

Numerische Optimierung - ein schneller Weg zu komplexer Modellbildung

von Reinhard OLDENBURG, Heidelberg

Einführung

Optimierung ist ein sehr flexibles Wort. Was verbirgt sich eigentlich dahinter? Kurz gesagt: Fast alles. Einige Beispiele:

- Evolution als Optimierungsprozess
- Extremalprinzipien der Physik
- Betriebswirtschaft und Politik

Natürlich gibt es extreme Unterschiede in der Frage, was jeweils optimiert wird! Und Optimierung im Mathematikunterricht? Da gibt es die vielen Extremwertaufgaben der Analysis, die sich nach dem Rezept „Ableiten und Nullsetzen“ so effektiv bearbeiten lassen. Mit diesem Ansatz schränkt man sich aber erheblich ein, wie die folgenden Beispiele zeigen werden. Näher an reale Anwendungen kommt man mit der linearen Optimierung heran. Allerdings ist das Simplexverfahren ein recht komplexer Algorithmus, und viele Probleme sind von ihrer Natur her nichtlinear.

Der Aufwand bei der numerischen Behandlung von nichtlinearen Optimierungsproblemen in mehreren Variablen ist - solange man nicht auf maximale Performanz aus ist - etwas geringer als beim Simplexverfahren, erschließt aber einen weitaus größeren Anwendungsbereich.

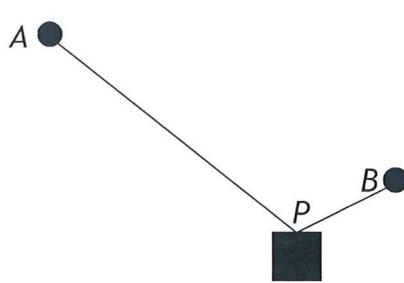
Insbesondere lassen sich reale physikalische Systeme damit modellieren und dann die Modellrechnungen mit der Realität vergleichen. Die so möglichen mathematischen Schülerexperimente "erden" den Modellbildungskreislauf (Problem - Realmodell - Mathematisches Modell - Mathematische Lösung - Validierung, vgl. Blum 1985) durch empirische Daten.

Die unmittelbare Überprüfbarkeit von Hypothesen in der Realität bewirkt, dass die Schüler immer sofort ein aus der Sache kommendes Feedback bekommen. Sie müssen den Lehrer nicht fragen, ob sie richtig gearbeitet haben. Die Lehrerrolle nähert sich der eines Trainers an, der unterstützend bei der Bewältigung der Probleme wirkt.

Den folgenden Darstellungen liegen die Erfahrungen zugrunde, die ich am Göttinger Schülerexperimentallabor mit diesem Thema machen konnte. Allerdings richtet sich die Darstellung der Inhalte an Lehrer, nicht an Schüler. Das XLAB-Skript für Schüler ist umfangreicher und stellt nicht die einzige Informationsquelle dar. Im regulären Unterricht wird man ohnehin eine Auswahl treffen müssen und die Inhalte entsprechend anpassen.

Ein erstes Beispiel

Ein dünner Faden der festen Länge L wird an zwei Haken A und B , die (damit es nicht zu einfach wird) nicht gleich hoch sind, befestigt. An den Faden wird ein Massstück gehängt, das auf ihm reibungsfrei gleiten kann. Es zieht den Faden nach unten und verleiht ihm einen Knickpunkt P . Bildlich sieht das dann folgendermaßen aus:



Wenn man nun Koordinaten für die Aufhängepunkte (z.B. $A(0, 5)$, $B(3, 2)$) und einen Wert für die Fadenlänge (z.B. $L = 8$) vorgibt, wohin rutscht dann P ? Natürlich wird die potentielle Energie minimiert, d.h. die y-Koordinate von P wird minimal unter der Nebenbedingung, dass die Strecken AP und PB zusammen die vorgegebene Länge L besitzen. Damit ist die Modellbildung abgeschlossen und wir können im Modell nach einer Lösung suchen.

Es gibt sehr viele Arten, dieses Optimierungsproblem zu lösen. In Ermangelung einer problemspezifischen Idee schauen wir erst einmal, was Excel für uns tun kann. Wenn man den „Solver“ mitinstalliert hat, ist das Problem schnell gelöst:

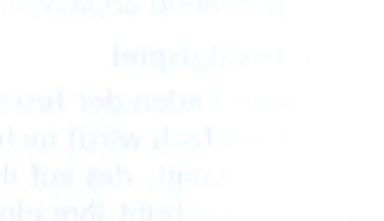
Schritt 1: Erstellen einer Arbeitsmappe mit den entscheidenden Formeln und Informationen:

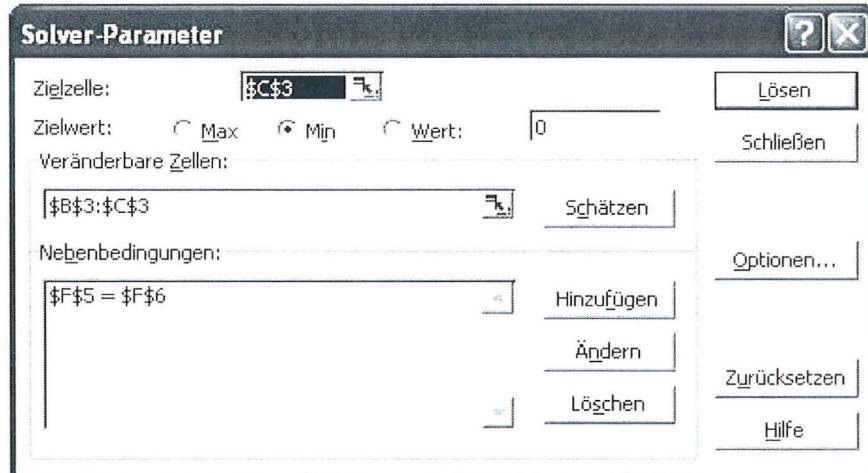
	A	B	C	D	E	F
1			Position der Masse P(x,y)			
2	x		y			
3	P		1,5	2	Abstand P, A	3,35410197
4	A		0	5	Abstand P, B	1,5
5	B		3	2	Fadenlänge	4,85410197
6					Vorgegebene Länge	8

Die Formeln in den Zellen sehen so aus:

	A	B	C	D	E	F
1			Position der I			
2	x		y			
3	P	1,5	2	Abstand P, A	=WURZEL((B3-B4)^2+(C3-C4)^2)	
4	A	0	5	Abstand P, B	=WURZEL((B3-B5)^2+(C3-C5)^2)	
5	B	3	2	Fadenlänge	=F3+F4	
6				Vorgegebene Länge	8	

Schritt 2: Aufruf des Solvers und Angabe, dass C3 minimiert werden soll und dabei B3 und C3 variiert werden dürfen, wobei F5=F6 einzuhalten ist. Im Solver-Menü sieht das folgendermaßen aus:



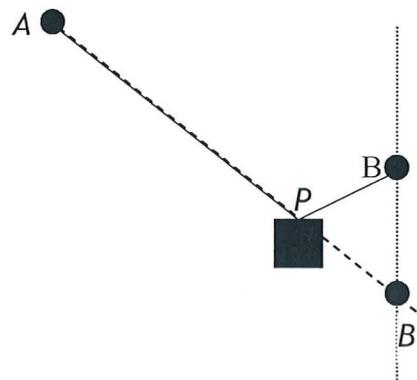


Schritt 3: Nach Klick auf den Lösen-Button findet Excel $P=(2,107; -0,208)$. Damit ist das Problem gelöst. Ein schnell aufgebautes Experiment bestätigt die Korrektheit.

Eine Rückschau: Zur Problemlösung muss das physikalische Optimierungsproblem mathematisch modelliert werden und zwar soweit, dass die zu minimierende Größe und die Nebenbedingungen als Term bzw. Gleichungen vorliegen. Dann kann der Computer die (numerische) Lösung übernehmen.

Nun zur Vorschau: Im Laufe dieses Aufsatzes werden wir viele weitere Problemstellungen auf diese Art modellieren und lösen. Dabei soll auch die Black-Box des Excel-Solvers entzaubert werden, indem Algorithmen für solche Problemstellungen entwickelt werden.

Das Problem des Gewichts am Faden kann auch durch Differentialrechnung gelöst werden (reichlich mühsam, die Theorie dazu werden wir noch kennen lernen) oder durch eine elementare geometrische Überlegung: Angenommen, P sei schon in optimaler Lage. Wenn man dann B an der Waagrechten durch P spiegelt, dann sind PB und PB' gleich lang und dann liegen A , P , B' auf einer Geraden (sonst könnte man B' und damit P tiefer ziehen). Die Entfernung AB' ist dann gleich L . Wenn umgekehrt der Punkt P noch nicht bekannt ist, kann er mit dieser Idee konstruiert werden: Man schlägt einen Kreis mit Radius L um A . Der Schnitt mit der Vertikalen durch B liefert B' . Die Mittelsenkrechte von B und B' schneidet AB' in P . En passant sei bemerkt, dass P auf einer Ellipse mit Brennpunkten A und B liegt - schließlich haben wir hier unmittelbar die Gärtnerkonstruktion vor uns.



