080303a tpsSplin.mcd

ORIGIN := 0

Fred Bookstein's Thin-Plate Splines implemented by F. James Rohlf in his TPS series of programs

prepared by Wm Stein

The TPS series of programs for Windows may be found at:

http://life.bio.sunysb.edu/morph/

Look under "software" "thin-plate spline"

The TPS series supports several different activities including data collection (tps.Dig2) and subsequent analysis (tpsSplin, tpsRelw, tpsTree, tpsRegr, tpsPLS, tpsSmall). The object of the tpsSplin program is to implement Fred Bookstein's (1991 Cambridge Univ. Press) pairwise comparison of morphological shapes registered by shifts in position of landmark points (LM) in two dimensions as a mathematical deformation in the original sense of D'Arcy Thompson (1917 "On Growth and Form"). In Bookstein's approach, shape change (positional changes in landmark points) is decomposed into a "uniform" (or affine) component and a "non-uniform" component. The non-uniform "warp" part is modeled using mathematics derived from warpage of thin metal plates analyzed in terms of "bending energy" (resistance to bending) - an approach called "thin-plate splines". The thin-plate spline warp is further decomposed into "partial warps" using a more-or-less standard eigenvector/eigenvalue approach. Partial warps are therefore, in a sense, a spectral decomposition of the total "warp" (non-affine component of shape change), and partial warps sum together to recreate the total "warp". The value of this style of shape decomposition is perhaps debatable. Similar to Principle Components Analysis (PCA) - also a form of spectral decomposition - it remains for the biologist to read specific biological interpretation(s) into the results. However, the approach is statistically "guilt free", and exploratory, in the sense that no specific model of shape change is assumed, and no statistical hypothesis is formed or tested.

Reading the Data:

Br := READPRN("c:/2008Morphometrics/Booksteinr.dta") **< landmarks coordinates on first (reference) form**

Bd := READPRN("c:/2008Morphometrics/Booksteind.dta") **< landmark coordinates on second (data) form**

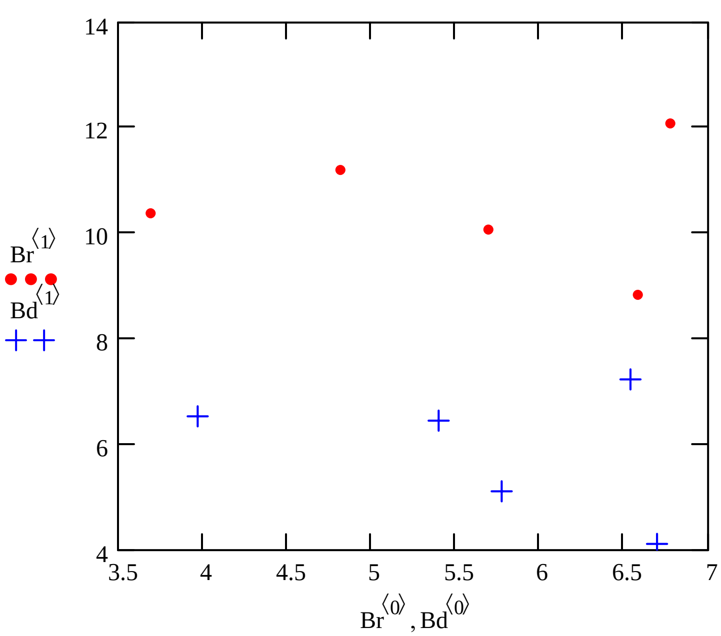
LMnum := rows(Br)   LMnum = 5   **< number of landmark points**

Landmark coordinate points

$$Br = \begin{pmatrix} 3.6929 & 10.3819 \\ 6.5827 & 8.8386 \\ 6.7756 & 12.0866 \\ 4.8189 & 11.2047 \\ 5.6969 & 10.0748 \end{pmatrix}$$

$$Bd = \begin{pmatrix} 3.972 & 6.535 \\ 6.697 & 4.118 \\ 6.539 & 7.236 \\ 5.402 & 6.453 \\ 5.776 & 5.114 \end{pmatrix}$$

Plotting Landmarks:



< Note that Br and Bd contain differences in LM locations that represent translation and rotation. These differences are probably related only to how the data were originally collected...

Calculating Matrix P_K (Bookstein 1991, p. 27, 32 & 294):

$i := 0..rows(Br) - 1 \quad j := 0..rows(Br) - 1$ **< index variables**

$x_i := \left(Br^{\langle 0\rangle}\right)_i \quad y_i := \left(Br^{\langle 1\rangle}\right)_i$ **< obtaining x & y coordinates from the reference form Br**

$r_{i,j} := \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}$ **< $r_{i,j}$ are the Euclidean distances between LM ponts i & j in reference form Br**

$$r = \begin{pmatrix} 0 & 3.276 & 3.523 & 1.395 & 2.027 \\ 3.276 & 0 & 3.254 & 2.951 & 1.521 \\ 3.523 & 3.254 & 0 & 2.146 & 2.283 \\ 1.395 & 2.951 & 2.146 & 0 & 1.431 \\ 2.027 & 1.521 & 2.283 & 1.431 & 0 \end{pmatrix}$$

$$P_{K_{i,j}} := \begin{cases} \left(r_{i,j}\right)^2 \cdot \ln\left[\left(r_{i,j}\right)^2\right] & \text{if } i \ne j \\ 0 & \text{otherwise} \end{cases}$$

< spline matrix P_K. Note that since ln(0) is undefined as a function, P_K has to be explicitly programmed here using a conditional ...

$$P_K = \begin{pmatrix} 0 & 25.472 & 31.251 & 1.294 & 5.81 \\ 25.472 & 0 & 24.98 & 18.851 & 1.939 \\ 31.251 & 24.98 & 0 & 7.036 & 8.602 \\ 1.294 & 18.851 & 7.036 & 0 & 1.467 \\ 5.81 & 1.939 & 8.602 & 1.467 & 0 \end{pmatrix}$$

^ Matrix P_K is an explicit calculation of "bending energy" between LM points in the reference form Br.

Matrix Q of 1's and x,y Coordinates of Landmark Points (Bookstein 1991, p. 32 & 320):

$ONE_i := 1 \quad k := 2 \quad ZERO_{k,2} := 0$

$$ONE = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad ZERO = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$Q := augment(ONE, Br)$

$$Q = \begin{pmatrix} 1 & 3.6929 & 10.3819 \\ 1 & 6.5827 & 8.8386 \\ 1 & 6.7756 & 12.0866 \\ 1 & 4.8189 & 11.2047 \\ 1 & 5.6969 & 10.0748 \end{pmatrix}$$

< In addition to "bending energy" P_K, a thin plate spline requires information about the (x,y) location of the "armature" points. This information is supplied by the LM points in the reference form Br.

Partioned Matrix L containing both affine and non-affine elements (Bookstein 1991, p. 32 & 320):

$L_{TOP} := augment\left(P_K, Q\right)$

$L_{BOT} := augment\left(Q^T, ZERO\right)$

$L := stack\left(L_{TOP}, L_{BOT}\right)$

$$L_{TOP} = \begin{pmatrix} 0 & 25.472 & 31.251 & 1.294 & 5.81 & 1 & 3.693 & 10.382 \\ 25.472 & 0 & 24.98 & 18.851 & 1.939 & 1 & 6.583 & 8.839 \\ 31.251 & 24.98 & 0 & 7.036 & 8.602 & 1 & 6.776 & 12.087 \\ 1.294 & 18.851 & 7.036 & 0 & 1.467 & 1 & 4.819 & 11.205 \\ 5.81 & 1.939 & 8.602 & 1.467 & 0 & 1 & 5.697 & 10.075 \end{pmatrix}$$

$$L_{BOT} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 3.693 & 6.583 & 6.776 & 4.819 & 5.697 & 0 & 0 & 0 \\ 10.382 & 8.839 & 12.087 & 11.205 & 10.075 & 0 & 0 & 0 \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 25.472 & 31.251 & 1.294 & 5.81 & 1 & 3.693 & 10.382 \\ 25.472 & 0 & 24.98 & 18.851 & 1.939 & 1 & 6.583 & 8.839 \\ 31.251 & 24.98 & 0 & 7.036 & 8.602 & 1 & 6.776 & 12.087 \\ 1.294 & 18.851 & 7.036 & 0 & 1.467 & 1 & 4.819 & 11.205 \\ 5.81 & 1.939 & 8.602 & 1.467 & 0 & 1 & 5.697 & 10.075 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 3.693 & 6.583 & 6.776 & 4.819 & 5.697 & 0 & 0 & 0 \\ 10.382 & 8.839 & 12.087 & 11.205 & 10.075 & 0 & 0 & 0 \end{pmatrix}$$

< Matrix L provides information about *potential* (i.e., before knowledge about any specific deformation) "bending energy" and LM locations in the reference form Br.

Vectors V & Y containing LM locations in the second (data) form (Bookstein 1991, p. 33 & 320):

$V := Bd$

$$V = \begin{pmatrix} 3.972 & 6.535 \\ 6.697 & 4.118 \\ 6.539 & 7.236 \\ 5.402 & 6.453 \\ 5.776 & 5.114 \end{pmatrix}$$

< Since the thin-plate spline models warping of a thin metal plate in the z direction given (x,y) information about position on the plate, Bookstein's procedure considers each column of V (V_x,V_y) separately as vectors V_x & V_y. Of course, matrix algebra doesn't care about this distinction...

$$Z := \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$Y := \text{stack}(V, Z)$

$$Y = \begin{pmatrix} 3.972 & 6.535 \\ 6.697 & 4.118 \\ 6.539 & 7.236 \\ 5.402 & 6.453 \\ 5.776 & 5.114 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

< Vectors Y_x & Y_y - columns of Y - provides an augmented description of vectors V_x & V_y to allow description of the affine component in matrix L.

For lack of better terminology provided by Bookstein or Rohlf, we can say here that vectors Y may be thought to "charge the spline" with information about the specific deformation involving comparison of Br with Bd.

