Reconstruction and Geometric Algorithms

Romain Vergne

Prashant Goswami

2013 - 2014

PLAN

- Introduction
- Parameterized / Ordered data
- Any supplied data
- Mini-project

PLAN

- Introduction
- Parameterized / Ordered data
- Any supplied data
- Mini-project

PLAN

- Introduction
- Parameterized / Ordered data
- Any supplied data
- Mini-project

PLAN

- Introduction
- Parameterized / Ordered data
- Any supplied data
- Mini-project

PLAN

Introduction

- General framework
- Types of data supplied
- Types of models
- Interpolation vs. Approximation

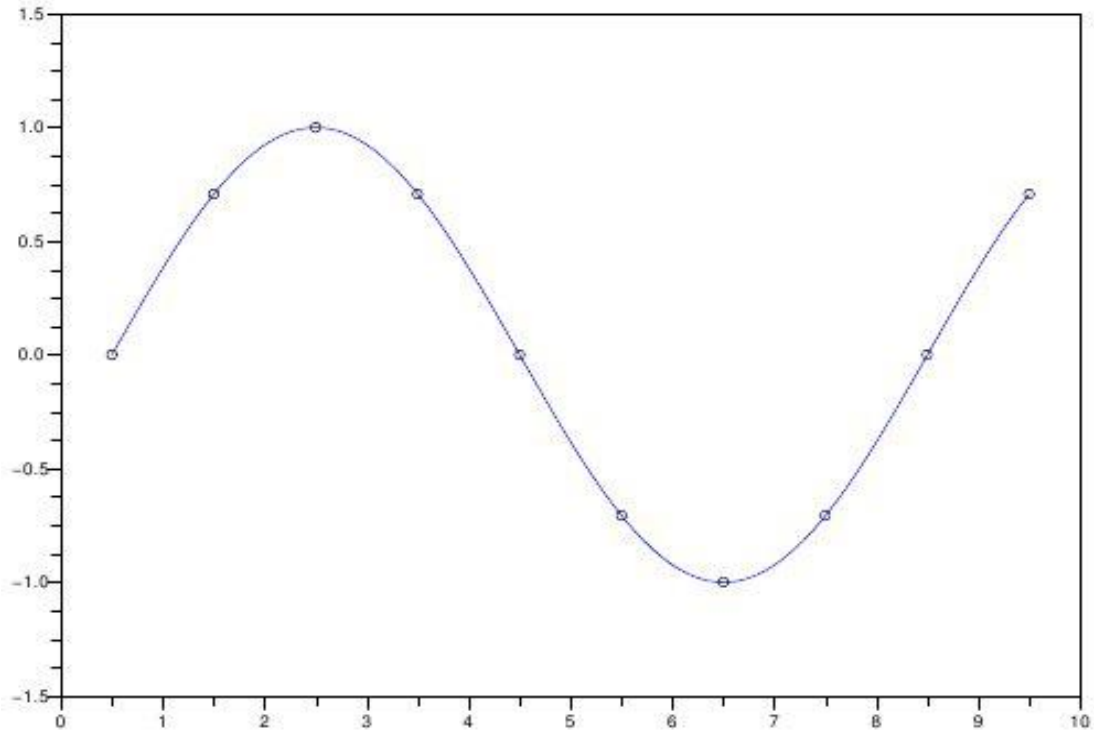
PLAN

Introduction

- General framework
- Types of data supplied
- Types of models
- Interpolation vs. Approximation

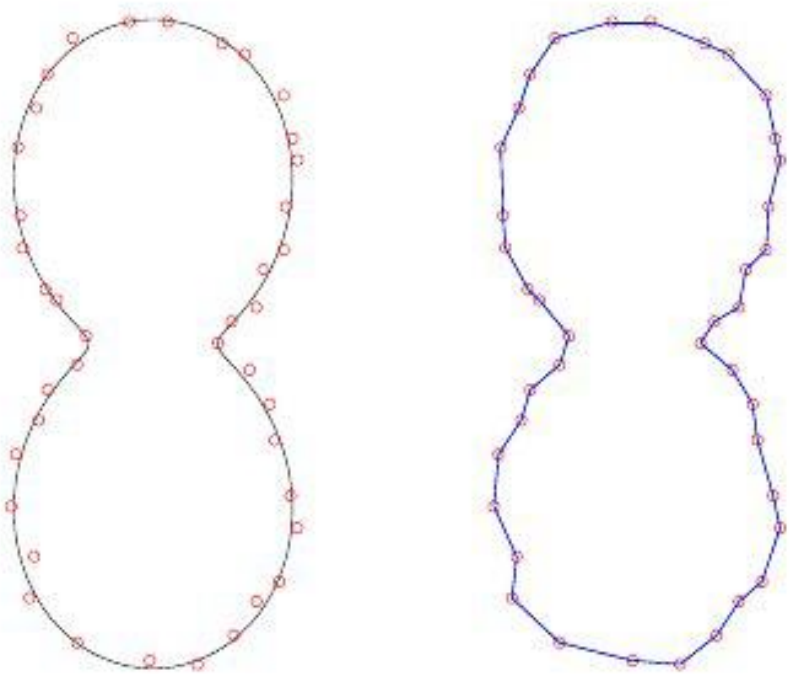
Reconstruction of Functions from Samples

Reconstruction of curves from points



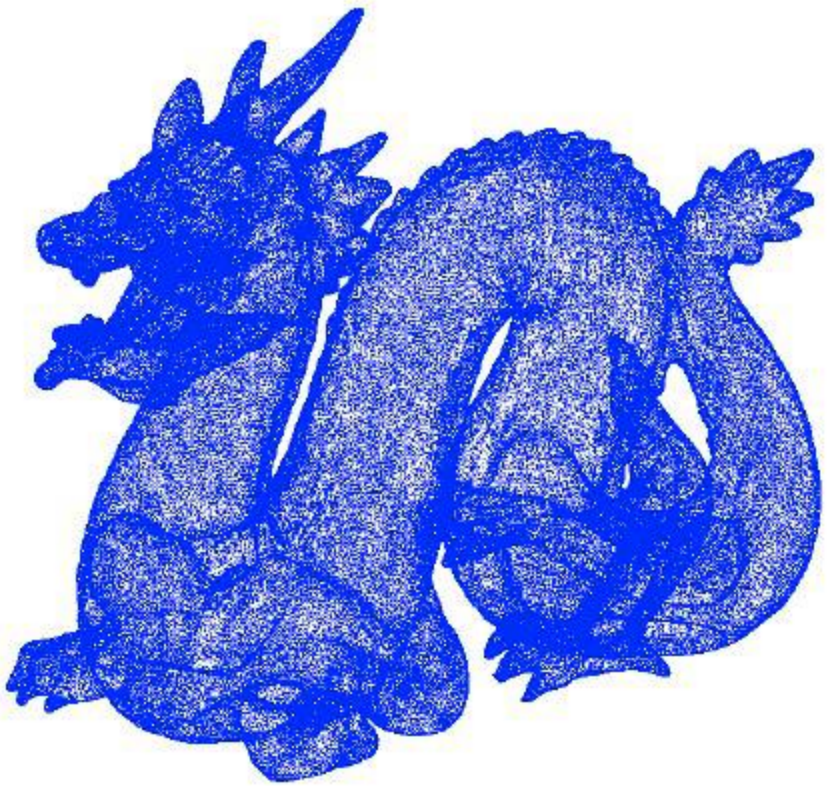
Reconstruction of Functions from Samples

Reconstruction of curves from points



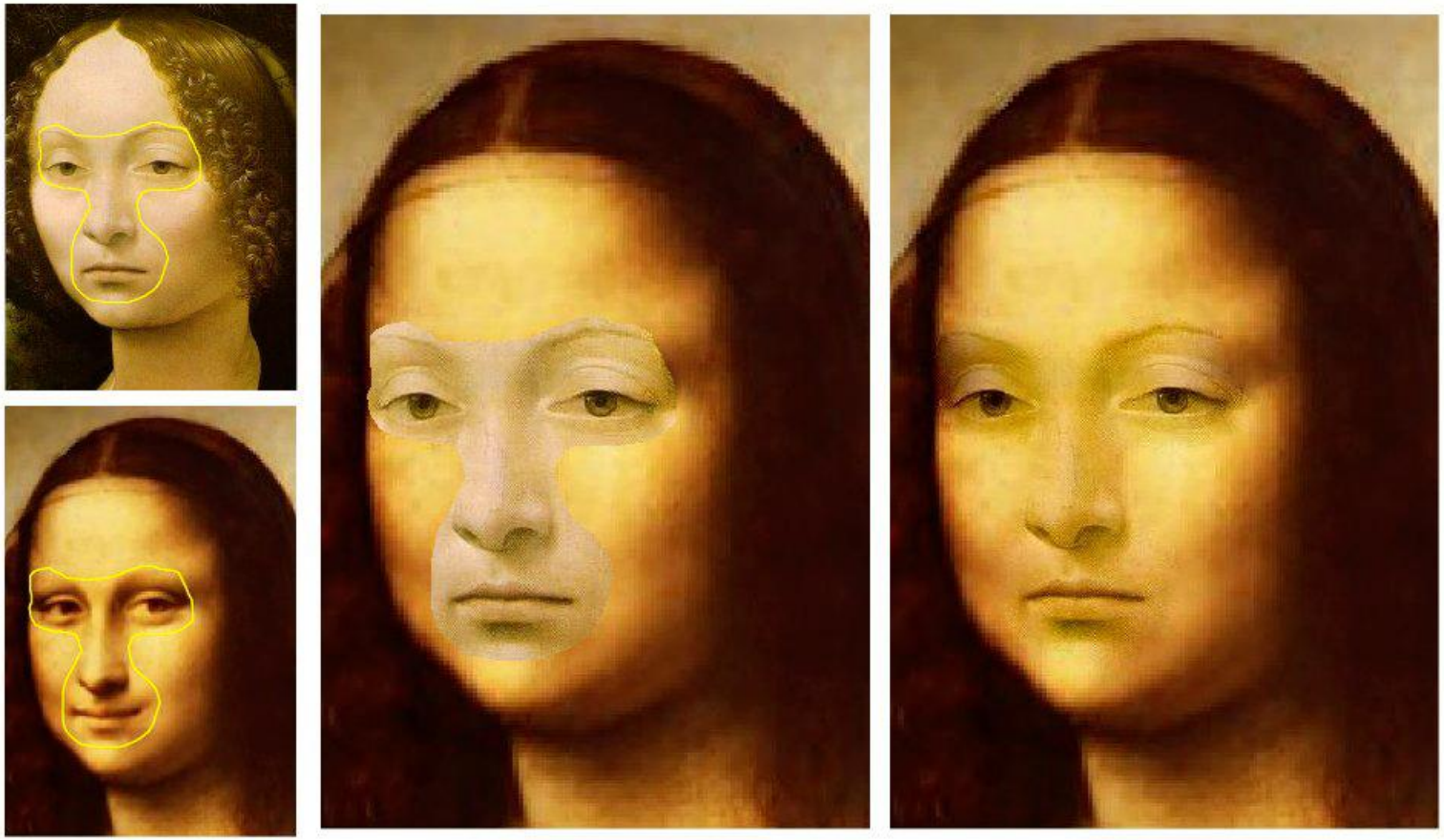
Reconstruction of Functions from Samples

Reconstruction of surfaces from point clouds



Reconstruction of Functions from Samples

Collage of images

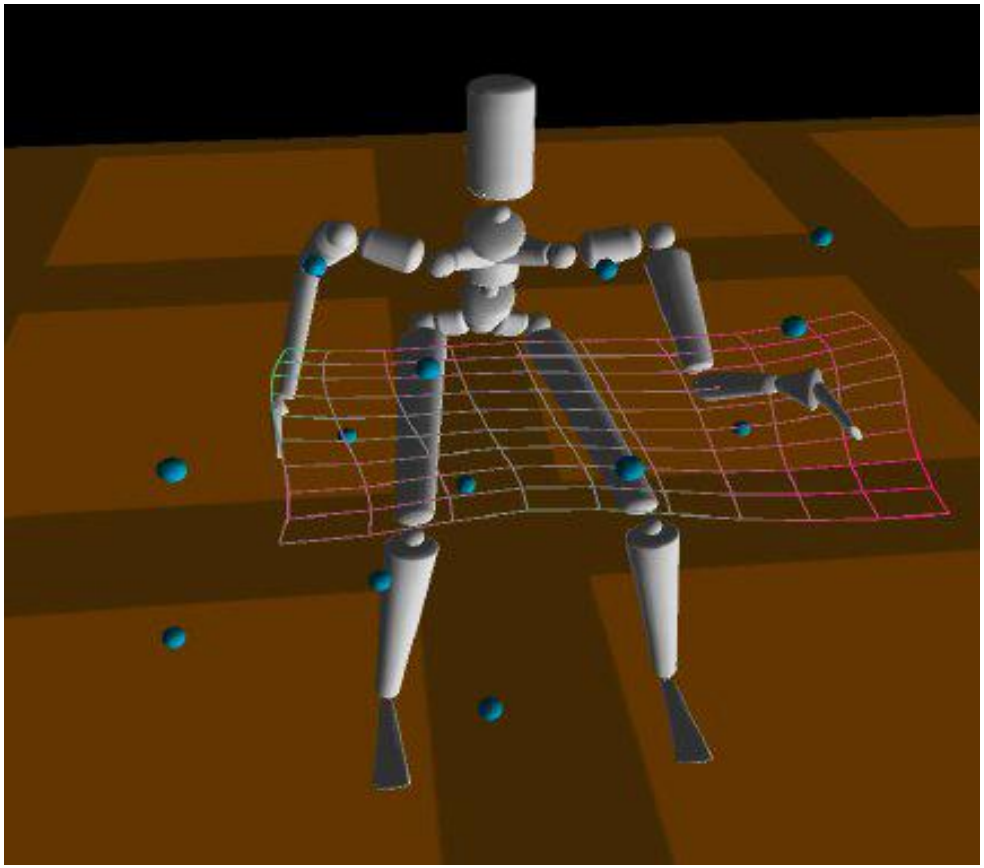


Reconstruction of Functions from Samples

Deformation of objects

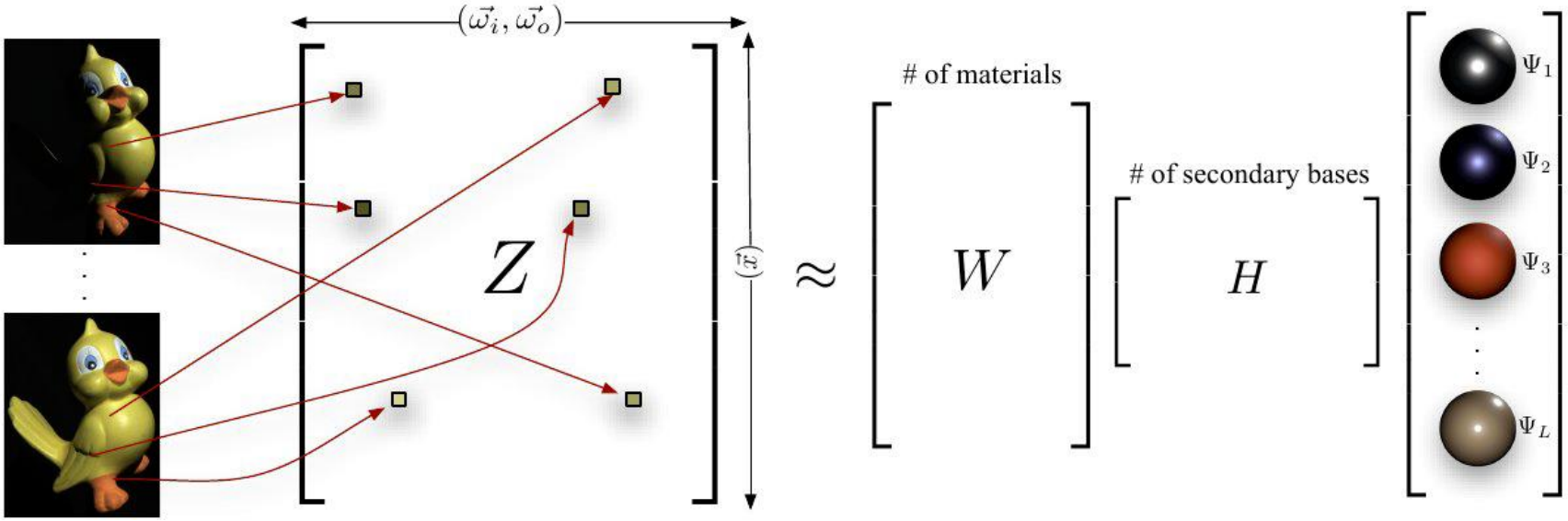
Reconstruction of Functions from Samples

Inverse Kinematics



Reconstruction of Functions from Samples

Estimation of reflection from images

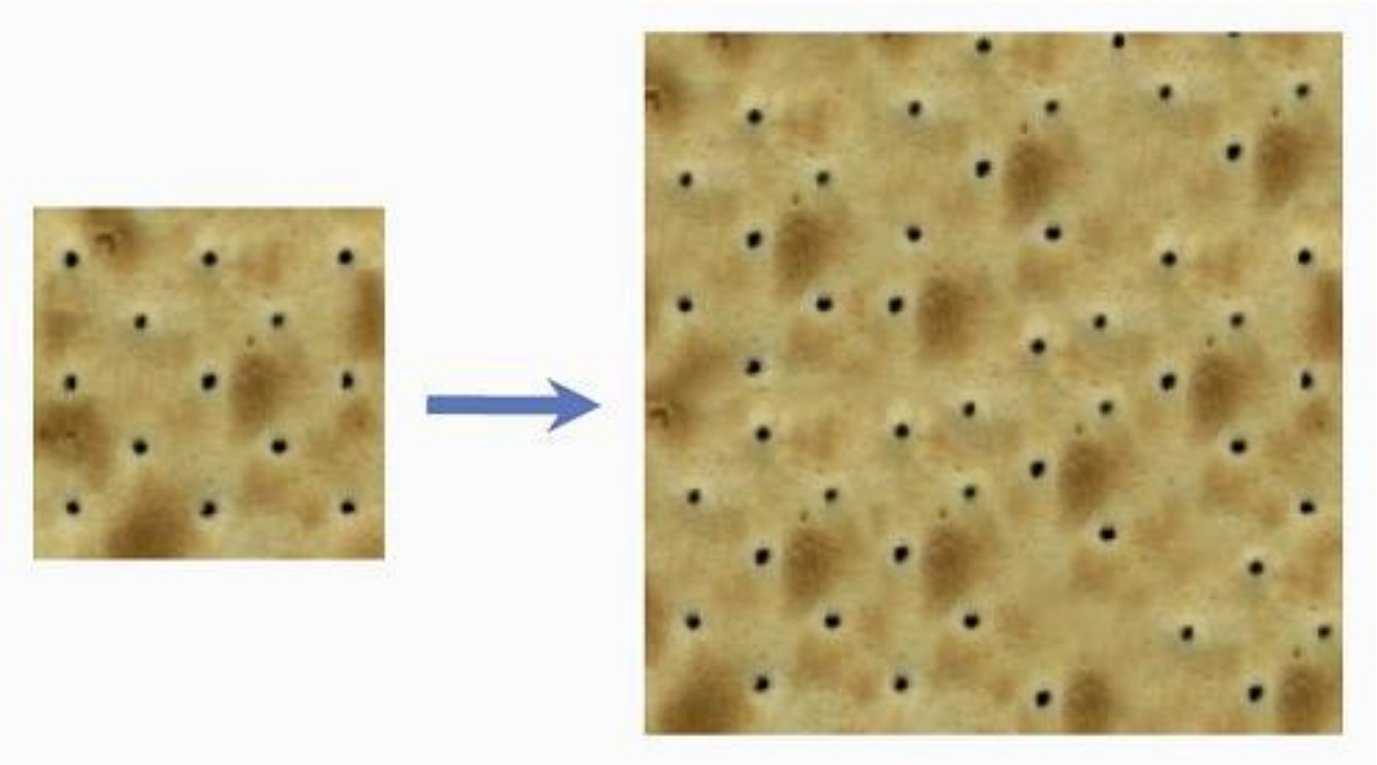


Reconstruction of Functions from Samples

Fluids

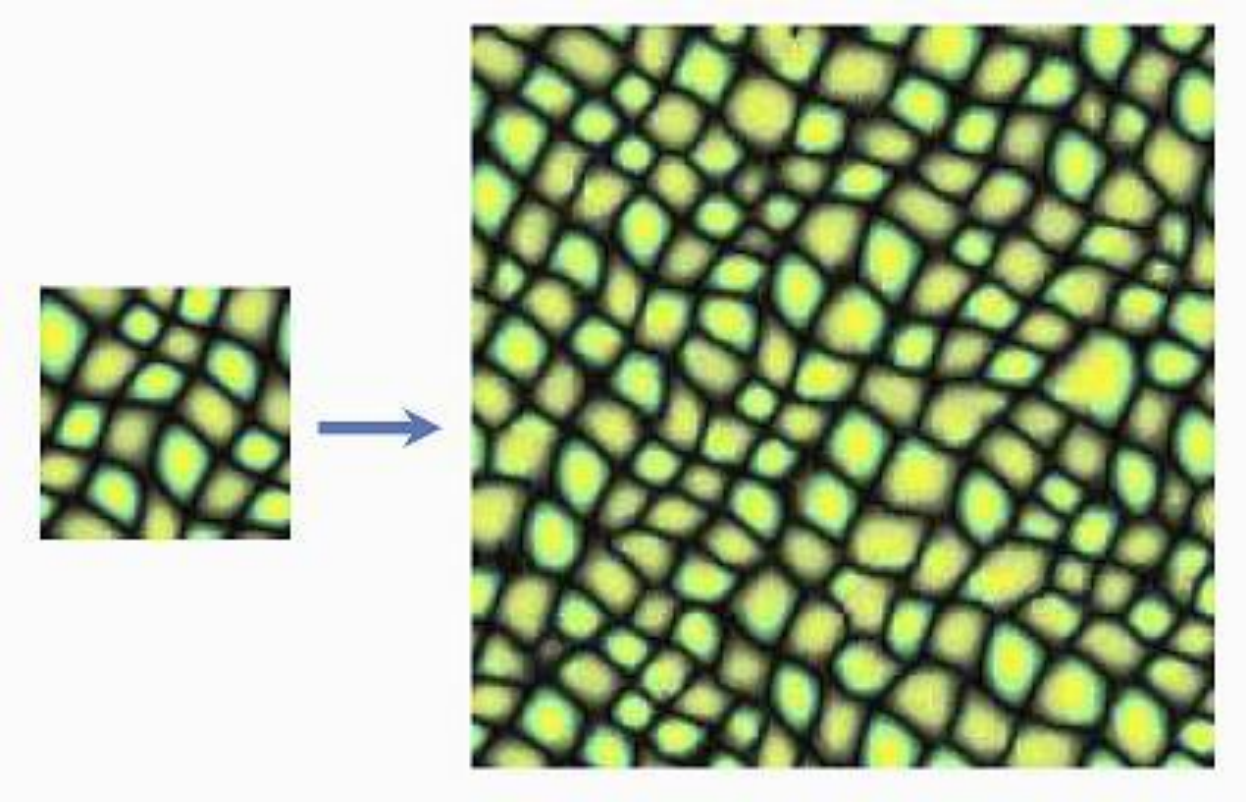
Reconstruction of Functions from Samples

Texture synthesis



Reconstruction of Functions from Samples

Texture synthesis



Reconstruction of Functions from Samples

But also:

- Neural networks
- Machine learning
- Data analysis / probability
- ….

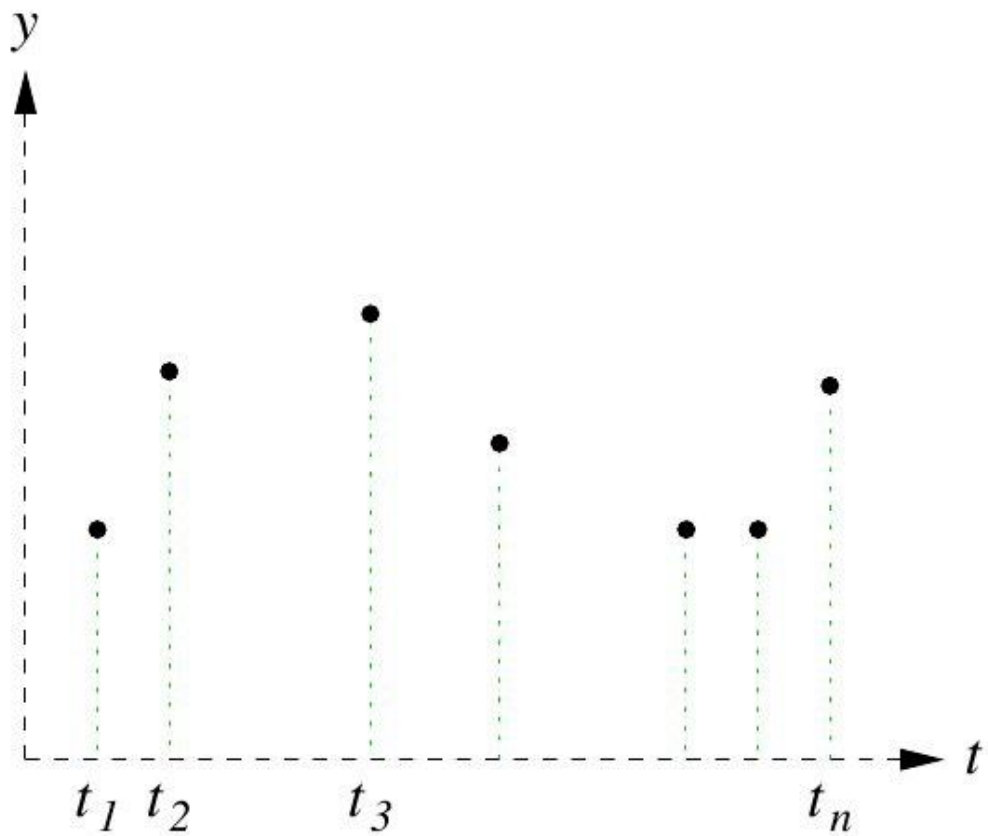
PLAN

Introduction

- General framework
- Types of data supplied
- Types of models
- Interpolation vs. Approximation

Point Parameters

Reconstruction of Curves

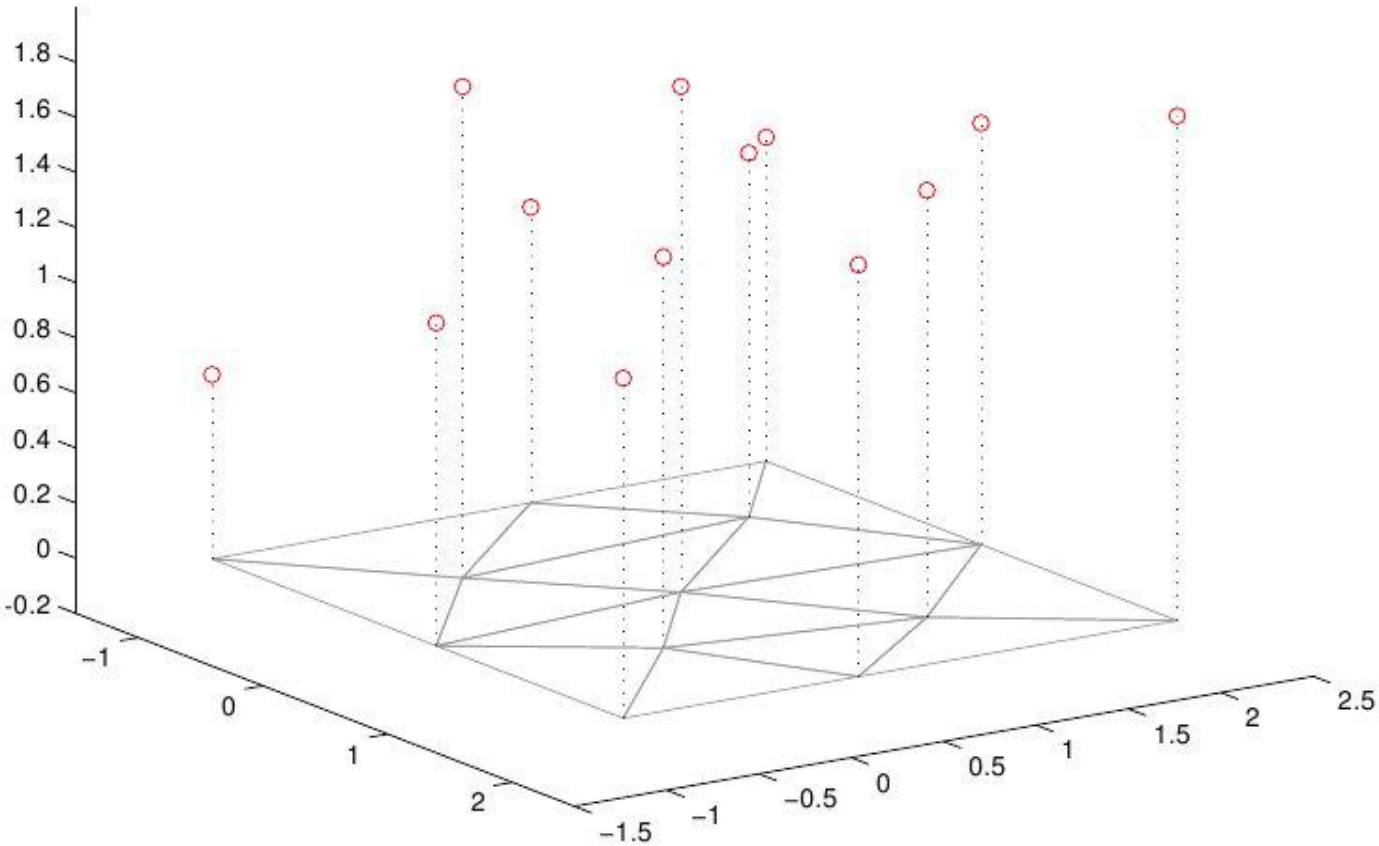


Data D = { point $P_i = (x_i , y_i)$; parameter t_i }

t_i : Parameter associated with point P_i *(here $x_i = t_i$)*

Point Parameters

Reconstruction of Surfaces

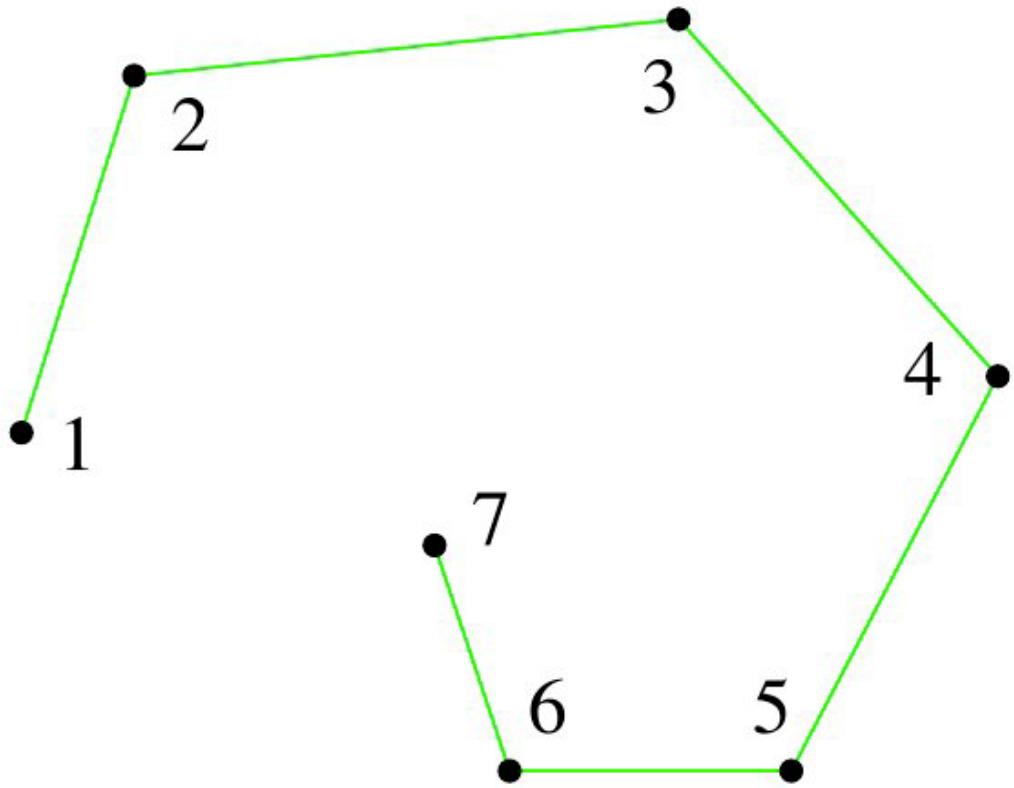


Data D = { point $P_i = (x_i, y_i, z_i)$; parameter (u_i, v_i) }

(u_i, v_i) : Parameter associated with point P_i *(here $u_i = x_i$, $v_i = y_i$)*

Points Ordering

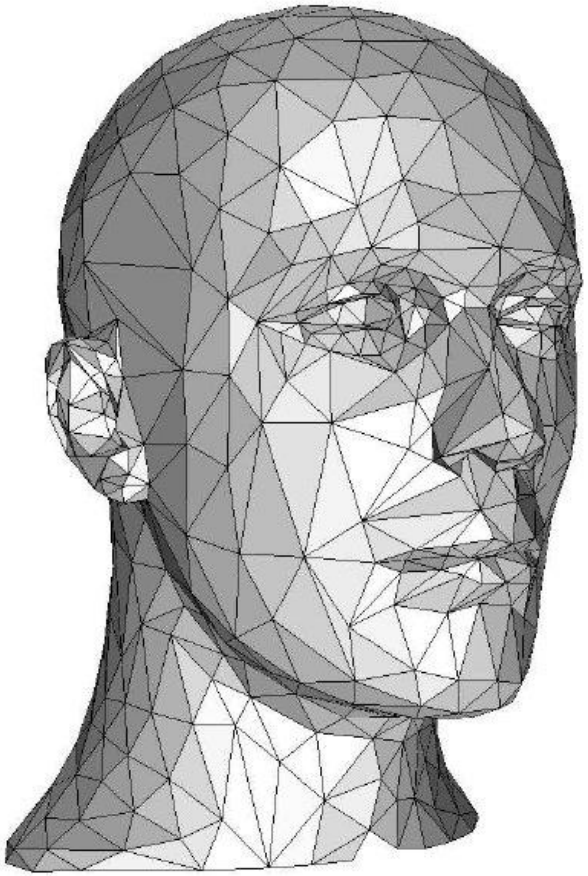
In 2D



Data D = point P_i + relation between neighboring points

Points Ordering

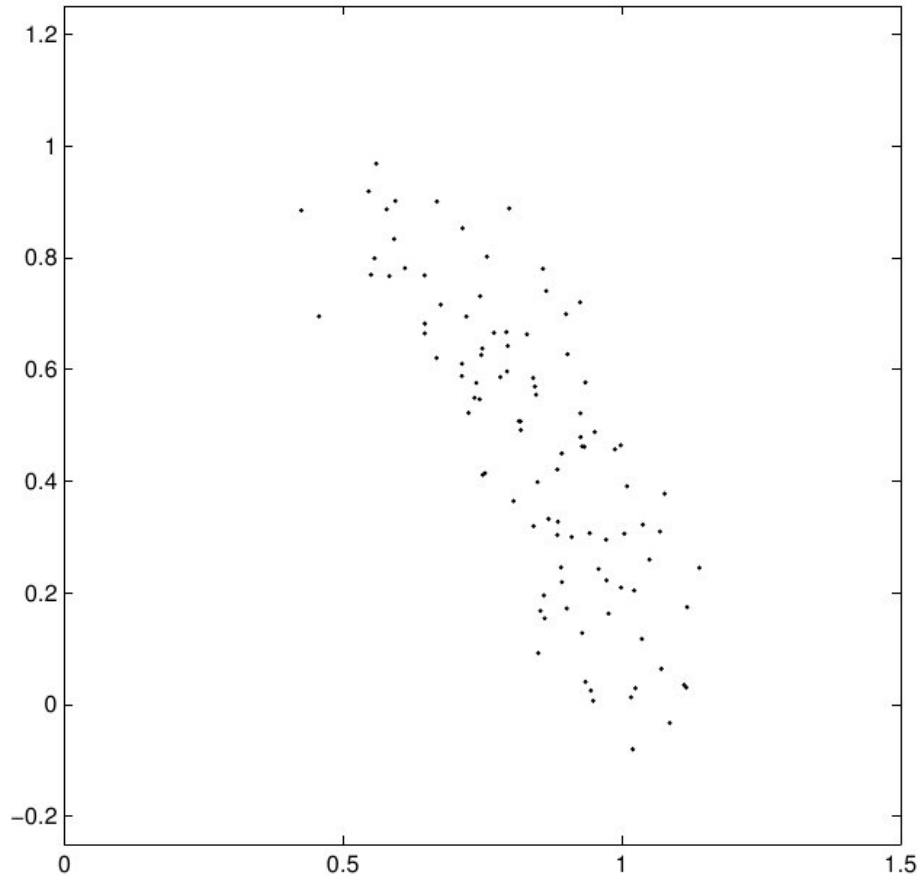
In 3D



Data D = point P_i + relation between neighboring points

Some points (Scattered data)

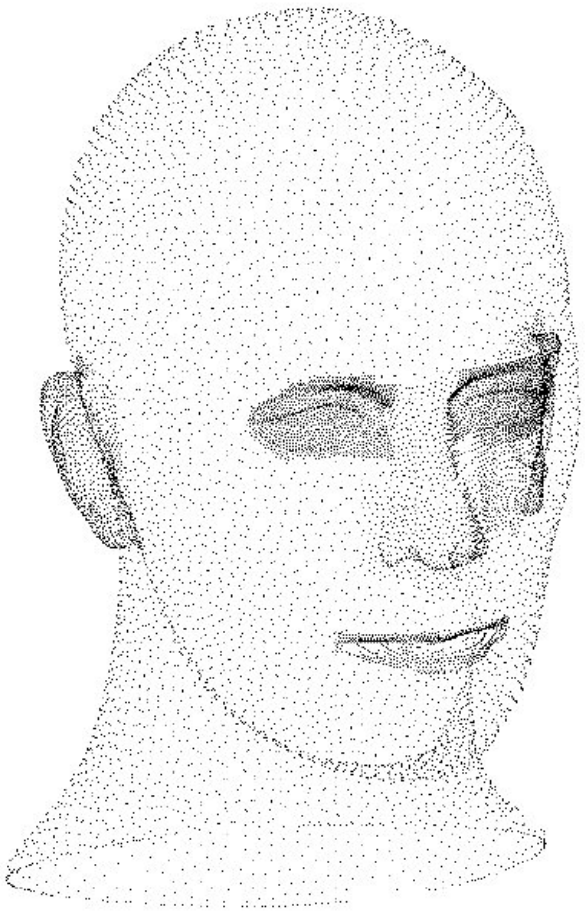
In 2D



Data D = { point $P_i = (x_i , y_i)$ }

Some points (Scattered data)

