Proceedings of the ASME 2022
Fluids Engineering Division Summer Meeting
FEDSM2022
August 3-5, 2022, Toronto, Canada

FEDSM2022-86472

A METHOD FOR NUMERICAL EVALUATION OF SINGULAR INTEGRALS IN CURVED HEXAHEDRA AND WITH HIGH-ORDER SOURCE FUNCTIONS

Adrin Gharakhani
Applied Scientific Research, Inc.
Irvine, California
Email: adrin@applied-scientific.com

Mark J. Stock
Applied Scientific Research, Inc.
Irvine, California
Email: markjstock@gmail.com

ABSTRACT

A method is developed for evaluating the 3-D Biot-Savart singular integral for the velocity field induced by arbitrarily high-order (discontinuous) vorticity in arbitrarily high-order curved hexahedral elements. The proposed method uses Duffy's coordinate transformation and singularity removal strategy, which, through a set of transformations, accommodates accurate evaluation of the transformed volume integrals using standard adaptive cubature techniques. In this paper, the new method is formulated in detail, followed by a series of benchmark tests demonstrating the convergence properties of the singular volume integral as a function of the discretization order of the vorticity (source) field.

Keywords: High Order Singular Integration, Biot-Savart Integral

NOMENCLATURE

el	Index to mesh element.
J	The Jacobian of coordinate transformation.
K_g	Number of grid points per element per coordinate direction.
K_s	Number of solution points per element per coordinate direction.
L	Lagrange interpolation basis function.
M	Number of mesh elements.
(p,q,s)	Coordinates in Duffy transformation.
Re	Reynolds number.
t	Time.
(u,v,w)	Shifted coordinates in parameter domain.
V_g	Vandermonde matrix.
$\boldsymbol{u}(\boldsymbol{x})$	Velocity vector.
$\boldsymbol{x}=(x,y,z)$	Position vector.
$\boldsymbol{X}^{el}_{i,j,k}$	Polynomial coefficients in the (i,j,k) directions describing the position vector $\boldsymbol{x}$ in element el.
$\boldsymbol{\Gamma}$	Volume integrated vorticity vector.
Δt	Computational time step.
Δx	Nominal mesh element size.
$(\xi,\eta,\zeta)_g$	Grid nodes in the parameter space.
$(\xi,\eta,\zeta)_s$	Solution nodes in the parameter space.
$\boldsymbol{\xi}=(\xi,\eta,\zeta)$	Coordinates in the parameter space.
$\boldsymbol{\omega}(\boldsymbol{x})$	Vorticity vector.
$\boldsymbol{\Omega}^{el}_{i,j,k}$	Polynomial coefficients in the (i,j,k) directions describing the vorticity vector $\boldsymbol{\omega}$ in element el.
$\times$	Cross product operation.
$\otimes$	Kronecker product operation.

INTRODUCTION

Vortex methods, whether in Lagrangian particle or Eulerian mesh formulation, are a class of Computational Fluid Dynamics (CFD) solvers ideal for simulations of unsteady incompressible vortex dominated flows. This is based on the observation that it is often more efficient to discretize the vorticity form of the Navier Stokes equations; i.e., the Vorticity Transport Equations (VTE), because VTE, especially in conservative form, preserve vorticity by construction; as a result, even low-order vortex methods tend to maintain the coherence of vortical structures for long times and distances [1]. Further, using vorticity alleviates the pressure-

velocity coupling challenges experienced in traditional primitive-variables-based CFD.

Compact high-order methods have received much attention by the CFD developer community in recent years, primarily because of their higher computational efficiency for a given level of accuracy vis-à-vis traditional low-order solvers. To this end, Discontinuous Galerkin formulations of 2-D VTE have appeared in the literature [2–4]. A discontinuous finite-difference method using unstructured arbitrarily high-order curvilinear (quadrilateral) elements has also been reported recently [5]. However, to the best of our knowledge, high-order VTE solvers in 3-D have not appeared in the literature.

One key component of a 3-D VTE solution algorithm is a Poisson solver module to evaluate the velocity induced by the field vorticity. This can be accomplished using traditional grid-based solvers. However, for external flow problems, where the vorticity field is very compact and much smaller than the flow domain, the latter approach may become quite inefficient. In such scenarios it is more efficient to apply the so-called Biot-Savart singular integral equation, because (1) it only operates on the compact vorticity source, and, by construction, it (2) satisfies the continuity constraint pointwise and (3) applies the far-field velocity boundary condition.

Biot-Savart volume integrals are relatively simple to evaluate analytically (or semi-analytically) when considering piecewise constant vorticity in *linear* elements (e.g., hexa- and tetrahedrons) [6]. For piecewise high-order vorticity in curved elements the integral must be evaluated numerically, paying particular attention to the integrand singularity and its efficient removal. Interestingly, though various Boundary Element Methods have been presented in the literature for the more challenging solution of singular integrals on curved 3-D surface elements [7–10], similar solutions for high-order sources in curved volumes are scarce and poorly presented [11, 12].

In this paper, the formulation and the numerical algorithm for a new method for evaluating the singular Biot-Savart integral due to an arbitrarily high-order (discontinuous) vorticity in an arbitrarily high-order curved hexahedral volume element is presented. The accuracy and convergence rates of the integral are then benchmarked for various polynomial orders of vorticity in uniform and warped volume elements using a manufactured test problem with an exact solution. Though the method described herein focuses on piecewise discontinuous vorticity in hexahedral elements, it is general and may be applied to vorticity distributions with C^0 (or higher) continuity across volume elements, and/or to tetrahedral (or other) curved volumes, with minimal modifications to the algorithm to reflect the aforementioned differences.

FORMULATION

The velocity $\boldsymbol{u}(\boldsymbol{x})$ anywhere in the field is obtained by the linear sum of the Biot-Savart integrals prescribing the influence of the vorticity $\boldsymbol{\omega}^{el}$ within each cell volume el, as follows

$$\boldsymbol{u}(\boldsymbol{x}) = \frac{1}{4\pi} \sum_{el} \int_{\mathcal{D}_{el}} \boldsymbol{\omega}^{el}(\boldsymbol{x}') \times \frac{(\boldsymbol{x}-\boldsymbol{x}')}{|\boldsymbol{x}-\boldsymbol{x}'|^3} d\boldsymbol{x}' \tag{1}$$

