

Local divergence-free polynomial interpolation on MAC grids

Craig Schroeder^{a,*}, Ritoban Roy Chowdhury^a, Tamar Shinar^{a,*}

^aComputer Science and Engineering, University of California, Riverside, 351 Winston Chung Hall, Riverside, CA 92521-0429

ARTICLE INFO

Article history:

ABSTRACT

Divergence-free vector fields play an important role in many types of problems, including the incompressible Navier-Stokes equations and the equations for magnetohydrodynamics. In the discrete setting, these fields are often obtained by projection, resulting in a discrete approximation of the continuous field that is discretely divergence-free. For many applications, such as tracing particles, this discrete field must then be extended to the entire region using interpolation. This interpolated field is continuous and differentiable (almost everywhere), but in general it will not be divergence-free. In this paper, we construct approximation schemes with the property that discretely divergence-free data interpolates to an analytically divergence-free vector field. Our focus is on data stored in a MAC grid layout that is divergence free under the second order central difference stencil, a case that is common in projection methods for the Navier-Stokes equations. While existing schemes with this property are known, they tend to be global (the interpolated value at a point depends on data stored on the grid far from that point) or discontinuous. We construct C^0 and C^1 continuous approximation schemes for 2D and 3D that are local and satisfy the divergence-free property. We also construct interpolating versions of the schemes that reproduce the MAC data at face centers. All eight schemes are explicit piecewise polynomials over small stencils.

© 2022 Elsevier Inc. All rights reserved.

1. Introduction

In this paper, we consider the problem of divergence-free interpolation of a vector field sampled on a Marker-and-Cell (MAC) grid and satisfying a discrete divergence-free condition. MAC grids, also known as staggered grids, were first proposed by Harlow and Welch [20] for storing velocity samples for incompressible fluid simulation. In particular, the components of a fluid velocity are stored at their respective cell faces: x -components are stored at faces whose normal points in the x direction, and y and z components are stored analogously (See Fig. 1). A key advantage of the staggered grid approach is that second-order accurate, centered finite difference discretizations of the gradient and divergence terms do not suffer from the odd-even pressure decoupling problem that can occur for collocated grid

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

data [32]. A similar staggered grid is commonly utilized in electromagnetism, where the components of the magnetic field B (or H) are stored at cell faces, so the divergence of the magnetic field (which must be zero, according to Maxwell's equations) can be calculated at cell centers [37].

Discretely divergence-free vector fields arise commonly in fractional-step methods for incompressible flow [11, 12]. In these methods, the velocity field is projected at every time step to satisfy a discrete divergence-free condition on each computational cell of the MAC grid. Typically, other parts of the algorithm require the velocity field to be evaluated at arbitrary points in the domain. Componentwise interpolation strategies do not yield a velocity field that is divergence-free pointwise, and this leads to inaccurate volume sources and sinks in the interpolated flow field.

The problem of interpolating vector-valued data so that the interpolant satisfies a divergence-free condition has seen interest in several areas of computational physics. In magnetohydrodynamics (MHD), Brackbill and Barnes [8] showed that small, nonzero divergence of the magnetic fields leads to spurious velocities when solving the conservative momentum equation. Several methods preserving the divergence-free property of the magnetic field have been developed, such as those by Balsara and colleagues [6, 5, 4]. Balsara further designed divergence-free interpolation methods for the magnetic field to support prolongation for solvers using adaptive mesh refinement [2, 3, 4], noting that refinement was insufficient in controlling errors in the divergence of the magnetic field. Similarly, Cervone et al. [9] used divergence-free interpolation in a hierarchical finite element method. Divergence-free interpolation strategies have also been sought in the context of fluid-structure interaction problems. Jenny et al. [21] developed a conservative velocity interpolation method for use in a hybrid mesh-particle scheme for turbulent reactive flow requiring accurate particle tracking. The hybrid immersed boundary method, where the Lagrangian structure moves with the interpolated Eulerian flow velocity, has been shown to exhibit better volume preservation, improved advection of tracer particles, and more uniform particle distribution when divergence-free interpolation is employed [7]. It has also been demonstrated that divergence-free interpolation allows for more accurate tracking of Lagrangian trajectories and Lagrangian coherent structures over long times [33] and better conservation of constants of motion [28]. Vennell and Beatson [35] use divergence-free interpolation to obtain improved accuracy in reconstruction of flow eddies from sparse, scattered data. Divergence-free basis functions have also been used in discontinuous Galerkin finite element methods for numerical solution of the Maxwell equations [13], magnetohydrodynamics equations [26], and incompressible flow equations [25]. Divergence-free reconstruction has been shown to be important for pressure-robustness of solutions for incompressible flow using mixed finite element methods [22, 24]. Kanschat [23] devised a discontinuous Galerkin scheme that when used with the lowest order Raviart-Thomas basis functions on a rectangular mesh is equivalent to the MAC scheme and gives a divergence-free interpolation scheme on the MAC grid, with tangential velocities discontinuous across cell boundaries.

Previous approaches for divergence-free interpolation of vector fields fall into several broad categories. Early work considered interpolation of sparse, scattered data subject to constraints on the interpolant. Focusing on applications to meteorology, Amodei and Benbourhim [1] developed a variational spline formulation for divergence-free or curl-free interpolation. Methods based on polyharmonic splines were derived in Handscomb [17, 19] and Dodu and Rabut [14]. Narcowich and Ward [31] introduced interpolation strategies based on matrix-valued radial basis functions (RBFs), and Lowitzsch [27] developed matrix-valued RBFs with compact support to improve computational efficiency. Matrix-valued RBFs were applied to MHD simulations by McNally [29]. While these previous methods required solution of a globally coupled system, a local method based on a partition of unity framework was recently developed by Drake et al. [15].

Many divergence-free interpolation approaches are based on reconstruction of a continuous vector potential from the vector field samples. The curl of the vector potential then gives a vector field that is pointwise divergence-free by construction. Finn and Chacón [16] reconstruct a tricubic spline representation of the vector potential to obtain volume-preserving integrators for solenoidal fields on a grid. Ravu et al. [33] directly fits the derivatives of the spline function for the potential to the velocities and solves a large global system of equations for the spline coefficients. Bao et al. [7] solve a vector Poisson equation to determine the vector potential. To reduce the computational cost, Silberman et al. [34] and Chang et al. [10] formulate methods based on sweeping which then require only the solution of a scalar Poisson problem for the vector potential.

Another set of approaches, which includes the present work, employ spline-based reconstruction of vector fields satisfying the pointwise divergence-free condition on structured meshes. Handscomb [18] presents a natural-spline-based method that uses a global solve to compute the coefficients. Several other approaches use more computationally efficient local reconstructions. Jenny et al. [21] developed conservative vector interpolation schemes for collocated grids in two dimensions. The reconstruction of the vector field is continuous only in normal direction across cell

faces. Meyer and Jenny [30] studied the evolution of passive particles under different velocity interpolation schemes, comparing nonconservative bilinear interpolation with the first and second order conservative scheme of Jenny et al. [21] and illustrate visually that particle distribution remained more uniform when conservative interpolation was used. The scheme of Jenny et al. [21] was extended to three dimensions by Wang et al. [36]. Like the MAC grids we consider, Balsara [2] considered Cartesian grids with the normal components of vectors stored at face centers. He formulates a polynomial basis for reconstruction in each cell and solves for coefficients enforcing the divergence-free constraint. Similar to the method of Jenny et al. [21], the method yields a vector field where u is quadratic in x and linear in y (with v analogous) and has normal continuity across cell faces but is discontinuous in tangential components. In the numerical tests that follow, we study this method and compare it with our approach. Balsara [3] devised an MHD simulation method that improved on [6] and extended [2] to complex geometries and unstructured meshes. Balsara [4] further shows how to extend the scheme to third and higher order accuracy using tensor products of the Legendre polynomials.

In this work, we present an efficient, local reconstruction of vector fields on MAC grids that is divergence-free and continuous across cell boundaries. In Section 2.2, we present a family of second-order accurate vector field reconstructions based on tensor products of B-spline basis functions which depend on a small, local stencil of the computational grid. These schemes approximate rather than interpolate the discrete vector field data. In Section 2.5 we formulate a more general construction based on spline chains. These can be used to incorporate divergence-free corrections to the B-spline formulation resulting in interpolating schemes. We give numerical results for our construction demonstrating the analytical divergence-free property, continuity, and convergence order on all schemes that we present: C^0 and C^1 approximating B-spline reconstructions and C^0 and C^1 interpolating reconstructions using spline chains (which we refer to as C^{0i} and C^{1i} , respectively). We compare our results with linear interpolation, cubic interpolation, cubic spline interpolation, and the second-order method of Balsara [2, 3].

2. Divergence-free construction

In this work, we assume that our input is data stored in a MAC grid layout with components $u_{i+\frac{1}{2},j,k}$, $v_{i,j+\frac{1}{2},k}$, and $w_{i,j,k+\frac{1}{2}}$ located at the centers of faces at positions $(x_{i+\frac{1}{2}}, y_j, z_k)$, $(x_i, y_{j+\frac{1}{2}}, z_k)$, and $(x_i, y_j, z_{k+\frac{1}{2}})$, where $x_i = i\Delta x$, $y_j = j\Delta y$, and $z_k = k\Delta z$. We are interested in schemes that produce from this data an interpolated vector field $\hat{\mathbf{u}}(\mathbf{x}) = (\hat{u}(x, y, z), \hat{v}(x, y, z), \hat{w}(x, y, z))$; we generally refer to schemes of this form as *interpolation schemes*. We restrict our focus to schemes of the form

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

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

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

where $P(x, y, z)$ is piecewise polynomial with compact support. In particular, the schemes we consider are explicit (they do not involve solving global systems). In order to guarantee second order convergence of the interpolation scheme, we also require $\hat{u}(x, y, z) = ax + by + cz + d$ when $u_{i+\frac{1}{2},j,k} = ax_{i+\frac{1}{2}} + by_j + cz_k + d$. That is, if the input data is sampled from an affine function, then the interpolation scheme recovers the affine function exactly.