Diese schöne geometrische Lösung hat leider den Nachteil, dass sie problemspezifisch ist: Variationen des Problems erfordern grundsätzlich neue Überlegungen. Die

rechnerische Lösung dagegen bahnt den Weg zur Beherrschung einer riesigen Problemklasse.

Grundlagen

Die Bedeutung des Wortes „Optimierung“ hängt vom Kontext ab, in dem es benutzt wird. Wir blicken von Anfang an durch eine mathematische Brille und betrachten nur solche Probleme, die - zumindest nach einer gewissen Modellbildung und Umformung - aufgefasst werden können, als Fragen nach den Stellen, an denen eine reellwertige Funktion einen minimalen Wert annimmt, d.h. wir suchen ein minimierendes Element x_0 aus der Definitionsmenge D , so dass der Funktionswert von $f: D \rightarrow \mathbb{R}$ an dieser Stelle minimal wird, d.h. $f(x_0) = \min_{x \in D} f(x)$.

Nach der Definition des Minimums gibt es dann keinen anderen Punkt in D , der zu einem kleineren Funktionswert führt: Ein solches $f(x_0)$ ist ein globales Minimum.

Natürlich interessieren oft auch lokale Minima, bei denen die Minimumseigenschaft nicht für ganz D , sondern nur für eine bestimmte Umgebung des minimierenden Punktes gilt.

Wir nennen die minimierende Stelle x_0 den Minimierer und den zugehörigen Funktionswert $f(x_0)$ das Minimum. Schließlich heißt f auch Zielfunktion.

Wenn D eine endliche Menge ist, kann das Minimierungsproblem immer gelöst werden, indem man nacheinander alle Elemente von D anschaut und unter den Funktionswerten das Minimum herausfindet:

```
Eingabe: Endliche Menge D, Funktion f
y0:=infinity;
x0:=undefined;
for all x in D do
    if f(x)<y0 then
        x0:=x; y0:=f(x0);
    end_if
end_for
Ausgabe: x0, ein Minimierer für f auf D
```

Allerdings ist in vielen interessanten Fällen D viel zu groß, als dass es vollständig aufgezählt werden könnte. Je nach Problemstellung muss man sich dann eines spezialisierten Programms bedienen.

Minimierung und Maximierung sind äquivalente Probleme: Wo f ein Minimum hat, hat $-f$ ein Maximum und umgekehrt. Es reicht also aus, Algorithmen für das Minimieren zu finden.

Man kann Minimierungsprobleme einteilen nach verschiedenen Kriterien, z.B.

- Eigenschaften der Menge D , sie kann sein:
 - diskret
 - (eine Teilmenge der) reellen Zahlen
 - (eine Teilmenge eines) Vektorraums
 - ein unendlichdimensionaler Raum
- Eigenschaften der Zielfunktion, sie kann sein:
 - stetig
 - differenzierbar

In diesem Aufsatz werden die diskreten Optimierungsprobleme ausgeklammert, obwohl sie in der Anwendung sehr wichtig und nach allen Unterrichtserfahrungen Schüler damit sehr gut umgehen können. Gute Materialien stellt das Berliner Projekt „Diskrete Mathematik für die Schule“ bereit.

Problemliste

Hier folgt eine weitgehend ungeordnete Liste von Problemen, die mit den Methoden, die später beschrieben werden, gelöst werden können. Der weitere Text enthält dann nur noch einfache Beispiele (zur Illustration) und komplexere (bei denen zumindest ein Teil der Lösung beschrieben werden muss, weil sie sonst zu schwierig wären).

Einige der Probleme können rein symbolisch geschlossen gelöst werden, andere erfordern, dass man numerische Werte verwendet (wenn sie nicht angegeben sind, muss man sie ggf. erfinden.).

Die methodische Idee hinter einer so gestalteten Problemliste ist, den Schülern Gelegenheit zu geben, die Passung von Problem und Methode selbst zu finden.

Viele der Probleme sind „Klassiker“, das ist im Sinne der Vernetzung und Anknüpfung an bekannte Inhalte durchaus beabsichtigt.

Optimale Konservendose

Welche zylindrische Konservendose von 850 cm^3 Volumen hat minimalen Blechverbrauch?

Wärmestrahlung

Jeder Körper sendet aufgrund seiner Temperatur elektromagnetische Strahlung, die Wärmestrahlung aus. Diese Strahlung enthält praktisch alle Wellenlängen, allerdings mit sehr unterschiedlichen Intensitäten. Bei einer festen Temperatur T (gemessen in Kelvin) ist die Intensität der Strahlung mit Wellenlänge λ proportional zu

$$\rho(\lambda, T) = \frac{\lambda^{-5}}{e^{T\cdot\lambda} - 1}, \text{ wobei } a = 0,01439 \cdot K \cdot m \text{ eine Naturkonstante ist. Dieses Planck-}$$

sche Strahlungsgesetz stand am Anfang der Quantenphysik.

Bei welcher Wellenlänge liegt die maximale Intensität bei 1000 K, bei einem Glühdraht einer Glühlampe (2000°C) oder bei der Sonnenoberfläche (sichtbares Spektrum 700 nm (rot) bis 400 nm (violett))?

Schnellster Weg I

Gegeben zwei Punkte A,B, die auf der selben Seite einer Geraden g liegen. Was ist der schnellste Weg von A nach B, der einmal g berührt?

Schnellster Weg II

Stadt B liegt an einer geraden Straße g , Stadt A irgendwo abseits der Straße. Auf der Straße bewegt man sich mit einer Geschwindigkeit von c_2 , abseits nur mit c_1 (z.B. $c_1 = 5$, $c_2 = 10$). Wie kommt man möglichst schnell von A nach B?

Werte mittendrin

Gegeben eine Reihe von Zahlen x_1, \dots, x_n . Welche Zahl m liegt am besten in der Mitte in dem Sinne, dass die Summe der Abstände $\sum_{i=1}^n |m - x_i|$ minimal wird, bzw.

dass die Quadratsumme $\sum_{i=1}^n (m - x_i)^2$ minimal wird?

Fermatpunkt:

Wo liegt in einem Dreieck ABC der Punkt P, für den die Abstandssumme $d(A, P) + d(B, P) + d(C, P)$ minimal wird ($d(A, P)$ steht für den Abstand von A und P)?

Verallgemeinerungen: Viereck statt Dreieck, unterschiedliche Gewichtung verschiedener Punkte.

Lemoine-Punkt:

Welcher Punkt P in einem Dreieck ABC liegt am nächsten an allen Seiten in dem Sinne, dass die Quadratsumme der Abstände von den Strecken minimal ist?

Regression, d.h. verschiedene Kurven-Fits:

- Kreis-Fit: Welcher Kreis passt am besten zu den Punkten (4, 2), (2, 4), (0, 5), (-1, 4), (-1, 3.7), (-3, 3)?
- Fitten an nichtlineare Funktionen: Stattd $y = m \cdot x + b$, wie bei der linearen Optimierung, können auch $y = a \cdot \exp(b)$ oder $y = a \cdot x^b + c$ oder $y = a \cdot \sin(b \cdot x + c)$ verwendet werden.

Optimale Approximation

Taylor-Reihen sind eine Möglichkeit, Funktionen zu approximieren, beispielsweise ist das Taylor-Polynom dritten Grades des Sinus gegeben durch $\sin(x) \approx x - \frac{x^3}{6}$. Fin-

den Sie ein Polynom dritten Grades $f(x) = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$, das auf dem ganzen Intervall $[0, \pi]$ die Sinusfunktion möglichst gut approximiert. Dazu soll das In-

tegral über das Fehlerquadrat $\int_0^\pi (\sin(x) - f(x))^2 \cdot dx$ minimal werden.

Eindimensionale Optimierung

Der Ausgangspunkt unserer Reise in die Welt der Optimierung ist das bekannte notwendige Kriterium, nach dem eine differenzierbare Funktion f , die in x_0 ein lokales Extremum hat, dort die Gleichung $f'(x_0) = 0$ erfüllt.

Beispiel: Gesucht ist der Punkt auf dem Einheitskreis um den Koordinatenursprung, der am nächsten an $A(4, 3)$ liegt.

Lösung: Der Kreis wird parametrisiert durch $x = \cos \varphi$; $y = \sin \varphi$, und der Abstand wird nach Pythagoras berechnet zu

$$d(\varphi) = \sqrt{(x - 4)^2 + (y - 3)^2} = \sqrt{(\cos \varphi - 4)^2 + (\sin \varphi - 3)^2}. \text{ Die Ableitung davon ist}$$

$\frac{8 \sin \varphi - 6 \cos \varphi}{2 \cdot \sqrt{26 - 8 \cos \varphi - 6 \sin \varphi}}$. Nach dem Nullsetzen muss man sich glücklicherweise nur noch um den Zähler kümmern und findet die beiden Lösungen $\varphi = \arctan \frac{3}{4}$ und $\varphi = \arctan \frac{3}{4} - \pi$, ein Maximum und ein Minimum, wie auch geometrisch einleuchtend ist.