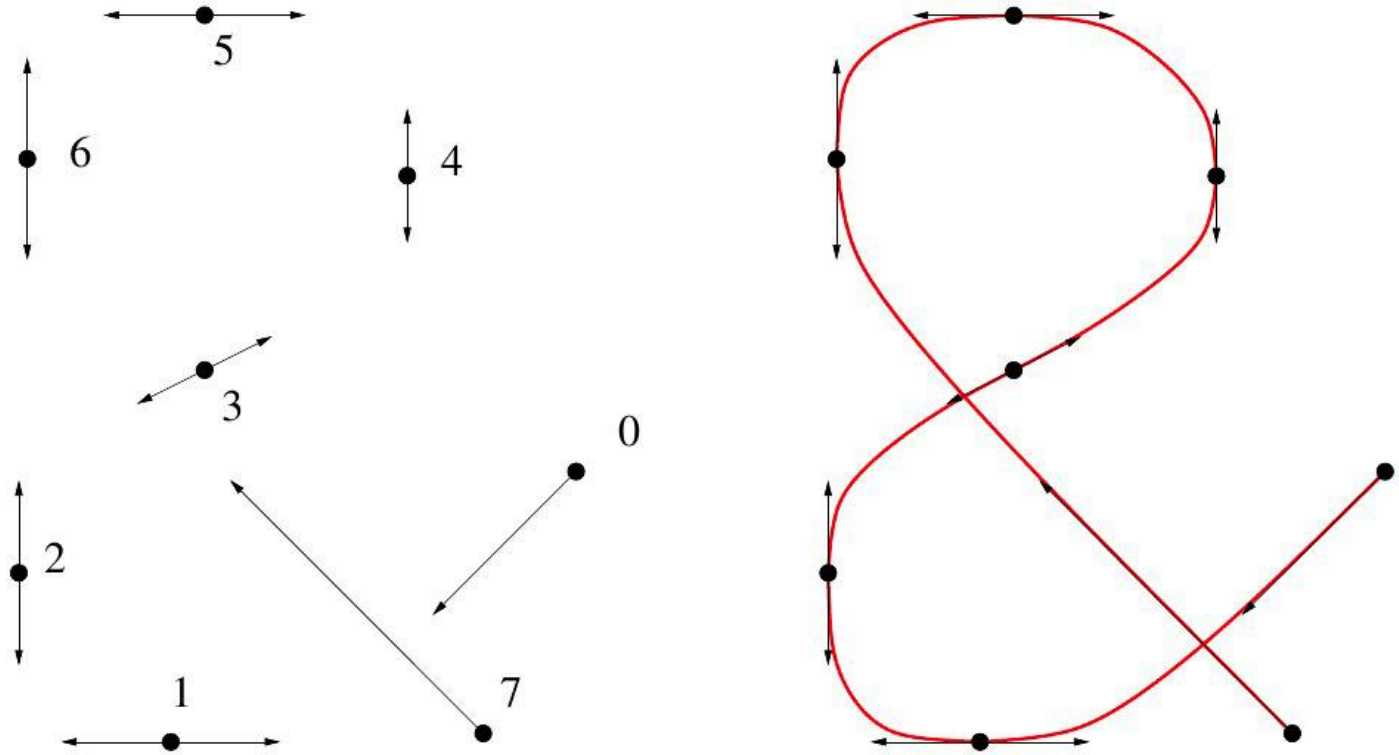
In 3D



Data D = { point $P_i = (x_i, y_i, z_i)$ }

Additional Geometrical Data

Ex: Tangents

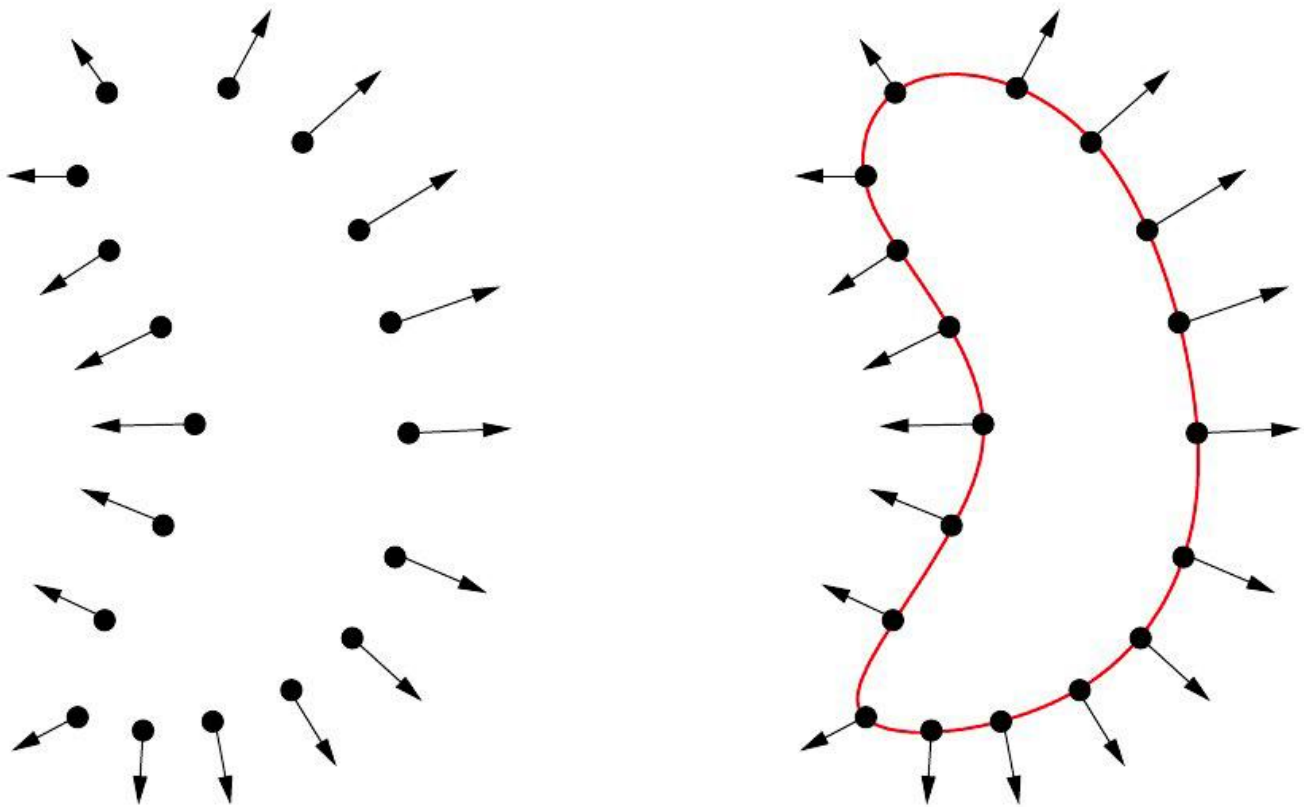


Data D = { point P_i + tangent T_i }

T_i is the tangent to the curve at P_i

Additional Geometrical Data

Ex: Normals

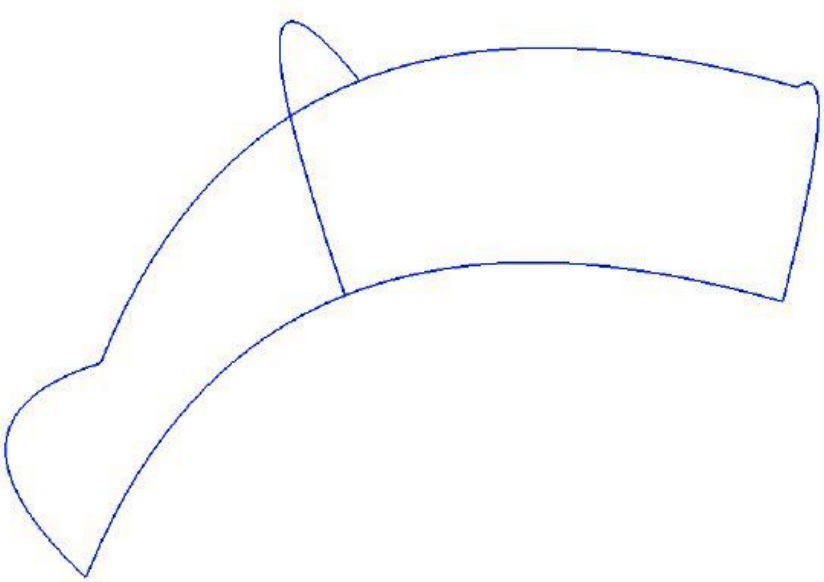


Data D = { point P_i + tangent N_i }

N_i is the normal to the curve at P_i

Additional Geometrical Data

Ex: Curve generated Surface



Data D = { curve C_i }

C_i being the boundary of the surface

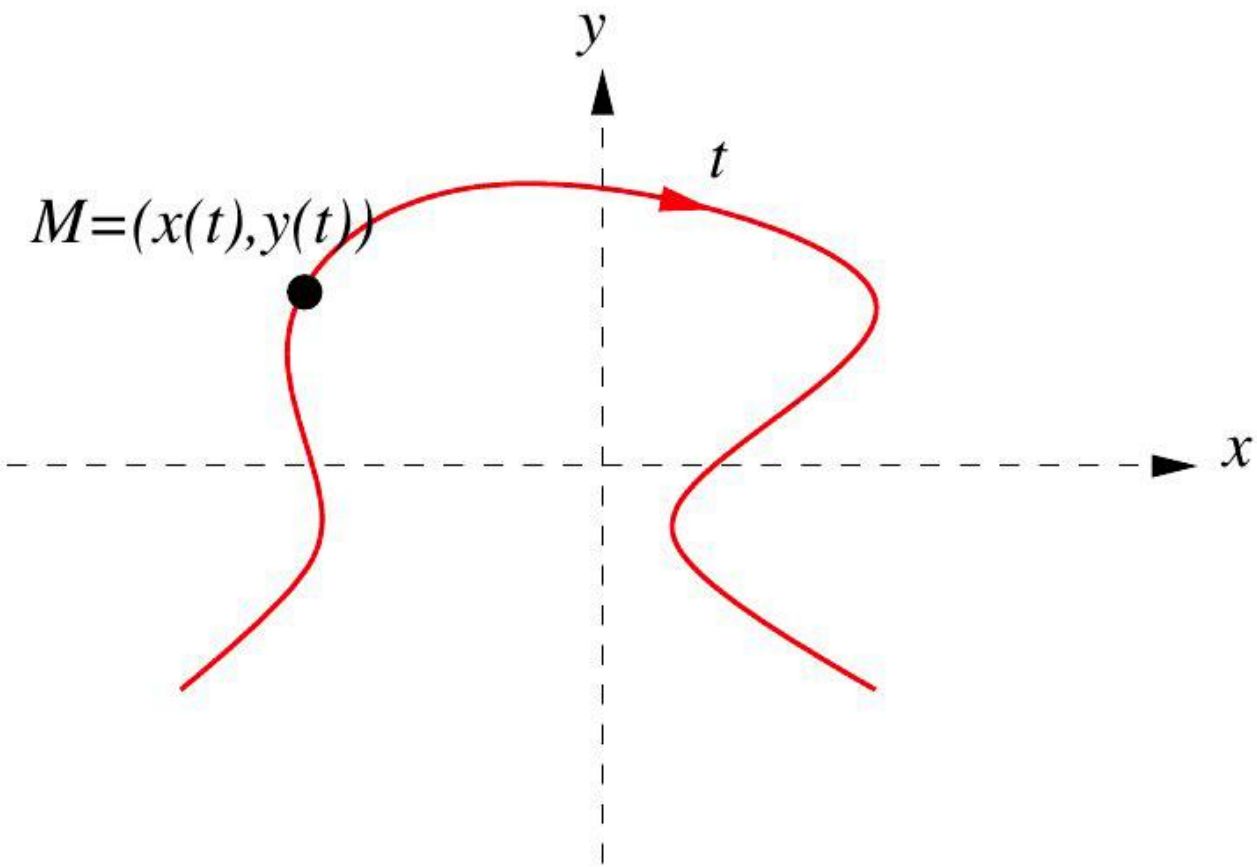
PLAN

Introduction

- General framework
- Types of data supplied
- Types of models
- Interpolation vs. Approximation

Curve Parameters

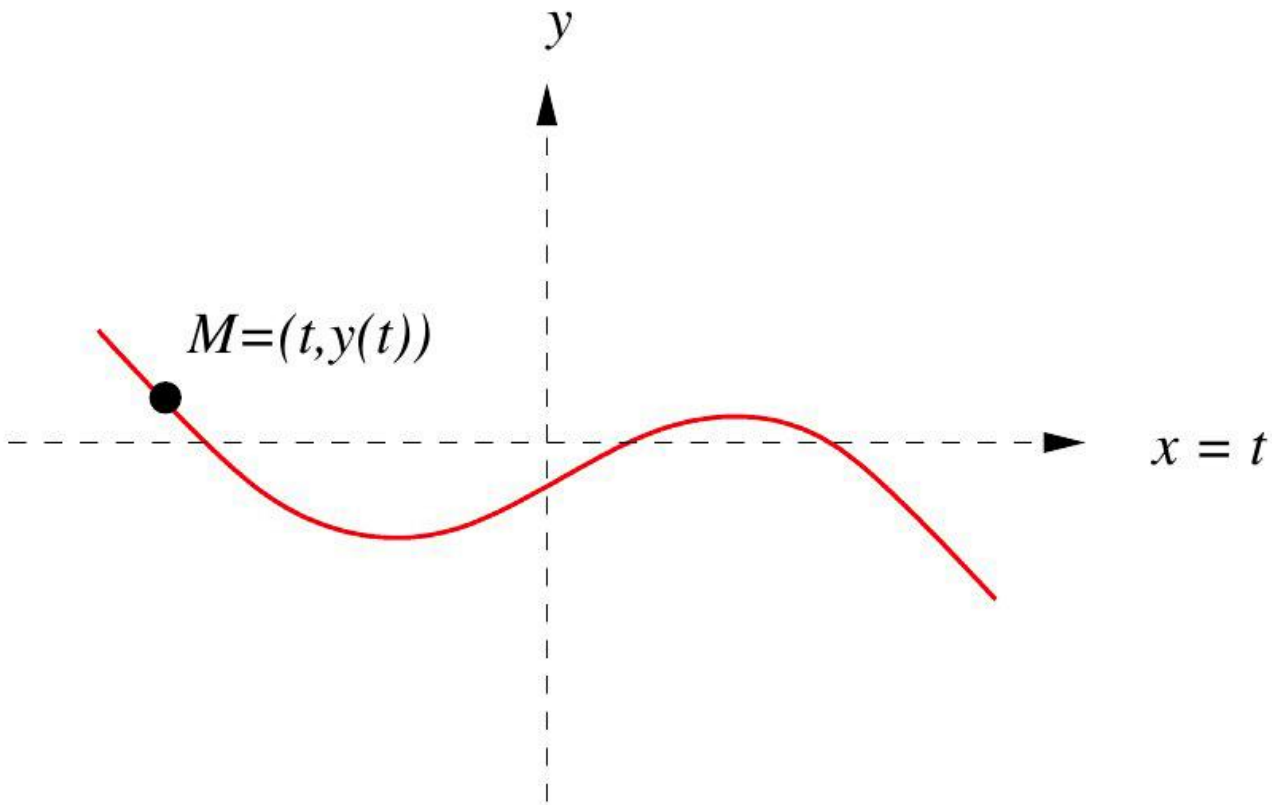
General case



Curve C = { (x(t) , y(t)), t ϵ [a,b] }

Curve Parameters

General case

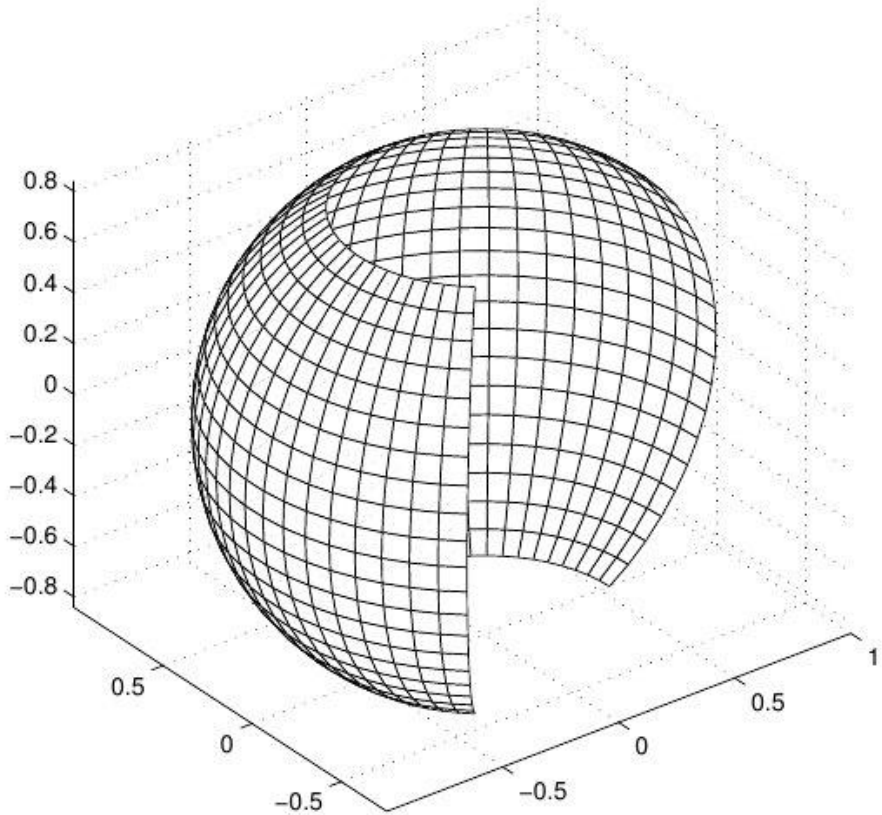


Curve C = { (x(t) , y(t)), t ϵ [a,b] }

Here *x(t) = t*

Surface Parameters

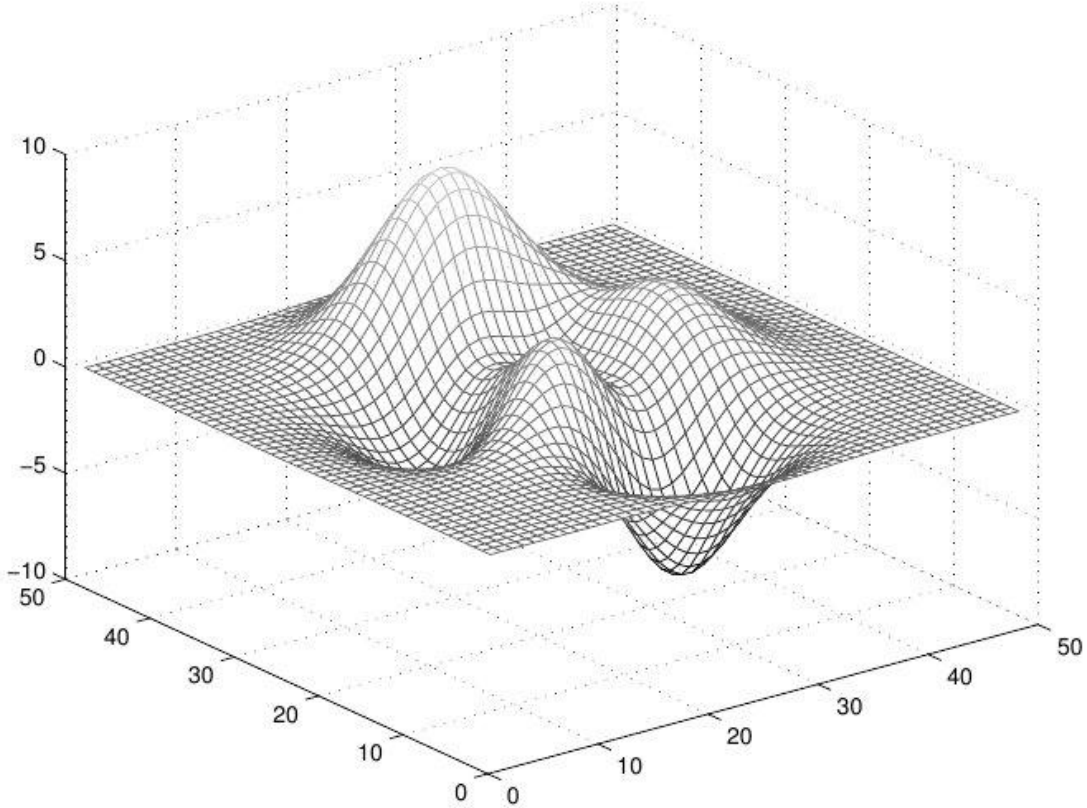
General case



$$\text{Surface } S = \{ (x(u, v), y(u, v), z(u, v)), \ (u, v) \in D, D \subset \mathbb{R} \}$$

Surface Parameters

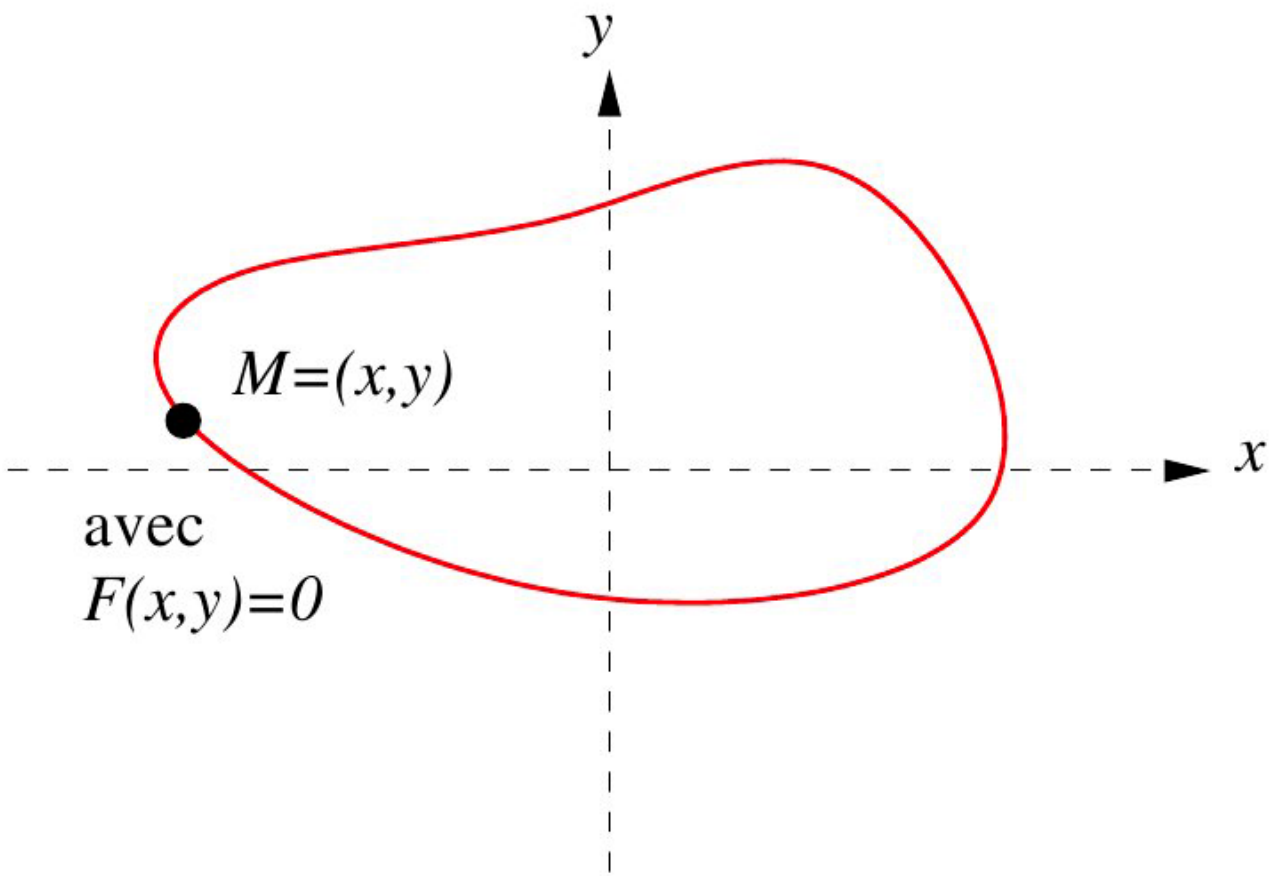
Special case



Surface S = { $(x(u, v), y(u, v), z(u, v))$, $(u, v) \in D, D \subset \mathbb{R}$ }

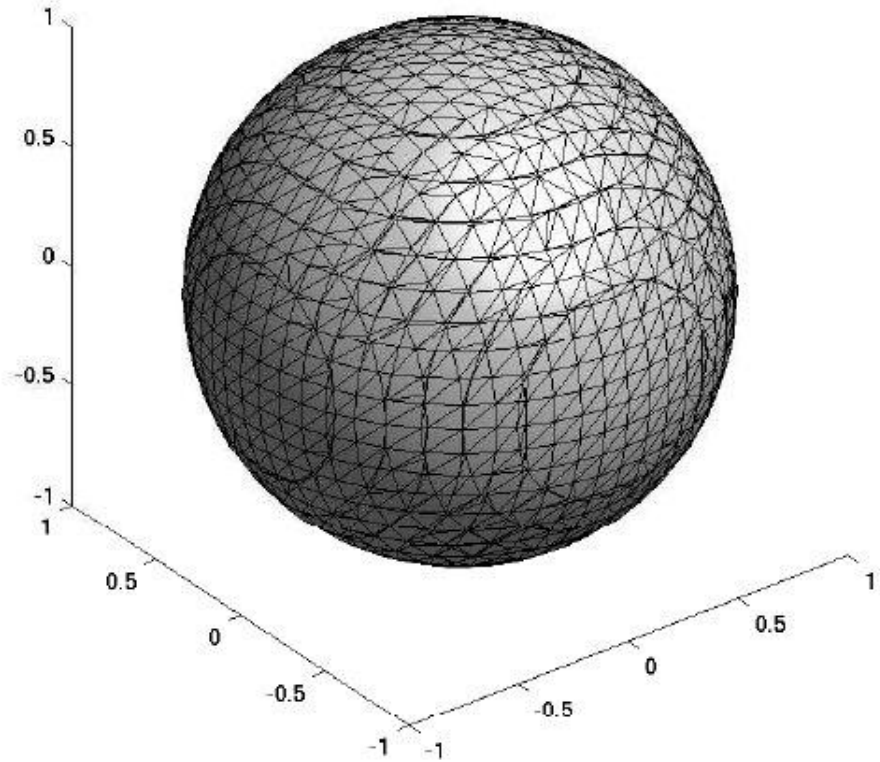
Here $x(u, v) = u$ and $y(u, v) = v$

Implicit Curve



Curve C = { (x , y), F(x, y) = 0 }

Implicit Surface



Surface S = { (x , y, z), F(x, y, z) = 0 }

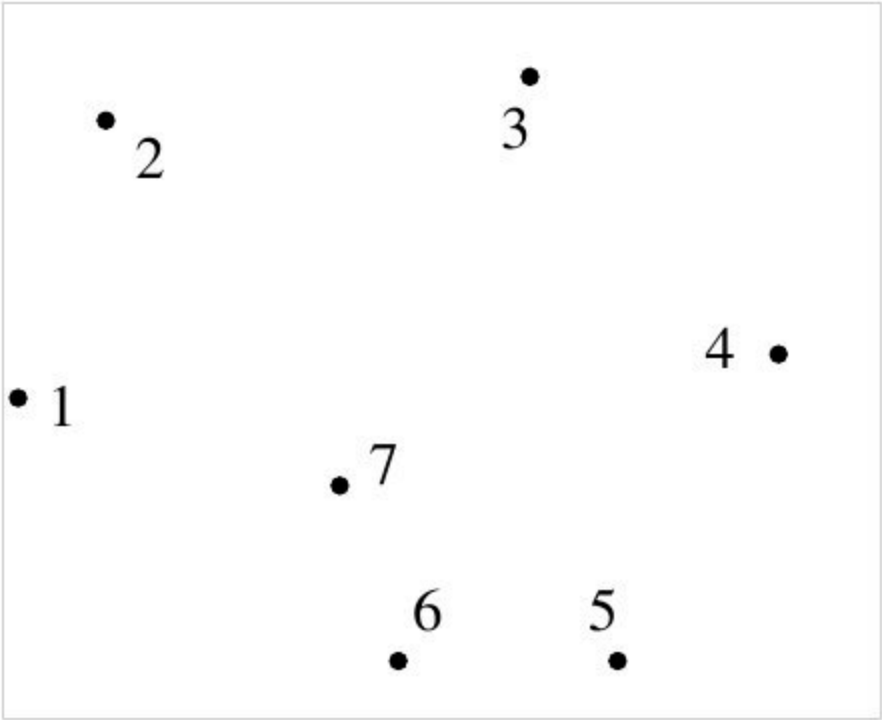
PLAN

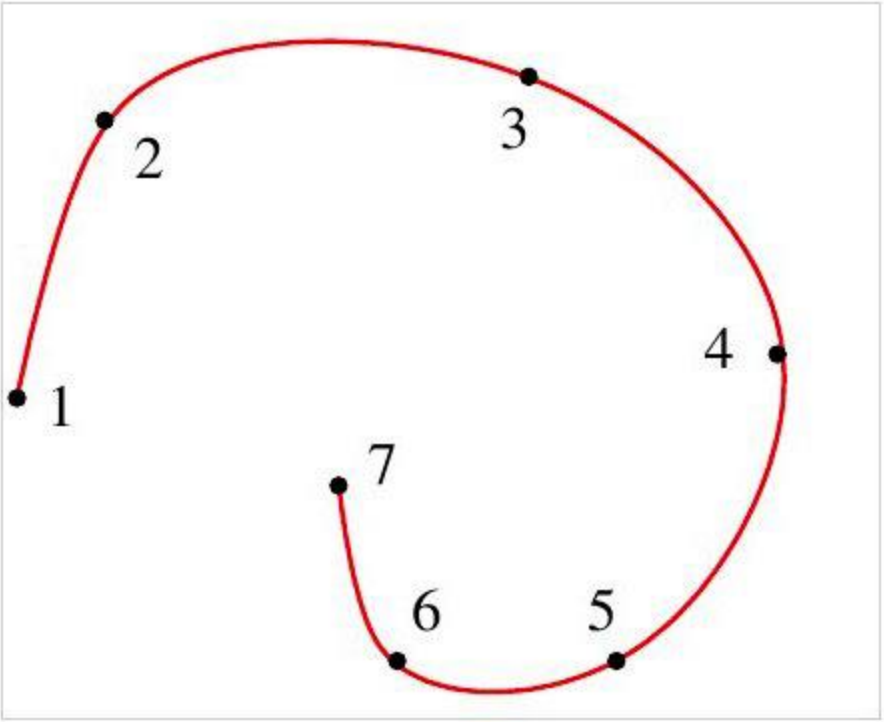
Introduction

- General framework
- Types of data supplied
- Types of models
- Interpolation vs. Approximation

Interpolation

Example 1



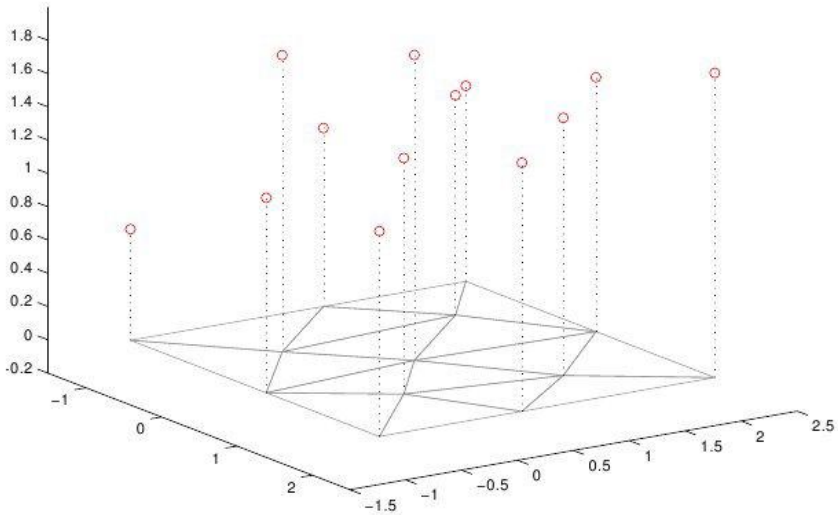


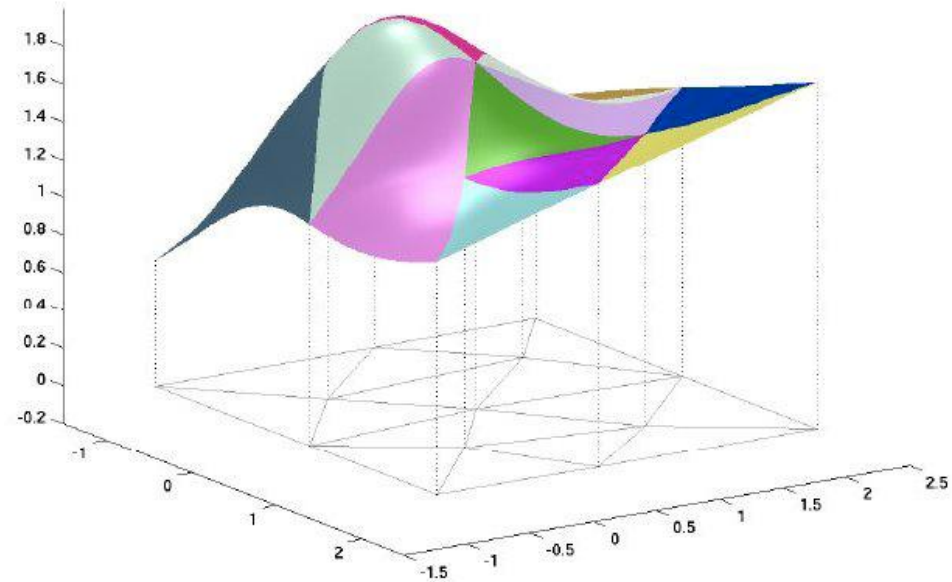
Data: Ordered points

Curve Interpolation

Interpolation

Example 2

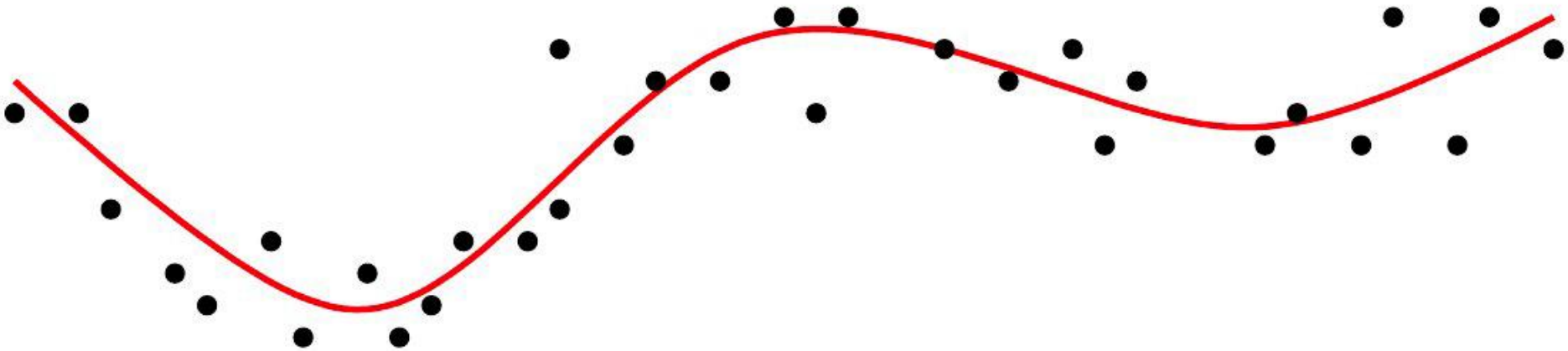




Triangulation plan
$S_i = (x_i , y_i)$ and points
$(S_i , z_i) = (x_i , y_i , z_i)$

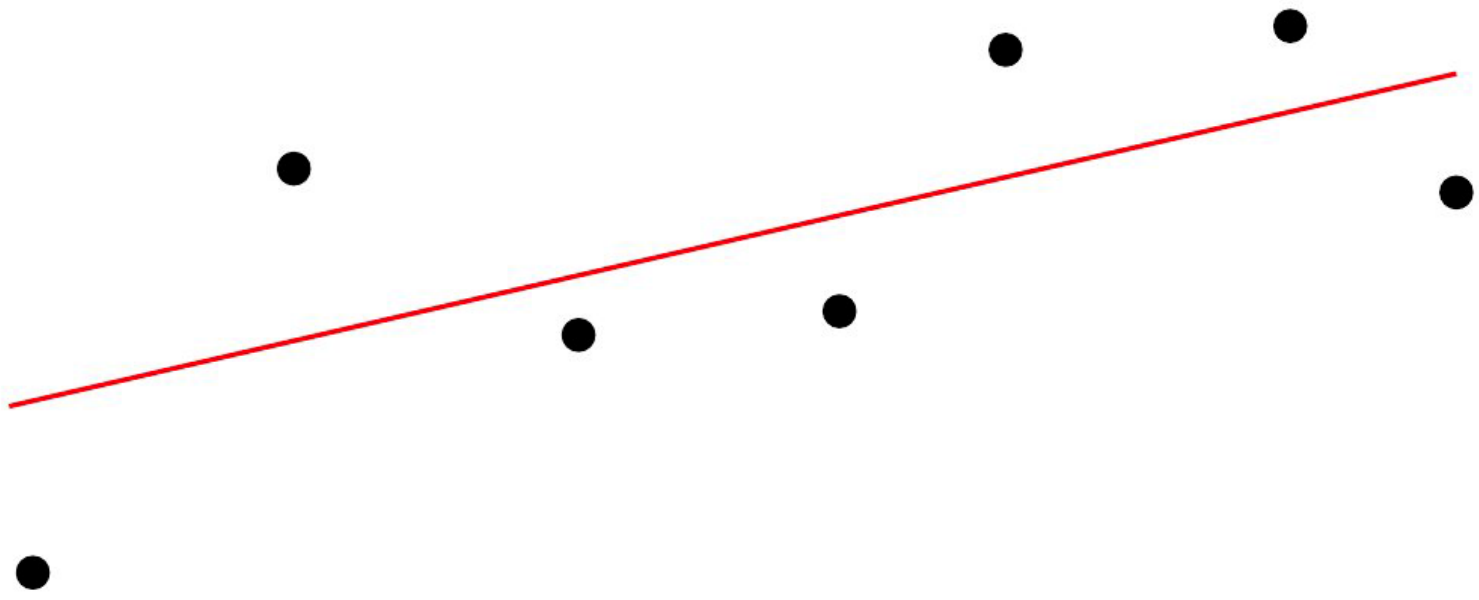
Smooth surface defined by piecewise interpolation inside triangles