In the present method, the computational domain used to discretize the Biot-Savart integrals consists of unstructured, arbitrarily-ordered curvilinear hexahedral meshes/cells. Each cell element el is mapped from a bi-unit cube $[1,1]^3$ in $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ parameter space to the $\boldsymbol{x} = (x, y, z)$ physical space; the variation of $\boldsymbol{x}^{el}(\boldsymbol{\xi})$ within el is provided by the tensor-product of 1-D Lagrange interpolants of order $K_g - 1$ in the ξ, η and ζ directions as follows

$$\boldsymbol{x}^{el}(\boldsymbol{\xi}) = \sum_{i,j,k=0}^{K_g-1} L_i(\xi)\, L_j(\eta)\, L_k(\zeta)\, \boldsymbol{x}^{el}\left(\xi_{g_i}, \eta_{g_j}, \zeta_{g_k}\right) \tag{2a}$$

$$L_i(\xi) = \prod_{l=0, l\neq i}^{K_g-1} \frac{\xi - \xi_{g_l}}{\xi_{g_i} - \xi_{g_l}} \tag{2b}$$

where (ξ_g, η_g, ζ_g) are standard uniformly distributed grid points prescribing the curved hexahedral cell topology with C^0 (or higher) continuity at the cell boundaries. Vorticity distribution within this cell is prescribed similarly using the tensor-product of 1-D Lagrange interpolants of order $K_s - 1$ in the ξ, η and ζ directions as follows

$$\boldsymbol{\omega}^{el}(\boldsymbol{\xi}) = \sum_{i,j,k=0}^{K_s-1} L_i(\xi)\, L_j(\eta)\, L_k(\zeta)\, \boldsymbol{\omega}^{el}\left(\xi_{s_i}, \eta_{s_j}, \zeta_{s_k}\right) \tag{3a}$$

$$L_i(\xi) = \prod_{l=0, l\neq i}^{K_s-1} \frac{\xi - \xi_{s_l}}{\xi_{s_i} - \xi_{s_l}} \tag{3b}$$

where (ξ_s, η_s, ζ_s) are the solution nodes describing a discontinuous variation of vorticity in the cell. Though the choice of the solution nodes is arbitrary and has no bearing on the accuracy of the method, in this paper they are assigned at Gauss-Legendre points.

Given Eqs. (2) and (3), Eq. (1) can now be recast in the parametric space as follows

$$\boldsymbol{u}(\boldsymbol{x}) = \frac{1}{4\pi} \sum_{el} \int_{\mathcal{D}_{el}} \boldsymbol{\omega}^{el}(\boldsymbol{\xi}) \times \frac{\left(\boldsymbol{x} - \boldsymbol{x}^{el}(\boldsymbol{\xi})\right)}{\left|\boldsymbol{x} - \boldsymbol{x}^{el}(\boldsymbol{\xi})\right|^3} J^{el}(\boldsymbol{\xi})\, d\boldsymbol{\xi} \tag{4a}$$

where

$$J^{el}(\boldsymbol{\xi}) = \frac{\partial \boldsymbol{x}^{el}(\boldsymbol{\xi})}{\partial \xi} \cdot \left(\frac{\partial \boldsymbol{x}^{el}(\boldsymbol{\xi})}{\partial \eta} \times \frac{\partial \boldsymbol{x}^{el}(\boldsymbol{\xi})}{\partial \zeta} \right) \tag{4b}$$

is the Jacobian of the transformation.

Change of Bases in Polynomial Interpolation

From a user's perspective, it is convenient to assign the position and vorticity vectors using the above Lagrange interpolation functions. However, from the computational standpoint in this paper, it is more appropriate to use monomial bases, in the following form

$$\boldsymbol{x}^{el}(\boldsymbol{\xi}) = \sum_{i,j,k=0}^{K_g-1} \boldsymbol{X}^{el}_{i,j,k}\, \xi^i \eta^j \zeta^k \tag{5a}$$

$$\boldsymbol{\omega}^{el}(\boldsymbol{\xi}) = \sum_{i,j,k=0}^{K_s-1} \boldsymbol{\Omega}^{el}_{i,j,k}\, \xi^i \eta^j \zeta^k \tag{5b}$$

The algorithm used in this work to obtain the coefficients for the trivariate monomials from the corresponding coefficients for the Lagrange interpolation function is the 3-D extension to the 2-D transformation introduced in [13]. The proof of the formulation is lengthy and beyond the scope of this paper. Here, only the basic concepts are provided to facilitate implementation by other researchers. The following discussion focuses on the vector position transformation; its extension to the vorticity vector is trivially self-evident.

We introduce here the following Vandermonde matrix

$$V_{\xi_g} = V_{\eta_g} = V_{\zeta_g} = V_g = \begin{bmatrix} 1 & \xi_{g_0} & \cdots & \xi_{g_0}^{K_g-1} \\ 1 & \xi_{g_1} & \cdots & \xi_{g_1}^{K_g-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \xi_{g_{K_g-1}} & \cdots & \xi_{g_{K_g-1}}^{K_g-1} \end{bmatrix} \tag{6}$$

where $V_{\xi_g} = V_{\eta_g} = V_{\zeta_g}$ due to the symmetry of the 1-D tensor-products. We also utilize the following flattening operations

$$vec\left(\boldsymbol{x}^{el}\left(\xi_g, \eta_g, \zeta_g\right)\right) = \begin{pmatrix} \boldsymbol{x}^{el}(\xi_{g_0}, \eta_{g_0}, \zeta_{g_0}) \\ \boldsymbol{x}^{el}(\xi_{g_1}, \eta_{g_0}, \zeta_{g_0}) \\ \vdots \\ \boldsymbol{x}^{el}(\xi_{g_{K_g-1}}, \eta_{g_0}, \zeta_{g_0}) \\ \boldsymbol{x}^{el}(\xi_{g_0}, \eta_{g_1}, \zeta_{g_0}) \\ \boldsymbol{x}^{el}(\xi_{g_1}, \eta_{g_1}, \zeta_{g_0}) \\ \vdots \\ \boldsymbol{x}^{el}(\xi_{g_{K_g-1}}, \eta_{g_1}, \zeta_{g_0}) \\ \vdots \\ \boldsymbol{x}^{el}(\xi_{g_{K_g-1}}, \eta_{g_{K_g-1}}, \zeta_{g_0}) \\ \boldsymbol{x}^{el}(\xi_{g_0}, \eta_{g_0}, \zeta_{g_1}) \\ \boldsymbol{x}^{el}(\xi_{g_1}, \eta_{g_0}, \zeta_{g_1}) \\ \vdots \\ \boldsymbol{x}^{el}(\xi_{g_{K_g-1}}, \eta_{g_0}, \zeta_{g_1}) \\ \vdots \\ \boldsymbol{x}^{el}(\xi_{g_0}, \eta_{g_{K_g-1}}, \zeta_{g_1}) \\ \vdots \\ \boldsymbol{x}^{el}(\xi_{g_{K_g-1}}, \eta_{g_{K_g-1}}, \zeta_{g_1}) \\ \vdots \\ \boldsymbol{x}^{el}(\xi_{g_0}, \eta_{g_0}, \zeta_{g_{K_g-1}}) \\ \vdots \\ \boldsymbol{x}^{el}(\xi_{g_{K_g-1}}, \eta_{g_{K_g-1}}, \zeta_{g_{K_g-1}}) \end{pmatrix} \tag{7a}$$

$$vec\left(\boldsymbol{X}^{el}_{i,j,k}\right) = \begin{pmatrix} \boldsymbol{X}^{el}_{0,0,0} \\ \boldsymbol{X}^{el}_{1,0,0} \\ \vdots \\ \boldsymbol{X}^{el}_{K_g-1,0,0} \\ \boldsymbol{X}^{el}_{0,1,0} \\ \boldsymbol{X}^{el}_{1,1,0} \\ \vdots \\ \boldsymbol{X}^{el}_{K_g-1,1,0} \\ \vdots \\ \boldsymbol{X}^{el}_{K_g-1,K_g-1,0} \\ \boldsymbol{X}^{el}_{0,0,1} \\ \boldsymbol{X}^{el}_{1,0,1} \\ \vdots \\ \boldsymbol{X}^{el}_{K_g-1,0,1} \\ \vdots \\ \boldsymbol{X}^{el}_{0,K_g-1,1} \\ \vdots \\ \boldsymbol{X}^{el}_{K_g-1,K_g-1,1} \\ \vdots \\ \boldsymbol{X}^{el}_{0,0,K_g-1} \\ \vdots \\ \boldsymbol{X}^{el}_{K_g-1,K_g-1,K_g-1} \end{pmatrix} \tag{7b}$$