Der Schritt der Parametrisierung ist essentiell, um das obige Verfahren anwenden zu können. Natürlich können noch symbolische Parameter auftauchen, aber es darf nur eine unabhängig veränderliche Variable geben. Das macht die zweite der folgenden Übungen etwas knifflig:

Weitere Übungen:

- Gleiches Problem wie eben, aber mit einer Ellipse, deren große Halbachse in Richtung der x-Achse liegt und Länge 4 hat und deren kleine Halbachse Länge 2 hat.
- Wenn kein Strom entnommen wird, besitzt eine Batterie eine Leerlaufspannung U_0 . Bei Belastung geht die gemessene Spannung U unter diesen Wert zurück. Das kann man durch den Innenwiderstand erklären: Man denkt sich die reale Batterie als Reihenschaltung einer idealen Batterie, die immer die Spannung U_0 besitzt, und einem Innenwiderstand R_i . Die außen gemessene Spannung ist dann nur $U = U_0 - R_i \cdot I$. Bei welchem Wert des externen Widerstandes R_e wird an ihm die maximale Leistung $P = U \cdot I$ frei?

Zur Lösung von b): Nach der Ohmschen Beziehung ist $U_0 = (R_e + R_i) \cdot I$. Daraus kann der Strom berechnet werden und eingesetzt werden in:

$$P = U \cdot I = (U_0 - R_i \cdot I) \cdot I = \left(U_0 - R_i \cdot \frac{U_0}{R_e + R_i} \right) \cdot \frac{U_0}{R_e + R_i}. \text{ Dies ist ein Term, der außer von } R_e \text{ nur noch von festen Parametern abhängt.}$$

Numerische Methoden

Der bisher beschrittene Weg der Analysis kann aus verschiedenen Gründen scheitern, z.B.

- könnte die Funktion nicht differenzierbar sein,
- könnte die Ableitung sehr komplex sein,
- könnte die notwendige Bedingung eine schwer oder gar nicht lösbare Gleichung sein.

In all diesen Fällen bietet es sich an, numerisch nach einer Lösung zu suchen.

Numerische Optimierung ohne Benutzung von Ableitungen

Jeder, der programmieren oder eine Tabellenkalkulation benutzen kann, ist in der Lage, numerisch zu minimieren. Man gibt eine Funktion f vor, die im Intervall $[a, b]$ ein Minimum hat, z.B. $\cos(x)$ auf $[0, 2\pi]$ - dann wissen wir ja schon, was herauskommen muss - und los geht's: Man könnte das Intervall äquidistant unterteilen und in der Umgebung des kleinsten gefundenen Funktionswertes die Schrittweite verkleinern. In einer Tabellenkalkulation realisiert man das folgendermaßen:

	A	B	C	D	E
1	x	cos(x)	x	cos(x)	
2		0	1		
3		0,5	0,87758256		
4		1	0,54030231		
5		1,5	0,0707372		
6		2	-0,41614684		
7		2,5	-0,80114362		
8		3	-0,9899925		
9		3,5	-0,93645669		
10		4	-0,65364362		
11		4,5	-0,2107958		
12		5	0,28366219		
13		5,5	0,70866977		
14		6	0,96017029		
15		6,5	0,97658763		
16		7	0,75390225		
17					

Das Verfahren ist ganz ähnlich der numerischen Bestimmung einer Nullstelle mit Vorzeichenwechsel. Es gibt aber einen entscheidenden Unterschied: Dort braucht man nur zwei Stellen a und b mit verschiedenem Vorzeichen, um (bei einer stetigen Funktion) sicher sein zu können, im Intervall $[a; b]$ eine Nullstelle zu finden. Aus der obigen Excel-Tabelle erkennt man aber, dass zwei Stellen für ein Minimum nicht ausreichen. Allerdings kann man mit drei Zahlen $a < b < c$ ein Minimum einsperren oder „einklammern“:

Wenn für $a < b < c$ auch $f(b) < f(a)$ und $f(b) < f(c)$ gilt, dann liegt zwischen a und c ein Minimum der stetigen Funktion f . Man sagt: Die drei Zahlen bilden eine Einklammerung des Minimums.

Auf Grundlage dieser Erkenntnis kann man ein Bisektionsverfahren entwickeln, das abwechselnd das linke oder das rechte Teilintervall verkleinert.

Man wählt einen neuen Punkt x , entweder aus $[a, b]$ oder aus $[b, c]$, und vergleicht seinen Funktionswert mit dem der Nachbarn. Angenommen, wie haben x aus $[a, b]$ gewählt, und es ist $f(x) > f(b)$, dann ist $x < b < c$ eine neue, kleinere Einklammerung des Minimums. Falls $f(x) < b$, dann ist $a < x < b$ ebenfalls eine neue, kleinere Einklammerung. Nur im Fall $f(a) = f(b)$ kommt man nicht zu einer vernünftigen Entscheidung und muss ein neues x wählen.

Durch Wiederholung dieser Verkleinerung erhält man einen iterativen Algorithmus, der ein Minimum mit beliebiger, vorgegebener Genauigkeit finden kann.

Aufgaben:

- Beweisen Sie, dass der Algorithmus korrekt arbeitet, d.h. dass das neue Tripel wieder die Klammerungsbedingung erfüllt.
- Implementieren Sie den Algorithmus.

Eine Lösung zu Aufgabe b) kann in Pseudocode (der sich aber fast genauso z.B. in MuPAD realisieren lässt) folgendermaßen aussehen:

```

Eingabe: Funktion f, Einklammerung a,b,c
if f(a)<f(b) or f(c)<f(b) or a>b or b>c or a>c then error("Error!")
while c-a>0.00001 do

```

```

x:=(b+c)/2
if f(x)>f(b) then c:=x; else a:=b; b:=x
x:=(a+b)/2
if f(x)>f(b) then a:=x else c:=b; b:=x
end_while;

```

Ausgabe: b, Näherung an die gesuchte Minimumsstelle, oder [a;c] als umgebendes Intervall

Für eine möglichst bequeme Anwendung ist es sinnvoll, auch eine Prozedur zu schreiben, die nur einen Startwert benötigt und dann eine erste Einklammerung liefert, mit der man den obigen Algorithmus füttern kann. Das ist nicht all zu schwierig, wer aber auf Effizienz Wert legt, sollte sich in den „Numerical recipes“ (Press et al. 1998) umschauen.

```

Eingabe: Funktion f, Startwert x0
eps:=1e-6
f0:=f(x0)
x1:=x0+eps
f1:=f(x1)
if f0>f1 then // suche nach rechts
    x2:=x1+eps; s:=eps
    while f(x2)<=f1 and x2<1e30 do x2:=x2+s; s:=s*1.2; end_while;
    if f(x2)>f1 then return([x0,x1,x2]) else return FAIL; end_if;
else // suche nach links
    x2:=x0-eps; s:=eps
    while f(x2)<=f1 and x2>-1e30 do x2:=x2-s; s:=s*1.2; end_while;
    if f(x2)>f1 then return([x2,x0,x1]) else return FAIL; end_if;
end_if

```

Eindimensionale Optimierung mit Ableitungen

Bei differenzierbaren Funktionen kann man ihre Glattheit ausnützen, um eine schnellere Konvergenz zu erzielen.

Beim Newton-Verfahren geht man aus von der Taylor-Entwicklung der Zielfunktion um den Startpunkt x_0 . Anschaulich gesprochen approximiert diese den Funktionsverlauf in der Nähe von x_0 durch eine Parabel, die an der Stelle x_0 in Funktionswert, Steigung und Krümmung mit dem Graphen von f übereinstimmt:

$$f(x) \approx f(x_0) + f'(x_0) \cdot (x - x_0) + \frac{1}{2} f''(x_0) \cdot (x - x_0)^2 =: p(x)$$

Als neuer Näherungswert für eine Minimumsstelle von f wird nun das Minimum von p genommen.

$$p'(x) = f'(x_0) + (x - x_0) \cdot f''(x_0)$$

$$p'(x_1) = 0 \Leftrightarrow f'(x_0) + (x_1 - x_0) \cdot f''(x_0) = 0 \Leftrightarrow x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)}$$

Die Iteration dieses Schrittes stellt das Newton-Verfahren dar. Es konvergiert oft extrem schnell, manchmal aber auch gar nicht. Vor allem aber ist man darauf angewiesen, dass der Startwert nahe bei einem Minimum liegt, weil man sonst z.B. gegen ein Maximum läuft. Gänzlich unbrauchbar ist es, wenn der Startwert eine flache Stelle ($f''(x_0) = 0$ oder sehr klein) ist.

Auch wenn es mehrere lokale Extrema gibt, hat das Verfahren seine Tücken: Es ist nämlich, außer in der unmittelbaren Nähe eines Optimums, nicht absehbar, gegen welches Extremum es konvergieren wird. Die Situation ist vollkommen analog zur Lösung von Gleichungen mit dem Newtonverfahren, wo bekanntermaßen sogar fraktale Einzugsgebiete existieren.

Die Implementierung des Verfahrens ist, zumal in einem CAS, wo man die Ableitung symbolisch bekommt, kein Problem.

