

Vortex Dominated Flows: A High-Order, Conservative Eulerian Simulation Method

Josh Bevan

April 23, 2015

Abstract

A high-order, conservative Eulerian method will be presented for the simulation of vortex dominated inviscid fluid flows. The primitive variable incompressible Euler equations are recast in the velocity-vorticity form to explicitly enforce conservation of vorticity. The advection of the vorticity is then calculated via a two-step process: the velocity field is first determined by evaluation of the Biot-Savart integral, and then a line-based discontinuous Galerkin (DG) Eulerian spatial discretization scheme is applied to accurately advect the vorticity field. The accuracy and convergence of this method is examined for test cases where an analytical solution exists, as well as more challenging test cases which lack an analytical solution. The influence the velocity field discretization has on the performance of the method is of particular interest.

Acknowledgments

Contents

List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Velocity-Vorticity equation	1
1.2 Lagrangian Methods	1
1.3 Lagrangian Method Difficulties	2
1.4 Existing Eulerian codes	2
1.5 Proposed Method	3
1.5.1 Alternate Advective Methods	3
1.5.2 Velocity Field Fidelity	4
1.6 Thesis Structure	4
2 Theory	6
2.1 Navier-Stokes: Velocity-vorticity form	6
2.2 Velocity Field Evaluation: The Biot-Savart Integral	7
2.3 Discontinuous Galerkin	9
3 Methodology	12
3.1 Overall Solver Structure	12
3.2 Use and Abuse of Structured Tensor Mesh	14
3.3 Mass Matrix	15
3.4 Modified Quadrature	15
3.5 Stiffness Matrix Integration	16
3.6 Velocity Evaluation	18
3.7 Vortex Sparseness	19
3.8 Reduced Order Velocity (Global)	19
3.9 Reduced Order Velocity (Far-field)	20
3.10 Explicit Time-Stepping	21
3.11 Validation and Convergence of Proposed Methods	22
4 Results	23
4.1 Analytical Test Cases	23
4.2 Elliptical Blob	27
4.3 Arbitrary Patch	30
4.4 Vortex Pair	33
4.5 Vortical System	36
4.6 Reduced Order Velocity	39
4.7 Runtime Speed and Memory Footprint of Solver	42

5	Discussion	43
5.1	Analytical Validation of Model	43
5.2	Empirical Convergence of Method	45
5.3	Computational Efficiency	47
5.4	Extended Body Fast Multipole Formulation	49
6	Conclusion	50
7	Recommendations	51
7.1	Extension to 3D	51
7.2	Flux Limiters	51
7.3	Impinging Geometry and Boundary Conditions	51
7.4	Viscosity	51
7.5	Implicit and Local Time Stepping	52
7.6	Parallelizability	52
7.7	FMM Improvements	52
7.8	2D Non-classical Quadrature for Nearly Singular Integrands	52
8	Implementation and Algorithms	53
8.1	Modified Two-part quadrature	53
8.2	Binary Singleton Expansion Matrix-Vector Products	54
8.3	Lagrange Bases and Constructed Quadrature	54
8.4	Velocity Evaluation	55
8.5	Semi-discrete System Construction	55
8.6	Low-storage Stability Optimized Runge-Kutta	56
9	Literature Cited	57

List of Tables

List of Figures

1 Introduction

1.1 Velocity-Vorticity equation

Direct solution of Navier-Stokes is impractical for many fluid problems and where possible simplifications should be made; one such possibility is for vortex dominated flows. It is possible to recast Navier-Stokes from a primitive variable form (u, v, p) to a velocity-vorticity (u, v, ω) form. This has several advantages: explicit conservation of vorticity, elimination of pressure, and reduction of required simulated DOFs to just those that form the vorticity support. For inviscid and incompressible flows the vorticity transport equation is:

$$\frac{\partial \omega}{\partial t} + u \cdot \nabla \omega - \omega \cdot \nabla u = S(x, t) \quad (1.1)$$

Traditionally simulation techniques take advantage of Helmholtz and Kelvin's theorems; vorticity exists as material surfaces, lines, etc. and therefore a Lagrangian approach is natural. Typically the vorticity is discretized into vortex particles, lines, or sheets at which point advection becomes trivial, one merely needs to calculate the velocity field at each particle and step forward in time [20, 38, 36]. Several monographs exist that enumerate the details and advantages of Lagrangian vortex methods [1, 2, 3].

