Journal of Computational Physics (2023)

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Higher order divergence-free and curl-free interpolation on MAC grids

Ritoban Roy-Chowdhury[a,b], Tamar Shinar[a,*], Craig Schroeder[a,*]

[a]*Department of Computer Science and Engineering, University of California, Riverside, 351 Winston Chung Hall, Riverside, CA 92521-0429, USA*

[b]*Department of Computer Science and Engineering, University of California, San Diego, 9500 Gilman Drive, Mail Code 0404, La Jolla, CA 92093-0404, USA*

ARTICLE INFO

Article history:

ABSTRACT

Divergence-free vector fields and curl-free vector fields play an important role in many types of problems, including the incompressible Navier-Stokes equations, Maxwell's equations, the equations for magnetohydrodynamics, and surface reconstruction. In practice, these fields are often obtained by projection, resulting in a discrete approximation of the continuous field that is discretely divergence-free or discretely curl-free. This field can then be interpolated to non-grid locations, which is required for many algorithms such as particle tracing or semi-Lagrangian advection. This interpolated field will not generally be divergence-free or curl-free in the analytic sense. In this work, we assume these fields are stored on a MAC grid layout and that the divergence and curl operators are discretized using finite differences. This work builds on and extends [39] in multiple ways: (1) we design a divergence-free interpolation scheme that preserves the discrete flux, (2) we adapt the general construction of divergence-free fields into a general construction for curl-free fields, (3) we extend the framework to a more general class of finite difference discretizations, and (4) we use this flexibility to construct fourth-order accurate interpolation schemes for the divergence-free case and the curl-free case. All of the constructions and specific schemes are explicit piecewise polynomials over a local neighborhood.

1. Introduction

In many engineering and scientific applications vector fields arise that satisfy a derivative constraint. For example, in the Navier-Stokes equations, the velocity field is divergence free, satisfying $\nabla \cdot \mathbf{u} = 0$. In Maxwell's equations, the magnetic field is divergence-free, $\nabla \cdot \mathbf{B} = 0$. Curl-free vector fields also arise in practical applications, including magnetic field mapping [41], surface reconstruction [23], and for fluid flow involving thermal conduction [11]. Throughout this paper, we will use the term *derivative constraint* to refer to either the divergence-free condition or

*Corresponding authors: email: shinar@cs.ucr.edu, craigs@cs.ucr.edu

the curl-free condition. We do not attempt to satisfy both simultaneously. Each scheme that we develop is designed strictly for one or the other. In practice, the problems of interpolating divergence and curl-free vector fields are intricately linked, and works often tackle both problems together.

Often, in algorithms for numerical solutions of these physical problems, the vector field is stored on a Marker-and-Cell (MAC) grid, and a projection step projects the field so that it discretely satisfies that the derivative constraint. This data often must subsequently be interpolated to arbitrary points in the domain (such as for particle tracing or semi-Lagrangian advection), but the interpolated vector field will not in general satisfy the derivative constraint. In this paper, we develop interpolation schemes that *do* satisfy the derivative constraint when the data being interpolated satisfies the constraint discretely.

Many previous works have developed methods for divergence-free and curl-free interpolation. These approaches generally fall into a few broad categories. For unstructured data, methods based on radial basis functions (RBFs) are commonly employed for divergence-free interpolation [27, 17], curl-free interpolation [23, 41], or both [1, 22, 21, 14]. For multiresolution problems, wavelet-based approaches have been similarly developed for interpolation of fields that are divergence-free [15], curl-free [33], or both [44]. Other works have developed wavelet-based interpolation schemes that satisfy boundary conditions in flow problems [20, 42, 32]. A continuous divergence-free interpolation scheme for unstructured data, for use in magnetohydrodynamics problems on toroidal geometry, was developed by Yang et al. [45] and requires a global constrained least-squares solve.

For interpolating structured data, several works take advantage of the fact that the gradient of a scalar potential is curl free, and that curl of a vector potential is divergence free. Interpolating the potential and analytically differentiating the interpolant then produces a vector field that satisfies the derivative constraint [28, 38, 13, 40, 19]. These methods typically use a global solve to compute the potential. Rather than a global solve, Solin et al. [41] train a network-based model incorporating a latent potential field variable.

Another class of approaches use spline-based interpolants, but require a solving a global linear system for the coefficients. Duchon [24] studied splines minimizing semi-norms in Sobolev spaces. Later, Handscomb [29] used natural cubic splines and a global linear solve to perform divergence-free interpolation. Yassine and McGraw [46] performs tensor interpolation using Bezier patches. Their approach is closely related to ours, as they use the same B-spline property (Eqn. 2) as we do. However, in their approach control points are determined through global sparse linear system, solved using a least squares approach, so the derivative constraint is only approximately enforced.

Finally, in contrast to these global spline-based interpolation schemes, another class of works, including the schemes proposed in the present work, utilize local splines. These approaches do not require the solution of a globally coupled system, so are more efficient and simpler to implement. Tóth and Roe [43] considered both divergence-free and curl-free restriction and prolongation operators for adaptive or hierarchical mesh grid algorithms and considered both MAC and colocated grids. Similarly a series of works by Balsara and Spicer [8], Balsara [2], Balsara and Kim [6], Balsara [3], Dumbser et al. [25] develops polynomial-based divergence-free interpolation methods for applications to magnetohydrodynamics. However, their interpolants are discontinuous in the tangential components at grid faces. Balsara et al. [11] extended this approach to curl-free interpolation, also with discontinuities in the normal component at dual grid cells.

The above spline-based schemes are second order accurate, but there has also been some interest in developing interpolation schemes with higher orders of accuracy. Balsara [4] extends their divergence free interpolation from Balsara [3] scheme up to 4th order accuracy by using higher-order Legendre polynomials as basis functions and WENO for approximating derivatives. Balsara et al. [11] uses an analogous approach for curl-free interpolation to achieve 4th order accuracy while [5] applies it to triangular elements, and [10] applies it to electrodynamics problems. Recent work [12] extended this approach to incorporate a source term, producing an interpolant $\mathbf{u}$ that satisfies $\nabla \cdot \mathbf{u} = \rho$, for some function ρ specified on grid cells. This was then subsequently applied to computational electrodynamic problems by Balsara and Sarris [7], to interpolate the electric displacement field $\mathbf{D}$ in an adaptive mesh refinement algorithm. Li [34] and Li [35] develop $3^{\text{rd}}$and 4^{th}-order divergence-free interpolation schemes, respectively. These approaches are local and spline based, but they are discontinuous in the tangential direction at cell faces. The interpolation scheme for unstructured data developed by Yang et al. [45] is also highly accurate, claiming convergence substantially faster than cubic spline interpolation

It is worth noting that many of the schemes referenced above for divergence-free scenarios are discontinuous in the tangential direction while many of the curl-free methods are discontinuous in the normal direction. This is at least partly a result of design tradeoffs and application requirements. Normal continuity suffices for $H(div)$ and is suitable for many applications. For example, Balsara et al. [9] was devised for use in Godunov schemes,

where fields are reconstructed from cell-averaged data and the ability to recover small jumps in the reconstruction is essential. Likewise, Hazra et al. [30] was devised for use in a discontinuous Galerkin scheme and achieves high order accuracy. For our applications, however, we generally favor a continuous interpolation. We found in our derivation that the div-free (or curl-free) constraint combined with C^0 continuity everywhere imposes significant limitations on what additional properties are possible for an interpolation scheme, preventing higher than second order interpolation accuracy with a second order accurate finite difference stencil. Reconstruction schemes and schemes that lack C^0 continuity, even if only slightly, are not necessarily limited in this way. Indeed, such schemes exist in the literature [4, 11, 30, 34, 35].

~~In contrast, *all* of the~~The schemes proposed in this paper (both div-free and curl-free) are everywhere C^0 or C^1 continuous ~~at all locations~~ in all components and in all directions. Full continuity in combination with div-free or curl-free imposes significant restrictions on the degrees of freedom available in constructing an interpolation scheme. In fact, the design space of such schemes is so small that it appears to be feasible to actually exhaustively enumerate them, as was effectively done by Schroeder et al. [39]. This work generalizes that framework to curl-free schemes and also schemes with higher order finite differencing stencils.

Indeed, these discontinuities are essential for higher-order Godunov schemes (i.e. [30]) and Discontinuous Galerkin schemes (i.e. [35, 30]). Further, discontinuities may be valuable for shock-capturing [9]. On the other hand, for the applications we're targeting, such as particle advection, continuity is desirable.

The present work builds on the approach to divergence-free interpolation developed in Schroeder et al. [39]. At a high level, Schroeder et al. [39] represents the interpolated vector field as sums of tensor products of polynomials whose derivatives take a particular form. As a result, the divergence of the interpolant is equal to the interpolation of the discrete divergences, so that if the interpolated data is discretely divergence free, then the interpolant will be analytically divergence free. In this paper, we build on their approach in four ways:

1. **Flux consistency.** We present an interpolation scheme in which the total flux of the interpolant, computed by integrating over the face of the MAC cell, exactly equals the value stored at that face times the face area.
2. **Curl-free interpolation.** We modify the general construction of divergence-free interpolants to the curl-free case.
3. **Generalizing the chain construction.** Hokpunna and Manhart [31] and Schroeder et al. [39] speculated that to achieve higher order accuracy in a continuous divergence-free interpolation scheme, a higher order finite difference stencil would need to be used in the construction the interpolation scheme. Following this, we extend their notion of a *chain* of polynomials to arbitrary finite difference stencils, and show that using such a modified spline chain, the divergence and curl are still the interpolation of the discrete divergence and curl.
4. **Fourth-order accurate divergence and curl-free interpolation.** Finally, we apply the generalized chain construction to a particular fourth-order finite difference stencil and use this to construct fourth-order divergence and curl-free interpolation schemes.

1.1. Summary of Schroeder et al. [39]

The interpolation schemes we present in this paper are generalizations and variations of those presented in Schroeder et al. [39]. As such, we begin with a brief summary of their interpolation schemes.

Given a MAC grid (in 2D) with data $u_{i+\frac{1}{2},j}$ and $v_{i,j+\frac{1}{2}}$, the goal is construct a continuous vector field $\hat{\mathbf{u}}(\mathbf{x}) = \langle \hat{u}(x,y), \hat{v}(x,y) \rangle$ with the properties that

1. $\hat{u}(x,y)$ is a piecewise polynomial that depends on $u_{i+\frac{1}{2},j}$, and $\hat{v}(x,y)$ is a piecewise polynomial that depends on $v_{i,j+\frac{1}{2}}$.
2. The functions have local support, in the sense that the evualation at a point (x,y) depends only on MAC data in the local neighborhood of the point.
3. The construction of the vector field is symmetrical.
4. The scheme is a divergence-free scheme; that is, $\nabla \cdot \hat{\mathbf{u}} = 0$ when the MAC grid data is discretely divergence free in the sense of

$$d_{i,j} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y} = 0. \tag{1}$$

5. The vector field interpolates the data. In practice, it suffices for $\hat{\mathbf{u}}(\mathbf{x})$ to exactly recover affine data.

The construction in Schroeder et al. [39] allows one to construct a basis for all vector fields (of bounded polynomial degree and support) satisfying the first four properties. Other constraints (including the interpolation property) are then used to eliminate the remaining degrees of freedom.

The construction of divergence-free vector fields begins with the concept of a *chain R*, a sequence of polynomial splines $R^0(x), R^1(x), \ldots$ satisfying the property

$$\frac{d}{dx}R^{n+1}(x) = R^n\left(x+\tfrac{1}{2}\right) - R^n\left(x-\tfrac{1}{2}\right). \tag{2}$$

The simplest example of a chain is the B-splines. In general, the splines $R(x)$ are assumed to be continuous, have compact support, and be symmetrical. They are not otherwise restricted.

Schroeder et al. [39] showed that for chains R and S, a general construction of a divergence-free vector field in 2D is given by

$$\hat{u}(x,y) = \sum_{i,j} u_{i+\frac{1}{2},j}\left[R^{m+1}\left(\frac{x-x_{i+\frac{1}{2}}}{\Delta x}\right)S^n\left(\frac{y-y_j}{\Delta y}\right) + S^{n+1}\left(\frac{x-x_{i+\frac{1}{2}}}{\Delta x}\right)R^m\left(\frac{y-y_j}{\Delta y}\right)\right] \tag{3}$$

$$\hat{v}(x,y) = \sum_{i,j} v_{i,j+\frac{1}{2}}\left[S^n\left(\frac{x-x_i}{\Delta x}\right)R^{m+1}\left(\frac{y-y_{j+\frac{1}{2}}}{\Delta y}\right) + R^m\left(\frac{x-x_i}{\Delta x}\right)S^{n+1}\left(\frac{y-y_{j+\frac{1}{2}}}{\Delta y}\right)\right]. \tag{4}$$

This vector field satisfies the first four properties for any choice of chains R and S. In addition, linear combinations of such vector fields also satisfy the same properties, so that a basis for such vector fields can be constructed by enumerating linearly independent splines up to the desired degree and support with the appropriate level of continuity (C^0 or C^1 in our case). Expressing vector fields in this form is very convenient, since there are very few degrees of freedom in this representation compared with the space of polynomials generally, and schemes with desirable properties tend to have very sparse representations in this form. Because the splines $R^m, R^{m+1}, S^n, S^{n+1}$ all have compact support, the summations above will have a finite number of terms. For example, Figure 1 illustrates the stencils for the five schemes proposed in this paper.

Using this construction, the divergence of the vector field above is equal to the interpolation of the discrete divergences. Specifically,

$$\nabla\cdot\hat{\mathbf{u}}(\mathbf{x}) = \sum_{i,j} d_{i,j}\left[R^m\left(\frac{x-x_i}{\Delta x}\right)S^n\left(\frac{y-y_j}{\Delta y}\right) + S^n\left(\frac{x-x_i}{\Delta x}\right)R^m\left(\frac{y-y_j}{\Delta y}\right)\right] \tag{5}$$

where

$$d_{i,j} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y}. \tag{6}$$

Thus, the interpolant $\hat{\mathbf{u}}(\mathbf{x})$ is divergence-free when the $d_{i,j} = 0$. Eq. 6 is a second-order accurate discretization of the divergence and $d_{i,j} \approx 0$ is typically enforced through the numerical projection of the vector field.

In three dimensions, given chains R, S, and T, a general construction of a divergence-free field is given by

$$\begin{aligned}
\hat{u}(x,y,z) = \sum_{i,j,k} u_{i+\frac{1}{2},j,k}\Bigg[& R^{m+1}\left(\frac{x-x_{i+\frac{1}{2}}}{\Delta x}\right)S^n\left(\frac{y-y_j}{\Delta y}\right)T^p\left(\frac{z-z_k}{\Delta z}\right) \\
& + R^{m+1}\left(\frac{x-x_{i+\frac{1}{2}}}{\Delta x}\right)T^p\left(\frac{y-y_j}{\Delta y}\right)S^n\left(\frac{z-z_k}{\Delta z}\right) \\
& + S^{n+1}\left(\frac{x-x_{i+\frac{1}{2}}}{\Delta x}\right)R^m\left(\frac{y-y_j}{\Delta y}\right)T^p\left(\frac{z-z_k}{\Delta z}\right) \\
& + S^{n+1}\left(\frac{x-x_{i+\frac{1}{2}}}{\Delta x}\right)T^p\left(\frac{y-y_j}{\Delta y}\right)R^m\left(\frac{z-z_k}{\Delta z}\right) \\
& + T^{p+1}\left(\frac{x-x_{i+\frac{1}{2}}}{\Delta x}\right)R^m\left(\frac{y-y_j}{\Delta y}\right)S^n\left(\frac{z-z_k}{\Delta z}\right) \\
& + T^{p+1}\left(\frac{x-x_{i+\frac{1}{2}}}{\Delta x}\right)S^n\left(\frac{y-y_j}{\Delta y}\right)R^m\left(\frac{z-z_k}{\Delta z}\right)\Bigg],
\end{aligned} \tag{7}$$

where $\hat{v}(x,y,z)$ and $\hat{w}(x,y,z)$ are formed by cycling axes. Similar to the 2D case, the divergence of the 3D interpolant is also equal to an interpolation of discrete divergences. Note that the splines R^m, S^n, and T^p in the construction above can be chosen completely independently of each other without losing the div-free property or continuity; there are no mutual compatibility restrictions between them. However, other properties do impose such restrictions, such as the interpolation property, flux consistency, or higher order accuracy. These properties are achieved only with very special combinations of splines, which can be found by using the constraints to eliminate degrees of freedom. This independence of R^m, S^n, and T^p is a general property of all of the general constructions in this paper.

