Discrete Thin-Plate Splines for Large Data Sets

Linda Stals Steve Roberts

Department of Mathematics
Australian National University

July 13, 2006

Thin Plate Splines
- Smoothing Splines
- Radial Basis Functions

Discrete Thin Plate Splines
- Finite Element Approximation
- System of Equations

Convergence Analysis
- Dirichlet Boundary Conditions
- Finite Element Convergence
- Interpolation Error

Results
- Holes
- 3D Examples

Future Work

THE AUSTRALIAN NATIONAL UNIVERSITY

Thin-Plate Splines

- 3D Image Recovery
- Finger Print Analysis
- Image Warping
- Medical Image Analysis
- Data Mining

Thin-Plate Splines

- 3D Image Recovery
- Finger Print Analysis
- Image Warping
- Medical Image Analysis
- Data Mining

Smoothing Splines

Given a set of attributes vectors $\mathbf{x} = (x_1, x_2, \cdots, x_d)^T$, build a predictive model

$$\mathbf{y} = f(\mathbf{x}).$$

$$\mathbf{y} \approx f(\mathbf{x}).$$

To estimate f by a 2nd-order smoothing spline minimise:

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

The first term penalises lack of fit, the second penalises roughness.

Smoothing Splines

Given a set of attributes vectors $\mathbf{x} = (x_1, x_2, \cdots, x_d)^T$, build a predictive model

$$\mathbf{y} \approx f(\mathbf{x}).$$

To estimate f by a 2nd-order smoothing spline minimise:

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

The first term penalises lack of fit, the second penalises roughness.

Smoothing Splines

Given a set of attributes vectors $\mathbf{x} = (x_1, x_2, \cdots, x_d)^T$, build a predictive model

$$\mathbf{y} \approx f(\mathbf{x}).$$

To estimate f by a 2nd-order smoothing spline minimise:

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

The first term penalises lack of fit, the second penalises roughness.

Smoothing Splines

Given a set of attributes vectors $\mathbf{x} = (x_1, x_2, \cdots, x_d)^T$, build a predictive model

$$\mathbf{y} \approx f(\mathbf{x}).$$

To estimate f by a 2nd-order smoothing spline minimise:

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

The first term penalises lack of fit, the second penalises roughness.

Radial Basis Functions

The standard approach is to represent f as a linear combination of radial basis functions

$$f(\mathbf{x}) = \sum_{k=1}^{M} a\phi_k(\mathbf{x}) + \alpha \sum_{i=1}^{n} w_i U(\mathbf{x}, \mathbf{x}^{(i)}),$$

where ϕ_k are monomials of order up to 1 and U are suitable radial basis functions.

Favoured method as it gives an analytical solution.

Radial Basis Functions

The standard approach is to represent f as a linear combination of radial basis functions

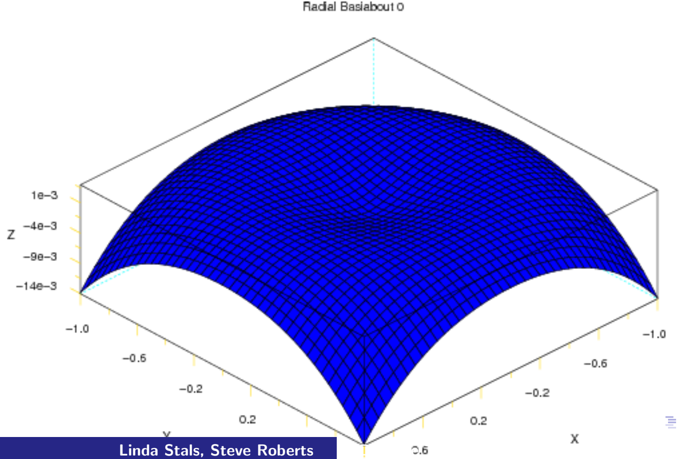
$$f(\mathbf{x}) = \sum_{k=1}^{M} a\phi_k(\mathbf{x}) + \alpha \sum_{i=1}^{n} w_i U(\mathbf{x}, \mathbf{x}^{(i)}),$$

where ϕ_k are monomials of order up to 1 and U are suitable radial basis functions.
Favoured method as it gives an analytical solution.

