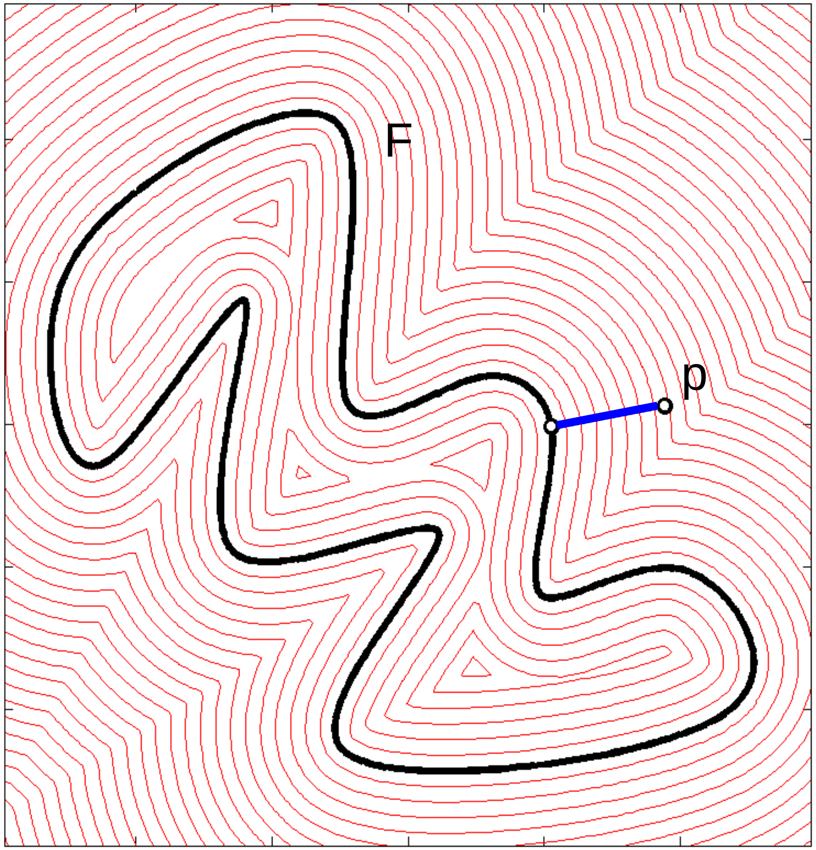
Distance Functions

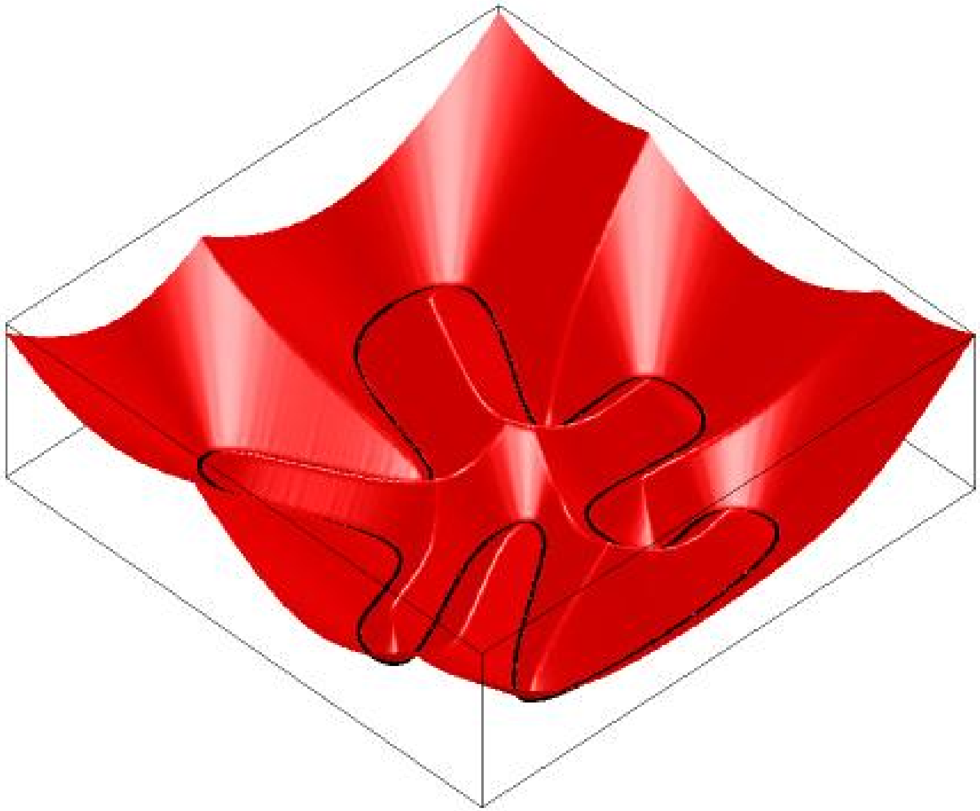
Distance function

- Given: geometric object *F* (curve, surface, solid, ...)
- Assigns to each point the shortest distance from *F*
- Level sets of the distance function are trimmed offsets



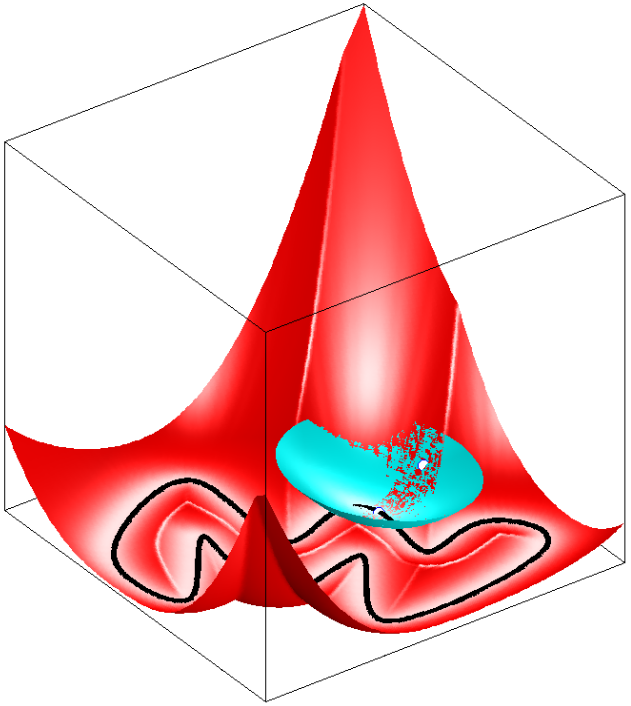
Graph surfaces of distance functions

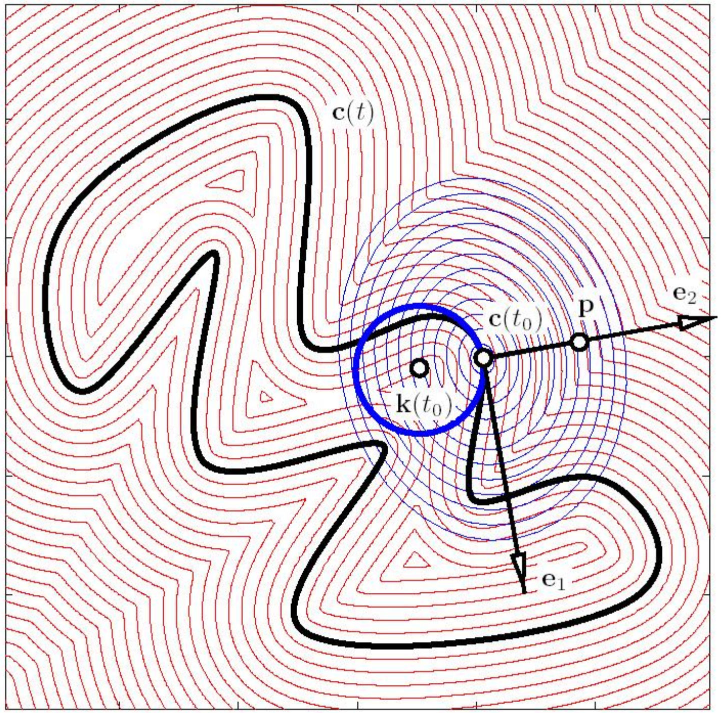
- Graph surfaces of distance functions: (developable) surfaces of constant slope
- Each tangent plane has the inclination angle 45 degrees
- Not smooth at the base curve and at the cut locus



squared distance function

- For geometric optimization algorithms: *local quadratic approximation of the* ***squared distance function***
- Outside cut locus



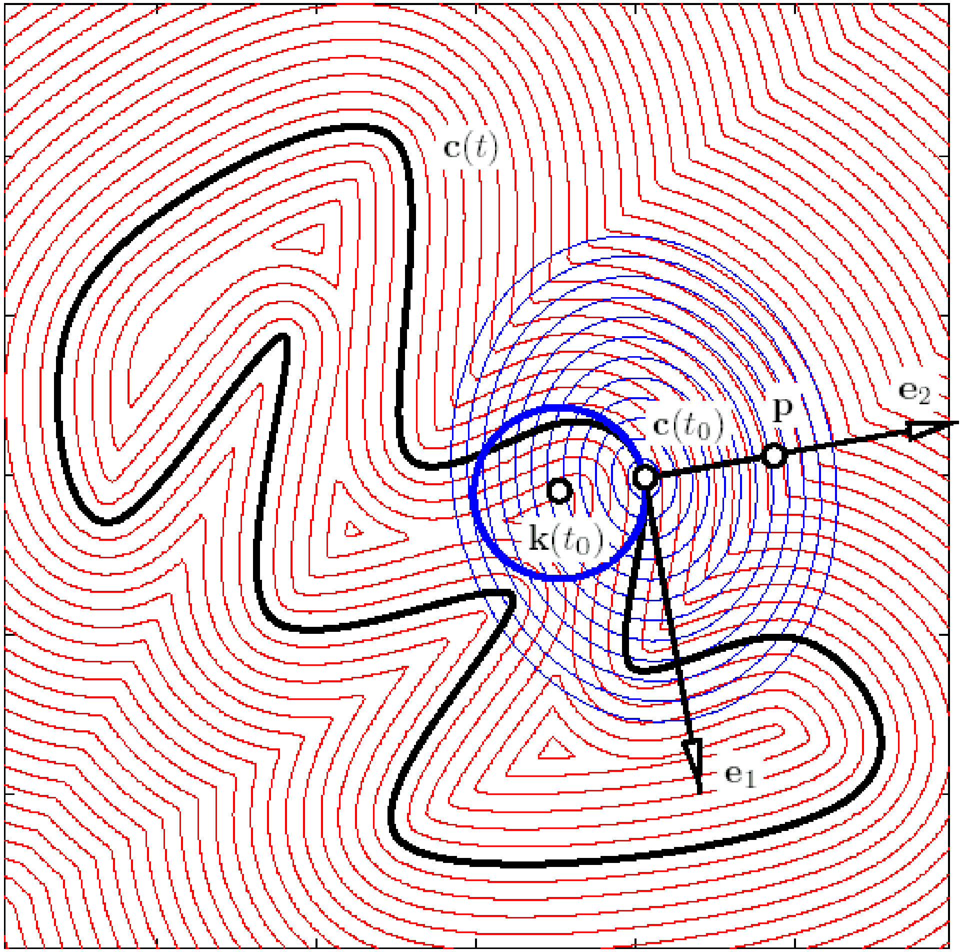


local quadratic approximation of the function d^2

quadratic Taylor approximant F_d of the squared distance function at p:

$$F_d(x_1, x_2) = \frac{d}{d-\rho} x_1^2 + x_2^2$$

x_1, x_2 are the coordinates in the Frenet frame at the normal footpoint $c(t_0)$ of p.



