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

Master's Defense

J. Bevan

Advisor: Dr. David J. Willis, Assistant Professor

Department of Mechanical Engineering
University of Massachusetts at Lowell

June 3, 2015

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

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.

Introduction: Presentation Structure

- Introduction
- Theory
- Methodology
- Implementation
- Results
- Discussion
- Conclusion

Motivation

- Direct solution of Navier-Stokes impractical for many fluid problems
- Vorticity-velocity formulation well-suited for inviscid, incompressible, vortex dominated flows
- Lagrangian vortex methods are common approach[1][2], but face several challenges [3]
- Initial vortex points become disorganized, re-meshing etc. required
- Eulerian approach avoids disorganization, extendable to high order
- Brown et al. successful with Eulerian approach for low order FVM[4]suggests high-order extension possible

[1] J. Strain. Fast adaptive 2D vortex methods. Journal of computational physics 132.1 (1997): 108-122.

[2] Moussa, C., Carley, M. J. (2008). A Lagrangian vortex method for unbounded flows. International journal for numerical methods in fluids, 58(2), 161-181.

[3] J. Strain. 2D vortex methods and singular quadrature rules. Journal of Computational Physics 124.1 (1996): 131-145.

[4] R.E. Brown. Rotor Wake Modeling for Flight Dynamic Simulation of Helicopters. AIAA Journal, 2000. Vol. 38(No. 1): p. 57-63.

Proposed Method

- Eulerian representation of velocity and vorticity
- Solution of inviscid velocity-vorticity PDE
- Velocity evaluation via eval of Biot-Savart kernel[5]
- Vorticity advection via a line-DG[6]approach
- Method of lines, explicit time-stepping with Runge-Kutta[7]

[5] Winckelmans, G. S., and A. Leonard. Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows. Journal of Computational Physics 109.2 (1993): 247-273.

[6] 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.

[7] Niegemann, Jens, Richard Diehl, and Kurt Busch. Efficient low-storage Runge–Kutta schemes with optimized stability regions. Journal of Computational Physics 231.2 (2012): 364-372.

Thesis Goals

Goals:

- Development of a high-order solver for inviscid incompressible vorticity-dominated flows in 2D
  - High-order advective solver capable of mixed order flux handling
  - High-order Biot-Savart evaluation routine

Contributions:

- Complete high-order method for velocity-vorticity inviscid flow
- Validation of solver and underlying Eulerian vortex approach
- Evaluation of convergence, error, and performance of method and solver

Theory: Navier-Stokes: Velocity-vorticity form

Navier-Stokes momentum equation

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\nabla^2\mathbf{u} + \tfrac{1}{3}\,\mu\nabla(\nabla\cdot\mathbf{u}) \tag{1}$$

where u is the velocity field, p is the pressure field, and ρ is the density. Define *vorticity* as

$$\omega = \nabla\times\mathbf{u} \tag{2}$$

Navier-Stokes can be recast as

$$\frac{\partial\omega}{\partial t} + \mathbf{u}\cdot\nabla\omega - \omega\cdot\nabla\mathbf{u} = S(x,t) \tag{3}$$

viscous generation of vorticity, S

Velocity-vorticity form

Advantages:

- Explicit conservation of vorticity.
- Frequently distribution of the vorticity is sparse.
- No pressure term.

Simplified in 2D, vortex stretching zero. Express in terms of vortex flux $f_i(\omega) = u_i\,\omega$

$$\frac{\partial \omega}{\partial t} + \frac{\partial f}{\partial x_i} = S(x,t) \tag{4}$$

Velocity Evaluation

For incompressible flows velocity related to vorticity by

$$\nabla^2 u = -\nabla \times \omega \tag{5}$$

Invert to obtain Biot-Savart integral

$$u(x^*) = \int_\Omega K(x^*, x) \times \omega(x) dx \tag{6}$$

x^* is velocity eval point, x is non-zero vorticity domain, $K(x^*, x)$ singular Biot-Savart kernel[8]

$$K(x^*, x) = \frac{-1}{2\pi} \frac{x^* - x}{|x^* - x|^2} \tag{7}$$

[8] Beale, J. Thomas, and Andrew Majda. High order accurate vortex methods with explicit velocity kernels. Journal of Computational Physics 58.2 (1985): 188-208.

Kernel De-singularization

- Singularity in Biot-Savart kernel generates non-physical velocities near vortex points, de-singularize by approximation of Dirac delta function using finite cutoff radius δ.
- Many de-singularized kernels possible, in Lagrangian method proofs must satisfy various "moment conditions". Main one used in this thesis is the spectral kernel[5]:

$$K_{spectral_0} = \frac{z}{2\pi|z|^2}(1 - J_0(\frac{z}{\delta})) \tag{8}$$

where J_0 is a Bessel function of the first kind and z is a radial distance.

- Infinite order kernel that preserves all moments of Dirac delta

Discontinuous Galerkin (DG)

We seek to solve the PDE Eqn. (4). An approximate solution $\widetilde{\omega}$ has residual

$$\frac{\partial \widetilde{\omega}}{\partial t} + \frac{\partial f}{\partial x_i} = R(x) \tag{9}$$

1-D case, vorticity sources omitted for simplicity.
DG approach[9]: minimize the L^2 norm by orthogonal projection of residual onto approximating space. Complete basis made from test functions ϕ_j, so:

$$\int_\Omega R(x)\phi_j \, dx = 0 \quad \text{for all } j \tag{10}$$

Substituting residual with conservation PDE yields:

$$\int_\Omega \frac{\partial \widetilde{\omega}}{\partial t} \phi_j \, dx + \int_\Omega \frac{\partial f(\widetilde{\omega})}{\partial x} \phi_j \, dx = 0 \quad \text{for all } j \tag{11}$$

[9] W.H Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. 1973.

Discontinuous Galerkin (DG)(cont.)

Use same space for both test functions and and approximation, so Mth order vorticity approximation is

$$\omega(x,t) \approx \widetilde{\omega}(x,t) = \sum_{i=0}^{M} a_i(t)\psi_i(x) \tag{12}$$

Substitute into Eqn. (11) and integrate by parts second term:

$$\sum_{i=0}^{M}\left[\frac{da_i(t)}{dt}\int_{x_L}^{x_R}\psi_i(x)\,\phi_j(x)\,dx\right]+f\phi_j(x)\Big|_{x_L}^{x_R}-\int_{x_L}^{x_R} f(\widetilde{\omega})\,\frac{d\phi_j(x)}{dx}\,dx=0 \tag{13}$$

Note: Local solution to PDE on an element.

Discontinuous Galerkin (DG)(cont.)

All local solutions decoupled, also vorticity multiply defined at overlapping element boundaries. Recover global solution and treat element boundaries via an upwind flux function[10](similar to finite volume method)

$$\hat{f}_{upwind}(x^+, x^-) = u\{\{\widetilde{\omega}\}\} + \frac{|u|}{2}[[\widetilde{\omega}]] \tag{14}$$

where $\{\{\omega^+\}\} = \frac{\omega^+ + \omega^-}{2}$ and $[[\omega]] = \omega^+ - \omega^-$

Applying change of variables to map to arbitrary computational element $X \in [-1, 1]$ results in:

$$\frac{\Delta x}{2} \sum_{i=0}^{M} \left[\frac{da_i}{dt} \int_{-1}^{1} \psi_i \, \phi_j \, dX \right] + \hat{f}\phi_j \Big|_{x_L}^{x_R} - \int_{-1}^{1} f(\widetilde{\omega}) \, \frac{d\phi_j}{dX} \, dX = 0 \tag{15}$$

[10] Hesthaven, Jan S., and Tim Warburton. Nodal discontinuous Galerkin methods: algorithms, analysis, and applications. Vol. 54. Springer Science & Business Media, 2007.

Solver Overview

```
Define problem parameters
Define solver parameters
Calculate derived solver parameters
Setup intitial conditions
Initialize solver
%Time stepping
for t=0 to end
   if datalog?=yes
      save system state to file and plot
   end
   %Loop through RK stages
   for s=1 to last_stage
      %For elements above threshold
      for each vorticity source
         calculate velocity contributions
      end
      %Calculate semi-discrete system terms
      interpolate boundary_vorticity
      calculate numerical_fluxes
      calculate total_surface_flux
      calculate internal_stiffness_flux

      vorticity_rate_of_change=...
         internal_stiffness_flux - total_surface_flux

      RK_stage= (RK_coeff_a*RK_stage) +...
         (time_step * vorticity_rate_of_change)
      vorticity= vorticity + RK_coeff_b * RK_stage
   end
end
```

Methodology: Method Specific Choices

- Choose basis functions to be the interpolating Lagrange polynomials

$$\psi_i(x) = \ell_i(x) = \prod_{\substack{p=0 \\ p \neq i}}^{M} \frac{x - x_p}{x_i - x_p} \tag{16}$$

- Choose vorticity interpolation nodes to be the Gauss-Legendre points, collocate with quadrature points, results in simplification of mass matrix

$$\int \ell_i(x)\ell_j(x)dx = \delta_{ij} w_j \tag{17}$$

- Take line-DG[6] approach, form 2D basis as tensor product of 1D bases

$$f(x,y) \approx \left[\sum_{j=0}^{L} z_j \ell_j(y)\right] \times \left[\sum_{i=0}^{M} z_i \ell_i(x)\right] = \sum_{j=0}^{M}\sum_{i=0}^{M} z_{ij}\ell_j\ell_i = \sum_{j=0}^{M} z_{ij}\ell_j \tag{18}$$

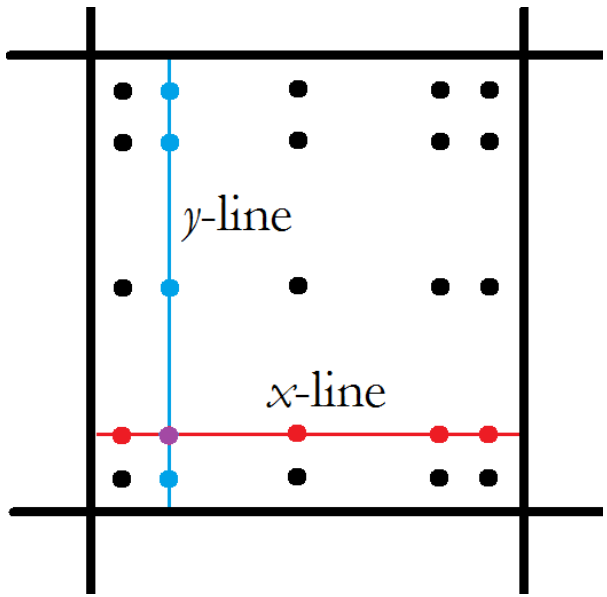
Method Specific Choices(cont.)