Radial Basis Functions

2D Eg:

$$U(\mathbf{x}, \mathbf{x}^{(i)}) = \frac{-1}{16\pi} r^2 ln(r).$$



Thin Plate Splines

- Requires a solution of a **dense** system of matrices.
- System may be ill-conditioned.
- **Size increases with the number of data points.**

Not practical for large data sets.

Thin Plate Splines

- Requires a solution of a **dense** system of matrices.
- System may be ill-conditioned.
- **Size increases with the number of data points.**

Not practical for large data sets.

Finite Element Approximation

Represent f as a linear combination of linear finite elements.
In vector notation f will be of the form

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

Minimise J_α over all f of this form

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

Finite Element Approximation

Represent f as a linear combination of linear finite elements.
In vector notation f will be of the form

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

Minimise J_α over all f of this form

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

THE AUSTRALIAN NATIONAL UNIVERSITY

Finite Element Approximation

Represent f as a linear combination of linear finite elements. In vector notation f will be of the form

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

Minimise J_α over all f of this form

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

Non-Conforming Finite elements

The smoothing term (derivatives) is not defined for piecewise multi-linear functions.

Use non-conforming finite elements.

Represent the gradient of f by $\mathbf{u} = (\mathbf{b}^T\mathbf{g}_1, ..., \mathbf{b}^T\mathbf{g}_d)$ where

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

for all piecewise multi-linear function v.

Non-Conforming Finite elements

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

is equivalent to

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

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

Non-Conforming Finite elements

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

is equivalent to

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

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

Finite Element Approximation

2nd-order smoothing spline: minimise

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

Finite element approximation: minimise

$$J_\alpha(c, g_1, g_2, \cdots, g_d) = \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}^{(i)}) - y^{(i)})^2 + \alpha \sum_{s=1}^{d} \mathbf{g}_s^T L \mathbf{g}_s,$$

subject to

$$L\mathbf{c} = \sum_{s=1}^{d} G_s \mathbf{g}_s.$$

Finite Element Approximation

2nd-order smoothing spline: minimise

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

Finite element approximation: minimise

$$J_\alpha(c, g_1, g_2, \cdots, g_d) = \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}^{(i)}) - y^{(i)})^2 + \alpha \sum_{s=1}^{d} \mathbf{g}_s^T L \mathbf{g}_s,$$

subject to

$$L\mathbf{c} = \sum_{s=1}^{d} G_s \mathbf{g}_s.$$

2D Formulation

2nd-order smoothing spline: minimise

$$
\begin{aligned}
J_\alpha(f) &= \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}^{(i)}) - y^{(i)})^2 \\
&+ \alpha \int_\Omega \left(\left(\partial_1^2 f(\mathbf{x})\right)^2 + 2\left(\partial_1\partial_2 f(\mathbf{x})\right)^2 + \left(\partial_2^2 f(\mathbf{x})\right)^2 \right) d\mathbf{x},
\end{aligned}
$$

$$
\begin{aligned}
J_\alpha(\mathbf{c}, \mathbf{g}_1, \mathbf{g}_2) &= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{b}(\mathbf{x}^{(i)})^T\mathbf{c} - y^{(i)})^2 \\
&+ \alpha \int_\Omega \nabla\mathbf{b}^T(\mathbf{x})\mathbf{g}_1.\nabla\mathbf{b}^T(\mathbf{x})\mathbf{g}_1 + \nabla\mathbf{b}^T(\mathbf{x})\mathbf{g}_2.\nabla\mathbf{b}^T(\mathbf{x})\mathbf{g}_2 \, d\mathbf{x}
\end{aligned}
$$