quadratic approximation of d^2

Taylor approximant

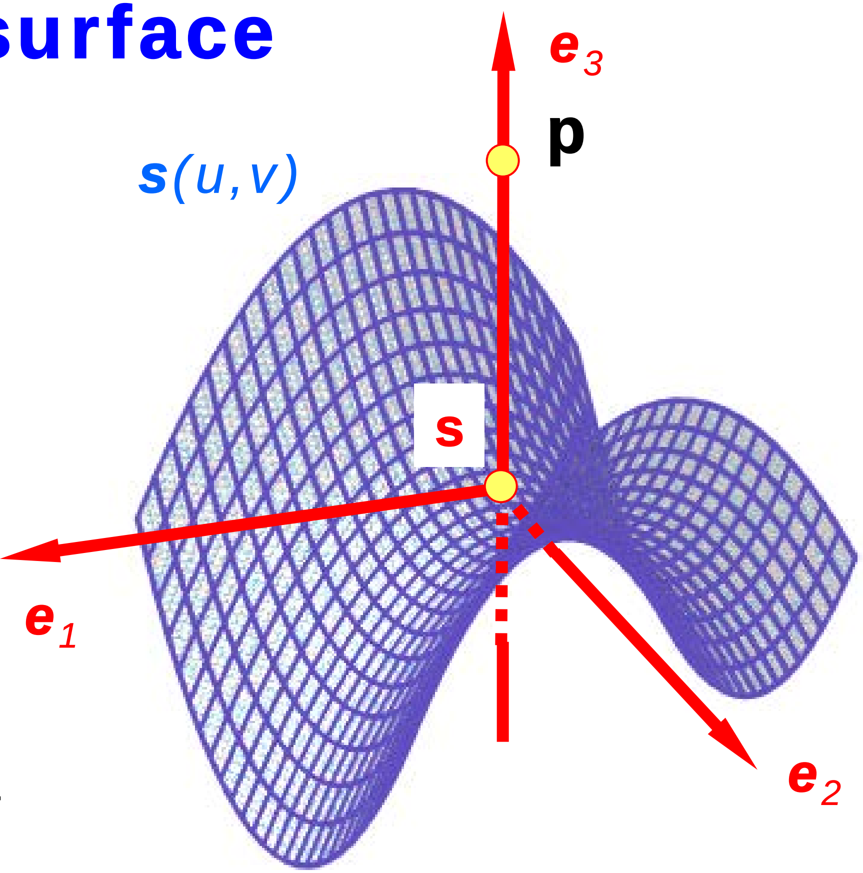
$$F_d(x_1, x_2) = \frac{d}{d-\rho} x_1^2 + x_2^2$$

for d=0 (p on the curve):

$$F_d(x_1, x_2) = x_2^2$$

Squared distance to the tangent at $c(t_0)$=p.

Second order approximant of the squared distance function to a **surface**

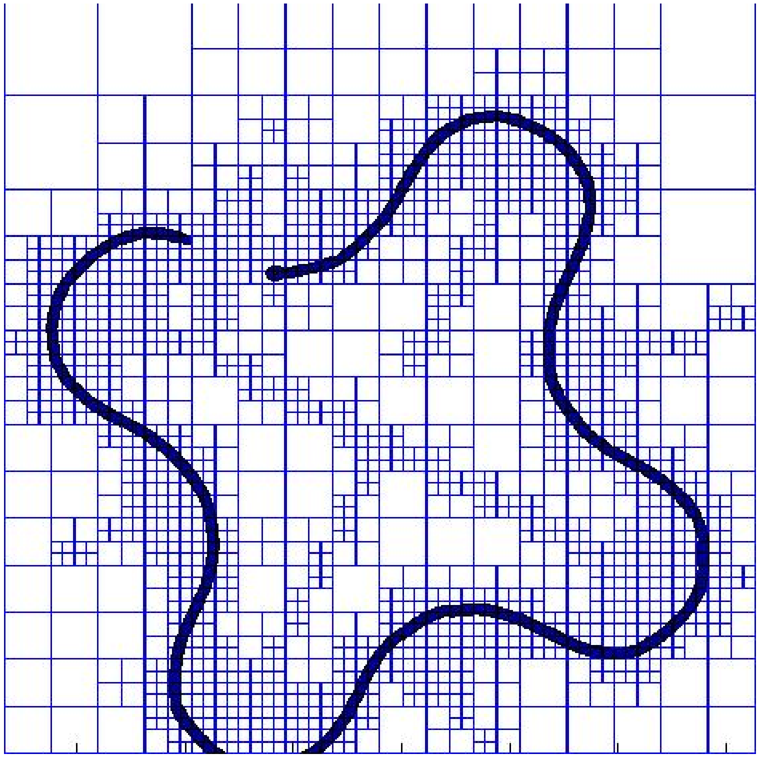


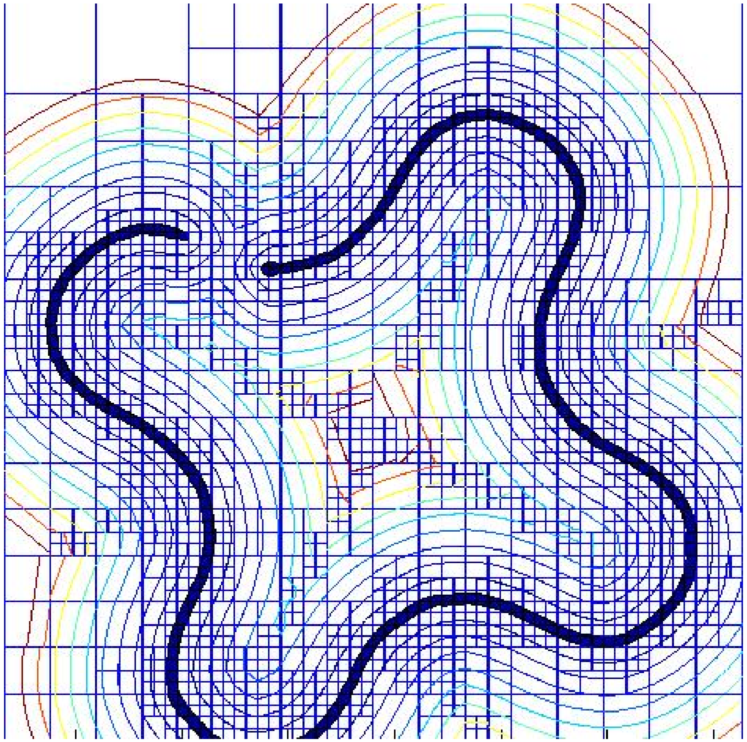
- The second order Taylor approximant F_d of the squared distance function to a surface at a point ***p*** is expressed in the principal frame at the normal footpoint **s** via

$$F_d(x_1, x_2, x_3) = \frac{d}{d-\rho_1} x_1^2 + \frac{d}{d-\rho_2} x_2^2 + x_3^2$$

Data structure

- Adaptive data structure whose cells carry local quadratic approximants of the squared distance function

