one iteration, while the **total accumulated error** is the global error accumulated after many n number of iterations

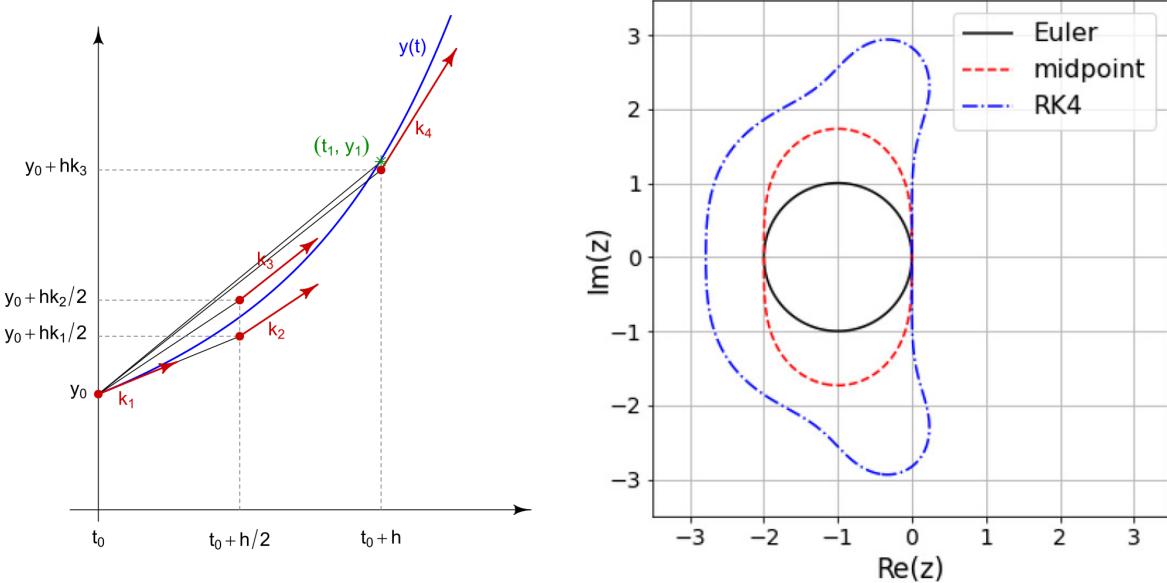


Figure 2: **Left:** Visual of the RK4 method. k_1, k_2, k_3 and k_4 slopes (in red) are added together as a weighted average to approximate the solution $y(t)$ at a later time t_1 [con25]. **Right:** Stability region of RK4 compared to the midpoint method and forward Euler [TY24].

4.2 Solving for the Point-Vortex Velocity

Solving for the velocity of the point vortices turns out to be simple matrix multiplication. The idea is to reconstruct the form of Equation 12 and 13 into an $n \times n$ matrix, making sure that each vortex does not feel its own contribution. The matrix operation equivalent to this is the following

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \vdots \\ \dot{y}_n \end{bmatrix} = \frac{1}{2\pi} \begin{bmatrix} 0 & \frac{x_1-x_2}{\|x_1-x_2\|^2} & \cdots & \frac{x_1-x_n}{\|x_1-x_n\|^2} \\ \frac{x_2-x_1}{\|x_2-x_1\|^2} & 0 & \vdots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \frac{x_n-x_1}{\|x_n-x_1\|^2} & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_n \end{bmatrix} = \frac{1}{2\pi} \bar{\mathbf{A}} \boldsymbol{\Gamma}$$

provided we supply an array of constant Γ_i 's. The matrix for the \dot{x}_i 's will be the same modulo a minus sign as well, and the x_i 's replaced by y_i 's. Notice the zeros on the diagonal, accounting for the null contribution of a point vortex with respect to itself. Additionally, the vector-matrix multiplication of the $\boldsymbol{\Gamma}$ into the $\bar{\mathbf{A}}$ does the job of summing up all contributions to a single point's velocity.

