

Institut für
Angewandte Mathematik

Institut für Wasserwirtschaft,
Hydrologie und landwirtschaft-
lichen Wasserbau

Universität Hannover

**Parameteridentifikation
für
Simulationsmodelle
in der
Hydrologie**

Diplomarbeit

von

Susanne Deyda

E R K L Ä R U N G

Ich erkläre hiermit an Eides Statt, die
vorliegende Arbeit selbständig und nur mit
den angegebenen Hilfsmitteln angefertigt zu
haben.

Hannover, den 26.4.91

INHALTSVERZEICHNIS

<u>Inhalt</u>	<u>Seite</u>
1. Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	2
2. Einführung in die mathematische Optimierung	3
2.1 Formulierung der mathematischen Aufgabenstellung	3
2.2 Verfahren zur Lösung der Optimierungsaufgabe	5
2.3 Die eindimensionale Minimierungsaufgabe	10
2.4 Numerische Iterationsverfahren zur mehrdimensionalen Optimierungsaufgabe	14
2.4.1 Ableitungsfreie Methoden	14
2.4.2 Gradientenmethoden	17
2.4.3 Newtonmethoden	24
2.5 Fehlernormen und Abbruchkriterien	27
3. Beschreibung des verwendeten Simulationsmodells	30
3.1 Allgemeines	30
3.2 Das Modell HYSRAD	30
3.3 Das Modell KMROUT	38

4. Anwendung des Optimierungsverfahrens auf das Simulationsmodell	42
4.1 Die hydrologische Aufgabenstellung	42
4.2 Verknüpfung der hydrologischen mit der mathematischen Aufgabenstellung	43
4.2.1 Die Parameterwahl	43
4.2.2 Die Meß- und Simulationsdaten	47
4.3 Ähnlichkeitsvergleich von Abflußganglinien	48
4.4 Sensitivität der Parameter	51
4.4.1 Der Parameter A_s	53
4.4.2 Der Parameter V_m	55
4.4.3 Der Parameter A_u	57
4.4.4 Der Parameter α_u	58
4.4.5 Zusammenfassung	60
4.5 Anwendung eines Optimierungsverfahrens auf das Modell HYSRAD/KMROUT	61
5. Anwendung des Optimierungsverfahrens	66
5.1 Autokalibrierung in einem realen Netz	66
5.2 Optimierung in realen Netzen und Meßdaten	77
6. Zusammenfassung und Ausblick	82
LITERATURVERZEICHNIS	84
BILDVERZEICHNIS	89
TABELLENVERZEICHNIS	91
ANHANG	

1. EINLEITUNG

1.1 Motivation

Die Entwicklung mathematischer Modelle zur Simulation von hydrologischen Vorgängen in der Natur nimmt in den letzten Jahren durch die rasanten Fortschritte in der Computertechnik einen immer größer werdenden Stellenwert in der Hydrologie ein.

Durch Beobachtung von physikalischen Teilprozessen aus der Umwelt wird hierfür zunächst ein hydrologisches System gebildet, das zeit- bzw. ortsabhängige Eingaben (Input) in Systemausgaben (Output) transformiert. Dieser meistens durch Messungen bestimmte eindeutige Zusammenhang zwischen quantitativer Ursache und quantitativer Wirkung wird dann in einem Modell mit Hilfe von mathematischen Ausdrücken abstrahiert.

Neben Input- und Outputvariablen enthält solch ein Modell in der Regel zusätzliche Parameter (System- oder Modellparameter), die wiederum zeit- bzw. ortsabhängig sein können und eine Anpassung an die natürlichen Prozesse bei unterschiedlichen Rahmenbedingungen ermöglichen.

Eine weitere Aufgabe besteht dann darin, diese Freiheitsgrade so zu bestimmen, daß das Modell bei vorgegebenen Eingabedaten zu den hieraus resultierenden Meßdaten eine möglichst gute Näherung als Ausgabe liefert.

Dieser Vorgang wird mit Modelleichung (Kalibrierung, Parameteridentifikation) bezeichnet. Er fordert von dem Modellanwender viel Geschick und Erfahrung und ist sehr zeitaufwendig.

Eine Automatisierung der Modellkalibrierung, die unabhängig von dem Benutzer gleiche Ergebnisse liefert, wäre daher von großem Nutzen.

1.2 Zielsetzung

In dieser Arbeit wird versucht mit Hilfe mathematischer Optimierungsverfahren eine Automatisierung des Kalibrierungsvorganges eines hydrologischen Modells durchzuführen. Die Untersuchungen sollen auf ein ausgewähltes hydrologisches Simulationsmodell angewendet werden.

Die Arbeit unterteilt sich so in einen mathematischen und einen hydrologischen Teil.

In dem ersten Teil wird zunächst eine konkrete mathematische Aufgabe gestellt, die später auf die hydrologische Aufgabenstellung übertragen werden kann. Im Anschluß daran werden verschiedene mathematische Optimierungsverfahren vorgestellt und diskutiert.

In dem hydrologischen Teil der Arbeit wird das ausgewählte Simulationsmodell mit seinen Modellansätzen und Systemparametern vorgestellt. Eine genauere Untersuchung der Modellparameter soll anschließend Aufschluß über die Wirkungsweise einer Parametervariation geben.

Aus den vorgestellten Optimierungsverfahren wird eine Methode ausgewählt und auf das hydrologische Simulationsmodell angewendet.

Es wird ein Computerprogramm geschrieben, das eine Parameteridentifikation für das Modell mit dem Optimierungsverfahren durchführt.

Abschließend wird in konkreten Rechenläufen die Anwendbarkeit des Optimierungsverfahrens getestet und dabei versucht, das Verhalten in dem Optimierungsprozeß mit den in den Voruntersuchungen gewonnenen Erkenntnissen zu erklären.

2. EINFÜHRUNG IN DIE MATHEMATISCHE OPTIMIERUNG

2.1 Formulierung der mathematischen Aufgabenstellung

Gegeben sind m Datenpunkte $(t_1, y_1), \dots, (t_m, y_m) \in \mathbb{R}^2$ und eine Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$, die von ihrer Variablen $t \in \mathbb{R}$ abhängt, sowie von n aus Teilmengen der reellen Zahlen wählbaren Parametern $x_1, \dots, x_n$ (geschrieben: $f = f(t; x) = f(t; x_1, \dots, x_n)$, $x = (x_1, \dots, x_n)^T \in X \subseteq \mathbb{R}^n$).

Die Funktion f wird hier Modell genannt und soll die Daten $y_1, \dots, y_m$ an den Stellen $t_1, \dots, t_m$ durch Wahl eines geeigneten Parametersatzes (optimaler Parametersatz) bestmöglichst nachbilden. Dafür werden die Daten y_i den mit dem Parametersatz x bestimmten Modellwerten $f(t_i; x)$ zugeordnet ($i=1, \dots, m$). Der von den Parametern abhängige Fehler der Modellanpassung ist dann ein Vektor in dem m -dimensionalen reellen Vektorraum $e(x) = (e_1(x), \dots, e_m(x))^T \in \mathbb{R}^m$ mit den Komponenten $e_i(x) = y_i - f(t_i; x)$ ($i = 1, \dots, m$).

Um die Güte solch einer Anpassung messen zu können, wird der Fehler $e(x)$ unter einer Norm $\|\dots\|$ des $\mathbb{R}^m$ betrachtet (wie z.B. $\|\dots\|_1, \|\dots\|_2, \|\dots\|_\infty$). Damit wird die sogenannte Zielfunktion (Fehlerfunktion) $F: X \rightarrow \mathbb{R}_{\geq 0}$, $F(x) := \|e(x)\|$ definiert, die einem Parametersatz $x \in X$ eine positive reelle Zahl als Maß der Anpassungsgüte zuordnet.

Ein Parametersatz $x^{(1)}$ liefert dann eine "bessere" Anpassung des Modells f an den Datensatz $y = (y_1, \dots, y_m)^T$ als $x^{(2)}$, falls $F(x^{(1)}) \leq F(x^{(2)})$ gilt, und demzufolge muß für einen optimalen Parametersatz $x^* \in X$ $F(x^*) \leq F(x)$ für alle möglichen Parametersätze x ($\forall x \in X$) erfüllt werden.

Dies ist ein Minimierungsproblem (hier Optimierungsproblem genannt) einer Funktion aus dem $\mathbb{R}^n$ in $\mathbb{R}$, und so ergibt sich eine Formulierung der Aufgabenstellung zu:

$$\left[x^* \in X : F(x^*) = \min_{x \in X} F(x) \right] \quad (2.1)$$

Der Definitionsbereich X des Parametersatzes x heißt in (2.1) zulässiger Bereich und ein Punkt $x \in X$ zulässiger Punkt (zulässige Lösung). Für den Fall $X = \mathbb{R}^n$ liegt ein Optimierungsproblem ohne Restriktionen (Nebenbedingungen) vor, ist X echte Teilmenge von $\mathbb{R}^n$, so heißt (2.1) Optimierungsproblem mit Restriktionen.

Der zulässige Bereich wird in restringierten Optimierungsaufgaben üblicherweise in Form von Gleichungs- bzw. Ungleichungssystemen $(g_i(x) \leq 0, g_i: \mathbb{R}^n \rightarrow \mathbb{R}, i=1, \dots, l)$ beschrieben, wobei die g_i in den hier betrachteten Fällen affine Abbildungen der Gestalt

$$g_i(x) = \sum_{k=1}^n (a_k x_k + b_k) \quad (a_k, b_k \in \mathbb{R}, k=1, \dots, n)$$

sind. Wird z.B. gefordert, daß alle Parameter x_i ($i=1, \dots, n$) nicht negativ sind, so läßt sich dieses durch die Nebenbedingungen $g_i(x) = -x_i \leq 0$ ($i=1, \dots, l=n$) erreichen. Der zulässige Bereich ergibt sich dann zu

$$X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i=1, \dots, l\}.$$

Die Dimension l kann hierbei kleiner, gleich oder größer als die Dimension n sein. Liegen $l \leq 1$ Gleichungsrestriktionen vor, so bildet $n - l$ den Freiheitsgrad der Optimierungsaufgabe (l Parameter werden nach den Restriktionen bestimmt, $n - l$ zur Minimierung der Funktion F). In dieser Arbeit werden jedoch nur Ungleichungsrestriktionen auftreten, so daß auf eine Betrachtung der speziellen Behandlung von Optimierungsaufgaben mit Gleichungsrestriktionen hier verzichtet werden kann. Verfahren zur Lösung dieser Aufgabe werden u.a. in [25, Kap.7], [16, Kap.7] erläutert.

Die Dimensionen n und m können unter mathematischen Gesichtspunkten in beliebiger Relation zueinander stehen. In der Praxis ist man jedoch daran interessiert, mit möglichst wenigen Parametern möglichst viele Daten zu reproduzieren, und aus diesem Grund sei im folgenden die Beziehung $m \gg n$ (m wesentlich größer als n) vorausgesetzt.

2.2 Verfahren zur Lösung der Optimierungsaufgabe

Aus der klassischen Analysis steht zur Behandlung von Extremwertproblemen über dem $\mathbb{R}^n$ der folgende, u.a. in [8, Satz 173.13] bewiesene Satz zur Verfügung:

Satz 2.1 (Extremwertkriterium) Seien $G \subset \mathbb{R}^n$ offen, $x^* \in G$. Ist $F: G \rightarrow \mathbb{R}$ eine mindestens einmal stetig differenzierbare Funktion, und gilt für ihren Gradienten ∇F an der Stelle x^* $\nabla F(x^*) = 0$ ($\in \mathbb{R}^n$), dann folgt:

- a) Ist die Hesse-Matrix $\nabla^2 F$ an der Stelle x^* positiv definit (negativ definit), d.h.

$$\sum_{i,j=1}^n \frac{\partial^2 F}{\partial x_i \partial x_j} (x^*) x_i x_j > 0 \quad (< 0) \quad \text{für alle } x = (x_1, \dots, x_n)^T \in G \setminus \{0\},$$

so besitzt F in x^* ein lokales Minimum (lokales Maximum).

- b) Ist die Hesse-Matrix indefinit, so hat F keinen Extremwert in x^* .
c) Für den Fall einer semidefinit Hesse-Matrix kann unter diesen Voraussetzungen keine Aussage über das Vorhandensein eines Extremwertes an der Stelle x^* gemacht werden.

Mit diesem Kriterium läßt sich ein methodisches Vorgehen zur Lösung der Minimierungsaufgabe (2.1) beschreiben:

- (M1): Bestimmung aller kritischen Stellen (stationäre Punkte) im Innern des Definitionsbereiches X der Funktion F , d.h. aller Punkte $x \in \overset{\circ}{X} := \{x \in X: x \text{ ist innerer Punkt von } X\}$ mit $\nabla F(x) = 0$.
(M2): Untersuchen, an welchen Stellen x aus (M1) lokale Minimalwerte vorliegen, d.h. für welche x die Hesse-Matrix positiv definit ist.
(M3): Den Rand von X (falls er zu X gehört) nach lokalen Minimalstellen absuchen. (Hierfür steht kein allgemeines methodisches Vorgehen aus der Analysis zur Verfügung.)

- (M4): Ermittlung des kleinsten lokalen Minimalwertes aus (M2) und (M3) und einer Stelle $x^* \in X$, an der dieses globale Minimum über X angenommen wird.

In der Praxis ist diese Vorgehensweise oft nicht anwendbar. Hierfür können folgende Gründe, die aus der Beschaffenheit des Modells f resultieren, verantwortlich sein:

- (G1): Die Funktion F ist nicht stetig differenzierbar, oder die Bestimmung ihres Gradienten ist zu aufwendig.
(G2): Der Gradient der Zielfunktion ist bekannt, aber die Aufgabe $\nabla F(x) = 0$ ist nicht analytisch lösbar.
(G3): Die Funktion F ist nicht zweimal stetig differenzierbar, oder die Bestimmung des Hesse-Matrix sowie der Test auf positive Definitheit sind zu aufwendig.

In diesem Kapitel werden numerische Iterationsverfahren vorgestellt, die an die analytische Methode (M1-M4) angelehnt sind, d.h. sie ermitteln (zunächst) lokale Minima, steht der Gradient zur Verfügung, so wird die notwendige Bedingung $\nabla F(x) = 0$ aus Satz 2.1 verwendet, usw..

Das Ziel dieser Verfahren ist, ausgehend von einem Startvektor $x^{(0)}$ eine Folge $(x^{(k)})_{k \in \mathbb{N}}$ von Parametersätzen aus X zu erzeugen, die gegen eine Näherung x^* einer lokalen Minimalstelle der Zielfunktion F konvergiert. Die Grundstruktur solch eines lokalen Iterationsverfahrens läßt sich wie folgt beschreiben:

- (L1): Wahl eines Startvektors $x^{(0)}$
→ (L2): Iterationsvorschrift zur Erzeugung einer Folge $(x^{(k)})_{k \in \mathbb{N}}$
(L3): Abbruchkriterium

Wenn das Abbruchkriterium nur notwendige Bedingungen zum Erreichen eines lokalen Minimalpunktes prüft, kann das Iterationsverfahren am Ende auch einen Sattelpunkt oder eine Maximalstelle erreicht haben. Dieser Fall soll in dem nachstehend Betrachteten mit eingeschlossen sein.

Da der Wert des globalen Minimums der Zielfunktion F über X in der hier betrachteten Aufgabenstellung nicht bekannt sein wird, muß zusätzlich zum Abbruchkriterium für ein lokales Minimum noch ein Kriterium zum Erreichen eines "akzeptablen" Minimums (geschätzter globaler Minimalwert) gewählt werden.

Ist die ausgehend von $x^{(0)}$ gefundene Stelle x^* nicht als Lösung der Optimierungsaufgabe akzeptabel, so kann durch einen Neustart des Iterationsverfahrens mit einem Startvektor aus einem anderen Bereich seines Definitionsbereiches versucht werden, einen weiteren lokalen Minimalpunkt zu erreichen, der dann evtl. einen kleineren Funktionswert als die zuerst gefundene Stelle liefert.

Wenn ein Algorithmus zur Bestimmung unterschiedlicher Startvektoren vorliegt, so ergibt sich damit eine sogenannte *Restart-Version* des lokalen Iterationsverfahrens (L1-L3) zur Lösung der Minimierungsaufgabe (2.1):

- (GL1): Wahl eines Startvektors
 - (GL2): Iterationsvorschrift
 - (GL3): Abbruchkriterium zum Finden eines lokalen Minimums (evtl. nur notwendiges Kriterium)
- (GL4): Abbruchkriterium zum Erreichen des geschätzten globalen Minimums

Es werden also solange lokale Minima gesucht, bis ein als global akzeptiertes gefunden ist.

Der Kern dieser Prozedur, das lokale Iterationsverfahren (L1-L3), wird im weiteren diskutiert. Dabei beschränkt sich die Auswahl der in dieser Arbeit betrachteten Verfahren auf solche, die eine Iterationsvorschrift (L2) der Form

$$x^{(k+1)} = x^{(k)} + \lambda_k r^{(k)} \quad (k \in \mathbb{N}) \quad (2.2)$$

enthalten.

Hierbei ist $r^{(k)} \in \mathbb{R}^n$ ein im k -ten Iterationsschritt zu wählender *Richtungsvektor* und $\lambda_k \in \mathbb{R}$ die *Schrittweite*, die von $x^{(k)}$ in Richtung $r^{(k)}$ gegangen werden soll.

In einigen Verfahren werden n Zwischenschritte der Form (2.2), bei denen jeweils nur eine Komponente von $x^{(k)}$ verändert wird, zu einem Iterationszyklus zusammengefaßt. Im j -ten Zwischenschritt ($j=1, \dots, n$) hat der gewählte Richtungsvektor dann die Gestalt

$$r_{(j)} = (r_1, \dots, r_n)^T \text{ mit } r_i = \begin{cases} r & i=j \\ 0 & i \neq j \end{cases} \quad (r \in \mathbb{R}, i=1, \dots, n)$$

(Relaxationsverfahren, s. Kap. 2.4.1).

Wird jeder Richtungsvektor $r^{(k)}$ ($k \in \mathbb{N}$) in (2.2) so gewählt, daß für eine positive Schrittweite $\bar{\lambda}_k \in \mathbb{R}_0$ die sogenannte *Abstiegsforderung*

$$F(x^{(k)} + \lambda_k r^{(k)}) < F(x^{(k)}) \quad \text{für alle } 0 < \lambda_k \leq \bar{\lambda}_k \quad (2.3)$$

erfüllt ist, so heißt das durch (2.2) beschriebene Verfahren *Abstiegsverfahren* und $r^{(k)}$ *Abstiegsrichtung* in $x^{(k)}$.

Die folgende u.a. in [10, Kap. 2.1] nachzulesende Aussage läßt Abstiegsrichtungen erkennen, falls der Gradient der Funktion F bekannt ist:

Satz 2.2: Seien $X \subset \mathbb{R}^n$, $F: X \rightarrow \mathbb{R}$ eine mindestens einmal stetig differenzierbare Funktion, $x \in X$ und $r \in \mathbb{R}^n$ mit

$$r^T \nabla F(x) < 0 \quad (2.4),$$

dann ist r eine Abstiegsrichtung in x .

Liegt mit der Aufgabe (2.1) ein restringiertes Optimierungsproblem vor, so kann an den Richtungsvektor zusätzlich eine sogenannte *Zulässigkeitsforderung* gestellt werden, die verlangt, jede Richtung $r^{(k)}$ ($k \in \mathbb{N}$) so zu wählen, daß eine positive Schrittweite $\bar{\lambda}_k$ existiert mit:

$x^{(k+1)} = x^{(k)} + \lambda_k r^{(k)}$ ist ein *zulässiger Punkt* ($x^{(k+1)} \in X$) für alle $0 < \lambda_k \leq \bar{\lambda}_k$.

Der Vektor $r^{(k)}$ heißt dann *zulässige Richtung*.

In vielen der hier betrachteten Verfahren tritt innerhalb eines Iterationsschrittes die *eindimensionale Minimierungsaufgabe*

$$\min_{\lambda} F(x + \lambda r)$$

auf, in der von x aus entlang der Richtung r für positive (evtl. durch die Zulässigkeitsforderung nach oben beschränkte) oder auch für beliebige $\lambda \in \mathbb{R}$ nach einem Minimum der Zielfunktion F gesucht wird.

Diese Teilaufgabe des Optimierungsverfahrens wird in diesem Zusammenhang dann auch *lineare Minimumsuche*, *lineare Suche*, *Strahlminimierung* oder *Schrittweitenbestimmung* nach dem *Cauchy-Prinzip* genannt und wird unter Punkt 2.3 zunächst behandelt, bevor auf Verfahren für die mehrdimensionale Optimierung in Kapitel 2.4 näher eingegangen wird.

Die lokalen Iterationsverfahren lassen sich je nach Umfang der Informationen über die Modellfunktion f und damit auch über die Zielfunktion F entsprechend der Punkte (G1-G3) in drei Gruppen unterteilen:

1. ~~ableitungsfreie Verfahren~~ (Suchmethoden)
2. ~~Gradientenmethoden~~
3. ~~Newtonmethoden~~.

Während bei den ableitungsfreien Verfahren nur mit den Funktionswerten von F gearbeitet wird, benutzt man bei den Gradientenmethoden neben den Funktionswerten noch die

1. Ableitungen und bei den Newtonverfahren zusätzlich die
2. Ableitungen, um nach lokalen Minima zu suchen.

Aus jeder dieser Gruppen werden unter Punkt 2.4.1 bis 2.4.3 im weiteren Verlauf dieses Kapitels repräsentative Verfahren vorgestellt.

2.3 Die eindimensionale Minimierungsaufgabe

Gesucht wird ein Minimum der Zielfunktion F über der Linie $L := \{(x + \lambda r) \in X : \lambda \in \mathbb{R}\}$ für feste $x \in X$ und $r \in \mathbb{R}^n$.

Um den Rechenaufwand dieser Berechnung klein zu halten, wird zur Vereinfachung angenommen, daß die Funktion F über L *unimodal* ist, d.h. sie besitzt hier nur eine lokale Minimalstelle, die also zugleich die globale über L ist.

Ziel aller hier behandelten Methoden ist, ein zu Beginn ermitteltes *Unbestimmtheitsintervall* $[a, b]$, in dem das Minimum liegt, mit wenigen Funktionswert- und evtl. auch Ableitungsberechnungen solange zu verkleinern, bis eine vorher bestimmte Genauigkeitsgrenze der Länge des Intervalls unterschritten ist.

Führt man die Hilfsfunktion $\varphi: \mathbb{R} \rightarrow \mathbb{R}$, $\varphi(\lambda) := F(x + \lambda r)$ ein, so ist φ insbesondere auf dem Intervall $[a, b]$ unimodal, und es gilt der u.a. in [2, Kap.5.4] gezeigte Satz:

Satz 2.3: Seien $\varphi: \mathbb{R} \rightarrow \mathbb{R}$ unimodal auf $[a, b]$, $\lambda^*, \lambda_1, \lambda_2 \in [a, b]$, wobei λ^* der Minimalpunkt von φ sei und $\lambda_1 < \lambda_2$ gelte. Gilt $\varphi(\lambda_1) > \varphi(\lambda_2)$, so ist $\lambda^* \in [\lambda_1, b]$, gilt hingegen $\varphi(\lambda_1) < \varphi(\lambda_2)$, so liegt λ^* in $[a, \lambda_2]$, und für den Fall $\varphi(\lambda_1) = \varphi(\lambda_2)$ ist $\lambda^* \in [\lambda_1, \lambda_2]$.

Mit Satz 2.3 kann das Unbestimmtheitsintervall durch zwei Funktionswertberechnungen eingeschränkt werden. Eine erste Phase sucht also zunächst ein Intervall $[a, b]$ mit

$$\varphi(a) > \varphi(\lambda^*) < \varphi(b) \quad (2.5)$$

In vielen der in dieser Arbeit auftretenden Anwendungen der linearen Minimumsuche gilt die Abstiegsforderung (2.4), d.h. es existiert ein $\bar{\lambda} > 0$ mit $\varphi(\lambda) < \varphi(0)$ für alle $0 \leq \lambda \leq \bar{\lambda}$. Dann wird oft schon das Intervall $[0, 1]$ die Bedingung (2.5) erfüllen, und auch sonst ist nicht zu erwarten, daß das Minimum der Funktion φ erst für sehr große λ^* ($\lambda^* \gg 1$, $\lambda^* \rightarrow \infty$) angenommen wird, s. [20, Kap.4.7]. Die folgende Prozedur zur Bestimmung des ersten Unbestimmtheitsintervalls $[a, b]$ ist daher in [20, Kap.4.7] angegeben:

Gilt $\varphi(1) > \varphi(0)$, dann wähle $[a,b]:=[0,1]$, sonst bestimme $\varphi(2), \varphi(2^2), \dots, \varphi(2^p), \dots$ ($p=1,2,\dots$) bis $\varphi(2^{i-1}) > \varphi(2^i) < \varphi(2^{i+1})$ für ein $i \in \{0,1,2,\dots\}$ erfüllt ist. (Ist $i=0$, dann sei 2^{-1} zu 0 gesetzt.) Die Minimalstelle λ^* liegt dann in dem Intervall $[a,b]:=[2^{i-1}, 2^{i+1}]$.

Dieses Intervall kann sofort verkleinert werden, wenn neben den Funktionswerten noch die 1. Ableitungen der Funktion φ bekannt sind. Analog zum Bisektionsverfahren (s. Punkt 3.) liegt λ^* hier in $[2^{i-1}, 2^i]$, falls $\varphi'(2^i) > 0$ ist, oder in $[2^i, 2^{i+1}]$, wenn $\varphi'(2^i) < 0$ gilt. Für den Fall $\varphi'(2^i) = 0$ ist das Minimum der Funktion φ an der Stelle $\lambda^*=2^i$ gefunden.

Gilt die Abstiegsforderung nicht, und sind auch negative Schrittweiten zugelassen, so kann das Intervall $[a,b]$ auf ähnliche Weise durch zusätzliche Funktionswertberechnungen an negativen Stellen ermittelt werden.