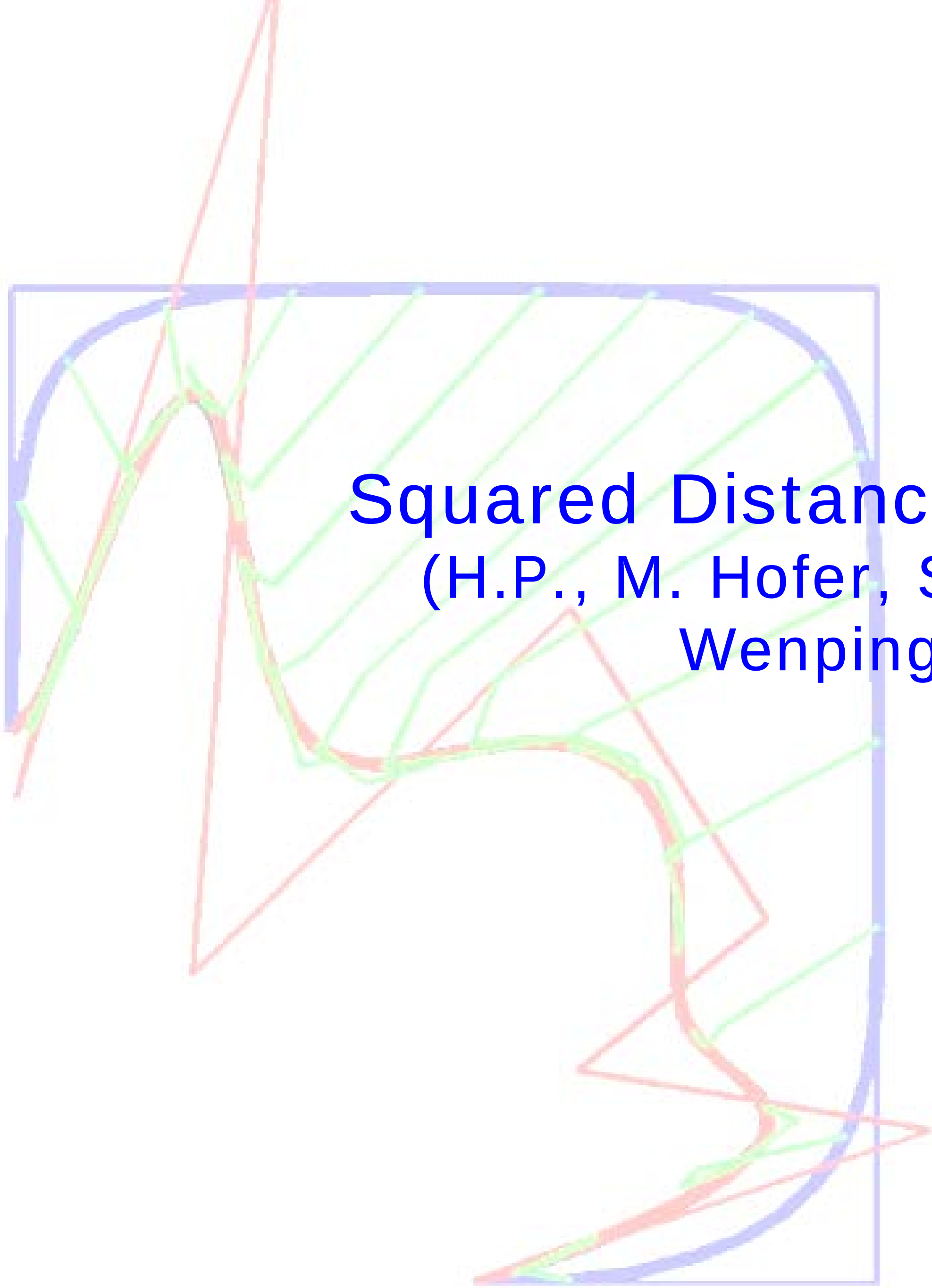




Curve and surface fitting

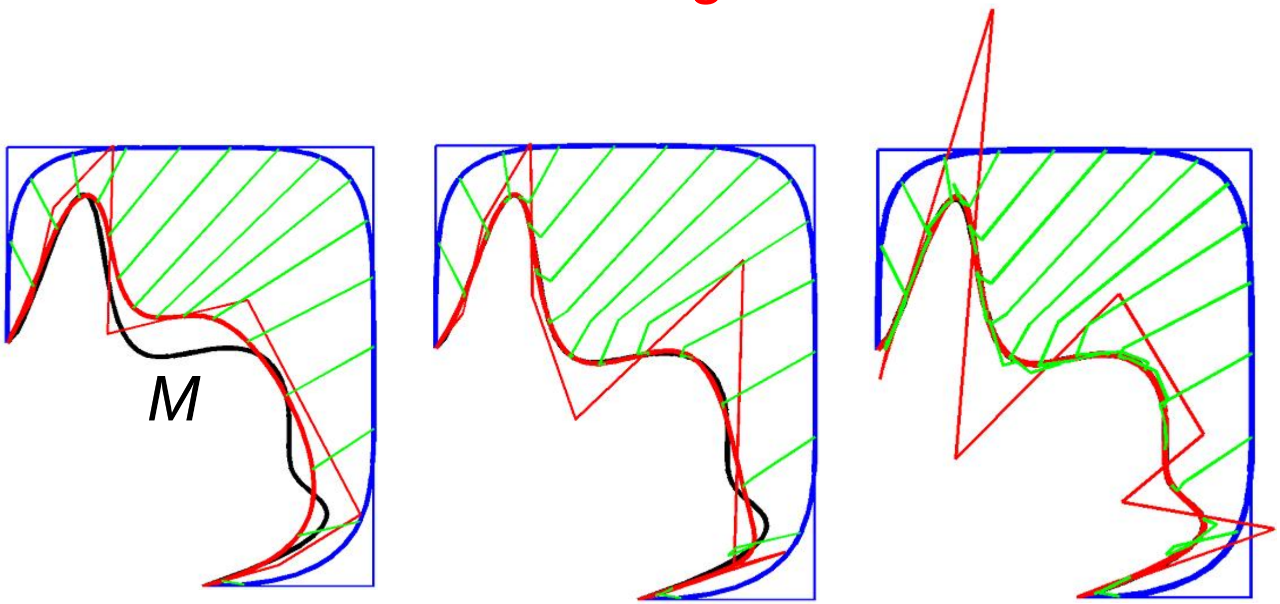
Institute of Geometry

Squared Distance Minimization I
(H.P., M. Hofer, S. Leopoldseder, Wenping Wang)



Squared Distance Minimization

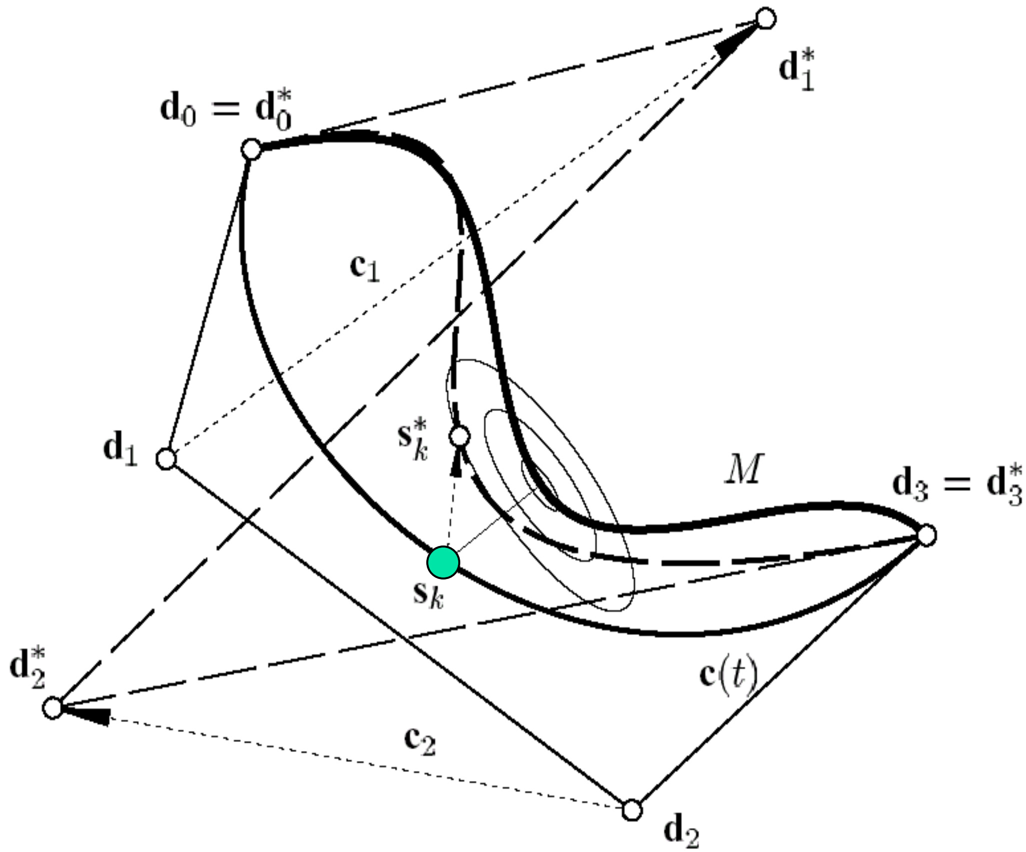
- B-spline curve or surface is iteratively deformed in the *squared distance field* of the model shape M (shape to be approximated: point cloud, mesh, implicit curve/surface,...)
- *Errors measured orthogonal to M*



Algorithm overview

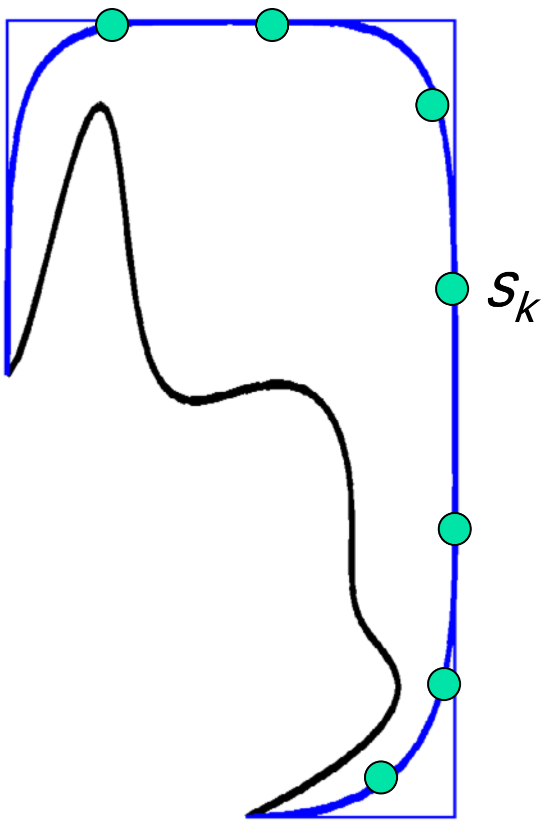
Iteration:

- evaluate current B-spline form at discrete number of 'sensor' points s_k
- compute local quadratic approximants of the squared distance field d^2 to M at s_k
- change control points such that sensors come closer to M (done with help of quadratic approximants to d^2)