- The PDE is now solved along each tensor direction

$$\frac{\Delta x}{2}\sum_{i=0}^{M}\left[\frac{dz_{ij}}{dt}\int_{-1}^{1}\ell_i\,\ell_j\,dX\right]+\hat{f}\ell_j\Big|_{x_L}^{x_R}-\int_{-1}^{1}f(\widetilde{\omega})\,\ell_j'\,dX=0 \tag{19}$$

- The rate of change at each node is the sum of the contribution along each tensor direction

$$\frac{\partial\omega_{ij}}{\partial t}=(\frac{\partial\omega_{ij}}{\partial t})_{x-line}+(\frac{\partial\omega_{ij}}{\partial t})_{y-line} \tag{20}$$



Flux Interpolation and Modified Quadrature

- Interpolate flux via product of interpolations of vorticity and velocity (rather than interpolation of their product)

$$\int u(x)\omega(x)\ell_j'(x)\,dx \approx \int \sum_i^M u_i\ell_i(x) \sum_k^M \omega_k\ell_k(x)\,\ell_j'(x)\,dx$$
$$= \sum_i^M u_i \sum_k^M \omega_k \int \ell_i(x)\ell_k(x)\ell_j'(x)\,dx \tag{21}$$

- To do so, generate a matrix of modified quadrature weights

$$\int \ell_i(x)\ell_k(x)\ell_j'(x)\,dx = \mathbf{W}_{ikj} \tag{22}$$

- The resultant quadrature rule is now

$$\int u(x)\omega(x)\ell_j'(x)\,dx \approx \mathbf{u}_i^T\mathbf{W}_{ikj}\,\boldsymbol{\omega}_k \tag{23}$$

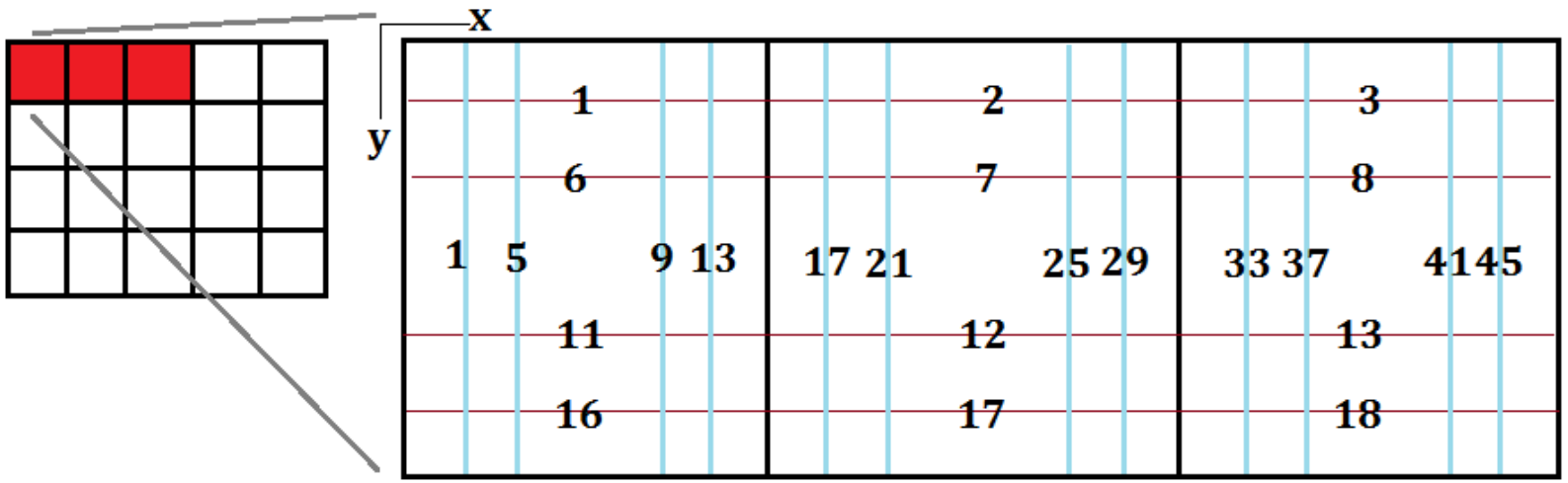
for each jth weighting basis function.

Modified Quadrature Advantages

- Able to extract *more* information out of *same* number of interpolation points with superior convergence of stiffness integral compared to standard Gauss-Legendre rule (integrate N point vorticity/velocity- 2N+2 order interpolation vs. N+1 order interpolation exactly)
- Including derivative of the weighting basis in integral for quadrature weight matrix means integrated exactly *a priori*
- Order and definition of the Lagrange basis functions for vorticity and velocity interpolations are *decoupled*
- Able to use Gauss-Legendre points for vorticity and Lobatto for velocity (avoids extra boundary velocity calcs)

Implementation: Structured Tensor Mesh

- Vorticity and velocity stored "line-wise" along tensor directions per element.
- Advection and velocity calcs operate line-wise



Mass Matrix

Making use of Eqn. (17) the mass matrix

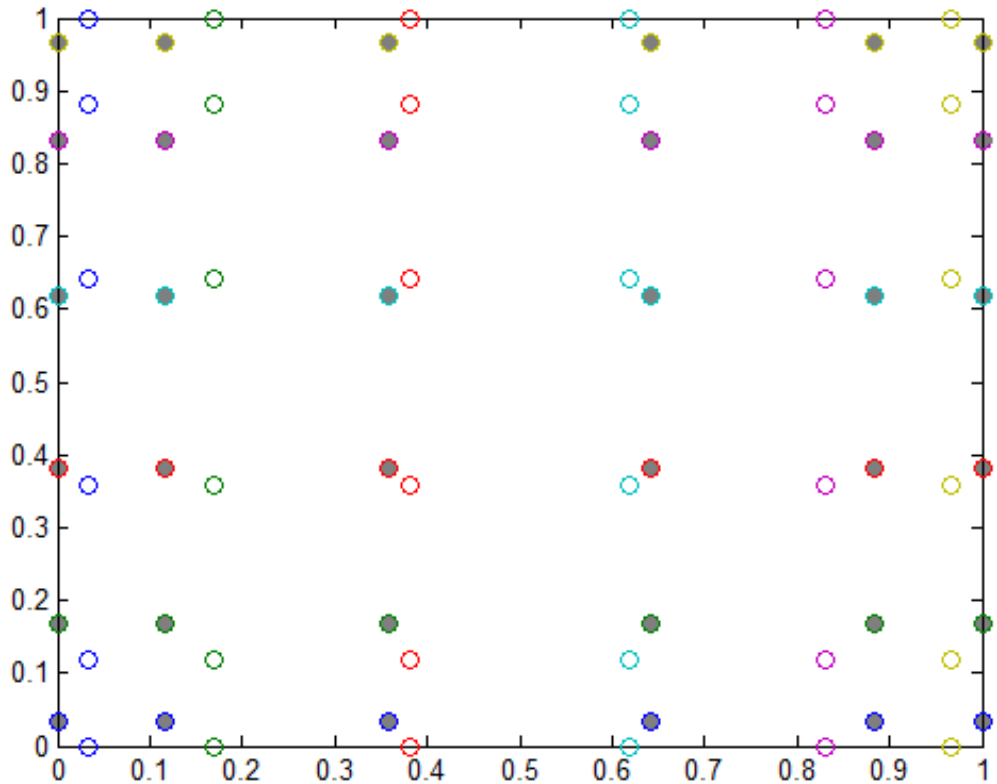
$$\boldsymbol{M} = \frac{\Delta x}{2} \sum_{i=0}^{M} \left[\frac{dz_{ij}}{dt} \int_{-1}^{1} \ell_i \, \ell_j \, dX \right] \tag{24}$$

is diagonalizable. We can simplify Eqn. (19):

$$\frac{d\widetilde{\omega}_j}{dt} \frac{w_j \Delta x}{2} + \hat{f} \ell_j \Big|_{x_L}^{x_R} - \int_{-1}^{1} f(\widetilde{\omega}) \, \ell_j' \, dX = 0 \tag{25}$$

Stiffness Matrix

- Velocity nodes for a particular line don't coincide with vorticity nodes, but are still co-linear
- Only the velocity component directed along the line is needed at a particular velocity node



Stiffness Matrix(cont.)

- Apply modified quadrature rule to the stiffness term

$$\boldsymbol{K} = \int_{-1}^{1} f(\widetilde{\omega})\, \ell_j'\, dX \tag{26}$$

yields:

$$\frac{d\widetilde{\omega}_j}{dt}\frac{w_j \Delta x}{2} + \hat{f}\ell_j\Big|_{x_L}^{x_R} - \mathbf{u}^T \mathbf{W}_{--j}\, \boldsymbol{\omega} = 0 \tag{27}$$

- Rearrange to obtain ODE in time:

$$\frac{d\widetilde{\omega}_j}{dt} = \frac{2}{w_j \Delta x}\left[\mathbf{u}^T \mathbf{W}_{--j}\, \boldsymbol{\omega} - \hat{f}\ell_j\Big|_{x_L}^{x_R}\right] \tag{28}$$

Velocity Evaluation

- Calculate velocity at all velocity nodes by summation of contribution from all vorticity using de-singularized kernel and Gauss-Legendre quadrature

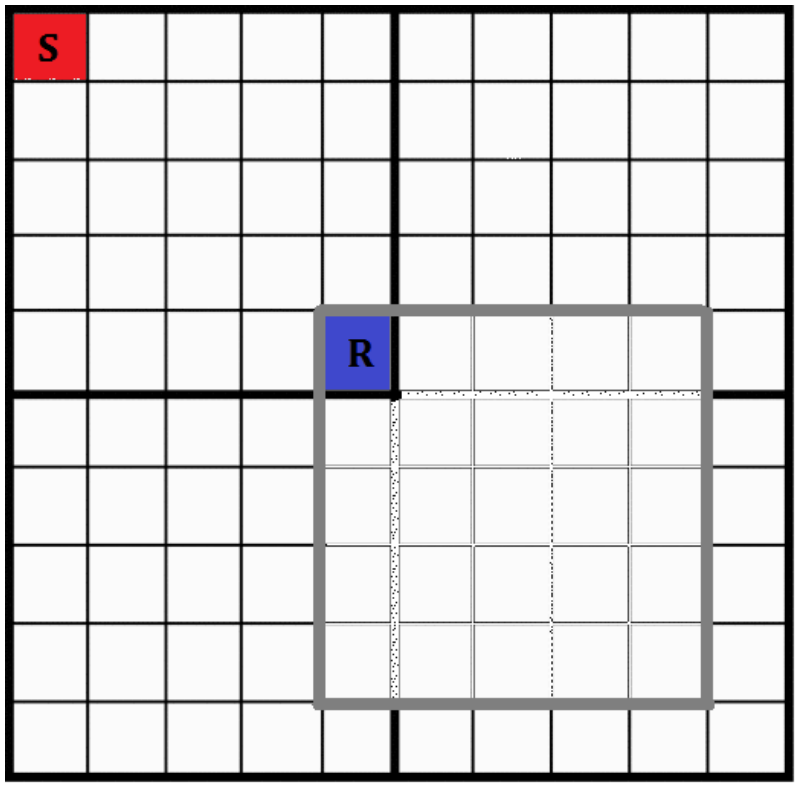
$$u(x^*) = \sum_{E=1}^{N_{mask}} [\omega_{pre}]^T K_\delta(x^* - [x_E]) \tag{29}$$

$$\omega_{pre} = [\omega(x_E)] . * [w_i \otimes w_j] \tag{30}$$

- Pre-multiplication of particular elemental vorticity source by outer product of Gauss-Legendre quadrature weights and computing as vector-vector product saves great deal of computational effort (2000% in current code)
- Kernel values pre-calculated for "generalized" reference frame

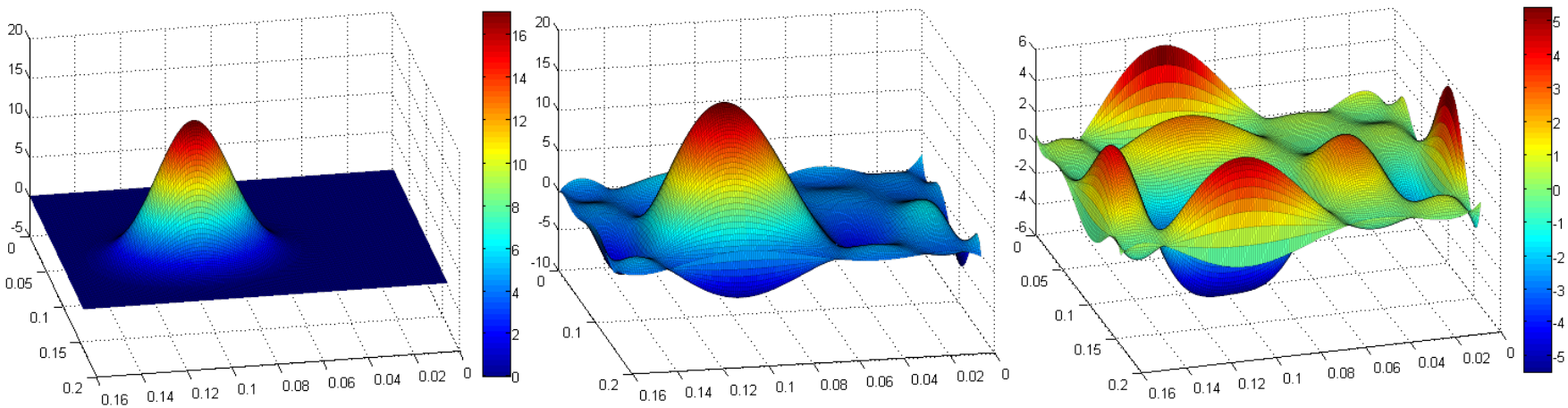
Kernel value calculations

- Constant re-calculation of kernel values computationally expensive, but storing all values at start not possible due to memory needs scaling as Np^4K^4
- Solution: calculate for generalized reference frame and map as needed for each vorticity element source