Probleme in mehreren Dimensionen

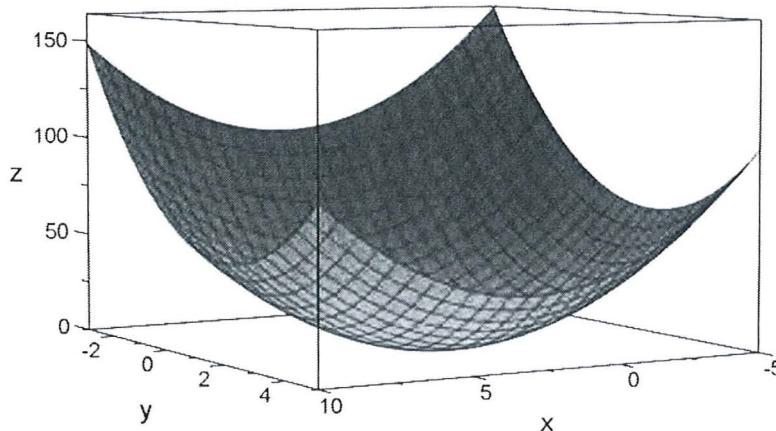
Die Reise geht jetzt schnell weiter zu Problemen in mehreren Variablen, weil diese Klasse besonders viele interessante Anwendungsfälle umfasst.

Viele Ideen werden eingeführt am Beispiel einer Funktion in zwei Variablen $f(x, y)$, können aber sofort verallgemeinert werden. Der Graph einer solchen Funktion kann als Fläche im Raum aufgefasst werden, wobei die Funktionswerte die Höhe der Fläche über der x - y -Grundebene angibt: $z = f(x, y)$.

Mit geometrischer Vorstellung zum Minimum

Je nach Zahl der Variablen und Verwendungszweck schreiben wir eine reellwertige Funktion f auf dem \mathbb{R}^n mal als $f(a)$ oder als $f(x_1, \dots, x_n)$ oder als $f(x, y)$, falls $n = 2$. Gesucht wird ein Minimierer $a^* \in \mathbb{R}^n$.

Ein Beispiel gibt die Funktion $f(x, y) = x^2 - 6x + 4y^2 - 16y + 25$. Der Graph zeigt ein Minimum in der Nähe des Ursprungs. Die genaue Lage des Minimums kann man hier sehr einfach ausrechnen, weil f eine besonders einfache Form hat - aber wir wollen hier nach allgemein anwendbaren Methoden Ausschau halten und solches Spezialwissen nur zur Kontrolle verwenden.



Wir bezeichnen die Minimumsstelle mit $a^* = (x^*, y^*)$ und nehmen an, wir würden längs eines geraden Weges in der x - y -Ebene durch a^* laufen. Längs dieses Weges hätte f in a^* den kleinsten Funktionswert. Der Lauf längs eines Weges ist aber ein eindimensionaler Vorgang, und deshalb bietet sich die Möglichkeit, das Wissen aus dem eindimensionalen Fall anzuwenden. Wenn die Richtung, mit der man durch a^* läuft, als r bezeichnet wird, kann der Weg in der x - y -Ebene als $p(t) = a^* + t \cdot r$ parametrisiert werden. Die Funktionswerte längs des Weges berechnen sich dann nach $\tilde{f}(t) = f(a^* + t \cdot r)$. Dies ist eine eindimensionale Funktion mit einem Minimum bei $t = 0$. Notwendige Minimumsbedingung ist, dass ihre Ableitung verschwindet.

Man nennt diese Ableitung $\tilde{f}'(t) = \frac{d}{dt} f(a^* + t \cdot r)$ die Richtungsableitung von f in Richtung r . Notwendiges Kriterium für ein Extremum von f ist das Verschwinden aller Richtungsableitungen.

Besonders einfach auszurechnen sind die Richtungsableitungen in Richtungen der Koordinatenachsen. Man nennt sie die partiellen Ableitungen und schreibt sie mit rundem Ableitungs-d. Aus der Definition folgt sofort die Grenzwertdarstellung

$$\frac{\partial f(a)}{\partial x_i} = \lim_{h \rightarrow 0} \frac{f(x_1, \dots, x_i + h, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{h}.$$

Man rechnet sie einfach aus, indem man die anderen Variablen als Konstanten betrachtet. In unserem Beispiel ergibt sich:

$$\frac{\partial f}{\partial x} = 2x - 6, \quad \frac{\partial f}{\partial y} = 8y - 16$$

Indem man diese Terme Null setzt und das Gleichungssystem nach x und y auflöst, bestimmt man die exakte Lage des Minimums (da wir nur ein notwendiges Kriterium verwendet haben, muss man an dieser Stelle im allgemeinen vorsichtig sein!).

Satz (Notwendiges Kriterium für Extremstelle): Wenn die differenzierbare Funktion $f(x_1, \dots, x_n)$ in $a^* = (x_1^*, \dots, x_n^*)$ ein lokales Extremum hat, dann gelten die Gleichungen:

$$\frac{\partial f(x_1, \dots, x_n)}{\partial x_i}(a^*) = 0; \quad i = 1 \dots n.$$

Die partiellen Ableitungen einer Funktion ergeben zusammen den Gradienten

$\text{grad } f = \nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$ von f. Das notwendige Kriterium schreibt sich damit kompakt als $\nabla f(a^*) = 0$.

Der Gradient hat eine schöne anschaulich-geometrische Bedeutung. Ausgewertet an einer Stelle $a = (x_1, \dots, x_n)$ ergibt er den Vektor, der in die Richtung des steilsten Anstiegs von f an dieser Stelle zeigt. Außerdem tritt er auf in der Verallgemeinerung der Taylorformel für Funktionen mehrerer Veränderlicher:

$$f(a + h) = f(a) + \nabla f(a) \cdot h + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} \cdot h_i h_j + \dots$$

Dabei sind a und h Vektoren, und die Produkte sind entsprechend Skalarprodukte.

Die Hesse-Matrix $H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$ kann wie die zweite Ableitung im eindimensionalen

Fall zur Formulierung eines hinreichenden Kriteriums und insbesondere zur Unterscheidung von Maxima und Minima herangezogen werden. Hinreichend für ein Minimum ist, dass diese Matrix positiv definit ist - das zu zeigen, kann allerdings recht mühsam sein. Wir wollen deshalb diesen theoretischen Pfad hier verlassen und statt dessen zu Anwendungen kommen.

Anwendung: Regressionsgerade

In den experimentellen Wissenschaften misst man oft zwei Größen, von denen man vermutet, dass sie über eine affin-lineare Beziehung verbunden sind. Im Diagramm sollte der Zusammenhang dann eine Gerade sein. Beispiel ist die Länge einer Feder in Abhängigkeit von der ausgeübten Kraft.

X: Kraft in N	0	2	4.2	6	7,5
Y: Länge in cm	12	16	26,5	39	47

Damit ein Computer eine Ausgleichsgeraden berechnen kann, die diesen Punkten optimal angepasst ist, muss man die Qualität der Ausgleichsgeraden berechenbar machen.

Die Datenpunkte seien $(x_i, y_i); i=1, \dots, n$. Gauss hatte die Idee, die Annäherungs-Qualität einer Geraden zu messen, indem man die Quadrate der Abweichungen aufsummiert. Die Geradengleichung sei $y = f(x) = mx + b$. Dann ist das Gaußsche Qua-

litätsmaß $Q(f) = \sum_{i=1}^n (y_i - f(x_i))^2$. Diejenige Gerade, die dieses Q minimiert, ist die „Least Square“-Approximation oder Regressionsgerade.

Um die Parameter m und b , die $Q(f)$ minimieren, zu bestimmen, bedient man sich des obigen Satzes. Notwendige Bedingung für das Vorliegen eines Extremums ist, dass die Ableitungen nach m bzw. b Null sind:

$$\frac{\partial}{\partial m} Q(f) = \frac{\partial}{\partial m} \sum_{i=1}^n (y_i - mx_i - b)^2 = \sum_{i=1}^n 2(y_i - mx_i - b) \cdot (-x_i)$$

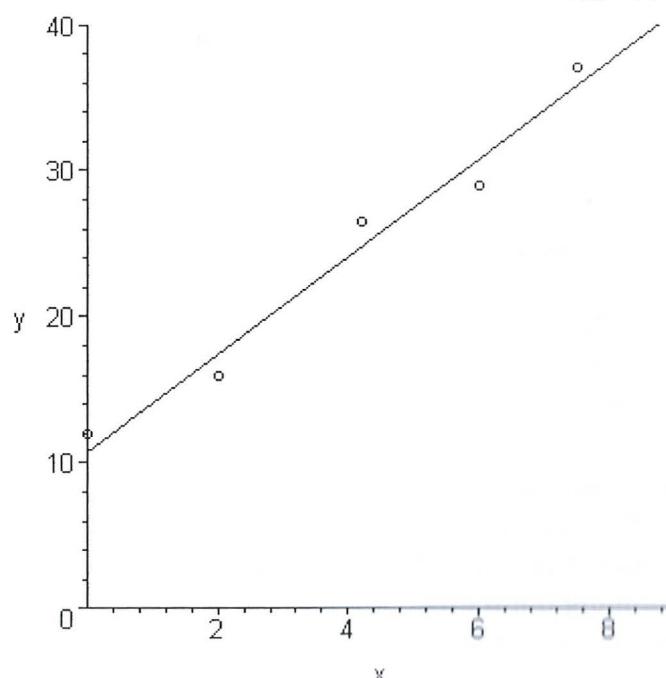
$$= m \sum_{i=1}^n 2x_i^2 + b \sum_{i=1}^n 2x_i - \sum_{i=1}^n 2y_i x_i = 0$$

$$\frac{\partial}{\partial b} Q(f) = \frac{\partial}{\partial b} \sum_{i=1}^n (y_i - mx_i - b)^2 = -2 \sum_{i=1}^n (y_i - mx_i - b) = m \sum_{i=1}^n 2x_i + 2nb - \sum_{i=1}^n 2y_i = 0$$