In der zweiten Phase, der Minimumsuche über dem Intervall $[a,b]$, können z.B. die folgenden Verfahren angewendet werden, die neben weiteren u.a. in [9,Kap.4.1.4],[17,Kap.6.3] und [11] beschrieben sind. Die Methoden 1., 2. und 4.a) arbeiten nur mit Funktionswertbestimmungen, während in 3. und 4.b) auch die 1. Ableitungen berechnet werden müssen. Die Ableitungen der Funktion φ nach der Schrittweite λ können explizit berechnet oder aus der Beziehung

$$\varphi'(\lambda) = \nabla F(x^{(k)} + \lambda r^{(k)})^T r^{(k)}$$

bestimmt werden.

1. Die äquidistante Suche: Das Intervall $[a,b]$ wird in $N+1$ gleich große Intervalle $[a,a+\delta], [a+\delta,a+2\delta], \dots, [a+N\delta,b]$ unterteilt mit $\delta = (b-a)/(N+1)$, und an den Stützstellen $a+k\delta$ ($k=1,\dots,n$) werden die Funktionswerte berechnet. Liegt an der Stelle $\bar{\lambda} = a + \bar{k}\delta$ für ein $\bar{k} \in \{1,\dots,n\}$ der kleinste Funktionswert, so ist λ^* in dem Intervall $[a-\delta, a+\delta]$ enthalten. N ist also größer als $2(b-a)/1 - 1$ zu wählen, wenn das Verfahren mit einer Länge des Unbestimmtheitsintervalls kleiner als 1 abbrechen soll.

2. Die Fibonacci-Suche verkleinert in N Iterationsschritten das Unbestimmtheitsintervall bis auf eine Länge kleiner als 1, wenn N vorweg durch die Beziehung $F_N > (b-a)/1$ bestimmt wird, wobei F_N aus den Fibonacci-Zahlen $(F_j)_{j \in \mathbb{N}}$ mit $F_0=F_1=1$ und $F_{j+1}=F_j+F_{j-1}$ ($j=1,2,\dots$) ist. Die Funktion φ wird, beginnend mit $[a_{(1)}, b_{(1)}]:=[a,b]$, im k -ten Iterationsschritt ($k=1,\dots,N$) an den Stellen

$$\lambda_1 = a_{(k)} + \frac{F_{N-k-1}}{F_{N-k+1}} (b_{(k)} - a_{(k)}) \quad \text{bzw.} \quad \lambda_2 = a_{(k)} + \frac{F_{N-k}}{F_{N-k+1}} (b_{(k)} - a_{(k)})$$

berechnet. (Nur im 1. Iterationsschritt müssen $\varphi(\lambda_1)$ und $\varphi(\lambda_2)$ bestimmt werden, in allen anderen Schritten ist einer der beiden Funktionswerte schon aus dem vorherigen Zyklus bekannt.)

Ist $\varphi(\lambda_1) > \varphi(\lambda_2)$, so setze $[a_{(k+1)}, b_{(k+1)}]:=[\lambda_1, b_{(k)}]$, und sonst wird $[a_{(k+1)}, b_{(k+1)}]:=[a_{(k)}, \lambda_2]$.

3. Das Bisektionsverfahren erreicht bei N Iterationsschritten ein Unbestimmtheitsintervall der Länge kleiner als 1, wenn für N : $(\frac{1}{2})^N < 1/(b-a)$ gilt.

Ausgehend vom Intervall $[a_{(1)}, b_{(1)}]:=[a,b]$ wird im k -ten Iterationsschritt ($k=1,\dots,N$) an der Stelle $\lambda = \frac{1}{2}(a_{(k)} + b_{(k)})$ die 1. Ableitung $\varphi'(\lambda)$ berechnet.

Gilt $\varphi'(\lambda) = 0$, so endet das Verfahren mit der gefundenen Minimalstelle $\lambda^* := \lambda$.

Ist $\varphi'(\lambda) > 0$, dann wird im nächsten Iterationsschritt in dem Intervall $[a_{(k+1)}, b_{(k+1)}]:=[a_{(k)}, \lambda]$ weitergesucht und sonst in $[a_{(k+1)}, b_{(k+1)}]:=[\lambda, b_{(k)}]$.

4. Interpolationsverfahren

a) Quadratische Interpolation:

Ausgehend von $[a_{(1)}, b_{(1)}]:=[a,b]$ und $\lambda_{(1)} := \frac{1}{2}(a+b)$ werden im k -ten Iterationsschritt ($k=1,2,\dots$) an den Stellen $a_{(k)}$, $\lambda_{(k)}$ und $b_{(k)}$ die Funktionswerte von φ berechnet, und das durch die Punkte $(a_{(k)}, \varphi(a_{(k)}))$, $(\lambda_{(k)}, \varphi(\lambda_{(k)}))$ und $(b_{(k)}, \varphi(b_{(k)}))$ eindeutig bestimmte quadratische Interpolationspolynom ϕ wird (z.B. mit der Lagrangeschen Interpolationsformel) ermittelt. Das Minimum von ϕ über $[a_{(k)}, b_{(k)}]$ läßt sich analytisch

bestimmen. Es liegt an der Stelle

$$\lambda_{(k)}^* = \frac{u_{\lambda b} \varphi(a_{(k)}) + u_{ba} \varphi(\lambda_{(k)}) + u_{a\lambda} \varphi(b_{(k)})}{v_{\lambda b} \varphi(a_{(k)}) + v_{ba} \varphi(\lambda_{(k)}) + v_{a\lambda} \varphi(b_{(k)})}$$

$$\text{mit } u_{ij} = i_{(k)}^2 - j_{(k)}^2 \quad \text{und} \quad v_{ij} = i_{(k)} - j_{(k)}.$$

(Eine Herleitung ist z.B. in [9, Kap. 4.1.4.3] nachzulesen.) Ist $\varphi(\lambda_{(k)}^*) > \varphi(\lambda_{(k)})$, dann wird für den nächsten Iterationsschritt $a_{(k+1)} := a_{(k)}$, $\lambda_{(k+1)} := \lambda_{(k)}$ und $b_{(k+1)} := \lambda_{(k)}^*$ und sonst wird $a_{(k+1)} := \lambda_{(k)}$, $\lambda_{(k+1)} := \lambda_{(k)}^*$ und $b_{(k+1)} := b_{(k)}$ gesetzt. Diese Iteration wird fortgeführt, bis ein Unbestimmtheitsintervall mit einer Länge kleiner als die gewünschte Genauigkeitsgrenze 1 übrigbleibt.

b) Kubische Interpolation:

Hier wird im k-ten Iterationsschritt das kubische Interpolationspolynom Φ durch die Punkte $(a_{(k)}, \varphi(a_{(k)}))$, $(a_{(k)}, \varphi'(a_{(k)}))$, $(b_{(k)}, \varphi(b_{(k)}))$ und $(b_{(k)}, \varphi'(b_{(k)}))$ z.B. mit der Hermiteschen Interpolationsformel bestimmt und die Minimalstelle von Φ mit

$$\lambda_{(k)}^* = b_{(k)} - (b_{(k)} - a_{(k)}) \left(\frac{\varphi'(b_{(k)}) + u_2 - u_1}{\varphi'(b_{(k)}) - \varphi'(a_{(k)}) + 2u_2} \right),$$

$$u_1 = \varphi'(a_{(k)}) + \varphi'(b_{(k)}) - 3 \left(\frac{\varphi(a_{(k)}) - \varphi(b_{(k)})}{a_{(k)} - b_{(k)}} \right) \quad \text{und}$$

$$u_2 = \sqrt{u_1^2 - \varphi'(a_{(k)}) \varphi'(b_{(k)})}$$

(s. [9, Kap. 4.1.4.3]) ermittelt. Wie im Bisektionsverfahren wird im Intervall $[a_{(k)}, \lambda_{(k)}^*]$ weitergesucht, falls $\varphi'(\lambda_{(k)}^*) > 0$ und in $[\lambda_{(k)}^*, b_{(k)}]$, wenn $\varphi'(\lambda_{(k)}^*) < 0$ gilt. Das Verfahren bricht ab, sobald $\varphi'(\lambda_{(k)}^*) = 0$ gilt oder das Intervall $[a_{(k)}, b_{(k)}]$ genügend klein ist (für ein $k \in \mathbb{N}$).

Die Verfahren 2. bis 4.b) können vereinfacht auch abbrechen, sobald nach dem Ausgangsintervall $[a, b]$ ein zweites Mal ein Intervall $[a_{(k)}, b_{(k)}]$ ($k > 1$) gefunden wird, in dem in

Anlehnung an die Beziehung (2.5) gilt:

$\varphi(a_{(k)}) > \varphi(\lambda) < \varphi(b_{(k)})$, wobei λ jeweils die dritte Stelle ($\lambda \in (a_{(k)}, b_{(k)})$) sein soll, zu der in den Verfahren der Funktionswert von φ berechnet wurde. Der Näherungswert für die Minimalstelle von φ ist dann $\lambda^* := \lambda$.

Dieses Abbruchkriterium wird z.B. in [20, Kap. 4.7] angewendet und ist unabhängig von der Länge des Unbestimmtheitsintervalls am Ende der Iteration.

2.4. Numerische Iterationsverfahren zur mehrdimensionalen Optimierungsaufgabe

Es werden Verfahren der in Kapitel 2.2 besprochenen Struktur zur Bestimmung lokaler Minima der Zielfunktion F aus der Optimierungsaufgabe (2.1) vorgestellt. Hierfür werden die unterschiedlichen Iterationsvorschriften der Form (2.2) beschrieben, sowie Vor- und Nachteile der einzelnen Verfahren diskutiert.

Mit $\|\dots\|$ sei im folgenden die euklidische Norm $\|\dots\|_2$ bezeichnet.

2.4.1. Ableitungsfreie Methoden:

Die einfachste Methode, ausschließlich durch Funktionswertberechnungen ein lokales Minimum von F zu suchen, ist die *sukzessive Variation der Variablen*. In diesem Relaxationsverfahren unterteilt sich ein Iterationszyklus in n Schritte, in denen jeweils in einer Koordinatenrichtung des Parametervektors nach einem lokalen Minimum der Zielfunktion F gesucht wird.

Der k-te Iterationsschritt ($k \in \mathbb{N}$) beginnt mit $x_{(1)} := x^{(k)}$ (für $k=0$ ist $x^{(k)}$ der Startvektor $x^{(0)}$, und sonst wurde $x^{(k)}$ im $(k-1)$ -ten Iterationszyklus bestimmt). Im j-ten Zwischenschritt ($j=1, \dots, n$) wird zu der Suchrichtung

$$r_{(j)} = (r_1, \dots, r_n)^T \quad \text{mit} \quad r_i = \begin{cases} 1, & i=j \\ 0, & i \neq j \end{cases} \quad (i=1, \dots, n)$$

eine optimale Schrittweite $\lambda_j \in \mathbb{R}$ gesucht mit :

$$F(x_{(j)} + \lambda_j r_{(j)}) = \min_{\lambda \in \mathbb{R}} F(x_{(j)} + \lambda r_{(j)})$$

(s. Kapitel 2.3.: ableitungsfreie Verfahren 1., 2. und 4.a)) und $x_{(j+1)} := x_{(j)} + \lambda_j r_{(j)}$ für den nächsten Zwischenschritt gesetzt. Der Ausgangspunkt für die Minimasuche im darauffolgenden Iterationszyklus ist dann $x^{(k+1)} := x_{(n+1)}$.

Diese Vorgehensweise erfordert jedoch unter Umständen eine hohe Anzahl von Funktionswertberechnungen, da in den n eindimensionalen Minimumsuchen, die in jedem Iterationsschritt enthalten sind, keine Informationen über Einschränkungsmöglichkeiten auf ein Suchintervall aus $\mathbb{R}$ vorliegen. Die Bestimmung eines ersten Unbestimmtheitsintervalls $[a, b]$ muß also zu Beginn jedes Zwischenschrittes durch Berechnung von Funktionswerten an positiven wie auch negativen Stellen erfolgen (s. Kapitel 2.3).

Weiterentwicklungen dieses Verfahrens, wie z.B. die Methoden von HOOKE-JEEVES, ROSENBROCK [9, Kap.4.2.5.1] oder POWELL [2, Kap.5.2] nutzen daher Informationen, die in einem Iterationsschritt gewonnen wurden, in den darauffolgenden aus.

Im ROSENBROCK-Verfahren kann auf die lineare Minimumsuche verzichtet werden. Die Schrittweiten werden innerhalb eines Iterationszyklusses je nach Erfolg oder Mißerfolg eines Versuchsschrittes in einer Richtung vergrößert oder verkleinert, und am Ende eines Iterationsschrittes wird das System der n Suchrichtungen aus Informationen über die erfolgreichen Schritte des Zyklusses neu konstruiert.

Zu Beginn der Iteration wird die Zielfunktion von einem Startvektor $x^{(0)}$ in den Koordinatenrichtungen $r_{(1)}^{(0)}, \dots, r_{(n)}^{(0)}$ (s. sukzessive Variation der Variablen) mit den Anfangsschrittweiten $\lambda_1 = \dots = \lambda_n = h > 0$ ($h=0,1$ nach einem Vorschlag von Rosenbrock) abgetastet.

Der k -te Iterationsschritt teilt sich dann in Phasen von jeweils n Zwischenschritten. Die erste Abtastphase beginnt an der Stelle $x_{(1)} := x^{(k)}$ und alle weiteren in $x_{(i)} := x_{(n+1)}$,

wobei $x_{(n+1)}$ die zuletzt erreichte Stelle der vorherigen Phase ist.

Liefert die Stelle $x = x_{(j)} + \lambda_j r_{(j)}^{(k)}$ im j -ten Zwischenschritt einer Phase einen kleineren Funktionswert als $x_{(j)}$, so war der Schritt $\lambda_j r_{(j)}^{(k)}$ erfolgreich, es wird $x_{(j+1)} := x$ gesetzt und die Schrittweite λ_j für einen Schritt in der Richtung $r_{(j)}^{(k)}$ in der nächsten Phase verdreifacht. Gilt hingegen $Q(x) \geq Q(x_{(j)})$, so war der Schritt $\lambda_j r_{(j)}^{(k)}$ nicht erfolgreich, es wird $x_{(j+1)} := x_{(j)}$ gesetzt und λ_j für den nächsten Schritt in der Richtung $r_{(j)}^{(k)}$ mit $(-\frac{1}{2})$ multipliziert.

Diese Phase von n Schritten wird mit den jeweils angepaßten Schrittweiten wiederholt, bis in jeder der n Suchrichtungen nach mindestens einem erfolgreichen Schritt schließlich ein erfolgloser aufgetreten ist. Der Parametersatz $x^{(k+1)}$ für den nächsten Iterationszyklus ergibt sich dann zu $x_{(n+1)}$ aus der letzten Phase. Aus den erfolgreichen Schritten des k -ten Iterationsschrittes werden n linear unabhängige Vektoren $w_{(1)}, \dots, w_{(n)} \in \mathbb{R}^n$ konstruiert mit:

$$w_{(i)} = \sum_{j=1}^n s_j r_{(j)}^{(k)} \quad (i=1, \dots, n)$$

wobei s_j die Summe aller erfolgreichen Schrittweiten λ_j in der Richtung $r_{(j)}^{(k)}$ ist ($j \in \{1, \dots, n\}$). Das System der n Suchrichtungen $r_{(1)}^{(k+1)}, \dots, r_{(n)}^{(k+1)}$ für den nächsten Iterationszyklus entsteht durch Transformation der Vektoren $w_{(1)}, \dots, w_{(n)}$ in ein orthonormales System (z.B. mit der GRAM-SCHMIDT'schen Methode und Normierungen):

$$\begin{aligned} r_{(1)}^{(k+1)} &:= \frac{w_{(1)}}{\|w_{(1)}\|} \quad \text{und} \\ r_{(j)}^{(k+1)} &:= \frac{w_{(j)} - \sum_{i=1}^{j-1} (w_{(j)}^T r_{(i)}^{(k+1)}) r_{(i)}^{(k+1)}}{\|w_{(j)} - \sum_{i=1}^{j-1} (w_{(j)}^T r_{(i)}^{(k+1)}) r_{(i)}^{(k+1)}\|} \quad \text{für } j=2, \dots, n \end{aligned} \quad (2.6)$$

Der aufwendigste Teil in dem ROSENBROCK-Verfahren ist die Orthogonalisierung nach GRAM-SCHMIDT. Eine Vereinfachung dieses Prozesses wurde daher für diese Anwendung von PALMER [14] entwickelt. Ein Vorteil des Verfahrens von ROSENBROCK ist, daß sich die Schrittweiten in den einzelnen Suchrichtungen unabhängig und schnell verändern können. In der Nähe des Minimums kann allerdings, bedingt durch die Wahl der Startschrittweiten eines Iterationszyklusses, ein "Herumirren" um den Minimalpunkt auftreten [9, Kap. 4.2.5.1.2].

In den hier beschriebenen Relaxationsverfahren können nicht nur bei verschiedenen Startvektoren, sondern auch durch Vertauschung der Koordinatenrichtungen als Suchrichtungen unterschiedliche lokale Minima am Ende des Iterationsverfahrens erreicht werden.

Im Anhang A.1 befinden sich die Flußdiagramme für die sukzessive Variation der Variablen und das ROSENBROCK-Verfahren.

Als ableitungsfreie Verfahren können auch die Gradientenmethoden (s. 2.4.2) benutzt werden, wenn statt der Gradienten ersatzweise Differenzenquotienten gebildet werden. Dabei ist jedoch zu beachten, daß, um z.B. neben dem Funktionswert auch den zentralen Differenzenquotienten an einer Stelle zu erhalten, zwei zusätzliche Funktionswertberechnungen notwendig sind. Die oben erwähnte POWELL-Methode [2, Kap. 5.2] ist z.B. eine ableitungsfreie Variante des konjugierten Gradientenverfahrens.

2.4.2 Gradientenmethoden

In den Gradientenverfahren werden die Richtungsvektoren unabhängig vom Koordinatensystem mit Hilfe der aus der Gradientenbestimmung zusätzlich gewonnenen Information über die Form der Zielfunktion gewählt.

Für das optimale Gradientenverfahren wird ausgenutzt, daß der negative Gradient $-\nabla F(x)$ einer mehrdimensionalen Funktion F in Richtung des steilsten Abstiegs der Funktion an der Stelle x zeigt.

Ausgehend von einem Startvektor $x^{(0)}$ wird im k -ten Iterationsschritt ($k \in \mathbb{N}$) die Suchrichtung $r^{(k)}$ auf den Wert des Antigradients $-\nabla F(x^{(k)})$ gesetzt und eine optimale Schrittweite $\lambda_k > 0$ mit einem Verfahren zur Strahlminimierung aus Kapitel 2.3 bestimmt. (Sinnvoll sind die Methoden 3. und 4.b), die auch mit den 1. Ableitungen arbeiten.) Der nächste Iterationspunkt ergibt sich dann zu

$$x^{(k+1)} = x^{(k)} - \lambda_k \nabla F(x^{(k)}) \quad (2.7)$$

Wegen $r^{(k)} \nabla F(x^{(k)}) = -\|\nabla F(x^{(k)})\|^2 < 0$ ist der k -te Richtungsvektor für $\nabla F(x^{(k)}) \neq 0$ ($k \in \mathbb{N}$) nach Satz 2.2 (Kap. 2.2) eine Abstiegsrichtung und die optimale Gradientenmethode demzufolge ein Abstiegsverfahren.

Sowohl die Folge der Iterationspunkte $(x^{(k)})_{k \in \mathbb{N}}$ als auch die ihrer Funktionswerte $(F(x^{(k)}))_{k \in \mathbb{N}}$ konvergieren unter gewissen Voraussetzungen an die Zielfunktion F mindestens linear. Dieses liefert die folgende Konvergenzaussage, die neben ähnlichen Konvergenzsätzen u.a. in [10, Kap. 2.5] oder [2, Kap. 5.1] bewiesen wird:

Satz 2.4: Ist F eine mindestens zweimal stetig differenzierbare Funktion über $\mathbb{R}^n$, und gibt es Konstanten $M \geq m > 0$, derart daß für die Hesse-Matrix $\nabla^2 F(x)$ die Beziehung

$$m\|y\|^2 \leq y^T \nabla^2 F(x) y \leq M\|y\|^2 \quad \text{für alle } x, y \in \mathbb{R}^n \text{ gilt,}$$

dann konvergiert die durch das optimale Gradientenverfahren erzeugte Folge $(x^{(k)})_{k \in \mathbb{N}}$ für jeden Startwert $x^{(0)}$ gegen die (eindeutige) Minimumstelle x^* von F , und es gibt eine Konstante $c < \infty$, so daß mit $q = (M-m)/(M+m)$

C paarweise konjugiert, d.h. es gilt $r^{(i)T}Cr^{(j)}=0$ für alle $i, j \in \{0, \dots, n-1\}$ mit $i \neq j$, dann erreicht das Iterationsverfahren

$$x^{(k+1)} = x^{(k)} + \lambda_k r^{(k)} \quad \text{mit} \quad \lambda_k = - \frac{(Cx^{(k)} + p)^T r^{(k)}}{r^{(k)T} C r^{(k)}}$$

für jeden Startvektor $x^{(0)}$ in höchstens n Schritten das Minimum von F.

Die Schrittweite λ_k ist dabei im k-ten Iterationsschritt ($k=0, \dots, n-1$) so gewählt, daß sie F auf der Geraden $x^{(k)} + \lambda r^{(k)}$ minimiert ($\lambda \in \mathbb{R}$).

Die konjugierten Richtungen $r^{(0)}, \dots, r^{(n-1)}$ werden bei HESTENES und STIEFEL während des Iterationsverfahrens erzeugt. Beginnend mit $r^{(0)} = -\nabla F(x^{(0)})$ wird $r^{(k+1)}$ im k-ten Iterationsschritt durch die Vorschrift

$$r^{(k+1)} = -\nabla F(x^{(k+1)}) + \beta_k r^{(k)} \quad \text{mit} \quad \beta_k = \frac{\nabla F(x^{(k+1)})^T C r^{(k)}}{r^{(k)T} C r^{(k)}} \quad (2.8)$$

berechnet.

Die Rekursionsformel (2.8) entsteht aus der GRAM-SCHMIDT-Methode zur Überführung von n linear unabhängigen Vektoren $p^{(0)}, \dots, p^{(n-1)}$ (hier: $p^{(i)} = -\nabla F(x^{(i)})$ ($i=0, \dots, n-1$)) in n bezüglich einer symmetrischen, positiv definiten (nxn)-Matrix C konjugierten Richtungen $r^{(0)}, \dots, r^{(n-1)}$, die den gleichen Raum wie $p^{(0)}, \dots, p^{(n-1)}$ aufspannen:

$$r^{(0)} := p^{(0)}, \quad r^{(i)} := p^{(i)} - \sum_{l=1}^{i-1} \frac{p^{(i)T} C r^{(l)}}{r^{(l)T} C r^{(l)}} r^{(l)} \quad (i=1, \dots, n-1)$$

In [2, Kap. 5.2] wird gezeigt, daß das Verfahren von HESTENES und STIEFEL die folgenden Eigenschaften besitzt:

- (E1) $r^{(k)T}Cr^{(l)}=0$ für alle $l < k \leq n-1$
(die Richtungsvektoren $r^{(k)}$ sind paarw. konjugiert bzgl. C)
- (E2) $\nabla F(x^{(k)})^T \nabla F(x^{(l)})=0$ für alle $l < k \leq n$
(die Gradienten $\nabla F(x^{(k)})$ sind paarweise orthogonal)

- (E3) $\nabla F(x^{(k)})^T d^{(l)}=0$ für alle $l < k \leq n$
- (E4) $\nabla F(x^{(k)})^T C d^{(l)}=0$ für alle $l < k-1 \leq n-1$

Die Rekursionsformeln aus dem Verfahren von HESTENES und STIEFEL lassen sich so umformen, daß sie auch für nicht-quadratische Zielfunktionen definiert sind. In dem Verfahren von FLETSCHER und REEVES geschieht dies durch die Beziehung

$$\begin{aligned} \nabla F(x^{(k+1)}) &= Cx^{(k+1)} + p = C(x^{(k)} + \lambda_k r^{(k)}) + p = \\ &= \nabla F(x^{(k)}) + \lambda_k C r^{(k)} \quad (k=0, \dots, n-1) \end{aligned}$$

Der Faktor β_k ($k=0, \dots, n-1$) aus (2.8) läßt sich damit und mit den Eigenschaften (E2) und (E3) umformen zu:

$$\begin{aligned} \beta_k &= \frac{\nabla F(x^{(k+1)})^T C r^{(k)}}{r^{(k)T} C r^{(k)}} = \\ &= \frac{\nabla F(x^{(k+1)})^T (\nabla F(x^{(k+1)}) - \nabla F(x^{(k)}))}{(-\nabla F(x^{(k)}) + \beta_{k-1} r^{(k-1)})^T (\nabla F(x^{(k+1)}) - \nabla F(x^{(k)}))} = \\ &= - \frac{\|\nabla F(x^{(k+1)})\|^2}{\|\nabla F(x^{(k)})\|^2} \end{aligned}$$

Für das Iterationsverfahren wird dann, ausgehend von einem Startvektor $x^{(0)}$ und der ersten Suchrichtung $r^{(0)} = -\nabla F(x^{(0)})$, im k-ten Iterationszyklus ($k \in \mathbb{N}$) eine optimale Schrittweite λ_k bestimmt, die F auf der Geraden $x^{(k)} + \lambda r^{(k)}$ ($\lambda \in \mathbb{R}$) minimiert (s. Kap. 2.3: 3. oder 4.b)), und für den nächsten Iterationsschritt

$$\begin{aligned} x^{(k+1)} &:= x^{(k)} + \lambda_k r^{(k)} \quad \text{und} \\ r^{(k+1)} &:= -\nabla F(x^{(k+1)}) + \beta_k r^{(k)} \quad \text{mit} \quad \beta_k = - \frac{\|\nabla F(x^{(k+1)})\|^2}{\|\nabla F(x^{(k)})\|^2} \end{aligned}$$

gesetzt.

