Visualization and Graphics Group (VISAGG) (2015)
Dept. of Computer Science, University of Calgary

Research Report 2015-1079-12

Component-wise Interpolation of Solenoidal Vector Fields: A Comparative Numerical Study

Dandong Yin[1] and Usman R. Alim[2]

[1]Dept. of Geography and Geographic Information Science, University of Illinois at Urbana-Champaign
[2]Dept. of Computer Science, University of Calgary

Abstract

Vector-field interpolation is a fundamental task in flow simulation and visualization. The common practice is to interpolate the vector field in a component-wise fashion. When the vector field of interest is solenoidal (divergence-free), such an approach is not conservative and gives rise to artificial divergence. In this work, we numerically compare some recently proposed scalar interpolation methods on the Cartesian and body-centered cubic lattices, and investigate their ability to conserve the solenoidal nature of the vector field. We start with a sampled version of a synthetic solenoidal vector field and use an interpolative component-wise reconstruction method to approximate the vector field and its divergence at arbitrary locations. Our results show that an improved scalar interpolation method does not necessarily lead to a more conservative vector field approximation.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Approximating a continuous phenomenon from discrete measurements is a fundamental task in many scientific visualization techniques. A quintessential example is volume visualization, where a sampled scalar dataset is used to approximate the underlying continuous scalar field via an interpolatory model. Of special interest is the case of the Cartesian grid as it lends itself to simple yet efficient visualization methods that can leverage the texture interpolation capabilities of modern Graphics Processing Units (GPUs). Another important example is flow visualization; approximating a vector field from discrete grid-based measurements is at the heart of many flow visualization techniques.

Owing to the popularity and simplicity of scalar interpolation methods, a common practice in flow visualization is to interpolate each velocity component independently via a scalar interpolation method. This straightforward treatment assumes that the components are truly independent. However, many velocity fields encountered in practice do not satisfy this requirement. The simulation of incompressible fluids aims at approximating vector fields that are inherently solenoidal (divergence-free) [Bri08]. Another example in the Helmholtz-Hodge decomposition that splits up a vector field into two parts: one that is divergence-free and one that is curl-free [BNPB13].

Our main focus in this work is the case of solenoidal steady three-dimensional vector fields that are measured on a regular grid. We restrict attention to those types of grids that can be mathematically described as a lattice. The main question we are interested in addressing is as follows: *Does an improved component-wise interpolation technique also lead to an improved preservation of the solenoidal nature of the vector field?*

It should not come as a surprise that a component-wise scalar interpolation of a solenoidal vector field does not guarantee that the resulting interpolating vector field is solenoidal. One strategy to reduce the artificial divergence is to improve the scalar interpolation model without increasing the number of measurements. If each component is better approximated, one would expect the resulting vector field to exhibit reduced divergence. In essence, the goal of this paper is to investigate this claim in detail.

We take a quantitative approach that is reminiscent of the method of manufactured solutions (MMS) [Roa02]. We start with a synthetic solenoidal vector field (the manufactured solution), sample it on a lattice, and approximate it at arbitrary

locations by applying a component-wise scalar interpolation model. By increasing the sampling rate, we can investigate how the divergence of the interpolated vector field diminishes. By considering the scalar interpolation error, we also obtain code verification since the error convergence rate for scalar interpolation is known.

The remainder of the paper is organized as follows. We discuss some related work in Section 2. In Section 3, we describe three fourth-order scalar interpolation methods, one on the Cartesian grid, and two on the body-centered cubic (BCC) grid. These methods represent the state-of-the-art in volume visualization and are described in sufficient detail so as to make our presentation self-contained. Numerical results are presented in Section 4. Section 5 concludes the paper by summarizing key findings and discussing possible avenues for future investigation.

2. Related Work

Trilinear and tricubic interpolation are the defacto standard for scalar data interpolation. These approaches are based on the 1D B-splines and can be efficiently implemented on modern GPUs [PF05, RT12]. In the past decade, a steady stream of work has focused on improving scalar data approximation by switching to non-Cartesian lattices. Owing to its optimality, the BCC lattice has generated some interest, and approximation approaches have been devised that make use of non-separable box-splines [EVM08, Kim13], or variants of the uniform B-splines [CH06, Csé13]. Some of these approaches have also been examined in the context of gradient estimation [AMC10, HAM11]. All of these approaches can be cast into the framework of shift-invariant spaces which offers a convenient way to quantitatively analyze them [BU99, Uns00, Ali12].