Mit den Abkürzungen $A = \sum_{i=1}^n x_i$, $B = \sum_{i=1}^n y_i$, $C = \sum_{i=1}^n x_i^2$, $D = \sum_{i=1}^n x_i y_i$ und nach

Division durch 2 kann dieses Gleichungssystem geschrieben werden als

$$mC + Ab - D = 0, \quad mA + nb - B = 0. \quad \text{Es hat die Lösung } m = \frac{nD - AB}{nC - A^2}, \quad b = \frac{CB - DA}{nC - A^2}.$$

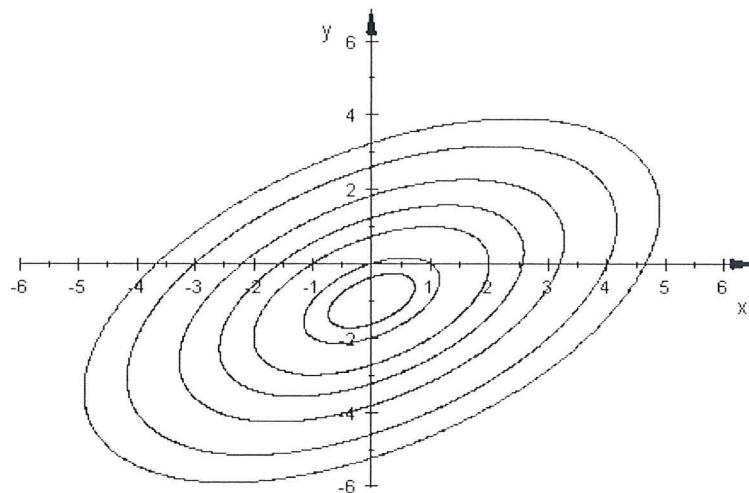


Dieser Weg zur Regressionsgeraden illustriert die Anwendung des obigen Satzes in einem anspruchsvollen Problem. Wenn die Schüler Regressionsgeraden noch nicht kennen, sollte weniger direkt vorgegangen werden.

Numerische Methoden

Für numerische Verfahren sprechen wieder die gleichen Gründe wie im eindimensionalen Fall. Glücklicherweise kann man genau wie in der analytischen Theorie die Hauptarbeit auf den eindimensionalen Fall reduzieren.

Zum Verständnis der Algorithmen ist es sinnvoll, von der Funktionsgraphdarstellung einer Funktion $f(x, y)$ wegzukommen und statt dessen ein Höhenliniendiagramm zu zeichnen. Das folgende Diagramm zeigt von außen nach innen die Höhenlinien $f(x, y) = z$ für $z = 15, z = 10, z = 5, z = 2, z = 0, z = -2, z = -2,6$ der Funktion $f(x, y) = (x - 2)^2 + y^2 - (x - 2) \cdot (y - 3)$. Man erkennt klar ein Minimum bei $(0, -1)$.



Ein Iterationsverfahren startet in einem Punkt der Ebene und berechnen sukzessive weitere Punkte, die dem Minimum immer näher kommen sollen. Es bietet sich folgende Grobstruktur für den Algorithmus an:

Eingabe:

f: Die Funktion

x0: der Startpunkt (Vektor)

eps: Kleine Zahl – Fehlertoleranz

Ausgabe: Eine Näherung für einen Minimierer

Lokale Variablen: r (Vektor), t,topt

loop

Wähle eine Suchrichtung r

Eindimensionale Minimierung von $f(x_0 + t \cdot r)$ in der Variablen t

$x_0 := x_0 + topt \cdot r$, wenn topt der eben gefundene Wert für t ist

if $|topt \cdot r| < \text{eps}$ return x_0

Als Suchrichtung kann man wählen:

abwechselnd die Koordinatenachsen (dann bewegt man sich im Zickzack zum Minimum)

den negativen Gradienten

eine Variation des Gradienten

Das wichtigste Beispiel der letzten Klasse ist das **Newtonverfahren**, das $r = -H^{-1}(x_0) \cdot \nabla f(x_0)$ als Suchrichtung verwendet.

Die Newton-Richtung ist oft so gut, dass man den Schritt der eindimensionalen Optimierung fallen lassen kann. Man kommt zu dem vereinfachten Iterationsschritt mit

$$x_{k+1} = x_k - H^{-1}(x_k) \cdot \nabla f(x_k).$$

Die kompakte Formulierung sollte nicht darüber hinwegtäuschen, dass hier einiges zu rechnen ist: Die Hessematrix muss ermittelt und invertiert werden. Es ist u.U. besser, den neuen Punkt x_{k+1} als Lösung des linearen Gleichungssystems

$H(x_k) \cdot x_{k+1} = H(x_k) \cdot x_k - \nabla f(x_k)$ zu ermitteln, als die Inverse explizit zu berechnen. Es ist es wert, kurz die Aussagen der obigen Theorie an dieser Iterationsformel zu reflektieren: Wenn x_k schon ein Extremum ist, verschwindet der Gradient und es ist $x_{k+1} = x_k$ ein Fixpunkt. An der Stelle eines Minimums ist H positiv definit, und damit (entsprechende Ableitungsstetigkeit vorausgesetzt) auch in einer Umgebung des Minimums. Wenn der Startpunkt der Iteration so gut ist, dass er schon in einer solchen Umgebung liegt, ist gewährleistet, dass die Hessematrix invertierbar ist.

Im Folgenden gehen wir davon aus, dass ein Optimierungsverfahren implementiert ist - welches, ist dabei (fast) unerheblich.

Anwendung: Die Brachistochrone

Wie muss eine Rutschbahn geformt sein, damit ein reibungsfreier Körper darauf schnellstmöglich von einem gegebenen zu einem anderen vorgegebenen Punkt rutscht? Diese Fragestellung trägt den schönen Namen Brachistochronen-Problem. Sie wurde von Johann Bernoulli schon im 17. Jahrhundert beantwortet und hat wesentlich zur Entwicklung der Variationsrechnung beigetragen (Pesch 2002). Leider ist diese Theorie für die Schule zu sperrig. Doch mit numerischer Optimierung kann man entscheidende Erkenntnisse verhältnismäßig einfach erzielen.

Es empfiehlt sich, mit Lichtschranken und flexiblen Eisenbahngleisen (gibt es im Modellbauhandel) oder einer Dardabahn die Situation praktisch zu erproben. In einem solchen Experiment können zunächst Hypothesen aufgebaut und geprüft werden. Das baut Erwartungen auf, was bei der Rechnung vernünftigerweise herauskommen sollte.

Die Rutschbahn wird modelliert durch einen Funktionsgraphen, der z.B. durch $(0, 0)$ als Startpunkt und $(x_e, y_e) = (3, -1)$ als Endpunkt verlaufen soll. Um die Zahl der Variablen endlich zu machen, diskretisiert man das Problem, d.h. man zerteilt das Intervall $[0; x_e]$ in n Teile durch die Stützstellen $x_i = \frac{i}{n} \cdot x_e$, $i = 0, \dots, n$.

Gesucht sind dann die dazugehörigen Funktionswerte y_i , die angeben, wie hoch die Rutschbahn an der jeweiligen Stelle ist. Natürlich muss gelten: $y_0 = 0$, $y_n = y_e$. Es bleiben also die Variablen y_1, \dots, y_{n-1} zu bestimmen.

Minimiert werden soll die gesamte Rutschzeit, die sich als Summe der Rutschdauern in den einzelnen Intervallen ergibt: $t = \sum_{i=0}^{n-1} t_i$. Die Rutschzeit t_i im Intervall

$[x_i, x_{i+1}]$ berechnet man aus der Geschwindigkeit v_i und der Länge der Strecke:

$$t_i = \frac{\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}}{v_i}.$$

Die Geschwindigkeit ergibt sich aus einer physikalischen Energieüberlegung: Die kinetische Energie stammt allein aus der potentiellen