Calculating the thin-plate spline (Bookstein 1991, p. 33 & 321):

$W := L^{-1} \cdot Y$

$$L^{-1} = \begin{pmatrix} 0.04928 & -0.00228 & 0.03286 & -0.0744 & -0.00546 & 1.92725 & -0.31551 & -0.05809 \\ -0.00228 & 0.03889 & -0.00042 & 0.04391 & -0.0801 & 1.81355 & 0.1776 & -0.29264 \\ 0.03286 & -0.00042 & 0.02195 & -0.04847 & -0.00592 & -3.78036 & 0.17271 & 0.24702 \\ -0.0744 & 0.04391 & -0.04847 & 0.15456 & -0.07561 & -0.72522 & -0.02746 & 0.14083 \\ -0.00546 & -0.0801 & -0.00592 & -0.07561 & 0.16709 & 1.76478 & -0.00735 & -0.03712 \\ 1.92725 & 1.81355 & -3.78036 & -0.72522 & 1.76478 & 701.06169 & -29.64533 & -50.94956 \\ -0.31551 & 0.1776 & 0.17271 & -0.02746 & -0.00735 & -29.64533 & 4.95546 & 0.14471 \\ -0.05809 & -0.29264 & 0.24702 & 0.14083 & -0.03712 & -50.94956 & 0.14471 & 4.83371 \end{pmatrix}$$

^ Inverse matrix of L

$$W = \begin{pmatrix} -0.038 & 0.0424 \\ 0.0232 & 0.0159 \\ -0.0248 & 0.0288 \\ 0.0798 & -0.0454 \\ -0.0402 & -0.0418 \\ 1.355 & -2.946 \\ 0.8747 & -0.2956 \\ -0.0289 & 0.9216 \end{pmatrix}$$

w's
^ non-affine coefficients

a_1
a_x
a_y
^ affine coefficients a_1, a_x, a_y

< coefficients of the thin-plate spline including the affine components. This information is used to compute the spline function f(x,y) for any arbitrary point (x,y) not necessarily LM points. See formula for f(x,y) in Bookstein p. 33.

Setting up a grid of points to display the spline:

$x_{min} := \min\left(Br^{\langle 0\rangle}\right)$ $y_{min} := \min\left(Br^{\langle 1\rangle}\right)$ **< calculating max and min coordinates of the LM points in Br.**

$x_{max} := \max\left(Br^{\langle 0\rangle}\right)$ $y_{max} := \max\left(Br^{\langle 1\rangle}\right)$

$n := 30$ **< specifying grid size**

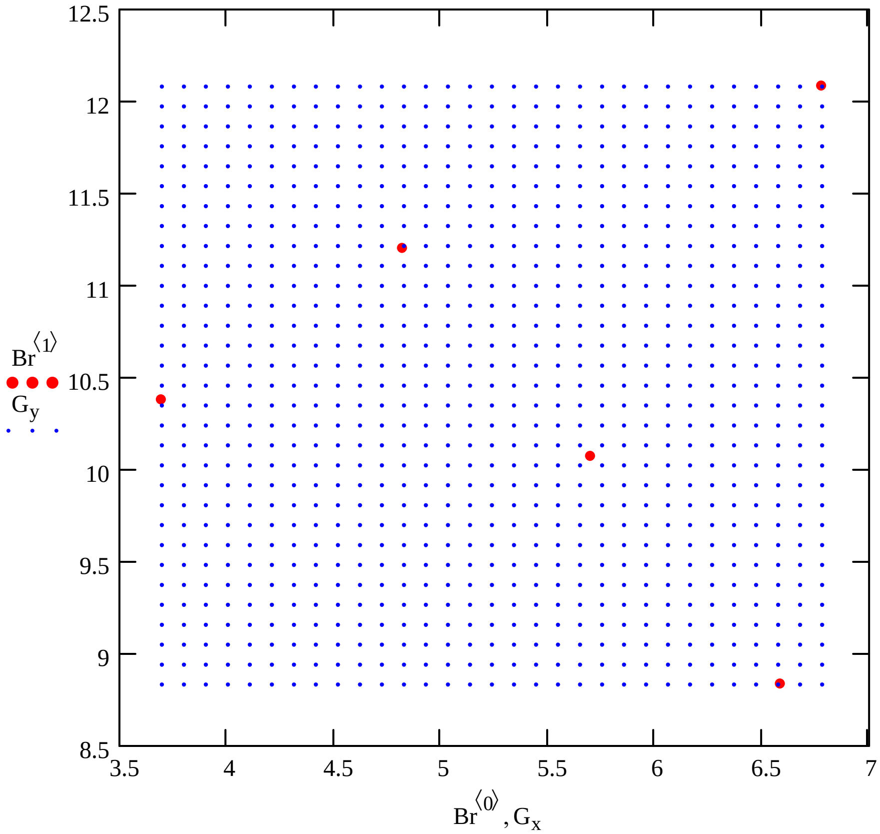
$$Br = \begin{pmatrix} 3.693 & 10.382 \\ 6.583 & 8.839 \\ 6.776 & 12.087 \\ 4.819 & 11.205 \\ 5.697 & 10.075 \end{pmatrix}$$

$s_x := \frac{(x_{max} - x_{min})}{n}$ $s_y := \frac{(y_{max} - y_{min})}{n}$ **< step size for point in the grid.**

$j := 0 .. n$ $k := 0 .. n$ **< index variables for grid.**

$Gx_{j,k} := x_{min} + j \cdot s_x$ $Gy_{j,k} := y_{min} + k \cdot s_y$ **< constructing the grid as separate matrices G_x and G_y.**

Plotting the initial LM points and grid:



matrix W as defined above:

$$W = \begin{pmatrix} -0.038 & 0.042 \\ 0.023 & 0.016 \\ -0.025 & 0.029 \\ 0.08 & -0.045 \\ -0.04 & -0.042 \\ 1.355 & -2.946 \\ 0.875 & -0.296 \\ -0.029 & 0.922 \end{pmatrix}$$

TPS Calculation for a single point on the grid:

$x := G_{x_{0,0}}$ $x = 3.693$ $y := G_{y_{0,0}}$ $y = 8.839$ **< choosing the point with indices 0,0 on the grid from among the points in Gx/Gy.**

$a1_x := \left(W^{\langle 0 \rangle}\right)_{LMnum}$ $a1_x = 1.355$ $a1_y := \left(W^{\langle 1 \rangle}\right)_{LMnum}$ $a1_y = -2.946$

$ax_x := \left(W^{\langle 0 \rangle}\right)_{LMnum+1}$ $ax_x = 0.875$ $ax_y := \left(W^{\langle 1 \rangle}\right)_{LMnum+1}$ $ax_y = -0.296$

$ay_x := \left(W^{\langle 0 \rangle}\right)_{LMnum+2}$ $ax_x = 0.875$ $ay_y := \left(W^{\langle 1 \rangle}\right)_{LMnum+2}$ $ay_y = 0.922$

^ defining affine coefficients a_1,a_x,a_y for each component x & y

Distance and Spline Energy:

$distLM_0 := \sqrt{\left[x - \left(Br^{\langle 0 \rangle}\right)_0\right]^2 + \left[y - \left(Br^{\langle 1 \rangle}\right)_0\right]^2}$ $distLM_0 = 1.543$ $ULM_0 := \left(distLM_0\right)^2 \cdot \ln\left[\left(distLM_0\right)^2\right]$ $ULM_0 = 2.067$

$distLM_1 := \sqrt{\left[x - \left(Br^{\langle 0 \rangle}\right)_1\right]^2 + \left[y - \left(Br^{\langle 1 \rangle}\right)_1\right]^2}$ $distLM_1 = 2.89$ $ULM_1 := \left(distLM_1\right)^2 \cdot \ln\left[\left(distLM_1\right)^2\right]$ $ULM_1 = 17.724$

$distLM_2 := \sqrt{\left[x - \left(Br^{\langle 0 \rangle}\right)_2\right]^2 + \left[y - \left(Br^{\langle 1 \rangle}\right)_2\right]^2}$ $distLM_2 = 4.478$ $ULM_2 := \left(distLM_2\right)^2 \cdot \ln\left[\left(distLM_2\right)^2\right]$ $ULM_2 = 60.125$

$distLM_3 := \sqrt{\left[x - \left(Br^{\langle 0 \rangle}\right)_3\right]^2 + \left[y - \left(Br^{\langle 1 \rangle}\right)_3\right]^2}$ $distLM_3 = 2.62$ $ULM_3 := \left(distLM_3\right)^2 \cdot \ln\left[\left(distLM_3\right)^2\right]$ $ULM_3 = 13.229$

$distLM_4 := \sqrt{\left[x - \left(Br^{\langle 0 \rangle}\right)_4\right]^2 + \left[y - \left(Br^{\langle 1 \rangle}\right)_4\right]^2}$ $distLM_4 = 2.355$ $ULM_4 := \left(distLM_4\right)^2 \cdot \ln\left[\left(distLM_4\right)^2\right]$ $ULM_4 = 9.496$

^ Euclidean distance (r) between point (x,y) and each LM **^ calculating TPS $U(r)=r^2\ln(r^2)$ function**

Weighted U functions : **U functions in x & y are weighted by the coefficients in W:**

$WUx_0 := \left(W^{\langle 0\rangle}\right)_0 \cdot ULM_0$ $WUx_0 = -0.079$ $WUy_0 := \left(W^{\langle 1\rangle}\right)_0 \cdot ULM_0$ $WUy_0 = 0.088$

$WUx_1 := \left(W^{\langle 0\rangle}\right)_1 \cdot ULM_1$ $WUx_1 = 0.411$ $WUy_1 := \left(W^{\langle 1\rangle}\right)_1 \cdot ULM_1$ $WUy_1 = 0.282$

$WUx_2 := \left(W^{\langle 0\rangle}\right)_2 \cdot ULM_2$ $WUx_2 = -1.488$ $WUy_2 := \left(W^{\langle 1\rangle}\right)_2 \cdot ULM_2$ $WUy_2 = 1.732$

$WUx_3 := \left(W^{\langle 0\rangle}\right)_3 \cdot ULM_3$ $WUx_3 = 1.055$ $WUy_3 := \left(W^{\langle 1\rangle}\right)_3 \cdot ULM_3$ $WUy_3 = -0.601$

$WUx_4 := \left(W^{\langle 0\rangle}\right)_4 \cdot ULM_4$ $WUx_4 = -0.382$ $WUy_4 := \left(W^{\langle 1\rangle}\right)_4 \cdot ULM_4$ $WUy_4 = -0.396$

^ Weighted function in x: **^ Weighted function in y:**

$$WUx = \begin{pmatrix} -0.079 \\ 0.411 \\ -1.488 \\ 1.055 \\ -0.382 \end{pmatrix} \qquad WUy = \begin{pmatrix} 0.088 \\ 0.282 \\ 1.732 \\ -0.601 \\ -0.396 \end{pmatrix}$$

$\sum WUx = -0.482$ $\sum WUy = 1.105$

< sum of the weighted functions Compare with na_x & na_y below.

