INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2005; **63**:455–471
Published online 3 March 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1296

Tricubic interpolation in three dimensions

F. Lekien*, †, 1, 2 and J. Marsden[2]

[1] *Mechanical and Aerospace Engineering, Princeton University, U.S.A.*
[2] *Control and Dynamical Systems, California Institute of Technology, U.S.A.*

SUMMARY

The purpose of this paper is to give a local tricubic interpolation scheme in three dimensions that is both C^1 and isotropic. The algorithm is based on a specific 64×64 matrix that gives the relationship between the derivatives at the corners of the elements and the coefficients of the tricubic interpolant for this element. In contrast with global interpolation where the interpolated function usually depends on the whole data set, our tricubic local interpolation only uses data in a neighbourhood of an element. We show that the resulting interpolated function and its three first derivatives are continuous if one uses cubic interpolants. The implementation of the interpolator can be downloaded as a static and dynamic library for most platforms. The major difference between this work and current local interpolation schemes is that we do not separate the problem into three one-dimensional problems. This allows for a much easier and accurate computation of higher derivatives of the extrapolated field. Applications to the computation of Lagrangian coherent structures in ocean data are briefly discussed. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: tricubic; interpolation; computational dynamics

1. INTRODUCTION

1.1. Motivation from ocean dynamics

There has been considerable interest in using observational and model data available in coastal regions to compute Lagrangian structures such as barriers to transport and alleyways in the flow. As an example, Figure 1 shows the Lyapunov exponent field computed using high-frequency radar data collected in the bay of Monterey, along the California shoreline. Red denotes zones of higher stretching in the sense of Reference [1]. The bright red lines in Figure 1 define a boundary between the open ocean and a re-circulating area inside the bay. These

*Correspondence to: F. Lekien, E-Quad J220, Princeton University, Princeton, NJ 08544, U.S.A.
†E-mail: lekien@princeton.edu

Contract/grant sponsor: Office of Naval Research; contract/grant number: N00014-01-1-0208
Contract/grant sponsor: Office of Naval Research; contract/grant number: N00014-02-1-0826

Received 13 May 2004
Revised 23 August 2004
Accepted 13 December 2004

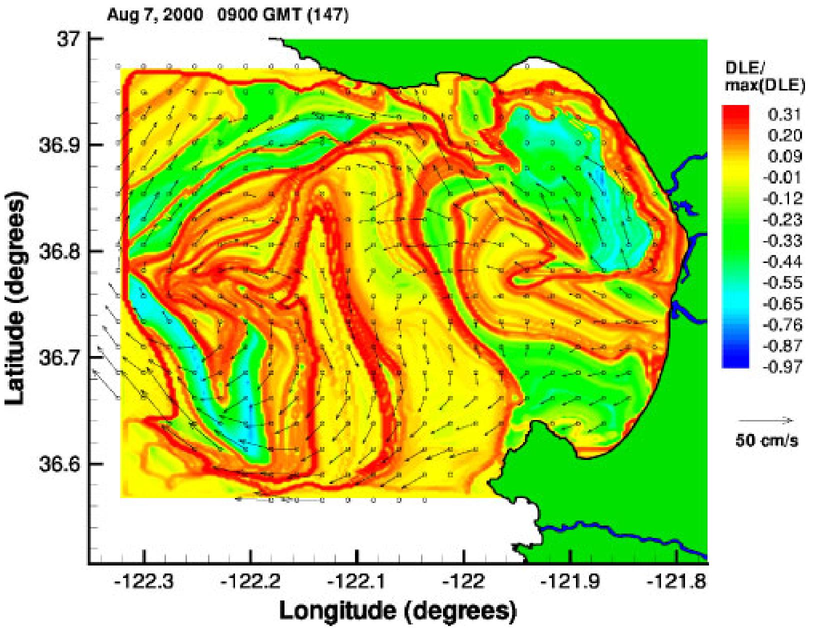

Figure 1. Lagrangian coherent structures computed using high-frequency radar data collected in Monterey Bay, CA. The bright red lines are those moving manifolds that experience the maximum stretching in time. They define a barrier between the bay area and the open ocean. Superimposed on this figure are the instantaneous velocity vectors measured by the radar. Interpolation of the velocity between the grid points is required to integrate trajectories and obtain the Lagrangian structures.

Lagrangian structures are of major interest for studying transport and mixing in coastal areas, as in, for example, References [1–3]. These techniques require the integration of an ODE of the form

$$\dot{\mathbf{x}} = \mathbf{v}(\mathbf{x}, t) \tag{1}$$

where the velocity field $\mathbf{v}(\mathbf{x}, t)$ is a sufficiently smooth function of space and time. However, in practice, the velocity field is usually given as a discrete set of measured vectors (see Figure 1 for example) or the numerical output of a large-scale ocean simulation. The velocity field then used in Equation (1) is generally chosen to be an interpolation of the given discrete data set.

The smoothness of the trajectories and of the extracted Lagrangian coherent structures is directly derived from the smoothness of the velocity field [4]. Linear interpolation of the data does not produce smooth Lagrangian structures and hence, there is a need for a smoother interpolation method. This paper develops such a method and was motivated by the need for data interpolation in ocean dynamics (see Reference [3]). Notice that many other works already use tricubic interpolation (see References [5–7] for examples in image processing and chemistry) and have shown its superiority. However, the tricubic interpolation is always split in three one-dimensional problems (see References [8, 9], for example). In this paper we propose an intrinsically three-dimensional approach and show that it is equivalent to a particular combination of three one-dimensional cubic interpolants. The advantages of our method is the reduced computational cost and the availability of accurate derivatives (first and higher orders) of the function interpolated.