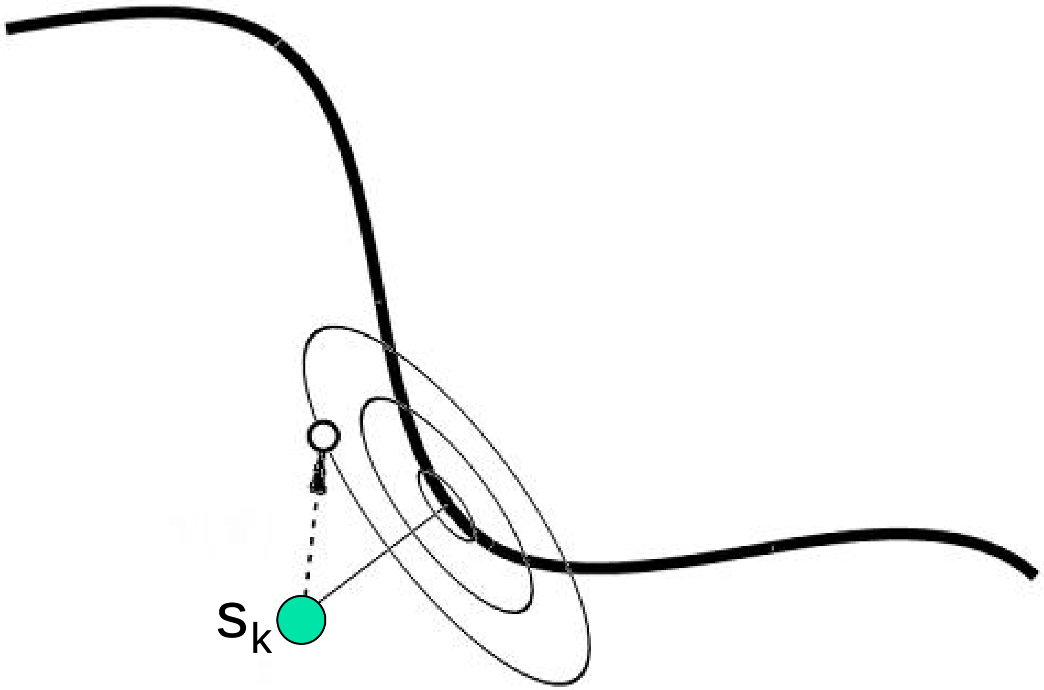
Initialization and sensor points

- Choose an initial shape s (B-spline curve or surface)
- compute sufficient number of *`sensor points s_k'* on s (evaluation at chosen parameter values)
- Note: if control points are displaced with vectors c_i (knots and weights fixed), the new *sensor locations s_k^* depend linearly on c_i*



local quadratic approximants at sensor points

- At each sensor point s_k, compute a nonnegative local quadratic approximant to the squared distance field of model shape

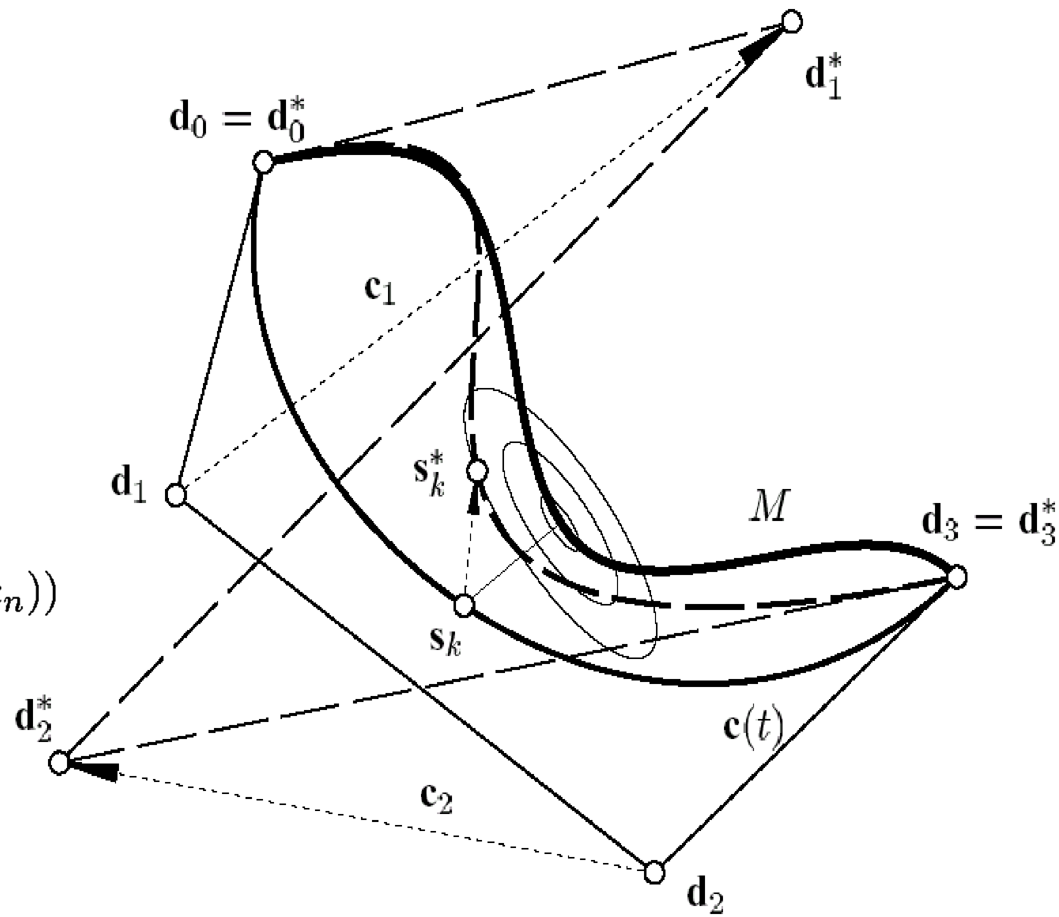


move closer to model shape

- Compute displacement vectors c_i of the control points such that sensors come closer to model shape by minimizing

$$F = \sum_{k=1}^{N} F_d^k(L_k(\mathbf{d}_1 + \mathbf{c}_1, \ldots, \mathbf{d}_n + \mathbf{c}_n))$$

- This is a quadratic function in c_i



Remarks

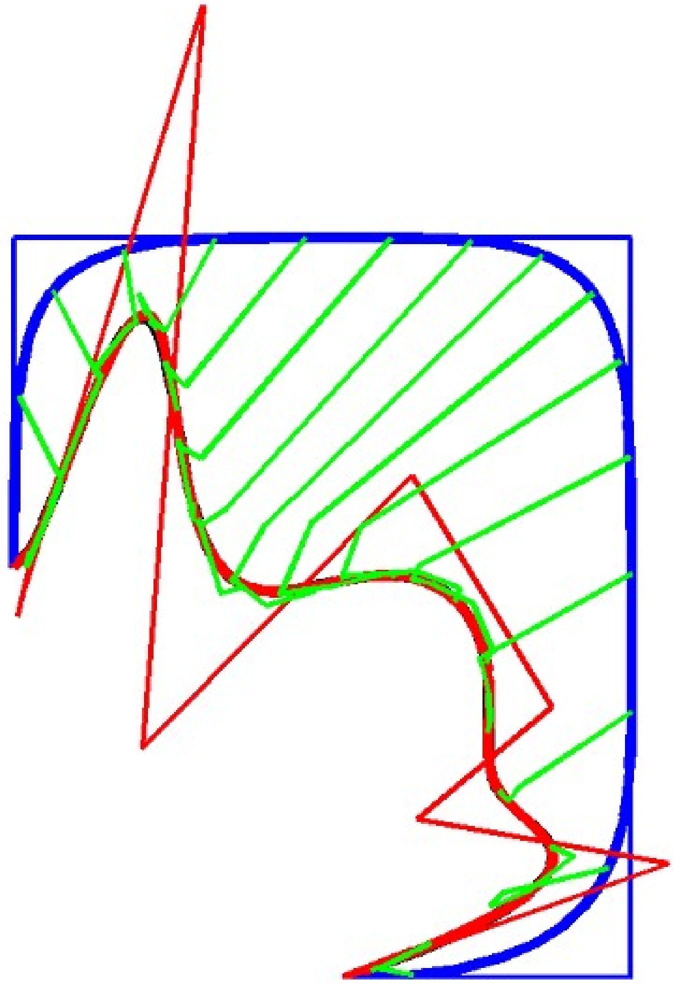
- For *regularization* and avoidance of overlappings/foldings of the final shape, an adequately weighted smoothing term F_s can be added to the functional, e.g. for surfaces:

$$F_s = \int\int (\mathbf{s}_{uu}^2 + 2\mathbf{s}_{uv}^2 + \mathbf{s}_{vv}^2)\,du\,dv$$

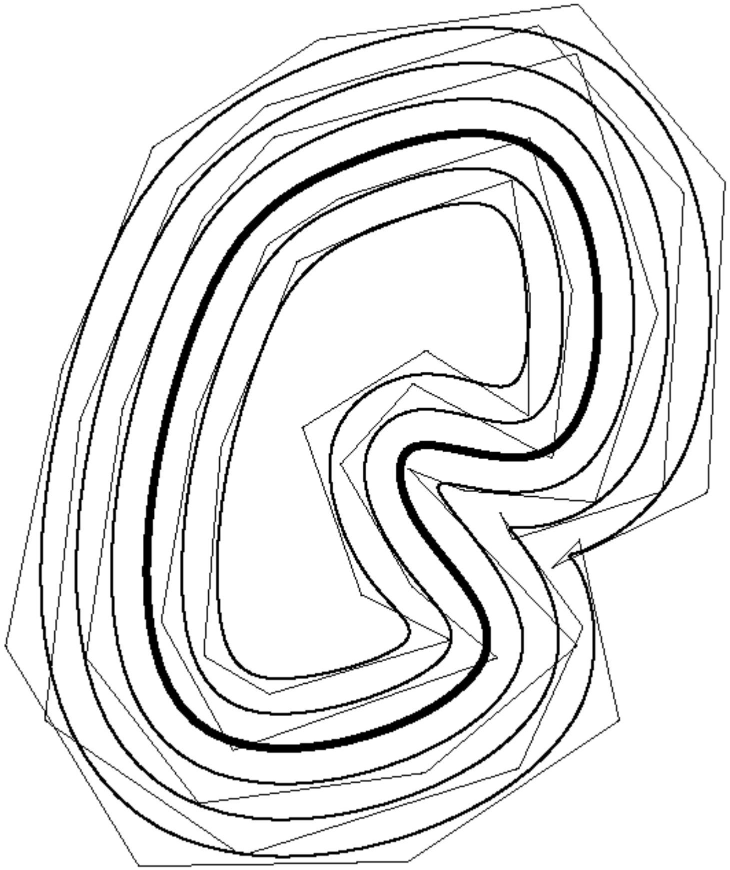
  Amounts to minimization of quadratic function

