B-Spline-Kurve und -Basisfunktionen

Eine **B-Spline-Kurve** der Ordnung k ist ein stückweise aus **B-Splines (Basisfunktion)** zusammengesetztes Polynom vom Grad $(k-1)$, das an den Segmentübergängen im allgemeinen C^{k-2} stetig differenzierbar ist.

Dabei seien B-Splines Stückweise Polynome, denen die folgenden geordneten Parameterwerte zugrundeliegen:

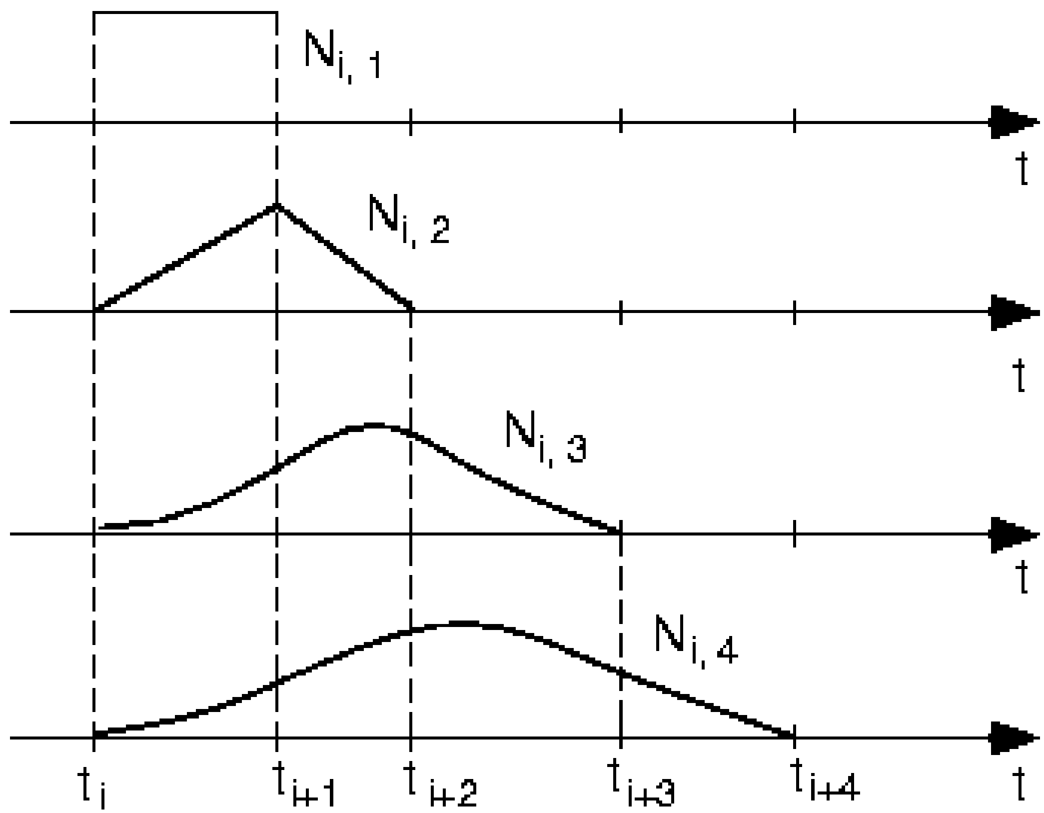
$$\mathbf{t} = (t_0, t_1, t_2, \ldots, t_m, t_{m+1}, \ldots, t_{m+k}),$$

wobei

- m: wird von der Anzahl der zu interpolierenden Punkte bestimmt
- k: die festgelegte Ordnung der B-Spline-Kurve

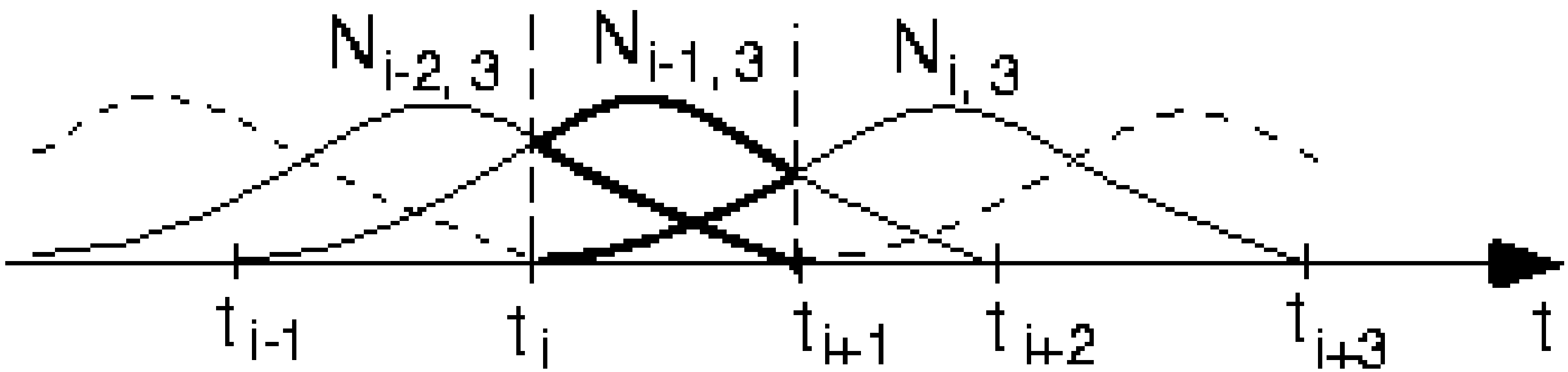
Beispiele von B-Splines

B-Splines der Ordnung 1, 2, 3 und 4:



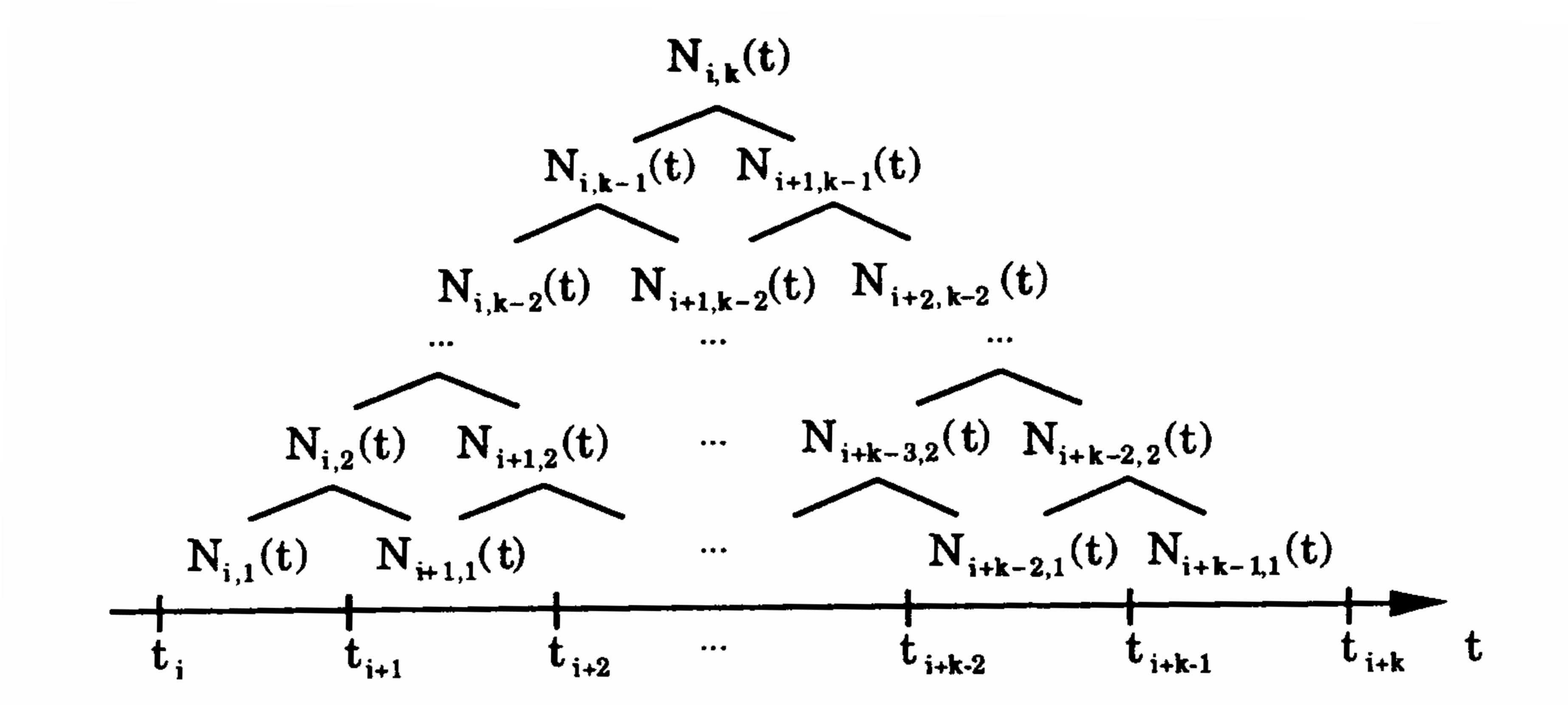
Innerhalb eines Parameterintervals gibt es k sich überlappende B-Splines.