2D Formulation

2nd-order smoothing spline: minimise

$$J_\alpha(f) = \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}^{(i)}) - y^{(i)})^2 + \alpha\int_\Omega \left(\left(\partial_1^2 f(\mathbf{x})\right)^2 + 2\left(\partial_1\partial_2 f(\mathbf{x})\right)^2 + \left(\partial_2^2 f(\mathbf{x})\right)^2\right) d\mathbf{x},$$

$$J_\alpha(\mathbf{c}, \mathbf{g}_1, \mathbf{g}_2) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{b}(\mathbf{x}^{(i)})^T\mathbf{c} - y^{(i)})^2 + \alpha\int_\Omega \nabla\mathbf{b}^T(\mathbf{x})\mathbf{g}_1.\nabla\mathbf{b}^T(\mathbf{x})\mathbf{g}_1 + \nabla\mathbf{b}^T(\mathbf{x})\mathbf{g}_2.\nabla\mathbf{b}^T(\mathbf{x})\mathbf{g}_2\ d\mathbf{x}$$

2D Formulation

Minimise:

$$J_\alpha(c, g_1, g_2) = \mathbf{c}^T A \mathbf{c} - 2\mathbf{d}^T \mathbf{c} + \|\mathbf{y}\|^2/n + \alpha(\mathbf{g}_1^T L \mathbf{g}_1 + \mathbf{g}_2^T L \mathbf{g}_2)$$

subject to

$$L\mathbf{c} = G_1\mathbf{g}_1 + G_2\mathbf{g}_2.$$

Where

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

and

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

Discrete System

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

$\mathbf{w}$ is a Lagrange multiplier.

The vectors $\mathbf{h}_1, \cdots, \mathbf{h}_4$ store the boundary information.

Conjugate Gradient

Eliminate all the variables except $\mathbf{g}_1$ and $\mathbf{g}_2$ to give

$$\left(\begin{bmatrix}\alpha L & 0\\ 0 & \alpha L\end{bmatrix} + \begin{bmatrix}G_1^T\\ G_2^T\end{bmatrix} L^{-1} A L^{-1}\begin{bmatrix}G_1 & G_2\end{bmatrix}\right)\begin{bmatrix}\mathbf{g}_1\\ \mathbf{g}_2\end{bmatrix} = \begin{bmatrix}G_1^T L^{-1}\mathbf{d}\\ G_2^T L^{-1}\mathbf{d}\end{bmatrix} - \begin{bmatrix}\widehat{\mathbf{h}}_2\\ \widehat{\mathbf{h}}_3\end{bmatrix},$$

$$\alpha \mathrm{diag}(L) + K^T K\mathbf{g} = \widehat{\mathbf{d}}$$

$$\mathbf{c} = L^{-1}\left(G_1\mathbf{g}_1 + G_2\mathbf{g}_2 - \widehat{\mathbf{h}}_4\right).$$

Conjugate Gradient

Eliminate all the variables except $\mathbf{g}_1$ and $\mathbf{g}_2$ to give

$$\left(\begin{bmatrix}\alpha L & 0\\ 0 & \alpha L\end{bmatrix} + \begin{bmatrix}G_1^T\\ G_2^T\end{bmatrix} L^{-1}AL^{-1}\begin{bmatrix}G_1 & G_2\end{bmatrix}\right)\begin{bmatrix}\mathbf{g}_1\\ \mathbf{g}_2\end{bmatrix} = \begin{bmatrix}G_1^T L^{-1}\mathbf{d}\\ G_2^T L^{-1}\mathbf{d}\end{bmatrix} - \begin{bmatrix}\widehat{\mathbf{h}}_2\\ \widehat{\mathbf{h}}_3\end{bmatrix},$$

$$\alpha \mathrm{diag}(L) + K^T K\mathbf{g} = \widehat{\mathbf{d}}$$

$$\mathbf{c} = L^{-1}\left(G_1\mathbf{g}_1 + G_2\mathbf{g}_2 - \widehat{\mathbf{h}}_4\right).$$

Preconditioned Conjugate Gradient

Current preconditioner

$$M = \begin{bmatrix} L^{-1} & 0 \\ 0 & L^{-1} \end{bmatrix}.$$

- large α: works well
- small α: help.

