European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2006
P. Wesseling, E. Oñate and J. Périaux (Eds)
© TU Delft, The Netherlands, 2006

A SCATTERED DATA INTERPOLANT FOR THE SOLUTION OF THREE-DIMENSIONAL PDES

Hassan Goldani Moghaddam* **and Wayne H. Enright**†

*Department of Computer Science
University of Toronto
10 King's College Road, Toronto, ON, M5S 3G4, Canada
e-mail: goldani@cs.toronto.edu

†Department of Computer Science
University of Toronto
10 King's College Road, Toronto, ON, M5S 3G4, Canada
e-mail: enright@cs.toronto.edu

Key words: Scattered Data, Scientific Visualization, Interpolation, Three-Dimensional PDE.

Abstract. *Using Differential Equation Interpolant (DEI), one can accurately approximate the solution of a Partial Differential Equation (PDE) at off-mesh points. The idea is to allocate a multi-variate polynomial to each mesh element and consequently, the collection of such polynomials over all mesh elements will define a piecewise polynomial approximation. In this paper we will investigate such interpolants on a three-dimensional unstructured mesh. As reported in [1], for a tetrahedron mesh in three dimensions, tensor product tri-quadratic and pure tri-cubic interpolants are the most appropriate candidates. We will report on the effectiveness of these alternatives on some typical PDEs.*

1 Introduction

1.1 Motivation

In many practical applications, the underlying system is modeled by Partial Differential Equations (PDEs). In most applications, the underlying PDEs do not have a closed form solution. In these cases, effective numerical methods can be applied to approximate the solution at a discrete set of mesh points in the domain associated with the problem definition. Although these approximations can be very accurate at mesh points, if one wishes to visualize some properties of the solution on the whole domain, some extra data at off-mesh points must be generated. In [1], Enright introduced the Differential Equation Interpolant (DEI) which approximates the solution of a PDE such that the approximations at off-mesh points have the same order of accuracy as those at mesh points.

In the case that the underlying numerical method produces approximations on an unstructured mesh, the DEI can be considered to be a scattered data interpolant. In [3], we investigated such problems and introduced the PCI, a scattered data interpolant, for a two-dimensional second-order elliptic PDE. The PCI is globally continuous and efficient in terms of time and error. In this paper, we will extend the PCI to a three dimensional tri-variate interpolant and identify and investigate the most efficient interpolants based on the DEI approach.

1.2 Outline

In the next section, the problem of scattered data interpolation in three dimensions will be defined followed by three candidate interpolants. In section 3, numerical results will be presented.

2 Problem Definition

In this paper, we focus on scattered discrete data associated with the numerical solution of a three-dimensional second-order elliptic PDE of the form

$$Lu = g(x, y, z, u, u_x, u_y, u_z),$$

where L is a given differential operator of the form

$$L = a_1(x, y, z)\frac{\partial^2}{\partial x^2} + a_2(x, y, z)\frac{\partial^2}{\partial y^2} + a_3(x, y, z)\frac{\partial^2}{\partial z^2}.$$

We assume that there are some accurate numerical results (approximate solution values, $u(x, y, z)$, as well as approximate derivative values, $u_x(x, y, z)$, $u_y(x, y, z)$ and $u_z(x, y, z)$) at some mesh points that are not necessarily structured. The mesh points partition the domain of the problem into a collection of mesh elements which are tetrahedra. Our approach is to associate with each mesh element e, a tri-variate polynomial $p_{d,e}(x, y, z)$ of degree d, which approximates $u(x, y, z)$ on mesh element e. In other words, one can determine a polynomial $p_{d,e}(x, y, z)$ that interpolates the data values associated with the mesh points of e and 'almost' satisfies the PDE at a predetermined set of collocation points of e. The number of collocation points depends on the degree d and type of interpolant (tensor product or pure). The collection of such polynomials over all mesh elements will then define a piecewise polynomial approximation $p_d(x, y, z)$, that is well defined for all (x, y, z) in the domain of interest.

In [1], Enright investigated the performance of this approach for both two and three dimensions. He reported that, for three-dimensional problems and tetrahedron meshes, pure tri-cubic interpolants and tensor product tri-quadratic are the most appropriate candidates. In this paper we compare the effectiveness of these alternatives on some test problems.