Das FLETCHER-REEVES-Verfahren ist wegen

$$\begin{aligned} \nabla F(x^{(k)})^T r^{(k)} &= \nabla F(x^{(k)})^T (-\nabla F(x^{(k+1)}) + \beta_{k-1} r^{(k-1)}) = \\ &= -\|\nabla F(x^{(k)})\|^2 < 0 \quad (k \in \mathbb{N}) \end{aligned}$$

für $\nabla F(x^{(k)}) \neq 0$ ($\in \mathbb{R}^n$) (hier geht die Eigenschaft (E3) ein)

ein Abstiegsverfahren. Es ist jedoch zu bedenken, daß die Schrittweiten λ_k ($k \in \mathbb{N}$) mit hoher Genauigkeit bestimmt werden müssen, damit die Richtungsvektoren konjugiert sind und die Eigenschaften (E1)-(E4) gelten.

Nach [9, Kap. 4.2.5.2.2] können mit dem FLETCHER-REEVES-Verfahren manchmal in aufeinanderfolgenden Iterationsschritten nahezu parallele Richtungsvektoren erzeugt werden. Dieses Verhalten kann durch eine *Restart-Version*, bei der nach je n Iterationszyklen wieder mit der Antigradientenrichtung als Suchrichtung begonnen wird, vermieden werden.

Eine weitere Methode mit konjugierten Richtungen ist das *Iterationsverfahren von FLETCHER und POWELL*.

Ausgehend von einem Startvektor $x^{(0)}$ und einer symmetrischen, positiv definiten $(n \times n)$ -Matrix $H^{(0)}$ wird hier im k -ten Iterationsschritt ($k \in \mathbb{N}$) eine optimale Schrittweite λ_k zu der Richtung $r^{(k)} = -H^{(k)} \nabla F(x^{(k)})$ berechnet und für den nächsten Iterationsschritt

$$\begin{aligned} x^{(k+1)} &:= x^{(k)} + \lambda_k r^{(k)} \quad \text{und} \\ H^{(k+1)} &:= H^{(k)} + A^{(k)} + B^{(k)} \quad \text{mit} \\ A^{(k)} &:= - \frac{\lambda_k r^{(k)} r^{(k)T}}{r^{(k)T} \nabla F(x^{(k)})}, \quad B^{(k)} := \frac{H^{(k)} u_k u_k^T H^{(k)}}{u_k^T H^{(k)} u_k} \quad \text{und} \\ u_k &:= (\nabla F(x^{(k+1)}) - \nabla F(x^{(k)})) \end{aligned} \quad (2.9)$$

gesetzt. Gilt $\nabla F(x^{(k)}) \neq 0$, so sind die Matrizen $A^{(k)}$, $B^{(k)}$ ($k \in \mathbb{N}$) definiert.

In [2, Kap. 5.2.] wird gezeigt, daß die Matrizen $H^{(k)}$ in jedem Iterationsschritt symmetrisch und positiv definit sind. Daher gilt für $\nabla F(x^{(k)}) \neq 0$ ($\in \mathbb{R}^n$):

$$\nabla F(x^{(k)})^T r^{(k)} = -\nabla F(x^{(k)})^T H^{(k)} \nabla F(x^{(k)}) < 0 \quad (k \in \mathbb{N})$$

Also ist das FLETCHER-POWELL-Verfahren ein Abstiegsverfahren. Für eine quadratische Zielfunktion $F = x^T C x + p$

(C symmetrisch und positiv definit) und die Einheitsmatrix als Startmatrix $H^{(0)}$ erzeugt das Verfahren die gleichen Iterationspunkte wie das Verfahren von HESTENES und STIEFEL, und es gilt $H^{(n)} = -C^{-1}$. Die Folge der Matrizen $(H^{(k)})_{k \in \mathbb{N}}$ liefert also Näherungen für die Hesse-Matrix $\nabla^2 F$ der Zielfunktion, und in n Schritten wird die Lösung der quadratischen Funktion F gefunden.

Das Verfahren fordert von den hier beschriebenen Gradientenmethoden durch die Berechnungen der Matrizen $H^{(k)}$ ($k \in \mathbb{N}$) den meisten Rechenaufwand in einem Iterationsschritt, und wie beim FLETCHER-REEVES-Verfahren kann die zu erwartende hohe Konvergenzgeschwindigkeit in Minimumnähe abnehmen, wenn die optimale Schrittweite λ_k ($k \in \mathbb{N}$) zu ungenau bestimmt wird. In [10, Kap. 2.7.6] wird an Beispielen demonstriert, daß das FLETCHER-POWELL-Verfahren schon bei kleinen Ungenauigkeiten in der Schrittweitenbestimmung langsamer als die in diesem Fall unempfindlichere optimale Gradientenmethode konvergieren kann.

Die Flußdiagramme zu der optimalen Gradientenmethode, dem FLETCHER-REEVES-Verfahren als Restart-Version und der FLETCHER-POWELL-Methode befinden sich im Anhang A.1.

2.4.3 Newtonmethoden:

Ist die Zielfunktion F mindestens zweimal stetig differenzierbar, so kann sie in der Umgebung eines Punktes $x^{(k)}$ mit Hilfe einer abbrechenden Taylorreihe durch

$$\begin{aligned} \tilde{F}(x) &:= F(x^{(k)}) + \nabla F(x^{(k)})^T (x - x^{(k)}) + \\ &\quad + \frac{1}{2} (x - x^{(k)})^T \nabla^2 F(x^{(k)}) (x - x^{(k)}) \end{aligned}$$

quadratisch approximiert werden. Wird nun statt $\nabla F(x^*) = 0$ näherungsweise $\nabla \tilde{F}(\tilde{x}) = 0$ gefordert, so kann $\tilde{x}$ für eine nicht singuläre Hesse-Matrix $\nabla^2 F(x^{(k)})$ eindeutig bestimmt werden mit:

$$\tilde{x} = x^{(k)} - (\nabla^2 F(x^{(k)}))^{-1} \nabla F(x^{(k)})$$

Aus diesen Überlegungen läßt sich ein Iterationsverfahren konstruieren, das sog. *gewöhnliche NEWTON-Verfahren* :
Ausgehend von einem Startvektor $x^{(0)}$ wird im k -ten Iterationsschritt ($k \in \mathbb{N}$) der nächste Suchpunkt $x^{(k+1)}$ mit der Vorschrift :

$$x^{(k+1)} = x^{(k)} - (\nabla^2 F(x^{(k)}))^{-1} \nabla F(x^{(k)}) \quad (2.10)$$

oder durch die Lösung des Gleichungssystems

$$\nabla^2 F(x^{(k)}) x^{(k+1)} = \nabla^2 F(x^{(k)}) x^{(k)} - \nabla F(x^{(k)}) \quad (2.11)$$

bestimmt.

Die Hesse-Matrix $\nabla^2 F(x^{(k)})$ muß in jedem Iterationsschritt ($k \in \mathbb{N}$) nicht singulär sein. Ist sie zusätzlich positiv definit, so ist die NEWTON-Methode ein Abstiegsverfahren.

Für eine quadratische Zielfunktion erreicht das NEWTON-Verfahren in nur einem Schritt die Lösung, und das läßt auch für nichtquadratische Zielfunktionen eine hohe Konvergenzgeschwindigkeit in der Nähe eines lokalen Minimums vermuten. Dieses bestätigt der folgende in [10, Kap. 2.6.1] bewiesene Satz :

Satz 2.7 : Ist die Zielfunktion $F: X \rightarrow \mathbb{R}$ mindestens dreimal stetig differenzierbar, und gibt es einen stationären Punkt $x^* \in X$, für den die Hesse-Matrix nicht singulär ist, dann existiert eine positive reelle Zahl ϵ , so daß das gewöhnliche NEWTON-Verfahren für $\|x^* - x^{(0)}\| < \epsilon$ durchführbar ist und die durch das Verfahren erzeugte Folge $(x^{(k)})_{k \in \mathbb{N}}$ mindestens quadratisch gegen x^* konvergiert.

Ein wesentliches Problem besteht also darin, einen geeigneten Startvektor zu suchen, für den das Verfahren konvergiert. Liegt der gewählte Startvektor zu weit von einer stationären Stelle x^* entfernt, so kann das Verfahren nach [9, Kap. 4.2.2.3] sogar instabil werden, d.h. die durch die Iterationsvorschrift erzeugte Folge kann sich noch weiter von x^* entfernen, da eine Approximation der Zielfunktion mit einer Taylorreihe bis zu dem Glied zweiter Ordnung hier nicht mehr genügend genau ist.

Ein weiterer Nachteil ist der hohe Rechenaufwand des gewöhnlichen NEWTON-Verfahrens durch die Bestimmung der zweiten Ableitungen und die Invertierung der Hesse-Matrix bzw. die Lösung des Gleichungssystems (2.11) in jedem Iterationsschritt.

Ist die Hesse-Matrix an der Stelle $x^{(0)}$ nicht singulär, so kann der Rechenaufwand durch die Iterationsvorschrift :

$$x^{(k+1)} = x^{(k)} - (\nabla^2 F(x^{(0)}))^{-1} \nabla F(x^{(k)}) \quad (2.12)$$

erheblich verringert werden, da die invertierte Hesse-Matrix nur einmal im ersten Iterationsschritt berechnet werden muß. Das hierdurch definierte *vereinfachte NEWTON-Verfahren* konvergiert für hinreichend gute Ausgangsnäherungen $x^{(0)}$ jedoch i.a. nur linear.

Weitere Modifikationen der NEWTON-Methode sind die *gedämpften NEWTON-Verfahren*, bei denen von den Hesse-Matrizen zusätzlich positive Definitheit gefordert wird und in der Iterationsvorschrift eine Schrittweite λ eingeführt wird, die nach dem Cauchy-Prinzip bestimmt wird.

Mit der Iterationsvorschrift :

$$x^{(k+1)} := x^{(k)} - (\nabla^2 F(x^{(0)}))^{-1} \nabla F(x^{(k)}) \quad , \lambda_k \text{ optimal} \quad , k \in \mathbb{N}$$

wird so das modifizierte vereinfachte NEWTON-Verfahren definiert, das wie die optimale Gradientenmethode linear konvergiert (vgl. Satz 2.4).

Eine entsprechende Modifikation des gewöhnlichen NEWTON-Verfahrens ist durch die Iterationsvorschrift :

$$x^{(k+1)} := x^{(k)} - (\nabla^2 F(x^{(k)}))^{-1} \nabla F(x^{(k)}) \quad , \lambda_k \text{ optimal} \quad , k \in \mathbb{N}$$

gegeben, dieses Verfahren konvergiert unter den Voraussetzungen :

- $\nabla^2 F(x^{(k)})$ nicht singulär und positiv definit ($k \in \mathbb{N}$),
- für den Startvektor $x^{(0)}$ gilt :

$$\{x \in X : F(x) < F(x^{(0)})\} \text{ ist beschränkt}$$

quadratisch gegen die unter diesen Bedingungen eindeutige Minimalstelle x^* der Zielfunktion F [10, Kap. 2.6.2].

2.5 Fehlernormen und Abbruchkriterien

Als Fehlernormen werden in dieser Arbeit die Normen $\|\dots\|_1$ und $\|\dots\|_2$ betrachtet. Werden die einzelnen Fehlerkomponenten gewichtet, so ergibt sich mit der euklidischen Norm als Zielfunktion die Summe der Fehlerquadrate :

$$F(x) = \|e(x)\|_2 = e(x)^T P e(x) = \sum_{i=1}^m p_i (y_i - f_i(x))^2 \quad (2.13)$$

mit einer Diagonalmatrix $P = \text{diag}(p_1, \dots, p_m)$, $p_i > 0$, $i \in \mathbb{N}$.

Ist die Zielfunktion mindestens einmal stetig differenzierbar, und sind ihre Gradienten bekannt, so kann als Abbruchkriterium die notwendige Bedingung zum Erreichen einer Minimalstelle x^* :

$$\|\nabla F(x^*)\| < \epsilon \quad (2.14)$$

für ein vorgewähltes $\epsilon > 0$ geprüft werden. Für die Zielfunktion aus (2.13) mit ihrem Gradienten :

$$\nabla F(x) = -2 \begin{pmatrix} \sum_{i=1}^m p_i (y_i - f_i(x)) \frac{\partial f_i}{\partial x_1}(x) \\ \vdots \\ \sum_{i=1}^m p_i (y_i - f_i(x)) \frac{\partial f_i}{\partial x_n}(x) \end{pmatrix} \quad (2.15)$$

ergibt sich mit der euklidischen Norm so ein Abbruchkriterium für den k -ten Iterationsschritt ($k \in \mathbb{N}$) :

$$\sqrt{\sum_{k=1}^n \left(-2 \sum_{i=1}^m p_i (y_i - f_i(x^{(k)})) \frac{\partial f_i}{\partial x_k}(x^{(k)}) \right)^2} < \epsilon \quad (2.16)$$

Für die Norm $\|\dots\|_1$ ergibt sich mit den positiven Wichtungsfaktoren $p_1, \dots, p_m$ als Zielfunktion die Summe der Fehlerbeträge :

$$F(x) = \sum_{i=1}^m p_i |y_i - f_i(x)| \quad (2.16)$$

Diese Funktion ist nicht in allen Punkten stetig differenzierbar, und daher kann nicht das Abbruchkriterium (2.14) gewählt werden.

Der (unstetige) Gradient der Funktion (2.17) wird wie folgt berechnet :

$$\nabla F(x) = - \begin{pmatrix} \sum_{i=1}^m p_i p_i \frac{\partial f_i}{\partial x_1}(x) \\ \vdots \\ \sum_{i=1}^m p_i p_i \frac{\partial f_i}{\partial x_n}(x) \end{pmatrix} \quad (2.18)$$

$$\text{mit } p_i = \begin{cases} 1, & \text{falls } |y_i - f_i(x)| \geq 0 \\ -1, & \text{falls } |y_i - f_i(x)| < 0 \end{cases} \quad (i=1, \dots, m) .$$

In diesem Fall und auch, wenn keine Gradienten zur Verfügung stehen, werden Kriterien, die nur Funktionswerte betrachten, benötigt. Solche Kriterien für den k -ten Iterationsschritt ($k \in \mathbb{N}$) können sein :

$$|F(x^{(k)}) - F(x^{(k+1)})| < \epsilon \quad (2.19) ,$$

$$|F(x^{(k-1)}) - F(x^{(k)})| < \epsilon \quad \text{und}$$

$$|F(x^{(k)}) - F(x^{(k+1)})| < \epsilon \quad (2.20) ,$$

sowie Kriterien, die die relative Änderung der Funktionswerte gegenüber der Änderung der Parameterwerte betrachten.

Eine weitere Möglichkeit ist in [10, Kap.2.5.3] beschrieben und basiert auf der Methode von AITKEN zur Konvergenzbeschleunigung.

Gilt für die durch das Iterationsverfahren erzeugte Folge der Funktionswerte $(F(x^{(k)}))_{(k \in \mathbb{N})}$

$$\lim_{k \rightarrow \infty} F(x^{(k)}) = F(x^*) \quad \text{und} \\ \lim_{k \rightarrow \infty} \frac{|F(x^{(k+1)}) - F(x^{(k)})|}{|F(x^{(k)}) - F(x^{(k-1)})|} = q, \quad 0 < q < 1$$

dann konvergiert die Folge $(F^*(x^{(k)}))_{(k \in \mathbb{N})}$ mit :

$$F^*(x^{(k)}) := \frac{(F(x^{(k)}))^2 - F(x^{(k-1)}) F(x^{(k+1)})}{2 F(x^{(k)}) - F(x^{(k-1)}) - F(x^{(k+1)})}$$

schneller als $(F(x^{(k)}))_{(k \in \mathbb{N})}$ gegen ihren Grenzwert $F(x_*)$.

Wegen

$|F(x^{(k)}) - F(x^*)| \leq |F(x^{(k)}) - F^*(x^{(k)})| + |F^*(x^{(k)}) - F(x^*)|$
ist daher ein sinnvolles Abbruchkriterium für den k-ten Iterationsschritt $(k \in \mathbb{N})$:

$$|F(x^{(k)}) - F^*(x^{(k)})| < \epsilon \quad (2.21)$$

Die Toleranzgrenze $\epsilon > 0$ sollte in jedem der Abbruchkriterien relativ zu der Größenordnung der geprüften Werte gewählt werden.

3. BESCHREIBUNG DES VERWENDETEN SIMULATIONSMODELLS

3.1 Allgemeines

Das in dieser Arbeit zugrunde gelegte Simulationsmodell ist ein *hydrologisches Niederschlag-Abfluß-Modell*. Es bildet die Abflußvorgänge auf der Oberfläche und im Kanalnetz eines Einzugsgebietes bei Niederschlagsereignissen nach.

Die kleinste ortsabhängige Berechnungseinheit eines Einzugsgebietes bildet hierfür eine *Haltung*, die Strecke zwischen zwei Schächten, mit einer zugehörigen Haltungsfäche. Aus Gebietskenngrößen und Kanalnetzdaten werden haltungsspezifische Parameter als Eingaben in das Modell abgeleitet.

Das Teilmodell HYSRAD berechnet ausgehend von gemessenen Niederschlagsdaten den *Oberflächenabfluß* des Einzugsgebietes und liefert Eingabedaten in Form von Zuflußganglinien in das Kanalnetz für das Modell KMROUT, das den *Abflußtransport* innerhalb des Netzes zu einem oder mehreren definierten Systemauslässen simuliert.

Die Beschreibung der beiden Teilmodelle beschränkt sich in dieser Arbeit neben allgemeinen Bemerkungen zur Methodik im wesentlichen auf die Simulationsgleichungen, die die zur Optimierung benutzten Systemparameter enthalten. Zu detaillierteren Beschreibungen und Herleitungen der Modellgleichungen sei auf die Literaturstellen [5], [6], [15] für HYSRAD und [19], [22], [23] für KMROUT hingewiesen.

3.2 Das Modell HYSRAD

HYSRAD ist eine Weiterentwicklung des Modells HYSTEM (*Hydrologisches Stadt-Entwässerungs-Modell*), das in [5] ausführlich beschrieben wird. Beide Modelle berechnen haltungs- und zeitschrittweise Oberflächenzuflüsse in ein

Kanalnetz. Als Berechnungszeitschritt Δt ist eine Minute fest vorgegeben.

Die Berechnung unterteilt sich in einen *Abflußbildungsprozeß*, in dem der in einem Zeitschritt zum Abfluß gelangende Niederschlag quantitativ bestimmt wird, und in einen *Abflußkonzentrationsprozeß*, in dem die zeitliche Verteilung der in das Kanalnetz eintretenden Oberflächenzuflüsse ermittelt wird. Dabei werden die zu den Haltungen gehörigen Flächen jeweils nach *durchlässigen* und *undurchlässigen Flächen* getrennt behandelt.

Die Abflußbildung für undurchlässige Flächen erfolgt nach der *Grenzwertmethode*, die der durchlässigen Flächen wahlweise nach dem *Infiltrationsmodell* von NEUMANN oder von HORTON.

Der funktionale Zusammenhang zwischen dem *abflußwirksamen Niederschlag* und dem direkten Oberflächenabfluß im Abflußkonzentrationsprozeß wird durch eine lineare und zeitinvariante Übertragungsfunktion, der Einheitsganglinie, die aus der *Standardeinheitsganglinie* ermittelt wird, bestimmt. Hierzu wird die den zeitlichen Ablauf des Abflusses auf der Oberfläche beschreibende *Schwerpunktlaufzeit* für die undurchlässigen Flächen aus Größe, Länge und Form der Flächen und für die durchlässigen Flächen aus Fließweg auf der Oberfläche, Gefälle, Rauigkeit und Regenintensität ermittelt.

In den hier vorgenommenen Untersuchungen werden nur undurchlässige Flächen betrachtet, d.h. es wird davon ausgegangen, daß vorhandene durchlässige Flächen der ausgewählten Einzugsgebiete keinen Abfluß liefern. Diese Einschränkung erfolgte, weil die physikalischen Vorgänge in durchlässigen Flächen sehr komplex und kompliziert sind, und eine Modellierung dieser einen zu großen Unsicherheitsfaktor bei gleichzeitiger Verdopplung der Parameteranzahl in der Optimierung bedeutet hätte. Aus diesem Grund wird auf die Methoden zur Berechnung von Abflüssen aus durchlässigen Flächen im weiteren nicht näher eingegangen.

Eine Erweiterung des Modells HYSRAD gegenüber HYSTEM ist die mögliche Simulation einer ungleichmäßigen Überregnung des Einzugsgebietes durch die Berücksichtigung mehrerer Regenschreiber. In dieser Arbeit wird jedoch ausschließlich mit gleichmäßiger Überregnung gearbeitet, da in den ausgewählten Gebieten jeweils nur eine Meßstation zur Aufnahme von Regendaten vorhanden ist.

Eine zweite Weiterentwicklung in dem Modell HYSRAD ist die Umstrukturierung der Zeitsteuerung zur On-line-Simulation, sowie die Berücksichtigung von Verdunstung und Muldenabtrocknung in Trockenzeiten zur Langzeitsimulation. Auf letztere Prozesse der Abflußbildung wird jedoch nicht näher eingegangen, da in allen Simulationen nur mit Einzelereignissen gerechnet wird.

Die letzte und auch für den Rahmen dieser Arbeit wesentliche Veränderung gegenüber dem Modell HYSTEM ist eine andere Implementation der Grenzwertmethode in der Abflußbildung für undurchlässige Flächen. Dieser im folgenden näher beschriebene Ansatz wurde von O. PAULSEN [15] entwickelt, in dem Modell KOSIM (kontinuierliches Niederschlag-Abfluß-Schmutzfracht-Modell) implementiert und nach eingehenden Untersuchungen als überlegen gegenüber dem alten Ansatz [5] eingestuft.

Die Parameter der Abflußbildung (für undurchlässige Flächen) sind :

- Benetzungsverlust V_{ben} [mm]
- Muldenverlust V_m [mm]
- Muldenauffüllgrad am Anfang des Ereignisses ϵ_0 [-]
- Abflußbeiwert am Anfang des Ereignisses A_a [-]
- Abflußbeiwert am Ende des Ereignisses A_e [-]

Der Benetzungsverlust repräsentiert die zur Benetzung der Flächen notwendige Niederschlagshöhe, die zuerst von den Niederschlagsdaten abgezogen wird, bevor es zum Abfluß auf der Oberfläche kommt. Der Abflußbeiwert A_a gibt den Anteil der nach Abdeckung des Benetzungsverlustes sofort zum

Abfluß führenden Fläche an und A_u den maximalen Anteil der Flächen, die insgesamt Abfluß liefern können. Wegen der gleichmäßigen Überregnung einer Haltungsfläche können die Parameter A_u und A_e auch als Anteile des zum Abfluß gelangenden Niederschlages interpretiert werden. Der Muldenverlust wird nach und nach abgedeckt, bis schließlich der durch die Größe des Flächenanteils A_e bedingte Grenzwert des abflußwirksamen Niederschlages erreicht ist. Die Simulationsgleichungen dieses Grenzwertansatzes für den i-ten Zeitschritt ($i \in \mathbb{N}$) lauten :

$$\epsilon_i = 1 - (1 - \epsilon_{i-1}) e^{-\frac{A_e - A_u}{V_{mr}} N_i} \quad (3.1)$$

$$R_i = N_i A_e - V_{mr} (\epsilon_i - \epsilon_{i-1}) \quad (3.2)$$

mit :

- N_i [mm] Niederschlagshöhe im i-ten Zeitintervall (nach Abzug der Benetzungsverluste)
- R_i [mm] abflußwirksame Niederschlagshöhe im i-ten Intervall
- ϵ_i [-] Muldenauffüllgrad im i-ten Intervall
- V_{mr} [mm] durch die Geländeneigung Q_g [%] reduzierter Muldenverlust :

$$V_{mr} := V_m e^{(-0.07 Q_g)} \quad (3.3) ,$$

wobei Q_g durch Angabe einer Neigungsklasse ($\in \{1, 2, 3, 4, 5\}$) der Haltungsfläche nach [5, Tab.1] bestimmt wird.

Ist der Muldenverlust einer Fläche sehr klein (< 0.01 mm), so wird ihr abflußwirksamer Niederschlag R_i in HYSRAD nach der Abflußprozentmethode berechnet :

$$R_i = N_i A_e \quad (3.4) .$$

Der einzige freie Parameter in der Abflußkonzentration für undurchlässige Flächen ist der Fließzeitparameter α_u [min], der in die Berechnung der Schwerpunktlaufzeit t_L [min] mit eingeht :

$$t_L = \alpha_u + 0.87 \ln A_{und} - 3 \frac{L}{l_f} \quad (3.5) .$$

Hierbei sind :

- A_{und} [ha] die Größe der undurchlässigen Fläche der Haltung
- L [m] die Haltungslänge
- l_f [m] der Fließweg auf der Oberfläche :

$$l_f = \sqrt{\left(\frac{L}{2}\right)^2 + \left(c \frac{b}{16}\right)^2}$$

mit :

- b [m] Breite der Haltungsfläche
- c [-] ($\in \{2, 3, 4, 5, 6, 8, 12\}$) Fließlängenkoeffizient zur Beschreibung der Schwerpunktlage in der Haltungsfläche s. [5, Bild 3.]

Die Formel (3.5) ist rein empirisch ermittelt und in dieser Form aus [4] entnommen. Für eine exakte Schreibweise müßten in den letzten beiden Termen noch einheitenbehaftete Konstanten eingeführt werden.