Constructing the matrix $\bar{\mathbf{A}}$ can be done by creating a first matrix with rows of the form $[x_1, \dots, x_n]$, transposing this matrix, and then taking the difference between the transposed and normal objects. Another with the $1/\|x_\alpha - x_\beta\|^2$ elements can then be generated with zeros on the diagonal and can

be multiplied element-wise into the previous matrix. This is better shown by the operations

$$\left(\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T - \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ x_1 & x_2 & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ x_1 & \cdots & \cdots & x_n \end{bmatrix} \right) \times \begin{bmatrix} 0 & \frac{1}{\|x_1-x_2\|^2} & \cdots & \frac{1}{\|x_1-x_n\|^2} \\ \frac{1}{\|x_2-x_1\|^2} & 0 & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{1}{\|x_n-x_1\|^2} & \cdots & \cdots & 0 \end{bmatrix}$$

where the \times symbol represents element-wise multiplication. The same can be done for \dot{x} (and in fact this can be generalized to one big matrix with a 1×2 array for each entry). As with any matrix multiplication in numerics, there is $O(n^2)$ computational complexity as we are in the 2D case of inviscid Euler.

4.3 Point-Vortices in a Circle

A great first test to see if the computation is working and to check the stability is to arrange the point-vortices in a shape with a predictable position evolution. For this section, we place 8 points in a circle, as seen in Figure 3, each with the same circulation Γ . We know that at any later time step, the points should remain constrained to move at a radial distance $R = 2$ from the circle's center, in this case (5, 5). It is then easy to calculate any change in position of the points due to the numerical methods, by taking the difference $|r_i - R|$ where r_i is the radial position of each point from the circle's center.

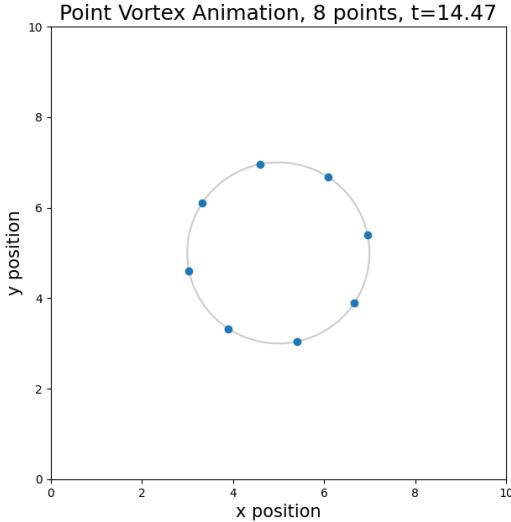


Figure 3: Position of a set of 8 point-vortices arranged uniformly in a circle of radius $R = 2$, evolved via Euler for $t = 14.47$ s and $h = 0.005$ s.

Running the simulation for 14.47s and $h = 0.005$ s yielded a difference of the order 10^{-13} , which indicates an extremely high degree of numerical accuracy. This level of error is well below the typical threshold for concern in double-precision floating-point computations (generally around 10^{-10} and 10^{-12}), and is consistent with the expected precision of the RK4 integrator. It should be clear that what governs this evolutionary shift in theoretical positioning is the numerical integrator method, as the method used to calculate the velocities of the points is already a predetermined approximation.

4.4 Dual Vortex Initial Condition

It is now time to explore more interesting initial conditions, which will certainly yield prettier images. In this section, we will deal with a dual vortex system⁷ with Gaussian initial conditions on the circulations, of the form⁸

$$\Gamma_i(\mathbf{x}_i) = -(g(\mathbf{x}_i - (0.15, 0) - \mathbf{m}) + g(\mathbf{x}_i + (0.15, 0) - \mathbf{m})) \quad (18)$$

$$g(\mathbf{x}_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{|\mathbf{x}_i|^2}{\sigma^2}\right) \quad (19)$$

with $\sigma = 0.07$ and $\mathbf{m} = (0, 0)$. Figure 4 provides the visual for the initial condition for $n = 10\,000$.

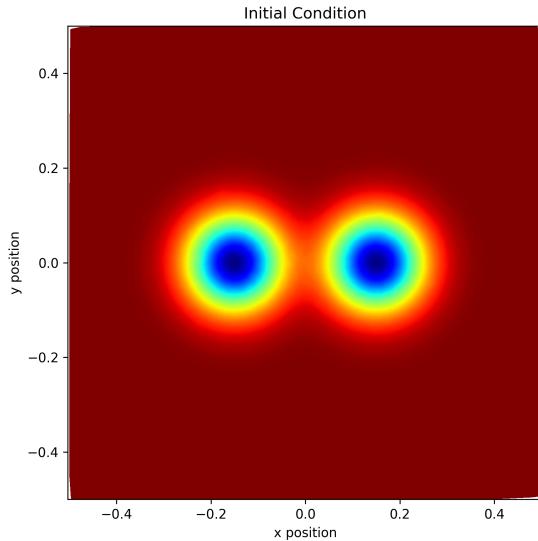


Figure 4: Dual vortex initial conditions. The colors directly map to the strength of the circulation Γ .