Convergence of Biot-Savart Integral

- Nearly singular nature of Biot-Savart kernel difficult to integrate numerically
- Smaller the cutoff radius, larger quadrature errors
- Larger cutoff radius, larger velocity approximation errors
- Cutoff radius should be selected to balance both



Explicit Time-Stepping

- Method of lines approach to semi-discrete system
- Low-storage explicit Runge-Kutta method used
- 14 stage-4th order "NRK14C" used to maximize stable time-step[7]
- Stability region almost 1.9 times larger per stage along negative real axis, chief consideration for dissipative upwind DG schemes

```
for t=0:delt:EndTime %Step
    for i=1:nS %Stage
        (Velocity calculations)
        (semi-discrete calculations for advection)
        wx_dt= permute(Stiff_x-SurfFlux_x,[4 1 3 2]);
        wy_dt= reshape(reshape(...
Stiff_y-SurfFlux_y,K(2),[])',Np,1,[]);

        k2= RKa(i)*k2 + delt*(wx_dt+wy_dt);
        wx= wx+RKb(i)*k2;
        wy= reshape(reshape(wx,K(1)*Np,[])',Np,1,[]);
    end
end
```

Results: Validating Test Cases

- Perlmann: Stationary vortex[11]

$$\omega(z) = (1 - |z|^2)^7, \ |z| \leq 1 \quad \omega(z) = 0, \ |z| > 1 \quad z^2 = x^2 + y^2 \tag{31}$$

- Has analytical solution to vorticity and velocity fields

$$u(z,t) = f(|z|)\begin{pmatrix} y \\ -x \end{pmatrix} \tag{32}$$

where

$$f(|z|) = \begin{cases} -\frac{1}{16|z|^2}(1 - (1 - |z|^2)^8) & |z| \leq 1 \\ -\frac{1}{16|z|^2} & |z| > 1 \end{cases}$$

[11] M. Perlman. On the accuracy of vortex methods, J. Comput. Phys. 59 (1985) 200–223.

Validating Test Cases(cont.)

- Strain: Interacting vortex patches[12]

$$\omega(x, y, 0) = \sum_{j=1}^{m} \Omega_j exp(-((x - x_j)^2 + (y - y_j)^2)/\rho_j^2) \quad (33)$$

Table : Interacting Vortex Patch Parameters

j	x_j	y_j	ρ_j	Ω_j
1	-0.6988	-1.7756	0.6768	-0.4515
2	1.4363	-1.4566	0.3294	0.4968
3	-0.1722	0.4175	0.5807	-0.9643
4	-1.5009	-0.0937	0.2504	0.3418

[12] J. Strain. 2D vortex methods and singular quadrature rules. Journal of Computational Physics 124.1 (1996): 131-145.

Validating Test Cases(cont.)

- Koumoutsakos: Elliptical vortex[13]

$$\omega^{II}(x,y,0)_{mod} = 20(1-((x/a)^2+(y/b)^2)^2/0.8^4) \quad a=1,\ b=2 \tag{34}$$

[13] P. Koumoutsakos. Inviscid axisymmetrization of an elliptical vortex, J. Comput. Phys. 138 (1997) 821–857.

Comparison of Biot-Savart Kernel Effects(P)

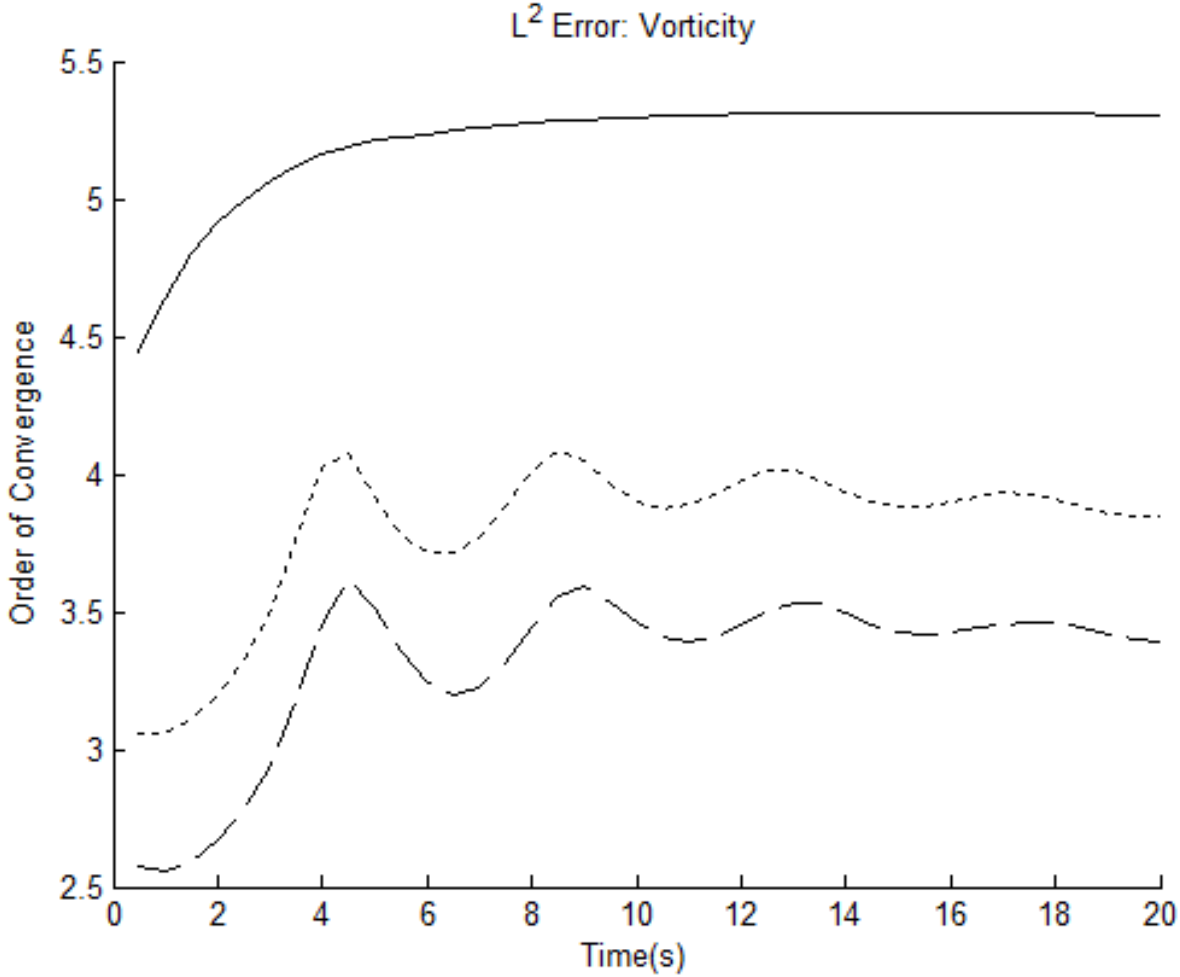


Figure : Comparison of convergence effects on vorticity by kernel for Spectral(-), WL(..), and RM(- -) kernels, 6th order method.

Comparison of Biot-Savart Kernel Effects(P)(cont.)

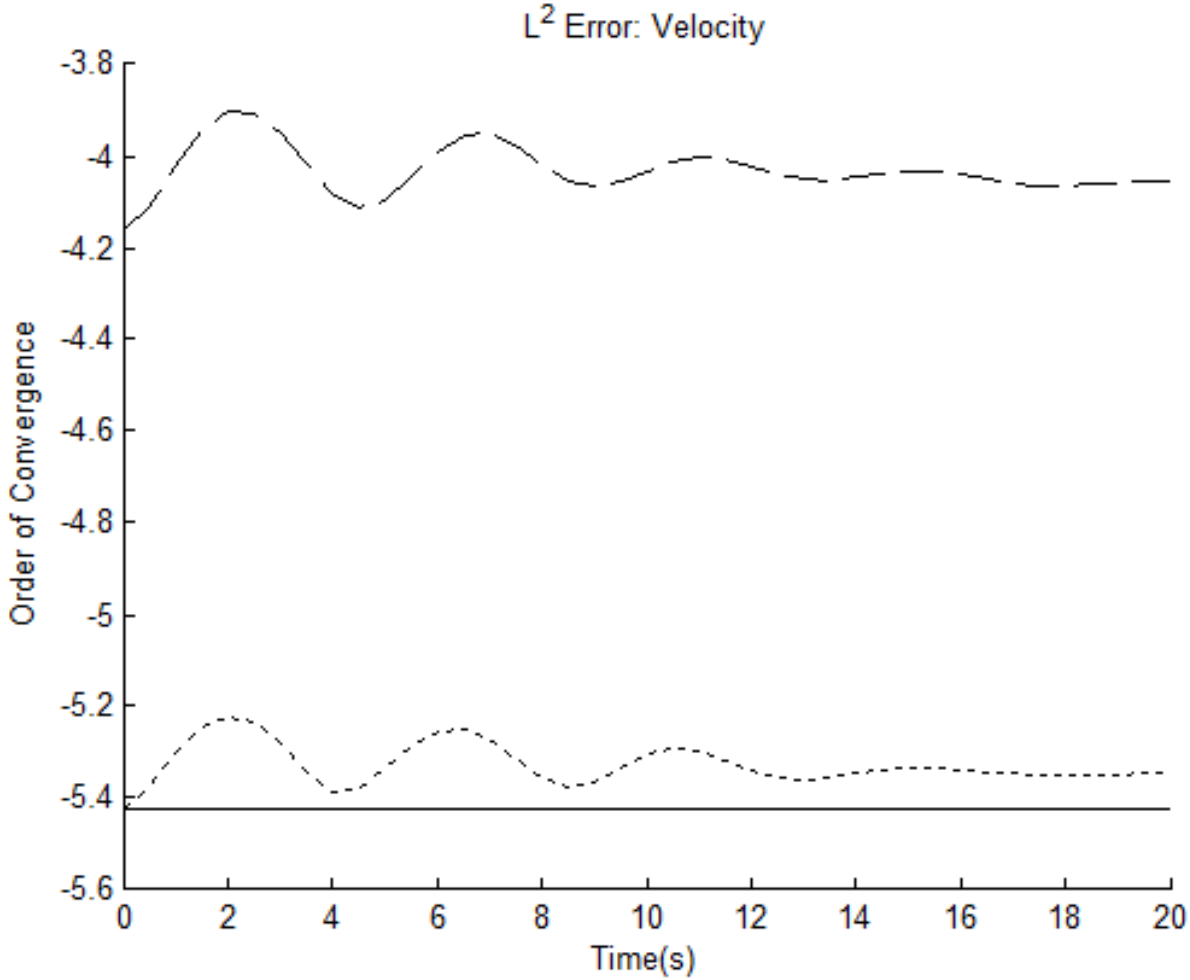


Figure : Comparison of convergence effects on velocity by kernel for Spectral(-), WL(..), and RM(- -) kernels, 6th order method.