Die dimensionslose, idealisierte Standardeinheitganglinie $Q_{se}(t)$ ist eine Funktion über der Zeit mit dem Volumen 1. Sie setzt sich aus den beiden folgenden Teilen zusammen :

- linearer Anstieg über dem Zeitintervall $[0, t_{ps}]$ bis zum Scheitelpunkt Q_{ps} :

$$Q_{se}(t) = \frac{Q_{ps}}{t_{ps}} t , \quad t \in [0, t_{ps}]$$

- exponentieller Abfall mit der Speicherkonstante k_s über dem Intervall $[t_{ps}, t_{gs}]$:

$$Q_{se}(t) = Q_{ps} e^{-\frac{(t_i - t_{ps})}{k_s}}, \quad t \in [t_{ps}, t_{gs}] ,$$

wobei t_{gs} die Zeit ist, ab der $Q_{se}(t)$ unter ein Prozent des Scheitelwertes Q_{ps} sinkt :

$$t_{gs} = -k_s \ln 0.01 + t_{ps} .$$

Aus der Standardeinheitganglinie lassen sich mit Hilfe der Haltungsparameter A_{und} und t_L dimensionsechte, gebietsabhängige Einheitganglinien $Q_e(t)$ mit den zu $Q_{se}(t)$ korrespondierenden Parametern t_p [min], Q_p [m³/s], t_q [min] und k [min] konstruieren. Das Volumen unter der Funktion $Q_e(t)$ entspricht dem Volumen von $A := 1$ mm abflußwirksamer Niederschlagshöhe auf der Fläche A_{und} :

$$\int_0^{t_q} Q_e(t) dt = Q_p \left(\frac{t_p}{2} + 0.99 k \right) = \frac{A A_{und}}{0.006} \quad (3.6) .$$

(Der Faktor 0.006 entsteht durch Einheitentransformationen.)

Durch Erweiterung in Gleichung (3.6) mit t_L und dem Volumen von $Q_{se}(t)$ über $[0, t_{gs}]$ ($= 1$) lassen sich Beziehungen zwischen den Funktionsparametern der Standardeinheitganglinie und der Einheitganglinie aufstellen :

$$t_p = t_{ps} t_L, \quad Q_p = \frac{Q_{ps} A A_{und}}{0.006 t_L}, \quad k = k_s t_L, \quad t_q = t_{gs} t_L \quad (3.7) .$$

Die Parameter der Einheitganglinie müssen so bestimmt werden, daß die Volumenbeziehung (3.6) erfüllt wird und eine Korrespondenz zwischen der Schwerpunktlaufzeit t_L der Haltungsfläche und der Zeit t_q bis zum Schwerpunkt der Fläche unter der Kurve $Q_e(t)$ hergestellt wird :

$$t_L + \frac{1}{2} \Delta t = t_q \quad (3.8) .$$

Aus den Beziehungen (3.6) bis (3.8) läßt sich eine Prozedur zur Berechnung der Funktionsparameter Q_p , t_p , k und t_q der Einheitganglinie konstruieren. Sie ist in [6] beschrieben und in HYSRAD implementiert :

- (P1) Parameterstartwerte der Standardeinheitganglinie und der Einheitganglinie :

$$k_1 = k_{ps} t_L = 0.82 t_L \quad (3.9)$$

$$t_p := t_{p1} = t_{ps} t_L = 0.5 t_L \quad (3.10)$$

$$Q_{ps1} = 1 + \frac{1}{3 \left(1 + \frac{2k_1}{t_{p1}} \right)^2} - \frac{1}{\frac{t_{p1}}{\Delta t} \left(1 + \frac{2k_1}{t_{p1}} \right)} \quad (3.11)$$

(Die Formel (3.11) entsteht aus (3.7), (3.8) und der Berechnung der Schwerpunkzeit t_q .)

- (P2) Berechnung von Q_{p1} aus Q_{ps1} mit (3.7) und einem zweiten k -Wert mit Formel (3.6) :

$$Q_{p1} = \frac{Q_{ps1} A A_{und}}{0.006 t_L} \quad (3.12)$$

$$k_2 = \frac{1}{0.99} \left(\frac{A A_{und}}{0.006 Q_{p1}} - \frac{t_p}{2} \right) \quad (3.13)$$

- (P3) Runden von t_{p1} auf das nächste ganzzahlige Vielfache von Δt :

$$t_p := \left\lceil \frac{t_{p1}}{\Delta t} \right\rceil \Delta t \quad (3.14)$$

(Dieses gerundete t_p bleibt in weiteren Verlauf der Prozedur unverändert.)

- (P4) Berechnung von Q_{p2} aus t_p und k_2 mit (3.6) :

$$Q_{p2} = \frac{A A_{und}}{0.006 \left(\frac{t_p}{2} + 0.99 k_2 \right)} \quad (3.15)$$

- (P5) Berechnung eines mittleren Q_p 's aus Q_{p1} und Q_{p2} :

$$Q_p = \frac{Q_{p1} + Q_{p2}}{2} \quad (3.16)$$

- (P6) Berechnung von k aus Q_p und t_p mit (3.13)

- (P7) Berechnung von Q_{ps2} aus k und t_p mit (3.11)

(P8) Fallunterscheidung :

Gilt $|Q_{ps1} - Q_{ps2}| > 0.01$, dann weise Q_{ps1} den neuen Wert zu :

$$Q_{ps1} := \frac{Q_{ps1} + Q_{ps2}}{2} \quad (3.17)$$

und gehe zu (P2), sonst gehe zu (P9).

(P9) Diskrete Ordinaten der Einheitganglinie im Abstand Δt für den ansteigenden Ast :

$$Q_{e,i} := Q_e(t_i) = \frac{Q_p}{t_p} t_i \quad (3.18)$$

(mit $t_i = i\Delta t$ und $i=1, \dots, \frac{t_p}{\Delta t}$)

für den abfallenden Ast :

$$Q_{e,i}^* := Q_e^*(t_i) = Q_p e^{-\frac{(t_i - t_p)}{k}} \quad (3.19)$$

(mit $t_i = i\Delta t$, $i = \frac{t_p}{\Delta t} + 1, \dots, \frac{t_g}{\Delta t}$)

und $t_g = \left\lceil (-k \ln 0.01 + t_p) \frac{1}{\Delta t} \right\rceil \Delta t$

(P10) Volumenkorrektur für den abfallenden Ast :

$$\gamma = \frac{\frac{AA_{und}}{0.006} - \frac{Q_p}{2} (t_p + \Delta t)}{\sum_{i=\frac{t_p}{\Delta t}+1}^{\frac{t_g}{\Delta t}} Q_{e,i}^* \Delta t} \quad (3.20)$$

Mit dem Faktor γ [-] werden die Ordinaten des abfallenden Astes korrigiert zu :

$$Q_{e,i}^* := \gamma Q_{e,i}^* \quad (3.21)$$

(mit $t_i = i\Delta t$ und $i = \frac{t_p}{\Delta t} + 1, \dots, \frac{t_g}{\Delta t}$)

Das Ziel dieser Prozedur ist, die durch Rundung von t_p auf ein Vielfaches des Zeitschrittes Δt bedingte Verschiebung der Schwerpunktzeit t_q durch Veränderung der Parameter Q_p und k auszugleichen sowie die durch die Diskretisierung der Funktion $Q_e(t)$ erfolgte Volumenvergrößerung zu korrigieren.

Die Koordinaten der Zuflußganglinien zu den Schächten der Haltungen ergeben sich durch Multiplikation der Einheitganglinien mit den in der Abflußbildung berechneten abflußwirksamen Niederschlagshöhen einzelner Zeitschritte und Überlagerung der so entstandenen Ganglinien (*Superpositionsprinzip*) :

$$Q_{z,j} = \sum_{i=1}^{r_{anz}} R_i Q_{e,j-i+1} \quad (0 < j-i+1 < e_{anz}) \quad (3.22) ,$$

wobei r_{anz} die Anzahl der Minuten der Ereignisdauer und e_{anz} die Anzahl der Ordinaten der Einheitganglinie ist. Für die Zuflußganglinie ergeben sich somit $r_{anz} + e_{anz} - 1$ Ordinaten im Abstand Δt .

3.3 Das Modell KMROUT

Das hydrologische Modell KMROUT basiert auf dem *Kalinin-Miljukov-Verfahren* [19] zur näherungsweise Berechnung des Wellenablaufs in offenen Gerinnen und wurde zur Abflußtransportberechnung in Kanalnetzen modifiziert [22], [23].

Grundlage des Verfahrens ist die *Kontinuitätsgleichung* :

$$\text{Zufluß} - \text{Abfluß} = \text{Retention}$$

$$Q_z - Q_a = \frac{dV}{dt}$$

und die bei stationärer Strömung eindeutige Volumen-Abfluß-Beziehung :

$$dV = K dQ_a .$$

Hierbei kann K als die Zeit interpretiert werden, die das Volumen V eines betrachteten Gerinneabschnittes benötigt, um durch diesen Abschnitt zu fließen.

Die wesentliche Idee im Kalinin-Miljukov-Verfahren ist, daß auch bei instationären Strömungsverhältnissen für einen Gerinneabschnitt einer zu bestimmenden Länge L (*charakteristische Länge*) diese eindeutige Beziehung zwischen Volumen und Abfluß gilt.

Das Modell KMROUT berechnet haltungs- und zeitschrittweise Durchflüsse in einem Kanalnetz. Zuflüsse aus Oberflächenabfluß und externe Zuflüsse aus Haushalts- und Industrie-Einleitungen werden für die einzelnen Haltungen jeweils in die oberen Schächte gegeben. Alle Parameter können aus den Kanalnetzdaten ermittelt werden. Die charakteristische Länge wird in KMROUT vereinfacht gleich der Haltungslänge gesetzt

Die Arbeitsgleichung im i-ten Berechnungsschritt für eine Haltung lautet :

$$Q_{a,i+1} = (1 - c_1) Q_{a,i} + (c_1 - c_2) Q_{z,i} + c_2 Q_{z,i+1} \quad (3.23)$$

Hierbei sind :

- $Q_{a,j}$ [m^3/s] der Abfluß aus der Haltung zum Zeitpunkt j
- $Q_{z,j}$ [m^3/s] der Zufluß zur Haltung zum Zeitpunkt j ($j \in N$)
- c_1, c_2 [-] Kalinin-Miljukov-Konstanten mit :

$$c_1 = 1 - e^{-\frac{\Delta t}{K}} \quad , \quad c_2 = 1 - \frac{K}{\Delta t} c_1 \quad (3.24)$$

wobei K [s] die durchflußabhängige Retentionszeit der Haltung ist, die nach der Formel :

$$K = L \frac{dA}{dQ_a} \quad (3.25)$$

mit :

- L [m] Länge der Haltung
- Q_a [m^3/s] Abfluß der Haltung in einem Zeitschritt
- A [m^2] durchflossene Querschnittsfläche der Haltung beim Abfluß Q_a

berechnet wird.

Der Zufluß $Q_{z,j}$ ($j \in N$) zu einer Haltung umfaßt in Gleichung (3.23) sowohl den Zufluß von der Oberfläche (s. (3.22)) und externen Einleitungen als auch den Abfluß aus obenliegend anschließenden Haltungen.

Für den Abfluß aus einer oberhalb anschließenden Haltung gilt auch eine Gleichung der Form (3.23). So entsteht nach SEMKE [22] durch rekursives Einsetzen von (3.23) für verschiedene Haltungen als Zustandsbeschreibung im i-ten Berechnungsschritt ($i \in N$) für ein Kanalnetz mit m Haltungen ein Gleichungssystem in Matrixschreibweise :

$$Q^{(i+1)} = A Q^{(i)} + B u^{(i)} + C u^{(i+1)} \quad (3.26)$$

Hierbei sind :

- A, B, C (mxm)-Koeffizientenmatrizen von unterer Dreiecksform, die in Abhängigkeit von der Netzverknüpfung die Zuordnung der Kalinin-Miljukov-Konstanten, s. (3.24), zu den m Haltungen beschreiben.
- $Q^{(j)}$ der Abflußvektor mit m Komponenten zum j-ten Zeitpunkt
- $u^{(j)}$ der Zuflußvektor mit m Komponenten zum j-ten Zeitpunkt ($j \in N$)

Sind $A = (a_{l,k})$, $B = (b_{l,k})$, $C = (c_{l,k})$ ($l, k \in \{1, \dots, m\}$), $Q^{(j)} = (Q_1^{(j)}, \dots, Q_m^{(j)})^T$ und $u^{(j)} = (u_1^{(j)}, \dots, u_m^{(j)})^T$ ($j \in N$), dann ergibt sich für die (i+1)-te Zeitkomponente der Abflußganglinie der l-ten Haltung ($l \in \{1, \dots, m\}$) :

$$Q_l^{(i+1)} = \sum_{k=1}^m a_{l,k} Q_k^{(i)} + \sum_{k=1}^m b_{l,k} u_k^{(i)} + \sum_{k=1}^m c_{l,k} u_k^{(i+1)} \quad (3.27) ,$$

wobei die Koeffizienten $a_{l,k}$, $b_{l,k}$, $c_{l,k}$ einheitenfrei sind und die Durchflüsse $Q_l^{(j)}$ und $u_l^{(j)}$ ($l,k \in \{1, \dots, m\}$, $j \in \mathbb{N}$) in $[m^3/s]$ berechnet werden.

Die durchflußabhängige Retentionszeit K kann mit (3.25) in KMROUT bei verschiedenen Füllungszuständen nach einem Zeitschritt neu berechnet werden oder als mittlere Konstante über alle Füllungszustände zu Beginn bestimmt und dann unverändert gelassen werden. Abgesehen von dieser Entscheidungsmöglichkeit kann nur noch durch die Wahl des Berechnungszeitschrittes Δt vom Benutzer ein Einfluß auf die Simulationsergebnisse bewirkt werden. Eine Untersuchung zu der Arbeitsweise des Modells unter diesen Variationsmöglichkeiten wurde in [4] vorgenommen.

4. ANWENDUNG DES OPTIMIERUNGSVERFAHRENS AUF DAS SIMULATIONSMODELL

4.1 Die hydrologische Aufgabenstellung

Es werden die Ziele der Kalibrierung eines Niederschlag-Abfluß-Modells an dem in Kap. 3. beschriebenen Modell kurz erläutert.

Um das Simulationsmodell HYSRAD/KMROUT in einem Einzugsgebiet anwenden zu können, müssen die in Kap. 3. beschriebenen freien Parameter so bestimmt werden, daß das Modell das Abflußgeschehen bei unterschiedlichen Regenereignissen möglichst genau nachbildet. Hierfür werden gemessene Niederschlag-Abfluß-Ereignisse nachsimuliert und die Ergebnisse mit den Meßdaten verglichen.

Die Auswahl repräsentativer Ereignisse spielt dabei eine wichtige Rolle, denn nur, wenn für möglichst unterschiedliche Regenereignisse bei gleichen Parameterwerten gute Simulationsergebnisse erreicht werden, kann davon ausgegangen werden, daß mit den gefundenen Parametern das Abflußgeschehen auch für andere Ereignisse befriedigend simuliert werden kann.

Um dieses zu kontrollieren, erfolgt im Anschluß an eine Kalibrierung eine *Verifizierung*. Hierfür werden Simulationen durchgeführt mit den bei der Kalibrierung ermittelten Parametern, aber mit Niederschlagsereignissen, die von den für die Kalibrierung verwendeten unabhängig sind.

Wird für ein Einzugsgebiet bei unterschiedlichen Regenereignissen kein gemeinsamer Parametersatz gefunden, so ist das Simulationsmodell in diesem Fall nicht anwendbar. Ursachen hierfür können fehlerhafte Meßdaten, Fehler in der Erfassung der Einzugsgebiets- und Kanalnetzdaten oder aber eine Überschreitung der Anwendungsmöglichkeiten des Modells sein.

Scheitert die Parametersuche bei mehreren voneinander unabhängigen Einzugsgebieten, so muß davon ausgegangen werden, daß die Modellansätze nicht kalibrierfähig sind. Dieses ist jedoch bei dem Modell HYSRAD/KMROUT nicht zu erwarten, da zumindest alle Teilprozesse des Modells in unterschiedlichen Arbeiten auf ihre Anwendbarkeit geprüft wurden, s. z.B. [4], [26].

4.2 Verknüpfung der hydrologischen mit der mathematischen Aufgabenstellung

Die oben gestellte hydrologische Aufgabenstellung wird auf die in Kapitel 2.1 eingeführten Begriffe übertragen.

Als erste Übertragung bezeichnet das Modell f in dieser Arbeit das hydrologische Niederschlag-Abfluß-Modell HYSRAD/KMROUT und setzt sich aus allen in Kap. 3. beschriebenen Simulationsgleichungen zusammen. Die Funktionswerte $f(t)$ sind also Ordinaten einer Abflußganglinie zu der Zeit t , $t \in \mathbb{R}_{\geq 0}$.

4.2.1 Die Parameterwahl

Aufgrund der hydrologischen Aufgabenstellung ist eine Trennung der Parameter in gebiets- und ereignisspezifische notwendig. Nur gebietsspezifische Parameter können zur Kalibrierung des Modells für ein Einzugsgebiet herangezogen werden.

Aus diesen Überlegungen werden in der hier vorliegenden Arbeit als Parameter in der Optimierungsaufgabe aus dem Modell HYSRAD/KMROUT die Abflußbeiwerte A_s und A_e und der Muldenverlust V_m aus der Abflußbildung für undurchlässige Flächen sowie der Fließzeitparameter α_u aus der Abflußkonzentration der undurchlässigen Flächen gewählt. Die Dimension der Aufgabe (2.1) ergibt sich damit zu $n=4$ und der Parametervektor x zu $x = (x_1, \dots, x_4)^T = (A_s, V_m, A_e, \alpha_u)^T$.

Für diese Parameter sind in HYSRAD Standardwerte implementiert, die als mittlere Werte aus zahlreichen Untersuchungen bestimmt wurden [5]. Diese Voruntersuchungen werden für die Optimierungsaufgabe ausgenutzt, indem die Standardwerte als Startwerte für das Optimierungsverfahren gewählt werden. Für den Startvektor $x^{(0)}$ gilt damit:
 $x^{(0)} = (A_s^{(0)}, V_m^{(0)}, A_e^{(0)}, \alpha_u^{(0)})^T := (0.25, 1.8, 0.85, 11)^T$
 Hierbei und auch im weiteren Verlauf dieser Arbeit seien der Muldenverlust mit der Einheit mm und der Fließzeitparameter mit der Einheit min behaftet.

Aus den Definitionen der Parameter und den Herleitungen der Simulationsgleichungen aus Kap. 3.2 gilt für die Abflußbeiwerte die Beziehung:

$$0 \leq A_s < A_e \leq 1 \quad (4.1)$$

und für den Muldenverlust V_m und den Fließzeitparameter α_u :

$$V_m \geq 0, \quad \alpha_u \geq 0 \quad (4.2)$$

Dieses sind die Restriktionen der hier gestellten Optimierungsaufgabe.

Zusätzlich zu den unteren Schranken könnten in (4.2) auch obere Schranken angegeben werden, denn nach Erfahrungswerten treten in realen Einzugsgebieten selten Muldenverluste $V_m > 5$ und Fließzeitparameter $\alpha_u > 20$ auf [1]. Solche empirisch ermittelten Grenzen werden allerdings nicht in die Restriktionen mit aufgenommen, sondern nur bei den Voruntersuchungen (s. Kap. 4.3) ausgenutzt.

Die Parameter Benetzungsverlust V_{ben} und Muldenauffüllgrad am Anfang des Ereignisses ϵ_0 werden nicht mit optimiert, da sie nicht nur gebiets- sondern auch ereignisabhängig sind. Hatte ein simuliertes Niederschlagsereignis eine Vorgesichte, so ist es möglich, daß der Benetzungsverlust der Gebietsflächen zu Ereignisbeginn noch abgedeckt ist und die Mulden sogar noch teilgefüllt sind. Eine Optimierung dieser Parameter wäre so nur für ein spezielles Ereignis, und damit nicht für die Anwendbarkeit des Modells auf ein Einzugsgebiet, gültig.

Es ist aber möglich mit den Parametern V_{ben} und ϵ_0 eine sogenannte *Nachkalibrierung* durchzuführen, d.h. nach einer Optimierung der gebietsspezifischen Parameter kann versucht werden, durch Veränderung der Parameter V_{ben} und ϵ_0 für einzelne Niederschlagsereignisse, die Simulationsergebnisse zu diesen Ereignissen zu verbessern.

Eine Wiederholung des Optimierungsvorganges mit diesen angepaßten ereignisspezifischen Parametern kann so evtl. einen anderen Parametersatz $(A_a, V_n, A_s, \alpha_n)^T$ liefern, der das Gebiet dann besser beschreibt.

Für die beiden Parameter V_{ben} und ϵ_0 werden, falls nicht anders erwähnt, die in HYSRAD implementierten Standardwerte $V_{ben} = 0.7$ [mm] und $\epsilon_0 = 0$ beibehalten.

Wie schon in Kap. 3.2 erwähnt, werden für die hier durchgeführten Untersuchungen in HYSRAD keine durchlässigen Flächen simuliert. Dies kann durch Nullsetzen des Abflußbeiwertes für durchlässige Flächen für jede Haltungsfläche erreicht werden.

Liefern bei einem betrachteten Einzugsgebiet für ein Regenereignis die durchlässigen Flächen nicht zu vernachlässigenden Abfluß, so wird bei der Optimierung der vier Parameter der undurchlässigen Flächen dieser Abfluß nicht zu erfassen sein.

Aus diesem Grund wurden in der hier vorliegenden Arbeit Einzugsgebiete ausgewählt, die einen geringen Anteil an durchlässigen Flächen besitzen, und Niederschlagsereignisse, die einen vernachlässigbar kleinen Abfluß von durchlässigen Flächen liefern. Da die durchlässigen Flächen erst bei sehr intensiven Ereignissen und dann mit einer zeitlichen Verzögerung gegenüber den undurchlässigen Flächen Abfluß liefern, können solche Ereignisse leicht erkannt werden. Es ist jedoch darauf zu achten, daß das Niederschlag-Abfluß-Modell mit den für ein Gebiet optimierten Parametern dann auch nur für derartige Ereignisse angewendet werden kann.

In dem Abflußtransportmodell KMROUT wird die durchflußabhängige Retentionszeit K nach einer Empfehlung von DIEKMANN [4] variabel mit der feinsten Einteilung gewählt, d.h. von Trockenfall bis Vollfüllung eines Kanalrohres wird K insgesamt 50 mal nach (3.25) neu berechnet. Bei dieser Wahl wird ein Volumenfehler innerhalb von KMROUT in Kauf genommen, der allerdings in den meisten Fällen den vertretbaren Rahmen nicht überschreitet. Demgegenüber tritt bei der Simulation mit konstanter Retentionszeit kein Volumenfehler auf. Nach anderen Beurteilungskriterien (s. Kap. 4.3) liefert das Modell aber hier insbesondere im unteren Füllungsbereich schlechtere Ergebnisse als bei variabler Retentionszeit [4].

Der Berechnungszeitschritt Δt wird in dieser Arbeit für KMROUT wie in HYSRAD auf eine Minute gesetzt. Mit einem kleineren Zeitschritt kann in KMROUT eine Verringerung des Volumenfehlers bei variabler Retentionskonstante erreicht werden [4]. Dieses zieht dann allerdings auch eine Erhöhung des Rechenaufwandes nach sich.

Da in dem Modell KMROUT keine zu optimierenden Parameter enthalten sind, müssen durch das Modell erzeugte Fehler, wie z.B. der Volumenfehler bei variablem K , mit den Parametern aus HYSRAD ausgeglichen werden.

Ein optimaler Parametersatz, der für ein Gebiet mit HYSRAD und KMROUT bei variabler Retentionszeit gefunden wurde, kann also verschieden sein von einem optimalen Parametersatz, der für das gleiche Gebiet mit HYSRAD und KMROUT bei konstantem K oder mit HYSRAD und einem anderen Abflußtransportmodell gefunden würde. Aus diesem Grund muß das Modell bei der Optimierung immer als Gesamtmodell HYSRAD/KMROUT betrachtet werden.

4.2.2 Die Meß- und Simulationsdaten

Die Einzugsgebiete werden in dieser Arbeit im wesentlichen aus [12], Urban Drainage Catchments, gewählt. Dabei handelt es sich um eine Sammlung von 20 verschiedenen Kanalnetzen (UDM-Netzen), für die neben einer Beschreibung der Einzugsgebiets- und Kanalnetzdaten auch diskretisierte Niederschlagsereignisse mit Niederschlag-Abfluß-Ganglinien und einer Beschreibung der Meßwerterfassung vorliegen.

In allen hier betrachteten Gebieten liegt jeweils eine Meßstelle zur Erfassung der Niederschlagsdaten vor. Dies bedeutet, daß in dem Modell über dem ganzen Einzugsgebiet auch mit einer gleichmäßigen Überregnung simuliert wurde.

Die Abflußganglinien wurden in [12] jeweils an einem definierten freien Auslaß des Kanalnetzes in teilweise unterschiedlichen Zeitabständen (≥ 1 min) gemessen. Mit linearer Interpolation zwischen den gegebenen Abflußordinaten wurden hieraus in Minutenabständen diskretisierte Abflußganglinien erzeugt. Eventuell vorhandene Basisabflüsse der gemessenen Ganglinien wurden abgetrennt.

Der Abfluß in dem Transportmodell KMROUT wurde entsprechend nur mit aus den Niederschlagsdaten resultierenden Zuflüssen und nicht zusätzlich mit Zuflüssen aus externen Einleitungen berechnet. Als Ausgabe lieferte das Modell HYSRAD/KMROUT die für den freien Auslaß berechneten Abflußganglinien zu einem Niederschlagsereignis.

Damit sind die Meßdaten $y_1, \dots, y_n$ als Ordinaten der gemessenen Ganglinie zu den Zeitpunkten $t_1, \dots, t_n$ und die den Meßwerten zugeordneten Modellwerte $f(t_i; x)$ ($i=1, \dots, n$) als Ordinaten der simulierten Ganglinie gegeben.

Eine weitere Möglichkeit, Meßdaten zu erhalten, ist die in der Hydrologie verwandte Methode der *Autokalibrierung*. Hierbei wird ein Parametersatz x^* für ein Einzugsgebiet vorgegeben und mit verschiedenen Regenereignissen eine Simulation durchgeführt. Die Abflußganglinien am freien

Auslaß liefern so die "künstlichen Meßdaten" zu den Ereignissen. Für die Optimierung wird anschließend versucht, von einem Startvektor $x^{(0)} \neq x^*$ ausgehend, die Meßdaten nachzusimulieren und dabei den Ausgangsparametersatz x^* als optimalen Parametersatz zu erhalten.