A pure tri-cubic polynomial for a mesh element e is defined by

$$p_1(x,y,z) = \sum_{i=0}^{3}\sum_{j=0}^{3-i}\sum_{k=0}^{3-i-j} c_{ijk} s^i t^j v^k,$$

where

$$s = \frac{(x-x_1)}{D_1}, t = \frac{(y-y_1)}{D_2}, v = \frac{(z-z_1)}{D_3},$$

and (x_1, y_1, z_1) is the corner of the associated enclosing box of e with the smallest values of (x, y, z); and D_1, D_2 and D_3 are the dimensions of the box.

The number of unknown coefficients, c_{ijk}s, for a pure three-dimensional interpolant of degree d can be expressed by $\sum_{k=0}^{d} \frac{(k+1)(k+2)}{2}$. Thus for a pure tri-cubic, there are 20 unknown coefficients. Since we already have 16 pieces of information (u, u_x, u_y and u_z for each of four nodes of a tetrahedron), we have to consider 4 collocation points to uniquely determine the interpolant.

A tensor product tri-quadratic polynomial can be also defined by

$$p_2(x,y,z) = \sum_{i=0}^{2}\sum_{j=0}^{2}\sum_{k=0}^{2} c_{ijk} s^i t^j v^k.$$

For a tensor product three-dimensional interpolant of degree d, there are $(d+1)^3$ unknown coefficients. Therefore for a tensor product tri-quadratic, we have 27 unknowns to identify. This results to consider 11 collocation points for each mesh elements.

Note that a pure tri-quadratic polynomial makes a total degree 2 and has only 10 unknowns. Since the number of unknowns is less than the number of linear equations provided by the information at four mesh points of e, it is not appropriate to investigate this type of interpolant. However we can consider a tri-quadratic polynomial of total degree 3 as follows:

$$p_3(x,y,z) = \sum_{i=0}^{2}\sum_{j=0}^{\min(2,3-i)}\sum_{k=0}^{\min(2,3-i-j)} c_{ijk} s^i t^j v^k.$$

Since p_3 has only 17 unknown coefficients, it requires less time to compute rather than p_2. Our results show that it also generates more accurate results than p_2 in practice.

3 Results

In this section, we will first introduce two test problems. We will then present test results comparing three mentioned candidates.

3.1 Test Problems

Both test problems have a known closed-form solution and we use this known solution to generate the required data at the unstructured mesh points. The first test problem is a three-dimensional second-order elliptic PDE:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 2\pi \cos(\pi x)\sin(\pi y)\sin(\pi z) - 3\pi^2 u$$

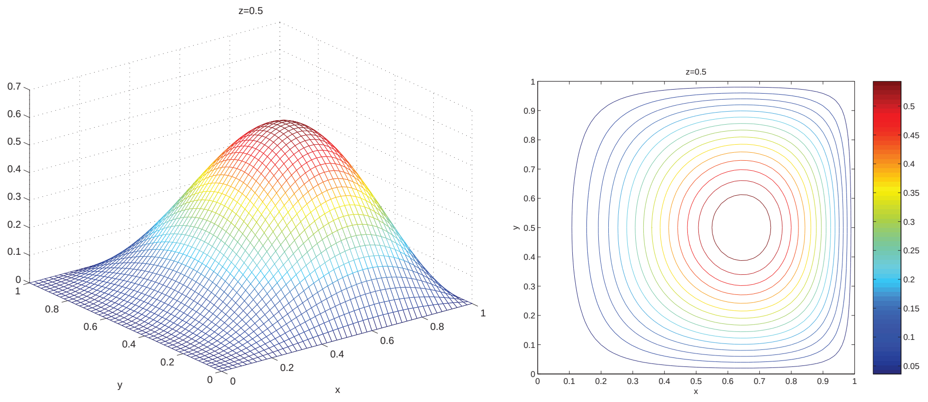
on the domain

$$0 \le x \le 1,\ 0 \le y \le 1,\ 0 \le z \le 1$$

and its closed-form solution is

$$u(x, y, z) = x \sin(\pi x)\sin(\pi y)\sin(\pi z).$$

Figure 1 shows its surface and contour plots for fixed $z = 0.5$.