MMS has its roots in Computational Fluid Dynamics (CFD) [Roa02]. Since numerical techniques are frequently employed in scientific visualization, a recent body of work has used this method to verify visualization techniques [ESN*09, KUMY10, EJR*14].

The problem of constructing divergence-free approximations of vector-fields has received some attention by researchers in Mathematics and CFD [Han93, Urb01, DP06]. However, within the fields of Computer Graphics and Visualization, we are unaware of any work that makes use of such techniques. A noteworthy example is the work of Lentine *et al*. [LAF11] that proposes a heuristic to conserve mass in fluid simulations.

3. Method

We begin with formulating the scalar interpolation problem in general terms, and then present examples of fourth-order interpolation schemes on the Cartesian cubic (CC) and BCC lattices.

Let $\mathbf{L}$ be the basis matrix of a 3D sampling lattice. Nodes on the sampling lattice are then given by the product $\boldsymbol{n}\mathbf{L}$, where $\boldsymbol{n}$ is an integer row-vector. Let $f[\boldsymbol{n}\mathbf{L}]$ denote the sample values of a function $f(\boldsymbol{x})$. The general interpolation scheme can be written as:

$$f(\boldsymbol{x}) \approx \tilde{f}(\boldsymbol{x}) = \sum_{\boldsymbol{n}\in\mathbf{Z}^3} (f*q)[\boldsymbol{n}\mathbf{L}]\varphi(\boldsymbol{x}-\boldsymbol{n}\mathbf{L}), \qquad (1)$$

where $\varphi(\boldsymbol{x})$ is the reconstruction kernel, and q is the deconvolution prefilter that makes the scheme interpolating and is the inverse of the filter obtained by sampling $\varphi(\boldsymbol{x})$ at the lattice nodes [Ali12, Chapter 1].

Since we are interested in the divergence of the interpolated vector-field, we take the analytic derivative of the reconstruction kernel:

$$\frac{\partial \tilde{f}}{\partial x_i} = \sum_{\boldsymbol{n}\in\mathbf{Z}^3} (f*q)[\boldsymbol{n}\mathbf{L}]\frac{\partial \varphi}{\partial x_i}. \qquad (2)$$

As we treat vector fields in a component-wise fashion, each component of the vector fields is interpolated as a scalar field. Therefore, the divergence of the reconstructed field is:

$$\boldsymbol{\nabla}\cdot\tilde{\boldsymbol{F}} = \frac{\partial \tilde{F}_1}{\partial x_1} + \frac{\partial \tilde{F}_2}{\partial x_2} + \frac{\partial \tilde{F}_3}{\partial x_3}. \qquad (3)$$

Combining (2) and (3), the divergence of reconstructed vector fields can be computed via the partial derivatives of the reconstruction kernel.

3.1. Cubic B-spline Interpolation

For the CC lattice, the corresponding basis $\mathbf{L}$ is the 3×3 identity matrix $\mathbf{I}$. We choose the tricubic B-spline as the fourth-order reconstruction kernel $\varphi(\boldsymbol{x})$. In the spatial domain, the tricubic B-spline $C(\boldsymbol{x})$ is defined as:

$$C(\boldsymbol{x}) = \beta^3(x_1)\cdot\beta^3(x_2)\cdot\beta^3(x_3) \text{ for } \boldsymbol{x} = [x_1,x_2,x_3]^T, \qquad (4)$$

which is the tensor product extension of the 1D uniform cubic B-spline kernel:

$$\beta^3(t) = \begin{cases} \frac{1}{2}|t|^3 - |t|^2 + \frac{2}{3} & \text{if } |t|<1 \\ -\frac{1}{6}|t|^3 + |t|^2 - 2|t| + \frac{4}{3} & \text{if } 1<|t|\le 2 \\ 0 & \text{otherwise} \end{cases}. \qquad (5)$$

The tricubic B-spline $C(\boldsymbol{x})$ has a support size of 64 which means that upto 64 coefficients contribute to the sum in (1) when reconstructing the scalar value at an arbitrary location.

The partial derivatives of $C(\boldsymbol{x})$ are straightforward:

$$\frac{\partial C}{\partial x_i} = \beta^{3\prime}(x_i)\prod_{j\neq i}\beta^3(x_j), \qquad (6)$$

and have the same support as $C(\boldsymbol{x})$.