Eight schemes were proposed in Schroeder et al. [39], four of which were constructed from B-splines and non-interpolating. We will occasionally make reference to these four schemes, and will refer to them as $u^{div,C0}$ and $u^{div,C1}$, using the same name for both the 2D and 3D versions.

2. Flux-Consistent Interpolation

Projecting a given vector field to be divergence free is an inherently global operation, since correcting local deviations from divergence-free may require changes far away. Indeed, a field may only be divergence free everywhere if it satisfies appropriate compatibility conditions. Discretely divergence-free vector fields are obtained by projecting the vector fields, often through a pressure projection or solving a KKT system. When second order central differences are used for this projection, the notion of discretely divergence free takes a particularly simple and intuitive form. If the velocity of a face is assumed constant, then the net flux flowing into any cell of the grid is zero.

Local schemes such as those in Schroeder et al. [39] leverage the discretely divergence-free property to achieve analytically divergence-free interpolation. However, those schemes are not fully consistent with the discrete divergence in the sense that their fluxes across grid cells are different from those predicted by the discrete projection. That is, the net flux of the interpolated vector field across a cell is zero (as implied by being analytically divergence free), but those fluxes are not the same as the discrete notion of flux. While not normally a problem, this discrepancy would complicate the handling of irregular boundaries, since the notion of flux would change in cut cells. Flux consistency would allow a simple scheme to be applied away from the boundaries with a separate (and likely more expensive) cut-cell-aware scheme employed near the boundaries. Provided the interpolation is continuous at the transition and the boundary interpolation scheme is consistent with the cut-cell formulation of flux, the overall interpolation scheme would be globally consistent.

We define an interpolation scheme $\hat{u}$ to be flux-consistent at a face $x_{i+\frac{1}{2},y,z}$ if

$$u_{i+\frac{1}{2},j,k}\Delta y\Delta z = \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{u}\left(x_{i+\frac{1}{2}},y,z\right)\,dy\,dz, \tag{8}$$

and similarly for the $\hat{v}$ and $\hat{w}$ components. In 2D, the corresponding property is

$$u_{i+\frac{1}{2},j}\Delta y = \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{u}\left(x_{i+\frac{1}{2}},y\right)\,dy. \tag{9}$$

Such a flux-consistency constraint over faces of Voronoi cells was used was used by Boscheri et al. [17] to design a divergence-free interpolation scheme as part of an algorithm for simulating free-surface flows. The scheme proposed by [2] also has this flux-consistency property.

2.1. Flux-consistent analytically divergence-free scheme

One particularly simple flux-consistent analytically divergence-free interpolation scheme is (the precise definitions of the splines P^2 and P^3 will be given later)

$$\hat{u}(x,y,z) = \sum_{i,j,k} u_{i+\frac{1}{2},j,k}P^3\left(\frac{x-x_{i+\frac{1}{2}}}{\Delta x}\right)P^2\left(\frac{y-y_j}{\Delta y}\right)P^2\left(\frac{z-z_k}{\Delta z}\right) \tag{10}$$

$$\hat{v}(x,y,z) = \sum_{i,j,k} v_{i,j+\frac{1}{2},k}P^2\left(\frac{x-x_i}{\Delta x}\right)P^3\left(\frac{y-y_{j+\frac{1}{2}}}{\Delta y}\right)P^2\left(\frac{z-z_k}{\Delta z}\right) \tag{11}$$

$$\hat{w}(x,y,z) = \sum_{i,j,k} w_{i,j,k+\frac{1}{2}}P^2\left(\frac{x-x_i}{\Delta x}\right)P^2\left(\frac{y-y_j}{\Delta y}\right)P^3\left(\frac{z-z_{k+\frac{1}{2}}}{\Delta z}\right). \tag{12}$$

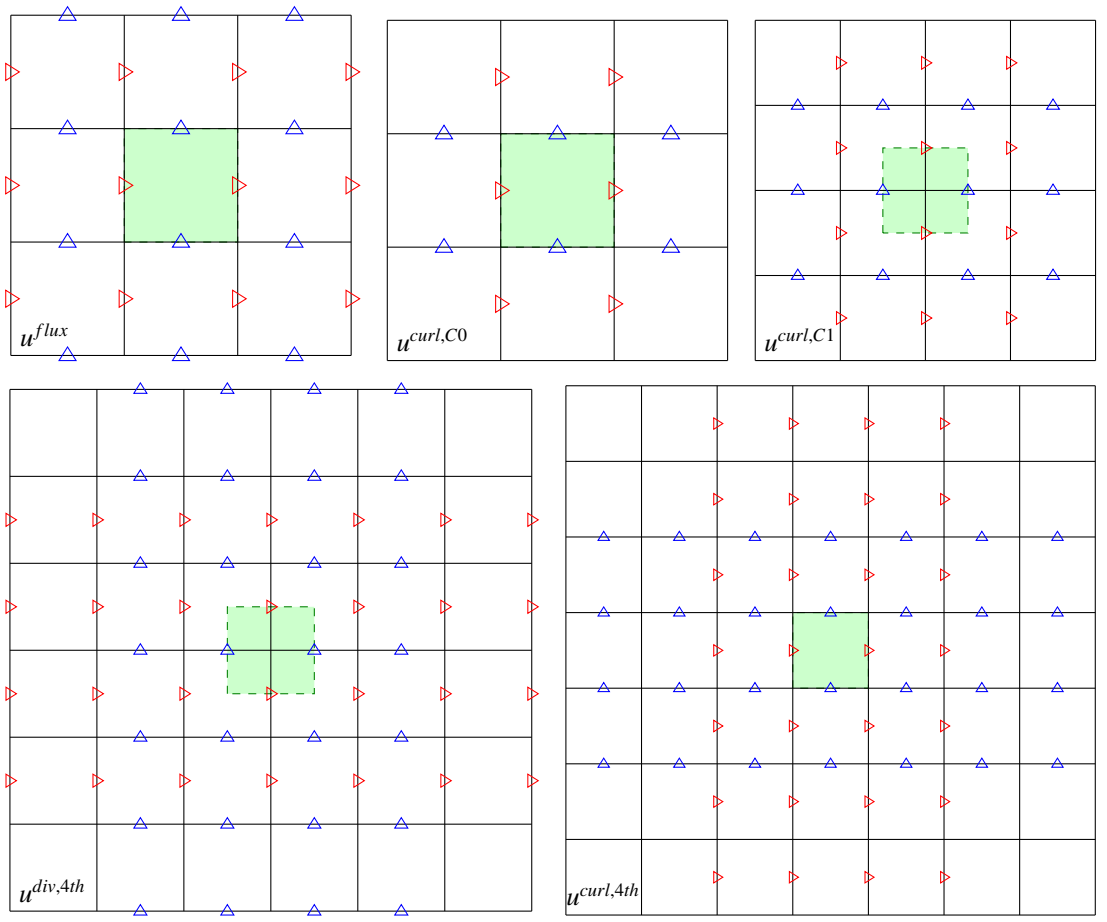


Fig. 1: The five interpolation schemes proposed in this paper interpolate data on a MAC grid, where normal vector components are stored at the centers of cell faces. The illustrations above show the stencils for each of the five schemes in 2D: u^{flux}, $u^{curl,C0}$, $u^{curl,C1}$, $u^{div,4th}$ and $u^{curl,4th}$. The shaded region shows the region of space over which the data is being interpolated. The x component depends on data at x-faces (▷), and the y component depends on data at y-faces (△). The stencils for 3D are similar. Observe that the region of influence for the 4$^{\text{th}}$ order schemes is elongated due to the application of the chain construction with a wider 4$^{\text{th}}$ order differencing stencil.

The 2D analog of this scheme has the same properties

$$\hat{u}(x,y) = \sum_{i,j} u_{i+\frac{1}{2},j} P^3\left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x}\right) P^2\left(\frac{y - y_j}{\Delta y}\right) \tag{13}$$

$$\hat{v}(x,y) = \sum_{i,j} v_{i,j+\frac{1}{2}} P^2\left(\frac{x - x_i}{\Delta x}\right) P^3\left(\frac{y - y_{j+\frac{1}{2}}}{\Delta y}\right). \tag{14}$$

These can be written in a form that is more convenient to implement

$$u^{flux}(x,y) = \sum_{ij} u_{i,j} P_i^3(x) P_j^2(y) \qquad u^{flux}(x,y,z) = \sum_{ijk} u_{i,j,k} P_i^3(x) P_j^2(y) P_k^2(z), \tag{15}$$

where the spline P^2 is given by

$$P_0^2 = \frac{(3x-1)(x-1)}{2} \qquad P_1^2 = -3x^2 + 3x + \frac{1}{2} \qquad P_2^2 = \frac{x(3x-2)}{2} \tag{16}$$

and the next spline in the chain P^3 is given by

$$P_0^3 = -\frac{x(x-1)^2}{2} \qquad P_1^3 = \frac{(x-1)(3x^2-2x-2)}{2} \qquad P_2^3 = -\frac{x(3x^2-4x-1)}{2} \qquad P_3^3 = \frac{x^2(x-1)}{2}. \tag{17}$$

Observe that this has the same simple form as the scheme $u^{div,C0}$ from Schroeder et al. [39], except that the B-splines are replaced with the splines P^n above. These schemes follow the divergence-free construction and are thus analytically divergence free. Next, we show that they are flux-consistent.

2.2. Flux-consistent analytically divergence-free scheme

Consider a scheme of the form

$$\hat{u}(x,y,z) = \sum_{i,j,k} u_{i+\frac{1}{2},j,k} A^{m+1}\left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x}\right) B^n\left(\frac{y-y_j}{\Delta y}\right) C^p\left(\frac{z-z_k}{\Delta z}\right). \tag{18}$$

This scheme is flux-consistent iff

$$\begin{aligned} u_{m+\frac{1}{2},n,p}\Delta y\Delta z &= \int_{z_{p-\frac{1}{2}}}^{z_{p+\frac{1}{2}}} \int_{y_{n-\frac{1}{2}}}^{y_{n+\frac{1}{2}}} \hat{u}\left(x_{m+\frac{1}{2}}, y, z\right) dy\, dz \qquad (19)\\ &= \int_{z_{p-\frac{1}{2}}}^{z_{p+\frac{1}{2}}} \int_{y_{n-\frac{1}{2}}}^{y_{n+\frac{1}{2}}} \sum_{i,j,k} u_{i+\frac{1}{2},j,k} A^{m+1}\left(\frac{x_{m+\frac{1}{2}} - x_{i+\frac{1}{2}}}{\Delta x}\right) B^n\left(\frac{y-y_j}{\Delta y}\right) C^p\left(\frac{z-z_k}{\Delta z}\right) dy\, dz \\ &= \sum_{i,j,k} u_{i+\frac{1}{2},j,k} A^{m+1}\left(\frac{x_{m+\frac{1}{2}} - x_{i+\frac{1}{2}}}{\Delta x}\right) \int_{y_{n-\frac{1}{2}}}^{y_{n+\frac{1}{2}}} B^n\left(\frac{y-y_j}{\Delta y}\right) dy \int_{z_{p-\frac{1}{2}}}^{z_{p+\frac{1}{2}}} C^p\left(\frac{z-z_k}{\Delta z}\right) dz \\ &= \Delta y\Delta z \sum_{i,j,k} u_{i+\frac{1}{2},j,k} A^{m+1}(m-i) \int_{n-j-\frac{1}{2}}^{n-j+\frac{1}{2}} B^n(y)\, dy \int_{p-k-\frac{1}{2}}^{p-k+\frac{1}{2}} C^p(z)\, dz \qquad (20) \end{aligned}$$

This implies

$$A^{m+1}(m-i) = \frac{1}{bc}\delta_{im} \qquad \int_{n-j-\frac{1}{2}}^{n-j+\frac{1}{2}} B^n(y)\, dy = b\delta_{jn} \qquad \int_{p-k-\frac{1}{2}}^{p-k+\frac{1}{2}} C^p(z)\, dz = c\delta_{kp} \tag{21}$$

for some constants b and c. Here δ_{im} is 1 if $i = m$ and 0 otherwise. We note that flux consistency in $\hat{v}$ and $\hat{w}$ adds similar constraints. These constraints can be summarized as

$$A^{m+1}(i) = \begin{cases} \frac{1}{bc} & i = 0 \\ 0 & i \neq 0 \end{cases} \qquad \int_{i-\frac{1}{2}}^{i+\frac{1}{2}} A^m(x)\, dx = \begin{cases} a & i = 0 \\ 0 & i \neq 0 \end{cases} \tag{22}$$

$$B^{n+1}(j) = \begin{cases} \frac{1}{ac} & j = 0 \\ 0 & j \neq 0 \end{cases} \qquad \int_{j-\frac{1}{2}}^{j+\frac{1}{2}} B^n(y)\, dy = \begin{cases} b & j = 0 \\ 0 & j \neq 0 \end{cases} \tag{23}$$

$$C^{p+1}(k) = \begin{cases} \frac{1}{ab} & k = 0 \\ 0 & k \neq 0 \end{cases} \qquad \int_{k-\frac{1}{2}}^{k+\frac{1}{2}} C^p(z)\, dz = \begin{cases} c & k = 0 \\ 0 & k \neq 0 \end{cases} \tag{24}$$

In our case, $A^{m+1} = B^{n+1} = C^{p+1} = P^3$ and $A^m = B^n = C^p = P^2$. From Figure 2, we see that these constraints follow from the easily-verified spline properties

$$P_1^3(0) = P_2^3(1) = 1 \qquad P_0^3(0) = P_0^3(1) = P_1^3(1) = P_2^3(0) = P_3^3(0) = P_3^3(1) = 0 \tag{25}$$

$$\int_0^1 P_1^2(x) = 1 \qquad \int_0^1 P_0^2(x) = \int_0^1 P_2^2(x) = 0. \tag{26}$$

Some of these properties can be readily observed from the graphs of P^2 and P^3 in Fig. 2. This demonstrates that the proposed interpolation scheme is flux-consistent. The proof for 2D is similar and omitted.

