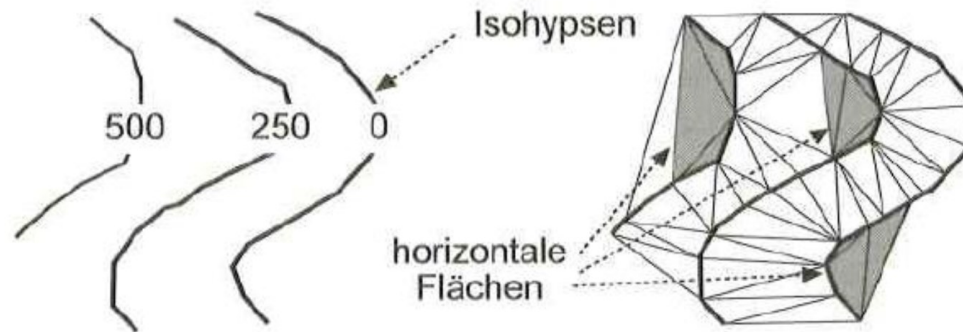
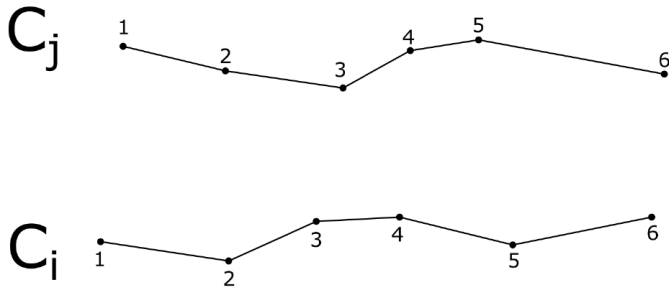


Parkettierung (*tiling*)

Triangulierung von Linienobjekten (z.B. Isohypsen, Isobaren) kann prinzipiell etwa als Constrained Delaunay Triangulation aufgefasst werden. Allerdings können dann u.U. ungewünschte Effekte auftreten (z.B. Plateaueffekte):



Triangulierung von Linienobjekten (z.B. Isohypsen, Isobaren) kann prinzipiell etwa als Constrained Delaunay Triangulation aufgefasst werden. Allerdings können dann u.U. ungewünschte Effekte auftreten (z.B. Plateaueffekte):

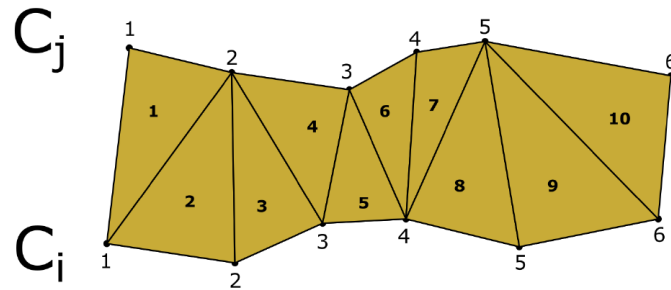


Es seien zwei Folgen $C_i = \{p_1, \dots, p_n\}$ und $C_j = \{q_1, \dots, q_m\}$ von Punkten gegeben. Eine **Parkettierung (Tiling)** mittels Dreiecken ist dann gegeben, wenn folgende Bedingungen erfüllt sind:

1. aufeinanderfolgende Punkte eines Linienobjektes C_i oder C_j sind durch eine Dreieckskante verbunden,
2. jedes Dreieck enthält maximal zwei Punkte, die in demselben Linienobjekt C_i oder C_j enthalten sind.

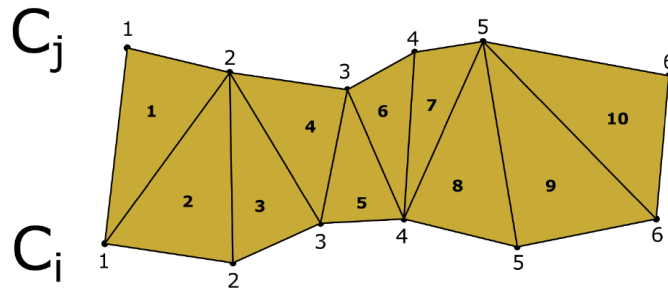
Parkettierung

Eine Parkettierung muss nicht zwangsläufig die gesamte konvexe Hülle aller Punkte umfassen. Ebenso ist es nicht ausgeschlossen, dass sich 3D Dreiecke überschneiden, wenn sie in die 2D Ebene projiziert werden (eine Parkettierung stellt also keine klassische Triangulierung dar).