3.2. Cosine weighted cubic B-Spline interpolation

The BCC lattice is the optimal 3D sampling lattice [EVM08]. The basis **B** for an unnormalized BCC lattice is:

$$\mathbf{B} = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}. \tag{7}$$

Csébfalvi [Csé13] presented a fast and high-quality reconstruction scheme for the BCC lattice by viewing the lattice as two interleaving CC lattices. In this scheme, the reconstruction result is a cosine weighted sum of the interpolation results on the two interleaving CC lattices. Interpolation in this scheme is formulated as follows:

$$\begin{aligned} f(\boldsymbol{x}) \approx \tilde{f}(\boldsymbol{x}) = W(\boldsymbol{x}) \cdot \underbrace{\sum_{\boldsymbol{n}\in\mathbf{Z}^3} (f*q)[\boldsymbol{n}]C(\boldsymbol{x}-\boldsymbol{n})}_{B(\boldsymbol{x})} \\ +(1-W(\boldsymbol{x}))\cdot \underbrace{\sum_{\boldsymbol{n}\in\mathbf{Z}^3} (f*q)[\boldsymbol{n}+\boldsymbol{h}]C(\boldsymbol{x}-\boldsymbol{n}-\boldsymbol{h})}_{B_h(\boldsymbol{x})}, \end{aligned} \tag{8}$$

where $\boldsymbol{h} = [\frac{1}{2},\frac{1}{2},\frac{1}{2}]^T$, $C(\boldsymbol{x})$ is the tricubic B-spline, $B(\boldsymbol{x})$ is the reconstructed function due to the primary CC lattice and $B_h(\boldsymbol{x})$ is the reconstructed function due to the secondary CC lattice. The weight function $W(\boldsymbol{x})$ depends on a scalar parameter λ and is defined as:

$$W(\boldsymbol{x}) = \frac{1}{2} + \lambda\frac{\cos(2\pi x_1)+\cos(2\pi x_2)+\cos(2\pi x_3)}{6}. \tag{9}$$

When $\lambda = 1$, the resulting scheme is interpolating but when $\lambda \neq 0$, it is not and a non-trivial deconvolution prefilter is needed. Due to the distributive property of the partial derivative operator, the partial derivative of each interpolated component of the vector field can be seperately calculated on each CC lattice:

$$\begin{aligned} \frac{\partial \tilde{f}}{\partial x_i}(\boldsymbol{x}) = &\frac{\partial W}{\partial x_i}(\boldsymbol{x})\cdot B(\boldsymbol{x}) + W(\boldsymbol{x})\frac{\partial B}{\partial x_i}(\boldsymbol{x}) \\ &- \frac{\partial W}{\partial x_i}(\boldsymbol{x})\cdot B_h(\boldsymbol{x}) + (1-W(\boldsymbol{x}))\frac{\partial B_h}{\partial x_i}(\boldsymbol{x}). \end{aligned} \tag{10}$$

Thus, the divergence computation routine for the tricubic B-spline can be reused to determine the divergence due to the cosine-weighed scheme. The cosine-weighted scheme (8) can also be written in the general form (1) to yield a reconstruction kernel that has a support size of 128 on the normalized (unit density) BCC lattice [Csé13]. This technique therefore uses twice as many coefficients as the tricubic B-spline scheme for reconstruction at an arbitrary location.

3.3. Quintic Box Spline Interpolation

As is shown by Entezari *et. al* [EVM08], box splines are another efficient instrument for BCC lattice interpolation. The quintic box spline $M(x,y,z)$ is a piecewise quintic polynomial completely supported in a rhombic dodecahedron. Using constants $\alpha := 1/3840$, $\beta := 1/1920$ and $\gamma := 1/960$, the polynomial expression in the positive octant $\{x>0, y>0, z>0\}$ is:

in region $R_1\{x+y<2\}, M(x,y,z) =$

$$\begin{aligned} &\alpha(x+y-4)^3(-3xy-5z^2+2x+2y+20z+x^2+y^2-24) \\ &+\beta(x+z-2)^3(x^2-9x-3xz+10y-5y^2+14+11z+z^2) \\ &+\beta(y+z-2)^3(46-30x-z-y+3zy+5x^2-y^2-z^2) \\ &-\gamma(x+y-2)^3(x^2+x-3xy-5z^2+y^2+y-6), \end{aligned}$$

