Reichweite der Schulmethode ?

notwend. Bed. — 1. Ableitung bilden und deren Nullstelle(n) bestimmen

hinr. Bed.? — 2. Ableitung bilden und deren Vorzeichen an Nullstelle(n) bestimmen
: >0 für Min , <0 für Max , $=0$?

k-te Abl. bilden, bis an Nullstelle verschieden von Null; Fallunterscheidung

$k \geq 1$ gerade $\quad f^{(k)}(x^*) > 0$ Min
$\quad\quad\quad\quad\quad\quad\quad\quad < 0$ Max

$k \geq 1$ ungerade Sattelpunkt

hinreichende Bed. für $n = 1$ [Mac Laurin, 1742]
für $n > 1$ [Scheefer & Stolz 1886 & 1893]

Voraussetzung: $f(x)$ k mal (partiell) differenzierbar d.h. analytisch
keine Nebenbedingungen

evtl. mehrere Lösungen : $f(x)$ multimodal
[Ggs: unimodal]

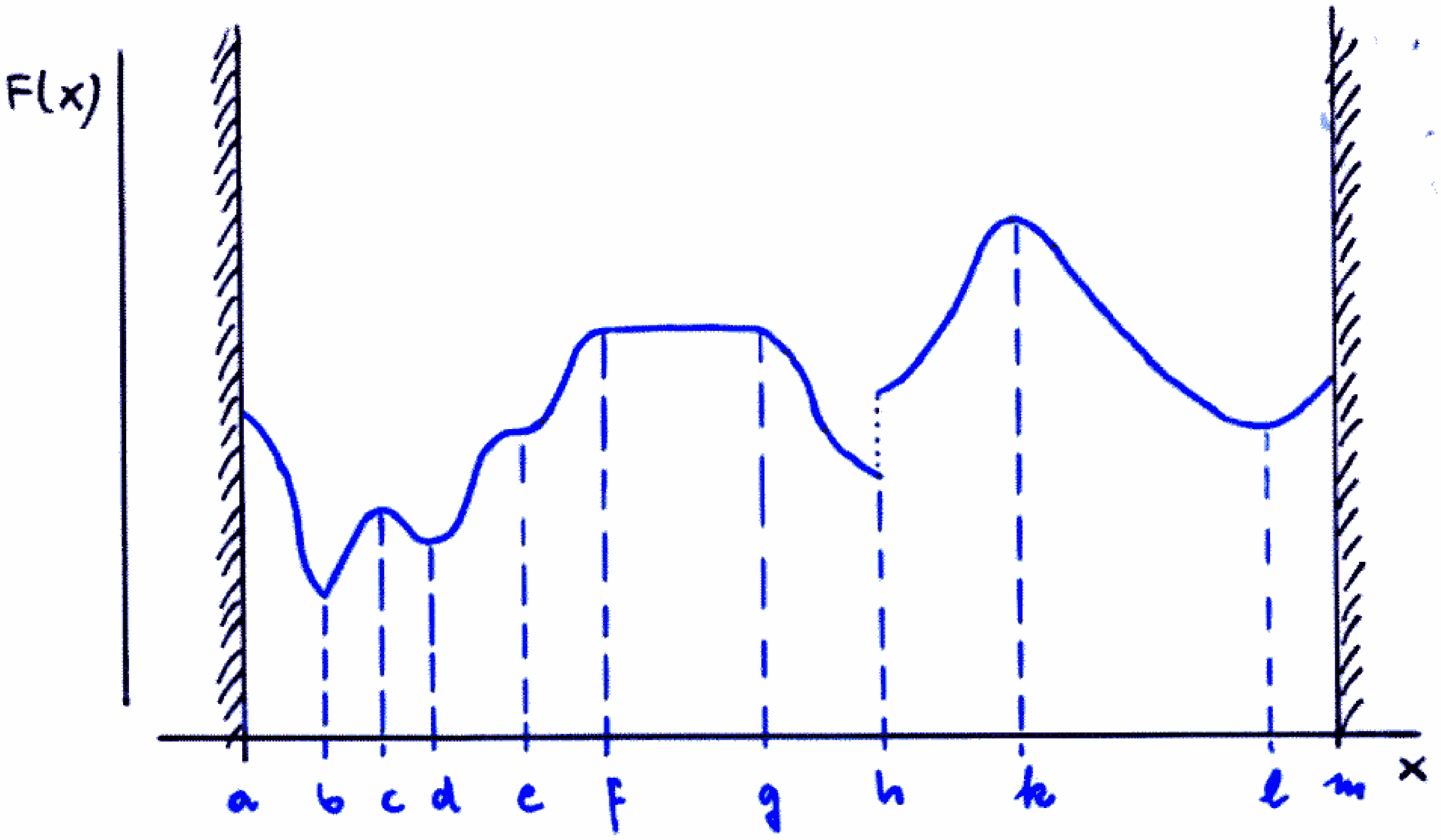
Nullstellensuche evtl. keine leichtere Aufgabe als Opt.

$n > 1$: $\nabla f(x^*) = 0$ als notwend. Bed.

Lösung eines simultanen Gleichungssystems
evtl. n nichtlineare Gl.
↓
Minimierung der Σ der quadrat. Abweichungen

Bsp. $f(x) = x_1^2 \sin\sqrt{x_1}\, e^{|x_1 - x_2|}$

Funktionenopt. ↪ System von Dgl. zu lösen



ausgezeichnete Stellen einer Funktion einer Variablen

Weierstraß: In einem abgeschlossenen Bereich $a \le x \le m$ hat jede dort stetige Fkt. mind. ein (absolutes) Minimum und Maximum

a lokales Max. am Rand

b lokales Min. an Unstetigkeitsstelle von $F'(x)$

c lokales Max. $(F'(x) = 0)$

d lokales Min. (")

e Sattel- bzw. Wendepunkt $(F' = 0)$

f, g } schwaches lokales Max. $(F' = 0)$

h lokales Min. an Unstetigkeitsstelle von $F(x)$

k lokales Max.

l lokales Min.

m lokales Max. am Rand

---

b globales Min.

k globales Max.

Gleichungssystem → Optimierungsaufgabe

z. B. $3x_1 - x_2 = 1$
$x_1 - 3x_2 = -13$

$3x_1 - x_2 - 1 = r_1(x_1, x_2) \stackrel{!}{=} 0$
$x_1 - 3x_2 + 13 = r_2(x_1, x_2) \stackrel{!}{=} 0$

Gauß'sche Approximation:
$F = r_1^2 + r_2^2 \rightarrow \min$ Höhenlinien: Ellipsen

Tschebyscheff'sche Appr.:
$F = \max\{r_1, r_2\} \rightarrow \min$ Höhenlinien: Rhomben

↑ lin. Gl. system n Unbek., m Gleichungen
wenn $n \neq m$: Approximation obligatorisch

↓ nichtlin. Gl. system $f_1(x_1, x_2) = 0$
evtl. mehrere Lösungen $f_2(x_1, x_2) = 0$
bei: Approximation zusätzlich (relative) Minima, die keine Lösung des Gl. syst. sind

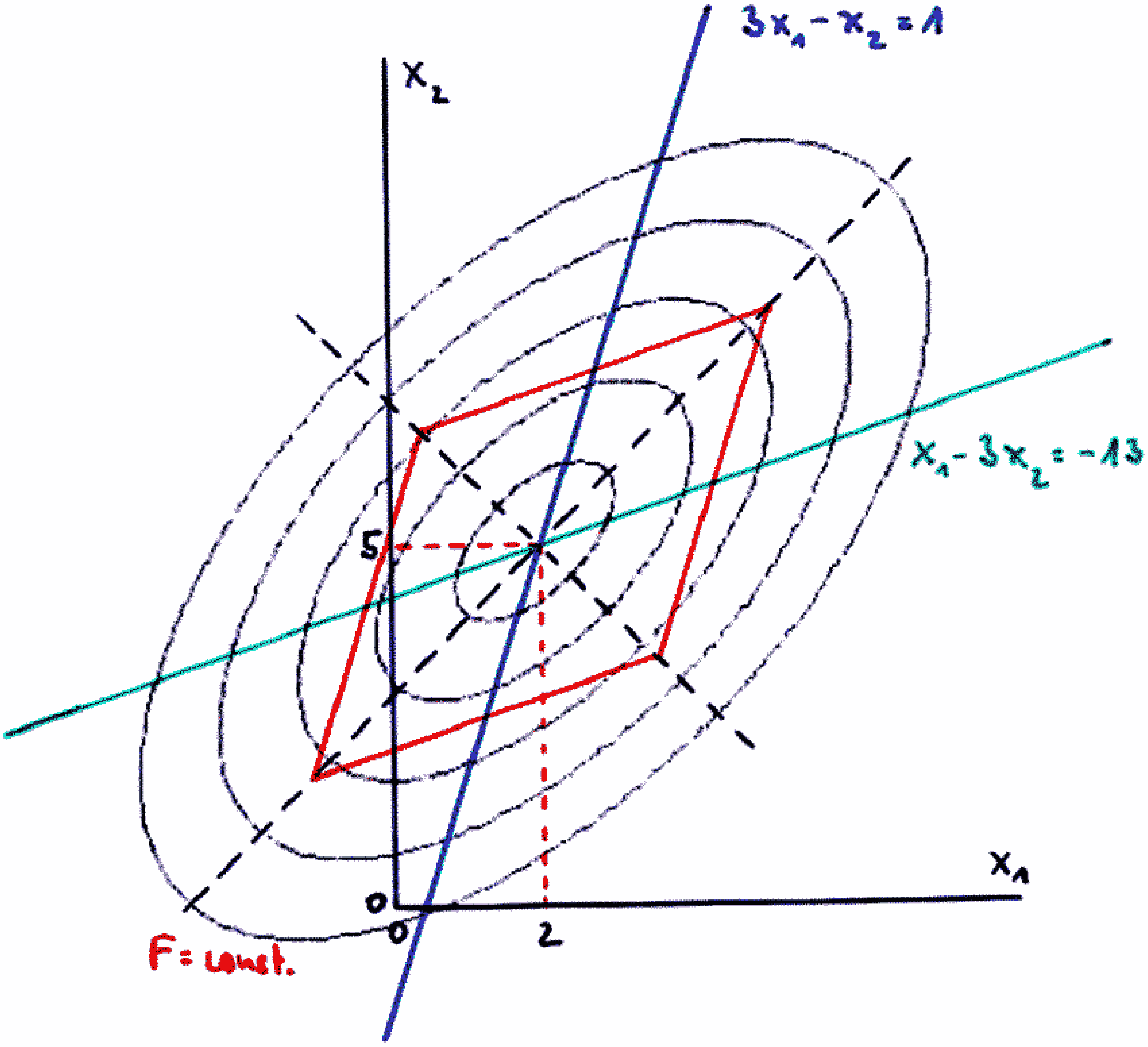
spezielle Verfahren für Quadratsummenminimierung

Gauß-Newton
Levenberg
Marquardt
Powell

$\Bigg\}$ verlangen Jacobische Matrix explizit

$$J = \left\{ \frac{\partial f_i}{\partial x_j} \; \forall\, i,j = 1 \dots n \right\}$$

Lineares Gleichungssystem

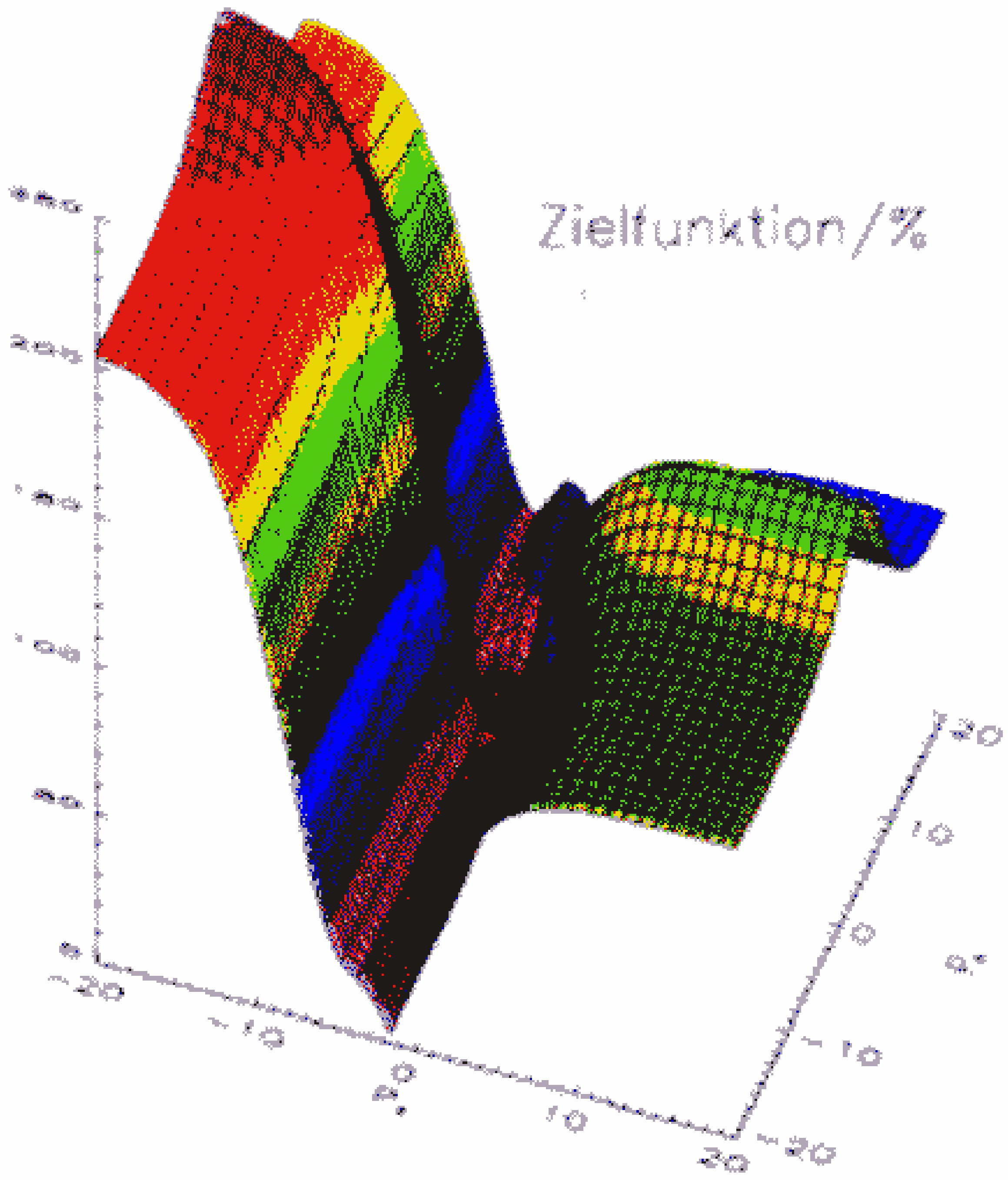


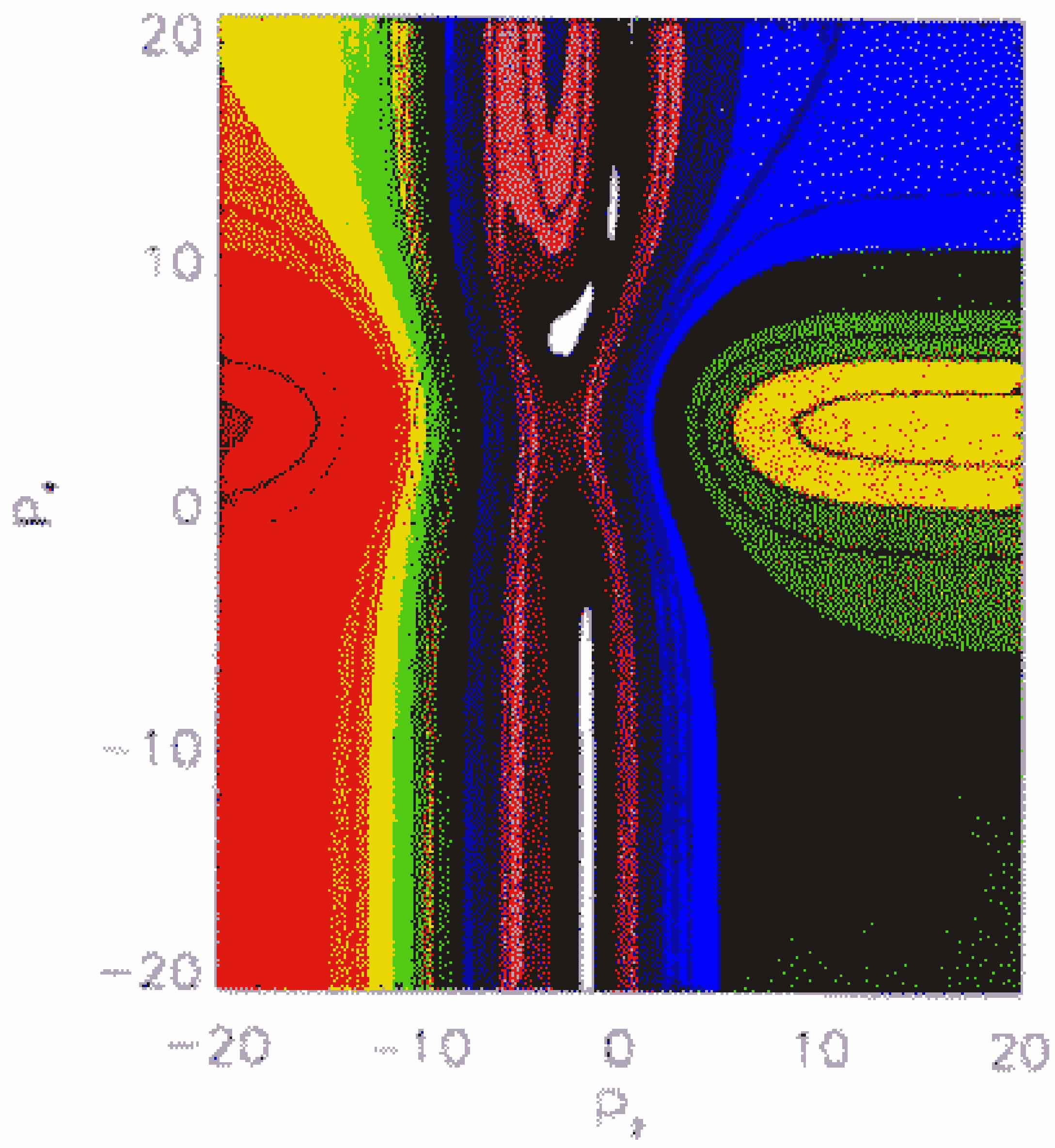
$$F = (3x_1 - x_2 - 1)^2 + (x_1 - 3x_2 + 13)^2 \quad \rightarrow \text{min.}$$

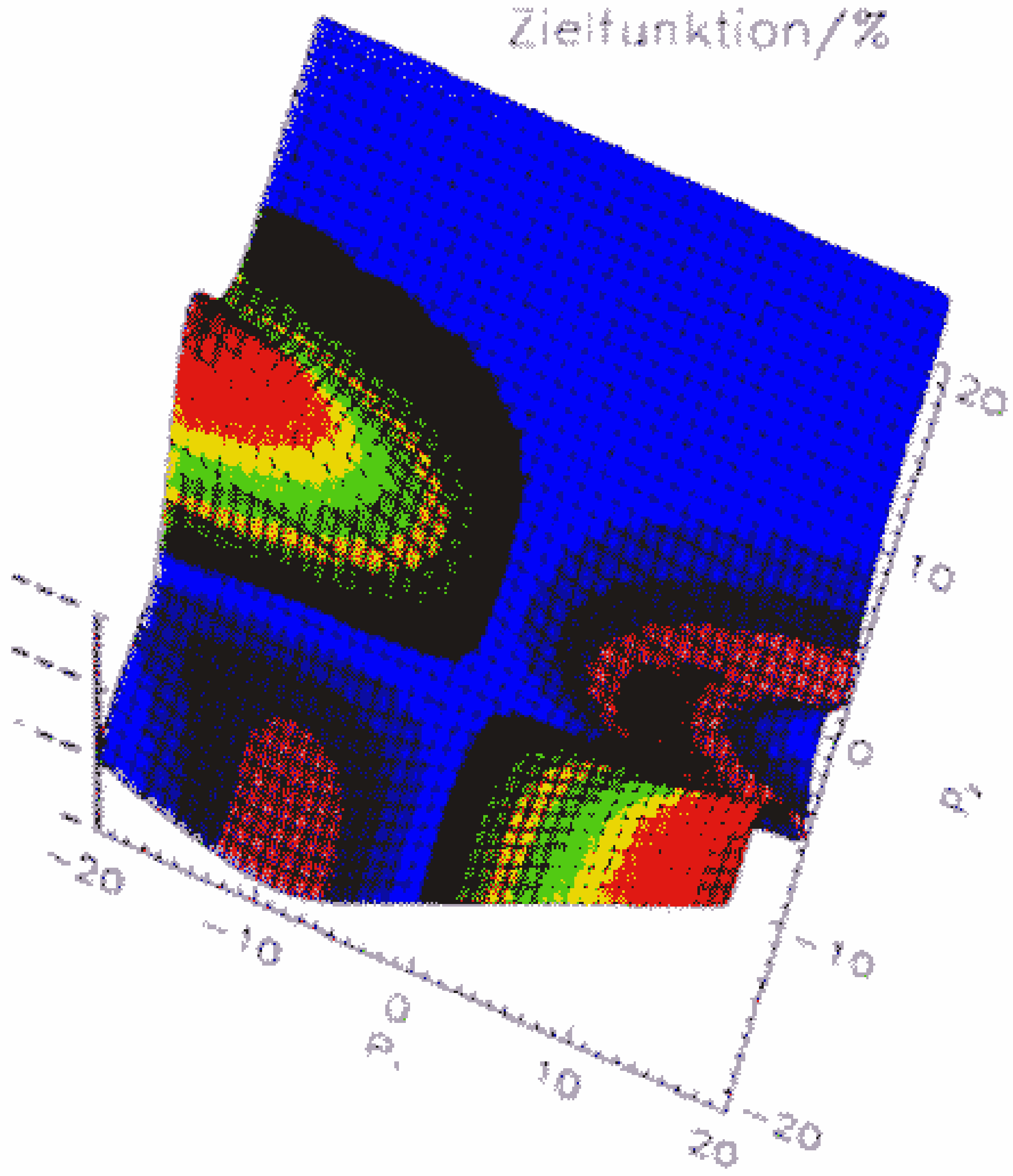
Gauß'sche Approximation

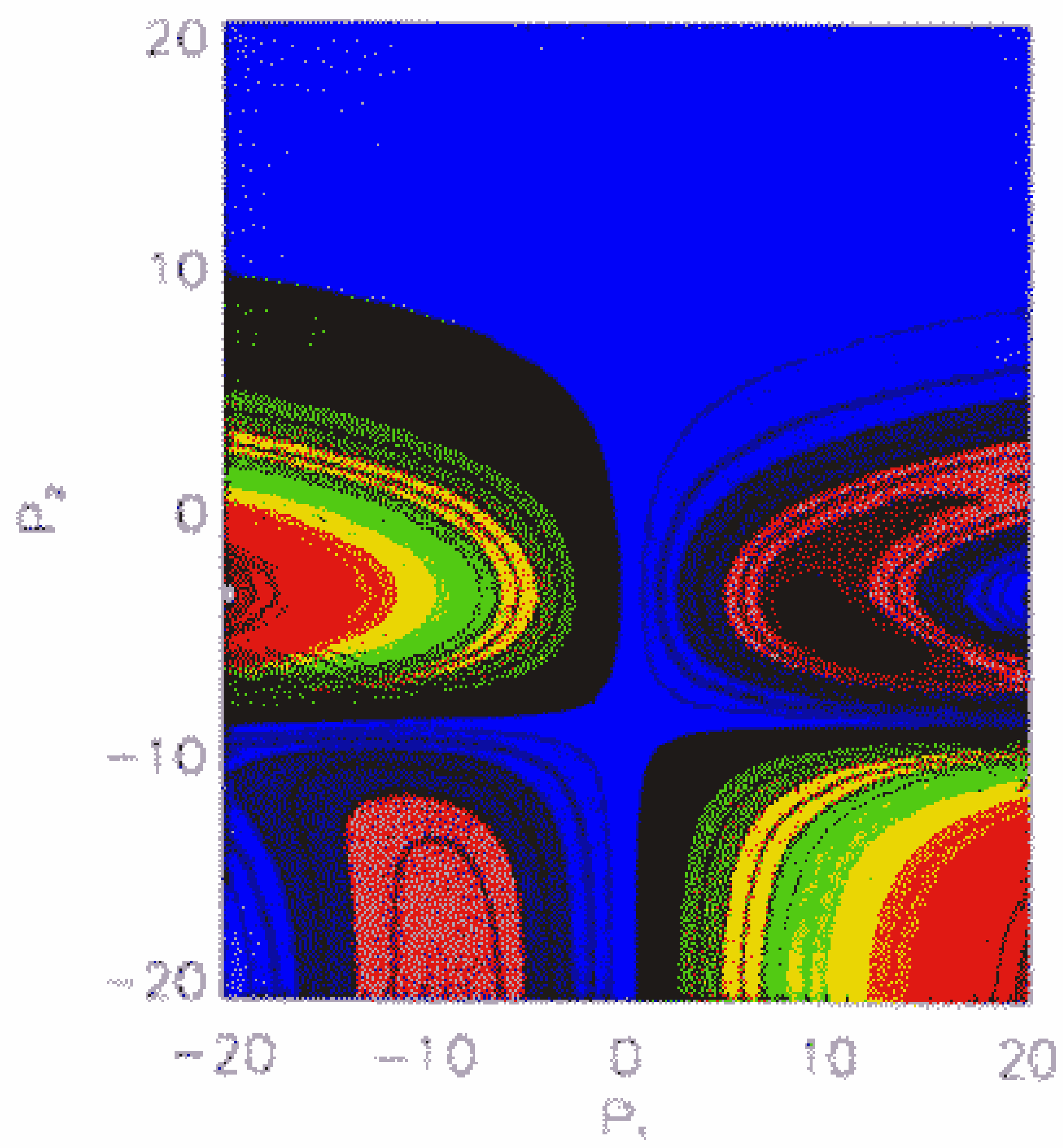
Tschebyscheff'sche Approximation:

$$F' = \max\left\{ |3x_1 - x_2 - 1| \; ; \; |x_1 - 3x_2 + 13| \right\} \rightarrow \text{min.}$$