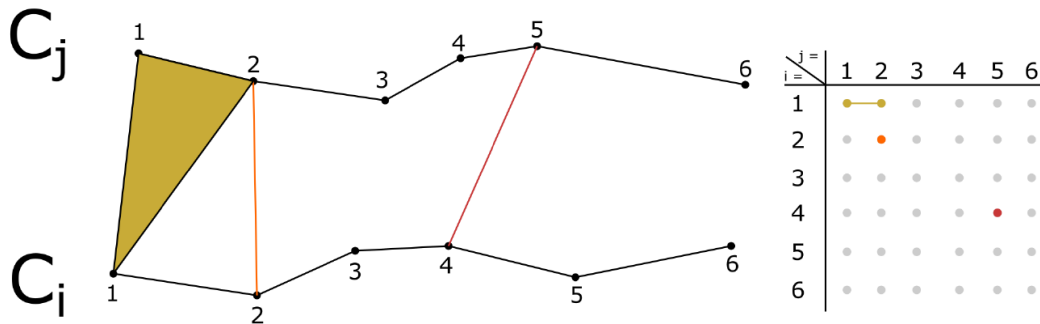


Parkettierung

Eine Parkettierung muss nicht zwangsläufig die gesamte konvexe Hülle aller Punkte umfassen. Ebenso ist es nicht ausgeschlossen, dass sich 3D Dreiecke überschneiden, wenn sie in die 2D Ebene projiziert werden (eine Parkettierung stellt also keine klassische Triangulierung dar).