⁷Not to be confused with the **point-vortices** which simply refers to our n points.

⁸This initial condition was strongly inspired by some results from the master's thesis of William Holman-Bissegger [Hol24].

As the points are time-evolved using the Point-Vortex Method, the Gaussians begin to swirl around each other (Figure 5), creating streaks reminiscent of the arms of some spiral galaxies. Delaunay triangulation is applied in order to show a more "continuum-like" like picture. Certain points can be observed to have diverged greatly from their central position. This is most likely due to the Point-Vortex Method being analogous to the Biot-Savart Law: when two point-vortices come in great proximity, their velocities diverge since we are working with an inverse square law.

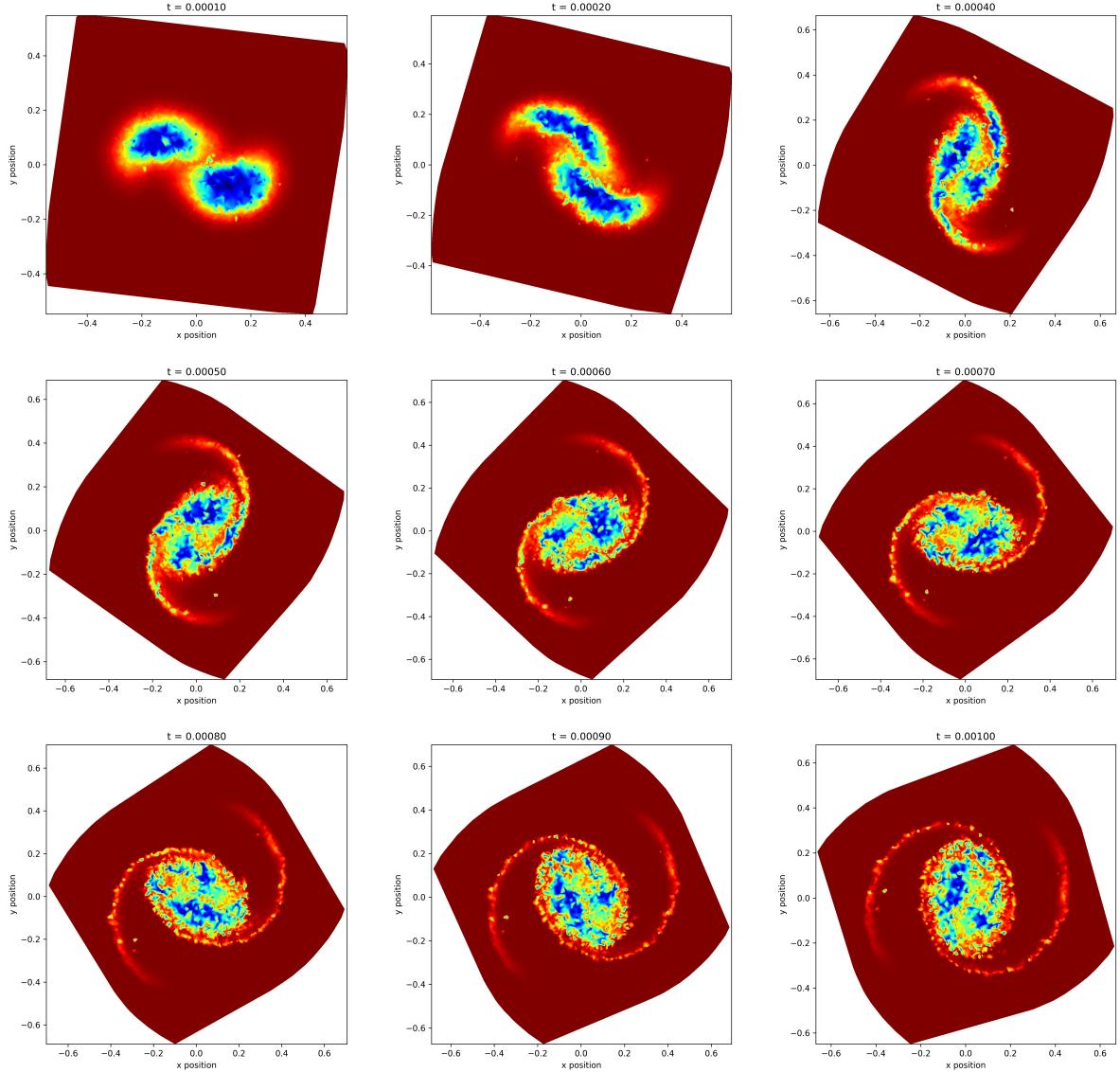


Figure 5: Evolution of a dual vortex merger for $h = 0.0001$ seconds and $n = 10\,000$.

As illustrated in the figure above, the Point-Vortex method breaks down at small spatial scales and requires a significantly large number of vortices to resolve fine-scale features such as turbulence or vorticity filaments.

4.5 Lamb-Chaplygin Dipole in a Cylindrical Boundary

We now tackle cylindrical boundary conditions computationally, building up from our previous derivations in Section 3.3. Implementing a solid circular boundary will follow from prior computational techniques (4.2), along with a few key changes which involve adding the contributions from image point vortices. The logic here is straight-forward: Recall from Equation 11 that for the discretized vorticity, we replaced the Delta function by the Laplacian of the Fundamental Solution, which was then mapped to the stream function $\tilde{\omega} = -\Delta \tilde{\psi}$. We are wishing to accomplish something similar, this time with a different Green's function which takes into account the cylindrical boundary, resulting in a modified stream function. Luckily, we have previously presented such a Green's function, i.e, Equation 17. Following from the previous computation, we derive our modified stream function:

$$\begin{aligned}\tilde{\psi}_C(\mathbf{x}, t) &= -\sum_{i=1}^n \Gamma_i G_{I,C}(\mathbf{x}; \mathbf{x}_i) \\ &= -\sum_{i=1}^n \Gamma_i \frac{1}{2\pi} \left[\log \|\mathbf{x} - \mathbf{x}_i\| - \log \left(\frac{R}{\|\mathbf{x}_i - \mathbf{x}_c\|} \cdot \frac{1}{\|\mathbf{x} - \mathbf{x}_i^*\|} \right) \right] \\ \xrightarrow{\tilde{\mathbf{u}} = \nabla^\perp \tilde{\psi}} \dot{x}_C &= -\frac{1}{2\pi} \sum_{\alpha \neq \beta}^n \Gamma_\alpha \frac{(y_\alpha - y_\beta)}{\|\mathbf{x}_\alpha - \mathbf{x}_\beta\|^2} - \frac{(y_\alpha - y_\beta^*)}{\|\mathbf{x}_\alpha - \mathbf{x}_\beta^*\|^2} \quad (20)\end{aligned}$$

$$\text{and } \dot{y}_C = \frac{1}{2\pi} \sum_{\alpha \neq \beta}^n \Gamma_\alpha \frac{(x_\alpha - x_\beta)}{\|\mathbf{x}_\alpha - \mathbf{x}_\beta\|^2} - \frac{(x_\alpha - x_\beta^*)}{\|\mathbf{x}_\alpha - \mathbf{x}_\beta^*\|^2} \quad (21)$$

where the * symbol denotes the image vortices and R , the radius of the cylinder. This modified form contains the same term as our previous system in \mathbb{R}^2 added to another term which accounts for the influence of the image points. This indicates that we must go through the construction of an additional matrix analogous to the one in Section 4.2 and add the vorticity extracted from it to the normal \mathbb{R}^2 vortex velocity. Going through all the steps again leads to a final operation of the form