in region $R_2\{x+y>2, x+z<2\}, M(x,y,z) =$

$$\begin{aligned} &\alpha(x+y-4)^3(-3xy-5z^2+2x+2y+20z+x^2+y^2-24) \\ &-\beta(x+z-2)^3(-z^2-11z+3xz-14+5y^2+9x-10y-x^2) \\ &-\beta(y+z-2)^3(-46+z+30x+y-3zy-5x^2+y^2+z^2), \end{aligned}$$

in Region $R_{3A}\{x+z>2, y+z<2, x-z>2\}, M(x,y,z) =$

$$\alpha(x+y-4)^3(-x^2+8x+3xy-y^2+5z^2-16-12y),$$

in region $R_{3B}\{x+z>2, y+z<2, x-z<2\}, M(x,y,z) =$

$$\begin{aligned} &\alpha(x+y-4)^3(-3xy-5z^2+2x+2y+20z+x^2+y^2-24) \\ &-\beta(y+z-2)^3(30x+z-46-3yz+y-5x^2+y^2+z^2), \end{aligned}$$

and in region $R_4\{y+z>2\}, M(x,y,z) =$

$$\alpha(x+y-4)^3(-3xy-5z^2+2x+2y+20z+x^2+y^2-24).$$

Since the quintic box spline is symmetric, the polynomial pieces in the remaining octants can be obtained via coordinate sign changes of the above polynomials. By differentiating these polynomials, the divergence of quintic box spline interpolated fields can be easily calculated. The quintic box spline is not an interpolating kernel and a non-trivial deconvolution prefiltering step is necessary. This reconstruction scheme is less expensive as compared to the previous two schemes as it only accesses 32 coefficients for reconstruction at an arbitrary location.

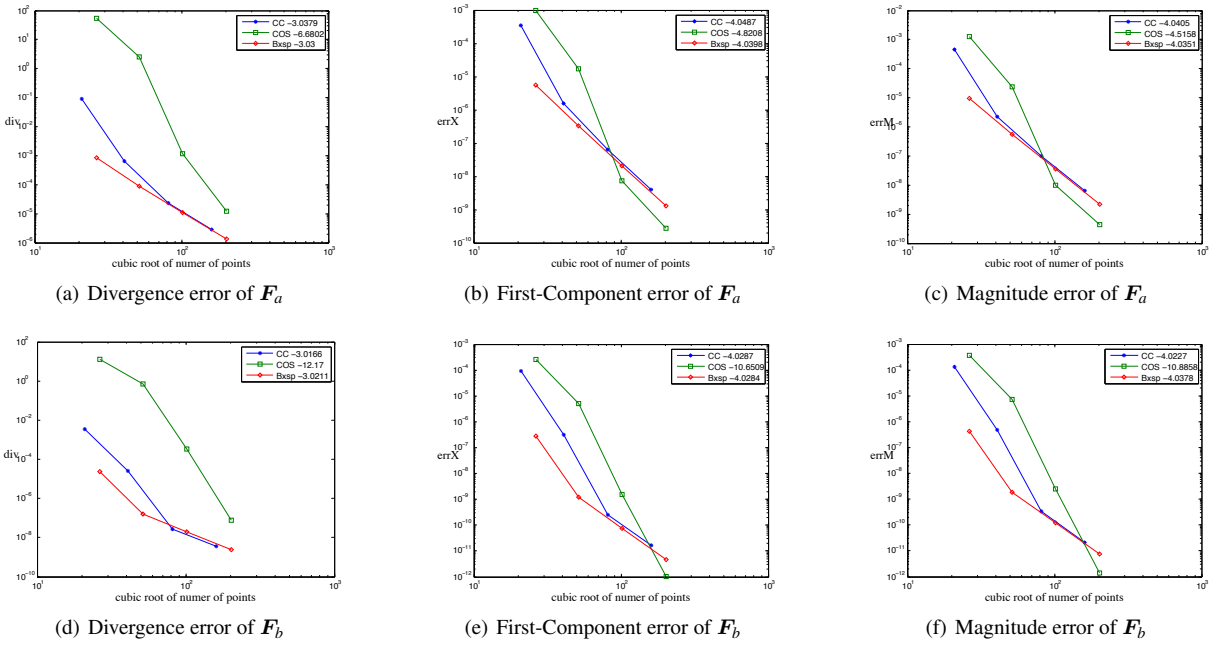
4. Results and Evaluation

We used the following two smooth solenoidal fields to evaluate the interpolation results:

$$\boldsymbol{F}_a = \begin{bmatrix} sin(2\pi x) - 2\pi x\cos(2\pi y) \\ sin(2\pi y) - 2\pi y\cos(2\pi z) \\ sin(2\pi z) - 2\pi z\cos(2\pi x) \end{bmatrix}, \tag{11}$$

and

$$\boldsymbol{F}_b = \begin{bmatrix} -2e^{-x^2-y^2-z^2}(y-z) \\ -2e^{-x^2-y^2-z^2}(z-x) \\ -2e^{-x^2-y^2-z^2}(x-y) \end{bmatrix}. \tag{12}$$