Die Vorteile der Autokalibrierung für die hier behandelte Aufgabe sind :

- Unabhängigkeit von Meßfehlern
- die Eindeutigkeit der Simulationsansätze kann untersucht werden
(Liefern zwei verschiedene Parametersätze eine Lösung der Optimierungsaufgabe ?)
- gute Kontrollmöglichkeit über die Arbeitsweise des Optimierungsverfahrens

Bei der Autokalibrierung liegt mit dem vorgegebenen Parametersatz eine globale Lösung mit dem Wert 0 der Zielfunktion vor. Die Existenz solch einer Lösung ist bei realen Meßdaten hingegen nicht zu erwarten, der Funktionswert für die globale Lösung allerdings auch nicht bekannt.

Das Ziel des Optimierungsverfahrens, bei der Autokalibrierung die globale Lösung zu finden, sollte daher für die Anwendung nicht überbewertet werden.

In dieser Arbeit wird sowohl mit Autokalibrierung wie auch mit realen Meßdaten gearbeitet.

4.3 Ähnlichkeitsvergleich von Abflußganglinien

Neben den in 2.5 beschriebenen mathematischen Beurteilungskriterien stehen in der Hydrologie speziell entwickelte Prüfverfahren zum Vergleich von einer gemessenen und einer simulierten Abflußganglinie zur Verfügung [18].

Grundlage dieser Fehlerbetrachtung sind die Kriterien :

- Absoluter Größenvergleich
- Absoluter Zeitvergleich
- Absoluter Mengenvergleich
- Formvergleich

Damit werden die Ordinate der Ganglinienspitze, die Zeit bis zum Auftreten der Spitze, die Schwerpunktlaufzeit, das Abflußvolumen und die gesamte Form der Abflußganglinie berücksichtigt.

Da kein Prüfverfahren existiert, das alle diese Kriterien gleichermaßen erfassen kann, sollten möglichst alle im folgenden beschriebenen Prüfgrößen untersucht werden, um eine objektive Beurteilung der Ganglinienanpassung zu erhalten.

Die *hydrologische Deviation* *DEVS* [%] wurde von SCHULTZ [21] eingeführt. Sie untersucht die Summe der Fehlerbeträge ($\|\dots\|_1$), versehen mit von der Zeit abhängigen Wichtungsfaktoren. Dabei werden Abweichungen in der Nähe der Spitze stärker bewertet (Wichtungsfaktor 2) als Abweichungen in Basisnähe, und so wirken sich z.B. auch Zeitverschiebungen zwischen zwei Ganglinien gleicher Form negativ in der Beurteilung der Anpassungsgüte aus. Die Deviation *DEVS* liefert eine Fehlergröße als Prozentangabe, bezogen auf den maximalen Abflußwert der gemessenen Ganglinie :

$$DEVS = 200 \sum_{i=1}^m |y_i - f(t_i; x)| \frac{y_i}{(m y_{\max} y_{\max})} \quad (4.1)$$

mit den Bezeichnungen aus Kap 2.1 :

- y_i [m^3/s] Ordinate der gemessenen Ganglinie zum Zeitpunkt t_i
- $f(t_i; x)$ [m^3/s] Ordinate der simulierten Ganglinie zum Zeitpunkt t_i
- und :
- $y_{\max}$ [m^3/s] Maximalwert der gemessenen Ganglinie

RITSCHER [18] schlägt vor, die hydrologische Deviation *DEVS* auf den Mittelwert der gemessenen Ganglinie zu beziehen, und erhält so die Prüfgröße *DEVM* [%], die sich aus *DEVS* durch Multiplikation mit einem Faktor > 1 ergibt:

$$DEVM = DEVS \frac{y_{\max}}{y_{mit}} \quad \text{mit} \quad y_{mit} = \frac{1}{m} \sum_{i=1}^m y_i \quad (4.2)$$

Eine dritte Prüfgröße, die in der Hydrologie verwendet wird, ist die sog. *modifizierte Standardabweichung* *STAN* [-] von MANIAK [13]. Sie berechnet die Summe der Fehlerquadrate ($\|\dots\|_2$) im Verhältnis zu der Streuung der gemessenen Werte um ihren Mittelwert :

$$STAN = \frac{(R_0^2 - R^2)}{R_0^2} = 1 - \frac{R^2}{R_0^2} \quad (4.3)$$

$$\text{mit} \quad R_0^2 = \frac{1}{m} \sum_{i=1}^m (y_i - y_{mit})^2$$

$$\text{und} \quad R^2 = \frac{1}{m} \sum_{i=1}^m (y_i - f(t_i; x))^2$$

Fallen die gemessene und die simulierte Ganglinie zusammen, so wird $R^2 = 0$ und damit $STAN = 1$. Bei Abweichungen zwischen den beiden Ganglinien ergibt sich für *STAN* ein Wert kleiner als 1.

Die *Volumenbilanz* zweier Ganglinien *VOL* [%] errechnet sich zu :

$$VOL = 100 \frac{(V_y - V_f)}{V_y} \quad (4.4)$$

mit :

- V_y [m^3] Volumen der gemessenen Ganglinie
- V_f [m^3] Volumen der simulierten Ganglinie

Auf die gleiche Weise können auch die prozentualen Differenzen der Schwerpunktlaufzeiten und der Spitzenabflüsse der beiden Ganglinien bestimmt werden.

In Abhängigkeit von den unterschiedlichen Maßstäben der Prüfgrößen schlagen ZIOR [22] und RITSCHER [15] die folgenden Einteilungen als Beurteilung der Ganglinienanpassung vor :

Prüfgröße	Anpassung			
	sehr gut	gut	brauchbar	schlecht
DEVS [%]	0 - 3	3 - 10	10 - 18	> 18
DEVM [%]	0 - 15	15 - 30	30 - 50	> 50
STAN [-]	1 - .85	.85 - .65	.65 - .35	< .35
VOL [%]	0 - 5	5 - 10	10 - 15	> 15

Tabelle 4.1 : Bewertung der hydrologischen Prüfgrößen

Eine Anpassung einer simulierten Ganglinie an eine gemessene wird nach Tabelle 4.1 z.B. mit "sehr gut" bezeichnet, wenn alle vier Prüfgrößen Werte aus dem Bereich "sehr gut" liefern.

4.4 Sensitivität der Parameter

Zunächst wurde die funktionale Abhängigkeit in dem Modell HYSRAD/KMROUT von den vier ausgewählten Parametern A_s , V_s , A_e und α_e untersucht. Das Ziel dieser Sensitivitätsanalyse ist die Beantwortung der folgenden Fragen :

- Wie wirkt sich eine Veränderung der Parameter auf die Abflußganglinie aus ?
- Wie stark muß ein Parameter verändert werden, um eine Veränderung des Abflusses am freien Auslaß zu bewirken?
- Welche Grenzen ergeben sich dabei für die Parameter ?
- Wie hängt die Sensitivität der einzelnen Parameter von den Eingabedaten, der Regenbelastung, ab ?
- Gibt es Abhängigkeiten (Korrelationen) zwischen den Parametern ?

Diese Untersuchungen wurden an einem synthetischen Netz durchgeführt. Das Netz ist aus [4] entnommen und besteht aus 15 Haltungen von je 100 m Länge und je 0.5 ha Einzugsgebietsfläche (100% undurchlässige Fläche). Die Kanalrohre haben ein Kreisprofil von 0.5 m Durchmesser und sind mit einem Gefälle von 0.1 % nach dem in Bild 4.1 dargestellten Grundriß angeordnet.

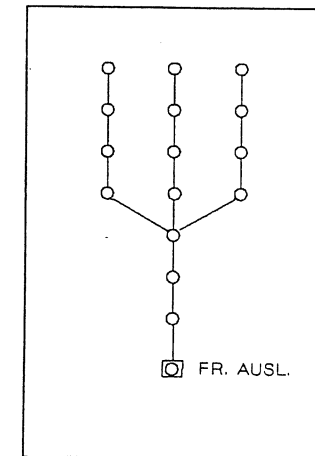


Bild 4.1 : Grundriß des synthetischen Netzes

Als Regenbelastung wurden drei verschiedenen starke Blockregen von je 30 Minuten Ereignisdauer gewählt, um die Abhängigkeiten von den Parametern möglichst gut erkennen zu können. Der schwache Regen hatte eine Intensität von 0.08 mm/min, der mittlere Regen von 0.16 mm/min und der starke Regen von 0.40 mm/min.

Für die Sensitivitätsuntersuchung wurde jeweils ein Parameter um seinen Standardwert in einem nach Erfahrungswerten ermittelten Bereich variiert, während die anderen drei Parameter auf ihren Standardwerten festgehalten wurden. Die Ergebnisse dieser unterschiedlichen Simulationen in graphischer Darstellungsweise befinden sich im Anhang A.3.

4.4.1 Der Parameter A_a

Mit dem schwachen Regen ergaben sich für den Abflußbeiwert A_a zu den Werten 0.05, 0.15, 0.25, 0.35 und 0.45 die in Bild 4.2 dargestellten Abflußganglinien.

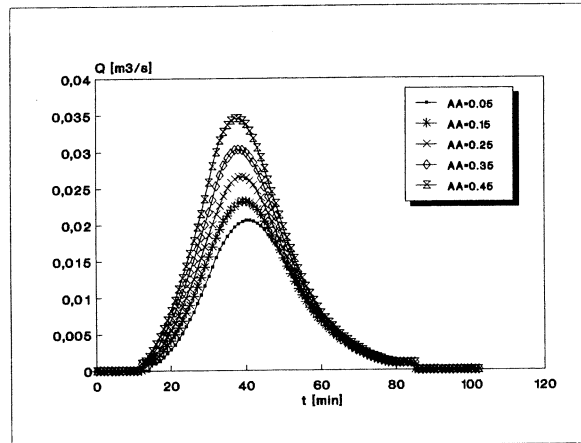


Bild 4.2: Abflußganglinien bei 0.08 mm/min Blockregen und verschiedenen A_a -Werten

Mit wachsendem A_a vergrößern sich auch das Volumen der Ganglinie und die Ordinate des Spitzenabflusses, während die Ordinaten in Basisnähe sowie die Schwerpunktlaufzeit dagegen nahezu unverändert bleiben.

Der Einfluß des Parameters A_a nimmt bei stärkeren Regenbelastungen jedoch ab (s. Anhang A.3). Nimmt das Volumen unter der Ganglinie von dem kleinsten untersuchten Anfangsabflußbeiwert (0.05) bis zu dem größten (0.45) bei dem schwachen Ereignis noch um 60 % zu (bezogen auf das Volumen für $A_a = 0.05$), so erreicht das Volumen bei einer Blockregenbelastung von 0.16 mm/min nur noch einen Zuwachs von 16 %, und bei dem starken Ereignis fallen die Abflußganglinien nahezu zusammen, der Volumenzuwachs beträgt hier nur noch ein Prozent.

Die auch im folgenden betrachteten prozentualen Volumendifferenzen beziehen sich, wenn es nicht zusätzlich angegeben ist, immer auf das aus dem kleineren Parameterwert berechnete Abflußvolumen.

In Bild 4.3 ist der in der Abflußbildung für die einzelnen Zeitschritte berechnete abflußwirksame Niederschlag für verschiedene Abflußbeiwerte A_a bei dem schwachen Ereignis dargestellt.

Der abflußwirksame Niederschlag wird üblicherweise in Niederschlagsblockdiagrammen dargestellt. Die hier gewählte Abbildung dient der besseren Unterscheidungsmöglichkeit für die einzelnen Parameterwerte.

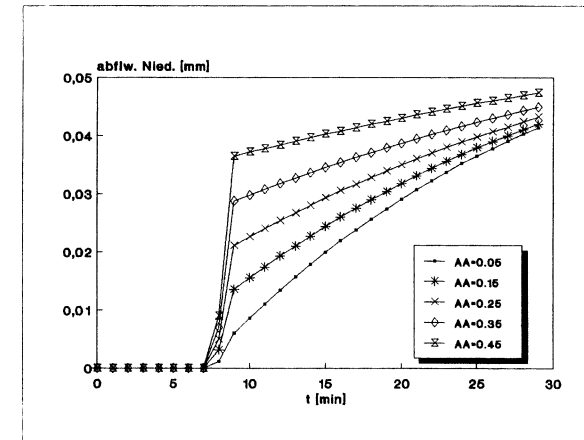


Bild 4.3 : Abflußwirksamer Niederschlag bei 0.08 mm/min Blockregen und verschiedenen A_a -Werten

Durch die geringe Intensität des Ereignisses füllen sich die Mulden erst langsam, und am Ende des Ereignisses ist der Grenzwert noch nicht erreicht. Der Parameter A_a beeinflusst den nach Abdeckung des Benetzungsverlustes sofort zum Abfluß gelangenden Niederschlag. Je höher der Wert von A_a ist und je langsamer sich die Mulden füllen, desto stärker wächst das gesamte Abflußvolumen. Dies erklärt auch den geringen Einfluß des Parameters A_a bei

stärkeren Ereignissen, denn hier füllen sich die Mulden schneller (s. Anhang A.3).

4.4.2 Der Parameter V_m

Für den Muldenverlust V_m wurden Simulationen zu den Werten 0.3, 0.8, 1.3, 1.8, 2.3, 2.8, 3.3 und 3.8 durchgeführt. Bild 4.4 enthält einige der Abflußganglinien zu den unterschiedlichen Parameterwerten bei dem starken Ereignis.

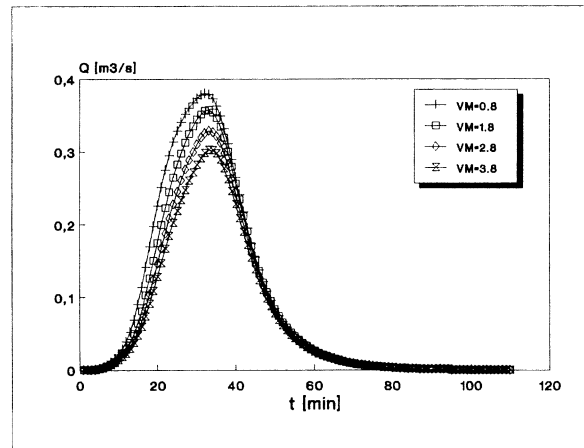


Bild 4.4 : Abflußganglinien bei 0.40 mm/min Blockregen und verschiedenen V_m -Werten

Auch dieser Parameter verliert bei zunehmender Regenintensität an Einfluß (s. Anhang A.3). Im Gegensatz zu dem Parameter A_m nimmt das Abflußvolumen jedoch für größer werdende Parameterwerte ab. Diese Volumenabnahme wird durch kleiner werdende Abflußwerte in der Nähe der Abflußspitze ausgelöst, während die Ordinaten in Basisnähe und die Schwerpunktlaufzeit wie bei der Veränderung des Abflußbeiwertes A_m nahezu unverändert bleiben.

Bei dem 0.08 mm/min Blockregen beträgt das Abflußvolumen für den größten Muldenverlust (3.8) 50 % des Volumens für den kleinsten Muldenverlust (0.3), bei dem 0.16 mm/min

intensiven Regen liegt die Volumendifferenz bei 46 % des aus 0.3 mm Muldenverlust berechneten Abflußvolumens und bei dem starken Ereignis nur noch bei 28 %.

Auffallend ist auch, daß die Volumendifferenzen für größer werdende Muldenverluste immer kleiner werden, so ergibt sich z.B. für den schwachen Regen zwischen den mit 3.3 und mit 3.8 Muldenverlust entstandenen Ganglinien eine Volumendifferenz von 3%, während der entsprechende Wert für 0.3 und 0.8 bei 23 % liegt. An Bild 4.5, in dem der abflußwirksame Niederschlag bei Blockregen mit 0.40 mm/min Intensität für die unterschiedlichen Muldenverluste aufgetragen ist, läßt sich erkennen, wie dieses zustande kommt.

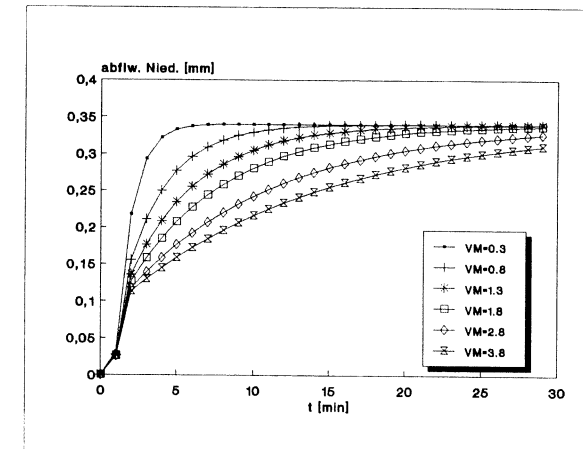


Bild 4.5 : abflußwirksamer Niederschlag bei 0.40 mm/min Blockregen und verschiedenen V_m -Werten

Die Parameter A_m und A_e als Anfangs- und Endabflußbeiwerte schränken den Bereich ein, zwischen dem die e-Funktion gespannt wird, die die Muldenauffüllung nachbildet. Je größer der Muldenverlust ist, desto flacher ist die e-Funktion. Der maximale Parameterwert von V_m , für den sich die Abflußganglinie noch verändert, hängt also von der Differenz $A_m - A_e$ (>0) ab.

Die Verringerung des Einflusses von V_a bei stärkeren Ereignissen hat die gleiche Ursache wie bei dem Parameter A_a . Das gleiche Phänomen würde auch bei den Ereignissen geringerer Intensität bei einer wesentlich längeren Ereignisdauer auftreten, da in diesem Fall der Grenzwert des abflußwirksamen Niederschlages erreicht wird, und je länger dieser noch beibehalten wird, desto geringer wirkt sich eine Volumendifferenz am Anfang des Ereignisses aus.

4.4.3 Der Parameter A_a

Für das mittlere Ereignis ergaben sich aus dem Endabflußbeiwert A_a für die Werte 0.55, 0.65, 0.75, 0.85 und 0.95 die in Bild 4.6 dargestellten Abflußganglinien:

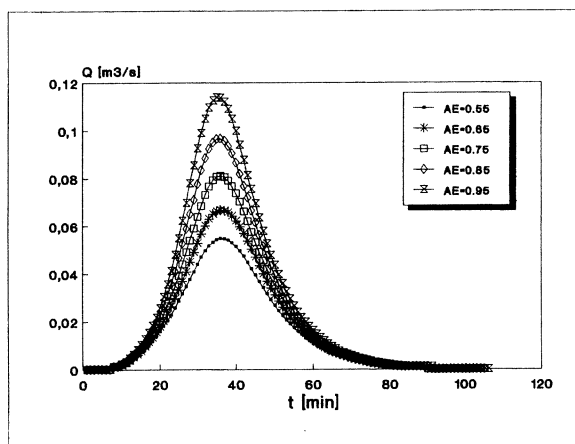


Bild 4.6 : Abflußganglinien bei 0.16 mm/min Blockregen und verschiedenen A_a -Werten

Das Abflußvolumen nimmt bei größer werdenden Endabflußbeiwerten zu. Bei einer Erhöhung des A_a -Wertes um 0.05 ergibt sich eine 7-8 %-ige Volumenzunahme. Wie bei den anderen beiden Parametern der Abflußbildung bleiben die Ordinaten in Basisnähe sowie die Schwerpunktlaufzeit nahezu unverändert, während sich die Ordinaten in der Nähe des Spitzenabflusses vergrößern.

Aus den Untersuchungen zu den Parametern A_a und V_a wurde schon deutlich, daß der Einfluß des Parameters A_a bei stärkeren Ereignissen größer wird. So ergibt sich für den schwachen Blockregen eine Volumendifferenz von 53 % zwischen einer Berechnung mit dem kleinsten und dem größten Abflußbeiwert, für den mittleren Regen 85 % und für das starke Ereignis 92 % (s. Anhang A.3). Die Betrachtung der berechneten abflußwirksamen Niederschläge, die in Bild 4.7 für den mittleren Regen dargestellt sind, verdeutlicht dies.

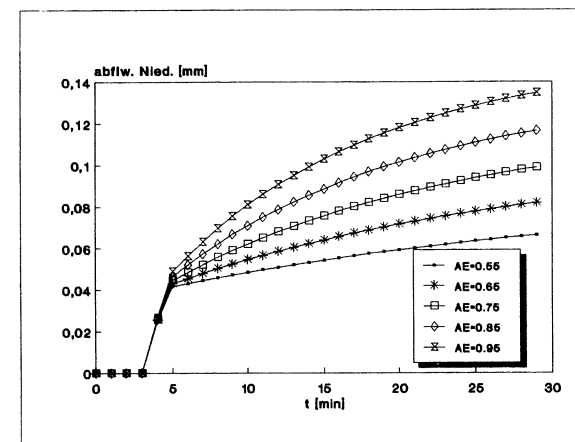


Bild 4.7 : Abflußwirksamer Niederschlag bei 0.16 mm/min Blockregen und verschiedenen A_a -Werten

Der Parameter A_a beeinflusst den Grenzwert, gegen den der abflußwirksame Niederschlag strebt. Eine Volumendifferenz tritt sofort nach dem ersten zum Abfluß gelangenden Niederschlag, der durch den Parameter A_a beeinflusst wurde, auf.

4.4.4 Der Parameter α_a

Es wurden Simulationen für Fließzeitparameter mit den Werten 4.0, 6.0, 8.0, 11.0, 14.0, 16.0, 18.0 und 20.0 durchgeführt.

Für den mittleren Blockregen sind einige der Abflußganglinien in Bild 4.8 dargestellt.

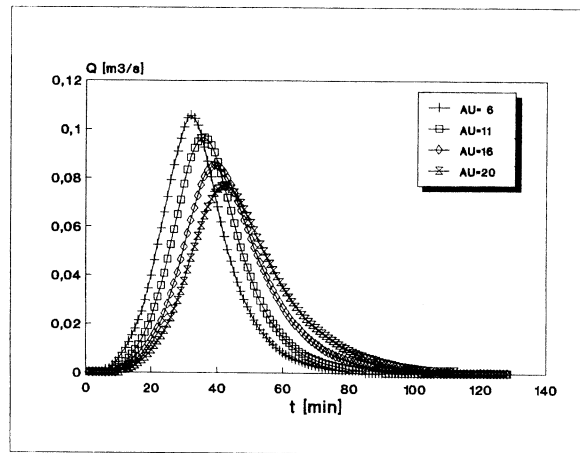


Bild 4.8 : Abflußganglinien bei 16 mm/min Blockregen und verschiedenen α_u -Werten

Wie aus der Herleitung des Abflußkonzentrationsprozesses zu erwarten war, wirkt sich eine Veränderung des Fließzeitparameters nicht auf das Abflußvolumen aus. Der Parameter beeinflusst die Schwerpunktlaufzeit der Ganglinien und damit auch die Ordinate der Abflußspitze. Größere Parameterwerte bewirken eine Vergrößerung der Schwerpunktlaufzeit und gleichzeitig eine Verkleinerung des maximalen Abflusses, die Ganglinie wird in der Zeitrichtung gestreckt.

Die unterschiedlichen Regenintensitäten haben so gut wie keinen Einfluß auf die Wirkung des Parameters (s. Anhang A.3). Ein Vergleich der Ganglinien zu den Fließzeitparametern von 6.0 und 20.0 liefert bei allen drei Ereignissen einen Anstieg der Schwerpunktlaufzeit um die 40 % und einen Abfall der maximalen Ordinate um die 25 %, bezogen auf die aus dem kleineren Parameterwert entstandene Abflußganglinie.

In Bild 4.9 sind die zu den unterschiedlichen Fließzeitparametern ermittelten Einheitganglinien dargestellt

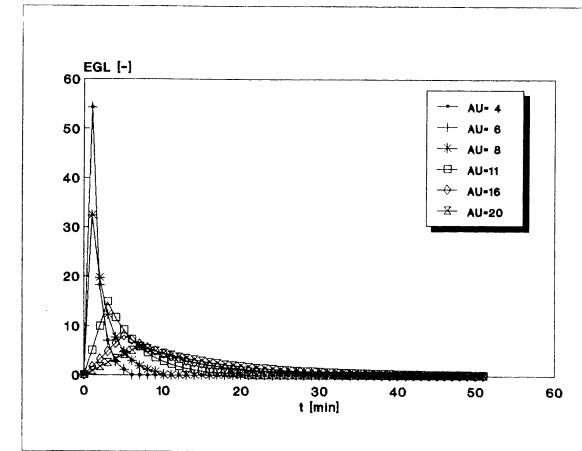


Bild 4.9 : Einheitganglinien bei verschiedenen α_u -Werten

Für Fließzeitparameter kleiner gleich 6.0 ergaben sich identische Einheitganglinien. Hierfür ist die in Kap. 3.2 beschriebene Prozedur zur Bestimmung der Einheitganglinienparameter verantwortlich, denn Laufzeiten t_L kleiner als zwei Minuten bewirken, daß die auf ein Vielfaches des Zeitschrittes γt gerundete Zeit t_p bis zur Spitze der Einheitganglinie jeweils den kleinstmöglichen Wert $t_p = \gamma t$ annimmt. Die untere Grenze des Parameters α_u ist von den Einzugsgebietsdaten abhängig, da aus α_u und den Gebietsdaten die Laufzeit t_L berechnet wird (s. (3.5)).

4.4.5 Zusammenfassung

Alle vier Parameter wirken sich in unterschiedlicher Weise auf die Abflußganglinie aus und sind so für die Optimierung von Bedeutung.

Den stärksten Einfluß haben die Parameter A_e und α_u . Während A_e das Abflußvolumen verändert, wird durch α_u die Abflußganglinie bei gleichem Volumen gestreckt oder

gestaucht. Die Parameter A_n und V_n wirken um so stärker, je kürzer und schwächer ein Ereignis ist.

Während A_n und A_n bei Vergrößerung auch das Abflußvolumen vergrößern, tritt bei V_n eine Verkleinerung des Volumens bei größer werdenden Parameterwerten auf.