Ein Beispiel der kubischen B-Splines



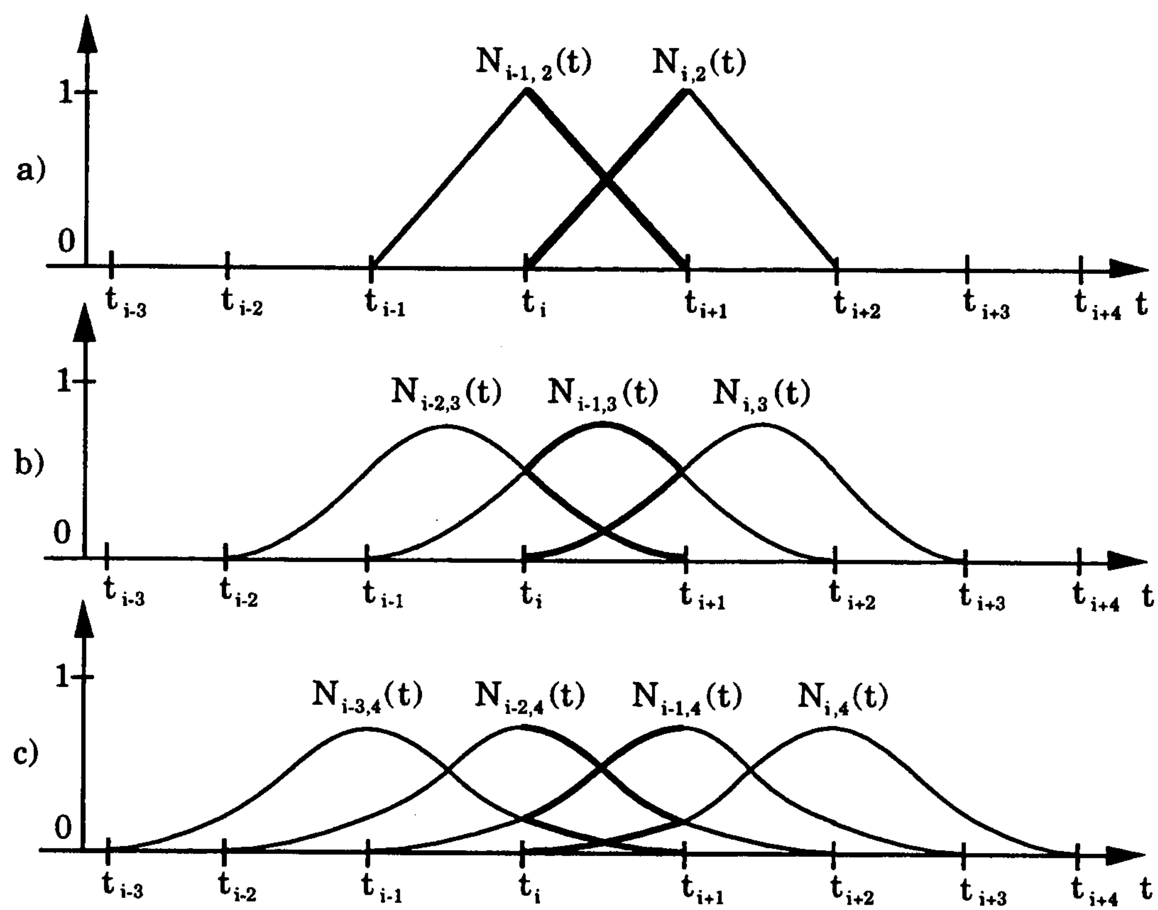
B-Splines der Ordnung k - I

Das rekursive Definitionsverfahren einer B-Spline-Basisfunktion $N_{i,k}(t)$:

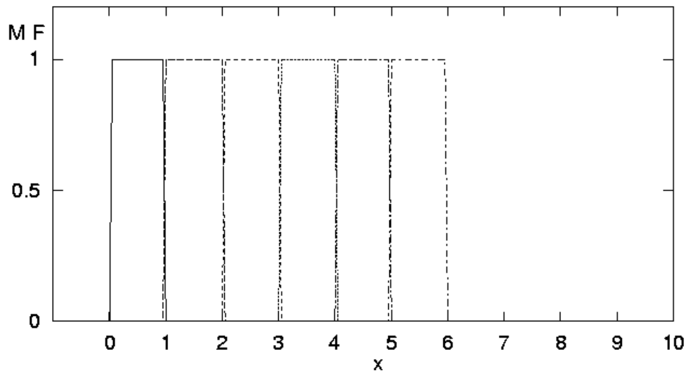


B-Splines der Ordnung k - II

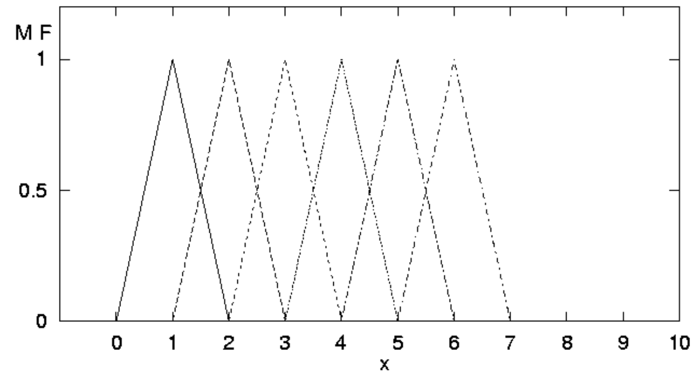
Aktuelle Segemente der B-Spline-Basisfunktionen der Ordnungen 2, 3 und 4 für $t_i \leq t < t_{i+1}$:



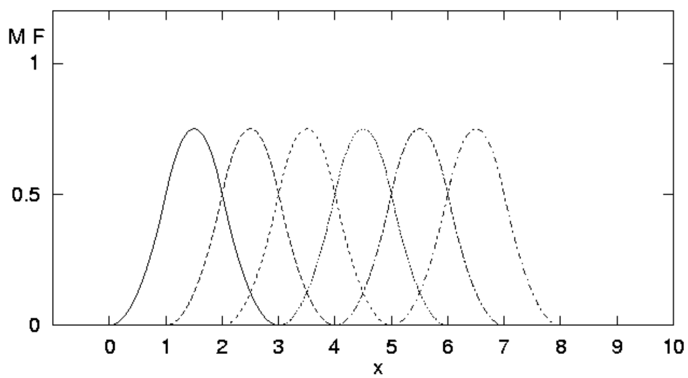
Uniforme B-Splines der Ordnung 1 bis 4



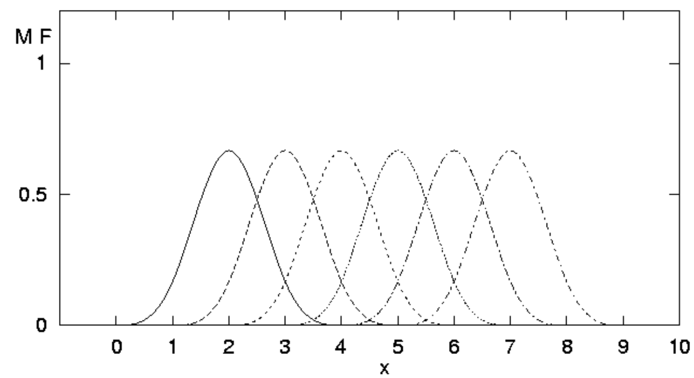
$k{=}1$



$k{=}2$



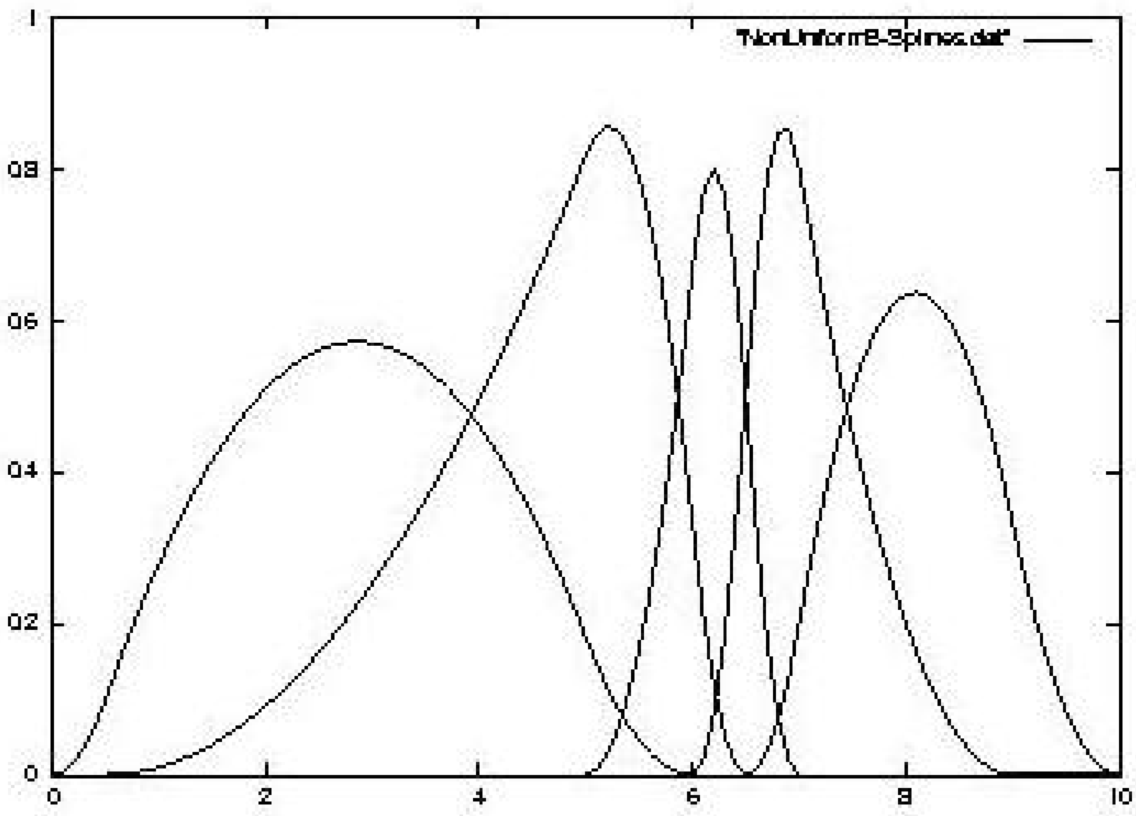
$k{=}3$



$k{=}4$

Nichtuniforme B-Splines

Ordnung 3:



Eigenschaften der B-Splines

Partition of unity: $\sum_{i=0}^{k} N_{i,k}(t) = 1$.
Positivity: $N_{i,k}(t) \geq 0$.
Local support: $N_{i,k}(t) = 0$ for $t \notin [t_i, t_{i+k}]$.
C^{k-2} *continuity:* If the knots $\{t_i\}$ are pairwise different from each other, then $N_{i,k}(t) \in C^{k-2}$, i.e. $N_{i,k}(t)$ is $(k-2)$ times continuously differentiable.

Gewinnung einer B-Spline-Kurve

Eine B-Spline-Kurve kann dadurch konstruiert werden, daß eine Menge von vorgegebenen Größen mit diesen B-Splines gemischt werden:

$$\mathbf{r}(\mathrm{t}) = \sum_{\mathrm{j}=0}^{\mathrm{m}} \mathbf{v}_{\mathrm{j}} \cdot \mathrm{N}_{\mathrm{j},\mathrm{k}}(\mathrm{t})$$

wobei $\mathbf{v}_j$ *Kontrollpunkte* (*de Boor-Punkte*) genannt werden.