Approximation



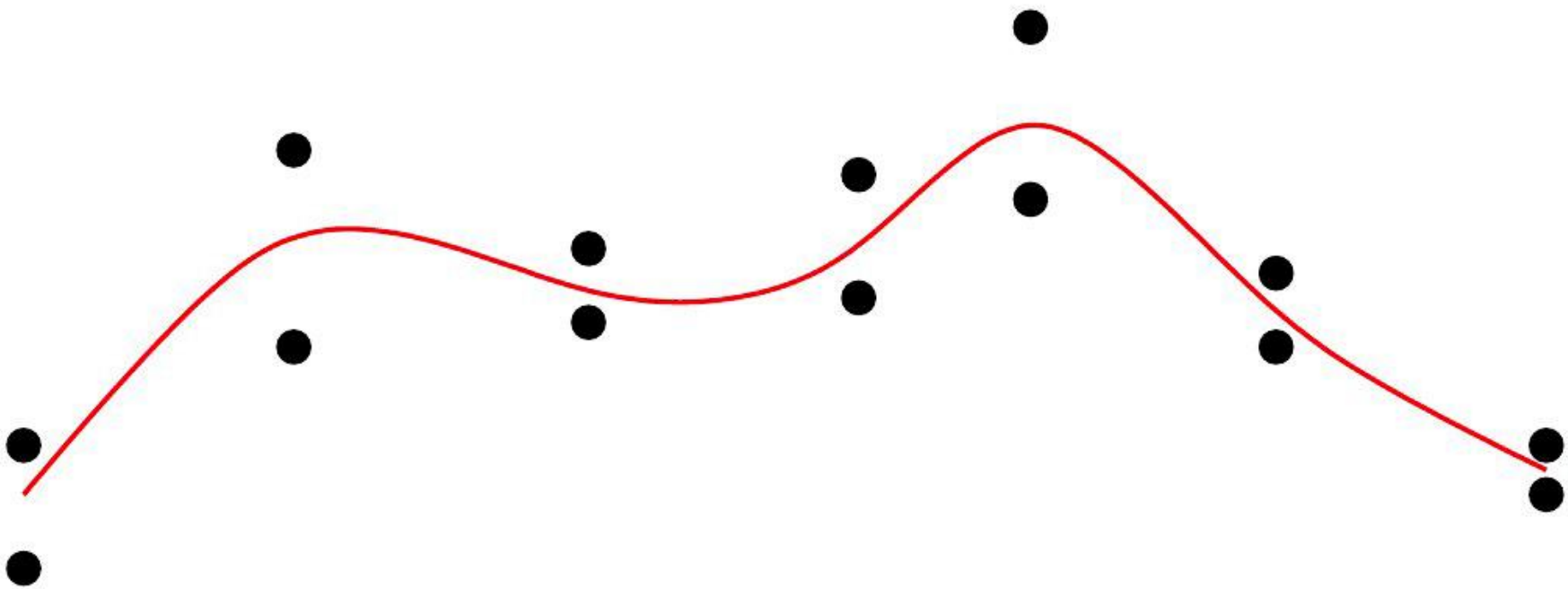
Smoothing noisy data

Approximation



Simple model does not allow interpolation

Approximation



Parameterization does not allow interpolation

PLAN

- Introduction
- Parameterized / Ordered data
- Any supplied data
- Mini-project

PLAN

Parameterized / Ordered data

- Curve interpolation
- Surface interpolation
- Curve approximation
- Surface approximation

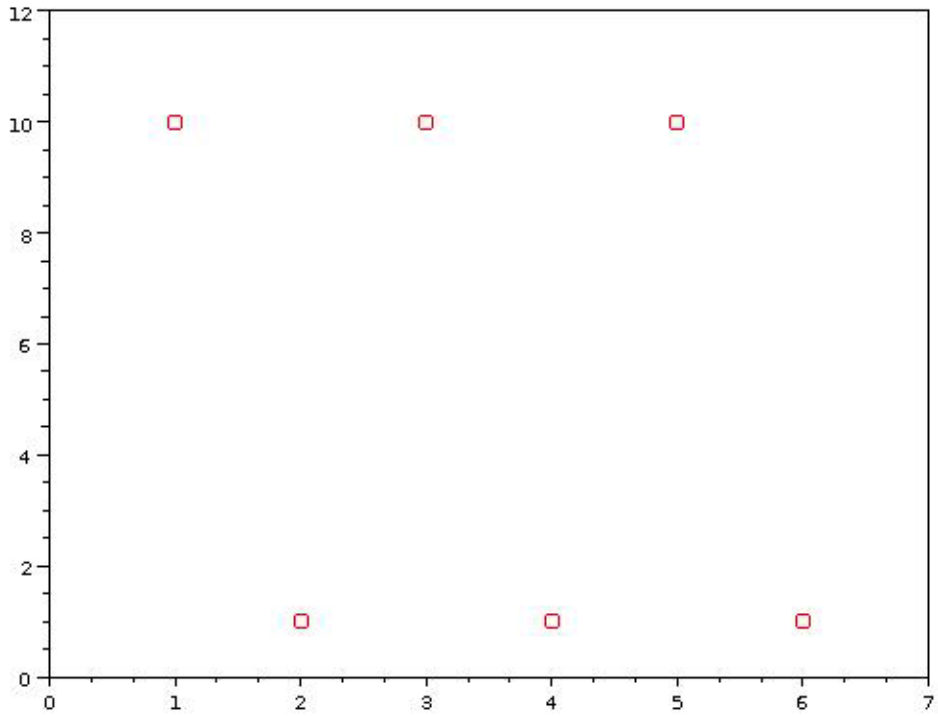
PLAN

Parameterized / Ordered data

- Curve interpolation
- Surface interpolation
- Curve approximation
- Surface approximation

General Framework

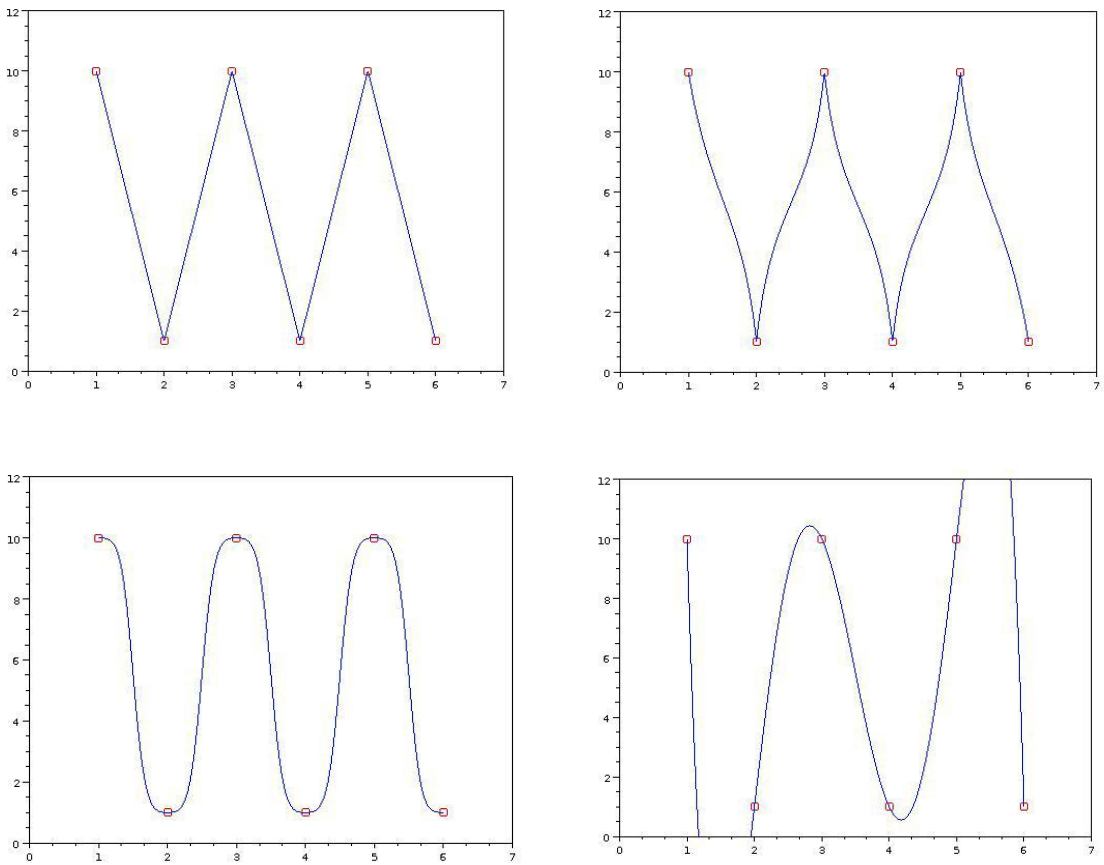
Functions f(x) = y



What is the best curve passing through these points?

General Framework

Functions f(x) = y



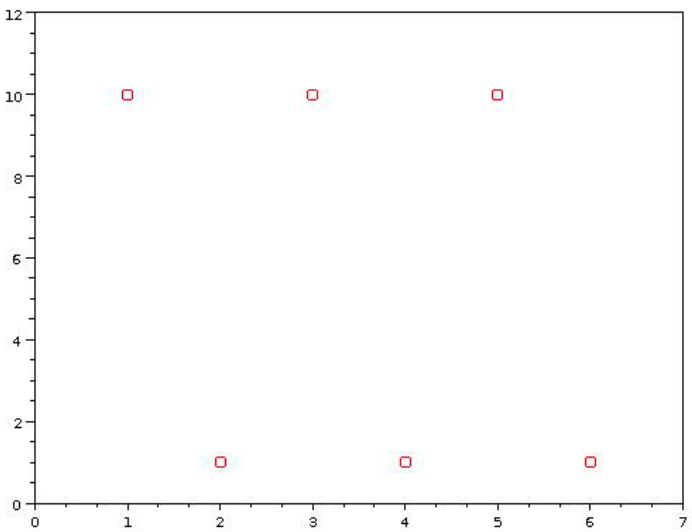
What is the best curve passing through these points?

Linear Interpolation

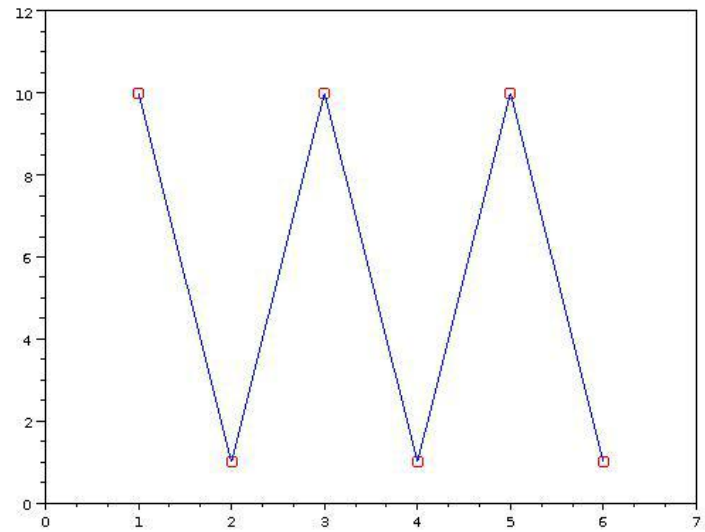
Functions f(x) = y

Straight line connecting P_i and P_{i+1}

$$f(x) = y_1 + (y_2 - y_1)\frac{x - x_1}{x_2 - x_1}$$



Data



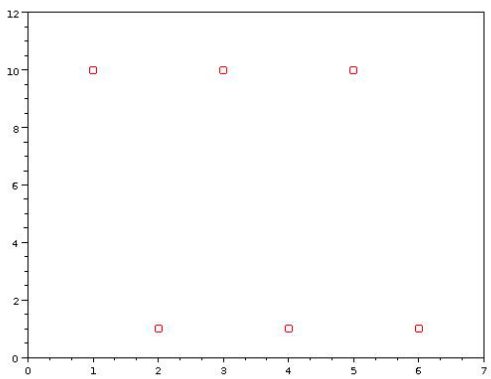
Linear Interpolation

Shepard Interpolation

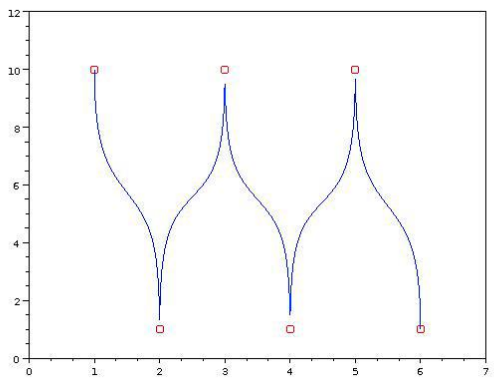
Functions f(x) = y

Weighted sum of points P_i , $i = 1....N$

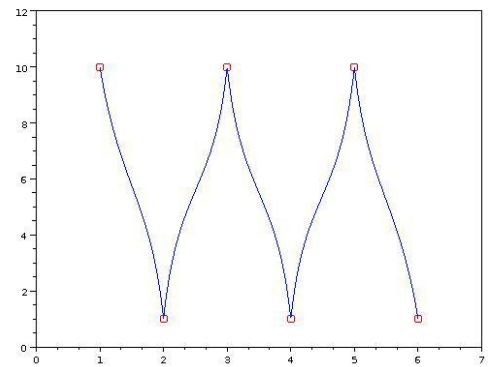
$$f(x) = \sum_{i}^{N} \frac{w_i(x) y_i}{\sum_i^N w_i(x)} \qquad w_i(x) = d(x, x_i)^{-p}$$



Data



p = 0.5



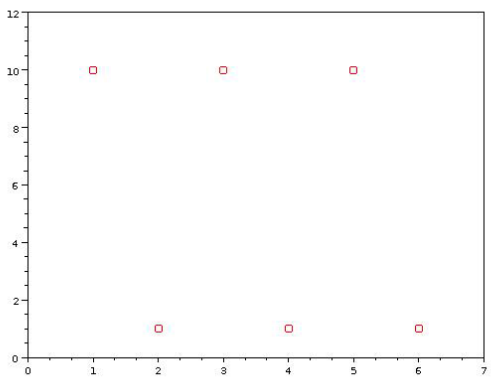
p = 1.0

Shepard Interpolation

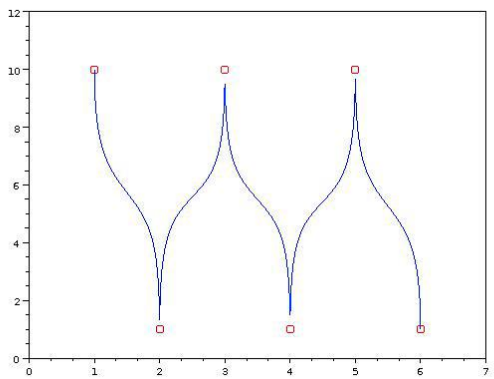
Functions f(x) = y

Weighted sum of points P_i , $i = 1....N$

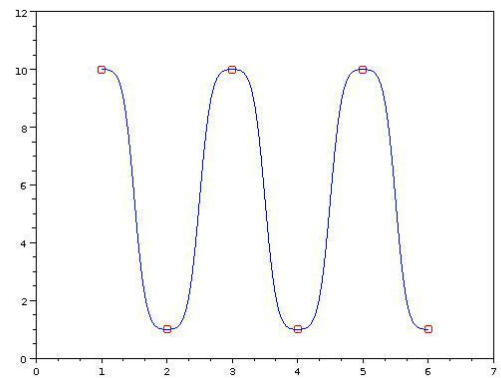
$$f(x) = \sum_{i}^{N} \frac{w_i(x) y_i}{\sum_i^N w_i(x)} \qquad w_i(x) = d(x, x_i)^{-p}$$



Data



p = 0.5



p = 3.0

Polynomial Interpolation

Functions f(x) = y

There exists a unique polynomial of degree N-1 obtained by interpolating N points

$$a_1 + a_2 x_1 + a_3 x_1^2 + \ldots\ldots\ldots a_N x_1^{N-1} = y_1$$

$$a_1 + a_2 x_2 + a_3 x_2^2 + \ldots\ldots\ldots a_N x_2^{N-1} = y_2$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad \ldots\ldots$$

$$a_1 + a_2 x_N + a_3 x_N^2 + \ldots\ldots\ldots a_N x_N^{N-1} = y_N$$

Polynomial Interpolation

Functions f(x) = y

There exists a unique polynomial of degree N-1 obtained by interpolating N points

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{N-1} \\ . & & & & \\ . & & & & \\ . & & & & \\ 1 & x_N & x_N^2 & \dots & x_N^{N-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_N \end{bmatrix}$$

Polynomial Interpolation

Functions f(x) = y

There exists a unique polynomial of degree N-1 obtained by interpolating N points

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{N-1} \\ . & & & & \\ . & & & & \\ . & & & & \\ 1 & x_N & x_N^2 & \dots & x_N^{N-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_N \end{bmatrix}$$

Expensive solution !!

Polynomial Interpolation

Functions f(x) = y

There exists a unique polynomial of degree N-1 obtained by interpolating N points

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{N-1} \\ . & & & & \\ . & & & & \\ . & & & & \\ 1 & x_N & x_N^2 & \dots & x_N^{N-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_N \end{bmatrix}$$

Expensive solution !!

Evaluation with the basis of Lagrange

Polynomial Interpolation

Functions f(x) = y

Weighting with the Lagrangian polynomial bases

$$f(x) = \sum_{i=1}^{N} y_i \mathcal{L}_{N,i}(x)$$

$$\mathcal{L}_{N,i}(x) = \prod_{j=1, j\neq i}^{N} \frac{x - x_j}{x_i - x_j} = \frac{(x - x_1)}{(x_i - x_1)} \cdots\cdots \frac{(x - x_{i-1})}{(x_i - x_{i-1})} \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \frac{(x - x_N)}{(x_i - x_N)}$$

Polynomial Interpolation

Functions f(x) = y

Weighting with the Lagrangian polynomial bases

$$f(x) = \sum_{i=1}^{N} y_i \mathcal{L}_{N,i}(x)$$

$$\mathcal{L}_{N,i}(x) = \prod_{j=1, j\neq i}^{N} \frac{x - x_j}{x_i - x_j} = \frac{(x - x_1)}{(x_i - x_1)} \cdots\cdots \frac{(x - x_{i-1})}{(x_i - x_{i-1})} \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \frac{(x - x_N)}{(x_i - x_N)}$$

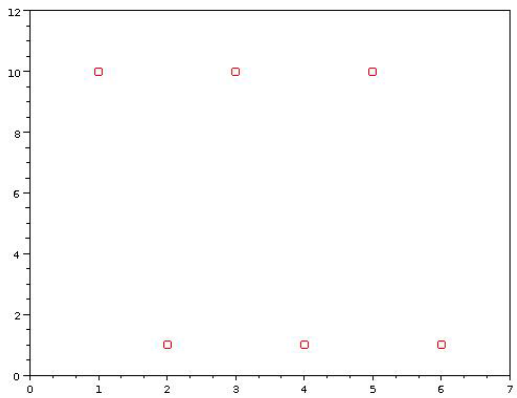
$$\mathcal{L}_{N,i}(x_j) = 0$$
$$\mathcal{L}_{N,i}(x_i) = 1$$

Polynomial Interpolation

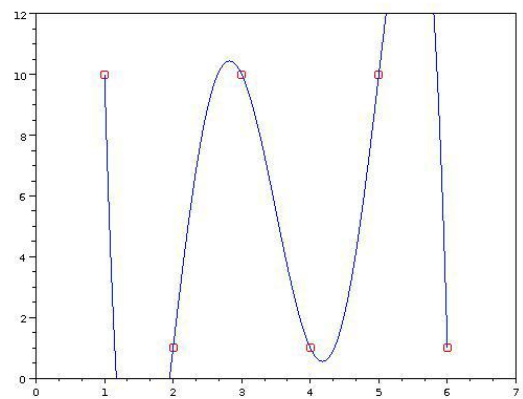
Functions f(x) = y

Weighting with the Lagrangian polynomial bases

$$f(x) = \sum_{i=1}^{N} y_i \mathcal{L}_{N,i}(x)$$



Data

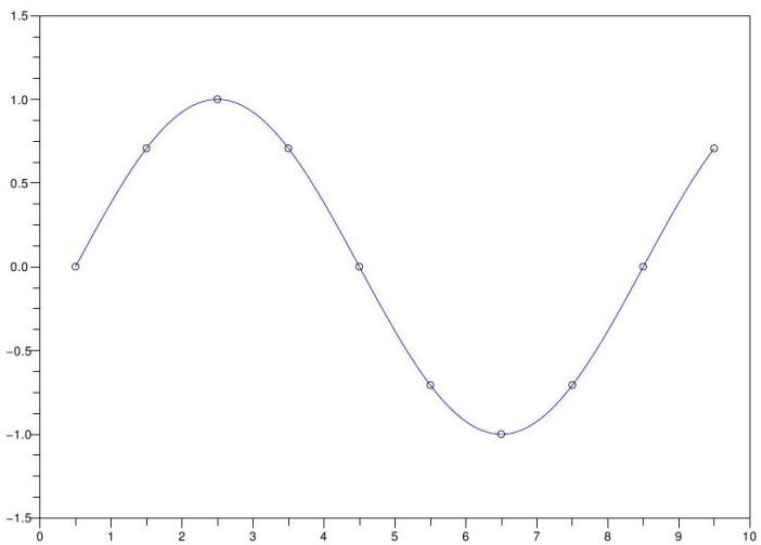


Polynomial Interpolation

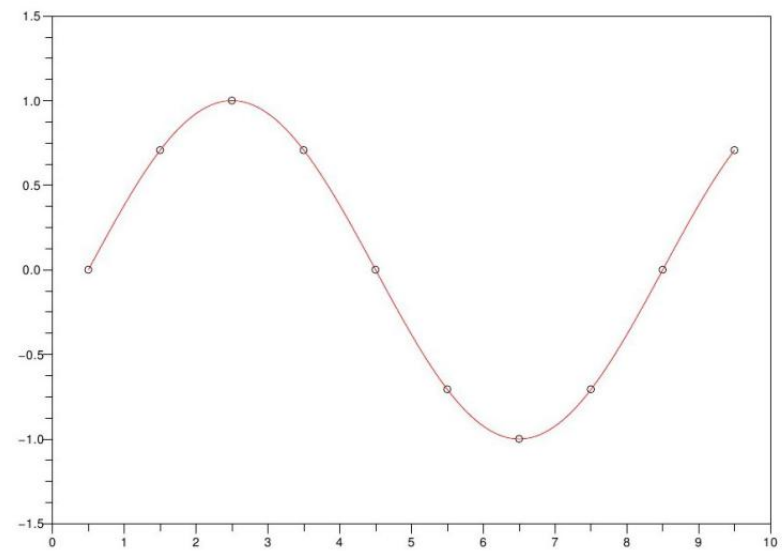
Polynomial Interpolation

Functions f(x) = y

Example $$y = sin(x_i \pi / 4)$$



Data



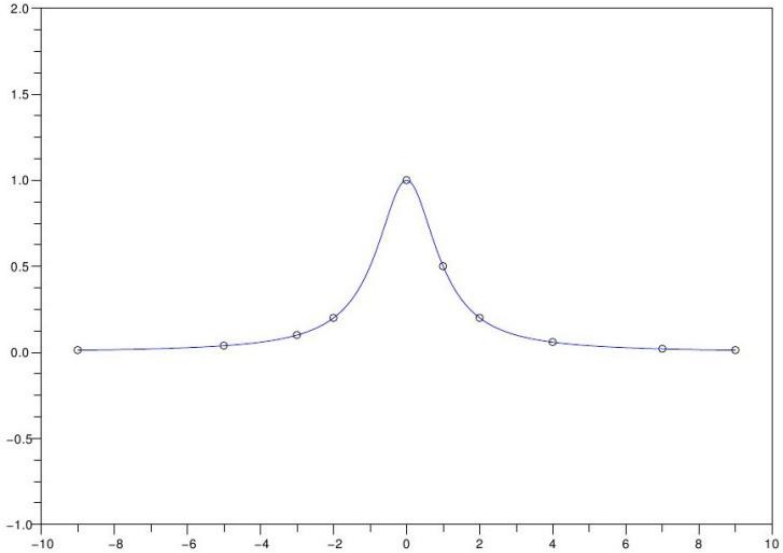
Polynomial Interpolation

Polynomial Interpolation

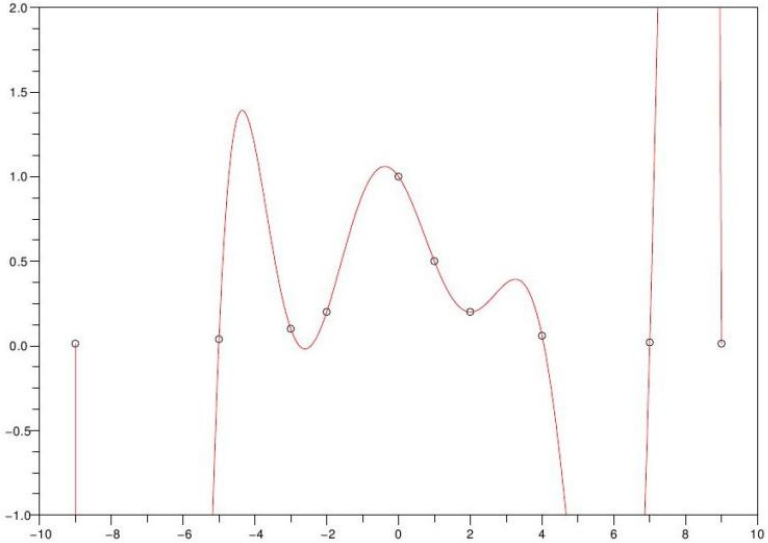
Functions f(x) = y

Example

$$y = \frac{1}{x^2 + 1}$$



Data

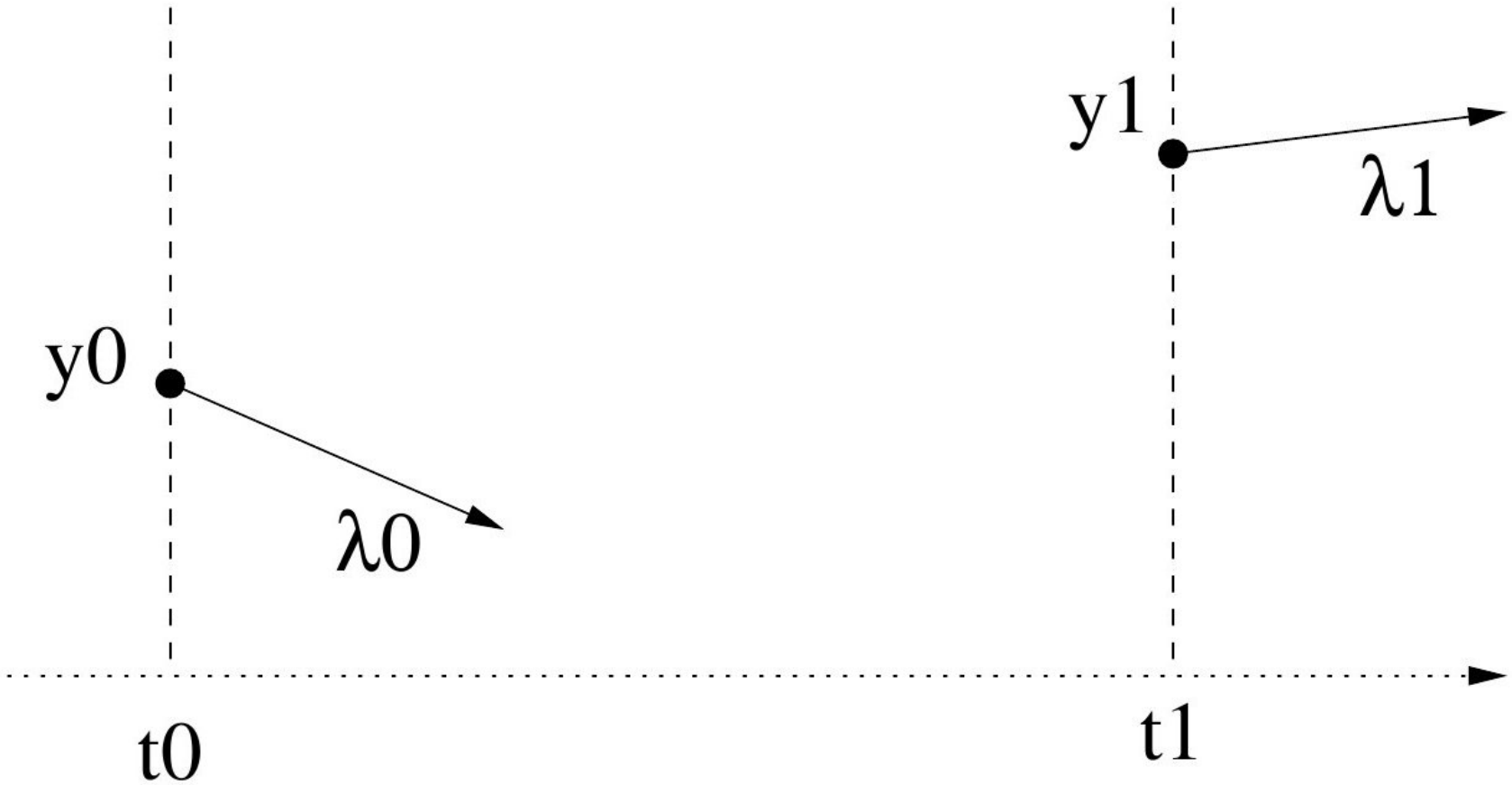


Polynomial Interpolation

Polynomial Interpolation by Piece

Functions f(x) = y and f'(x) = λ

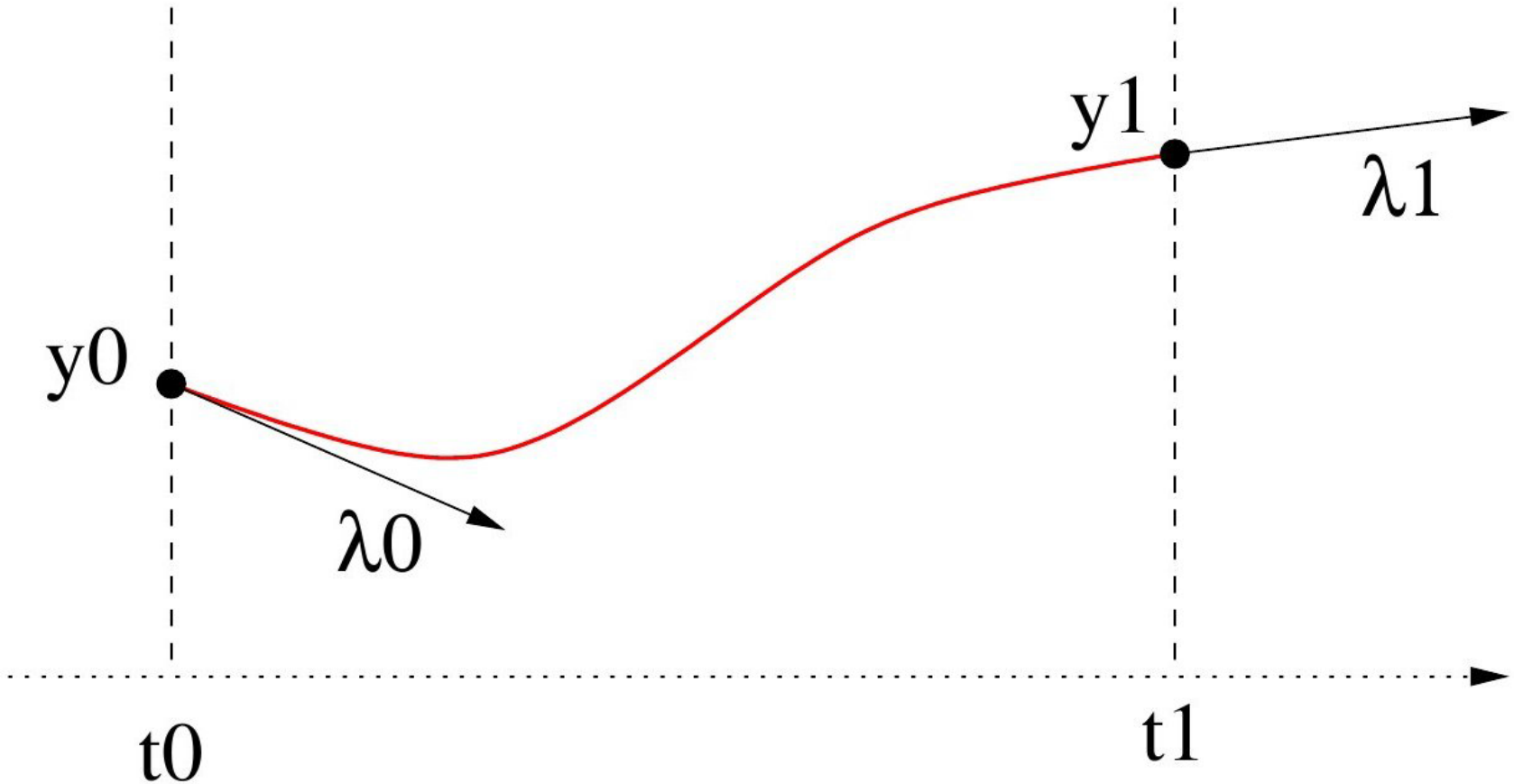
Hermite Interpolation



Polynomial Interpolation by Piece

Functions f(x) = y and f'(x) = λ

Hermite Interpolation



Polynomial Interpolation by Piece

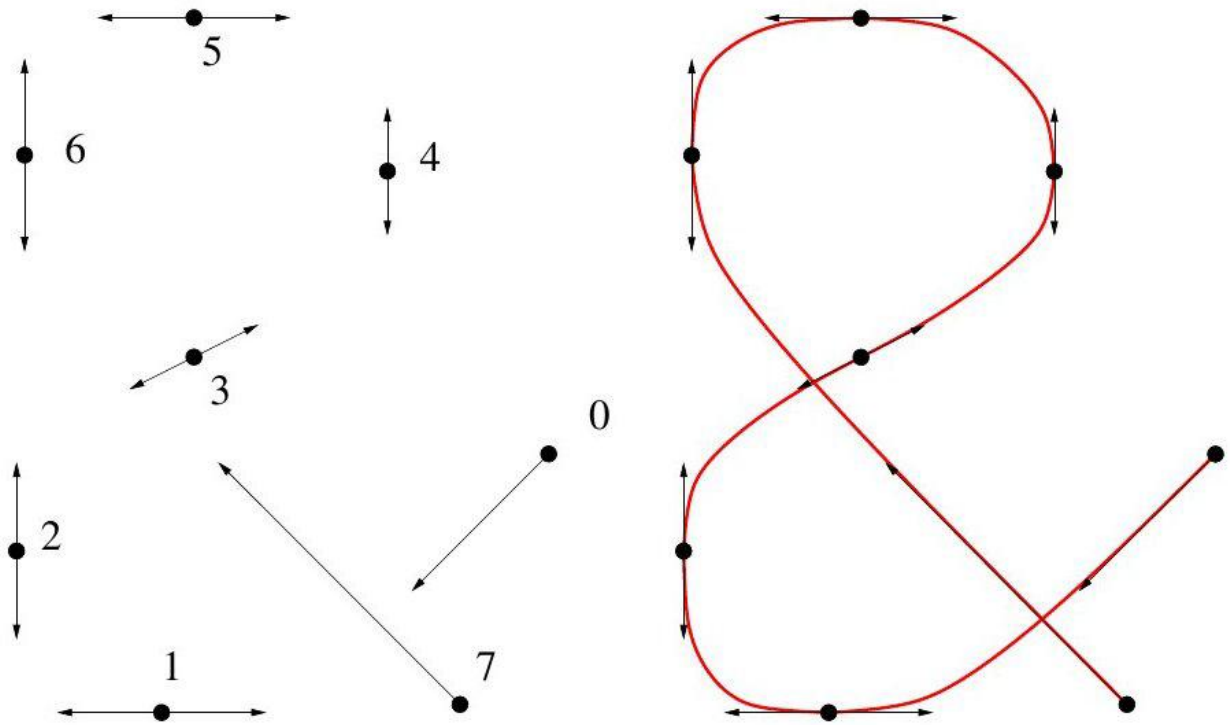
Functions f(x) = y and f'(x) = λ

$$f(x) = \begin{bmatrix} s^3 \\ s^2 \\ s \\ 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \lambda_1 \\ \lambda_2 \end{bmatrix}$$

where $s = \dfrac{(x - x_1)}{(x_2 - x_1)} \in [0, 1]$

Polynomial Interpolation by Piece

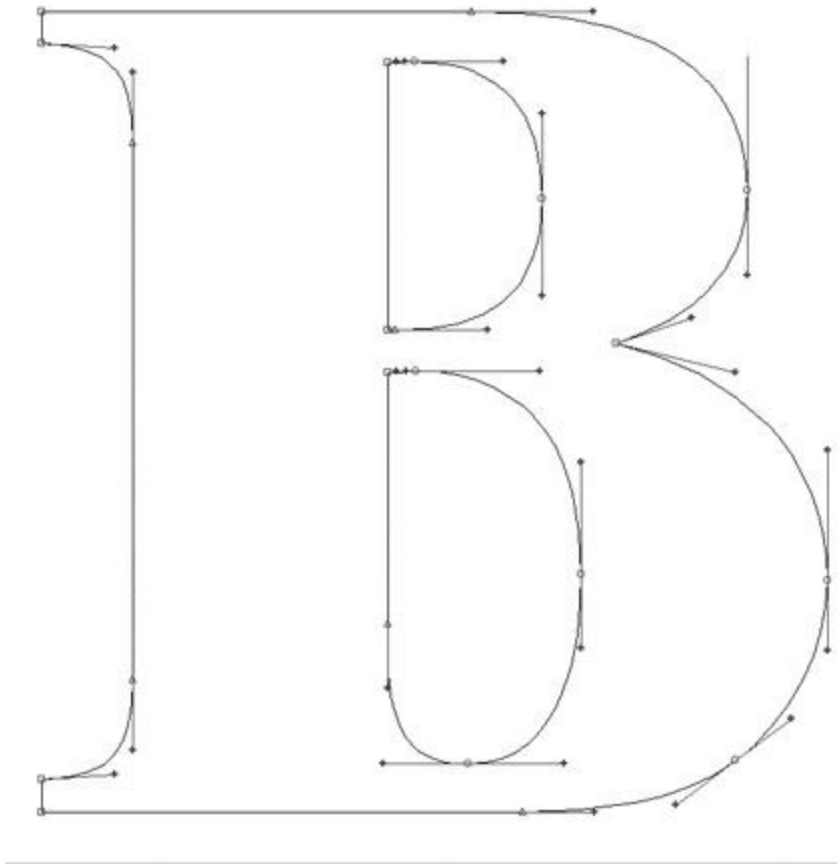
Functions f(x) = y and f’(x) = λ



Hermite Interpolation

Polynomial Interpolation by Piece

Functions $f(x) = y$ and $f'(x) = \lambda$



Variation : Bezier Curves

Polynomial Interpolation by Piece

Functions $f_0(x_i) = f_1(x_i)$, $f'_0(x_i) = f'_1(x_i)$ and $f''_0(x_i) = f''_1(x_i)$