We define the *discrete divergence* of a grid cell i, j, k to be

$$d_{i,j,k} = \frac{u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}}{\Delta x} + \frac{v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}}{\Delta y} + \frac{w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}}{\Delta z}. \quad (4)$$

If $d_{i,j,k} = 0$ for all grid cells, then we say that the input data is *discretely divergence free*. If an interpolation scheme satisfies $\nabla \cdot \hat{\mathbf{u}}(\mathbf{x}) = 0$ for all discretely divergence-free inputs, then we call the interpolation scheme *analytically divergence free*. If a scheme satisfies the property $\hat{u}(x_{i+\frac{1}{2}}, y_j, z_k) = u_{i+\frac{1}{2},j,k}$, then we say that the interpolation scheme is *interpolating* (in the sense of interpolating polynomials). We say that an interpolating scheme is C^0 if it is continuous everywhere and C^1 if in addition its gradient is continuous everywhere.

In this paper, we present four interpolation schemes in 2D and a corresponding set of four interpolation schemes in 3D, which we call C^0 (C^0 continuous, not interpolating), C^1 (C^1 continuous, not interpolating), C^{0i} (C^0 continuous, interpolating), C^{1i} (C^1 continuous, interpolating). We call the corresponding interpolation functions $\hat{u}^{C^0}(\mathbf{x})$, $\hat{u}^{C^1}(\mathbf{x})$, $\hat{u}^{C^{0i}}(\mathbf{x})$, and $\hat{u}^{C^{1i}}(\mathbf{x})$. Of these, the non-interpolating versions are simplest, so we begin with their construction.

2.1. B-spline basis functions

The simplest divergence-free interpolation schemes are formulated in terms of the B-spline basis functions, centered at the origin. These are piecewise polynomial functions with compact support, which can be generated by the recurrence

$$B^0(x) = \begin{cases} 1 & -\frac{1}{2} \leq x \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}, \quad B^{n+1}(x) = \frac{\left(\frac{n}{2} + 1 - x\right) B^n\left(x - \frac{1}{2}\right) + \left(\frac{n}{2} + 1 + x\right) B^n\left(x + \frac{1}{2}\right)}{n + 1}. \quad (5)$$

Observe that $B^n(x)$ is composed of $n + 1$ nonzero polynomials of degree n . It is C^{n-1} continuous everywhere and nonzero exactly when $-\frac{n+1}{2} < x < \frac{n+1}{2}$. These functions also satisfy $B^n(x) = B^n(-x)$ and

$$\frac{d}{dx} B^{n+1}(x) = B^n\left(x + \frac{1}{2}\right) - B^n\left(x - \frac{1}{2}\right). \quad (6)$$

The splines up to degree four are

$$B^1(x) = \begin{cases} 1 - |x| & |x| \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad B^2(x) = \begin{cases} \frac{3}{4} - x^2 & |x| \leq \frac{1}{2} \\ \frac{1}{8}(2|x| - 3)^2 & \frac{1}{2} < |x| \leq \frac{3}{2} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$B^3(x) = \begin{cases} \frac{2}{3} - x^2 + \frac{1}{2}|x|^3 & |x| \leq 1 \\ \frac{1}{6}(2 - |x|)^3 & 1 < |x| \leq 2 \\ 0 & \text{otherwise} \end{cases} \quad B^4(x) = \begin{cases} \frac{115}{192} - \frac{5}{8}x^2 + \frac{1}{4}x^4 & |x| \leq \frac{1}{2} \\ \frac{55}{96} + \frac{5}{24}|x| - \frac{5}{4}x^2 + \frac{5}{6}|x|^3 - \frac{1}{6}x^4 & \frac{1}{2} < |x| \leq \frac{3}{2} \\ \frac{1}{384}(5 - 2|x|)^4 & \frac{3}{2} < |x| \leq \frac{5}{2} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

2.2. Base scheme construction

Let $\hat{\mathbf{u}}^n(\mathbf{x}) = (\hat{u}^n(x, y, z), \hat{v}^n(x, y, z), \hat{w}^n(x, y, z))$ where

$$\hat{u}^n(x, y, z) = \sum_{i,j,k} u_{i+\frac{1}{2},j,k} B^{n+1}\left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x}\right) B^n\left(\frac{y - y_j}{\Delta y}\right) B^n\left(\frac{z - z_k}{\Delta z}\right) \quad (9)$$

$$\hat{v}^n(x, y, z) = \sum_{i,j,k} v_{i,j+\frac{1}{2},k} B^n\left(\frac{x - x_i}{\Delta x}\right) B^{n+1}\left(\frac{y - y_{j+\frac{1}{2}}}{\Delta y}\right) B^n\left(\frac{z - z_k}{\Delta z}\right) \quad (10)$$

$$\hat{w}^n(x, y, z) = \sum_{i,j,k} w_{i,j,k+\frac{1}{2}} B^n\left(\frac{x - x_i}{\Delta x}\right) B^n\left(\frac{y - y_j}{\Delta y}\right) B^{n+1}\left(\frac{z - z_{k+\frac{1}{2}}}{\Delta z}\right). \quad (11)$$

We can define the schemes C^0 and C^1 (and in fact C^{n-1} for any $n \geq 1$) as $\hat{\mathbf{u}}^{C^0}(\mathbf{x}) = \hat{\mathbf{u}}^1(\mathbf{x})$ and $\hat{\mathbf{u}}^{C^1}(\mathbf{x}) = \hat{\mathbf{u}}^2(\mathbf{x})$. (Note that the 1 in $\hat{\mathbf{u}}^1$ refers to the polynomial degree of the B-splines, while the 1 in $\hat{\mathbf{u}}^{C^1}$ refers to the continuity.) Although the sum is written over all indices for simplicity, only a finite number are actually required due to the compact support of $B^n(x)$. The interpolation scheme is defined similarly in 2D. Although the scheme is meaningful for $n \geq 0$, $\hat{\mathbf{u}}^0$ is discontinuous and thus not very useful. We note that $\hat{\mathbf{u}}^0$ is equivalent to the divergence-free discontinuous Galerkin scheme presented in Kanschat [23]. We focus our numerical tests on $\hat{\mathbf{u}}^1$ and $\hat{\mathbf{u}}^2$, since they are likely to be the most useful in practice. The stencils for $\hat{\mathbf{u}}^1$ and $\hat{\mathbf{u}}^2$ in 2D are shown in Figure 1. We note that this construction is essentially the same as Handscomb [18], which used natural splines and solved a linear system to obtain the coefficients to achieve C^2 continuity using quadratic and cubic splines. We use the same combination of B-splines but as a local C^1 scheme, sacrificing continuity for locality.

The interpolated vector field $\hat{\mathbf{u}}^n(\mathbf{x})$ satisfies a number of simple properties:

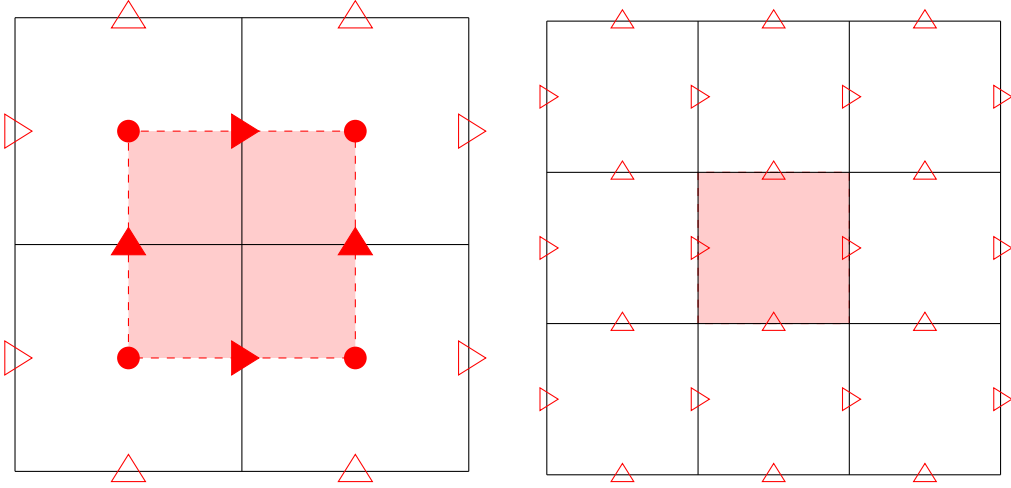


Fig. 1: The MAC grid stores data in a staggered arrangement, with pressures at cell centers and normal velocity components at cell boundaries. (Left) The stencil for $\mathbf{u}^1 = (\hat{u}^1, \hat{v}^1)$ over the shaded region is shown. $\hat{u}^1$ and $\hat{v}^1$ are polynomials (not piecewise) over this region. $\hat{u}^1$ depends on x -faces ($\triangleright, \blacktriangleright$), and $\hat{v}^1$ depends on y -faces ($\triangle, \blacktriangle$). The interpolated values at the corners of the shaded region ($\bullet$) are central averages of the surrounding samples. $\hat{u}^1$ may alternatively be implemented as a tensor product Bezier patch (degree 1 in the y direction and degree 2 in the x direction) with control points on the shaded region ($\blacktriangleright, \bullet$). $\hat{v}^1$ is similar, and the construction extends naturally to 3D. The interpolation stencil is only slightly wider than a linear interpolation stencil. (Right) The stencil for $\mathbf{u}^2 = (\hat{u}^2, \hat{v}^2)$ over the shaded region is shown. $\hat{u}^2$ and $\hat{v}^2$ are polynomials (not piecewise) over this region. $\hat{u}^2$ depends on x -faces ($\triangleright$), and $\hat{v}^2$ depends on y -faces ($\triangle$).