Sei ein Parameter t gegeben, ist $\mathbf{r}(t)$ ein Punkt dieser B-Spline-Kurve.

Wenn t von t_{k-1} bis zu t_{m+1} variiert, so stellt $\mathbf{r}(\mathrm{t})$ eine C^{k-2} stetig differenzierbare Kurve dar.

Berechnung von Kontrollpunkten aus Datenpunkten

Die Punkte $\mathbf{v}_j$ sind nur bei $k = 2$ identisch mit den Datenpunkten zur Interpolation, sonst nicht.

Ein Kontrollpunktzug bildet eine konvexe Hülle für die Interpolationskurve.

Zwei Verfahren zur Berechnung von Kontrollpunkten aus Datenpunkten:

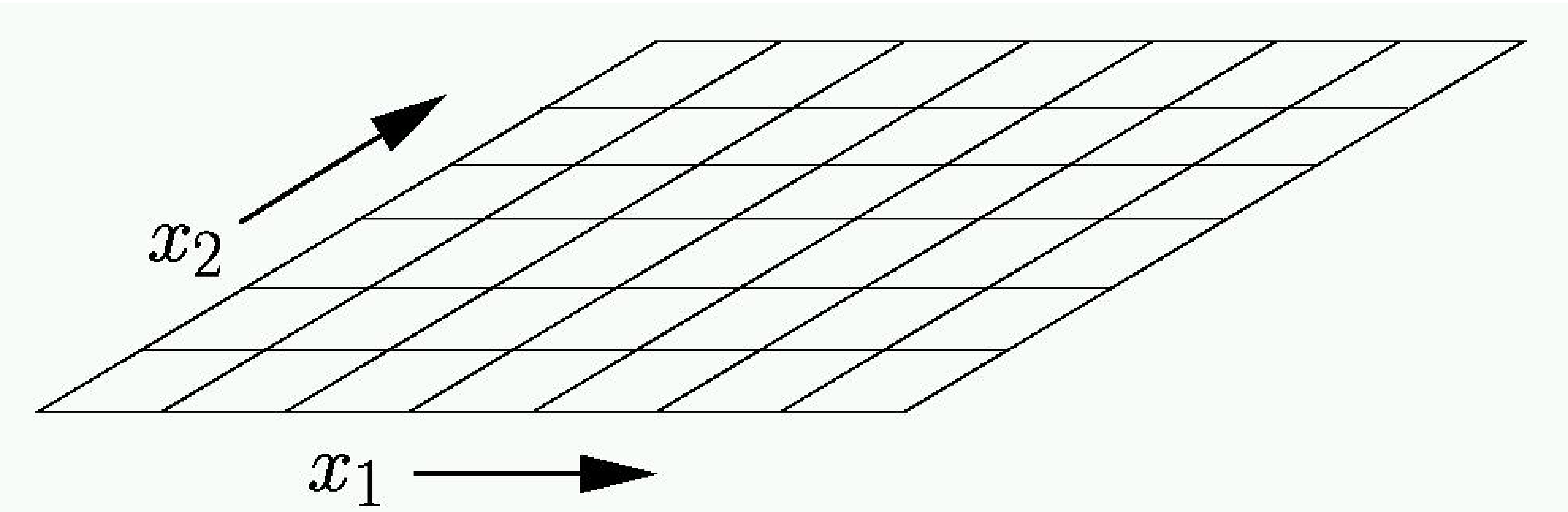
1 Durch die Lösung des folgenden Gleichungssytems (**Böhm84**):

$$\mathbf{q}_j(t) = \sum_{j=0}^{m} \mathbf{v}_j \cdot N_{j,k}(t)$$

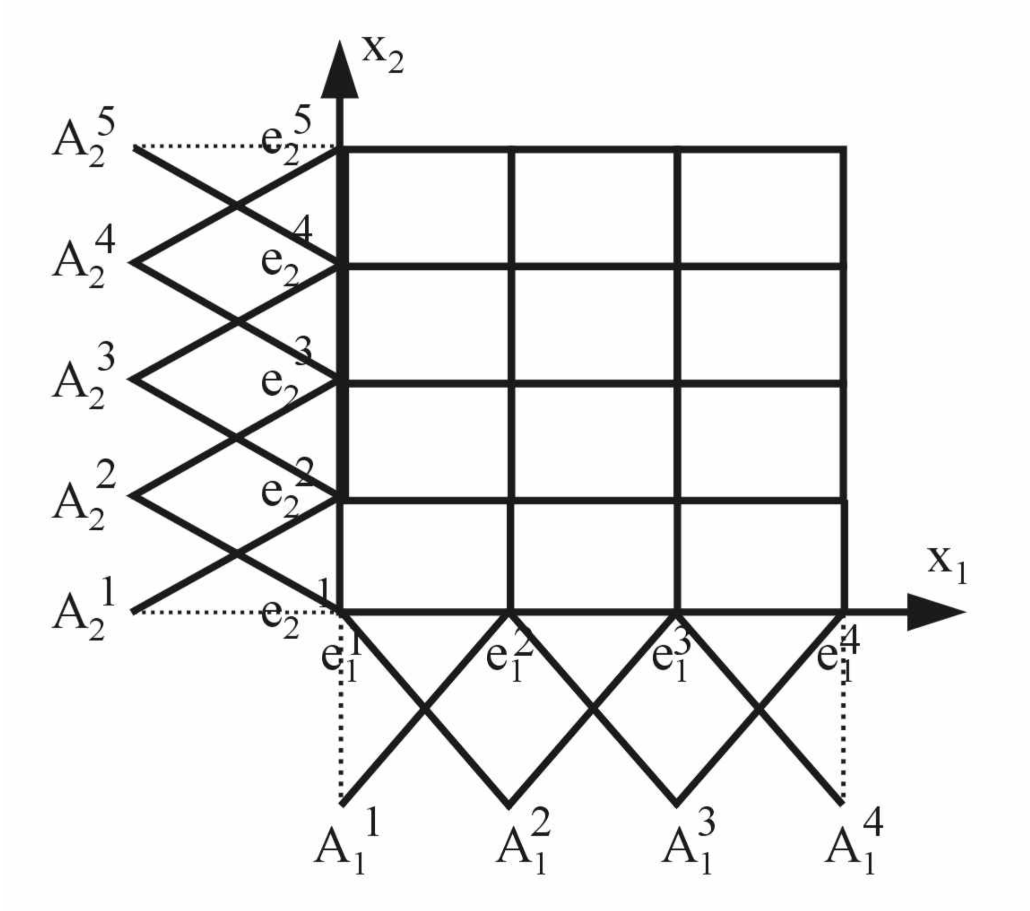
wobei $\mathbf{q}_j$ die Datenpunkte für die Interpolation sind, $j = 0, \cdots, m$.

2 Durch Lernen basierend auf dem Gradient-Abstieg (**Zhang98**).

Lattice - I

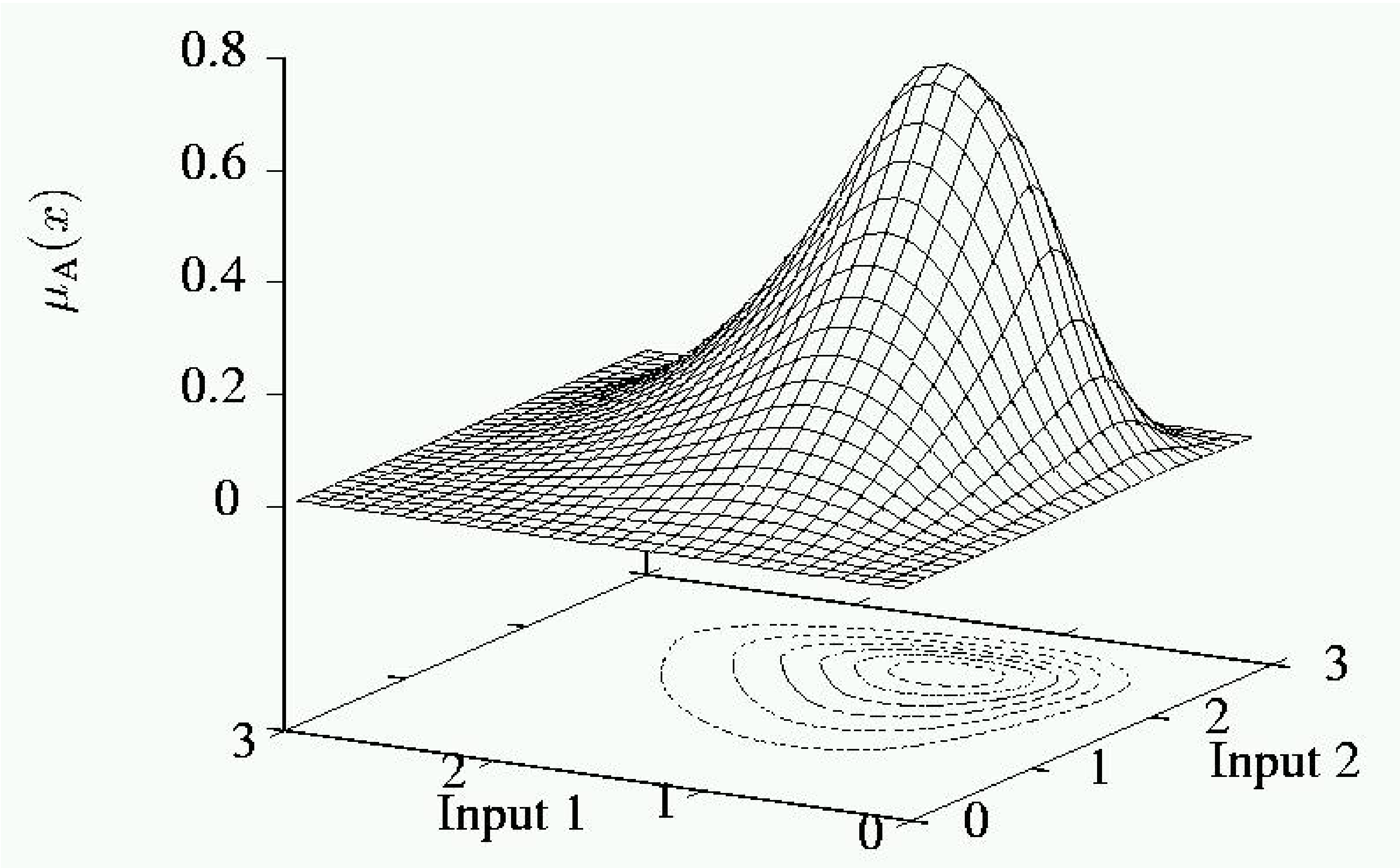


Lattice - II



x_2
x_1
A_2^5
A_2^4
A_2^3
A_2^2
A_2^1
e_2^5
e_2^4
e_2^3
e_2^2
e_2^1
e_1^1
e_1^2
e_1^3
e_1^4
A_1^1
A_1^2
A_1^3
A_1^4

