

# Tricubic Engine

## Technical Notes and Full Matrix

Francois Lekien, Chad Coulliette and Jerry Marsden  
California Institute of Technology

January 20, 2004

### Abstract

The objective of this paper is to document a local interpolation scheme in three dimensions, available as a package called `tricubic`. In contrast to global interpolation where the interpolated function usually depends on the whole data set, local interpolation only uses data in a neighborhood of an element. The function interpolated with the tricubic interpolator and its three first derivatives are continuous. As shown in Lekien, Coulliette and Marsden, 2004, this is the only cubic  $C^1$  local interpolator. The implementation of the interpolator can be downloaded as a static and dynamic library for most platforms.

## Contents

<b>1</b>	<b>Interpolator Equation</b>	<b>1</b>
<b>2</b>	<b>Numerical Implementation</b>	<b>11</b>
<b>3</b>	<b>Example</b>	<b>12</b>
<b>4</b>	<b>Further reading, Comments, Remarks and Bugs</b>	<b>14</b>

## 1 Interpolator Equation

We assume that the value of the function  $f$  is given at the corners of a regular mesh (see Fig. 1). The function  $f$  to interpolate is represented in each element by

$$f(x, y, z) = \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N a_{ijk} x^i y^j z^k . \quad (1)$$

The coefficients  $a_{ijk}$  are determined in Lekien, Coulliette and Marsden, 2004 in such a way that the function  $f$  described by Eq. 1 is  $C^1$ , i.e.  $f$  is continuous and its 3 first derivatives are also continuous.

For the sake of simplicity, the 64 coefficients  $a_{ijk}$  are stacked in a vector  $\bar{\alpha}$  by defining

$$\alpha_{1+i+4j+16k} = a_{ijk} \quad \forall i, j, k \in \{0, 1, 2, 3\} . \quad (2)$$

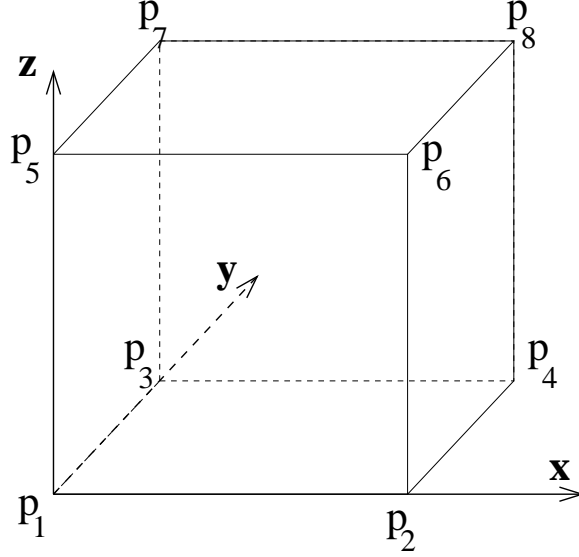


Figure 1: Element for interpolation in three dimensions.

Similarly, we give a unique indice to each corner of the cube as shown in Fig. 1. We stack the constraints on  $f$  and its derivatives in a vector  $\bar{b}$  by defining

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

Based on Eq. 1, the derivatives of  $f$  can be computed and evaluated for the 8 points  $p_i$ . This gives a linear system in the 64 unknown coefficients  $\alpha_i$

$$B\bar{\alpha} = \bar{b}, \quad (4)$$

The matrix  $B$  is  $64 \times 64$  and has only integer entries. It can be computed by a computer program. The elements of the matrix  $B$  are irrelevant for the methods, so we do not give them in this document. It is important to note that the determinant of  $B$  is

$$\det(B) = 1. \quad (5)$$

As a result,  $B$  is invertible and the coefficient  $\alpha_i$  can be computed using the linear relationship

$$\bar{\alpha} = B^{-1}\bar{b}. \quad (6)$$

The matrix  $B^{-1}$  is the core of the tricubic interpolator. It can be found in Table 1 to Table 8.