Zielfunktion/%
−20
−10
0
10
20
−20
−10
0
10
20


20
10
0
−10
−20
−20
−10
0
10
20


Zielfunktion/%
−20
−10
0
10
20
−20
−10
0
10
20


20
10
0
−10
−20
−20
−10
0
10
20

Funktionenoptimierung

gesucht: $x(t)$ bzw. $\{x_1(t), x_2(t), \ldots, x_n(t)\}$

oftmals Zeit als 'Parameter'

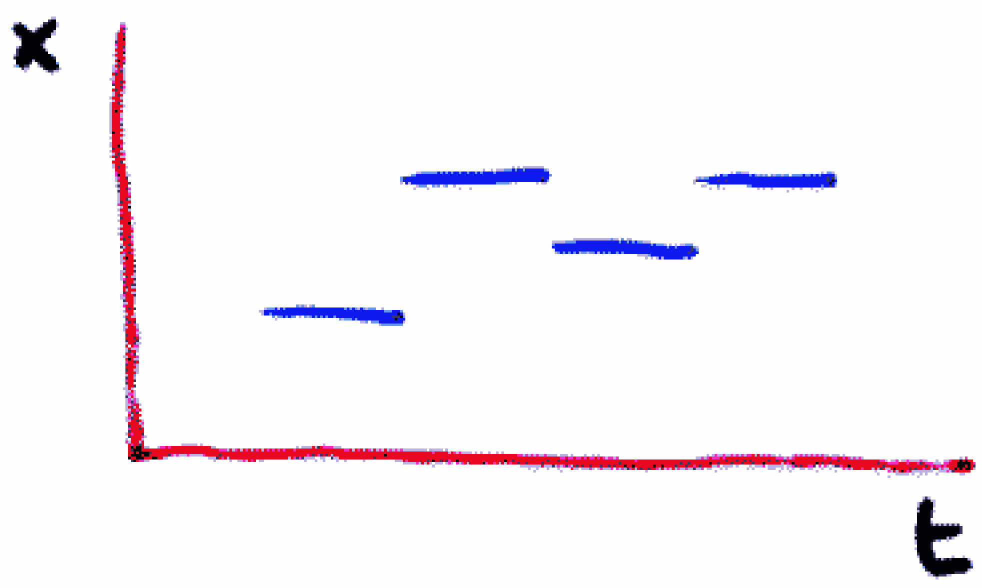
zu optimieren: $F(x(t))$ Funktionenfunktion
Zielfunktional

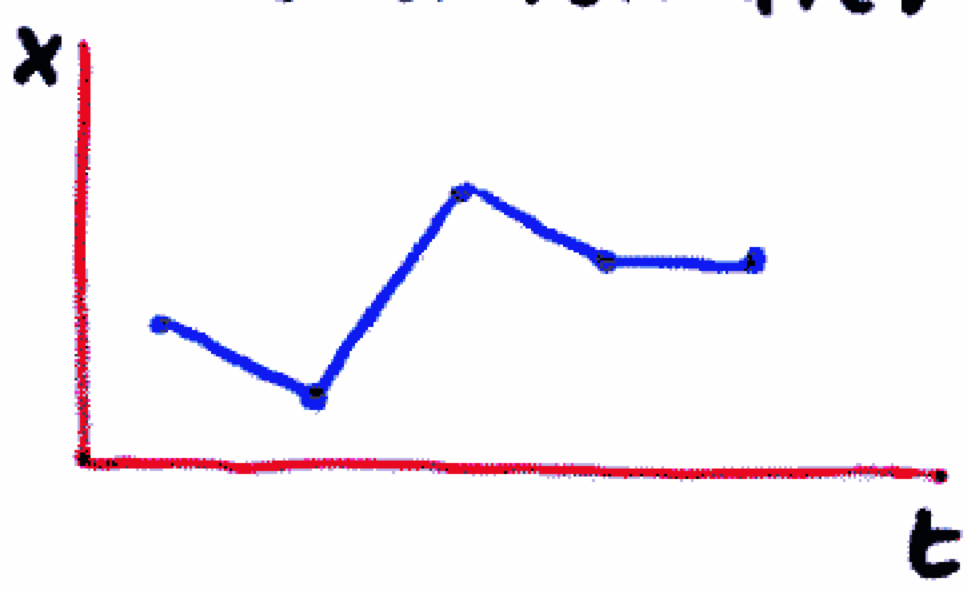
ggf. Nebenbedingungen $\dot{y}(t) = g(y(t), x(t))$

z.B. $F(x) = \int_{t_1}^{t_2} f(x(t), \dot{x}(t), t)\, dt \rightarrow$ Extr.

siehe: Variationsrechnung, Kontrolltheorie
Steuerung + Regelung

gut mit Analog- u. Hybridrechnern zu lösen
auf Digitalrechnern: Diskretisierung
Parametrisierung





$x(t) = a + b \cdot e^{ct}$ oder Polynom in t ...

$\curvearrowright$ Parameter a, b, c gesucht

Funktionenoptimierung → Lösung einer Dgl.

z.B. $I(x) = \int_0^1 F(t, x(t), \dot{x}(t))\, dt \rightarrow$ Extr.

notwend. Bed. : Euler'sche Differentialgleichung

$$\frac{\partial F}{\partial x} - \frac{d}{dt}\frac{\partial F}{\partial \dot{x}} \overset{!}{=} 0$$

nichtlin. Dgl. → Optimierungsaufgabe

z.B. $\frac{dz}{dx} - z^2 = 1$ mit Randbedingung
$z(x=0) = 0$; $0 \le x \le 1$

Ansatz : $\bar{z} = a_1 + a_2 x + a_3 x^2 + \cdots$

$$\frac{d\bar{z}}{dx} = a_2 + 2a_3 x + \cdots$$

$$\bar{z}(x=0) = a_1$$

Term für Randbed.

$$\sum_{i=1}^{m} \left\{ \left.\frac{d\bar{z}}{dx}\right|_{x_i} - \left.\bar{z}^2\right|_{x_i} - 1 \right\}^2 + a_1^2 \rightarrow \text{Min}$$

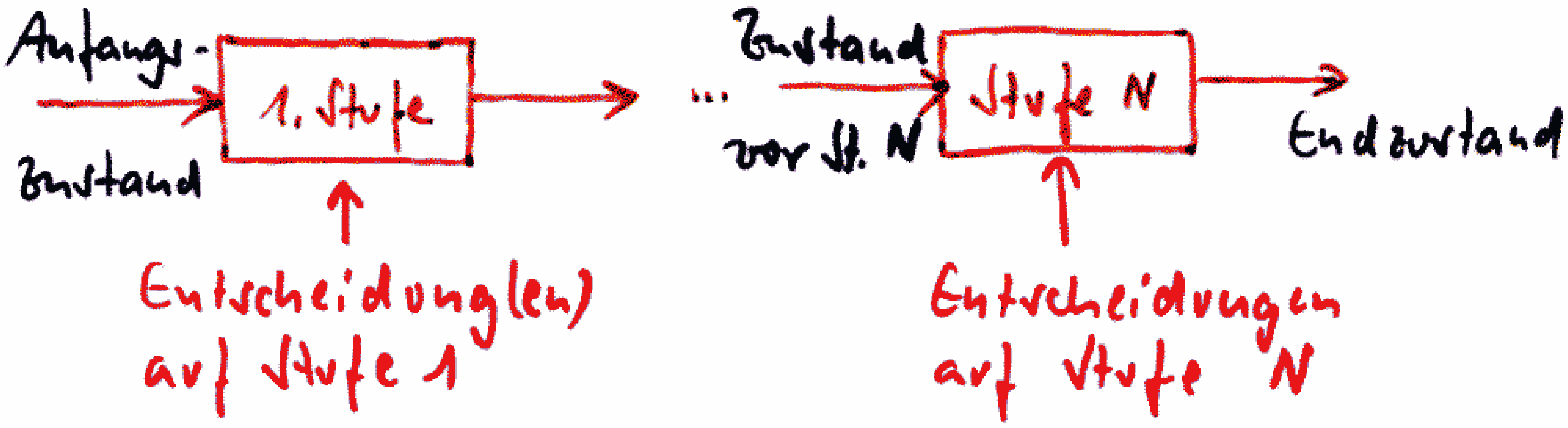
m äquidistante Stützstellen im Intervall

$\rightsquigarrow$

$F(a_1, a_2, \cdots) \rightarrow$ Min.

Spezialfall Stufenoptimierung

Dekomposition
dynamische Programmierung



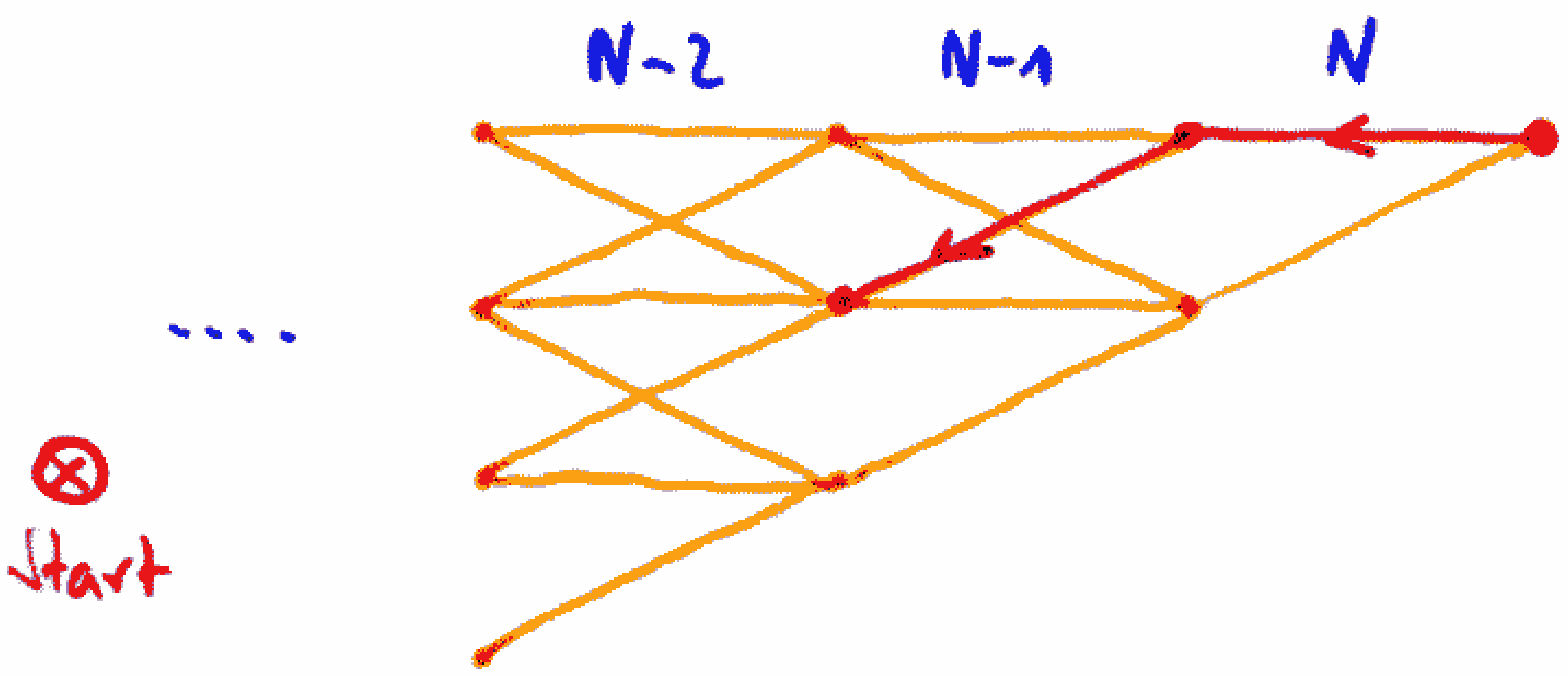
Bedingung: Rückwirkungsfreiheit

Folge: Jedes Teilsystem, welches Endzust. enthält, ist einzeln optimierbar

$F = f(\text{Endzustand})$

$\;\;= \hat{f}(\text{Zustand } N-1,\ \text{Entscheidung } N)$ .....

also Optimierung sukzessive vom Endzustand her (meist für diskrete, determinist. Entsch. prozesse)



Stufenoptimierung ist für Parameteropt. anwendbar, wenn Zielfunktion separierbar,

z. B. $F(x) = \sum_{i=1}^{N} f_i(x_i)$

Aufwand : $N \cdot b^{m+l}$

N = Stufenzahl

b = Zahl der diskreten Parameterwerte

m = Zahl der pro Stufe beteiligten Entscheid. variablen

l = Zahl der Nebenbedingungen

Bsp. 1 $F = x_1 + x_2$

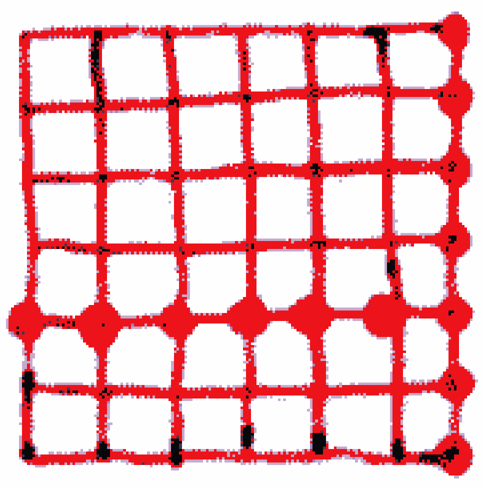
$N = 2$

$m = 1$

$l = 0$

$b = 7$

$2 \cdot 7$ Versuche



Bsp. 2 $F = x_1 \cdot x_2$

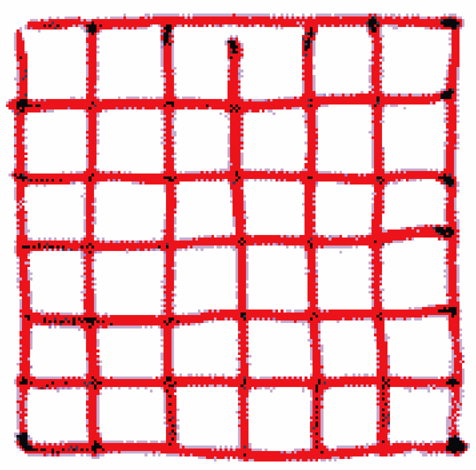
$N = 1$

$m = 2$

$l = 0$

$b = 7$

$1 \cdot 7^2$ Versuche



alle Gitterpunkte testen

≙ totale Enumeration

Behandlung von Nebenbedingungen

a) in Form von Gleichungen

$$f(x) \to \min \mid g_j(x) = 0 \quad \forall j = 1(1)m$$

a1) Elimination / Reduktion der Freiheitsgrade

z. B. $x_1 x_2 \to \min$

$x_1 + x_2 = 1$

$x_1(1 - x_1) \to \min$

a2) Lagrange'sche Multiplikatoren

$$F(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x) \to \text{Sattelpunkt}$$

$$\frac{\partial F}{\partial x_i} = 0 \quad \forall i = 1(1)n$$

$$\frac{\partial F}{\partial \lambda_j} = 0 \quad \forall j = 1(1)m$$

↷ (nicht-)lineares Gleich. system

b) in Form von Ungleichungen

b1) explizite Grenzen : Transformation

$x_i \geq 0$ Substitution $x_i = y_i^2$

$x_i = e^{y_i}$

$0 \leq x_i \leq 1$ Subst. $x_i = \sin^2 y_i$

$$x_i = \frac{e^{y_i}}{e^{y_i} + e^{-y_i}}$$

$a_i \leq x_i \leq b_i$ Subst. $x_i = a_i + (b_i - a_i) \sin^2 y_i$

b2) implizite Grenzen : Straffunktion
penalty function

$f(x) \to \min$

$g_j(x) \geq 0$

$h_k(x) = 0$

$$F(x, r^{(p)}) = f(x) + r^{(p)} \sum_{j=1}^{m} \frac{w_j}{g_j(x)} + \frac{1}{r^{(p)}} \sum_{k=1}^{l} v_k h_k^2(x)$$

$p = 1, 2, \ldots$
wobei: $r^{(p)} > r^{(p+1)} > 0$

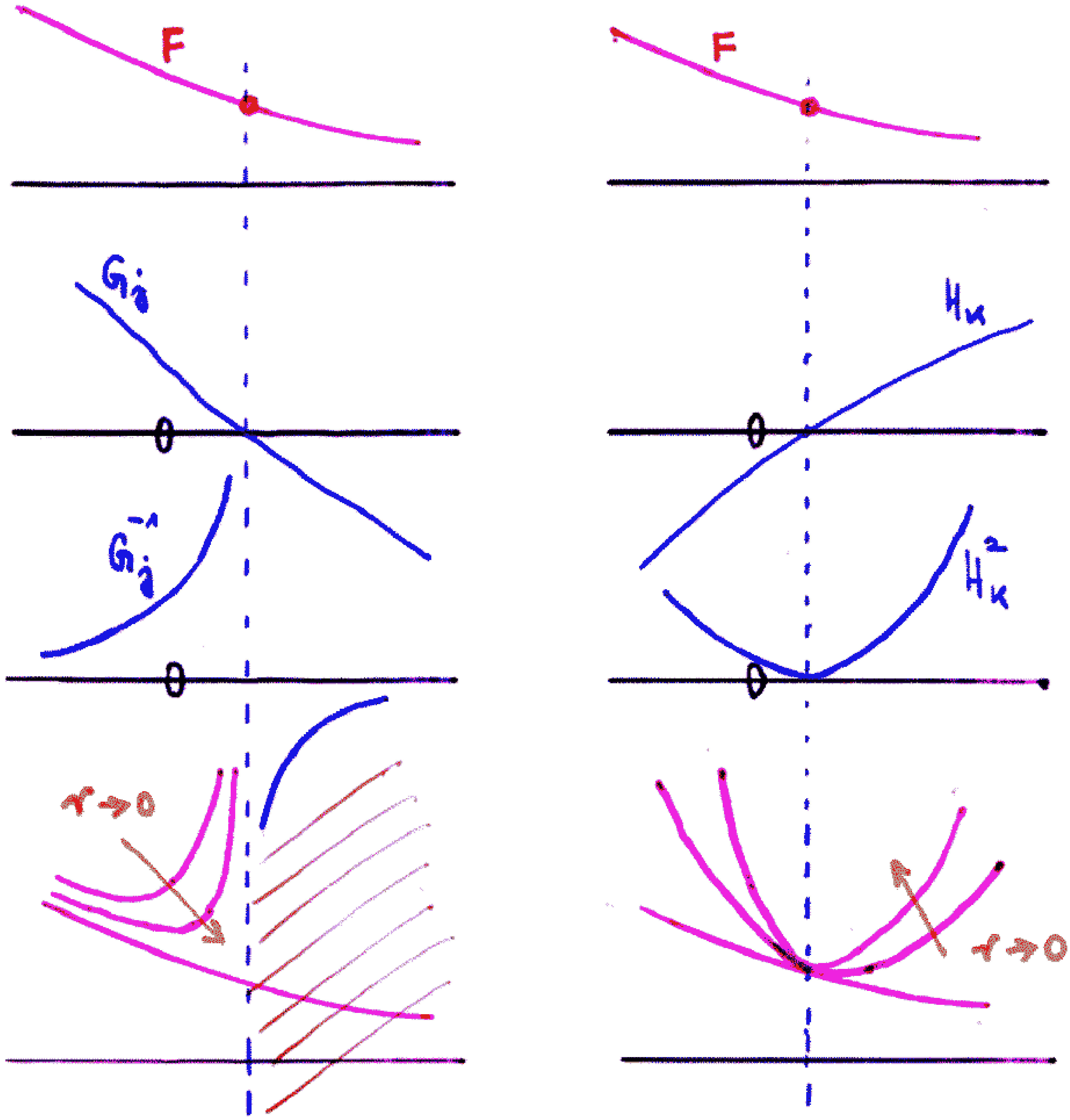
SUMT sequential unconstrained minimization technique (Fiacco & McCormick)

b3) <u>oder</u> : bei Verletzung von (Ungleichungs-) Restriktionen Lösung verwerfen, neuer Versuch (bei ‚trial and error' Verfahren)

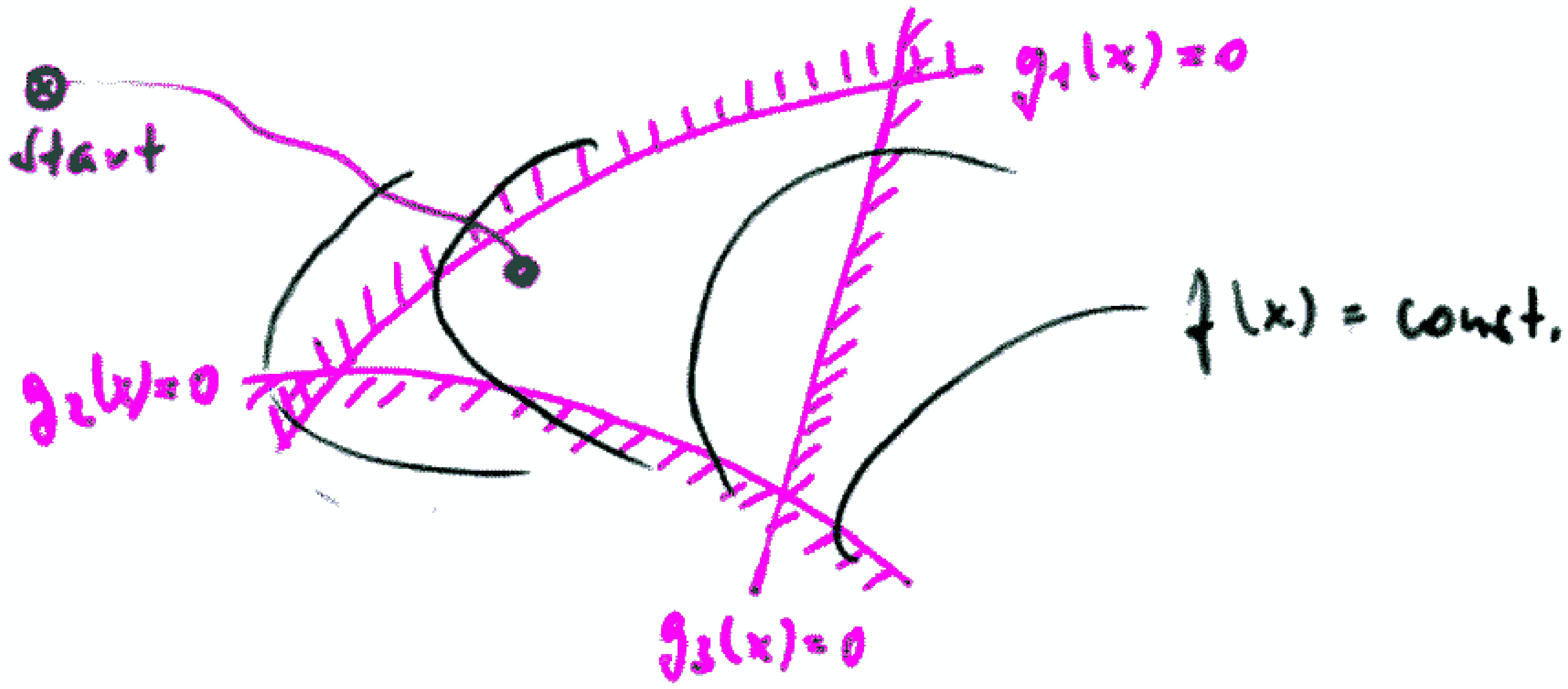
SUMT Sequential Unconstrained Minimization Technique
(Fiacco & McCormick)

$$F(x) \to \min \; ; \quad G_j(x) \geq 0 \; ; \quad H_k(x) = 0$$
$$j = 1(1)m \qquad k = 1(1)l$$

$$F'(x) = F(x) + r \sum_{j=1}^{m} \frac{w_j}{G_j(x)} + \frac{1}{r} \sum_{k=1}^{l} v_k H_k^2(x)$$