1.2 Lagrangian Methods

There are several variations of Lagrangian methods depending on how the vorticity is discretized; point vortex methods were the earliest with the work of Rosenhead [4] in 2D with singular point vortices, which was later built upon and improved [5, 6, 7, 8, 9].

Higher dimensional discretizations take the form of vortex line [10, 11, 12, 13], sheets [14, 15, 16], and volumes [17, 18, 19].

The velocity field due to the vorticity is usually calculated from solving the Poisson problem, or through inversion of the Poisson problem via evaluation of the Biot-Savart integral. There are several challenges with evaluation of the velocity: boundary conditions in the Poisson solution method, and singularities in the Biot-Savart volume integral method. Typically Lagrangian point vortex codes de-singularize the kernel [32, 33]. Direct summation for the velocity field has a $\mathcal{O}(N^2)$ complexity. Tree-codes [31] and Fast Multipole Methods (FMM) [30] are common ways of achieving a more efficient.

The convergence of vortex methods was proven early on in 2D [25] and was later extended to 3D [26].

1.3 Lagrangian Method Difficulties

Lagrangian point methods present several problems. Point disorganization can occur as the fluid evolves, this typically requires temporary meshing to recondition the discretization. This has been handled with various methods; recalculation of the quadrature weights at each time step [21, 22], regridding/rezoning [23], and remeshing [24] among them. Additionally, most Lagrangian approaches are limited to low order; careful point locations must be chosen and maintained through frequent remeshing etc. in order to achieve high-order convergence [30].

1.4 Existing Eulerian codes

In comparison, far less literature exists for Eulerian approaches. Chief among them is the work of Brown et al. [27] who adapted the velocity-vorticity approach to a low order Finite Volume (FV) solver. Later they were able to extend their method

to be accelerated via a FMM [28]. However, like most FV approaches, to extend to high order solution approximations requires extended stencils that ultimately limit the geometrical freedom of the mesh. Steinhoff et al. also used an Eulerian approach, but solved a modified form of the inviscid Euler equations with "vorticity confinement" rather than the velocity-vorticity equation [39].

1.5 Proposed Method

1.5.1 Alternate Advective Methods

The desire to resolve fine vortical structures motivates the need for a high order solver. Considering the challenges associated with achieving a high order Lagrangian method and the preliminary success of Brown et al.'s low order method, it is natural to attempt a high order Eulerian vorticity-velocity method, but what of the spatial discretization choice?

Finite difference methods suffer from similar problems as FV with extended stencils, as well as not being explicitly conservative. A finite element approach is ill-suited to the hyperbolic nature of vorticity advection. Spectral methods are promising, but for sparse vorticity domains are less-efficient due to the global support of the harmonic bases. However, discontinuous Galerkin (DG) methods are a natural choice; they are conservative, able to take advantage of vorticity sparseness, are well-suited to handle advection via intelligent choice of a numerical flux function, and have bases with local/compact support.

For domains free of impinging bodies a hexahedral mesh with a tensor product grid of interpolation points is convenient to implement. This permits the use of a line-DG [34] approach that considerably simplifies multi-dimensional cases by allowing reuse of 1D methods. A 2D domain will be considered to evaluate whether the method is practical, as well as to limit solution times to those that are reasonable on a

workstation. This has the advantage of removing the vortex stretching term and reducing the vorticity to a scalar. The resultant partial differential equation (PDE) takes on the familiar form of a scalar conservation law.

1.5.2 Velocity Field Fidelity

Unlike Lagrangian methods where advection is trivial, or Brown's method which is at low order, the necessary fidelity of the velocity field to maintain a particular global convergence rate that one would expect from the order of the vorticity field is not immediately obvious. The method by which one recovers the velocity field is decoupled from the advective solver, which has no a priori knowledge or assumptions about the quality of the velocity field approximation.

To maintain maximum flexibility for investigative purposes and to remove as much approximation error as possible the velocity field for validation of the underlying method is calculated via direct evaluation of the BS integral.

1.6 Thesis Structure

In Chapter 2, a brief overview of the underlying theory of the velocity-vorticity formulation, velocity evaluation, the Biot-Savart kernel and the discontinuous Galerkin method are covered. The extension of one-dimensional DG methods to 2D via a line-DG style method will also be reviewed.

