# Using multidimensional scaling with Duchon splines for reliable finite area smoothing

David Lawrence Miller

Mathematical Sciences
University of Bath

Recent advances in spatial statistics in ecology
5 May 2011
St Andrews

# Outline

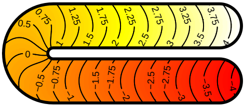
# Outline

# Smoothing in 2 dimensions

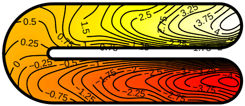
- Have some geographical region and wish to find out something about the biological population in it.
- Response is eg. animal distribution, wish to predict based on $(x, y)$ and other covariates eg. habitat, size, sex, etc.
- This problem is relatively easy if the domain is simple.

# Leakage (or, “whales don’t live in glaciers”)

- Smoothing of complex domains makes this a lot more difficult.
- Problem of leakage.
- Wrong metric.
- Euclidean distance doesn’t always make sense.



(modified) Ramsay test function



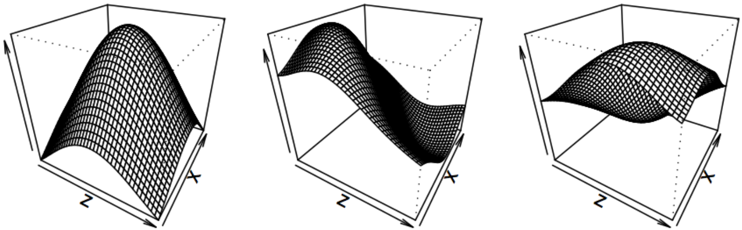
Thin plate spline fit

# Smoothing with penalties

- Do smoothing with splines in an additive model framework.
- Take some linear combination of (known) basis functions.
- Penalize based on integral of second squared derivative:

$$\int\int_{\mathbf{R}^2} \left( \left( \frac{\partial^2 f(x_1, x_2)}{\partial x_1^2} \right)^2 + \left( \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} \right)^2 + \left( \frac{\partial^2 f(x_1, x_2)}{\partial x_2^2} \right)^2 \right) \mathrm{d}x_1 \mathrm{d}x_2.$$

- Using thin plate regression splines (eg Wood (2003)).



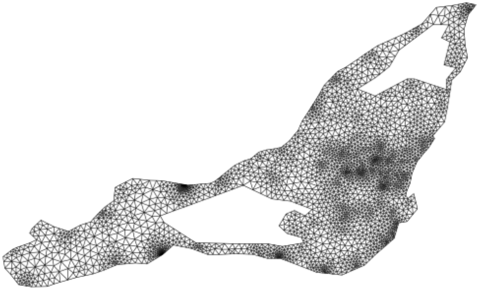
# Proposed solutions to leakage problems

Two classes:

- PDEs:
  - FELSPLINE (Ramsay, 2002).
  - Soap film smoothers (Wood *et al*, 2008).
- Within-area distance
  - Geodesic Low-Rank Thin Plate Splines (GLTPS) (Wang and Ranalli, 2007).

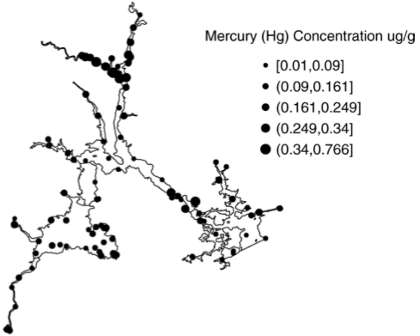
# FELSPLINE (Ramsay, 2002)

- Triangulate domain.
- Setup PDEs on triangles.
- Solve PDE system $\Rightarrow$ solve smoothing problem.
- BAD: unrealistic boundary condition.



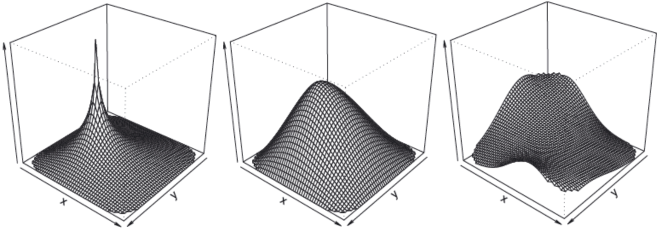
# Geodesic Low-Rank Thin Plate Splines (GLTPS) (Wang and Ranalli, 2007).