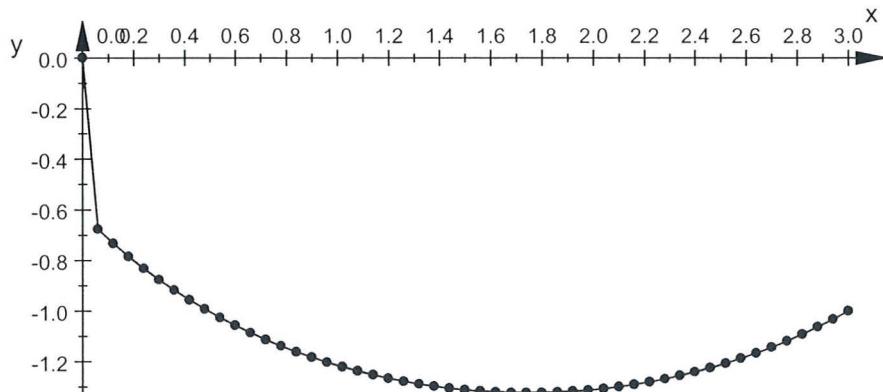
Energie der Höhe y , also gilt $\frac{1}{2} \cdot m \cdot v_i^2 = m \cdot g \cdot y$. Dabei bietet es sich an, als Höhe die mittlere Höhe im Intervall zu nehmen: $y = \frac{y_i + y_{i+1}}{2}$. Das Ganze lässt sich natürlich schnell nach der Geschwindigkeit auflösen. Alles zusammen eingesetzt ergibt einen Term für die Rutschdauer, der nur noch von den y_1, \dots, y_{n-1} abhängt, sich also prima optimieren lassen sollte. Die folgenden Zeilen Pseudocode (die sich aber fast so in MuPAD umsetzen lassen) zeigen, wie man das Problem in einem Mathematikprogramm lösen kann. Dabei wird eine Optimierungsfunktion NEWTON verwendet, die die Zielfunktion, eine Liste der Variablen und eine Liste der Startwerte übergeben bekommt.

```

n:=50                                // Zahl der Stützstellen
ya:=0.0                                 // Start in (0,ya)
xe:=3; ye:=-1                           // Ende in (xe,ye)
v:= y -> sqrt(-2*9.81*y)               // Geschwindigkeit v in Höhe y
ttime:=sum(1/v(0.5*(y[i]+y[i+1]))*sqrt((y[i]-y[i+1])^2+(xe/n)^2), i=0..n-1)
// ttime=total time= Summe über die einzelnen Zeiten ti
term:=subs(ttime, y[0]=ya, y[n]=ye) // Randbedingungen einbauen
ys:=NEWTON(term,[y[1],...,y[n]],[1,...,-1])

```

Aus den Werten lassen.



Das Ergebnis ist nicht unbedingt so, wie es Schüler erwarten: Zwar bestätigt sich die Überlegung, dass es besser ist, erst Schwung zu holen und dann mit großer Geschwindigkeit die restliche Strecke zu fahren, aber soll man tatsächlich „untertunneln“ und am Ende wieder den Berg hochfahren? Das Experiment bestätigt aber die numerische Rechnung: Es gelingt tatsächlich, durch Untertunneln die Zeit zu verbessern. Nachträglich lässt es sich dann auch inhaltlich verstehen, dass das Schwungholen sinnvoll ist. Besonders klar wird es - wie so oft - im Grenzfall: Wenn der Zielpunkt auf der gleichen Höhe wie der Startpunkt liegt, würde man auf der direkten Verbindungsgeraden keine Hangabtriebskraft erfahren, also in Ruhe bleiben und bräuchte damit - formal - unendlich lange. Durch das Untertunneln schafft man es dagegen in endlicher Zeit ans Ziel.

Die analytische Theorie der Brachistochrone erfordert Variationsrechnung und die Lösung einer unangenehmen Differentialgleichung. Letztlich ergeben sich als Lösung Zykloiden, also die Bahnen, die ein Punkt auf dem Umfang eines rollenden Rades beschreibt.

Beispiel: Der Kranausleger

Stahlträger verhalten sich nahezu wie elastische Federn - zumindest solange sie nicht übermäßig belastet werden. Das bedeutet, dass sie eine Spannenergie besitzen, die proportional ist zum Quadrat der Abweichung der aktuellen Länge von der

Ruhelänge: $E = \frac{1}{2} \cdot k \cdot (l - l_0)^2$ (dabei ist k die Hooke'sche Federkonstante).

Interessant ist die Frage, wie sich der Ausleger verformt, wenn man ein Gewicht anhängt. Als Minimierungsproblem ist das leicht formuliert: Gesucht sind die Koordinaten x_i, y_i der Federendpunkte, die zum Minimum der Gesamtenergie (bestehend aus der Summe der Spannenergien aller Federn und der potentiellen Energie der angehängten Last) führen.

Hier der Pseudocode eines CAS-Skripts, das das Verfahren umsetzt:

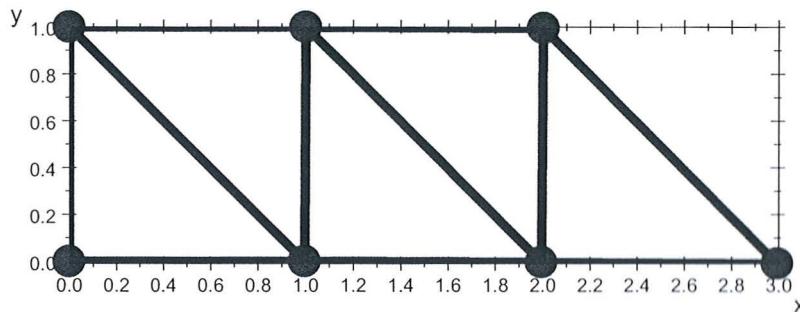
```
// Die Federendpunkte sind (x[i],y[i])
// Die Funktion feder berechnet die Energie einer Feder mit Feder-Konstante K und Ruhelänge L
feder:=(i,j,K,L) -> 1/2*K*(L-sqrt((x[i]-x[j])^2+(y[i]-y[j])^2))^2
vars:=[x[3],y[3],x[4],y[4],x[5],y[5],x[6],y[6],x[7],y[7]]:
gesEnergieTerm:=
    feder(2,3,100,1)+feder(4,3,100,1)+feder(4,5,100,1)+  

    feder(1,7,100,1)+feder(7,6,100,1) // horizontal  

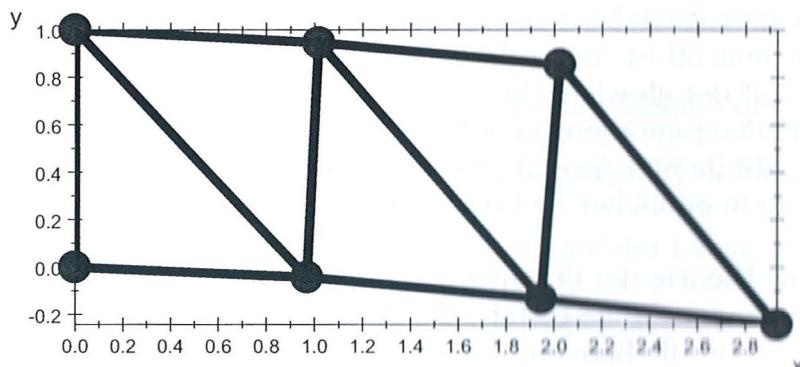
    feder(3,7,100,1)+feder(4,6,100,1) // vertikal  

    feder(1,3,100,1.41)+feder(7,4,100,1.41)+feder(6,5,100,1.41)+  

    9.81*m*y[5]: // m Kg Gewicht am Ausleger
// Jetzt werden die fixen Koordinaten eingesetzt
gesEnergie:=nr
Belastet:=NEW
```



Belastet2:=NEWTON(gesEnergie (0.1),vars): zeichne(Belastet2)



Was passiert, wenn die angreifende Kraft noch größer wird? Gibt es Grenzen der Modellbildung? Oder gar Grenzen der Realität?

Anwendung: Lösung überbestimmter Gleichungssysteme

Wenn Koeffizienten von Gleichungssystemen aus fehlerbehafteten Messungen stammen, resultieren oft überbestimmte Gleichungssysteme, die gar keine Lösung mehr besitzen. In der Praxis ist es dann nötig, diese Systeme zumindest näherungsweise zu lösen. Das gelingt, indem man das Lösen des Gleichungssystems als Optimierungsproblem auffasst.

Aus einem System von m Gleichungen $g_i(x) = 0, \quad i = 1, \dots, m, \quad x \in \mathbb{R}^n$ bildet man die

Zielfunktion $f(x) = \sum_{i=1}^m g_i(x)^2$. Jede Lösung des Systems ist ein Minimierer von f mit

Minimum 0 und umgekehrt.

Weil die numerische Optimierung normalerweise nur ein lokales Minimum liefert, auch wenn es mehrere gibt, findet man hier natürlich auch (höchstens) eine Lösung eines Gleichungssystems.

Einige Beispiele, in denen dieser Ansatz weiter hilft:

- Bestimmung der 3D-Koordinaten eines Punktes aus mehreren Fotografien
- Berechnung der Absorptionswerte bei der Computertomographie
- Positionsbestimmung bei GPS

Aufgabe: Aus Echo-Messungen ist bekannt, dass sich der Punkt P(x, y) 5 Einheiten von (8, 8), 3 Einheiten von (5, 1) und 7 Einheiten von (0, 0) entfernt befindet. Wo sollte man ihn vermuten?

Eine weitere Anwendung haben wir schon in etwas anderer Betrachtungsweise kennen gelernt: Die Regressionsgerade durch n gegebene Datenpunkte (x_i, y_i) . Ihre Parameter ergeben sich aus den Lösungen m und b des überbestimmten Gleichungssystems $y_i = m \cdot x_i + b$. Auch die verallgemeinerten Regressionsfragestellungen ordnen sich hier nahtlos ein.