Finally, the monomial coefficients are obtained using the

corresponding Lagrange coefficients as follows

$$vec\left(\boldsymbol{X}^{el}_{i,j,k}\right)=\left(V_g^{-1}\otimes\left(V_g^{-1}\otimes V_g^{-1}\right)\right)\cdot vec\left(\boldsymbol{x}^{el}\left(\xi_g,\eta_g,\zeta_g\right)\right) \quad (8)$$

For the sake of completeness, the following is the operation for the vorticity coefficients

$$vec\left(\boldsymbol{\Omega}^{el}_{i,j,k}\right)=\left(V_s^{-1}\otimes\left(V_s^{-1}\otimes V_s^{-1}\right)\right)\cdot vec\left(\boldsymbol{\omega}^{el}\left(\xi_s,\eta_s,\zeta_s\right)\right) \quad (9)$$

Biot-Savart Integration for Coincident Cells

The Biot-Savart singular volume integration for coincident cells (i.e., the target point is within the vorticity/source cell) is the 3-D curved hexahedral extension of the concepts presented in [14] for 2-D curved quadrilateral surfaces using arbitrary-order basis functions for the source. This approach uses the method of Duffy for singularity cancellation [15, 16] and is adopted in this work because it was demonstrated in [14] to be faster and to require fewer Gauss integration points for a given level of accuracy than the alternatives used in the literature; i.e., (1) singularity extraction (or removal) method, which consists of analytical integration of principal singular part of the integrand [17]; (2) polar transformation method for singularity cancellation, similar to that of the Duffy method [18]; and (3) quadratic and cubic rectangular transformation methods for singularity cancellation [19, 20].

The singular volume integration of arbitrary-order vorticity distribution in curved hexahedra proceeds as follows. The target or evaluation point $\boldsymbol{x}$ corresponds to $\boldsymbol{\xi}_0=(\xi_0,\eta_0,\zeta_0)$ in the parametric space, as depicted in Fig.1. The bi-unit cube is subdivided into six pyramids, each consisting of a vertex at $\boldsymbol{\xi}_0$ and a base at one of the six cube faces $(\xi^+,\xi^-,\eta^+,\eta^-,\zeta^+,\zeta^-)$ Figure 1 shows the pyramid with base on the ξ^+ face (red edges). The Biot-Savart velocity at $\boldsymbol{x}$ induced by a piecewise discontinuous vorticity in the curved hexahedral element is the linear sum of the velocities due to these six pyramids

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{x})=&\boldsymbol{u}_{\xi^+}(\boldsymbol{\xi_0})+\boldsymbol{u}_{\xi^-}(\boldsymbol{\xi_0})+\boldsymbol{u}_{\eta^+}(\boldsymbol{\xi_0})+\boldsymbol{u}_{\eta^-}(\boldsymbol{\xi_0})\\&+\boldsymbol{u}_{\zeta^+}(\boldsymbol{\xi_0})+\boldsymbol{u}_{\zeta^-}(\boldsymbol{\xi_0})\end{aligned} \quad (10)$$

As template and example, and using Eqs. (4) and (5), the velocity due to the ξ^+ pyramid is evaluated via

$$\boldsymbol{u}_{\xi^+}(\boldsymbol{\xi_0})=\frac{1}{4\pi}\int_{\mathcal{D}_{el}^{\xi^+}}\frac{\left(\boldsymbol{x}^{el}(\boldsymbol{\xi})-\boldsymbol{x}^{el}(\boldsymbol{\xi_0})\right)}{\left|\boldsymbol{x}^{el}(\boldsymbol{\xi})-\boldsymbol{x}^{el}(\boldsymbol{\xi_0})\right|^3}\times\boldsymbol{\omega}^{el}(\boldsymbol{\xi})\,J^{el}(\boldsymbol{\xi})\,d\boldsymbol{\xi} \quad (11a)$$

where $\boldsymbol{\omega}^{el}(\boldsymbol{\xi})$ is assigned using Eq. (5b), and

$$\boldsymbol{x}^{el}(\boldsymbol{\xi})-\boldsymbol{x}^{el}(\boldsymbol{\xi_0})=\sum_{i,j,k=0}^{K_g-1}\boldsymbol{X}^{el}_{i,j,k}\left(\xi^i\eta^j\zeta^k-\xi_0^i\eta_0^j\zeta_0^k\right) \quad (11b)$$

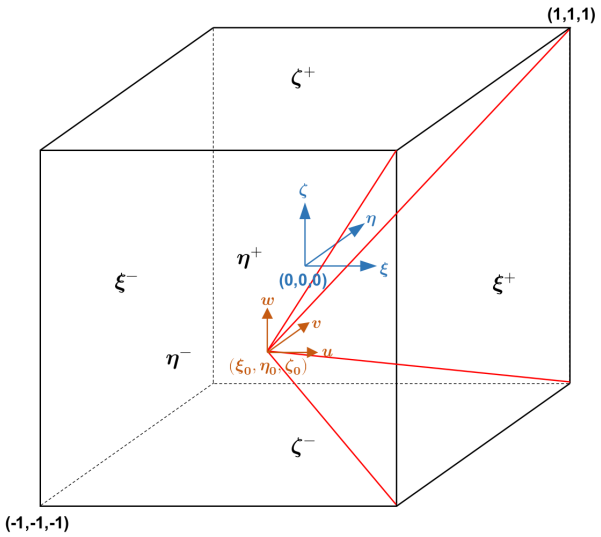

FIGURE 1. DUFFY METHOD FOR SINGULARITY CANCELLATION AT $\boldsymbol{\xi_0}=(\xi_0,\eta_0,\zeta_0)$ IN A BI-UNIT CUBE IN 3-D PARAMETRIC SPACE. THE CUBE IS DIVIDED INTO SIX PYRAMIDS, EACH CONSISTING OF A VERTEX AT $\boldsymbol{\xi_0}$ AND A BASE AT ONE OF THE SIX CUBE FACES $(\xi^+,\xi^-,\eta^+,\eta^-,\zeta^+,\zeta^-)$.

$$\begin{aligned}J^{el}(\boldsymbol{\xi})=&\left(\sum_{i,j,k=0}^{K_g-1}\boldsymbol{X}^{el}_{i,j,k}\,i\xi^{i-1}\eta^j\zeta^k\right)\cdot\\&\left(\sum_{i,j,k=0}^{K_g-1}\boldsymbol{X}^{el}_{i,j,k}\,j\xi^i\eta^{j-1}\zeta^k\right)\times\left(\sum_{i,j,k=0}^{K_g-1}\boldsymbol{X}^{el}_{i,j,k}\,k\xi^i\eta^j\zeta^{k-1}\right)\end{aligned} \quad (11c)$$

We proceed using the following shift in the coordinate system such that the pyramid vertex $\boldsymbol{\xi_0}$ is at the origin

$$\begin{aligned}\xi&=\xi_0+u\\\eta&=\eta_0+v\\\zeta&=\zeta_0+w\end{aligned} \quad (12)$$