- Form the complete graph ($G$) on the data.
- Run $k-$NN, to create the reduced graph ($G_k$).
- Use Floyd's algorithm to find distances ($O(n^3)$).
- BAD: few data $\Rightarrow$ inaccurate distances.
- BAD: many data $\Rightarrow$ slow!



# Soap film smoothers (Wood *et al*, 2008)

- Coat hangers, soapy water, PDEs.
- Solve PDEs (Laplace + Poisson) $\Rightarrow$ basis functions.
- Cyclic spline gives boundary conditions.
- BAD: mathematically complex.

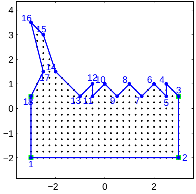


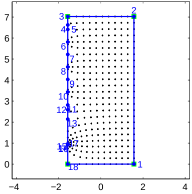
# Domain morphing

- Domain morphing - think of domain as silly putty.
- Takes into account within-area distance.
- Potentially less computationally intensive.
- Eilers (2006) proposes Schwarz-Christoffel transform.
- Gives a known domain - easier to smooth over.

**However:**

- Crowding - numerical singularities.
- Not clear what this does to the smoothness penalty - artefacts, penalty issues - but NOT GOOD!



# Outline

# Multidimensional scaling and within-area distances

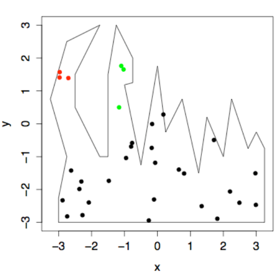
- Idea: use MDS to arrange points in the domain according to their distance within the domain.

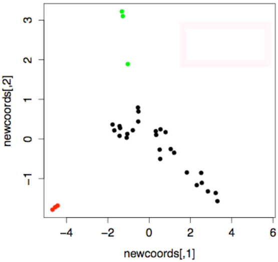
**Scheme:**

- First need to find the within-area distances.
- Perform MDS on the matrix of within-area distances.
- Find the new configuration of the points.
- Smooth over the new points.

# Multidimensional scaling refresher

- Use double centred matrix of between point distances, $D$, using the eigen-decomposition of $DD^T$ we can find new points.
- Finds a projection of points such that Euclidean distance between points in new arrangement is approximately the same as the distances in $D$.
- New points can be added into the MDS configuration via Gower's interpolation (Gower, 1968)

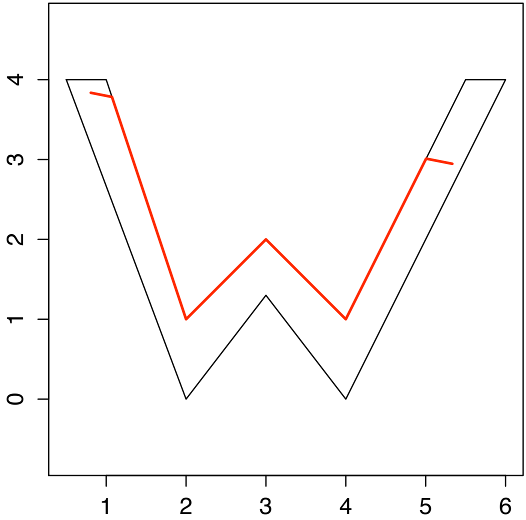


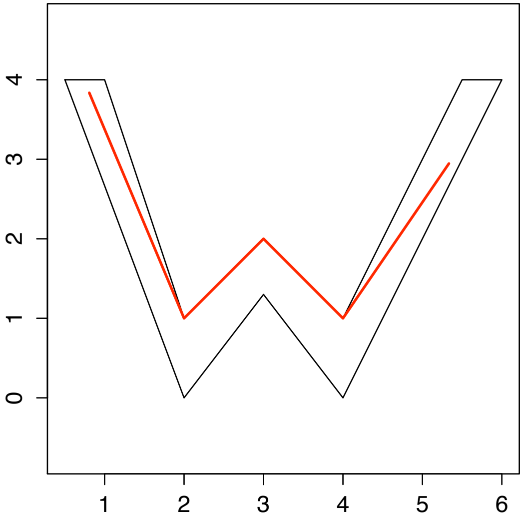
# A “new” algorithm for finding within-area distances