(a) The surface plot (b) The contour plot

Figure 1: The first test problem: The surface and contour plots.

The Second test problem is also a three-dimensional second-order elliptic PDE:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 6x(y^2 - y)(z^2 + 1) + (x^2 - x)(6y - 2)(z^2 + 1) + 2(x^2 - x)(y^2 - y),$$

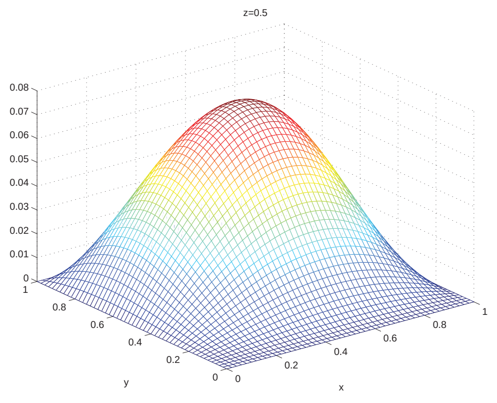
on the domain

$$0 \le x \le 1,\ 0 \le y \le 1,\ 0 \le z \le 1$$

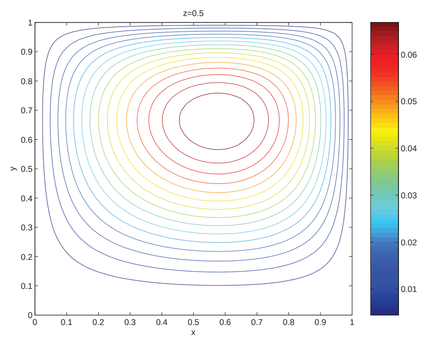
and its closed-form solution is

$$u(x, y, z) = (x^3 - x)(y^3 - y^2)(z^2 + 1).$$

Figure 2 shows its surface and contour plots for fixed $z = 0.5$.



(a) The surface plot



(b) The contour plot

Figure 2: The second test problem: The surface and contour plots.

3.2 Test Results

In this section, we compare three candidate interpolants in terms of time, accuracy and visualization. As of accuracy, the average error over 1000 regular points in the domain has been computed. The scattered data is generated using a random approach and triangularized by the built-in MATLAB *delaunay3* function [2].

	# Mesh Points	# Mesh Elements	*Interpolant*		
			p_1	p_2	p_3
First Test Problem	$4 \times 4 \times 4$	303	8.4×10^{-3}	1.065×10^{-1}	2.95×10^{-2}
	$8 \times 8 \times 8$	3146	4.97×10^{-4}	9.6×10^{-3}	5.8×10^{-3}
	$16 \times 16 \times 16$	26880	3.76×10^{-5}	1.3×10^{-3}	1.31×10^{-4}
	$32 \times 32 \times 32$	219273	5.56×10^{-6}	1.67×10^{-4}	4.04×10^{-5}
Observed Order			3.52	3.10	3.17
Second Test Problem	$4 \times 4 \times 4$	303	1.5×10^{-3}	2.03×10^{-2}	8.6×10^{-3}
	$8 \times 8 \times 8$	3146	5.42×10^{-5}	2.8×10^{-3}	2.3×10^{-3}
	$16 \times 16 \times 16$	26880	4.76×10^{-6}	1.72×10^{-4}	3.14×10^{-5}
	$32 \times 32 \times 32$	219273	4.77×10^{-7}	2.98×10^{-5}	1.02×10^{-5}
Observed Order			3.87	3.14	3.24

Table 1: Average error for both test problems.

Table 1 shows the average error of the candidate interpolants for different number of mesh points. As expected, the pure cubic interpolant p_1 has the most accurate results for both test problems. Furthermore, table 2 represents the corresponding required computer time for the candidate interpolants and both test problems. It can be seen that

p_3 needs less computer time than p_1 and p_2 as it includes less unknown coefficients to identify. Considering both test problems, the required time depends on the number of mesh elements and number of unknowns and is almost independent of the test problem.