Substituting (12) into (11b), performing a Binomial expansion, collecting terms, and further simplifying yields

$$\boldsymbol{x}^{el}(\boldsymbol{\xi})-\boldsymbol{x}^{el}(\boldsymbol{\xi_0})=\sum_{i,j,k=0}^{K_g-1}\boldsymbol{X}^{el}_{i,j,k}\begin{Bmatrix}\eta_0^j\zeta_0^kU+\xi_0^i\zeta_0^kV+\\\xi_0^i\eta_0^jW+\zeta_0^kUV+\\\eta_0^jUW+\xi_0^iVW+\\UVW\end{Bmatrix} \quad (13a)$$

where

$$U = \sum_{l=1}^{K_g-1} \binom{i}{l} \xi_0^{i-l} u^l \tag{13b}$$

$$V = \sum_{m=1}^{K_g-1} \binom{j}{m} \eta_0^{j-m} v^m \tag{13c}$$

$$W = \sum_{n=1}^{K_g-1} \binom{k}{n} \zeta_0^{k-n} w^n \tag{13d}$$

and $\binom{0}{i} = 0$ by definition. Application of the approach above to $\boldsymbol{\omega}^{el}(\boldsymbol{\xi})$ and $J^{el}(\boldsymbol{\xi})$ is trivial and not presented here.

Equation (13) in its current form is computationally intensive because (U,V,W) have to be evaluated for every (i,j,k) triple summation, each, in turn, evaluated in an adaptive cubature process. Fortunately, it is possible to reduce the computational cost *significantly* by rearranging the order and the limits of the summations, leading to the following simple form

$$\boldsymbol{x}^{el}(\boldsymbol{\xi}) - \boldsymbol{x}^{el}(\boldsymbol{\xi_0}) = \sum_{l,m,n=0}^{K_g-1} \boldsymbol{\pi}^{el}_{l,m,n} u^l v^m w^n \tag{14a}$$

where

$$\boldsymbol{\pi}^{el}_{l,m,n} = \sum_{i=l,j=m,k=n}^{K_g-1} \boldsymbol{X}^{el}_{i,j,k} \binom{i}{l} \binom{j}{m} \binom{k}{n} \xi_o^{i-l} \eta_0^{j-m} \zeta_0^{k-n} \tag{14b}$$

$$\boldsymbol{\pi}^{el}_{0,0,0} = 0 \tag{14c}$$

The following Duffy transformation rule is then applied to the above compact formulation, as well as its equivalents for $\boldsymbol{\omega}^{el}(\boldsymbol{\xi})$ and $J^{el}(\boldsymbol{\xi})$, to accommodate singularity cancellation at $\boldsymbol{\xi_0}$

$$\begin{aligned} u &= c \cdot p \quad , \qquad 0 \le p \le 1 \\ v &= p \cdot q \quad , -1-\eta_0 \le q \le 1-\eta_0 \\ w &= p \cdot s \quad , -1-\zeta_0 \le s \le 1-\zeta_0 \\ c &= 1-\xi_0 \end{aligned} \tag{15}$$

For the sake of completeness, the following is the final form of the components that contribute to evaluating $\boldsymbol{u}_{\xi^+}(\boldsymbol{\xi_0})$ upon integrand singularity cancellation

$$\boldsymbol{u}_{\xi^+}(\boldsymbol{\xi_0}) = \int_{-1-\zeta_0}^{1-\zeta_0} \int_{-1-\eta_0}^{1-\eta_0} \int_0^1 Q \, dp \, dq \, ds \tag{16a}$$

$$Q = \frac{1}{4\pi} \frac{\Delta \boldsymbol{x}^{el}}{|\Delta \boldsymbol{x}^{el}|^3} \times \boldsymbol{\omega}^{el} \, c \frac{\partial \boldsymbol{x}^{el}}{\partial \xi} \cdot \left(\frac{\partial \boldsymbol{x}^{el}}{\partial \eta} \times \frac{\partial \boldsymbol{x}^{el}}{\partial \zeta} \right) \tag{16b}$$

$$\begin{aligned} \Delta \boldsymbol{x}^{el} &= \boldsymbol{\alpha}^{el}_{1,0,0} + q\boldsymbol{\alpha}^{el}_{0,1,0} + s\boldsymbol{\alpha}^{el}_{0,0,1} \\ &\quad + pq\boldsymbol{\alpha}^{el}_{1,1,0} + ps\boldsymbol{\alpha}^{el}_{1,0,1} + pqs\boldsymbol{\alpha}^{el}_{0,1,1} \\ &\quad + \sum_{l,m,n=1}^{K_g-1} \boldsymbol{\alpha}^{el}_{l,m,n} \left(p^{l+m+n-1} q^m s^n \right) \end{aligned} \tag{16c}$$

$$\begin{aligned} c\frac{\partial \boldsymbol{x}^{el}}{\partial \xi} &= \boldsymbol{\alpha}^{el}_{1,0,0} + pq\boldsymbol{\alpha}^{el}_{1,1,0} + ps\boldsymbol{\alpha}^{el}_{1,0,1} \\ &\quad + \sum_{l,m,n=1}^{K_g-1} \boldsymbol{\alpha}^{el}_{l,m,n} \left(l p^{l+m+n-1} q^m s^n \right) \end{aligned} \tag{16d}$$

$$\begin{aligned} \frac{\partial \boldsymbol{x}^{el}}{\partial \eta} &= \boldsymbol{\alpha}^{el}_{0,1,0} + p\boldsymbol{\alpha}^{el}_{1,1,0} + ps\boldsymbol{\alpha}^{el}_{0,1,1} \\ &\quad + \sum_{l,m,n=1}^{K_g-1} \boldsymbol{\alpha}^{el}_{l,m,n} \left(m p^{l+m+n-1} q^{m-1} s^n \right) \end{aligned} \tag{16e}$$

$$\begin{aligned} \frac{\partial \boldsymbol{x}^{el}}{\partial \zeta} &= \boldsymbol{\alpha}^{el}_{0,0,1} + p\boldsymbol{\alpha}^{el}_{1,0,1} + pq\boldsymbol{\alpha}^{el}_{0,1,1} \\ &\quad + \sum_{l,m,n=1}^{K_g-1} \boldsymbol{\alpha}^{el}_{l,m,n} \left(n p^{l+m+n-1} q^m s^{n-1} \right) \end{aligned} \tag{16f}$$

$$\boldsymbol{\alpha}^{el}_{l,m,n} = c^l \, \boldsymbol{\pi}^{el}_{l,m,n} \tag{16g}$$

$$\boldsymbol{\omega}^{el} = \sum_{l,m,n=0}^{K_s-1} \boldsymbol{\beta}^{el}_{l,m,n} \left(p^{l+m+n} q^m s^n \right) \tag{16h}$$

$$\boldsymbol{\beta}^{el}_{l,m,n} = c^l \cdot \sum_{i=l,j=m,k=n}^{K_s-1} \boldsymbol{\Omega}^{el}_{i,j,k} \binom{i}{l} \binom{j}{m} \binom{k}{n} \xi_0^{i-l} \eta_0^{j-m} \zeta_0^{k-n} \tag{16i}$$

where the integration is performed numerically, in this work, utilizing an open-source module for adaptive multi-dimensional cubature of vector-valued integrands [21].