1. $\hat{\mathbf{u}}^n(\mathbf{x})$ is piecewise polynomial with total degree $3n + 1$ (or $2n + 1$ in 2D).
2. $\hat{\mathbf{u}}^n(\mathbf{x})$ is C^{n-1} -continuous.
3. If n is even, then $\hat{\mathbf{u}}^n(\mathbf{x})$ is a polynomial on MAC grid cells.
4. If n is odd, then $\hat{\mathbf{u}}^n(\mathbf{x})$ is a polynomial on dual cells (cells centered at grid nodes).
5. The polynomials depend on the data stored at the faces of the neighboring $(n + 1) \times (n + 1) \times (n + 1)$ MAC cells.
6. The analytic divergence $\nabla \cdot \hat{\mathbf{u}}^n(\mathbf{x})$ interpolates the discrete divergences of the cells it depends on (see Section 2.3).
7. If the MAC vector field is discretely divergence free in the neighboring cells, then the interpolated vector field will be analytically divergence free ($\nabla \cdot \hat{\mathbf{u}}^n(\mathbf{x}) = 0$).
8. The interpolation scheme ($n \geq 1$) exactly interpolates affine data.
9. For $n \geq 1$, $\hat{\mathbf{u}}^n(\mathbf{x})$ are not interpolating polynomials. That is, $\hat{u}^n(x_{i+\frac{1}{2}}, y_j, z_k) \neq u_{i+\frac{1}{2}, j, k}$ in general.

Property 6 is proven in the next section, and the divergence-free property (Property 7) readily follows from it. In particular, an analytically divergence-free vector field is obtained from a discretely divergence-free field without any need for an additional global solve. The rest of the properties follow readily from the properties of the B-spline basis polynomials.

2.3. Divergence-free vector field

Due to the properties of the B-spline basis polynomials, the partial derivatives (denoted with $\hat{\mathbf{u}}^n_x$) of the interpolated vector take the form of a central difference. For example, from (6) and (9) we have (following Handscomb [18])

$$\hat{u}^n_x(x, y, z) = \sum_{i,j,k} \frac{u_{i+\frac{1}{2},j}}{\Delta x} \left(B^n \left(\frac{x - x_i}{\Delta x} \right) - B^n \left(\frac{x - x_{i+1}}{\Delta x} \right) \right) B^n \left(\frac{y - y_j}{\Delta y} \right) B^n \left(\frac{z - z_k}{\Delta z} \right) \quad (12)$$

$$= \sum_{i,j,k} \frac{u_{i+\frac{1}{2},j}}{\Delta x} B^n \left(\frac{x - x_i}{\Delta x} \right) B^n \left(\frac{y - y_j}{\Delta y} \right) B^n \left(\frac{z - z_k}{\Delta z} \right) - \sum_{i,j,k} \frac{u_{i+\frac{1}{2},j}}{\Delta x} B^n \left(\frac{x - x_{i+1}}{\Delta x} \right) B^n \left(\frac{y - y_j}{\Delta y} \right) B^n \left(\frac{z - z_k}{\Delta z} \right) \quad (13)$$

$$= \sum_{i,j,k} \frac{u_{i+\frac{1}{2},j}}{\Delta x} B^n \left(\frac{x - x_i}{\Delta x} \right) B^n \left(\frac{y - y_j}{\Delta y} \right) B^n \left(\frac{z - z_k}{\Delta z} \right) - \sum_{i,j,k} \frac{u_{i-\frac{1}{2},j}}{\Delta x} B^n \left(\frac{x - x_i}{\Delta x} \right) B^n \left(\frac{y - y_j}{\Delta y} \right) B^n \left(\frac{z - z_k}{\Delta z} \right) \quad (14)$$

$$= \sum_{i,j,k} \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} B^n \left(\frac{x - x_i}{\Delta x} \right) B^n \left(\frac{y - y_j}{\Delta y} \right) B^n \left(\frac{z - z_k}{\Delta z} \right) \quad (15)$$

We substitute in the derivative property of the B-splines (Eqn. 6) into the interpolation scheme (Eqn. 9) to obtain Eqn. 12¹. Expanding out the parenthesis yields Eqn. 13. To obtain Eqn. 14, we shift the index i to $i - 1$ in the second summation (so x_{i+1} becomes x_i and $u_{i+\frac{1}{2}}$ becomes $u_{i-\frac{1}{2}}$). Because the summation is written over all grid cells, we do not have to modify its bounds to account for the shift. Finally, we can recombine the two summations, showing that the derivative of $\hat{u}$ is the n -th order B-spline interpolation of the central difference approximations of the derivative (Eqn. 15). The derivatives $\hat{v}_y^n(x, y, z)$ and $\hat{w}_z^n(x, y, z)$ are similar. We define the *discrete divergence* of a grid cell to be

$$d_{i,j,k} = \frac{u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}}{\Delta x} + \frac{v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}}{\Delta y} + \frac{w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}}{\Delta z}. \quad (16)$$

With this definition we see that

$$\nabla \cdot \hat{\mathbf{u}}^n(\mathbf{x}) = \hat{u}_x^n(x, y, z) + \hat{v}_y^n(x, y, z) + \hat{w}_z^n(x, y, z) = \sum_{i,j,k} d_{i,j,k} B^n\left(\frac{x - x_i}{\Delta x}\right) B^n\left(\frac{y - y_j}{\Delta y}\right) B^n\left(\frac{z - z_k}{\Delta z}\right). \quad (17)$$

The divergence of the interpolated vector field interpolates the discrete divergence. In particular, if the MAC vector field is discretely divergence free ($d_{i,j,k} = 0$) then the interpolated vector field will be analytically divergence free ($\nabla \cdot \hat{\mathbf{u}}^n(\mathbf{x}) = 0$). This property is local: the vector field will be analytically divergence free as long as nearby cells are discretely divergence free.

2.4. Spline chains

The property of B-splines that leads to divergence-free vector fields is (6). More generally, we say that $R^n(x)$ and $R^{n+1}(x)$ are part of a *chain* if

$$\frac{d}{dx} R^{n+1}(x) = R^n\left(x + \frac{1}{2}\right) - R^n\left(x - \frac{1}{2}\right). \quad (18)$$

The B-splines form one long chain. The superscript indicates that spline R^n has polynomial degree n . Compared with R^n , the spline R^{n+1} is one polynomial degree higher, has one additional level of continuity, and one more piecewise polynomial segment. The chain property (18), along with compact support, can be used to construct R^{n+1} from R^n . The chain continues forever. The chain continues downward (constructing R^n from R^{n+1}) if it is continuous, has at least two piecewise polynomial segments, and satisfies the property

$$\sum_{a=-\infty}^{\infty} R^n(x + a) = \text{const} \quad \text{for all } x.$$

We only consider symmetrical chains ($R^n(x) = R^n(-x)$). For exposition purposes, we refer to R^n as the parent and R^{n+1} as the child.

2.5. Generalized construction of divergence-free vector fields

Let R^m , S^n , and T^p be splines from arbitrary chains of polynomial degrees m , n , and p . In 2D, the vector field

$$\hat{u}^n(x, y) = \sum_{i,j} u_{i+\frac{1}{2},j} \left(R^{m+1}\left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x}\right) S^n\left(\frac{y - y_j}{\Delta y}\right) + S^{n+1}\left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x}\right) R^m\left(\frac{y - y_j}{\Delta y}\right) \right) \quad (19)$$

$$\hat{v}^n(x, y) = \sum_{i,j} v_{i,j+\frac{1}{2}} \left(S^n\left(\frac{x - x_i}{\Delta x}\right) R^{m+1}\left(\frac{y - y_{j+\frac{1}{2}}}{\Delta y}\right) + R^m\left(\frac{x - x_i}{\Delta x}\right) S^{n+1}\left(\frac{y - y_{j+\frac{1}{2}}}{\Delta y}\right) \right) \quad (20)$$

¹This substitution must be performed with some care. $\frac{d}{dx} B^{n+1}\left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x}\right) = \frac{d}{dx} B^{n+1}\left(\frac{x}{\Delta x} - i - \frac{1}{2}\right) = \frac{1}{\Delta x} \left[B^n\left(\frac{x}{\Delta x} - i\right) - B^n\left(\frac{x}{\Delta x} - i - 1\right) \right] = \frac{1}{\Delta x} \left[B^n\left(\frac{x - x_i}{\Delta x}\right) - B^n\left(\frac{x - x_{i+1}}{\Delta x}\right) \right].$

has polynomial degree $m + n + 1$ and is analytically divergence free whenever the discrete field is discretely divergence free. More generally,

$$\nabla \cdot \hat{\mathbf{u}}^n(\mathbf{x}) = \sum_{i,j,k} d_{i,j,k} \left(R^m \left(\frac{x - x_i}{\Delta x} \right) S^n \left(\frac{y - y_j}{\Delta y} \right) + S^n \left(\frac{x - x_i}{\Delta x} \right) R^m \left(\frac{y - y_j}{\Delta y} \right) \right). \quad (21)$$

The derivation of this property is analogous to that of (17). This analytically divergence-free vector field construction generalizes the form of the interpolation schemes formed from B-splines, which is a special case when $m = n$ and $R^m = 2S^n = B^n$. In 3D, the generalized construction is

$$\begin{aligned} \hat{u}^n(x, y, z) = & \sum_{i,j,k} u_{i+\frac{1}{2},j,k} \left(R^{m+1} \left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x} \right) S^n \left(\frac{y - y_j}{\Delta y} \right) T^p \left(\frac{z - z_k}{\Delta z} \right) + R^{m+1} \left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x} \right) T^p \left(\frac{y - y_j}{\Delta y} \right) S^n \left(\frac{z - z_k}{\Delta z} \right) \right. \\ & + S^{n+1} \left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x} \right) R^m \left(\frac{y - y_j}{\Delta y} \right) T^p \left(\frac{z - z_k}{\Delta z} \right) + S^{n+1} \left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x} \right) T^p \left(\frac{y - y_j}{\Delta y} \right) R^m \left(\frac{z - z_k}{\Delta z} \right) \\ & \left. + T^{p+1} \left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x} \right) R^m \left(\frac{y - y_j}{\Delta y} \right) S^n \left(\frac{z - z_k}{\Delta z} \right) + T^{p+1} \left(\frac{x - x_{i+\frac{1}{2}}}{\Delta x} \right) S^n \left(\frac{y - y_j}{\Delta y} \right) R^m \left(\frac{z - z_k}{\Delta z} \right) \right) \end{aligned} \quad (22)$$

with $\hat{v}^n(x, y, z)$ and $\hat{w}^n(x, y, z)$ defined similarly. The discrete divergence is interpolated similarly to the 2D case (but with six terms), and the vector field is analytically divergence free whenever the discrete divergence is zero. The polynomial degree is $m + n + p + 1$. The special case $n = p$ and $S^n = T^p$ reduces the number of terms from six to three. The special case $m = n = p$ and $R^m = S^n = T^p$ reduces this to a single term, as occurred in the B-spline-based scheme.