**So, how do we find the distances to put in D?**

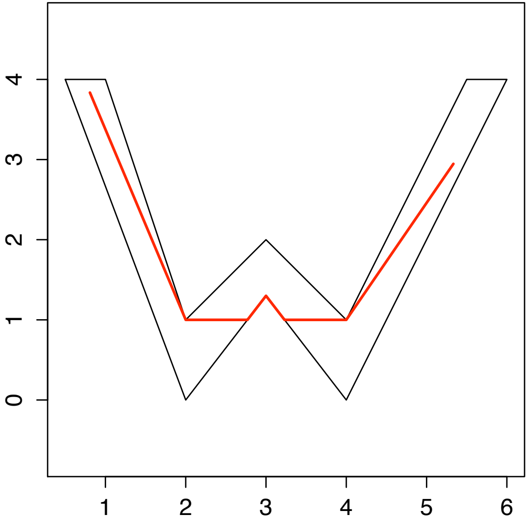
# Finding the path - 1



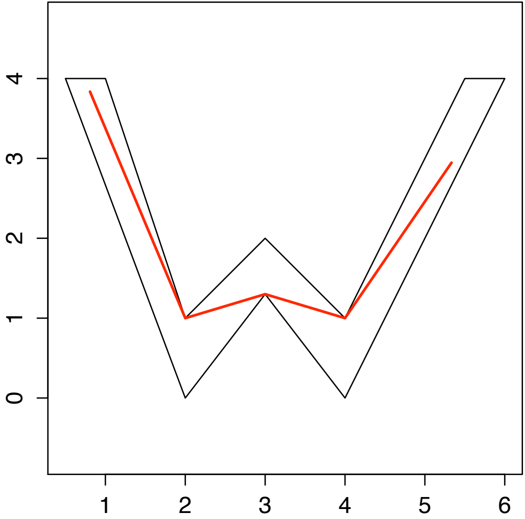
# Finding the path - 2



# Finding the path - 3



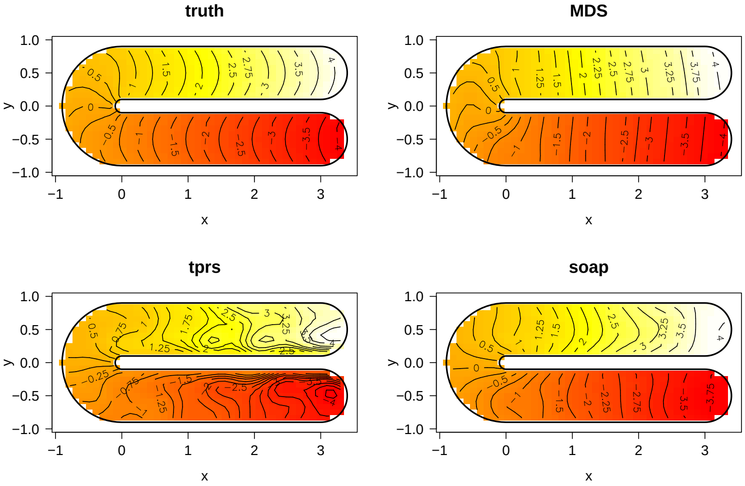
# Finding the path - 4



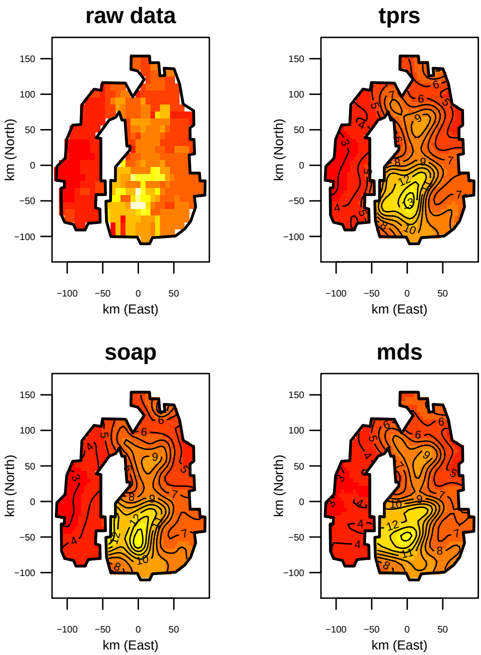
# Simulations

**But how does it perform?**

# Ramsay simulations



# The Aral sea

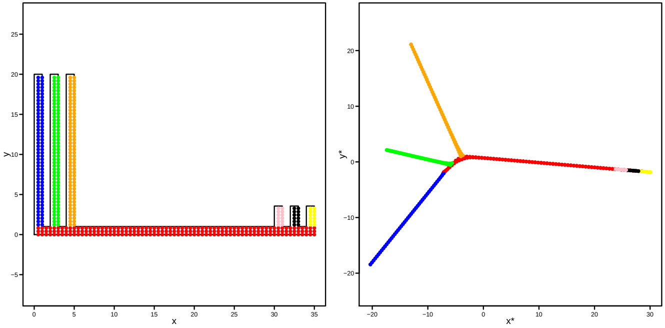


raw data
tprs
soap
mds
km (North)
km (East)
150
100
50
0
−50
−100
−100
−50
0
50

# But...

- Computationally costly to find the within-area distances. (Esp. prediction.)
- Comparative in MSE terms to the soap film smoother.
- Artefacts – ordering?
- Taking a $m$-D projection of $n$-D space – higher dimensions $==$ better?