Tensor-Produkt 2D-NURBS



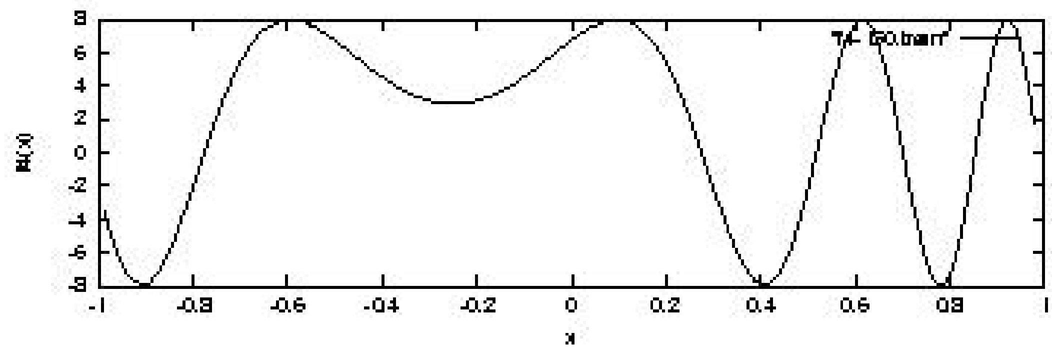
Problematik in der realen Welt

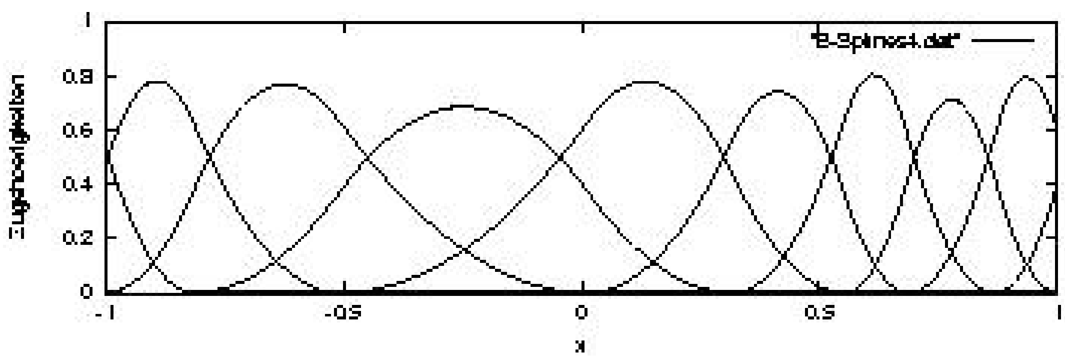
- **Modellierung**: Lernen aus Beispielen, selbstoptimierende Gestaltung, Vorhersagen, ...
- **Regelung**: Perzeption-Aktion-Zyklus, Zustandsregelung, Identifikation dynamischer Systeme, ...

Funktionsapproximation als Benchmark zur Wahl eines Modells

Funktionsapproximation - 1D Beispiel

Eine Testfunktion $f(x) = 8sin(10x^2 + 5x + 1)$ mit $-1 < x < 1$ und die richtig verteilten B-Splines:





Lattice

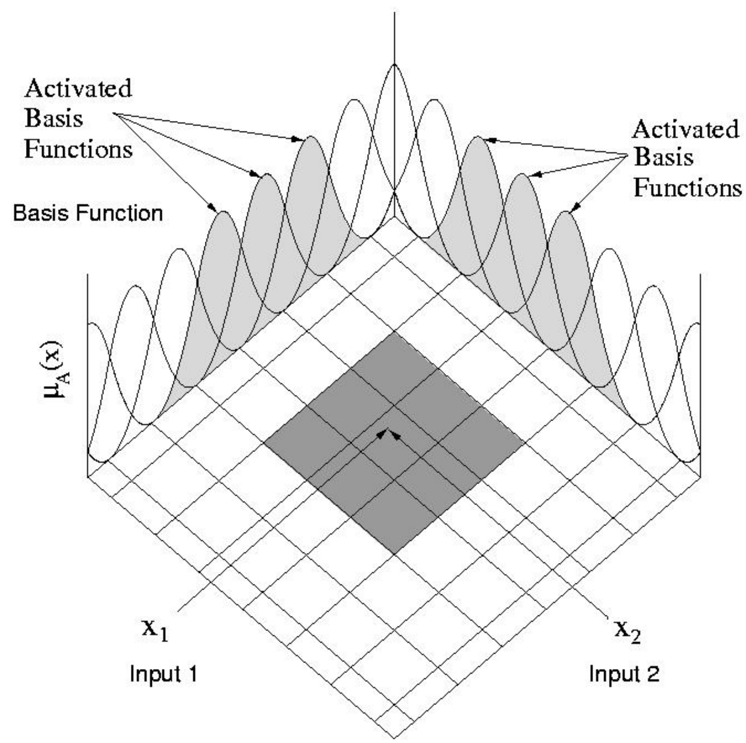
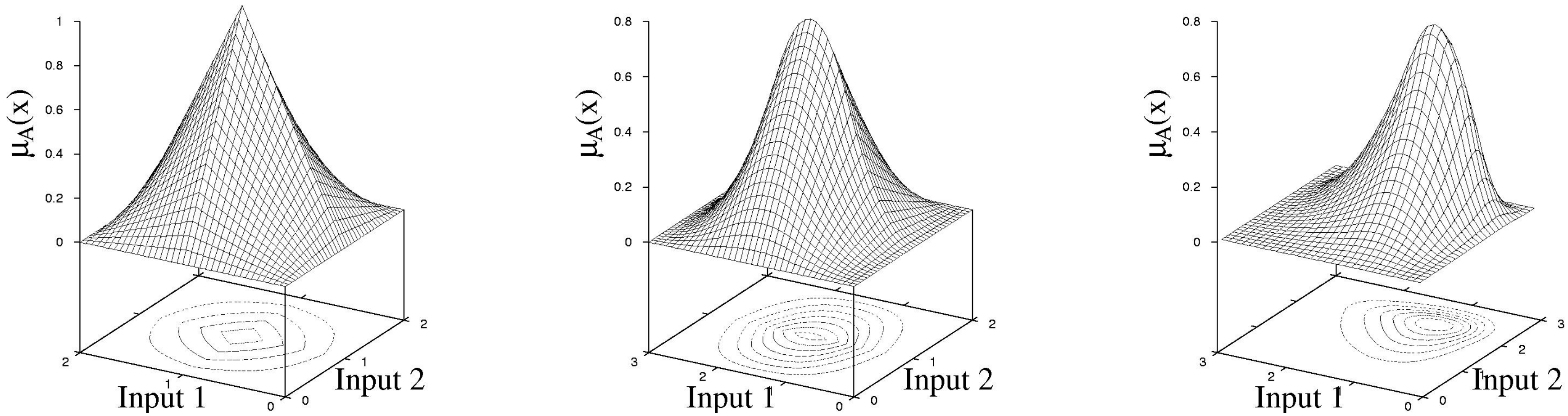


Abbildung 1: The B-spline model – a two-dimensional illustration.

Jedes n-dimensionale Viereck $(n > 1)$ wird von dem j^{th} multivariaten B-spline $N_k^j(x)$ bedeckt. $N_k^j(x)$ ist über den Tensorprodukt n univariate B-splines:

$$N_k^j(x) = \prod_{j=1}^{n} N_{i_j,k_j}^j(x_j) \tag{1}$$



(a) Tensor product of two, order 2 univariate B-splines.

(b) Tensor product of one order 3 and one order 2 univariate B-splines.

(c) Tensor product of two univariate B-splines of order 3.

Abbildung 2: Bivariate B-splines formed by taking the tensor product of two univariate B-splines.

Allgemeine Anforderungen an einen Approximator

- Universalität: Approximation von beliebigen Funktionen
- Generalisierung: gute Approximation ohne *Overfitting*
- Adaptivität: selbsteinstellend anhand von neuen Daten
- Parallelität: Rechnen nach biologischen Vorbildern
- Interpretierbarkeit: mindestens als “Grey-box” anstatt “Black-box”

Bedeutung der Interpretierbarkeit eines Modells

Richard P. Feynman: “the way we have to describe nature is generally incomprehensible to us”.

Albert Einstein: “it should be possible to explain the laws of physics to a barmaid”.

Wichtige Gründe für symbolische Interpretierbarkeit eines Approximators:

- Linguistische Modellierung bietet ein Mittel zur Fertigkeitsübertragung von einem Experten auf einen Computer oder Roboter.