Since linear combinations of divergence-free vector fields are also divergence free, additional divergence-free vector fields can be constructed by taking linear combinations of the divergence-free vector fields described above. All analytically divergence-free interpolation schemes we have found that satisfy the properties that this construction provides (piecewise polynomial, compact support, symmetries, and continuity) are generated by this construction, though we do not have a proof that this is always true.

Corrections to divergence-free fields

We can use this freedom provided by the generalized construction to make corrections to the original interpolation scheme in order to obtain additional properties, at the cost of higher polynomial order and computational cost. For any fixed stencil size, maximum polynomial degree, and level of continuity, there are a finite number of degrees of freedom remaining in the coefficients of the underlying polynomials. The resulting space of splines can be spanned by a finite basis of linearly independent splines. We can then construct the child of each spline in the basis, which leads to a finite basis for analytically divergence-free vector fields. If no linear combination of the basis fields satisfies the desired properties, then the search must be repeated with an expanded basis, which can be obtained by allowing a wider stencil size or a higher polynomial degree. We prioritize minimizing the stencil size over minimizing the total polynomial degree.

In particular, we can construct *interpolating* versions of the analytically divergence-free interpolation schemes; these versions agree with the MAC data at the MAC locations. These schemes are presented in Section 2.6 and included in our numerical tests. These schemes are optimal in the sense that no scheme with tighter stencil or smaller polynomial order exists with the same properties as the ones presented, but they are not unique. This was established using brute force in maple; the search procedure we used is described below. The extra freedom was chosen as a compromise between (1) minimizing the L^2 norm of the gradients of the blending functions and (2) selecting schemes that are simple.

2.6. Precise definition of scheme variants

We now present all eight interpolation schemes in a form that is convenient for implementing them. The color-coding of the terms and their properties will be further discussed in the following sections. We describe the schemes in terms of their *blending functions*, which are related to a piecewise polynomial in the following way: Given a piecewise polynomial function $P(x)$ that is nonzero on $-k/2 < x < k/2$, there are k blending functions $P_i(x)$ for $i = 0, 2, \dots, k-1$ such that $P_i(x) = P(x + k/2 - (i + 1))$, $0 \leq x \leq 1$. Essentially, if P is a piecewise polynomial with pieces of length

1 (as is the case with the B-splines, as well as all of the other polynomials we use), the blending functions are each piece of that polynomial shifted to the domain $[0, 1]$.

For the remainder of this section, we assume that the summations are over the u values that are actually included in the stencil (see Fig. 1). If s is the stencil-size in cells (2 for C^0 interpolation, 3 for C^1), then i ranges from 0 to $s + 1$ and j and k range from 0 to s . Using this convention, the i -th u sample can be directly multiplied by the i -th blending function without any shifting necessary.

The four non-interpolating schemes are

$$u^{C^0}(x, y) = \sum_{ij} u_{i,j} B_i^2(x) B_j^1(y) \quad u^{C^0}(x, y, z) = \sum_{ijk} u_{i,j,k} B_i^2(x) B_j^1(y) B_k^1(z) \quad (23)$$

$$u^{C^1}(x, y) = \sum_{ij} u_{i,j} B_i^3(x) B_j^2(y) \quad u^{C^1}(x, y, z) = \sum_{ijk} u_{i,j,k} B_i^3(x) B_j^2(y) B_k^2(z) \quad (24)$$

These schemes use only the B-spline chain B_i^n , whose blending functions are

$$\begin{aligned} B_0^0 &= 1 \\ B_0^1 &= 1 - x & B_1^1 &= x \\ B_0^2 &= \frac{1}{2}(x - 1)^2 & B_1^2 &= -x^2 + x + \frac{1}{2} & B_2^2 &= \frac{1}{2}x^2 \\ B_0^3 &= -\frac{1}{6}(x - 1)^3 & B_1^3 &= \frac{1}{2}x^3 + \frac{2}{3} - x^2 & B_2^3 &= -\frac{1}{2}x^3 + \frac{1}{6} + \frac{1}{2}x^2 + \frac{1}{2}x & B_3^3 &= \frac{1}{6}x^3 \end{aligned}$$

The C^{0i} interpolating schemes are

$$u^{C^{0i}}(x, y) = \sum_{ij} u_{i,j} [B_i^2(x) B_j^1(y) - 4C_i^3(x) D_j^3(y) - 4D_i^4(x) C_j^2(y)] \quad (25)$$

$$u^{C^{0i}}(x, y, z) = \sum_{ijk} u_{i,j,k} [B_i^2(x) B_j^1(y) B_k^1(z) - 4C_i^3(x) (D_j^3(y) C_k^2(z) + D_k^3(z) C_j^2(y)) - 4D_i^4(x) C_j^2(y) C_k^2(z)] \quad (26)$$

Here, we have used two additional chains. The C_i^n chain has blending functions

$$\begin{aligned} C_0^2 &= (x - 1)(3x - 1) & C_1^2 &= x(3x - 2) \\ C_0^3 &= -x(x - 1)^2 & C_1^3 &= -x(x - 1) & C_2^3 &= x^2(x - 1) \\ C_0^4 &= \frac{1}{12}(3x + 1)(x - 1)^3 & C_1^4 &= -\frac{1}{4}x^4 + \frac{1}{6} + x^3 - x^2 & C_2^4 &= -\frac{1}{4}x^4 - \frac{1}{12} + \frac{1}{2}x^2 & C_3^4 &= \frac{1}{12}x^3(3x - 4) \end{aligned}$$

and the D_i^n chain has blending functions

$$\begin{aligned} D_0^2 &= 6x^2 - 6x + 1 \\ D_0^3 &= -x(2x - 1)(x - 1) & D_1^3 &= x(2x - 1)(x - 1) \\ D_0^4 &= \frac{1}{2}x^2(x - 1)^2 & D_1^4 &= -x^2(x - 1)^2 & D_2^4 &= \frac{1}{2}x^2(x - 1)^2 \\ D_0^5 &= -\frac{1}{60}(6x^2 + 3x + 1)(x - 1)^3 & D_1^5 &= \frac{3}{10}x^5 - \frac{1}{30} - \frac{3}{4}x^4 + \frac{1}{2}x^3 & D_2^5 &= -\frac{3}{10}x^5 + \frac{1}{60} + \frac{3}{4}x^4 - \frac{1}{2}x^3 \\ D_3^5 &= \frac{1}{60}x^3(6x^2 - 15x + 10) \end{aligned}$$

The C^{1i} schemes are significantly more complex. Unlike the other schemes, the 2D and 3D versions are quite different. The 2D scheme is

$$u^{C^{1i}}(x, y) = \sum_{ij} u_{i,j} \left[B_i^3(x) B_j^2(y) + \frac{8}{35} (F_i^5(x) C_j^3(y) + C_i^4(x) F_j^4(y)) - 4D_i^5(x) B_j^2(y) - 4B_i^3(x) D_j^4(y) \right] \quad (27)$$

and uses the spline chain F_i^n given by

$$\begin{aligned} F_0^4 &= (x-1)^2(25x^2 - 4x - 3) & F_1^4 &= 130x^4 - 260x^3 + 132x^2 - 2x - 3 & F_2^4 &= x^2(25x^2 - 46x + 18) \\ F_0^5 &= -\frac{1}{2}(2x+1)(5x-1)(x-1)^3 & F_1^5 &= -21x^5 + 1 + \frac{103}{2}x^4 - 34x^3 + 2x^2 \\ F_2^5 &= 21x^5 - \frac{1}{2} - \frac{107}{2}x^4 + 38x^3 - x^2 - 3x & F_3^5 &= \frac{1}{2}x^3(5x-4)(2x-3) \end{aligned}$$

Finally, the 3D version of the scheme is

$$\begin{aligned} u^{cli}(x, y, z) &= \sum_{ijk} u_{i,j,k} \left[B_i^3(x)B_j^2(y)B_k^2(z) + \frac{1}{21} \left(H_i^4(x)C_j^3(y)C_k^3(z) + C_i^4(x)(H_j^3(y)C_k^3(z) + C_j^3(y)H_k^3(z)) \right) \right. \\ &\quad + \frac{1}{7} (G_i^5(x)(D_j^4(y)B_k^2(z) + B_j^2(y)D_k^4(z)) + D_i^5(x)(G_j^4(y)B_k^2(z) + B_j^2(y)G_k^4(z)) \\ &\quad \left. + B_i^3(x)(G_j^4(y)D_k^4(z) + D_j^4(y)G_k^4(z)) \right) \end{aligned} \quad (28)$$