Comparison of Cutoff Radius Effects(P)

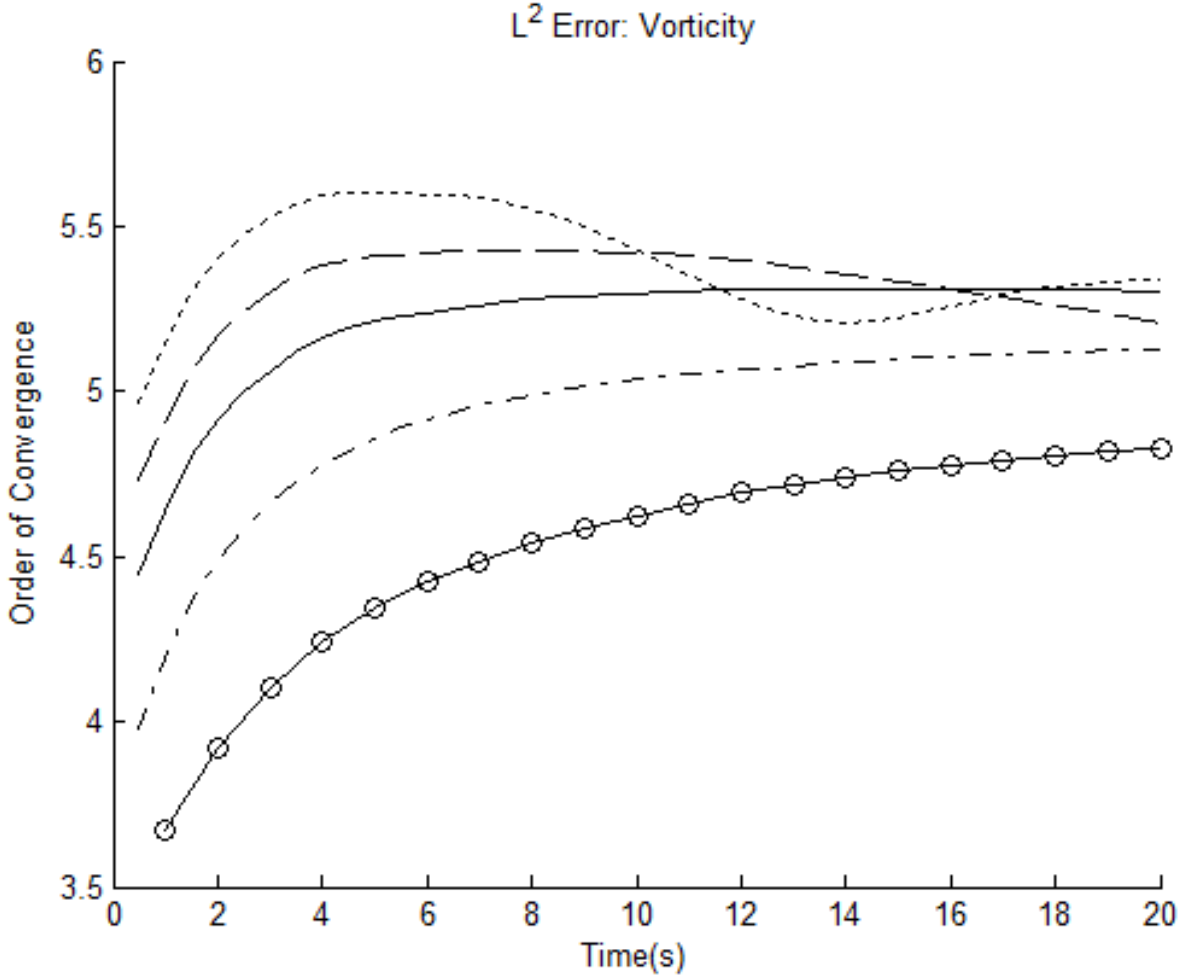


Figure : Comparison of convergence effects on vorticity by cutoff radius for $\delta/\Delta x$=0.1(..), 0.2(- -), 0.3(-), 0.5(-.), 0.9(-o).

Comparison of Cutoff Radius Effects(P)(cont.)

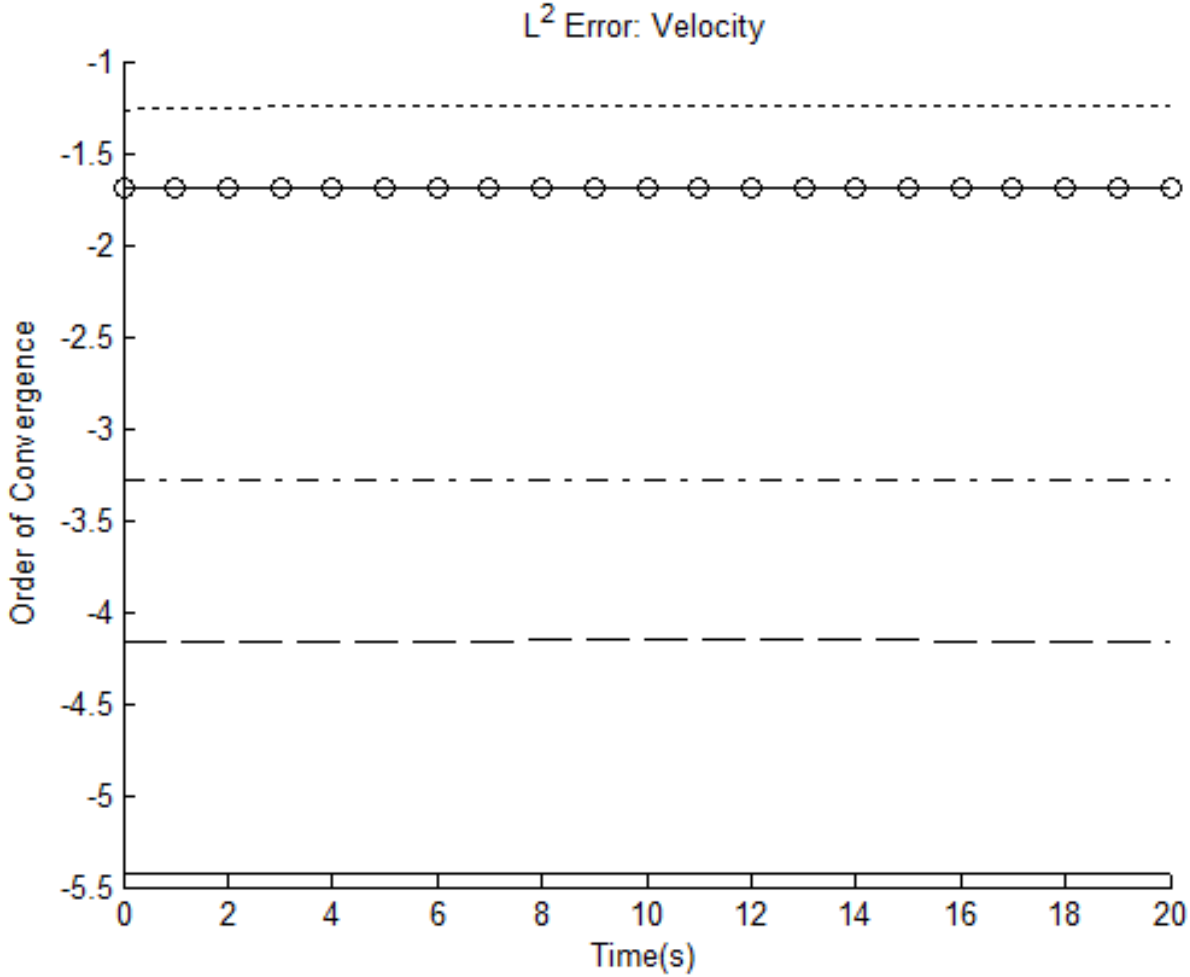


Figure : Comparison of convergence effects on velocity by cutoff radius for $\delta/\Delta x$=0.1(..), 0.2(- -), 0.3(-), 0.5(-.), 0.9(-o).

Comparison of Reduced Order Velocity(P)

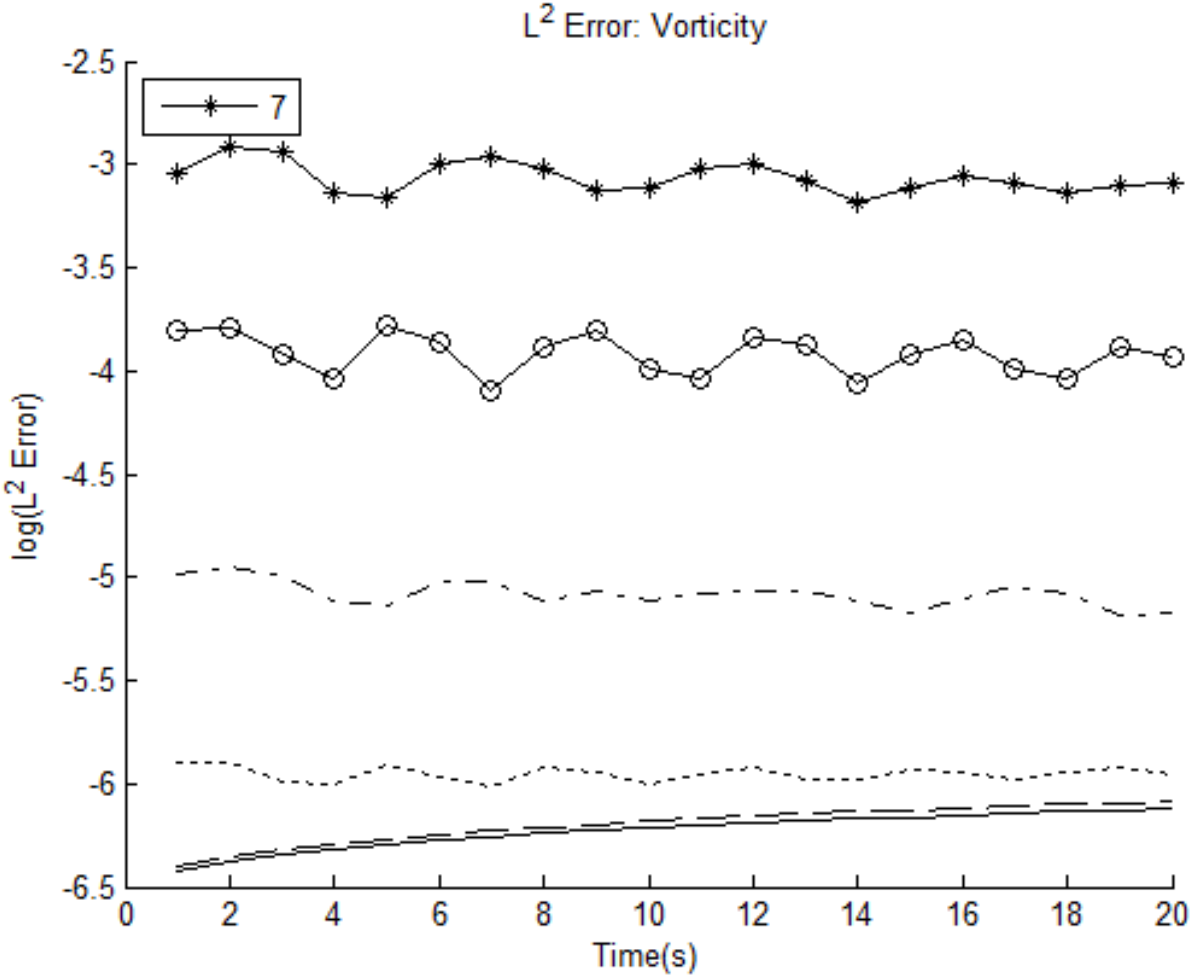


Figure : L^2 error in vorticity for velocity order: 2(-*), 3(-o), 4(-.), 5(..), 6(- -), and 7(-).

Comparison of Reduced Order Velocity(P)(cont.)