Absolute Veränderungen der Abflußbeiwerte A_n und A_n um 0.05 liefern eine Auswirkung auf den Abfluß in etwa der gleichen Größenordnung wie absolute Veränderungen des Muldenverlustes V_n um 0.5 oder des Fließzeitparameters um 5.0 .

Die Parameter der Abflußbildung sind eng miteinander verknüpft, da die Abflußbeiwerte den Bereich aufspannen, in dem die Muldenauffüllung durch den Parameter V_n zum Tragen kommt.

Betrachtet man nur das Volumen, so ist der Parameter α_n unabhängig von den drei Parametern der Abflußbildung, da aber alle Parameter die Ordinate der Abflußspitze beeinflussen, besteht auch hier eine Korrelation.

4.5 Anwendung eines Optimierungsverfahrens auf das Modell HYSRAD/KMROUT

Es wurde ein FORTRAN-Programm OPTI erstellt, das eine Optimierung der vier ausgewählten Parameter aus dem Modell HYSRAD/KMROUT zu einem Einzugsgebiet durchführt. Als Eingabedaten werden hierfür eine Kanalnetzdatei mit den Einzugsgebiets- und Kanalnetzdaten sowie Regen- und Meßdateien mit Daten aus vorgegebenen Niederschlag-Abfluß-Ereignissen (gemessene oder selbst erzeugte) benötigt. Dabei ist es möglich, optimale Parameter zu mehreren Abflußereignissen gleichzeitig zu bestimmen. In den in dieser Arbeit durchgeführten Untersuchungen wurden jeweils Parameter zu drei Niederschlag-Abfluß-Ereignissen ermittelt.

Der Kern des Programmes OPTI ist ein mathematisches Optimierungsverfahren. Innerhalb dieses Iterationsverfahrens wird das Modell HYSRAD/KMROUT für die einzelnen Regenereignisse aufgerufen, um die Modellwerte zu den jeweils ermittelten Parametersätzen zu berechnen.

Als Optimierungsverfahren wurde in dieser Arbeit das optimale Gradientenverfahren aus Kapitel 2.4 gewählt. Dabei wird die langsame Konvergenz des Verfahrens in der Nähe einer Minimalstelle in Kauf genommen. Ausschlaggebend für die Wahl war dagegen das schnelle Fortschreiten zu Beginn der Iteration sowie die Unempfindlichkeit des Verfahrens bei ungenauer Schrittweitenbestimmung. Desweiteren ist der Rechenaufwand im Gegensatz zu den Verfahren der konjugierten Richtungen oder dem Newton-Verfahren bei dem optimalen Gradientenverfahren geringer und die Arbeitsweise des Verfahrens in der Anwendung durch den einfachen Algorithmus leicht zu kontrollieren.

Die wichtigste Voraussetzung, um das optimale Gradientenverfahren anwenden zu können, ist die Bestimmung der ersten Ableitungen der Zielfunktion in Abhängigkeit von den Parametern. Unabhängig von der Wahl der Fehlnorm in der Zielfunktion müssen für die Berechnung ihres Gradienten die ersten partiellen Ableitungen des Modells f nach seinen Parametern bestimmt werden.

Dies bedeutete für das hier betrachtete Modell HYSRAD/KMROUT, daß alle Simulationsgleichungen (3.1) bis (3.27) aus Kapitel 3. nach den vier Parametern A_n , V_n , A_n und α_n abgeleitet werden mußten. Der Gradient des Modells setzt sich dann aus den partiellen Ableitungen der mit (3.27) berechneten Abflüsse nach den vier Parametern zusammen, die durch mehrfache Anwendung der Kettenregel aus den partiellen Ableitungen zu den Simulationsgleichungen (3.1) bis (3.22) berechnet wurden.

Die partiellen Ableitungen der Gleichungen (3.1) bis (3.27) nach den Parametern A_n , V_n , A_n und α_n befinden sich in Anhang A.2.

Als Zielfunktion F wurde zunächst die Summe der Fehlerquadrate mit einer möglichen Wichtung der Fehlerkomponenten mit Hilfe einer positiv definiten Diagonalmatrix $P = \text{diag}(p_1, \dots, p_n)$ gewählt. Die Gradienten der Zielfunktion wurden so nach (2.14) in Kap. 2.5 berechnet.

In weitergehenden Untersuchungen wurde auch die hydrologische Deviation $DEVS$ (4.1) aus Kapitel 4.3 als Zielfunktion getestet. Sie geht auf die Summe der Fehlerbeträge (s. Kap. 2.5) zurück, und ihr Gradient in Abhängigkeit eines Parametervektors x ergibt sich aus (2.17) mit:

$$p_i = 200 \frac{y_i}{m y_{\max} y_{\min}}, \quad i=1, \dots, m$$

Zur Schrittweitenbestimmung wurde die in Kap. 2.3 beschriebene Prozedur zur Bestimmung eines ersten Unsicherheitsintervalls und das Bisektionsverfahren oder die kubische Interpolation für eine Verkleinerung des Unbestimmtheitsintervalls gewählt. Ein Abbruch der Schrittweitenbestimmung erfolgt wie in Kap. 2.3 beschrieben, nachdem ein zweites Mal eine ermittelte Schrittweite eine lokale Minimalstelle im Vergleich zu den zuvor bestimmten Schrittweiten bildet.

Für das Bisektionsverfahren und die kubische Interpolation werden die ersten Ableitungen der in Kap. 2.3 definierten Funktion $\phi(\lambda)$ in Abhängigkeit von der Schrittweite λ benötigt. Für die Simulationsgleichungen (3.1) bis (3.4) der Abflußbildung wurden die Ableitungen nach λ explizit bestimmt, um Rechenzeit zu sparen. Die entsprechenden Ableitungen befinden sich ebenfalls in Anhang A.2.

Um die Restriktionen (4.1) und (4.2) zu berücksichtigen, werden in dem Gradientenverfahren nur zulässige Richtungen erlaubt. In einem Iterationsschritt wird in der berechneten Richtung eine Schrittweite gesucht, für die sich ein zulässiger Punkt ergibt (vgl. Kap. 2.1).

Als Abbruchkriterium für die Zielfunktionen, die sich auf die euklidische Norm $\|\dots\|_2$ beziehen, wurde aus Kap. 2.5 die Formel (2.15) gewählt, die ein Verschwinden des Gradienten testet. Für Zielfunktionen, die auf der Fehler-norm $\|\dots\|_1$ beruhen, wurden die Differenzen aufeinanderfolgender Funktionswerte der Zielfunktion auf ein mögliches Verschwinden getestet (s. (2.19)). In jedem Fall ist auch ein Abbruch aufgrund einer Überschreitung der vom Benutzer angegebenen maximalen Anzahl der Iterationsschritte in dem Programm implementiert.

Damit sind alle Komponenten des Algorithmusses für das optimale Gradientenverfahren festgelegt (s. Anhang A.1.3). Zusätzlich eingeführte Modifikationen ergaben sich aus eingeführten Vereinfachungen, insbesondere um die Anzahl der Aufrufe des Modells HYSRAD/KMROUT zu verringern und dadurch Rechenzeit zu sparen, sowie aus Erweiterungen, die aufgrund der im Laufe der Untersuchungen auftretenden Probleme bei der Optimierung notwendig wurden.

Ist die Länge des ersten gefundenen Unsicherheitsintervalls durch die jeweilige Verdopplung sehr groß geworden, so kann nicht mehr zur Vereinfachung angenommen werden, daß die Zielfunktion über dem Intervall unimodal ist (s. Kap. 2.3), die Minimumsuche in diesem Intervall bestimmt eine von mehreren lokalen Minimalstellen aus diesem Intervall. In diesem Fall wird die Suche schon nach dem ersten gefundenen lokalen Minimum abgebrochen. Um die Zahl der Modellaufrufe weiter gering zu halten, sind in einem Intervallverkleinerungsprozeß mit dem Bisektionsverfahren oder der kubischen Interpolation nur eine begrenzte Anzahl von Intervallteilungen erlaubt.

Die vier Parameter können in HYSRAD nur in dem begrenzten Format von jeweils zwei Stellen nach dem Komma eingelesen werden. Da die Abflußbeiwerte in HYSRAD als Prozentwerte eingelesen werden, können die Parameter A_n und A_s mit vier und die Parameter V_n und α_n mit zwei gültigen Stellen nach dem Komma bestimmt werden.

Die durch das Iterationsverfahren berechneten Parameterwerte werden zunächst mit der computerabhängigen Genauigkeit bestimmt und anschließend auf das für HYSRAD benötigte Format gerundet. Dies führt dazu, daß Änderungen der Parameter um weniger als 10^{-4} bzw. 10^{-2} keine Auswirkungen mehr haben. Aus diesem Grund bricht das Iterationsverfahren neben den oben erwähnten Abbruchkriterien auch ab, wenn in mehreren aufeinanderfolgenden Iterationsschritten sehr kleine Schrittweiten berechnet wurden.

Ein Abbruch der Iteration kann auch erfolgen, wenn die durch die ermittelten Parameter berechneten Abflußganglinien im Vergleich zu den Meßdaten sehr gute Deviationen DEVS und DEVM sowie gute Volumenbilanzen VOL (nach Tabelle 4.1) liefern.

Die Komponenten des Richtungsvektors können mit konstanten Faktoren multipliziert werden, um die Größenordnung der Veränderung der einzelnen Parameter in den Iterationsschritten zu beeinflussen.

Durch Nullsetzen einzelner Komponenten kann erreicht werden, daß ein Parameter auf seinem zuletzt bestimmten Wert festgehalten und im weiteren Verlauf der Iteration nicht mehr optimiert wird. Dieses Festhalten erfolgt, falls ein Parameter in mehreren aufeinanderfolgenden Iterationsschritten nur mit sehr kleinen Schrittweiten einen zulässigen Punkt liefert oder wenn ein Parameter mit einer kleinen Frequenz um einen Wert oszilliert.

Werden Entscheidungen nach einer bestimmten Anzahl von Iterationsschritten getroffen, so kann diese Anzahl je nach Zielsetzung gewählt werden. Bei allen Entscheidungen, die in Abhängigkeit von der Größe der Schrittweiten, Richtungsvektoren oder Parameterwerte erfolgen, werden die Grenzen in Relation zu den unterschiedlichen Größenordnungen gewählt.

Im Anhang A.6 befinden sich eine Kurzbeschreibung und die Quelltexte des Programms OPTI.

5. ANWENDUNG DES OPTIMIERUNGSVERFAHRENS

5.1 Autokalibrierung in einem realen Netz

Ausgewählt wurde das UDM-Netz Vika NO01 in Oslo, Norwegen [12]. Es besitzt eine gesamte Einzugsgebietsfläche von 10.1 ha, davon sind 9.8 ha undurchlässige Flächen, was einem Versiegelungsgrad von 97 % entspricht. Das Kanalnetz ist in 25 Haltungen unterteilt mit den durchschnittlichen Kenngrößen :

- Haltungsfläche 0.38 ha, davon 0.36 ha undurchlässig
- Rohrdurchmesser 0.44 m (Kreisprofil)
- Haltungslänge 71.4 m
- Gefälle 2.6 %.

Der Grundriß des Kanalnetzes ist in Anhang A.4 dargestellt.

Es standen insgesamt sechs diskretisierte Niederschlagsereignisse zur Verfügung :

Bezeichnung	Regendauer	max. Regenintensität	Kategorie
Regen 9	140 min	0.09 mm/min	schwach
Regen 10	290 min	0.20 mm/min	mittel
Regen 11	280 min	0.72 mm/min	stark
Regen 12	275 min	0.17 mm/min	mittel
Regen 13	200 min	0.20 mm/min	mittel
Regen 14	160 min	0.04 mm/min	schwach

Tabelle 5.1: Niederschlagsereignisse zu dem UDM-Netz NO01

Zunächst wurden Simulationen der Ereignisse mit dem Modell HYSRAD/KMROUT zu den folgenden Parametern durchgeführt :

- Anfangsabflußbeiwert : $A_a = 0.10$
- Muldenverlust : $V_m = 2.5$
- Endabflußbeiwert : $A_e = 0.60$
- Fließzeitparameter : $\alpha_u = 5$

Die hieraus resultierenden Abflußganglinien werden im folgenden als Meßwerte interpretiert. Der Ausgangsparametersatz bildet also eine Lösung einer Optimierung des Einzugsgebietes mit den sechs Ereignissen mit dem Funktionswert 0 der Zielfunktion.

In Optimierungslauf 1 wurden optimale Parameter zu den Regen 12, 13 und 14 bestimmt. Als Zielfunktion wurde die Summe der Fehlerquadrate ohne Wichtung der Fehlerkomponenten gewählt. In der Schrittweitenbestimmung wurde das Bisektionsverfahren verwendet.

Das Programm berechnete nach 20 Iterationsschritten die Parameter :

$A_a = 0.1427$, $V_m = 1.8497$, $A_e = 0.5250$ und $\alpha_u = 11.0043$

In Anhang A.5.1-2 befinden sich Graphiken, die zu den drei Ereignissen die simulierten Ganglinien aus den im 5. und im 20. Iterationsschritt berechneten Parametern im Vergleich zu den Meßdaten darstellen.

In Tabelle 5.2 sind die am Ende der Iteration erreichten Werte für die hydrologische Deviation DEVS und die Volumendifferenz VOL zwischen gemessener und simulierter Ganglinie angegeben.

	Regen 12	Regen 13	Regen 14
DEVS (%)	5.69	4.20	6.27
VOL (%)	- 3.20	3.46	17.52

Tabelle 5.2 : Prüfgrößen zu Optimierungslauf 1

Die Deviationen zu den Ganglinienvergleichen liegen für alle drei Ereignisse nach Tab. 4.1 im guten Bereich, ebenso die Volumendifferenzen für die Regen 12 und 13. Der Regen 14 liefert hingegen eine schlechte Volumenbilanz.

Die Graphik in Bild 5.1 stellt die in den Iterationsschritten 0 bis 5 berechneten Parameter im Verhältnis zu den erwarteten Parametern (Ausgangsparameter) dar. Liefert die Kurve zu einem Parameter den Wert 1, so wurde der Wert des zu erwarteten Parameters erreicht.

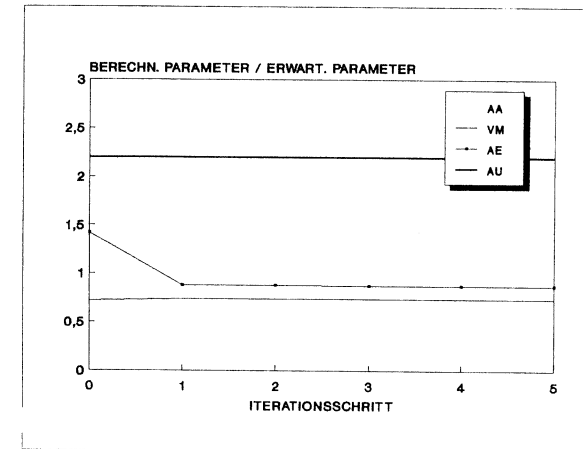


Bild 5.1: Veränderung der Parameter in Optimierungslauf 1

Nur im ersten Iterationsschritt wurden die Parameter wesentlich verändert. In allen weiteren Iterationsschritten bleibt die geringe Veränderung der Parameter ohne Einfluß auf die Simulationganglinien.

Betrachtet man den berechneten Richtungsvektor im ersten Iterationsschritt mit seinen Komponenten :

$r_1 = - 0.1033$, $r_2 = 0.0477$, $r_3 = - 0.3142$, $r_4 = 0.0048$, so fällt auf, daß bei einer gemeinsamen Schrittweite die Parameter V_m in der Richtung r_2 und α_u in der Richtung r_4 im Verhältnis zu den Parametern A_a und A_e viel zu gering verändert werden. Diese Unverhältnismäßigkeit wirkt sich in den weiteren Iterationsschritten in der Schrittweitenbe-

stimmung aus, denn nur für kleine Schrittweiten ist in den hier bestimmten Richtungen ein Abstieg der Zielfunktion zu erreichen. Durch die kleinen Schrittweiten verändern sich die Parameterwerte im Laufe der weiteren Iteration nicht mehr wesentlich.

In einem weiteren Rechenlauf, in dem statt dem Bisektionsverfahren die kubische Interpolation in der Schrittweitenbestimmung angewendet wurde, bricht das Iterationsverfahren sogar durch zu kleine Schrittweiten ab. Die Ergebnisse gleichen sonst denen aus Optimierungslauf 1.

Das in diesem Rechenlauf auftretende Verhalten war aufgrund der unterschiedlichen Größenordnungen der vier Parameter zu erwarten. In Kap. 2.4 wurde schon erwähnt, daß durch eine Skalierung der Parameter eine Konvergenzbeschleunigung erreicht werden kann.

Eine Skalierung kann in dieser Aufgabenstellung nur behelfsweise durch Multiplikation der Komponenten des Richtungsvektors mit unterschiedlichen Faktoren erfolgen. Da die Abflußbeiwerte in dem Intervall $[0;1]$ liegen, kann als Bezugswert der Wert 0.5 gewählt werden. Geht man bei dem Muldenverlust von einem mittleren Wert 1.8 und bei dem Fließzeitparameter von 11 (Standardwerte) aus, so ergibt sich aus diesen Überlegungen für die Richtungskomponente r_z der Faktor 4 und für r_r der Faktor 20.

Zieht man die aus der Sensitivitätsanalyse in Kap. 4.4 gewonnenen Erkenntnisse hinzu, so müßte eine Veränderung des Fließzeitparameters α_u um den Faktor 100 größer sein als eine der Parameter A_s und A_r , um eine Auswirkung auf die Abflußganglinie in der gleichen Größenordnung zu erhalten. Ein entsprechender Faktor ergibt sich für den Muldenverlust zu 10.

Beide Möglichkeiten wurden im Laufe der Untersuchungen verwendet.

In Optimierungslauf 2 wurde die gleiche Berechnung wie in Lauf 1 wiederholt. Dabei wurden die Richtungskomponenten für den Muldenverlust und dem Fließzeitparameter in jedem Iterationsschritt mit dem Faktor 10 bzw. 100 multipliziert.

Nach 20 Iterationsschritten wurden die folgenden Parameter berechnet :

$$A_s = 0.1102, V_m = 2.7196, A_r = 0.6121, \alpha_u = 5.3401$$

Bild 5.2 stellt wie Bild 5.1 die Parameterveränderungen dar. In Anhang A.5.3-4 befinden sich die Abflußganglinien zu diesem Optimierungslauf.

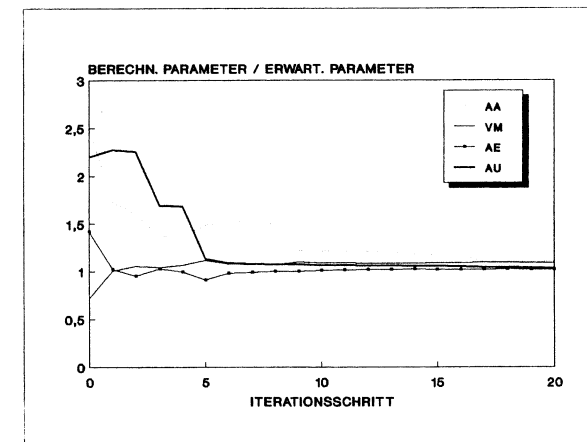


Bild 5.2: Veränderung der Parameter in Optimierungslauf 2

In Tabelle 5.3 sind die Deviationen DEVS und die Volumendifferenzen zu verschiedenen Iterationsschritten eingetragen. Ein Abbruch hätte hiernach schon nach dem 5. Iterationsschritt mit guten Ganglinienanpassungen für die drei Ereignisse oder nach dem 10. Iterationsschritt mit sehr guten Anpassungen stattfinden können.

Die Parameterwerte nach dem 5. Iterationsschritt waren :

$$A_s = 0.1484, V_m = 2.7929, A_r = 0.5479, \alpha_u = 5.6665,$$

und nach dem 10. Iterationsschritt :

$$A_s = 0.1213, V_m = 2.2743, A_r = 0.6067 \text{ und } \alpha_u = 5.3401.$$

Iterations- schritt	Regen 12	Regen 13	Regen 14
5	DEVS= 4.97% VOL = -11.29%	DEVS= 1.63% VOL = -5.18%	DEVS= 4.47% VOL = 13.10%
10	DEVS= 0.85% VOL = -0.64%	DEVS= 0.17% VOL = 1.35%	DEVS= 2.82% VOL = 7.99%
20	DEVS= 0.26% VOL = 0.23%	DEVS= 0.08% VOL = 0.28%	DEVS= 1.33% VOL = 3.67%

Tabelle 5.3 Prüfgrößen zu Optimierungslauf 2

In einer dritten Berechnung wurden die Regen 9, 10 und 11 zur Optimierung herangezogen, um zu untersuchen, wie unterschiedliche Eingabedaten auf das Optimierungsverfahren wirken. Die Optimierung erfolgte unter den gleichen Bedingungen wie bei Optimierungslauf 2. Der Unterschied bei dieser Kombination der Regenereignisse liegt darin, daß mit dem Regen 11 ein starkes Ereignis mit aufgenommen wurde.

Nach 40 Iterationsschritten wurden die Parameter :
 $A_s = 0.1842$, $V_s = 2.6306$, $A_u = 0.6079$ und $\alpha_u = 5.2345$
 berechnet. Bild 5.3 zeigt die Veränderungen der Parameter in den einzelnen Iterationsschritten, die Abflußganglinien befinden sich in Anhang A5.5-6.

Auffallend ist, daß die Parameter A_s und α_u sehr schnell zu dem erwarteten Wert gelangen und bis zum Ende der Iteration um diesen Wert oszillieren. Diese Oszillation in der Nähe der Lösung ist typisch für das Gradientenverfahren (s. Kap.2.4). Das Verhalten der Parameter V_s , A_u und α_u ähnelt sehr dem aus Bild 5.2. Nur der Parameter A_s weicht deutlich in seinem Verhalten ab. In Optimierungslauf 2 verändert sich der Parameter vom Beginn der Iteration an in der richtigen Richtung, in Lauf 3 hingegen geht er bis zum 4. Iterationsschritt in die falsche Richtung, um sich dann langsam, bedingt durch die mittlerweile kleinen Schrittweiten, an seinen erwarteten Wert anzunähern.

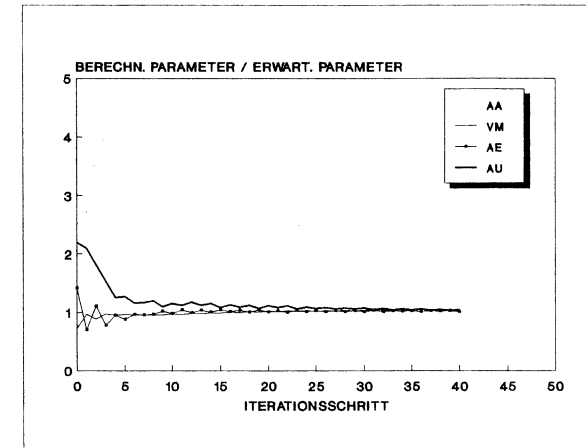


Bild 5.3: Veränderung der Parameter in Optimierungslauf 3

Verantwortlich hierfür ist einerseits das starke Regenereignis 11, zum anderen aber auch die Korrelation der Parametern untereinander.

Dieses wird in dem nächsten Rechenlauf deutlich. Hier wurden die Parameter A_s und α_u zu Beginn der Iteration auf ihren erwarteten Wert gesetzt und im Laufe der Iteration nicht mehr verändert. Für die Optimierung sind also nur noch zwei Parameter zu berechnen. Dieses entspricht für den gesamten Rechenlauf einem Festhalten der Parameter, falls sie in einem genügend kleinen Intervall um einen Wert oszillieren.

Schon im ersten Iterationsschritt werden die beiden Parameter von ihrem Standardwert als Startwert ausgehend auf die Werte $A_s = 0.1135$ und $V_s = 2.5997$ gesetzt, und im 5. Iterationsschritt liegen die Werte bereits bei $A_s = 0.0999$ und $V_s = 2.4989$.

Ein wesentlicher Punkt, um die Optimierung zu beschleunigen, ist also eine Reduzierung der zu optimierenden Parameter. Eine Möglichkeit besteht in diesem Fall darin, um einen Wert oszillierende Parameter für den Rest der Iteration festzuhalten.

Die Graphiken zu diesem Optimierungslauf befinden sich in Anhang A.5.7-8.

Für den nächsten Rechenlauf wurde eine weitere Kombination aus den sechs Niederschlagsereignissen getestet. Es wurden die beiden schwachen Ereignisse 9 und 14 und das mittlere Ereignis 13 ausgewählt. Diese Kombination erwies sich wie zu erwarten als nicht repräsentabel für die gesamten sechs Ereignisse.

In Bild 5.4 ist die Veränderung der Parameter dargestellt, die Abflußganglinien befinden sich in Anhang A.5.9-10.

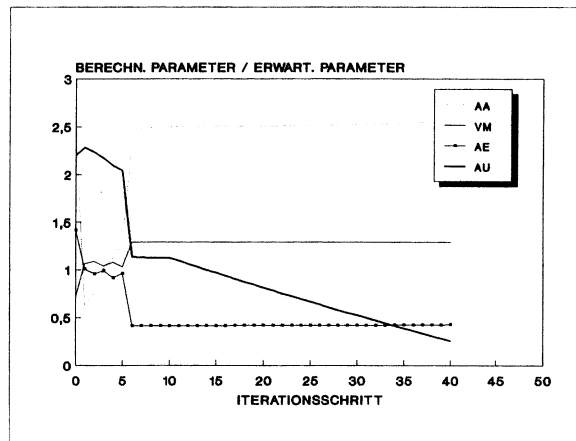


Bild 5.4: Veränderung der Parameter in Optimierungslauf 5

Interessant ist, daß die Parameter der Abflußbildung bis zu dem 5. Iterationsschritt den erwarteten Parametern sehr nahe gekommen sind, während der Fließzeitparameter sich kaum verändert hat. Die Werte in dem 5. Schritt waren:

$A_n = 0.1289$, $V_n = 2.5862$, $A_o = 0.5790$ und $\alpha_n = 10.2004$
In dem 6. Iterationsschritt wurde ein optimale Schrittweite von 60.75 berechnet, die dazu führte, daß der Parameter α_n in die Nähe seines erwarteten Wertes auf 5.6353 abfiel, während für die anderen Parameter die Werte $A_n = 0.2372$, $V_n = 3.2250$, $A_o = 0.2498$ eine deutliche Verschlechterung der Ganglinienanpassung lieferten.