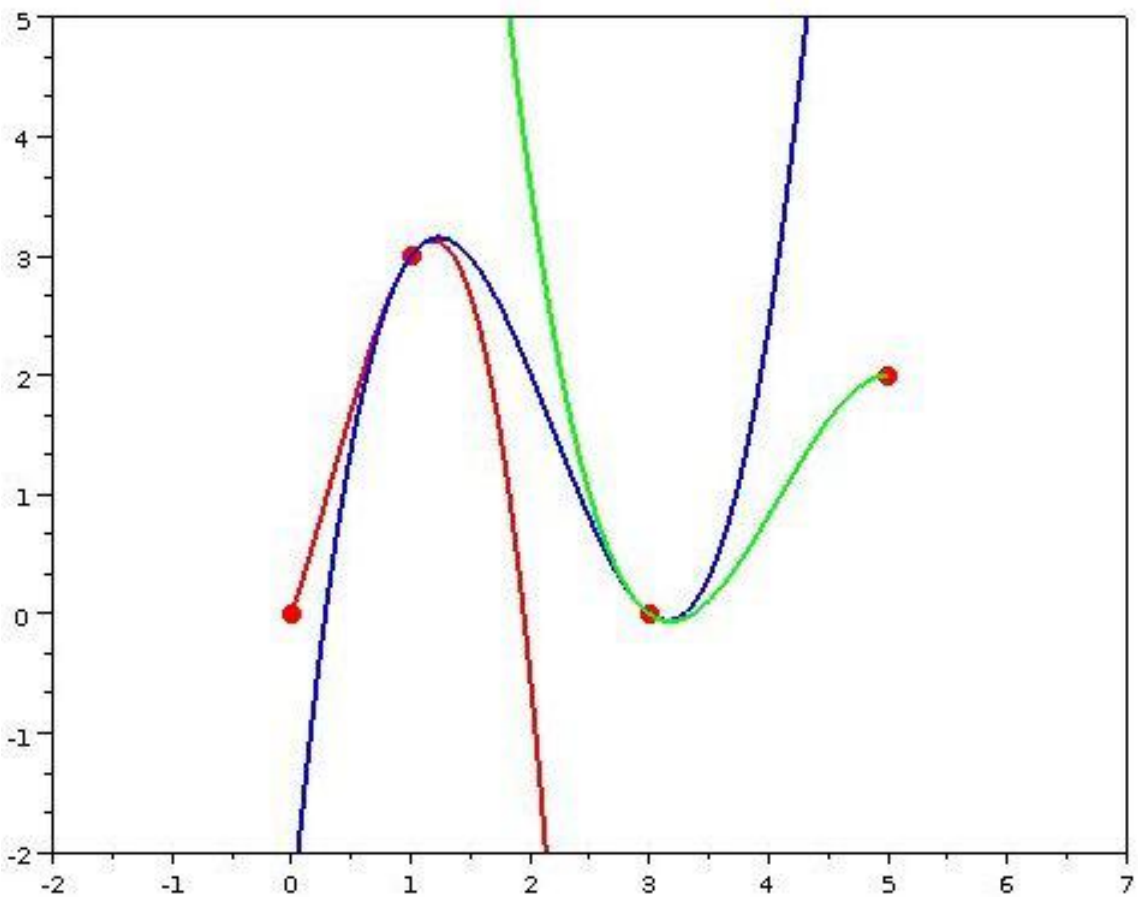
Spline Interpolation (for 4 points)

$$
\begin{bmatrix}
0 & 1 & 2x_1 & 3x_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & x_1 & x_1^2 & x_1^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & x_2 & x_2^2 & x_2^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2x_2 & 3x_2^2 & 0 & -1 & -2x_2 & -3x_2^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 6x_2 & 0 & 0 & -2 & -6x_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & x_2 & x_2^2 & x_2^3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & x_3 & x_3^2 & x_3^3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 2x_3 & 3x_3^2 & 0 & -1 & -2x_3 & -3x_3^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 6x_3 & 0 & 0 & -2 & -6x_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_3 & x_3^2 & x_3^3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_4 & x_4^2 & x_4^3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2x_4 & 3x_4^2
\end{bmatrix}
\begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \\ a_1 \\ b_1 \\ c_1 \\ d_1 \\ a_2 \\ b_2 \\ c_2 \\ d_2 \end{bmatrix}
=
\begin{bmatrix} \lambda_1 \\ y_1 \\ y_2 \\ 0 \\ 0 \\ y_2 \\ y_3 \\ 0 \\ 0 \\ y_3 \\ y_4 \\ \lambda_4 \end{bmatrix}
$$

Polynomial Interpolation by Piece

Functions $f_0(x_i) = f_1(x_i)$, $f'_0(x_i) = f'_1(x_i)$ and $f''_0(x_i) = f''_1(x_i)$

Spline Interpolation (for 4 points)

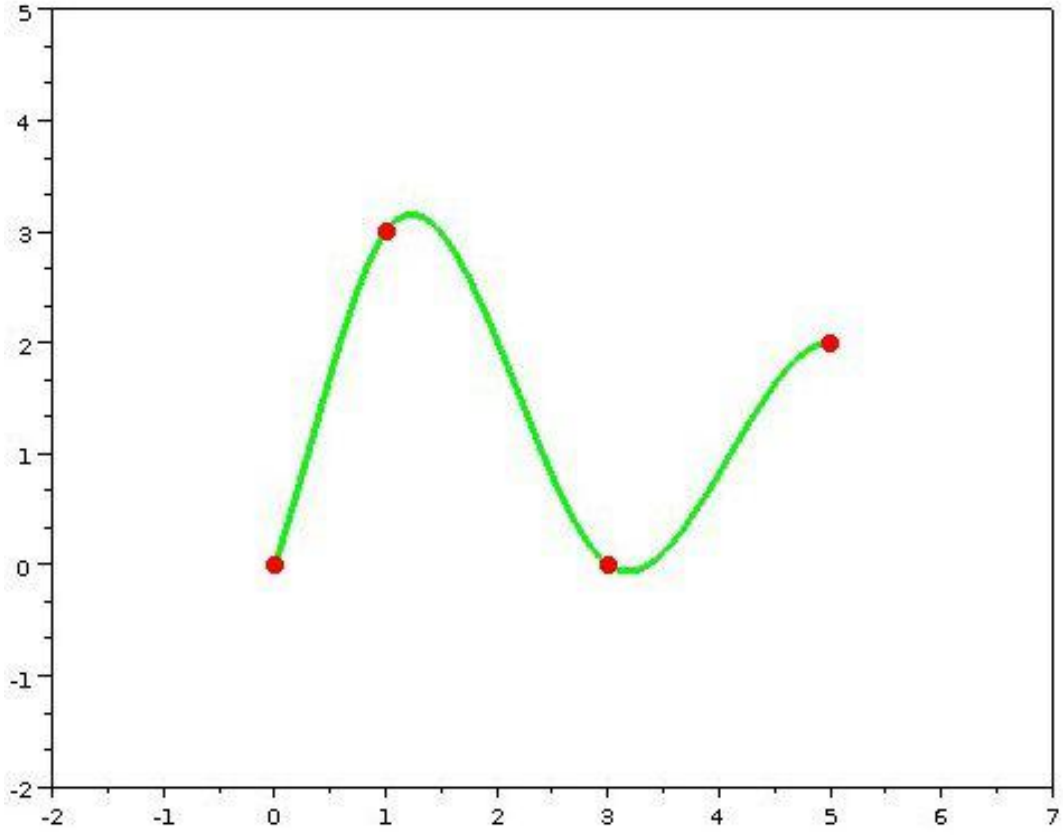


3 cubics obtained

Polynomial Interpolation by Piece

Functions $f_0(x_i) = f_1(x_i)$, $f'_0(x_i) = f'_1(x_i)$ and $f''_0(x_i) = f''_1(x_i)$

Spline Interpolation (for 4 points)

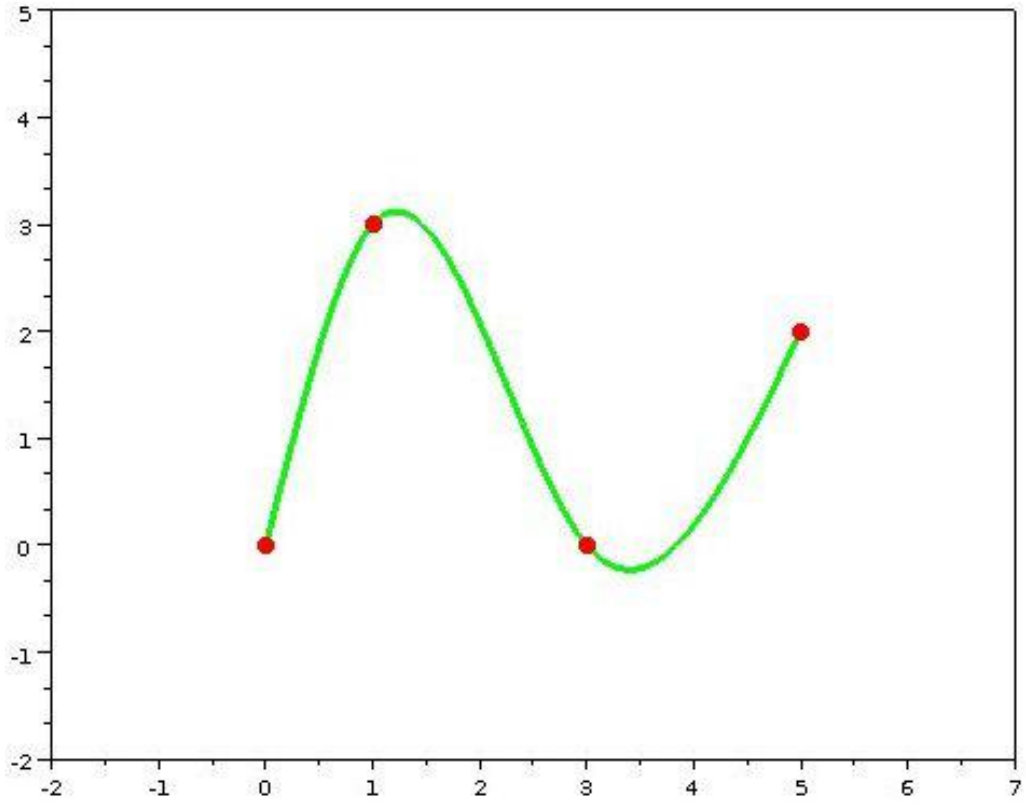


Splines with tangents for the borders given

Polynomial Interpolation by Piece

Functions $f_0(x_i) = f_1(x_i)$, $f'_0(x_i) = f'_1(x_i)$ and $f''_0(x_i) = f''_1(x_i)$

Spline Interpolation (for 4 points)

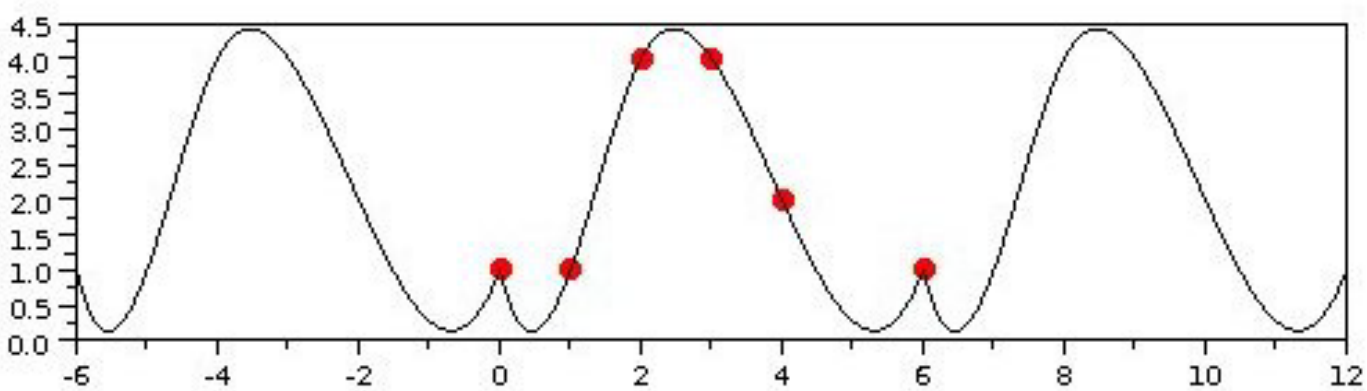


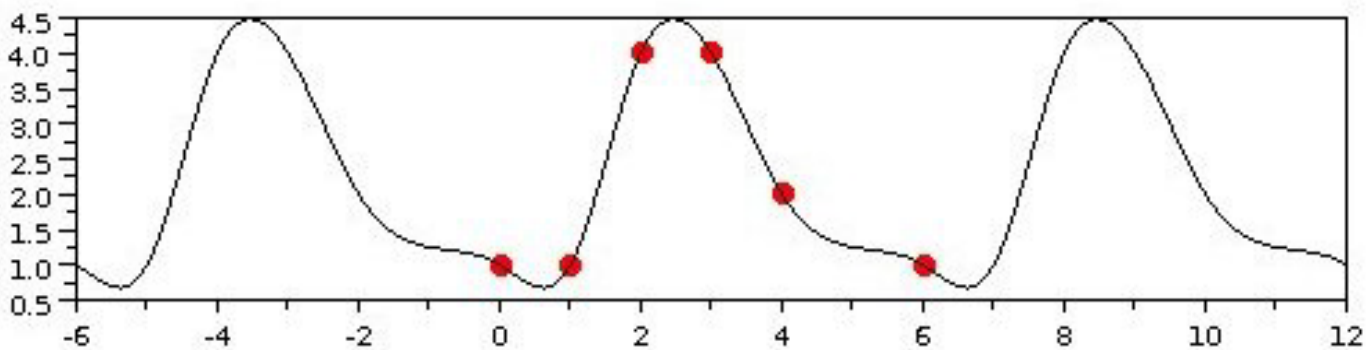
Spline with $f''_0(x_0) = 0$ and $f''_2(x_4) = 0$

Polynomial Interpolation by Piece

Functions $f_0(x_i) = f_1(x_i)$, $f'_0(x_i) = f'_1(x_i)$ and $f''_0(x_i) = f''_1(x_i)$

Spline Interpolation



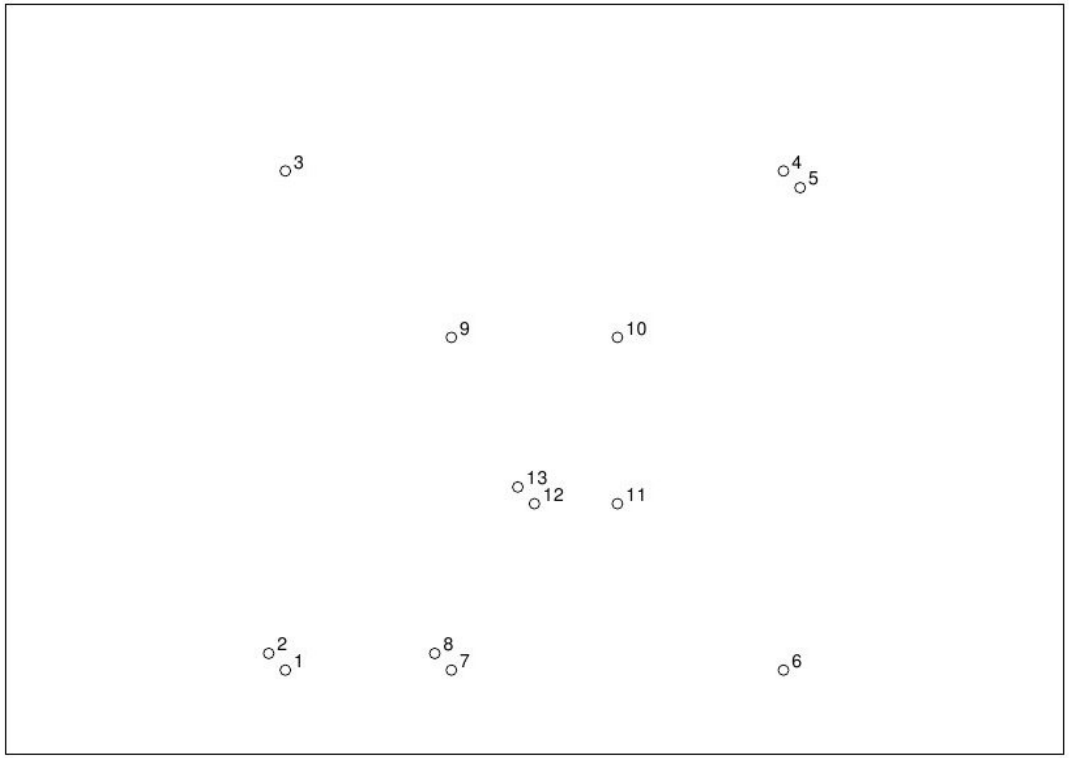


The spline with periodic constraints

Polynomial Interpolation

Functions $f(t) = (x(t), y(t))$ (Data = t, x, y)

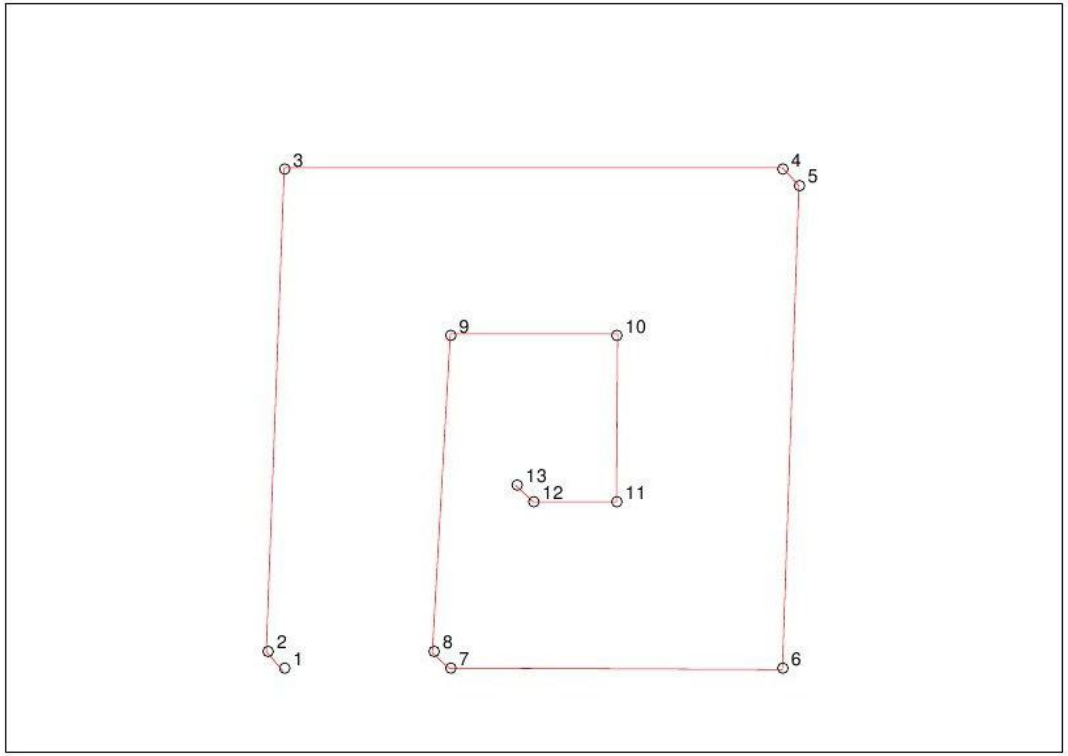
- Interpolate (t_i, x_i) by a function $x(t)$
- Interpolate (t_i, y_i) by a function $y(t)$



Polynomial Interpolation

Functions $f(t) = (x(t), y(t))$ (Data = t, x, y)

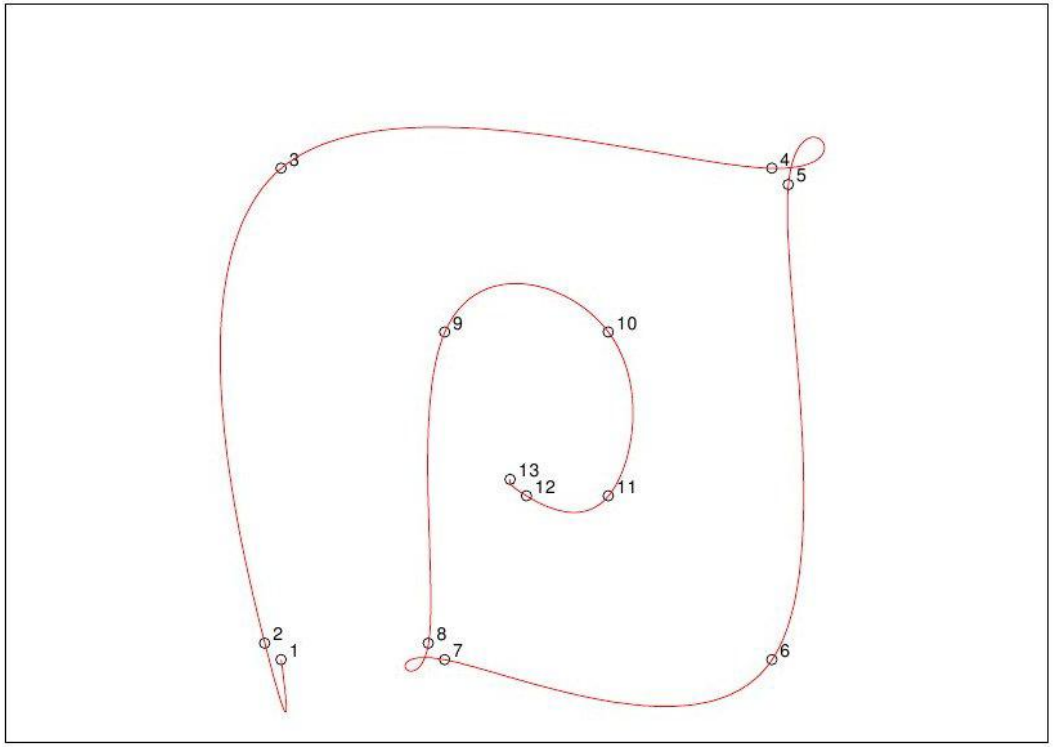
- Interpolate (t_i, x_i) by a function $x(t)$
- Interpolate (t_i, y_i) by a function $y(t)$



Polynomial Interpolation

Functions $f(t) = (x(t), y(t))$ (Data = t, x, y)

- Interpolate (t_i, x_i) by a function $x(t)$
- Interpolate (t_i, y_i) by a function $y(t)$

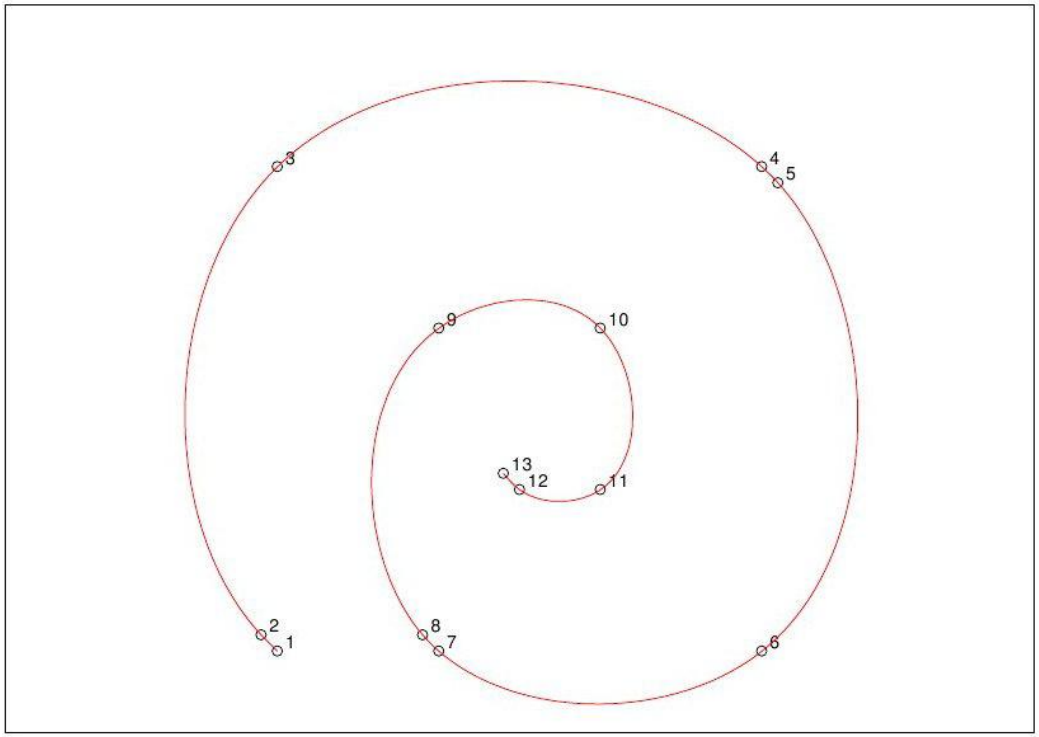


Cubic spline interpolation with $t_i = i$

Polynomial Interpolation

Functions $f(t) = (x(t), y(t))$ (Data = t, x, y)

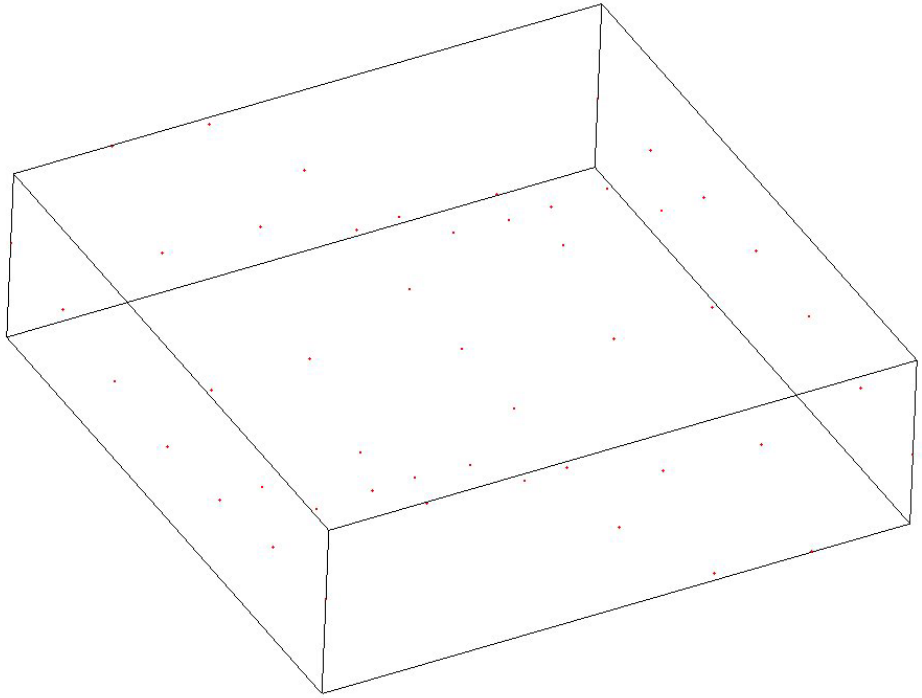
- Interpolate (t_i, x_i) by a function $x(t)$
- Interpolate (t_i, y_i) by a function $y(t)$



Cubic spline interpolation with $t_{i+1} - t_i = ||P_{i+1} - P_i||$

General Framework

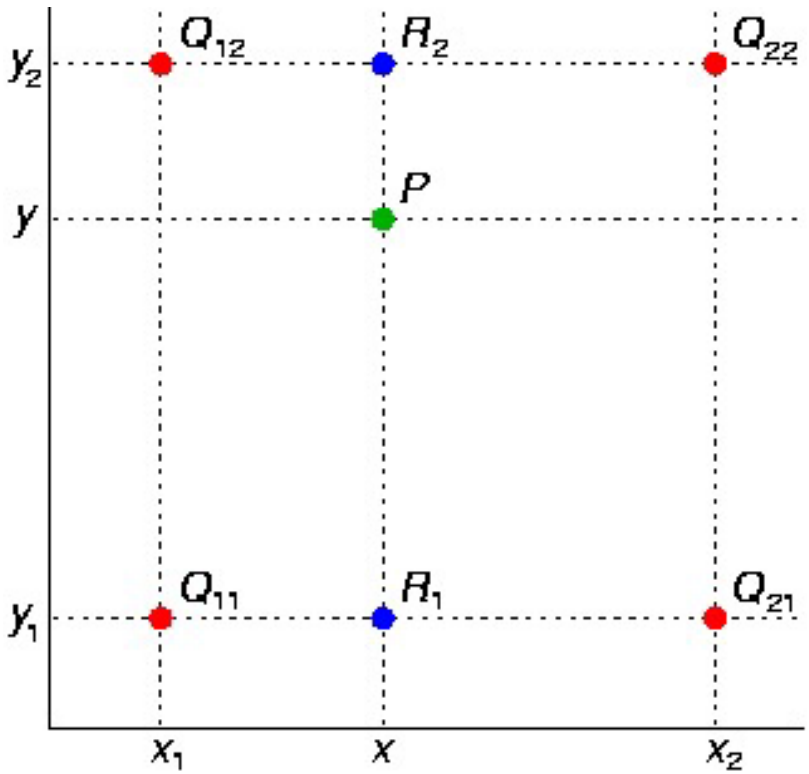
Functions $f(x, y) = z$



4 x 4 points such that $f(x, y) = \cos(x) * \sin(y)$

Bilinear Interpolation

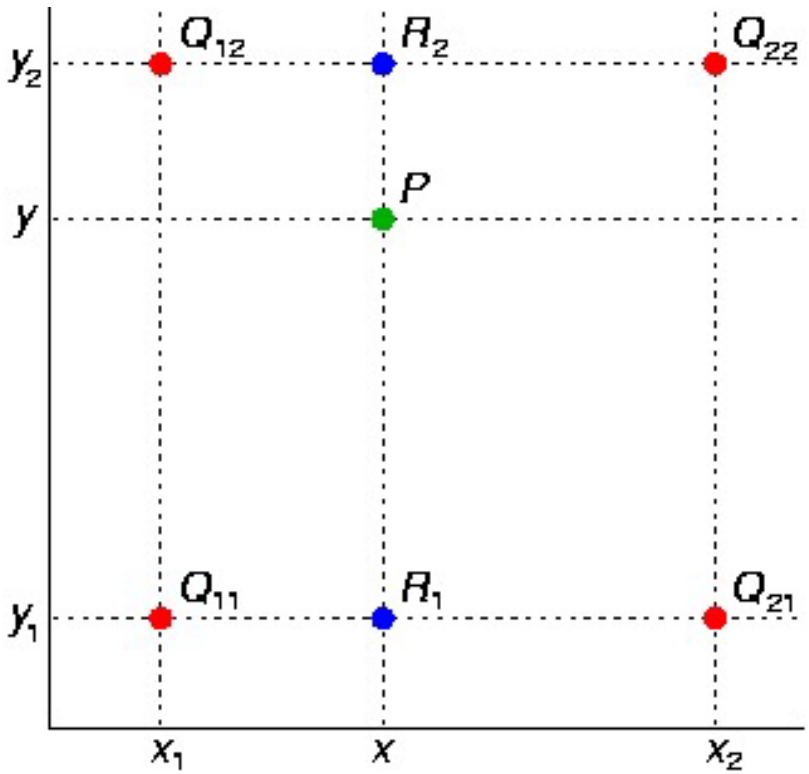
Functions $f(x, y) = z$



Bilinear Interpolation

Functions *f(x, y) = z*

$$R_1 = z(1,1) + (z(2,1) - z(1,1))\frac{x - x_1}{x_2 - x_1}$$

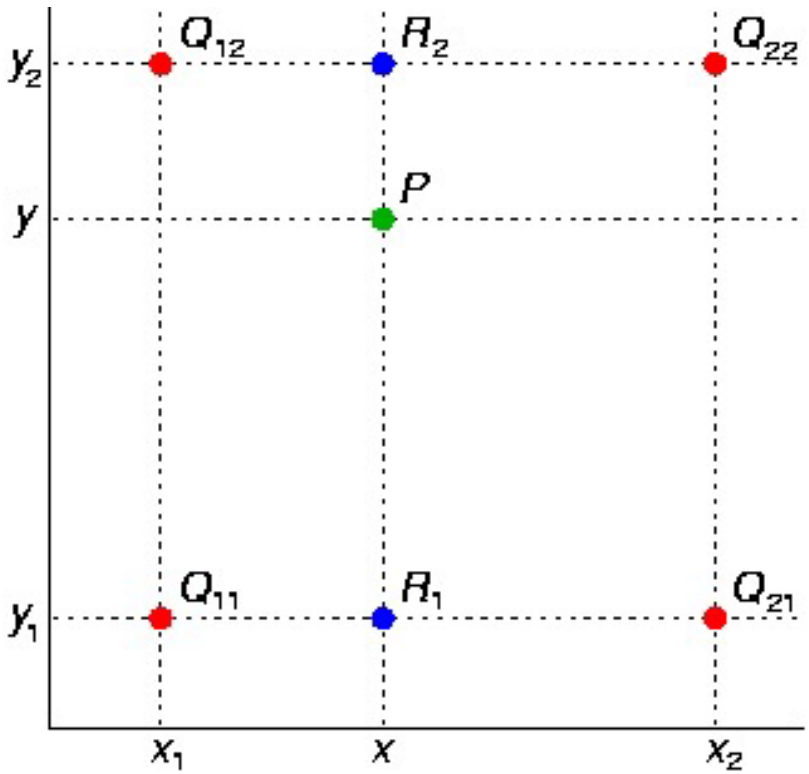


Bilinear Interpolation

Functions *f(x, y) = z*

$$R_1 = z(1,1) + (z(2,1) - z(1,1))\frac{x - x_1}{x_2 - x_1}$$

$$R_2 = z(1,2) + (z(2,2) - z(1,2))\frac{x - x_1}{x_2 - x_1}$$



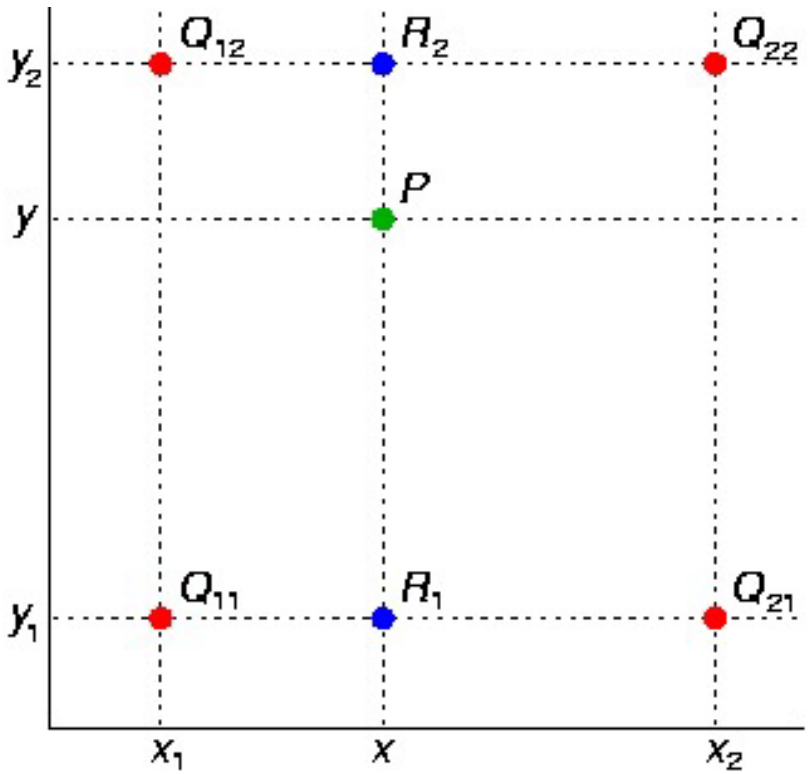
Bilinear Interpolation

Functions *f(x, y) = z*

$$R_1 = z(1,1) + (z(2,1) - z(1,1))\frac{x - x_1}{x_2 - x_1}$$

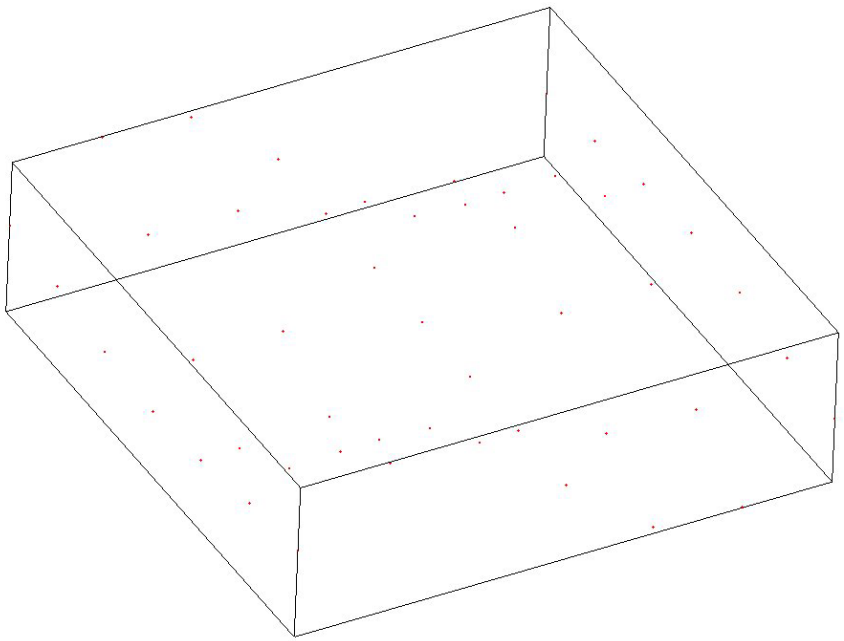
$$R_2 = z(1,2) + (z(2,2) - z(1,2))\frac{x - x_1}{x_2 - x_1}$$

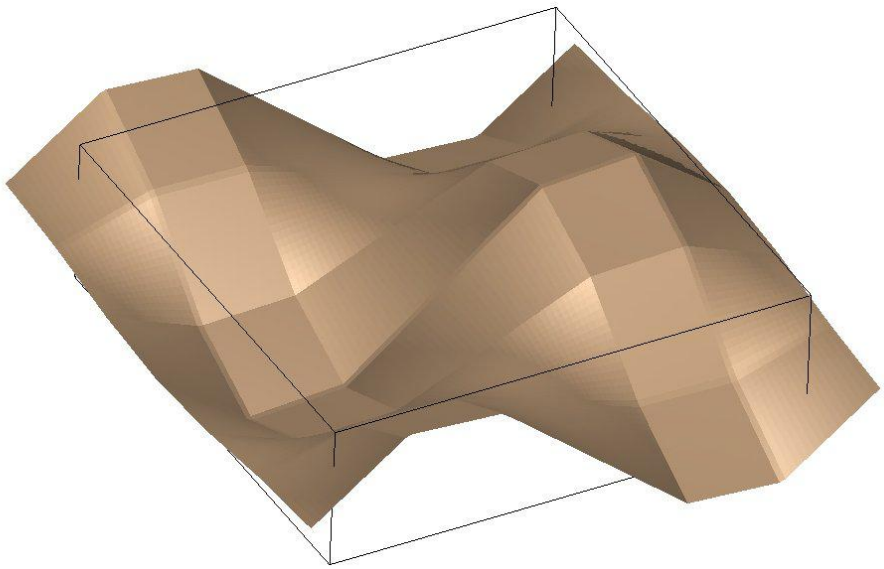
$$z = R_1 + (R_2 - R_1)\frac{y - y_1}{y_2 - y_1}$$