Spezialfall: Suche nach zulässiger Lösung



$$F'(x) = -\sum w_j \, g_j(x) \quad \to \min$$

$$\text{wobei} \quad w_j = \begin{cases} 1 & \text{wenn } g_j(x) < 0 \\ 0 & \text{sonst} \end{cases}$$

Stop, sobald $F'(x) = 0$

dann Fortsetzung mit einem Opt.-verfahren, welches Ungleichungen einhalten kann, mehrfache Zielsetzung

MCDM multiple criteria decision making

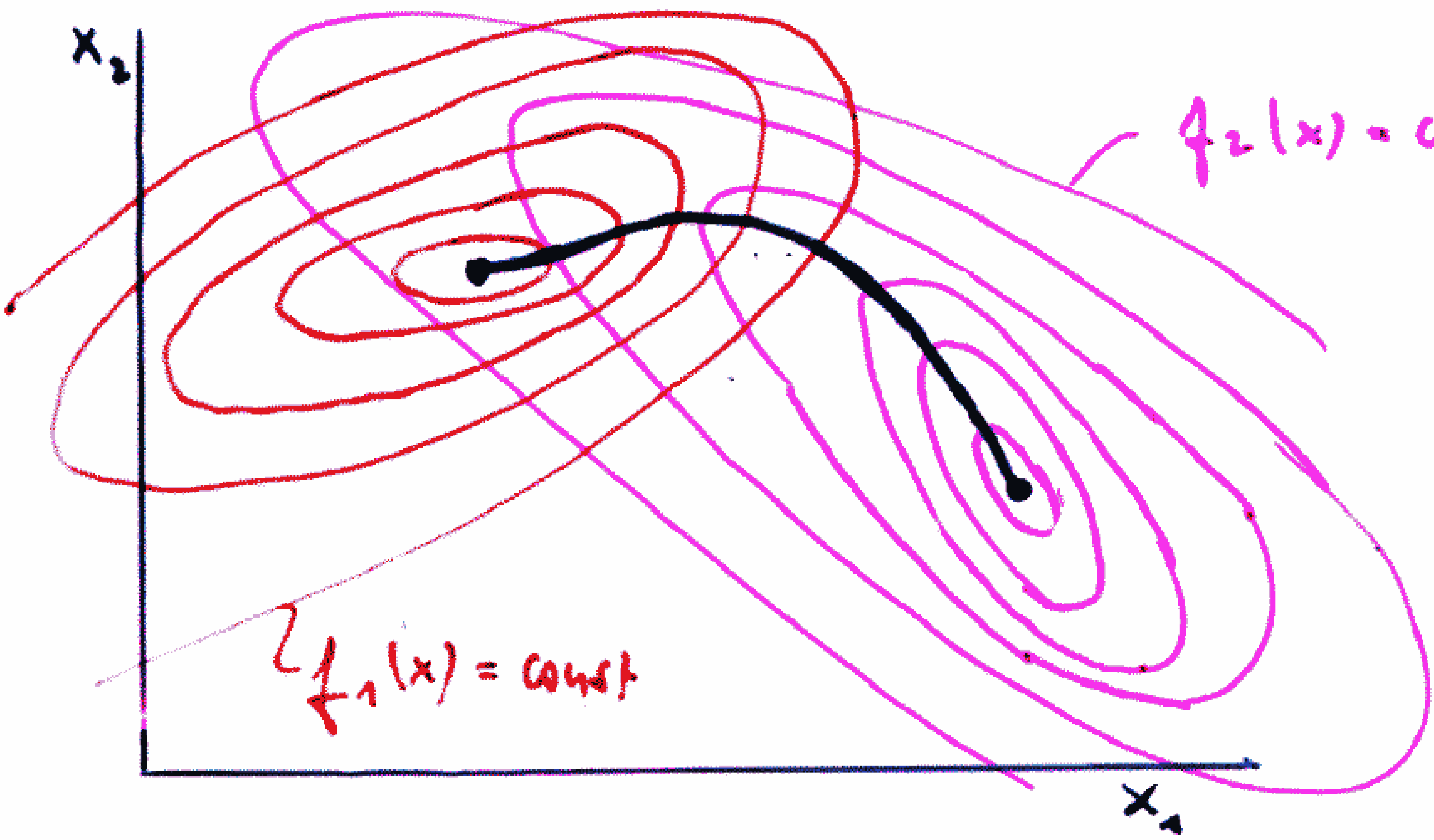
Vektor- oder Poly-Optimierung

$f: M \subseteq \mathbb{R}^n \to \mathbb{R}^l$ $\quad$ l Teilziele

stets Lösungsmenge

gesucht: effiziente (Pareto-optimale) Lösungen

nicht dominierte



• eigennützige Optima

— effiziente Lösungen $x^* \in M$ gesucht

$x^* \in M: \forall x \in M \quad f_k(x^*) \leq f_k(x) \quad \forall k \in \{1, 2, \dots, l\}$

und $\quad f_k(x^*) < f_k(x)$ für mind. ein k

d.h. kein Teilziel läßt sich mehr verbessern ohne nicht mindestens ein anderes zu verschlechtern

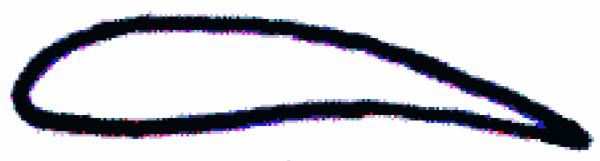
(nicht-dominierte Lösungen)

übliche Lösungsansätze

– Teilziele gewichten $\qquad \sum w_k f_k(x)$

↑ subjektives Urteil

z.B. Segelflugzeug, Tragflächenprofil



→ Widerstand W ↘ min

↑ Auftrieb A ↗ max

nicht additive Ziele

richtiger: $\frac{A}{W} = \frac{c_A}{c_W} \nearrow$ max

$A = c_A \cdot \frac{\varrho}{2} v^2 \cdot F$

$W = c_W \cdot \frac{\varrho}{2} v^2 \cdot F$

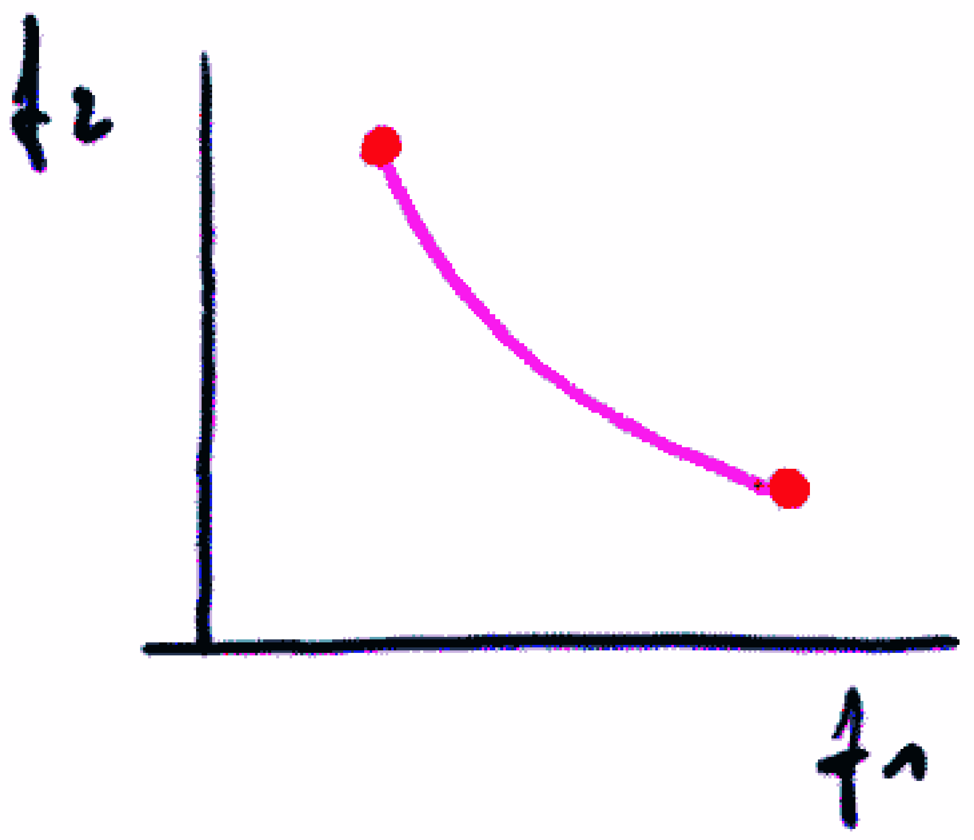
↳ flachster Gleitwinkel
Reichweite wird maximiert

oder: $\frac{c_A}{c_W^2} \nearrow$ max

↳ Flugzeit wird maximiert

„Physik" muß verstanden sein

– ein Teilziel auswählen und Mindestforderungen für andere Teilziele aufstellen



• eigennützige Optima

trade-off Kurve —

– goal programming

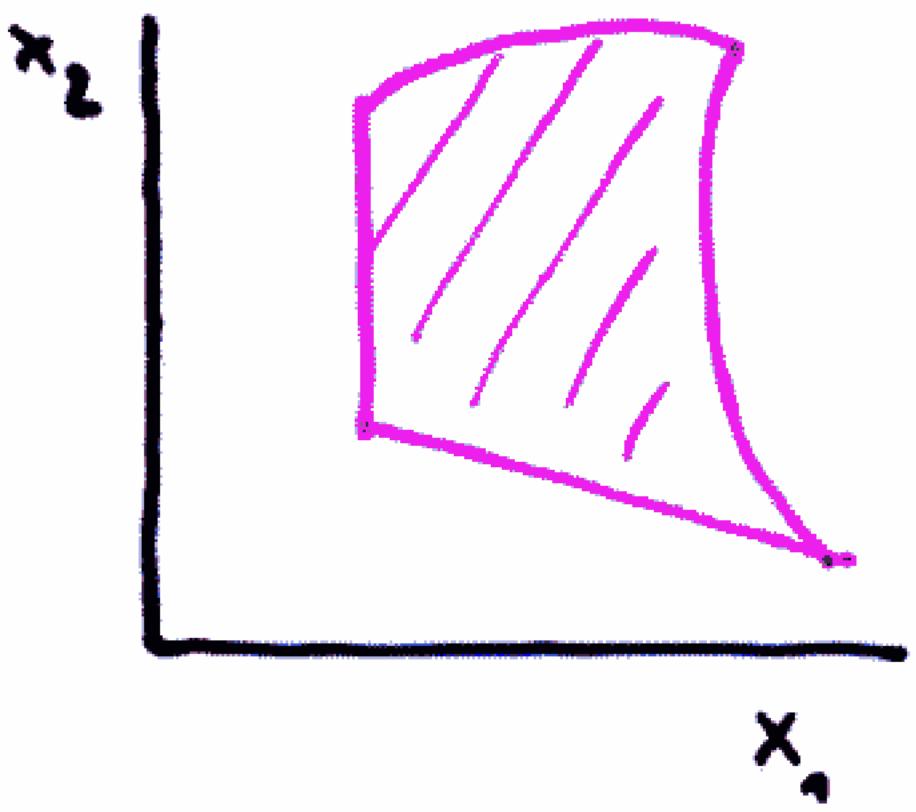
eigennützige Optima f_k^* bestimmen

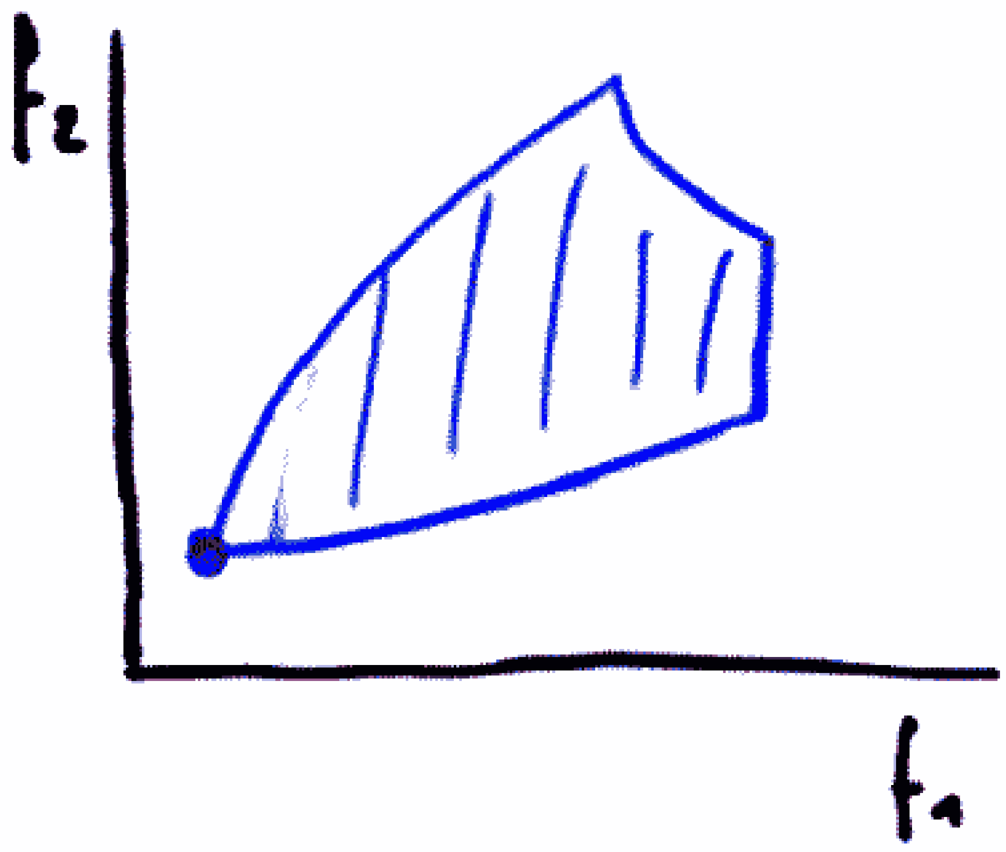
Utopia Punkt im f-Raum

$$F = \sum_{(k)} \left(f_k - f_k^* \right)^2 \to \min$$

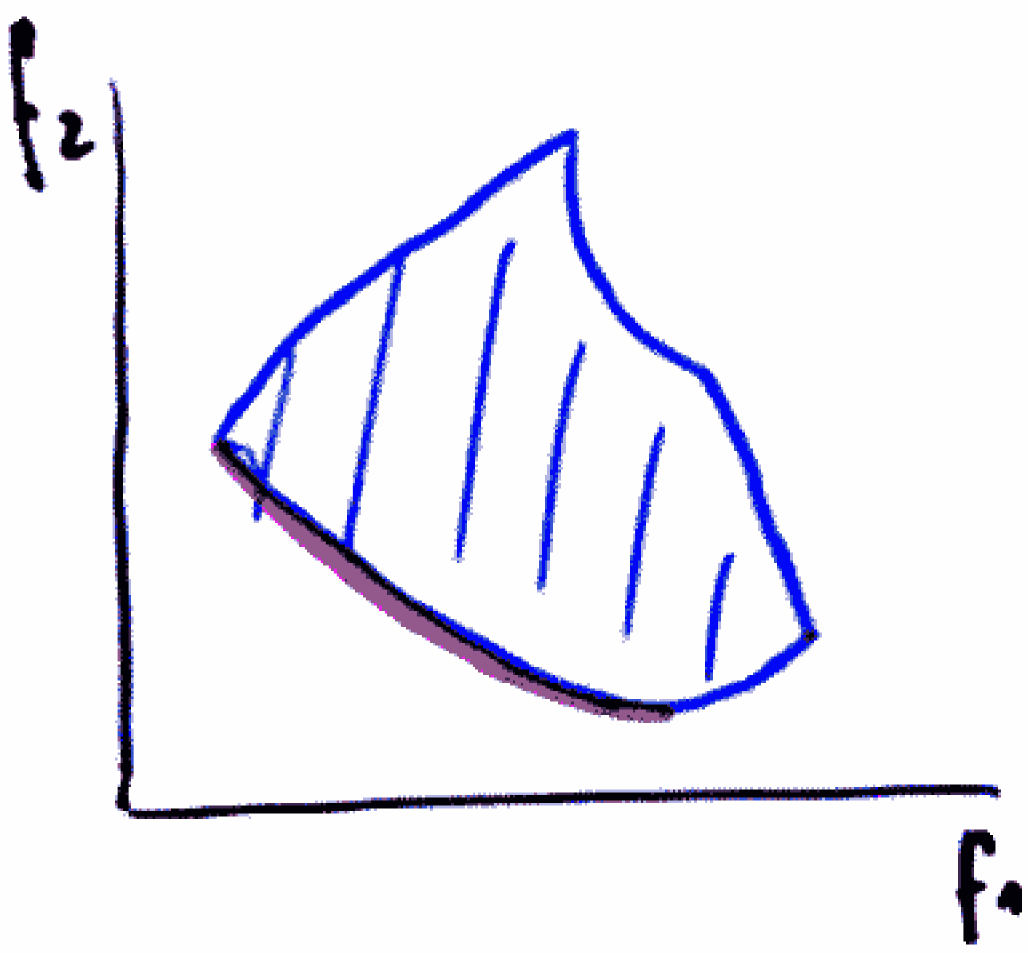
oder andere Abstandsmaße

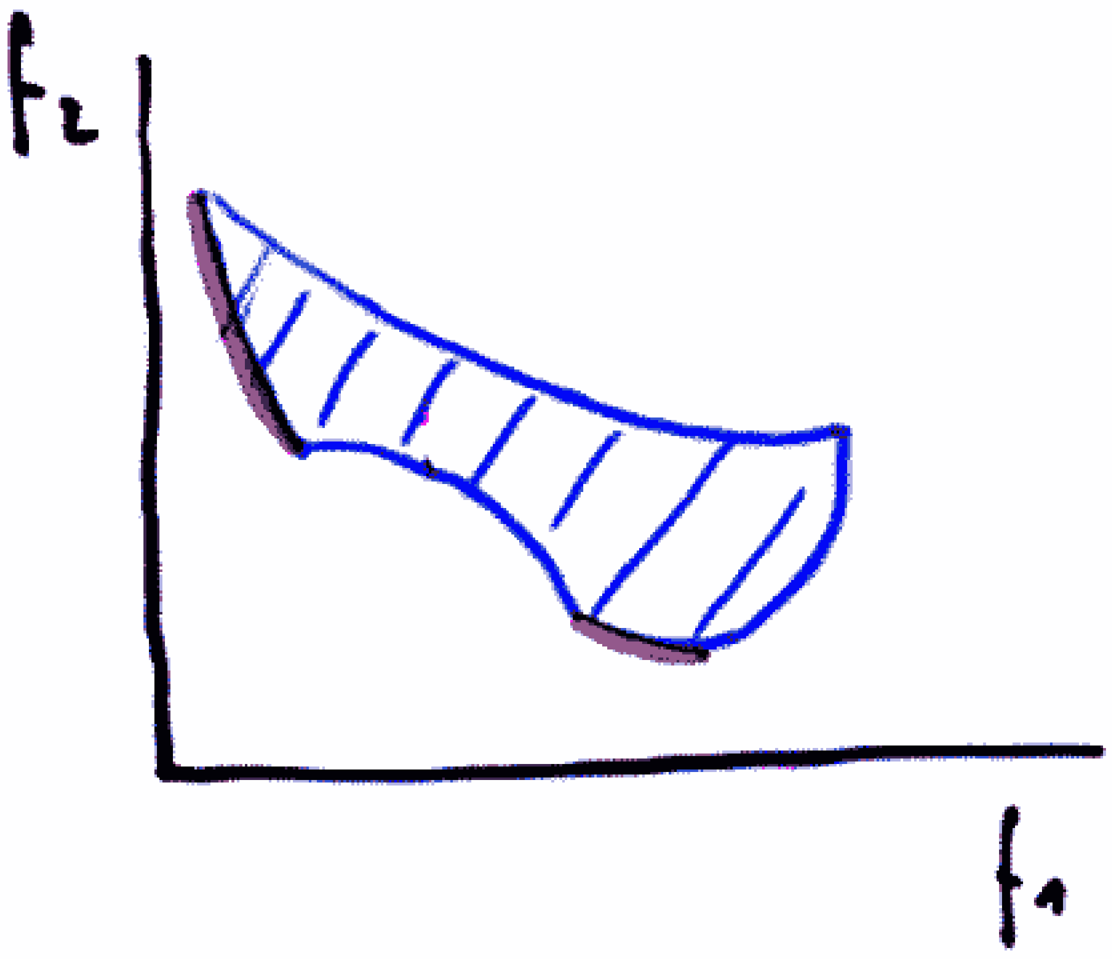
Beispiele für $n=2$, $l=2$





nicht-konfliktäre Ziele





konfliktäre Ziele; Lösungen zus. hängend oder auch nicht mehrere Entscheidungsträger mit unterschiedlichen Kompetenzen und Zielsetzungen

A maximiert $f_A(x, z)$

seine freien Entsch. variablen

B maximiert $f_B(y, z)$

seine freien Entsch. variablen

z sind Verhandlungsvariablen

Spieltheorie

z. B. Gefangenendilemma
Matrix-Spiele
(einmalig oder mit Wiederholung)

z. B. Nash Verhandlungslösung
(kein Spieler dominiert)

Schritt 1a bestimme $f_A^*(x^*, z=0)$

1b $f_B^*(y^*, z=0)$

Status quo-Lösungen

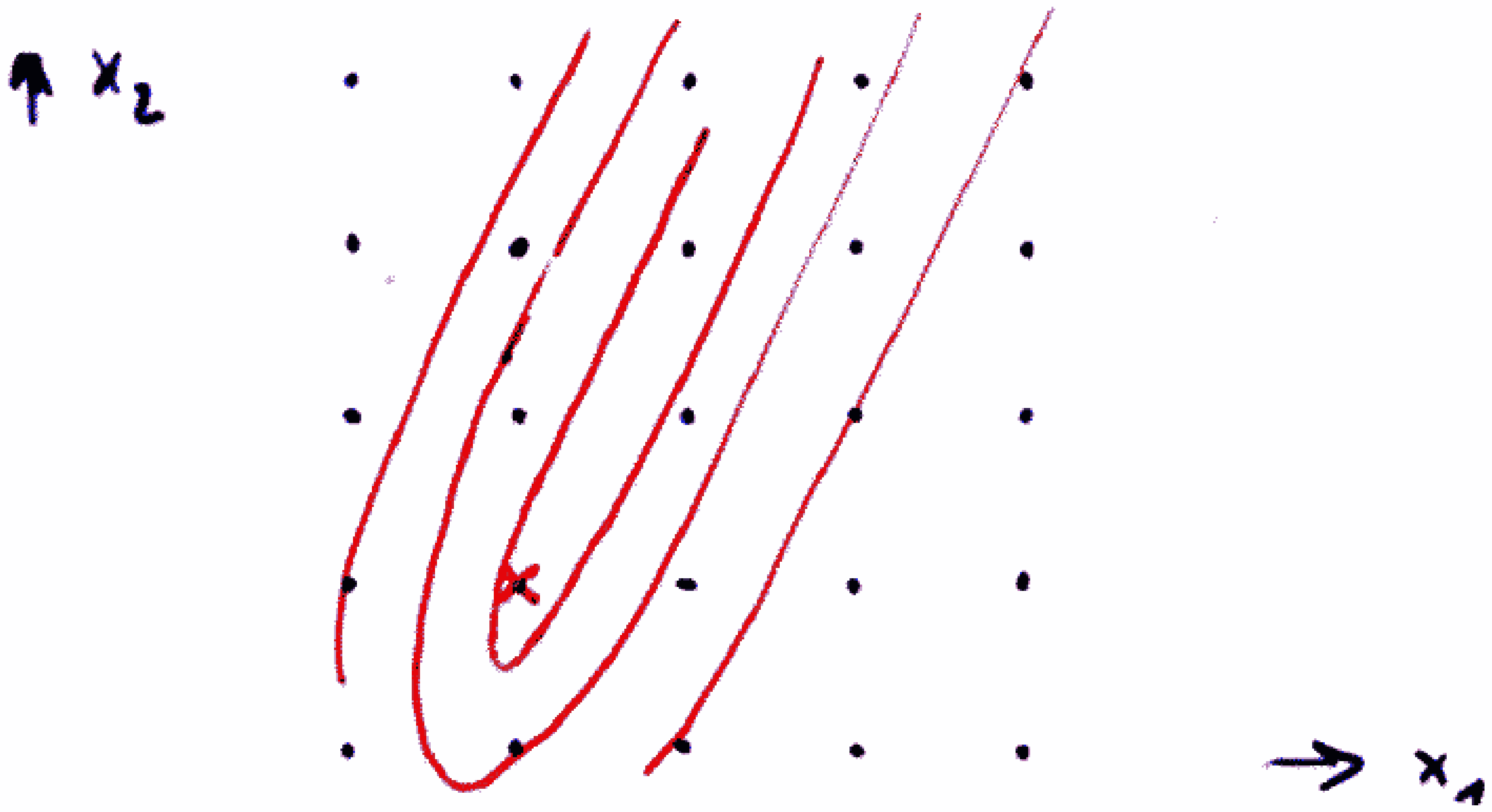
Schritt 2 maximiere gemeinsamen Zugewinn

$$F(x, y, z) = (f_A - f_A^*) \cdot (f_B - f_B^*)$$

andere Konzepte : Nash-Gleichgewicht
Stackelberg- "

weitere Schwierigkeiten beim Optimieren

– diskrete Variable (z.B. ganzzahlig, Null-Eins)



in Nachbarschaft von x keine Verbesserung erzielbar

– stochastische Störungen

$$f(x+\varepsilon) \quad \text{oder} \quad f(x)+\eta$$

– Erwartungswert optimieren

– stochastische Approximation (n = 1)