The spline function:

$$f_x := a1_x + ax_x \cdot x + ay_x \cdot y + \left(W^{\langle 0\rangle}\right)_0 \cdot ULM_0 + \left(W^{\langle 0\rangle}\right)_1 \cdot ULM_1 + \left(W^{\langle 0\rangle}\right)_2 \cdot ULM_2 + \left(W^{\langle 0\rangle}\right)_3 \cdot ULM_3 + \left(W^{\langle 0\rangle}\right)_4 \cdot ULM_4$$

$$f_y := a1_y + ax_y \cdot x + ay_y \cdot y + \left(W^{\langle 1\rangle}\right)_0 \cdot ULM_0 + \left(W^{\langle 1\rangle}\right)_1 \cdot ULM_1 + \left(W^{\langle 1\rangle}\right)_2 \cdot ULM_2 + \left(W^{\langle 1\rangle}\right)_3 \cdot ULM_3 + \left(W^{\langle 1\rangle}\right)_4 \cdot ULM_4$$

$f_x = 3.848$

$f_y = 5.213$

< displacements calculated separately for x & y as if each is to be independently displaced by warping the spline in the z direction for a given a pont on the plane (x,y). Bookstein's innovative trick it to use these values in the original (x,y) plane by considering each as components ($x+f_x$, $y+f_y$) of displacement.

Decomposition of the spline:

$affine_x := a1_x + ax_x \cdot x + ay_x \cdot y$ $affine_x = 4.33$

$affine_y := a1_y + ax_y \cdot x + ay_y \cdot y$ $affine_y = 4.109$

< Bookstein's "uniform" (affine) displacement separately for x & y.

$$na_x := \left(W^{\langle 0\rangle}\right)_0 \cdot ULM_0 + \left(W^{\langle 0\rangle}\right)_1 \cdot ULM_1 + \left(W^{\langle 0\rangle}\right)_2 \cdot ULM_2 + \left(W^{\langle 0\rangle}\right)_3 \cdot ULM_3 + \left(W^{\langle 0\rangle}\right)_4 \cdot ULM_4 \qquad na_x = -0.482$$

$$na_y := \left(W^{\langle 1\rangle}\right)_0 \cdot ULM_0 + \left(W^{\langle 1\rangle}\right)_1 \cdot ULM_1 + \left(W^{\langle 1\rangle}\right)_2 \cdot ULM_2 + \left(W^{\langle 1\rangle}\right)_3 \cdot ULM_3 + \left(W^{\langle 1\rangle}\right)_4 \cdot ULM_4 \qquad na_y = 1.105$$

^ The "non-uniform" displacements separately for x & y.

The ulimate goal here is to provide pictures showing TOTAL displacement ($x+f_x$, $y+f_y$) as well as partial displacements ($x+affine_x$, $y+affine_y$) for the affine part, and ($x+na_x$, $y+na_y$) for the non-affine part.

Now doing TPS calculations for all points on the grid:

$x_{j,k} := Gx_{j,k}$ $y_{j,k} := Gy_{j,k}$ **< specifying each point in G_x/G_y in turn**

Distances between points (x,y) in G_x/G_y and each LM point in Br is now calculated and from this, the spline energy function U is determined.

$$rd0_{j,k} := \sqrt{\left[x_{j,k} - \left(Br^{\langle 0\rangle}\right)_0\right]^2 + \left[y_{j,k} - \left(Br^{\langle 1\rangle}\right)_0\right]^2} \qquad Ud0_{j,k} := \begin{cases} \left(rd0_{j,k}\right)^2 \cdot \ln\left[\left(rd0_{j,k}\right)^2\right] & \text{if } rd0_{j,k} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$rd1_{j,k} := \sqrt{\left[x_{j,k} - \left(Br^{\langle 0\rangle}\right)_1\right]^2 + \left[y_{j,k} - \left(Br^{\langle 1\rangle}\right)_1\right]^2} \qquad Ud1_{j,k} := \begin{cases} \left(rd1_{j,k}\right)^2 \cdot \ln\left[\left(rd1_{j,k}\right)^2\right] & \text{if } rd1_{j,k} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$rd2_{j,k} := \sqrt{\left[x_{j,k} - \left(Br^{\langle 0\rangle}\right)_2\right]^2 + \left[y_{j,k} - \left(Br^{\langle 1\rangle}\right)_2\right]^2} \qquad Ud2_{j,k} := \begin{cases} \left(rd2_{j,k}\right)^2 \cdot \ln\left[\left(rd2_{j,k}\right)^2\right] & \text{if } rd2_{j,k} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$rd3_{j,k} := \sqrt{\left[x_{j,k} - \left(Br^{\langle 0\rangle}\right)_3\right]^2 + \left[y_{j,k} - \left(Br^{\langle 1\rangle}\right)_3\right]^2} \qquad Ud3_{j,k} := \begin{cases} \left(rd3_{j,k}\right)^2 \cdot \ln\left[\left(rd3_{j,k}\right)^2\right] & \text{if } rd3_{j,k} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$rd4_{j,k} := \sqrt{\left[x_{j,k} - \left(Br^{\langle 0\rangle}\right)_4\right]^2 + \left[y_{j,k} - \left(Br^{\langle 1\rangle}\right)_4\right]^2} \qquad Ud4_{j,k} := \begin{cases} \left(rd4_{j,k}\right)^2 \cdot \ln\left[\left(rd4_{j,k}\right)^2\right] & \text{if } rd4_{j,k} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

^ Euclidean distances

^ Spline Energy function U
Note here that one must control for the pathological situation where ln(0) is undefined as done for P_K above.

Weighted U functions:

$$WUdx0_{j,k} := \left(W^{\langle 0\rangle}\right)_0 \cdot Ud0_{j,k} \qquad WUdy0_{j,k} := \left(W^{\langle 1\rangle}\right)_0 \cdot Ud0_{j,k}$$

$$WUdx1_{j,k} := \left(W^{\langle 0\rangle}\right)_1 \cdot Ud1_{j,k} \qquad WUdy1_{j,k} := \left(W^{\langle 1\rangle}\right)_1 \cdot Ud1_{j,k}$$

$$WUdx2_{j,k} := \left(W^{\langle 0\rangle}\right)_2 \cdot Ud2_{j,k} \qquad WUdy2_{j,k} := \left(W^{\langle 1\rangle}\right)_2 \cdot Ud2_{j,k}$$

$$WUdx3_{j,k} := \left(W^{\langle 0\rangle}\right)_3 \cdot Ud3_{j,k} \qquad WUdy3_{j,k} := \left(W^{\langle 1\rangle}\right)_3 \cdot Ud3_{j,k}$$

$$WUdx4_{j,k} := \left(W^{\langle 0\rangle}\right)_4 \cdot Ud4_{j,k} \qquad WUdy4_{j,k} := \left(W^{\langle 1\rangle}\right)_4 \cdot Ud4_{j,k}$$

$$W = \begin{pmatrix} -0.038 & 0.042 \\ 0.023 & 0.016 \\ -0.025 & 0.029 \\ 0.08 & -0.045 \\ -0.04 & -0.042 \\ 1.355 & -2.946 \\ 0.875 & -0.296 \\ -0.029 & 0.922 \end{pmatrix}$$

Sums:

$$WUdx_{sum_{j,k}} := WUdx0_{j,k} + WUdx1_{j,k} + WUdx2_{j,k} + WUdx3_{j,k} + WUdx4_{j,k}$$

$$WUdy_{sum_{j,k}} := WUdy0_{j,k} + WUdy1_{j,k} + WUdy2_{j,k} + WUdy3_{j,k} + WUdy4_{j,k}$$

Decomposition of the spline:

$WUdx_{sum} =$

	0	1	2	3	4
0	-0.482	-0.479	-0.475	-0.47	-0.464
1	-0.471	-0.468	-0.464	-0.459	-0.452
2	-0.46	-0.457	-0.452	-0.447	-0.44
3	-0.449	-0.445	-0.441	-0.435	-0.428
4	-0.437	-0.434	-0.429	-0.423	-0.416
5	-0.425	-0.422	-0.417	-0.411	-0.404
6	-0.413	-0.41	-0.405	-0.399	-0.392
7	-0.401	-0.398	-0.393	-0.388	-0.38
8	-0.389	-0.386	-0.382	-0.376	-0.369
9	-0.377	-0.374	-0.371	-0.365	-0.358
10	-0.365	-0.363	-0.36	-0.355	-0.348

Matrix of weighted sums for x and for y. Note that entries with indices (0,0) report the same value calculated for this point above.

$WUdy_{sum} =$

	0	1	2	3	4
0	1.105	1.096	1.087	1.079	1.072
1	1.08	1.07	1.06	1.051	1.042
2	1.054	1.043	1.032	1.022	1.012
3	1.028	1.016	1.004	0.992	0.981
4	1.003	0.989	0.976	0.963	0.95
5	0.977	0.962	0.948	0.933	0.919
6	0.952	0.936	0.92	0.904	0.888
7	0.928	0.91	0.893	0.875	0.858
8	0.905	0.886	0.866	0.847	0.829
9	0.882	0.862	0.841	0.821	0.8
10	0.862	0.84	0.818	0.796	0.774
11	0.842	0.819	0.796	0.772	0.749

partial tables of size (n X n)

$n = 30$ **< as specified above.**

^ This is the non-uniform part as recorded for each point x,y on the grid G_x/G_y.

$m := 0..2$

$adx_m := \left(W^{\langle 0\rangle}\right)_{LMnum+m}$ $\quad adx = \begin{pmatrix} 1.355 \\ 0.875 \\ -0.029 \end{pmatrix}$

$ady_m := \left(W^{\langle 1\rangle}\right)_{LMnum+m}$ $\quad ady = \begin{pmatrix} -2.946 \\ -0.296 \\ 0.922 \end{pmatrix}$

$Aff_{x_{j,k}} := adx_0 + x_{j,k} \cdot adx_1 + y_{j,k} \cdot adx_2$

^ Uniform (affine) part in the x's

$Aff_{y_{j,k}} := ady_0 + x_{j,k} \cdot ady_1 + y_{j,k} \cdot ady_2$

^ Uniform (affine) part in the y's

Total spline:

$TSx_{j,k} := Aff_{x_{j,k}} + WUdx_{sum_{j,k}}$

$TSy_{j,k} := Aff_{y_{j,k}} + WUdy_{sum_{j,k}}$

$TSx =$

	0	1	2	3	4	5
0	3.848	3.848	3.849	3.851	3.854	3.858
1	3.949	3.949	3.95	3.952	3.955	3.96
2	4.05	4.05	4.051	4.054	4.057	4.062
3	4.151	4.151	4.153	4.155	4.159	4.164
4	4.253	4.253	4.255	4.257	4.261	4.267
5	4.354	4.355	4.356	4.359	4.363	4.369
6	4.456	4.457	4.458	4.461	4.465	4.471
7	4.558	4.558	4.56	4.562	4.567	4.573
8	4.66	4.66	4.661	4.664	4.668	4.674
9	4.762	4.762	4.762	4.765	4.768	4.774

$TSy =$

	0	1	2	3	4	5
0	5.213	5.304	5.396	5.487	5.579	5.672
1	5.158	5.248	5.338	5.428	5.52	5.611
2	5.102	5.19	5.28	5.369	5.459	5.549
3	5.046	5.133	5.221	5.309	5.398	5.487
4	4.99	5.076	5.162	5.249	5.336	5.424
5	4.934	5.019	5.104	5.189	5.275	5.361
6	4.879	4.962	5.046	5.129	5.214	5.298
7	4.824	4.906	4.988	5.07	5.153	5.236
8	4.77	4.851	4.932	5.012	5.093	5.174
9	4.718	4.797	4.876	4.955	5.035	5.114