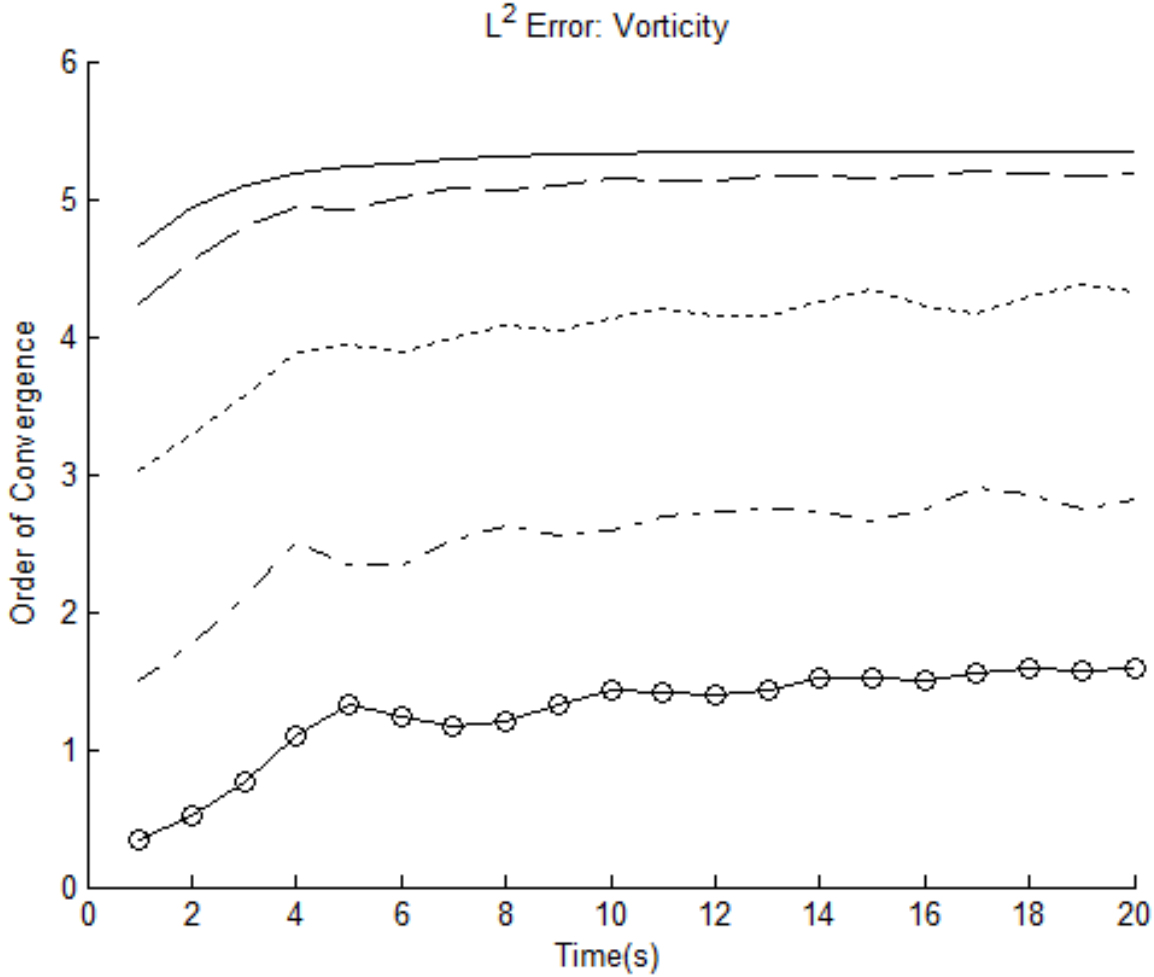


Figure : Comparison of convergence rate for reduced velocity order experiment, velocity order: 2(-o), 3(-.), 4(..), 5(- -), 6/7(-).

Observed Convergence Rate of Various Orders(P)

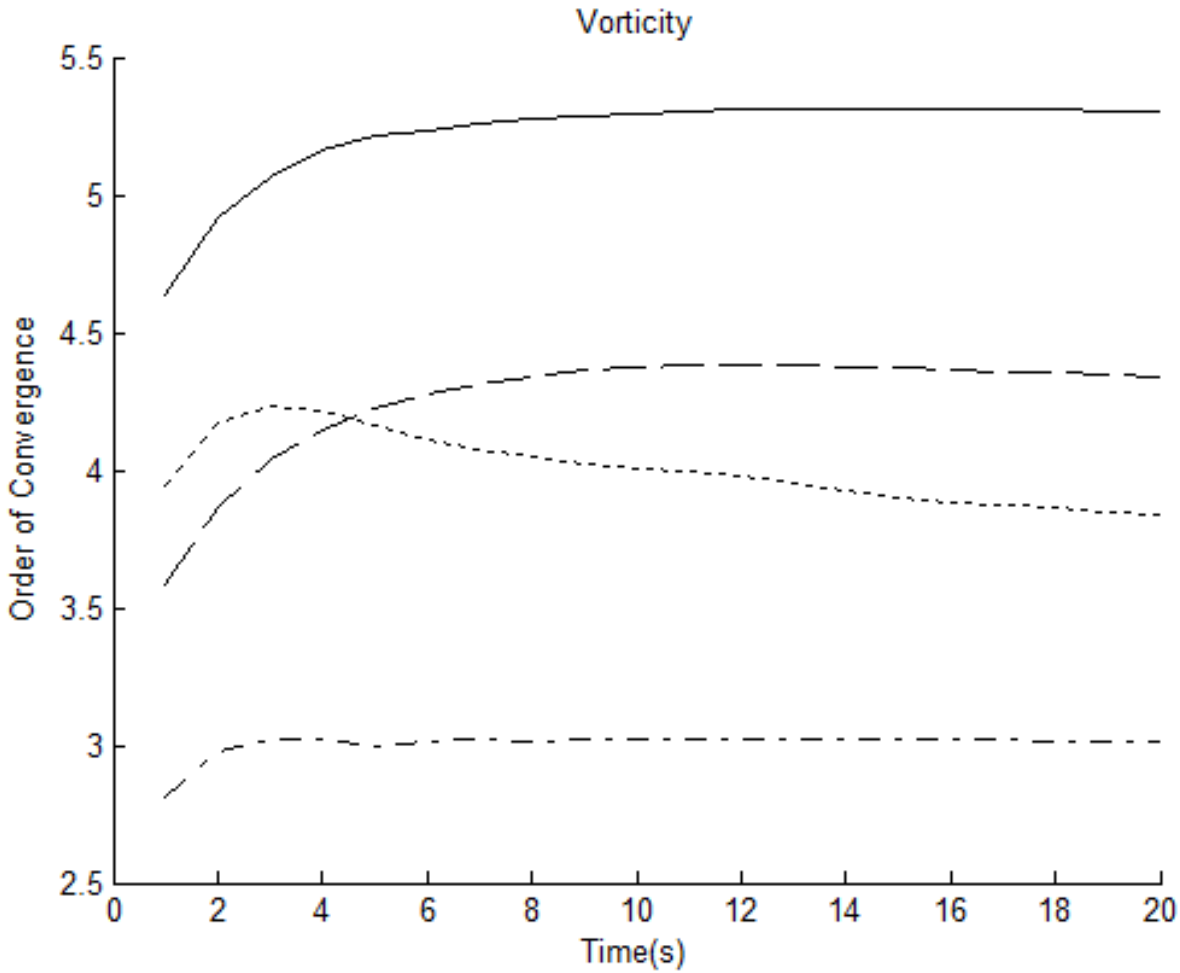


Figure : Comparison of convergence rate for various order methods: 3rd(-.), 4th(..), 5th(- -), and 6th(-).

Conservation of Vorticity(P)

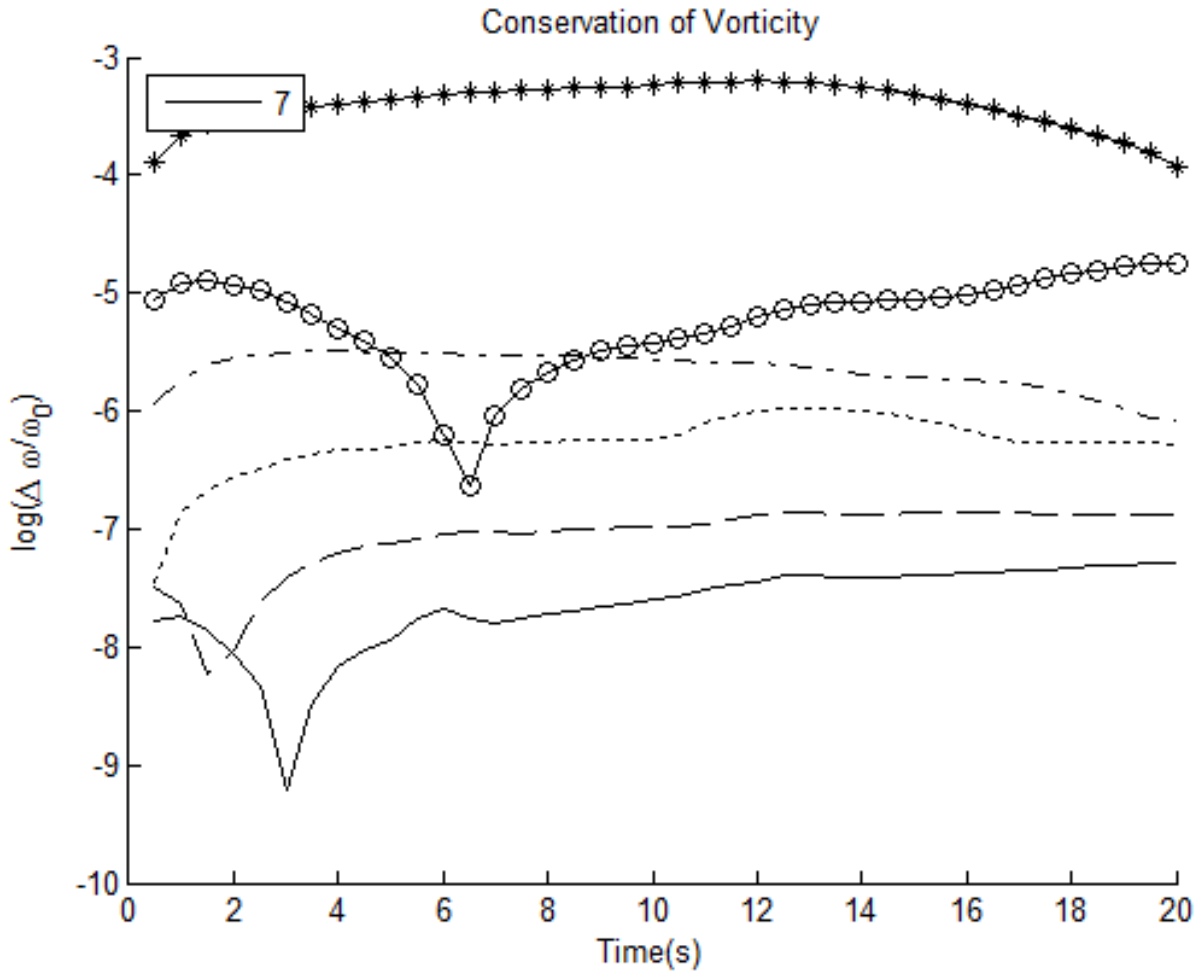


Figure : Conservation of vorticity in a 6th order method, K×K elements: K=2(-*), 3(-o), 4(-.), 5(..), 6(- -), and 7(-).