It uses the spline chain $G_i^n = 7B_i^{n-2} + 32C_i^{n-1} - 8F_i^n$

$$\begin{aligned} G_0^4 &= -\frac{5}{2}(x-1)^2(80x^2 - 11) & G_1^4 &= -1040x^4 + 2080x^3 - 1095x^2 + 55x + \frac{55}{2} \\ G_2^4 &= -\frac{5}{2}x^2(80x^2 - 160x + 69) \\ G_0^5 &= \frac{5}{2}(16x^2 + 8x - 1)(x-1)^3 & G_1^5 &= 168x^5 + 2 - 420x^4 + \frac{615}{2}x^3 - 55x^2 \\ G_2^5 &= \frac{5}{2} + \frac{55}{2}x - 168x^5 + 420x^4 - \frac{615}{2}x^3 + \frac{55}{2}x^2 & G_3^5 &= -\frac{5}{2}x^3(16x^2 - 40x + 23) \end{aligned}$$

as well as spline chain $H_i^n = 36B_i^{n-1} + 14C_i^n$

$$\begin{aligned} H_0^3 &= -2(7x-9)(x-1)^2 & H_1^3 &= -50x^2 + 50x + 18 & H_2^3 &= 2x^2(7x+2) \\ H_0^4 &= \frac{1}{6}(21x-29)(x-1)^3 & H_1^4 &= -\frac{7}{2}x^4 + \frac{79}{3} + 32x^3 - 50x^2 & H_2^4 &= -\frac{7}{2}x^4 + \frac{29}{6} - 18x^3 + 25x^2 + 18x \\ H_3^4 &= \frac{1}{6}x^3(21x+8) \end{aligned}$$

2.7. Classifying splines

A somewhat better understanding of the terms that occur in the proposed schemes can be obtained by classifying splines into four types. The first type are the B-splines themselves, which are colored as $B_i^n(x)$ in the presentation of the schemes. These splines have the partition of unity property, which is necessary for the construction of interpolation schemes. They also have the property that they preserve affine data.

All of the interpolation schemes consist of a tensor product B-spline along with tensor product spline terms as corrections. The second type of spline (colored as $C_i^n(x)$) has the property that it is equivalent to interpolating finite differences. For example,

$$C_0^3(x)a_0 + C_1^3(x)a_1 + C_2^3(x)a_2 = -C_0^3(x)(a_1 - a_0) + C_2^3(x)(a_2 - a_1)$$

The third type of spline (colored as $C_i^n(x)$) has the property that it is equivalent to interpolating second differences. For example,

$$C_0^4(x)a_0 + C_1^4(x)a_1 + C_2^4(x)a_2 + C_3^4(x)a_3 = C_0^4(x)(a_2 - 2a_1 + a_0) + C_3^4(x)(a_3 - 2a_2 + a_1)$$

The fourth type of spline (colored as $G_i^4(x)$) has none of these properties and must be combined with other types of splines. The corrections that are applied to the base all have the property that they are zero when applied to affine data. This can be accomplished in two ways: (1) using a spline of the third type (like $D_i^4(x)C_j^2(y)$) or (2) with two splines of the second type (like $C_i^3(x)D_j^3(y)$). These vanish since if $a_{i,j} = m + ni + pj$ for some m, n, p then $a_{i+2,j} - 2a_{i+1,j} + a_{i,j} = 0$ and $a_{i+1,j+1} - a_{i+1,j} - a_{i,j+1} + a_{i,j} = 0$. With the aid of the color scheme, one can readily observe that all of the correction terms vanish for affine data.

Achieving higher order accuracy

It does not appear to be possible to construct a div-free scheme that is more accurate than second order and also satisfies basic properties such as continuity and symmetry. A higher order scheme is possible if continuity is not required [4]. We suspect that this is because a higher-order vector field would lead to a higher-order computation of the divergence everywhere, which contradicts the fact that the divergence is interpolated from a discrete second order accurate finite difference approximation. Although the divergence might in general be one order less accurate than the vector field itself (so that the argument above might not apply to a third-order interpolation scheme), we have not found any schemes more accurate than the second order schemes demonstrated in this paper. We have verified by brute force that no such scheme exists in 2D with a stencil size of at most 8×8 and polynomial degree at most 13.

Details of original construction

We describe here how we constructed the schemes as they are presented. The eight schemes presented were originally constructed and implemented without the benefit of the general construction using brute force in maple. We describe the procedure we followed below. The first step in this process was to construct the solution in unfactored form. First, we fix a stencil size and maximum degree. Then (in 2D) we look for a solution of the form

$$u(x, y) = \sum_{ijrs} u_{i,j} a_{i,j,r,s} x^r y^s,$$

where i, j vary over the MAC x -face indices of the stencil and r, s vary over the permitted polynomial degrees. This leads to a large number of unknowns $a_{i,j,r,s}$. $v(x, y)$ is constructed from $u(x, y)$ by reflection. Next, we enforce that the solution satisfies a number of constraints by enforcing (1) the appropriate level of continuity, (2) symmetry, (3) affine data is reproduced exactly, (4) analytically divergence free subject to discretely divergence free data. The interpolating versions also enforced the constraint that $u(x, y)$ agree with the MAC grid data at the center of the MAC faces. Each of these constraints is linear, so although the system may have thousands of degrees of freedom, they were feasible to solve analytically. In some cases, this leads to families of solutions, where there are a few $a_{i,j,r,s}$ degrees of freedom that could not be eliminated. If no such scheme with the fixed stencil size and polynomial degree exists, the system of equations has no solution.

We started with a tight stencil and low polynomial degree, repeating the procedure above until either a solution is found or the degree was so high that the system became intractable to solve analytically. If no solution is found, we increase the stencil size and repeat the search. For the eight schemes presented, this procedure terminated at the schemes presented. In the case of the search for higher order schemes, no solution was found. This leads to a scheme described by a large number of multivariate polynomials (one per MAC face in the stencil). In this form, the schemes are impractical to implement, and even if implemented would be impractically inefficient to use.

The next step is factoring the stencil into a sum of tensor products. The B-spline schemes factor easily on inspection. The original process that we used for the others was to identify the maximum polynomial degree $x^m y^n$ remaining in the interpolation stencil, create arbitrary splines $P(x)$ and $Q(y)$ of that maximum degree, and solve for the coefficients of these splines to eliminate as many of the highest order polynomial terms as possible. This process is repeated, adding in additional tensor product terms until the entire interpolation stencil is eliminated. This process can be computationally extremely expensive (especially in 3D), since it produces a rather large system of quadratic (2D) or cubic (3D) polynomials in a large number of variables. Where possible, degrees of freedom in the scheme were chosen to reduce the number of tensor product terms required. This worked quite well for the easier schemes (especially C^{0i}). For the hardest scheme (C^{1i}), the best factorization we were able to achieve using this method had 10 terms and used 20 distinct splines whose coefficients were large and irrational.

The general divergence-free construction in 2D was deduced while trying to prove in a simple way that the C^{0i} scheme satisfied the divergence-free property. The 3D form was guessed from the 2D form. We then went back over all of the schemes and instead searched for factorizations in the form of the general construction. Although the procedure is otherwise the same (try to eliminate the highest degree), it is far simpler. The number of arbitrary splines needed to factor C^{1i} is reduced from 30 to just 9, of which only 5 were distinct. (The other 5 splines are their children and are constructed using the chain property.) The system of equations is still nonlinear, but it is much smaller and easier to solve. It also produces simpler solutions.

The schemes presented use six chains: $B_i^1(x)$, $C_i^2(x)$, $D_j^3(y)$, $F_i^4(x)$, $G_i^4(x)$, and H_2^3 . The construction of these chains is not obvious. The B chain is well-known and arises immediately when factoring the non-interpolating schemes. The

C and D chains arise unambiguously when factoring the 2D C^{0i} scheme using the procedure above. The B , C , and D chains suffice to factor the 3D C^{0i} scheme as well, but they are unable to factor the 2D C^{1i} scheme. Factoring this scheme using the general construction while using B , C , or D wherever possible leads to the F chain. The same procedure on the 3D version leads to the G and H chains, both of which are linear combinations of the B , C , and F chains as noted above.

3. Numerical results

In the numerical tests that follow, we use eight vector fields. In 2D, we test using the velocity fields:

$$\begin{aligned}\mathbf{u}_{2a} &= \begin{pmatrix} \sin(370x + 2) \sin(370y + 4) \\ \cos(370x + 2) \cos(370y + 4) \end{pmatrix} \\ \mathbf{u}_{2b} &= \begin{pmatrix} \sin(x + 2) \sin(y + 4) \\ \cos(x + 2) \cos(y + 4) \end{pmatrix} \\ \mathbf{u}_{2c} &= \begin{pmatrix} x^3 - 6xy^2 + y^3 \\ -3x^2y + 2y^3 \end{pmatrix} \\ \mathbf{u}_{2d} &= \begin{pmatrix} \sin(x + 2) + \sin(y + 4) \\ \cos(x + 2) + \cos(y + 4) \end{pmatrix}\end{aligned}$$

In 3D, we test using the velocity fields:

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

Fields $\mathbf{u}_{2a}$ and $\mathbf{u}_{3a}$ are discretely divergence free but under-resolved on the grids used by the tests that employ them. These fields effectively function as pseudorandom but discretely divergence-free fields. Fields $\mathbf{u}_{2b}$ and $\mathbf{u}_{3b}$ are discretely divergence free and well-resolved by the grid. $\mathbf{u}_{2c}$ and $\mathbf{u}_{3c}$ are analytically divergence free but not discretely divergence free. Finally, $\mathbf{u}_{2d}$ and $\mathbf{u}_{3d}$ are not divergence free. Since some of the schemes (especially C^{0i} and C^{1i}) are quite different in 2D and 3D, we perform all of our numerical tests in both dimensions.

3.1. Analytically divergence free

In this example we numerically verify the analytic divergence-free property by comparing (1) the C^0 continuous div-free scheme, (2) the C^1 continuous div-free scheme, (3) the C^{0i} continuous interpolating div-free scheme, (4) the C^{1i} continuous interpolating div-free scheme, (5) the second order Balsara scheme [3], (6) linear interpolation (denoted B^1), (7) cubic spline interpolation (denoted B^3), and (8) cubic interpolation. Schemes (1)-(4) are proposed here. Scheme (5) is a published divergence-free scheme for comparison. Schemes (6)-(8) are commonly-used but non-divergence-free interpolation schemes.