	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	0	0	0	0	1	0	0	0	0	0	0	0
3	-3	3	0	0	0	0	0	0	-2	-1	0	0	0	0	0	0
4	2	-2	0	0	0	0	0	0	1	1	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	-3	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	-3	0	3	0	0	0	0	0
11	9	-9	-9	9	0	0	0	0	6	3	-6	-3	0	0	0	0
12	-6	6	6	-6	0	0	0	0	-3	-3	3	3	0	0	0	0
13	2	0	-2	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	2	0	-2	0	0	0	0	0
15	-6	6	6	-6	0	0	0	0	-4	-2	4	2	0	0	0	0
16	4	-4	-4	4	0	0	0	0	2	2	-2	-2	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 1: Part1

	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	-3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	2	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	-2	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	6	-6	3	-3	0	0	0	0	0	0	0	0	0	0	0	0
12	-4	4	-2	2	0	0	0	0	0	0	0	0	0	0	0	0
13	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	-3	3	-3	3	0	0	0	0	0	0	0	0	0	0	0	0
16	2	-2	2	-2	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	-3	3	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	2	-2	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	-3	0	3	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	9	-9	-9	9	0	0	0	0
28	0	0	0	0	0	0	0	0	-6	6	6	-6	0	0	0	0
29	0	0	0	0	0	0	0	0	2	0	-2	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	-6	6	6	-6	0	0	0	0
32	0	0	0	0	0	0	0	0	4	-4	-4	4	0	0	0	0

Table 2: Part2

	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	-2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	-2	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
11	4	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0
12	-2	-2	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	-2	-1	-2	-1	0	0	0	0	0	0	0	0	0	0	0	0
16	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	-2	-1	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	-3	0	3	0	0	0	0	0
27	0	0	0	0	0	0	0	0	6	3	-6	-3	0	0	0	0
28	0	0	0	0	0	0	0	0	-3	-3	3	3	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	2	0	-2	0	0	0	0	0
31	0	0	0	0	0	0	0	0	-4	-2	4	2	0	0	0	0
32	0	0	0	0	0	0	0	0	2	2	-2	-2	0	0	0	0

Table 3: Part3

	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
23	-3	3	0	0	0	0	0	0	-2	-1	0	0	0	0	0	0
24	2	-2	0	0	0	0	0	0	1	1	0	0	0	0	0	0
25	-2	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	-2	0	-1	0	0	0	0	0
27	6	-6	3	-3	0	0	0	0	4	2	2	1	0	0	0	0
28	-4	4	-2	2	0	0	0	0	-2	-2	-1	-1	0	0	0	0
29	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
31	-3	3	-3	3	0	0	0	0	-2	-1	-2	-1	0	0	0	0
32	2	-2	2	-2	0	0	0	0	1	1	1	1	0	0	0	0

Table 4: Part4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
33	-3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	-3	0	0	0	3	0	0	0
35	9	-9	0	0	-9	9	0	0	6	3	0	0	-6	-3	0	0
36	-6	6	0	0	6	-6	0	0	-3	-3	0	0	3	3	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	9	0	-9	0	-9	0	9	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	9	0	-9	0	-9	0	9	0
43	-27	27	27	-27	27	-27	-27	27	-18	-9	18	9	18	9	-18	-9
44	18	-18	-18	18	-18	18	18	-18	9	9	-9	-9	-9	-9	9	9
45	-6	0	6	0	6	0	-6	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	-6	0	6	0	6	0	-6	0
47	18	-18	-18	18	-18	18	18	-18	12	6	-12	-6	-12	-6	12	6
48	-12	12	12	-12	12	-12	-12	12	-6	-6	6	6	6	6	-6	-6
49	2	0	0	0	-2	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	2	0	0	0	-2	0	0	0
51	-6	6	0	0	6	-6	0	0	-4	-2	0	0	4	2	0	0
52	4	-4	0	0	-4	4	0	0	2	2	0	0	-2	-2	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
57	-6	0	6	0	6	0	-6	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	-6	0	6	0	6	0	-6	0
59	18	-18	-18	18	-18	18	18	-18	12	6	-12	-6	-12	-6	12	6
60	-12	12	12	-12	12	-12	-12	12	-6	-6	6	6	6	6	-6	-6
61	4	0	-4	0	-4	0	4	0	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0	4	0	-4	0	-4	0	4	0
63	-12	12	12	-12	12	-12	-12	12	-8	-4	8	4	8	4	-8	-4
64	8	-8	-8	8	-8	8	8	-8	4	4	-4	-4	-4	-4	4	4