(deterministisches Verfahren!)

$$x^{(k+1)} = x^{(k)} - 2\, a^{(k)} \frac{F(x^{(k)}+c^{(k)}) - F(x^{(k)}-c^{(k)})}{2\, c^{(k)}}$$

$$a^{(k)} = \frac{1}{k}\, a^{(0)} \quad ; \quad a^{(0)} > 0$$

$$c^{(k)} = \frac{1}{\sqrt[4]{k}}\, c^{(0)} \quad ; \quad c^{(0)} > 0$$

$\hat{=}$ Grad. meth. mit Tastschritten $c^{(k)}$

Arbeitsschritten $a^{(k)}$

- wanderndes Optimum

permanente on-line Optimumsuche

dynamische Optimierung

Opt. finden und halten

gewonnene Information (z. B. über Gradienten) veraltet : ‚Vergessen' wird notwendig

---

weitere Spezialfälle

- gebrochen rationale Progr. $F(x) = \dfrac{g(x)}{h(x)}$

- geometrische Optimierung $F(x)$ = Posynom
  gebrochen-rationale Exponenten

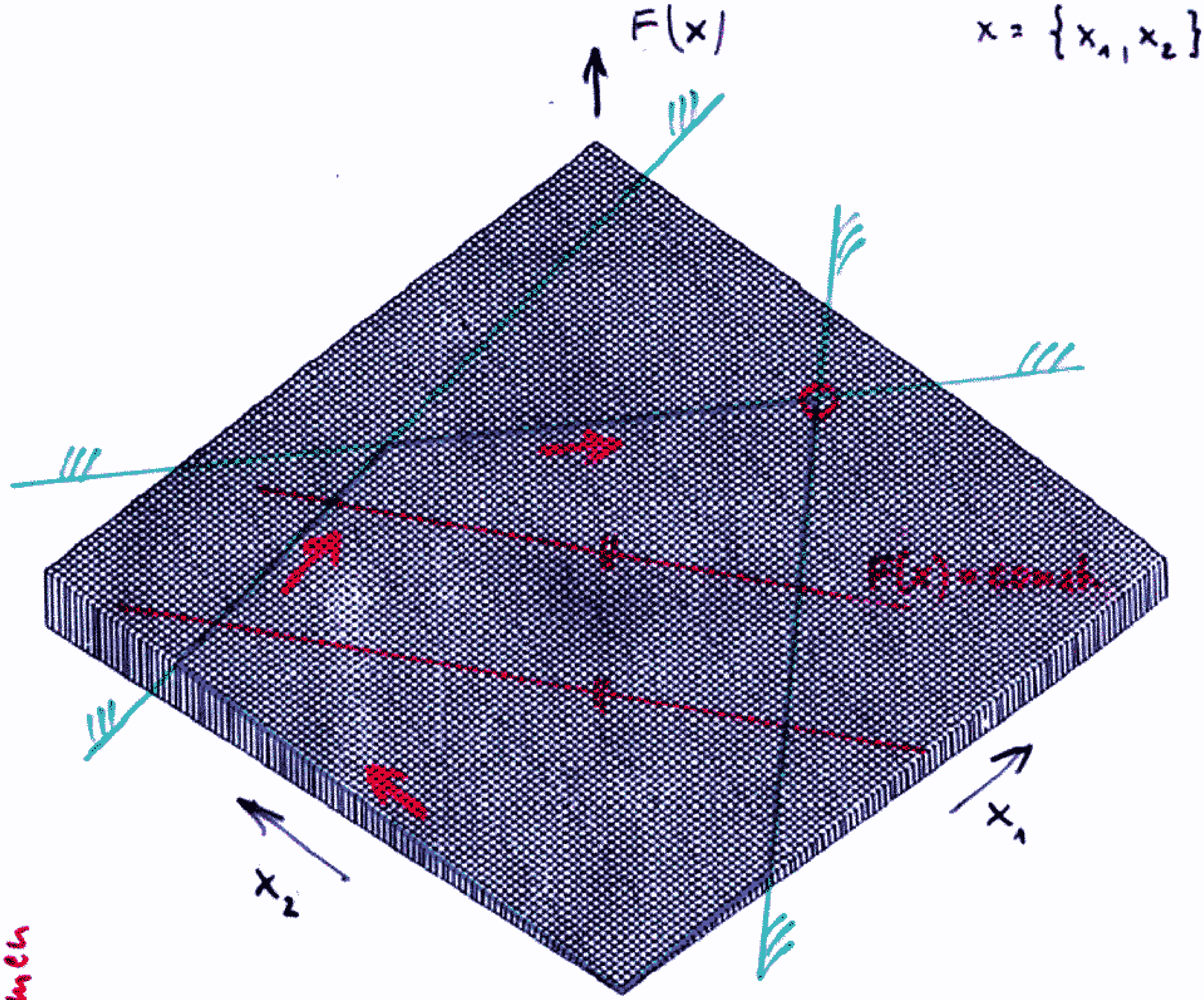
- Reihenfolgen - Optimierung (z. B. TSP)
  kombinatorische Opt.

Spezialgebiet OR Unternehmensforschung

- Transport
- Rundreise
- Zuordnung
- Rucksack
- Zuschnitt
- Stundenplan
- ⋮

– Probleme

stets spezielle Verfahren für einfache, eng umgrenzte Aufgaben



sukzessives Lösen von Gleichungssystemen

lineare Zielfunktion
lineare Nebenbedingungen
+ Nichtnegativität

konvexer zulässiger Bereich
Lösung: eine der Ecken
(außer bei Degeneration)

Simplex-Methode (G. Dantzig): LP Lineare Programmierung
(Gradientensuche entlang den Kanten)
(Pivotisierung: 'beste' Nachbarecke)
sukzessive Lösung von linearen Gleich. syst.
Vorwissen genutzt: Lösung in Ecke

Lineare Programmierung (LP)

hier: Normalform

$z = cx \to \min$ $\qquad\qquad$ $\bar{z} = b\bar{x} \to \max$

$Ax \leq b$ $\qquad\qquad$ $A^T\bar{x} \geq c$

$x \geq 0$ $\qquad\qquad$ $\bar{x} \geq 0$

primales Problem $\rightarrow$ duales Problem

duales $\leftarrow$ primales

Lösung sei x^* bzw. $\bar{x}^*$

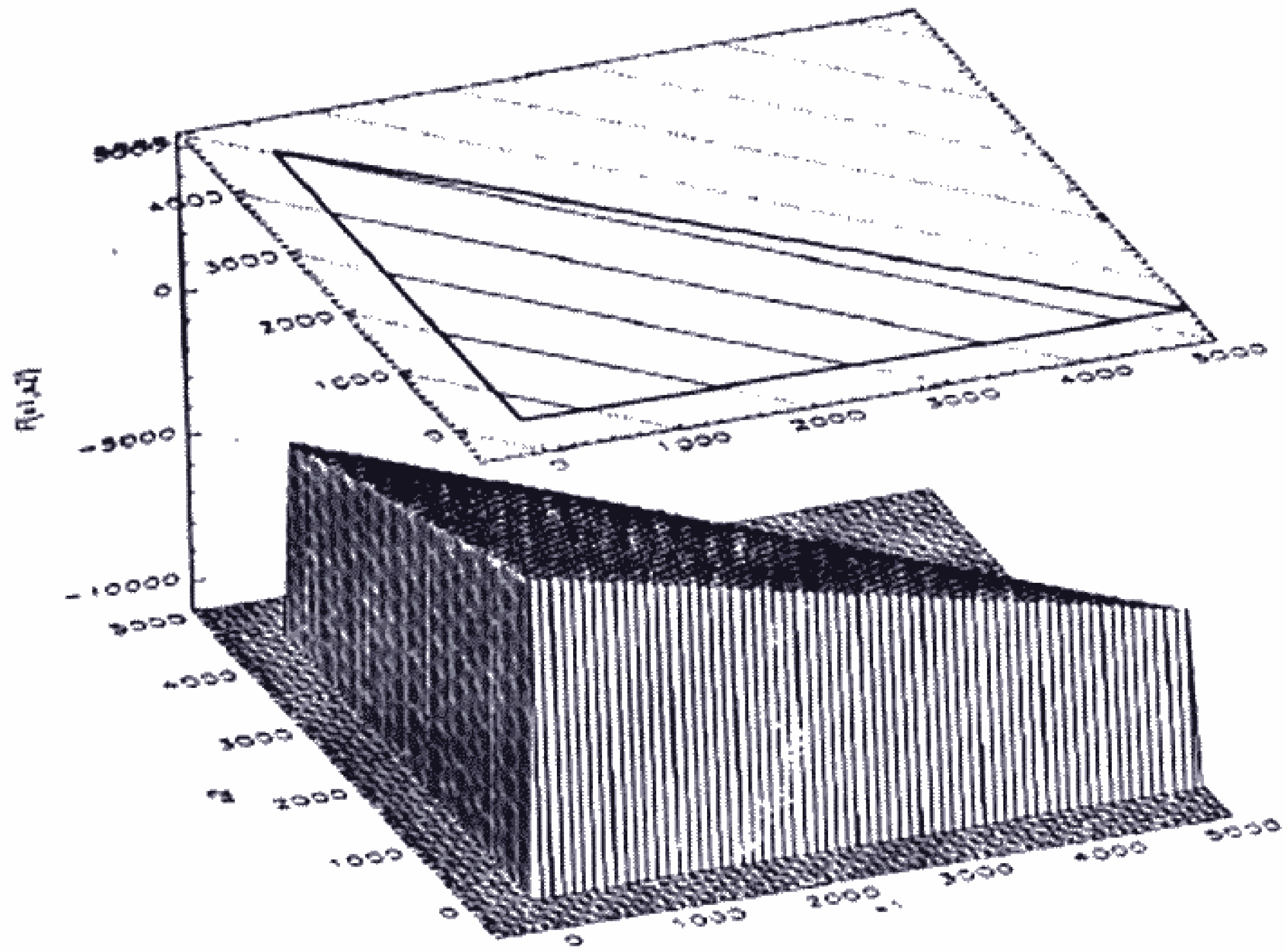
dann gilt $z(x^*) = \bar{z}(\bar{x}^*)$

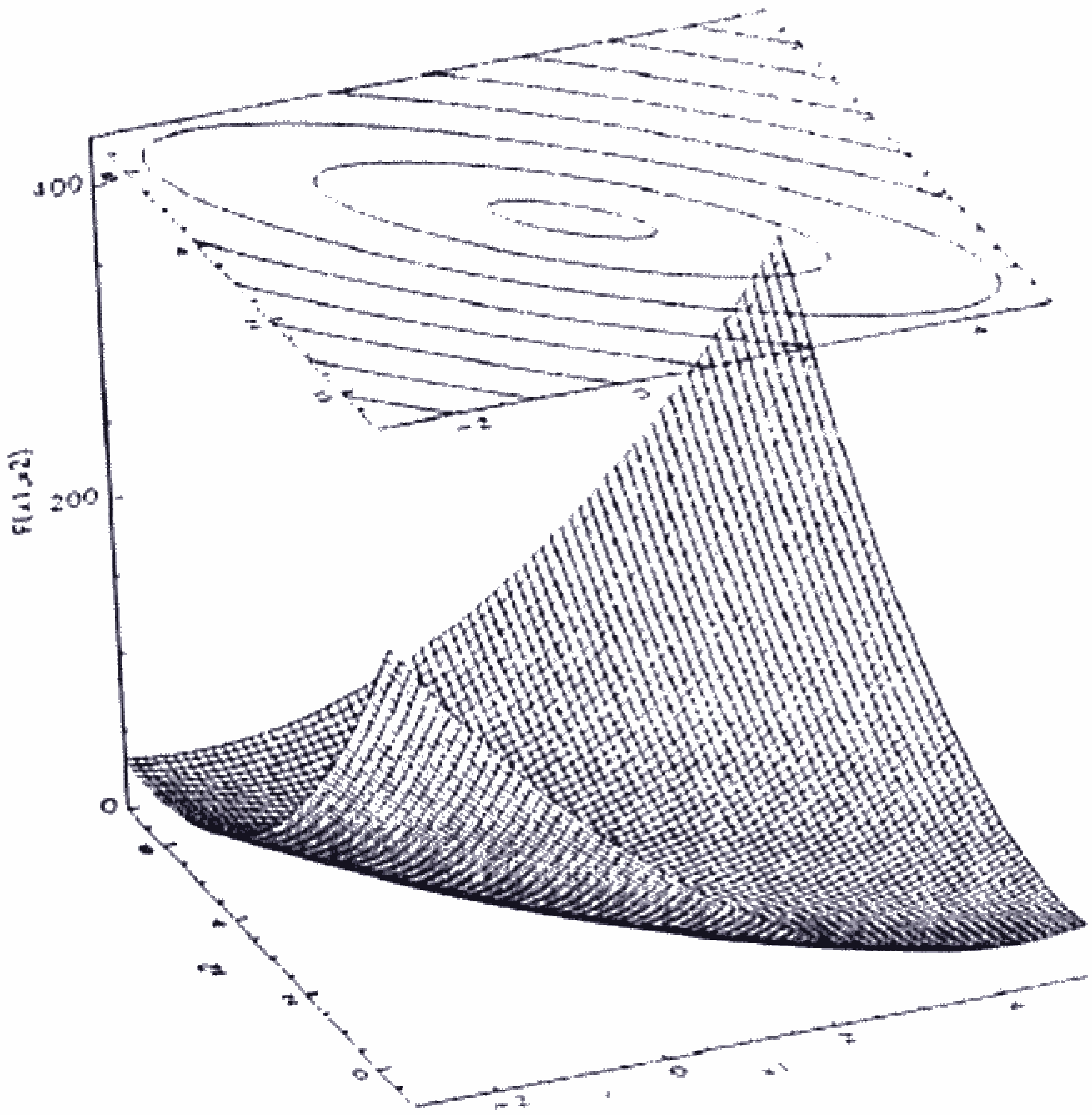
Dualitätssatz (Voraus. Lösbarkeit)

Standardverfahren: Simplexmethode
(G. B. Dantzig, ~1940)

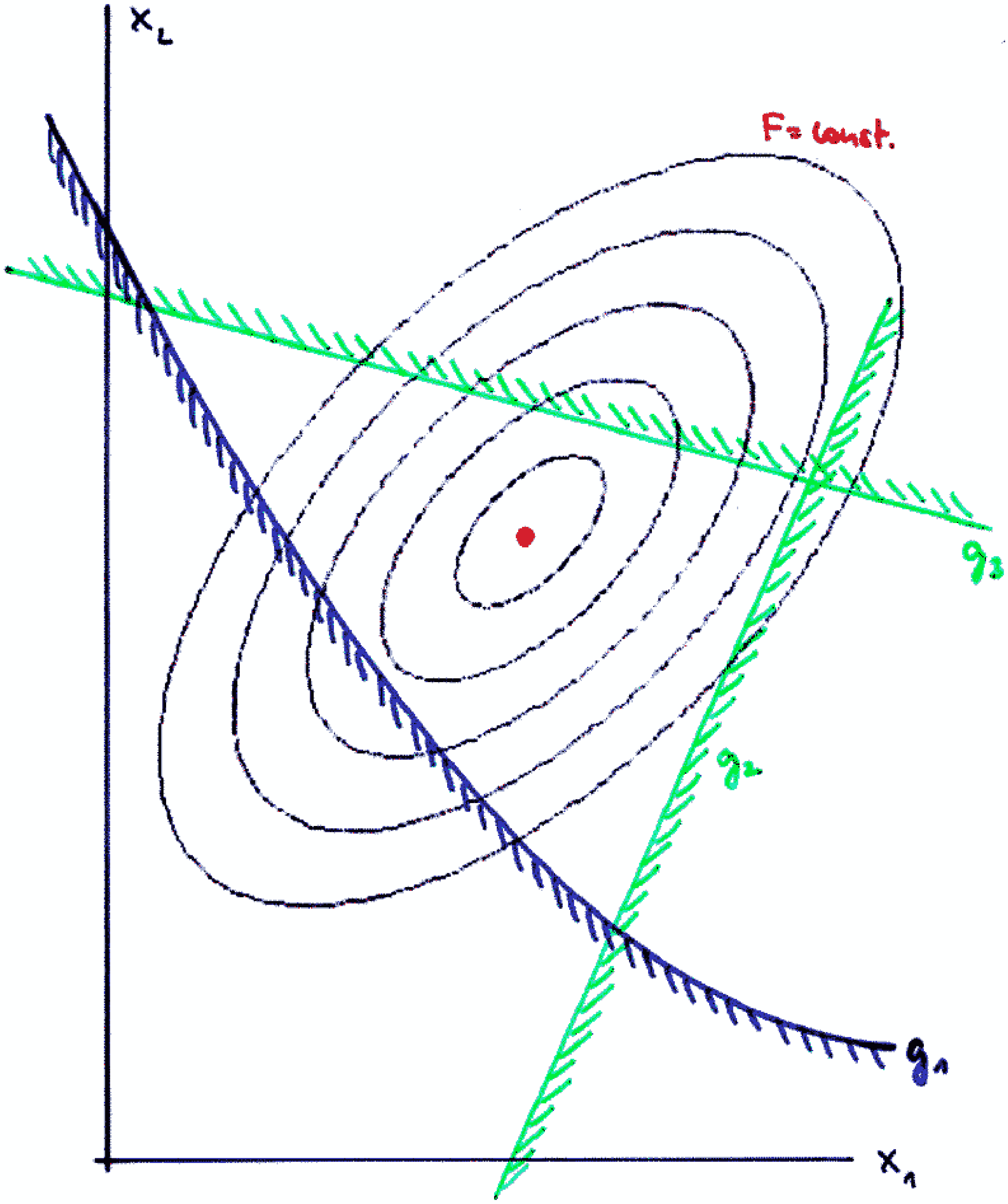
Simplextableau

$$
\begin{array}{ccccc|cccc|c}
x_1 & x_2 & \cdots & x_n & & s_1 & s_2 & \cdots & s_m & \\
a_{11} & a_{12} & \cdots & a_{1n} & & 1 & 0 & \cdots & 0 & b_1 \\
a_{21} & a_{22} & \cdots & a_{2n} & & 0 & 1 & \cdots & 0 & b_2 \\
\vdots & & & & & & & & & \\
a_{m1} & & \cdots & a_{mn} & & 0 & 0 & \cdots & 1 & b_m \\
\hline
c_1 & c_2 & \cdots & c_n & & \underbrace{0 \quad 0 \quad \cdots \quad 0}_{\text{hier } -x^* \text{ ablesbar}} & & & & z
\end{array}
$$

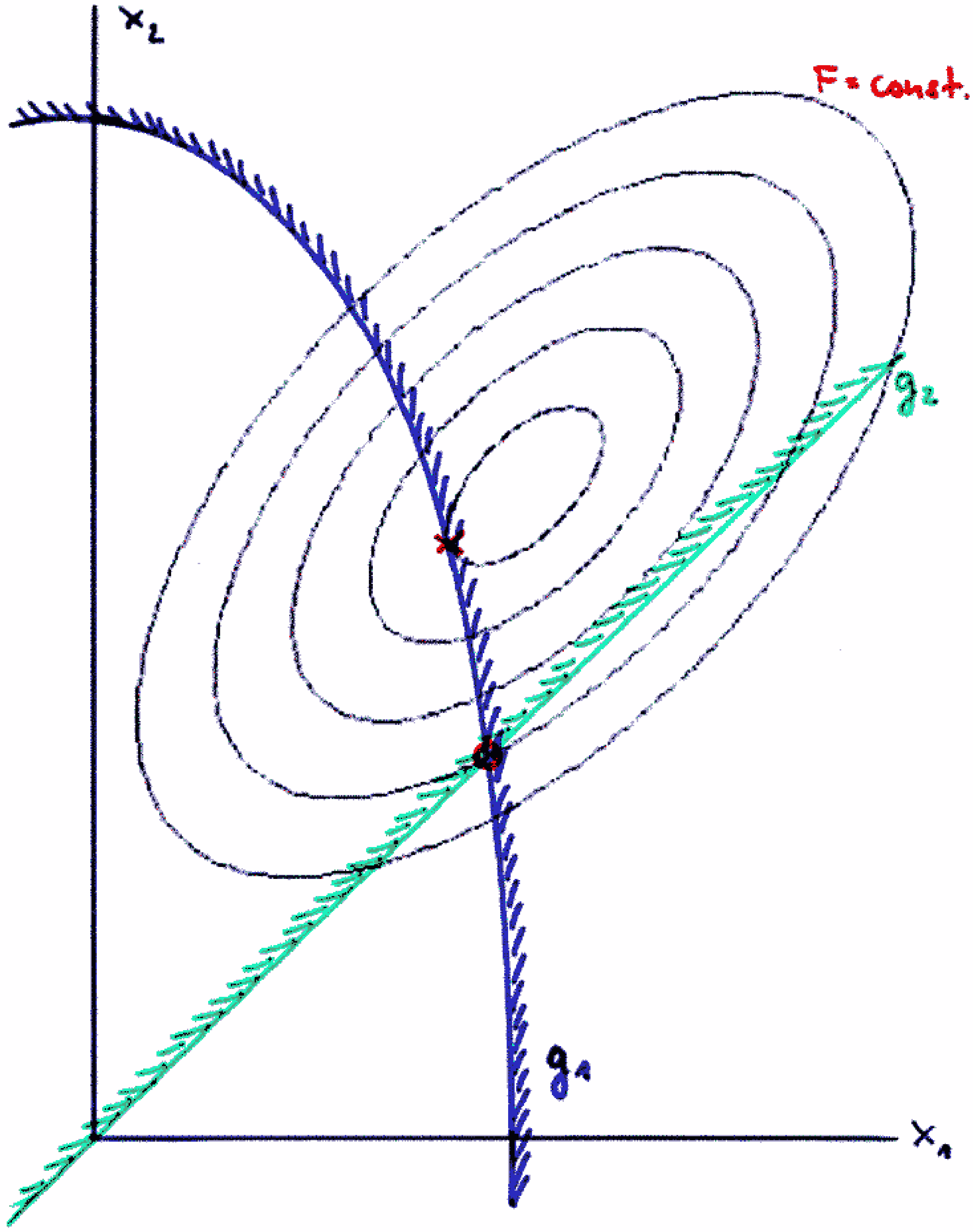

5000
4000
3000
2000
1000
0
-5000
-10000
f(x1,x2)