We perform our tests on a 16^D grid over the region $[0, 1]^D$ (for $D = 2, 3$). We seed the domain with one million uniformly random test locations, which are held fixed across all tests, and reject and resample locations that are within a distance of δ from a cell face. We use all eight of the vector fields to contrast the behaviors of the schemes under different conditions.

scheme	$\mathbf{u}_{2a}$	$\mathbf{u}_{2b}$	$\mathbf{u}_{2c}$	$\mathbf{u}_{2d}$	$\mathbf{u}_{3a}$	$\mathbf{u}_{3b}$	$\mathbf{u}_{3c}$	$\mathbf{u}_{3d}$
B^0	$2.23 \cdot 10^1$	$3.10 \cdot 10^{-2}$	$5.57 \cdot 10^{-1}$	$5.56 \cdot 10^{-1}$	$2.92 \cdot 10^1$	$5.10 \cdot 10^{-2}$	$3.05 \cdot 10^{-1}$	2.78
cubic	$1.07 \cdot 10^1$	$2.02 \cdot 10^{-5}$	$9.99 \cdot 10^{-10}$	$5.84 \cdot 10^{-1}$	$1.47 \cdot 10^1$	$3.36 \cdot 10^{-5}$	$2.89 \cdot 10^{-9}$	2.80
B^3	$1.21 \cdot 10^1$	$9.56 \cdot 10^{-4}$	$1.17 \cdot 10^{-2}$	$5.84 \cdot 10^{-1}$	$1.63 \cdot 10^1$	$1.11 \cdot 10^{-3}$	$3.91 \cdot 10^{-3}$	2.80
Balsara	$1.67 \cdot 10^{-9}$	$1.67 \cdot 10^{-10}$	$2.94 \cdot 10^{-3}$	$5.56 \cdot 10^{-1}$	$2.24 \cdot 10^{-9}$	$2.78 \cdot 10^{-10}$	$9.77 \cdot 10^{-4}$	2.78
C^0	$1.56 \cdot 10^{-9}$	$2.01 \cdot 10^{-10}$	$2.93 \cdot 10^{-3}$	$5.84 \cdot 10^{-1}$	$1.88 \cdot 10^{-9}$	$4.02 \cdot 10^{-10}$	$9.77 \cdot 10^{-4}$	2.80
C^{0i}	$2.55 \cdot 10^{-8}$	$2.54 \cdot 10^{-10}$	$2.93 \cdot 10^{-3}$	$5.84 \cdot 10^{-1}$	$4.18 \cdot 10^{-8}$	$5.41 \cdot 10^{-10}$	$9.77 \cdot 10^{-4}$	2.80
C^1	$2.11 \cdot 10^{-9}$	$2.36 \cdot 10^{-10}$	$2.93 \cdot 10^{-3}$	$5.83 \cdot 10^{-1}$	$1.96 \cdot 10^{-9}$	$4.58 \cdot 10^{-10}$	$9.77 \cdot 10^{-4}$	2.80
C^{1i}	$2.66 \cdot 10^{-8}$	$3.40 \cdot 10^{-10}$	$2.93 \cdot 10^{-3}$	$5.84 \cdot 10^{-1}$	$5.90 \cdot 10^{-8}$	$5.97 \cdot 10^{-10}$	$9.77 \cdot 10^{-4}$	2.80

Fig. 2: Maximum divergence E as computed by central differences. The divergence-free interpolation schemes are highlighted (previous work in red, proposed schemes in green). These schemes exhibit small divergence when when interpolating discretely divergence-free vector fields ($\mathbf{u}_{2a}$, $\mathbf{u}_{2b}$, $\mathbf{u}_{3a}$, and $\mathbf{u}_{3b}$).

At each test location (x, y, z) , we compute a second order accurate approximation of the divergence by sampling the interpolated fields $\hat{u}(x, y, z)$, $\hat{v}(x, y, z)$, and $\hat{w}(x, y, z)$ using the central difference stencil

$$E = \frac{\hat{u}(x + \delta, y) - \hat{u}(x - \delta, y) + \hat{v}(x, y + \delta) - \hat{v}(x, y - \delta)}{2\delta} \quad \text{in 2D}$$

$$E = \frac{\hat{u}(x + \delta, y, z) - \hat{u}(x - \delta, y, z) + \hat{v}(x, y + \delta, z) - \hat{v}(x, y - \delta, z) + \hat{w}(x, y, z + \delta) - \hat{w}(x, y, z - \delta)}{2\delta} \quad \text{in 3D}$$

and $\delta = 10^{-6}$, and report the maximum divergence. This gives us two sources of errors: truncation error $O(\delta^2)$ and cancellation $O(\epsilon/\delta)$, where $\epsilon \approx 2 \times 10^{-16}$ is the machine precision. In our tests, the cancellation error dominates, and we may expect an error on the order of 2×10^{-10} . This is not intended to be a refinement study. Rather, we perform this test at a low resolution because we are verifying that the analytic divergence-free property holds regardless of resolution.

Results are shown in Figure 2. The 3D results mirror the 2D results, so we only discuss the 2D results here. The first column shows the results for $\mathbf{u}_{2a}$, which is discretely divergence free but under-resolved. The five divergence-free interpolation schemes (Balsara, C^0 , C^{0i} , C^1 , and C^{1i}) all produce errors on the order of $10^{-9} - 10^{-8}$, which is expected

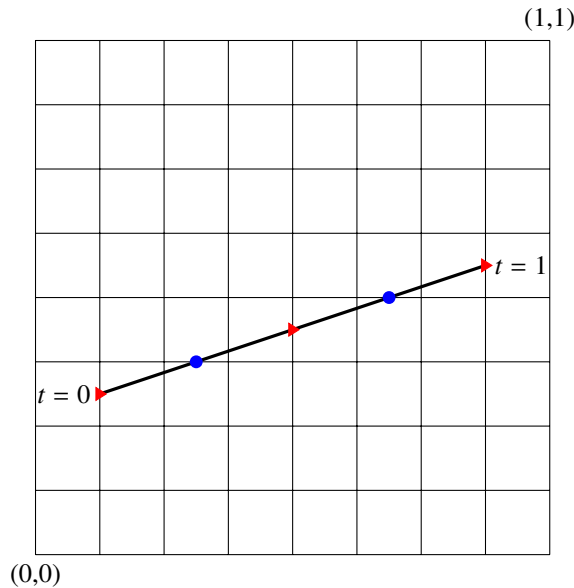


Fig. 3: Schematic for the path continuity test in 2D. The path is a line segment with endpoints at $(\Delta x, 2.5\Delta x)$ and $(7\Delta x, 4.5\Delta x)$, where $\Delta x = \frac{1}{8}$. The path passes through u degrees of freedom (red dot) at $t=0$, $t=\frac{1}{2}$, and $t=1$ and v degrees of freedom (blue dot) at $t=\frac{1}{4}$ and $t=\frac{3}{4}$.

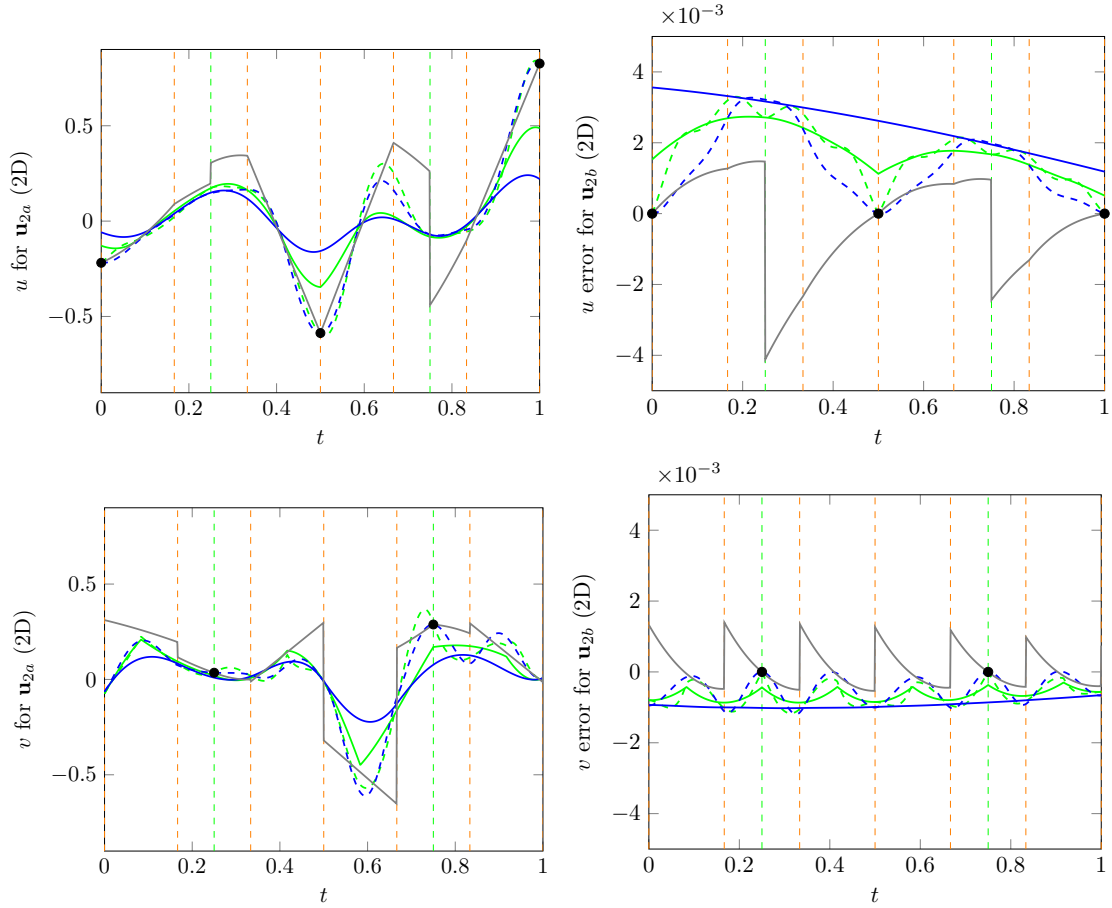


Fig. 4: Path traces for $\mathbf{u}_{2a}$ (first column) and $\mathbf{u}_{2b}$ (second column). Results for u (x component; top row) and v (y component; bottom row) are plotted separately. Both vector fields are discretely divergence free, but the field $\mathbf{u}_{2a}$ is not resolved at the grid resolution 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 C^0 (—), C^{0i} (---), C^1 (—), C^{1i} (---), and Balsara (—). Dashed vertical lines indicate cell crossings across u (---) and v (---) faces. Black dots (•) mark places where the path traces through grid data.

noting that the derivatives that must cancel are themselves on the order of 370. By contrast, the three non-divergence-free interpolation schemes produce divergences in the range 10 – 23. The second column ($\mathbf{u}_{2b}$) is similar, but now the function is resolved on the grid. Since the non-divergence-free interpolation schemes are accurately approximating a divergence-free vector field, their divergences are approximately zero, with the more accurate interpolations producing smaller errors.