Example: curve approximation

- Displacement polygons of sensor points show that later moves are mainly in tangential direction
- Related to the computation of a good parameterization



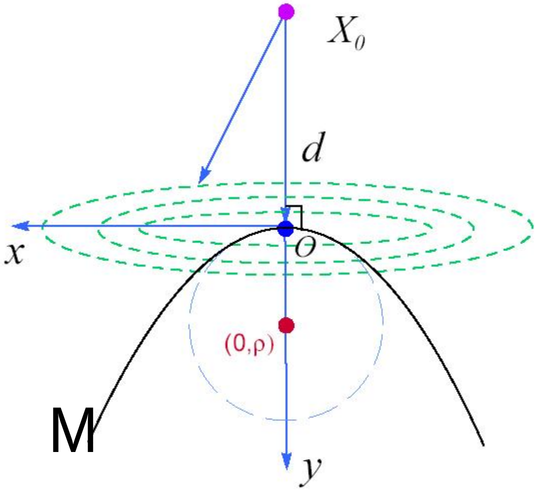
Example: Offset approximation



Sensor points do NOT move towards offset along normals of the progenitor curve

SDM, TDM, PDM

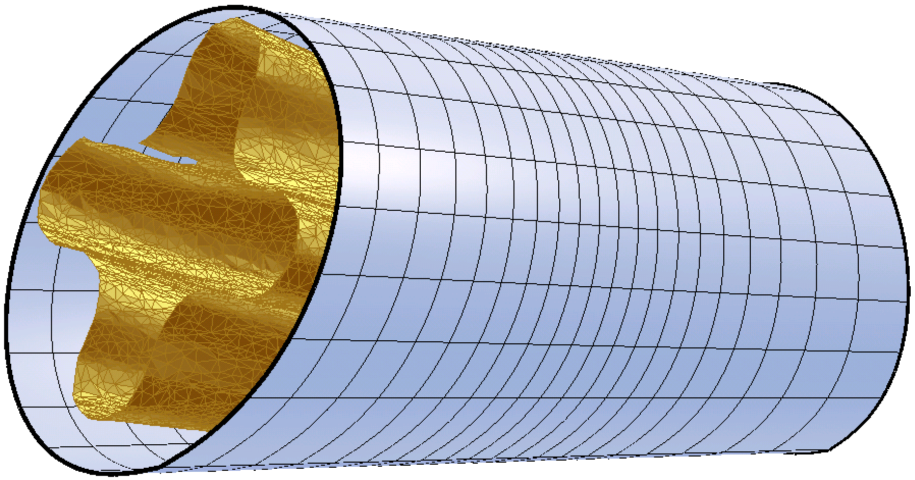
- SDM employs curvature at closest point to compute 2nd order approximation of squared distance function
- TDM uses only *squared distance to tangent at footpoint* (requires regularization and stepsize control)
- PDM (mostly used) employs squared distances to footpoint
- TDM regularized with a PDM term works very well



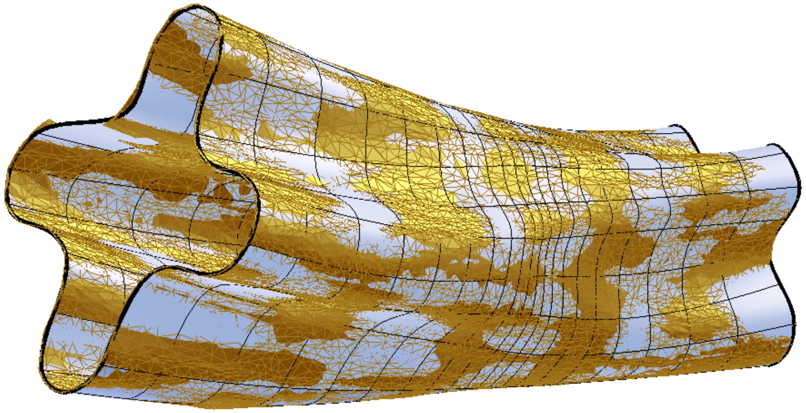
SDM, TDM, PDM

- SDM: Newton algorithm with *quadratic convergence*
- TDM: Gauss-Newton iteration for nonlinear least squares problem; *quadratic convergence for zero residual problem* and good initial position; requires regularization (add multiple of identity matrix to approximate Hessian or add a PDM term with low weight) and step size control
- PDM: only linearly convergent and prone to be trapped in local minimizer

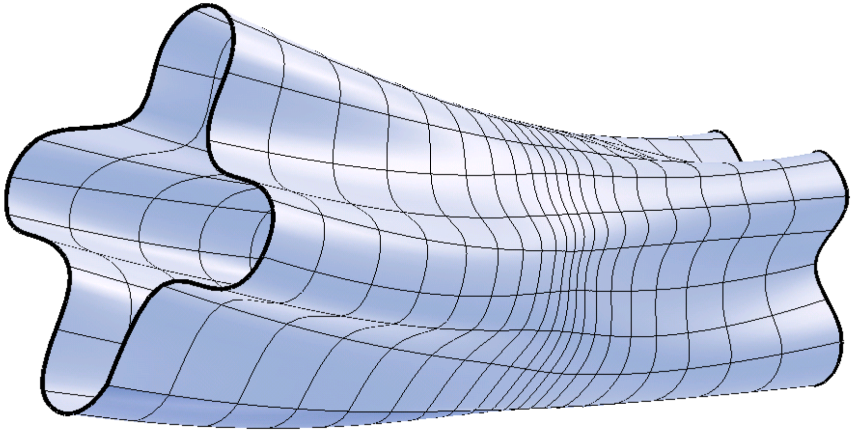
Example:
NURBS surface approximation



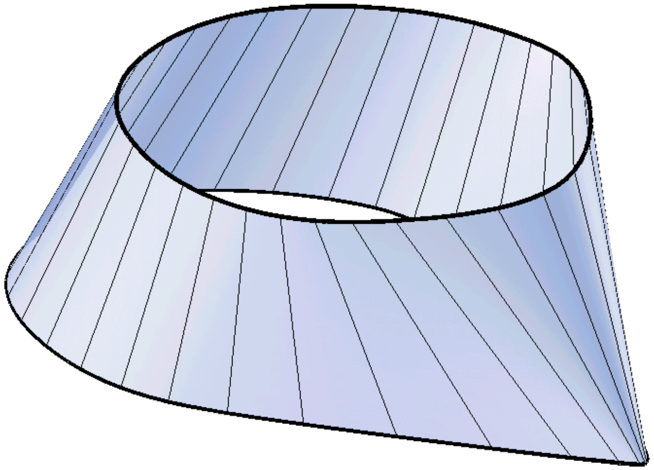
Model surface (gold) and initial position of active NURBS surface (blue)

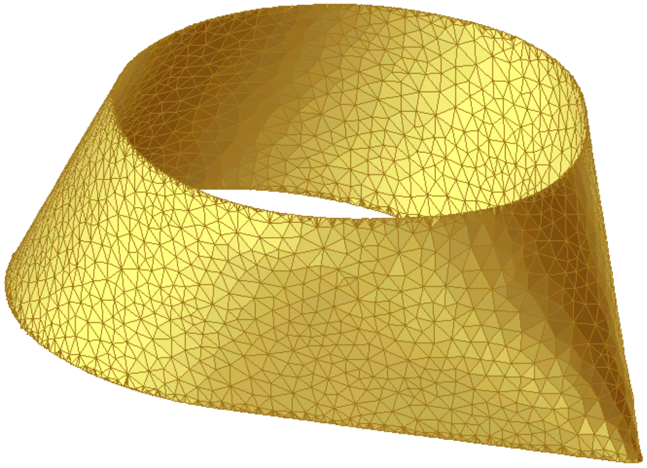


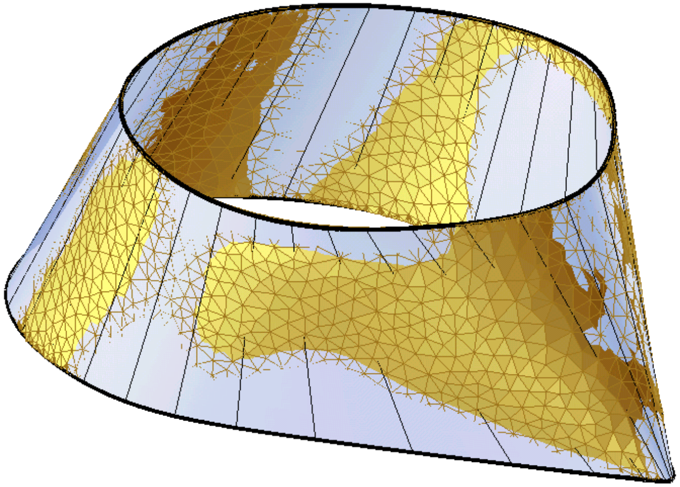
Final position of active NURBS surface



Approximation with ruled surface

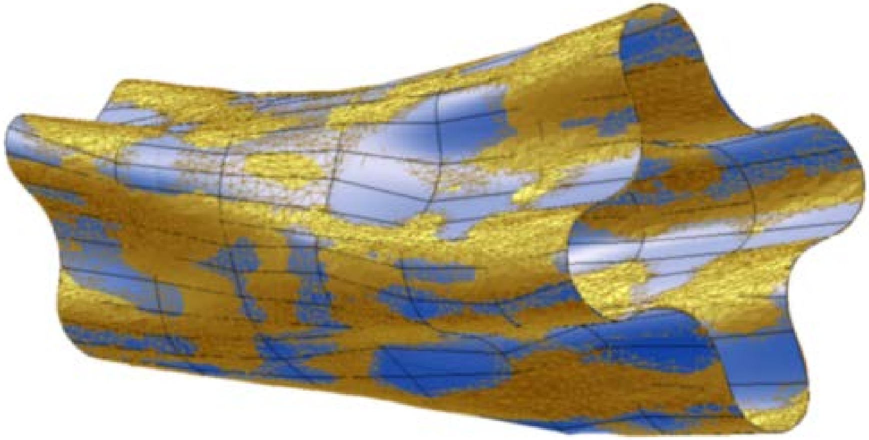


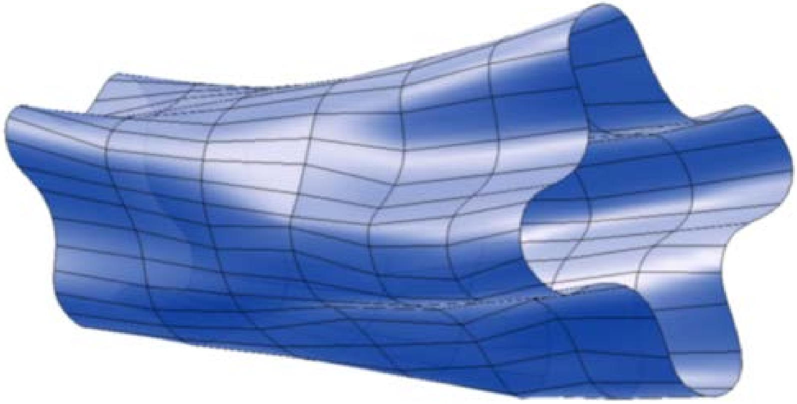




- active B-spline surface of degree (1,n)