Probleme mit Einschränkungen

Unter Umständen sucht man einen Minimierer nicht im ganzen Definitionsbereich der Zielfunktion, sondern nur unter denjenigen Punkten, die eine bestimmte Bedingung erfüllen.

Beispiel: Gesucht ist der Punkt auf dem Einheitskreis um den Koordinatenursprung, der am nächsten an A(4 | 3) liegt.

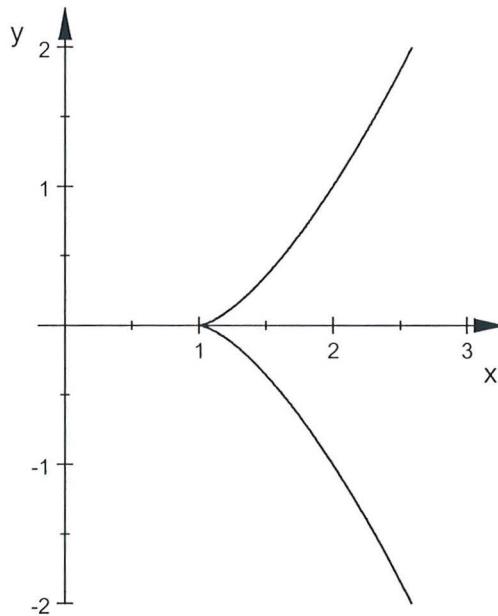
Die Zielfunktion ist hier die Abstandsfunktion $f(x, y) = \sqrt{(x - 4)^2 + (y - 3)^2}$ wobei die Punkte (x, y) aber auf dem Einheitskreis liegen müssen, also die Nebenbedingung $g(x, y) := x^2 + y^2 - 1 = 0$ erfüllen müssen.

Ganz allgemein kann ein **eingeschränktes Optimierungsproblem** folgendermaßen formuliert werden:

Minimiere $f(x)$ unter der Nebenbedingung $g_i(x) = 0$, $i=1,\dots,m$. In bestimmten Situationen ist es auch sinnvoll, Ungleichungen als weitere Einschränkungen hinzuzunehmen.

Das obige Beispiel für ein eingeschränktes Optimierungsproblem haben wir schon zu Beginn dieses Textes gelöst. Im Rückblick liefert das dortige Vorgehen ein Rezept zum Lösen eingeschränkter Optimierungsaufgaben: Man nutze die Nebenbedingungen, um die Zahl der Variablen zu verringern, bis (möglichst) gar keine Nebenbedingungen mehr notwendig sind. Dieses Eliminationsverfahren hat aber seine Nachteile: Manchmal ist es rechnerisch schwer oder gar nicht durchführbar, und selbst wenn es gelingt, muss man auf der Hut sein:

Beispiel: Minimiere $x^2 + y^2$ unter der Bedingung $y^2 = (x - 1)^3$. Aus der geometrischen Interpretation (siehe Bild) ist klar, dass die Lösung derjenige Punkt der Kurve $y^2 = (x - 1)^3$ ist, der dem Ursprung am nächsten kommt, das ist $(1, 0)$. Durch Elimination von y gelangt man aber zum nichteingeschränkten Optimierungsproblem $x^2 + (x - 1)^3$, das zu $x \rightarrow -\infty$ führt.



Die Strafterm-Methode

In der Mathematik gibt es manchmal Situationen, in denen nur der Mut der Verzweiflung weiter hilft. Eingeschränkte Optimierungsaufgaben können einen in solche Situationen bringen. Einerseits sind die Fragestellungen so klar und einfach, dass es eine Lösung geben muss, andererseits ist die komplette Theorie äußerst diffizil. Das einleitende Beispiel in diesem Text, das Gewicht, das an einem masselosen Faden hängt, ist ein solches Problem. Mit etwas Nachdenken oder mit einfachen Experimenten wird klar, dass es eine eindeutige Lösung geben muss, ja man kann sogar im Experiment „sehen“, dass die Lösung stetig von den Parametern (Aufhängepunkte, Fadenlänge) abhängt. Dennoch ist eine Lösung recht schwierig (versuchen Sie es!) zu finden, wenn man nicht die eingangs geschilderte geometrische Idee hat.

Wenn man ein solches oder ähnliches reales Problem (siehe unten) vor sich hat, ist man für eine Lösung vielleicht auch bereit, eine wirklich haarsträubende Nähe-

rungsmethode zu akzeptieren. Eine solche ist die Strafterm-Methode (penalty-Methode). Als Rezept lautet sie:

Wenn eine Zielfunktion f gegeben ist, die unter den Nebenbedingungen

$g_i(x) = 0, \quad i = 1, \dots, m, \quad x \in \mathbb{R}^n$ zu optimieren ist, definiert man eine neue Ersatz-

Zielfunktion $\tilde{f}(x) = f(x) + \mu \cdot \sum_{i=1}^m g_i(x)^2$ und minimiert diese ohne Beachtung von Einschränkungen.

(Man sieht also hier die bekannte heuristische Strategie der Reduktion auf einen bekannten Fall am Werke!). Die Zahl $\mu > 0$ wird dabei nach Gefühl gewählt.

Warum kann so etwas überhaupt funktionieren? Für alle x , die die Bedingungsgleichungen erfüllen, ist der letzte Summand 0. Umgekehrt, für unzulässige x ist dieser Term größer als 0, und wenn μ eine sehr große Zahl ist, macht das den Funktionswert von $\tilde{f}(x)$ so groß, dass x nicht mehr als Minimierer in Frage kommt. Demnach müsste man μ so groß wählen wie nur möglich, um jede Verletzung der Bedingung maximal abzustrafen. Allerdings bringt man damit u.U. den Optimierungsalgorithmus in Probleme, wenn eine Summe sehr unterschiedlich großer Terme minimiert werden soll. In der Praxis reichen oft verhältnismäßig kleine Werte für μ aus. In dieser Hinsicht ist das obige Beispiel etwas untypisch. Zur Illustration wurde das obige Problem „Minimiere $x^2 + y^2$ unter der Bedingung $y^2 = (x - 1)^3$ “ nach der Penalty-Methode mit einer einfachen Newton-Minimierung gelöst. Die exakte Lösung ist, wie oben gezeigt, $x = 1, y = 0$. Die folgende Tabelle zeigt den gefundenen Wert für x in Abhängigkeit von $\mu = 10^k$:

k	x
0	0,349480479
1	0,5505022086
2	0,7022358881
4	0,8760957887
6	0,9498687574
8	0,9799176899
10	1,342663076
12	1,568134189

Diese Ergebnisse sollten zur Vorsicht mahnen, aber nicht generell Unsicherheit schüren: Es reicht schon, die Abbruchbedingung in der Newton-Iteration zu verkleinern, um die Werte besser aussehen zu lassen. An einem solchen Beispiel können die Schüler gleich noch eine wichtige Heuristik lernen: Numerischen Ergebnissen ist dann zu misstrauen, wenn das Ergebnis wesentlich von der Abbruchbedingung der Iteration abhängt, denn dann ist man wohl (noch) nicht auf einem guten, konvergenten Weg.

Beispiel: Die hängende Kette

Dies ist ein weiteres klassisches Problem und experimentell ebenfalls leicht zugänglich (möglichst feingliedrige, nicht zu leichte Kette verwenden). Nachdem bereits bei der Brachistochrone diskretisiert wurde, ist es nicht überraschend, dass das hier nach dem gleichen Schema läuft. Man stellt sich die Kette vor als Polygonzug mit Stützstellen $(x_i, y_i), \quad i = 0, \dots, n$. Minimiert werden soll die potentielle Energie

der Kette, die proportional ist zur Summe der Länge der Segmentstücke multipliziert mit ihrer mittleren Höhe: $E = \sum_{i=0}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \cdot \frac{y_i + y_{i+1}}{2}$

Als Einschränkung ergibt sich, dass die Summe der Längen aller Segmente gleich der Länge L der Kette ist (Für L gibt man irgendeine Zahl vor):

$$L = \sum_{i=0}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

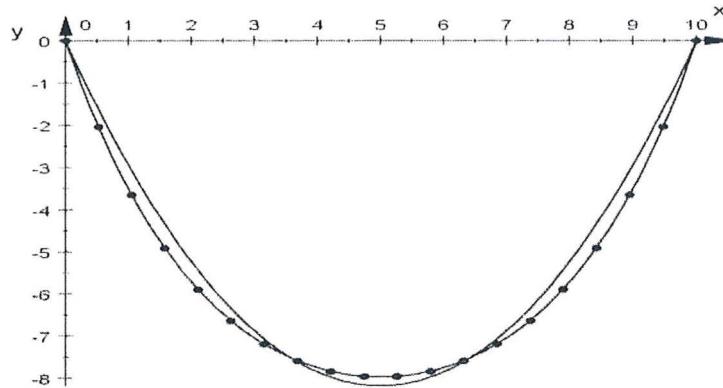
Nach der Penalty-Methode muss man also das Minimum suchen von

$$\tilde{f} = \sum_{i=0}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \cdot \frac{y_i + y_{i+1}}{2} + \mu \cdot \left(L - \sum_{i=0}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \right)^2$$

Mit $\mu = 100$ oder $\mu = 1000$ erzielt man wunderbare Ergebnisse. Die Genauigkeit der numerischen Rechnung ist in jedem Falle gut genug, die naheliegende Vermutung, es könne sich um eine Parabel handeln, zu widerlegen. Die folgende Grafik zeigt einen Parabelbogen der Bogenlänge 20 durch die Punkte $(0 | 0)$ und $(10 | 0)$ und den durch numerische Optimierung mit 20 Stützstellen gewonnenen Verlauf (Punkte). Offensichtlich hängt die Parabel in der Mitte tiefer - aber um den Preis, an den Seiten deutlich zu hoch zu sein. Eingezeichnet ist auch die Funktion mit

$$y = \frac{\cosh\left(c \cdot \left(x - \frac{d}{2}\right)\right) - \cosh\left(c \cdot \frac{d}{2}\right)}{c}, \text{ die sich aus der analytischen Theorie ergibt.}$$

Dies ist der Term der Kettenlinie durch die Punkte $(0, 0)$ und $(d, 0)$, wobei c die Lösung von $\frac{L \cdot c}{2} = \sinh\left(\frac{d \cdot c}{2}\right)$ ist.