$$\begin{aligned}\begin{bmatrix} \dot{y}_{C,1} \\ \dot{y}_{C,2} \\ \vdots \\ \dot{y}_{C,n} \end{bmatrix} &= \frac{1}{2\pi} \left(\begin{bmatrix} 0 & \frac{x_1 - x_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|^2} & \cdots & \frac{x_1 - x_n}{\|\mathbf{x}_1 - \mathbf{x}_n\|^2} \\ \frac{x_2 - x_1}{\|\mathbf{x}_2 - \mathbf{x}_1\|^2} & 0 & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{x_n - x_1}{\|\mathbf{x}_n - \mathbf{x}_1\|^2} & \cdots & \cdots & 0 \end{bmatrix} - \begin{bmatrix} 0 & \frac{x_1 - x_2^*}{\|\mathbf{x}_1 - \mathbf{x}_2^*\|^2} & \cdots & \frac{x_1 - x_n^*}{\|\mathbf{x}_1 - \mathbf{x}_n^*\|^2} \\ \frac{x_2 - x_1^*}{\|\mathbf{x}_2 - \mathbf{x}_1^*\|^2} & 0 & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{x_n - x_1^*}{\|\mathbf{x}_n - \mathbf{x}_1^*\|^2} & \cdots & \cdots & 0 \end{bmatrix} \right) \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_n \end{bmatrix} \\ &= \frac{1}{2\pi} (\bar{\mathbf{A}} - \bar{\mathbf{B}}) \boldsymbol{\Gamma}\end{aligned}$$

where $\bar{\mathbf{A}}$ is the same matrix as before and $\bar{\mathbf{B}}$ is the matrix accounting for the contributions from the image vortices. As before, an analogous statement can be made for $\dot{\mathbf{x}}_C$.

We are now ready to deal with an interesting example known as the Lamb-Chaplygin dipole in a cylindrical boundary. This system is a steady, axisymmetric solution of the 2D incompressible Euler equations describing a pair of counter-rotating vortices that propagate together at constant speed without changing shape. The stream function associated with this is

$$\psi = \begin{cases} \frac{-2UJ_1(kr)}{kJ_0(kR)} \sin \theta, & \text{for } r < R \\ U\left(\frac{R^2}{r} - r\right) \sin \theta, & \text{for } r \geq R \end{cases} \quad (22)$$

where J_0 and J_1 are the zeroth and first order Bessel functions of the first kind and R is the radius of the dipole, not to be confused with the radius of the cylinder. It turns out that the vorticity is linearly related to the stream function via $\omega = k^2\psi$ where k is the first non-trivial zero of the first Bessel function of the first kind, which gives $kR = 3.8317\dots$. A visual of the initial conditions is provided in Figure 6.

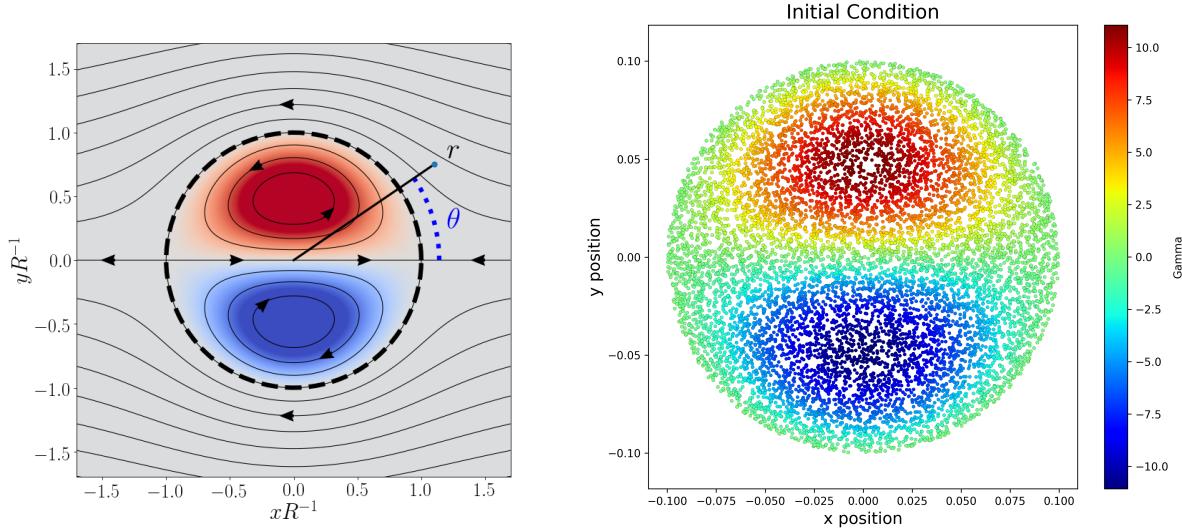


Figure 6: **Left:** Flow structure of the Lamb-Chaplygin dipole [con24]. **Right:** Dipole initial condition with 10 000 point-vortices.