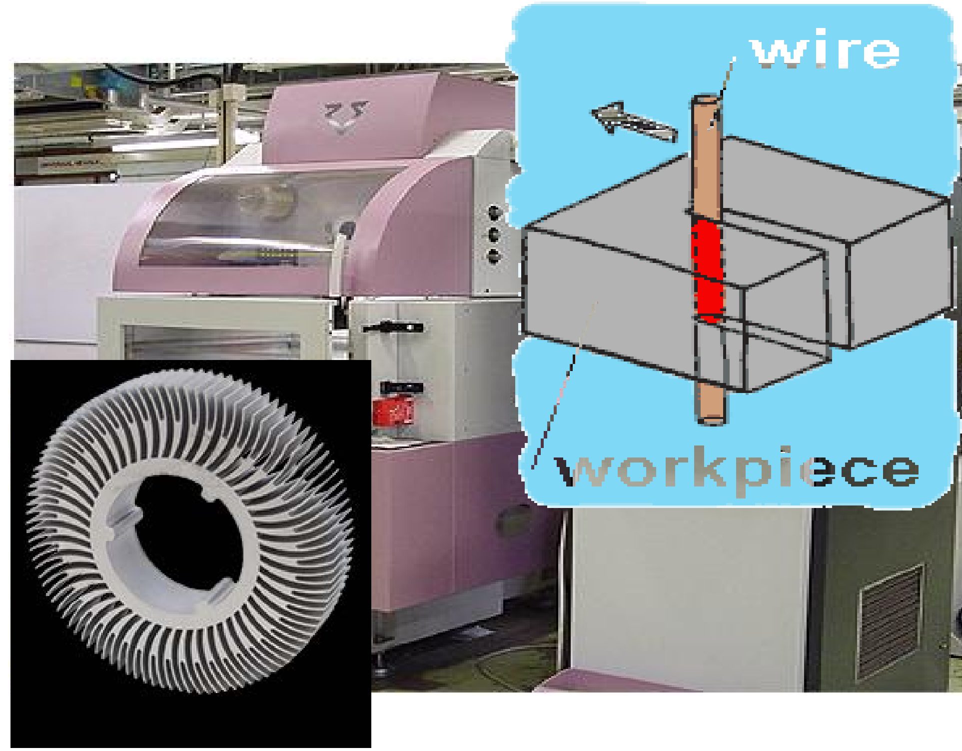
Approximation by a piecewise ruled surface



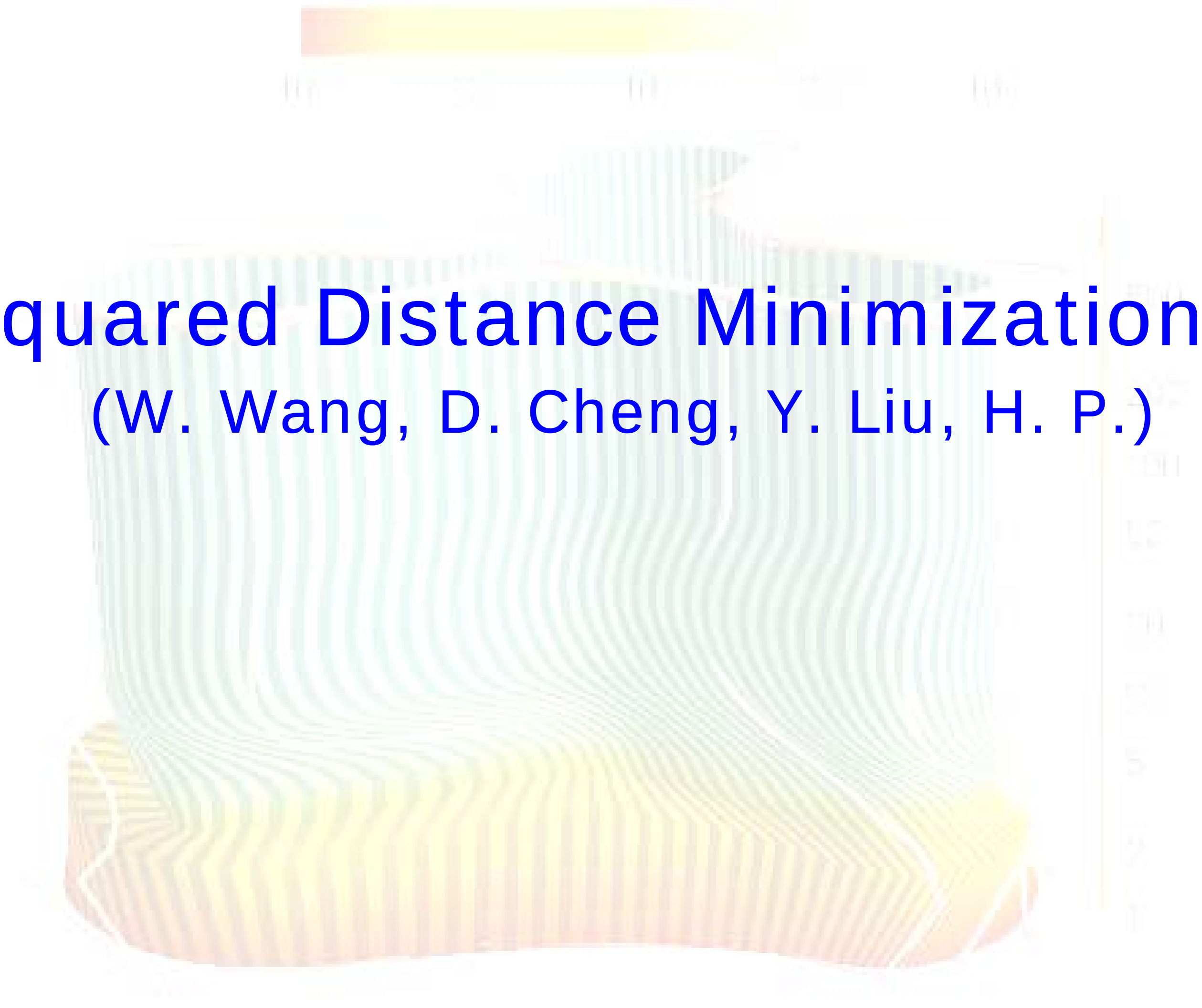


Application of ruled surface approximation

hot wire cutting
(styrofoam)



wire EDM
(metal)

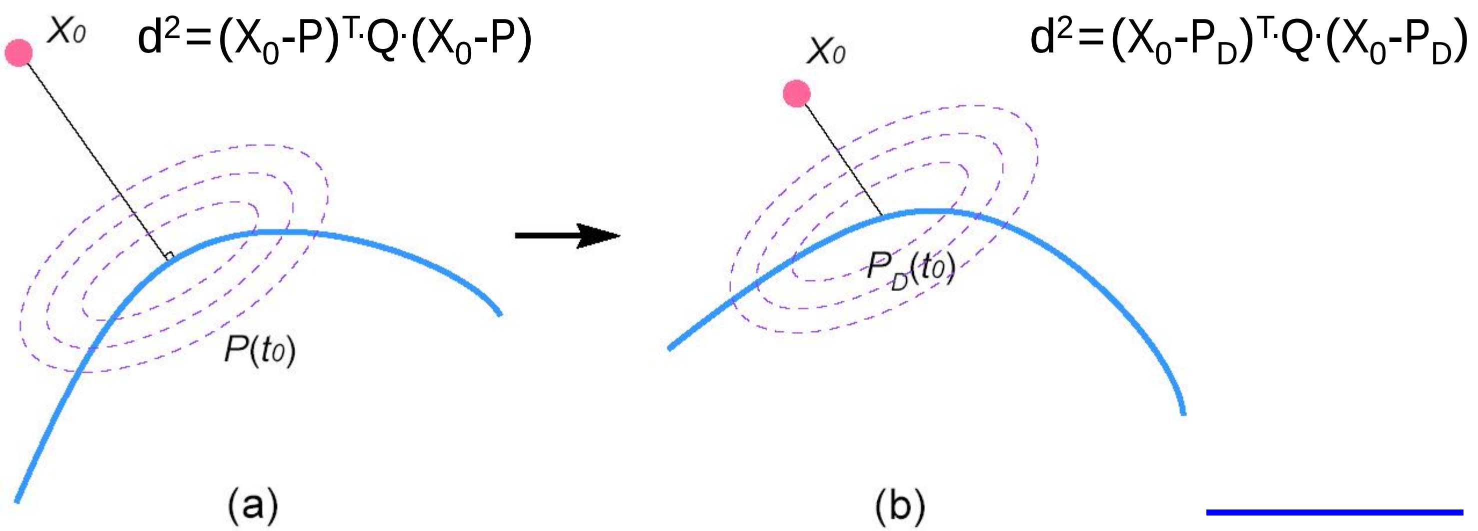


Squared Distance Minimization II

(W. Wang, D. Cheng, Y. Liu, H. P.)