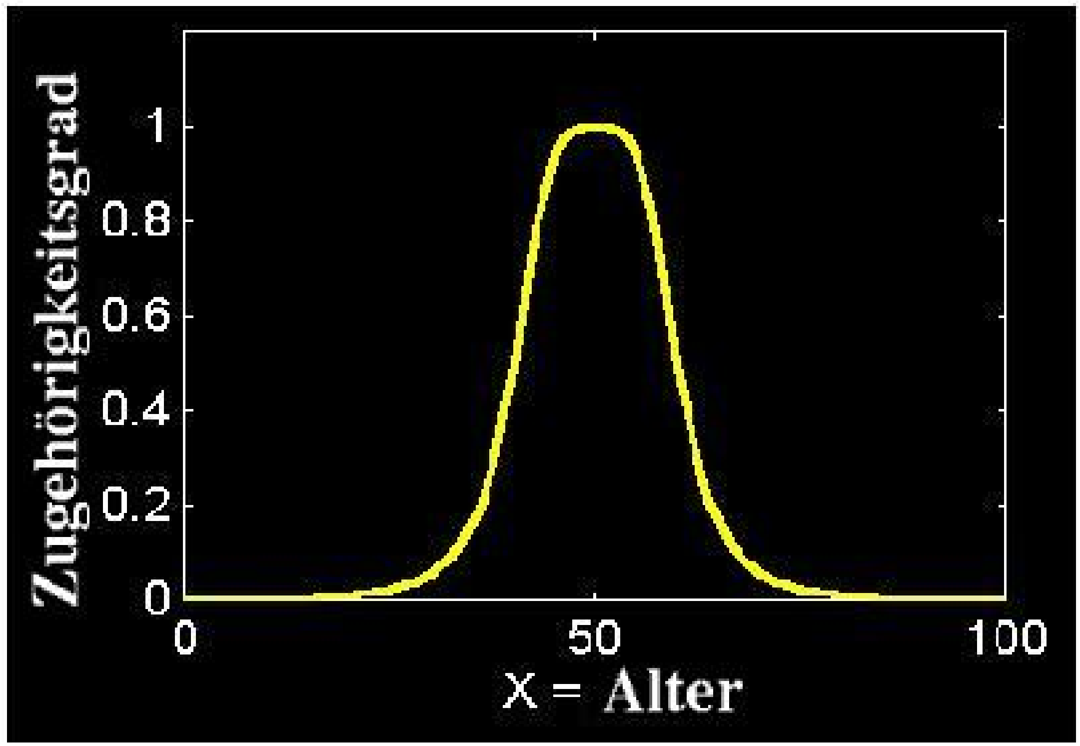
- Automatisches Lernen eines transparenten Modells erleichtert die Analyse, Validierung und Überwachung bei der Entwicklung eines Modells bzw. eines Reglers.

- Transparente Modelle besitzen vielfältige Anwendungsmöglichkeiten in *Decision-Support Systems*.

Symbolumwandlung der Kernfunktionen

Positiv definierte, konvexe Kernfunktionen können als Fuzzy-Mengen betrachtet werden. Z.B.:

$$\mu_B(x) = \frac{1}{1 + \left(\frac{x-50}{10}\right)^2}$$



B-Spline ANFIS

Bei einem B-Spline ANFIS mit n Eingängen $x_1, x_2, \ldots, x_n$, werden Regeln der folgenden Form benutzt:

$\{Regel(i_1, i_2, \ldots, i_n)$: IF ($x_1$ IS $N^1_{i_1,k_1}$) AND (x_2 IS $N^2_{i_2,k_2}$) AND $\ldots$ AND (x_n IS $N^n_{i_n,k_n}$) THEN y IS $Y_{i_1 i_2 \ldots i_n}\}$,

wobei

- x_j: Eingangsgröße j $(j = 1, \ldots, n)$,
- k_j: Ordnung der B-Spline-Basisfunktion für x_j,

- $N^{j}_{i_j,k_j}$: mit dem i-ten linguistischen Term für x_j assoziierte B-Spline-Funktion,

- $i_j = 0, \ldots, m_j$, Partitionierung von Eingang j,

- $Y_{i_1 i_2 \ldots i_n}$: Kontrollpunkte der $Regel(i_1, i_2, \ldots, i_n)$.

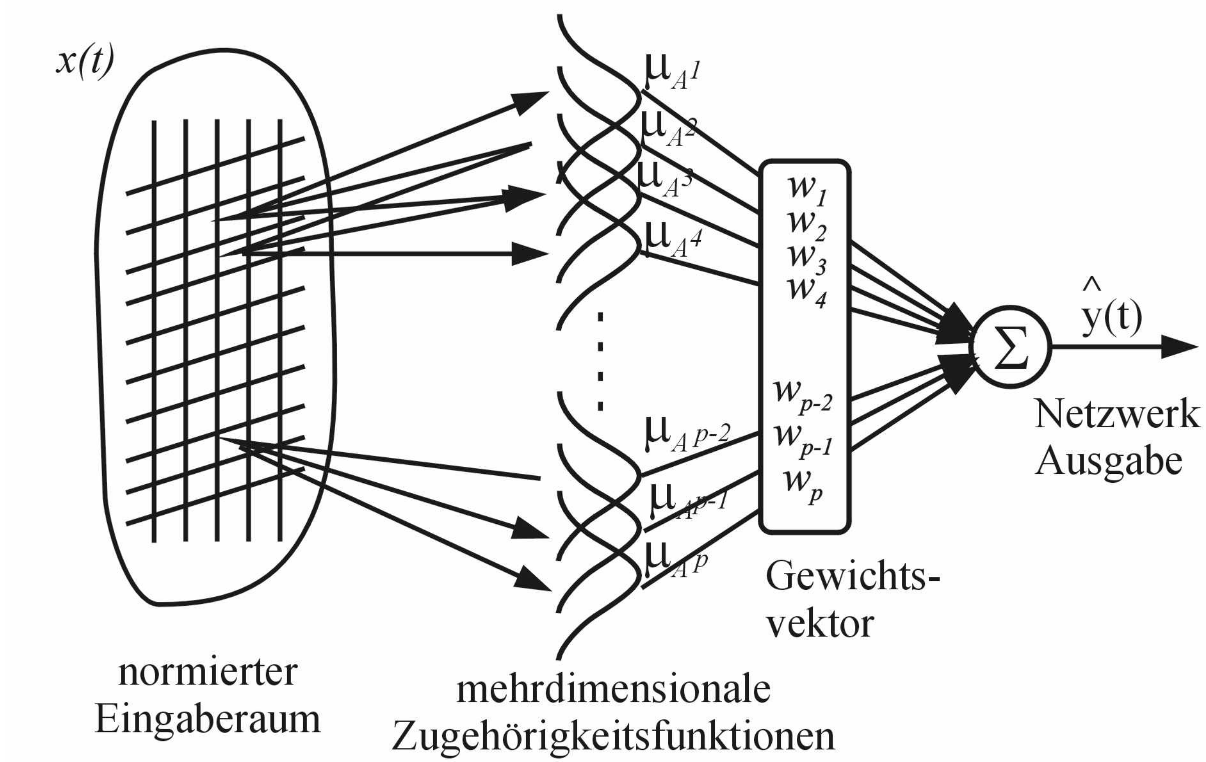
- der "AND"-Operator: Produkt

Dann ist der Ausgang y eines MISO Regelungssystems:

$$y = \sum_{i_1=1}^{m_1} \cdots \sum_{i_n=1}^{m_n} (Y_{i_1,\ldots,i_n} \prod_{j=1}^{n} N^j_{i_j,k_j}(x_j))$$

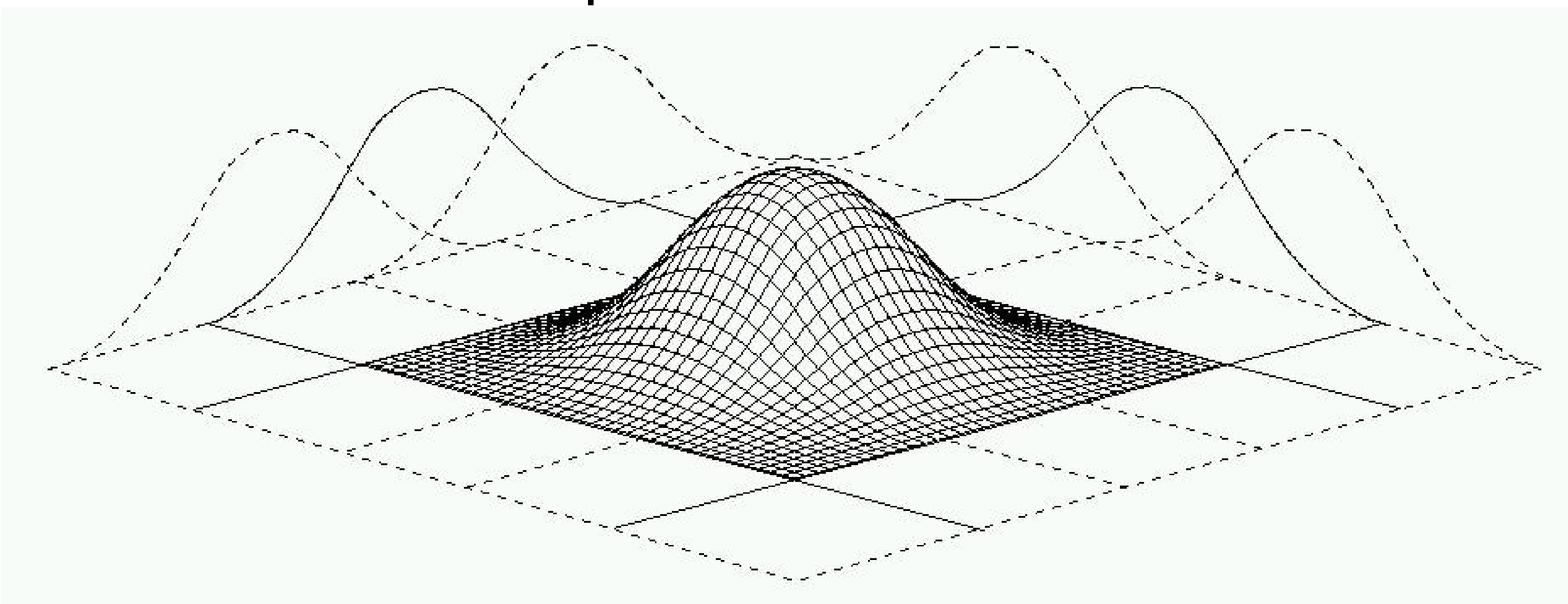
Das ist ein allgemeines B-Spline-Modell, das die Hyperfläche *NUBS (nonuniform B-spline)* darstellt.