Table 5: Part5

	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	0	0	0	0	0	0	0	0	-2	0	0	0	-1	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	6	-6	0	0	3	-3	0	0
36	0	0	0	0	0	0	0	0	-4	4	0	0	-2	2	0	0
37	-3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	9	-9	0	0	-9	9	0	0	0	0	0	0	0	0	0	0
40	-6	6	0	0	6	-6	0	0	0	0	0	0	0	0	0	0
41	6	0	3	0	-6	0	-3	0	6	0	-6	0	3	0	-3	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	-18	18	-9	9	18	-18	9	-9	-18	18	18	-18	-9	9	9	-9
44	12	-12	6	-6	-12	12	-6	6	12	-12	-12	12	6	-6	-6	6
45	-3	0	-3	0	3	0	3	0	-4	0	4	0	-2	0	2	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	9	-9	9	-9	-9	9	-9	9	12	-12	-12	12	6	-6	-6	6
48	-6	6	-6	6	6	-6	6	-6	-8	8	8	-8	-4	4	4	-4
49	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	-3	3	0	0	-3	3	0	0
52	0	0	0	0	0	0	0	0	2	-2	0	0	2	-2	0	0
53	2	0	0	0	-2	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	-6	6	0	0	6	-6	0	0	0	0	0	0	0	0	0	0
56	4	-4	0	0	-4	4	0	0	0	0	0	0	0	0	0	0
57	-4	0	-2	0	4	0	2	0	-3	0	3	0	-3	0	3	0
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
59	12	-12	6	-6	-12	12	-6	6	9	-9	-9	9	9	-9	-9	9
60	-8	8	-4	4	8	-8	4	-4	-6	6	6	-6	-6	6	6	-6
61	2	0	2	0	-2	0	-2	0	2	0	-2	0	2	0	-2	0
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
63	-6	6	-6	6	6	-6	6	-6	-6	6	6	-6	-6	6	6	-6
64	4	-4	4	-4	-4	4	-4	4	4	-4	-4	4	4	-4	-4	4

Table 6: Part6



	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	-2	0	0	0	-1	0	0	0
35	0	0	0	0	0	0	0	0	4	2	0	0	2	1	0	0
36	0	0	0	0	0	0	0	0	-2	-2	0	0	-1	-1	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	-3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
39	6	3	0	0	-6	-3	0	0	0	0	0	0	0	0	0	0
40	-3	-3	0	0	3	3	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	6	0	3	0	-6	0	-3	0	6	0	-6	0	3	0	-3	0
43	-12	-6	-6	-3	12	6	6	3	-12	-6	12	6	-6	-3	6	3
44	6	6	3	3	-6	-6	-3	-3	6	6	-6	-6	3	3	-3	-3
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	-3	0	-3	0	3	0	3	0	-4	0	4	0	-2	0	2	0
47	6	3	6	3	-6	-3	-6	-3	8	4	-8	-4	4	2	-4	-2
48	-3	-3	-3	-3	3	3	3	3	-4	-4	4	4	-2	-2	2	2
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
51	0	0	0	0	0	0	0	0	-2	-1	0	0	-2	-1	0	0
52	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54	2	0	0	0	-2	0	0	0	0	0	0	0	0	0	0	0
55	-4	-2	0	0	4	2	0	0	0	0	0	0	0	0	0	0
56	2	2	0	0	-2	-2	0	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
58	-4	0	-2	0	4	0	2	0	-3	0	3	0	-3	0	3	0
59	8	4	4	2	-8	-4	-4	-2	6	3	-6	-3	6	3	-6	-3
60	-4	-4	-2	-2	4	4	2	2	-3	-3	3	3	-3	-3	3	3
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62	2	0	2	0	-2	0	-2	0	2	0	-2	0	2	0	-2	0
63	-4	-2	-4	-2	4	2	4	2	-4	-2	4	2	-4	-2	4	2
64	2	2	2	2	-2	-2	-2	-2	2	2	-2	-2	2	2	-2	-2