Conservation of Linear Impulse(P)

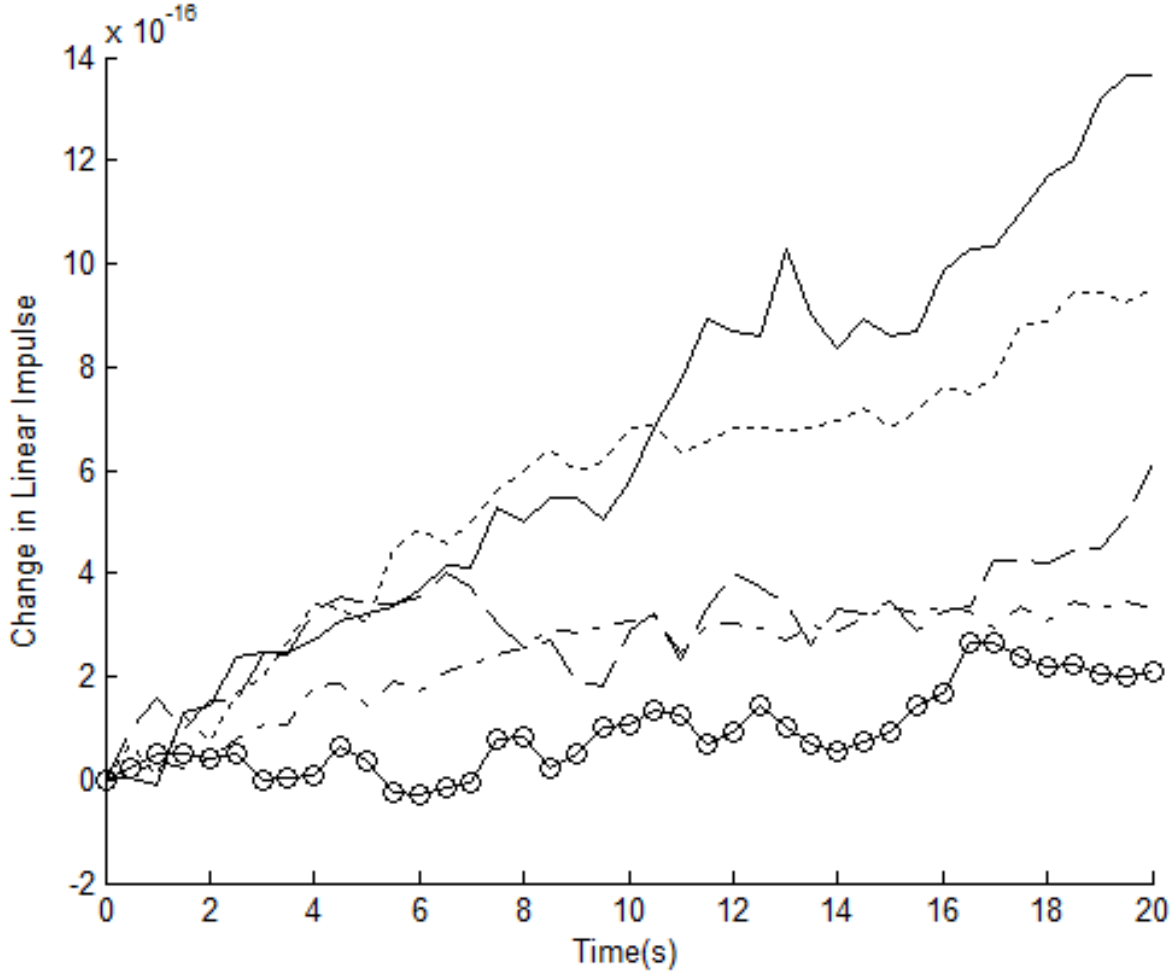


Figure : Conservation of linear impulse, sixth order method with K×K elements K= 3(-o), 4(-.), 5(..), 6(- -), and 7(-).

Conservation of Linear Impulse(P)

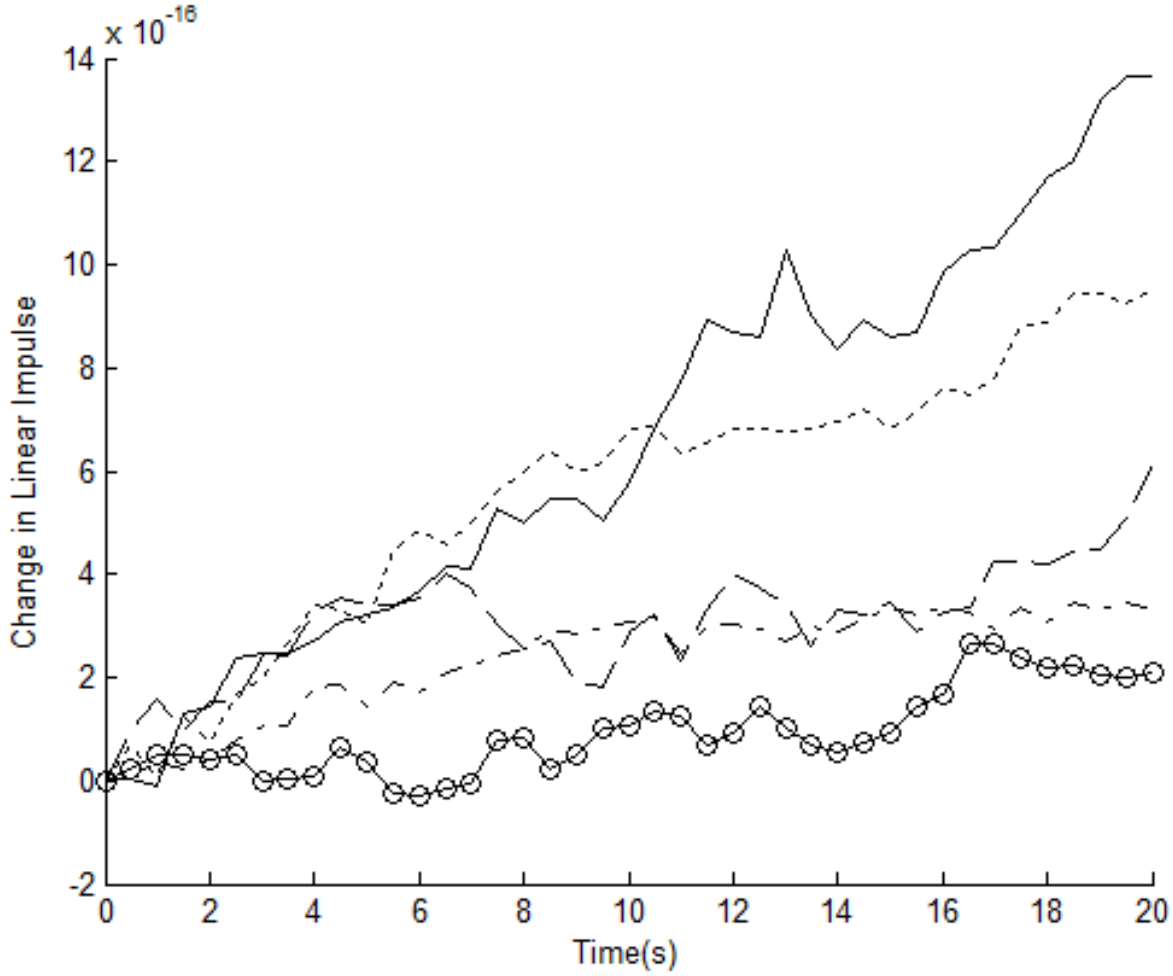


Figure : Conservation of linear impulse, sixth order method with K×K elements K= 3(-o), 4(-.), 5(..), 6(- -), and 7(-).

Qualitative Comparison of Vorticity(S)

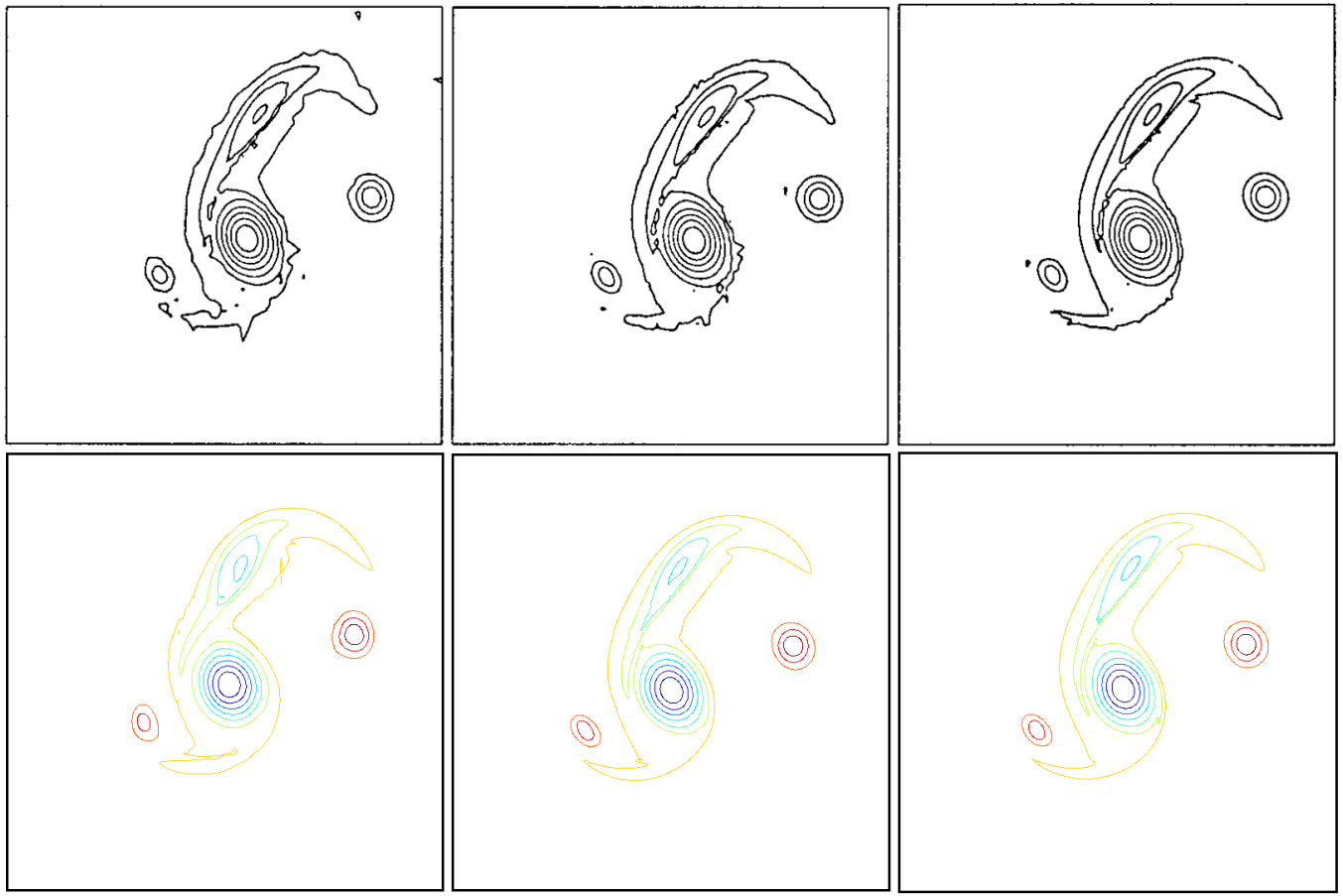


Figure : Comparison of Strain's results with present method t=28. Left to right, top to bottom, DOF= 6400, 12800, 25600; 3136, 7056, 63504. Reprinted with permission from Elsevier.