1.2. Goals of the paper

This paper focuses on the cases of time-dependent two-dimensional flows and time-independent three-dimensional flows. In both cases, there are three directions of interpolation, namely (x, y, t) or (x, y, z), respectively. For time-dependent two-dimensional flows, the usual interpolation method is a bicubic interpolation in space combined with a one-dimensional interpolation in time, such as a Lagrange polynomial [10]. This provides C^1 continuity in space and C^1 continuity in time. C^1 continuity in space–time can be achieved with such a method provided that the derivatives with respect to time are also interpolated with the bicubic interpolator and used directly in the one-dimensional interpolator afterward. However, the derivatives of the function are not easily accessible in such schemes. One can use finite differences to recover the derivatives but the global tricubic interpolator provides a more efficient and accurate way to compute them.

The purpose of this paper is to derive the equations for a full C^1 interpolation method for functions with three variables. We will show that a tricubic representation is the minimum order that is necessary to maintain global C^1 continuity and we will derive the equations for the local representation.

2. MINIMUM REQUIREMENTS FOR C^1 INTERPOLATION

We assume that a function f is given at the corner of a regular mesh. Without loss of generality, we assume that the element is a cube of side 1, as in Figure 2. For an element with arbitrary size, each variable can be scaled to obtain the generic element described in Figure 2. Accordingly, derivatives used in this section must be appropriately scaled. As a result, the method presented works for meshes where the elements are rectangular parallelepipeds, not necessarily of equal size.

We represent the function f as a piecewise polynomial. Inside each cube, f takes the values given by an expression of the following form:

$$f(x,y,z) = \sum_{i,j,k=0}^{N} a_{ijk} x^i y^j z^k \tag{2}$$

where the N^3 coefficients a_{ijk} must be determined based on the data and the desired degree of smoothness. The resulting function f, of course, is C^∞ inside each element. The overall smoothness of f depends on the properties of f and its derivative on the faces of the elements. One achieves C^1 continuity if and only if f and its three first derivatives are continuous on each of the six faces of the cubes. An obvious necessary condition is the continuity of f and the first derivatives at each of the eight corners $p_1, p_2, \ldots, p_8$ of the cube. The value of f is assumed to be given at these corners. We aim towards local interpolation, so we also assume

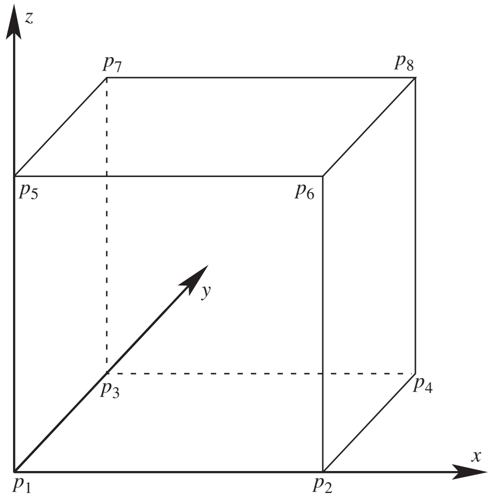

Figure 2. Element for interpolation in three dimensions.

that we know the value of the first derivatives at the eight corners.[‡] A finite difference method can be used to extract an approximation of the derivatives of f based on nearby values of f.

In this situation, the necessary condition for C^1 continuity results in four constraints at the eight corners of each element, giving 32 constraints. An examination of Equation (2) shows that N must be greater or equal to 3 to keep the number of coefficients above 32. The resulting tricubic form of f on the elements is

$$f(x,y,z) = \sum_{i,j,k=0}^{3} a_{ijk} x^i y^j z^k \tag{3}$$

We will derive the equations for the coefficients a_{ijk} in such a way that they satisfy the necessary condition above (C^1 continuity at the eight corners) and then the strategy is to show that this condition is also sufficient that is, the corner conditions automatically imply C^1 continuity of f on each face.

3. ADDITIONAL CONSTRAINTS

The tricubic interpolation form given by Equation (3) uses 64 coefficients and enforcing C^1 continuity at the eight corners only provides 32 constraints. Notice that there is not a unique choice of extra constraints. We will show in Section 5 that the first 32 conditions also guarantee

[‡] In comparison, global interpolation only requires the continuity of the derivatives at the faces of the elements. This produces multi-element constraints and requires the simultaneous computation of the interpolants for all squares. In this paper, we assume that a fair approximation of the first derivatives at the corner are available, so the interpolants can be computed for each element independently. This choice is mainly dictated by the structure of the data. Footprints of geophysical flows are usually sparse and the error may increase greatly at the edges of the domain. Computing the interpolant only with data collected at the corner of an element avoids the propagation of high measurement errors from the edges. In addition, the boundary conditions are not always known, making it impossible to derive a full system of equation for global splines.