The third column ($\mathbf{u}_{2c}$) is a well-resolved divergence free but not discretely divergence-free vector field. It is instructive to see that all of the divergence-free interpolation schemes produce the same error. This is because their residual divergences are actually interpolations of the same discrete divergence (as in (17)). Since this vector field is a cubic polynomial, cubic interpolation exactly recovers the divergence-free vector field, resulting in near-zero divergence. The fourth column ($\mathbf{u}_{2d}$) is a not a divergence-free function. All of the schemes accurately approximate this function and its (nonzero) divergence. Over the domain, the maximum absolute value of divergence is $1 + \cos 2 \approx 0.58385$.

3.2. Path continuity

In this section, we compare the continuity and interpolating properties of the five divergence-free schemes we considered in the previous section. We do this by plotting a cross section of an interpolated vector field along a path. Jumps in these curves reveal violations of C^0 continuity in the interpolation schemes. Kinks in the curves reveal violations of C^1 continuity.

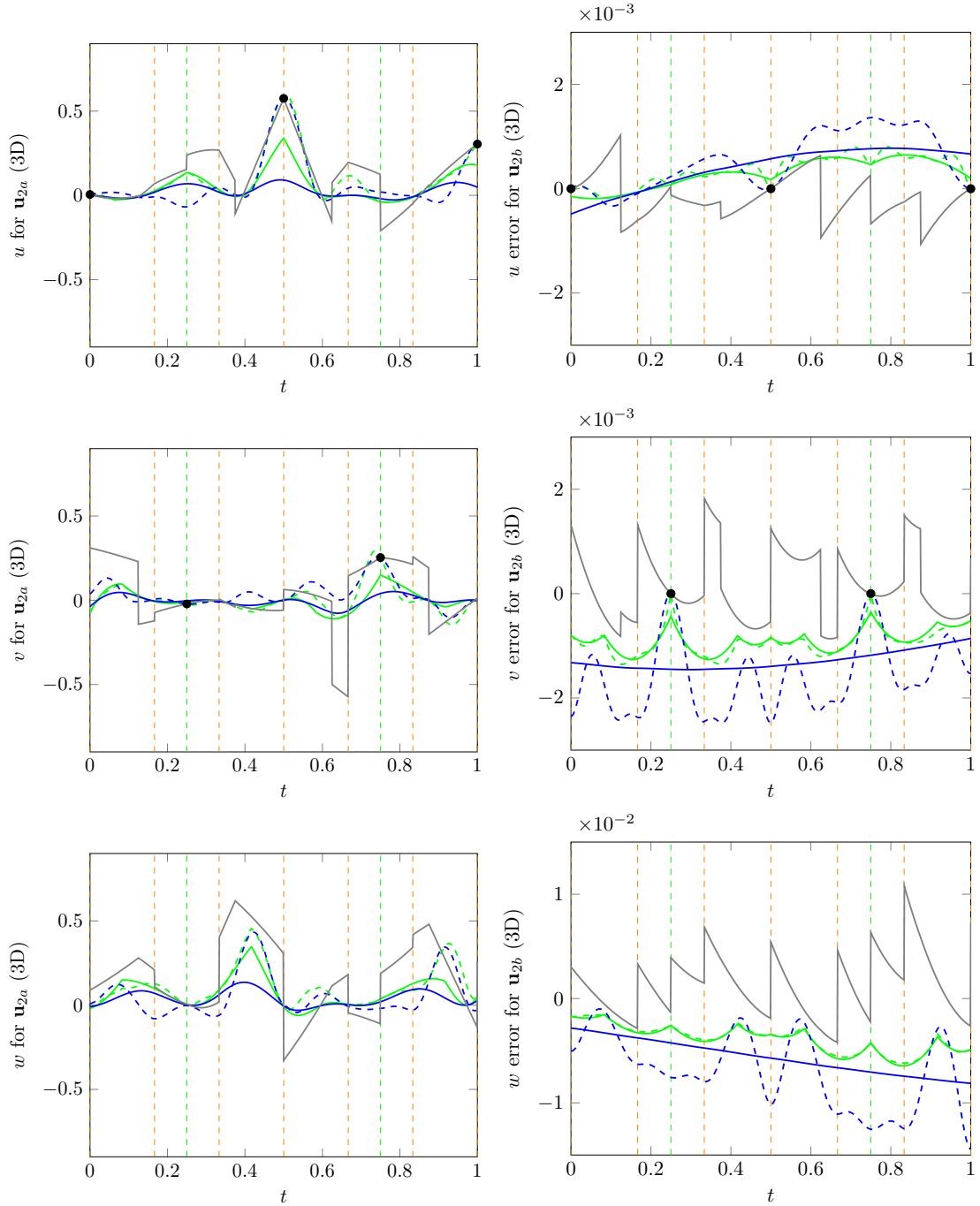


Fig. 5: Path traces for $\mathbf{u}_{3a}$ (first column) and $\mathbf{u}_{3b}$ (second column). Results for u (x component; top row), v (y component; middle row), and w (z component; bottom row) are plotted separately. Both vector fields are discretely divergence free, but the field $\mathbf{u}_{3a}$ 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 C^0 (—), C^0i (---), C^1 (—), C^1i (---), and Balsara (—). Dashed vertical lines indicate cell crossings across u (---) and v (---) faces. Black dots (•) mark places where the path traces through grid data.

We begin in 2D on a 8^2 grid over the region $[0, 1]^2$. A schematic showing the grid layout and path is shown in Figure 3. We use this coarse grid so that the cell crossings and interpolation points can be readily observed in the results. In all cases, enough ghost data is filled for all of the interpolation schemes.

For the first trace, we use the field $\mathbf{u}_{2a}$, which mimics pseudorandom discretely divergence-free data at our resolution. This will tend to emphasize discontinuities and overshoots in the interpolation. The results from this test are shown in Figure 4. From these plots, a few properties of these interpolation schemes can be readily observed. (a) Balsara is discontinuous. The u values are discontinuous at v faces and have kinks at u faces. The v values are discontinuous at u faces and have kinks at v faces. The other schemes are continuous. (b) The schemes C^0 and C^{0i} display kinks. As one would expect, the C^1 and C^{1i} interpolation schemes have continuous derivatives and therefore no kinks. (c) The C^{0i} , C^{1i} , and Balsara schemes interpolate the data; they pass through the black circles in the plots. (d) The C^{0i} and C^{1i} schemes are more oscillatory than the C^0 and C^1 schemes, which reflects the higher degree of the polynomials involved.

For the second trace, we use the field $\mathbf{u}_{2b}$, which is discretely divergence free and resolved at our resolution. This test gives a better idea of how the schemes behave on smooth data. The results from this test are also shown in Figure 4. Since this vector field is accurately interpolated by all of the schemes, the traces overlap, so we instead plot the errors. The same properties (a)–(d) are observed in the errors. In all of the schemes (except C^1), the sensitivity of errors on cell crossings is noticeable. Note that the green curves corresponding to C^0 and C^{0i} exhibit their artifacts halfway between the cell crossings, since they are discretized over the dual nodal grid.

Next, we consider a similar test for the 3D versions of the schemes. The setup is the same as the 2D case, except that now we also have a z direction. We use a 8^3 grid over the region $[0, 1]^3$. The endpoints of the path are at $(\Delta x, 2.5\Delta x, 1.5\Delta x)$ and $(7\Delta x, 4.5\Delta x, 5.5\Delta x)$. As with the 2D case, the path passes through u degrees of freedom at $t = 0$, $t = \frac{1}{2}$, and $t = 1$ and v degrees of freedom at $t = \frac{1}{4}$. The path does not pass through w degrees of freedom.

For the first 3D trace, we use the field $\mathbf{u}_{3a}$. For the second 3D trace, we use the field $\mathbf{u}_{3b}$. The results from these tests are shown in Figure 5. The qualitative properties and relationships between the schemes are similar to the 2D case.

3.3. Convergence order

In this section, we compare the convergence order of the proposed divergence-free interpolation schemes. We use a $[0, 1]^D$ domain and the vector field $\mathbf{u}_{Db}$. As with the divergence-free test, we seed the domain with one million uniformly random test locations, which are held fixed across all tests. We compute the error at each resolution as the maximum error at any of the sample locations. The results are shown in Figure 6. We use a discretely divergence-free vector field for this comparison since the Balsara scheme was constructed under this assumption.