Another type of SDM

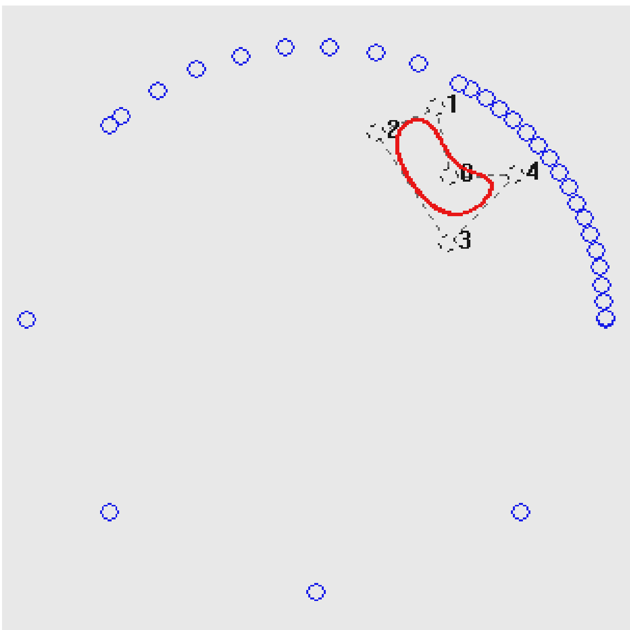
- *Error measurement orthogonal to the active curve/surface*
- Squared distance field (quadratic approximants) attached to the moving curve/surface
- Quasi-Newton method (neglects rotation of tangent and change of curvature)



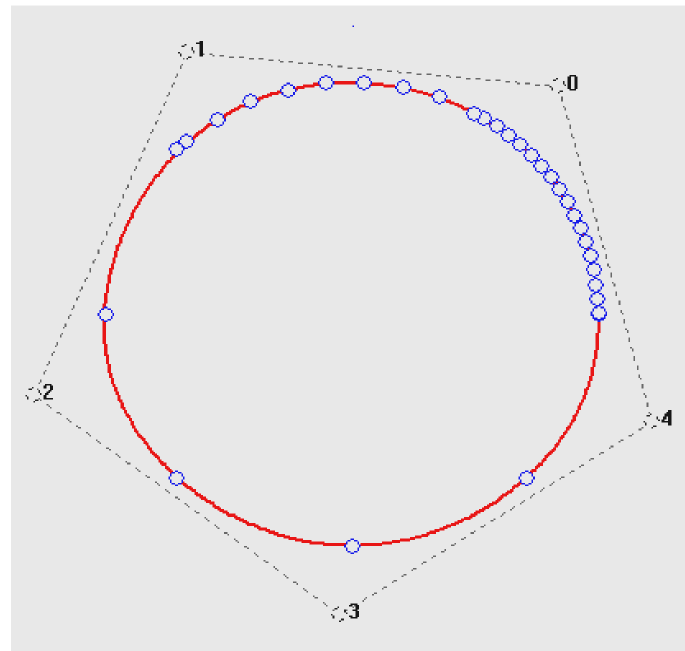
simplifications

- TDM: measures the error via squared distance to tangent at closest point; requires regularization and step size control
- PDM (standard approach): measures error via squared distance to closest point
- TDM regularized with a PDM term (small weight) works very well