C^1 continuity. Therefore, there are many tricubic interpolants of the form given by Equation (3) satisfying C^1 continuity. Fortunately, by adding two basic requirements on what the remaining 32 constraints can be, the solution becomes unique. First, we have chosen to *favour smoothness* over accuracy. Our objective in designing a C^1 tricubic interpolator is to increase the smoothness of the resulting interpolated function, not minimizing the error between the actual (unknown) field and the interpolated function. This naturally suggests to use the smoothness of additional derivatives as the remaining constraints. Second, we require that the choice of extra constraints be *isotropic*. In other words, we require that the set of extra constraints is invariant under rotations of the axis x, y and z. There are only two sets of four derivatives satisfying this condition. We will show that the set

$$\left\{\frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial y^2}, \frac{\partial^2 f}{\partial z^2}, \frac{\partial^3 f}{\partial x \partial y \partial z}\right\} \tag{4}$$

is linearly dependent with the first 32 constraints. This choice leads to a linearly dependent set of constraints for the coefficients a_{ijk}, as is shown in the next section. As a result, a tricubic interpolant satisfying prescribed values of these functions at the eight corners does not usually exist and this choice must be discarded. The only isotropic choice (i.e. invariant under axis rotation) that satisfies our two requirements above and guarantees existence and uniqueness of the solution is

$$\left\{\frac{\partial^2 f}{\partial x \partial y}, \frac{\partial^2 f}{\partial x \partial z}, \frac{\partial^2 f}{\partial y \partial z}, \frac{\partial^3 f}{\partial x \partial y \partial z}\right\} \tag{5}$$

Therefore, we adopt the strategy that the coefficients a_{ijk} will be determined for each element in such a way that the values of each function in the set

$$\left\{f, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}, \frac{\partial^2 f}{\partial x \partial y}, \frac{\partial^2 f}{\partial x \partial z}, \frac{\partial^2 f}{\partial y \partial z}, \frac{\partial^3 f}{\partial x \partial y \partial z}\right\} \tag{6}$$

take prescribed values at the eight corners of the element.

One might wonder why the value of f, its first derivatives and the four functions in the set given in Equation (4) at the eight corners are not linearly independent variables. It is not too surprising because f is a tricubic function as defined by Equation (3). The set of tricubic functions is much smaller than C^1 and we expect some relationships between certain derivatives arising from their particular form.

In this particular case, one can show that, if f and its first derivatives are prescribed at the eight corners, the values of $\partial^2 f/\partial x^2$, $\partial^2 f/\partial y^2$, $\partial^2 f/\partial z^2$ are always fixed at each corner and cannot be used as extra constraints. The explanation for this fact lies in the fact that along one axis (i.e. setting the two other variables to zero), the tricubic form given in Equation (3) reduces to a cubic spline. The values of f and its first derivative along that axis at two points along that axis determine a unique spline and the second derivative along that axis is already constrained by the cubic spline. As an example, we derive the relationship for $\partial^2 f/\partial x^2$. Along the x-axis, we have $y = z = 0$ and Equation (3) reduces to

$$f(x, 0, 0) = \sum_{i=0}^{3} a_{i00} x^i \tag{7}$$

To achieve C^1 continuity, we have prescribed the value of f and its first derivatives at the eight corners of the element shown in Figure 2. In particular, f and $\partial f/\partial x$ are given at the points $p_1=(0,0,0)$ and $p_2=(1,0,0)$. According to Reference [10], for example, this means that we must have

$$\begin{aligned} a_{000} &= f|_{p_1} \\ a_{100} &= \left.\frac{\partial f}{\partial x}\right|_{p_1} \\ a_{200} &= 3f|_{p_2} - 3f|_{p_1} - 2\left.\frac{\partial f}{\partial x}\right|_{p_1} - \left.\frac{\partial f}{\partial x}\right|_{p_2} \\ a_{300} &= \left.\frac{\partial f}{\partial x}\right|_{p_2} + \left.\frac{\partial f}{\partial x}\right|_{p_1} - 2f|_{p_2} + 2f|_{p_1} \end{aligned} \tag{8}$$

The values of the second derivative in the x direction at the point p_1 are given by

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{p_1} = 2a_{200} = 6f|_{p_2} - 6f|_{p_1} - 4\left.\frac{\partial f}{\partial x}\right|_{p_1} - 2\left.\frac{\partial f}{\partial x}\right|_{p_2} \tag{9}$$

and are therefore not independent of the values of f and $\partial f/\partial x$ at p_1 and p_2. Adding $\partial^2 f/\partial x^2$ in the set of constraints will always create linearly dependent constraints and a generally non-existent solution for the interpolant. Notice that this result is valid for any tricubic interpolant in any dimension. For example, tricubic interpolation in six dimensions will require many extra constraints in addition to C^1 continuity at the corners. In no case should derivatives such as $\partial^2 f/\partial x^2$, $\partial^3 f/\partial x^2\partial y$ be added to the set of constraints. An important corollary is the following: In any dimension, no local tricubic interpolant can guarantee C^2 continuity. In n dimensions, each element has 2^n corners and there are n first-order derivatives; so maintaining C^1 continuity only gives $2^n(n+1)$ constraints. On the other hand, the cubic spline in an element is given by a four-dimensional tensor[§] of order n. This gives 2^{2n} coefficients to determine. The number of extra constraints is $2^n(2^n-n-1)$ or 2^n-n-1 per corner. Since there are exactly $\frac{1}{2}n(n+1)$ second derivatives of f in n dimensions, for $n\geqslant 4$, it is potentially possible to try to add all the second derivatives in the extra constraints and try to make the interpolated function C^2. However, the result above shows that this will not be a valid choice because $\partial^2 f/\partial x^2$ will be generically discontinuous at the corners and faces of the elements. To achieve C^2 continuity, one must use higher order polynomials.