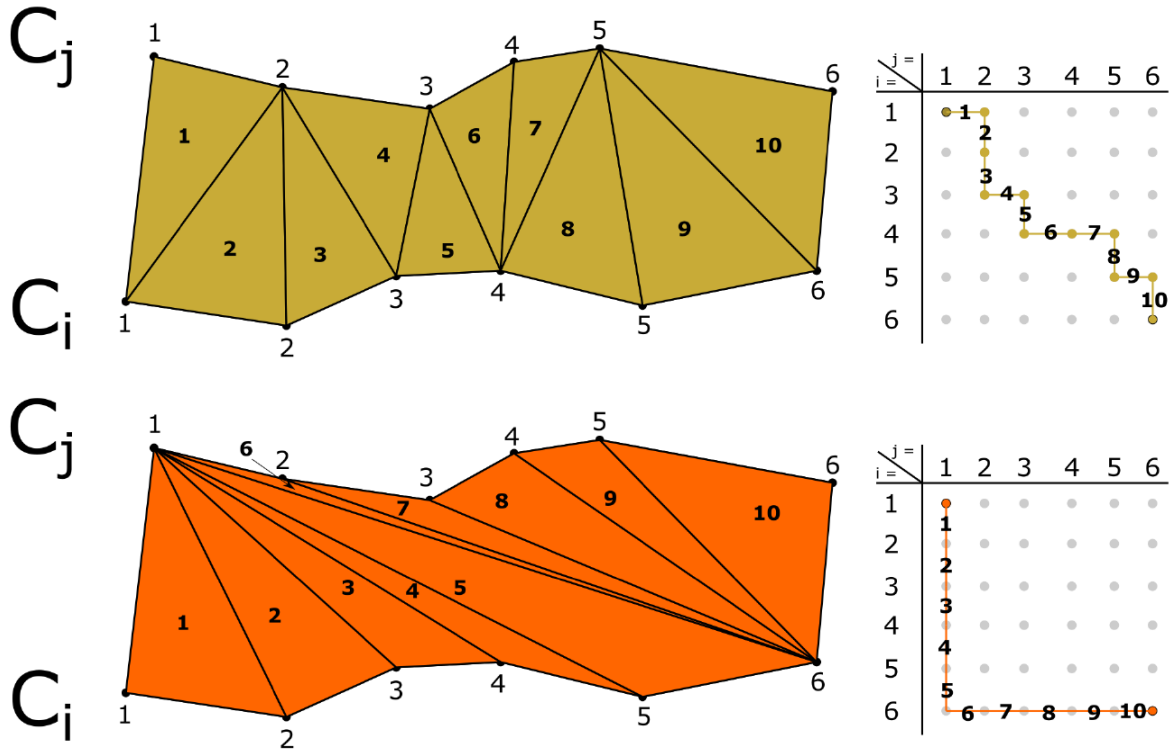


Parkettierungen können mit Hilfe eines Suchgraphens dargestellt werden: jeder Eintrag der Matrix beschreibt eine Verbindungslinie zwischen den Punkten der jeweiligen Zeile und Spalte.



Parkettierung

Jede Parkettierung kann mittels eines Pfades in dem Suchgraphen dargestellt werden



Parkettierung

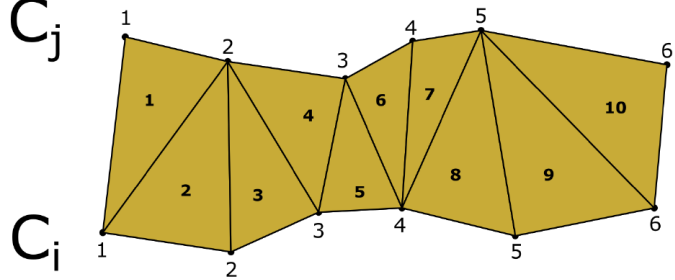
Eine **Gewichtungen der Kanten** in dem Suchgraphen erlauben es, die in einem gewissen Sinne **optimale Parkettierungen** zu finden (also Parkettierungen, die eine gewissen Zielfunktion minimieren; etwa die Gesamtfläche der Parkettierung).

Dazu stellt man Graphen mit gewichteten Kanten auf: einzeln gewichtet, als auch kumulativ gewichtet.

Parkettierung

Eine **Gewichtung der Kanten** in dem Suchgraphen erlauben es, die in einem gewissen Sinne **optimale Parkettierungen** zu finden (also Parkettierungen, die eine gewisse Zielfunktion minimieren).

Dazu stellt man C_j **zeln gewichtet**, als auch kumulativ gewichtet.



Gewichte

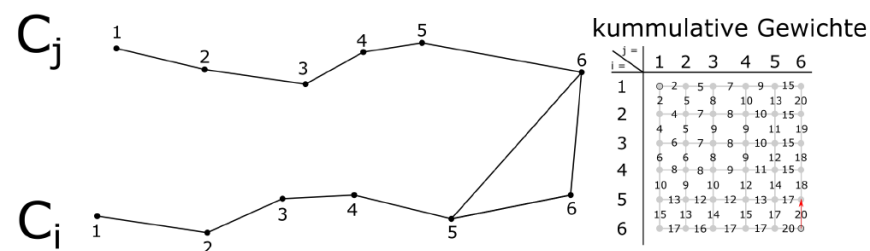
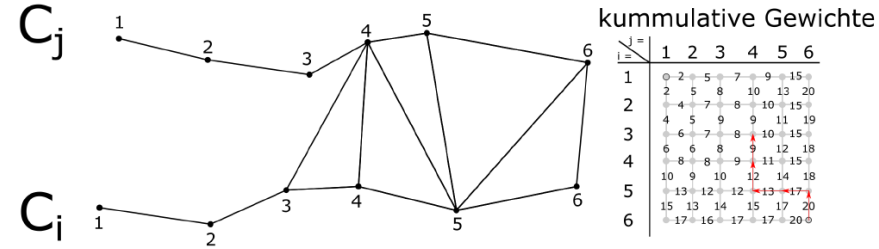
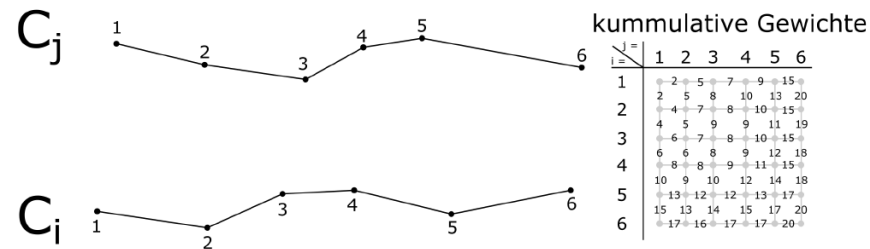
$i \backslash j =$	1	2	3	4	5	6
1	2	3	2	2	6	
2	2	3	3	3	4	5
3	2	1	2	1	1	4
4	2	2	1	2	5	
5	2	1	1	1	2	3
6	4	3	2	3	3	3
1	3	3	2	1	4	
2	5	4	4	3	4	3
3	2	3	3	2	3	