Bilinear Interpolation

Functions $f(x, y) = z$





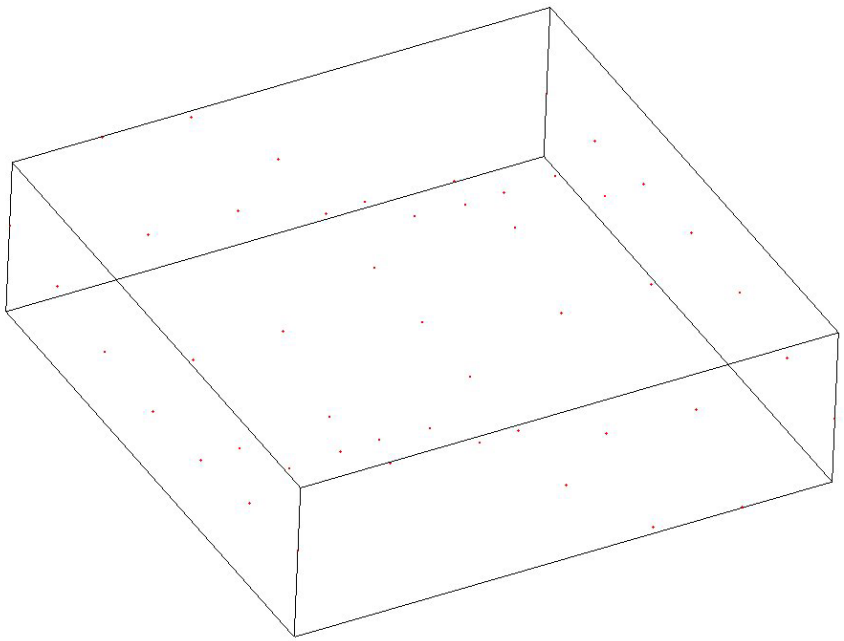
4 x 4 points such that $f(x, y) = cos(x) * sin(y)$

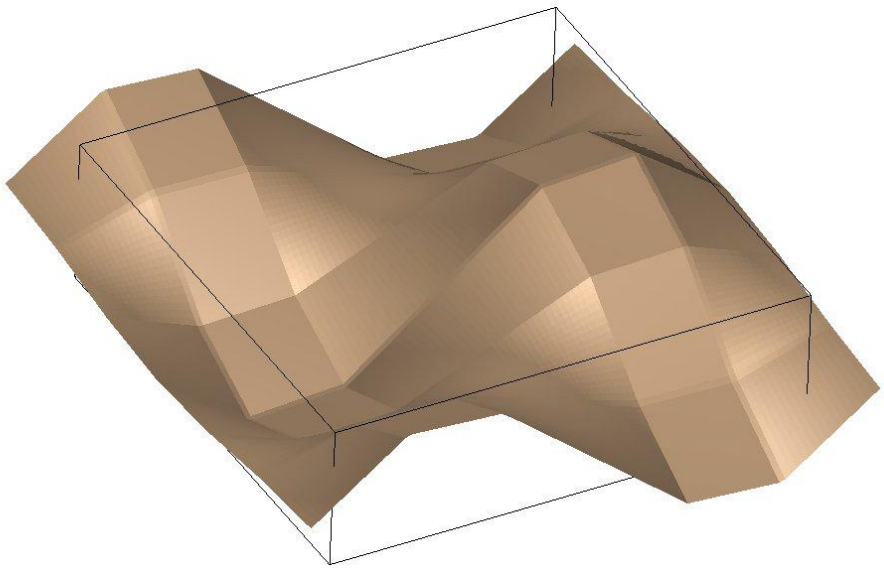
Bilinear Interpolation

Functions *f(x, y) = z*

Only distance calculation changes from 1D case !

$$f(x,y) = \sum_{i=1}^{N} \frac{w_i(x,y) z_i}{\sum_{i=1}^{N} w_i(x,y)} \quad , w_i(x,y) = d((x,y),(x_i,y_i))^{-p}$$





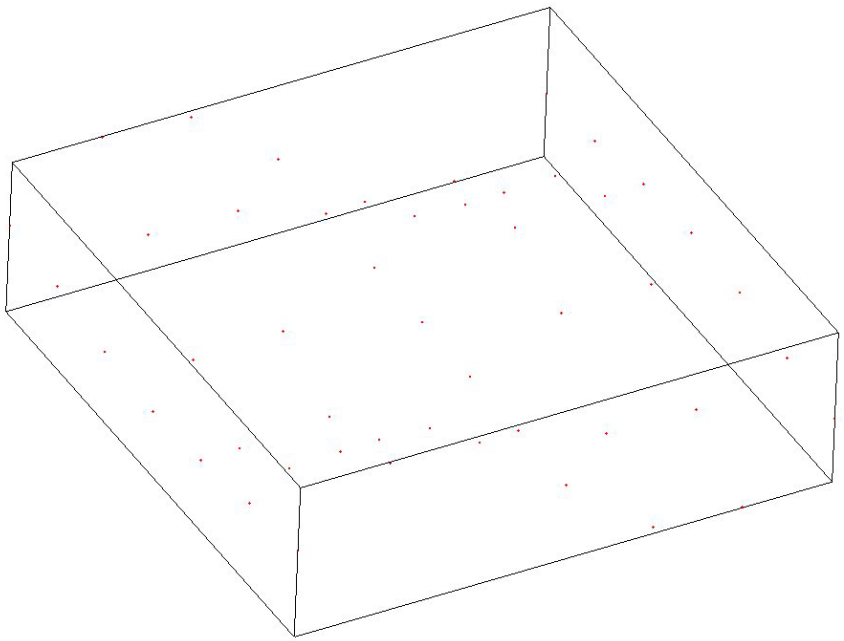
4 x 4 points such that *f(x, y) = cos(x) * sin(y)* and p = 1

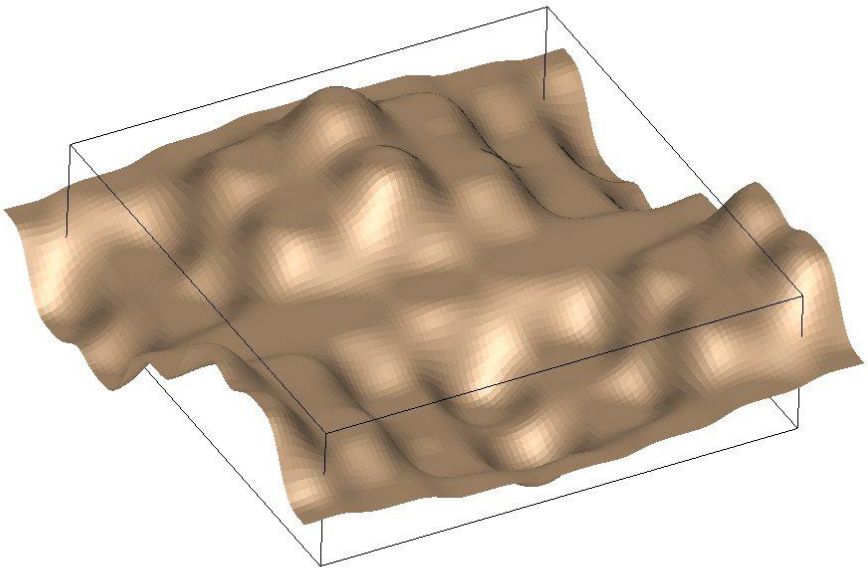
Bilinear Interpolation

Functions *f(x, y) = z*

Only distance calculation changes from 1D case !

$$f(x,y) = \sum_{i=1}^{N} \frac{w_i(x,y) z_i}{\sum_{i=1}^{N} w_i(x,y)} \quad , w_i(x,y) = d((x,y),(x_i,y_i))^{-p}$$





4 x 4 points such that *f(x, y) = cos(x) * sin(y)* and p = 3

Lagrangian Interpolation

Functions *f(x, y) = z*

$$1D : f(x) = \sum_{i=1}^{N} y_i \mathcal{L}_{n,i}(x)$$

Lagrangian Interpolation

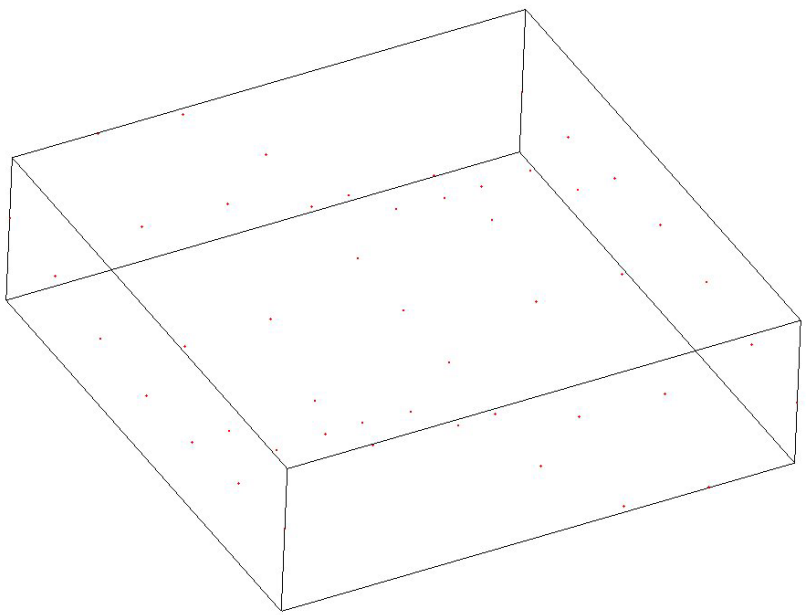
Functions *f(x, y) = z*

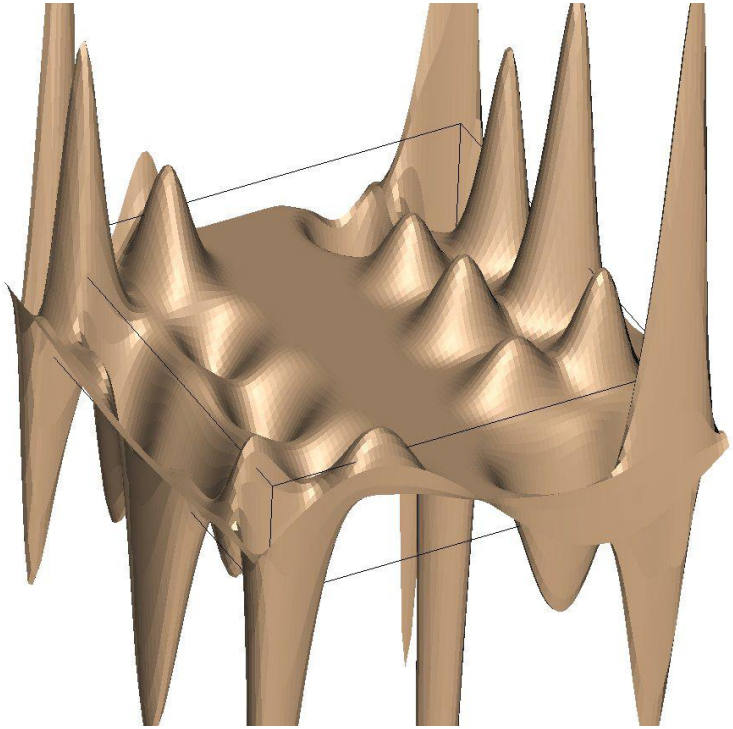
$$1D: f(x) = \sum_{i=1}^{N} y_i \mathcal{L}_{n,i}(x) \quad , 2D: f(x,y) = \sum_{i=1}^{N} \sum_{j=1}^{M} z_{ij} \mathcal{L}_{n,i}(x) \mathcal{L}_{m,j}(y)$$

Lagrangian Interpolation

Functions *f(x, y) = z*

$$1D: f(x) = \sum_{i=1}^{N} y_i \mathcal{L}_{n,i}(x) \quad , 2D: f(x,y) = \sum_{i=1}^{N} \sum_{j=1}^{M} z_{ij} \mathcal{L}_{n,i}(x) \mathcal{L}_{m,j}(y)$$

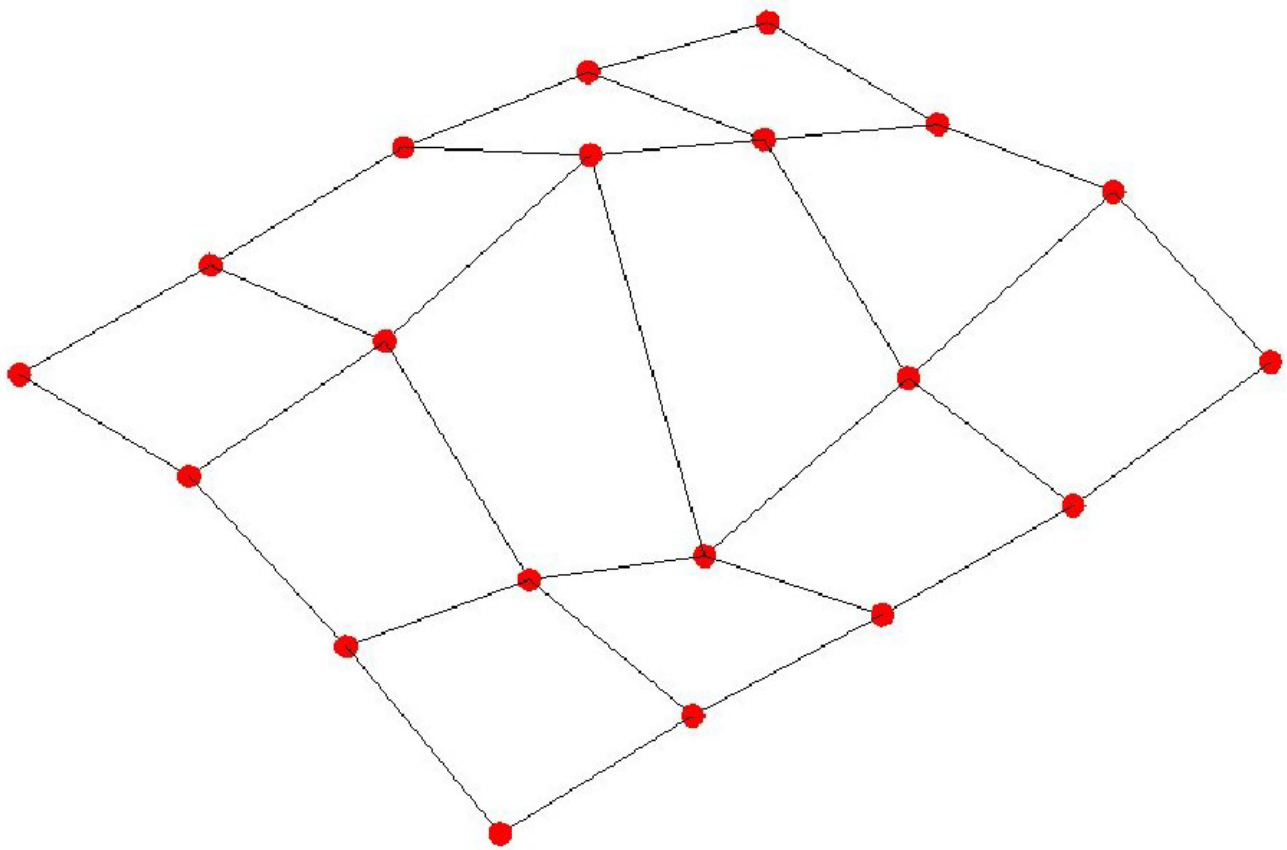




4 x 4 points such that *f(x, y) = cos(x) * sin(y)*

Tensor Product Interpolation

Functions $f(x, y) = z$

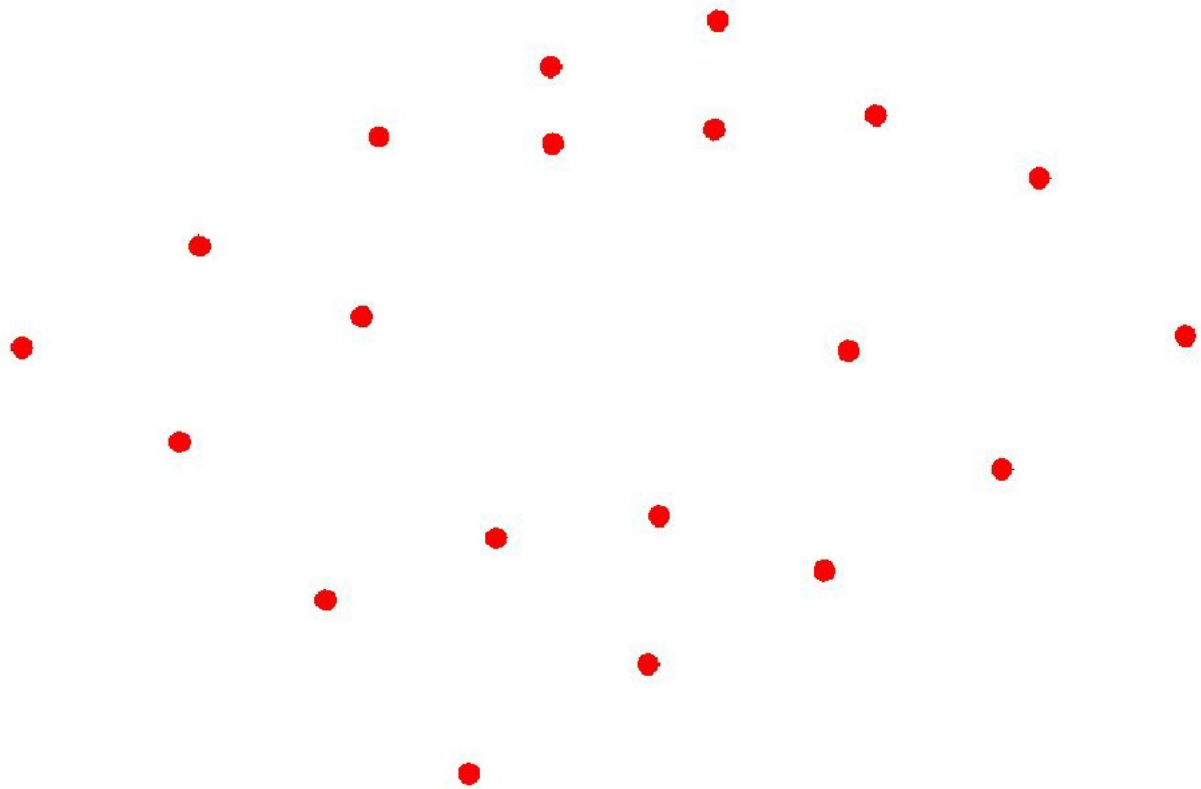


$P_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j} ; (u_i, v_j))$ $1 \leq i \leq n$ AND $1 \leq j \leq m$

forming a regular quadrangle

Tensor Product Interpolation

Functions $f(x, y) = z$

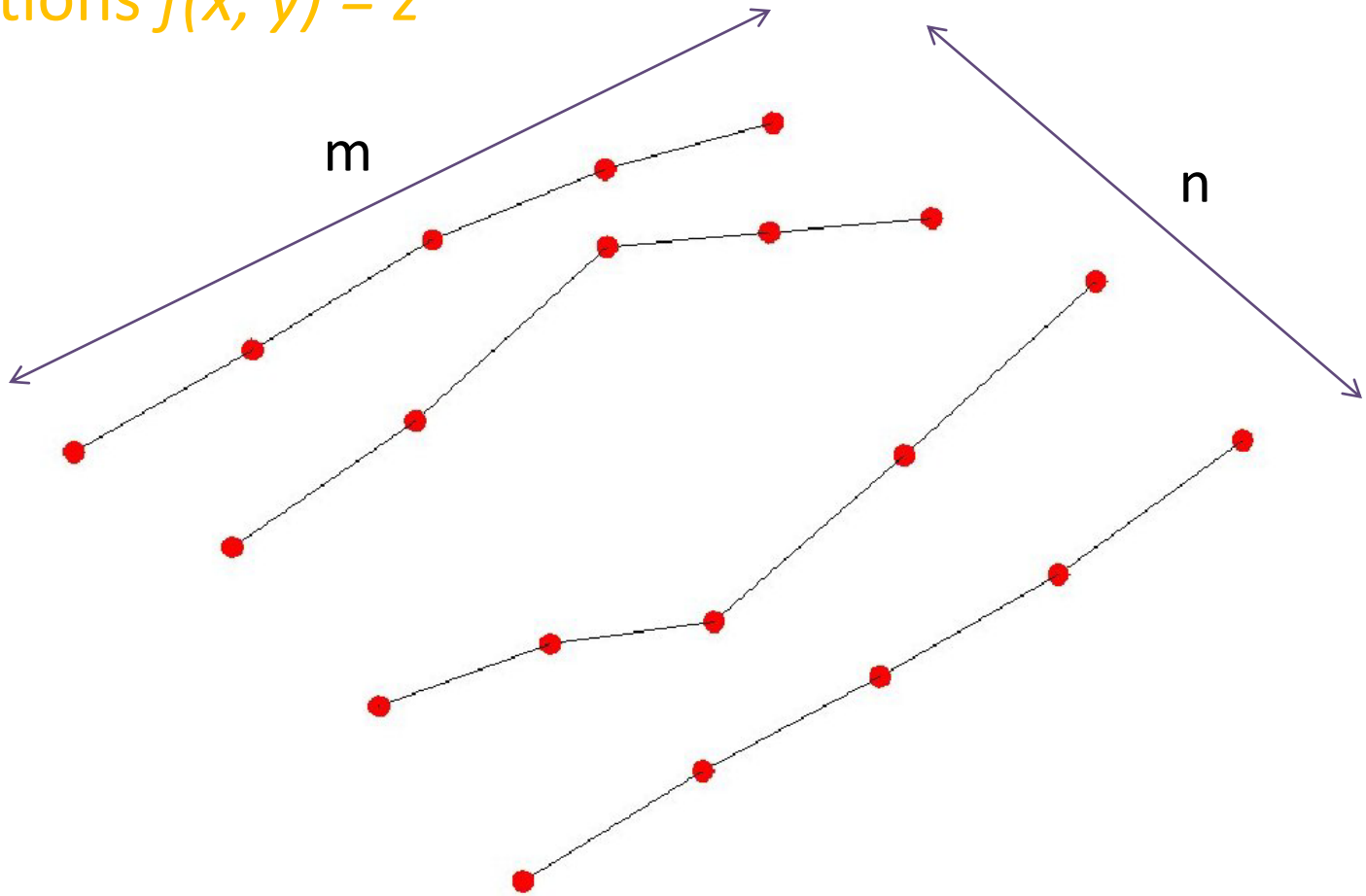


$P_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j} ; (u_i, v_j))$ $1 \leq i \leq n$ AND $1 \leq j \leq m$