(a) Divergence error of $\boldsymbol{F}_a$ (b) First-Component error of $\boldsymbol{F}_a$ (c) Magnitude error of $\boldsymbol{F}_a$

(d) Divergence error of $\boldsymbol{F}_b$ (e) First-Component error of $\boldsymbol{F}_b$ (f) Magnitude error of $\boldsymbol{F}_b$

Figure 1: *The divergence evaluation of the two solenoidal fields, with their first-component interpolation error and reconstructed magnitude error, relative to the number of sampling points. The grid sizes used are, CC:* $20^3, 40^3, 80^3, 160^3$*, and BCC:* $2\times20^3, 2\times40^3, 2\times80^3, 2\times160^3$*. For each plot, the slope of the line segment between the last two data points is indicated.*

We sampled each component of these fields within the unit cube $[0,1]^3$ for interpolation. The sampling lattice is in accordance with the corresponding interpolation method. When using the cosine weighted scheme on the BCC lattice, we set $\lambda = 0.01$ as this value leads to better results [Csé13]. For each component, the sample values are prefiltered by applying the corresponding deconvolution prefilter. This prefiltering operation is carried out in the Fourier domain by using the tensor product fast Fourier transform (FFT) for the CC lattice, and the BCC FFT [AM09] for the BCC lattice.

We conducted a monte-carlo integration to estimate the L_2 divergence error $\left(\int \boldsymbol{\nabla}\cdot\tilde{\boldsymbol{F}}\,d\boldsymbol{x}\right)^{1/2}$ within the unit cube. This is done by computing the divergence — using the method described in Section 3 — at 100,000 points randomly distributed inside the unit cube. In addition, we also quantified the L_2-error in the reconstruction of the first component of each field as well as the L_2-magnitude-error between the reconstructed vector field and the original field. Both measurements are conducted with similar monte-carlo integrations.

As illustrated in Figure 1, the divergence error curves behave significantly differently from the first-component (scalar field) error curve, which is similar to the magnitude error curve. For $\boldsymbol{F}_a$, the first-component and magnitude curves exhibit a fourth-order convergence as expected. The cosine-weighted scheme converges more slowly as compared to the other two schemes but eventually (as the sampling rate increases) outperforms its competitors. This is in accordance with the results of Csébfalvi [Csé13]. On the other hand, the divergence error curves only exhibit a third-order convergence for the tricubic B-spline and quintic box spline schemes. The cosine-weighted scheme, despite having superior scalar interpolation performance, shows poor divergence performance. For the range of sampling rates shown, the divergence curve has not sufficiently converged indicating that a higher rate is needed to further reduce the divergence. The field $\boldsymbol{F}_b$ exhibits very similar trends for the tricubic B-spline and quintic box spline schemes. Observe that the box spline scheme has a slightly better first-component curve but a slightly worse divergence curve as compared to the tricubic B-spline. For the cosine-weighted scheme, this difference is even more stark: at the highest sampling rate, it shows the best scalar interpolation performance but the worst divergence performance.

In either case, we observe that a better component-wise scalar interpolation method does not guarantee a commensurate reduction in divergence. Our results suggest that the cosine-weighted scheme, while being suitable for scalar interpolation, is not a good fit for component-wise interpolation of solenoidal vector fields.

5. Conclusion

We numerically compared three component-wise interpolation methods in the context of approximating solenoidal vector fields and empirically demonstrated that merely improving the scalar interpolation accuracy of each component is not a reasonable strategy to reduce artificial divergence. In future, we plan to carry out a more thorough divergence error analysis of component-wise schemes so as to investigate alternate component-wise approximation schemes that relax the interpolation requirement in order to reduce artificial divergence. We also plan to study the impact of these improved schemes on flow visualization and simulation techniques.

Acknowledgements

The first author would like to acknowledge the MITACS Globlink program for providing funding for this research.