4. INTERPOLATOR EQUATIONS

We stack the 64 coefficients a_{ijk} of the interpolant in Equation (3) in a vector $\bar{\alpha}$ by defining

$$\alpha_{1+i+4j+16k} = a_{ijk} \quad \text{for all } i,j,k\in\{0,1,2,3\} \tag{10}$$

[§] In n dimensions, we consider the coefficients of the tricubic interpolation $a_{i_1 i_2\cdots i_n}$ as a tensor of order n. The n four-dimensional vectors $(1, x_i, x_i^2, x_i^3)$ for $i=1,\ldots,n$ are applied to this tensor to get a scalar value, representing the value of the function at this point.

Similarly, we give a unique index $p_1, p_2, \ldots, p_8$ to each corner of the cube, as in Figure 2. We stack the constraints on f and its derivatives in a vector $\bar{b}$ by defining

$$b_i = \begin{cases} f(p_i) & \text{if } 1 \leqslant i \leqslant 8 \\ \dfrac{\partial f}{\partial x}(p_{i-8}) & \text{if } 9 \leqslant i \leqslant 16 \\ \dfrac{\partial f}{\partial y}(p_{i-16}) & \text{if } 17 \leqslant i \leqslant 24 \\ \dfrac{\partial f}{\partial z}(p_{i-24}) & \text{if } 25 \leqslant i \leqslant 32 \\ \dfrac{\partial^2 f}{\partial x \partial y}(p_{i-32}) & \text{if } 33 \leqslant i \leqslant 40 \\ \dfrac{\partial^2 f}{\partial x \partial z}(p_{i-40}) & \text{if } 41 \leqslant i \leqslant 48 \\ \dfrac{\partial^2 f}{\partial y \partial z}(p_{i-48}) & \text{if } 49 \leqslant i \leqslant 56 \\ \dfrac{\partial^3 f}{\partial x \partial y \partial z}(p_{i-56}) & \text{if } 57 \leqslant i \leqslant 64 \end{cases} \tag{11}$$

Based on Equation (3), the derivatives of f can be computed and evaluated for the eight points p_i. This gives a linear system in the 64 unknown coefficients α_i of the form

$$B\bar{\alpha} = \bar{b} \tag{12}$$

The 64×64 matrix B is readily found computationally. Its elements are integer numbers and are computed exactly (i.e. without numerical error). Because of its large size, we do not display the matrix in this paper but they are explicitly available in different forms (HTML, C-code) at `http://gyre.cds.caltech.edu/pub/software/tricubic/doc`

The determinant of the matrix B is

$$\det(B) = 1 \tag{13}$$

As a result, the matrix B is invertible and the coefficients α_i can be computed using the linear relationship

$$\bar{\alpha} = B^{-1}\bar{b} \tag{14}$$

The 64×64 matrix B^{-1} is the core of the tricubic interpolator and can be explicitly found at `http://gyre.cds.caltech.edu/pub/software/tricubic/doc`

Remark

Adding the isotropic set of derivatives in the set given by Equation (4) to the conditions determining f and its first derivatives, gives another matrix B'. However, we have shown that constraining $\partial^2 f/\partial x^2$ is not possible because its value at the corners is not independent of the values of f and $\partial f/\partial x$. As a result, the choice of derivatives given by Equation (4) leads to $\det(B')=0$ and therefore B' cannot be used to determine a valid set of coefficients a_{ijk} for the interpolator.

5. PROOF OF C^1 CONTINUITY

In this section, we show that the function f, defined piecewise by Equation (3), is actually C^1 continuous. Inside each element, f is polynomial and therefore C^∞. The smoothness of f depends on its continuity and the continuity of its derivatives at the boundary of each element. As a result, proving that f is C^1 is equivalent to showing that f is continuous on each face of the elements and that the 3 first derivatives are also continuous on these faces. The coefficients $a_{ijk}=\alpha_{1+i+4j+16k}$ defined by Equation (14) are designed in such a way that the values of the eight functions in the set given in Equation (6) take prescribed values at the eight corners of the element. We will show that this implies that the eight functions of the set given in Equation (6) are also continuous on the six faces of the element.

Lemma 5.1
The interpolated function f is continuous on the horizontal faces of the elements.

Proof
We investigate the values of the function f on the upper face of an element and the bottom face of the element right above it (see Figure 3). In the lower element, f can be written as

$$f^{\mathrm{L}}(x,y,z)=\sum_{i,j,k=0}^{N} a^{\mathrm{L}}_{ijk}x^i y^j z^k \tag{15}$$

and its value on the common face is the following bicubic polynomial in x and y:

$$f^{\mathrm{L}}(x,y,1)=\sum_{i,j=0}^{N}\left(\sum_{k=0}^{N} a^{\mathrm{L}}_{ijk}\right)x^i y^j \tag{16}$$

Define

$$b^{\mathrm{L}}_{ij}=\sum_{k=0}^{N} a^{\mathrm{L}}_{ijk} \tag{17}$$

so that Equation (16) becomes

$$f^{\mathrm{L}}(x,y,1)=\sum_{i,j=0}^{N} b^{\mathrm{L}}_{ij}x^i y^j \tag{18}$$

Similarly, for the upper cube, we have

$$f^{\mathrm{U}}(x,y,z)=\sum_{i,j,k=0}^{N} a^{\mathrm{L}}_{ijk}x^i y^j z^k \tag{19}$$