kummulative Gewichte

$i \backslash j =$	1	2	3	4	5	6
1	2	5	7	9	15	
2	4	7	8	10	15	
3	6	7	8	10	15	
4	6	6	8	9	12	18
5	8	8	9	11	15	
6	10	9	10	12	14	18
1	13	12	12	13	17	
2	15	13	14	15	17	20
3	17	16	17	17	20	

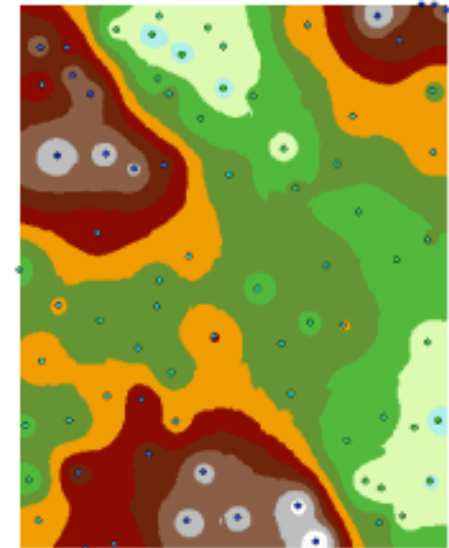
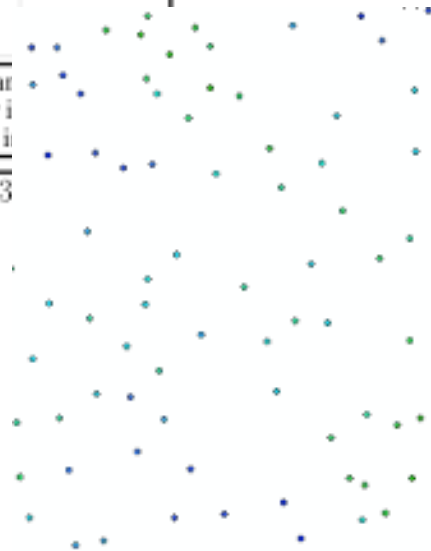
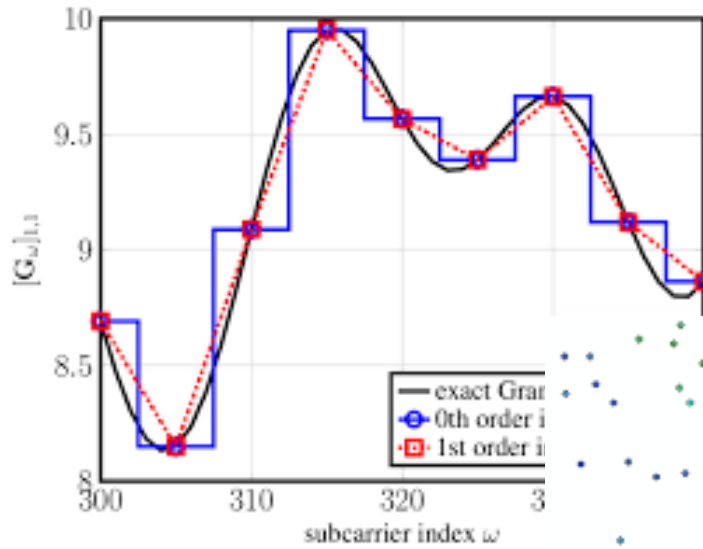
Parkettierung

Die optimale Parkettierung ergibt sich durch den kürzesten Pfad im kumuliert gewichteten Graphen. Diese optimale Parkettierung muss nicht eindeutig sein.