Minimalflächen

Seifenfilme versuchen sich zusammenzuziehen, also unter den gegebenen Randbedingungen die kleinste Fläche einzunehmen. Bei frei schwebenden Seifenblasen ergibt sich die Randbedingung aus der eingeschlossenen Luftmenge: Das umspannte Volumen ist eine feste Größe (Seifenhäute können durch den kleinen Binnendruck die Luft nicht nennenswert komprimieren). Die umspannende Fläche minimalen Inhalts ist natürlich die Kugeloberfläche.

Wenn in einem Gebilde aus Seifenfilm drei flüssige Kanten zusammenstoßen, tun sie das stets unter einem Winkel von 120° . Diese Regel hängt unmittelbar mit dem Fermatpunkt zusammen, wie man sich an der Skizze der Situation in einer zur Kante orthogonalen Schnittebene leicht überlegen kann.

Interessant ist auch, eine Seifenhaut in einen Drahtrahmen einzuspannen. Wenn man aus Draht eine ebene Form biegt (z.B. einen Kreis), wird das Innere durch eine ebene Haut erfüllt. Interessant wird es, wenn der Rand so verbogen wird, dass eine ebene Fläche nicht mehr hineinpasst. Dann bekommt man schöne Minimalflächen in der typischen Sattelform. Wir wollen jetzt solche berechnen.

Dazu nehmen wir vereinfachend an, dass die Fläche als Funktionsgraph $z = f(x, y)$ einer Funktion in zwei Variablen dargestellt werden kann. (Dies ist nicht zwingend, reduziert aber den Rechenaufwand etwas).

Außerdem nehmen wir an, dass die Minimalfläche in ein räumliches Polygon eingespannt ist, das durch die Punkte $(0, 0, 0), (1, 0, 1), (1, 1, 0), (0, 1, 1), (0, 0, 0)$ gegeben ist (dies ist ein räumliches Viereck, denn die Projektion in die x-y-Ebene ist ein Quadrat, und die z-Komponente (Höhe) steigt und fällt abwechselnd). In diskreter Form wird die Fläche (als Funktionsgraph) durch die Höhe $z_{ij} = f(a_{ij})$ der Fläche an den $(n+1)^2$ Stützstellen $a_{ij} = \left(\frac{i}{n}, \frac{j}{n}\right)$ (mit $i = 0, \dots, n; j = 0, \dots, n$) beschrieben. Das Randpolygon legt dabei die äußersten $4n$ Werte $z_{0j}, z_{nj}, z_{i0}, z_{in}$ fest.

Die restlichen Variablen werden als Optimierungsvariablen bestimmt.

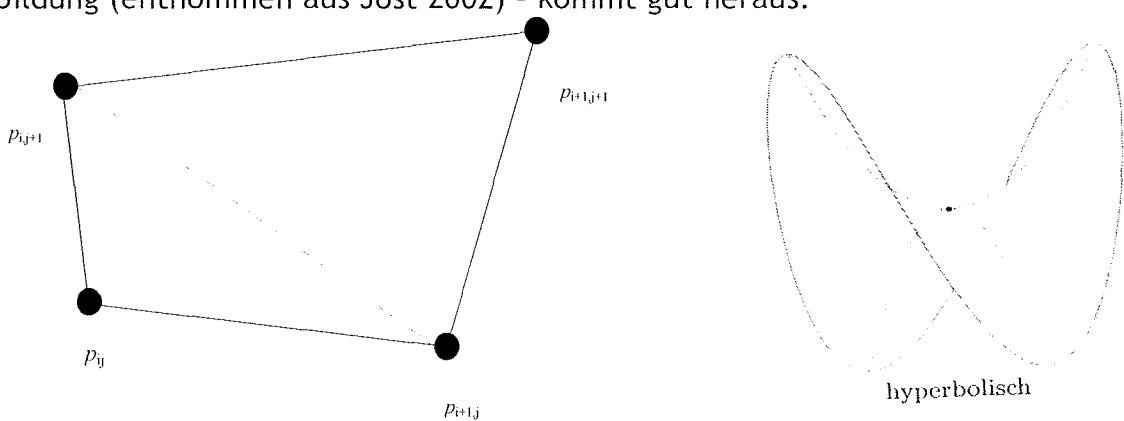
Die Zielfunktion ist der Flächeninhalt. Um ihn zu berechnen, schauen wir ein Viereck an, das durch vier benachbarte Stützstellen gegeben wird. Die Ortsvektoren

der Eckpunkte sind $\bar{p}_{ij} = \left(\frac{i}{n}, \frac{j}{n}, z_{ij}\right)$. Die Vierecke werden weiter in zwei Dreiecke

zerlegt, für die die Flächenberechnung mit dem Kreuzprodukt möglich ist. Der Flächeninhalt eines Vierecks ist dann gegeben durch

$$A_{ij} = \frac{1}{2} \left| (\bar{p}_{i+1,j} - \bar{p}_{i,j}) \times (\bar{p}_{i,j+1} - \bar{p}_{i,j}) \right| + \frac{1}{2} \left| (\bar{p}_{i+1,j} - \bar{p}_{i+1,j+1}) \times (\bar{p}_{i,j+1} - \bar{p}_{i+1,j+1}) \right|.$$

Die Zielfunktion ist die Summe über all solche Flicken: $A = \sum_{i,j} A_{ij}$. Eine einschränkende Nebenbedingung fehlt, so dass man numerisch schnell loslegen kann. Die Ergebnisse sind natürlich nichtssagende Datenwüsten, die man deswegen z.B. mit einem Computeralgebrasystem graphisch darstellen sollte. Die Ergebnisse lassen sich sehr schön mit dem Experiment vergleichen. Die sattelförmige Gestalt der Minimalflächen - es handelt sich um hyperbolische Flächen, siehe die unten stehende Abbildung (entnommen aus Jost 2002) - kommt gut heraus.



Fazit

In diesem Aufsatz wurde ein ganzer Katalog von realen Situationen beschrieben, die sich als ein Optimierungsproblem beschreiben lassen. Verhältnismäßig oft scheitert eine analytische Lösung, so dass die hier entwickelten numerischen Methoden zum Einsatz kommen müssen. Schüler können die prinzipielle Arbeitsweise einer numerischen Optimierungsprozedur gut verstehen; die Finessen, die für schnellst mögliche Algorithmen nötig sind, brauchen sie dagegen nicht zu kennen. Ein professioneller Algorithmus kann als black-box benutzt werden. Das ist hier unbedenklich, da die Bedeutung des Algorithmus - also das, was man von seiner Lösung erwarten kann - leicht zugänglich und verständlich ist. Durch den Umgang mit einfachen Iterationsverfahren gewinnen die Schülerinnen und Schüler die nötigen Grundvorstellungen, die sie vor einer unkritischen Nutzung bewahren sollten. Nachdem eine solche Grundvorstellung gewonnen und die Anwendung einer Optimierungsprozedur klar ist, besteht die wesentlich Schüleraktivität darin, die Situation als Optimierungsproblem zu modellieren, also eine Zielfunktion aufzustellen (funktionale Modellierung) und ggf. Bedingungen als Gleichungen zu beschreiben (relationale Modellierung). Die Lösungen können bei realen Problemen oft von den Schülern selbst bewertet werden, optimal z.B. dann, wenn der Vergleich mit einem Experiment möglich ist.

Die verwendeten MuPAD Dateien können über die Internetseite www.ph-heidelberg.de/wp/filler/istrong9 bezogen werden.

Literatur

- W. Blum: Anwendungsorientierter Mathematikunterricht in der didaktischen Diskussion. MS 32 (2) 1985, 195-232.
J. Jost: Differentialgeometrie und Minimalflächen, Berlin 2002.
J. Nocedal, S. J. Wright: Numerical Optimization, New York 1999.
J. Oprea: The Mathematics of Soap Films, 2000.
H. J. Pesch: Schlüsseltechnologie Mathematik, Wiesbaden 2002.
Press et al: Numerical Recipes, Cambridge 1996.

Adresse des Autors:

Prof. Dr. Reinhard Oldenburg, Pädagogische Hochschule Heidelberg, Im Neuenheimer Feld 561

E-Mail: oldenburg@ph-heidelberg.de