Singular integration of the remaining five pyramids follows (16) very closely. For pyramid ξ^-, one only needs to modify c to $c = -1 - \xi_0$ and reverse the integration order in the direction of p; i.e., $\int_1^0 (.) dp$. For pyramid pairs (η^+, η^-) and (ζ^+, ζ^-), one simply rotates the (ξ, η, ζ) and (u, v, w) triplets in (15) and (16) in the counter-clockwise and clockwise directions, respectively; i.e., (η, ζ, ξ) and (v, w, u), and (ζ, ξ, η) and (w, u, v), respectively.

Biot-Savart Integration for non-Coincident Cells

The Biot-Savart volume integration for non-coincident cells is performed numerically without difficulty because the integrand

is non-singular. The following is the formulation in parametric space

$$\boldsymbol{u}(\boldsymbol{x}) = \frac{1}{4\pi}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}\frac{\left(\boldsymbol{x}^{el}(\boldsymbol{\xi})-\boldsymbol{x}\right)}{\left|\boldsymbol{x}^{el}(\boldsymbol{\xi})-\boldsymbol{x}\right|^3}\times\boldsymbol{\omega}^{el}(\boldsymbol{\xi})\,J^{el}(\boldsymbol{\xi})\,d\xi\,d\eta\,d\zeta \quad (17)$$

where $\boldsymbol{x}^{el}(\boldsymbol{\xi})$, $\boldsymbol{\omega}^{el}(\boldsymbol{\xi})$ and $J^{el}(\boldsymbol{\xi})$ are evaluated using Eqs. (5a), (5b), and (11c), respectively.

RESULTS AND DISCUSSION

To demonstrate the accuracy of the present method with respect to K_g and K_s, we will perform two tests: use the method above to compute the velocity field at an array of test points within a radially-symmetric vortex blob, and simulate a 3-D lid-driven cavity until steady state is reached.

Single Vortex Blob

Comparing the present method to a flow with an analytical solution will determine the order of the error with respect to the number of solution points (K_s). For a compact radially-symmetric vorticity, with its peak at the origin, defined as

$$r = \sqrt{x^2+y^2+z^2} \quad (18a)$$

$$A(a,r) = (a-\min(a,r))^5 \quad (18b)$$

$$\omega_z(r) = \frac{A(3,r)-6A(2,r)+15A(1,r)}{120\pi} \quad (18c)$$

the analytical velocity is

$$\overline{B}(b,r) = b-\min(b,r) \quad (19a)$$

$$B(b,r) = \frac{b^8-\overline{B}^6(b,r)\left(21\overline{B}^2(b,r)-48b\overline{B}(b,r)+28b^2\right)}{168} \quad (19b)$$

$$\boldsymbol{u}(\boldsymbol{x}) = (-y,x,0)\,\frac{B(3,r)-6B(2,r)+15B(1,r)}{120\pi r^3}. \quad (19c)$$

Uniform meshes cover the range $[-4,4]$ in $\mathbb{R}^3$ with 2^m elements, where $2 \geq m \geq 7$. Warped versions of these meshes undergoing the following transformation are also benchmarked in this work:

$$d = 0.75\sin\left(\frac{\pi}{4}x\right)\sin\left(\frac{\pi}{4}y\right)\sin\left(\frac{\pi}{4}z\right) \quad (20a)$$

$$\boldsymbol{x}_{\text{warp}} = \boldsymbol{x}+(d,d,d)\,. \quad (20b)$$

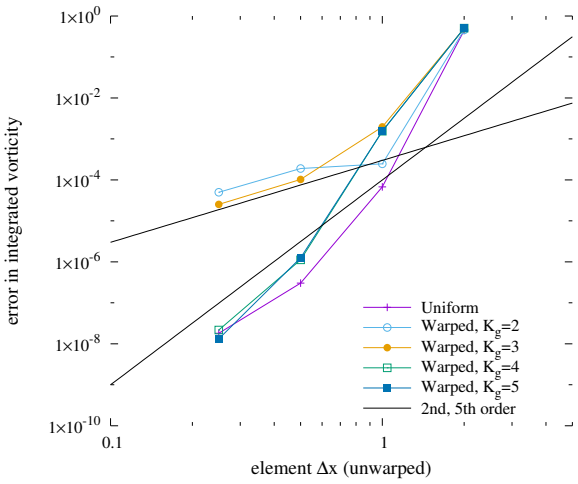

FIGURE 2. ERROR IN INTEGRATED VORTICITY VS. ELEMENT SIZE, $K_s = 1$.

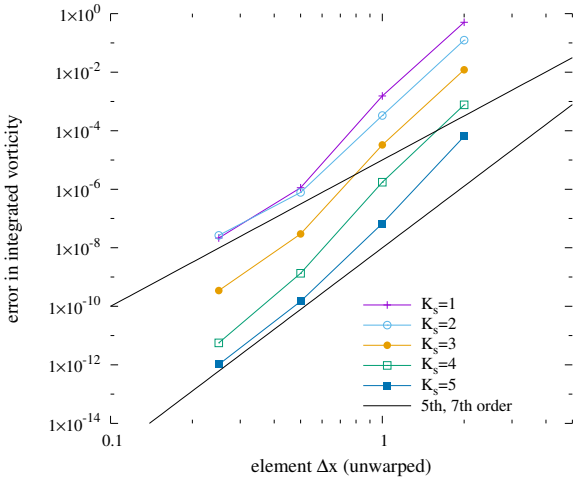

FIGURE 3. ERROR IN INTEGRATED VORTICITY VS. ELEMENT SIZE FOR 3RD ORDER WARPED MESH.

The vorticity at each solution node is set by Eq. (18c), and the integrated vorticity in each element in each cell obtained as follows

$$\boldsymbol{\Gamma}^{el} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}\boldsymbol{\omega}^{el}(\boldsymbol{\xi})\,J^{el}(\boldsymbol{\xi})\,d\xi\,d\eta\,d\zeta \quad (21)$$

Figures 2-3 show the errors in the discretization of the vorticity field on the solution nodes. With only one solution node per element ($K_s = 1$, Fig. 2), the error in integrated vorticity for the uniform mesh drops rapidly, as the center and volume of each

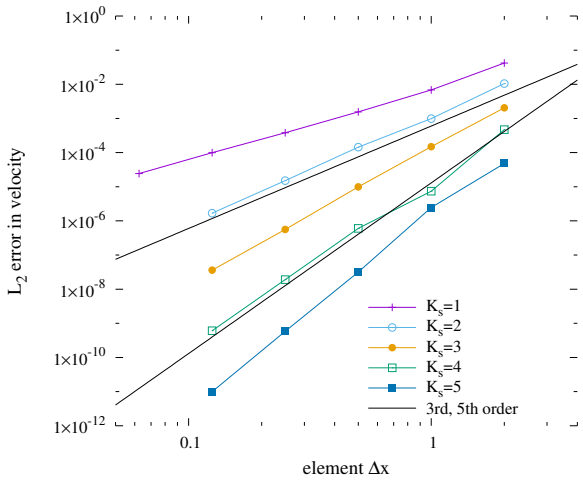

FIGURE 4. L_2 ERROR IN VELOCITY VS. ELEMENT SIZE, UNIFORM MESH.