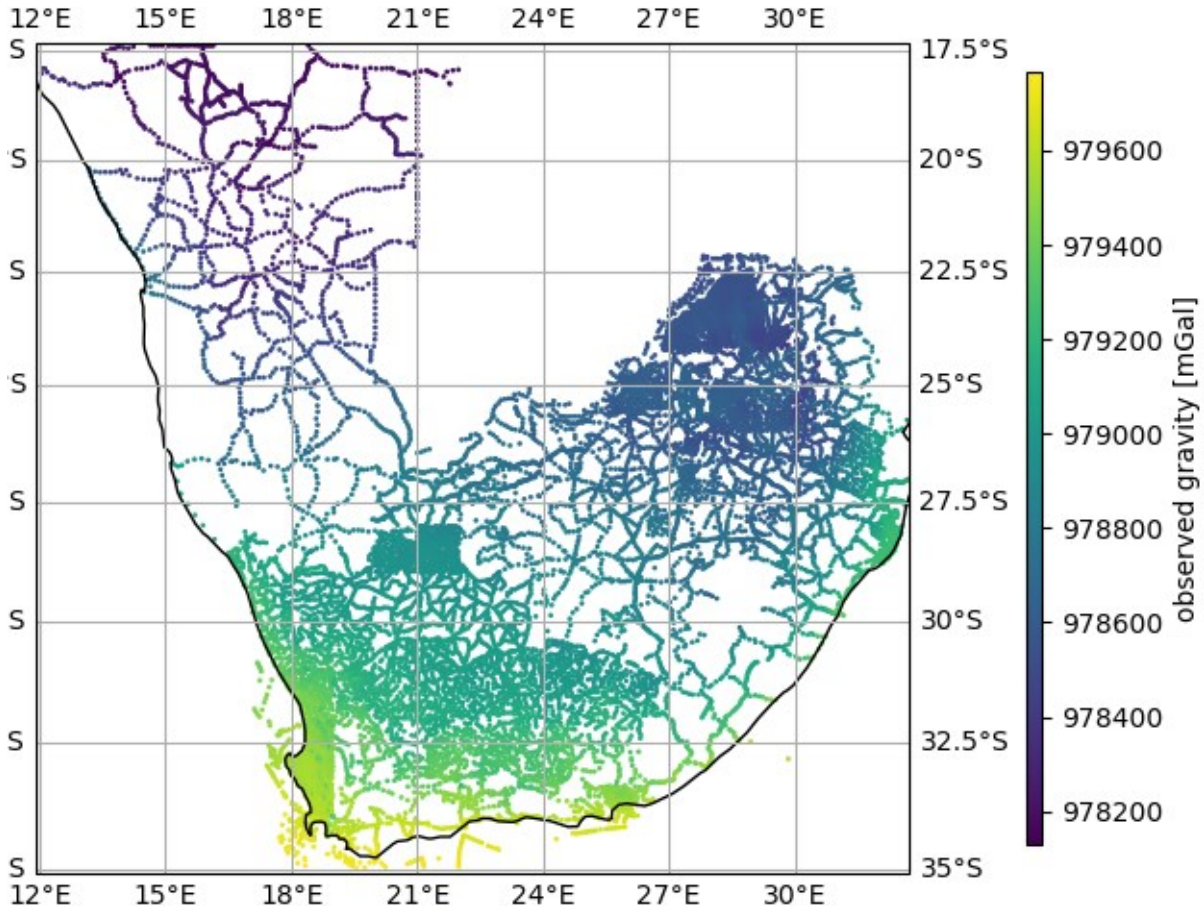


1. Definitionen, Funktionen, Anwendungen
2. Koordinatensysteme und -transformationen
3. Räumliche Datenmodellierung
4. Vermaschungen
- 5. Räumliche Interpolation**
6. Transformationen, Filtermethoden