Table 7: Part7

	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	-2	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	-2	0	0	0	-1	0	0	0
39	6	-6	0	0	3	-3	0	0	4	2	0	0	2	1	0	0
40	-4	4	0	0	-2	2	0	0	-2	-2	0	0	-1	-1	0	0
41	4	0	2	0	2	0	1	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	4	0	2	0	2	0	1	0
43	-12	12	-6	6	-6	6	-3	3	-8	-4	-4	-2	-4	-2	-2	-1
44	8	-8	4	-4	4	-4	2	-2	4	4	2	2	2	2	1	1
45	-2	0	-2	0	-1	0	-1	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	-2	0	-2	0	-1	0	-1	0
47	6	-6	6	-6	3	-3	3	-3	4	2	4	2	2	1	2	1
48	-4	4	-4	4	-2	2	-2	2	-2	-2	-2	-2	-1	-1	-1	-1
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
53	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
55	-3	3	0	0	-3	3	0	0	-2	-1	0	0	-2	-1	0	0
56	2	-2	0	0	2	-2	0	0	1	1	0	0	1	1	0	0
57	-2	0	-1	0	-2	0	-1	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	-2	0	-1	0	-2	0	-1	0
59	6	-6	3	-3	6	-6	3	-3	4	2	2	1	4	2	2	1
60	-4	4	-2	2	-4	4	-2	2	-2	-2	-1	-1	-2	-2	-1	-1
61	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0
63	-3	3	-3	3	-3	3	-3	3	-2	-1	-2	-1	-2	-1	-2	-1
64	2	-2	2	-2	2	-2	2	-2	1	1	1	1	1	1	1	1

Table 8: Part8

## 2 Numerical Implementation

An algorithm using the tricubic interpolator consist of the following 3 steps:

- Vector  $\bar{b}$ . The values of  $f$  and the derivatives at the 8 corners of the cube must be stacked in vector of dimension 64. The values of  $f$  are usually given at the corner and the derivatives can be evaluated by finite differences and the boundary conditions. Notice that the method described here assumes a cube of side 1. As a result, of the cube has arbitrary sides  $\delta_x, \delta_y$ , and  $\delta_z$ , the derivatives in  $x, y, z$  must be respectively multiplied by  $\delta_x, \delta_y$  and  $\delta_z$ . For example, if the derivative  $\partial^2 \partial x \partial y$  is computed in the initial cube, it must be multiplied by  $\delta_x \delta_y$  before computing the tricubic coefficients.
- Vector  $\bar{a}$ . The value of the coefficients are computed using the relation  $\bar{a} = B^{-1} \bar{b}$ .
- Value of the function. The value of the function  $f$  can be computed at any point of the cube by the relationship given by Eq. 1. Notice that the coordinates  $x, y, z$  are relative to the unit cube. In other words, the coordinates  $x, y$ , and  $z$  must be translated to the corner  $p_1$  of the cube and scaled respectively by  $\delta_x, \delta_y$  and  $\delta_z$ . If the value of the derivatives of the function  $f$  are needed, one can differentiate Eq. 1 with respect to any set of variables and get a form for the derivatives of  $f$ . Notice that the derivatives are given in terms of the unit cube and must be divided by the actual size of the elements. In other words, the value of  $\partial^{i+j+k} f / \partial x^i \partial y^j \partial z^k$  must be divided by  $\delta_x^i \delta_y^j \delta_z^k$ .

This algorithm has been implemented in a library called `libtricubic` and is available for download at <http://gyre.cds.caltech.edu/software/tricubic> . Programs using the tricubic library usually have an include line that imports the prototype of the tricubic function

```
#include 'tricubic.h'
```

The source code is usually linked with the tricubic library with the following arguments

```
gcc source.cpp -L/path/to/tricubiclib/ -ltricubic
```

If the installation of `libtricubic` is system-wide (i.e. run `make install` as root), the above code and command are simplified to

```
#include <tricubic.h>
```

The source code is usually linked with the tricubic library with the following arguments

```
gcc source.cpp -ltricubic
```

The functions available in the library are

- `char *tricubic_version(void)`. Returns a string containing the version of the library.
- `void tricubic_get_coeff(double a[64], double f[8], double dfdx[8], double dfdy[8], double dfdz[8], double d2fdxdy[8], double d2fdxdz[8], double d2fdydz[8], double d3fdxdydz[8])`. Takes 8 vectors of dimension 8 containing the value of  $f$  and the required derivatives for each point of Fig. 1. Returns the 64 coefficients of the tricubic interpolation in `a[64]`.

- `double tricubic_eval(double a[64], double x, double y, double z, int derx, int dery, int derz)`. Takes the 64 coefficients `a[64]` as input and compute a value at point `x,y,z`. The three coordinates are given with respect to the corner  $p_1$  in Fig. 1 and are scaled in such a way that the element is a cube of side 1. The function computed is

$$\frac{\partial^{derx+dery+derz} f}{\partial x^{derx} \partial y^{dery} \partial z^{derz}}, \quad (7)$$

where we use the convention  $f = \frac{\partial^0}{\partial x^0 \partial y^0 \partial z^0}$  to compute the value of the function  $f$ .