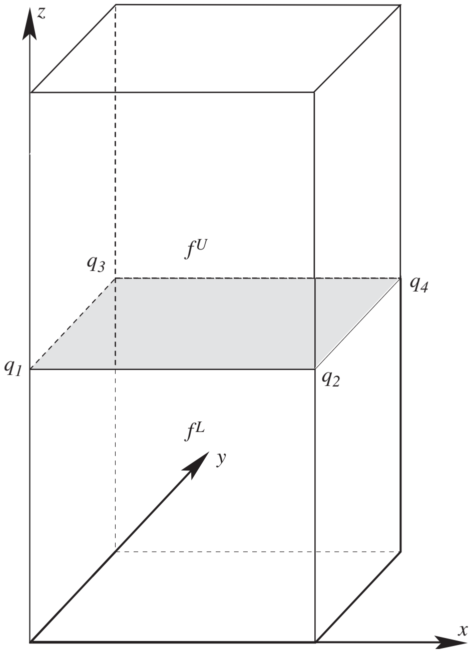

Figure 3. Study of C^1 continuity at the common face of two elements.

and the value of f on the common face is

$$f^{\mathrm{U}}(x, y, 0) = \sum_{i,j=0}^{N} a_{ij0}^{\mathrm{L}} x^i y^j \tag{20}$$

also a bicubic polynomial in x and y. Define

$$b_{ij}^{\mathrm{U}} = a_{ij0}^{\mathrm{U}} \tag{21}$$

so that Equation (20) becomes

$$f^{\mathrm{U}}(x, y, 0) = \sum_{i,j=0}^{N} b_{ij}^{\mathrm{U}} x^i y^j \tag{22}$$

To show that $f^{\mathrm{L}}(x, y, 1) = f^{\mathrm{U}}(x, y, 0)$, it is necessary and sufficient to show that the coefficients corresponding to the same monomials in Equations (18) and (22) are identical.

For this purpose, we define the vectors β^{L} and β^{U} by

$$\beta_{1+i+4j}^{\mathrm{L}} = b_{ij}^{\mathrm{L}} \quad \text{and} \quad \beta_{1+i+4j}^{\mathrm{U}} = b_{ij}^{\mathrm{U}} \tag{23}$$

The value of the function f and the derivatives $\partial f/\partial x$, $\partial f/\partial y$, $\partial^2 f/\partial x \partial y$ can be computed using the coefficients in $\bar{\beta}$. We define the vector $\bar{\rho}$ by stacking the values of these functions

Table I. The matrix M, describing the relationship between the coefficients of the interpolant to a horizontal plane.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
8	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
9	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
11	0	0	0	0	1	0	0	0	2	0	0	0	3	0	0	0
12	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
13	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	1	2	3	0	0	0	0	0	0	0	0
15	0	0	0	0	0	1	0	0	0	2	0	0	0	3	0	0
16	0	0	0	0	0	1	2	3	0	2	4	6	0	3	6	9

at the four common corners q_1, q_2, q_3 and q_4 (see Figure 3). Namely,

$$\rho_i = \begin{cases} f(q_i) & \text{if } 1 \leqslant i \leqslant 4 \\ \dfrac{\partial f}{\partial x}(q_{i-4}) & \text{if } 5 \leqslant i \leqslant 8 \\ \dfrac{\partial f}{\partial y}(q_{i-8}) & \text{if } 9 \leqslant i \leqslant 12 \\ \dfrac{\partial^2 f}{\partial x \partial y}(q_{i-12}) & \text{if } 13 \leqslant i \leqslant 16 \end{cases} \tag{24}$$

The relationship between the derivatives in $\bar{\rho}$ and the coefficients in β is linear and can be written as

$$\bar{\rho} = M\bar{\beta} \tag{25}$$

where the 16×16 matrix M is given in Table I.

By definition of the tricubic interpolator, the values of the function in $\bar{\rho}$ are fixed and given. As a result, their computation using either $\bar{\beta}^{\mathrm{U}}$ or $\bar{\beta}^{\mathrm{L}}$ must give the same result and therefore

$$M\bar{\beta}^{\mathrm{L}} = M\bar{\beta}^{\mathrm{U}} \tag{26}$$

A computation shows that the determinant of the matrix M is equal to -1 and M is therefore invertible. Its inverse M^{-1} is given in Table II and is equivalent (after line and column rearrangements) to the bicubic interpolator matrix presented in Reference [10]. Using M^{-1}, we

Table II. The matrix M^{-1} is the inverse of M, given in Table I.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	−3	3	0	0	−2	−1	0	0	0	0	0	0	0	0	0	0
4	2	−2	0	0	1	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	0	−3	3	0	0	−2	−1	0	0
8	0	0	0	0	0	0	0	0	2	−2	0	0	1	1	0	0
9	−3	0	3	0	0	0	0	0	−2	0	−1	0	0	0	0	0
10	0	0	0	0	−3	0	3	0	0	0	0	0	−2	0	−1	0
11	9	−9	−9	9	6	3	−6	−3	6	−6	3	−3	4	2	2	1
12	−6	6	6	−6	−3	−3	3	3	−4	4	−2	2	−2	−2	−1	−1
13	2	0	−2	0	0	0	0	0	1	0	1	0	0	0	0	0
14	0	0	0	0	2	0	−2	0	0	0	0	0	1	0	1	0
15	−6	6	6	−6	−4	−2	4	2	−3	3	−3	3	−2	−1	−2	−1
16	4	−4	−4	4	2	2	−2	−2	2	−2	2	−2	1	1	1	1

can write Equation (26) as

$$\bar{\beta}^{\mathrm{L}} = M^{-1} M \bar{\beta}^{\mathrm{U}} = \bar{\beta}^{\mathrm{U}} \tag{27}$$

and, therefore, the coefficients of the two bicubic polynomials of Equations (18) and (22) are identical. □