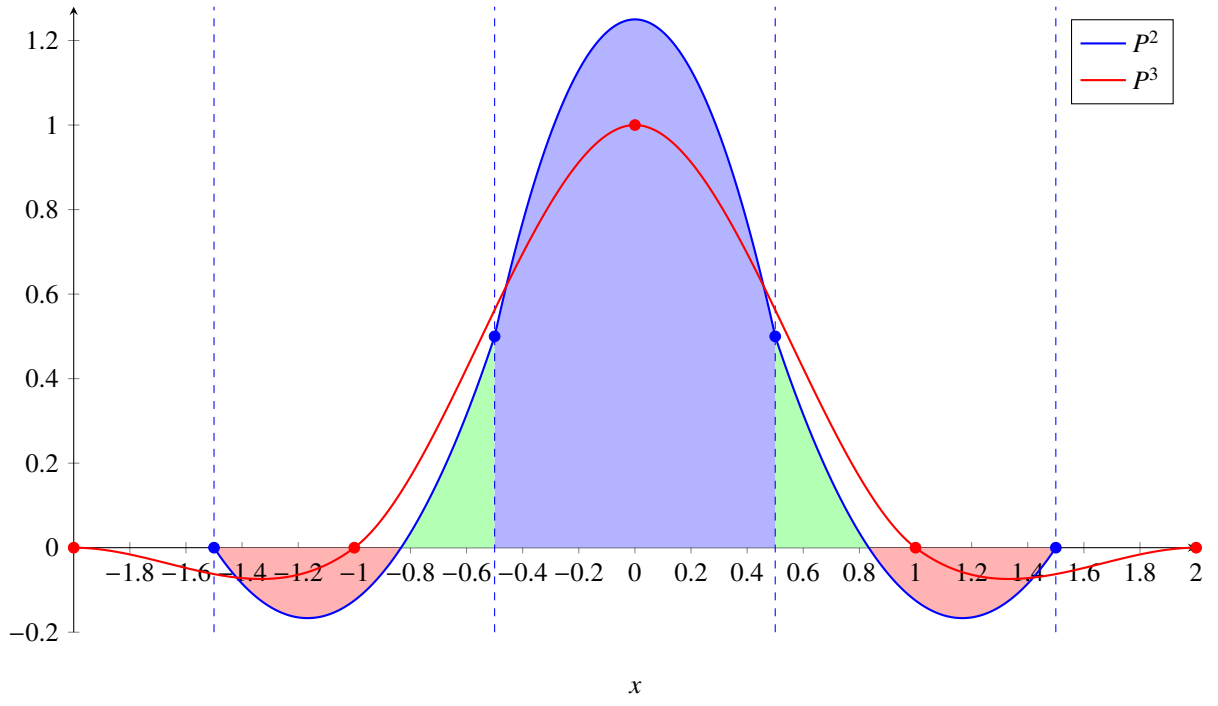


Fig. 2: The splines P^2 and P^3 graphed. P^3 is zero at nonzero integer x values (●), and $P^3(0) = 1$, as expected by the flux consistency constraints. It is not obvious merely from the graph, but it is also true that the area under P^2 is 0 for the two outer pieces (■■), and 1 for the central piece (■). The boundaries of each piecewise polynomial are marked with circles (●).

3. Curl-Free

3.1. General Construction

The general construction of divergence-free fields from [39] can be extended very naturally to curl-free fields. In particular, for any spline chains R^m, S^n, and T^p the vector field

$$\hat{u}_{RS}(x,y) = \sum_{i,j} u_{i-\frac{1}{2},j} R^m\left(\frac{x-x_i}{\Delta x}\right) S^{n+1}\left(\frac{y-y_{j-\frac{1}{2}}}{\Delta y}\right) \tag{27}$$

$$\hat{v}_{RS}(x,y) = \sum_{i,j} v_{i,j-\frac{1}{2}} R^{m+1}\left(\frac{x-x_{i-\frac{1}{2}}}{\Delta x}\right) S^{n}\left(\frac{y-y_j}{\Delta y}\right) \tag{28}$$

is analytically curl-free ($\hat{u}_y - \hat{v}_x = 0$) provided the data is discretely curl-free in the sense of

$$c_{i-\frac{1}{2},j-\frac{1}{2}} = \frac{u_{i-\frac{1}{2},j} - u_{i-\frac{1}{2},j-1}}{\Delta y} - \frac{v_{i,j-\frac{1}{2}} - v_{i-1,j-\frac{1}{2}}}{\Delta x} = 0. \tag{29}$$

The 3D equivalent is

$$\hat{u}_{RST}(x,y,z) = \sum_{i,j,k} u_{i-\frac{1}{2},j,k} R^m\left(\frac{x-x_i}{\Delta x}\right) S^{n+1}\left(\frac{y-y_{j-\frac{1}{2}}}{\Delta y}\right) T^{p+1}\left(\frac{z-z_{k-\frac{1}{2}}}{\Delta z}\right) \tag{30}$$

$$\hat{v}_{RST}(x,y,z) = \sum_{i,j,k} v_{i,j-\frac{1}{2},k} R^{m+1}\left(\frac{x-x_{i-\frac{1}{2}}}{\Delta x}\right) S^{n}\left(\frac{y-y_j}{\Delta y}\right) T^{p+1}\left(\frac{z-z_{k-\frac{1}{2}}}{\Delta z}\right) \tag{31}$$

$$\hat{w}_{RST}(x,y,z) = \sum_{i,j,k} w_{i,j,k-\frac{1}{2}} R^{m+1}\left(\frac{x-x_{i-\frac{1}{2}}}{\Delta x}\right) S^{n+1}\left(\frac{y-y_{j-\frac{1}{2}}}{\Delta y}\right) T^{p}\left(\frac{z-z_k}{\Delta z}\right). \tag{32}$$

Note the differences compared to the divergence-free case. In the divergence-free case, the component being interpolated has polynomial degree one higher than the other dimensions, since the divergence operator lowers the degree by one during differentiation. In the curl-free case, it is the opposite. The curl stencil differentiates along the axes that are different from the component being computed, so those polynomials must have higher degree instead. That is, in the divergence-free case we are interested in $u_x + v_y = 0$, while in the curl-free case we are interested in $u_y - v_x = 0$.

Unless $R^m = S^n = T^p$, the construction above will be asymmetrical. As in [39], this can be symmetrized by using

$$\hat{u}(x,y) = \hat{u}_{RS}(x,y) + \hat{u}_{SR}(x,y) \tag{33}$$

$$\hat{u}(x,y,z) = \hat{u}_{RRS}(x,y,z) + \hat{u}_{RSR}(x,y,z) + \hat{u}_{SRR}(x,y,z) \qquad \text{when } R = T \tag{34}$$

$$\begin{aligned}\hat{u}(x,y,z) = {} & \hat{u}_{RST}(x,y,z) + \hat{u}_{SRT}(x,y,z) + \hat{u}_{RTS}(x,y,z) \\ & + \hat{u}_{STR}(x,y,z) + \hat{u}_{TRS}(x,y,z) + \hat{u}_{TSR}(x,y,z).\end{aligned} \tag{35}$$

As in the divergence-free case, linear combinations of curl-free fields are also curl-free. Thus, symmetrized schemes will also be curl-free. The continuity of the interpolating scheme is inherited from the splines used to construct it. As far as we are aware, this construction is general in the sense that all local polynomial continuous curl-free schemes can be expressed as linear combinations of terms of the form above, though we do not have a proof of this.

In section 4.4 we extend this construction to generalizations of (29) and (2). We prove the analytic curl-free property in the more general case in Section 4.4.1.

3.2. Using B-Splines in Curl-Free

As in the divergence-free case, the B-splines form a very convenient interpolation scheme with a relatively simple form and compact stencil. The C^0 and C^1 continuous versions of this scheme are given by

$$u^{curl,C0}(x,y) = \sum_{ij} u_{i,j} B_i^1(x) B_j^2(y) \qquad u^{curl,C0}(x,y,z) = \sum_{ijk} u_{i,j,k} B_i^1(x) B_j^2(y) B_k^2(z) \tag{36}$$

$$u^{curl,C1}(x,y) = \sum_{ij} u_{i,j} B_i^2(x) B_j^3(y) \qquad u^{curl,C1}(x,y,z) = \sum_{ijk} u_{i,j,k} B_i^2(x) B_j^3(y) B_k^3(z). \tag{37}$$

The splines B^n are the standard B-splines, whose polynomials are given by

$$B_0^1 = 1 - x \qquad B_1^1 = x \tag{38}$$

$$B_0^2 = \frac{1}{2}(1-x)^2 \qquad B_1^2 = \frac{1}{2} + x - x^2 \qquad B_2^2 = \frac{1}{2}x^2 \tag{39}$$

$$B_0^3 = \frac{1}{6}(1-x)^3 \qquad B_1^3 = \frac{2}{3} - x^2 + \frac{1}{2}x^3 \qquad B_2^3 = \frac{1}{6} + \frac{1}{2}x + \frac{1}{2}x^2 - \frac{1}{2}x^3 \qquad B_3^3 = \frac{1}{6}x^3. \tag{40}$$

The stencils for these schemes (in 2D) are shown in Figure 1. These schemes are very similar to the corresponding divergence-free schemes $u^{div,C0}$ and $u^{div,C1}$, as can be observed from the path traces in Section 5.4.2.

4. Higher Order Interpolation

Schroeder et al. [39] presented second-order accurate divergence-free interpolation schemes. They also speculated that construction a divergence-free interpolation scheme with a higher order accuracy would not be possible without sacrificing continuity or using a different stencil for computing the discrete divergence. In this section, we do the latter, presenting a fourth-order accurate interpolation, which is analytically divergence-free when the data it interpolates is discretely divergence-free with respect to a fourth-order accurate discrete divergence stencil.

4.1. Difference Stencils

To perform divergence or curl-free interpolation at higher order accuracy, we need to specify a higher order difference stencil with respect to which our data is discretely divergence or curl free. We will assume that our stencil of interest contains M entries. Let $w_i/\Delta x$ be the differencing weights. For example, for the second order central differencing stencil ($M = 2$),

$$w_0 = -1 \qquad w_1 = 1. \tag{41}$$

For the standard fourth-order central differencing stencil ($M = 4$),

$$w_{-1} = \frac{1}{24} \qquad w_0 = -\frac{27}{24} \qquad w_1 = \frac{27}{24} \qquad w_2 = -\frac{1}{24}. \tag{42}$$

We assume that a stencil has been fixed. In practice, we will be using the second or fourth-order stencils above, but what follows holds for any stencil. Additional assumptions on stencil properties will be stated when they are required. With this, we define the differencing notation $\mathcal{D}^x_{m,n,p}[u]$ as

$$\mathcal{D}^x_{m,n,p}[u] = \frac{1}{\Delta x} \sum_\alpha w_\alpha u_{m+\alpha-\frac{1}{2},n,p}. \tag{43}$$

The other combinations (for y and z) as well as the 2D versions are defined similarly. The m, n, p are indices, which indicate where in the grid the quantity naturally lives. For example the quantity $\mathcal{D}^y_{i-\frac{1}{2},j-\frac{1}{2},k}[u]$ naturally lives at the grid location $x_{i-\frac{1}{2},j-\frac{1}{2},k}$. With this notation, we extend the notion of chains from Schroeder et al. [39] to arbitrary stencils in the natural way

$$\frac{\mathrm{d}}{\mathrm{d}x} R^{n+1}(x) = \sum_\alpha w_\alpha R^n\left(x + \alpha - \frac{1}{2}\right). \tag{44}$$

In particular, for the 4th order stencil, the chain construction becomes

$$\frac{\mathrm{d}}{\mathrm{d}x} R^{n+1}(x) = \frac{1}{24} R^n\left(x - \frac{3}{2}\right) - \frac{27}{24} R^n\left(x - \frac{1}{2}\right) + \frac{27}{24} R^n\left(x + \frac{1}{2}\right) - \frac{1}{24} R^n\left(x + \frac{3}{2}\right). \tag{45}$$

4.2. Interpolation notation

The general construction of divergence-free fields follows the same pattern as was derived for the second order stencil, except that the chains are extended using general stencils according to (44). To simplify the construction, we define the notation $\mathcal{I}^{abc}[\cdot]$ to concisely describe the types of interpolation required by the schemes.

$$\mathcal{I}^{abc}_{ABC}[f(i,j,k)] = \sum_{i,j,k} f(i,j,k) A^{m+a}\left(\frac{x - x_{i+a/2}}{\Delta x}\right) B^{n+b}\left(\frac{y - y_{j+b/2}}{\Delta y}\right) C^{p+c}\left(\frac{z - z_{k+c/2}}{\Delta z}\right). \tag{46}$$

The superscripts *abc* indicate which spline(s) have raised degrees. The subscripts indicate which splines are used in the construction. As examples,

$$\mathcal{I}^{000}_{ABC}[d_{i,j,k}] = \sum_{i,j,k} d_{i,j,k} A^m\left(\frac{x - x_i}{\Delta x}\right) B^n\left(\frac{y - y_j}{\Delta y}\right) C^p\left(\frac{z - z_k}{\Delta z}\right) \tag{47}$$

$$\mathcal{I}^{100}_{ABC}[u_{i+\frac{1}{2},j,k}] = \sum_{i,j,k} u_{i+\frac{1}{2},j,k} A^{m+1}\left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x}\right) B^n\left(\frac{y - y_j}{\Delta y}\right) C^p\left(\frac{z - z_k}{\Delta z}\right) \tag{48}$$

$$\mathcal{I}^{011}_{PQR}[u_{i-\frac{1}{2},j,k}] = \sum_{i,j,k} u_{i-\frac{1}{2},j,k} P^m\left(\frac{x - x_i}{\Delta x}\right) Q^{n+1}\left(\frac{y - y_{j+\frac{1}{2}}}{\Delta y}\right) R^{p+1}\left(\frac{z - z_{k+\frac{1}{2}}}{\Delta z}\right) \tag{49}$$

The first is interpolating the data d located at cell centers (i, j, k) using the piecewise polynomials A^m, B^n, and C^p. The second interpolates the data u located at x faces $i + \frac{1}{2}, j, k$ using the polynomials A^{m+1}, B^n, and C^p. And finally, the last is interpolating data living at the x faces $i - \frac{1}{2}, j, k$ using the polynomials P^m, Q^{n+1}, R^{p+1}. When we are considering the 2D case, the notation is similar

$$\mathcal{I}^{ab}_{AB}[f(i,j)] = \sum_{i,j} f(i,j) A^{m+a}\left(\frac{x - x_{i+a/2}}{\Delta x}\right) B^{n+b}\left(\frac{y - y_{j+b/2}}{\Delta y}\right). \tag{50}$$

4.3. General construction of divergence-free fields

The differencing stencils naturally induce a corresponding notion of divergence

$$d_{i,j} = \mathcal{D}^x_{i,j}[u] + \mathcal{D}^y_{i,j}[v] \qquad \text{in 2D} \tag{51}$$

$$d_{i,j,k} = \mathcal{D}^x_{i,j,k}[u] + \mathcal{D}^y_{i,j,k}[v] + \mathcal{D}^z_{i,j,k}[v] \qquad \text{in 3D.} \tag{52}$$

Discretely divergence-free simply means $d_{i,j} = 0$ or $d_{i,j,k} = 0$. Note that the second order derivative stencil produces a second order accurate divergence finite difference stencil, and the fourth-order derivative stencil produces a fourth-order accurate divergence finite difference stencil.

In 2D, the basic form of a divergence-free field is constructed from two splines A^m and B^n.

$$\hat{u}(x,y) = \mathcal{I}^{10}_{AB}[u_{i+\frac{1}{2},j}] \qquad \hat{v}(x,y) = \mathcal{I}^{01}_{AB}[v_{i,j+\frac{1}{2}}] \tag{53}$$

In 3D, the basic form is constructed from three splines A^m, B^n, and C^p.

$$\hat{u}(x,y,z) = \mathcal{I}^{100}_{ABC}[u_{i+\frac{1}{2},j,k}] \qquad \hat{v}(x,y,z) = \mathcal{I}^{010}_{ABC}[v_{i,j+\frac{1}{2},k}] \qquad \hat{w}(x,y,z) = \mathcal{I}^{001}_{ABC}[w_{i,j,k+\frac{1}{2}}] \tag{54}$$

Note that the 2D and 3D functions defined above are analytically divergence free for any choice of splines A^m, B^n, and C^p if the data is discretely divergence free in the sense of (52), as we show below. As in Schroeder et al. [39], the construction above is generalized by noting that linear combinations of divergence-free fields are also divergence free.