In Chapter 3 the methodology taken to construct a high-order Eulerian velocity-vorticity is covered. Formation of the semi-discrete system, along with the development of some specialized tools is addressed. A high-order velocity evaluation method is laid out, as well as some computation-saving modifications. Explicit time-stepping via a Runge-Kutta method optimized for the spectra of the DG operator is briefly reviewed. Finally a test plan is laid out for validation and investigation of the overall solver and method.

In Chapter 4 results of the validation tests, convergence studies, and performance of the possible improvements and modifications is presented. Additionally runtime performance of the solver is examined, dependent on solver parameters, tolerances, order, etc.

In Chapter 5 the successful performance of the method and solver is presented and discussed. Additionally an un-implemented extension to the velocity evaluation via a modified Fast Multipole Method is presented. Chapter 7 points in the direction of possible expansion of the method to include additional physical behaviors and configurations, as well as several next steps that should be taken to improve the solver for practical solution of larger problems beyond what is currently possible.

Finally, Chapter 8 presents the specific implementation details necessary to implement the solver practically and efficiently. Several Matlab functions and routines are presented that either makeup the solver itself, or necessary auxiliary tools that are tangentially needed.

2 Theory

2.1 Navier-Stokes: Velocity-vorticity form

If the quantity *vorticity* is defined as

$$\omega = \nabla \times u \quad (2.1)$$

Then the traditional form of the Navier-Stokes equations can be recast, assuming an inviscid incompressible flow

$$\frac{\partial \omega}{\partial t} + u \cdot \nabla \omega - \omega \cdot \nabla u = S(x, t) \quad (2.2)$$

There are several benefits to the recast form; first, it explicitly conserves vorticity. The vorticity is recoverable from the primitive variable Navier-Stokes, but the diffusive nature of most upwinded advection means coherent vortical structures are quickly smeared. Also, frequently the distribution of the vorticity is quite sparse. The primitive variable form requires the velocity and pressure be evaluated over the whole domain.

In comparison, the velocity-vorticity form only requires velocity to be evaluated in areas of non-zero vorticity. In areas where there is no vorticity, nothing needs to be advected. The sparser and more concentrated the vorticity, the larger the gap in performance between the two forms; this is especially true in 3D domains where the scaling from surfaces to volumes gives even greater opportunity for empty local regions in the domain.

Additionally, the primitive-variable form requires the pressure be solved via a separate elliptic equation. The velocity-vorticity form is absent the pressure and

automatically ensures satisfaction of continuity.

Restricting to examining 2-D distributions of vorticity, several simplifications can be made. The originally vectorial vorticity becomes a scalar quantity, all vorticity is directed normal to the plane. As a result, the vortex stretching term in (2.2) becomes zero. The only non-zero component of ω is in the z-direction, however the gradient of the velocity field is zero in the z-direction, so the product is therefore zero. The result is

$$\frac{\partial \omega}{\partial t} + u \cdot \nabla \omega = S(x, t) \quad (2.3)$$

or if instead the second term is expressed in terms of the flux of the vorticity (where $f_i(\omega) = u_i \omega$):

$$\frac{\partial \omega}{\partial t} + \frac{\partial f}{\partial x_i} = S(x, t) \quad (2.4)$$

2.2 Velocity Field Evaluation: The Biot-Savart Integral

We shall decide that ω is the quantity of interest in the solution method. However, if this is the case then the question must be posed: how does one determine the velocity field? For an incompressible flow

$$\nabla^2 u = -\nabla \times \omega \quad (2.5)$$

If inverted, the Biot-Savart integral is obtained

$$u(x^*) = \int_{\Omega} K(x^*, x) \times \omega(x) dx \quad (2.6)$$

with the singular Biot-Savart kernel

$$K(x^*, x) = \frac{-1}{4\pi} \frac{x^* - x}{|x^* - x|^3} \quad (2.7)$$

There are several important points to note that are a consequence of this inversion. First, rather than solving the Poisson equation for the entirety of the domain we

choose to evaluate the velocity at some subset. For the purposes of advecting the velocity we will only need velocities near the vorticity itself. However, if the number of required velocity evaluation points is roughly proportional to the N DOFs of our vorticity approximation, then it might be expected for the velocity calculations to scale as $\mathcal{O}(N^2)$.