	# Mesh Points	# Mesh Elements	*Interpolant*		
			p_1	p_2	p_3
First Test Problem	$4 \times 4 \times 4$	303	1.021	2.193	0.621
	$8 \times 8 \times 8$	3146	10.71	24.68	6.098
	$16 \times 16 \times 16$	26880	118.9	332.2	56.31
	$32 \times 32 \times 32$	219273	3976	10682	1143
Second Test Problem	$4 \times 4 \times 4$	303	1.091	2.003	0.711
	$8 \times 8 \times 8$	3146	10.33	21.87	6.259
	$16 \times 16 \times 16$	26880	119.3	291.8	52.81
	$32 \times 32 \times 32$	219273	3995	10629	1016

Table 2: Total required time (in seconds) for both test problems.

Unfortunately, none of these interpolants are globally continuous along the boundaries of the mesh elements. In fact, they provide continuity on the shared edges, but on the shared faces. Figure 3 shows the contour plots associated with the different interpolants on a tetrahedron mesh with 512 mesh points for the second test problem. The contour plots have been generated by the built-in MATLAB *contour* procedure on a fine grid of size $40 \times 40 \times 40$. The pure tri-cubic generates more suitable results for visualization. As can be seen, tri-quadratic with total degree 3 generates better results rather than tensor product tri-quadratic with total degree 6.

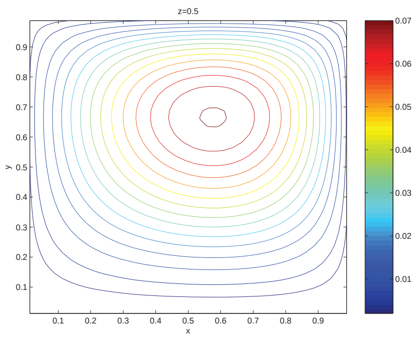
4 Conclusions

We compared three candidate interpolants defined for three-dimensional elliptic PDEs over an unstructured mesh. Test results show that pure tri-cubic interpolant generates more accurate results rather than tri-quadratic interpolants. It is also the best one in terms of visualization.

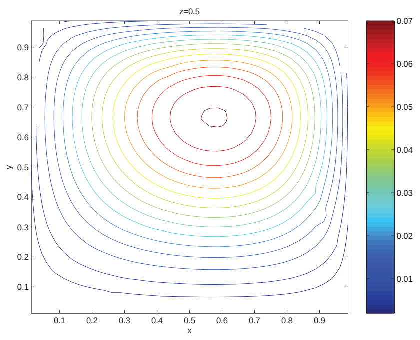
REFERENCES

[1] Wayne H. Enright. Accurate approximate solution of partial differential equations at off-mesh points. *ACM Transaction on Mathematical Software*, 26(2):274–292, June 2000.

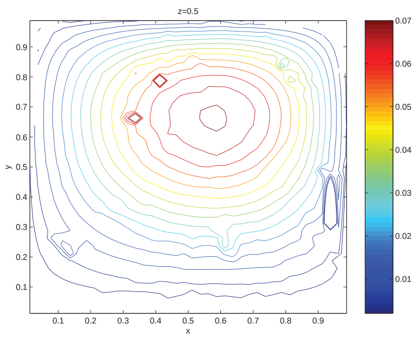
[2] MathWorks. *MATLAB online documentation*, 12 edition.



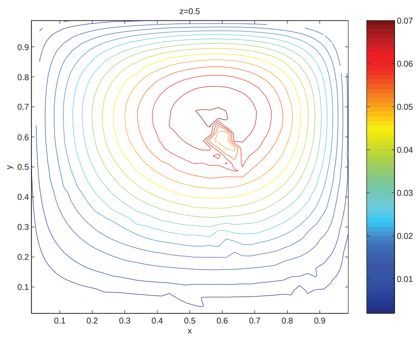
(a) The exact solution



(b) pure tri-cubic (p_1)



(c) tensor product tri-quadratic (p_2)



(d) modified tri-quadratic (p_3)

Figure 3: The contour plots of the exact solution and candidate interpolants for the second test problem on a tetrahedron mesh with 512 mesh points.

[3] Hassan Goldani Moghaddam and Wayne H. Enright. The PCI: A Scattered Data Interpolant For the Solution of Partial Differential Equations. In *Proceedings of International Conference on Adaptive Modeling and Simulation*, ADMOS 2005, Barcelona, Spain, September 2005.