

# Verifying convergence rates of discrete thin-plate splines in 3D.

Linda Stals\*

Stephen Roberts\*

(Received 11 November 2004, revised 29 April 2005)

## Abstract

Traditional thin-plate splines use radial basis functions that produce dense linear system of equations whose size increases with the number of data points. We present a discrete thin-plate spline method that uses polynomials with local support defined on finite-element grids. The resulting system of equations is sparse and its size depends only on the number of nodes in the finite element grid. Theory is developed for general  $d$ -dimensional data sets and model problems are presented in 3D to study the convergence behaviour.

## Contents

### 1 Introduction

C517

---

\*Mathematical Sciences Institute, Australian National University, Canberra, ACT 0200, AUSTRALIA. <mailto:stals@maths.anu.edu.au>

See <http://anziamj.austms.org.au/V46/CTAC2004/Stal> for this article, © Austral. Mathematical Soc. 2005. Published June 19, 2005. ISSN 1446-8735

<i>Contents</i>	C517
<b>2 Discrete thin-plate splines</b>	<b>C518</b>
<b>3 3D interpolation spline</b>	<b>C520</b>
<b>4 Model problems</b>	<b>C522</b>
4.1 Results . . . . .	C523
<b>5 Example applications</b>	<b>C525</b>
<b>6 Future research</b>	<b>C528</b>
<b>References</b>	<b>C528</b>

# 1 Introduction

Thin-plate splines are commonly used for data fitting. The thin-plate spline for a general domain  $\Omega$ , as formulated by Wahba [6] and Duchon [2], is the function,  $f$ , that minimises the functional

$$J_\alpha(f) = \frac{1}{n} \sum_{i=1}^n [f(\mathbf{x}^{(i)}) - y^{(i)}]^2 + \alpha \int_{\Omega} \sum_{|\nu|=2} \binom{2}{\nu} (D^\nu f(\mathbf{x}))^2 d\mathbf{x}, \quad (1)$$

where  $\nu = (\nu_1, \dots, \nu_d)$  is a  $d$ -dimensional multi-index,  $|\nu| = \sum_{s=1}^d \nu_s$ ,  $\mathbf{x}$  is a predictor variable in  $\mathbf{R}^d$ , and  $\mathbf{x}^{(i)}$  and  $y^{(i)}$  are the corresponding  $i$ th predictor and response data value ( $1 \leq i \leq n$ ). This particular formulation is valid for  $d \leq 3$ , but more general formulations are available for higher dimensions using higher order smoothing terms. The smoothing parameter  $\alpha$  controls the trade-off between smoothness and fit. In the limit  $\alpha \rightarrow 0$  the function  $f$  becomes an interpolant. If  $\alpha$  is large,  $f$  becomes very smooth but may not reflect the data very well. The value of  $\alpha$  may be automatically calculated using generalised cross validation [6]. We have implemented this method and are currently preparing a paper that documents the results.

Often radial basis functions are used to represent  $f$  as they give an analytical solution to the minimiser of the functional in Equation (1). See for example [5] and the references mentioned there. The system of equations resulting from the radial basis functions is dense and its size is directly proportional to the number of data points. In applications with a large number of data points the system is very expensive to compute.

Here we propose a discrete thin-plate spline method that uses polynomial basis function with local support defined on a finite element mesh. The resulting system of equations is sparse and its size depends only on the number of grid points in the finite element mesh. A preconditioned conjugate gradient (CG) method is used to solve the system of equations.

This research is an extension of the work presented in [5] and focuses on the formulation of a set of model problems to verify that the code is working correctly and compares the results with the theory given in [4].

In Section 2 a description of the discrete thin-plate spline method is given in  $d$ -dimensions. Section 3 formulates the resulting system of equations in 3D and describes the CG method used to solve the system. A set of model problems is presented in Section 4 and some example applications are discussed in Section 5.

## 2 Discrete thin-plate splines

Standard thin-plate splines require the solution of a dense linear system of equations whose size increases with the number of data points. An alternative approach uses finite element approximation of thin-plate splines [1, 3, 4].