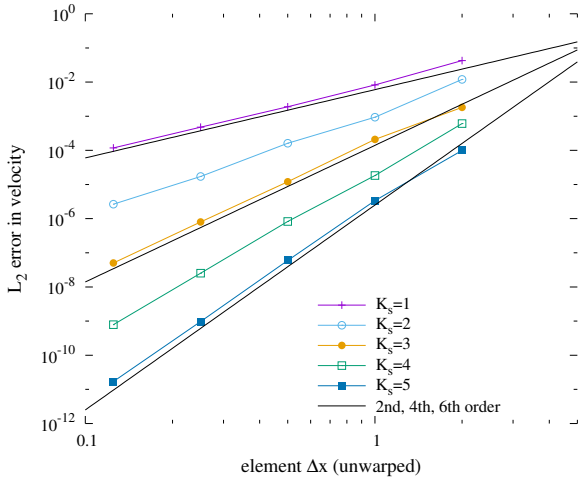

FIGURE 5. L_2 ERROR IN VELOCITY VS. ELEMENT SIZE FOR 3RD ORDER WARPED MESH.

element are known exactly. For the warped mesh, though, both $K_g = 2$ (linear) and $K_g = 3$ (quadratic) meshes exhibit 2nd order convergence, whereas $K_g \geq 4$ converge much more rapidly. Fig. 3 shows the effect of K_s on the error in integrated vorticity for the same 3rd order warped mesh ($K_g = 4$); here all *solution* orders allow convergence at 5th to 7th order.

The true test is how well the present method replicates the theoretical velocity field from Eq. (19c). To this end, one full evaluation of the Biot-Savart integration using cubatures is performed for each of 20^3 target points in the positive octant $[0,4]$ for each combination of solution (K_s) and mesh (K_g) node count. Important results appear in Figs. 4-5. Here we see that the method achieves desired accuracy on uniform meshes, with the L_2-norm of the error in the velocity evaluations approaching zero at (K_s+1) order. A similar result appears in Fig. 5 for the case with a 3rd order warped mesh: all values of K_s converge to the analytical solution at (K_s+1) order, with magnitudes similar to the uniform mesh case.

Lid-Driven Cavity

The lid-driven cavity is a common test of methods for convection-diffusion equations. In this section we demonstrate the effectiveness of the current method on the three-dimensional lid-driven (cubic) cavity. Previous results can be drawn from spectral methods solutions from Ku *et al.* [22] and a 2nd-order finite difference solution of a 3-D vorticity vector-potential method by Chen & Xie [23] for cubic cavities at Re = 100, 400, 1,000. Note, we have not yet published the details of our algorithm for a 3-D high-order VTE solver based on the Flux Reconstruction method of Huynh [24, 25]. However, it is an extension of a 2-D Flux Reconstruction method for VTE, which was published earlier and is available as reference for a general understanding of the method and its accuracy [5].

To study the effect of order K_s on the solution, we ran dynamic simulations of the most diffusive case (Re = 100) with 4^3 elements and $\Delta t = 0.02$ for all $K_s < 5$; $K_s = 5$ required $\Delta t = 0.01$. Flow properties on the solution nodes at $t = 10$ were projected to the geometry nodes of the mesh; each mesh was one order less than the solution nodes ($K_s = 2$ used a 1st order mesh with 2 nodes per linear edge). Note that for this case, Ku *et al.* [22] use $25 \times 25 \times 13$ modes (fewer modes along the direction of the primary vortex) and $\Delta t = 5 \times 10^{-4}$, requiring about 7,500 time steps ($t = 3.75$) to reach stationary state. Chen & Xie [23] reported neither the number of grid points nor the time step size for these simulations.

Velocities along two centerlines ($\hat{x}$, the direction of the lid motion, and $\hat{z}$, the axis perpendicular to the lid) for these runs appear in Fig. 6. The $K_s = 2$ case is clearly insufficient to resolve the flow, only capturing the basic character and magnitude of the circulation. Larger K_s shows very little variation, with $K_s = 4$ and $K_s = 5$ overlapping with themselves and previous results. This is not surprising, as $K_s = 4$ contains twice the spatial resolution of $K_s = 2$ and eight times the *total* degrees of freedom (number of solution nodes).

In addition, we investigated several cases with similar degrees of freedom, all at Re = 400: $K_s = 2$ with 15^3 cells and $\Delta t = 0.025$, $K_s = 3$ with 10^3 cells and $\Delta t = 0.02$, $K_s = 4$ with 8^3 cells and $\Delta t = 0.01\bar{3}$, and $K_s = 5$ with 6^3 and $\Delta t = 0.01$. These results appear in Fig. 7. Again, the present method shows excellent agreement, especially for $K_s > 2$. A close-up reveals that as K_s increases, the curves approach the results of Chen & Xie [23], but not Ku *et al.* [22] (which were carefully digitized from a photocopy, and may not be as precise as required).

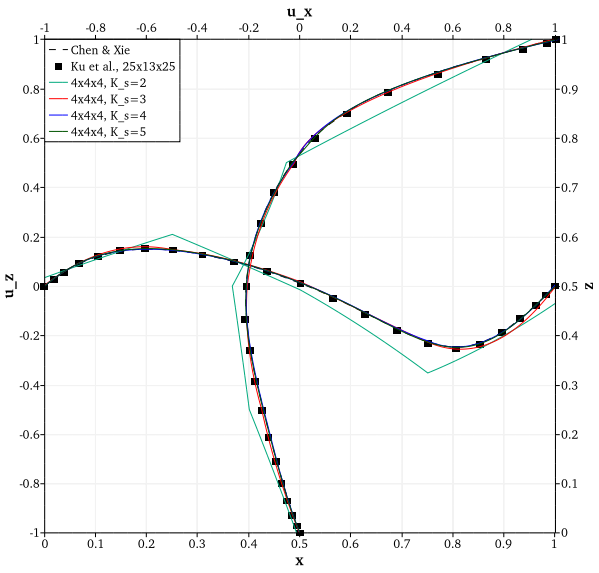

FIGURE 6. 3-D LID-DRIVEN CAVITY, RE=100, VELOCITY ALONG CENTERLINES, $T = 10$.

Finally, the least-diffusive case (Re = 1,000) was run with $K_s = 3$ and 4 and compared to previous results in Fig. 8. This run used 16^3 cells, $K_s = 3$ used $\Delta t = 0.01$ and $K_s = 4$ used $\Delta t = 0.008$, and results are reported at $t = 40$. For this larger Reynolds number, Ku *et al.* [22] use $31 \times 31 \times 16$ modes and $\Delta t = 5 \times 10^{-4}$, where stationary state was reported to have been achieved after 25,000 time steps ($t = 12.5$). In this case, results from the present simulations converge to curves closer to the results of Ku *et al.* [22] than of Chen & Xie [23], though again the reader should be warned of the imprecise nature of digitizing the results of the former.

PERFORMANCE OPTIMIZATION

Depending on the relative or absolute error thresholds, the numerical cubature may require $O(10^8)$ evaluations of the Biot-Savart kernel. While relaxing these thresholds will increase the performance of the above method, leveraging parallelization, caching, and other methods can return a combined $O(10^2)$ speedup compared to the serial, unoptimized case.