Preconditioned Conjugate Gradient

Current preconditioner

$$M = \begin{bmatrix} L^{-1} & 0 \\ 0 & L^{-1} \end{bmatrix}.$$

- large α: works well
- small α: help.

Lagrange Multiplier

Recall the discrete system

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

where $\mathbf{w}$ is a Lagrange multiplier. We use Dirichlet boundary conditions as L^{-1} is unique, although Neumann is also possible. What is the Dirichlet boundary value for $\mathbf{w}$?

Lagrange Multiplier

Recall the discrete system

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

where $\mathbf{w}$ is a Lagrange multiplier. We use Dirichlet boundary conditions as L^{-1} is unique, although Neumann is also possible.
What is the Dirichlet boundary value for $\mathbf{w}$?

Lagrange Multiplier

Recall the discrete system

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

where $\mathbf{w}$ is a Lagrange multiplier. We use Dirichlet boundary conditions as L^{-1} is unique, although Neumann is also possible. What is the Dirichlet boundary value for $\mathbf{w}$?

Dirichlet Boundary Conditions

Use Karush-Kuhn-Tucker (KKT) condition with calculus of variations to rewrite weak finite element equations into a system of strong equations.
Then

$$\Delta\widetilde{\lambda}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{f}(\mathbf{x}) - y^{(i)}\right)\delta(\mathbf{x} - \mathbf{x}^{(i)}) \quad \text{in } \Omega,$$

$$-\alpha\Delta\widetilde{u}_1(\mathbf{x}) = \partial_1\widetilde{\lambda}(\mathbf{x}) \quad \text{in } \Omega,$$

$$-\alpha\Delta\widetilde{u}_2(\mathbf{x}) = \partial_2\widetilde{\lambda}(\mathbf{x}) \quad \text{in } \Omega,$$

$$\Delta\widetilde{f}(\mathbf{x}) = \nabla.\widetilde{\mathbf{u}}(\mathbf{x}) \quad \text{in } \Omega.$$

$\widetilde{f}$ = minimiser, $\widetilde{u}$ = gradient, $\widetilde{\lambda}$ = lagrange multiplier.

Dirichlet Boundary Conditions

$$\Delta\widetilde{\lambda}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{f}(\mathbf{x}) - y^{(i)}\right)\delta(\mathbf{x} - \mathbf{x}^{(i)}),$$

$$-\alpha\Delta\widetilde{u}_1(\mathbf{x}) = \partial_1\widetilde{\lambda}(\mathbf{x}),$$

$$-\alpha\Delta\widetilde{u}_2(\mathbf{x}) = \partial_2\widetilde{\lambda}(\mathbf{x}),$$

$$\Delta\widetilde{f}(\mathbf{x}) = \nabla.\widetilde{\mathbf{u}}(\mathbf{x}).$$

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

Dirichlet Boundary Conditions

Use Karush-Kuhn-Tucker (KKT) condition with calculus of variations to rewrite weak finite element equations into a system of strong equations.
Then

$$
\begin{aligned}
\Delta\Delta\widetilde{f}(\mathbf{x}) &= \Delta\nabla.\widetilde{\mathbf{u}}(\mathbf{x}) \\
&\vdots \\
&= \frac{-1}{\alpha}\frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{f}(\mathbf{x}) - y(\mathbf{x})\right)\delta(\mathbf{x}-\mathbf{x}^{(i)}).
\end{aligned}
$$

Conclusion: Boundary conditions do not matter, always get a minimiser.

Dirichlet Boundary Conditions

Use Karush-Kuhn-Tucker (KKT) condition with calculus of variations to rewrite weak finite element equations into a system of strong equations.
Then

$$\begin{aligned} \Delta\Delta\widetilde{f}(\mathbf{x}) &= \Delta\nabla.\widetilde{\mathbf{u}}(\mathbf{x}) \\ &\vdots \\ &= \frac{-1}{\alpha}\frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{f}(\mathbf{x}) - y(\mathbf{x})\right)\delta(\mathbf{x}-\mathbf{x}^{(i)}). \end{aligned}$$

Conclusion: Boundary conditions do not matter, always get a minimiser.