There are numerous methods to reduce the computational complexity of similar N-body type problems to $\mathcal{O}(N \log N)$ or $\mathcal{O}(N)$. Cyclic reduction [40], tree-codes [31, 41], and FMM [42] are all possibilities. However, because the velocity field calculation method is essentially decoupled from the discretization of the PDE, there is no a priori assurance that the velocity calculated is of sufficient fidelity to ensure convergence of the overall method (let alone convergence at the order one might expect based on solely the discretization). For maximum flexibility in investigating this dependency of overall convergence on the calculated velocity field we shall eschew more efficient techniques so that we can directly control the fidelity of the velocity field.

The second point to consider regarding the Biot-Savart integral is the singular nature of the exact kernel. Lagrangian point vortex methods have the benefit that the singularity and its associated non-physical velocities occur in a relatively small region; they assume that the advective effect of the point vortex on itself is negligible. Generally, self-influence merely results in solid body rotation. Additionally they will frequently de-singularize the kernel by introducing a core function. The classical one used is the Rosenhead-Moore kernel [32, 33]:

$$K(x^*, x) = \frac{-1}{4\pi} \frac{x^* - x}{(|x^* - x|^2 + \sigma^2)^{3/2}} \quad (2.8)$$

The high-order Eulerian approach taken here however means that the Biot-Savart integral diverges everywhere within any of the extended vorticity patches thanks to self-influence. The approach taken by Brown [28] was to use the Rosenhead-Moore

kernel, choosing a core size such that the maximum velocity occurred on the face of the finite volume unit. This is constructible a priori because the vorticity is taken as constant across the volume, as is typical in a FV approach. If the vorticity is spatially varying however, the choice of core size is more troublesome. This will be examined in greater detail in the next chapter.

2.3 Discontinuous Galerkin

In order to solve (2.9) we adopt a method-of-lines approach. We will first spatially discretize the system to obtain the semi-discrete system, then we use an explicit time discretization method to march forward in time. Note that (2.9) has the form of a scalar conservation law, with ω being the conserved quantity. The velocity field that advects the conserved quantity has been calculated by evaluation of the Biot-Savart integral for the current timestep.

Our spatial discretization is at best an approximate solution to (2.9), that we shall denote as $\tilde{\omega}(x, t)$. Some residual will remain for the approximate solution of the PDE at any time t

$$\frac{\partial \tilde{\omega}}{\partial t} + \frac{\partial f}{\partial x_i} = R(x) \quad (2.9)$$

where we have shown a 1-D case and omitted vorticity sources for simplicity.

The discontinuous Galerkin (DG) approach attempts to approximately satisfy the PDE in the following way: Seek the best approximation in a finite vector test space \mathbb{W}_h in the L^2 norm sense. Minimize the L^2 norm by an orthogonal projection of the residual onto \mathbb{W}_h . Form a complete basis for \mathbb{W}_h with a set of test functions $\phi_j \in \mathbb{W}_h$, such that the orthogonal projection satisfies:

$$\int_{\Omega} R(x) \phi_j \, dx = 0 \quad \text{for all } j \quad (2.10)$$

Substituting the residual for our conservation PDE yields:

$$\int_{\Omega} \frac{\partial \tilde{\omega}}{\partial t} \phi_j dx + \int_{\Omega} \frac{\partial f(\tilde{\omega})}{\partial x} \phi_j dx = 0 \quad \text{for all } j \quad (2.11)$$

Choose the space of piecewise polynomials for the finite dimensional approximation space, and decompose the global domain into K elements where the local approximation space \mathbb{V}_h has basis functions $\psi_i^k(x)$ over the local domain $x \in [x_L, x_R]$ (henceforth we drop the element index k for brevity, except when describing two distinct elements that must be distinguished). Note that continuity of vorticity is not enforced across elements, and that the local approximation spaces for each element are independently defined. Finally, take the Bubnov-Galerkin choice of setting the test space to be the same as the approximation space so that $\phi_j = \psi_i \in \mathbb{V}_h$ if $i = j$. The local M th order approximation to vorticity takes the form

$$\omega(x, t) \approx \tilde{\omega}(x, t) = \sum_{i=0}^M a_i(t) \psi_i(x) \quad (2.12)$$

where a_i is some set of coefficients that are to be determined. The elemental approximating PDE is then

$$\sum_{i=0}^M \left[\frac{da_i(t)}{dt} \int_{x_L}^{x_R} \psi_i(x) \phi_j(x) dx \right] + \int_{x_L}^{x_R} \frac{\partial f(\tilde{\omega})}{\partial x} \phi_j dx = 0 \quad (2.13)$$