The smoothing problem from Equation (1) is approximated with finite elements so that the discrete smoother  $f$  is a piecewise multi-linear function.

In vector notation  $f$  will be of the form

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}.$$

The idea is to minimise  $J_\alpha$  over all  $f$  of this form. The smoothing term is not defined for piecewise multi-linear functions, but the non-conforming finite element principle is used to introduce piecewise multi-linear functions  $\mathbf{u} = (\mathbf{b}^T \mathbf{g}_1, \dots, \mathbf{b}^T \mathbf{g}_d)$  to represent the gradient of  $f$ . The functions  $f$  and  $\mathbf{u}$  satisfy the relationship

$$\int_{\Omega} \nabla f(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} \mathbf{u}(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \, d\mathbf{x}, \quad (2)$$

for all piecewise multi-linear function  $v$ . This is equivalent to the relationship

$$L\mathbf{c} = \sum_{s=1}^d G_s \mathbf{g}_s, \quad (3)$$

where  $L$  is a discrete approximation to the negative Laplace operator and  $(G_1, \dots, G_d)$  is a discrete approximation to the transpose of the gradient operator.

In this formulation the boundary conditions are assumed to be Dirichlet. If the values along the boundary  $\Gamma_h$  are unknown, then the data is placed well inside the domain, away from the boundary, and  $\mathbf{c}, \mathbf{g}_1, \dots, \mathbf{g}_d$  are set to zero along the boundary. This is consistent with a cubic spline fit with clamped boundary conditions. Section 4 shows some model problems where the values for  $\mathbf{c}, \mathbf{g}_1, \dots, \mathbf{g}_d$  along the boundary are known.

Now consider the minimiser of the functional

$$\begin{aligned} J_\alpha(\mathbf{c}, \mathbf{g}_1, \dots, \mathbf{g}_d) &= \frac{1}{n} \sum_{i=1}^n [\mathbf{b}(\mathbf{x}^{(i)})^T \mathbf{c} - y^{(i)}]^2 \\ &\quad + \alpha \int_{\Omega} \sum_{s=1}^d \nabla(\mathbf{b}^T \mathbf{g}_s) \cdot \nabla(\mathbf{b}^T \mathbf{g}_s) \, d\mathbf{x} \end{aligned}$$

$$= \frac{1}{n} \sum_{i=1}^n [\mathbf{b}(\mathbf{x}^{(i)})^T \mathbf{c} - y^{(i)}]^2 + \alpha \sum_{s=1}^d \mathbf{g}_s^T L \mathbf{g}_s. \quad (4)$$

Our smoothing problem consists of minimising this functional over all vectors  $\mathbf{c}, \mathbf{g}_1, \dots, \mathbf{g}_d$ , defined on the domain  $\Omega_h$ , subject to Constraint (3).

### 3 3D interpolation spline

We now concentrate on the interpolation of 3D data sets.

In the 3D case the discrete minimisation problem (4) is equivalent to finding the minimum of

$$J_\alpha(\mathbf{c}, \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3) = \mathbf{c}^T A \mathbf{c} - 2 \mathbf{d}^T \mathbf{c} + \mathbf{y}^T \mathbf{y} / n + \alpha (\mathbf{g}_1^T L \mathbf{g}_1 + \mathbf{g}_2^T L \mathbf{g}_2 + \mathbf{g}_3^T L \mathbf{g}_3), \quad (5)$$

subject to

$$L \mathbf{c} - G_1 \mathbf{g}_1 - G_2 \mathbf{g}_2 - G_3 \mathbf{g}_3 = 0. \quad (6)$$

The matrices  $L, G_1, G_2$  and  $G_3$  are independent of the data points but the matrix

$$A = \frac{1}{n} \sum_{i=1}^n \mathbf{b}(\mathbf{x}^{(i)}) \mathbf{b}(\mathbf{x}^{(i)})^T,$$

and vector

$$\mathbf{d} = \frac{1}{n} \sum_{i=1}^n \mathbf{b}(\mathbf{x}^{(i)}) y^{(i)},$$

must be assembled by sweeping through the data points. The matrix  $A$  is symmetric indefinite and sparse.