400
200
0

Nichtlineare Optimierung

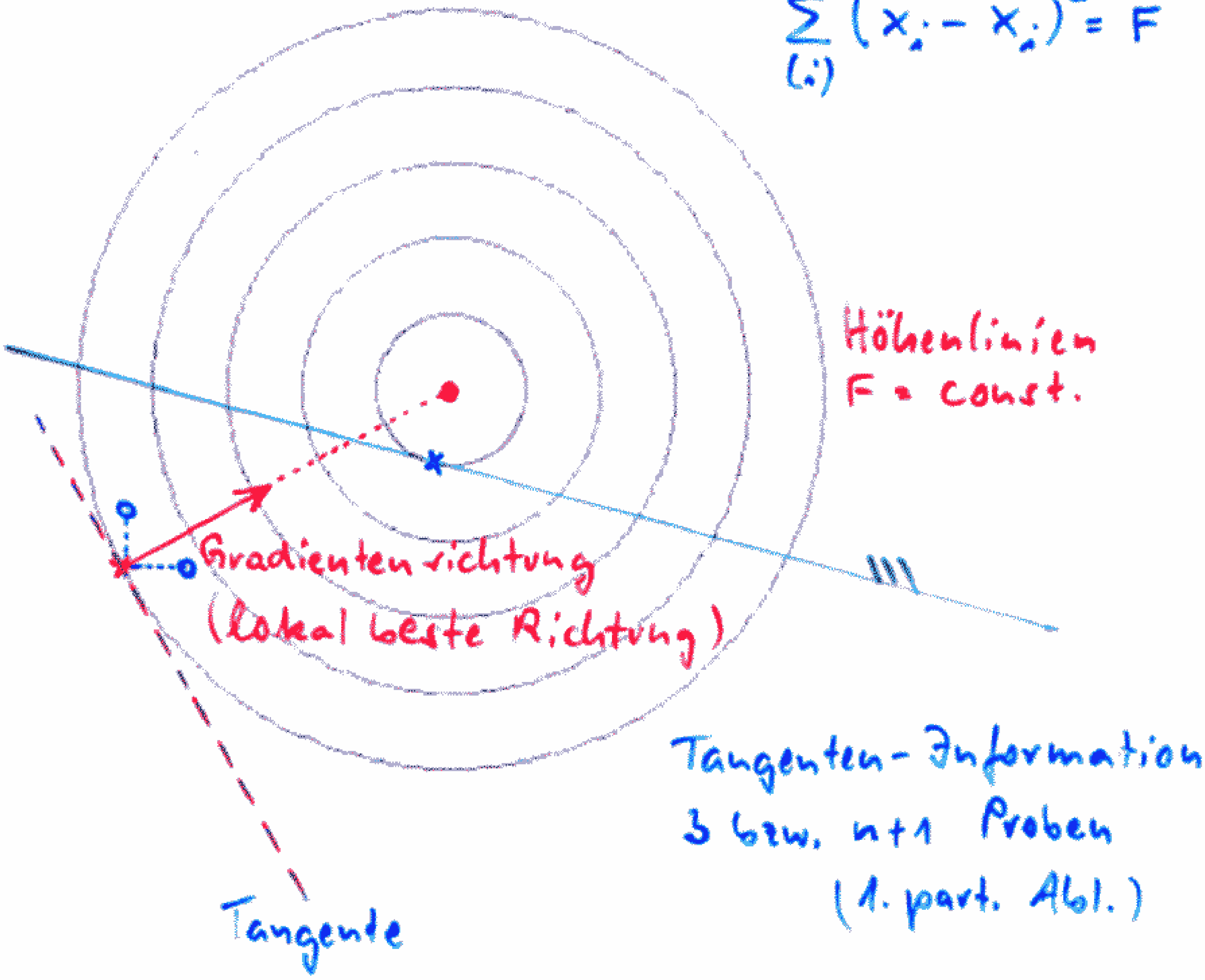


Nichtlineare Programmierung

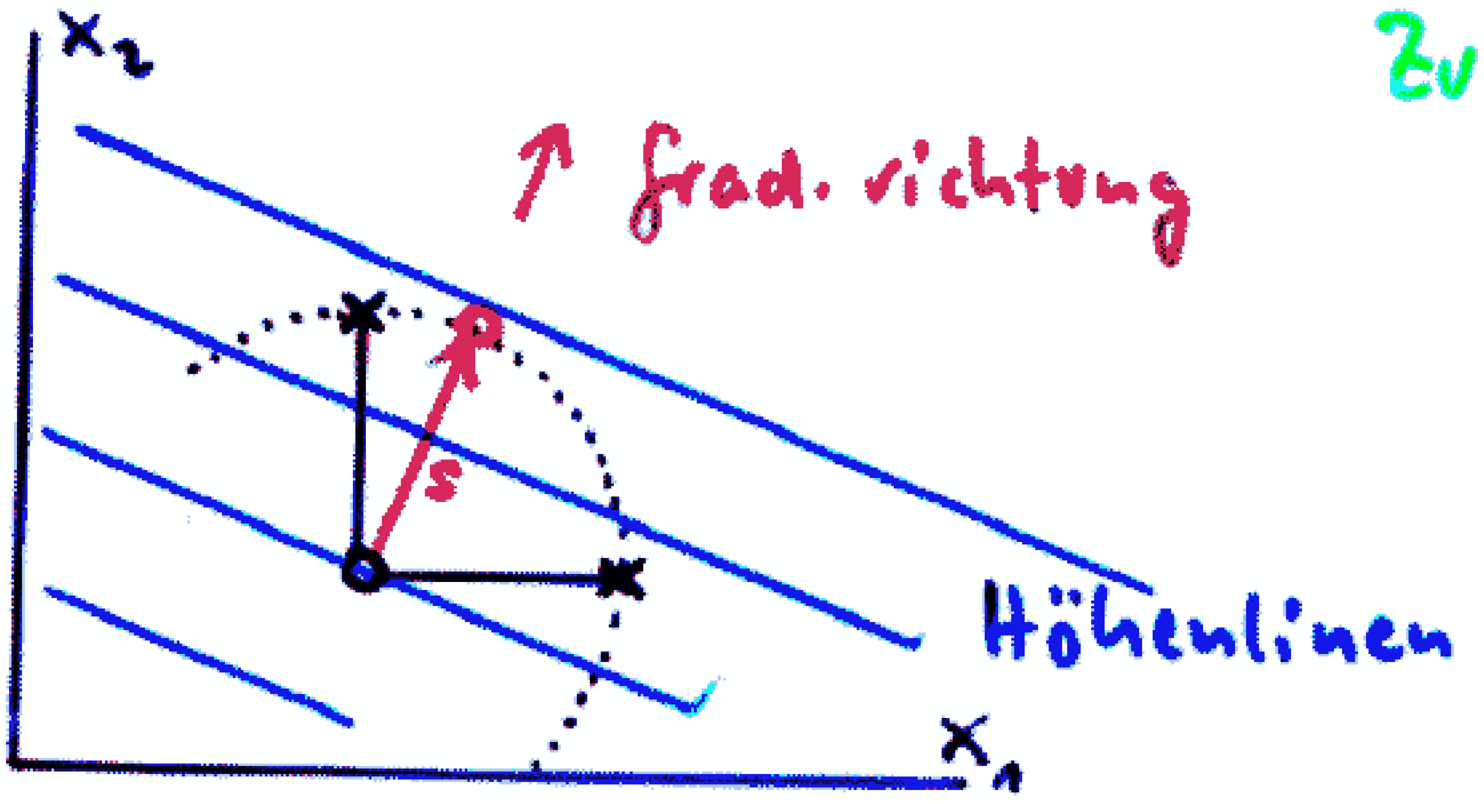


'Künstliches' Gebirge II

$$\sum_{(i)} (x_i - \hat{x}_i)^2 = F$$



Vergleich: experimentelle Grad. strat.
Zufallsstrat.



2 Parameter : 2 Tastschritte
1 Arbeitsschritt

n Parameter : n+1 Experimente

$$\text{Konvergenzgeschw.} = \frac{\text{zurückgelegter Weg}}{\text{Anzahl der Versuche}}$$

$$\varphi = \frac{s}{n+1}$$

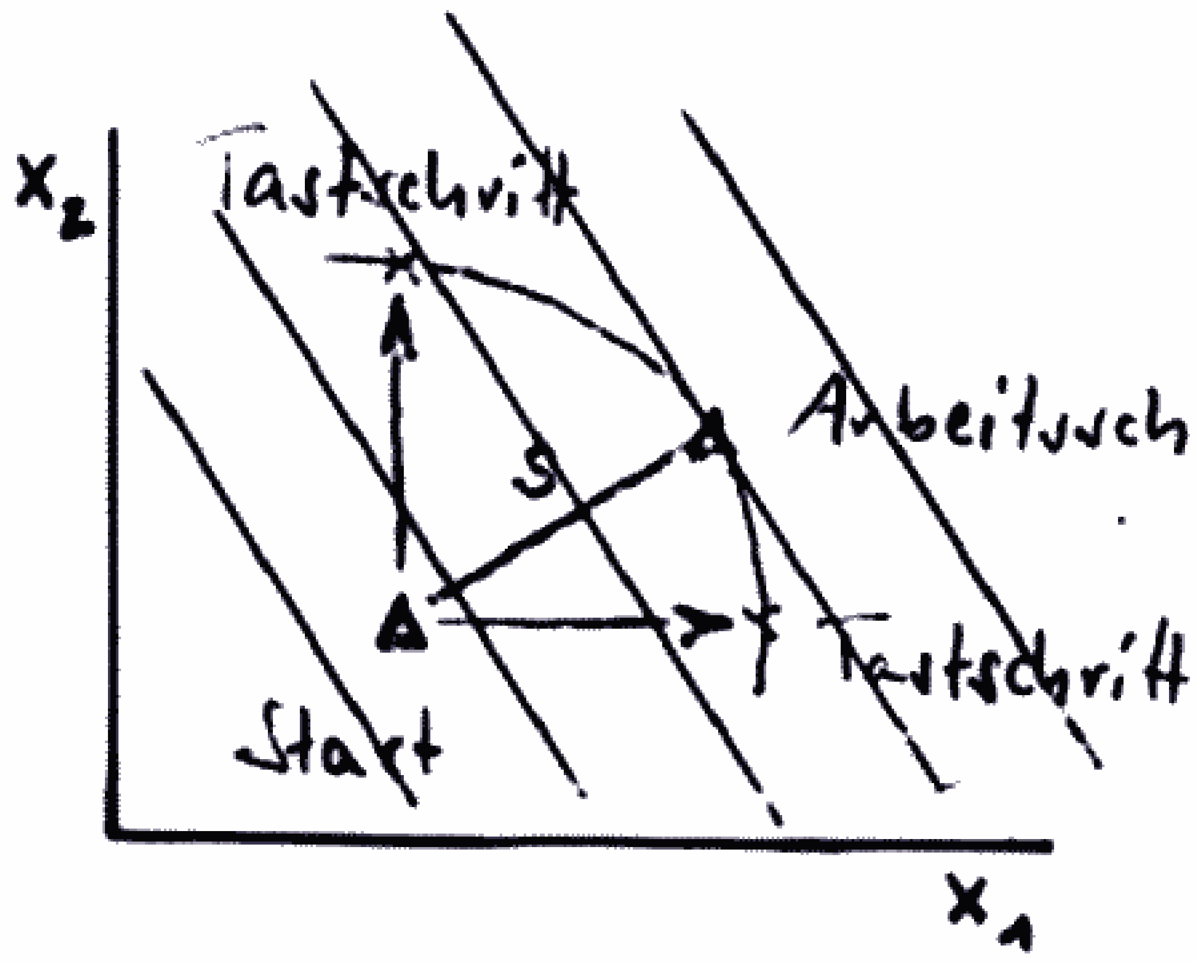
s kann so groß sein, daß man bis zum rel. Opt. gelangt

Erlernen der Topologie aus lokaler Analyse mit „innerem Modell"

Vergleich deterministische / stochastische Gradientenmethode

Bedingung: kleine Schritte / lineare Situation (lokal)

A



Arbeitsschritt in Ri. des Gradienten

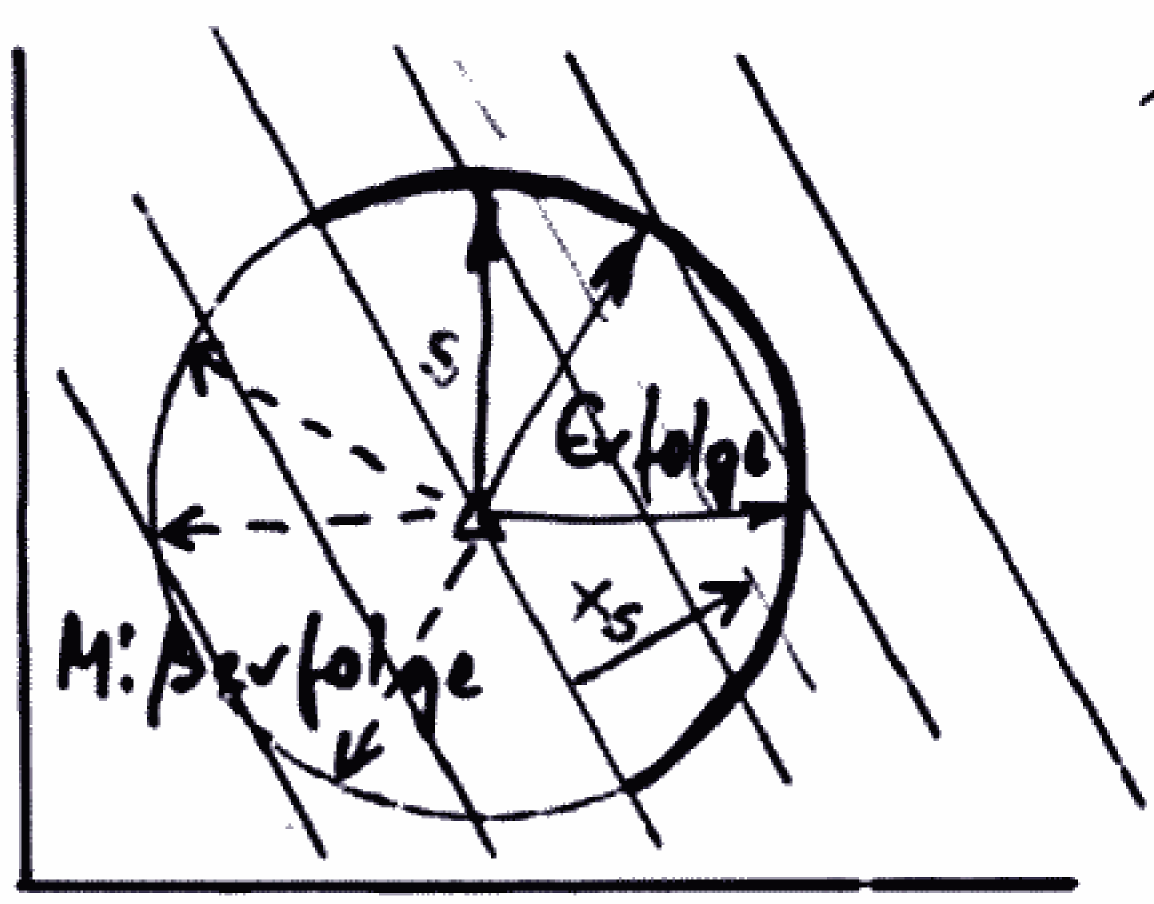
$n = 2$ 3 Zielf. auswertungen

allg. $n+1$ "

$$\text{Konvergenzgeschwind.} = \frac{\text{zurückgelegter Weg in Ri. Ziel}}{\text{Anzahl Zielf. auswertungen}}$$

$$\varphi_{det.} = \frac{s}{n+1} \sim \frac{1}{n} \quad \text{für } n \gg 1$$

B



stochastische Versuche gleichverteilt auf Kreisrand (Kugelrand) mit Radius s

$n = 2$

$$\varphi_{stoch.} = \underset{W_e}{\frac{1}{2}} \cdot x_s$$

W_e = Erfolgswahrscheinlichkeit

x_s = Linienschwerp. des erfolgreichen Halbkreises (Erwartungswert für Weg in Ri. Ziel)

Guldinsche Regel für x_s

$$x_s = \frac{1}{2\pi} \cdot \frac{\text{erzeugte Fläche bei Rotation}}{\text{Länge der erzeugenden Linie}}$$

$$= \frac{1}{2\pi} \cdot \frac{\text{Kugeloberfläche}}{\frac{1}{2}\ \text{Kreisumfang}} = \frac{1}{\pi} \cdot \frac{O_3}{O_2}$$

wobei O_n = Oberfläche einer n-dim. Kugel

$$O_2 = 2\pi s \quad ; \quad O_3 = 4\pi s^2$$

für $n=2$ also $x_s = \frac{1}{\pi} \cdot \frac{4\pi s^2}{2\pi s} = \frac{2s}{\pi}$

damit $\varphi_{stoch.} = \frac{1}{2} \cdot \frac{2s}{\pi} = \frac{s}{\pi}$

für beliebiges n

$$x_s = \frac{1}{\pi} \cdot \frac{O_{n+1}}{O_n} \quad \text{wobei } O_n = \frac{2\pi^{\frac{n}{2}} s^{n-1}}{\Gamma\left(\frac{n}{2}\right)}$$

$$= \frac{s}{\sqrt{\pi}} \cdot \frac{\Gamma\left(\frac{n}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right)}$$

Stirling-Formel für $m \gg 1$ $\quad \Gamma(m) \approx \sqrt{2\pi}\, m^{m-\frac{1}{2}} e^{-m}$

ferner gilt $\lim\limits_{x\to\infty} \left(1 + \frac{1}{x}\right)^x = e$

für $n \gg 1$ folgt $x_s \approx \sqrt{\frac{2}{\pi}} \cdot \frac{s}{\sqrt{n}}$

$$\varphi_{stoch.} \approx \frac{s}{\sqrt{2\pi n}} \sim \frac{1}{\sqrt{n}}$$

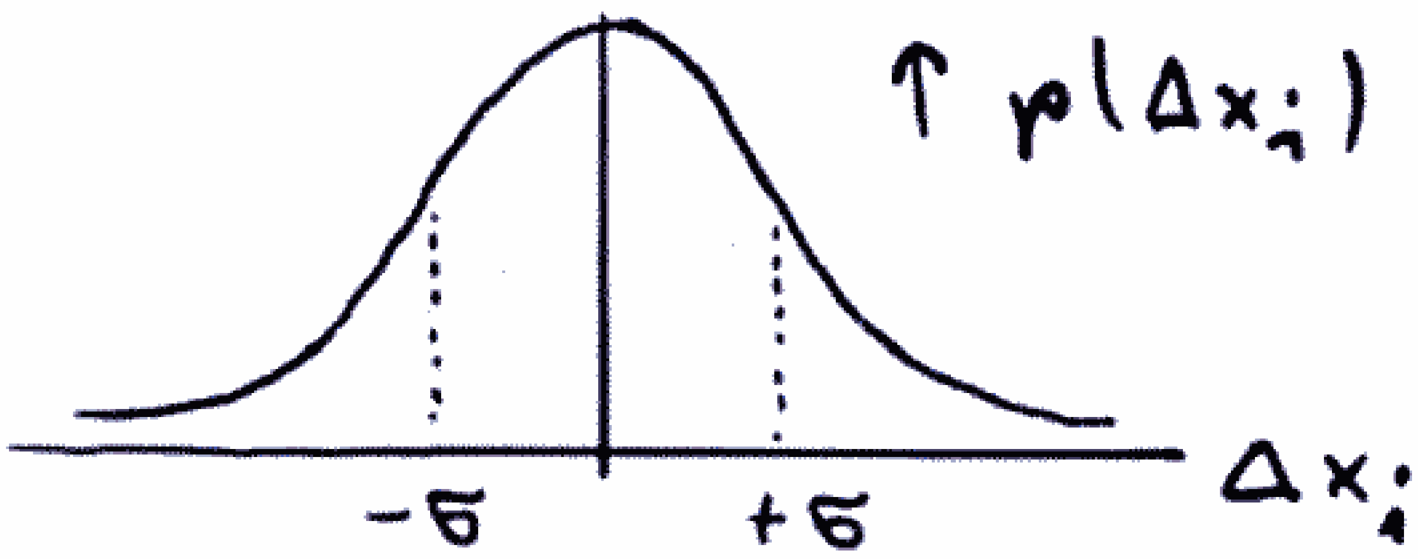
Schlußfolgerung: Informationsbedarf $\sim n$ ist nicht nötig; man kommt mit $\sim\sqrt{n}$ aus!

stochast. Gradientenmethode

- starte irgendwo im $\mathbb{R}^n$
- plaziere Probe auf Kugelrand um Ausgangsposition (fester Radius s)*
- wenn neue Position besser,
  dann übernimm diese als neue Ausgangspos.
  sonst behalte die alte Pos. bei (neue vergessen)

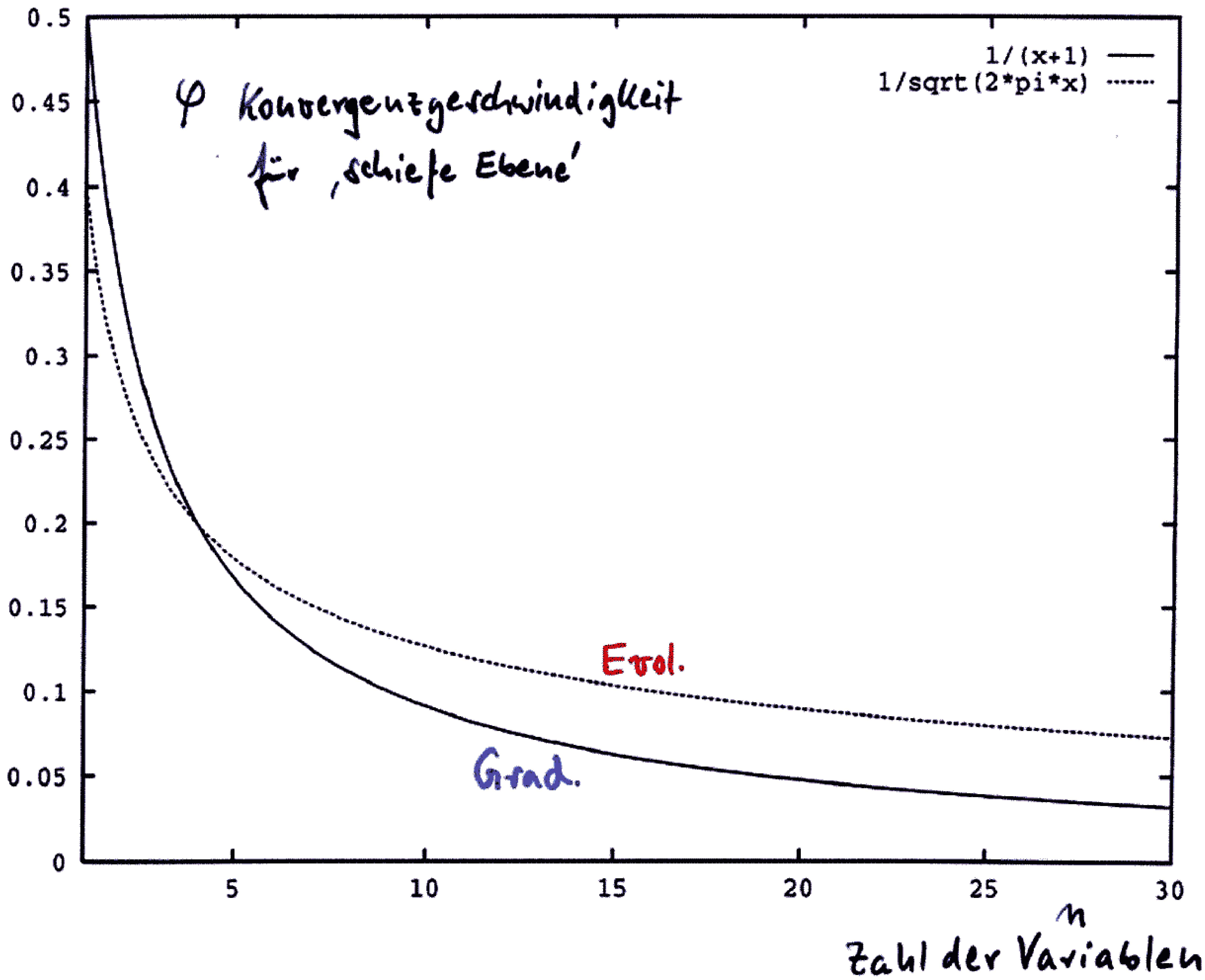
* alternativ: Normalverteilung mit σ als Standardabweichung für jede Variablenänderung Δx_i
$i = 1(1)n$

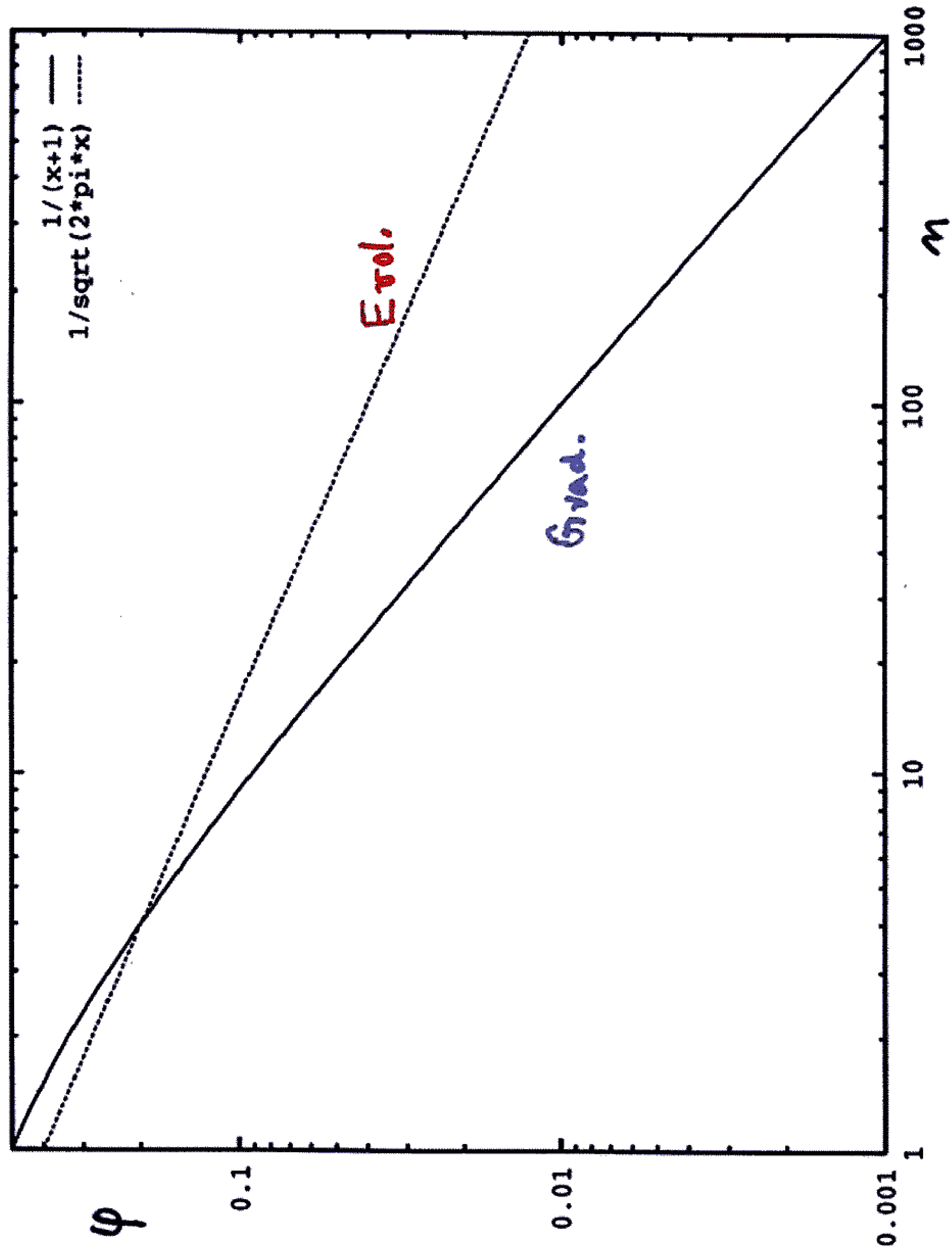
Mittelwert des Gesamtschritts : $\sigma \cdot \sqrt{n} \mathrel{\hat{=}} s$



rein zufällig ist nur die Richtung
Einzelschrittweiten sind normalverteilt
die Gesamtschrittweite wird mit wachsendem n
immer wahrscheinlicher $\sigma \cdot \sqrt{n}$
(entsprechend einer Kugelrandverteilung
mit Radius $s = \sigma \cdot \sqrt{n}$)