Dies ist auch aus der Tabelle 5.4 zu entnehmen.

Iterations-schritt	Regen 9	Regen 13	Regen 14
5	DEVS= 4.49% VOL = 3.89%	DEVS= 5.37% VOL = -0.39%	DEVS= 5.67% VOL = 8.99%
10	DEVS= 2.73% VOL = 23.57%	DEVS= 3.19% VOL = -4.72%	DEVS=14.44% VOL =51.55%
40	DEVS= 3.52% VOL = 26.88%	DEVS= 3.38% VOL = -2.22%	DEVS=14.76% VOL =55.61%

Tabelle 5.4 : Prüfgrößen zu Optimierungslauf 5

Die Verbesserung des Fließzeitparameters nach dem 5. Iterationsschritt bewirkte für das mittlere Ereignis 13 und das schwache Ereignis 9 noch einen besseren Wert für die hydrologische Deviation DEVS, während sich bei dem schwächsten Ereignis 14 die deutliche Verschlechterung der Volumenbilanz auch auf die Deviation DEVS auswirkte. Im Verlauf der weiteren Iteration konnte nur das Volumen des mittleren Ereignisses leicht verbessert werden.

Das Verfahren wurde nach 40 Iterationsschritten aufgrund einer Überschreitung der maximalen Anzahl der Schritte ohne Lösung abgebrochen. In dem 6. Iterationsschritt wurde in der Zielfunktion ein Tal erreicht, aus dem das Optimierungsverfahren im Laufe der weiteren Iteration nicht mehr herausgekommen ist.

In dem letzten in diesem Kapitel vorgestellten Rechenlauf wurde wieder eine Optimierung für die Regen 9, 10 und 11 durchgeführt. Als Fehlerfunktion wurde die hydrologische Deviation DEVS gewählt. Die Fehlerkomponenten zu dem schwachen Ereignis 9 wurden in der Fehlerfunktion zusätzlich mit dem Faktor 10 gewichtet. Die Komponenten des Richtungsvektors wurden mit den Faktoren 4 in Richtung V_n und 20 in Richtung α_n multipliziert.

Die Iteration brach nach 35 Iterationsschritten nach den hydrologischen Beurteilungskriterien ab. Dabei wurden die in Tabelle 5.5 eingetragenen Werte erreicht :

Iterations-schritt	Regen 9	Regen 10	Regen 11
5	DEVS= 6.06% DEVM= 41.80% VOL = 56.03%	DEVS= 4.67% DEVM= 16.13% VOL = 6.04%	DEVS= 2.64% DEVM=39.84% VOL = 8.38%
20	DEVS= 4.96% DEVM= 34.24% VOL = 42.44%	DEVS= 3.78% DEVM= 13.07% VOL = 3.35	DEVS= 2.16% DEVM=32.62% VOL = 5.32%
35	DEVS= 1.81% DEVM= 12.51% VOL = -1.13%	DEVS= 2.65% DEVM= 9.15% VOL = 6.40%	DEVS= 0.56% DEVM= 8.47% VOL = 7.35%

Tabelle 5.5 : Prüfgrößen zu Optimierungslauf 6

Die Parameter nach dem 5. Iterationsschritt lauteten :

$A_a = 0.2535$, $V_m = 1.8890$, $A_o = 0.5610$, $\alpha_u = 10.7823$.

In dem 20. Iterationsschritt wurden die Parameter :

$A_a = 0.2174$, $V_m = 1.8783$, $A_o = 0.5788$, $\alpha_u = 9.4279$

berechnet und die von dem Programm gefundenen optimalen Parameter waren :

$A_a = 0.0476$, $V_m = 1.8783$, $A_o = 0.5953$, $\alpha_u = 6,7143$.

In diesem Parametersatz sind der Abflußbeiwert am Anfang des Ereignisses A_a und der Muldenverlust V_m zu klein gewählt gegenüber den vorgegebenen Parameterwerten. Nach Kap. 4.4 bedeutet dies für den Parameter A_a eine Verkleinerung des Abflußvolumens, für V_m dagegen eine Vergrößerung. Mit einer Verifizierung der Parameter durch die Niederschlagsereignisse 12, 13 und 14, wurde untersucht, ob der Parametersatz nach hydrologischen Kriterien für das Gebiet akzeptabel ist. Tabelle 5.6 enthält die entsprechenden Werte der hydrologischen Prüfgrößen :

	Regen 12	Regen 13	Regen 14
DEVS (%)	4.86	1.60	5.47
DEVM (%)	13.10	10.22	19.33
STAN (-)	0.95	0.97	0.93
VOL (%)	3.58	6.70	-10.92

Tabelle 5.6 : Prüfgrößen zu der Verifizierung von Lauf 6

Die Prüfgrößen liegen nach Tabelle 4.1 noch alle im guten Bereich, so daß der Parametersatz nach hydrologischen Kriterien für das Einzugsgebiet akzeptiert werden kann. Der Unterschied der von dem Optimierungsprogramm ermittelten Parameterwerte gegenüber den zu erwartenden Parameterwerten wirkt sich bei schwachen Ereignissen durch ein zu niedriges Abflußvolumen und bei stärkeren Ereignissen durch ein zu hohes Volumen aus.

Optimierungslauf 6 zeigt, daß ein durch das Programm berechneter optimaler Parametersatz nicht notwendig die globale Lösung des Optimierungsproblems sein muß, um nach hydrologischen Kriterien akzeptiert zu werden.

Die Graphiken mit den Abflußganglinien und der Veränderung der Parameter befinden sich in Anhang A.5.11.

5.2 Optimierung mit realen Netzen und Meßdaten

Es wurde das UDM-Netz Malvern CA01 in Burlington, Canada [11] ausgewählt, für das zu sechs diskretisierten Niederschlagsereignissen gemessene Abflußganglinien zur Verfügung standen. Das Gebiet Malvern hat eine Gesamtfläche von 23 ha, wovon 7.9 ha undurchlässige Flächen sind (34 % Versiegelung). Das Kanalnetz besteht aus 21 Haltungen mit den durchschnittlichen Kenngrößen :

- Haltungsfläche 1.11 ha, davon 0.38 durchlässige Fläche
- Rohrdurchmesser 0.54 m (Kreisprofil)
- Haltungslänge 88.6 m
- Gefälle 1.0 %

Der Grundriß des Kanalnetzes ist in Anhang A.4 dargestellt.

Die jeweilige Dauer der sechs Niederschlagsereignisse, sowie ihre maximale Regenintensität sind der folgenden Tabelle zu entnehmen :

Bezeichnung	Regendauer	max. Regenintensität	Kategorie
Regen 2	196 min	0.70 mm/min	stark
Regen 8	205 min	0.13 mm/min	schwach
Regen 16	81 min	1.02 mm/min	sehr stark
Regen 21	150 min	0.76 mm/min	stark
Regen 23	143 min	0.76 mm/min	stark
Regen 24	190 min	0.51 mm/min	mittel

Tabelle 5.7 : Niederschlagsereignisse zu dem UDM-Netz CA01

Die zu den Regenereignissen gemessenen Abflußganglinien lieferten die in Tabelle 5.8 angegebenen Volumina und Spitzenordinaten.

Regenbez.	Abflußvolumen	Abflußspitze	Abflußkoeffizient
Regen 2	736.98 m³	0.717 m³/s	34.55 %
Regen 8	517.95 m³	0.135 m³/s	29.14 %
Regen 16	451.11 m³	0.770 m³/s	33.10 %
Regen 21	1025.07 m³	0.428 m³/s	28.83 %
Regen 23	341.16 m³	0.210 m³/s	31.98 %
Regen 24	546.21 m³	0.249 m³/s	30.72 %

Tabelle 5.8 : Abflußereignisse zu dem UDM-Netz CA01

Der Abflußkoeffizient gibt an, wieviel Prozent von dem gefallenem Niederschlag insgesamt zum Abfluß gelangt ist. Den Abflußkoeffizienten ist zu entnehmen, daß die Abflußvolumina zu den sechs Ereignissen auch bei einer Simulation nur mit undurchlässigen Flächen erreicht werden können. Ein Vergleich der Abflußkoeffizienten mit dem Versiegelungsgrad des Einzugsgebietes von 34 % zeigt schon, daß über die undurchlässigen Flächen bei allen sechs Ereignissen bis zu 100 % des gefallenem Niederschlages auch zum Abfluß kommen.

In den durchgeführten Rechenläufen konnte dieses bestätigt werden.

Für Optimierungslauf 7 wurden die Regen 2, 8 und 24 ausgewählt. Es wurden die gleichen Faktoren zur Multiplikation der Richtungskomponenten gewählt wie in Rechenlauf 6. Als Zielfunktion wurde die Summe der Fehlerquadrate ohne Wichtung eingesetzt.

Die Benetzungsverluste wurden aus eigenen Handkalibrierungen übernommen, um möglichst von Vorereignissen unabhängige Meßdaten zu erhalten. Für Regen 2 ergab sich ein Benetzungsverlust $V_{ben} = 0.36$ mm, für Regen 8 wurde $V_{ben} = 0.60$ und für Regen 24 $V_{ben} = 0.50$ gesetzt.

In Bild 5.5 ist die Veränderung der Parameter dargestellt. Da die zu erwartenden Parameter nicht bekannt sind, kann eine Darstellung wie in Kap. 5.1 nicht erfolgen. In Bild 5.5 sind die absoluten Werte der Parameter eingetragen. Die Abflußbeiwerte wurden in der Graphik mit dem Faktor 10 multipliziert, um einen einheitlichen Maßstab zu erhalten.

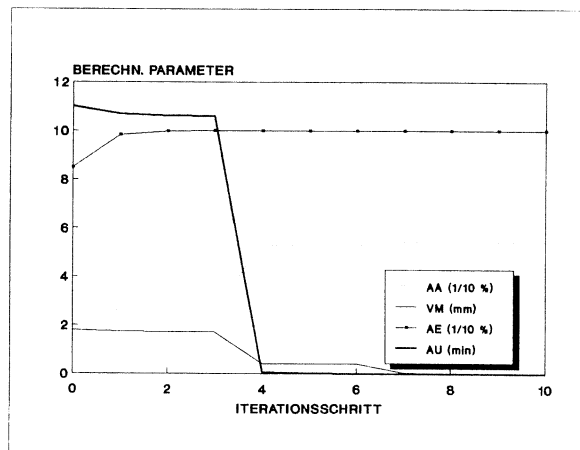


Bild 5.5: Veränderung der Parameter in Optimierungslauf 7

Schon im ersten Iterationsschritt mußten sehr kleine Schrittweiten gewählt werden, damit der Parameter A_a kleiner als 1 bleibt. Hierdurch wurden die anderen drei Parameter kaum variiert. Nach dem dritten Iterationsschritt wurde der Parameter A_a auf seinem Grenzwert 1 festgehalten. In dem vierten Schritt konnten so die drei anderen Parameter mit größeren Schrittweiten in ihren Richtungen verändert werden.

Der Parameter α_u wurde auf die gleich Weise wie A_a nach dem 6. Iterationsschritt festgehalten, jedoch auf seiner unteren Grenze 0. Auch in diesem Schritt ist zu erkennen, wie die zwei verbleibenden Parameter sich anschließend stärker verändern.

In dem 8. Iterationsschritt wurde schließlich der Parameter V_a auf seiner unteren Grenze 0 festgehalten, damit wird in HYSRAD die Abflußprozentmethode in der Abflußbildung verwendet (s. Kap 4.2, (3.4)), und die Parameter A_a und V_a wirken sich im weiteren Verlauf der Iteration nicht mehr auf die Abflußganglinien aus.

Das Verfahren wurde im 10. Iterationsschritt abgebrochen, da kein Parameter mehr zu optimieren war. Es wurden die Parameter :

$A_a = 0.5424$, $V_a = 0.0$, $A_u = 1.0$ und $\alpha_u = 0.0$ berechnet. Der Abflußbeiwert A_a hat hierbei aus oben genanntem Grund keinen Einfluß als Parameter.

Die Prüfgrößen zu diesem Parametersatz sind in Tabelle 5.9 dargestellt:

	Regen 2	Regen 8	Regen 24
DEVS (%)	.98	4.05	2.42
DEVN (%)	11.13	13.06	12.63
STAN (-)	0.97	0.95	0.91
VOL (%)	-12.42	6.99	1.23

Tabelle 5.9 : Prüfgrößen zu Optimierungslauf 7

Nach Tabelle 4.1 liegen alle Prüfgrößen in sehr guten bis guten Bereichen. Die Abflußganglinien hierzu befinden sich in Anhang A.5.13.

In dem letzten in dieser Arbeit beschriebenen Rechenlauf wurde eine Optimierung unter den gleichen Bedingungen wie in Nr. 7, jedoch mit den Niederschlagsereignissen 16, 21 und 23, und ihren Benetzungsverlusten 0.10 mm, 0.40 mm und 0.37 mm durchgeführt.

Das Verhalten der Parameter ist in Bild 5.6 dargestellt. Das Verfahren brach nach 10 Iterationsschritten aus den gleichen Gründen wie bei Optimierungslauf 7 ab.

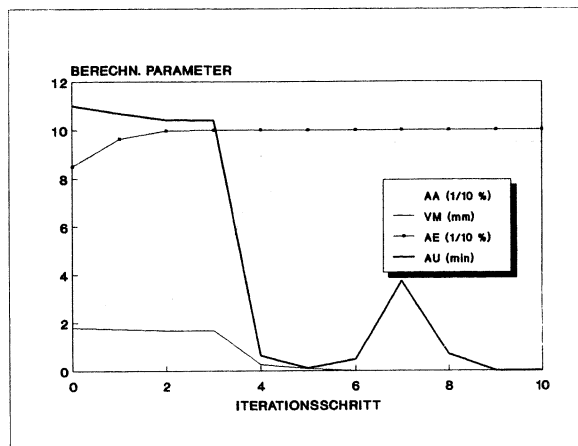


Bild 5.6: Veränderung der Parameter in Optimierungslauf 8

Der Parameter A_a wurde in Iterationsschritt 3 auf seine obere Grenze 1 festgesetzt. Als zweiter Parameter erreichte der Muldenverlust V_m im 6. Iterationsschritt seine untere Grenze 0, und dadurch wurde auch der Parameter A_a festgehalten. In der Abflußbildung wurde in den weiteren Iterationsschritten die Abflußprozentmethode angewendet. Der Fließzeitparameter α_a wurde als einziger Parameter nach dem 6. Iterationsschritt noch variiert. So erklärt sich die starke Veränderung des Parameters im 7. Iterationsschritt. Im 10. Iterationsschritt erreichte jedoch auch dieser Parameter seine untere Grenze, und das Verfahren stoppte.

Sieht man von dem Parameter A_a , der ohne Einfluß ist, ab, so wurden in beiden Optimierungsläufen die gleichen Parameter berechnet. Der Rechenlauf 8 ist also gleichzeitig eine Verifizierung des Optimierungslaufes 7 und umgekehrt.

6. ZUSAMMENFASSUNG UND AUSBLICKE

In dieser Arbeit wurde mit Hilfe eines mathematischen Optimierungsverfahrens der Kalibrierungsvorgang des hydrologischen Niederschlag-Abfluß-Modells HYSRAD/KMROUT automatisiert.

Aus den Verfahren der Optimierungstheorie wurde das optimale Gradientenverfahren ausgewählt.

Um dieses Gradientenverfahren auf das Niederschlag-Abfluß-Modell anwenden zu können, mußten zu allen Simulationsgleichungen des Modells die ersten Ableitungen nach ihren Parametern gebildet werden.

Mit Hilfe eines in dieser Arbeit erstellten Computerprogramms wurden sowohl Autokalibrierungen wie auch Kalibrierungen anhand von realen Meßdaten der gebiets-spezifischen Modellparameter für zwei Einzugsgebiete vorgenommen.

Insbesondere die Optimierung der Parameter zu realen Meßdaten konnte mit dem Programm erfolgreich durchgeführt werden. Bei den Rechenläufen zu der Autokalibrierung konnten Erkenntnisse sowohl aus der mathematischen Theorie als auch aus Voruntersuchungen zu dem hydrologischen Modell bestätigt werden.

Als wichtiger Faktor bei der Optimierung des Modells HYSRAD/KMROUT erwies sich die Auswahl repräsentativer Ereignisse für ein Gebiet. Bei der Autokalibrierung wurden für Optimierungen mit unterschiedlichen Kombinationen mehrerer Ereignisse teilweise auch unterschiedliche optimale Parameter berechnet.

Hier ist noch die Erfahrung des Anwenders erforderlich, eine geeignete Auswahl der Regenereignisse zu treffen. Wünschenswert wäre eine Unabhängigkeit von der Wahl der Ereignisse. Durch verschiedene Fehlerfunktionen und Wich-

tungen der Fehlerkomponenten konnte diese noch nicht erreicht werden.

Korrelationen unter den Parametern, sowie unterschiedliche Größenordnungen wirken sich negativ auf den Erfolg der Optimierung aus. Daher sollte die Anzahl der Parameter wenn möglich reduziert werden, und eine Skalierung der Parameter oder der Richtungen, in denen die Parameter verändert werden, vorgenommen werden.

Da in diesen beiden Punkten das Gradientenverfahren besonders empfindlich reagiert, könnte in weitergehenden Untersuchungen ein anderes Optimierungsverfahren angewendet werden, wie z.B. das FLETCHER-REEVES-Verfahren. Hier ist dann zu prüfen, in wie weit dieses Verfahren auf eine ungenaue Schrittweitenbestimmung reagiert.

Ein wesentlicher Punkt für die Anwendbarkeit eines Optimierungsverfahrens sind Voruntersuchungen über die Verhaltensweise der Parameter bei Variationen (Sensitivitätsanalyse). Nur mit diesen Vorkenntnissen kann die Arbeitsweise des Optimierungsverfahrens in der Testphase nachvollzogen und kontrolliert werden.

6. LITERATURVERZEICHNIS

- [1] Abwassertechnische Vereinigung e.V. (ATV)

Arbeitsbericht der ATV-Arbeitsgruppe 1.2.6:
"Hydrologie der Stadtentwässerung"
gemeinsam mit dem DVWK, Heft 2/86
- [2] BLUM, E. OETTLI, W.

Mathematische Optimierung
Ökonometrie und Unternehmensforschung XX
Springer-Verlag Berlin Heidelberg New York 1975
- [3] DENNIS JR., J.E. SCHNABEL, ROBERT B.

Numerical Methods for Unconstrained Optimization
and Nonlinear Equations
Prentice-Hall Series in Computational Mathematics
Clever Moler, Advisor 1983
- [4] DIEKMANN, R.

Anwendungsbereiche und Grenzen eines detaillierten
hydrologischen Kanalnetzbrechnungsmodells
Diplomarbeit am Institut für Wasserwirtschaft,
Hydrologie und landwirtschaftlichen Wasserbau
der Universität Hannover 1990, unveröffentlicht
- [5] FUCHS, L. VERWORN, H.-R.

Mikrocomputer in der Stadtentwässerung
Kanalnetzberechnung I
Kursunterlagen, 1988, (S.: 6-21)

- [6] HARMS, R.W.

Synthese maßgebender Abflußdaten für teilbebaute kleine Einzugsgebiete
Mitteilungen des Insituts für Wasserwirtschaft, Hydrologie und landwirtschaftlichen Wasserbau der Universität Hannover, Heft 55, 1984, (S.: 94-104)
- [7] HARTUNG PROF. DR.-ING., F.

Dynamische Optimierung der Fließbeiwerte in mathematischen Flußmodellen unter Berücksichtigung der Vorlandüberströmung
(von Dr.-Ing. Orhan Uslu)
Versuchsanstalt für Wasserbau der Technischen Universität München - Oskar v. Miller Institut - Bericht Nr. 30, 1974
- [8] HEUSER, HARRO

Lehrbuch der Analysis Teil 2
B.G. Teubner Stuttgart 1988
- [9] HOFFMANN, ULRICH HOFMANN, HANNS

Einführung in die Optimierung mit Anwendungs - beispielen aus dem Chemie-Ingenieur-Wesen
erweiterte Fassung eines Dechema-Kurses
Verlag Chemie GmbH 1971
- [10] HORST, REINER

Nichtlineare Optimierung
Carl Hanser Verlag München Wien 1979
- [11] LEXIKON DER OPTIMIERUNG

Optimierung und optimale Steuerung
Akademischer Verlag Berlin 1986

- [12] MAKSIMOVIC, C. RADOJKOVIC, M.

Urban Drainage Catchments
Pergamon Press, London, 1986
- [13] MANIAK, U.

Vergleich von Niederschlag-Abfluss-Modellen
5. Fortbildungslehrgang für Hydrologie des DVWK
Barsinghausen, 1973
- [14] PALMER, J.R.

An improved procedure for orthogonalizing the search vektors in Rosenbrock's and Swann's direct search method
Computer J.12 (1969), Seite 69-71
- [15] PAULSEN, O.

Kontinuierliche Simulation von Abflüssen und Stofffrachten in der Trennentwässerung
Mitteilungen des Instituts für Wasserwirtschaft, Hydrologie und landwirtschaftlichen Wasserbau der Universität Hannover, Heft 62, 1986, (S.:103-116)
- [16] PERESSINI, A.L. SULLIVAN, F.E. UHL JR., J.J.

The Mathematics of Nonlinear Programming
Springer-Verlag New York Berlin Heidelberg
London Paris Tokyo 1988
- [17] PIERRE, DONALD A.

Optimization Theory with Applications
Dover Publications, Inc., New York 1986

- [18] RITSCHER, UWE
- Ein mathematisch physikalisches Abflußmodell zur Berechnung von Hochwasserabflüssen und Untersuchung des Betriebes von Rückhaltebecken
Dissertation an der Fakultät für Bauwesen der TU Braunschweig, 1971
- [19] ROSEMANN, H.-J. VEDRAL, J.
- Das Kalinin-Miljukov-Verfahren zur Berechnung des Ablaufs von Hochwasserwellen
Schriftenreihe der Bayrischen Landesstelle für Gewässerkunde, Heft 6, 1970
- [20] SCHMIDT, RÜDIGER
- Advances in Nonlinear Parameter Optimization
Springer-Verlag Berlin Heidelberg New York 1982
- [21] SCHULTZ, GERT A.
- Bestimmung theoretischer Abflußganglinien durch elektronische Berechnung von Niederschlagskonzentration und Retention
Versuchsanstalt für Wasserbau der Technischen Hochschule München, Bericht Nr.11, München 1968
- [22] SEMKE, MARITA
- Mathematische Simulation einer zentralen Abflußregelung
Diplomarbeit am Institut für angewandte Mathematik und am Institut für Wasserwirtschaft, Hydrologie und landwirtschaftlichen Wasserbau der Universität Hannover 1984, unveröffentlicht
(Kap. 4.2.2.2 und Anhang A)

- [23] SIEKER, F. GROTEHUSMANN, D.
- Erarbeitung und Entwicklung eines kombinierten hydrologisch hydrodynamischen Modells zur Langzeitsimulation des Abflußprozesses in städtischen Kanalnetzen unter quantitativen und qualitativen Aspekten
Abschlußbericht zum Forschungsvorhaben I/62 917 der Stiftung Volkswagen, 1991, (Kap. 3.2 und 5.4)
- [24] SOSNOWSKI DIPL.-PHYS., PAUL
- Optimierung der Parameter hydrologischer Modelle
Zeitschrift Wasserwirtschaft-Wassertechnik
21. Jahrgang, Heft 8, 1971, S. 265-270
- [25] WISMER, DAVID A. CHATTERGY, R.
- Introduction to Nonlinear Optimization
A Problem Solving Approach
North-Holland New York 1978
- [26] ZINN, H.
- Entwicklung eines Expertensystems zur Kalibrierung eines Niederschlag-Abfluß-Modells
Diplomarbeit am Institut für Wasserwirtschaft, Hydrologie und landwirtschaftlichen Wasserbau der Universität Hannover 1990, unveröffentlicht.
- [27] ZIOR, F.
- Regenwasserabfluß auf Fahrbahnoberflächen - experimentelle und theoretische Untersuchungen -
Technische Berichte über Ingenieurhydrologie und Hydraulik, Institut für Wasserbau der Technischen Hochschule Darmstadt, Heft 38, 1987

Bildverzeichnis

<u>Bild</u>	<u>Seite</u>
Bild 2.1: Zick-Zack-verhalten des optimalen Gradientenverfahrens bei der Minimierung einer zweidimensionalen Funktion F in Höhenliniendarstellung	20
Bild 4.1: Grundriß des synthetischen Netzes	52
Bild 4.2: Abflußganglinien bei 0.08 mm/min Blockregen und verschiedenen A_n -Werten	53
Bild 4.3: Abflußwirksamer Niederschlag bei 0.08 mm/min Blockregen und verschiedenen A_n -Werten	54
Bild 4.4: Abflußganglinien bei 0.40 mm/min Blockregen und verschiedenen V_n -Werten	55
Bild 4.5: Abflußwirksamer Niederschlag bei 0.40 mm/min Blockregen und verschiedenen V_n -Werten	56
Bild 4.6: Abflußganglinien bei 0.16 mm/min Blockregen und verschiedenen A_n -Werten	57
Bild 4.7: Abflußwirksamer Niederschlag bei 0.16 mm/min Blockregen und verschiedenen A_n -Werten	58
Bild 4.8: Abflußganglinien bei 16 mm/min Blockregen und verschiedenen α_n -Werten	59
Bild 4.9: Einheitganglinien bei verschiedenen α_n -Werten	60

Bild 5.1: Veränderung der Parameter in Optimierungslauf 1	68
Bild 5.2: Veränderung der Parameter in Optimierungslauf 2	70
Bild 5.3: Veränderung der Parameter in Optimierungslauf 3	72
Bild 5.4: Veränderung der Parameter in Optimierungslauf 5	73
Bild 5.5: Veränderung der Parameter in Optimierungslauf 7	79
Bild 5.6: Veränderung der Parameter in Optimierungslauf 8	81

Tabellenverzeichnis

<u>Tabelle</u>	<u>Seite</u>
Tabelle 4.1: Bewertung der hydrologischen Prüfgrößen	51
Tabelle 5.1: Niederschlagsereignisse zu dem UDM-Netz NO01	66
Tabelle 5.2: Prüfgrößen zu Optimierungslauf 1	67
Tabelle 5.3: Prüfgrößen zu Optimierungslauf 2	71
Tabelle 5.4: Prüfgrößen zu Optimierungslauf 5	74
Tabelle 5.5: Prüfgrößen zu Optimierungslauf 6	75
Tabelle 5.6: Prüfgrößen zu der Verifizierung von Lauf 6	76
Tabelle 5.7: Niederschlagsereignisse zu dem UDM-Netz CA01	77
Tabelle 5.8: Abflußereignisse zu dem UDM-Netz CA01	78
Tabelle 5.9: Prüfgrößen zu Optimierungslauf 7	80

Korrekturen:

Seite 37: Formel (3.18) muß lauten :

$$Q_{e,i}^* := Q_e^*(t_i) = \frac{Q_p}{t_p} t_i \quad .$$

Formel (3.21) muß lauten :

$$Q_{e,i} := \gamma Q_{e,i}^* \quad .$$

Seite 60: Bild 4.9 : Achsenbezeichnung :

'Q_e [m³/s]' statt 'EGL [-]'.