forming a regular quadrangle

Tensor Product Interpolation

Functions $f(x, y) = z$

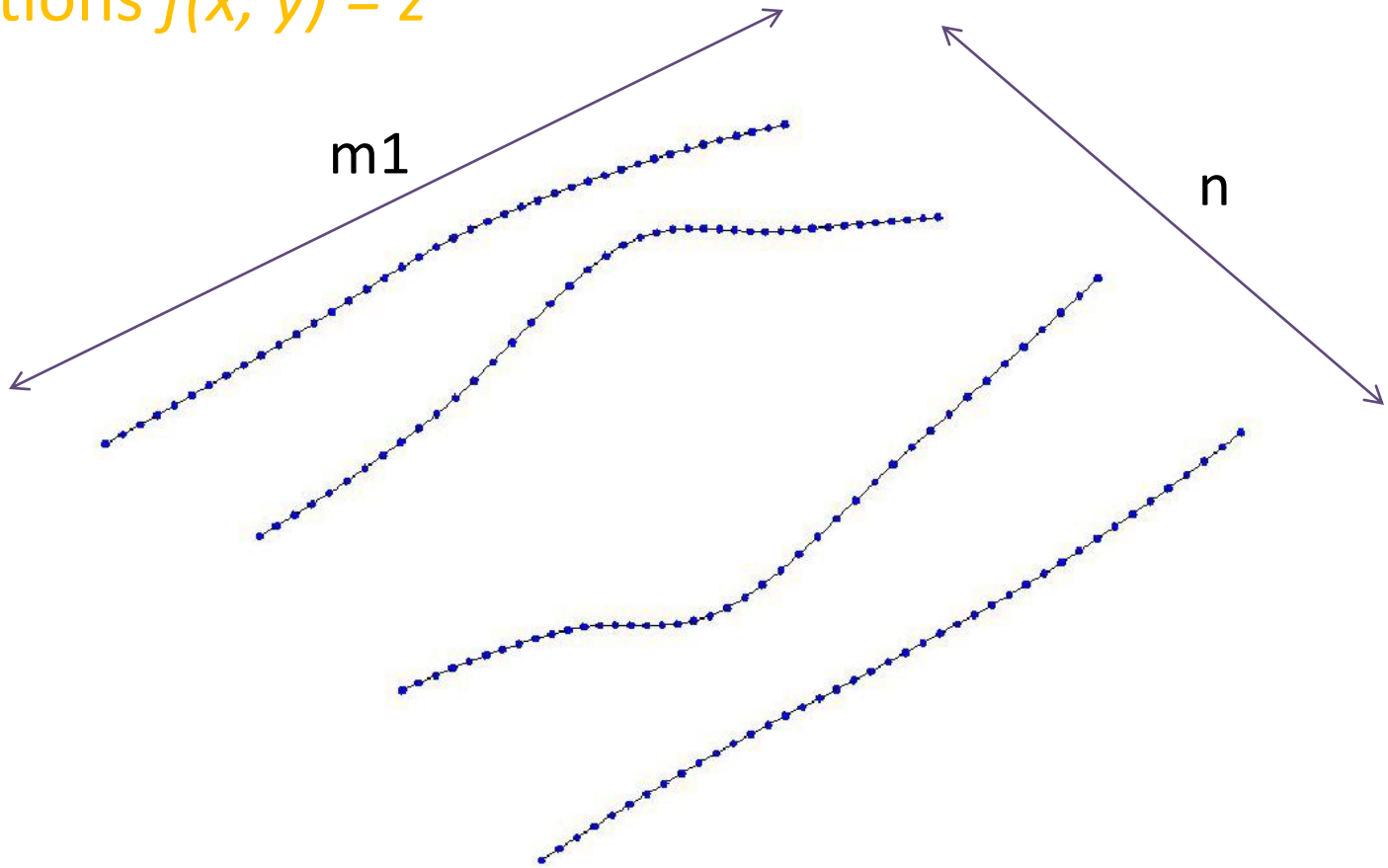


Interpolation after 1st dimension : interpolation of *n* curves

Here each curve interpolates m points $P_{i,j}$, $1 \leq j \leq m$

Tensor Product Interpolation

Functions $f(x, y) = z$

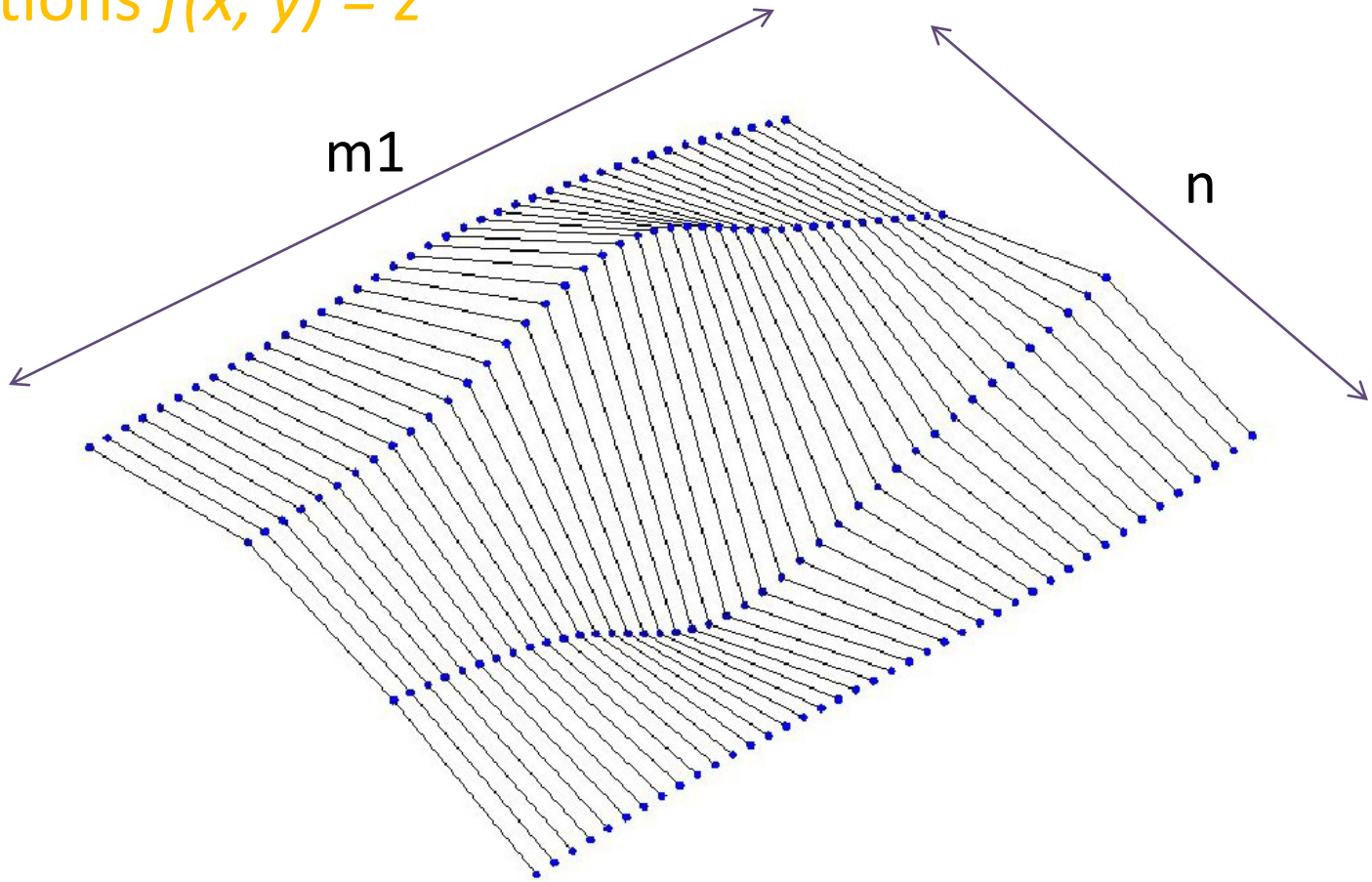


Interpolation after 1st dimension : *n* curves

Here each curve consists of *m1* points

Tensor Product Interpolation

Functions $f(x, y) = z$

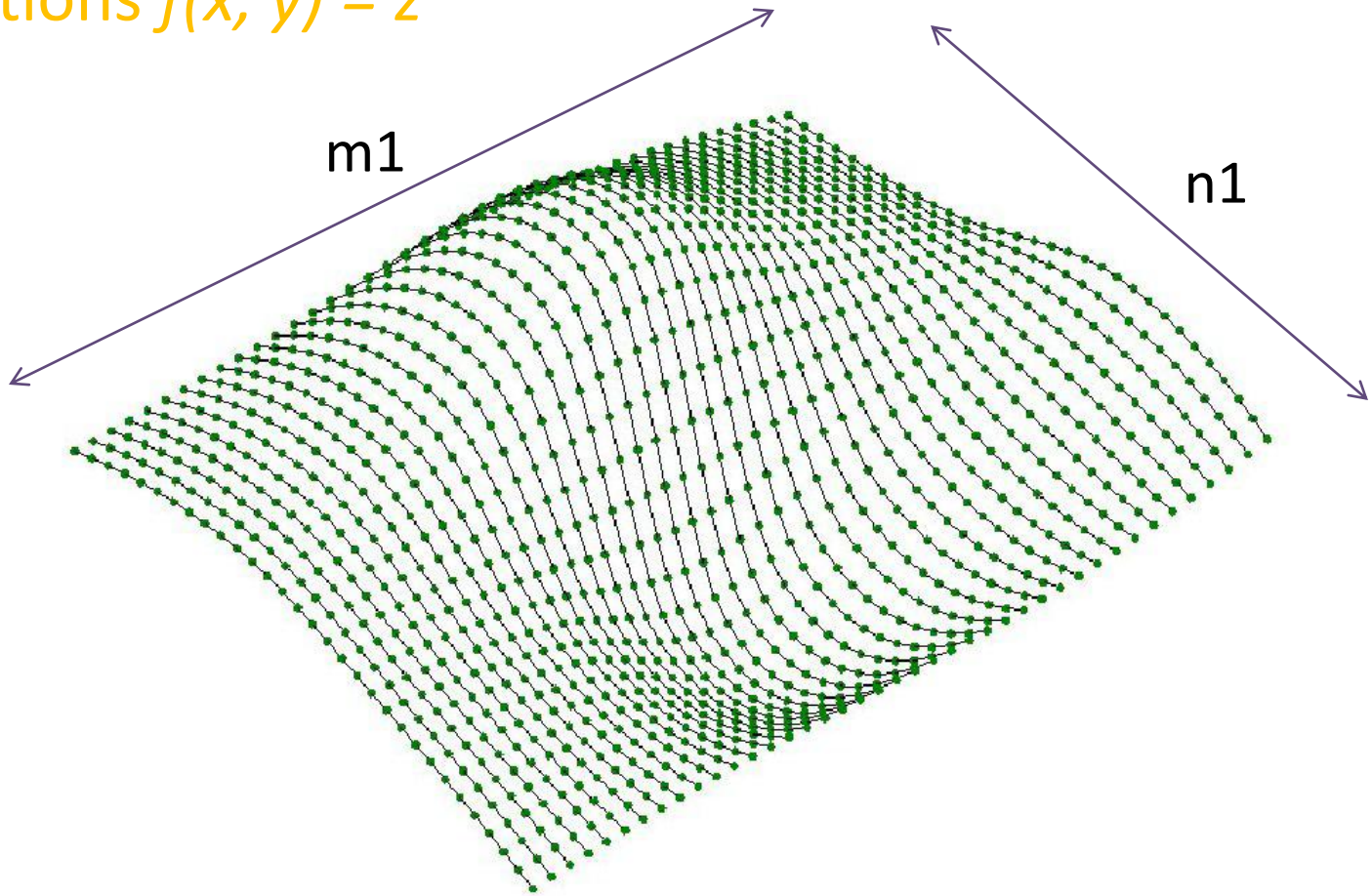


Interpolation along 2nd dimension : interpolation of *m1* curves

Each curve C_j interpolates n points $P_{i,j}$, $1 \leq j \leq n$

Tensor Product Interpolation

Functions $f(x, y) = z$

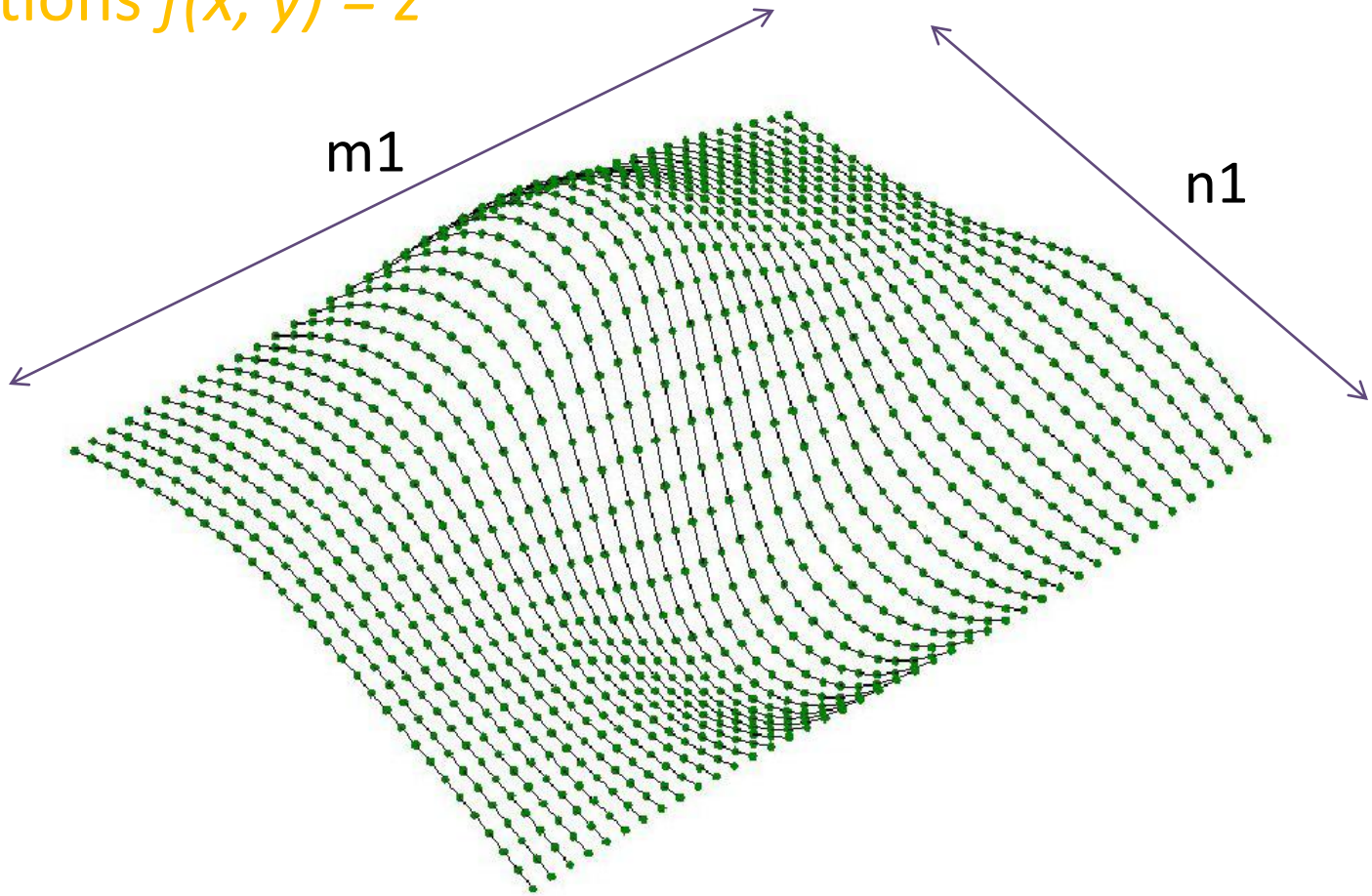


Interpolation along 2nd dimension : *m1* curves

Each curve C_j consists of n points

Tensor Product Interpolation

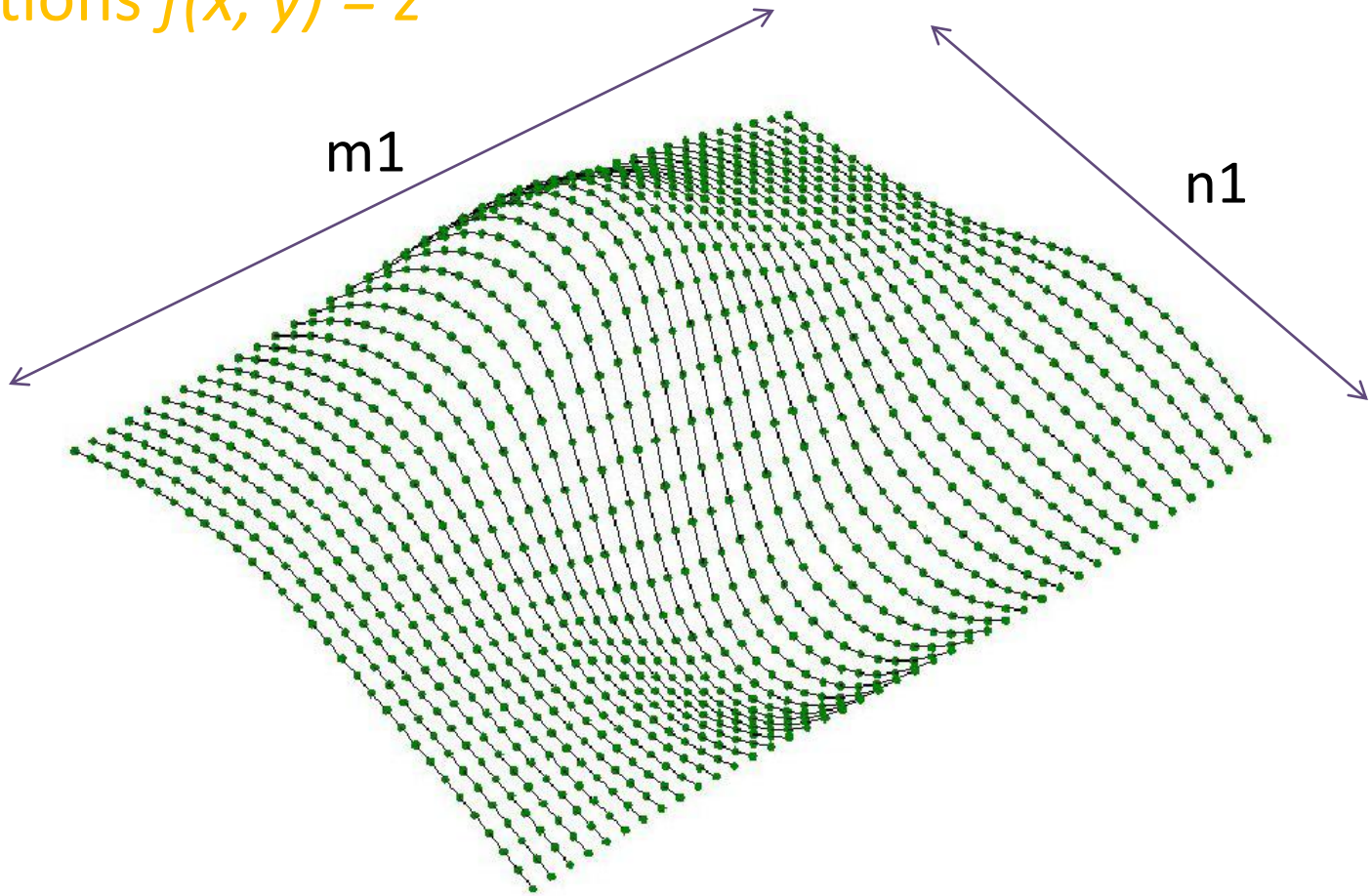
Functions *f(x, y) = z*



Quadrangular mesh of dimension *m1* x *n1* points

Tensor Product Interpolation

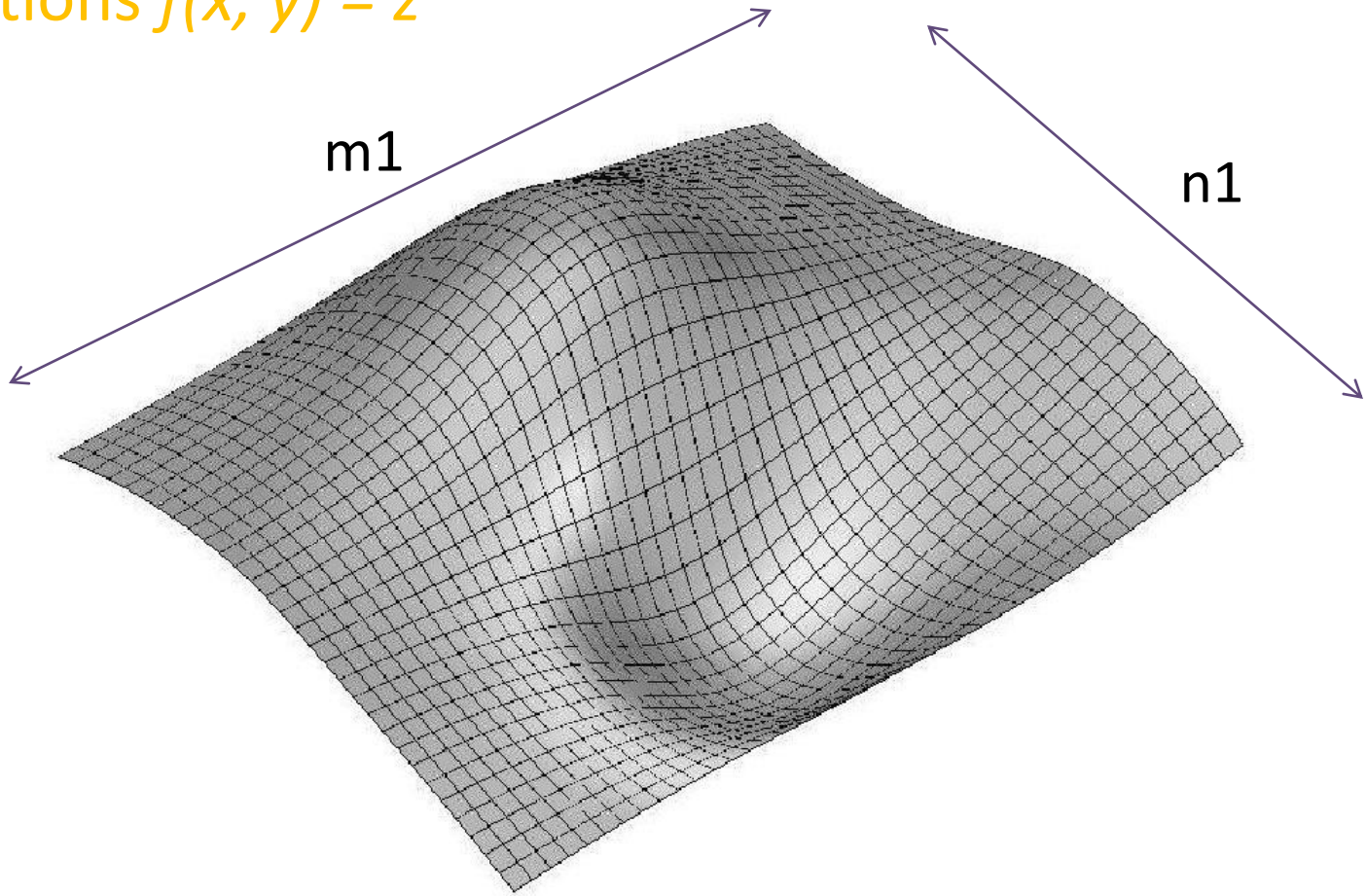
Functions *f(x, y) = z*



Resultant Surface Tensor Product $\mathcal{S}$

Tensor Product Interpolation

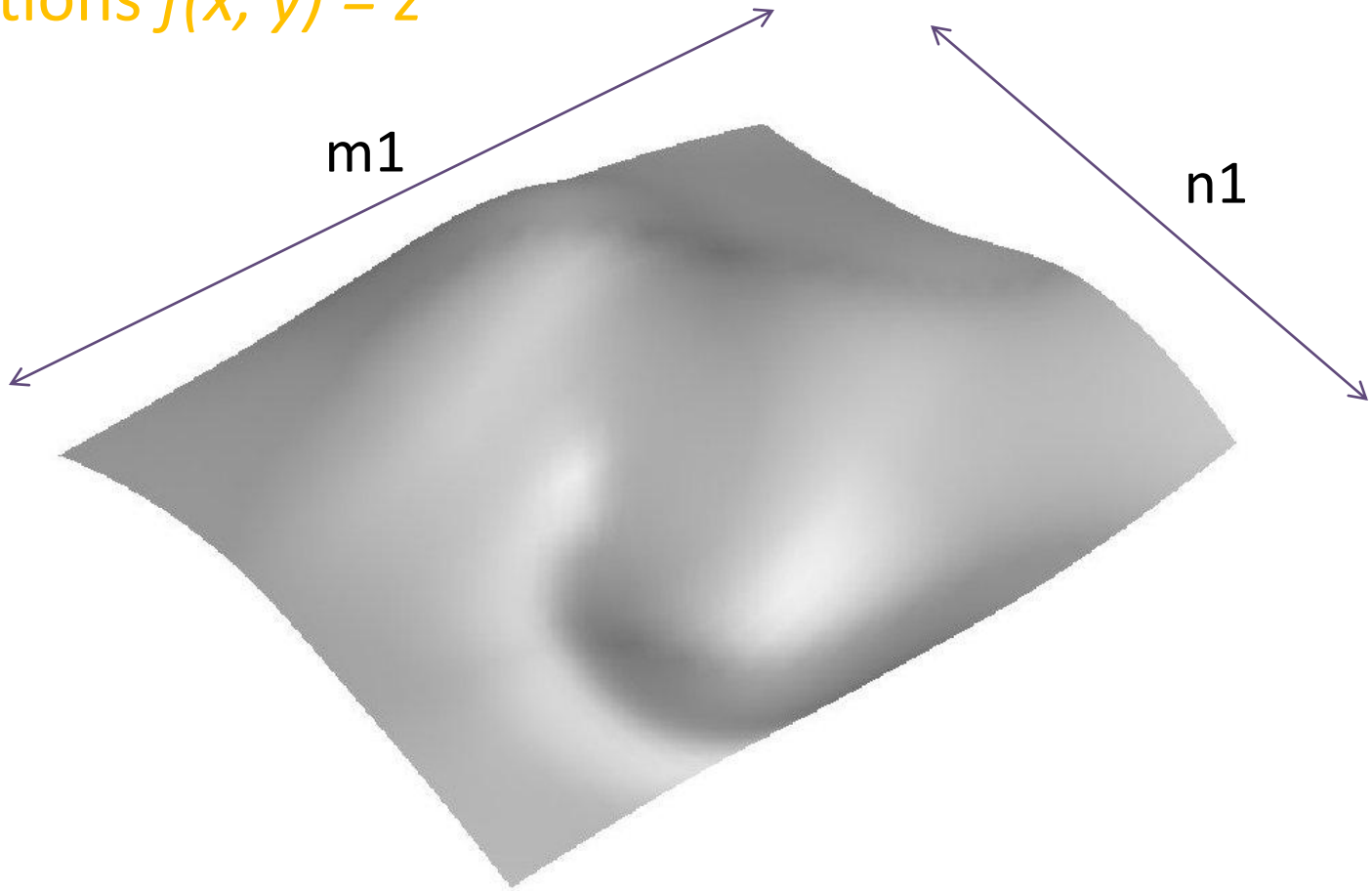
Functions $f(x, y) = z$



Resultant Surface Tensor Product $\mathcal{S}$

Tensor Product Interpolation

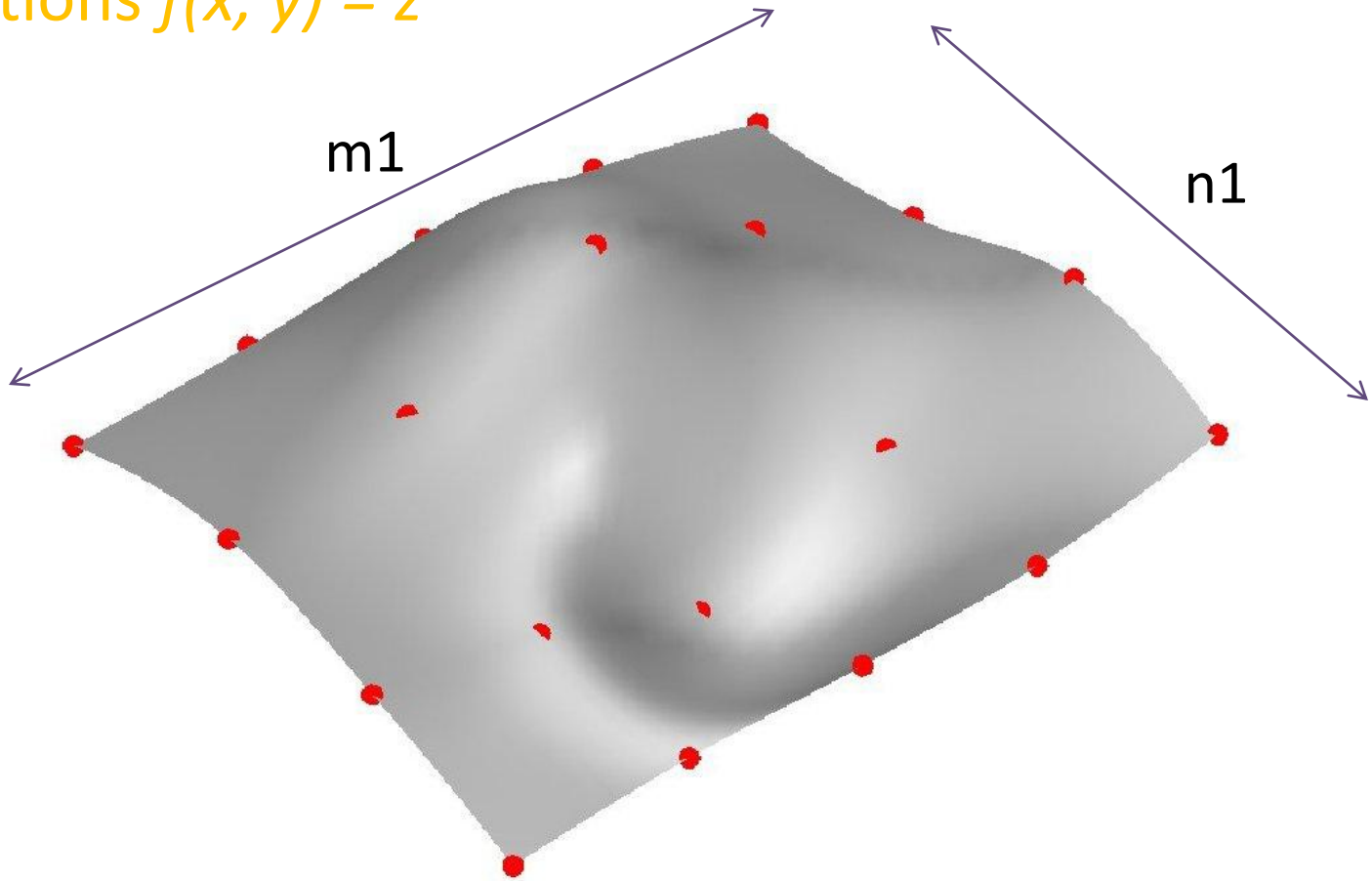
Functions $f(x, y) = z$



Resultant Surface Tensor Product $\mathcal{S}$

Tensor Product Interpolation

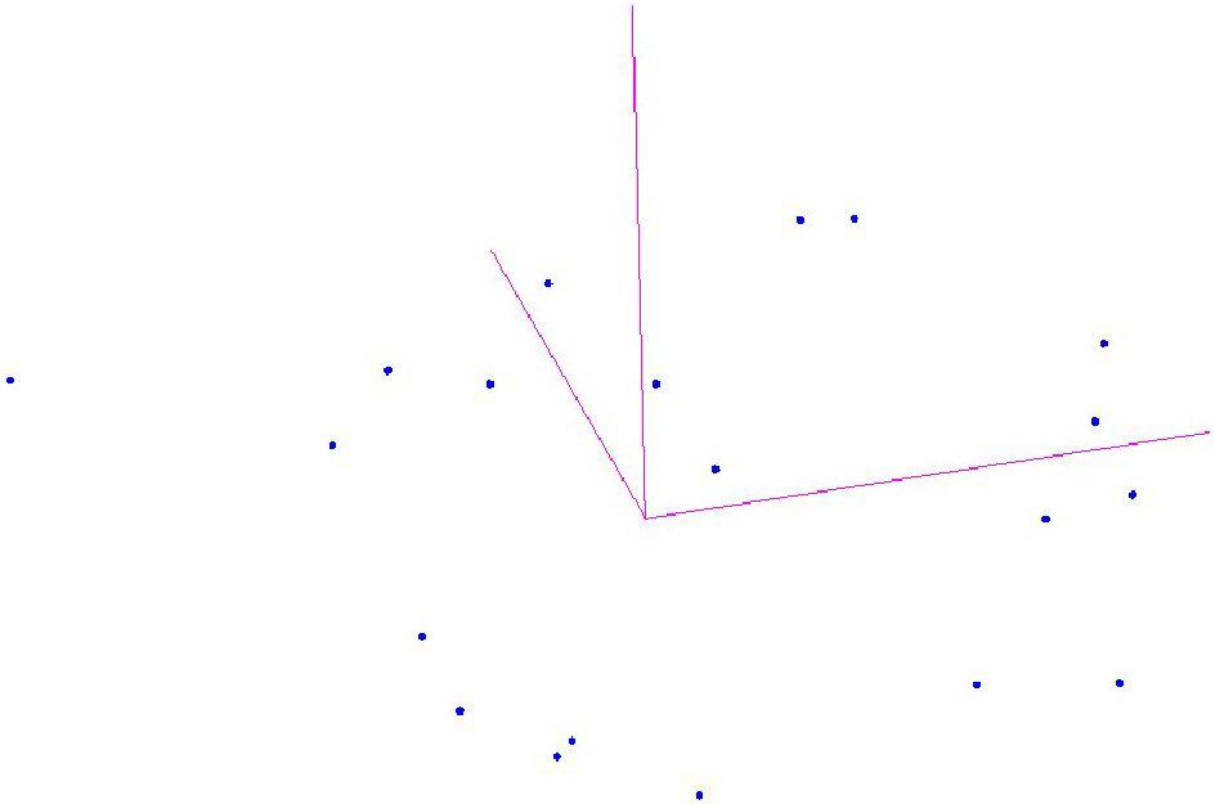
Functions $f(x, y) = z$



Resultant Surface Tensor Product $\mathcal{S}$ + Initial data

Spline Interpolation

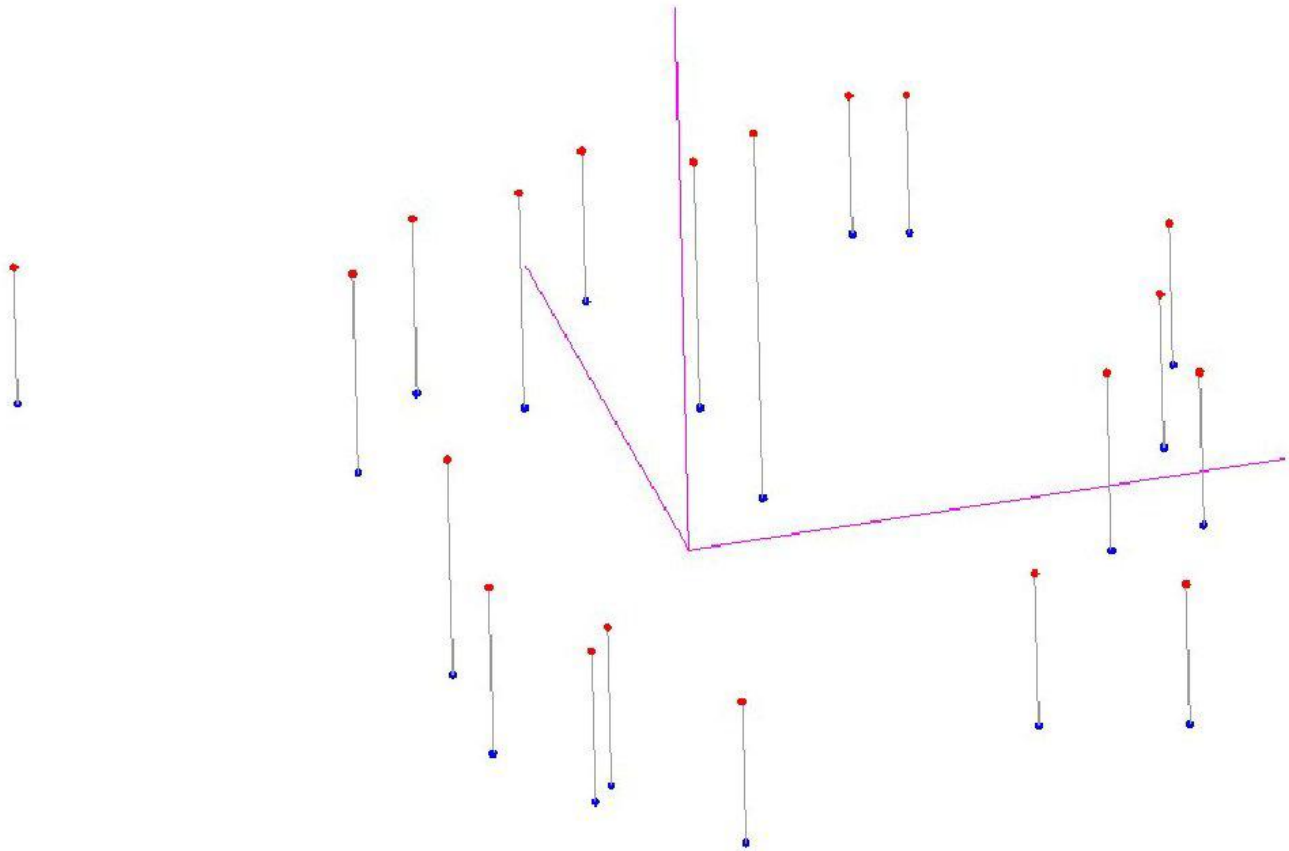
Functions $f(x, y) = z$



Point Setting (u_i, v_i)