Example 1: Non-uniform data

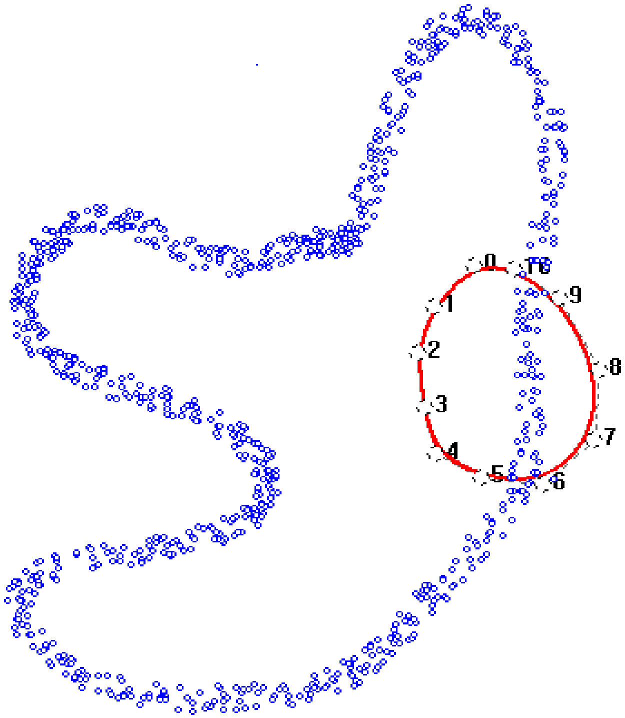


Initial position

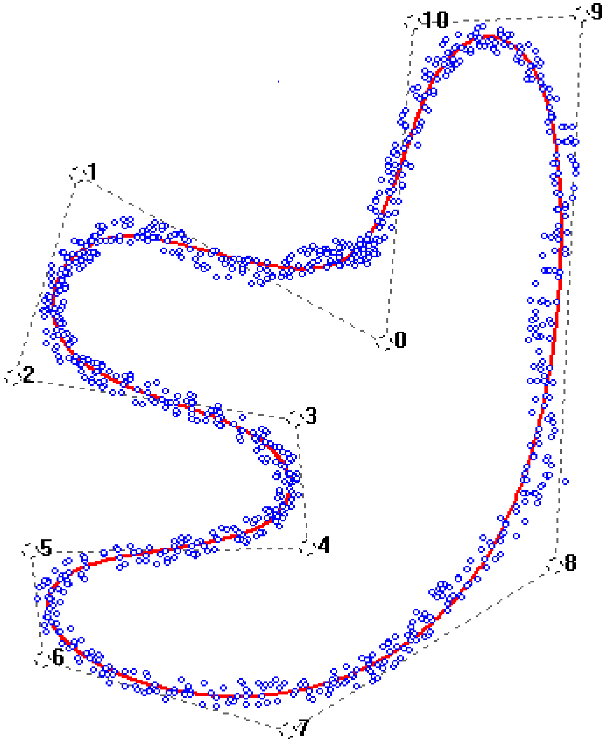


SDM after 8 iterations

Example 2: Thick point cloud

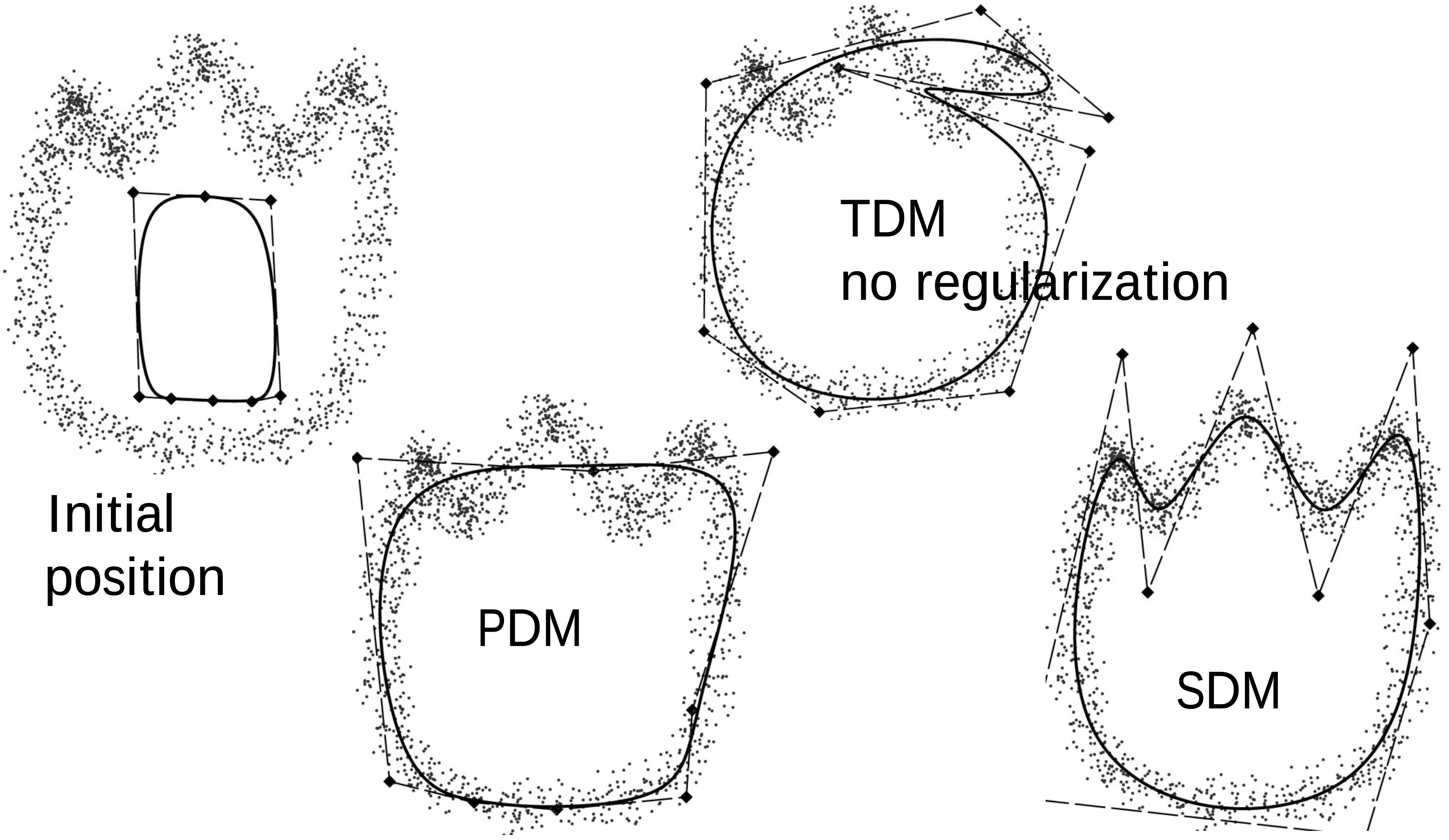


Initial position

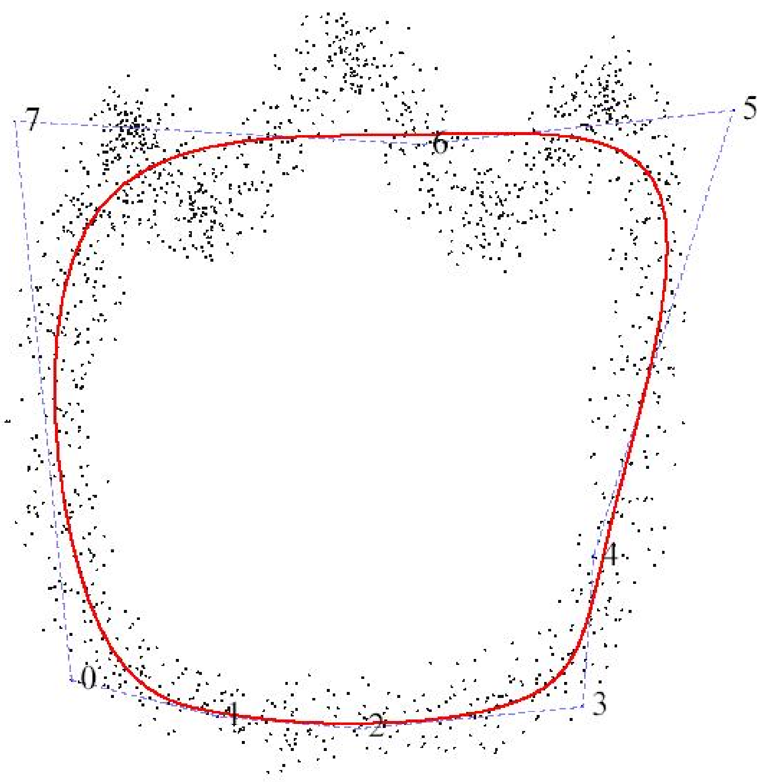


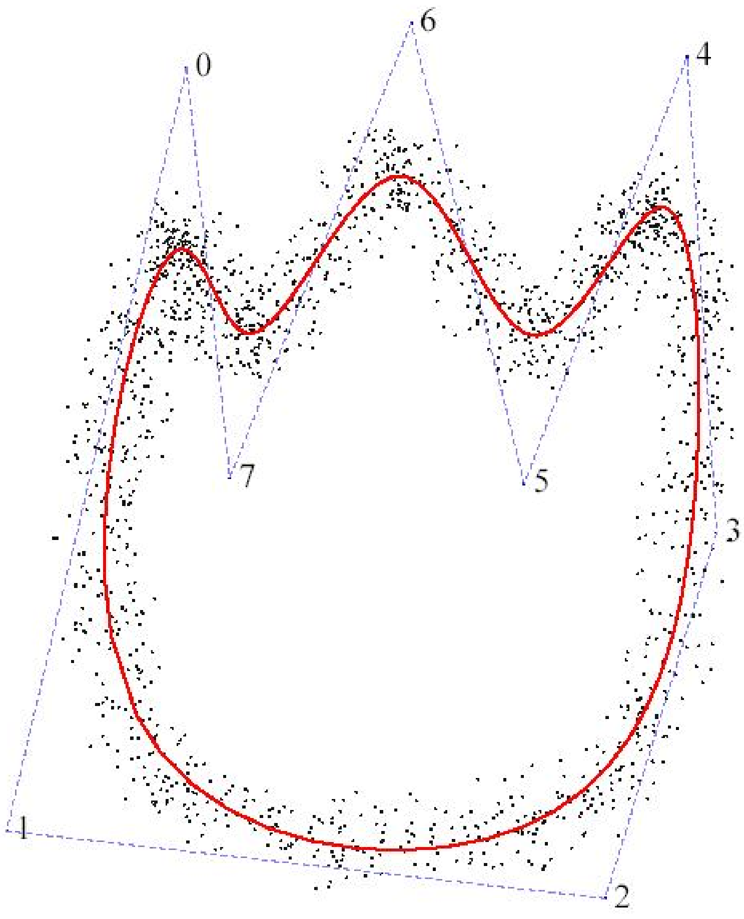
SDM after 31 iterations

Noisy data set; 50 iterations



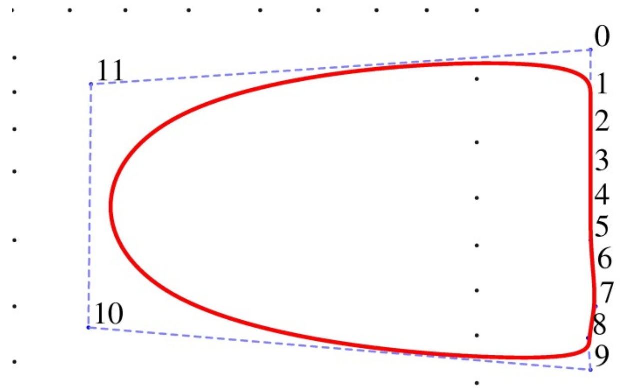
Comparison of PDM and SDM after 50 iteration steps



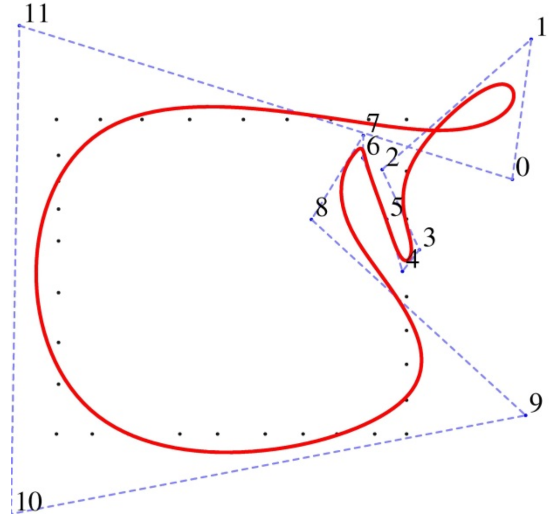


- Point Distance Min. (PDM)
- SDM

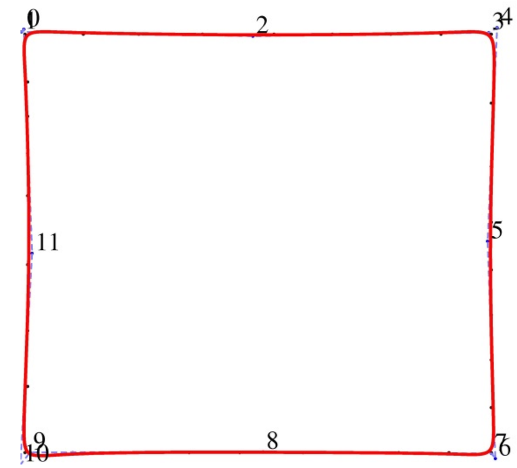
Comparison: PDM vs. SDM



Target shape and Initial position



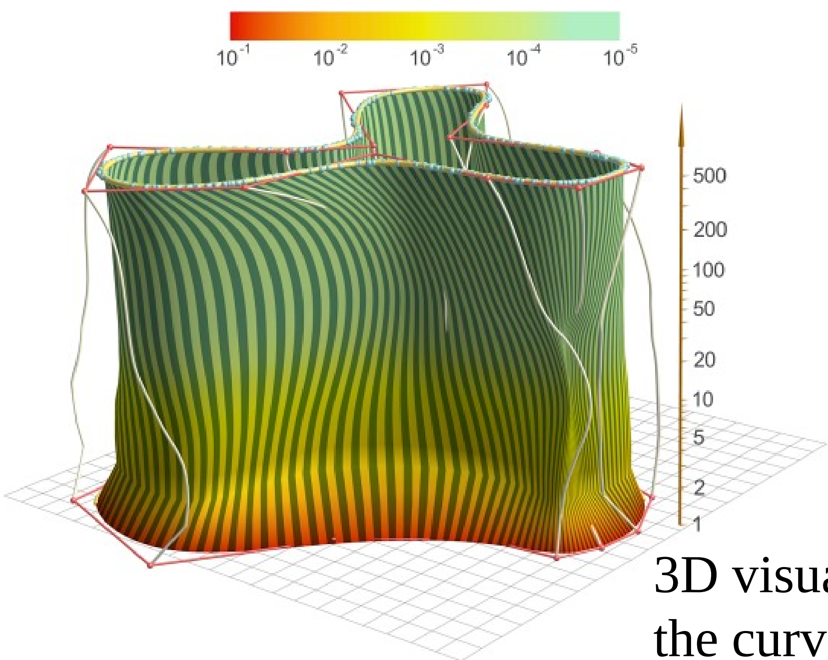
PDM after 20 iterations



SDM after 20 iterations

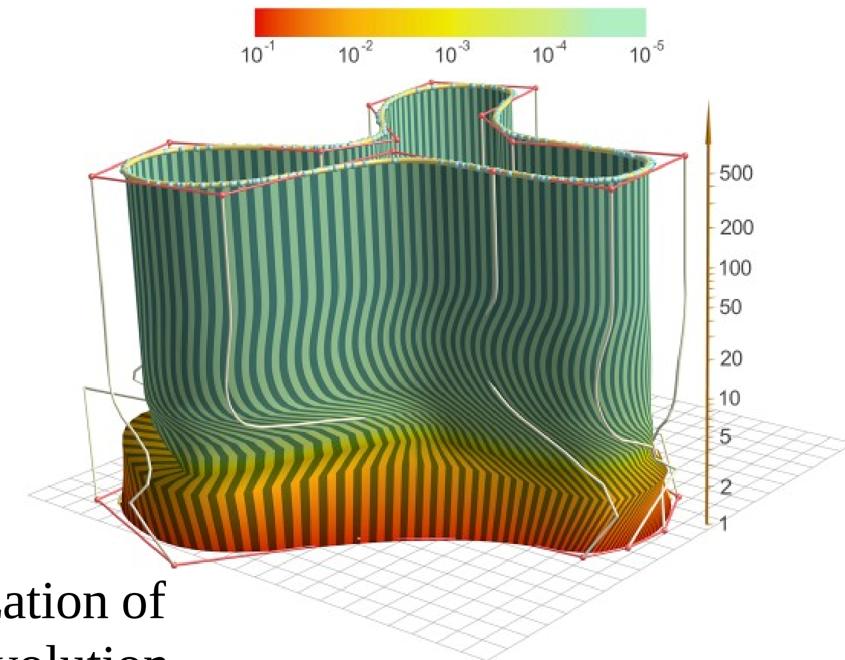
Convergence behavior

- Standard method (PDM): linear convergence
- SDM: quadratic convergence



PDM

3D visualization of the curve evolution



SDM