Architektur des B-Spline ANFIS

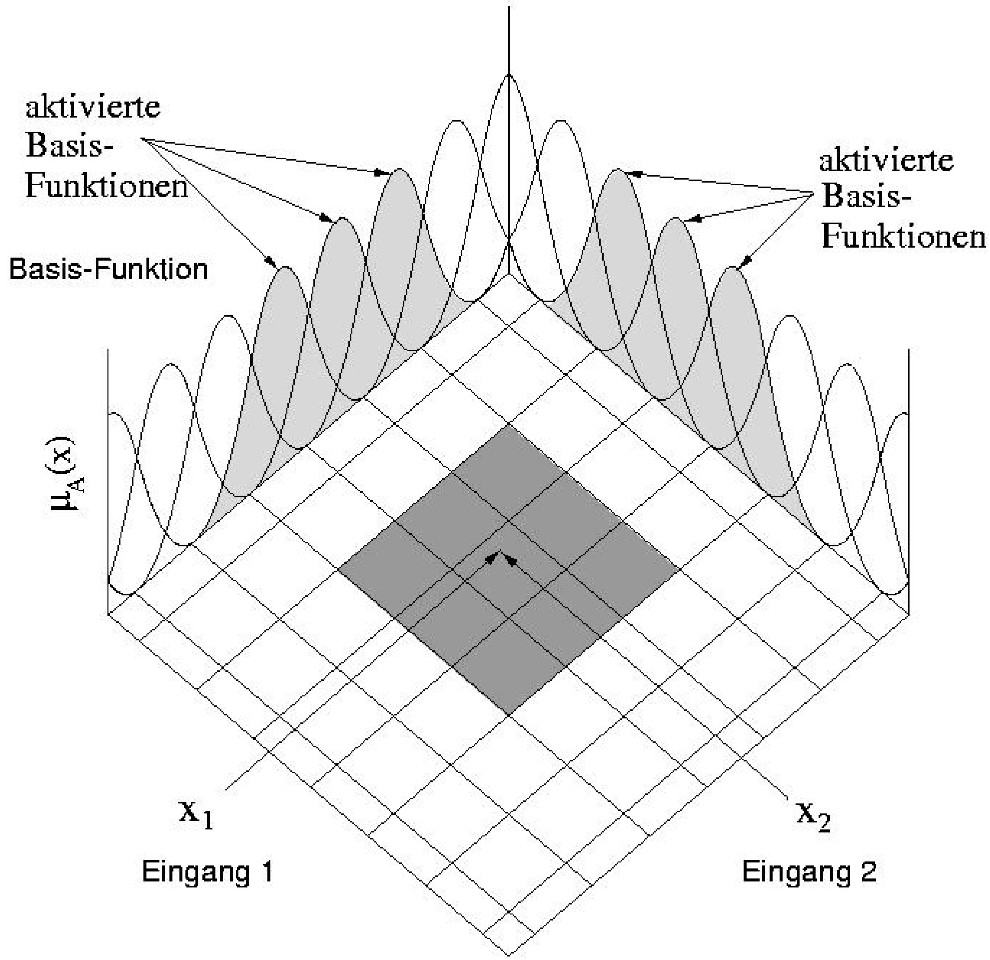


ZF-Formulierung - Tensorprodukt

Tensor-Produkt 2D-Splines:



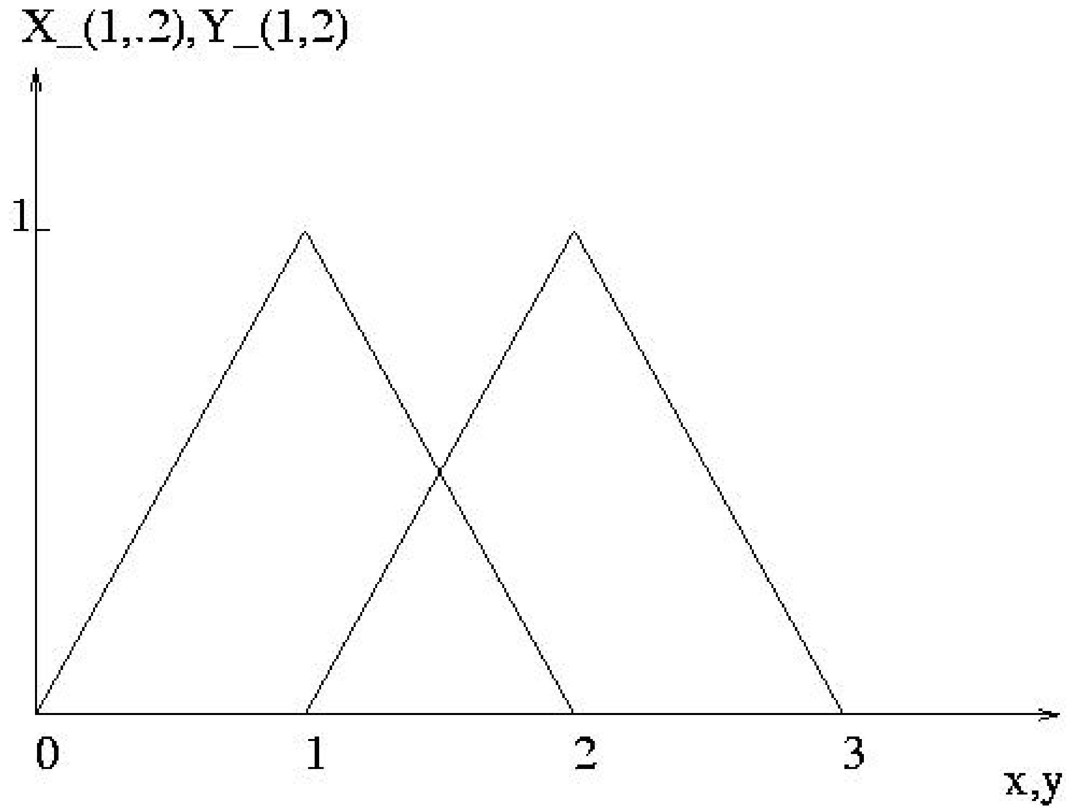
Die Aktivierung der ZF über den Eingang



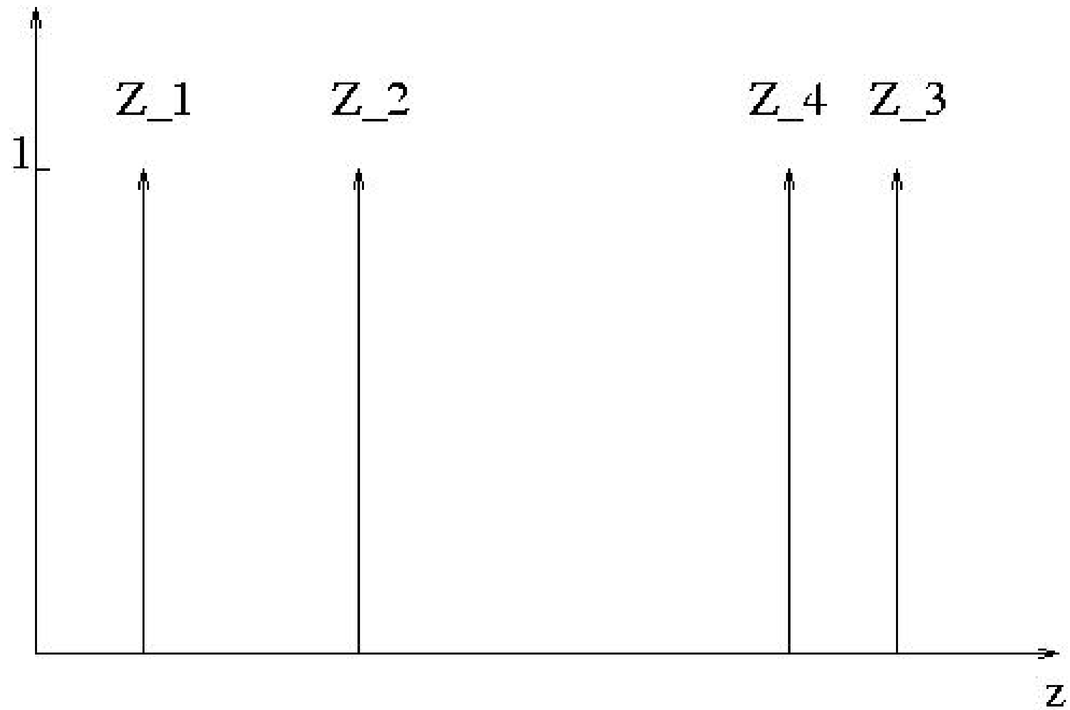
B-Spline ANFIS: ein Besipiel

Ein Beispiel mit zwei Eingangsvariablen (x und y) und einem Ausgang z. Die Parameter der DANN-Teile sind Z_1, Z_2, Z_3, Z_4.

Die linguistischen Terme der Eingänge (WENN-Teile):



Die Parameter der DANN-Teile:

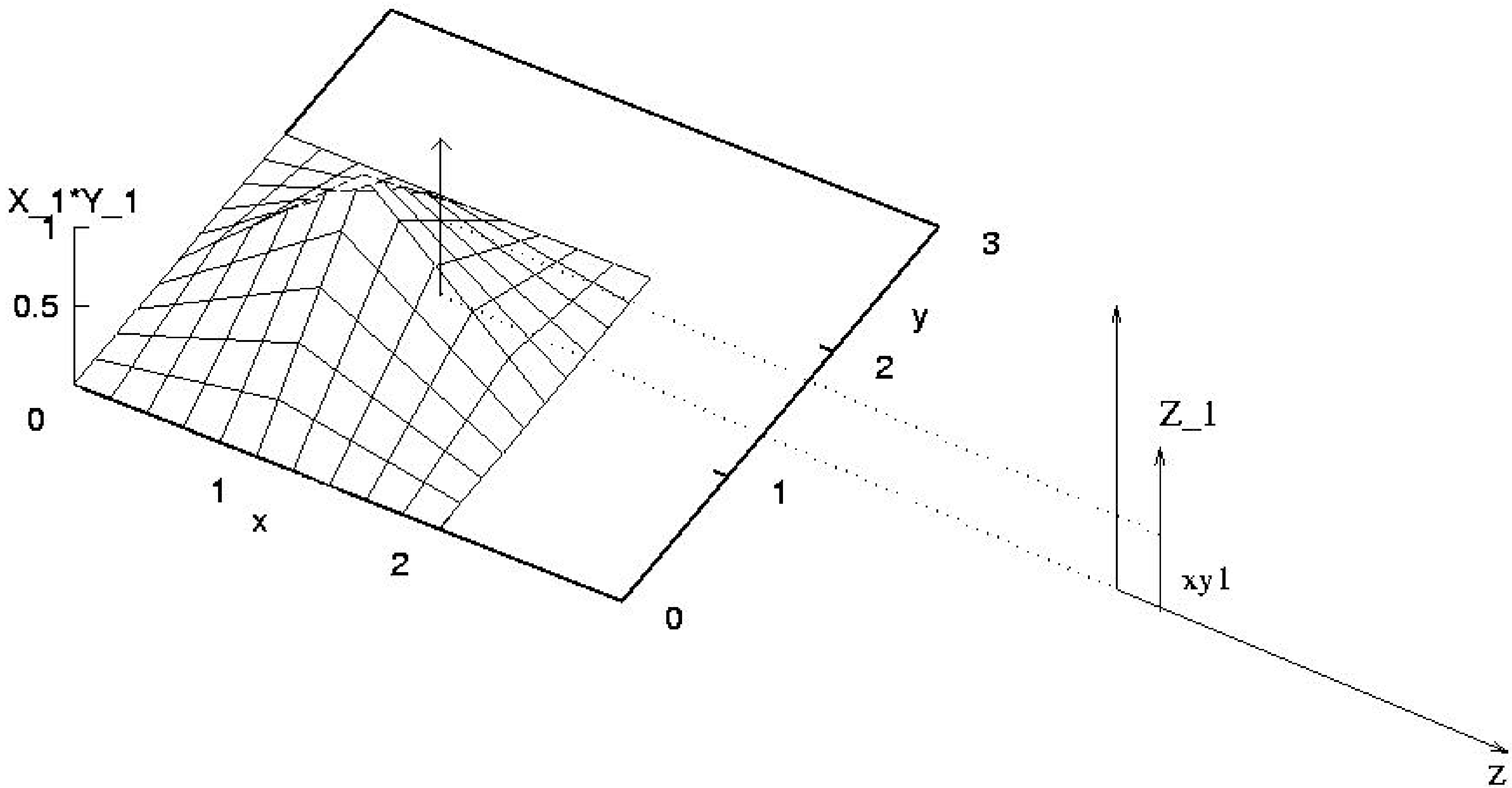


Ein Beispiel-Regelbasis

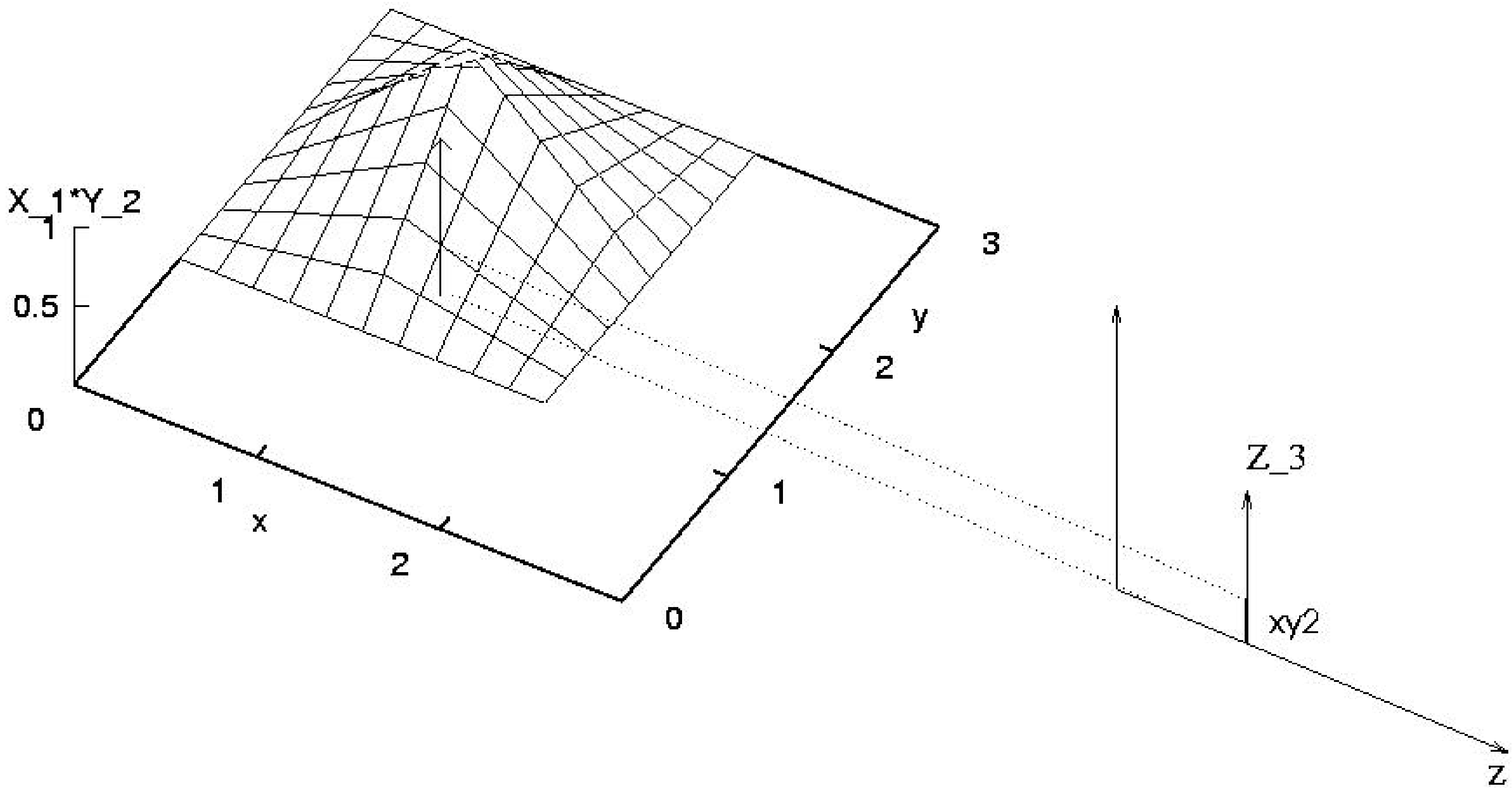
Die Beispiel-Regelbasis besteht aus vier Regeln:

Regel									
1)	IF	x is	X_1	and	y is	Y_1	THEN	z is	Z_1
2)	IF	x is	X_1	and	y is	Y_2	THEN	z is	Z_2
3)	IF	x is	X_2	and	y is	Y_1	THEN	z is	Z_3
4)	IF	x is	X_2	and	y is	Y_2	THEN	z is	Z_4