Using Lagrange multipliers, rewrite the Minimisation Problem (5) as the

solution of the linear system:

$$\begin{bmatrix} A & 0 & 0 & 0 & L \\ 0 & \alpha L & 0 & 0 & -G_1^T \\ 0 & 0 & \alpha L & 0 & -G_2^T \\ 0 & 0 & 0 & \alpha L & -G_3^T \\ L & -G_1 & -G_2 & -G_3 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \\ \mathbf{h}_4 \\ \mathbf{h}_5 \end{bmatrix}, \quad (7)$$

where  $\mathbf{w}$  is a Lagrange multiplier associated with Constraint (6). The vectors  $\mathbf{h}_1, \dots, \mathbf{h}_5$  store the Dirichlet boundary information.

One of the advantages of this approach is that the size of the linear system depends on the discretisation size  $m$  instead of the number of the data points  $n$ . All sub-systems in (7) have dimension  $m$ . The time required to assemble the matrix does depend on  $n$  (that is, to build  $A$  and  $\mathbf{d}$ ) but it depends only linearly on  $n$  and the observation data only have to be read from secondary storage once if using a uniform finite element grid.

One way to solve Equation (7) is to eliminate all of the variables except  $\mathbf{g}_1$ ,  $\mathbf{g}_2$  and  $\mathbf{g}_3$ , which gives

$$\begin{bmatrix} \alpha L + G_1^T Z G_1 & G_1^T Z G_2 & G_1^T Z G_3 \\ G_2^T Z G_1 & \alpha L + G_2^T Z G_2 & G_2^T Z G_3 \\ G_3^T Z G_1 & G_3^T Z G_2 & \alpha L + G_3^T Z G_3 \end{bmatrix} \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} = \begin{bmatrix} G_1^T L^{-1} \mathbf{d} \\ G_2^T L^{-1} \mathbf{d} \\ G_3^T L^{-1} \mathbf{d} \end{bmatrix} - \begin{bmatrix} \mathbf{h}_2 + G_1^T L^{-1} \mathbf{h} \\ \mathbf{h}_3 + G_2^T L^{-1} \mathbf{h} \\ \mathbf{h}_4 + G_3^T L^{-1} \mathbf{h} \end{bmatrix}, \quad (8)$$

where matrix  $Z = L^{-1} A L^{-1}$ , vector  $\mathbf{h} = \mathbf{h}_1 - A L^{-1} \mathbf{h}_5$  and vector  $\mathbf{c} = L^{-1} (G_1 \mathbf{g}_1 + G_2 \mathbf{g}_2 + G_3 \mathbf{g}_3 - \mathbf{h}_5)$ .

TABLE 1: Model problem specifications

Model Problem	$f(\mathbf{x})$	$\mathbf{u}_j$	$\lambda$
1	$\ \mathbf{x}\ _2^2$	$2\mathbf{x}_j$	0
2	$\mathbf{x}_1^3 + \mathbf{x}_2^3 + \mathbf{x}_3^3$	$3\mathbf{x}_j^2$	$-6\alpha\ \mathbf{x}\ _1$ .
3	$\ \mathbf{x} + [0.5, 0.5, 0.5]^T\ _2$	$(\mathbf{x}_j + 0.5)/f(\mathbf{x})$	$-2\alpha/f(\mathbf{x})$
4	$\ \mathbf{x} - [0.01, 0.01, 0.01]^T\ _2$	$(\mathbf{x}_j - 0.01)/f(\mathbf{x})$	$-2\alpha/f(\mathbf{x})$

## 4 Model problems

To study the convergence behaviour of the finite element formulation, and verify that the code is working correctly, a set of model test problems were devised.