To reduce the smoothness requirements on the flux, integrate the second term by parts to yield

$$\sum_{i=0}^M \left[\frac{da_i(t)}{dt} \int_{x_L}^{x_R} \psi_i(x) \phi_j(x) dx \right] + f \phi_j(x) \Big|_{x_L}^{x_R} - \int_{x_L}^{x_R} f(\tilde{\omega}) \frac{d\phi_j(x)}{dx} dx = 0 \quad (2.14)$$

The local solution offers no way of recovering the global solution, until one realizes that the flux at the element boundaries is multiply defined between elements. The global solution is allowed to be piecewise discontinuous across boundaries, so there is no guarantee that the flux between neighboring elements would agree. To resolve this (and at the same time recover a global solution) define a numerical flux analogous

to a Finite Volume Method that takes as input the vorticity at the adjacent element boundaries. We will use an upwind flux, where defining the average as $\{\{\omega^+\}\} = \frac{\omega^+ + \omega^-}{2}$ and the jump as $[[\omega]] = \omega^+ - \omega^-$, yields

$$\hat{f}_{upwind}(x^+, x^-) = u\{\{\tilde{\omega}\}\} + \frac{|u|}{2}[[\tilde{\omega}]] \quad (2.15)$$

We would also like to map any element to a computational element by means of a mapping $x = g(X)$ and it's inverse $X = G(x)$. If we set the domain of our computational element to be $X \in [-1, 1]$ and the element size to be $\Delta x = x_R - x_L$, then the mapping is

$$x = g(X) = \frac{X+1}{2}\Delta x + x_L \quad , \quad X = G(x) = \frac{2(x - x_L)}{\Delta x} + 1 \quad (2.16)$$

Applying a change of variables using the mapping to (2.17), accounting for the numerical flux, and being sure to include the determinant of the Jacobian matrix that results from the mapping ($J = g'(X) = \Delta x/2$) in the first integral:

$$\frac{\Delta x}{2} \sum_{i=0}^M \left[\frac{da_i}{dt} \int_{-1}^1 \psi_i \phi_j dX \right] + \hat{f}\phi_j \Big|_{x_L}^{x_R} - \int_{-1}^1 f(\tilde{\omega}) \frac{d\phi_j}{dX} dX = 0 \quad (2.17)$$

(Note: no determinant of the Jacobian matrix appears in the second integral because one of the basis functions has been differentiated. When the chain rule is applied during the change of variables an extra $1/g'(X)$ appears that cancels out with the $g'(X)$ that occurred from the change of variables on the integral).

We have purposely left the flux in the second integral unevaluated in terms of the basis functions, as well as the choice of basis functions left as undecided. We defer these choices to a more in depth investigation in Methodology. We also have developed the equations past the initial PDE in a 1-D form. We will shortly show that the basis coefficients are chosen to be nodal values. The 2-D solution will be recovered following the Line-DG methodology [29] element will be composed of a tensor product of two 1-D discretization.

3 Methodology

3.1 Overall Solver Structure

IC and BC initialization, Lagrange pre-calculation, mass and stiffness matrices generation, 2 step spatial discretization, time discretization, post processing

3.2 Use and Abuse of Structured Tensor Mesh

3.3 Mass Matrix

3.4 Modified Quadrature

3.5 Stiffness Matrix Integration

3.6 Velocity Evaluation

3.7 Vortex Sparseness

3.8 Reduced Order Velocity (Global)

3.9 Reduced Order Velocity (Far-field)

3.10 Explicit Time-Stepping

3.11 Validation and Convergence of Proposed Methods

(analytical cases, elliptical blob, 2-D "ring", several interacting patches) vs (convergence rate of: constant order velocity field, linear far field velocity [far= 1 element or X element separation], variable far field velocity order [heuristic?])

The method will be validated via comparison with test cases for which analytical solutions exist (Perlman [?] test case, stationary isentropic vortex, elliptical patch [?]), as well a replication of empirical results from test cases [?]. The qualitative performance of the method will also be evaluated with more complex test cases, including the collision of two vortex pairs, "leap-frogging", and several arbitrary patch configurations.