Lemma 5.2
The derivatives

$$\frac{\partial f}{\partial x}, \quad \frac{\partial f}{\partial y}, \quad \frac{\partial^2 f}{\partial x \partial y}, \quad \frac{\partial^2 f}{\partial x^2}, \quad \frac{\partial^2 f}{\partial y^2}, \quad \frac{\partial^3 f}{\partial x^2 \partial y}, \quad \frac{\partial^3 f}{\partial x \partial y^2}, \quad \frac{\partial^3 f}{\partial x^3}, \quad \frac{\partial^3 f}{\partial y^3}$$

are continuous on the horizontal faces of the elements.

Proof
Based on Lemma 5.1, f is represented as a unique polynomial in x and y on the horizontal faces between elements. As a result, higher derivatives of this polynomial (i.e. in x and y, but not z) are also identical on these faces. □

Lemma 5.3
The partial derivative $\partial f/\partial z$ is continuous on the horizontal faces of the elements.

Proof
Notice that Equation (3) implies that

$$\frac{\partial f}{\partial z}(x, y, z) = \sum_{i,j=0}^{N} \sum_{k=1}^{N} a_{ijk} k x^i y^j z^{k-1} \tag{28}$$

As a result, the function $\partial f/\partial z$ can also be represented on a common horizontal face, by Equations (18) and (22), with

$$b_{ij}^{\mathrm{L}} = \sum_{k=1}^{N} k a_{ijk}^{\mathrm{L}} \tag{29}$$

and

$$b_{ij}^{\mathrm{U}} = a_{ij1}^{\mathrm{U}} \tag{30}$$

We define

$$g = \frac{\partial f}{\partial z} \tag{31}$$

and we notice that values of

$$g, \quad \frac{\partial g}{\partial x}, \quad \frac{\partial g}{\partial y} \quad \text{and} \quad \frac{\partial^2 g}{\partial x \partial y}$$

are fixed and equal for the two polynomials at the four common corners. By applying the same reasoning as in the proof of Lemma 5.1, we see that the value of g (i.e. $\partial f/\partial z$) on the common face is independent of the element (lower or upper) used for the computation. □

Lemma 5.4
The derivatives

$$\frac{\partial^2 f}{\partial x \partial z}, \quad \frac{\partial^2 f}{\partial y \partial z}, \quad \frac{\partial^3 f}{\partial x \partial y \partial z}, \quad \frac{\partial^3 f}{\partial x^2 \partial z} \quad \text{and} \quad \frac{\partial^3 f}{\partial y^2 \partial z}$$

are continuous on the horizontal faces of the elements.

Proof
Based on Lemma 5.3, $\partial f/\partial z$ is represented as a unique polynomial in x and y on the horizontal faces between elements. As a result, higher derivatives of this polynomial (i.e. in x and y, not z) are also identical on these faces. □

Similar theorems can be written for the other faces; the results are summarized in Table III, which gives the continuity properties of each derivative on each face.

Lemma 5.5
The functions

$$f, \quad \frac{\partial f}{\partial x}, \quad \frac{\partial f}{\partial y}, \quad \frac{\partial f}{\partial z}, \quad \frac{\partial^2 f}{\partial x \partial y}, \quad \frac{\partial^2 f}{\partial x \partial z}, \quad \frac{\partial^2 f}{\partial y \partial z}, \quad \frac{\partial^3 f}{\partial x \partial y \partial z}$$

are continuous.

Proof
Table III summarizes the smoothness of each derivative through each face. We have shown that the smoothness of any of these derivative depends only on its smoothness through the faces of

Table III. Continuity of the interpolated function and its derivatives.

	$x = 0, 1$	$y = 0, 1$	$z = 0, 1$	Global
f	y	y	y	y
$\frac{\partial f}{\partial x}$	y	y	y	y
$\frac{\partial f}{\partial y}$	y	y	y	y
$\frac{\partial f}{\partial z}$	y	y	y	y
$\frac{\partial^2 f}{\partial x^2}$	n	y	y	n
$\frac{\partial^2 f}{\partial y^2}$	y	n	y	n
$\frac{\partial^2 f}{\partial z^2}$	y	y	n	n
$\frac{\partial^2 f}{\partial x \partial y}$	y	y	y	y
$\frac{\partial^2 f}{\partial x \partial z}$	y	y	y	y
$\frac{\partial^2 f}{\partial y \partial z}$	y	y	y	y
$\frac{\partial^3 f}{\partial x^3}$	n	y	y	n
$\frac{\partial^3 f}{\partial y^3}$	y	n	y	n
$\frac{\partial^3 f}{\partial z^3}$	y	y	n	n
$\frac{\partial^3 f}{\partial x^2 \partial y}$	n	y	y	n
$\frac{\partial^3 f}{\partial x^2 \partial z}$	n	y	y	n
$\frac{\partial^3 f}{\partial x \partial y^2}$	y	n	y	n
$\frac{\partial^3 f}{\partial y^2 \partial z}$	y	n	y	n
$\frac{\partial^3 f}{\partial x \partial z^2}$	y	y	n	n
$\frac{\partial^3 f}{\partial y \partial z^2}$	y	y	n	n
$\frac{\partial^3 f}{\partial x \partial y \partial z}$	y	y	y	y

The three first columns give the continuity with respect to a particular face. The last column gives the global continuity properties in 3D spaces. Functions marked with ‘y’ are necessarily continuous. Functions marked with ‘n’ are not necessarily continuous for any data set.

the elements. As a result, functions that are continuous through all faces in Table III are also continuous everywhere. □

Using these lemmas, we are now ready to give the main result of this paper.