View the system of equations given in (7) as the discrete form of

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n f(\mathbf{x}^{(i)})\psi(\mathbf{x}^{(i)}) + \int_{\Omega} \nabla\lambda\nabla\psi &= \frac{1}{n} \sum_{i=1}^n y^{(i)}\psi(\mathbf{x}^{(i)}), \quad (9) \\ \alpha \int_{\Omega} \nabla\mathbf{u}_j\nabla v_j + \int_{\Omega} \partial_j\lambda v_j &= 0, \quad \text{for all } j = 1, 2, 3, \\ \int_{\Omega} (\nabla f - \mathbf{u})\nabla v &= 0, \end{aligned}$$

for all  $\psi, v, v_j \in H_0^1$ . We assume that  $f$  and  $\mathbf{u}_j$  have Dirichlet boundary conditions. Functions  $f$  and  $\mathbf{u}$  such that  $\Delta f = 0$ ,  $\mathbf{u} = \Delta f$  with  $y^{(i)} = f(\mathbf{x}^{(i)})$  and  $\lambda = -\alpha\nabla^2 f$  satisfy (9). To find the boundary conditions of  $\mathbf{w}$  needed in (8) we let  $\mathbf{b}(\mathbf{x})^T \mathbf{w}$  be the discrete approximation to  $\lambda$ .

In the following experiments the data points  $\mathbf{x}^{(i)}$  ( $1 \leq i \leq n$ ) sit in the unit cube on the lattice found by dividing the cube into  $16 \times 16 \times 16$  equally spaced sub-cubes.

Table 1 gives a list of model problems with  $\nabla^4 f = 0$ . For these specific examples  $\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{u}_3$ .

TABLE 2: Convergence results for Model Problem 1. The convergence results labelled by (\*) indicates that the error is only due to the iterative solver.

$m$	$\ f(\mathbf{x}) - \widehat{f}(\mathbf{x})\ _{L_2}$	$\log_2(\text{Conv})$	$\ \mathbf{u}_i - \widehat{\mathbf{u}}_i\ _{L_2}$	$\log_2(\text{Conv})$	CG
65	$2.23 \times 10^{-2}$		$5.93 \times 10^{-6}$		3
369	$6.54 \times 10^{-3}$	1.8	$3.23 \times 10^{-6}$	(*)	5
2465	$1.76 \times 10^{-3}$	1.9	$2.73 \times 10^{-7}$	(*)	6
17985	$4.42 \times 10^{-4}$	2.0	$8.45 \times 10^{-7}$	(*)	7
137345	$1.10 \times 10^{-4}$	2.0	$2.17 \times 10^{-7}$	(*)	7

TABLE 3: Convergence results for Model Problem 2.

$m$	$\ f(\mathbf{x}) - \widehat{f}(\mathbf{x})\ _{L_2}$	$\log_2(\text{Conv})$	$\ \mathbf{u}_i - \widehat{\mathbf{u}}_i\ _{L_2}$	$\log_2(\text{Conv})$	CG
65	$3.40 \times 10^{-2}$		$2.44 \times 10^{-2}$		3
369	$1.02 \times 10^{-2}$	1.7	$7.25 \times 10^{-3}$	1.8	6
2465	$2.79 \times 10^{-3}$	1.9	$1.93 \times 10^{-3}$	1.9	6
17985	$7.06 \times 10^{-4}$	2.0	$4.86 \times 10^{-4}$	2.0	7
137345	$1.75 \times 10^{-4}$	2.0	$1.21 \times 10^{-4}$	2.0	7

## 4.1 Results

For all of the model problems the stopping criterion of the CG method was  $10^{-8}$ . The model problems are very smooth and our conclusions are typical for a wide range of  $\alpha$  so we have only tabulated results for  $\alpha = 1$ . The finite element grid size ranges from  $h = 7.07107 \times 10^{-1}$  for 65 grid nodes to  $h = 4.41942 \times 10^{-2}$  for 137345 grid nodes.

Tables 2, 3, 4 and 5 list the  $L_2$  error norm for the model problems given in Table 1. The values labelled  $\widehat{f}$  and  $\widehat{\mathbf{u}}$  are the approximations to  $f$  and  $\mathbf{u}$ .