Now TPS calculation for LM points in the reference form Br:

$$Brx_i := \left(Br^{\langle 0\rangle}\right)_i \qquad Bry_i := \left(Br^{\langle 1\rangle}\right)_i$$

$$Br = \begin{pmatrix} 3.693 & 10.382 \\ 6.583 & 8.839 \\ 6.776 & 12.087 \\ 4.819 & 11.205 \\ 5.697 & 10.075 \end{pmatrix}$$

$$Brx = \begin{pmatrix} 3.693 \\ 6.583 \\ 6.776 \\ 4.819 \\ 5.697 \end{pmatrix} \qquad Bry = \begin{pmatrix} 10.382 \\ 8.839 \\ 12.087 \\ 11.205 \\ 10.075 \end{pmatrix}$$

< specifying (x,y) for each point in Br.

$$W = \begin{pmatrix} -0.038 & 0.042 \\ 0.023 & 0.016 \\ -0.025 & 0.029 \\ 0.08 & -0.045 \\ -0.04 & -0.042 \\ 1.355 & -2.946 \\ 0.875 & -0.296 \\ -0.029 & 0.922 \end{pmatrix}$$

$$ii := 0 \,..\, LMnum - 1$$

$$Brd_{i,ii} := \sqrt{\left[Brx_i - \left(Br^{\langle 0\rangle}\right)_{ii}\right]^2 + \left[Bry_i - \left(Br^{\langle 1\rangle}\right)_{ii}\right]^2}$$

< Calculating Euclidean distances

$$BrU_{i,ii} := \begin{cases} \left(Brd_{i,ii}\right)^2 \cdot \ln\left[\left(Brd_{i,ii}\right)^2\right] & \text{if } Brd_{i,ii} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

< Calculating Spline energy function U

$$Brd = \begin{pmatrix} 0 & 3.276 & 3.523 & 1.395 & 2.027 \\ 3.276 & 0 & 3.254 & 2.951 & 1.521 \\ 3.523 & 3.254 & 0 & 2.146 & 2.283 \\ 1.395 & 2.951 & 2.146 & 0 & 1.431 \\ 2.027 & 1.521 & 2.283 & 1.431 & 0 \end{pmatrix} \qquad BrU = \begin{pmatrix} 0 & 25.472 & 31.251 & 1.294 & 5.81 \\ 25.472 & 0 & 24.98 & 18.851 & 1.939 \\ 31.251 & 24.98 & 0 & 7.036 & 8.602 \\ 1.294 & 18.851 & 7.036 & 0 & 1.467 \\ 5.81 & 1.939 & 8.602 & 1.467 & 0 \end{pmatrix}$$

< Evaluations

$$BrWUx_{i,ii} := \left(W^{\langle 0\rangle}\right)_i \cdot BrU_{i,ii} \qquad BrWUy_{i,ii} := \left(W^{\langle 1\rangle}\right)_i \cdot BrU_{i,ii}$$

< Calculating weighted U functions

$$BrWUx = \begin{pmatrix} 0 & -0.969 & -1.188 & -0.049 & -0.221 \\ 0.591 & 0 & 0.579 & 0.437 & 0.045 \\ -0.774 & -0.618 & 0 & -0.174 & -0.213 \\ 0.103 & 1.504 & 0.561 & 0 & 0.117 \\ -0.233 & -0.078 & -0.346 & -0.059 & 0 \end{pmatrix} \qquad BrWUy = \begin{pmatrix} 0 & 1.081 & 1.327 & 0.055 & 0.247 \\ 0.405 & 0 & 0.398 & 0.3 & 0.031 \\ 0.9 & 0.72 & 0 & 0.203 & 0.248 \\ -0.059 & -0.856 & -0.32 & 0 & -0.067 \\ -0.243 & -0.081 & -0.359 & -0.061 & 0 \end{pmatrix}$$

Decomposition of the spline:

Non-uniform part:

$$BrWUsum_{x_i} := \sum BrWUx^{\langle i\rangle} \qquad BrWUsum_{y_i} := \sum BrWUy^{\langle i\rangle}$$

< Calculating non-uniform part

$$BrWUsum_x = \begin{pmatrix} -0.313 \\ -0.161 \\ -0.394 \\ 0.155 \\ -0.272 \end{pmatrix} \qquad BrWUsum_y = \begin{pmatrix} 1.005 \\ 0.864 \\ 1.045 \\ 0.496 \\ 0.459 \end{pmatrix}$$

< Evaluation for each LM point in Br

$$adx = \begin{pmatrix} 1.355 \\ 0.875 \\ -0.029 \end{pmatrix} \qquad ady = \begin{pmatrix} -2.946 \\ -0.296 \\ 0.922 \end{pmatrix}$$

Uniform Part:

$$BrAff_{x_i} := adx_0 + Brx_i \cdot adx_1 + Bry_i \cdot adx_2$$

$$BrAff_{y_i} := ady_0 + Brx_i \cdot ady_1 + Bry_i \cdot ady_2$$

^ Calculation of affine part

$$BrAff_x = \begin{pmatrix} 4.286 \\ 6.858 \\ 6.933 \\ 5.247 \\ 6.047 \end{pmatrix} \qquad BrAff_y = \begin{pmatrix} 5.531 \\ 3.254 \\ 6.191 \\ 5.956 \\ 4.656 \end{pmatrix}$$

< Evaluations

Total:

$$TSBr_{x_i} := BrAff_{x_i} + BrWUsum_{x_i} \qquad TSBr_{y_i} := BrAff_{y_i} + BrWUsum_{y_i}$$

Displacements visualized in x & y:

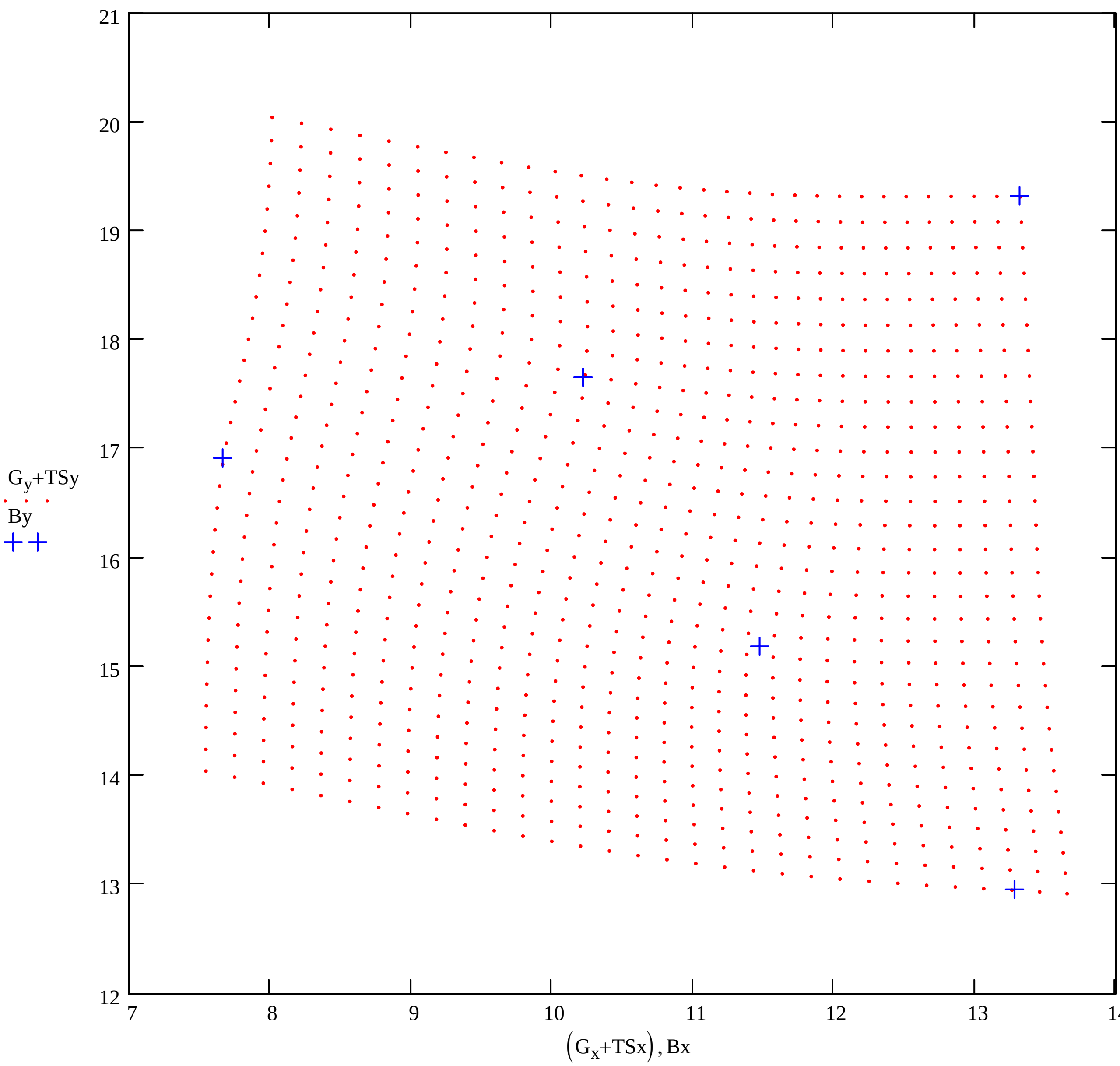
$$Bx := Br^{\langle 0 \rangle} + TSBr_x \qquad By := Br^{\langle 1 \rangle} + TSBr_y$$

$$Bx = \begin{pmatrix} 7.665 \\ 13.28 \\ 13.315 \\ 10.22 \\ 11.472 \end{pmatrix} \qquad By = \begin{pmatrix} 16.917 \\ 12.957 \\ 19.323 \\ 17.657 \\ 15.189 \end{pmatrix}$$