**[[3D PLOT]]**



# Outline

# High dimensional smoothing with Duchon splines

**How can we fix this?**

# Nullspaces

- TPRS functions of the form:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \delta_i \eta_{md}(\mathbf{x} - \mathbf{x_i}) + \sum_{j=1}^{M} \alpha_j \phi_j(\mathbf{x})$$
$$= \text{radial basis functions} + \text{"planes"}$$

  Where

$$\begin{aligned} m &= \text{the derivative order in the penalty} \\ d &= \text{smoothing dimension} \\ n &= \text{\# data} \\ M &= \text{nullspace dimension} \end{aligned}$$

- $\phi_j(\mathbf{x})$ are linearly independent polynomials of degree $< m$.

# When nullspaces get big...(I)

- For TPRS, nullspace size is

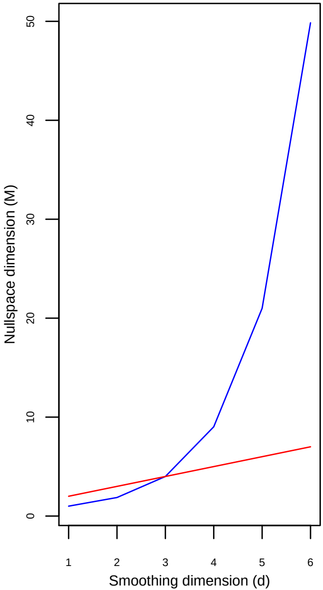
$$M = \binom{m+d-1}{d}$$

  Where

$$m = \text{the derivative order in the penalty}$$
$$d = \text{smoothing dimension}$$

- **AND** must obey $2m > d$
  smoothing dimension++ $\Rightarrow$ derivative order++++.
- $M$ gets very big, very fast.

# When nullspaces get big... (II)



# Enter Jean Duchon...

## SPLINES MINIMIZING ROTATION-INVARIANT SEMI-NORMS IN SOBOLEV SPACES


Jean DUCHON

Université Scientifique et Médicale

Laboratoire de Mathématiques Appliquées

B.P. 53 - 38041 Grenoble - France



ABSTRACT

We define a family of semi-norms $\|u\|_{m,s} = \left( \int_{\mathbb{R}^n} |\tau|^{2s} |\mathcal{F} D^m u(\tau)|^2 \, d\tau \right)^{1/2}$. Minimizing such semi-norms, subject to some interpolating conditions, leads to functions of very simple forms, providing interpolation methods that : 1°) preserve polynomials of degree $\leq m-1$ ; 2°) commute with similarities as well as translations and rotations of $\mathbb{R}^n$ ; and 3°) converge in Sobolev spaces $H^{m+s}(\Omega)$.

# High dimensional smoothing with Duchon splines

**TPRS to Duchon splines.**

# High dimensional smoothing with Duchon splines

Wood (2003) type penalty in 2D, order 2 derivatives:

$$\int\int_{\mathbf{R}^2} \left(\left(\frac{\partial^2 f(x_1,x_2)}{\partial x_1^2}\right)^2 + \left(\frac{\partial^2 f(x_1,x_2)}{\partial x_1 \partial x_2}\right)^2 + \left(\frac{\partial^2 f(x_1,x_2)}{\partial x_2^2}\right)^2\right) \mathsf{d}x_1 \mathsf{d}x_2.$$

More generally:

$$J_{m,d} = \int \dots \int_{\mathbb{R}^d} \sum_{\nu_1+\dots+\nu_d=m} \frac{m!}{\nu_1! \dots \nu_d!} \left(\frac{\partial^m f(x_1,\dots,x_d)}{\partial x_1^{\nu_1} \dots \partial x_d^{\nu_d}}\right)^2 \mathsf{d}x_1 \dots \mathsf{d}x_d$$