Referring back to Equation (9), if  $f, \mathbf{u}$  are in  $H_2$  the expected standard finite element convergence rates of  $\mathcal{O}(h^2)$  apply. The results given in the column labelled ‘Conv’ in Tables 2, 3 and 4 show the expected  $\mathcal{O}(h^2)$  convergence rate. In Model Problem 1  $\mathbf{u}$  is a linear function and is represented



TABLE 4: Convergence results for Model Problem 3.

$m$	$\ f(\mathbf{x}) - \widehat{f}(\mathbf{x})\ _{L_2}$	$\log_2(\text{Conv})$	$\ \mathbf{u}_i - \widehat{\mathbf{u}}_i\ _{L_2}$	$\log_2(\text{Conv})$	CG
65	$4.38 \times 10^{-3}$		$1.68 \times 10^{-3}$		3
369	$1.40 \times 10^{-3}$	1.7	$5.64 \times 10^{-4}$	1.6	5
2465	$3.39 \times 10^{-4}$	2.0	$1.64 \times 10^{-4}$	1.8	6
17985	$1.00 \times 10^{-4}$	1.8	$4.25 \times 10^{-5}$	2.0	7
137345	$2.50 \times 10^{-5}$	2.0	$1.07 \times 10^{-5}$	2.0	7

TABLE 5: Convergence results for Model Problem 4. The convergence result labelled (-) indicates that the error is due to a singularity.

$m$	$\ f(\mathbf{x}) - \widehat{f}(\mathbf{x})\ _{L_2}$	$\log_2(\text{Conv})$	$\ \mathbf{u}_i - \widehat{\mathbf{u}}_i\ _{L_2}$	$\log_2(\text{Conv})$	CG
65	$6.60 \times 10^{-3}$		$5.99 \times 10^{-2}$		3
369	$2.72 \times 10^{-3}$	1.3	$9.98 \times 10^{-3}$	2.6	5
2465	$8.96 \times 10^{-4}$	1.6	$1.57 \times 10^{-3}$	2.7	6
17985	$2.49 \times 10^{-4}$	1.9	$6.22 \times 10^{-4}$	1.3	7
137345	$6.57 \times 10^{-5}$	1.9	$6.34 \times 10^{-4}$	(-)	7

exactly by the linear basis functions, the errors reported in Table 2 are from the linear solver.

The column labelled ‘CG’ gives the number of conjugate gradient iterations required to solve the linear system to the given tolerance. For these model problems the solver worked very well and only showed slight dependence on the grid size  $m$ .

Table 5 shows the errors for Model Problem 4. For this example there is a singularity at the point (0.01, 0.01, 0.01) and its effect is seen in the errors for  $\mathbf{u}$ . As  $f \in H_{5/2}$  we expect  $f$  to still demonstrate second order accuracy, but  $\widehat{\mathbf{u}}$  should converge to a projection of  $\mathbf{u}$  onto a smoother space  $H_2$  (at least away from the data points). Hence we do not expect the discrete gradient to converge to  $\mathbf{u}$ . An important point to note is that our finite element grid does not have a node at the point (0.01, 0.01, 0.01).