Symmetrizing. Note that if $A^m = B^n = C^p$, then the schemes above will be symmetrical (in the sense of permuting axes). If $A^m \neq B^n = C^p$, then a symmetric stencil can be obtained (if desired) by adding the permutations

$$\hat{u}(x,y) = \mathcal{I}^{10}_{AB}[u_{i+\frac{1}{2},j}] + \mathcal{I}^{10}_{BA}[u_{i+\frac{1}{2},j}] \tag{55}$$

$$\hat{u}(x,y,z) = \mathcal{I}^{100}_{ABB}[u_{i+\frac{1}{2},j,k}] + \mathcal{I}^{100}_{BAB}[u_{i+\frac{1}{2},j,k}] + \mathcal{I}^{100}_{BBA}[u_{i+\frac{1}{2},j,k}] \tag{56}$$

Similarly, if all of the splines are different in 3D, then a symmetric stencil is obtained as the sum of all six permutations of the splines. None of the other properties of our schemes rely on symmetry, so we do not assume it during the general construction or the proofs for simplicity. All of the specific schemes that we propose, however, are symmetrical.

4.3.1. Proof of analytic divergence-free for general stencils

We begin by noting the following identity for the derivative of our MAC-based interpolation.

$$\frac{\partial}{\partial x}\mathcal{I}^{100}_{ABC}[u_{i+\frac{1}{2},j,k}] = \sum_{i,j,k} u_{i+\frac{1}{2},j,k} \frac{\partial}{\partial x}\left[A^{m+1}\left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x}\right)\right] B^n\left(\frac{y-y_j}{\Delta y}\right) C^p\left(\frac{z-z_k}{\Delta z}\right) \tag{57}$$

$$= \sum_{i,j,k} u_{i+\frac{1}{2},j,k}\left[\frac{1}{\Delta x}\sum_\alpha w_\alpha A^m\left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x} + \alpha - \frac{1}{2}\right)\right] B^n\left(\frac{y-y_j}{\Delta y}\right) C^p\left(\frac{z-z_k}{\Delta z}\right) \tag{58}$$

$$= \sum_{i,j,k} u_{i+\frac{1}{2},j,k}\left[\frac{1}{\Delta x}\sum_\alpha w_\alpha A^m\left(\frac{x - x_{i+\frac{1}{2}-(\alpha-\frac{1}{2})}}{\Delta x}\right)\right] B^n\left(\frac{y-y_j}{\Delta y}\right) C^p\left(\frac{z-z_k}{\Delta z}\right) \tag{59}$$

$$= \sum_{i,j,k}\left[\frac{1}{\Delta x}\sum_\alpha w_\alpha u_{i+\alpha-\frac{1}{2},j,k}\right] A^m\left(\frac{x-x_i}{\Delta x}\right) B^n\left(\frac{y-y_j}{\Delta y}\right) C^p\left(\frac{z-z_k}{\Delta z}\right) \tag{60}$$

$$= \sum_{i,j,k} \mathcal{D}^x_{i,j,k}[u] A^m\left(\frac{x-x_i}{\Delta x}\right) B^n\left(\frac{y-y_j}{\Delta y}\right) C^p\left(\frac{z-z_k}{\Delta z}\right) \tag{61}$$

$$= \mathcal{I}^{000}_{ABC}[\mathcal{D}^x_{i,j,k}[u]] \tag{62}$$

Then, the divergence is

$$\nabla \cdot \langle \hat{u}, \hat{v}, \hat{w} \rangle = \frac{\partial}{\partial x}\mathcal{I}^{100}_{ABC}[u_{i+\frac{1}{2},j,k}] + \frac{\partial}{\partial y}\mathcal{I}^{010}_{ABC}[v_{i,j+\frac{1}{2},k}] + \frac{\partial}{\partial z}\mathcal{I}^{001}_{ABC}[w_{i,j,k+\frac{1}{2}}] \tag{63}$$

$$= \mathcal{I}^{000}_{ABC}[\mathcal{D}^x_{i,j,k}[u]] + \mathcal{I}^{000}_{ABC}[\mathcal{D}^y_{i,j,k}[v]] + \mathcal{I}^{000}_{ABC}[\mathcal{D}^z_{i,j,k}[w]] \tag{64}$$

$$= \mathcal{I}^{000}_{ABC}[\mathcal{D}^x_{i,j,k}[u] + \mathcal{D}^y_{i,j,k}[v] + \mathcal{D}^z_{i,j,k}[w]] \tag{65}$$

$$= \mathcal{I}^{000}_{ABC}[d_{i,j,k}] \tag{66}$$

so that $\nabla \cdot \langle \hat{u}, \hat{v}, \hat{w} \rangle = 0$ when $d_{i,j,k} = 0$. The proof in the case of 2D is similar.

4.4. General construction of curl-free fields

The differencing stencil also implies a notion of curl.

$$c^{xy}_{i-\frac{1}{2},j-\frac{1}{2}} = \mathcal{D}^{x}_{i-\frac{1}{2},j-\frac{1}{2}}[v] - \mathcal{D}^{y}_{i-\frac{1}{2},j-\frac{1}{2}}[u] \qquad \text{in 2D} \tag{67}$$

$$c^{xy}_{i-\frac{1}{2},j-\frac{1}{2},k} = \mathcal{D}^{x}_{i-\frac{1}{2},j-\frac{1}{2},k}[v] - \mathcal{D}^{y}_{i-\frac{1}{2},j-\frac{1}{2},k}[u] \qquad \text{in 3D} \tag{68}$$

In 2D, discretely curl-free means $c^{xy}_{i-\frac{1}{2},j-\frac{1}{2}} = 0$. In 3D, it implies $c^{xy}_{i-\frac{1}{2},j-\frac{1}{2},k} = c^{xz}_{i-\frac{1}{2},j,k-\frac{1}{2}} = c^{yz}_{i,j-\frac{1}{2},k-\frac{1}{2}} = 0$.

The curl-free construction is similar but in a sense opposite to the divergence-free construction. The 2D and 3D definitions are

$$\hat{u}(x,y) = \mathcal{I}^{01}_{AB}[u_{i-\frac{1}{2},j}] \qquad \hat{v}(x,y) = \mathcal{I}^{10}_{AB}[v_{i,j-\frac{1}{2}}] \tag{69}$$

$$\hat{u}(x,y,z) = \mathcal{I}^{011}_{ABC}[u_{i-\frac{1}{2},j,k}] \qquad \hat{v}(x,y,z) = \mathcal{I}^{101}_{ABC}[v_{i,j-\frac{1}{2},k}] \qquad \hat{w}(x,y,z) = \mathcal{I}^{110}_{ABC}[w_{i,j,k-\frac{1}{2}}]. \tag{70}$$

The 2D and 3D functions defined above are analytically curl free for any choice of splines A^m, B^n, and C^p if the data is discretely curl free in the sense of (68). The construction above is generalized by noting that linear combinations of divergence-free fields are also divergence free. As in the divergence-free case, the construction can be symmetrized by adding permutations of the splines.

4.4.1. Proof of analytic curl-free for general stencils

The curl is (in 3D)

$$\nabla \times \begin{pmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial \hat{v}}{\partial x} - \dfrac{\partial \hat{u}}{\partial y} \\ \dfrac{\partial \hat{w}}{\partial y} - \dfrac{\partial \hat{v}}{\partial z} \\ \dfrac{\partial \hat{u}}{\partial z} - \dfrac{\partial \hat{w}}{\partial x} \end{pmatrix}, \tag{71}$$

The discrete curl-free property begins with the identity

$$\frac{\partial}{\partial y}\mathcal{I}^{011}_{ABC}[u_{i-\frac{1}{2},j,k}] = \mathcal{I}^{001}_{ABC}[\mathcal{D}^{y}_{i-\frac{1}{2},j-\frac{1}{2},k}[u]]. \tag{72}$$

The derivation of this identity is nearly identical to that of (60) and is omitted. The first component of the curl is

$$\frac{\partial \hat{v}}{\partial x} - \frac{\partial \hat{u}}{\partial y} = \frac{\partial}{\partial x}\mathcal{I}^{101}_{ABC}[v_{i,j-\frac{1}{2},k}] - \frac{\partial}{\partial y}\mathcal{I}^{011}_{ABC}[u_{i-\frac{1}{2},j,k}] \tag{73}$$

$$= \mathcal{I}^{001}_{ABC}[\mathcal{D}^{x}_{i-\frac{1}{2},j-\frac{1}{2},k}[v]] - \mathcal{I}^{001}_{ABC}[\mathcal{D}^{y}_{i-\frac{1}{2},j-\frac{1}{2},k}[u]] \tag{74}$$

$$= \mathcal{I}^{001}_{ABC}[\mathcal{D}^{x}_{i-\frac{1}{2},j-\frac{1}{2},k}[v] - \mathcal{D}^{y}_{i-\frac{1}{2},j-\frac{1}{2},k}[u]] \tag{75}$$

$$= \mathcal{I}^{001}_{ABC}[c^{xy}_{i-\frac{1}{2},j-\frac{1}{2},k}]. \tag{76}$$

If the vector field is discretely divergence free, then $c^{xy}_{i-\frac{1}{2},j-\frac{1}{2},k} = 0$ so that the first component of curl is analytically zero. The other components are similar.

4.5. Explicit 4th Order Schemes

Our primary interest in considering more general differencing stencils is to construct an interpolation scheme that is divergence-free (or curl-free) and also more accurate than second order. As noted in [39], this does not appear to be possible when using second order central differences. Indeed, a Poisson projection using the second order central differencing stencil would only be second order accurate anyway. To improve accuracy, we need a more accurate stencil. This suggests the use of the standard 4th order differencing stencil, which implies the chain construction (45).

Practically speaking, this allows us to project vectors fields with fourth-order accuracy. As we shall see, this also allows us to perform divergence-free or curl-free interpolation with fourth-order accuracy. It is important to note that while the general construction always yields interpolation schemes that are divergence-free or curl-free, the use of a higher order stencil does not imply that the scheme will actually be higher order accurate. The interpolation schemes

$$u^{div,4th}(x,y) = \frac{1}{6}\sum_{ij} u_{i,j}[\hat{B}^2_{i-1}(x)I^3_j(y) + \hat{I}^4_i(x)B^1_{j-1}(y)] \tag{77}$$

$$u^{div,4th}(x,y,z) = \frac{1}{6}\sum_{ijk} u_{i,j,k}[\hat{B}^2_{i-1}(x)B^1_{j-1}(y)J^3_k(z) + \hat{B}^2_{i-1}(x)J^3_j(y)B^1_{k-1}(z) + \hat{J}^4_i(x)B^1_{j-1}(y)B^1_{k-1}(z)] \tag{78}$$

$$u^{curl,4th}(x,y) = \frac{1}{6}\sum_{ij} u_{i,j}[B^1_{i-1}(x)\hat{I}^4_j(y) + I^3_i(x)\hat{B}^2_{j-1}(y)] \tag{79}$$

$$u^{curl,4th}(x,y,z) = \frac{1}{6}\sum_{ijk} u_{i,j,k}[B^1_{i-1}(x)\hat{B}^2_{j-1}(y)\hat{J}^4_k(z) + B^1_{i-1}(x)\hat{J}^4_j(y)\hat{B}^2_{k-1}(z) + J^3_i(x)\hat{B}^2_{j-1}(y)\hat{B}^2_{k-1}(z)] \tag{80}$$

are all fourth-order accurate. We have used hats ($\hat{B}^2$, $\hat{I}^4$) to indicate that the polynomials have been formed from the parent spline (B^1, I^3) using the fourth-order chain. In particular, this distinguishes $\hat{B}^2$ (formed from B^1 using (45)) from B^2 (formed from B^1 using (2)). Note that "missing" polynomials are zero. Thus, for example, $B^1_{-1}(x) = B^1_2(x) = 0$ occur in the schemes above.

Note that the 2D and 3D schemes are subtly different, in that the 2D schemes use the I chain while the 3D schemes use the J chain, which are defined as

$$I^3_0 = x(1-x)(x-2) \qquad I^3_1 = -3(1-x)(x^2-x-1) \qquad I^3_2 = -3x(x^2-x-1) \qquad I^3_3 = -x(1-x)(x+1) \tag{81}$$

$$J^3_0 = x(1-x)(x-2) \qquad J^3_1 = -(1-x)(3x^2-3x-2) \qquad J^3_2 = -x(3x^2-3x-2) \qquad J^3_3 = -x(1-x)(x+1). \tag{82}$$

The remaining splines (with hats) are obtained from existing splines using (45). For completeness and ease of implementation, these splines are

$$\hat{B}^2_0 = -\frac{1}{48}(1-x)^2 \qquad \hat{B}^2_1 = \frac{25}{48} - \frac{9}{8}x + \frac{7}{12}x^2 \tag{83}$$

$$\hat{B}^2_2 = \frac{25}{48} + \frac{9}{8}x - \frac{9}{8}x^2 \qquad \hat{B}^2_3 = -\frac{1}{48} - \frac{1}{24}x + \frac{7}{12}x^2 \qquad \hat{B}^2_4 = -\frac{1}{48}x^2 \tag{84}$$

$$\hat{I}^4_0 = -\frac{1}{96}(x^2-2x-1)(1-x)^2 \qquad \hat{I}^4_1 = -\frac{11}{32} + \frac{1}{8}x + \frac{9}{8}x^2 - \frac{29}{24}x^3 + \frac{5}{16}x^4 \tag{85}$$

$$\hat{I}^4_2 = \frac{11}{6} - \frac{27}{8}x - \frac{17}{16}x^2 + \frac{41}{12}x^3 - \frac{37}{32}x^4 \qquad \hat{I}^4_3 = \frac{11}{6} + \frac{27}{8}x - \frac{5}{3}x^2 - \frac{41}{12}x^3 + \frac{41}{24}x^4 \tag{86}$$

$$\hat{I}^4_4 = -\frac{11}{32} - \frac{1}{8}x + \frac{9}{4}x^2 + \frac{29}{24}x^3 - \frac{37}{32}x^4 \qquad \hat{I}^4_5 = \frac{1}{96} - \frac{5}{8}x^2 - \frac{1}{24}x^3 + \frac{5}{16}x^4 \qquad \hat{I}^4_6 = -\frac{1}{96}x^2(x^2-2) \tag{87}$$

$$\hat{J}^4_0 = -\frac{1}{96}(x^2-2x-1)(1-x)^2 \qquad \hat{J}^4_1 = -\frac{31}{96} + \frac{1}{12}x + \frac{55}{48}x^2 - \frac{29}{24}x^3 + \frac{5}{16}x^4 \tag{88}$$

$$\hat{J}^4_2 = \frac{21}{16} - \frac{9}{4}x - \frac{79}{48}x^2 + \frac{41}{12}x^3 - \frac{37}{32}x^4 \qquad \hat{J}^4_3 = \frac{21}{16} + \frac{9}{4}x - \frac{13}{24}x^2 - \frac{41}{12}x^3 + \frac{41}{24}x^4 \tag{89}$$

$$\hat{J}^4_4 = -\frac{31}{96} - \frac{1}{12}x + \frac{5}{3}x^2 + \frac{29}{24}x^3 - \frac{37}{32}x^4 \qquad \hat{J}^4_5 = \frac{1}{96} - \frac{29}{48}x^2 - \frac{1}{24}x^3 + \frac{5}{16}x^4 \qquad \hat{J}^4_6 = -\frac{1}{96}x^2(x^2-2) \tag{90}$$

The stencils for these schemes are shown in Figure 1.

Note on scheme construction. The specific stencils presented in this paper were constructed following the strategy in [39], where one begins with an arbitrary vector field with arbitrary polynomial splines of specified degree and support. Then, one uses constraints (continuity, analytic divergence-free or curl-free, symmetry, etc.) to solve for as many degrees of freedom as possible. If optimal degree and support are chosen, one is typically left with a small number of free parameters. We then use the general construction for divergence-free or curl-free fields to *factor* the stencil as a sum of tensor products, the form in which all of the schemes in this paper have been presented. We

choose the free parameters in order to use the fewest number of tensor products. Using the general construction for factorization has many advantages. It produces very simple factorizations in practice, making it possible to implement schemes that defied practical implementation without it. The general construction drastically reduces the difficulty of factoring the schemes, since only the simplest spline of a chain must be solved for; the other splines are obtained from the chain construction and therefore do not contribute additional unknowns to the problem. Finding the original scheme is a linear problem, but factoring it leads to a nonlinear system of polynomials; reducing the difficulty of the factoring problem is important.