We will be looking to see the effects of this dipole hitting the wall of the cylindrical boundary. Figure 7 shows a sample simulation with $R_{cyl} = 3$ and $n = 1000$ point-vortices. Such parameters were chosen as simulating with more points rendered it visually difficult to decipher the components of the dipole. The dipole can be seen moving towards the boundary with velocity U , leaving behind a trail due to numerical artifacts. As it hits the boundary, the dipole splits into two components

with positive and negative vorticity. As they go their separate ways, they eventually meet back and collide again.

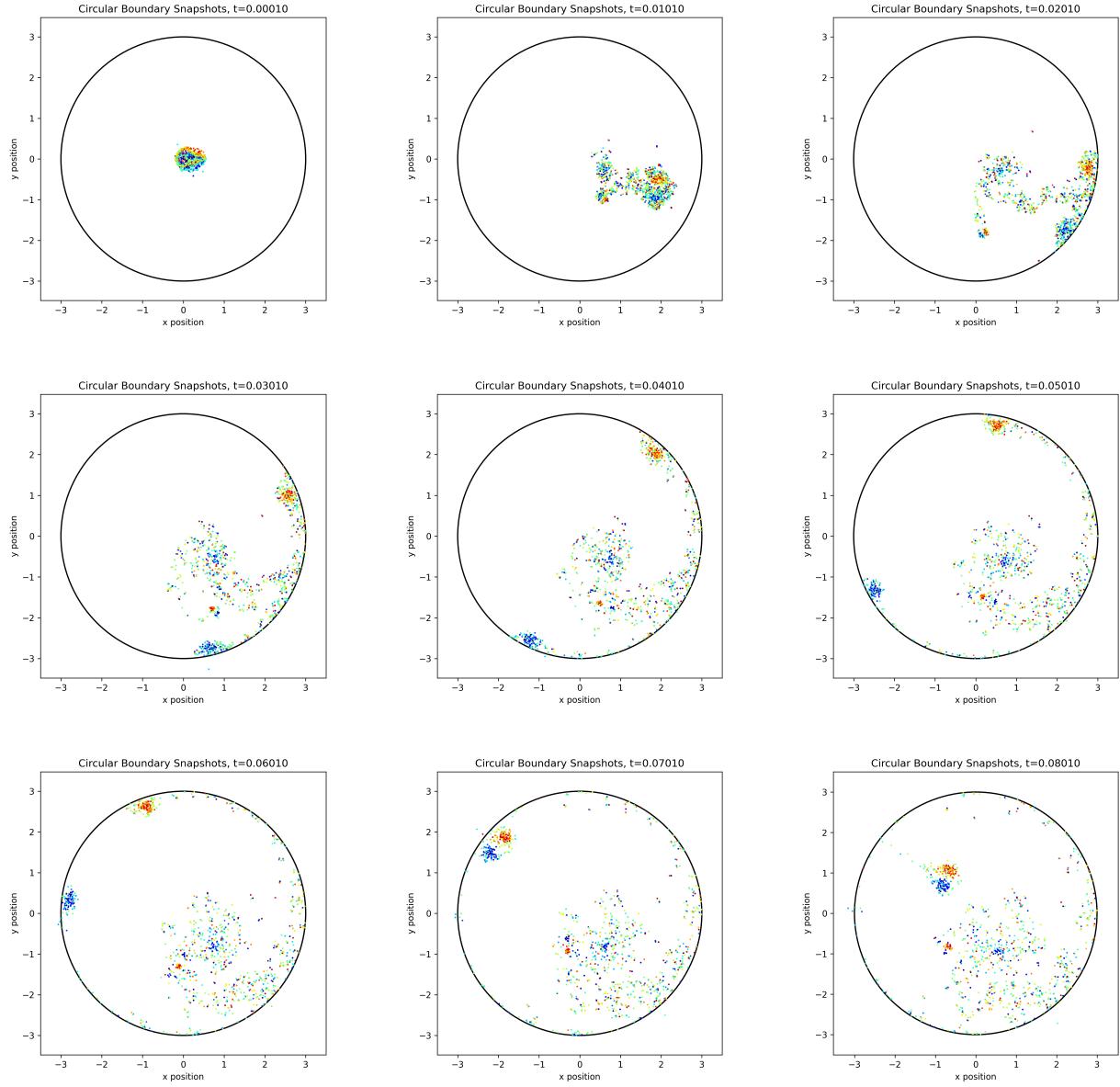


Figure 7: Evolution of a Lamb–Chaplygin dipole made up of 1000 point vortices.