Parallelism is improved via both multithreading with OpenMP and manual vectorization using Vc [26]. OpenMP parallelism is achieved using the `pragma omp for` construct (with `schedule(dynamic)`) over the outer loop of target cells. Each thread, then, integrates the Biot-Savart formula over the entire fluid domain for all of the target solution nodes in a single high-order cell. The loop also starts with the target cell's self-influence, as this is easily the most costly cell-cell interaction to compute. Since there are generally an order of magnitude more cells than

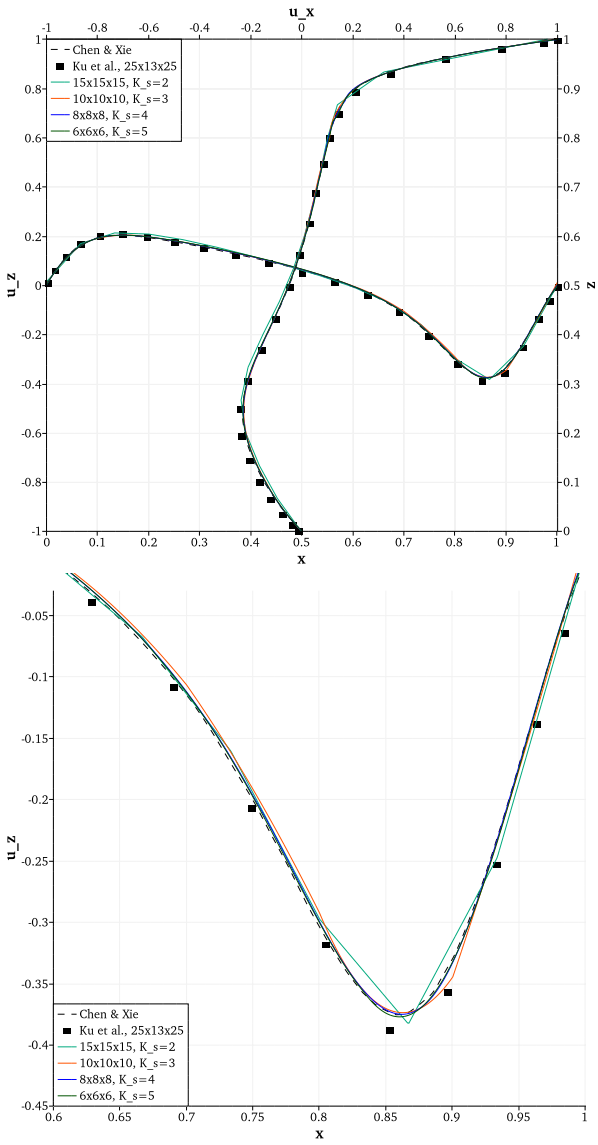

FIGURE 7. 3-D LID-DRIVEN CAVITY, RE=400, VELOCITY ALONG CENTERLINES, $T = 20$.

threads, and because the algorithm front-loads the work for each work block, the overall parallel efficiency remains quite high.

Further parallelism is achieved via explicit (super-scalar) vectorization. Because each call to the kernel from the cubature integration is rather involved, compilers do not always automatically recognize and vectorize them. Thus, we used *Vc*'s packed floating-point data types inside of `cubature`'s h-adaptive vector integrand, which allowed computing four values at once (for double-precision and when 256-bit registers and AVX instructions are available). Explicitly vectorizing the kernel improves performance in cases which require large numbers of kernel calls

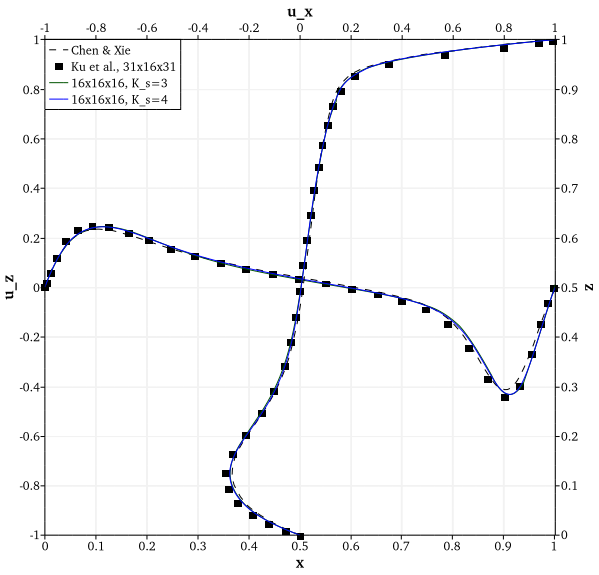

FIGURE 8. 3-D LID-DRIVEN CAVITY, RE=1,000, VELOCITY ALONG CENTERLINES, $T = 40$.

to complete the cubature.

While the geometry matrix for each cell is dense for a general high-order curvilinear cell, many cases use meshes with regular geometry. In these cases, the geometry matrix contains many zeros, and some of the arithmetic in the integration kernel becomes unnecessary. The algorithm detects this on a cell-by-cell basis and sends a flag to the kernel indicating that it can skip some of the loops. While this optimization had only a small effect on this test problem, in real cases, it can increase performance by up to 20% versus the full calculation with no effect on the result.

Recognizing that the self-influence of a cell is the most costly of the cell-cell influence calculations, and that meshes often do not change shape or conformation during a simulation, we can pre-calculate and cache a coefficient matrix for a specific cell's self-interaction. Each matrix contains $(3K_s^3)^2$ numbers and, for $K_s = 3$ and double precision, consumes 51 kB of storage ($K_s = 4$ cells require 288 kB). Thus, during the velocity evaluation step, a simple matrix-vector operation replaces the costly self-interaction cubatures. If all cells share identical geometry, the coefficient matrix for each cell is identical, and it is calculated and stored only once and shared. This method provides a $2\times \rightarrow 3\times$ speed improvement once the simulation is underway, depending on the number of cells, the order $K_s - 1$, and the shape of the vorticity field (though the performance benefit on our test case is more muted). Obviously, more than just self-interactions may be cached, though the performance benefit is not as substantial and the storage requirement increases quickly.

Finally, because the kernel influence falls off as r^2, where $r = |\boldsymbol{x}^{el}(\boldsymbol{\xi}) - \boldsymbol{x}|$, certain source cells that are far enough away from the target cell can use a lower-accuracy integration. Instead of a full cubature evaluation, these cell-cell interactions lump the source cell's vorticity distribution onto K_s^3 equivalent particles, and the much simpler particle-particle Biot-Savart formulation is performed. The threshold for this simpler calculation is the "box-opening criterion" common to particle treecodes: where the distance between the two cells divided by the maximum diagonal size of the cells is greater than some value—here 3.5. On small (8^3) problems, very few cell-cell interactions breach this threshold (7.5%), but this increases substantially for larger problems (75.3% for 16^3).

TABLE 1. SOLUTION TIMES (WALL CLOCK SECONDS) FOR ONE VELOCITY EVALUATION GIVEN VARIOUS OPTIMIZATIONS.

Optimization	M=8^3, $K_s = 3$	M=8^3, $K_s = 4$	M=16^3, $K_s = 4$
serial	66.03	332.2	15,720
OpenMP	5.157	22.84	1,134
above + Vc	2.763	8.079	459.9
above + geom	2.653	5.671	390.7
above + far-field	2.409	4.853	84.39
above + caching	2.381	4.672	83.41
total speedup	27.73	71.10	188.5

Table 1 summarizes the performance improvements provided by the above optimizations on the first validation test problem. These used meshes extending $[-4:4]$ and either $K_g = K_s = 3$ or $K_g = K_s = 4$. All runs were performed on a machine with a 16-core AMD 3950X CPU, using GCC 9.3.1 with `-Ofast -march=native`. Altogether, moderate-sized regular-grid solutions can be performed at least 100× faster with these optimizations than without.