Dirichlet Boundary Conditions

Use Karush-Kuhn-Tucker (KKT) condition with calculus of variations to rewrite weak finite element equations into a system of strong equations.

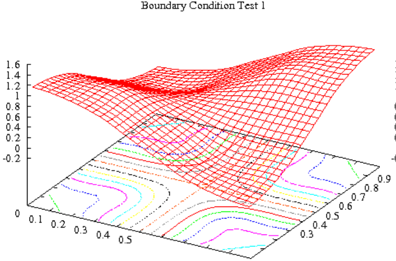
Then

$$
\begin{aligned}
\Delta\Delta\widetilde{f}(\mathbf{x}) &= \Delta\nabla.\widetilde{\mathbf{u}}(\mathbf{x}) \\
&\vdots \\
&= \frac{-1}{\alpha}\frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{f}(\mathbf{x}) - y(\mathbf{x})\right)\delta(\mathbf{x}-\mathbf{x}^{(i)}).
\end{aligned}
$$

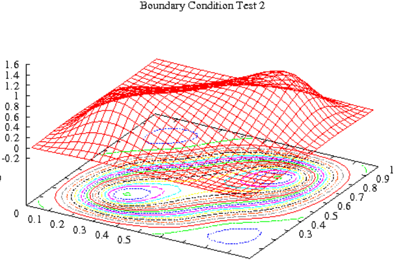
Conclusion: Boundary conditions do not matter, always get a minimiser.

Boundary Condition Examples

$\mathbf{x}^{(1)} = (0.25, 0.25)$, $\mathbf{x}^{(2)} = (0.75, 0.25)$, $\mathbf{x}^{(3)} = (0.25, 0.75)$, $\mathbf{x}^{(4)} = (0.75, 0.75)$; $y^{(1)} = 1$, $y^{(2)} = 0$, $y^{(3)} = 0$ and $y^{(4)} = 1$



$h_f(\mathbf{x}) = \text{tps fit.}$



$h_f(\mathbf{x}) = 0.$

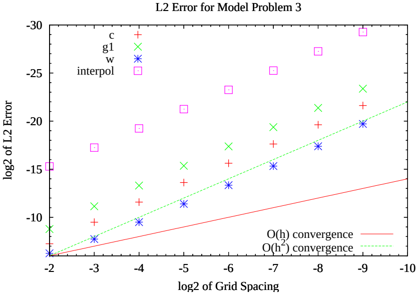
$\mathbf{h}_u = \nabla h_f(\mathbf{x})$, $h_\lambda(\mathbf{x}) = -\alpha \Delta h_f$

THE AUSTRALIAN NATIONAL UNIVERSITY

Convergence on Smooth Problem

$$y^{(i)} = \widetilde{f}_y(\mathbf{x}^{(i)}) \text{ where } \nabla^4 \widetilde{f}_y = 0.$$

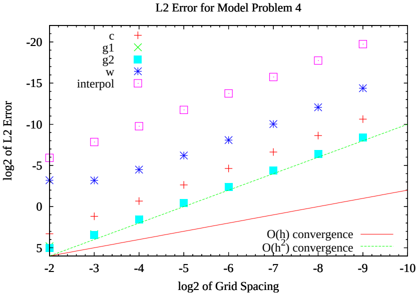
$$\widetilde{f}(\mathbf{x}) = \widetilde{f}_y(\mathbf{x}) = \left\| \mathbf{x} + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right\|_2$$



Convergence on Smooth Problem

$$y^{(i)} = \widetilde{f}_y(\mathbf{x}^{(i)}) \text{ where } \nabla^4 \widetilde{f}_y = 0.$$

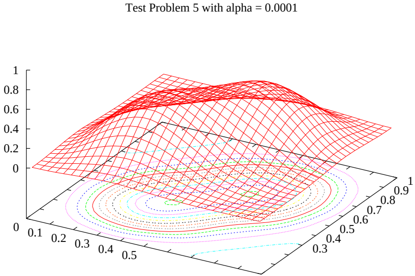
$$\widetilde{f}(\mathbf{x}) = \widetilde{f}_y(\mathbf{x}) = \cosh(2\pi x_1)\sin(2\pi x_2)$$