Beispiele

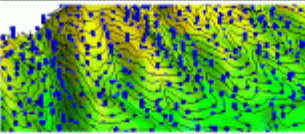

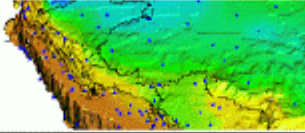
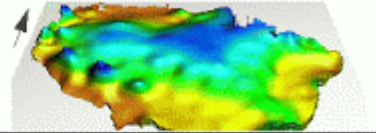
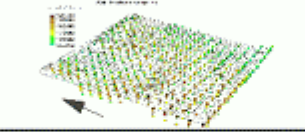
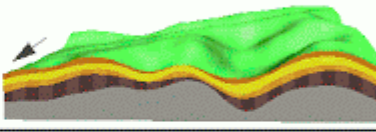
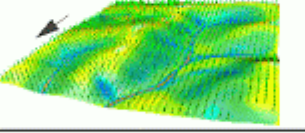
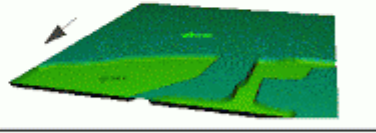
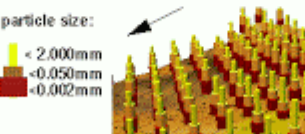
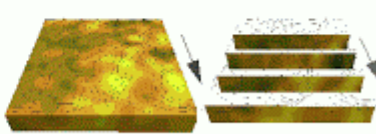
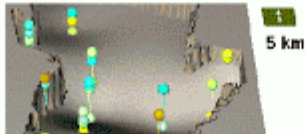
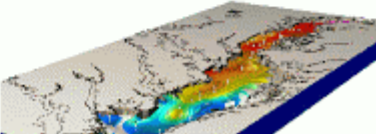


Beispiele



Beispiele

Figure 1. Representation of landscape phenomena as multivariate fields with interpolation performed by the RST method.

phenomenon (field)	point data	3D dynamic map
elevation: $z = f(x, y)$		
precipitation: $p_i = f_i(x, y); i = 1, \dots, 12$		
soil soil horizons: $z_i = f_i(x, y); i = 1, \dots, 5$		
land cover: $z + h_i = f_i(x, y), i = 1, \dots, 12$		
soil particle size (% clay): $c = f(x, y, z)$	<p>particle size:</p> <ul style="list-style-type: none"> ■ < 2.000mm ■ < 0.050mm ■ < 0.002mm 	
conc. of chemicals in water: $w = f(x, y, z, t)$	 <p>5 km</p>	

Mitasova, Mitas, Brown

fatra.cnr.ncsu.edu/~hmitaso/gmslab/asae97/mitasova973034.html

Interpolation

Im Allgemeinen liegen Messwerte nur an endlich vielen Positionen x_1, \dots, x_N vor (dies können physische Positionen im Raum, Zeitpunkte oder andere abstrakte Positionen sein). Für jede Position x_i liegt ein Messwert y_i für einen Parameter vor.

Das Ziel einer Interpolation ist die „Vorhersage“ des gesuchten Parameters y an einer beliebigen Position x , an dem initial kein Messwert vorliegt.

Dies erfolgt zumeist, indem eine Funktion f , mit

$f(x_i) = y_i, i = 1, \dots, N$ (**klassische / exakte Interpolation**) oder zumindest

$f(x_i) \approx y_i, i = 1, \dots, N$ (**Approximation**),

gefunden wird, welche bestimmte Eigenschaften aufweisen soll.

Die „Vorhersage“ für eine Position x erfolgt dann über $y = f(x)$.