CONCLUSIONS

A method is presented for evaluating the 3-D Biot-Savart singular volume integral used to obtain the velocity field induced by arbitrarily high-order (discontinuous) vorticity in arbitrarily high-order curved hexahedral elements. The formulation relies on Duffy's coordinate transformation and singularity removal strategy, leading to high-accuracy evaluation of the transformed volume integrals using standard adaptive cubature techniques. In this paper, the formulations for the new method are detailed followed by a series of benchmark tests, as well as strategies to optimize the performance of the integration module. Tests with a manufactured problem, for which an analytical formula for the

velocity field exists; i.e. , a single vortex blob, demonstrate that the L_2-norm of the errors in velocity evaluations converge with order (K_s+1) for piecewise polynomial distribution of vorticity sources of order (K_s-1), matching theoretical expectations *perfectly*. Most importantly, tests with severely warped cells show that both accuracy and convergence rates are maintained in this scenario. The Biot-Savart integration module was incorporated into a 3-D VTE solver, which was then benchmarked using the classical problem of lid-driven flow in a cubic cavity. Simulations at Re = $100, 400, 1{,}000$ using various K_s showed *excellent* comparisons of the centerline velocity profiles vis-à-vis published results. Finally, Biot-Savart volume integration of arbitrary-order vorticity sources in arbitrary-order curvilinear hexahedra is quite CPU intensive. To this end, performance optimization techniques such as OpenMP parallelization, vectorization, pre-caching, and others are discussed in this paper, showing up to *two orders of magnitude* speedup compared to the naïve serial version of the module.

ACKNOWLEDGMENT

Research reported in this publication was supported by the National Institute Of Biomedical Imaging And Bioengineering of the National Institutes of Health under Award Number R01EB022180. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

REFERENCES

[1] Brown, R. E., 2000. "Rotor wake modeling for flight dynamic simulation of helicopters". *AIAA Journal,* ***38***(1), pp. 57–63.

[2] Liu, J.-G., and Shu, C.-W., 2000. "A high-order discontinuous Galerkin method for 2D incompressible flows". *Journal of Computational Physics,* ***160***(2), pp. 577–596.

[3] Mozolevski, I., Süli, E., and Bösing, P., 2006. "Discontinuous Galerkin finite element approximation of the two–dimensional Navier-Stokes equations in streamfunction formulation". *Communications in Numerical Methods in Engineering,* ***23***, 06, pp. 447 – 459.

[4] Einkemmer, L., and Wiesenberger, M., 2014. "A conservative discontinuous Galerkin scheme for the 2D incompressible Navier–Stokes equations". *Computer Physics Communications,* ***185***(11), pp. 2865–2873.

[5] Gharakhani, A., 2021. "A High-Order Flux Reconstruction Method For 2-D Vorticity Transport". In Proceedings of the ASME 2021 Fluids Engineering Division Summer Meeting, no. FEDSM2021-63196.

[6] Suh, J.-C., 2000. "The evaluation of the Biot–Savart integral". *Journal of Engineering Mathematics,* ***37***(4), pp. 375–395.

[7] Wala, M. M., 2019. "High-order numerical methods for layer potential evaluation". PhD thesis, University of Illinois at Urbana-Champaign.

[8] Gao, X.-W., Feng, W.-Z., Yang, K., and Cui, M., 2015. "Projection plane method for evaluation of arbitrary high order singular boundary integrals". *Engineering Analysis with Boundary Elements,* ***50***, pp. 265–274.

[9] Yang, Q.-N., and Miao, Y., 2016. "An improved exponential transformation for accurate evaluation of nearly singular boundary integrals in 3D BEM". *Engineering Analysis with Boundary Elements,* ***71***, pp. 27–33.

[10] Lv, J.-H., Feng, X.-T., Chen, B.-R., Jiang, Q., and Guo, H.-S., 2016. "The CPCT based CBIE and HBIE for potential problems in three dimensions". *Engineering Analysis with Boundary Elements,* ***67***, pp. 53–62.

[11] Nunes, A., Chadebec, O., Kuo-Peng, P., Dular, P., and Cucco, B., 2018. "Contribution on the source field calculation through the Biot-Savart equation using curvilinear elements and an adaptive process". *Progress In Electromagnetics Research C,* ***88***, pp. 145–161.

[12] Selcuk, G., Kurt, S., and Koc, S. S., 2015. "Solution of volume integral equations with novel treatment to strongly singular integrals". In 2015 9th European Conference on Antennas and Propagation (EuCAP), IEEE, pp. 1–4.

[13] Varsamis, D., and Karampetakis, N., 2014. "Transformations between bivariate polynomial bases". *International Journal of Computational and Mathematical Sciences,* ***8***, 01, pp. 1204 – 1207.

[14] Manić, A. B., Djordjević, M., and Notaroš, B. M., 2014. "Duffy method for evaluation of weakly singular SIE potential integrals over curved quadrilaterals with higher order basis functions". *IEEE Transactions on Antennas and Propagation,* ***62***(6), pp. 3338–3343.

[15] Duffy, M. G., 1982. "Quadrature over a pyramid or cube of integrands with a singularity at a vertex". *SIAM Journal on Numerical Analysis,* ***19***(6), pp. 1260–1262.

[16] Mousavi, S., and Sukumar, N., 2010. "Generalized Duffy transformation for integrating vertex singularities". *Computational Mechanics,* ***45***(2), pp. 127–140.

[17] Wilton, D., Rao, S., Glisson, A., Schaubert, D., Al-Bundak, O., and Butler, C., 1984. "Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains". *IEEE Transactions on Antennas and Propagation,* ***32***(3), pp. 276–281.

[18] Ding, W., and Wang, G., 2009. "Treatment of singular integrals on generalized curvilinear parametric quadrilaterals in higher order method of moments". *IEEE Antennas and Wireless Propagation Letters,* ***8***, pp. 1310–1313.

[19] Jørgensen, E., Volakis, J. L., Meincke, P., and Breinbjerg, O., 2004. "Higher order hierarchical Legendre basis functions for electromagnetic modeling". *IEEE Transactions on Antennas and Propagation,* ***52***(11), pp. 2985–2995.

[20] Yuan, H., Wang, N., and Liang, C., 2009. "Combining the higher order method of moments with geometric modeling by NURBS surfaces". *IEEE Transactions on Antennas and Propagation,* ***57***(11), pp. 3558–3563.

[21] Johnson, S. G., 2020. "Multi-Dimensional Adaptive Integration (Cubature) in C". `https://github.com/stevengj/cubature`.

[22] Ku, H. C., Hirsh, R. S., and Taylor, T. D., 1987. "A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations". *J. Comput. Phys.,* ***70***(2), pp. 439–462.

[23] Chen, Y., and Xie, X., 2016. "Vorticity vector-potential method for 3D viscous incompressible flows in time-dependent curvilinear coordinates". *J. Comput. Phys.,* ***312***, pp. 50–81.

[24] Huynh, H. T., 2007. "A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods". In 18th AIAA Computational Fluid Dynamics Conference, no. AIAA-2007-4079.

[25] Huynh, H. T., 2009. "A Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin for Diffusion". In 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, no. AIAA-2009-403.

[26] Kretz, M., and Lindenstruth, V., 2012. "Vc: A C++ Library for Explicit Vectorization". *Software: Practice and Experience,* ***42***(11), pp. 1409–1430.