References

[Ali12] ALIM U. R.: *Data Processing on the Body-Centered Cubic Lattice*. PhD thesis, Simon Fraser University, Burnaby, BC, Canada, July 2012. 2

[AM09] ALIM U. R., MÖLLER T.: A fast Fourier transform with rectangular output on the BCC and FCC lattices. In *Proceedings of the Eighth International Conference on Sampling Theory and Applications (SampTA'09)* (Marseille, France, May 2009). 4

[AMC10] ALIM U. R., MÖLLER T., CONDAT L.: Gradient estimation revitalized. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2010) 16*, 6 (Nov. 2010), 1494–1503. 2

[BNPB13] BHATIA H., NORGARD G., PASCUCCI V., BREMER P.-T.: The helmholtz-hodge decomposition — a survey. *IEEE Transactions on Visualization and Computer Graphics 19*, 8 (Aug 2013), 1386–1404. 1

[Bri08] BRIDSON R.: *Fluid Simulation For Computer Graphics*. Ak Peters Series. A. K. Peters, 2008. 1

[BU99] BLU T., UNSER M.: Quantitative Fourier analysis of approximation techniques: Part I—Interpolators and projectors. *IEEE Transactions on Signal Processing 47*, 10 (Oct. 1999), 2783–2795. 2

[CH06] CSÉBFALVI B., HADWIGER M.: Prefiltered B-spline reconstruction for hardware-accelerated rendering of optimally sampled volumetric data. In *Proceedings of Vision, Modeling, and Visualization* (2006), pp. 325–332. 2

[Csé13] CSÉBFALVI B.: Cosine-weighted b-spline interpolation: A fast and high-quality reconstruction scheme for the body-centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics 19*, 9 (Sept. 2013), 1455–1466. 2, 3, 4

[DP06] DERIAZ E., PERRIER V.: Divergence-free and curl-free wavelets in two dimensions and three dimensions: application to turbulent flows. *Journal of Turbulence*, 7 (2006). 2

[EJR*14] ETIENE T., JONSSON D., ROPINSKI T., SCHEIDEGGER C., COMBA J. L., NONATO L. G., KIRBY R. M., YNNERMAN A., SILVA C. T.: Verifying volume rendering using discretization error analysis. *IEEE Transactions on Visualization and Computer Graphics 20*, 1 (2014), 140–154. 2

[ESN*09] ETIENE T., SCHEIDEGGER C., NONATO L. G., KIRBY R. M., SILVA C. T.: Verifiable visualization for isosurface extraction. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (2009), 1227–1234. 2

[EVM08] ENTEZARI A., VILLE D. V. D., MÖLLER T.: Practical box splines for reconstruction on the body centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics 14*, 2 (2008), 313–328. 2, 3

[HAM11] HOSSAIN Z., ALIM U. R., MÖLLER T.: Toward high quality gradient estimation on regular lattices. *IEEE Transactions on Visualization and Computer Graphics 17*, 4 (Apr. 2011), 426–439. 2

[Han93] HANDSCOMB D.: Local recovery of a solenoidal vector field by an extension of the thin-plate spline technique. *Numerical Algorithms 5*, 2 (1993), 121–129. 2

[Kim13] KIM M.: Quartic box-spline reconstruction on the BCC lattice. *Visualization and Computer Graphics, IEEE Transactions on 19*, 2 (2013), 319–330. 2

[KUMY10] KRONANDER J., UNGER J., MÖLLER T., YNNERMAN A.: Estimation and modeling of actual numerical errors in volume rendering. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 893–902. 2

[LAF11] LENTINE M., AANJANEYA M., FEDKIW R.: Mass and momentum conservation for fluid simulation. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aug. 2011), pp. 91–100. 2

[PF05] PHARR M., FERNANDO R.: *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation (Gpu Gems)*. Addison-Wesley Professional, 2005. 2

[Roa02] ROACHE P. J.: Code verification by the method of manufactured solutions. *Journal of Fluids Engineering 124*, 1 (2002), 4–10. 1, 2

[RT12] RUIJTERS D., THÉVENAZ P.: GPU prefilter for accurate cubic B-spline interpolation. *The Computer Journal 55*, 1 (2012), 15–20. 2

[Uns00] UNSER M.: Sampling-50 years after Shannon. *Proceedings of the IEEE 88*, 4 (2000), 569–587. 2

[Urb01] URBAN K.: Wavelet bases in h(div) and h(curl). *Mathematics of Computation 70*, 234 (2001), pp. 739–766. 2