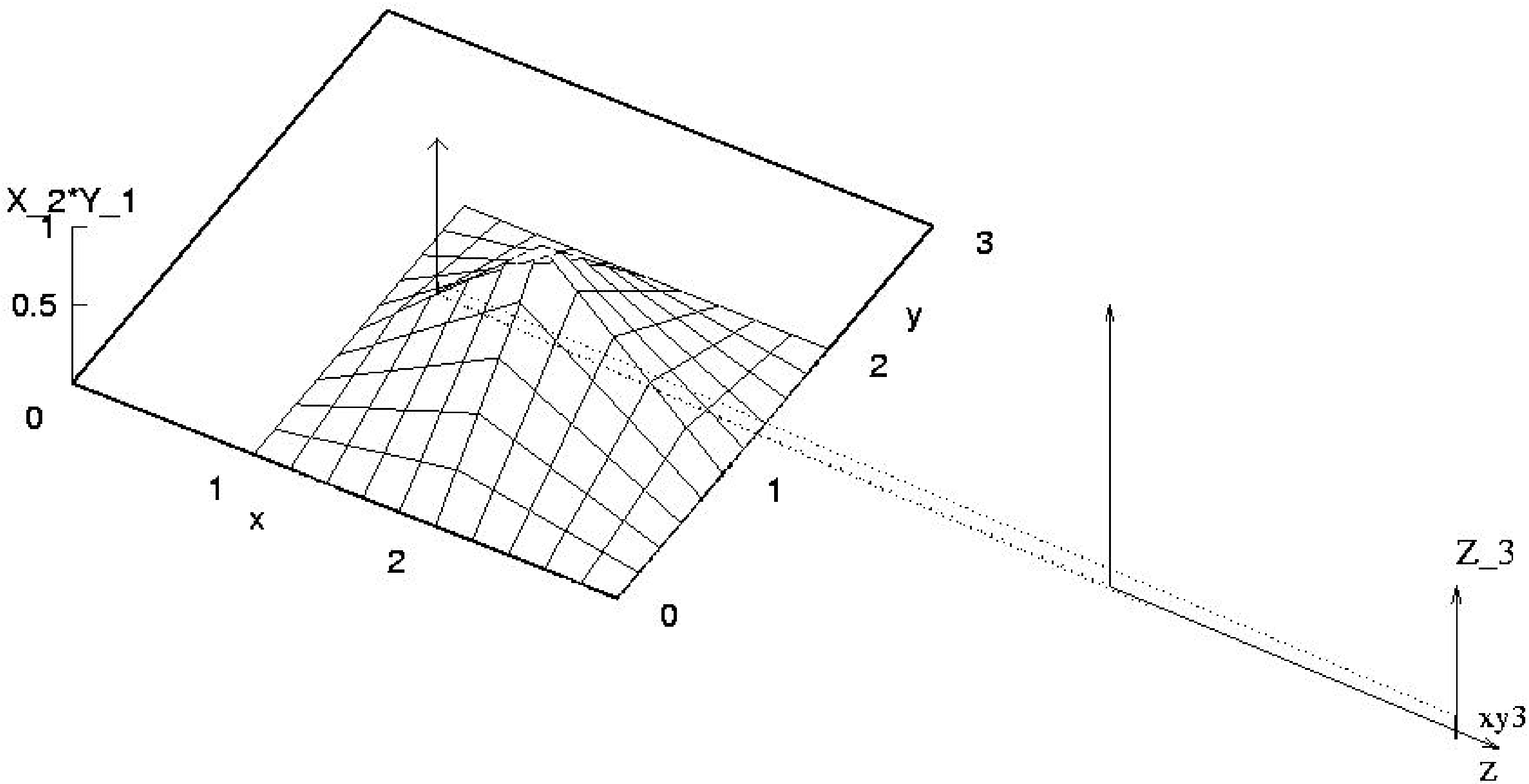
Illustrierung der Fuzzy-Inferenz



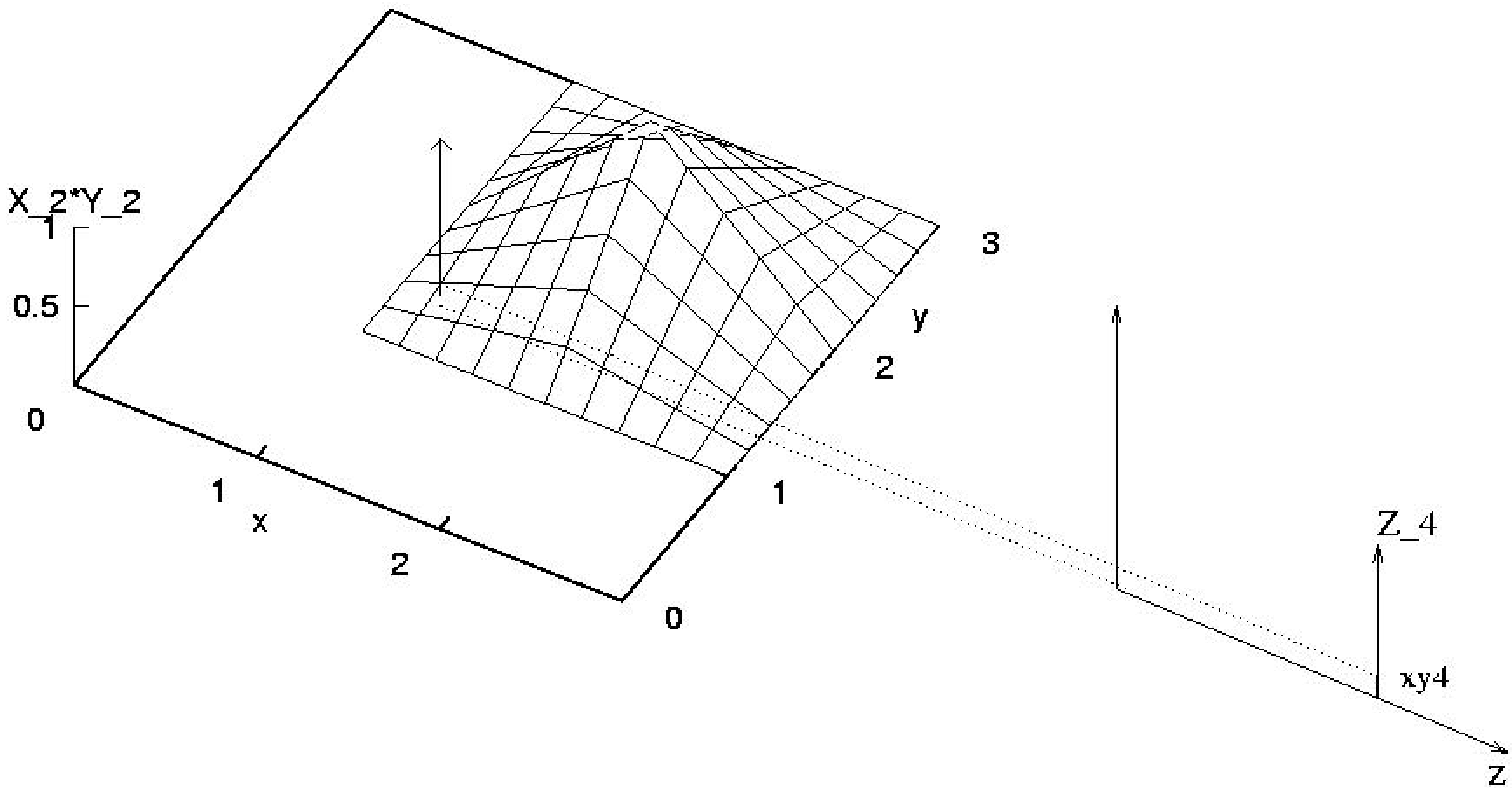
IF (x is X_1) and (y is Y_1) THEN z is Z_1



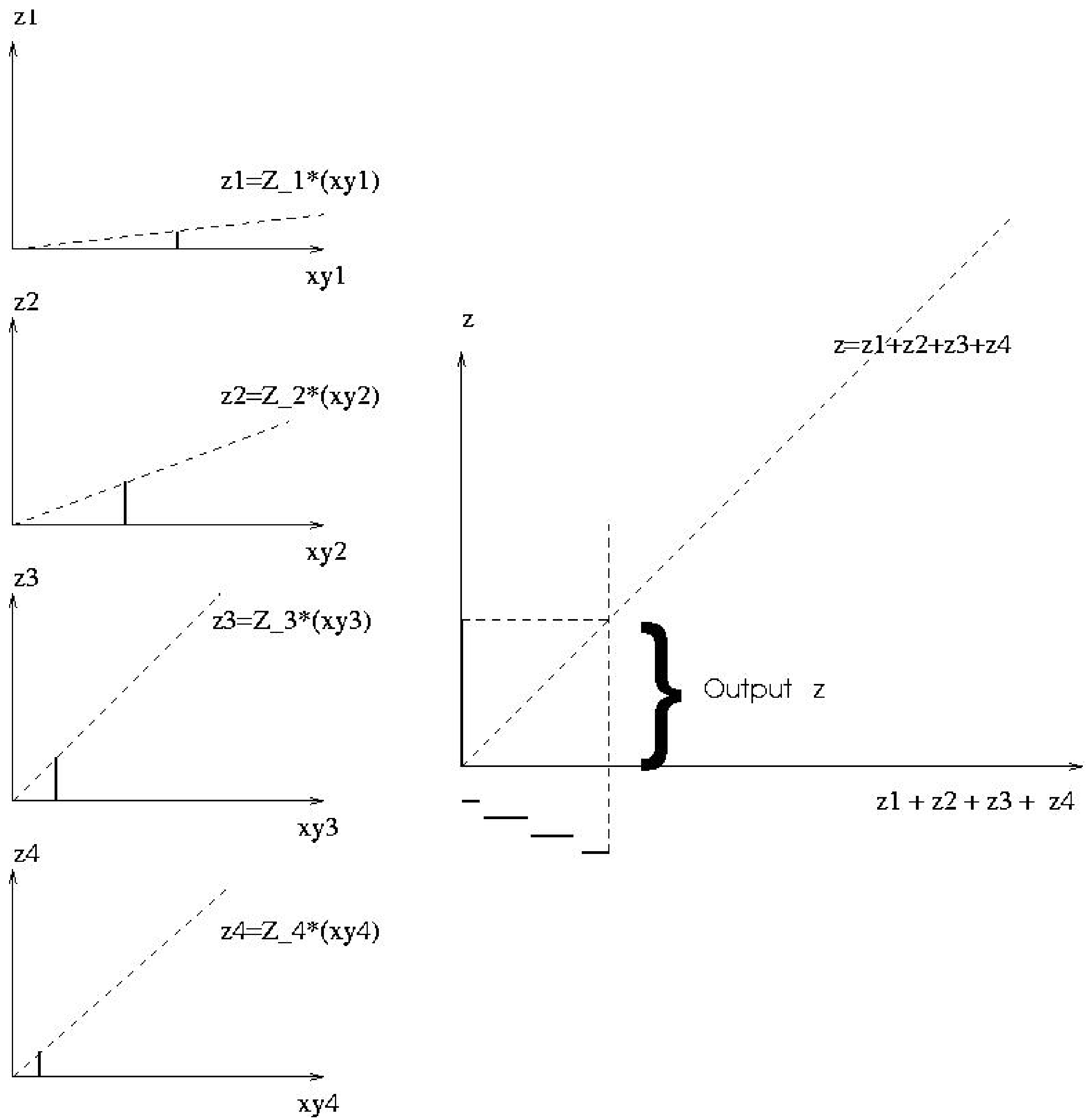
IF (x is X_1) and (y is Y_2) THEN z is Z_2



IF (x is X_2) and (y is Y_1) THEN z is Z_3



IF (x is X_2) and (y is Y_2) THEN z is Z_4


z1
z1=Z_1*(xy1)
xy1
z2
z2=Z_2*(xy2)
xy2
z3
z3=Z_3*(xy3)
xy3
z4
z4=Z_4*(xy4)
xy4
z
z=z1+z2+z3+z4
Output z
z1 + z2 + z3 + z4

Algorithm zum überwachten Lernen - I

Angenommen sei $\{(\mathbf{X},\ y_d)\}$ eine Menge von Trainingsdaten, wobei

- $\mathbf{X} = (x_1, x_2, \ldots, x_n)$: der Vektor der Eingangsdaten,
- y_d : der gewünschte Ausgang für $\mathbf{X}$.

Der LSE ist:

$$E = \frac{1}{2}(y_r - y_d)^2, \tag{2}$$

wobei y_r der aktuelle reale Ausgangswert während des Tranings ist.

Die zu findenen Parameter sind $Y_{i_1,i_2,\ldots,i_n}$, die den Fehler in (2) minimiert, d.h.

$$E = \frac{1}{2}(y_r - y_d)^2 \equiv \text{ MIN.} \tag{3}$$

Algorithm zum überwachten Lernen - II

Jeder Controlpunkt $Y_{i_1,\ldots,i_n}$ kann über das folgende Gradientabstiegsverfahren verbessert werden:

$$\Delta Y_{i_1,\ldots,i_n} = -\epsilon \frac{\partial E}{\partial Y_{i_1,\ldots,i_n}} \tag{4}$$

$$= \epsilon(y_r - y_d) \prod_{j=1}^{n} N^j_{i_j,k_j}(x_j) \tag{5}$$

wobei $0 < \epsilon \leq 1$.

Das Gradientabstiegsverfahren gewährleistet, dass der Lernalgorithmus zum globalen Minimum der LSE-Funktion konvergiert, weil die 2. partielle Ableitung bezüglich zu $Y_{i_1,i_2,\ldots,i_n}$ konstant ist:

$$\frac{\partial^2 E}{\partial^2 Y_{i_1,\ldots,i_n}} = (\prod_{j=1}^{n} N^j_{i_j,k_j}(x_j))^2 \geq 0. \qquad (6)$$

Dies bedeutet dass die LSE-Funktion (2) konvex im Raum $Y_{i_1,i_2,\ldots,i_n}$ ist und deshalb nur einen (globalen) Minimum besitzt.

Funktionsapproximation - Demonstrationen

$sin(x^2)$ Stop 1d-demo

$sin(x^2y)$ Stop 2d-demo