Interpolation

Gegeben sei $(x_i, y_i), i = 1, \dots, N$. Die Interpolationsfunktion liegt zumeist in einen von zwei Formen vor, entweder

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

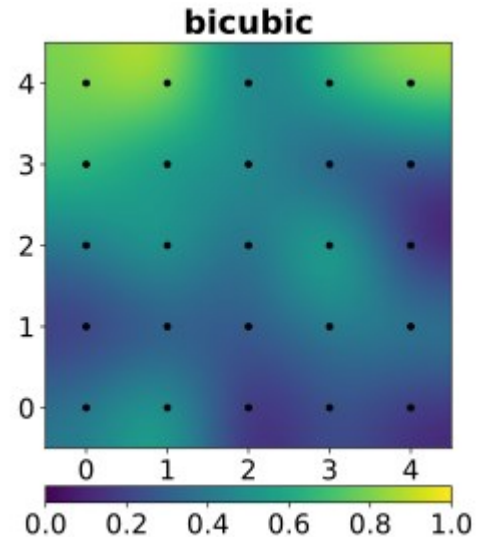
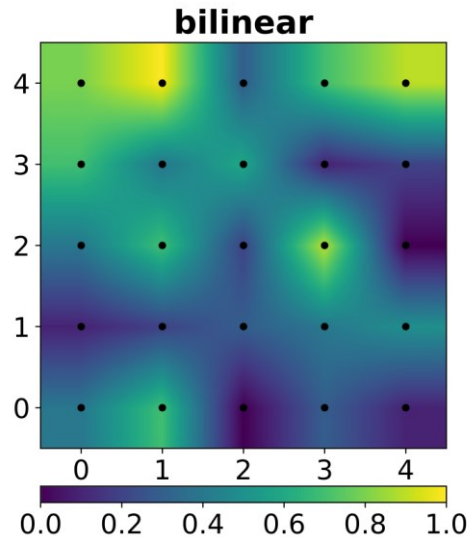
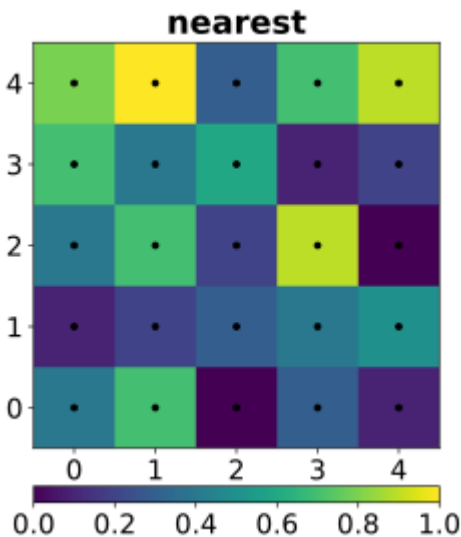
wobei hier die Gewichte λ_i von x abhängen, oder

$$f(x) = \sum_{i=1}^N \lambda_i f_i(x),$$

mit konstanten Gewichten λ_i (abhängig von y_i).

Gitter-basierte / Element-basierte Interpolation

- **Nearest Neighbour Interpolation** (basiert auf Voronoi Vermaschung): $f(x)$ diskontinuierlich
- **Natural Neighbour Interpolation** (Voronoi Vermaschung benötigt): $f(x)$ kontinuierlich + differenzierbar
- **Lineare Interpolation auf Triangulationen**: $f(x)$ kontinuierlich
- **Bilinear Interpolation** (benötigt Rechteck-Gitter): $f(x)$ kontinuierlich



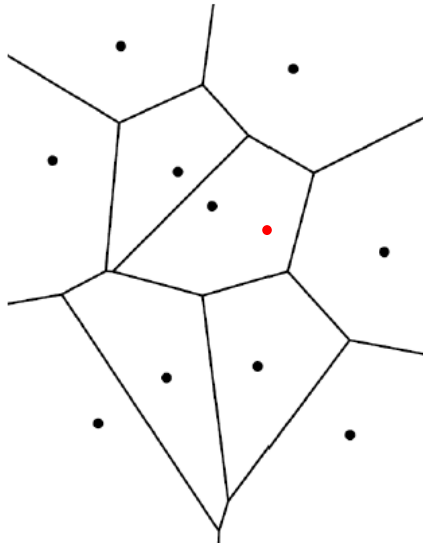
Nearest Neighbour Interpolation

Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ die Voronoi Vermaschung der Punktmenge $P = \{p_1, \dots, p_N\}$ und sei y_i ein Parameterwert an einem Punkt p_i , dann