Implementation notes. In our implementation, we implement the interpolation schemes using the tensor product form obtained from the general construction. For the simplest schemes, we hard-code the spline polynomial computations to enable compiler optimization. For the more complicated schemes (such as the fourth-order schemes) and also for rapid prototyping, we store the spline coefficients in a table. We use the chain construction to automatically generate the tables for the higher degree splines in a chain.

4.6. Relationship with Raviart-Thomas Finite Elements

There is a notable similarity between the degrees of the splines in the schemes presented in this section, and the Raviart-Thomas function spaces used for $H(div)/H(curl)$-conforming finite-elements on quadrilaterals [37, 36].

In particular, the Raviart-Thomas function space is $\mathcal{RT}_k = P_{k+1,k,k} \times P_{k,k+1,k} \times P_{k,k,k+1}$, and the Nédélec function space is $\mathcal{N}_k = P_{k,k+1,k+1} \times P_{k+1,k,k+1} \times P_{k+1,k+1,k}$, where $P_{l,m,n}$ denotes the set of polynomials with degrees l, m, n in x, y, z respectively (and analogously in 2D) [16]. The degrees on these polynomials are identical to the degrees on the splines in all of the interpolation schemes we have presented. Therefore, our polynomials can be seen as a subset of these function spaces.

However, where Raviart-Thomas/Nédélec finite elements use L^2 inner products with test functions (i.e. moments) as the degrees of freedom, using higher-order test functions for higher-order interpolation, our approach instead uses a larger stencil width, directly using the vector field values on the grid. While the interpolation operators for Raviart-Thomas elements are divergence/curl free because of the function spaces involved, and the surjectivity of the derivative operator between B-spline spaces [18, 26], while our interpolants are divergence/curl free as a result of the chain property of the splines that we use.

4.7. Practical implementation

It is worth taking a moment to make the process of going from the tensor product spline representation presented into an actual implementation. Consider the x component of the curl-free interpolation (36).

$$u^{curl,C0}(x,y) = \sum_{ij} u_{i,j} B_i^1(x) B_j^2(y). \tag{91}$$

This scheme uses two splines B^1 and B^2, which are

$$B_0^1 = 1 - x \qquad B_1^1 = x \qquad B_0^2 = \frac{1}{2}(1-x)^2 \qquad B_1^2 = \frac{1}{2} + x - x^2 \qquad B_2^2 = \frac{1}{2}x^2. \tag{92}$$

The shape of this stencil is shown in Figure 1 in the top middle illustration. In the figure, the x components we are using are stored on the grid at the locations labeled with red triangles (▷), and the region of the grid we are interpolating data it is shaded green. For the purpose of computing the interpolation, we let $0 \le x, y < 1$ over the shaded green square. That is, $x = y = 0$ at the lower left corner of the green square and $x = y = 1$ at the top right of the green square. In the case of this stencil, the green region corresponds to a cell of the MAC grid, though this is not always

the case. Letting (i, j) be the index of the green cell, the interpolated x component is

$$u^{curl,C0}(x,y) = B_0^1(x)B_0^2(y)u_{i-\frac{1}{2},j-1} + B_0^1(x)B_1^2(y)u_{i-\frac{1}{2},j} + B_0^1(x)B_2^2(y)u_{i-\frac{1}{2},j+1} \tag{93}$$

$$+ B_1^1(x)B_0^2(y)u_{i+\frac{1}{2},j-1} + B_1^1(x)B_1^2(y)u_{i+\frac{1}{2},j} + B_1^1(x)B_2^2(y)u_{i+\frac{1}{2},j+1} \tag{94}$$

$$= (1-x)\left(\frac{1}{2}(1-y)^2\right)u_{i-\frac{1}{2},j-1} + (1-x)\left(\frac{1}{2}+y-y^2\right)u_{i-\frac{1}{2},j} + (1-x)\left(\frac{1}{2}y^2\right)u_{i-\frac{1}{2},j+1} \tag{95}$$

$$+ x\left(\frac{1}{2}(1-y)^2\right)u_{i+\frac{1}{2},j-1} + x\left(\frac{1}{2}+y-y^2\right)u_{i+\frac{1}{2},j} + x\left(\frac{1}{2}y^2\right)u_{i+\frac{1}{2},j+1} \tag{96}$$

$$= \begin{pmatrix} 1-x \\ x \end{pmatrix}^T \begin{pmatrix} u_{i-\frac{1}{2},j-1} & u_{i-\frac{1}{2},j} & u_{i-\frac{1}{2},j+1} \\ u_{i+\frac{1}{2},j-1} & u_{i+\frac{1}{2},j} & u_{i+\frac{1}{2},j+1} \end{pmatrix} \begin{pmatrix} \frac{1}{2}(1-y)^2 \\ \frac{1}{2}+y-y^2 \\ \frac{1}{2}y^2 \end{pmatrix} \tag{97}$$

In practice, we implement our splines in the last form, where we collect the appropriate entries from the MAC grid and copy them to a matrix. Then we evaluate the splines to get two vectors. The interpolation is performed as a matrix-vector multiply and a dot product. The process is similar in 3D, except now the data is copied to a rank-3 tensor, three vectors are filled in by evaluating the appropriate set of splines (one for x, y, and z), and then the result is obtained by contracting the tensor with the vectors.

5. Numerical Tests

In the numerical tests that follow, we use the following vector fields in 2D.

$$\mathbf{u}_{2a} = \begin{pmatrix} \sin(370x+2)\sin(370y+4) \\ \cos(370x+2)\cos(370y+4) \end{pmatrix} \qquad \mathbf{u}_{2e} = \begin{pmatrix} \sin(370x+2)\cos(y+4) \\ \cos(370x+2)\sin(370y+4) \end{pmatrix} \tag{98}$$

$$\mathbf{u}_{2b} = \begin{pmatrix} \sin(x+2)\sin(y+4) \\ \cos(x+2)\cos(y+4) \end{pmatrix} \qquad \mathbf{u}_{2f} = \begin{pmatrix} \sin(x+2)\cos(y+4) \\ \cos(x+2)\sin(y+4) \end{pmatrix} \tag{99}$$

$$\mathbf{u}_{2c} = \begin{pmatrix} x^3 - 6xy^2 + y^3 \\ -3x^2y + 2y^3 \end{pmatrix} \qquad \mathbf{u}_{2g} = \begin{pmatrix} -3x^2y + y^3 \\ -x^3 + 3xy^2 \end{pmatrix} \tag{100}$$

$$\mathbf{u}_{2d} = \begin{pmatrix} \sin(x+2) + \sin(y+4) \\ \cos(x+2) + \cos(y+4) \end{pmatrix} \tag{101}$$

In 3D, we test using the vector fields:

$$\mathbf{u}_{3a} = \begin{pmatrix} \sin(370x+2)\sin(370y+4)\sin(370z+6) \\ \cos(370x+2)\cos(370y+4)\cos(370z+6) \\ \cos(370x+2)\sin(370y+4)(\cos(370z+6)+\sin(370z+6)) \end{pmatrix} \qquad \mathbf{u}_{3c} = \begin{pmatrix} y^3z^2 - 6xy^2z + x^3 \\ 3x^2y + 2xyz + z^2 \\ 3y^2z^2 - 6x^2z - xz^2 \end{pmatrix} \tag{102}$$

$$\mathbf{u}_{3b} = \begin{pmatrix} \sin(x+2)\sin(y+4)\sin(z+6) \\ \cos(x+2)\cos(y+4)\cos(z+6) \\ \cos(x+2)\sin(y+4)(\cos(z+6)+\sin(z+6)) \end{pmatrix} \qquad \mathbf{u}_{3d} = \begin{pmatrix} \sin(x+2)+\sin(y+4) \\ \cos(y+2)+\cos(z+4) \\ \cos(z+2)+\cos(x+4) \end{pmatrix} \tag{103}$$

$$\mathbf{u}_{3e} = \begin{pmatrix} \sin(370x+2)\cos(370y+4)\cos(370z+6) \\ \cos(370x+2)\sin(370y+4)\cos(370z+6) \\ \cos(370x+2)\cos(370y+4)\sin(370z+6) \end{pmatrix} \qquad \mathbf{u}_{3g} = \begin{pmatrix} -3x^2y + y^3 \\ -x^3 + 3xy^2 \\ 0 \end{pmatrix} \tag{104}$$

$$\mathbf{u}_{3f} = \begin{pmatrix} \sin(x+2)\cos(y+4)\cos(z+6) \\ \cos(x+2)\sin(y+4)\cos(z+6) \\ \cos(x+2)\cos(y+4)\sin(z+6) \end{pmatrix}. \tag{105}$$

These vector fields were chosen such that the 2D and 3D versions have similar properties. Fields (a) through (d) are used for the divergence tests, while (d) through (g) are used for the curl tests. The (a) and (e) fields are discretely divergence and curl free, respectively, but the functions are under-resolved on the grid (so function as pseudorandom vector fields). The (b) and (f) fields are also discretely divergence/curl free, but are resolved on the grid. Fields (c) and (g) are analytically but not discretely divergence/curl free. Finally (d) does not have any special properties with respect to divergence or curl.

Scheme	2D	3D
Linear	0.0610	0.163
Cubic	0.0237	0.0657
Cubic spline	0.0223	0.0623
$u^{div,C0}$	0.116	0.236
u^{flux}	5.55×10^{-16}	7.77×10^{-16}
Balsara	5.55×10^{-16}	1.11×10^{-15}

Table 1: Maximum difference between face flux computed by sixth order Gaussian quadrature and velocity at that face (times area). Using velocity field $\mathbf{u}_{2a}$ and $\mathbf{u}_{3a}$

Scheme	$\mathbf{u}_{2a}$	$\mathbf{u}_{2b}$	$\mathbf{u}_{2c}$	$\mathbf{u}_{2d}$	$\mathbf{u}_{2\mathcal{D}^2}$	$\mathbf{u}_{2\mathcal{D}^4}$
Linear	1.98×10^{1}	2.89×10^{-2}	4.54×10^{-1}	5.56×10^{-1}	6.74	6.35
Cubic	9.92	1.95×10^{-5}	5.00×10^{-10}	5.82×10^{-1}	4.40	3.98
Cubic spline	1.18×10^{1}	9.25×10^{-4}	1.11×10^{-2}	5.83×10^{-1}	3.22	3.19
Balsara	1.53×10^{-9}	1.27×10^{-10}	2.93×10^{-3}	5.56×10^{-1}	5.83×10^{-10}	5.98×10^{-1}
$u^{div,C0}$	1.05×10^{-9}	1.53×10^{-10}	2.93×10^{-3}	5.82×10^{-1}	5.02×10^{-10}	4.39×10^{-1}
$u^{div,4th}$	1.34×10^{-9}	1.80×10^{-10}	5.55×10^{-10}	5.82×10^{-1}	5.85×10^{-1}	8.84×10^{-10}
u^{flux}	9.65×10^{-9}	2.08×10^{-10}	2.93×10^{-3}	5.83×10^{-1}	3.82×10^{-9}	9.11×10^{-1}

Scheme	$\mathbf{u}_{3a}$	$\mathbf{u}_{3b}$	$\mathbf{u}_{3c}$	$\mathbf{u}_{3d}$	$\mathbf{u}_{3\mathcal{D}^2}$	$\mathbf{u}_{3\mathcal{D}^4}$
Linear	2.48×10^{1}	4.69×10^{-2}	2.16×10^{-1}	2.73	8.96	8.71
Cubic	1.27×10^{1}	3.12×10^{-5}	1.19×10^{-9}	2.74	7.03	6.73
Cubic spline	1.06×10^{1}	9.17×10^{-4}	3.90×10^{-3}	2.74	4.48	3.97
Balsara	1.92×10^{-9}	1.94×10^{-10}	9.77×10^{-4}	2.73	7.70×10^{-10}	1.11
$u^{div,C0}$	9.51×10^{-10}	2.43×10^{-10}	9.77×10^{-4}	2.74	4.30×10^{-10}	6.69×10^{-1}
$u^{div,4th}$	1.30×10^{-9}	2.57×10^{-10}	1.11×10^{-9}	2.74	9.13×10^{-1}	9.09×10^{-10}
u^{flux}	1.67×10^{-8}	5.27×10^{-10}	9.77×10^{-4}	2.74	6.14×10^{-9}	1.54

Table 2: Maximum divergence of interpolation at 10^6 sample points for 6 velocity fields in 2D (top) and 3D (bottom). $\mathbf{u}_{*_{\mathcal{D}^2}}$ is random data projected to be divergence-free under the second order stencil, while $\mathbf{u}_{*_{\mathcal{D}^4}}$ is random data projected to be divergence-free under the 4^{th} order stencil.

5.1. Flux consistency

In this test, we verify that the flux consistent scheme u^{flux} analytically preserves the flux across faces. For this we use the $\mathbf{u}_{2a}$ (for 2D) and $\mathbf{u}_{3a}$ (for 3D) vector fields over a 10^3 grid over the domain $[0, 1]^3$. We compare with a variety of schemes: linear, cubic, cubic spline, and $u^{div,C0}$. On the grid, the flux is computed as value stored at the face center multiplied by the face area. Analytically, the flux of the interpolated vector field is calculated by integrating over the face using sixth order Gaussian quadrature, which is high enough order to compute the integrals exactly. The results are shown in Figure 1. As expected, the flux error for the flux-consistent scheme is at roundoff error levels, while the flux differences for the other schemes are on the order of truncation errors. We note that the second order scheme from Balsara [2] is also flux-consistent even when used for interpolation, which fits nicely with the flux-based nature of their numerical method for magnetohydrodynamics. Reconstruction schemes, including higher order variants, such as [4], are typically constructed to be flux-consistent when used for reconstruction.

5.2. Numerical tests for divergence-free interpolation

We demonstrate numerically that the divergence-free interpolation schemes produce analytically divergence-free vector fields. We perform tests over a 16^3 grid over the domain $[0, 1]^3$ using the vector fields $\mathbf{u}_{2a}$, $\mathbf{u}_{2b}$, $\mathbf{u}_{2c}$, $\mathbf{u}_{2d}$, $\mathbf{u}_{\mathcal{D}^2}$, and $\mathbf{u}_{\mathcal{D}^4}$ (for 2D) and $\mathbf{u}_{3a}$, $\mathbf{u}_{3b}$, $\mathbf{u}_{3c}$, $\mathbf{u}_{3d}$, $\mathbf{u}_{\mathcal{D}^2}$, and $\mathbf{u}_{\mathcal{D}^4}$ (for 3D). All of these except $\mathbf{u}_{2d}$ and $\mathbf{u}_{3d}$ are analytically divergence free. We perform this test with four divergence-free schemes (~~second order Balsara~~the second order scheme from Balsara [2], $u^{div,C0}$, $u^{div,4th}$, and u^{flux}) in both 2D and 3D. For comparison, we include three non-divergence-free schemes: linear, cubic, and cubic spline. The divergence of the interpolation schemes is calculated numerically using finite differences at a large number of sample points; the maximum absolute value of divergence is reported. Because

of the finite differences used for the test, analytically divergence-free vector fields will produce divergence errors on the order of $10^{-9} - 10^{-10}$.

The results are essentially the same in 2D (Table 2, top) and 3D (Table 2, bottom), so we only describe the 2D results.