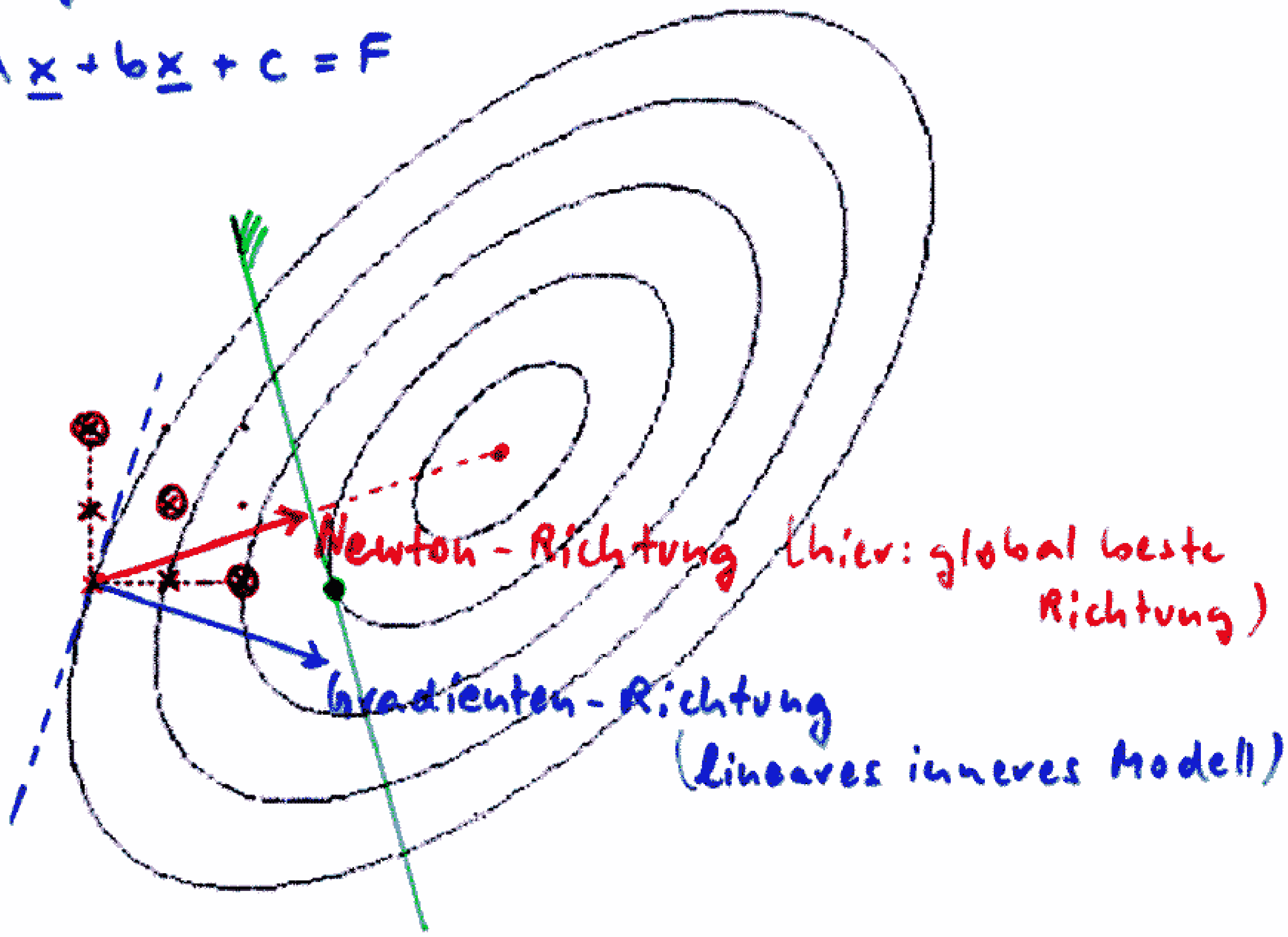
wichtig: trotz Zufallskomponente ~~weniger~~ strat. weniger aufwendig als det. Grad. meth. mit Tastschritten


1/(x+1)
1/sqrt(2*pi*x)
φ Konvergenzgeschwindigkeit
für ‚schiefe Ebene'
Evol.
Grad.
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
5
10
15
20
25
30
n
Zahl der Variablen


1/(x+1)
1/sqrt(2*pi*x)
Erol.
Grad.
0.1
0.01
0.001
1
10
100
1000
φ
n

'Künstliches'
Gebirge III

$\underline{x}^T A \underline{x} + b\underline{x} + c = F$



Nichtlineare Programmierung

quadratisches inneres Modell

benötigte Information:

2 bzw. n 1. part. Abl.

3 bzw. $\frac{n}{2}(n+1)$ 2. part. Abl.

für $n = 10^5$

etwa 10^{10} örtliche Versuche nötig

vgl. Weltpopulation $5 \cdot 10^9$

QP lineare Zielfkt, eine nichtlin. Neb.
nichtlin. " , nur lineare Neb.

SLP
projizierte Gradienten-Methode

NLP Kuhn Tucker

SQP
allgemeine Suchverfahren

OR

---

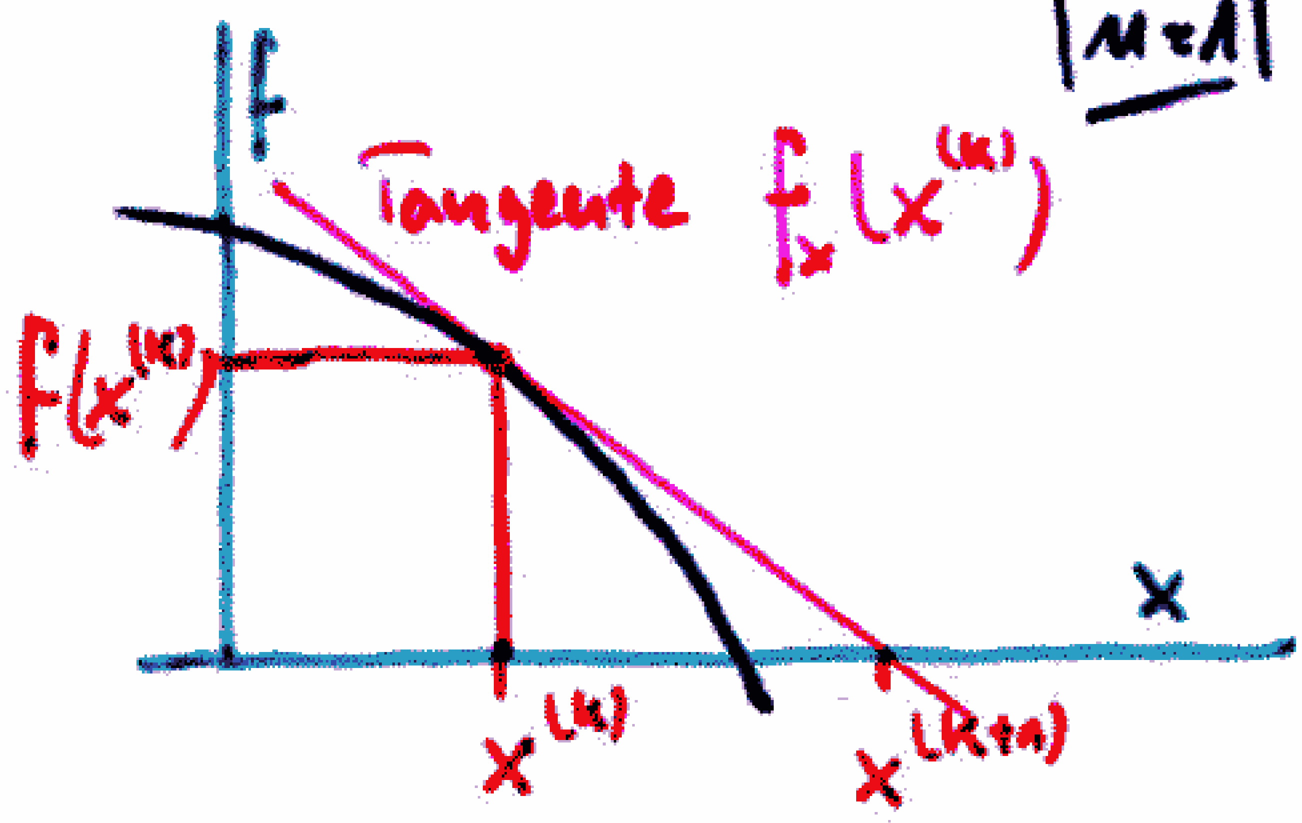
quadrat. Problem ohne Restr.

$$f(x) = x A x^T + b x + c$$

$$\nabla f(x^*) = 0$$

Newton-Raphson
Nullstellen Bestimmung

$\boxed{n=1}$



$$f_x(x^{(k)}) = \frac{f(x^{(k)}) - 0}{x^{(k)} - x^{(k+1)}}$$

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f_x(x^{(k)})}$$

n beliebig

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{\nabla f(x^{(k)})}$$

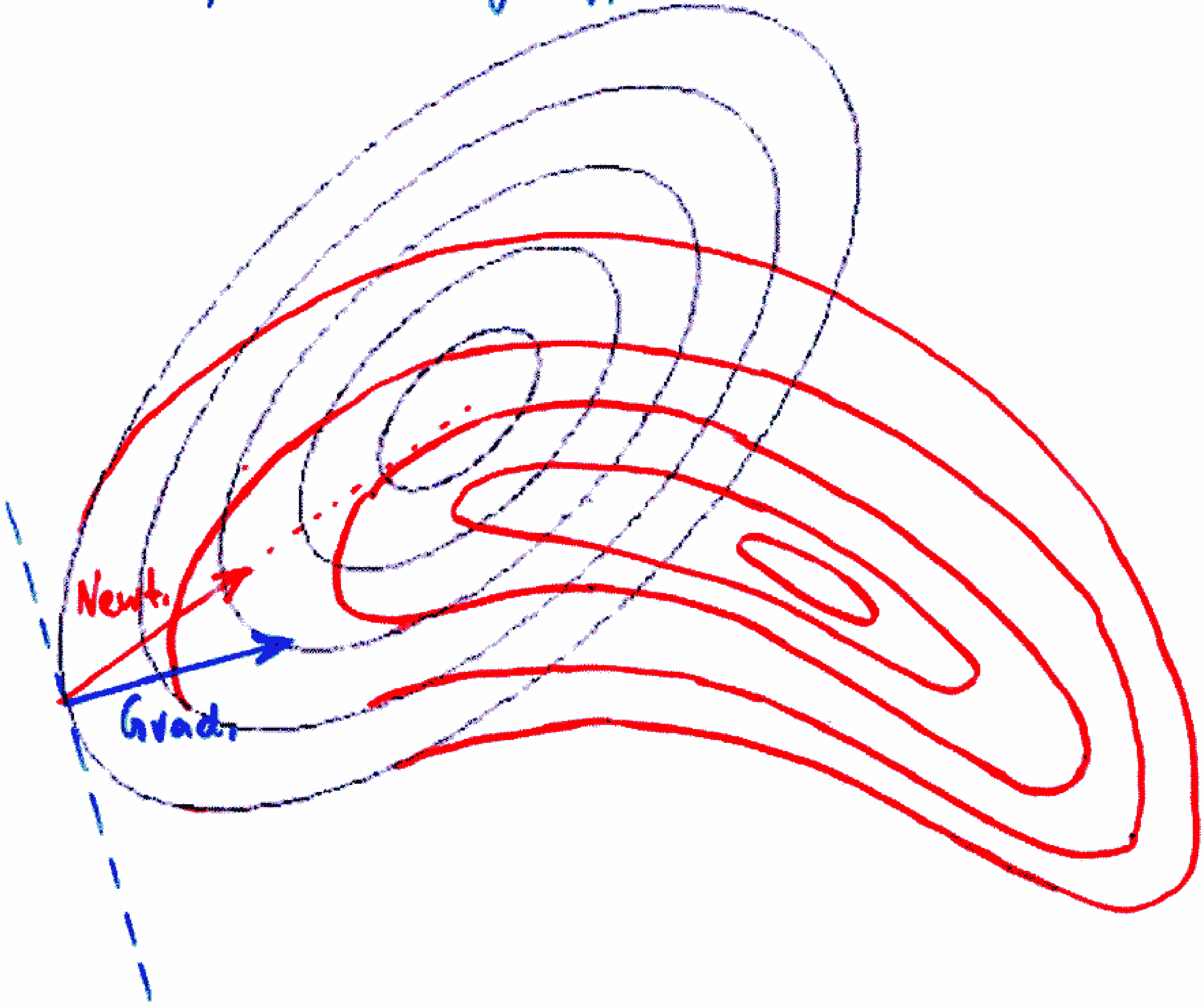
für Nullstelle von ∇f:

$$x^{(k+1)} = x^{(k)} - \frac{\nabla f(x^{(k)})}{\nabla^2 f(x^{(k)})} \quad s^{(k)}$$

Q 1
$O(n^3)$ Rechenzeit

Künstl. Gebirge IV

F unimodal, 2mal stetig differenzierbar



'genaueres' Modell liefert schlechtere Ergebnisse

immer noch : glatt
unimodal (eingipflig)
2mal stetig differenzierbar
zus. hängend
keine Störungen

Optimization Software

If you are looking for information about a specific optimization software product (and you know its name!), use the list below.

In case you are looking for software for a particular kind of optimization problem, we have also indexed the list by category.

Much of the information on these products is drawn from a book - the Optimization Software Guide (Jorge J. Moré and Stephen J. Wright, SIAM Publications, 1993). It is reproduced here by kind permission of SIAM.

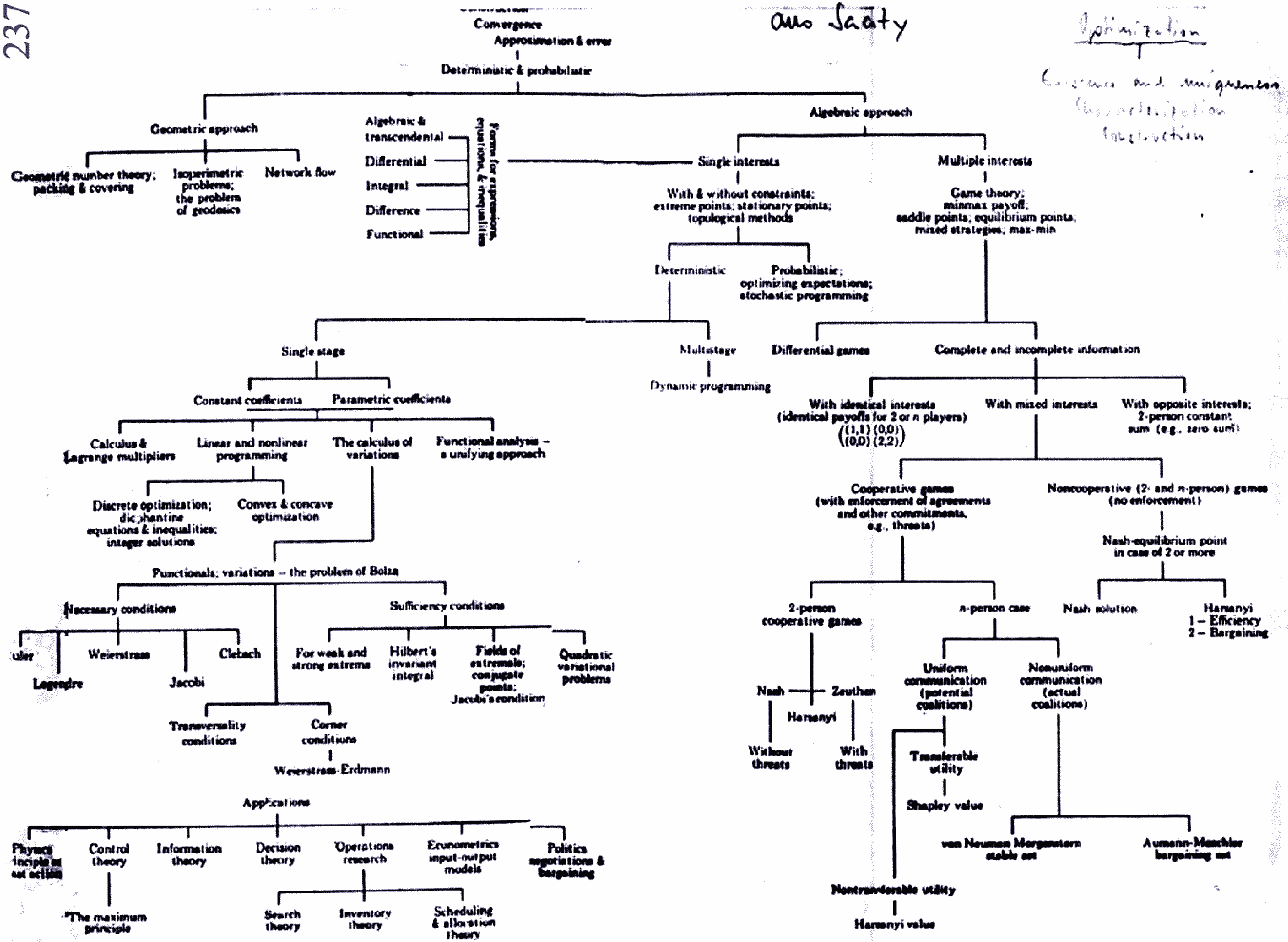
You can also create a custom home page containing the software packages you are most interested in.

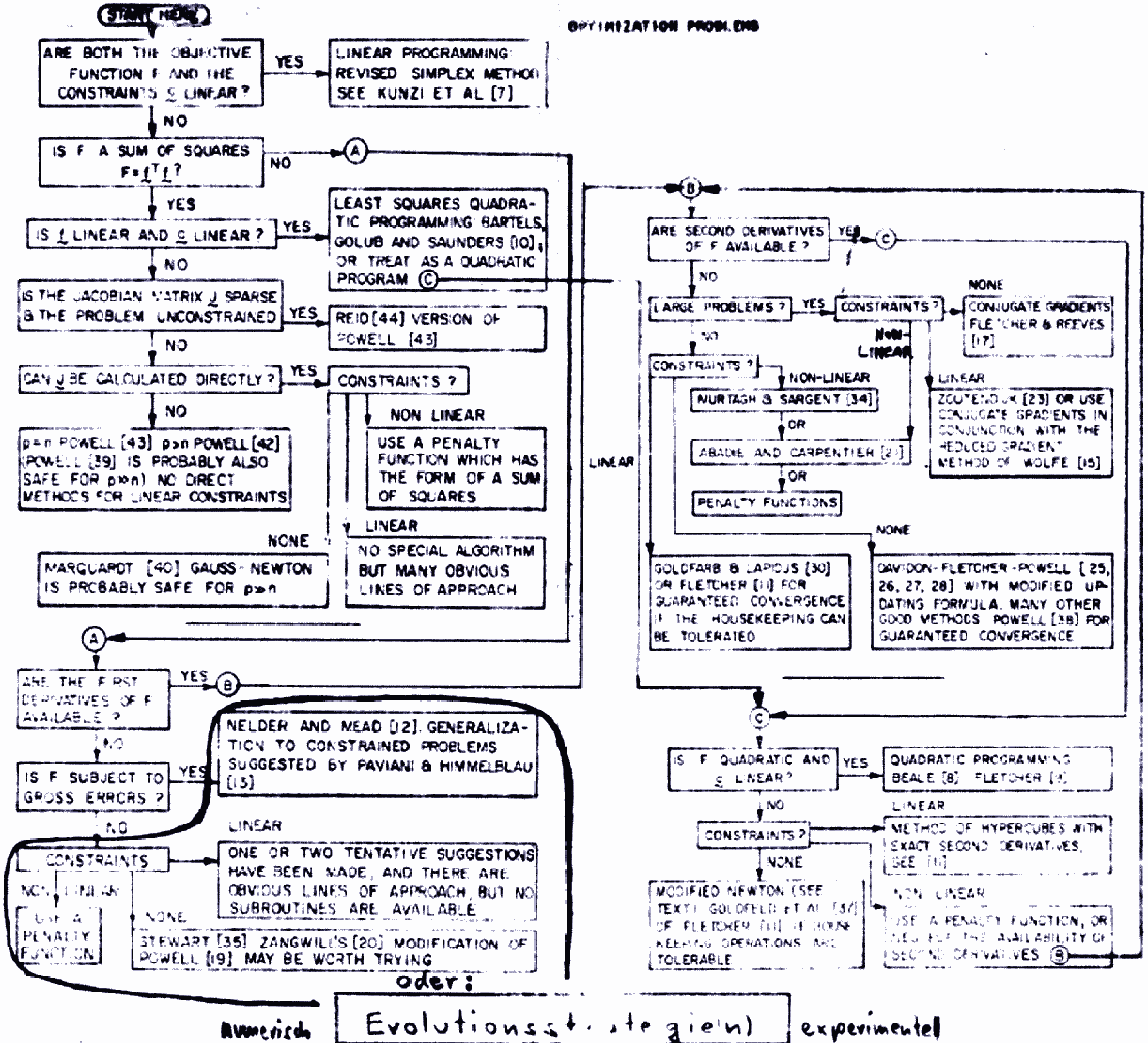
- The AMPL modeling language.
- BQPD - quadratic programming.
- BT - minimization.
- BTN - block truncated Newton.
- CNM - linear algebra and minimization.
- CONOPT - nonlinear programming.
- CONSOL-OPTCAD - engineering system design.
- CONTIN - systems of nonlinear equations.
- CPLEX - linear programming.