Approximated Convergence Rate(S)

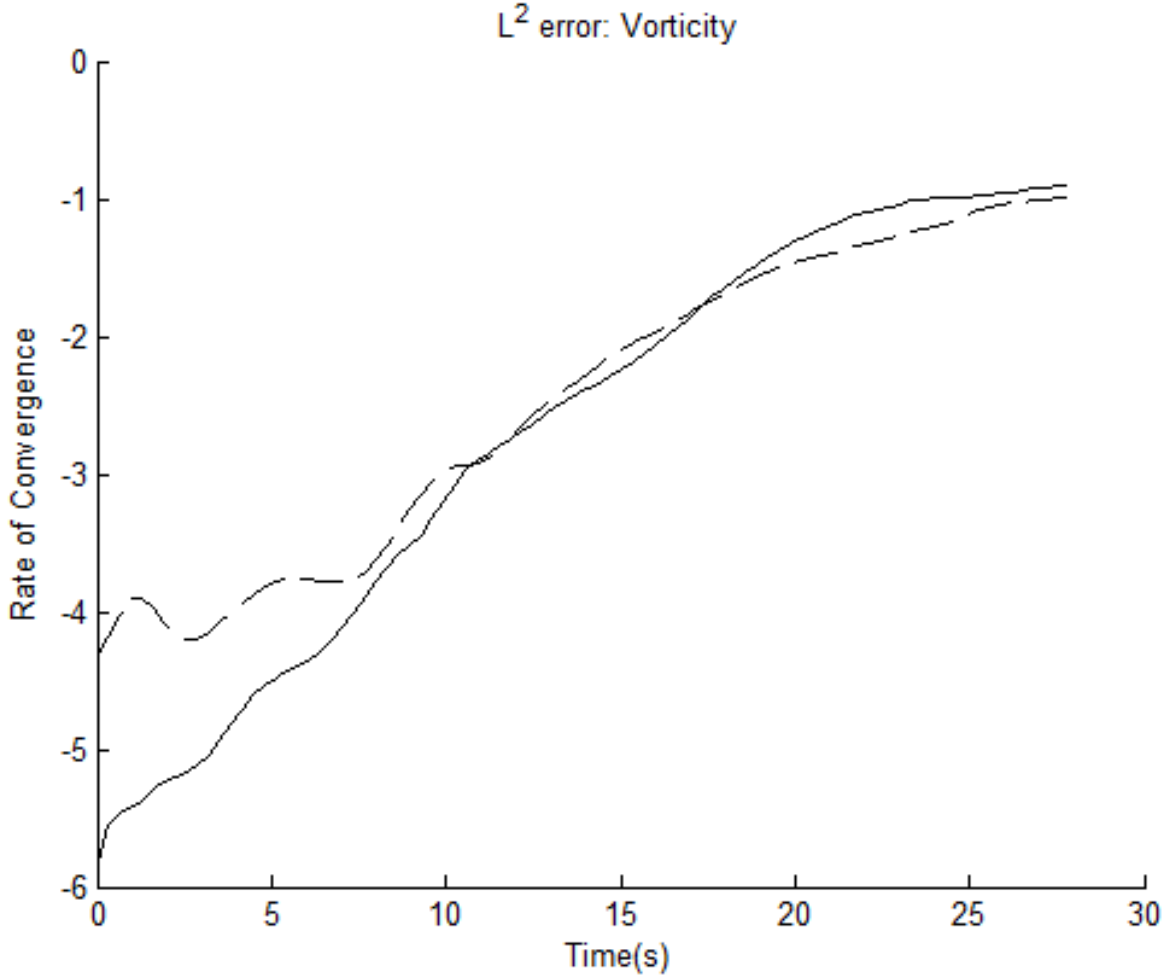


Figure : Dependency of rate of convergence on order of method, for a 4th order(- -) and 6th order(-) method.

Approximated Convergence Rate(S)(cont.)

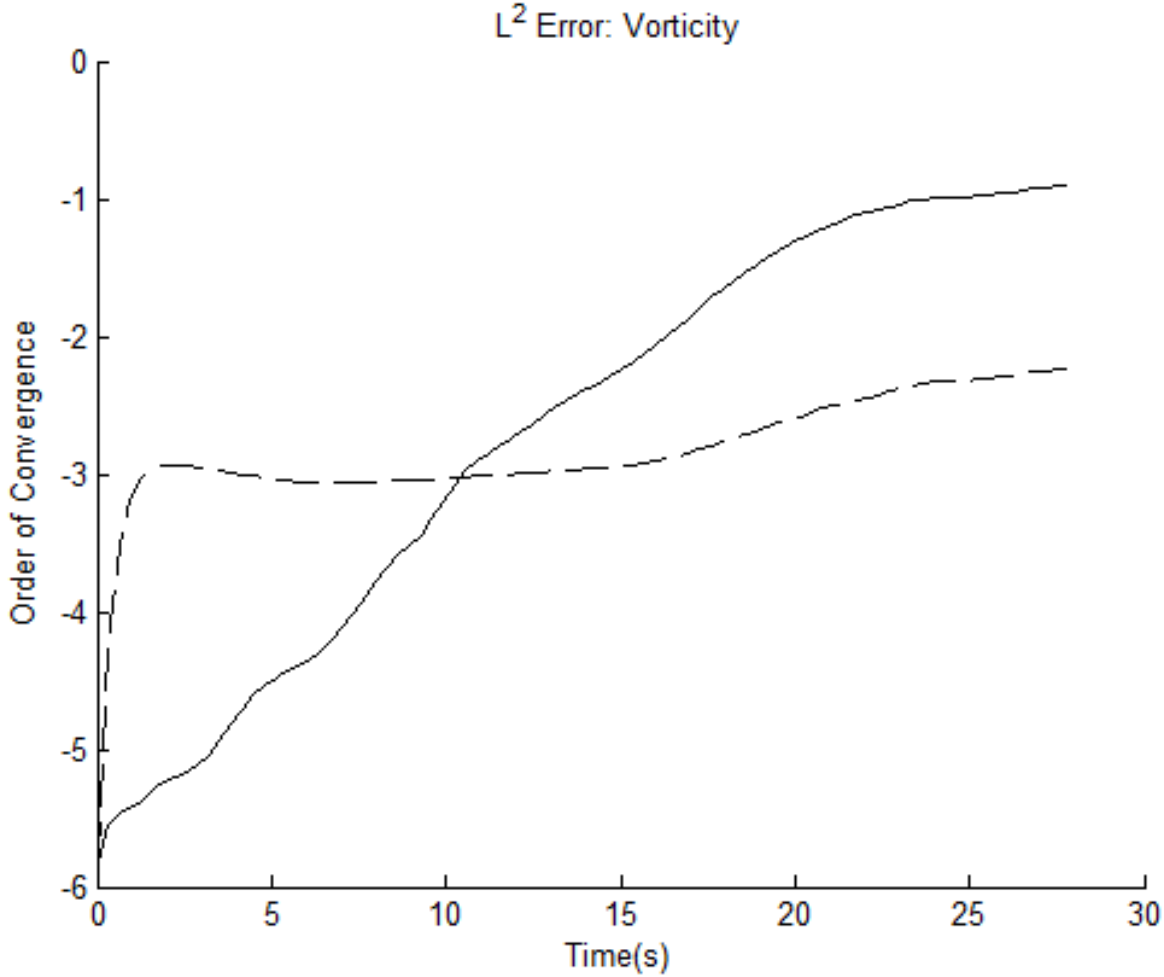


Figure : Dependency of rate of convergence on cutoff radius in sixth order method, for a $\delta/\Delta x =$ 0.5(- -) and 0.25(-).

Qualitative Comparison of Vorticity(K)

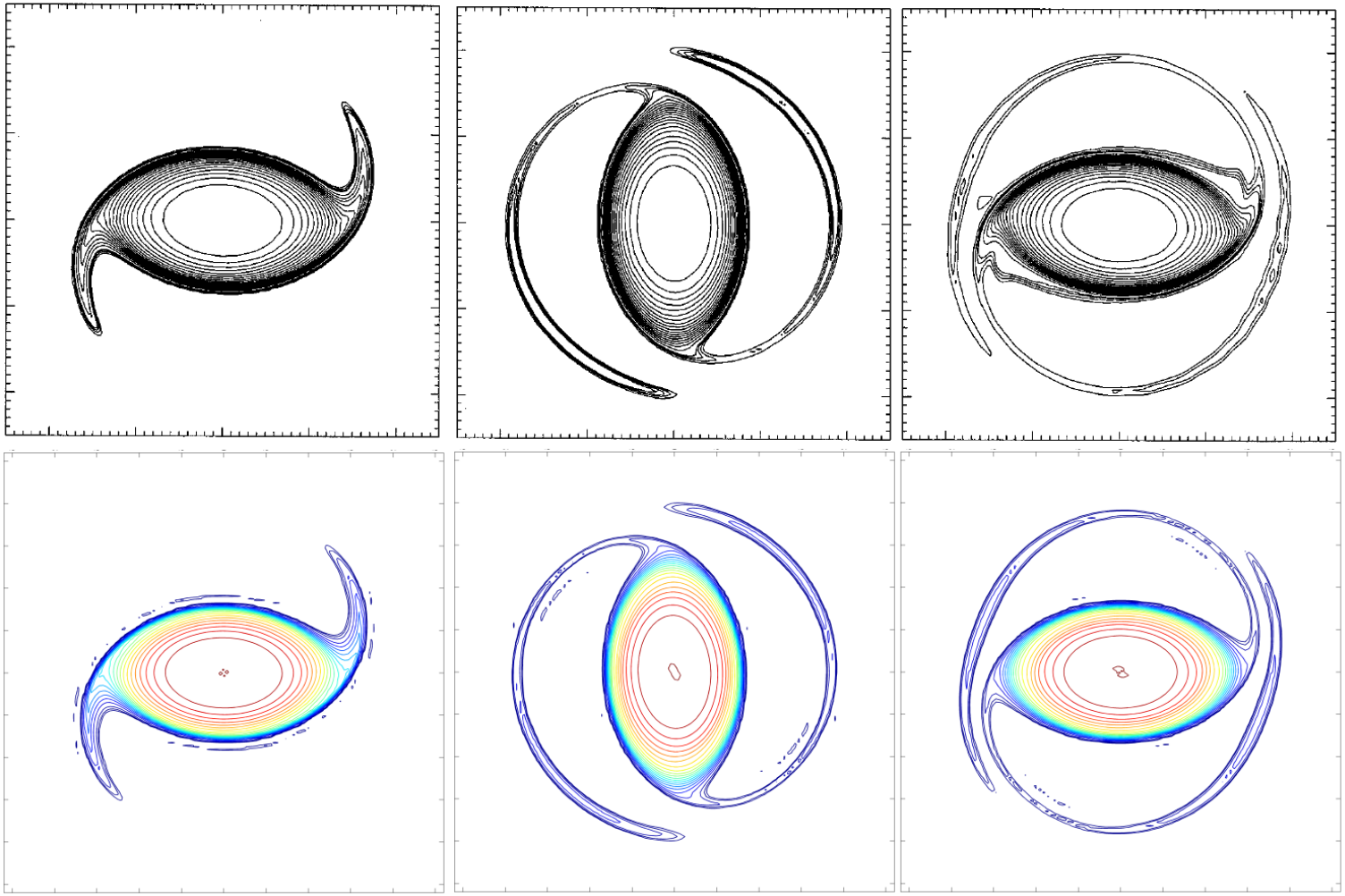


Figure : Comparison of vorticity, Koumoutsakos (top) and present method (bottom). From left to right, top to bottom t=1, 2, 4; 0.80, 1.93, 2.32. Reprinted with permission from Elsevier.

Qualitative Comparison of Vorticity(K)(cont.)