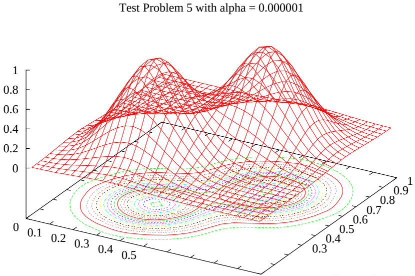


Further Model Problems - Exponential

$$\exp\left(-30\|0.65 - \mathbf{x}\|_2^2\right) + \exp\left(-30\|0.35 - \mathbf{x}\|_2^2\right)$$

Finite element grid of size $m = 4225$ with different values of α.



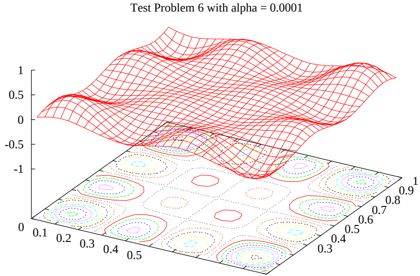


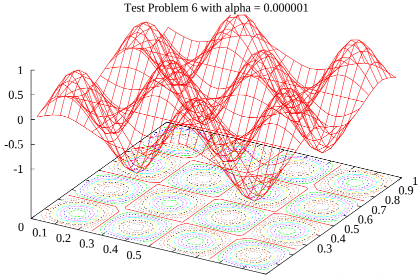
Boundary conditions: $h_f = \widetilde{f}_y$, $\mathbf{h}_u = \nabla \widetilde{f}_y$ and $h_\lambda = -\alpha \Delta \widetilde{f}_y$.

Further Model Problems - Sin

$$\sin(4\pi x_1)\sin(4\pi x_2)$$

Finite element grid of size $m = 4225$ with different values of α.





Boundary conditions: $h_f = \widetilde{f}_y$, $\mathbf{h}_u = \nabla\widetilde{f}_y$ and $h_\lambda = -\alpha\Delta\widetilde{f}_y$.

Interpolation Error

Finite element analysis shows that the interpolation error is given by

$$\sqrt{(k^4+\alpha)\|f_0\|^2_{H^2} + h^{2m}\|f\|^2_{H^m} + \frac{C\sigma^2}{n\alpha^{d/(2m)}}},$$

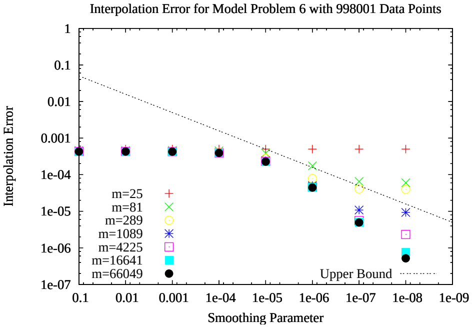
where

- $y^i = f_0(x^i) + \epsilon^i$, $\mathcal{E}(\epsilon) = 0$, s.d. σ,
- h is the grid size,
- k is spacing between data points, uniform spacing, no holes,
- d is the dimension,
- C is a constant.

Example Interpolation Error - No Noise

$$\sin(4\pi x_1)\sin(4\pi x_2)$$

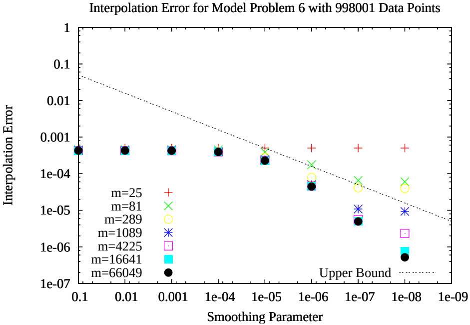
$$\sqrt{(k^4+\alpha)\|f_0\|_{H^2}^2 + h^{2m}\|f\|_{H^m}^2 + \frac{C\sigma^2}{n\alpha^{d/(2m)}}},$$