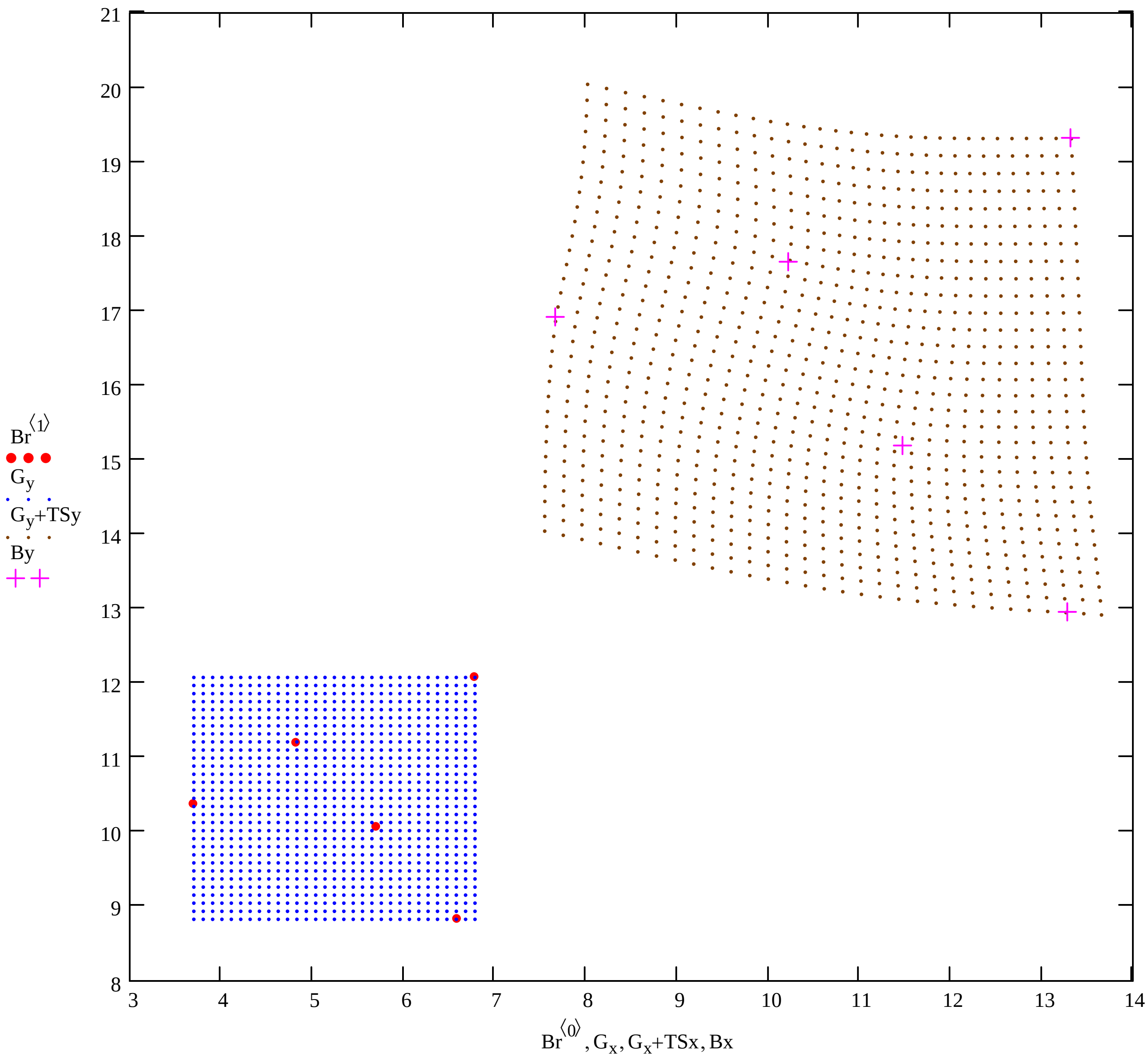
$$TSBr_x = \begin{pmatrix} 3.972 \\ 6.697 \\ 6.539 \\ 5.402 \\ 5.776 \end{pmatrix} \qquad TSBr_y = \begin{pmatrix} 6.535 \\ 4.118 \\ 7.236 \\ 6.453 \\ 5.114 \end{pmatrix}$$

^ Evaluations for each LM point in Br

Plot of "deformed" grid and LM Points in the second (data) configuration:



Combined Plots:



Note that the calculations above do not specify an "opitimal" overlap of the grids. This might be accomplished by centering each grid or form first before making the above calculations. In general, however, this doesn't matter.

Partial Plots:

It is now possible to decompose the thin plate spline calculated above into subcomponents that all sum to the total spline displayed in graphs above. The first step involves separating the "uniform" (affine) from the "non-uniform" parts as already calculated.

Affine Part:

Points on the grid (calculated above):

$Affx_{j,k} := adx_0 + x_{j,k} \cdot adx_1 + y_{j,k} \cdot adx_2$

^ Uniform (affine) part in the x's

$Affy_{j,k} := ady_0 + x_{j,k} \cdot ady_1 + y_{j,k} \cdot ady_2$

^ Uniform (affine) part in the y's

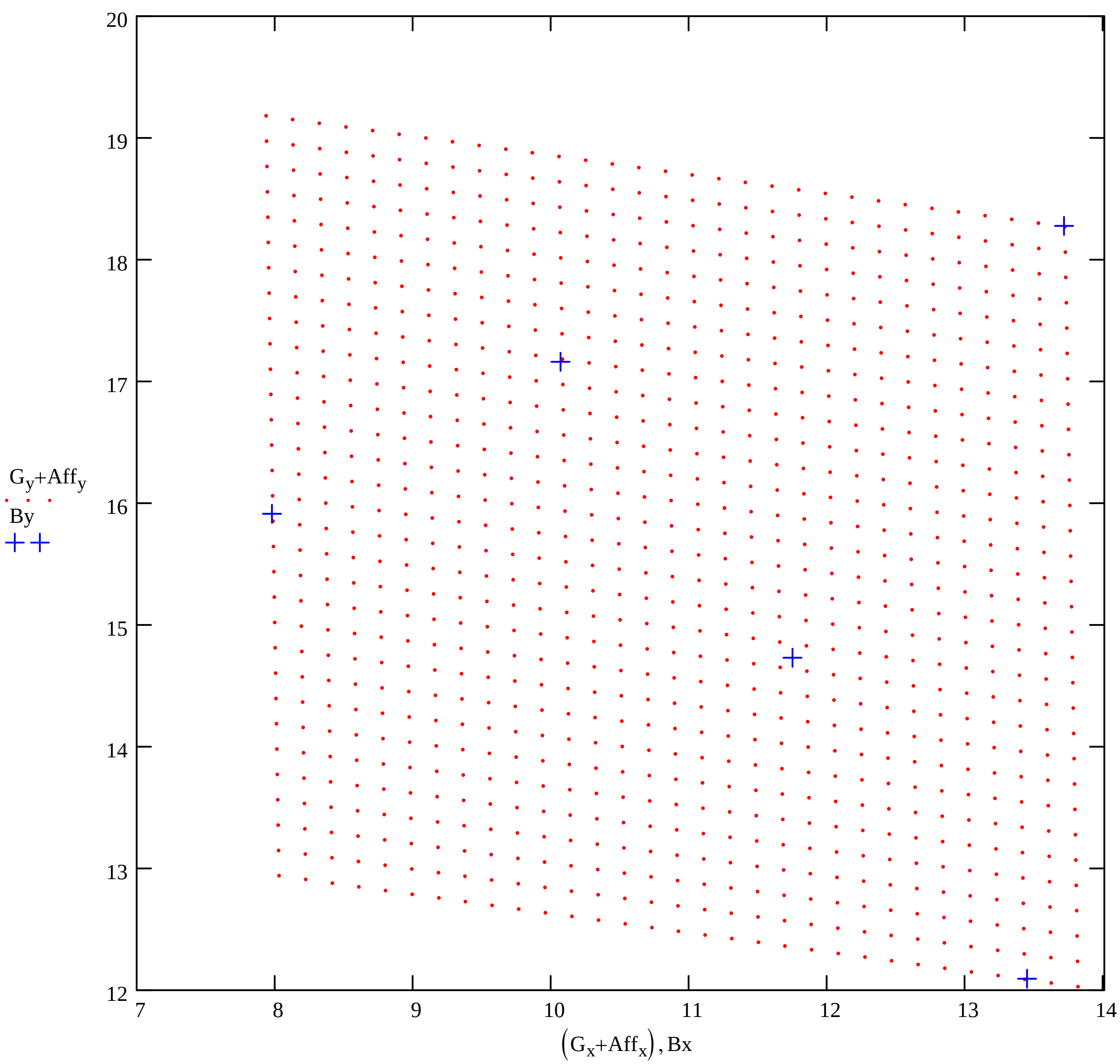
LM Points in Br (calculated above)

$BrAffx_i := adx_0 + Brx_i \cdot adx_1 + Bry_i \cdot adx_2$

$BrAffy_i := ady_0 + Brx_i \cdot ady_1 + Bry_i \cdot ady_2$

^ Calculation of affine part

$Bx := Br^{\langle 0 \rangle} + BrAff_x \qquad By := Br^{\langle 1 \rangle} + BrAff_y$



Non-affine Part:

Non uniform sums for points on the grid (calculated above):

$WUdx_{sum} =$

	0	1	2	3	4
0	-0.482	-0.479	-0.475	-0.47	-0.464
1	-0.471	-0.468	-0.464	-0.459	-0.452
2	-0.46	-0.457	-0.452	-0.447	-0.44
3	-0.449	-0.445	-0.441	-0.435	-0.428
4	-0.437	-0.434	-0.429	-0.423	-0.416
5	-0.425	-0.422	-0.417	-0.411	-0.404