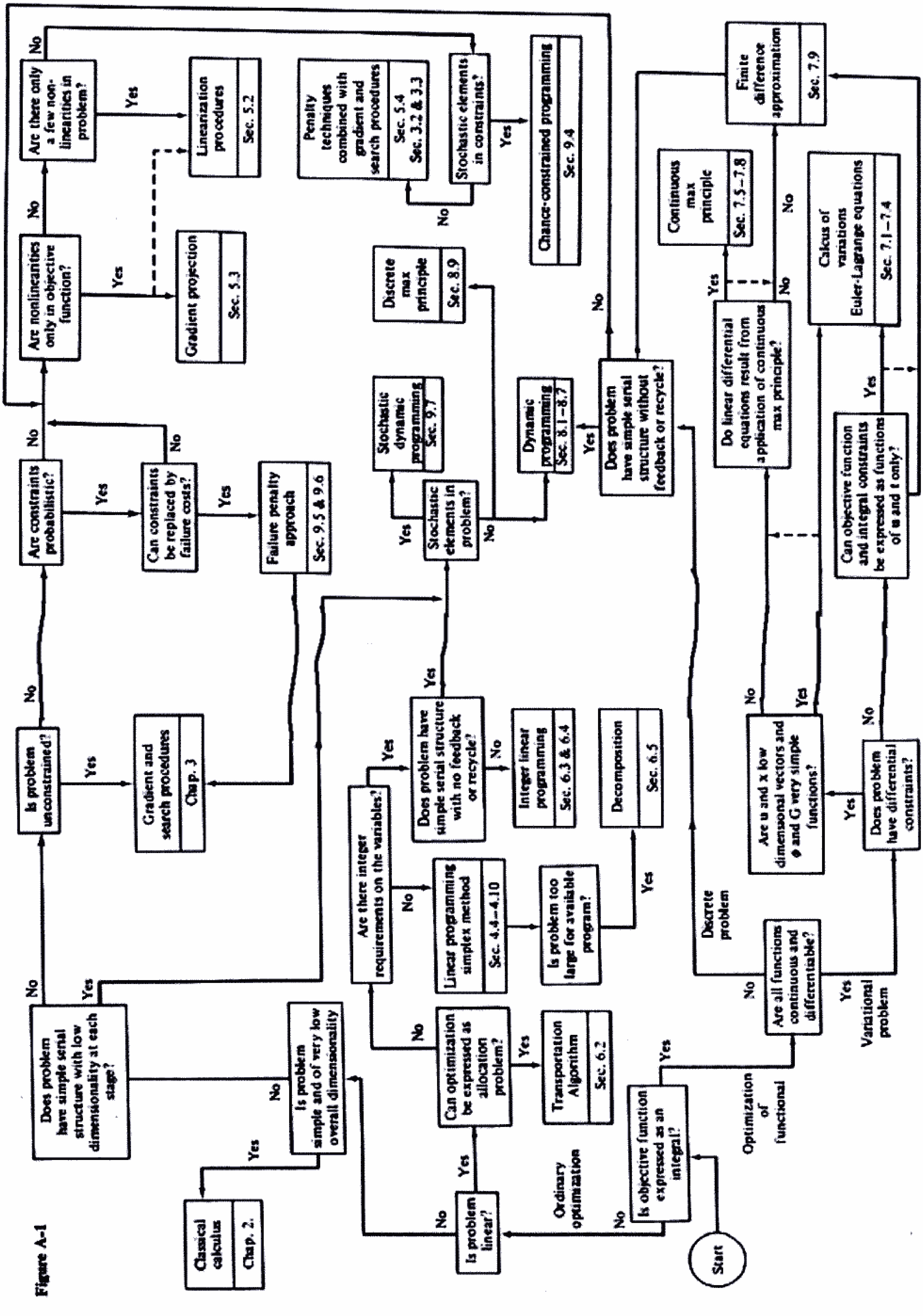
- C-WHIZ - linear programming models.
- DFNLP - nonlinear data fitting.
- DOC - Design Optimization Control Program.
- DOT - Design Optimization Tools.
- FortLP - linear and mixed integer programming.
- FSQP/CFSQP - nonlinear and minmax optimization.
- GAMS - modeling language.
- GAUSS - matrix programming language.
- GENESIS - structural optimization software.
- GENOS 1.0 - nonlinear network optimization.
- GINO - nonlinear programming.
- GRG2 - nonlinear programming.
- HOMPACK - nonlinear equations and polynomials.
- HOPDM - linear programming (interior-point).
- IMSL - Fortran and C Library.
- LAMPS - linear and mixed-integer programming.
- LANCELOT - large-scale problems.
- LBFGS - unconstrained minimization.
- LINDO - linear, mixed-integer and quadratic programming.
- LINGO - modeling language.
- LNOS - linear programming/network flow problems.
- LSGRG2 - nonlinear programming.
- LSNNO - large scale optimization.
- LSSOL - least squares problems.
- M1QN2 and M1QN3 - unconstrained optimization.
- MATLAB - optimization toolbox.
- MINOS - linear programming and nonlinear optimization.
- MINPACK-1 - nonlinear equations and least squares.
- MIPIII - mixed integer programming.
- MODULOPT - unconstrained problems and simple bounds.
- NAG C Library - nonlinear and quadratic programming, minimization
- NAG Fortran Library - nonlinear and quadratic programming, minimization

- NETFLOW - network optimization.
- NITSOL - systems of nonlinear equations.
- NLPE - minimization and least squares problems.
- NLPQL - nonlinear programming.
- NLPQLB - nonlinear programming with constraints.
- NLSFIT - parameter-estimation problems.
- NLSSOL - constrained nonlinear least squares problems.
- NLPSPR - nonlinear programming.
- NPSOL - nonlinear programming.
- ODRPACK - NLS and ODR problems
- OPSYC - OPtimisation de SYstèmes Creux.
- OptiA - optimization and quadratic programming.
- OPTIMA Library - optimization and sensitivity analysis.
- OPTPACK - constrained and unconstrained optimization.
- OSL - linear, quadratic and mixed-integer programming.
- PORT 3 - minimization, least squares, etc.
- PROC NLP - nonlinear minimization or maximization.
- Q01SUBS - quadratic programming for matrices.
- QAPP - quadratic assignment problems.
- QPOPT - linear and quadratic problems.
- SPEAKEASY - numerical problems and operations research.
- SQP - nonlinear programming.
- TENMIN - unconstrained optimization.
- TENSOLVE - nonlinear equations and least squares.
- TN/TNBC - minimization.
- TNPACK - nonlinear unconstrained minimization.
- UNCMIN - unconstrained optimization.
- VE08 - nonlinear optimization.
- VE10 - nonlinear least squares.
- VIG and VIMDA - decision support system.
- What'sBest - linear and mixed integer programming.

aus Saaty


Optimization
Existence and uniqueness
Convergence
Approximation & error
Deterministic & probabilistic
Geometric approach
Geometric number theory; packing & covering
Isoperimetric problems; the problem of geodesics
Network flow
Algebraic & transcendental
Differential
Integral
Difference
Functional
Forms for expressions, equations, & inequalities
Algebraic approach
Single interests
With & without constraints; extreme points; stationary points; topological methods
Multiple interests
Game theory; minmax payoff; saddle points; equilibrium points; mixed strategies; max-min
Deterministic
Probabilistic; optimizing expectations; stochastic programming
Single stage
Multistage
Dynamic programming
Differential games
Complete and incomplete information
Constant coefficients
Parametric coefficients
Calculus & Lagrange multipliers
Linear and nonlinear programming
The calculus of variations
Functional analysis – a unifying approach
Discrete optimization; diophantine equations & inequalities; integer solutions
Convex & concave optimization
Functionals; variations – the problem of Bolza
Necessary conditions
Sufficiency conditions
Weierstrass
Clebsch
Legendre
Jacobi
For weak and strong extrema
Hilbert's invariant integral
Fields of extremals; conjugate points; Jacobi's condition
Quadratic variational problems
Transversality conditions
Corner conditions
Weierstrass-Erdmann
Applications
Control theory
Information theory
Decision theory
Operations research
Econometrics input-output models
Politics negotiations & bargaining
"The maximum principle
Search theory
Inventory theory
Scheduling & allocation theory
With identical interests (identical payoffs for 2 or n players)
With mixed interests
With opposite interests; 2-person constant sum (e.g., zero sum)
Cooperative games (with enforcement of agreements and other commitments, e.g., threats)
Noncooperative (2- and n-person) games (no enforcement)
Nash-equilibrium point in case of 2 or more
2-person cooperative games
n-person case
Nash solution
Harsanyi
1 – Efficiency
2 – Bargaining
Nash
Zeuthen
Without threats
With threats
Uniform communication (potential coalitions)
Nonuniform communication (actual coalitions)
Transferable utility
Shapley value
von Neumann Morgenstern stable set
Aumann-Maschler bargaining set
Nontransferable utility
Harsanyi value

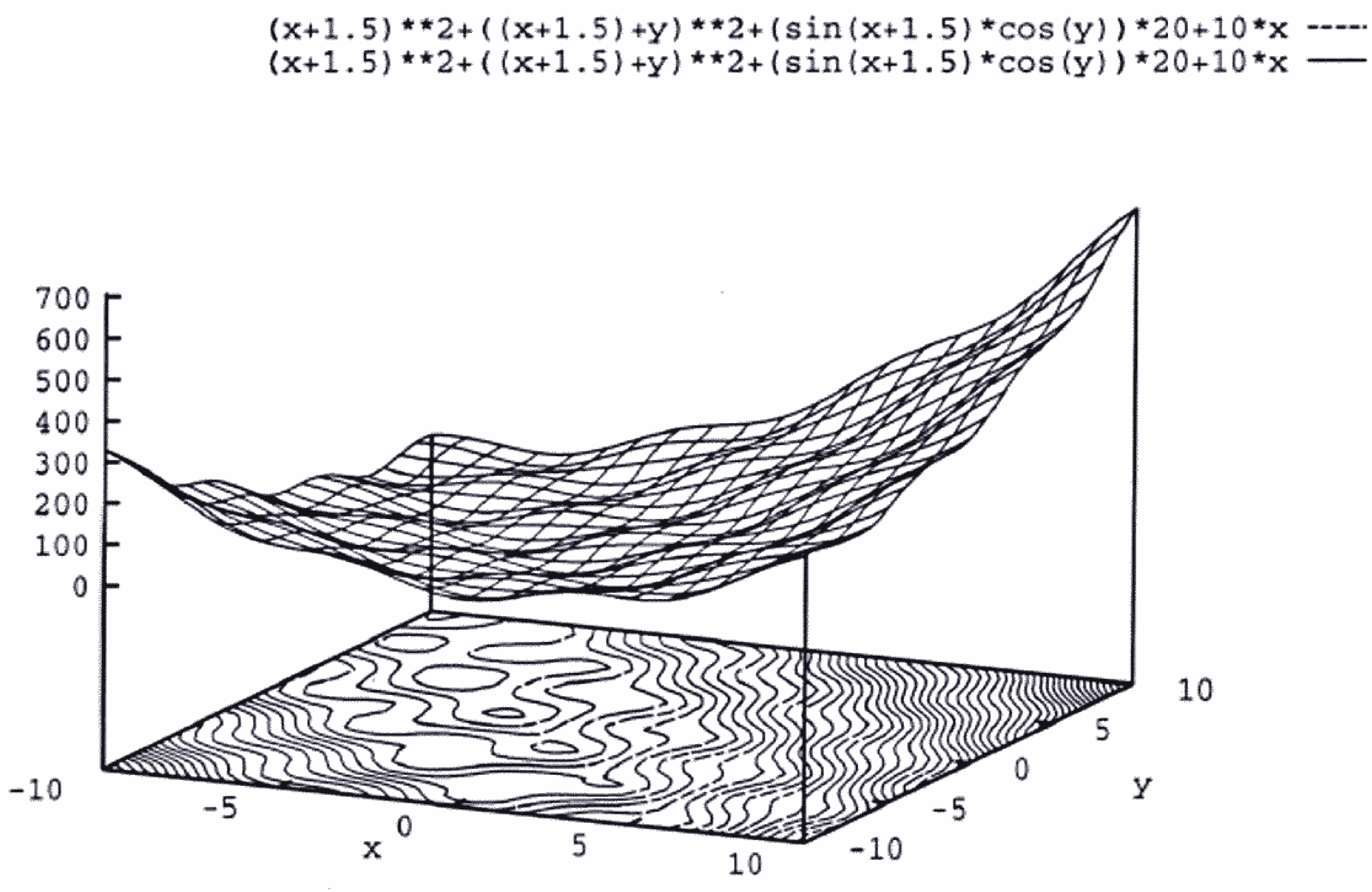

OPTIMIZATION PROBLEMS
START HERE
ARE BOTH THE OBJECTIVE FUNCTION F AND THE CONSTRAINTS c LINEAR?
YES
LINEAR PROGRAMMING: REVISED SIMPLEX METHOD SEE KUNZI ET AL [7]
NO
IS F A SUM OF SQUARES F=f^T f?
NO
A
YES
IS f LINEAR AND c LINEAR?
YES
LEAST SQUARES QUADRATIC PROGRAMMING BARTELS, GOLUB AND SAUNDERS [10], OR TREAT AS A QUADRATIC PROGRAM C
NO
IS THE JACOBIAN MATRIX J SPARSE & THE PROBLEM UNCONSTRAINED
YES
REID [44] VERSION OF POWELL [43]
NO
CAN J BE CALCULATED DIRECTLY?
YES
CONSTRAINTS?
NON LINEAR
USE A PENALTY FUNCTION WHICH HAS THE FORM OF A SUM OF SQUARES
LINEAR
NO SPECIAL ALGORITHM BUT MANY OBVIOUS LINES OF APPROACH
NO
p=n POWELL [43] p>n POWELL [42] (POWELL [39] IS PROBABLY ALSO SAFE FOR p>n) NO DIRECT METHODS FOR LINEAR CONSTRAINTS
NONE
MARQUARDT [40] GAUSS-NEWTON IS PROBABLY SAFE FOR p>n
A
ARE THE FIRST DERIVATIVES OF F AVAILABLE?
YES
B
NO
IS F SUBJECT TO GROSS ERRORS?
YES
NELDER AND MEAD [12]. GENERALIZATION TO CONSTRAINED PROBLEMS SUGGESTED BY PAVIANI & HIMMELBLAU [13]
NO
CONSTRAINTS
LINEAR
ONE OR TWO TENTATIVE SUGGESTIONS HAVE BEEN MADE, AND THERE ARE OBVIOUS LINES OF APPROACH, BUT NO SUBROUTINES ARE AVAILABLE
NON-LINEAR
USE A PENALTY FUNCTION
NONE
STEWART [35] ZANGWILL'S [20] MODIFICATION OF POWELL [19] MAY BE WORTH TRYING
B
ARE SECOND DERIVATIVES OF F AVAILABLE?
YES
C
NO
LARGE PROBLEMS?
YES
CONSTRAINTS?
NONE
CONJUGATE GRADIENTS FLETCHER & REEVES [17]
NON-LINEAR
LINEAR
ZOUTENDIJK [23] OR USE CONJUGATE GRADIENTS IN CONJUNCTION WITH THE REDUCED GRADIENT METHOD OF WOLFE [15]
NO
CONSTRAINTS?
NON-LINEAR
MURTAGH & SARGENT [34]
OR
ABADIE AND CARPENTIER [21]
OR
PENALTY FUNCTIONS
NONE
GOLDFARB & LAPIDUS [30] OR FLETCHER [11] FOR GUARANTEED CONVERGENCE IF THE HOUSEKEEPING CAN BE TOLERATED
DAVIDON-FLETCHER-POWELL [25, 26, 27, 28] WITH MODIFIED UPDATING FORMULA. MANY OTHER GOOD METHODS. POWELL [38] FOR GUARANTEED CONVERGENCE
LINEAR
C
IS F QUADRATIC AND c LINEAR?
YES
QUADRATIC PROGRAMMING BEALE [8], FLETCHER [9]
NO
CONSTRAINTS?
LINEAR
METHOD OF HYPERCUBES WITH EXACT SECOND DERIVATIVES, SEE [11]
NONE
MODIFIED NEWTON (SEE TEXT) GOLDFELD ET AL [37] OR FLETCHER [11] IF HOUSEKEEPING OPERATIONS ARE TOLERABLE
NON LINEAR
USE A PENALTY FUNCTION, OR NEGLECT THE AVAILABILITY OF SECOND DERIVATIVES B
oder:
numerisch
Evolutionsstrategien
experimentell



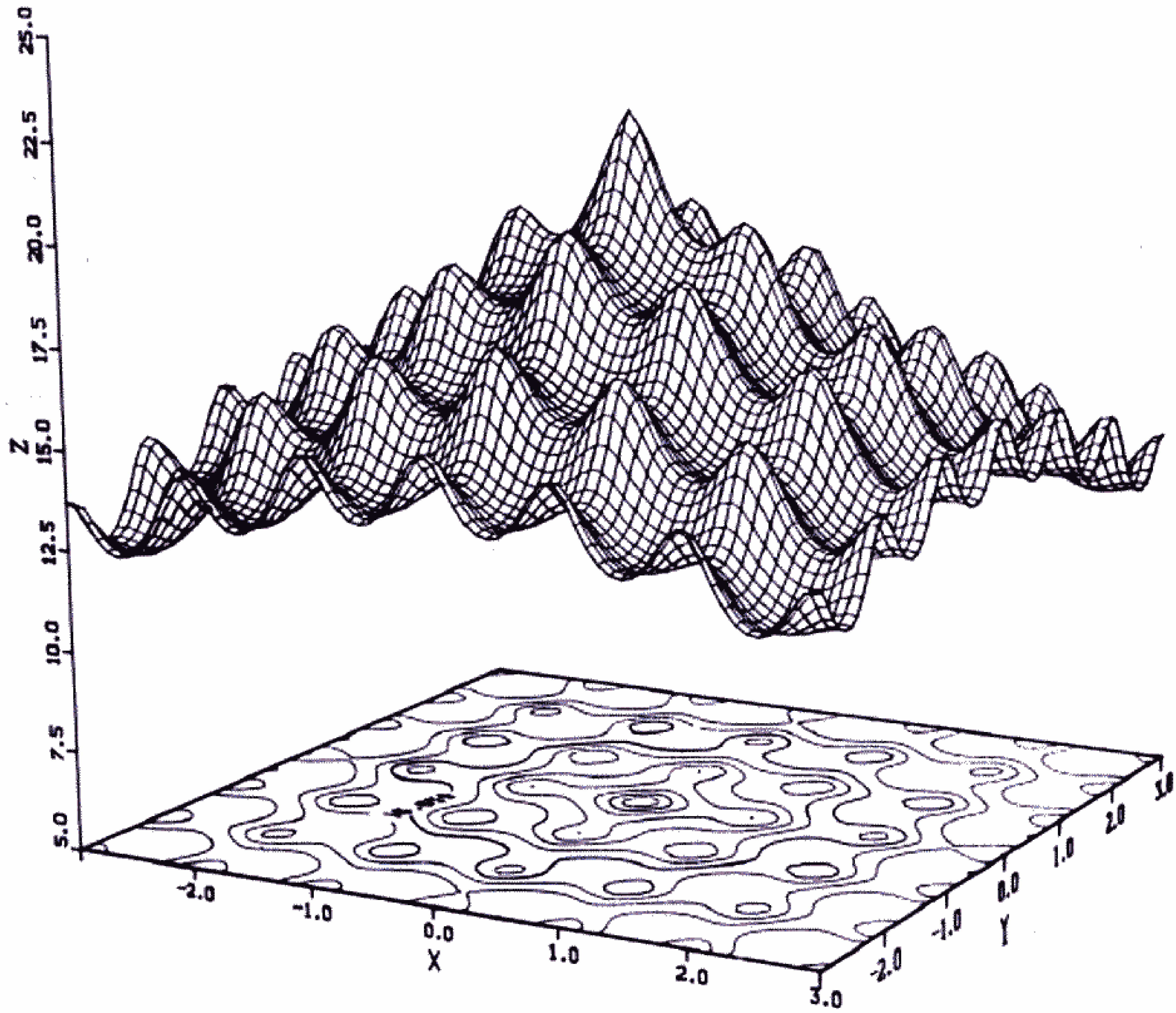
Start
Is objective function expressed as an integral?
Optimization of functional
Ordinary optimization
Is problem linear?
Can optimization be expressed as allocation problem?
Transportation Algorithm
Sec. 6.2
Is problem simple and of very low overall dimensionality
Classical calculus
Chap. 2.
Does problem have simple serial structure with low dimensionality at each stage?
Is problem unconstrained?
Gradient and search procedures
Chap. 3
Are constraints probabilistic?
Can constraints be replaced by failure costs?
Failure penalty approach
Sec. 9.5 & 9.6
Are nonlinearities only in objective function?
Gradient projection
Sec. 5.3
Are there only a few non-linearities in problem?
Linearization procedures
Sec. 5.2
Penalty techniques combined with gradient and search procedures
Sec. 5.4
Sec. 3.2 & 3.3
Stochastic elements in constraints?
Chance-constrained programming
Sec. 9.4
Are there integer requirements on the variables?
Linear programming simplex method
Sec. 4.4–4.10
Is problem too large for available program?
Decomposition
Sec. 6.5
Does problem have simple serial structure with no feedback or recycle?
Integer linear programming
Sec. 6.3 & 6.4
Stochastic elements in problem?
Stochastic dynamic programming
Sec. 9.7
Dynamic programming
Sec. 8.1–8.7
Discrete max principle
Sec. 8.9
Does problem have simple serial structure without feedback or recycle?
Are all functions continuous and differentiable?
Discrete problem
Variational problem
Are u and x low dimensional vectors and φ and G very simple functions?
Does problem have differential constraints?
Do linear differential equations result from application of continuous max principle?
Continuous max principle
Sec. 7.5–7.8
Finite difference approximation
Sec. 7.9
Can objective function and integral constraints be expressed as functions of u and t only?
Calculus of variations
Euler-Lagrange equations
Sec. 7.1–7.4
Yes
No