4.6 Future Improvements: Fast Multipole Method

One major limitation of the Point-Vortex Method is its computational cost, which scales as $\mathcal{O}(n^2)$ due to the need to compute pairwise interactions between all n vortices at each time step. This becomes prohibitively expensive for large systems. A promising avenue for improving the efficiency of the method is the incorporation of the **Fast Multipole Method** (FMM), an algorithm originally developed to accelerate N -body interactions in physics and astronomy. The core idea of FMM is to approximate the effect of distant groups of vortices using multipole expansions, significantly reducing the number of direct calculations required. By organizing vortices into a hierarchical tree structure and computing interactions between clusters rather than individual particles, the overall complexity can be reduced to nearly $\mathcal{O}(n)$. Integrating FMM into the Point-Vortex framework would allow for the simulation of much larger vortex systems while preserving accuracy, making it an attractive direction for future development and application to realistic fluid flows.

5 Conclusion

This exploratory paper examined numerical approaches to solving partial differential equations, focusing on the incompressible Euler equations, which describe the behavior of inviscid fluids. The Point-Vortex Method was introduced as a powerful tool for reducing this complex PDE system to a set of coupled ordinary differential equations through the concept of vorticity. By discretizing the fluid domain, we explored several illustrative systems, including a vortex pair and the Lamb–Chaplygin dipole, within this framework.

Beyond fluid dynamics, it is worth noting that the Point-Vortex Method can be generalized to PDEs reducible to the form $d\mathbf{x}/dt = \sum_i a_i f(\mathbf{x} - \mathbf{x}_i)$. In the classical vortex formulation, $f(\mathbf{x} - \mathbf{x}_i)$ reflects the inverse-square interaction characteristic of point vortices. However, replacing f with alternative interaction kernels allows this framework to model the collective behavior of other systems exhibiting self-organization, such as flocks or herds [Gon+24].

6 Acknowledgments

I would like to extend my deepest thanks to my mentor, William Holman-Bissegger, for his invaluable guidance and insight into this project, as well as to Mikey and Parker for their support. I would also like to thank the organizers of this year’s DRP at McGill for providing resources and organizing the program. While many sources were used in the compilation of this project, much of its foundational material comes from Paul K. Newton [New01] and C. Mimate & I Mortazavi [MM21].

When finalized, the code for this project will be uploaded on GitHub via <https://github.com/ameliechd>.

References

- [GHL90] Jonathan Goodman, Thomas Y. Hou, and John Lowengrub. “Convergence of the point vortex method for the 2-D Euler equations”. In: *Communications on Pure and Applied Mathematics* 43.3 (1990), pp. 415–430.
- [New01] Paul K. Newton. *The N-Vortex Problem: Analytic Techniques*. Springer, 2001.
- [MM21] C. Mimateau and I. Mortazavi. “A Review of Vortex Methods and Their Applications: From Creation to Recent Advances”. In: *Fluids* (2021). DOI: <https://doi.org/10.3390/fluids6020068>.
- [Cho22] Rustum Choksi. *Partial Differential Equations: A First Course*. AMS, 2022.
- [Gon+24] I. Gonzalez et al. “Emergence of collective behaviours from local Voronoi topological perception”. In: *Royal Society Open Science* (2024). DOI: <https://doi.org/10.1098/rsos.231537>.
- [Hol24] William Holman-Bissegger. “A volume-preserving characteristic mapping method for the 2D incompressible Euler equations”. MA thesis. McGill University, 2024.
- [TY24] S. Terakawa and T. Yaguchi. “Modeling Error and Nonuniqueness of the Continuous-Time Models Learned via Runge–Kutta Methods”. In: *Mathematics* (2024). DOI: <https://doi.org/10.3390/math12081190>.
- [con24] Wikipedia contributors. “Lamb–Chaplygin dipole”. In: *Wikipedia* (Last edited: 2024). URL: https://en.wikipedia.org/wiki/Lamb%20%93Chaplygin_dipole.
- [con25] Wikipedia contributors. “Runge–Kutta methods”. In: *Wikipedia* (Last edited: 2025). URL: https://en.wikipedia.org/wiki/Runge%20%93Kutta_methods.