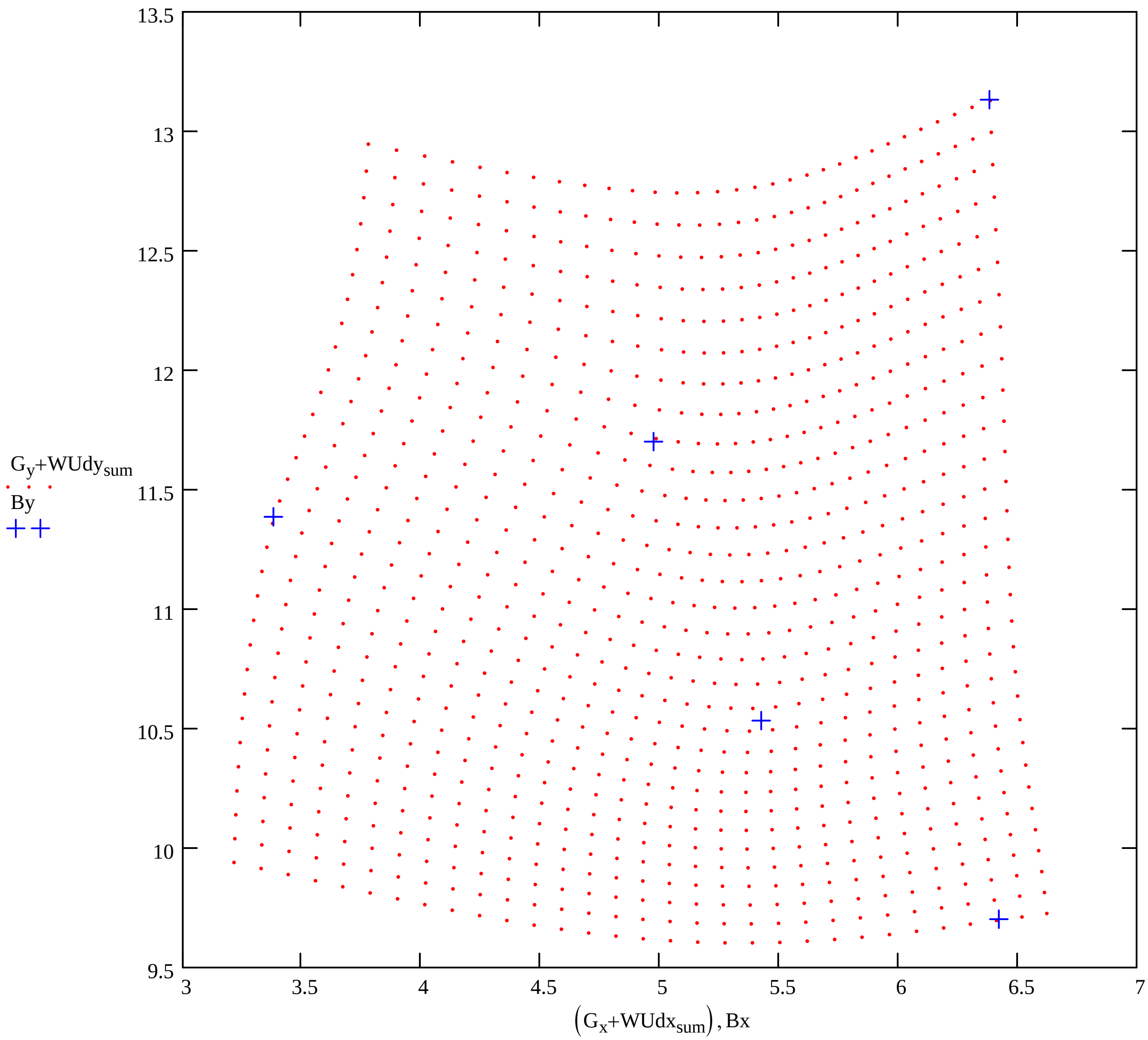
$WUdy_{sum} =$

	0	1	2	3	4
0	1.105	1.096	1.087	1.079	1.072
1	1.08	1.07	1.06	1.051	1.042
2	1.054	1.043	1.032	1.022	1.012
3	1.028	1.016	1.004	0.992	0.981
4	1.003	0.989	0.976	0.963	0.95
5	0.977	0.962	0.948	0.933	0.919

Non uniform sums for LM points in Br (calculated above):

$$BrWUsum_x = \begin{pmatrix} -0.313 \\ -0.161 \\ -0.394 \\ 0.155 \\ -0.272 \end{pmatrix} \qquad BrWUsum_y = \begin{pmatrix} 1.005 \\ 0.864 \\ 1.045 \\ 0.496 \\ 0.459 \end{pmatrix}$$

$$Bx := Br^{\langle 0\rangle} + BrWUsum_x \quad By := Br^{\langle 1\rangle} + BrWUsum_y$$



Principal Warps & Partial Warps:

Bookstein (1991) describes (and Rohlf impliments in TPS) an further spectral decomposition of the non-affine part of the thin-plate spline very much in the spirit of PCA. The mathematics, involving eigenvectors and eigenvalues, is nearly identical to PCA although based on the very different matrix L_K^{-1} described below.

From above, Bookstein set up the partitioned matrix L:

$$L = \begin{pmatrix} 0 & 25.472 & 31.251 & 1.294 & 5.81 & 1 & 3.693 & 10.382 \\ 25.472 & 0 & 24.98 & 18.851 & 1.939 & 1 & 6.583 & 8.839 \\ 31.251 & 24.98 & 0 & 7.036 & 8.602 & 1 & 6.776 & 12.087 \\ 1.294 & 18.851 & 7.036 & 0 & 1.467 & 1 & 4.819 & 11.205 \\ 5.81 & 1.939 & 8.602 & 1.467 & 0 & 1 & 5.697 & 10.075 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 3.693 & 6.583 & 6.776 & 4.819 & 5.697 & 0 & 0 & 0 \\ 10.382 & 8.839 & 12.087 & 11.205 & 10.075 & 0 & 0 & 0 \end{pmatrix}$$

comprised of elements:

$$P_K = \begin{pmatrix} 0 & 25.472 & 31.251 & 1.294 & 5.81 \\ 25.472 & 0 & 24.98 & 18.851 & 1.939 \\ 31.251 & 24.98 & 0 & 7.036 & 8.602 \\ 1.294 & 18.851 & 7.036 & 0 & 1.467 \\ 5.81 & 1.939 & 8.602 & 1.467 & 0 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1 & 3.693 & 10.382 \\ 1 & 6.583 & 8.839 \\ 1 & 6.776 & 12.087 \\ 1 & 4.819 & 11.205 \\ 1 & 5.697 & 10.075 \end{pmatrix} \qquad ZERO = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

< 1 & reference LM (x,y)

From this, he calculated the matrix inverse L^{-1}:

$$L^{-1} = \begin{pmatrix} 0.04928 & -0.00228 & 0.03286 & -0.0744 & -0.00546 & 1.92725 & -0.31551 & -0.05809 \\ -0.00228 & 0.03889 & -0.00042 & 0.04391 & -0.0801 & 1.81355 & 0.1776 & -0.29264 \\ 0.03286 & -0.00042 & 0.02195 & -0.04847 & -0.00592 & -3.78036 & 0.17271 & 0.24702 \\ -0.0744 & 0.04391 & -0.04847 & 0.15456 & -0.07561 & -0.72522 & -0.02746 & 0.14083 \\ -0.00546 & -0.0801 & -0.00592 & -0.07561 & 0.16709 & 1.76478 & -0.00735 & -0.03712 \\ 1.92725 & 1.81355 & -3.78036 & -0.72522 & 1.76478 & 701.06169 & -29.64533 & -50.94956 \\ -0.31551 & 0.1776 & 0.17271 & -0.02746 & -0.00735 & -29.64533 & 4.95546 & 0.14471 \\ -0.05809 & -0.29264 & 0.24702 & 0.14083 & -0.03712 & -50.94956 & 0.14471 & 4.83371 \end{pmatrix}$$

And he "charged the spline" using Y:

- to create the coefficients W:

data LM (x,y) with zeros >

$$Y = \begin{pmatrix} 3.972 & 6.535 \\ 6.697 & 4.118 \\ 6.539 & 7.236 \\ 5.402 & 6.453 \\ 5.776 & 5.114 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$L^{-1} \cdot Y = \begin{pmatrix} -0.038 & 0.042 \\ 0.023 & 0.016 \\ -0.025 & 0.029 \\ 0.08 & -0.045 \\ -0.04 & -0.042 \\ 1.355 & -2.946 \\ 0.875 & -0.296 \\ -0.029 & 0.922 \end{pmatrix} \qquad W = \begin{pmatrix} -0.038 & 0.042 \\ 0.023 & 0.016 \\ -0.025 & 0.029 \\ 0.08 & -0.045 \\ -0.04 & -0.042 \\ 1.355 & -2.946 \\ 0.875 & -0.296 \\ -0.029 & 0.922 \end{pmatrix}$$

$$K := \text{length}\left(Br^{\langle 0 \rangle}\right) \qquad K = 5$$

^ number of reference LM points in Br, and therefore the spline's non-affine part.

Matrix W (comprised of column vectors in x and y) was then used to weight distances U(r) for each point (x,y) on a grid to calculate the spline f(x,y). Due to the partioned nature of L, the last three rows of W give the weights for the affine part of the spline with columns representing x & y respectively. The remaining upper rows give weights based on each LM point in Br for the non-affine part. Note that the non-affine parts of W are based only on the upper left (K X K) sub-block of L^{-1}.

Now Bookstein defines the sub-block matrix L_K^{-1}:

$$invL_K := submatrix\left(L^{-1}, 0, K-1, 0, K-1\right)$$

$$invL_K = \begin{pmatrix} 0.04928 & -0.00228 & 0.03286 & -0.0744 & -0.00546 \\ -0.00228 & 0.03889 & -0.00042 & 0.04391 & -0.0801 \\ 0.03286 & -0.00042 & 0.02195 & -0.04847 & -0.00592 \\ -0.0744 & 0.04391 & -0.04847 & 0.15456 & -0.07561 \\ -0.00546 & -0.0801 & -0.00592 & -0.07561 & 0.16709 \end{pmatrix}$$

^ (K X K) upper left sub-block of L^{-1}

Note that given the data LM points V:

$$V = \begin{pmatrix} 3.972 & 6.535 \\ 6.697 & 4.118 \\ 6.539 & 7.236 \\ 5.402 & 6.453 \\ 5.776 & 5.114 \end{pmatrix} \qquad invL_K \cdot V = \begin{pmatrix} -0.038 & 0.042 \\ 0.023 & 0.016 \\ -0.025 & 0.029 \\ 0.08 & -0.045 \\ -0.04 & -0.042 \end{pmatrix}$$

^ is the first (K X 2) non-affine sub-block of W.

Since the term $L_K^{-1}V$ can be thought of as a linear transformation of points in column vectors of V into W, the transformation can be efficiently described in terms of this transformation's "canonical" directions (eigenvectors) and scalar changes (eigenvalues) in these directions of L_K^{-1}.

Eigenanalysis of L_K^{-1}:

$$\lambda := eigenvals\left(invL_K\right) \qquad \lambda = \begin{pmatrix} 0 \\ 0.148 \\ 0.2838 \\ 0 \\ 0 \end{pmatrix}$$

$$\lambda\lambda := \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \qquad \lambda\lambda = \begin{pmatrix} 0.148 \\ 0.284 \end{pmatrix}$$

< there are only two non-zero eigenvalues, so there are only two eigenvectors to extract.

$\varepsilon^{\langle 0 \rangle} := eigenvec\left(invL_K, \lambda_1\right)$ **< eigenvector in first column of ε corresponding to the lesser eigenvalue λ**

$\varepsilon^{\langle 1 \rangle} := eigenvec\left(invL_K, \lambda_2\right)$ **< eigenvector in second column of ε corresponding to the greater eigenvalue λ**

$$\varepsilon = \begin{pmatrix} -0.49407 & 0.21524 \\ -0.24154 & -0.32651 \\ -0.33697 & 0.13458 \\ 0.47001 & -0.65535 \\ 0.60257 & 0.63204 \end{pmatrix} \qquad \left|\varepsilon^{\langle 0 \rangle}\right| = 1 \qquad \left|\varepsilon^{\langle 1 \rangle}\right| = 1$$

< Eigenvectors, each standardized to a length of 1.