# High dimensional smoothing with Duchon splines

$$J_{m,d} = \int \dots \int_{\mathbb{R}^d} \sum_{\nu_1+\dots+\nu_d=m} \frac{m!}{\nu_1! \dots \nu_d!} \left( \frac{\partial^m f(x_1, \dots, x_d)}{\partial x_1^{\nu_1} \dots \partial x_d^{\nu_d}} \right)^2 \mathsf{d}x_1 \dots \mathsf{d}x_d$$

Why not take the Fourier transform?

$$J_{m,d} = \int \dots \int_{\mathbb{R}^d} \sum_{\nu_1+\dots+\nu_d=m} \frac{m!}{\nu_1! \dots \nu_d!} \left( \mathfrak{F} \frac{\partial^m f}{\partial x_1^{\nu_1} \dots \partial x_d^{\nu_d}} (\boldsymbol{\tau}) \right)^2 \mathsf{d}\boldsymbol{\tau}$$

(Equivalent by Plancherel theorem)

# Duchon splines penalty

Putting in some arbitrary weighting, $w(\boldsymbol{\tau})$:

$$\breve{J}_{m,d} = \int \dots \int_{\mathbb{R}^d} w(\boldsymbol{\tau}) \sum_{\nu_1+\dots+\nu_d=m} \frac{m!}{\nu_1! \dots \nu_d!} \left( \mathfrak{F} \frac{\partial^m f}{\partial x_1^{\nu_1} \dots \partial x_d^{\nu_d}} (\boldsymbol{\tau}) \right)^2 \mathsf{d}\boldsymbol{\tau}$$

Let $w(\boldsymbol{\tau}) = |\boldsymbol{\tau}|^{2s}$ (Duchon 1977)

$$\breve{J}_{m,d} = \int \dots \int_{\mathbb{R}^n} |\boldsymbol{\tau}|^{2s} \sum_{\nu_1+\dots+\nu_d=m} \frac{m!}{\nu_1! \dots \nu_d!} \left( \mathfrak{F} \frac{\partial^m f}{\partial x_1^{\nu_1} \dots \partial x_d^{\nu_d}} (\boldsymbol{\tau}) \right)^2 \mathsf{d}\boldsymbol{\tau}$$

# Duchon splines penalty

$$\breve{J}_{m,d} = \int \dots \int_{\mathbb{R}^n} |\boldsymbol{\tau}|^{2s} \sum_{\nu_1+\dots+\nu_d=m} \frac{m!}{\nu_1!\dots\nu_d!} \left( \mathfrak{F} \frac{\partial^m f}{\partial x_1^{\nu_1} \dots \partial x_d^{\nu_d}} (\boldsymbol{\tau}) \right)^2 \mathsf{d}\boldsymbol{\tau}$$

- This penalises high frequency components of the derivatives.
- For continuity now just need:

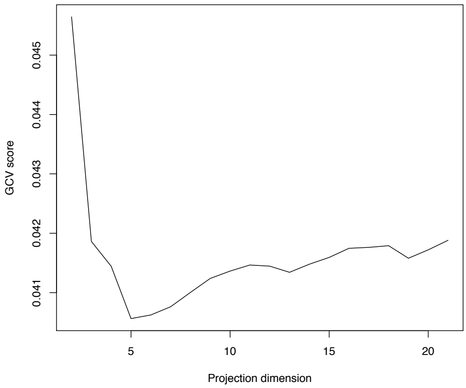
$$d/2 - m \leq s$$

- Can now always use $m = 2$ (some pseudo-theoretical reasons for that) and we have a set $d$, determine $s$ by

$$s = d/2 - 2$$

# Using ML/GCV score to select projection dimension

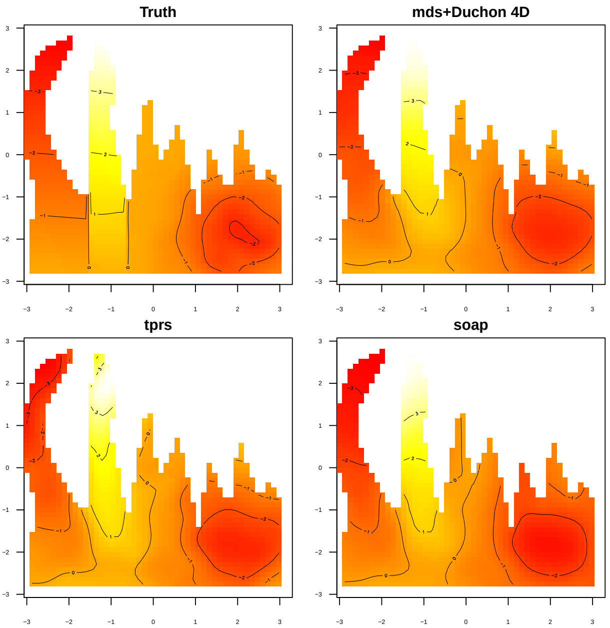
- So, we can now smooth in higher dimensions, **but how many?**
- Select MDS dimension using GCV or ML score.
- Fit model in a range of dimensions, pick best score.
- No guarantee of having a “nice” function to optimise.