- $\mathbf{u}_{2a}$: This vector field is discretely divergence-free under both stencils but under-resolved on the grid. As a result, none of the schemes accurately approximate it, and the non-divergence-free schemes all produce large divergence errors. The discretely divergence-free schemes produce errors consistent with an analytically divergence-free vector field.

- $\mathbf{u}_{2b}$: This vector field is discretely divergence-free under both stencils and well-resolved on the grid. Since all of the schemes accurately approximate it, the non-divergence-free schemes produce truncation-error-level divergence errors. The discretely divergence-free schemes produce errors consistent with an analytically divergence-free vector field.

- $\mathbf{u}_{2c}$: This vector field is divergence-free but not discretely so (under either stencil). As a result, the low order schemes (including the divergence-free ones) produce divergence errors on the order of truncation errors. The two higher order schemes (cubic interpolation and $u^{div,4th}$) exactly recover the cubic polynomial vector field and as a result produce divergence errors consistent with an analytically divergence-free vector field.

- $\mathbf{u}_{2d}$: This vector field is not divergence-free, and all of the schemes compute a divergence consistent with the actual divergence of the vector field.

- $\mathbf{u}_{\mathcal{D}^2}$: This vector field was constructed from random data that has been projected to be discretely divergence free under the second order stencil. This data effectively corresponds to an under-resolved high-frequency vector field that none of the schemes can approximate well (much like $\mathbf{u}_{2a}$). Unlike $\mathbf{u}_{2a}$, this data is divergence free under the second order stencil but not the fourth-order stencil. We see that the three non-divergence-free schemes produce large divergences, while the three schemes that are divergence-free under the second order stencil (Balsara [2], $u^{div,C0}$, and u^{flux}) are analytically divergence-free. It is interesting to note that while $u^{div,4th}$ does not produce an analytically divergence-free field (since the data is not discretely divergence-free under the correct stencil), the divergence errors are an order of magnitude smaller than the non-divergence-free schemes.

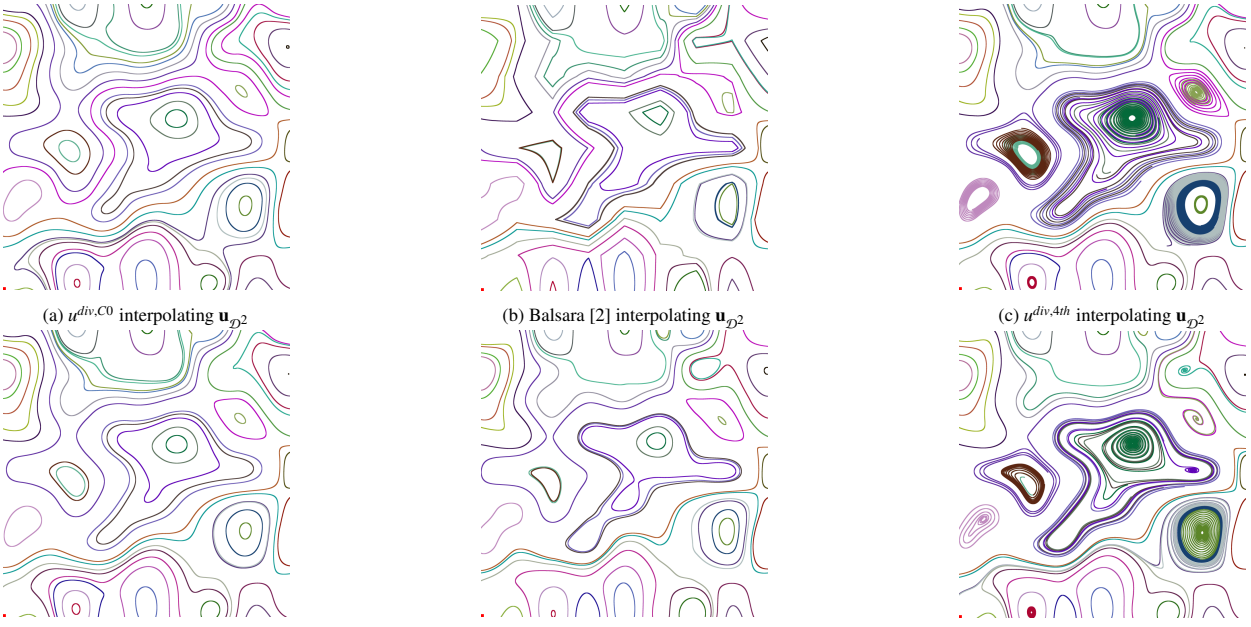
- $\mathbf{u}_{\mathcal{D}^4}$: This vector field is random data projected under the fourth-order stencil, and the results are analogous to the previous case. $u^{div,4th}$ is analytically divergence-free, the other non-divergence-free schemes have large errors, and the other divergence-free schemes have somewhat smaller errors.

5.3. Streamlines of reconstructed vector fields

Divergence-free vector fields have the property that particles traveling along them move along closed curves. More generally, particle trajectories through vector fields with negative divergence will spiral inwards, and particle trajectories through vector fields with positive divergence will spiral outwards. In methods where particles are traced through grid-based vector fields, this can lead to gaps or clumps in particle coverage.

Here, we demonstrate a property of divergence-free interpolation schemes such as the ones we propose. Using a divergence-free interpolation scheme and higher-order integration, approximately closed-loop streamlines in an incompressible flow can be reproduced. In contrast, non-divergence-free interpolation schemes result in streamlines that incorrectly spiral in/out from stagnation points.

Figure 3 illustrates the streamlines resulting from six different interpolation schemes. We first generate a random periodic vector field on a coarse 8^2 grid over the domain $[0, 1]^2$ and project it to be discretely divergence free using either the second-order stencil ($\mathbf{u}_{\mathcal{D}^2}$) or fourth-order stencil ($\mathbf{u}_{\mathcal{D}^4}$) described in Sec. 4.5. We seed particles in random locations and integrate their paths along the streamlines of the flow using 3^{rd}-order Runge-Kutta and the vector interpolation scheme indicated. Note that the divergence-free interpolation schemes used in Figures 3a, 3b, 3d, and 3e produce approximately closed-loop streamlines, while the schemes that do not produce analytically divergence-free fields used in Figure 3c and 3f produce streamlines that spiral in/out of stagnation points. In particular, streamlines produced by the 4^{th} order divergence-free scheme $u^{div,4th}$ forms are closed when interpolating $\mathbf{u}_{\mathcal{D}^4}$, but not when interpolating $\mathbf{u}_{\mathcal{D}^2}$, because interpolation is only analytically divergence free if the interpolated data is discretely divergence free under the 4^{th} order difference stencil $\mathcal{D}^4$. A single streamline of the same flow is depicted in Figures 4a-4f for



(a) $u^{div,C0}$ interpolating $\mathbf{u}_{\mathcal{D}^2}$ (b) Balsara [2] interpolating $\mathbf{u}_{\mathcal{D}^2}$ (c) $u^{div,4th}$ interpolating $\mathbf{u}_{\mathcal{D}^2}$

(d) $u^{div,4th}$ interpolating $\mathbf{u}_{\mathcal{D}^4}$ (e) u^{flux} interpolating $\mathbf{u}_{\mathcal{D}^2}$ (f) Cubic spline interpolating $\mathbf{u}_{\mathcal{D}^2}$

Fig. 3: For a given incompressible flow field, streamlines of the flow are traced through third-order Runge-Kutta integration of particle trajectories using different spatial interpolation schemes. The spatial interpolation scheme used is indicated in each subfigure caption. Analytically divergence-free interpolants (3a,3b,3d,3e) result in approximately closed streamlines, while non-divergence-free schemes (3c,3f) exhibit streamlines that spiral in/out of stagnation points.

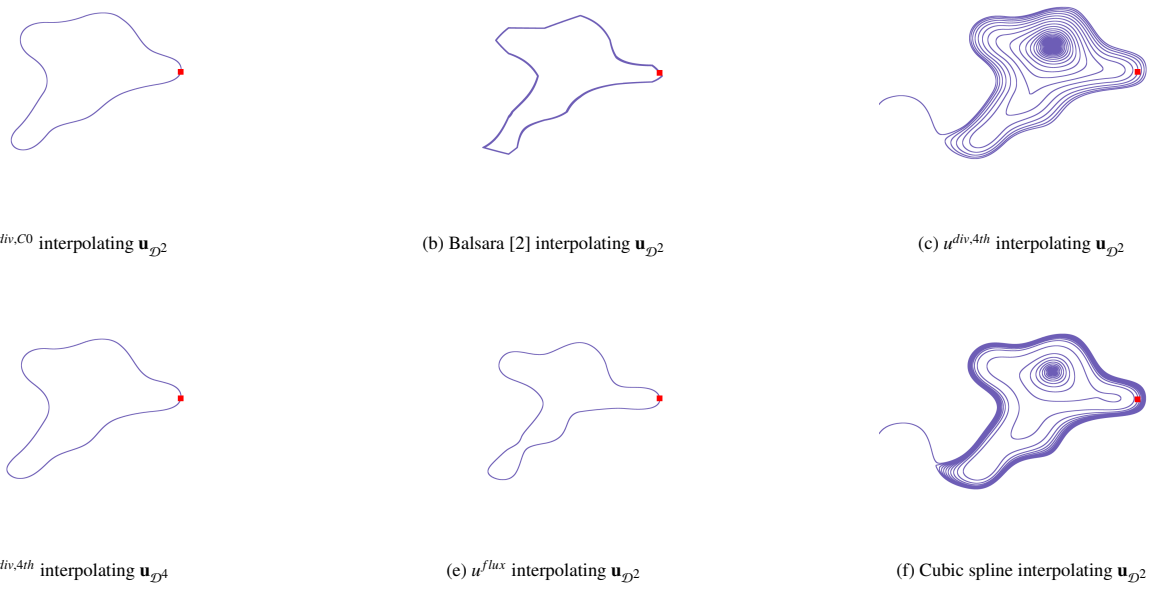
Scheme	$\mathbf{u}_{2e}$	$\mathbf{u}_{2f}$	$\mathbf{u}_{2g}$	$\mathbf{u}_{2d}$	$\mathbf{u}_{\mathcal{D}^2}$	$\mathbf{u}_{\mathcal{D}^4}$
Linear	2.25×10^{1}	3.11×10^{-2}	3.56×10^{-1}	1.19	8.11	7.86
Cubic	1.08×10^{1}	2.03×10^{-5}	6.66×10^{-10}	1.19	5.14	4.92
Cubic spline	1.20×10^{1}	8.81×10^{-4}	7.81×10^{-3}	1.19	4.25	3.73
$u^{curl,C0}$	1.55×10^{-9}	2.22×10^{-10}	1.95×10^{-3}	1.19	4.96×10^{-10}	6.56×10^{-1}
$u^{curl,C1}$	2.09×10^{-9}	2.36×10^{-10}	1.95×10^{-3}	1.19	1.12×10^{-9}	2.99×10^{-1}
$u^{curl,4th}$	1.74×10^{-9}	2.50×10^{-10}	5.55×10^{-10}	1.19	7.54×10^{-1}	8.53×10^{-10}

Table 3: Maximum curl of interpolation at 10^6 sample points for 4 velocity fields in 2D.

clarity. The sharp corners in the curves produced by the ~~Balsara~~ schemes from Balsara [2, 4] are caused by the discontinuities in the scheme and are especially visible because of the low resolution. These discontinuities should be much less significant on data that is better resolved on the grid, as would be the case in higher resolution simulations. We would also expect these discontinuities to become less significant in the 4th and 5th order versions presented in Balsara et al. [12, 9], though we do not have implementations of these schemes.

5.4. Numerical tests for curl-free interpolation

In this test, we demonstrate numerically that the curl-free schemes are analytically curl free. We perform these tests over a 16^3 grid over the domain $[0, 1]^3$ using the vector fields $\mathbf{u}_{2e}$, $\mathbf{u}_{2f}$, $\mathbf{u}_{2g}$, $\mathbf{u}_{2d}$, $\mathbf{u}_{\mathcal{D}^2}$, and $\mathbf{u}_{\mathcal{D}^4}$ (for 2D) and $\mathbf{u}_{3e}$, $\mathbf{u}_{3f}$, $\mathbf{u}_{3g}$, $\mathbf{u}_{3d}$, $\mathbf{u}_{\mathcal{D}^2}$, and $\mathbf{u}_{\mathcal{D}^4}$ (for 3D). All of these except $\mathbf{u}_{2d}$ and $\mathbf{u}_{3d}$ are analytically curl-free. We perform this test with all three of the proposed curl-free schemes ($u^{curl,C0}$, $u^{curl,C1}$, and $u^{curl,4th}$) in both 2D and 3D. For comparison, we include three non-curl-free schemes: linear, cubic, and cubic spline. The curl of the interpolation schemes is calculated numerically using finite differences at 10^6 random sample points; the maximum magnitude of curl is reported. Because of the finite differences used for the test, analytically curl-free vector fields will produce curl errors on the order of $10^{-9} - 10^{-10}$. The overall analysis and observations for the curl-free tests closely follow the analysis and observations for the divergence-free tests, so we do not repeat them. The results are shown in Table 2.



(a) $u^{div,C0}$ interpolating $\mathbf{u}_{\mathcal{D}^2}$ (b) Balsara [2] interpolating $\mathbf{u}_{\mathcal{D}^2}$ (c) $u^{div,4th}$ interpolating $\mathbf{u}_{\mathcal{D}^2}$

(d) $u^{div,4th}$ interpolating $\mathbf{u}_{\mathcal{D}^4}$ (e) u^{flux} interpolating $\mathbf{u}_{\mathcal{D}^2}$ (f) Cubic spline interpolating $\mathbf{u}_{\mathcal{D}^2}$

Fig. 4: A streamline of the flow is traced through third order Runge-Kutta integration of a single particle trajectory using different spatial interpolation schemes. Divergence-free interpolation (4a,4b,4d,4e) results in an approximately closed streamline whereas the non-divergence-free schemes (4c,4f) exhibit a spiral.

5.4.1. Convergence Test

In this test, we demonstrate the convergence order of the divergence and curl-free schemes presented. For the divergence-free schemes, we initialize the grid using the vector field $\mathbf{u}_{2b}$ or $\mathbf{u}_{3b}$. ~~However, for the 3rd order Balsara divergence-free interpolation scheme~~We also compare against the 3rd order divergence-free scheme of Balsara [4], with the WENO limiting from the accompanying work [9]. In that case, we initialize the grid with face-averaged values, as required by that scheme. In particular, we set $u_{i,j} = \int_{(j-\frac{1}{2})\Delta y}^{(j+\frac{1}{2})\Delta y} \mathbf{u}_{2b}(x,y)dy$ or $u_{i,j,k} = \int_{(k-\frac{1}{2})\Delta z}^{(k+\frac{1}{2})\Delta z} \int_{(j-\frac{1}{2})\Delta y}^{(j+\frac{1}{2})\Delta y} \mathbf{u}_{3b}(x,y,z)dy\,dz$. We then select 10^6 random locations and for each one, compute the error between the interpolated value at the location and the actual value of $\mathbf{u}_{2b}$ or $\mathbf{u}_{3b}$. We run this test for grid resolutions ranging from 8^3 to 4096^3, and plot the maximum error over all of the locations in Fig. 5.

5.4.2. Path continuity

Following [39], we examine the interpolated vector field along a path through a coarse grid (8^2 grid over $[0,1]^2$ in 2D or 8^3 over $[0,1]^3$ in 3D) to illustrate general properties of the interpolation schemes presented, such as discontinuities (jumps), derivative discontinuities (kinks), general accuracy, and the tendency of some schemes to overshoot the data being interpolated. Enough ghost data is filled for all of the interpolation schemes. The paths traced (in 2D and 3D) are exactly the same as in [39], as are the vector fields used.