Bookstein calls these eigenvectors and associated splines "Principal Warps"

As with PCA, each coordinate value (single number in a column representing an eigenvector) can be considered a "loading". In this context, the "loadings" represent relative *potential* displacement of each reference LM point above or below a previously "uncharged" thin-plate spline for each Principal Warp separately for x and y. Bookstein (1991, p. 322) also shows (and Rolf implements in TPS) thin-plate splines of the Principal Warps themselves as deformations in the (x+Δx, y+Δy) plane. These diagrams show the principal ways points on the reference form will move simultaneously in (x,y) dependent only on choices of LM points made on the reference form, but lacking information about any specific change or comparison between reference and data sets. Principal Warps are not at all like the planar figures of total, affine, or non-affine deformations, also shown in the (x,y) plane, in the sense that the latter contain "charged spline" (comparison) information between forms. As with PCA, the eigenvectors (=Principal Warps) should probably be viewed as an intermediate step in constructing Bookstein's "Partial Warps" which are in fact interpretable in the same "charged spline" manner as the above figures. I won't construct the Principle Warp splines here, although I acknowledge possible value of looking at them in tpsSplin. More important, perhaps, is how forms *do in fact change or differ* in the *cardinal ways of change* (determined by the Principal Warps) imposed on the form by the choice of reference LM. *Actual change or comparisons* in these cardinal ways are recorded by Partial Warps. In my opinion, interpretation of Principal Warps and Partial Warps is often confused.

Partial Warps:

Partial warps are calculated using the spectral decomposition of L_K^{-1}:

$$invL_K = \begin{pmatrix} 0.04928 & -0.00228 & 0.03286 & -0.0744 & -0.00546 \\ -0.00228 & 0.03889 & -0.00042 & 0.04391 & -0.0801 \\ 0.03286 & -0.00042 & 0.02195 & -0.04847 & -0.00592 \\ -0.0744 & 0.04391 & -0.04847 & 0.15456 & -0.07561 \\ -0.00546 & -0.0801 & -0.00592 & -0.07561 & 0.16709 \end{pmatrix} \qquad \lambda\lambda = \begin{pmatrix} 0.148 \\ 0.284 \end{pmatrix} \qquad \varepsilon = \begin{pmatrix} -0.494 & 0.215 \\ -0.242 & -0.327 \\ -0.337 & 0.135 \\ 0.47 & -0.655 \\ 0.603 & 0.632 \end{pmatrix}$$

Eigen Partials:

$$P_0 := \lambda\lambda_0 \cdot \varepsilon^{\langle 0 \rangle} \cdot \varepsilon^{\langle 0 \rangle T} \qquad P_0 = \begin{pmatrix} 0.03613 & 0.01766 & 0.02464 & -0.03437 & -0.04407 \\ 0.01766 & 0.00864 & 0.01205 & -0.0168 & -0.02154 \\ 0.02464 & 0.01205 & 0.01681 & -0.02344 & -0.03005 \\ -0.03437 & -0.0168 & -0.02344 & 0.0327 & 0.04192 \\ -0.04407 & -0.02154 & -0.03005 & 0.04192 & 0.05374 \end{pmatrix}$$

$$P_1 := \lambda\lambda_1 \cdot \varepsilon^{\langle 1 \rangle} \cdot \varepsilon^{\langle 1 \rangle T} \qquad P_1 = \begin{pmatrix} 0.01315 & -0.01994 & 0.00822 & -0.04003 & 0.0386 \\ -0.01994 & 0.03025 & -0.01247 & 0.06072 & -0.05856 \\ 0.00822 & -0.01247 & 0.00514 & -0.02503 & 0.02414 \\ -0.04003 & 0.06072 & -0.02503 & 0.12187 & -0.11753 \\ 0.0386 & -0.05856 & 0.02414 & -0.11753 & 0.11335 \end{pmatrix}$$

$$P_0 + P_1 = \begin{pmatrix} 0.04928 & -0.00228 & 0.03286 & -0.0744 & -0.00546 \\ -0.00228 & 0.03889 & -0.00042 & 0.04391 & -0.0801 \\ 0.03286 & -0.00042 & 0.02195 & -0.04847 & -0.00592 \\ -0.0744 & 0.04391 & -0.04847 & 0.15456 & -0.07561 \\ -0.00546 & -0.0801 & -0.00592 & -0.07561 & 0.16709 \end{pmatrix}$$

< Partials sum to original L_K^{-1} matrix...

Calculating Partial Warps for a single point on the grid:

Considering the same point (x,y) above: $x := G_{x_{0,0}}$ $\quad x = 3.693$ $\qquad y := G_{y_{0,0}}$ $\quad y = 8.839$

We calculated weighted U functions and their sums:

$$WUx = \begin{pmatrix} -0.0786 \\ 0.411 \\ -1.4884 \\ 1.0554 \\ -0.3816 \end{pmatrix} \qquad WUy = \begin{pmatrix} 0.088 \\ 0.282 \\ 1.732 \\ -0.601 \\ -0.396 \end{pmatrix}$$

< calculated above

$$\sum WUx = -0.4822 \qquad \sum WUy = 1.1049$$

< sums calculated as above

unweighted U:

$$ULM = \begin{pmatrix} 2.067 \\ 17.7238 \\ 60.1247 \\ 13.2288 \\ 9.4959 \end{pmatrix}$$

These may be more directly calculated from the unweighted distance function U:

$$ULM^T \cdot invL_K \cdot V = (-0.4822 \quad 1.1049)$$

^ same sum as above

$$invL_K \cdot V = \begin{pmatrix} -0.038 & 0.0424 \\ 0.0232 & 0.0159 \\ -0.0248 & 0.0288 \\ 0.0798 & -0.0454 \\ -0.0402 & -0.0418 \end{pmatrix}$$

< weighting coefficients

And from Partials of spectral decomposition:

$$\lambda\lambda_0 \cdot \varepsilon^{\langle 0\rangle} \cdot \varepsilon^{\langle 0\rangle T} \cdot V = \begin{pmatrix} -0.0172 & 0.04 \\ -0.0084 & 0.0196 \\ -0.0117 & 0.0273 \\ 0.0164 & -0.0381 \\ 0.021 & -0.0488 \end{pmatrix} \qquad \lambda\lambda_1 \cdot \varepsilon^{\langle 1\rangle} \cdot \varepsilon^{\langle 1\rangle T} \cdot V = \begin{pmatrix} -0.0208 & 0.0024 \\ 0.0316 & -0.0037 \\ -0.013 & 0.0015 \\ 0.0634 & -0.0073 \\ -0.0612 & 0.0071 \end{pmatrix}$$

< partial weighting coefficients

$$\lambda\lambda_0 \cdot \varepsilon^{\langle 0\rangle} \cdot \varepsilon^{\langle 0\rangle T} \cdot V + \lambda\lambda_1 \cdot \varepsilon^{\langle 1\rangle} \cdot \varepsilon^{\langle 1\rangle T} \cdot V = \begin{pmatrix} -0.038 & 0.0424 \\ 0.0232 & 0.0159 \\ -0.0248 & 0.0288 \\ 0.0798 & -0.0454 \\ -0.0402 & -0.0418 \end{pmatrix}$$

< same >

$$invL_K \cdot V = \begin{pmatrix} -0.038 & 0.0424 \\ 0.0232 & 0.0159 \\ -0.0248 & 0.0288 \\ 0.0798 & -0.0454 \\ -0.0402 & -0.0418 \end{pmatrix}$$

$$\lambda\lambda = \begin{pmatrix} 0.148 \\ 0.284 \end{pmatrix}$$

Applying the weighting coefficients:

$$ULM^T \cdot \left(\lambda\lambda_0 \cdot \varepsilon^{\langle 0\rangle} \cdot \varepsilon^{\langle 0\rangle T} \cdot V\right) = (-0.4743 \quad 1.1039)$$

< displacement for Partial Warp associated with $\lambda\lambda_0$

$$ULM^T \cdot \left(\lambda\lambda_1 \cdot \varepsilon^{\langle 1\rangle} \cdot \varepsilon^{\langle 1\rangle T} \cdot V\right) = (-0.0079 \quad 0.00091)$$

< displacement for Partial Warp associated with $\lambda\lambda_1$

$$ULM^T \cdot \left(\lambda\lambda_0 \cdot \varepsilon^{\langle 0\rangle} \cdot \varepsilon^{\langle 0\rangle T} \cdot V\right) + ULM^T \cdot \left(\lambda\lambda_1 \cdot \varepsilon^{\langle 1\rangle} \cdot \varepsilon^{\langle 1\rangle T} \cdot V\right) = (-0.4822 \quad 1.1049)$$

< total displacement

^ same total displacement summed above

Now doing Partial Warp calculations for all point on the Grid:

$x_{j,k} := Gx_{j,k} \qquad y_{j,k} := Gy_{j,k}$ **< specifying each point in Gx/Gy in turn**

Unweighted U:

$$UD_{j,k} := \left(Ud0_{j,k} \quad Ud1_{j,k} \quad Ud2_{j,k} \quad Ud3_{j,k} \quad Ud4_{j,k}\right)$$

$$UD_{0,0} = (2.067 \quad 17.724 \quad 60.125 \quad 13.229 \quad 9.496)$$

^ first point same as single point above

$$invL_K \cdot V = \begin{pmatrix} -0.038 & 0.0424 \\ 0.0232 & 0.0159 \\ -0.0248 & 0.0288 \\ 0.0798 & -0.0454 \\ -0.0402 & -0.0418 \end{pmatrix}$$

< weighting coefficients

$$\lambda\lambda_0 \cdot \varepsilon^{\langle 0\rangle} \cdot \varepsilon^{\langle 0\rangle T} \cdot V = \begin{pmatrix} -0.0172 & 0.04 \\ -0.0084 & 0.0196 \\ -0.0117 & 0.0273 \\ 0.0164 & -0.0381 \\ 0.021 & -0.0488 \end{pmatrix} \qquad \lambda\lambda_1 \cdot \varepsilon^{\langle 1\rangle} \cdot \varepsilon^{\langle 1\rangle T} \cdot V = \begin{pmatrix} -0.0208 & 0.0024 \\ 0.0316 & -0.0037 \\ -0.013 & 0.0015 \\ 0.0634 & -0.0073 \\ -0.0612 & 0.0071 \end{pmatrix}$$

< partial weighting coefficients

Applying the weighting coefficients:

$$PDD0_{j,k} := UD_{j,k} \cdot \left(\lambda\lambda_0 \cdot \varepsilon^{\langle 0\rangle} \cdot \varepsilon^{\langle 0\rangle T} \cdot V\right) \qquad PDD0_{0,0} = (-0.4743 \quad 1.1039)$$

< displacement for Partial Warp associated with $\lambda\lambda_0$

$$PDD1_{j,k} := UD_{j,k} \cdot \left(\lambda\lambda_1 \cdot \varepsilon^{\langle 1\rangle} \cdot \varepsilon^{\langle 1\rangle T} \cdot V\right) \qquad PDD1_{0,0} = (-0.0079 \quad 0.00091)$$