- `double tricubic_eval(double a[64], double x, double y, double z)`. Shortcut for the previous function with `derx=dery=derz=0`.

### 3 Example

```

/*****
/* example1.cpp : illustrates the use of libtricubic          */
/* Francois Lekien <lekien@mit.edu> 2004-01-20                */
/*****/

//Required Include File: tricubic.h
//If tricubic.h has not been installed in a directory
// accessible by the compiler, use -I/path/to/tricubic
#include <tricubic.h>

//Define the box
//Tricubic is written for cubes of side 1. Multiplications and
// divisions are needed along the way for rectangular box with
// arbitrary sides. See below for details.
#define dx 0.2
#define dy 2.0
#define dz 0.3

//These are the 8 functions that need to be known at the 8 corners.
//The functions are f, the three first derivatives dfdx, dfdy, dfdz,
// the 3 mixed 2nd order derivatives and the mixed 3rd order derivative.
//The derivatives can be obtained by numerical differentiation of f at the
// other corners of the grid. See Numerical Recipies for examples in 2D
//The order of the points is as follow:
// 0: x=0; y=0; z=0;
// 1: x=1; y=0; z=0;
// 2: x=0; y=1; z=0;
// 3: x=1; y=1; z=0;
// 4: x=0; y=0; z=1;
// 5: x=1; y=0; z=1;
// 6: x=0; y=1; z=1;

```

```

/// 7: x=1; y=1; z=1;
///For convenience, the ordering of the points is available at run time
/// using tricubic_pointID2xyz(). See man page for details
double fval[8]={1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 7.8, 8.9};
double dfdxval[8]={1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 7.8, 8.9};
double dfdyval[8]={1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 7.8, 8.9};
double dfdzval[8]={1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 7.8, 8.9};
double d2fdxdyval[8]={1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 7.8, 8.9};
double d2fdxdzval[8]={1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 7.8, 8.9};
double d2fdydzval[8]={1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 7.8, 8.9};
double d3fdxdydzval[8]={1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 7.8, 8.9};

int main(int narg, char *arg[]) {
    double a[64];
    double f1, f2, dfdx, d2fdxdy;
    int i;
    double x,y,z;
    ///First we compute the 64 coefficients for the cube.
    ///The first step is to scale the derivatives that have been computed
    /// in a rectangular box instead of the cube of side 1.
    ///Notice that this step can be avoided by computing the numerical
    /// derivatives without dividing by the length of the boxes.
    for (i=0;i<8;i++) {
        fval[i]*=1.0;
        dfdxval[i]*=dx;
        dfdyval[i]*=dy;
        dfdzval[i]*=dz;
        d2fdxdyval[i]*=dx*dy;
        d2fdxdzval[i]*=dx*dz;
        d2fdydzval[i]*=dy*dz;
        d3fdxdydzval[i]*=dx*dy*dz;
    }
    ///Next we get the set of coefficient for this cube
    tricubic_get_coeff(a,fval,dfdxval,dfdyval,dfdzval,d2fdxdyval,
                      d2fdxdzval,d2fdydzval,d3fdxdydzval);
    ///To get the value in the middle of the cube, we always use (.5,.5,.5)
    /// i.e. relative coordinates
    f1=tricubic_eval(a,.5,.5,.5);
    ///To get the value at a point x,y,z (with reference to corner ID 0
    /// we divide by each length
    f2=tricubic_eval(a,x/dx,y/dy,z/dz);
    ///Derivatives can be computed similarly but need to be scaled by dx,dy,dz
    dfdx=tricubic_eval(a,x/dx,y/dy,z/dz,1,0,0)/dx;
    d2fdxdy=tricubic_eval(a,x/dx,y/dy,z/dz,1,1,0)/(dx*dy);
}

```

## 4 Further reading, Comments, Remarks and Bugs

The proofs of  $C^1$  continuity and the uniqueness of the tricubic  $C^1$  interpolator has been published in the *International Journal for Numerical Methods in Mechanical Engineering*. See *F. Lekien, C. Coulliette and J. Marsden, Tricubic Interpolation, IJNME vol ??, pp 1–10, 2004* for more details. Send comments, remarks and bug reports to Francois Lekien, Caltech 107-81, 1201 East California blvd, Pasadena, CA 91125 or [lekien@mit.edu](mailto:lekien@mit.edu).