The three non-divergence-free schemes give a good benchmark for accuracy. Cubic interpolation is fourth order accurate, cubic spline interpolation is third order accurate, and multilinear interpolation is second order accurate. All of the divergence-free interpolation schemes tested here are second order accurate. The C^0 and C^1 schemes are very similar to multilinear interpolation in terms of accuracy. Although the C^{0i} and C^{1i} schemes are forced to interpolate the data, they appear to be less accurate than the C^0 and C^1 schemes.

4. Conclusion

We have presented eight schemes that give analytically divergence-free approximating functions for discretely divergence-free data on a MAC grid. The schemes are local and piecewise polynomial. The schemes are C^0 or C^1 continuous, and interpolating or non-interpolating. We have also presented a general construction of continuous, piecewise polynomial divergence-free interpolation schemes. This provides a convenient and concise description and representation of such schemes that is amenable to efficient implementation. This construction also provides a convenient framework for the construction of divergence-free interpolating schemes with desired properties. The schemes presented are demonstrated to be second-order accurate. There are a few limitations to the presented schemes. They are limited to second-order accuracy, which seems to be related to the second-order accurate discrete divergence-free condition. In our implementation, the interpolating versions are notably more computationally expensive, with the 2D C^{0i} , 3D C^{0i} , 2D C^{1i} , and 3D C^{1i} schemes requiring 2.9, 3.2, 6.2, and 7.3 times the computation time of the corresponding non-interpolating versions. We note, however, that the performance could be improved significantly by taking advantage of redundancies in the tensor product terms. The non-interpolating schemes are quite efficient, with the C^0 schemes being only $\sim 17\%$ more expensive than multilinear interpolation and less costly than cubic interpolation.

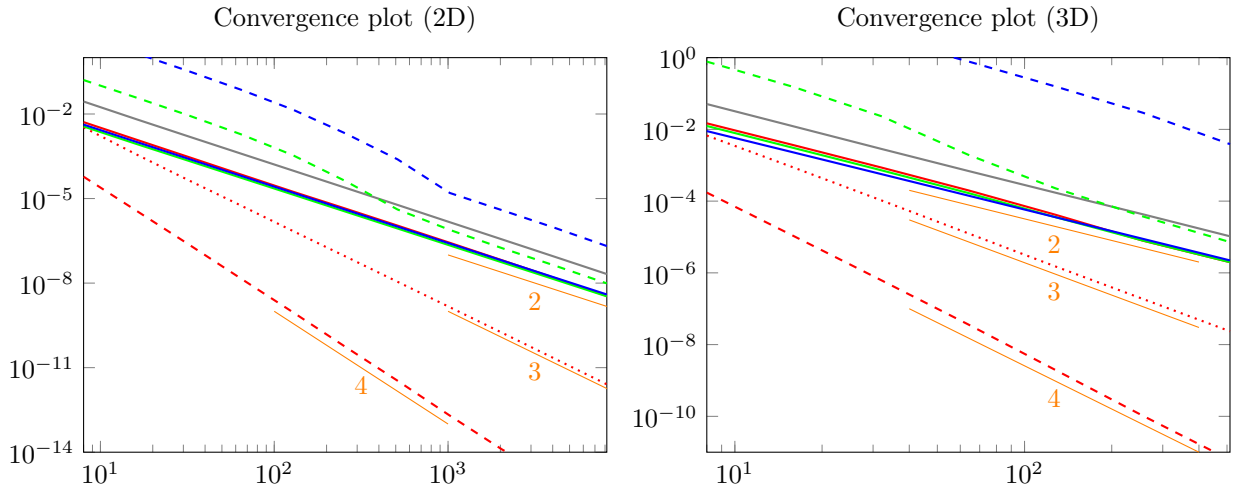


Fig. 6: Convergence test demonstrating the rate of convergence for various methods in 2D (left) and 3D (right). This test compares the convergence rates for multilinear interpolation (—), cubic (---), cubic spline (.....), Balsara (—), C^0 (—), C^{0i} (---), C^1 (—), and C^{1i} (---). The x axis is grid 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 has fourth order accuracy, and cubic spline has third order accuracy. Multilinear interpolation and the divergence-free schemes are all second order accurate. The three schemes shown with solid lines (linear, C^0 , and C^1) have nearly identical accuracy and are partially overlapped in the convergence plots.

5. Acknowledgements

This work was supported in part by National Science Foundation award NSF-2006570.

References

- [1] Luca Amodè 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 Jongsoo Kim. A comparison between divergence-cleaning and staggered-mesh formulations for numerical magnetohydrodynamics. *The Astrophysical Journal*, 602(2):1079, 2004.
- [6] 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.
- [7] 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.
- [8] Jeremiah U Brackbill and Daniel C Barnes. The effect of nonzero $\nabla \cdot \mathbf{B} = 0$ on the numerical solution of the magnetohydrodynamic equations. *Journal of Computational Physics*, 35(3):426–430, 1980.
- [9] Antonio Cervone, Sandro Manservigi, and Ruben Scardovelli. An optimal constrained approach for divergence-free velocity interpolation and multilevel vof method. *Computers & fluids*, 47(1):101–114, 2011.
- [10] Jumyung Chang, Vinicius C Azevedo, and Christopher Batty. Curl-flow: Pointwise incompressible velocity interpolation for grid-based fluids. *arXiv preprint arXiv:2104.00867*, 2021.
- [11] Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.
- [12] Alexandre Joel Chorin, Jerrold E Marsden, and Jerrold E Marsden. *A mathematical introduction to fluid mechanics*, volume 168. Springer, 1990.
- [13] Bernardo Cockburn, Fengyan Li, and Chi-Wang Shu. Locally divergence-free discontinuous galerkin methods for the maxwell equations. *Journal of Computational Physics*, 194(2):588–610, 2004.
- [14] Fabrice Dodu and Christophe Rabut. Irrotational or divergence-free interpolation. *Numerische Mathematik*, 98(3):477–498, 2004.
- [15] 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.
- [16] John M Finn and Luis Chacón. Volume preserving integrators for solenoidal fields on a grid. *Physics of Plasmas*, 12(5):054503, 2005.
- [17] David Handscomb. Local recovery of a solenoidal vector field by an extension of the thin-plate spline technique. *Numerical Algorithms*, 5(2):121–129, 1993.
- [18] DC Handscomb. Spline representation of incompressible flow. *IMA journal of numerical analysis*, 4(4):491–502, 1984.
- [19] DC Handscomb. Interpolation and differentiation of multivariate functions and interpolation of divergence-free vector fields using surface splines. In *Report 91/5*. Citeseer, 1991.
- [20] F. Harlow and E. Welch. Numerical calculation of time dependent viscous flow of fluid with a free surface. *Phys Fluid*, 8(12):2182–2189, 1965.
- [21] Patrick Jenny, Stephen B Pope, Metin Muradoglu, and David A Caughey. A hybrid algorithm for the joint pdf equation of turbulent reactive flows. *Journal of Computational Physics*, 166(2):218–252, 2001.
- [22] Volker John, Alexander Linke, Christian Merdon, Michael Neilan, and Leo G Rebholz. On the divergence constraint in mixed finite element methods for incompressible flows. *SIAM review*, 59(3):492–544, 2017.
- [23] Guido Kanschat. Divergence-free discontinuous galerkin schemes for the stokes equations and the mac scheme. *International journal for numerical methods in fluids*, 56(7):941–950, 2008.
- [24] Philip L Lederer and Sander Rhebergen. A pressure-robust embedded discontinuous galerkin method for the stokes problem by reconstruction operators. *SIAM Journal on Numerical Analysis*, 58(5):2915–2933, 2020.
- [25] Christoph Lehrenfeld and Joachim Schöberl. High order exactly divergence-free hybrid discontinuous galerkin methods for unsteady incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 307:339–361, 2016.
- [26] Fengyan Li and Chi-Wang Shu. Locally divergence-free discontinuous galerkin methods for mhd equations. *Journal of Scientific Computing*, 22(1):413–442, 2005.
- [27] Svenja Lowitzsch. Matrix-valued radial basis functions: stability estimates and applications. *Advances in Computational Mathematics*, 23(3):299–315, 2005.
- [28] F Mackay, R Marchand, and K Kabin. Divergence-free magnetic field interpolation and charged particle trajectory integration. *Journal of Geophysical Research: Space Physics*, 111(A6), 2006.
- [29] Colin P McNally. Divergence-free interpolation of vector fields from point values—exact $\nabla \cdot \mathbf{B} = 0$ in numerical simulations. *Monthly Notices of the Royal Astronomical Society: Letters*, 413(1):L76–L80, 2011.
- [30] DW Meyer and P Jenny. Conservative velocity interpolation for pdf methods. In *PAMM: Proceedings in Applied Mathematics and Mechanics*, volume 4, pages 466–467. Wiley Online Library, 2004.
- [31] Francis J Narowich and Joseph D Ward. Generalized hermite interpolation via matrix-valued conditionally positive definite functions. *Mathematics of Computation*, 63(208):661–687, 1994.
- [32] Suhas V Patankar. *Numerical heat transfer and fluid flow*. CRC press, 2018.
- [33] 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.

- 397 [34] Zachary J Silberman, Thomas R Adams, Joshua A Faber, Zachariah B Etienne, and Ian Ruchlin. Numerical generation of vector potentials
398 from specified magnetic fields. *Journal of Computational Physics*, 379:421–437, 2019.
- 399 [35] Ross Vennell and Rick Beatson. A divergence-free spatial interpolator for large sparse velocity data sets. *Journal of Geophysical Research:*
400 *Oceans*, 114(C10), 2009.
- 401 [36] Hongliang Wang, Roberto Agrusta, and Jeroen van Hunen. Advantages of a conservative velocity interpolation (cvi) scheme for particle-in-
402 cell methods with application in geodynamic modeling. Technical report, Wiley Online Library, 2015.
- 403 [37] Kane Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Transactions on*
404 *antennas and propagation*, 14(3):302–307, 1966.