< displacement for Partial Warp associated with $\lambda\lambda_1$

$$TDD_{j,k} := UD_{j,k} \cdot \left(invL_K \cdot V\right) \qquad TDD_{0,0} = (-0.4822 \quad 1.1049)$$

< total displacement

^ evaluation of first point same as above

Now Partial Warp calculation for LM points in the reference form Br:

$$Brx = \begin{pmatrix} 3.693 \\ 6.583 \\ 6.776 \\ 4.819 \\ 5.697 \end{pmatrix} \quad Bry = \begin{pmatrix} 10.382 \\ 8.839 \\ 12.087 \\ 11.205 \\ 10.075 \end{pmatrix}$$

< specifying (x,y) for each point in Br.

$$\lambda\lambda = \begin{pmatrix} 0.148 \\ 0.284 \end{pmatrix} \quad \varepsilon = \begin{pmatrix} -0.494 & 0.215 \\ -0.242 & -0.327 \\ -0.337 & 0.135 \\ 0.47 & -0.655 \\ 0.603 & 0.632 \end{pmatrix}$$

Unweighted U:

$$BrU = \begin{pmatrix} 0 & 25.472 & 31.251 & 1.294 & 5.81 \\ 25.472 & 0 & 24.98 & 18.851 & 1.939 \\ 31.251 & 24.98 & 0 & 7.036 & 8.602 \\ 1.294 & 18.851 & 7.036 & 0 & 1.467 \\ 5.81 & 1.939 & 8.602 & 1.467 & 0 \end{pmatrix}$$

$$invL_K \cdot V = \begin{pmatrix} -0.038 & 0.0424 \\ 0.0232 & 0.0159 \\ -0.0248 & 0.0288 \\ 0.0798 & -0.0454 \\ -0.0402 & -0.0418 \end{pmatrix}$$

< weighting coefficients

$$\lambda\lambda_0 \cdot \varepsilon^{\langle 0\rangle} \cdot \varepsilon^{\langle 0\rangle T} \cdot V = \begin{pmatrix} -0.0172 & 0.04 \\ -0.0084 & 0.0196 \\ -0.0117 & 0.0273 \\ 0.0164 & -0.0381 \\ 0.021 & -0.0488 \end{pmatrix} \quad \lambda\lambda_1 \cdot \varepsilon^{\langle 1\rangle} \cdot \varepsilon^{\langle 1\rangle T} \cdot V = \begin{pmatrix} -0.0208 & 0.0024 \\ 0.0316 & -0.0037 \\ -0.013 & 0.0015 \\ 0.0634 & -0.0073 \\ -0.0612 & 0.0071 \end{pmatrix}$$

< partial weighting coefficients

$m := 0 .. 1$

Applying the weighting coefficients:

$$PBr0x_i := \left[BrU^{\langle i\rangle T} \cdot \left(\lambda\lambda_0 \cdot \varepsilon^{\langle 0\rangle} \cdot \left(\varepsilon^{\langle 0\rangle}\right)^T \cdot V^{\langle 0\rangle}\right)\right]_0 \qquad PBr0y_i := \left[BrU^{\langle i\rangle T} \cdot \left(\lambda\lambda_0 \cdot \varepsilon^{\langle 0\rangle} \cdot \left(\varepsilon^{\langle 0\rangle}\right)^T \cdot V^{\langle 1\rangle}\right)\right]_0$$

$$PBr1x_i := \left[BrU^{\langle i\rangle T} \cdot \left(\lambda\lambda_1 \cdot \varepsilon^{\langle 1\rangle} \cdot \varepsilon^{\langle 1\rangle T} \cdot V^{\langle 0\rangle}\right)\right]_0 \qquad PBr1y_i := \left[BrU^{\langle i\rangle T} \cdot \left(\lambda\lambda_1 \cdot \varepsilon^{\langle 1\rangle} \cdot \varepsilon^{\langle 1\rangle T} \cdot V^{\langle 1\rangle}\right)\right]_0$$

$$TBrx_i := \left[BrU^{\langle i\rangle T} \cdot \left(invL_K \cdot V^{\langle 0\rangle}\right)\right]_0 \qquad TBry_i := \left[BrU^{\langle i\rangle T} \cdot \left(invL_K \cdot V^{\langle 1\rangle}\right)\right]_0$$

$$PBr0x = \begin{pmatrix} -0.437772 \\ -0.3820616 \\ -0.4520239 \\ -0.2325326 \\ -0.1931488 \end{pmatrix} \quad PBr0y = \begin{pmatrix} 1.01895 \\ 0.88928 \\ 1.05212 \\ 0.54124 \\ 0.44957 \end{pmatrix}$$

< displacement for Partial Warp associated with $\lambda\lambda_0$

$$PBr1x = \begin{pmatrix} 0.1245269 \\ 0.2209934 \\ 0.0584597 \\ 0.3872926 \\ -0.07871 \end{pmatrix} \quad PBr1y = \begin{pmatrix} -0.01441 \\ -0.02557 \\ -0.00676 \\ -0.04482 \\ 0.00911 \end{pmatrix}$$

< displacement for Partial Warp associated with $\lambda\lambda_1$

Totals are the same as calculation above:

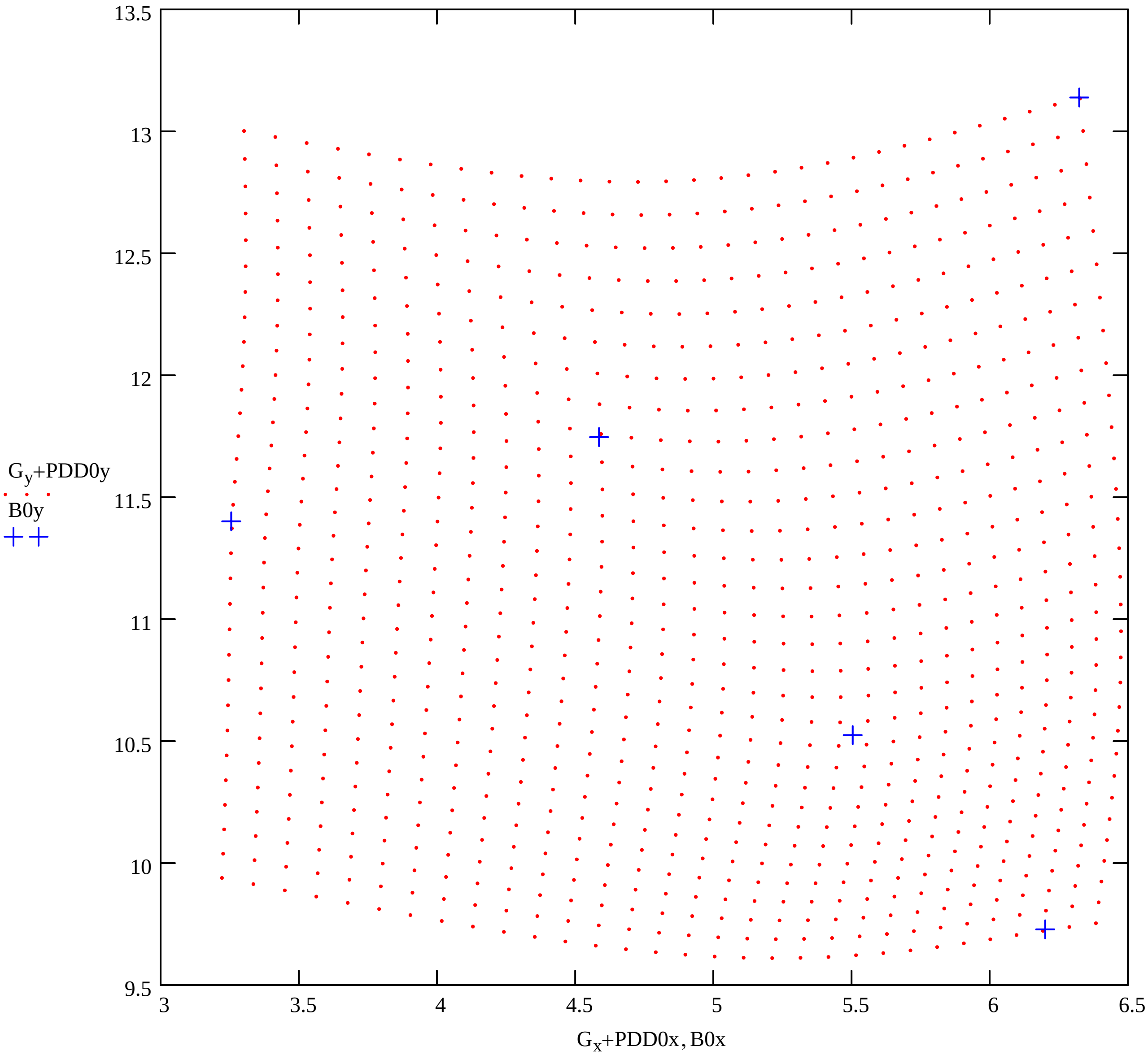
$$TBrx = \begin{pmatrix} -0.31325 \\ -0.16107 \\ -0.39356 \\ 0.15476 \\ -0.27186 \end{pmatrix} \quad TBry = \begin{pmatrix} 1.00454 \\ 0.86371 \\ 1.04536 \\ 0.49642 \\ 0.45868 \end{pmatrix}$$

$$BrWUsum_x = \begin{pmatrix} -0.31325 \\ -0.16107 \\ -0.39356 \\ 0.15476 \\ -0.27186 \end{pmatrix} \quad BrWUsum_y = \begin{pmatrix} 1.00454 \\ 0.86371 \\ 1.04536 \\ 0.49642 \\ 0.45868 \end{pmatrix}$$

Plot of Partial Warp associated with $\lambda\lambda_0$: $\lambda\lambda_0 = 0.148$

$B0x := Brx + PBr0x$ $B0y := Bry + PBr0y$ **< for LM points in Br**

$PDD0x_{j,k} := \left[\left(PDD0_{j,k}\right)^{\langle 0 \rangle}\right]_0$ $PDD0y_{j,k} := \left[\left(PDD0_{j,k}\right)^{\langle 1 \rangle}\right]_0$ **< extracting points from nested matrix for grid**



Plot of Partial Warp associated with $\lambda\lambda_1$: $\lambda\lambda_1 = 0.284$

$B1x := Brx + PBr1x$ $B1y := Bry + PBr1y$ **< for LM points in Br**

$PDD1x_{j,k} := \left[\left(PDD1_{j,k}\right)^{\langle 0\rangle}\right]_0$ $PDD1y_{j,k} := \left[\left(PDD1_{j,k}\right)^{\langle 1\rangle}\right]_0$ **< extracting points from nested matrix for grid**