Scheme	$\mathbf{u}_{3e}$	$\mathbf{u}_{3f}$	$\mathbf{u}_{3g}$	$\mathbf{u}_{3d}$	$\mathbf{u}_{\mathcal{D}^2}$	$\mathbf{u}_{\mathcal{D}^4}$
Linear	2.96×10^{1}	4.29×10^{-2}	3.63×10^{-1}	1.56	1.24×10^{1}	1.21×10^{1}
Cubic	1.45×10^{1}	2.81×10^{-5}	8.16×10^{-10}	1.56	7.04	6.70
Cubic spline	1.17×10^{1}	8.81×10^{-4}	7.80×10^{-3}	1.56	5.70	4.91
$u^{curl,C0}$	9.56×10^{-10}	2.58×10^{-10}	1.95×10^{-3}	1.56	5.32×10^{-10}	6.10×10^{-1}
$u^{curl,C1}$	1.32×10^{-9}	2.79×10^{-10}	1.95×10^{-3}	1.56	1.17×10^{-9}	3.20×10^{-1}
$u^{curl,4th}$	1.36×10^{-9}	3.82×10^{-10}	6.75×10^{-10}	1.56	8.27×10^{-1}	9.94×10^{-10}

Table 4: Maximum curl of interpolation at 10^6 sample points for 4 velocity fields in 3D.

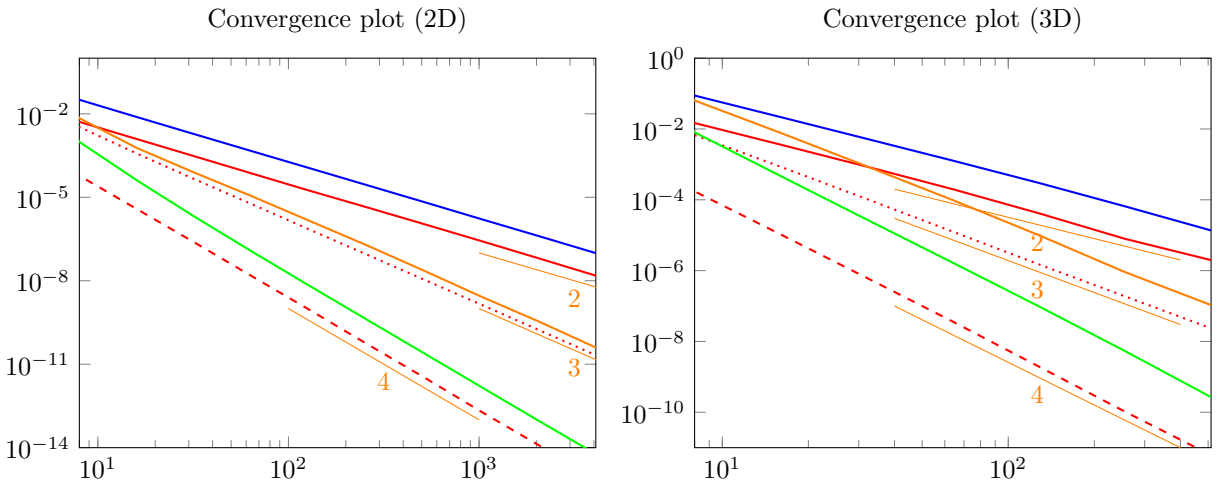


Fig. 5: Convergence test demonstrating the rate of convergence for various divergence-free methods in 2D (left) and 3D (right). This test compares the convergence rates for multilinear interpolation (——), cubic (– – –), cubic spline (……), 3rd order Balsara (——), flux preserving u^{flux} (——), and 4th order $u^{div,4th}$ (——). The x axis is grid resolution, and the y axis is maximum interpolation error at computed at one million fixed sample points. The orange guides show the slopes corresponding to second, third, and fourth order accuracy. Cubic interpolation and $u^{div,4th}$ have fourth order accuracy, and cubic spline and 3rd order Balsara have third order accuracy. Multilinear interpolation and flux preserving are second order accurate. For 3rd order Balsara, we computed face fluxes analytically and used those as inputs to the interpolation instead of the values directly, as is required by the scheme [4]. Without doing this, the interpolation is only second order.

For the first trace, we use the field $\mathbf{u}_{2a}$, which mimics pseudorandom discretely divergence-free data (under both the second and fourth-order stencils) at our resolution. This will tend to emphasize discontinuities and overshoots in the interpolation. The results from this test are shown in Figure 7. It is worth noting that we perform the path trace for curl-free schemes on the same divergence-free data; we use divergence-free data because the scheme from [2] requires this. All of the interpolation schemes proposed work fine on generic data. From these plots, a few properties of these interpolation schemes can be readily observed. (1) The 2nd ~~and 3rd~~ order scheme from [2] is discontinuous; the rest of the schemes are continuous. (2) The schemes $u^{div,C0}$, $u^{curl,C0}$, u^{flux}, and both fourth-order schemes $u^{div,4th}$ and $u^{curl,4th}$ are C^0 continuous and display kinks. Only $u^{div,C1}$ and $u^{curl,C1}$ are C^1 continuous. ~~(3) The 3rd order Balsara scheme is generally poorly behaved on this random-ish data. We suspect this is because the scheme was designed to be used on averaged data, which one might suspect to be smoother. The scheme performs much better on the smoother data used in the second part of this test. (4) The 3rd order Balsara scheme and~~ (3) u^{flux} tends to overshoot the data significantly more than the other schemes (the black dots in the figures show places along the path where grid data is located; values beyond these dots are overshoots). The effects of this could be mitigated in a numerical method using WENO or some other stabilization scheme. It is interesting to note that the 4th order schemes are actually quite well-behaved and do not significantly overshoot the data. (4) The curl-free schemes behave qualitatively very similarly to their divergence-free counterparts (these pairs are shown with solid and dashed lines of the same color). This should not be too surprising, since they use the same splines.

For the second trace, we use the field $\mathbf{u}_{2b}$, which is discretely divergence free and resolved at our resolution and illustrates how the methods behave on smooth data. Since this vector field is accurately interpolated by all of the schemes, the traces overlap, so we instead plot the errors. Both of the 4th order schemes have errors that are indistinguishable from zero in all of the plots (the blue lines, both solid and dashed, coincide at the x axis). Even at this extremely low resolution, these scheme are orders of magnitude more accurate than the second order schemes (the maximum error for them is about 10^{-5}, much smaller than the thickness of the lines in these plots). Note that we have used the raw data as input to the 3rd order Balsara scheme on this test, not face-averaged data. As such, the errors observed in this scheme on this test are comparable to the second order accurate methods. On face averaged data, the scheme is third order accurate, as shown in Section 5.4.1.

We repeat both tests in 3D following the same setup as [39], using the fields $\mathbf{u}_{3a}$ and $\mathbf{u}_{3b}$. The results from these tests are shown in Figure 8 and very closely mirror the 2D results.

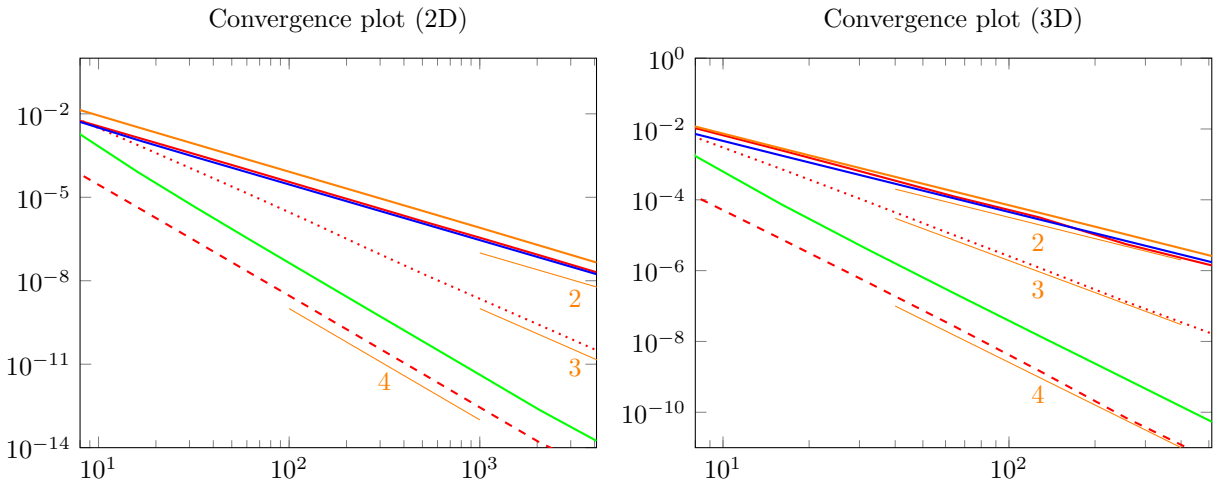


Fig. 6: Convergence test demonstrating the rate of convergence for various curl-free methods in 2D (left) and 3D (right). This test compares the convergence rates for multilinear interpolation (——), cubic (- - -), cubic spline (.....), $u^{curl,C0}$ (——), $u^{curl,C1}$ (——), and 4th order $u^{curl,4th}$ (——). The x axis is grid resolution, and the y axis is maximum interpolation error at computed at one million fixed sample points. The orange guides show the slopes corresponding to second, third, and fourth order accuracy. Cubic interpolation and $u^{curl,4th}$ have fourth order accuracy, and cubic spline has third order accuracy. The rest are second order accurate.

6. Note on other types of analytic consistency

One obvious way to extend the results above would be to extend the idea of divergence-free or curl-free interpolation to, for example, harmonic interpolation. That is, if a scalar field satisfies $\nabla^2 u = 0$ discretely under some finite difference stencil, then interpolate it such that the interpolated field satisfies the differential equation analytically. Or one might consider the same problem for the Helmholtz equation $\nabla^2 u + ku = 0$. Here we show that a harmonic interpolation scheme ($\nabla^2 u = 0$) with basic properties (local, polynomial, continuous) does not exist in 2D.

We begin by noting that there are many harmonic polynomials, but they take a very simple form. In particular, the harmonic polynomials of total degree at most n take the form

$$\phi(x,y) = \sum_{k=0}^{n} \left[A_k (x - iy)^k + B_k (x + iy)^k \right] \tag{106}$$

for some (possibly complex) A_k and B_k. Observe that the polynomials of different total degree are independent from each other. The form above can be obtained by looking for solutions of the form

$$\phi(x,y) = \sum_{j=0}^{n} a_j x^j y^{n-j} \tag{107}$$

and substituting them into the Laplace equation to yield a recurrence on a_j. Solving the recurrence yields the general form.

There are not many degrees of freedom in the general form above, and enforcing continuity with neighboring grid cells constrains many of them. Consider two adjacent cells, (i, j) and $(i, j + 1)$, and evaluate the interpolation scheme at the boundary between them, say at $y = y_0$. Continuity at the boundary between the two cells implies $\phi_{i,j}(x, y_0) = \phi_{i,j+1}(x, y_0)$, which is a polynomial equation in x. For polynomials to be equal, all of their coefficients must independently be equal. Fixing a $n > 0$ we then have

$$A^m_{i,j}(x - iy_0)^n + B^n_{i,j}(x + iy_0)^n = A^m_{i,j+1}(x - iy_0)^n + B^n_{i,j+1}(x + iy_0)^n, \tag{108}$$

which has only the solution $A^m_{i,j} = A^m_{i+1,j}$ and $B^n_{i,j} = B^n_{i+1,j}$. (This can be seen by plugging in $x = iy_0$ or $x = -iy_0$ into the equation above.) The case $n = 0$ is the constant polynomial, which must be the same everywhere to be continuous. Thus, the full polynomial for each cell must be exactly the same as its neighbors. The solution is purely non-local, since the solution everywhere is determined by the solution within one cell.

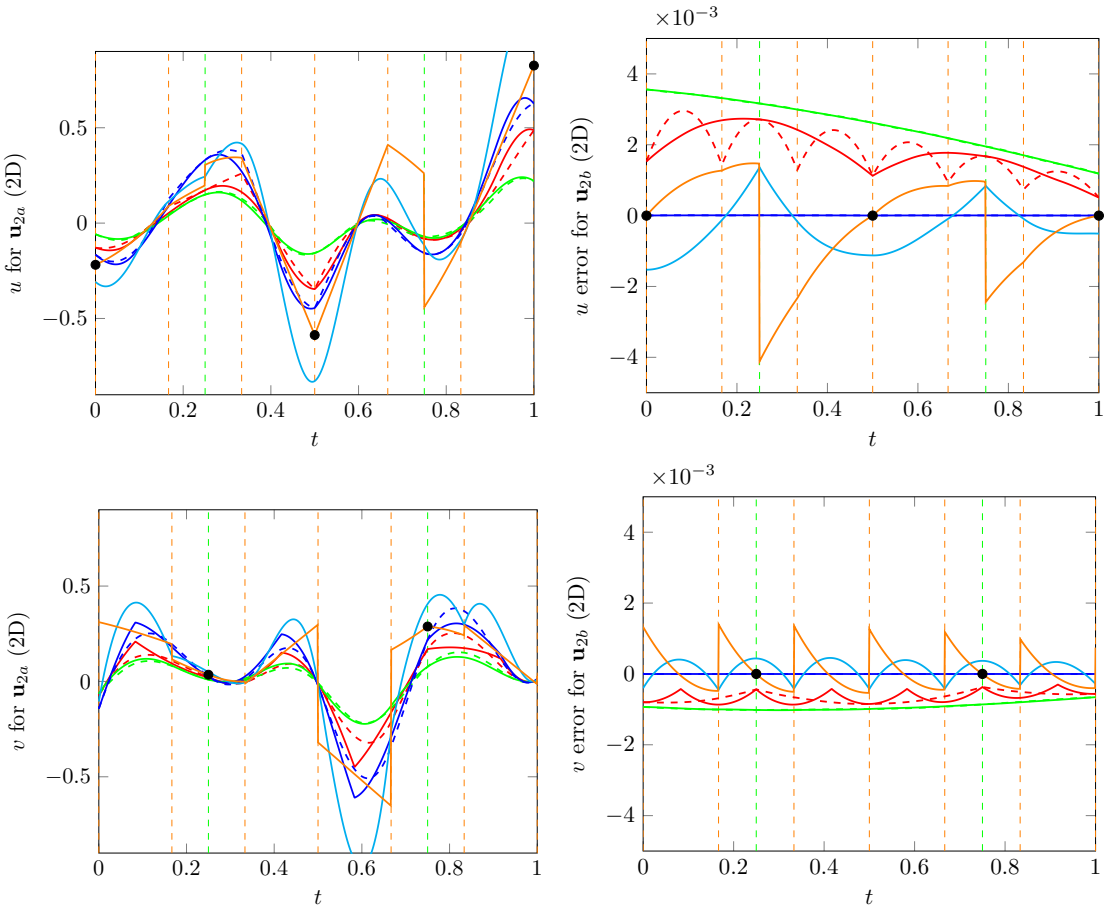


Fig. 7: Path traces for $\mathbf{u}_{2a}$ (first column) and $\mathbf{u}_{2b}$ (second column). Results for u (x component; top row) and v (y component; bottom row) are plotted separately. Both vector fields are discretely divergence free, but the field $\mathbf{u}_{2a}$ is not resolved at the grid resolution being used. Since all of the schemes accurately interpolate $\mathbf{u}_{2b}$, errors (difference between interpolated and analytic fields) are plotted instead to highlight differences between the schemes. The curves represent traces for $u^{div,C0}$ (——), $u^{curl,C0}$ (– – –), $u^{div,C1}$ (——), $u^{curl,C1}$ (– – –), $u^{div,4th}$ (——), $u^{curl,4th}$ (– – –), u^{flux} (——), and 2nd order Balsara (——). Dashed vertical lines indicate cell crossings across u (– – –) and v (– – –) faces. Black dots (•) mark places where the path traces through grid data.