Theorem 5.1
The tricubic interpolated function f is C^1 in three dimensions.

Proof
Lemma 5.5 implies that f is continuous and its three first derivatives are also continuous and therefore f is C^1. □

6. BOUNDARY CONDITIONS AND EXAMPLE

Boundary conditions can be enforced in a variety of ways. For a step-size boundary that coincides with the Cartesian grid, natural and Dirichlet boundary conditions are enforced by setting the corresponding components of the velocity or derivatives to zero before computing the interpolator coefficients.

In the case of a complex coastal problem such as Figure 1, the boundary is usually a polygonal line. Notice that the boundary conditions cannot be enforced directly in the interpolation method. Instead, we interpolate the velocity first (setting the velocity to zero for grid points outside the domain) and apply a mask that smoothly decreases the magnitude of the velocity or its normal component as the point approaches the boundary. Figure 4 shows an example of this procedure. The red arrows are the vectors measured by the radar, as was shown in Figure 1. The black arrows are sampled vectors obtained with the tricubic

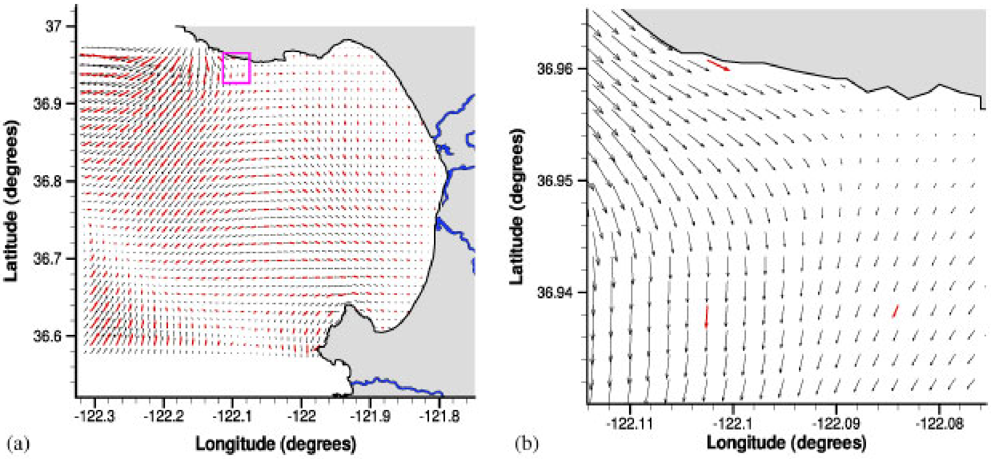

Figure 4. Experimentally observed velocity vectors (red arrows) in Monterey Bay, CA (see Reference [11] for details) and sampled velocity vectors (black arrows) resulting from the tricubic interpolation of the experimental data. Panel (a) shows the whole bay and panel (b) enlarges a small portion of the domain close to the coastline where a Dirichlet boundary condition has been properly enforced.

interpolant. In Figure 4(b) a magnified region shows how the normal component of the velocity vanishes near the coastline, as it should.

7. CONCLUSION

This paper provides a tricubic interpolation scheme in three dimensions that is C^1 and is isotropic. The core of the algorithm is a 64×64 matrix that gives the relationship between the derivatives at the corners of an element and the coefficients of the tricubic interpolant for this element. The resulting algorithm can be downloaded at `http://gyre.cds.caltech.edu/software/tricubic`

Additional files (html, C headers and Fortran headers) containing the interpolation matrix can also be downloaded at this location. Notice that the algorithm can be reproduced by a sequence of one-dimensional cubic interpolations. However, the numerical complexity of the triple cubic interpolator is higher than the tricubic interpolator. To achieve the same smoothness, the functions f, $\partial f/\partial y$, $\partial f/\partial z$ and $\partial^2 f/\partial y \partial z$ must be interpolated in the x direction. Next, the functions $f|_x$ and $(\partial f/\partial z)\big|_x$ can be interpolated along the y-axis. This allows a one-dimensional interpolation of $f|_{x,y}$ along the z-axis. If the value of f is to be evaluated at only one point inside an element, the two schemes are equivalent. However, tricubic interpolation allows the computation of unique coefficients for the whole element that can be stored and used repetitively for subsequent interpolation in the same element. In addition, the tricubic interpolator also allows the extraction of the exact derivatives of the interpolated function by *analytical* differentiation of the tricubic polynomial, which is not the case for a sequence of one-dimensional cubic interpolants.

Based on the previous remarks, the use of the tricubic interpolator described in this paper becomes advantageous when

- C^1-smoothness is required. In the case of the example presented in Figure 4 and many other geophysical problems, C^1 continuity is required to apply basic tools in dynamical system theory. These tools are usually not applicable to fields that are only C^0.
- Local interpolation (as opposed to global splines) is preferred. In cases such as the one presented in this paper, local interpolation is usually preferred to global splines because the data sets might be sparse, noisy and the boundary conditions might not be known everywhere [12]. Local interpolation also avoids propagating experimental error from poorly sampled regions to the element of interest.
- Either one of these two conditions is satisfied:
  1. The derivatives of the interpolated field are also needed. In such a case, direct tricubic interpolation (as opposed to three one-dimensional cubic interpolators) directly gives the value of the interpolated function and the derivatives. The sequence of one-dimensional engines would have to be run at least once for each derivative, or
  2. The interpolation has to be made at several points inside at least one element. In such a case, the full tricubic interpolator given in this paper provides 64 coefficients for the whole element. These can be stored after the first use. In comparison, the scheme composed of three one-dimensional interpolators will only give the interpolated function at a point. The whole process has to be repeated if interpolation is required at another

point in the same elements. The tricubic interpolator described in this paper provides 64 coefficients for the whole element. In such a case, the efficiency of the tricubic interpolator will be much higher.