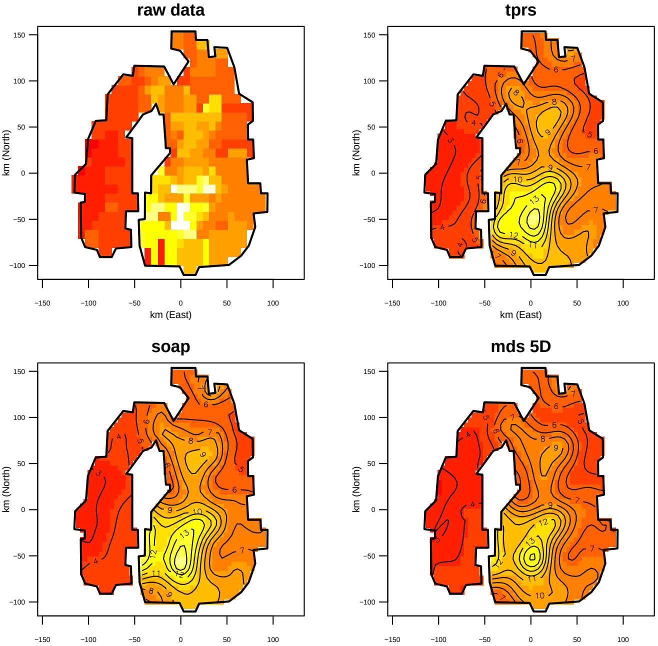
# More results

**How well do Duchon splines do?**

# Fake coastline simulation



# Aral sea


raw data
tprs
soap
mds 5D
km (North)
km (East)
150
100
50
0
−50
−100
−150
−100
−50
0
50
100

# Outline

# General distance smoothing

- Given a general distance matrix, $D$.
- Project the points in $D$ using MDS.
- Smooth over those points – full GAM framework available.

Examples:

- Other bio/ecol distances (?)
- MP voting data
- Microarrays
- Socio-economic data

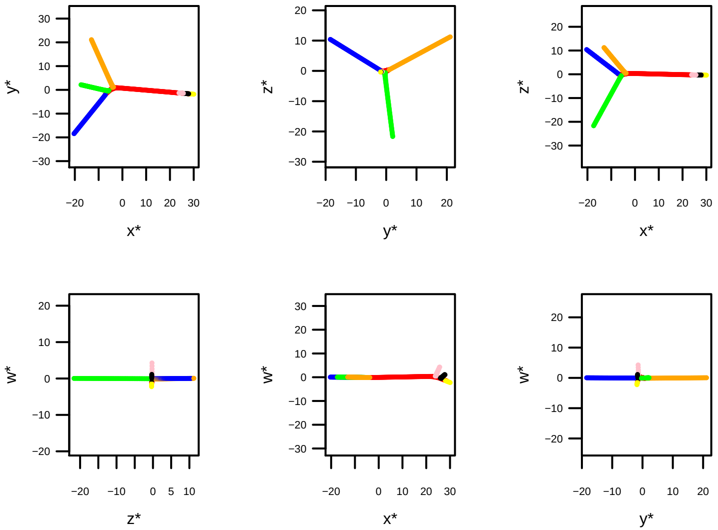
# Outline

# Conclusions

- Using Duchon splines gets around problems with high dimensional smoothing.
- MDS projections + Duchon splines + WAD useful for spatial smoothing.
- Perhaps applications in a broader set of problems involving distances.

Slides available at `http://people.bath.ac.uk/dlm27`

# Appendix - Ordering problems in 4-D



# Appendix - A "new" algorithm for finding within-area distances

- Algorithm "bounces" around inside the polygon.
- Initial path is just the path around the edge, then iterate over two steps: *delete* and *alter*.
- Delete (iterating over all nodes):
  - If we can shorten the path by simply deleting a node, do that.
- Alter (iterating over all nodes):
  - If we can find a shorter sub-path by bouncing off the other "side" of the polygon, do that.