Spline Interpolation

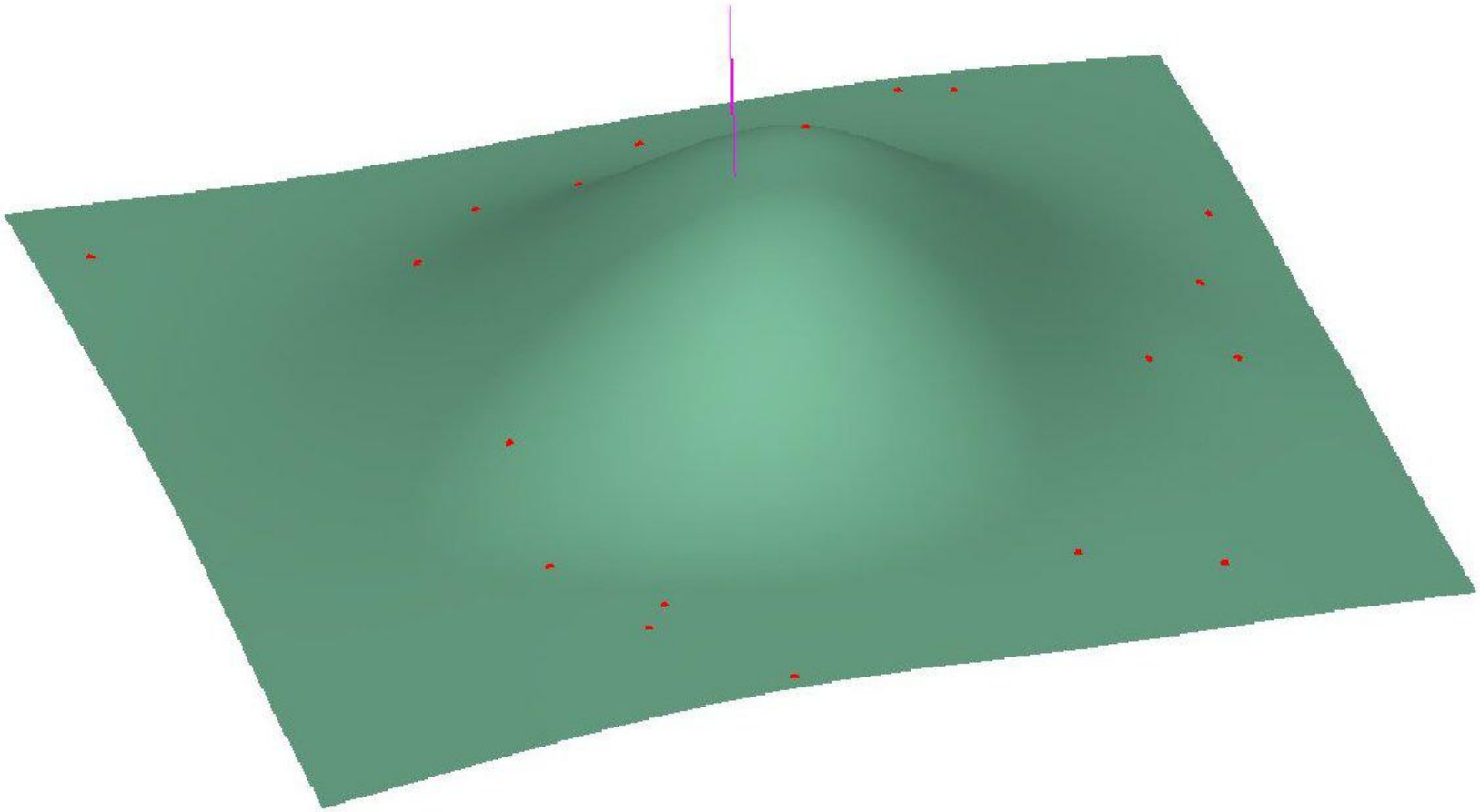
Functions $f(x, y) = z$



Data Points $(x_i = u_i, y_i = v_i, z_i)$

Spline Interpolation

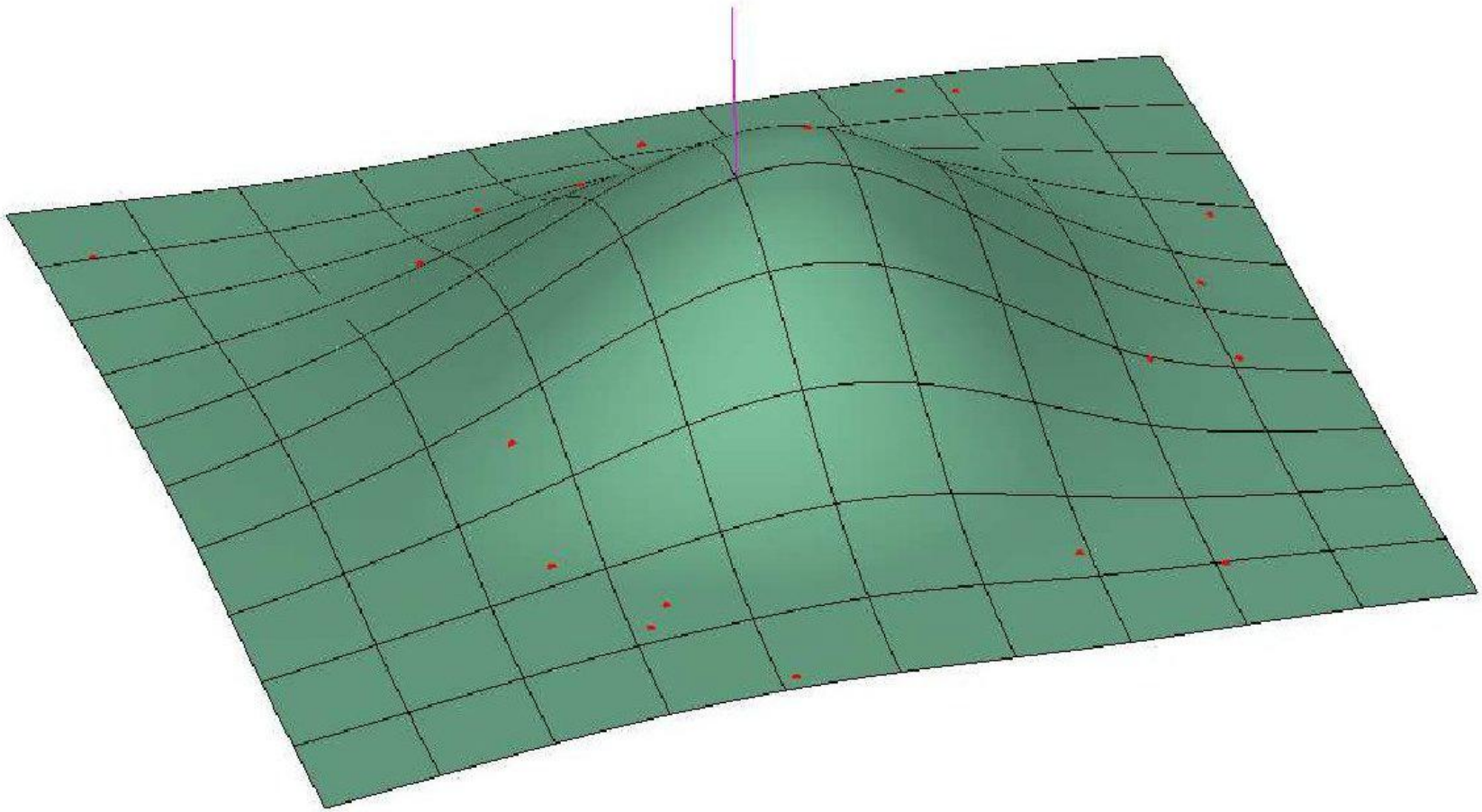
Functions $f(x, y) = z$



Cubic Spline Interpolation $h(r) = r^3$

Spline Interpolation

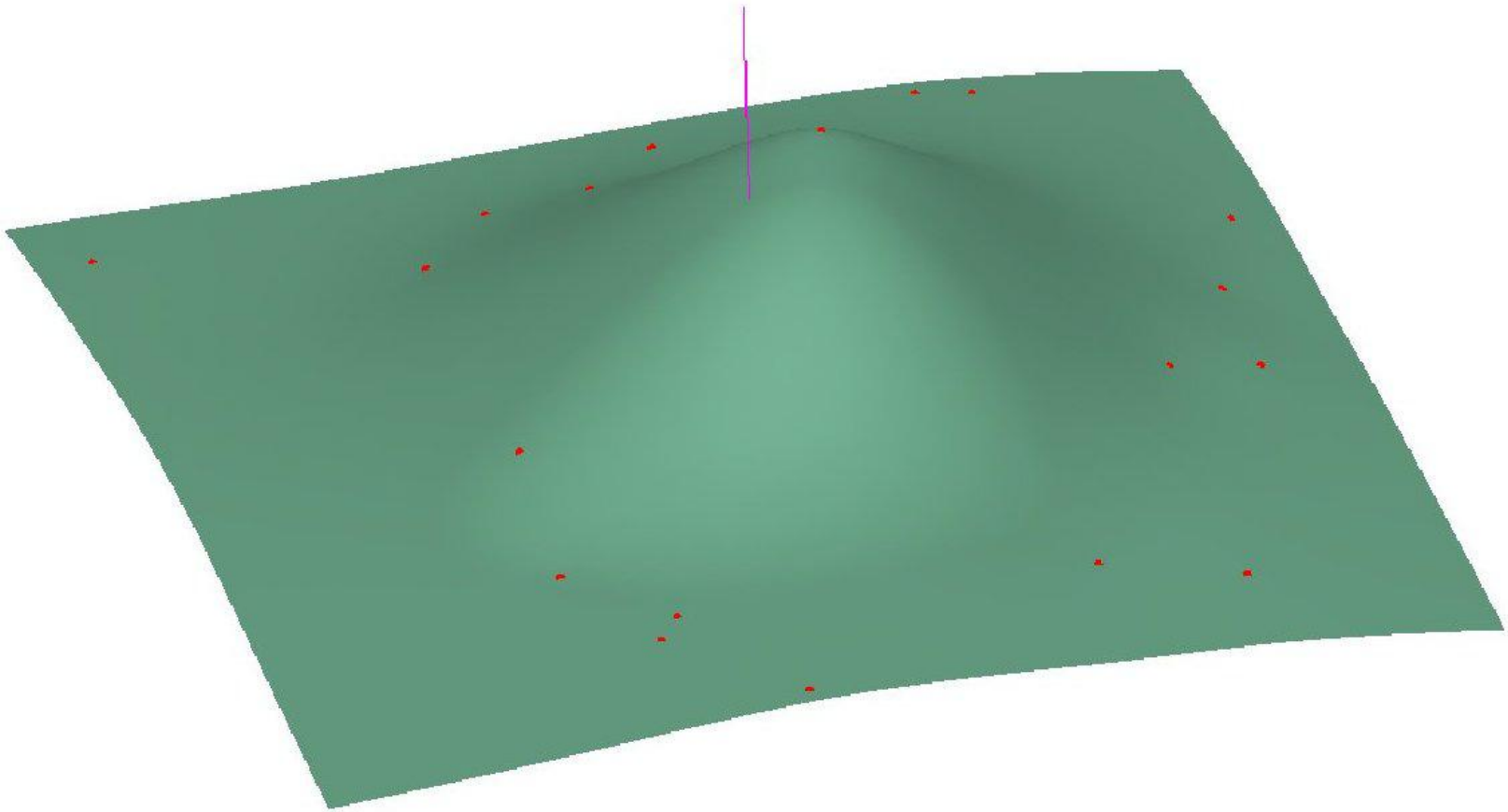
Functions $f(x, y) = z$



Cubic Spline Interpolation $h(r) = r^3$

Spline Interpolation

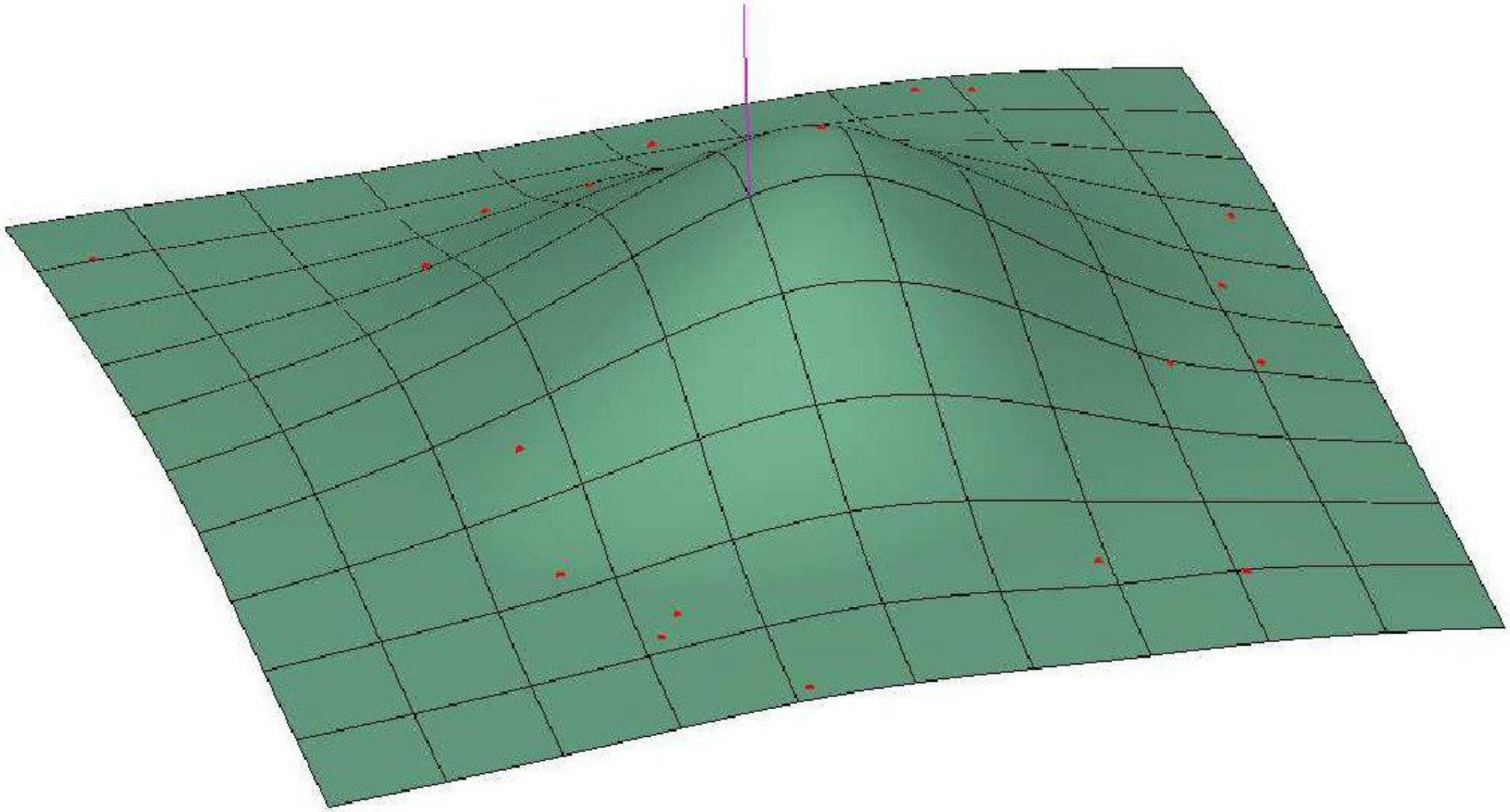
Functions $f(x, y) = z$



Cubic Spline Interpolation $h(r) = r^2 \log(r)$

Spline Interpolation

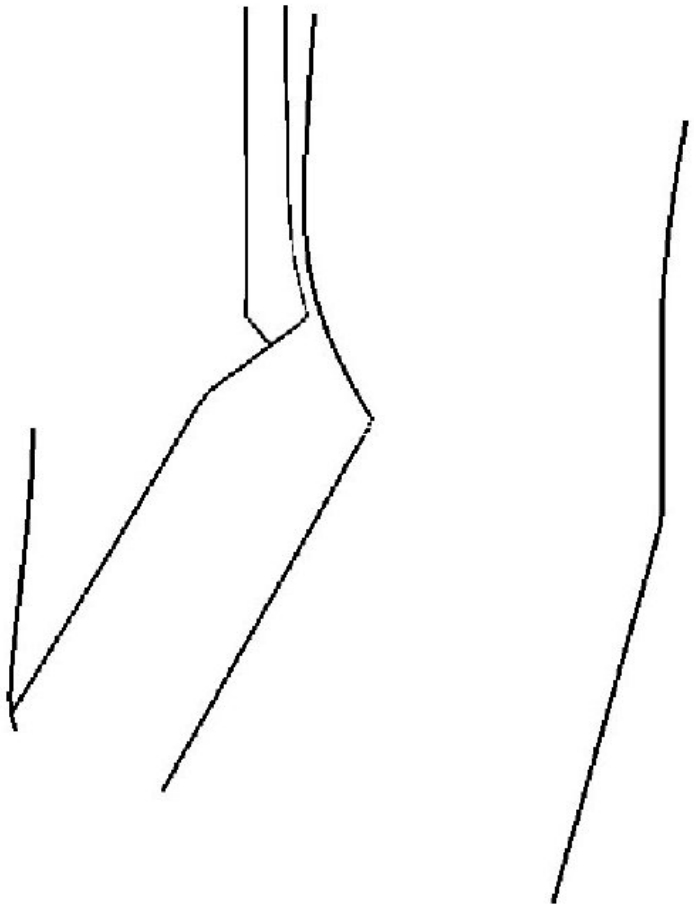
Functions $f(x, y) = z$



Cubic Spline Interpolation $h(r) = r^2 \log(r)$

Surface Filling

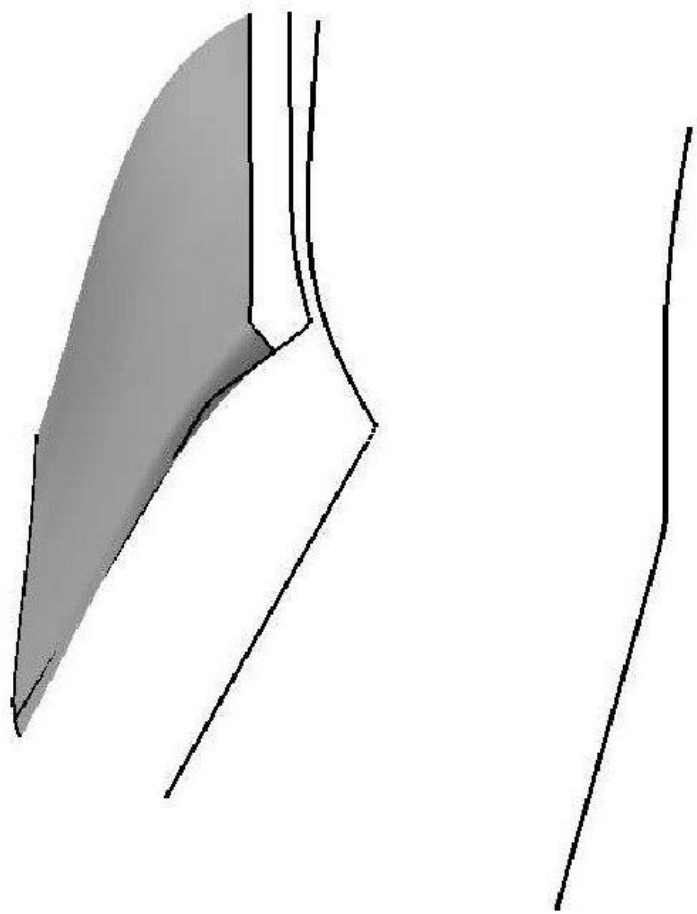
Border Curves



Border Curves

Surface Filling

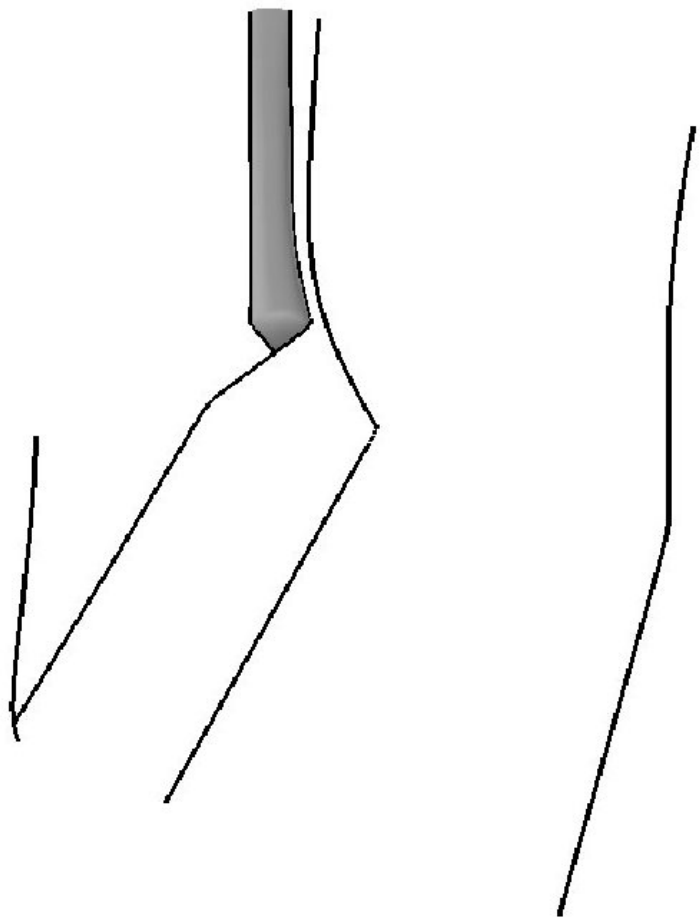
Border Curves



Interpolating Surface : C_1 and C_2

Surface Filling

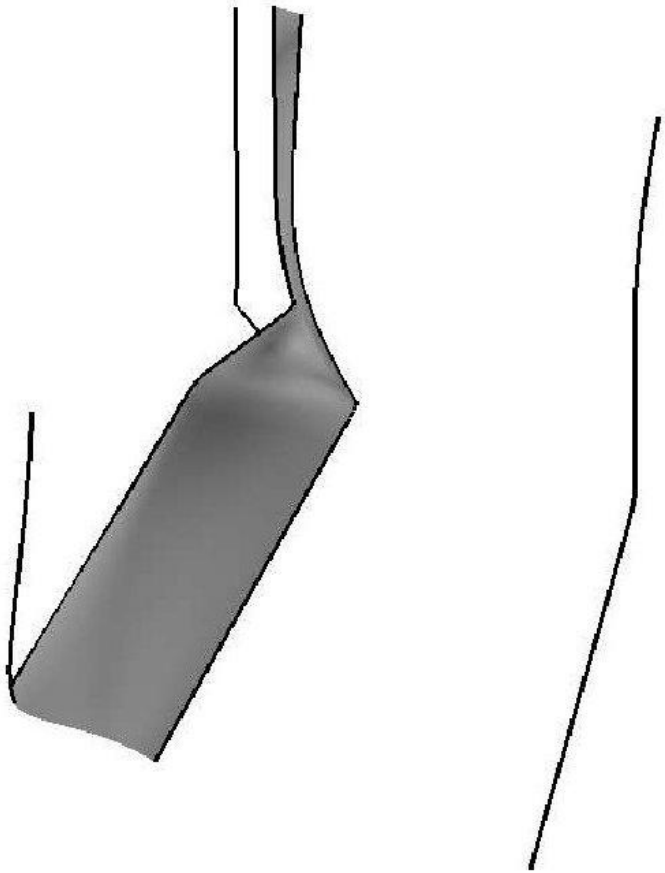
Border Curves



Interpolating Surface : C_2 and C_3

Surface Filling

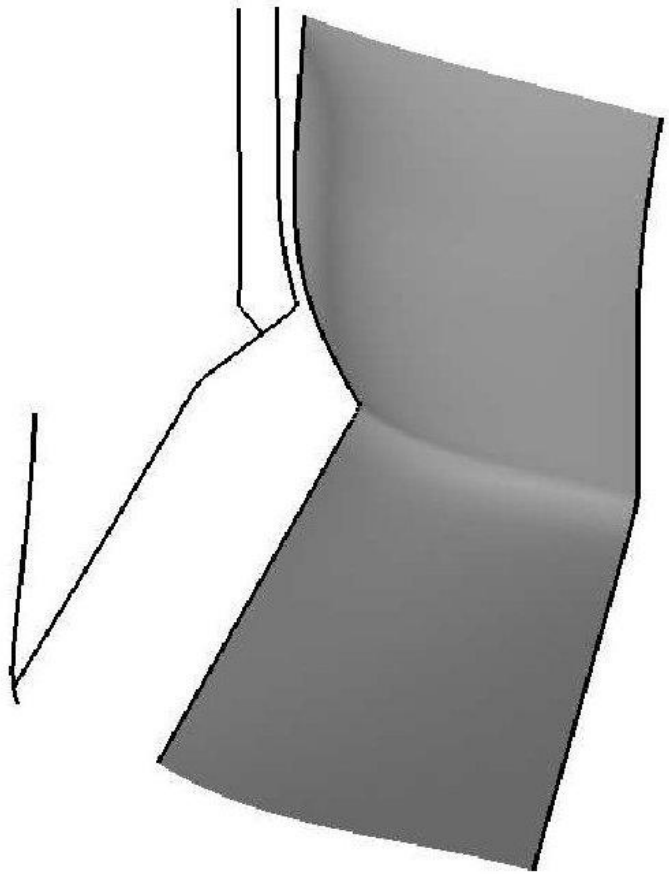
Border Curves



Interpolating Surface : C_3 and C_4

Surface Filling

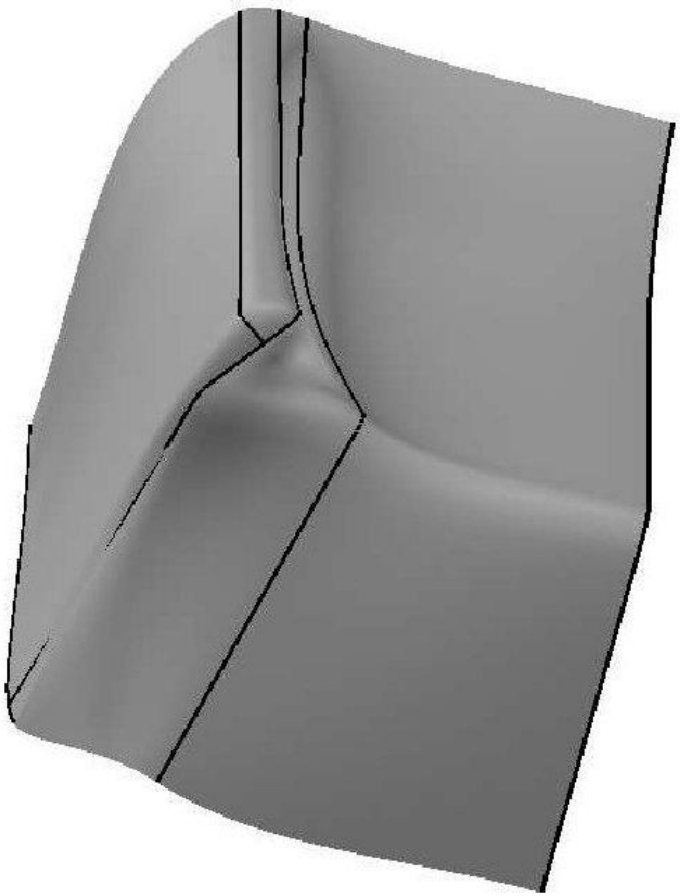
Border Curves



Interpolating Surface : C_4 and C_5

Surface Filling

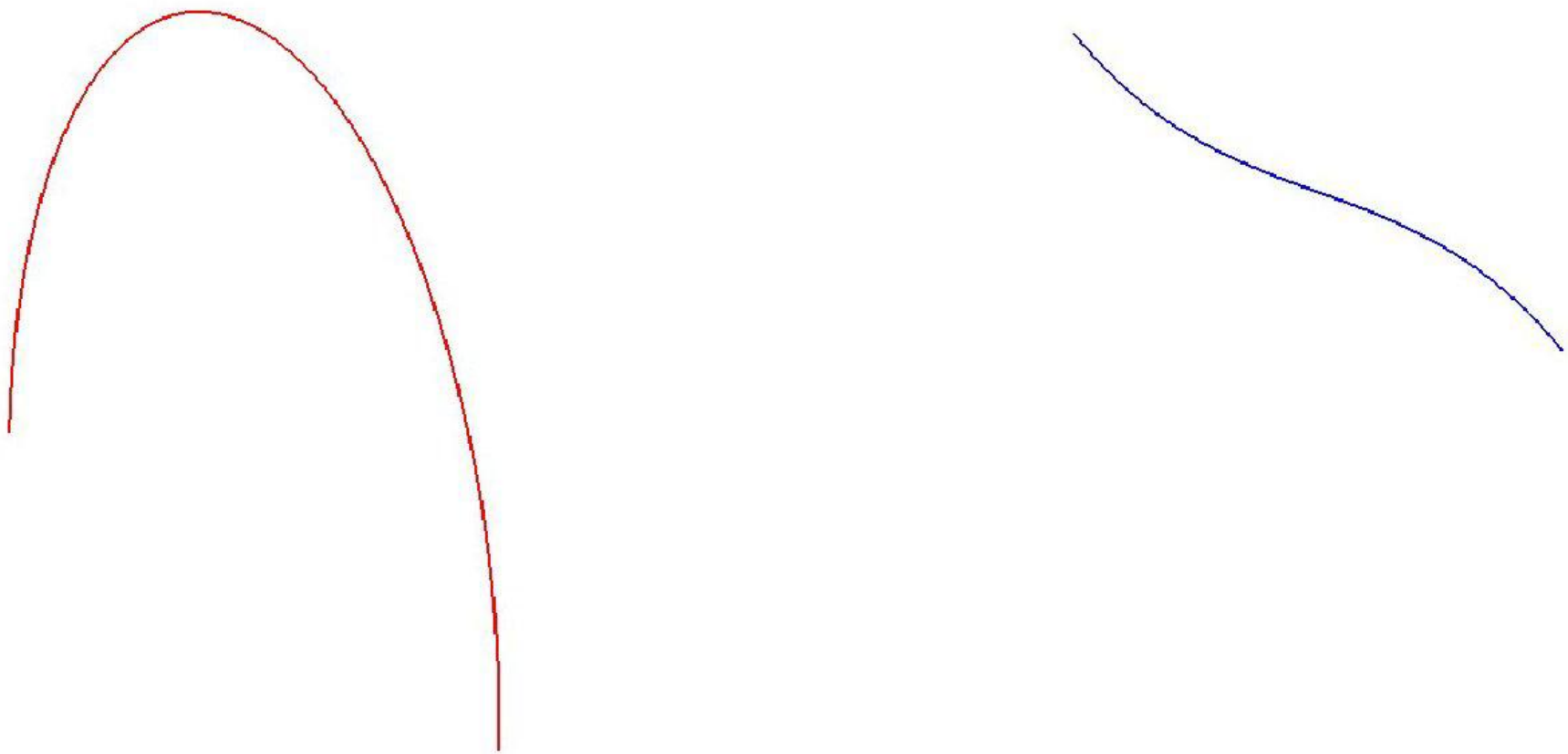
Border Curves



Resultant Surface

Interpolation between 2 Curves

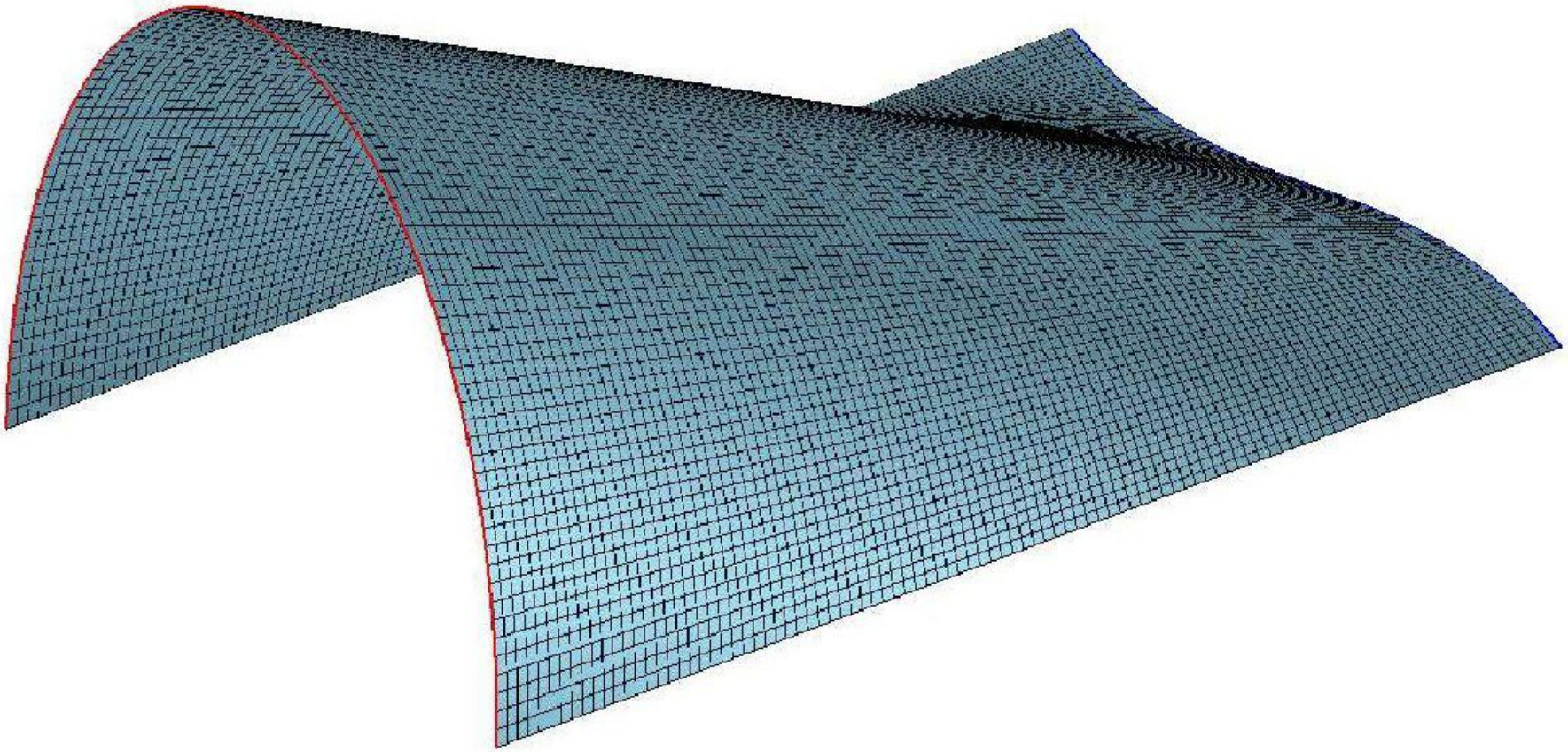
Linear Process



Curves $C_1(u)$ and $C_2(u)$

Interpolation between 2 Curves

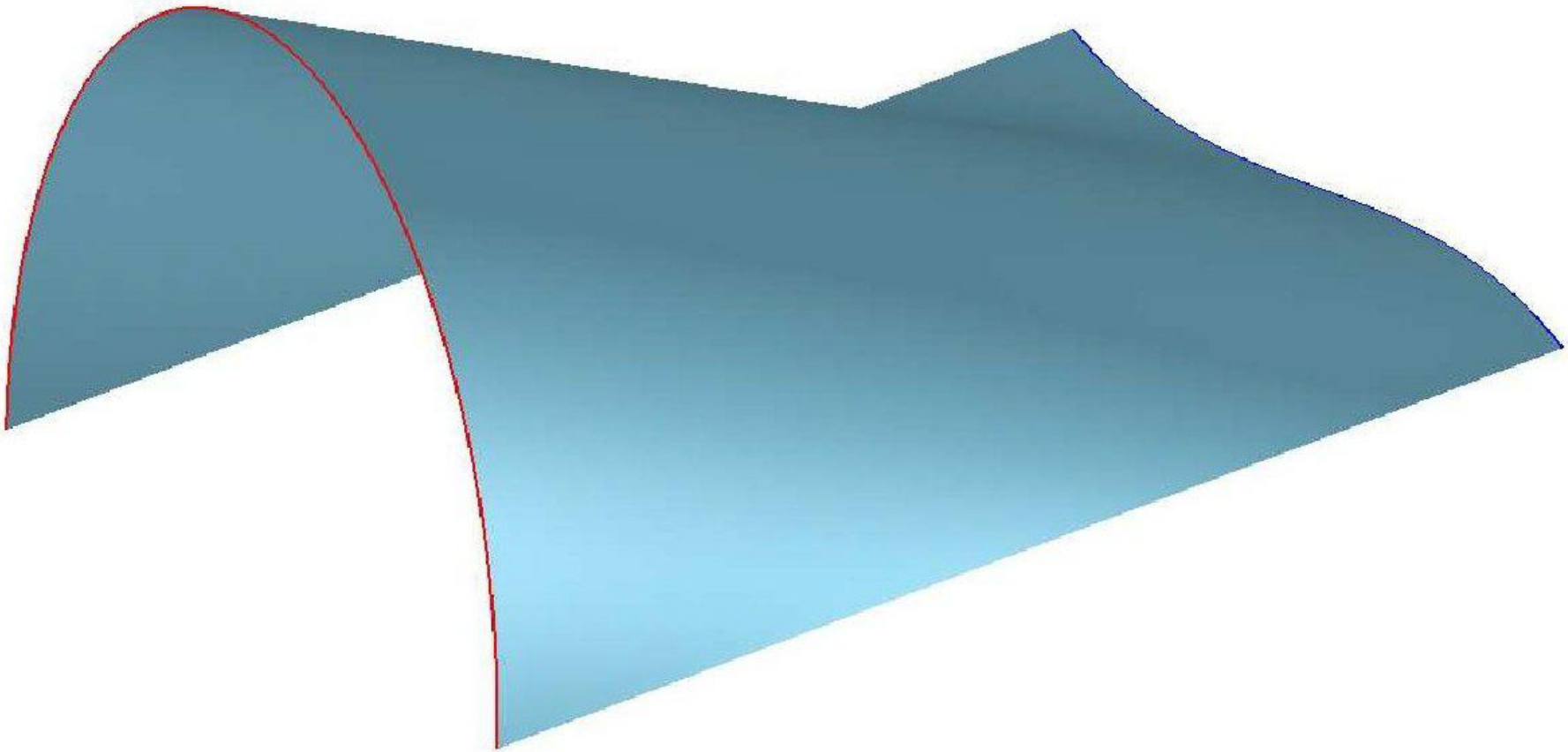
Linear Process



Surface S = $(1 - v) C_1(u) + v C_2(u)$, where $v \in [0, 1]$

Interpolation between 2 Curves

Linear Process



Surface S = $(1 - v) C_1(u) + v C_2(u)$, where $v \in [0, 1]$

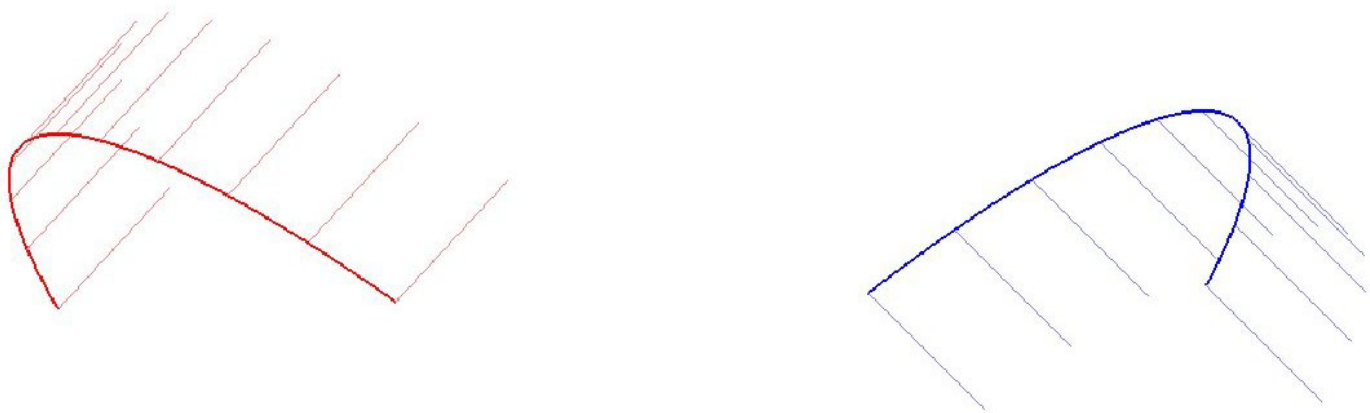
Interpolation between 2 Curves

Hermite Process

Curves $C_1(u)$ and $C_2(u)$

Interpolation between 2 Curves

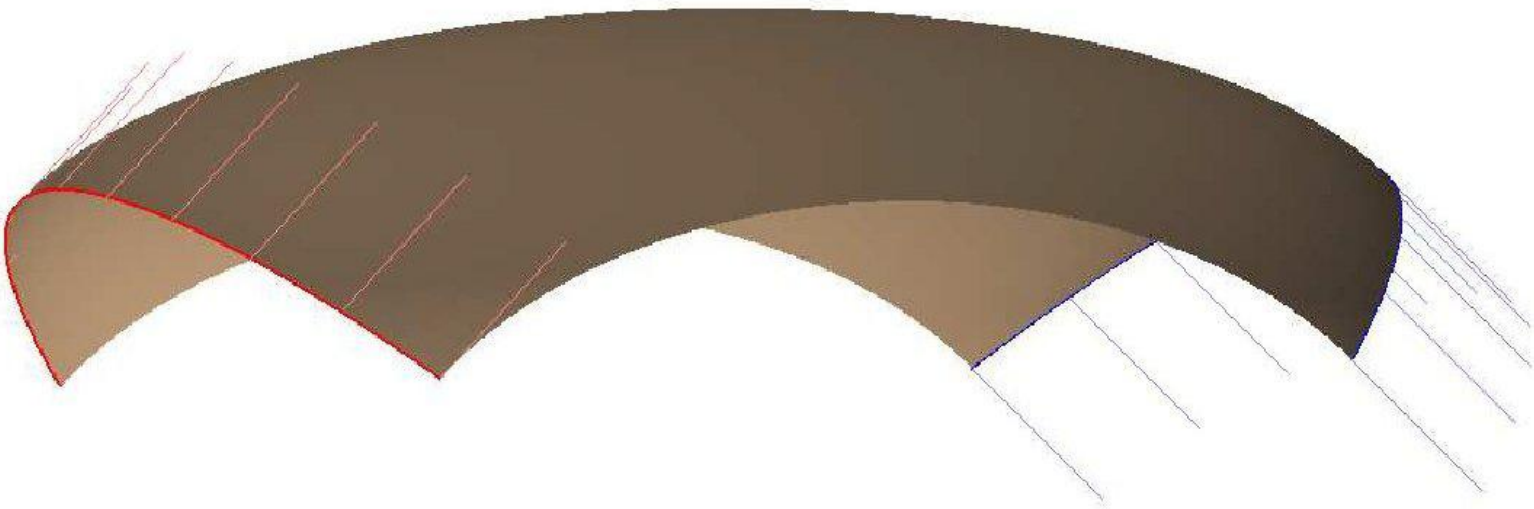
Hermite Process



Curves $C_1(u)$ and $C_2(u)$ + derived tangents $T_1(u)$ and $T_2(u)$

Interpolation between 2 Curves

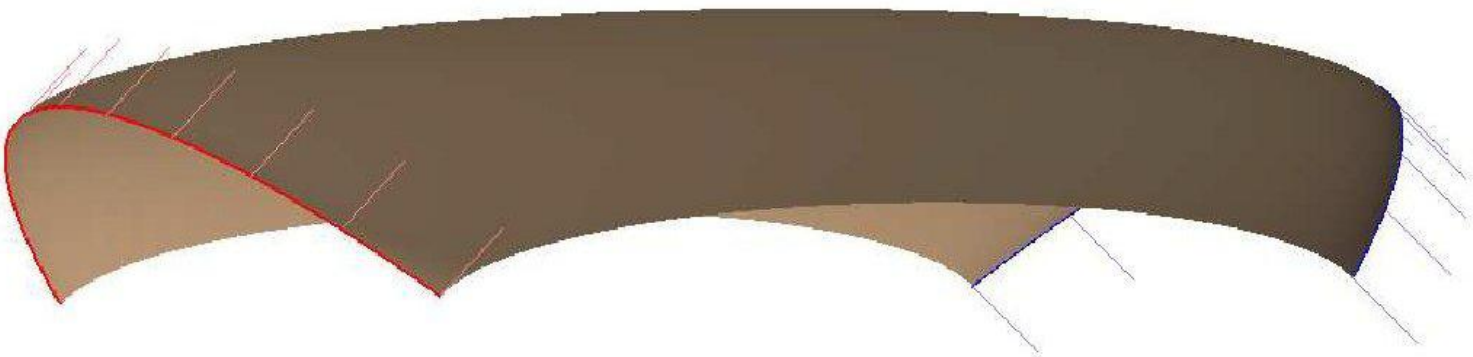
Hermite Process



Curves $C_1(u)$ and $C_2(u)$ + derived tangents $T_1(u)$ and $T_2(u)$
Same principle/formula for the interpolation of points

Interpolation between 2 Curves

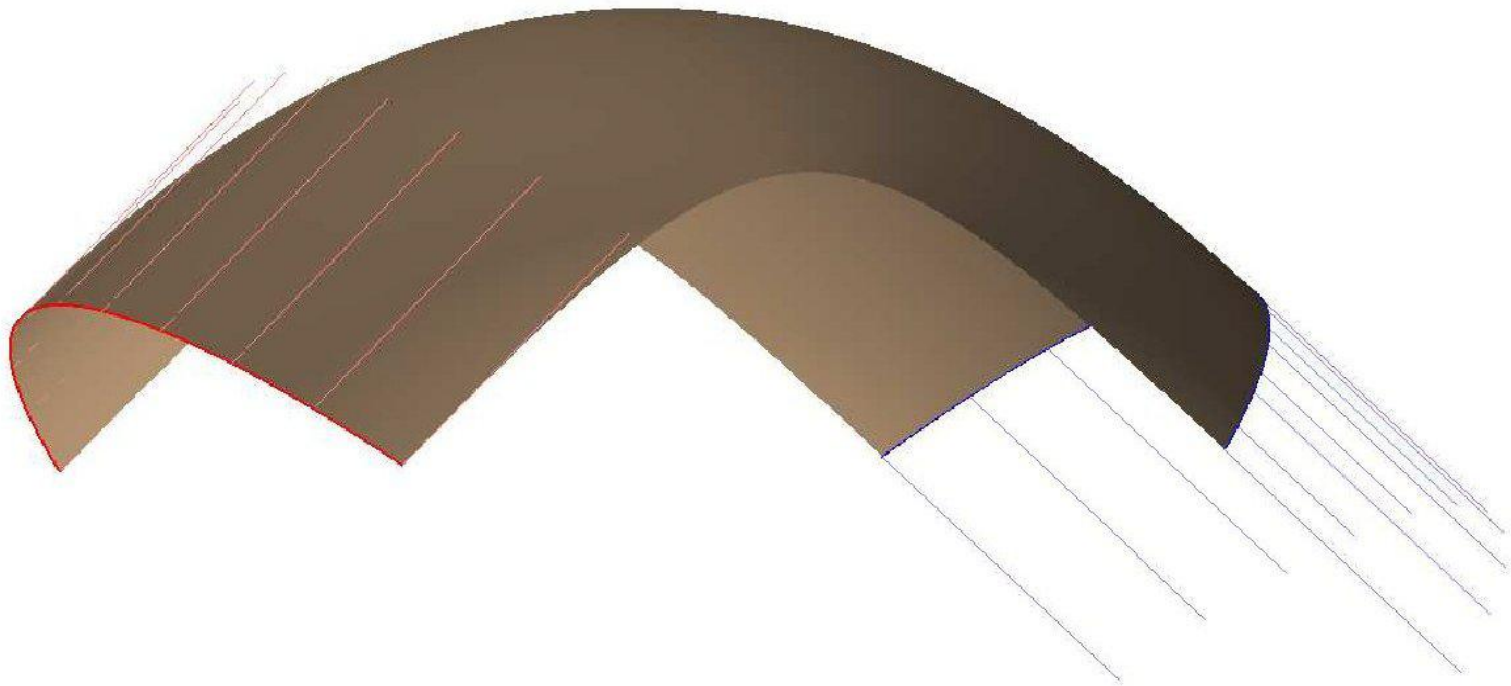
Hermite Process



Curves $C_1(u)$ and $C_2(u)$ + derived tangents $T_1(u)$ and $T_2(u)$
Same principle/formula for the interpolation of points

Interpolation between 2 Curves

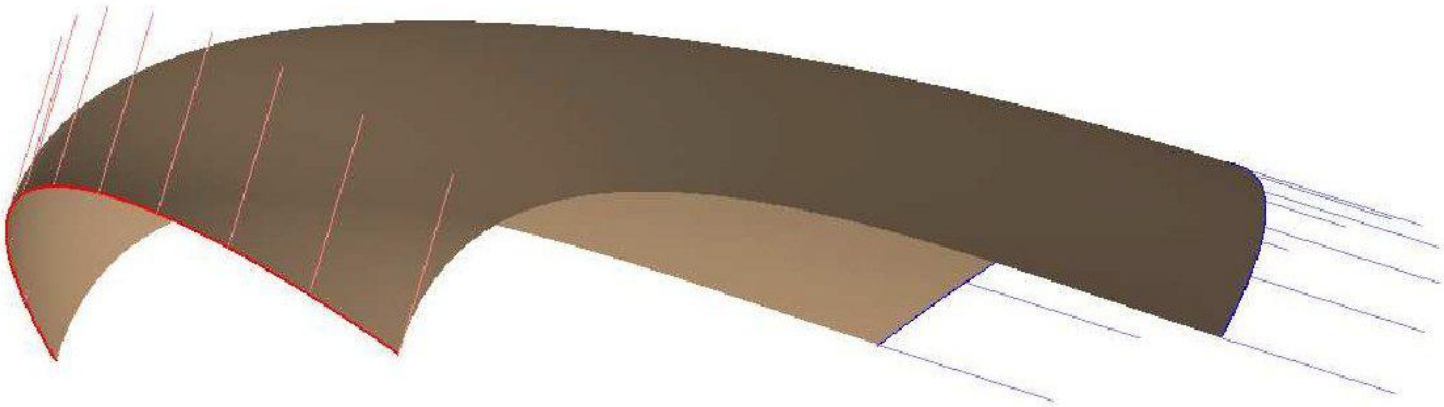
Hermite Process



Curves $C_1(u)$ and $C_2(u)$ + derived tangents $T_1(u)$ and $T_2(u)$
Same principle/formula for the interpolation of points

Interpolation between 2 Curves

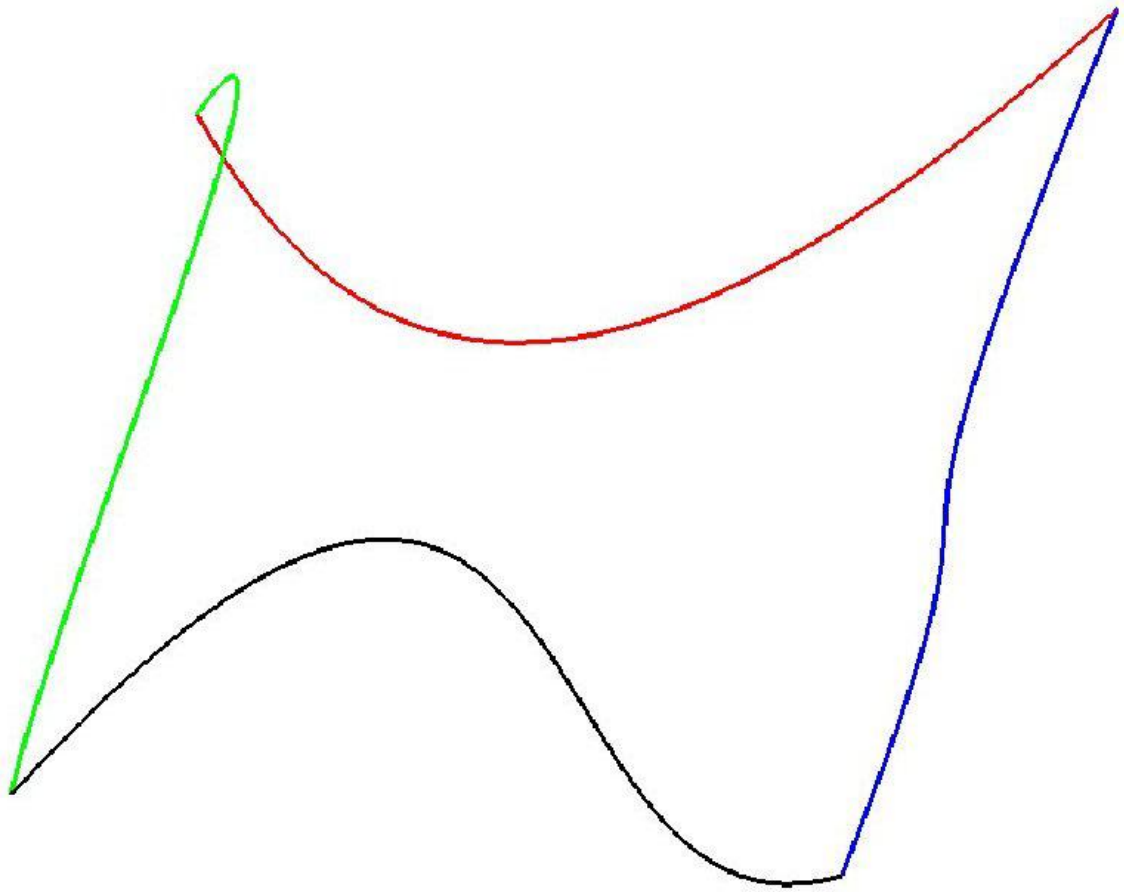
Hermite Process



Curves $C_1(u)$ and $C_2(u)$ + derived tangents $T_1(u)$ and $T_2(u)$
Same principle/formula for the interpolation of points

Interpolation between 4 Curves

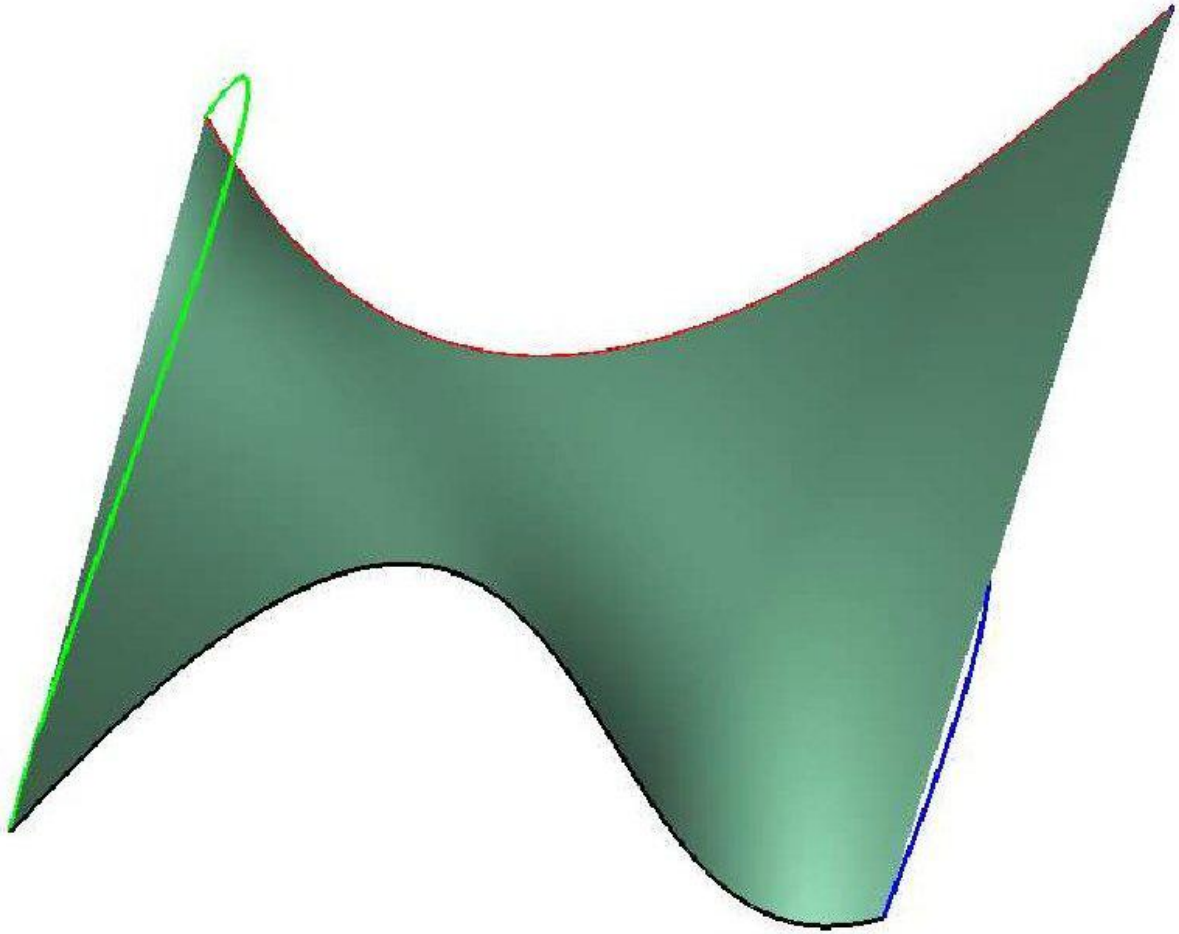
Coon's Process (Linear)