$$f(x) = \sum_{i=1}^N \lambda_i(x) y_i$$

mit den Gewichten

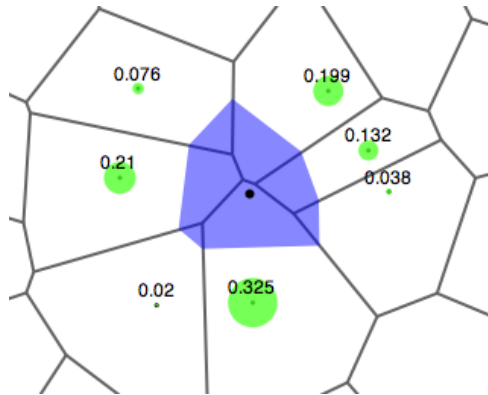
$$\lambda_i(x) = \begin{cases} 1, & x \in V_i \\ 0, & x \notin V_i \end{cases}$$



Eigenschaften:

- $f(x)$ diskontinuierlich;
- Lokaler Interpolator, der nur auf (p_i, y_i) basiert, wenn $x \in V_i$;
- Extrapolation ist konzeptuell möglich.

Natural Neighbour Interpolation



Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ die Voronoi Vermaschung der Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^n$ und sei y_i ein Parameterwert an einem Punkt p_i . Sei des weiteren $x \in \mathbb{R}^n$ ein beliebiger Punkt und \mathcal{V}' die Voronoi Vermaschung bezüglich $P \cup \{x\}$. Sind die Punkte $p_{i_x}, i = 1, \dots, n_x$ die **starken** Voronoi Nachbarn von x bezüglich \mathcal{V}' , dann gilt

$$\lambda_{i_x} = \frac{\text{area}(V'(x) \cap V(p_{i_x}))}{\text{area}(V'(x))}.$$

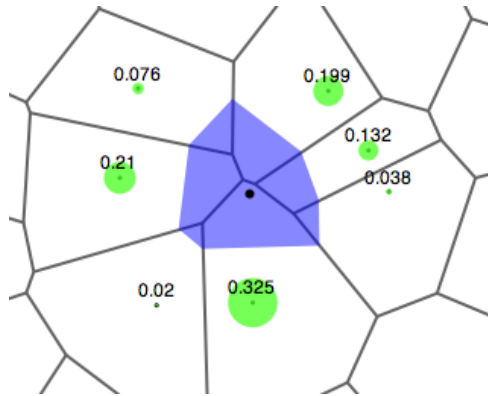
Sind Messwerte y_i für die Punkte p_i gegeben, kann man $f(x)$ wiefolgt vorhersagen

$$f(x) = \sum_{i=1}^{n_x} \lambda_{i_x} y_{i_x}.$$

Für $\lambda_{i_x} > 0$ gilt $\sum_{i=1}^{n_x} \lambda_{i_x} = 1$.

Dies ist eine typische Eigenschaft von Interpolationsgewichten.
(Isoparametrischer Interpolator)

Natural Neighbour Interpolation



Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ die Voronoi Vermaschung der Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^n$ und sei y_i ein Parameterwert an einem Punkt p_i . Sei des weiteren $x \in \mathbb{R}^n$ ein beliebiger Punkt und \mathcal{V}' die Voronoi Vermaschung bezüglich $P \cup \{x\}$. Sind die Punkte $p_{i_x}, i = 1, \dots, n_x$ die **starken** Voronoi Nachbarn von x bezüglich \mathcal{V}' , dann gilt

$$\lambda_{i_x} = \frac{\text{area}(V'(x) \cap V(p_{i_x}))}{\text{area}(V'(x))}$$

Sind Messwerte y_i für die Punkte p_i gegeben, kann man $f(x)$ wiefolgt vorhersagen

$$f(x) = \sum_{i=1}^{n_x} \lambda_{i_x} y_{i_x}$$

Eigenschaften:

- $f(x)$ ist kontinuierlich und differenzierbar;
- Lokaler Interpolator, der nur Tupel (p_i, y_i) berücksichtigt, die die starken Voronoi-Nachbarn von x sind;
- Extrapolation ist konzeptuell möglich.