4 Results

4.1 Analytical Test Cases

4.2 Elliptical Blob

4.3 Arbitrary Patch

4.4 Vortex Pair

4.5 Vortical System

4.6 Reduced Order Velocity

4.7 Runtime Speed and Memory Footprint of Solver

5 Discussion

5.1 Analytical Validation of Model

Comparison of existing analytical solutions vs model (Euler vortex, 5th order poly, other Saffmann test cases)

5.2 Empirical Convergence of Method

5.3 Computational Efficiency

5.4 Extended Body Fast Multipole Formulation

6 Conclusion

7 Recommendations

7.1 Extension to 3D

7.2 Flux Limiters

7.3 Impinging Geometry and Boundary Conditions

7.4 Viscosity

7.5 Implicit and Local Time Stepping

7.6 Parallelizability

7.7 FMM Improvements

7.8 2D Non-classical Quadrature for Nearly Singular Integrands

8 Implementation and Algorithms

8.1 Modified Two-part quadrature

Vectorized Lagrange evaluation, Lagrange derivatives, etc

8.2 Binary Singleton Expansion Matrix-Vector Products

8.3 Lagrange Bases and Constructed Quadrature

8.4 Velocity Evaluation

8.5 Semi-discrete System Construction

8.6 Low-storage Stability Optimized Runge-Kutta

9 Literature Cited

Bibliography

- [1] H. J. Lugt, Vortex Flow in Nature and Technology, Krieger Publishing Company, Malabar, FL, USA, 1983.
- [2] P. G. Saffman, Vortex Dynamics, Cambridge Univ. Press, Cambridge, UK, 1992.
- [3] C. G. Speziale, On the advantages of the vorticity-velocity formulation of the equations of fluid dynamics, J. Comput. Phys. 73 (1987) 476-480.
- [4] L. Rosenhead, The formation of vortices from a surface of discontinuity, Proc. Roy. Soc. London Ser. A 134 (1931) 170-192.
- [5] G. Birkhoff, J. Fisher, Do vortex sheets roll up?, Rend. Circ. Math. Palermo, Ser. 2 8 (1959) 77-90.
- [6] F. H. Abernathy, R. E. Kronauer, The formation of vortex streets, J. Fluid Mech. 13 (1962) 1-20.
- [7] A. J. Chorin, P. S. Bernard, Discretization of a vortex sheet, with an example of roll-up, J. Comput. Phys. 13 (3) (1973) 423-429.
- [8] K. Kuwahara, H. Takami, Numerical studies of two-dimensional vortex motion by a system of point vortices, J. Physical Society of Japan 34 (1) (1973) 247-253.
- [9] R. G. Zalosh, Discretized simulation of vortex sheet evolution with buoyancy and surface tension effects, AIAA Journal 14 (11) (1976) 1517-1523.
- [10] G. K. Batchelor, An Introduction to Fluid Dynamics, Cambridge University Press, 1967, 1973.

- [11] W. T. Ashurst, E. Meiburg, Three-dimensional shear layers via vortex dynamics, *J. Fluid Mech.* 189 (1988) 87-116.
- [12] J. E. Martin, E. Meiburg, Numerical investigation of three-dimensional evolving jets subject to axisymmetric and azimuthal perturbation, *J. Fluid Mech.* 230 (1991) 271-318.
- [13] A. Leonard, Numerical simulation of interacting, three-dimensional vortex filaments, in: *Proceedings of the IV Intl. Conference on Numerical Methods of Fluid Dynamics*, no. 35 in *Lecture Notes in Physics*, Springer-Verlag, 1975, pp. 245-250.
- [14] M. E. Agishtein, A. A. Migdal, Dynamics of vortex surfaces in three dimensions: Theory and simulations, *Physica D* 40 (1989) 91-118.
- [15] O. M. Knio, A. F. Ghoniem, Three-dimensional vortex simulation of rollup and entrainment in a shear layer, *J. Comput. Phys.* 97 (1991) 172-223.
- [16] O. M. Knio, A. F. Ghoniem, Vortex simulation of a three-dimensional reacting shear layer with finite-rate kinetics, *AIAA J.* 30 (1) (1992) 105-116.
- [17] G. Russo, J. A. Strain, Fast triangulated vortex methods for the 2D Euler equations, *J. Comput. Phys.* 111 (1994) 291-323.
- [18] M. Carley, A triangulated vortex method for the axisymmetric Euler equations, *J. Comput. Phys.* 180 (2002) 616-641.
- [19] S. A. Huyer, J. R. Grant, Solution of two-dimensional vorticity equation on a Lagrangian mesh, *AIAA Journal* 38 (5) (2000) 774-783.
- [20] J. Strain. 2D vortex methods and singular quadrature rules. *Journal of Computational Physics* 124.1 (1996): 131-145.