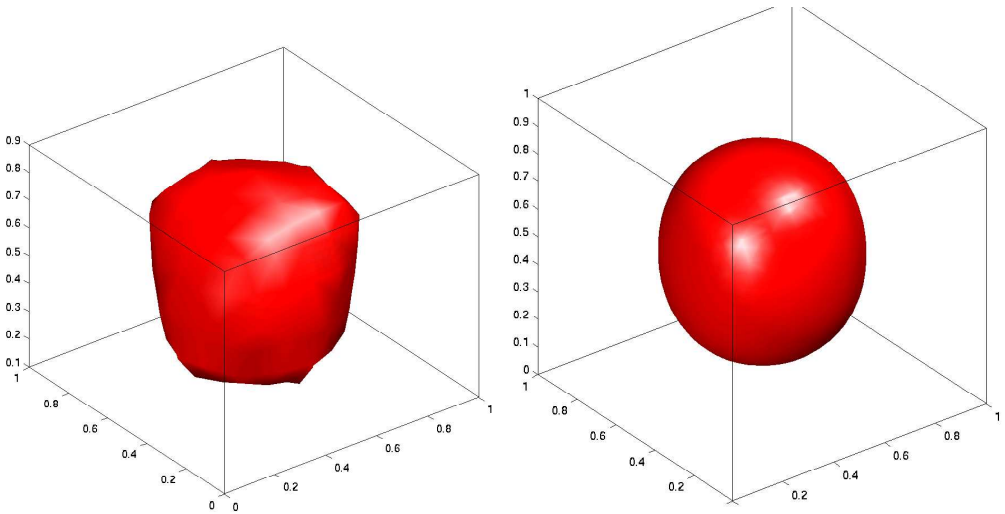


FIGURE 1: Iso-surface plot of the sphere on a grid with 189 and 68705 nodes.

## 5 Example applications

We now briefly present some examples of data fitting in 3D. For these examples  $f$  is unknown so the data is placed well within the interior of the domain and the boundary conditions assumed to be zero. The tolerance used for the stopping criterion for the linear solver is  $10^{-3}$ . A fairly high tolerance is chosen for these examples as we are only interested in the visual output. The values of  $\alpha$  were chosen by trial and error.

The first data set contains  $10^6$  points generated randomly on a sphere with centre  $(0.5, 0.5, 0.5)$  and radius  $1/3$ . The data points are assigned a value of 1. The smoothing parameter  $\alpha$  is set to  $10^{-3}$ . Figure 1 shows an iso-surface plot of the discrete thin plate spline with 189 nodes and 68705 nodes respectively.

The next data set is designed to represent the case where there is missing

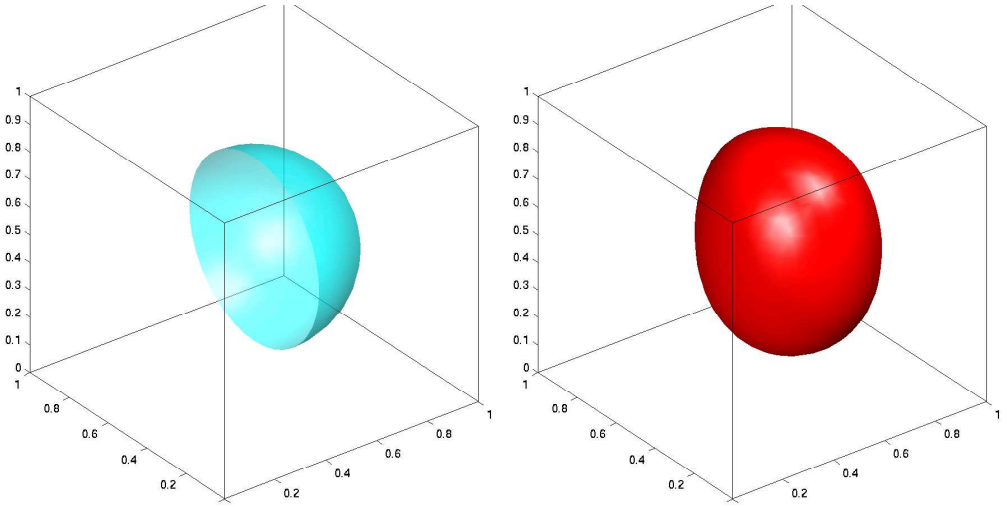


FIGURE 2: Iso-surface representation of the original data set for the semi-sphere example and approximation on a grid with 68705 nodes.

data. Specifically, we take the data from the example described above and remove all of those points with  $x$ -coordinate less than 0.5. See Figure 2. The data set contains approximately  $1/2 \times 10^6$  points. Note that the spline attempts to fill in the missing data. The value for  $\alpha$  is set to  $10^{-3}$ .