There are three possible ways to implement a tricubic interpolator that must be called several times. First, one can simply compute the 64 coefficients for the element each time the interpolated function is required at a triplet (x, y, z), inside this element. This procedure does not give an advantage over the combination of one-dimensional cubic interpolators unless the derivatives are also required. Second, one can compute the 64 coefficients for each element beforehand and use the saved coefficients during the interpolation step. This procedure is only advantageous when the interpolation is done at a large number of points, preferably in as many elements as possible. Finally, a hybrid algorithm can combine the advantages of the two first methods. For each triplet (x, y, z) where interpolation has to be done, we determine the corresponding element. If the 64 coefficients have not been computed yet for this element, they are computed and saved in memory. If the 64 coefficients have already been computed (for another point in the same element), the saved coefficients are used. The hybrid method keeps the advantage of not repeating the same computation several times, but does not require the computation of the coefficients in elements where interpolation is not needed. This is particularly advantageous for dynamical systems tools that concentrate only on certain regions. These tools will require interpolation at many points, but located in only a few elements close to a dynamical feature.

The hybrid version of the tricubic interpolator has been included in MANGEN (see Reference [3] for example), a software package implementing various ocean and coastal environment analysis tools. A typical run of MANGEN requires more than 1000 points inside at least one element (see Figure 4(b) for example). The use of the tricubic interpolator combined with the storage of the coefficients (hybrid method) reduced the computation time by a factor of 100 compared to the combination of one-dimensional interpolators. Notice that this high efficiency would not be achieved if the interpolation was only required in only 1 or 2 points inside each element.

The method presented in this paper is designed for regular Cartesian meshes in three dimensions. Such meshes are made exclusively of rectangular parallelepipeds but not necessarily all of the same size. The results do not translate easily to other types of elements. In particular, future directions include development of a tricubic interpolant on an arbitrary hexahedron. Notice that for systems where the third axis is time and not a spatial dimension, one can assume that the mesh in space does not change and the elements are prisms whose bases are irregular parallelograms in the (x, y) plane. Another direction for future work is the development of higher order interpolants. We have shown that, for any n-dimensional space, a cubic interpolant cannot guarantee C^2 continuity. Increased smoothness can only be achieved by increasing the order of the interpolating polynomials.

ACKNOWLEDGEMENTS

This project was supported by the Office of Naval Research through ONR contract N00014-01-1-0208 and the AOSN-II project under ONR contract N00014-02-1-0826 to observe and understand geophysical flows and coastal processes. The authors are grateful to Dallas Trinkle and MANGEN users for their ideas, remarks and tests of the tricubic interpolator. The HF radar data used in this paper was collected and processed by Jeffrey Paduan and Michael Cook (see Reference [11]).

REFERENCES

1. Haller G. Lagrangian coherent structures from approximate velocity data. *Physics of Fluids* 2002; **14**: 1851–1861.
2. Coulliette C, Wiggins S. Intergyre transport in a wind-driven, quasi-geostrophic double gyre: an application of lobe dynamics. *Nonlinear Processes in Geophysics* 2001; **8**:69–94.
3. Lekien F, Coulliette C, Marsden J. Lagrangian structures in very high-frequency radar data and optimal pollution timing. *American Institute of Physics Conference Proceedings* 2003; **676**:162–168.
4. Wiggins S. *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. Springer: Berlin, 1990.
5. Kadosh A, Cohen-Or D, Yagel R. Tricubic interpolation of discrete surfaces for binary volumes. *IEEE Transactions on Visualization and Computer Graphics* 2003; **9**(4):580–586.
6. LaMar E, Hamann B, Joy KI. High-quality rendering of smooth isosurfaces. *Journal of Visualization and Computer Animation* 1999; **10**(2):79–90.
7. Katan C, Rabiller R, Lecomte C, Guezo M, Oison V, Souhassou M. Numerical computation of critical properties and atomic basins from three-dimensional grid electron densities. *Journal of Applied Crystallography* 2003; **36**:65–73.
8. Catmull E, Rom R. A class of local interpolating splines. In *Computer Aided Geometric Design*, Barnhill R, Riesenfeld R (eds). Academic Press: New York, 1974; 317–326.
9. Farin G. *Curves and Surfaces for Computer Aided Geometric Design* (4th edn). Academic Press: Boston, MA, 1997.
10. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in C++, the Art of Scientific Computing*. Cambridge University Press: Cambridge, U.K., 2002.
11. Paduan JD, Cook MS. Mapping surface currents in Monterey Bay with radar-type HR data. *Oceanography* 1997; **10**:49–52.
12. Lekien F, Coulliette C, Banks R, Marsden J. Open-boundary modal analysis: interpolation, extrapolation and filtering. *Journal of Geophysical Research—Oceans* 2004; **109**(C12): Art. No. C12004.