- [21] J. T. Beale, On the accuracy of vortex methods at large times, in: IMA Workshop on Computational Fluid Dynamics and Reacting Gas Flows, Springer-Verlag, 1988, p. 19.
- [22] J. S. Marshall, J. R. Grant, Penetration of a blade into a vortex core: vorticity response and unsteady blade forces, *J. Fluid Mech.* 306 (1996) 83-109.
- [23] H. O. Nordmark, Rezoning for higher order vortex methods, *J. Comput. Phys.* 97 (1991) 366-397.
- [24] H. N. Najm, R. B. Milne, K. D. Devine, S. N. Kempa, A coupled Lagrangian-Eulerian scheme for reacting flow modeling, *ESAIM Proc.* 7 (1999) 304-313.
- [25] O. H. Hald, V. D. Prete, Convergence of vortex methods for Euler's equations, *Math. Comput.* 32 (1978) 791-809.
- [26] J. T. Beale, A convergent 3-D vortex method with grid-free stretching, *Math. Comput.* 46 (174) (1986) 401-424.
- [27] R.E. Brown. Rotor Wake Modeling for Flight Dynamic Simulation of Helicopters. *AIAA Journal*, 2000. Vol. 38(No. 1): p. 57-63.
- [28] A.J. Line, R.E. Brown. Efficient High-Resolution Wake Modelling using the Vorticity Transport Equation. in 60th Annual Forum of the American Helicopter Society. 2004. Baltimore, MD.
- [29] P.O. Persson. A Sparse and High-Order Accurate Line-Based Discontinuous Galerkin Method for Unstructured Meshes. *J. Comp. Phys.*, Vol. 233, pp. 414-429, Jan 2013.
- [30] J. Strain. Fast adaptive 2D vortex methods. *Journal of computational physics* 132.1 (1997): 108-122.

- [31] K. Lindsay, and R. Krasny. A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow. *Journal of Computational Physics* 172.2 (2001): 879-907.
- [32] L. Rosenhead. The spread of vorticity in the wake behind a cylinder, *Proc. Roy. Soc. London Ser. A* 127, 590 (1930).
- [33] D. W. Moore. Finite amplitude waves on aircraft trailing vortices, *Aero. Quart.* 23, 307 (1972).
- [34] M. Perlman. On the accuracy of vortex methods, *J. Comput. Phys.* 59 (1985) 200–223.
- [35] P. Koumoutsakos. Inviscid axisymmetrization of an elliptical vortex, *J. Comput. Phys.* 138 (1997) 821–857.
- [36] Koumoutsakos, P., Leonard, A. (1995). High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *Journal of Fluid Mechanics*, 296, 1-38.
- [37] K. B. Southerland, J. R. P. III, W. J. A. Dahm, K. A. Buch. An experimental study of the molecular mixing process in an axisymmetric laminar vortex ring, *Phys. Fluids A* 3 (5) (1991) 1385–1392.
- [38] Moussa, C., Carley, M. J. (2008). A Lagrangian vortex method for unbounded flows. *International journal for numerical methods in fluids*, 58(2), 161-181.
- [39] J. Steinhoff, D. Underhill, Modification of the Euler equations for "vorticity confinement": Application to the computation of interacting vortex rings, *Phys. Fluids* 6 (8) (1994) 2738-2743.

- [40] Schumann, U., Sweet, R. A. (1976). A direct method for the solution of Poisson's equation with Neumann boundary conditions on a staggered grid of arbitrary size. *Journal of Computational Physics*, 20(2), 171-182.
- [41] Barnes, J., Hut, P. (1986). A hierarchical $O(N \log N)$ force-calculation algorithm.
- [42] Greengard, L., Rokhlin, V. (1987). A fast algorithm for particle simulations. *Journal of computational physics*, 73(2), 325-348.