Example Interpolation Error - No Noise

$$\sin(4\pi x_1)\sin(4\pi x_2)$$

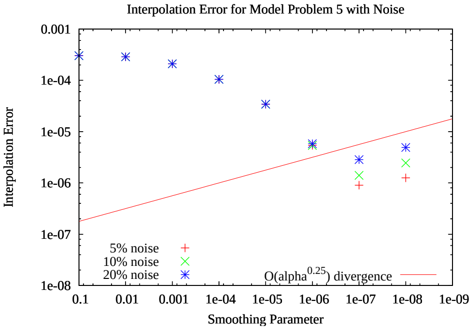
$$\sqrt{(k^4+\alpha)\|f_0\|^2_{H^2} + h^{2m}\|f\|^2_{H^m} + \frac{C\sigma^2}{n\alpha^{d/(2m)}}},$$



Example Interpolation Error - Noise

$$\exp\left(-30\|0.65-\mathbf{x}\|_2^2\right)+\exp\left(-30\|0.35-\mathbf{x}\|_2^2\right)$$

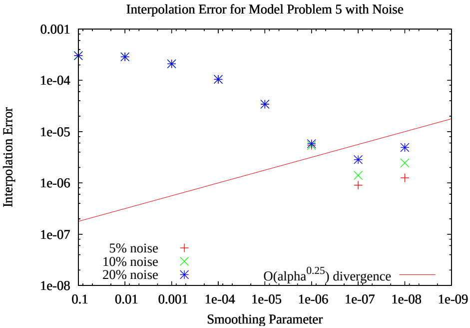
$$\sqrt{(k^4+\alpha)\|f_0\|_{H^2}^2+h^{2m}\|f\|_{H^m}^2+\frac{C\sigma^2}{n\alpha^{d/(2m)}}},$$



Example Interpolation Error - Noise

$$\exp\left(-30\|0.65-\mathbf{x}\|_2^2\right)+\exp\left(-30\|0.35-\mathbf{x}\|_2^2\right)$$

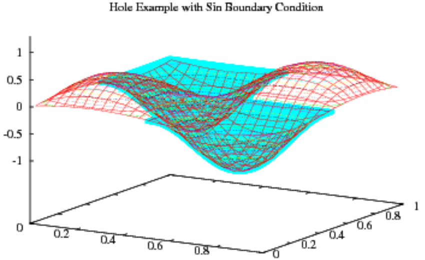
$$\sqrt{(k^4+\alpha)\|f_0\|_{H^2}^2+h^{2m}\|f\|_{H^m}^2+\frac{C\sigma^2}{n\alpha^{d/(2m)}}},$$



Sine Example with Holes

$$y(x, y) = \sin(2\pi x)\sin(2\pi y), \text{ such that } y(x, y) < 0.$$

$n = 179401$, $m = 4229$ with $\alpha = 10^{-6}$

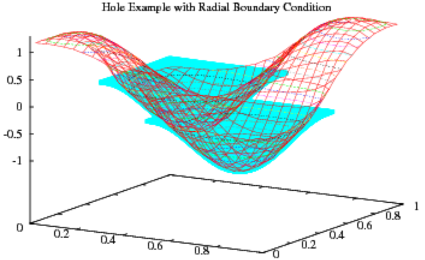


Boundary: $h_f(\mathbf{x}) = y(\mathbf{x})$, $\mathbf{h}_u = \nabla h_f(\mathbf{x})$, $h_\lambda(\mathbf{x}) = -\alpha \Delta h_f$.