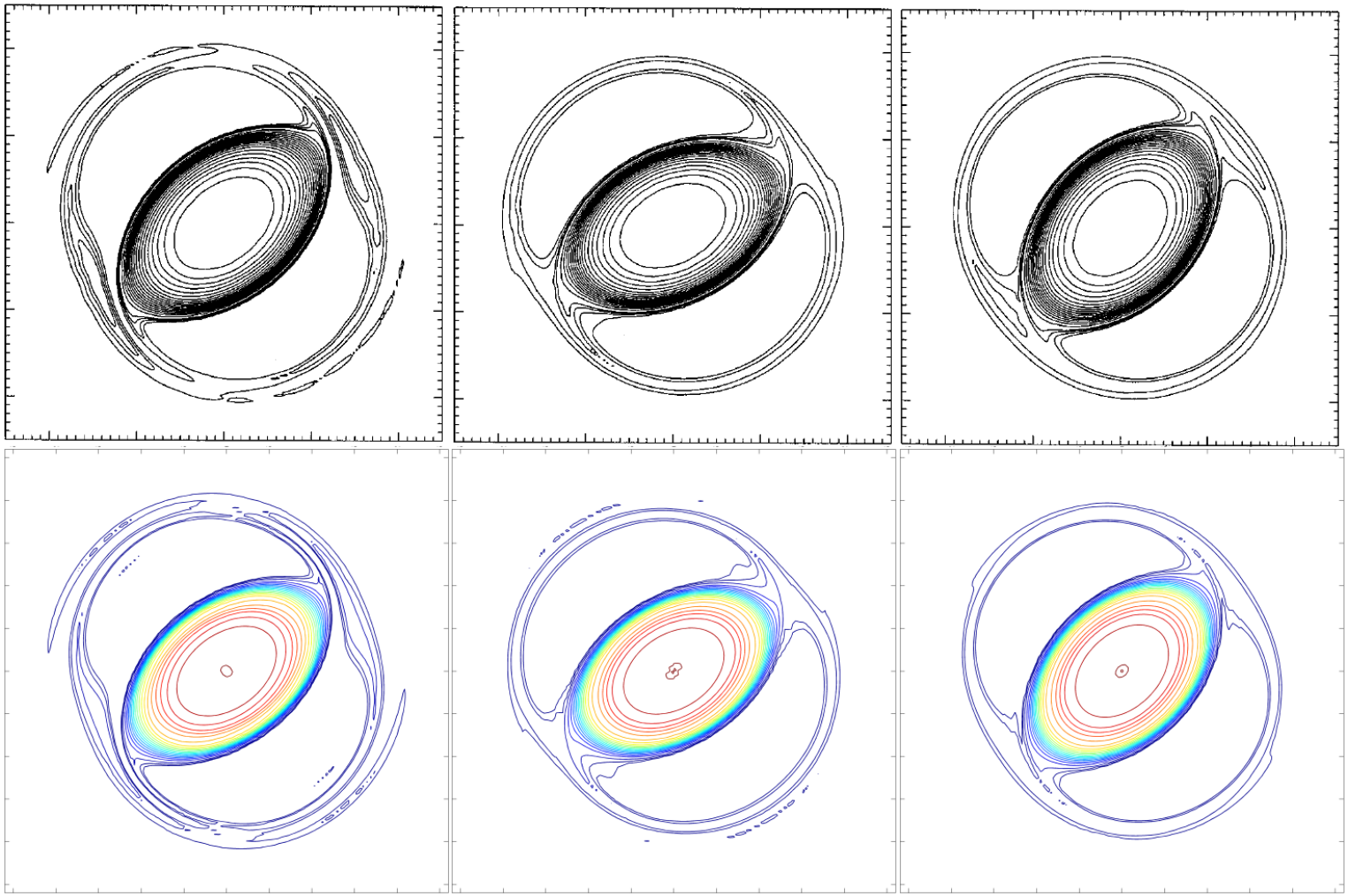


Figure : Comparison of vorticity, Koumoutsakos (top) and present method (bottom). From left to right top to bottom,: t=6, 12, 18; 5.94, 11.99, 17.94. Reprinted with permission from Elsevier.

Qualitative Comparison of Vorticity(K)(cont.)

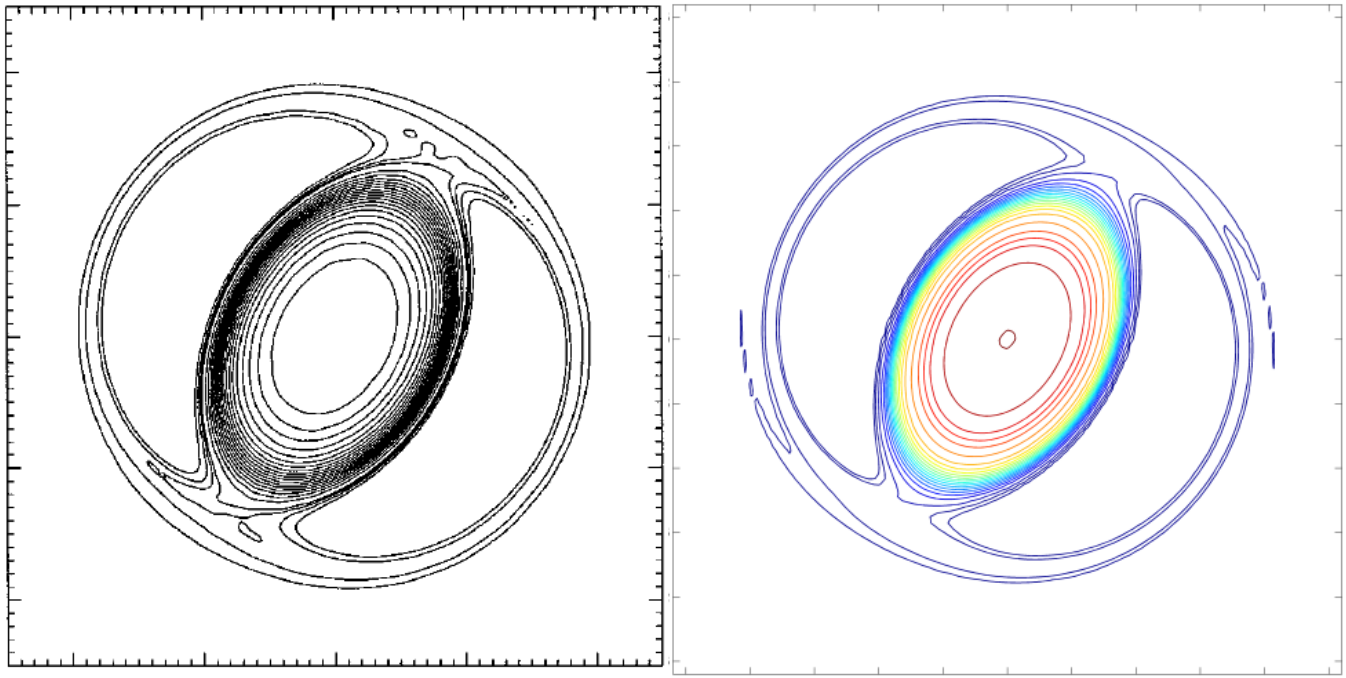


Figure : Comparison of vorticity, Koumoutsakos (left) and present method (right). From left to right: t=24; 23.98. Reprinted with permission from Elsevier.

Comparison of Aspect Ratio(K)

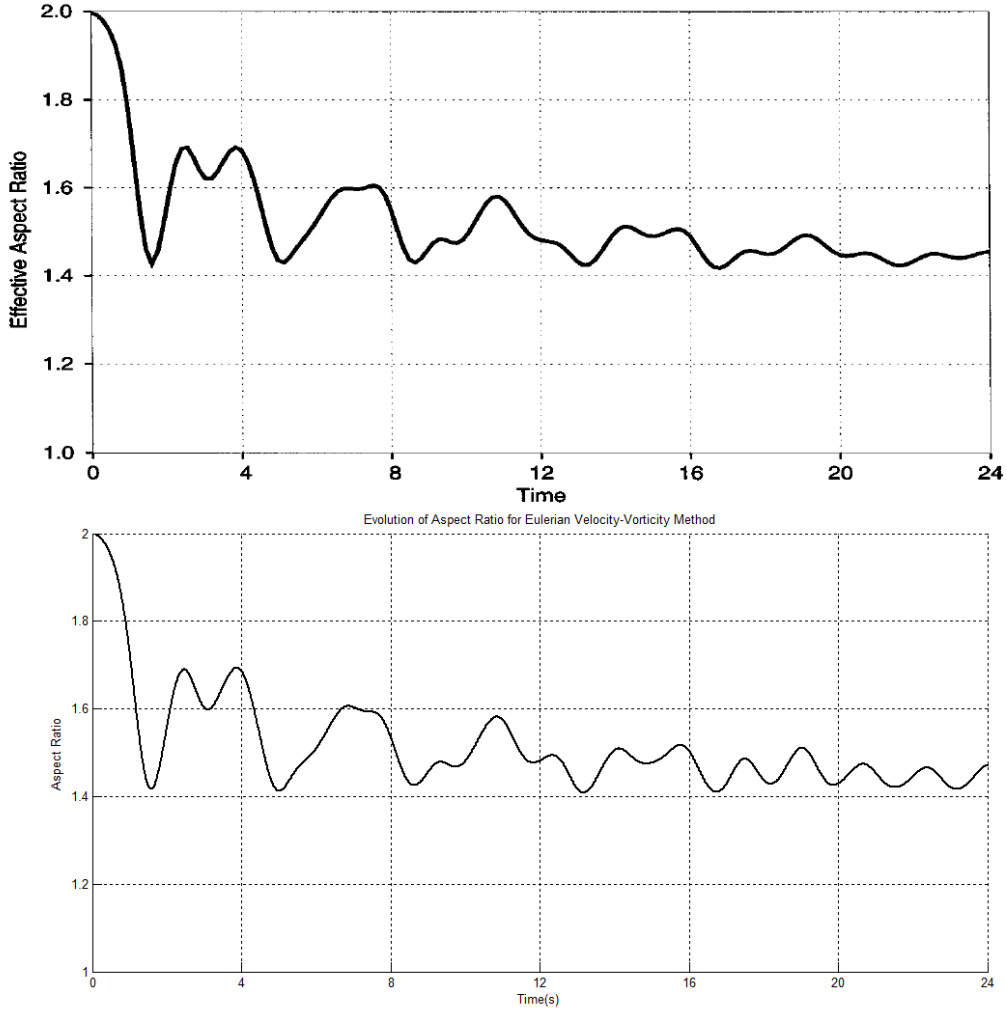


Figure : Comparison of effective aspect ratio, Koumoutsakos (top) and present method (bottom). Reprinted with permission from Elsevier.

Discussion: Validation

- Analytical: With proper choice of kernel, cutoff radius, stage-wise velocity evaluation, and matching velocity order method able to obtain solution within discretization error of exact solution for velocity and vorticity.
- Qualitative: Excellent agreement in Strain test case even for test with far fewer DOFs
- Good agreement in Koumoutsakos test case, with some minor deviation towards end of period studied and minor artifacting at vortex body boundaries.
- DOFs required about equal to Koumoutsakos's results, despite being higher-order method.
- Arm filaments and vortex body boundary challenging for polynomial basis functions, discontinuous derivatives affect bound on interpolation error.

Convergence Rate

- In Perlmann test case, capable of near optimal convergence rates for stationary vortex test case.
- Half-order less convergence rate for higher order methods due to lack of as many vorticity derivatives.
- Non-optimal apprroximated convergence observed in Strain test case.
- Choice of two options, cutoff radius too small and gradual decay of convergence to first order from optimal. Cutoff radius too large, constant but non-optimal order of convergence.
- Flow insufficiently smooth in Strain test case to reach acceptable level of approximation and quadrature error
- More accurate quadrature rule needed for nearly singular Biot-Savart integral.
- Convergence rate in Koumoutsakos test case not measured due to insufficient convergence of lower fidelity tests.

Conclusion: Convergence Rate

- Overarching thesis goal of high-order solver capable of solving inviscid incompressible vorticity-dominated flows in 2D met
- Advective solver capable of mixed order flux handling was constructed capable of high-order convergence
- High-order Biot-Savart evaluation routine using a direct summation approach created
- Result is a complete high-order method for velocity-vorticity inviscid flow that integrates all necessary subsystems
- Both solver and underlying Eulerian vortex approach were validated against analytical and qualitative test cases
- L^2 error, vortex diagnostics, and convergence rate were measured
- Method was shown capable of achieving near optimal convergence rate in smooth flows
- Flows where vorticity is changing rapidly spatially the method suffers from insufficient Biot-Savart quadrature convergence
- If improvement made to quadrature routine to integrate nearly singular Biot-Savart kernel, may be possible to maintain overall convergence of method without decay