In the final example an ‘inner layer’ of data points are added to force the discrete thin plate spline approximation to more closely follow the shape of the semi-sphere. Along with the data points from the previous example, additional data points sitting on the semi-sphere with centre  $(0.5, 0.5, 0.5)$  and radius  $1/6$  are added and given a value of 0. See Figure 3. There are approximately  $2 \times (1/2 \times 10^6)$  data points. The smoothing parameter  $\alpha$  is reduced to  $10^{-7}$ .

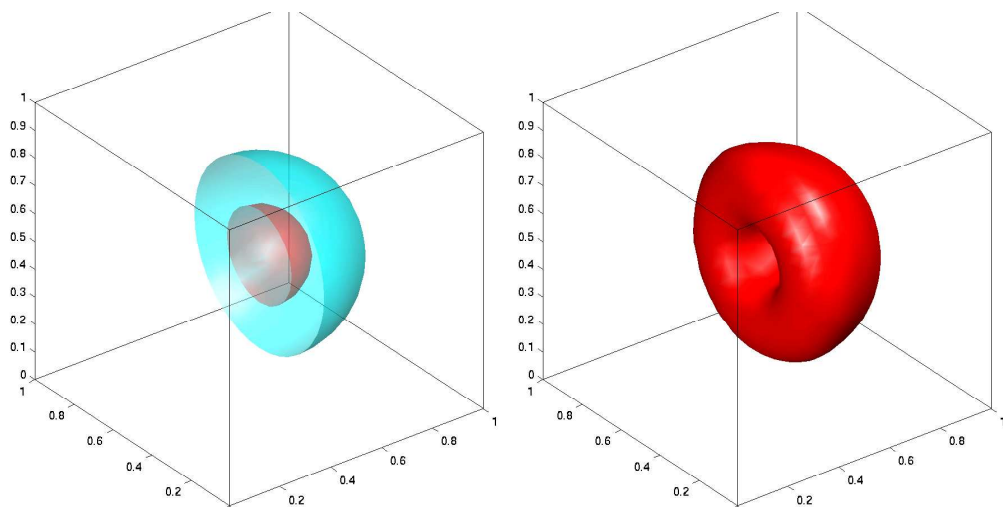


FIGURE 3: Iso-surface representation of the original data set for the semi-sphere example with an additional layer of data points placed in the interior and the approximation on a grid with 68705 nodes.

## 6 Future research

Other options that we wish to explore include: the use of higher-order finite element basis functions, adaptive grid refinement, different solution techniques, automatic calculation of  $\alpha$  and different boundary conditions.

## References

- [1] P. Christen, M. Hegland, O. Nielsen, S. Roberts, P. Strazdins, and I. Altas, Scalable parallel algorithms for surface fitting and data mining, *Parallel Computing*, 27 (2001), pp. 941–961. C518
- [2] J. Duchon, Splines minimizing rotation-invariant, in *Lecture Notes in Math*, vol. 571, Springer-Verlag, 1977, pp. 85–100. C517
- [3] M. Hegland, S. Roberts, and I. Altas, Finite element thin plate splines for data mining applications, in *Mathematical methods for curves and surfaces, II* (Lillehammer, 1997), Innov. Appl. Math., Vanderbilt Univ. Press, Nashville, TN, 1998, pp. 245–252. C518
- [4] S. Roberts, M. Hegland, and I. Altas, Approximation of a thin plate spline smoother using continuous piecewise polynomial functions, *SIAM J. Numer. Anal.*, 41 (2003), pp. 208–234. [Online] <http://epubs.siam.org/sam-bin/dbq/article/38329>. C518
- [5] S. Roberts and L. Stals, Discrete thin plate spline smoothing in 3D, in *Proc. of 11th Computational Techniques and Applications Conference CTAC-2003*, J. Crawford and A. J. Roberts, eds., vol. 45, July 2004, pp. C646–C659. [Online] <http://anziamj.austms.org.au/V45/CTAC2003/Rob2/home.html> [July 18, 2004]. C518

- [6] G. Wahba, Spline models for observational data, vol. 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.  
C517