Figure A-1

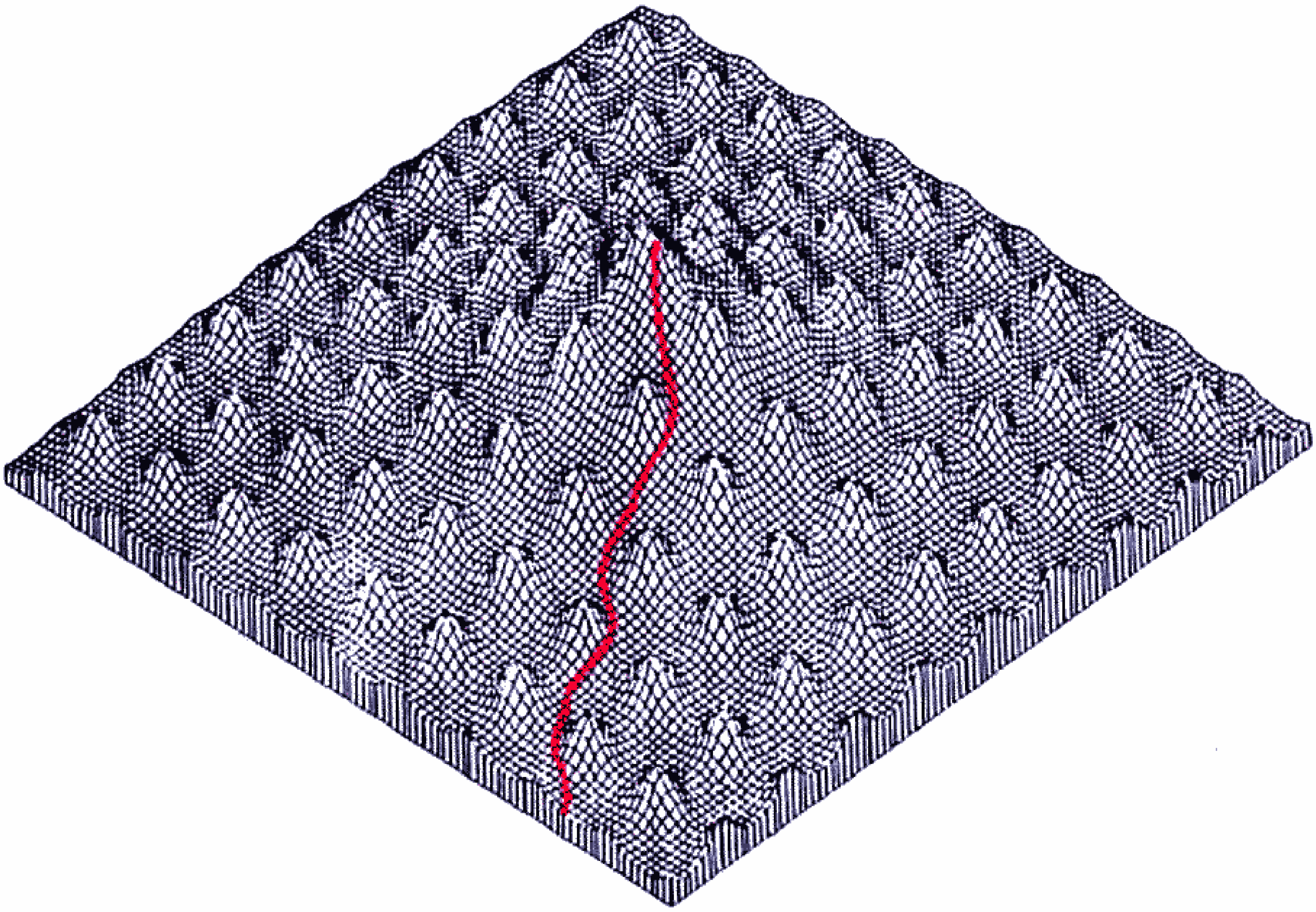
Hermes'sche Spiegelei-Funktion



Verallgemeinerte Ackley - Funktion



$$f(x) = -20\, e^{-0{,}2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$$



nichtlinear

multimodal

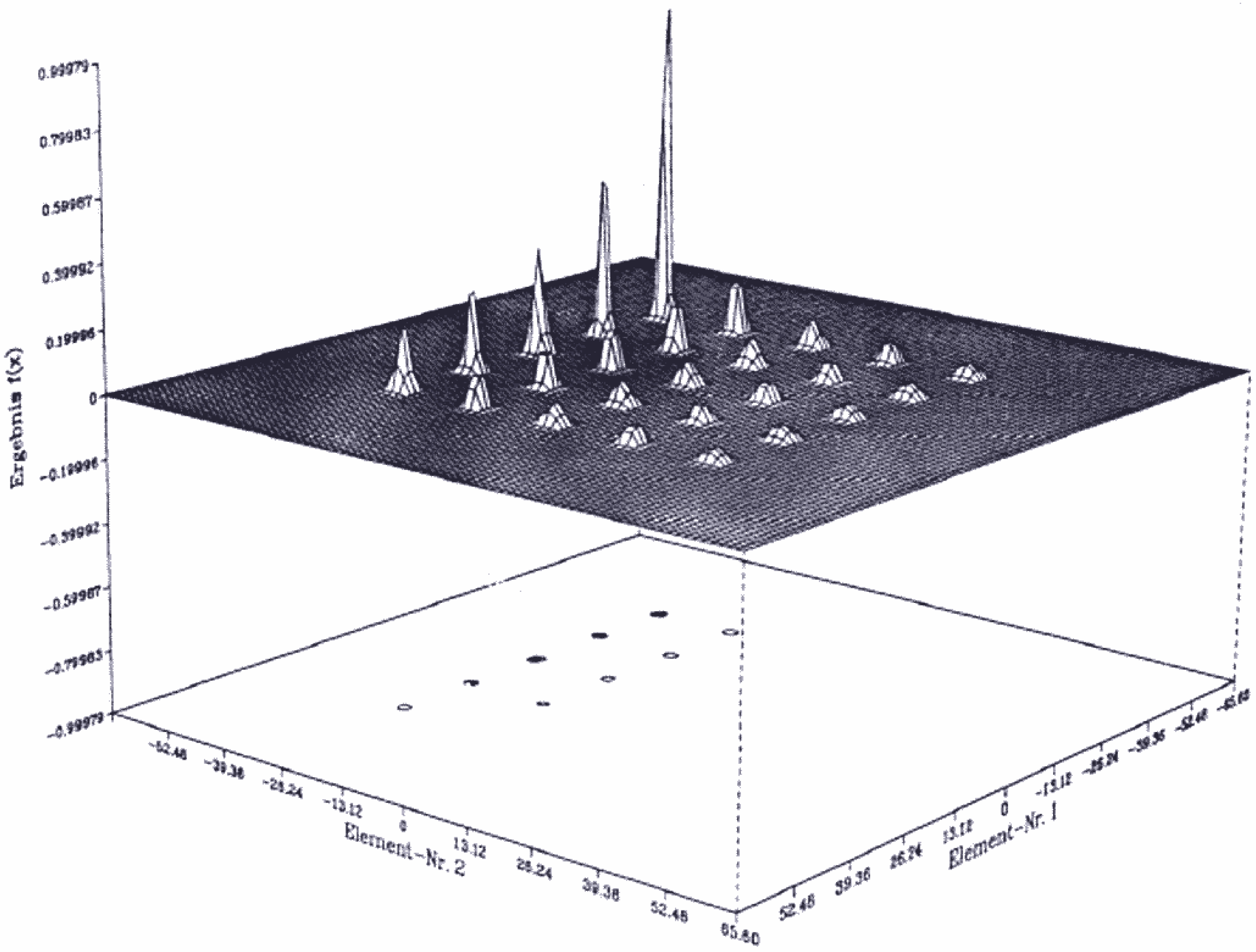
es gibt Wege 'aufwärts' zum Gipfel

Konzept 'steilster Anstieg' führt aber nur zum nächsten lokalen Maximum

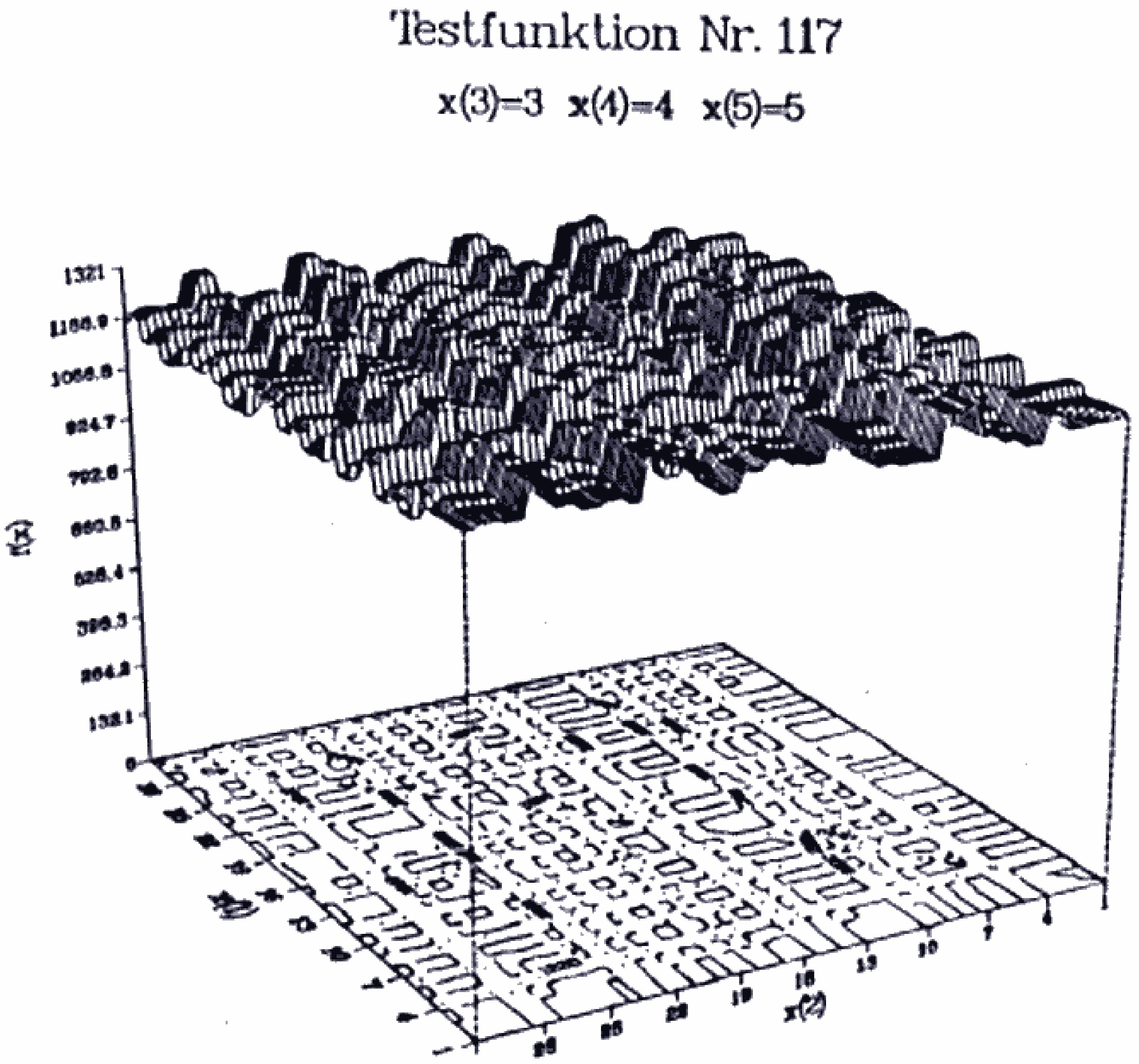
Heilmittel: Neustart von verschied. Ausgangsp.

oder: Evolutionsstrategie mit 'sanfter' Selektion

Testfunktion Nr. 80 f(x)
Konstanten – keine

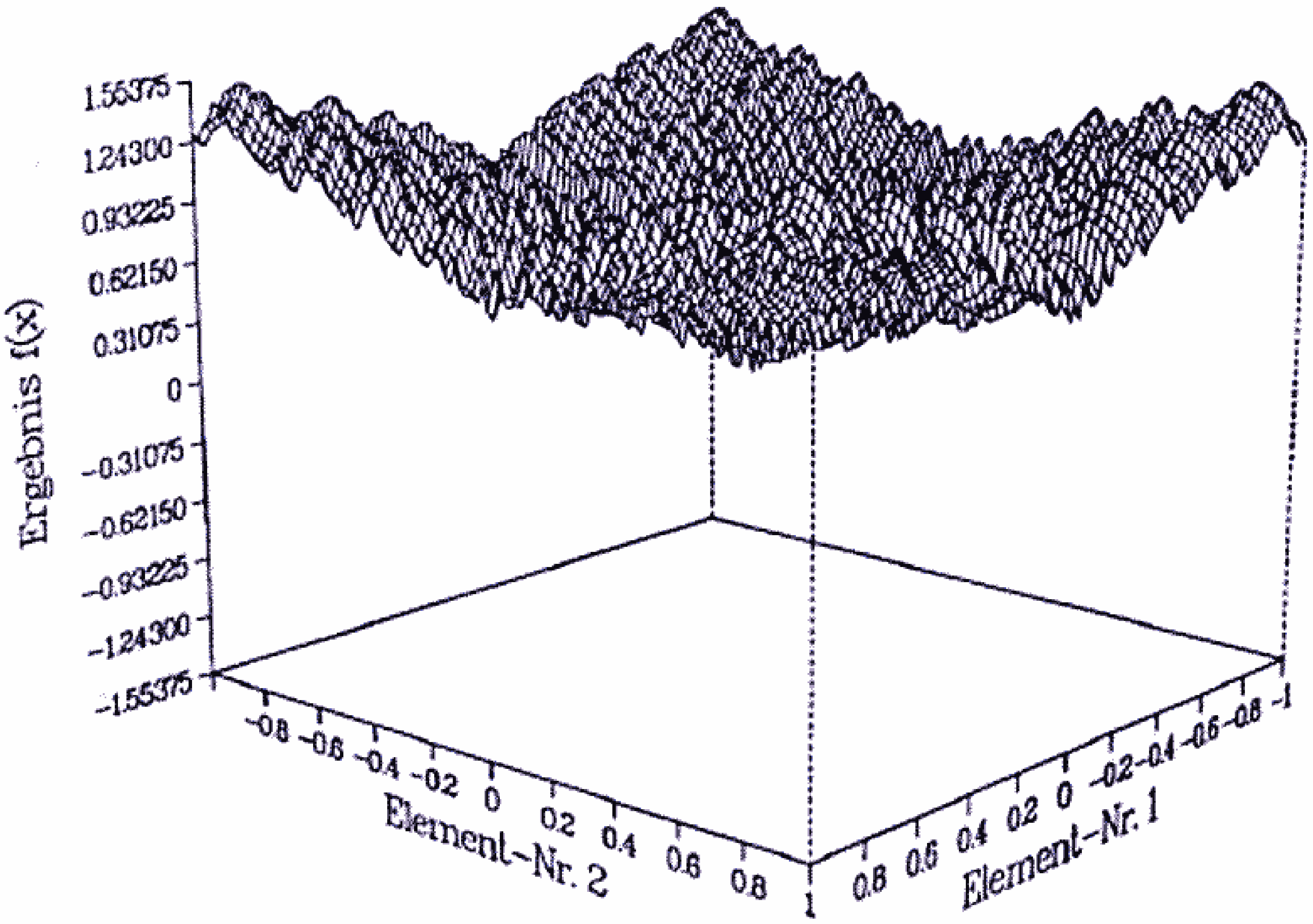


TSP-Zielfunktionstopologie (diskret)



Testfunktion Nr. 88 f(x)

Konstanten – keine

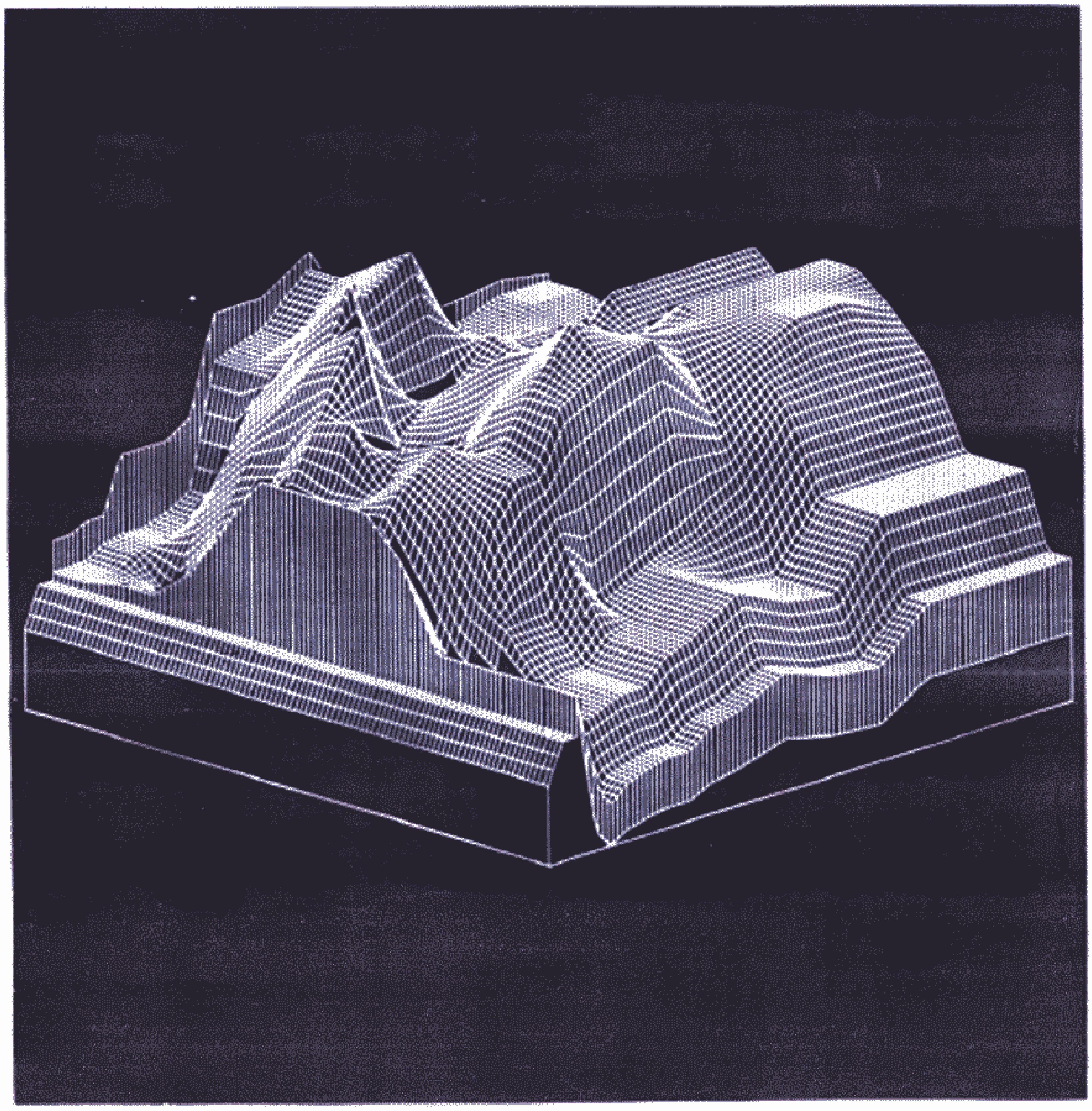


Weierstraß-Mandelbrot's fraktale Cosinus-Fkt.

Verfahren, die nicht mit infinitesimalen Schrittweiten operieren, können dem Optimum relativ nahe kommen

Kennlinienfeld einer Zündkerze

'technisches' Gebirge



nicht : linear, quadratisch

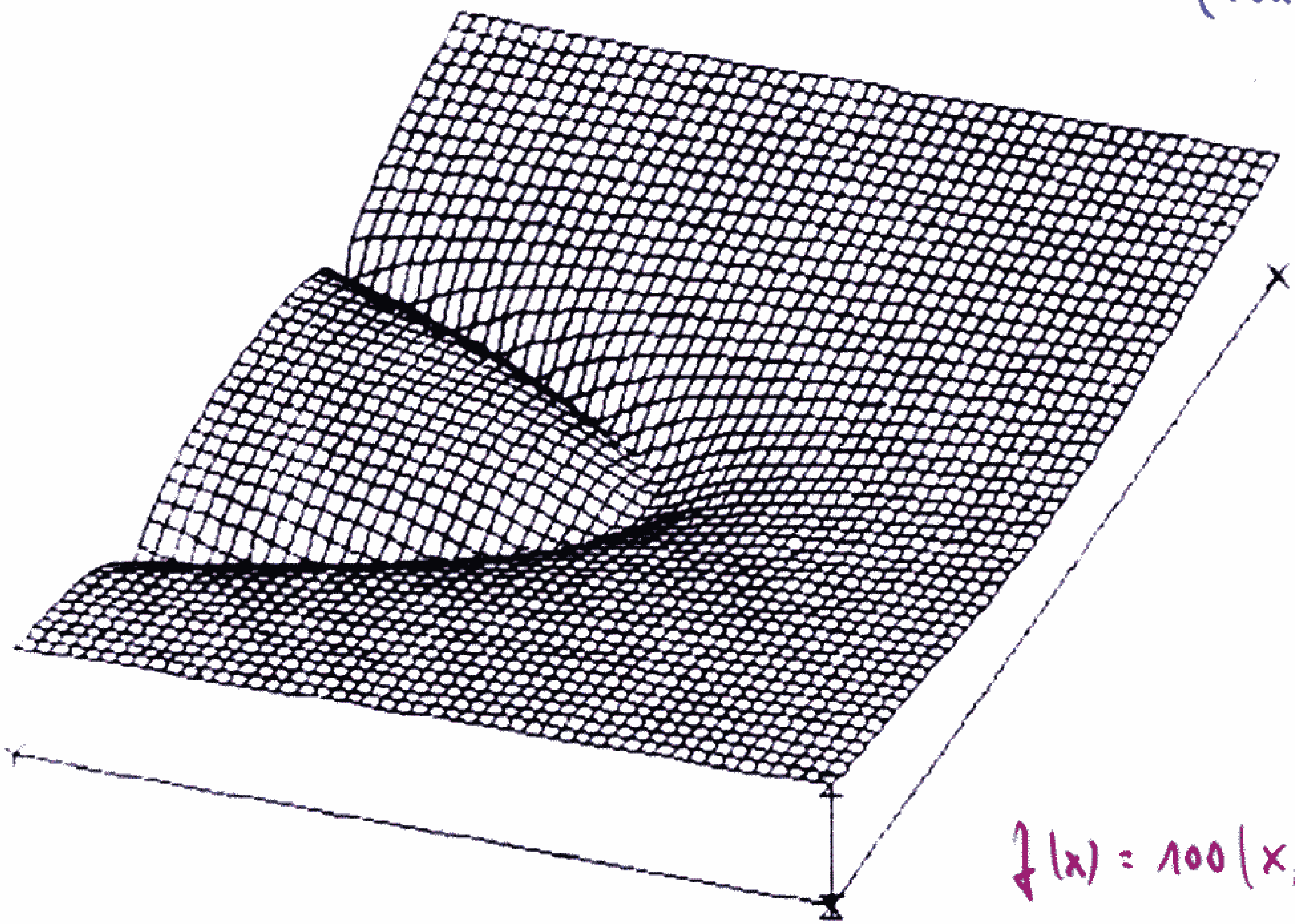
Ableitungen unstetig

multimodal : mehrere lokale Optima

(nicht zus. häng. zul. Bereich)

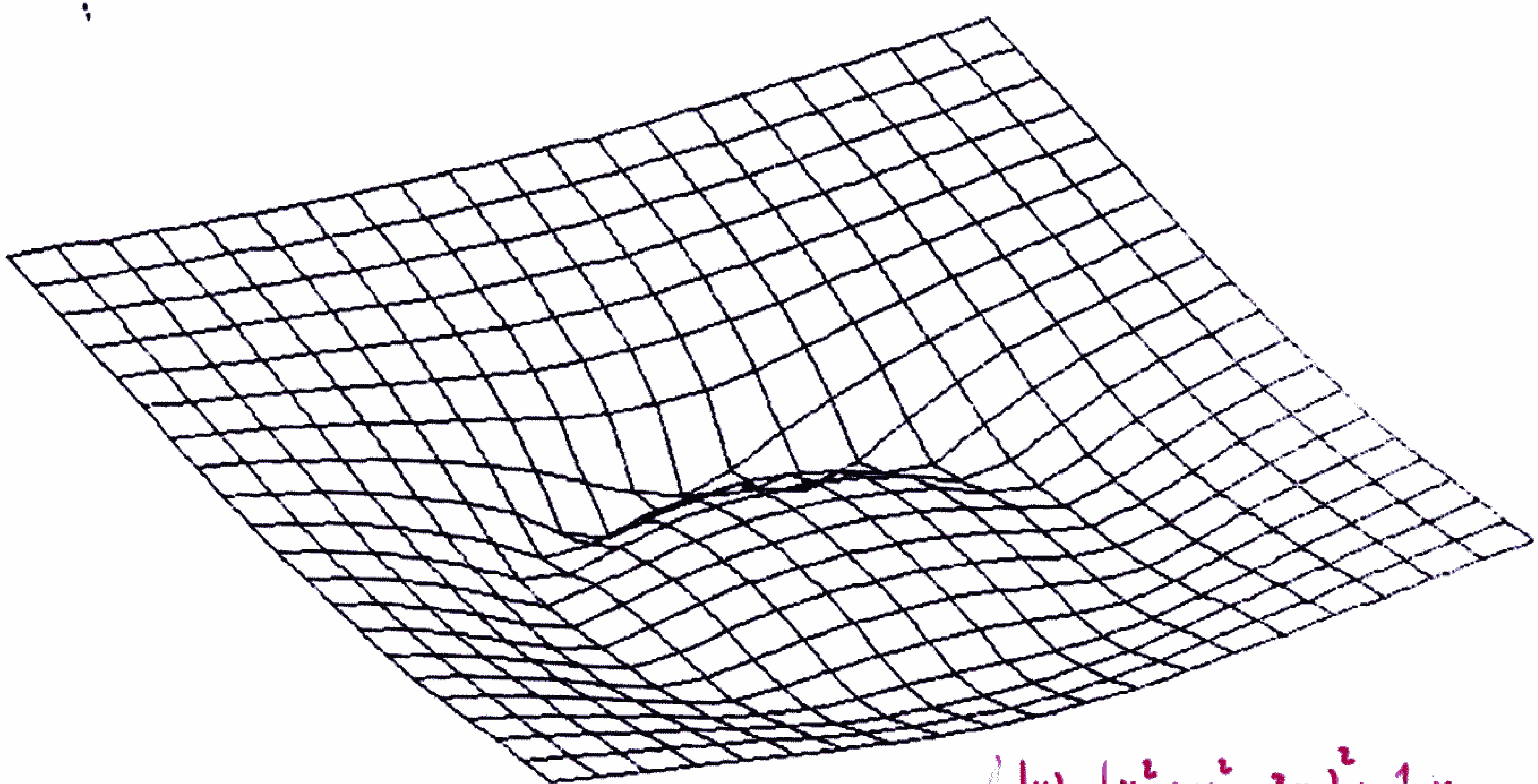
Rundungsfehler / Störungen

Bananenfunktion
(von Rosenbrock)



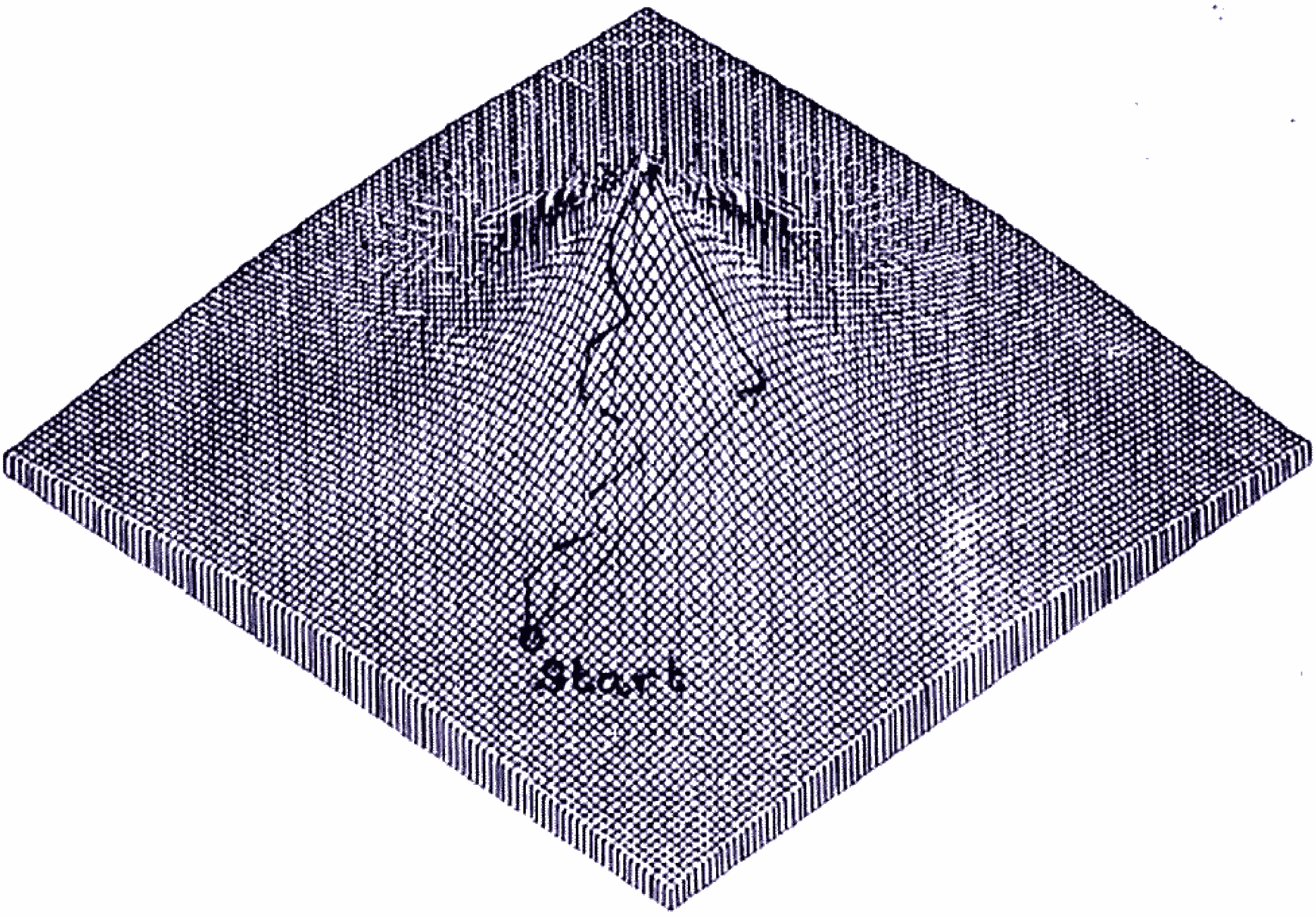
$$f(x) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2$$

Sombrerofunktion
(von Zettl)



$f(x) = (x_1^2 + x_2^2 - 2x_1)^2 + \frac{1}{4} x_1$

Vesuvius function



$$f(x) = -e^{-\sqrt{x_1^2 + x_2^2}}$$