Given the failure of harmonic interpolation in 2D, its solution in 3D seems doomed (since we can extrude the 2D case to 3D to solve the 2D case). This also casts doubt on the Helmholtz equation, of which the Poisson equation is a special case.

Another natural question is whether an interpolating scheme can be both a divergence-free scheme and a curl-free scheme. Although several previous works have targeted such schemes, they have either been discontinuous [43], or do not exactly impose the derivative constraint (i.e. they only approximately impose it via a least squares solve or as a minimization problem) [46, 1, 21, 24]. Our constructions suggest that performing divergence or curl free interpolation exactly, while maintaining continuity, in the same interpolation scheme is likely impossible. ~~since~~ The degrees required of the polynomials are incompatible. Further, if the data were to be both divergence and curl free, the interpolant would have vector Laplacian zero, which would seem to contradict the impossibility of local Laplacian-free interpolation. Of course, the general constructions assume many things (explicit, local, polynomial splines, continuity, symmetry), and breaking these assumptions may broaden the scope of what is achievable.

7. Conclusion

In this work we have presented a flux-preserving scheme for interpolating divergence-free data as well as a general construction for interpolation schemes that produce analytically curl-free vector fields from discretely curl-free data.

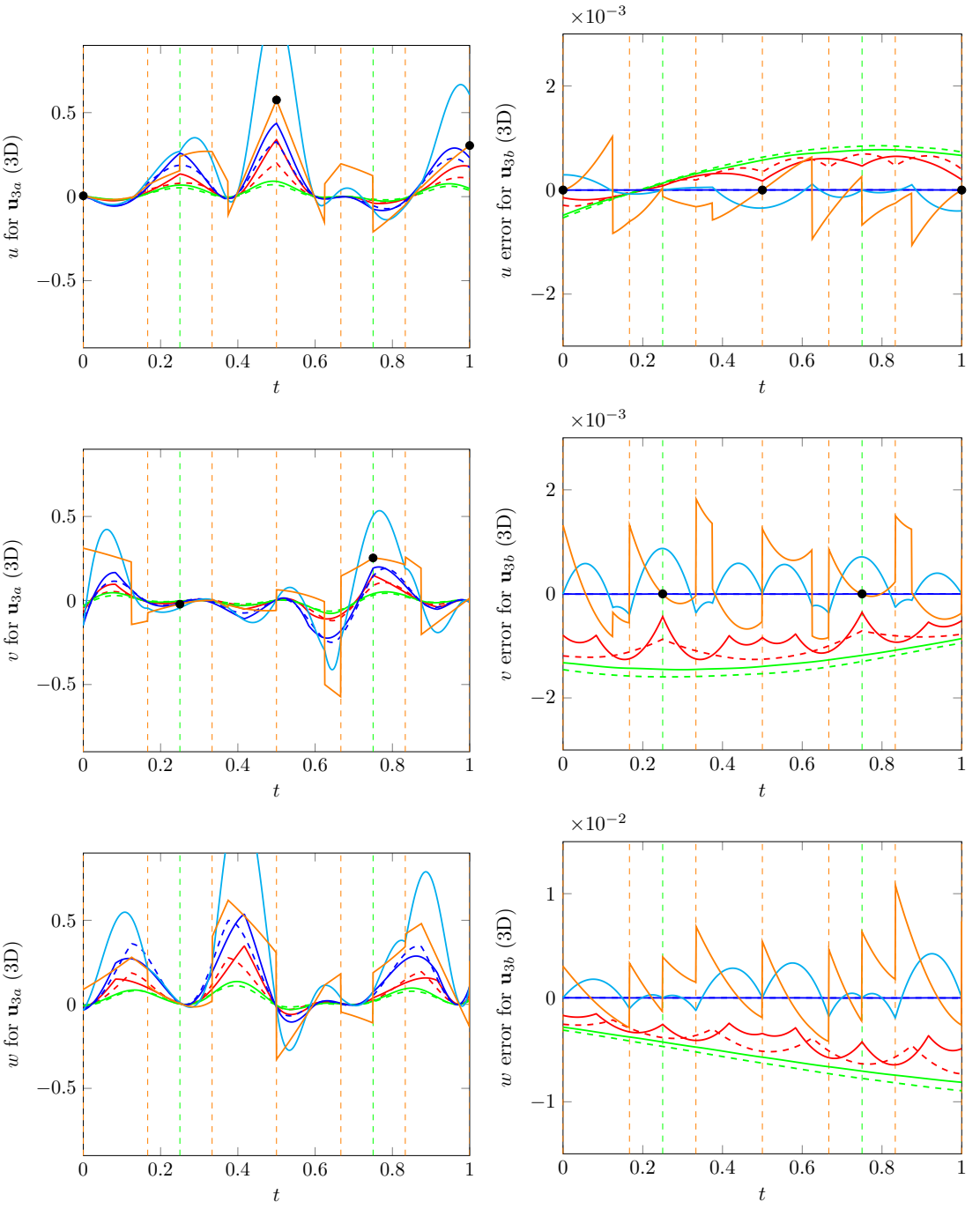


Fig. 8: Path traces for $\mathbf{u}_{3a}$ (first column) and $\mathbf{u}_{3b}$ (second column). Results for u (x component; top row), v (y component; middle row), and w (z component; bottom row) are plotted separately. Both vector fields are discretely divergence free, but the field $\mathbf{u}_{3a}$ is not resolved at the grid resolution being used. Since all of the schemes accurately interpolate $\mathbf{u}_{3b}$, errors (difference between interpolated and analytic fields) are plotted instead to highlight differences between the schemes. The curves represent traces for $u^{div,C0}$ (——), $u^{curl,C0}$ (– – –), $u^{div,C1}$ (——), $u^{curl,C1}$ (– – –), $u^{div,4th}$ (——), $u^{curl,4th}$ (– – –), u^{flux} (——), and 2nd order Balsara (——). Dashed vertical lines indicate cell crossings across u (– – –) and v (– – –) faces. Black dots (•) mark places where the path traces through grid data.

We use this general construction to create C^0 and C^1 continuous interpolation schemes for curl-free data. We also extend the general constructions for divergence-free and curl-free vector fields to a wider class of finite difference stencils, which includes the fourth-order finite difference stencil (and also stencils of higher order). This allows us to overcome the obstruction to higher order interpolation stencils encountered in [39] and construct two fourth-order accurate interpolation schemes. All of the general constructions as well as the five specific schemes operate on MAC grid data, are local and piecewise polynomial, are C^0 or C^1 continuous, and can be applied in 2D or 3D.

8. Acknowledgements

This work was supported in part by National Science Foundation award NSF-2006570 and University of California award M23PL6076.

References

[1] Luca Amodei and Mohamed-Najib Benbourhim. A vector spline approximation. *Journal of approximation theory*, 67(1):51–79, 1991.
[2] Dinshaw S Balsara. Divergence-free adaptive mesh refinement for magnetohydrodynamics. *Journal of Computational Physics*, 174(2): 614–648, 2001.
[3] Dinshaw S Balsara. Second-order-accurate schemes for magnetohydrodynamics with divergence-free reconstruction. *The Astrophysical Journal Supplement Series*, 151(1):149, 2004.
[4] Dinshaw S Balsara. Divergence-free reconstruction of magnetic fields and weno schemes for magnetohydrodynamics. *Journal of Computational Physics*, 228(14):5040–5056, 2009.
[5] Dinshaw S Balsara and Michael Dumbser. Divergence-free mhd on unstructured meshes using high order finite volume schemes based on multidimensional riemann solvers. *Journal of Computational Physics*, 299:687–715, 2015.
[6] Dinshaw S Balsara and Jongsoo Kim. A comparison between divergence-cleaning and staggered-mesh formulations for numerical magnetohydrodynamics. *The Astrophysical Journal*, 602(2):1079, 2004.
[7] Dinshaw S Balsara and Costas D Sarris. A systematic approach to adaptive mesh refinement for computational electrodynamics. *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, 2023.
[8] Dinshaw S Balsara and Daniel S Spicer. A staggered mesh algorithm using high order godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations. *Journal of Computational Physics*, 149(2):270–292, 1999.
[9] Dinshaw S. Balsara, Tobias Rumpf, Michael Dumbser, and Claus-Dieter Munz. Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. 228(7):2480–2516. URL `https://www.sciencedirect.com/science/article/pii/S0021999108006396`.
[10] Dinshaw S Balsara, Sudip Garain, Allen Taflove, and Gino Montecinos. Computational electrodynamics in material media with constraint-preservation, multidimensional riemann solvers and sub-cell resolution–part ii, higher order fvtd schemes. *Journal of Computational Physics*, 354:613–645, 2018.
[11] Dinshaw S Balsara, Roger Käppeli, Walter Boscheri, and Michael Dumbser. Curl constraint-preserving reconstruction and the guidance it gives for mimetic scheme design. *Communications on Applied Mathematics and Computation*, pages 1–60, 2021.
[12] Dinshaw S Balsara, Saurav Samantaray, and Sethupathy Subramanian. Efficient weno-based prolongation strategies for divergence-preserving vector fields. *Communications on Applied Mathematics and Computation*, 5(1):428–484, 2023.
[13] Yuanxun Bao, Aleksandar Donev, Boyce E Griffith, David M McQueen, and Charles S Peskin. An immersed boundary method with divergence-free velocity interpolation and force spreading. *Journal of computational physics*, 347:183–206, 2017.
[14] Mohammed-Najib Benbourhim and Abderrahman Bouhamidi. Meshless pseudo-polyharmonic divergence-free and curl-free vector fields approximation. *SIAM journal on mathematical analysis*, 42(3):1218–1245, 2010.
[15] Kai Bittner and Karsten Urban. On interpolatory divergence-free wavelets. *Mathematics of computation*, 76(258):903–929, 2007.
[16] Daniele Boffi, Franco Brezzi, Michel Fortin, et al. *Mixed finite element methods and applications*, volume 44. Springer, 2013.
[17] Walter Boscheri, Giuseppe Roberto Pisaturo, and Maurizio Righetti. High-order divergence-free velocity reconstruction for free surface flows on unstructured voronoi meshes. *International Journal for Numerical Methods in Fluids*, 90(6):296–321, 2019.
[18] A. Buffa, J. Rivas, G. Sangalli, and R. Vázquez. Isogeometric Discrete Differential Forms in Three Dimensions. 49(2):818–844. ISSN 0036-1429. doi: 10.1137/100786708. URL `https://epubs.siam.org/doi/10.1137/100786708`.
[19] Jumyung Chang, Vinicius C Azevedo, and Christopher Batty. Curl-flow: Pointwise incompressible velocity interpolation forgrid-based fluids. *arXiv preprint arXiv:2104.00867*, 2021.
[20] Erwan Deriaz and Valérie Perrier. Towards a divergence-free wavelet method for the simulation of 2d/3d turbulent flows. *J. Turbul*, 7(3): 1–37, 2006.
[21] Fabrice Dodu and Christophe Rabut. Irrotational or divergence-free interpolation. *Numerische Mathematik*, 98(3):477–498, 2004.
[22] Kathryn P Drake, Edward J Fuselier, and Grady B Wright. A partition of unity method for divergence-free or curl-free radial basis function approximation. *SIAM Journal on Scientific Computing*, 43(3):A1950–A1974, 2021.
[23] Kathryn P Drake, Edward J Fuselier, and Grady B Wright. Implicit surface reconstruction with a curl-free radial basis function partition of unity method. *SIAM Journal on Scientific Computing*, 44(5):A3018–A3040, 2022.
[24] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables: Proceedings of a Conference Held at Oberwolfach April 25–May 1, 1976*, pages 85–100. Springer, 1977.
[25] Michael Dumbser, Dinshaw S Balsara, Maurizio Tavelli, and Francesco Fambri. A divergence-free semi-implicit finite volume scheme for ideal, viscous, and resistive magnetohydrodynamics. *International Journal for Numerical Methods in Fluids*, 89(1-2):16–42, 2019.
[26] John Andrews Evans. Divergence-free B-spline discretizations for viscous incompressible flows. URL `https://repositories.lib.utexas.edu/handle/2152/ETD-UT-2011-12-4506`.

[27] Patricio Farrell, Kathryn Gillow, and Holger Wendland. Multilevel interpolation of divergence-free vector fields. *IMA Journal of Numerical Analysis*, 37(1):332–353, 2017.

[28] John M Finn and Luis Chacón. Volume preserving integrators for solenoidal fields on a grid. *Physics of Plasmas*, 12(5):054503, 2005.

[29] DC Handscomb. Spline representation of incompressible flow. *IMA journal of numerical analysis*, 4(4):491–502, 1984.

[30] Arijit Hazra, Praveen Chandrashekar, and Dinshaw S Balsara. Globally constraint-preserving fr/dg scheme for maxwell's equations at all orders. *Journal of computational physics*, 394:298–328, 2019.

[31] Arpiruk Hokpunna and Michael Manhart. Compact fourth-order finite volume method for numerical solutions of navier–stokes equations on staggered grids. *Journal of Computational Physics*, 229(20):7545–7570, 2010.

[32] Yingchun Jiang. Anisotropic curl-free wavelets with boundary conditions. *Journal of Inequalities and Applications*, 2012:1–10, 2012.

[33] Yingchun Jiang and Youming Liu. Interpolatory curl-free wavelets and applications. *International journal of wavelets, multiresolution and Information processing*, 5(05):843–858, 2007.

[34] Shengtai Li. High order central scheme on overlapping cells for magneto-hydrodynamic flows with and without constrained transport method. *Journal of Computational Physics*, 227(15):7368–7393, 2008.

[35] Shengtai Li. A fourth-order divergence-free method for mhd flows. *Journal of Computational Physics*, 229(20):7893–7910, 2010.

[36] Jean-Claude Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numerische Mathematik*, 35:315–341, 1980.

[37] Pierre-Arnaud Raviart and Jean-Marie Thomas. A mixed finite element method for 2-nd order elliptic problems. In *Mathematical Aspects of Finite Element Methods: Proceedings of the Conference Held in Rome, December 10–12, 1975*, pages 292–315. Springer, 2006.

[38] Bharath Ravu, Murray Rudman, Guy Metcalfe, Daniel R Lester, and Devang V Khakhar. Creating analytically divergence-free velocity fields from grid-based data. *Journal of Computational Physics*, 323:75–94, 2016.

[39] Craig Schroeder, Ritoban Roy Chowdhury, and Tamar Shinar. Local divergence-free polynomial interpolation on mac grids. *Journal of Computational Physics*, page 111500, 2022.

[40] Zachary J Silberman, Thomas R Adams, Joshua A Faber, Zachariah B Etienne, and Ian Ruchlin. Numerical generation of vector potentials from specified magnetic fields. *Journal of Computational Physics*, 379:421–437, 2019.

[41] Arno Solin, Manon Kok, Niklas Wahlström, Thomas B Schön, and Simo Särkkä. Modeling and interpolation of the ambient magnetic field by gaussian processes. *IEEE Transactions on robotics*, 34(4):1112–1127, 2018.

[42] Rob Stevenson. Divergence-free wavelet bases on the hypercube: Free-slip boundary conditions, and applications for solving the instationary stokes equations. *Mathematics of computation*, 80(275):1499–1523, 2011.

[43] G Tóth and PL Roe. Divergence-and curl-preserving prolongation and restriction formulas. *Journal of Computational Physics*, 180(2): 736–750, 2002.

[44] Karsten Urban. *Wavelets in numerical simulation: problem adapted construction and applications*, volume 22. Springer Science & Business Media, 2002.

[45] Minglei Yang, Diego del Castillo-Negrete, Guannan Zhang, and Matthew Beidler. A divergence-free constrained magnetic field interpolation method for scattered data. *arXiv preprint arXiv:2207.12349*, 2022.

[46] Inas Yassine and Tim McGraw. 4th order diffusion tensor interpolationwith divergence and curl constrained bézier patches. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 634–637. IEEE, 2009.