4 boundary curves

Interpolation between 4 Curves

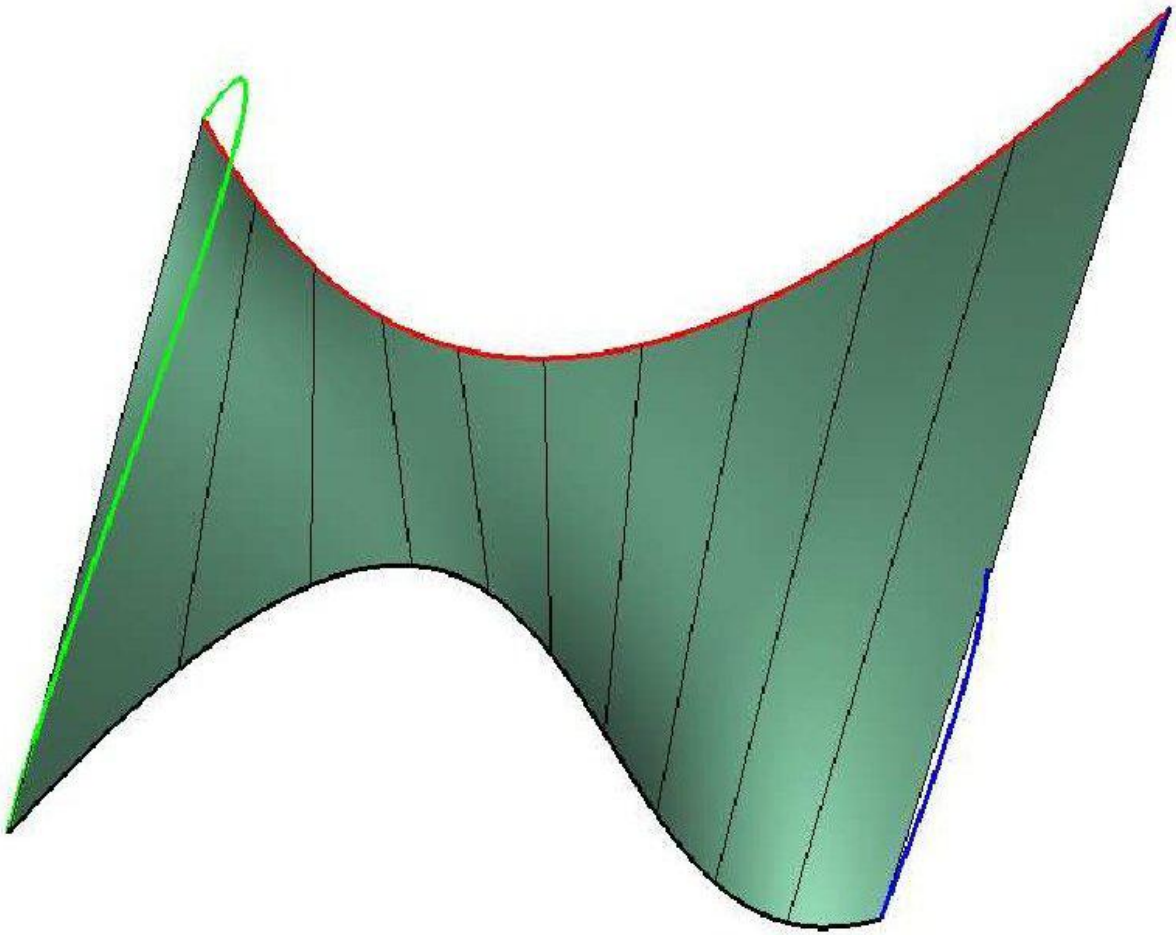
Coon's Process (Linear)



P_u : Interpolation of curves $C_{v=0}$ *and* $C_{v=1}$

Interpolation between 4 Curves

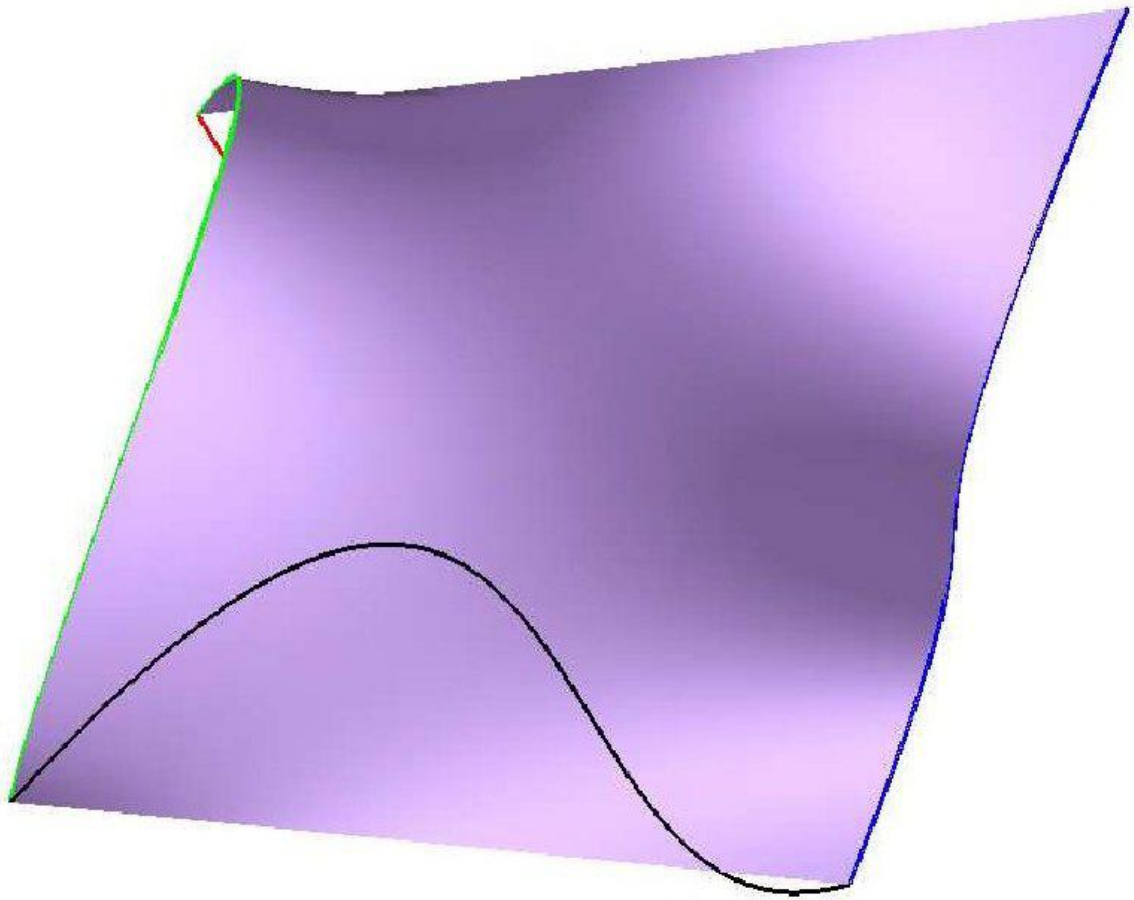
Coon's Process (Linear)



P_u : Interpolation of curves $C_{v=0}$ *and* $C_{v=1}$

Interpolation between 4 Curves

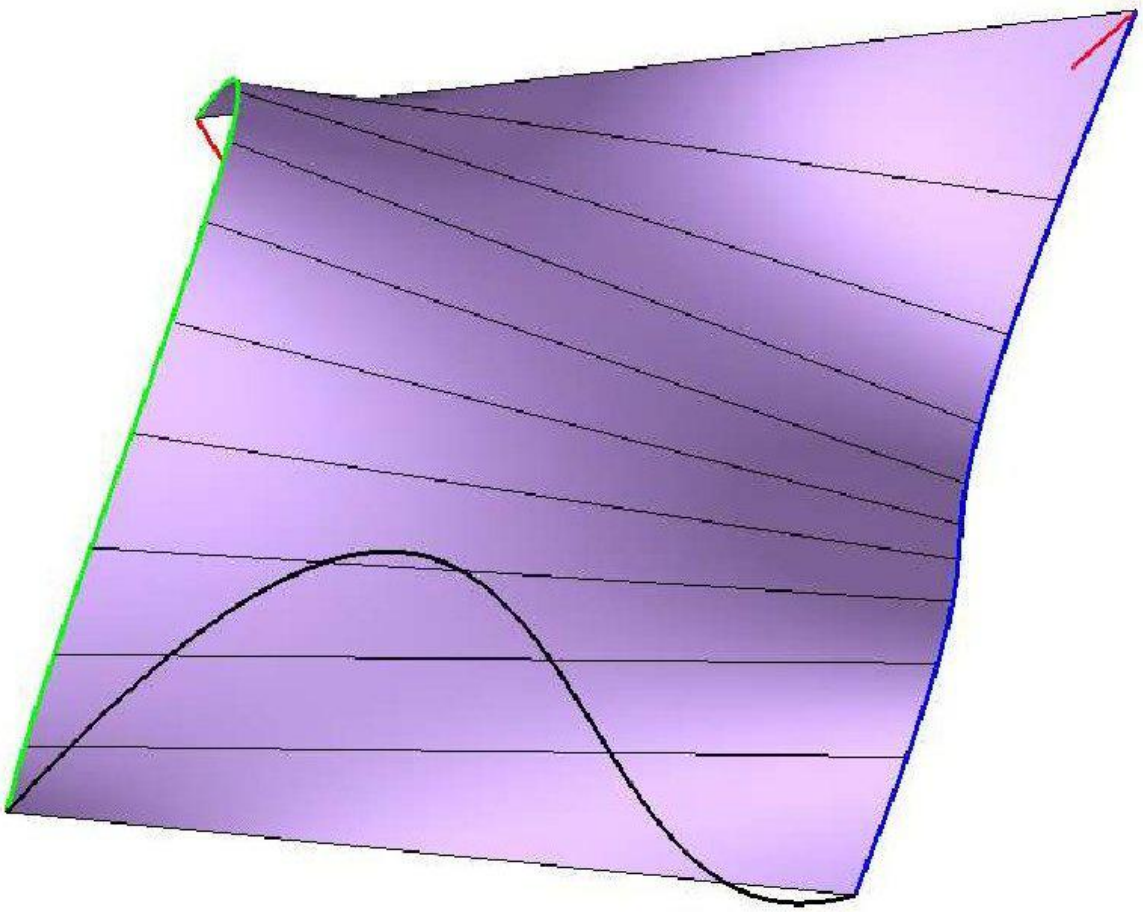
Coon's Process (Linear)



P_v : Interpolation of curves $C_{u=0}$ *and* $C_{u=1}$

Interpolation between 4 Curves

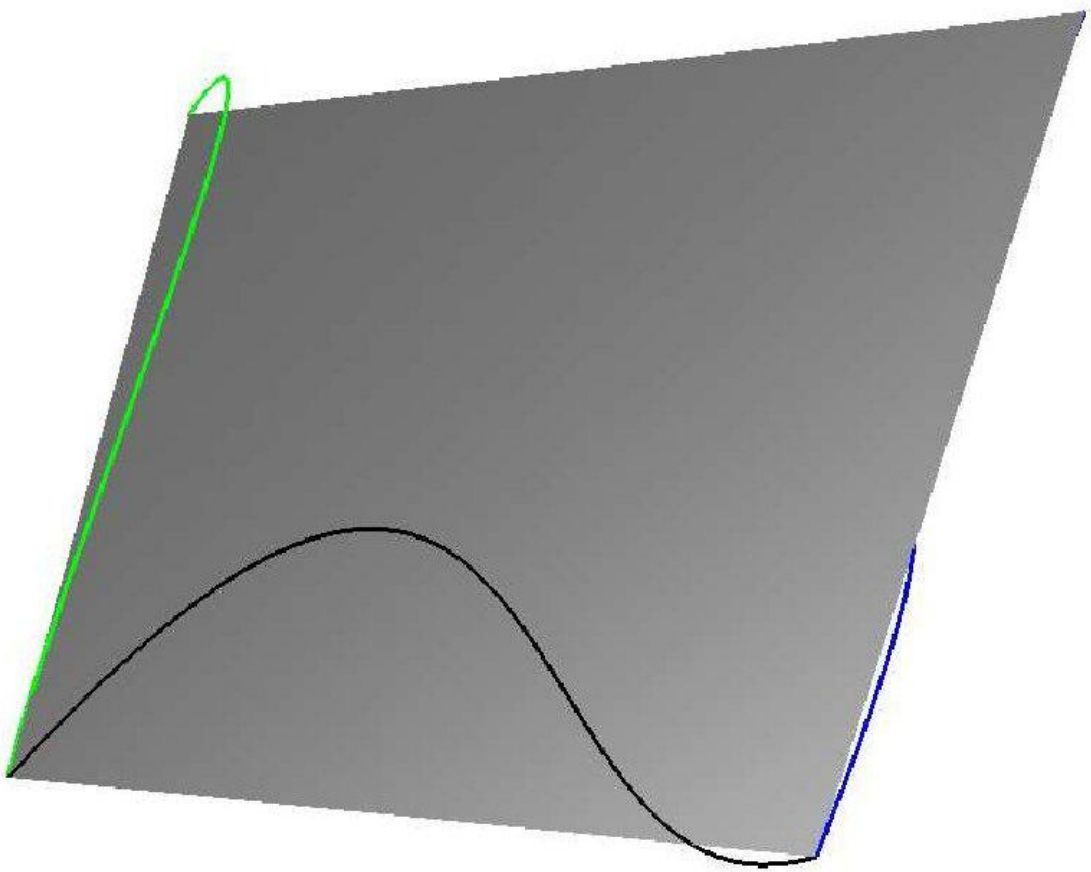
Coon's Process (Linear)



P_v : Interpolation of curves $C_{u=0}$ *and* $C_{u=1}$

Interpolation between 4 Curves

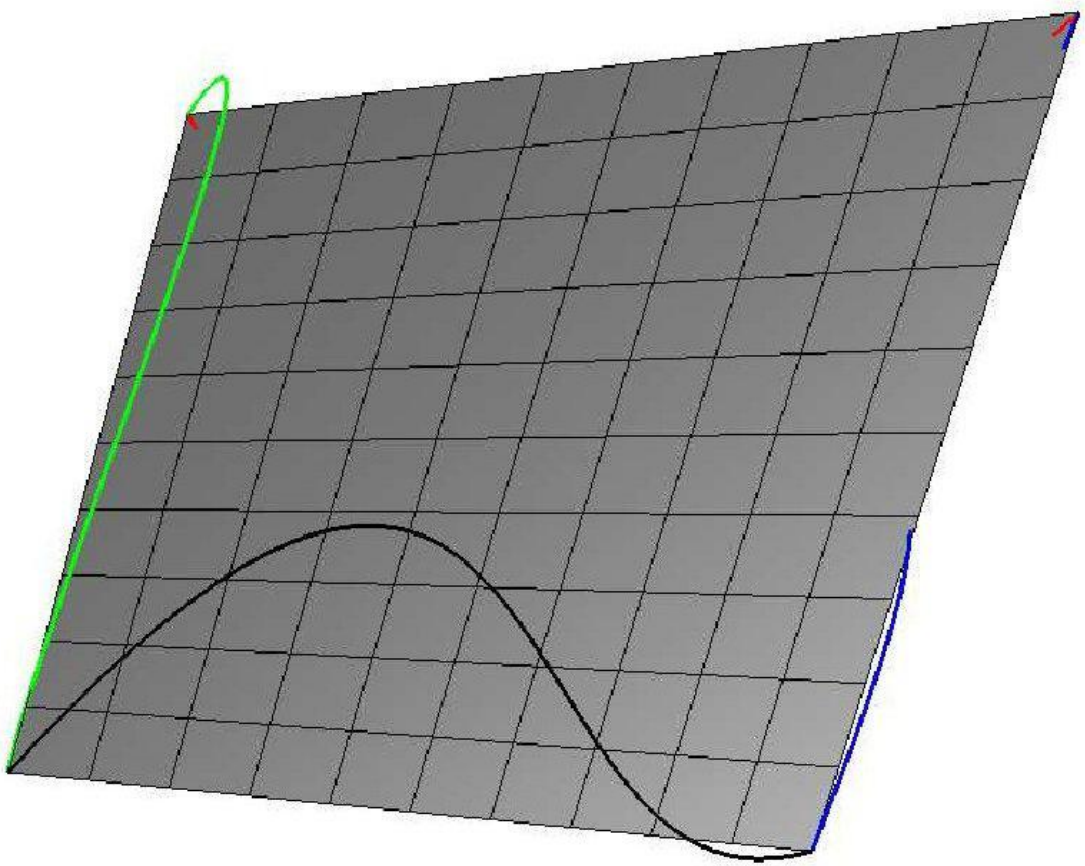
Coon's Process (Linear)



$P_u P_v$: Interpolation of the vertices of the patch

Interpolation between 4 Curves

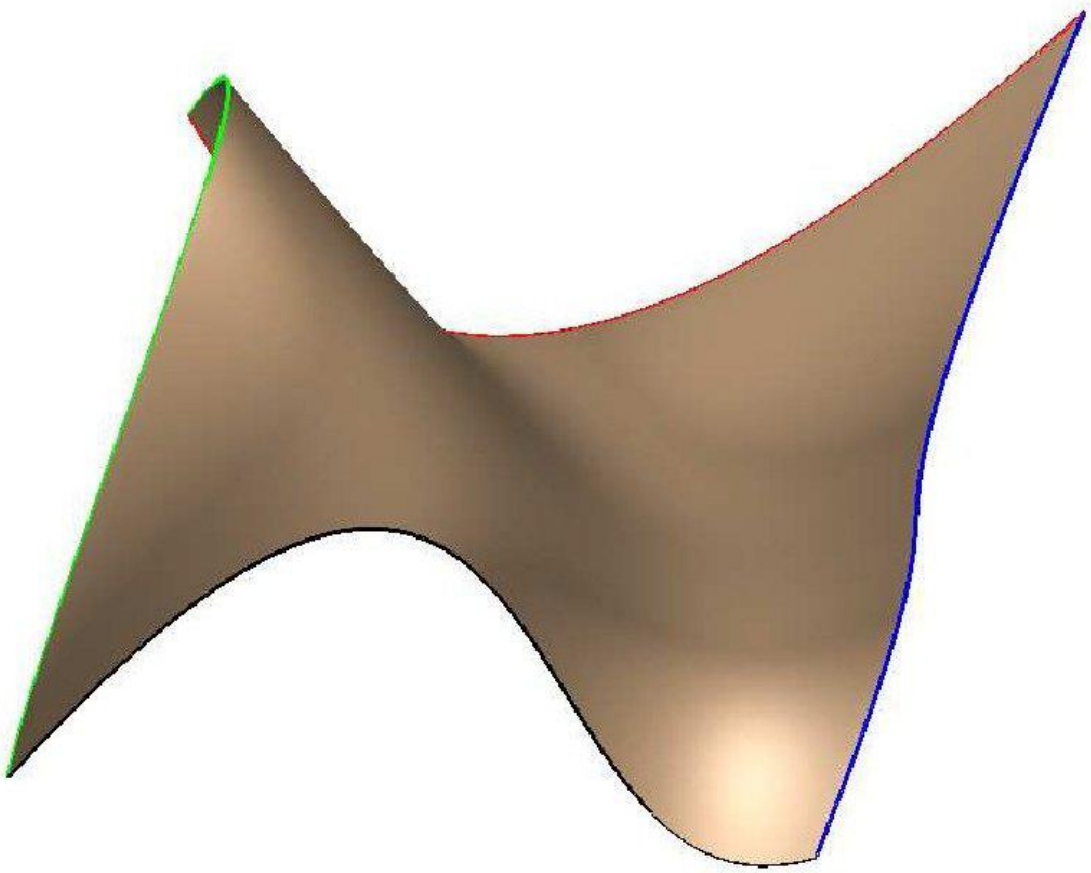
Coon's Process (Linear)



$P_u P_v$: Interpolation of the vertices of the patch

Interpolation between 4 Curves

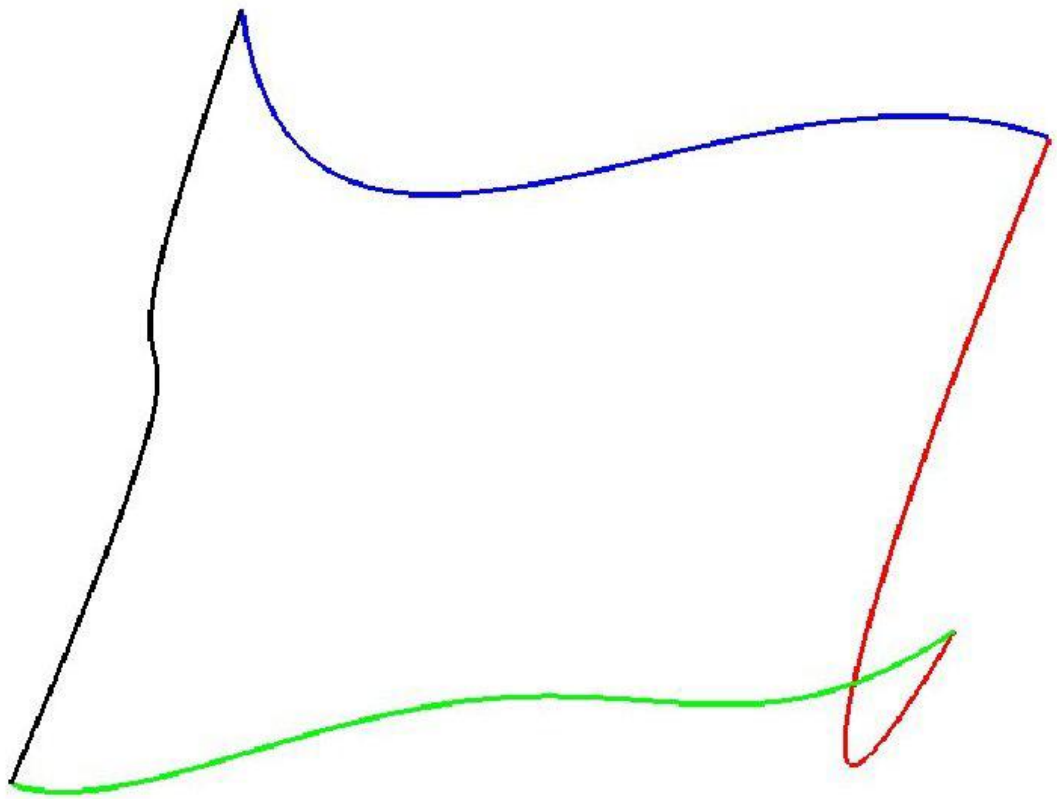
Coon's Process (Linear)



Final Surface : $P_u + P_v - P_u P_v$

Interpolation between 4 Curves

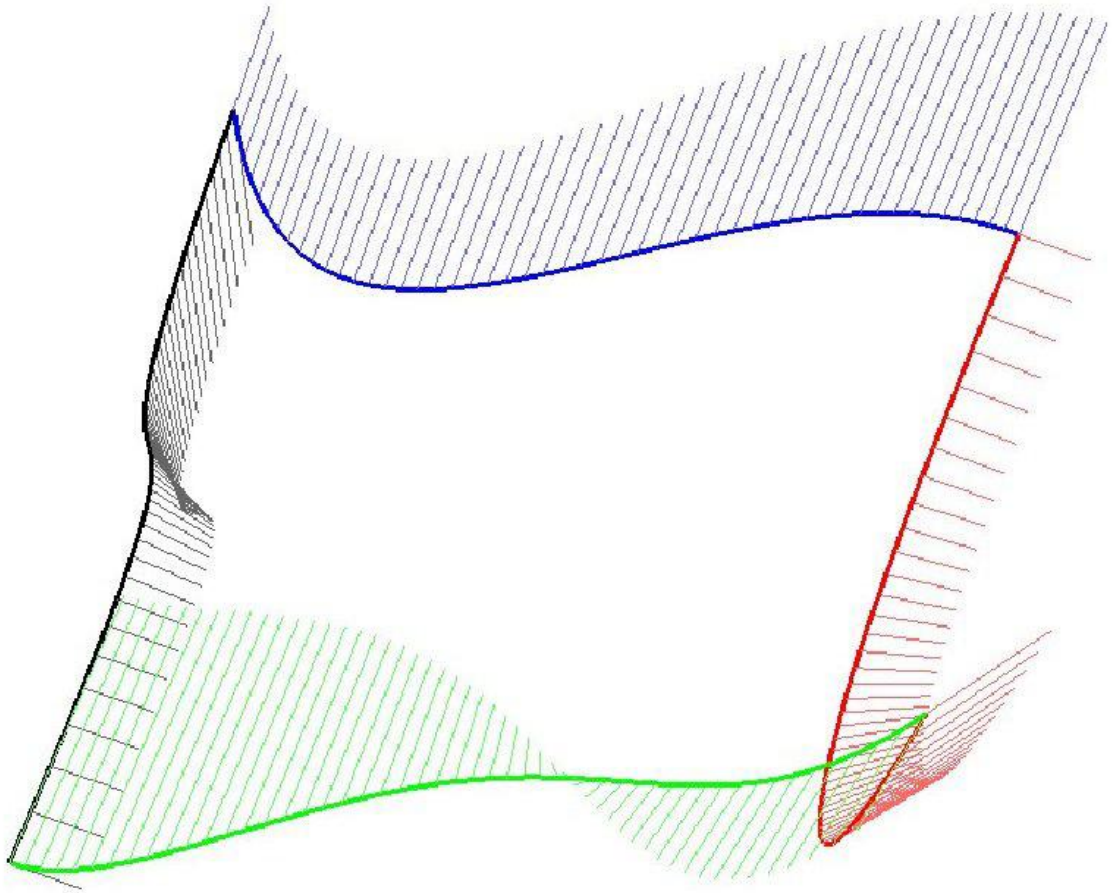
Coon's Process (Hermite)



4 boundary curves

Interpolation between 4 Curves

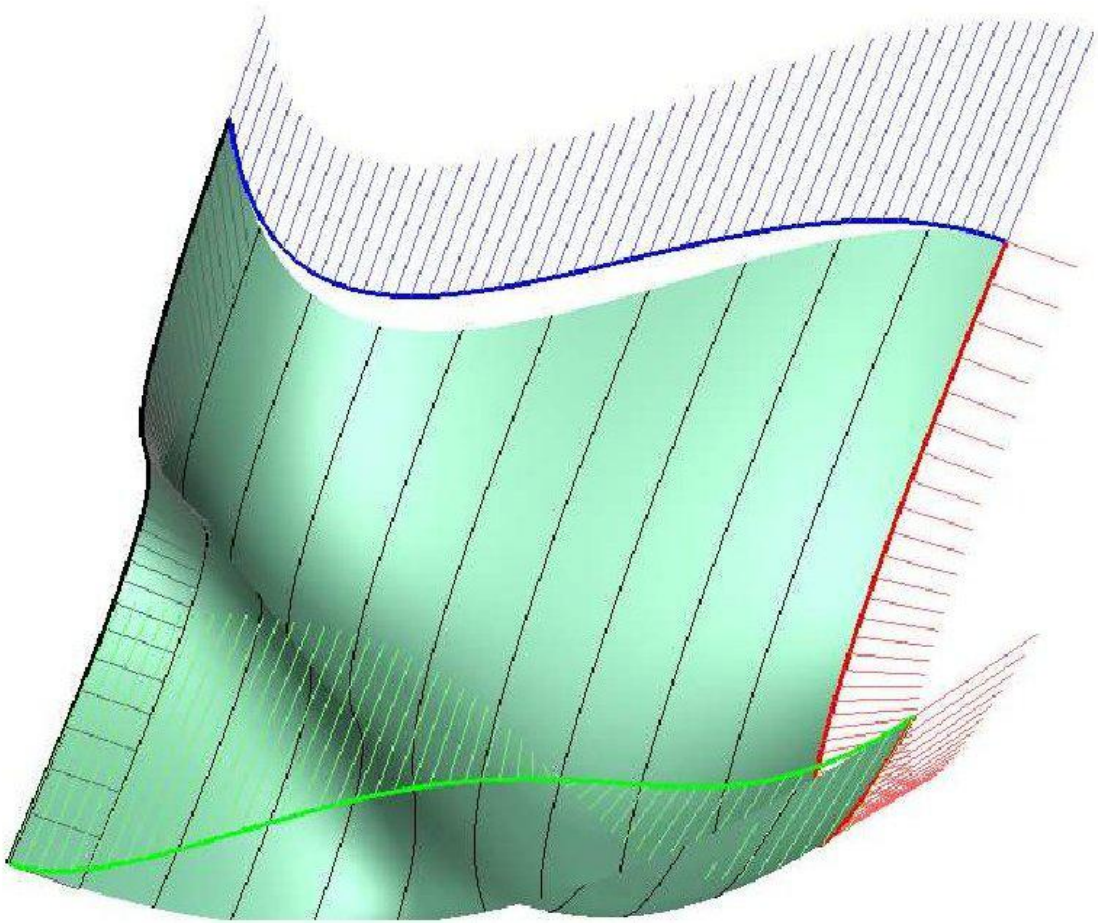
Coon's Process (Hermite)



4 boundary curves + derived tangents

Interpolation between 4 Curves

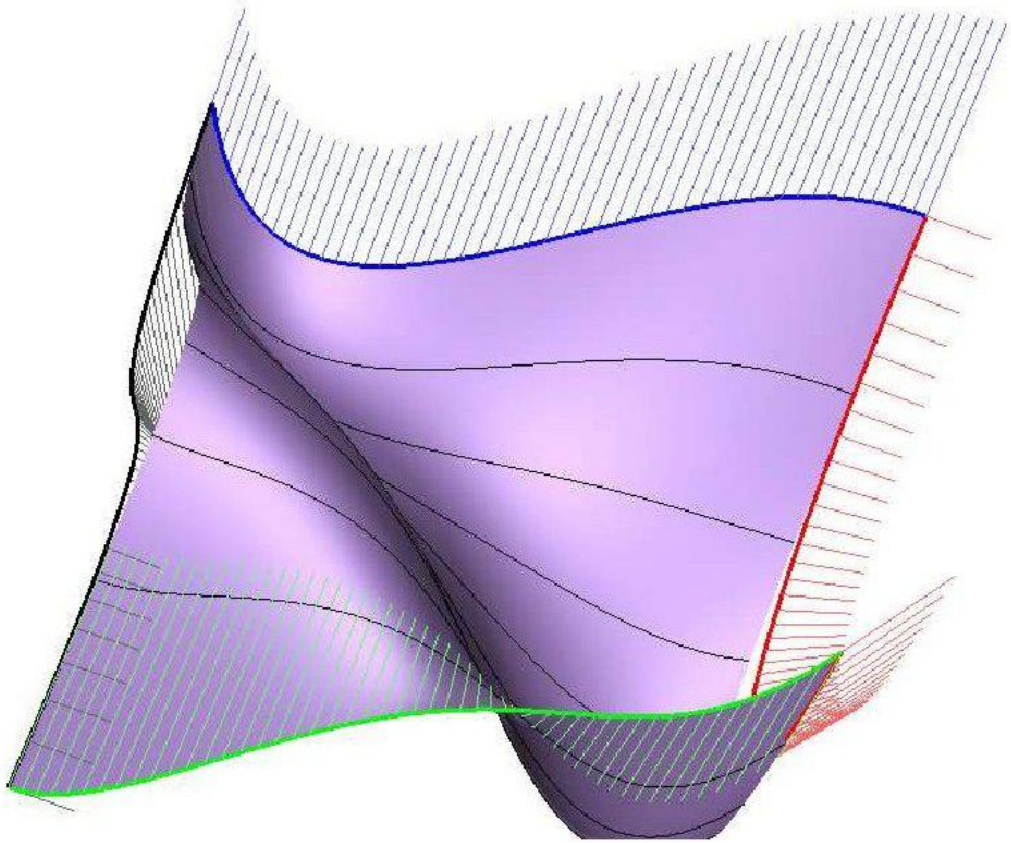
Coon's Process (Hermite)



P_u : Interpolation of curves $C_{v=0}$ *and* $C_{v=1}$

Interpolation between 4 Curves

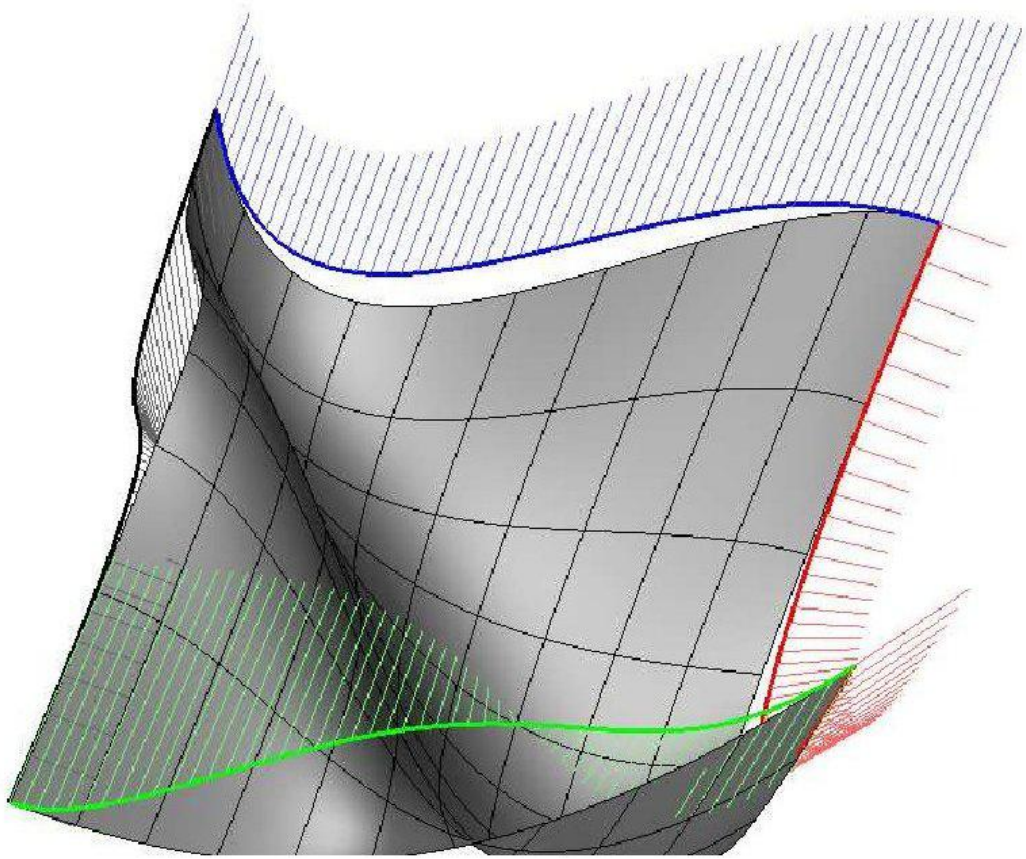
Coon's Process (Hermite)



P_v : Interpolation of curves $C_{u=0}$ *and* $C_{u=1}$

Interpolation between 4 Curves

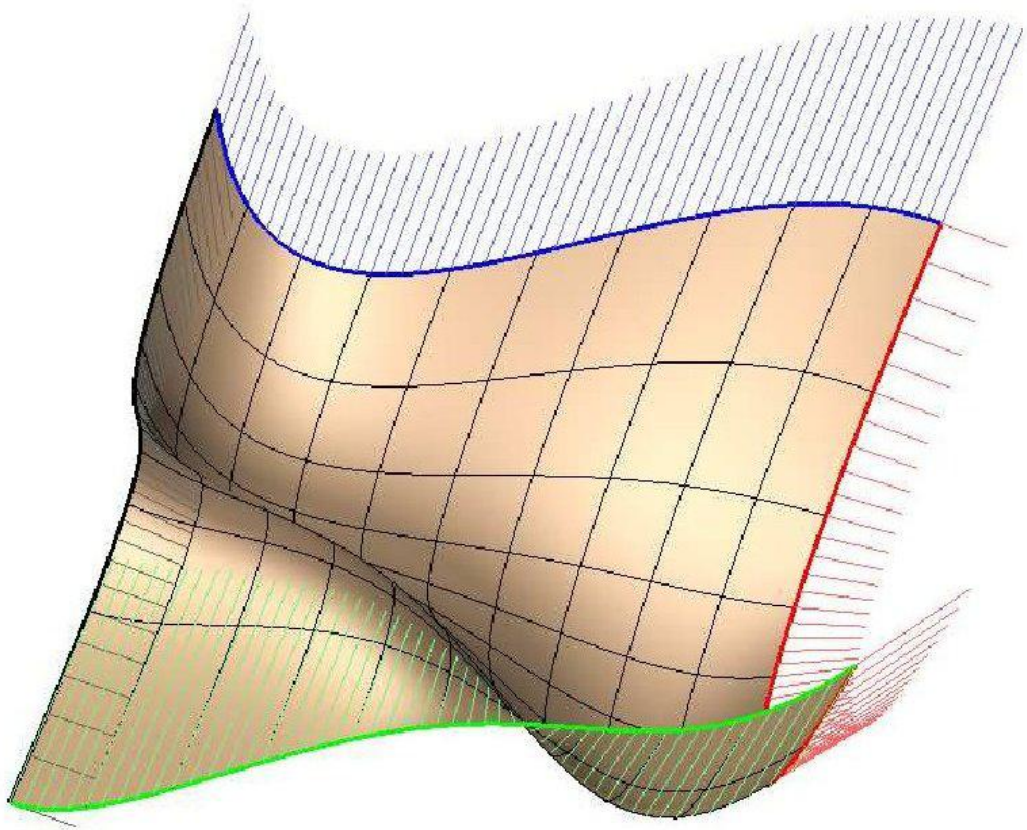
Coon's Process (Linear)



$P_u P_v$: Interpolation of the vertices of the patch

Interpolation between 4 Curves

Coon's Process (Hermite)



Final Surface : $P_u + P_v - P_u P_v$

Bibliography

- Cours de Nicolas Szafran 2011/2012 (http://www-ljk.imag.fr/membres/Nicolas.Szafran/)
- Scattered Data Interpolation and Approximation for Computer Graphics (Siggraph Asia course 2010)
- Implicit surface reconstruction from point clouds (Johan Huysmans's thesis)

PLAN

Parameterized / Ordered data

- Curve interpolation
- Surface interpolation
- Curve approximation
- Surface approximation

PLAN

Parameterized / Ordered data

- Curve interpolation
- Surface interpolation
- Curve approximation
- Surface approximation

PLAN

Parameterized / Ordered data

- Curve interpolation
- Surface interpolation
- Curve approximation
- Surface approximation