Sine Example with Holes

$$y(x,y) = \sin(2\pi x)\sin(2\pi y), \text{ such that } y(x,y) < 0.$$

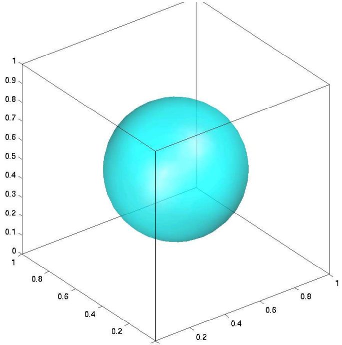
$n = 179401,\ m = 4229$ with $\alpha = 10^{-6}$

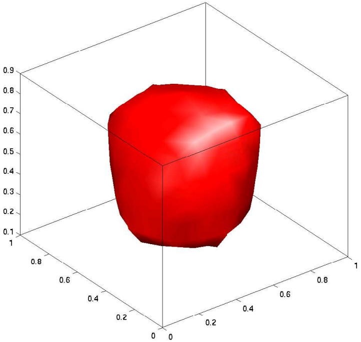


Boundary: $h_f(\mathbf{x}) =$ tps fit, $\mathbf{h}_u = \nabla h_f(\mathbf{x})$, $h_\lambda(\mathbf{x}) = -\alpha \Delta h_f$.

Sphere Example

Grid - 189 Nodes, $\alpha = 10^{-3}$.

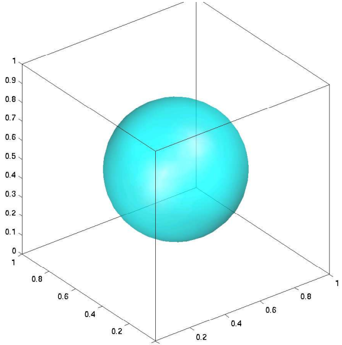


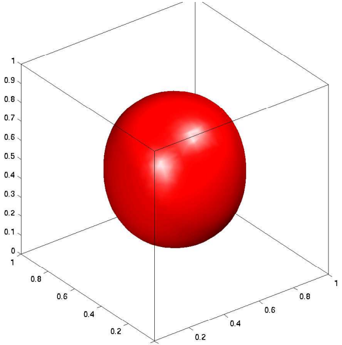


THE AUSTRALIAN NATIONAL UNIVERSITY

Sphere Example

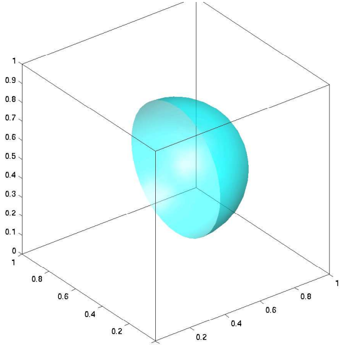
Grid - 68705 Nodes, $\alpha = 10^{-3}$.

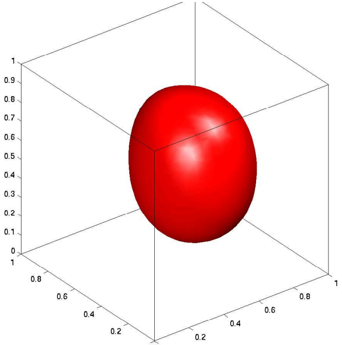




Semi Sphere Example

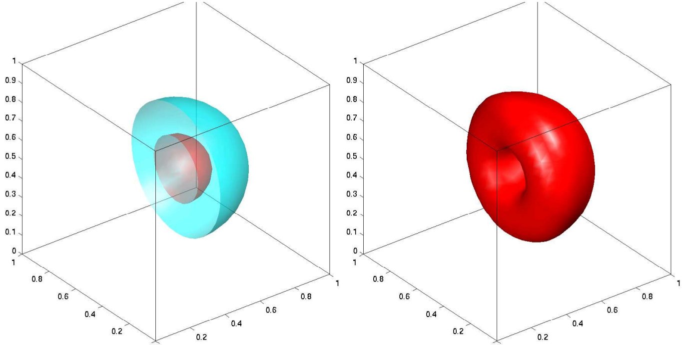
Grid - 68705 Nodes, $\alpha = 10^{-3}$.





Two Sphere Example

Grid - 68705 Nodes, $\alpha = 10^{-7}$.



What Now?

- Adaptive Refinement: Reduce number of times data has to be read.
- Parallel Implementation: Grid v's data.
- Preconditioners for Small α:
- Higher Dimensions: Hierarchical, sparse grids.
- Finite Element Formulation: Linear operators, different smoothers, different norms.
- Holes: Include a-prior information.