Anhang: A.3 : Sensitivität des Parameters V_a mit

0.08 mm/min Blockregen :

Maßstab der y-Achse (abflw. Nied.) :

0.01 bis 0.07 statt 100 bis 700 .

Sensitivität des Parameters A_a mit

0.08 mm/min Blockregen :

Achsenbezeichnung :

'abflw. Nied. [mm]' statt

'abflw. Nied. [m³/s]'.

Sensitivität des Parameters α_a :

Einheitganglinien :

Achsenbezeichnung :

'Q_e [m³/s]' statt 'EGL [-]'.

A.5.1-2 : Die berechneten Ganglinien aus dem 20.

Iterationsschritt fehlen. Sie sind nahezu

identisch zu den Ganglinien aus dem 5.

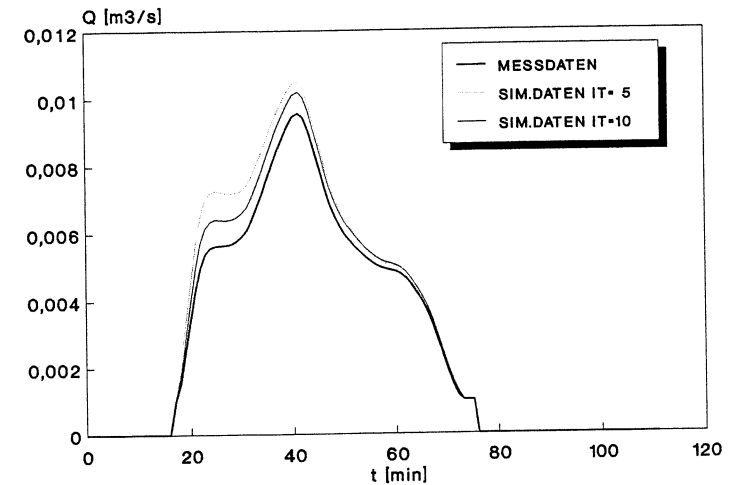
Iterationsschritt.

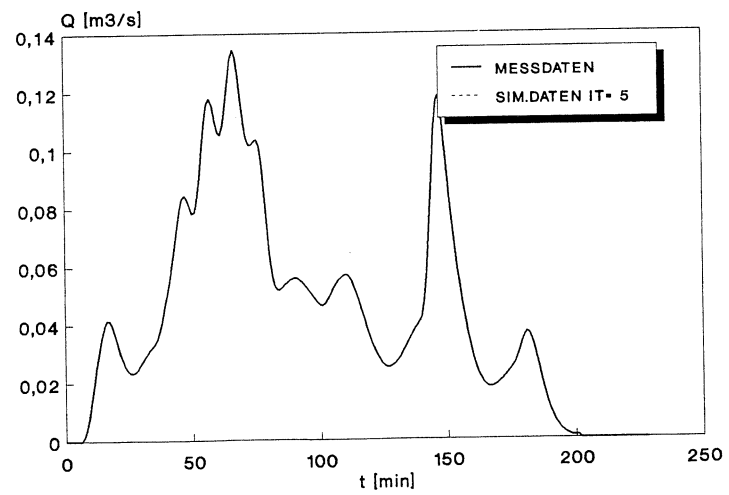
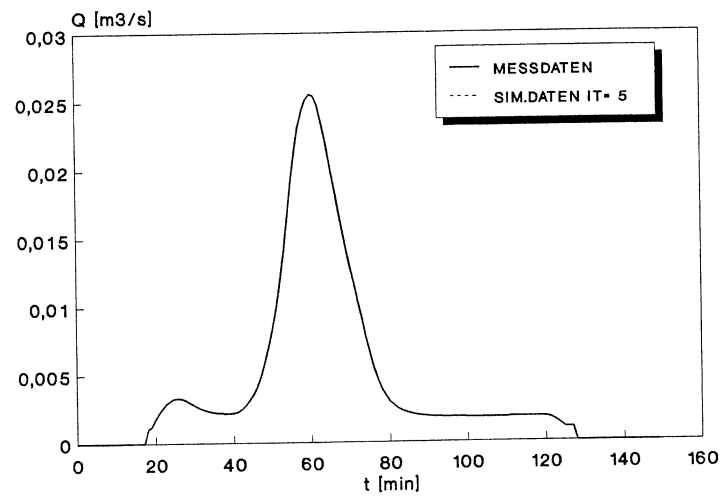
A.5.4 : Die obere Abbildung zeigt Regen 13 und nicht

Regen 14. Die Abbildung zu Regen 14 liegt

anbei.

A.5.7 : Die Seite fehlt. Sie liegt anbei.





A.5.7 Optimierungslauf 4 : Regen 9 und 10

Institut für
Angewandte Mathematik

Institut für Wasserwirtschaft,
Hydrologie und landwirtschaft-
lichen Wasserbau

Universität Hannover

**Parameteridentifikation
für
Simulationsmodelle
in der
Hydrologie**

Anhang

Diplomarbeit

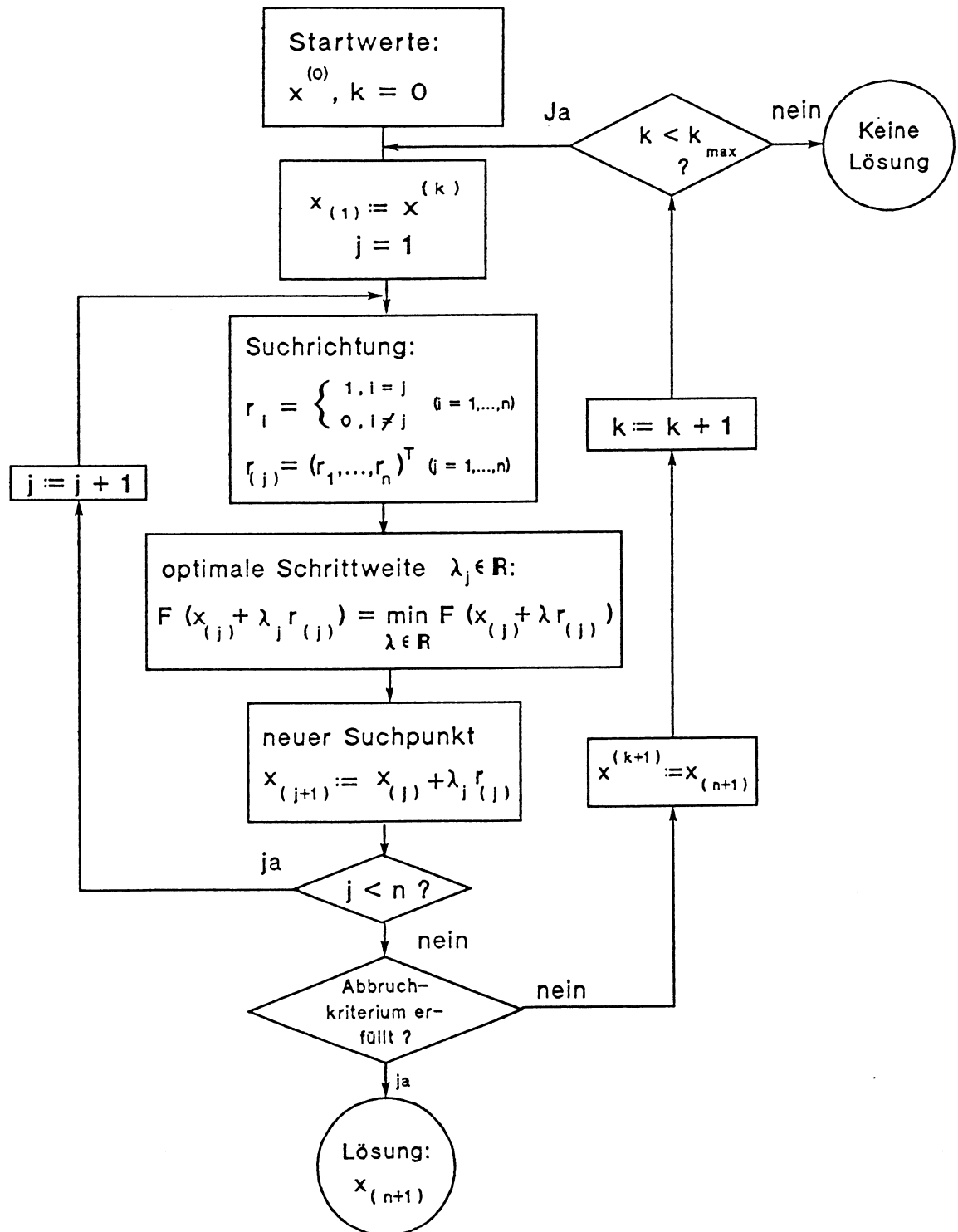
von

Susanne Deyda

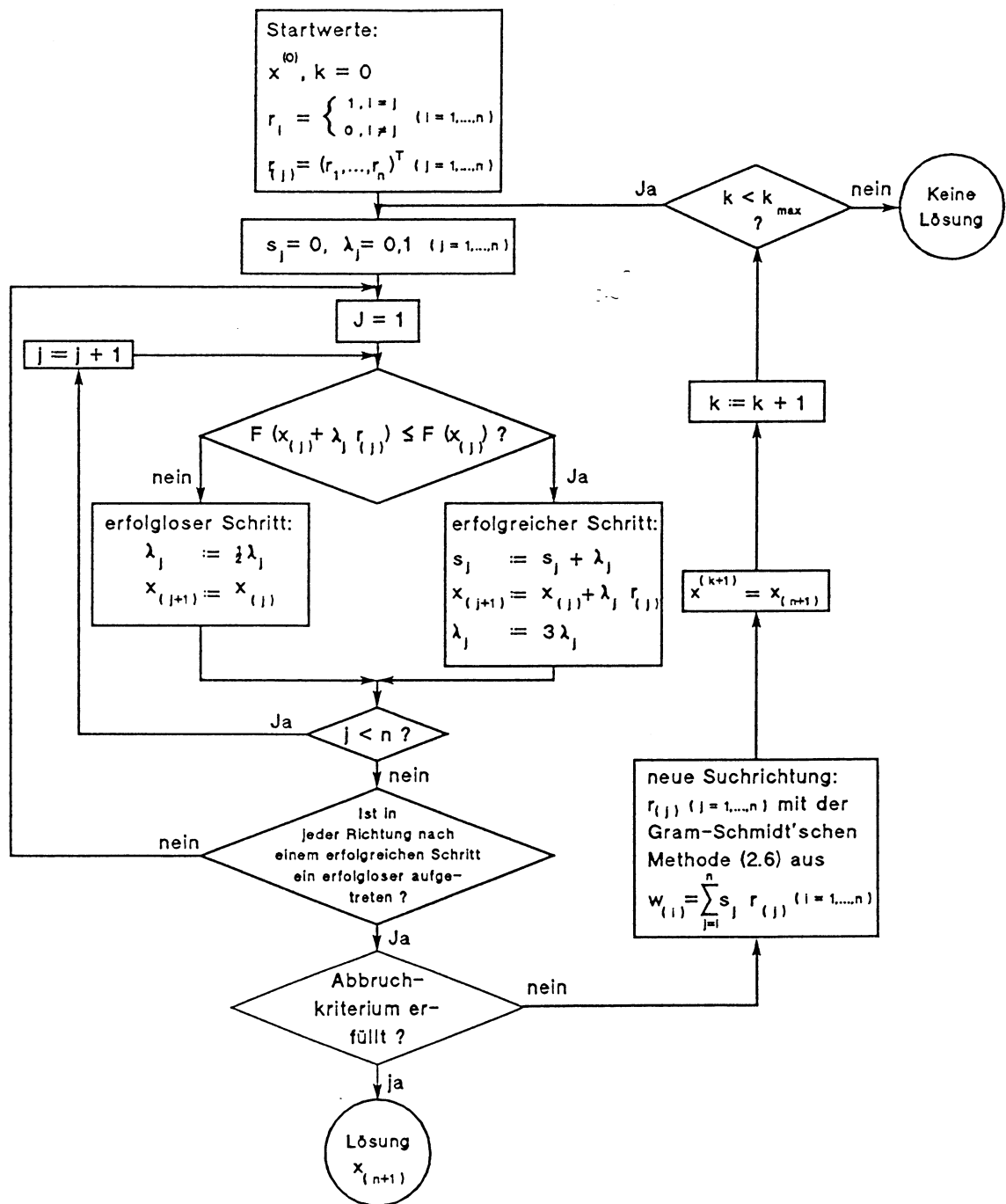
Anhang

- A.1 Flußdiagramme
- A.2 Die Ableitungen der Simulationsgleichungen
- A.3 Sensitivitätsanalyse
- A.4 UDM-Netze
- A.5 Optimierungsläufe
- A.6 Kurzbeschreibung und Quelltexte des Programms
OPTI

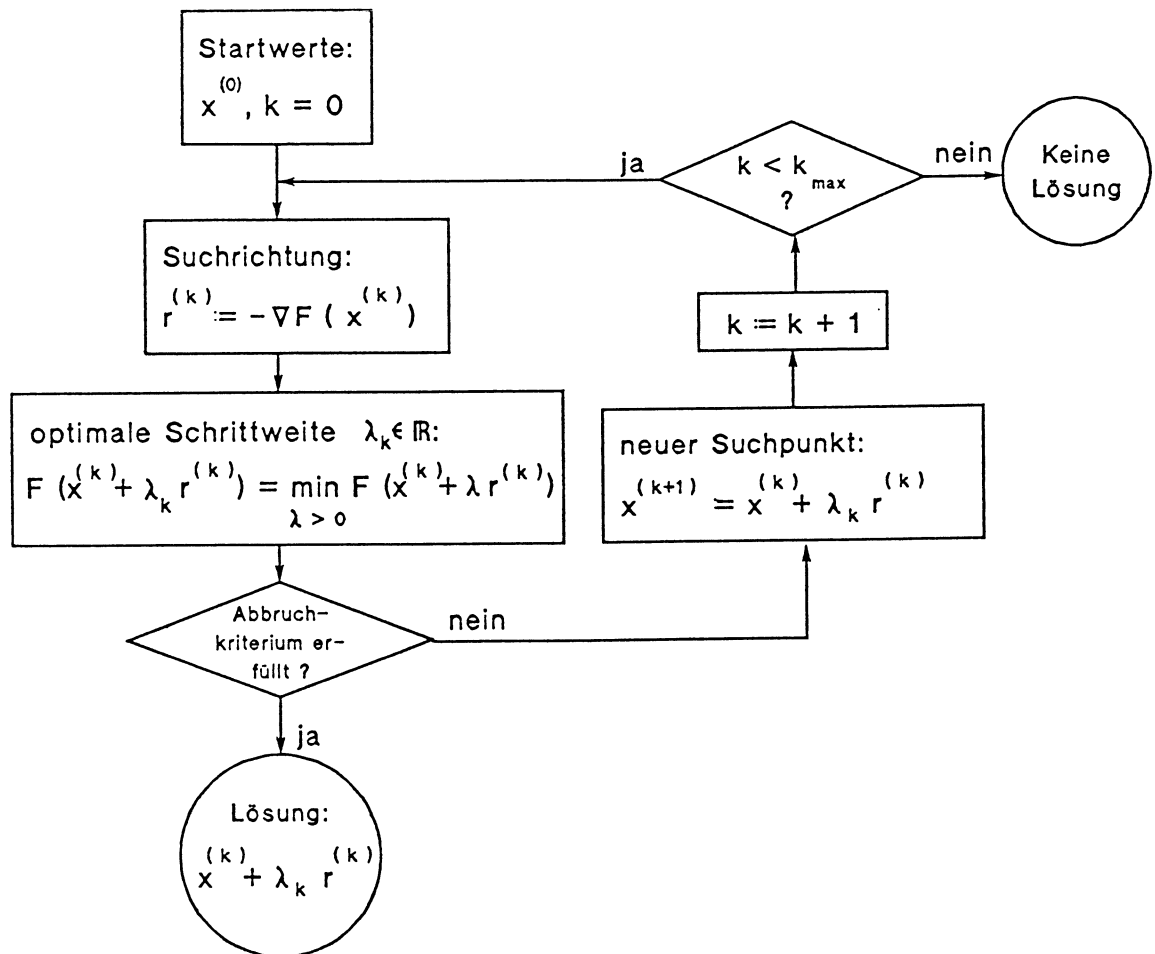
Anhang A.1 Flußdiagramme



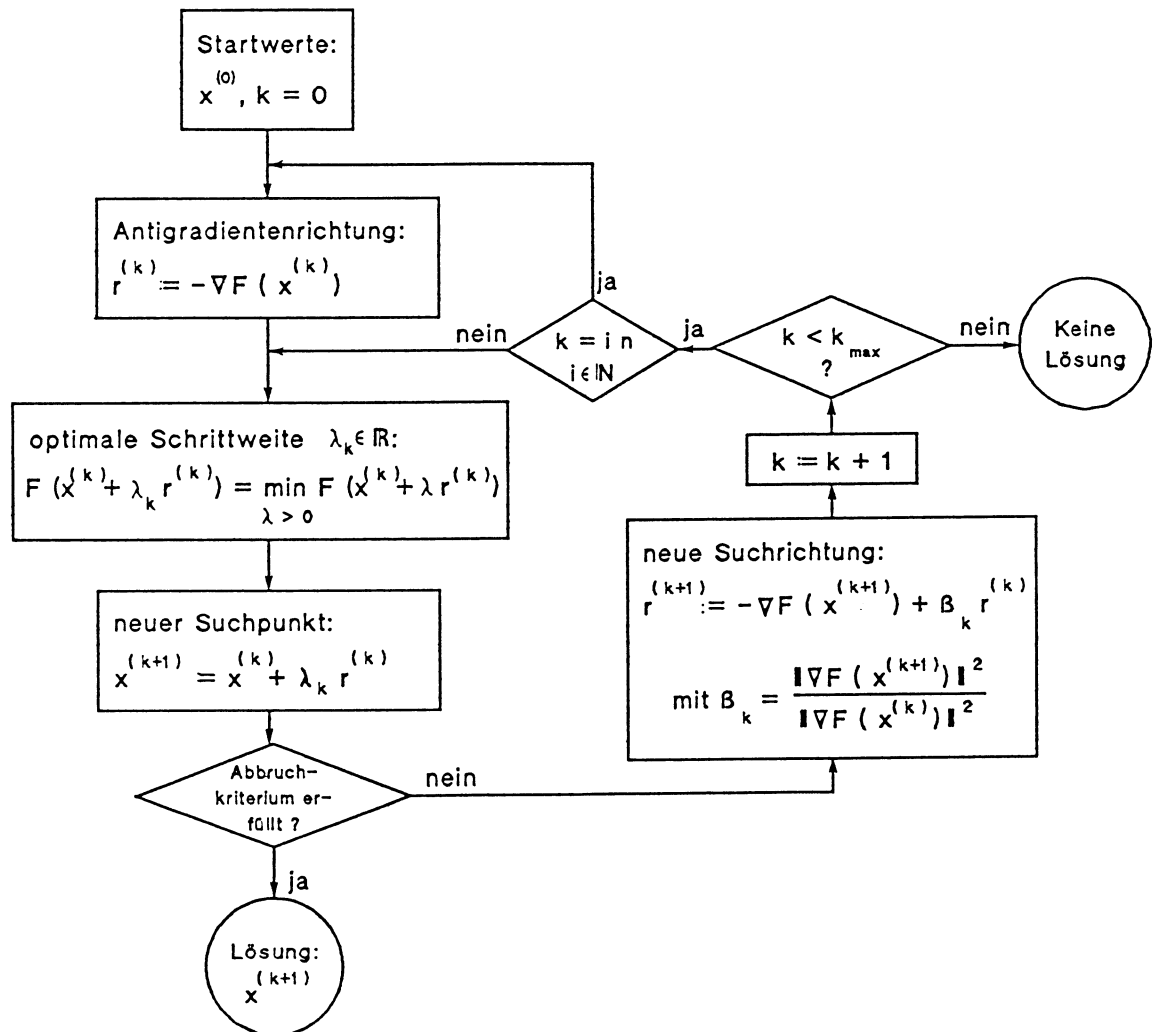
Die sukzessive Variation der Variablen



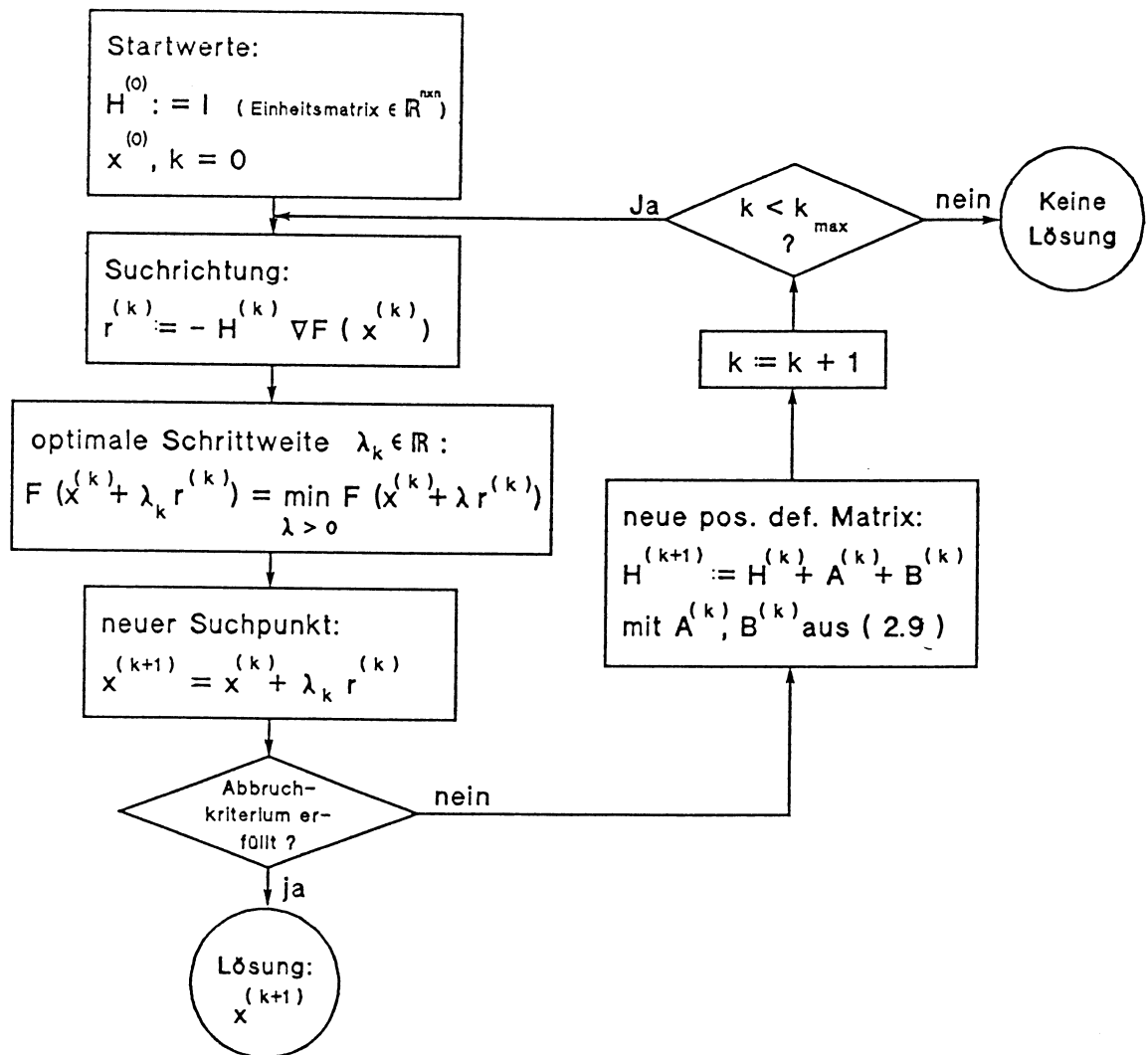
Die ROSENBRACK-Methode



Die optimale Gradientenmethode



DAS FLETCHER-REEVES-Methode



Das FLETCHER-POWELL-Verfahren

Institut für
Angewandte Mathematik

Institut für Wasserwirtschaft,
Hydrologie und landwirtschaft-
lichen Wasserbau

Universität Hannover

Parameteridentifikation
für
Simulationsmodelle
in der
Hydrologie

Anhang

Diplomarbeit

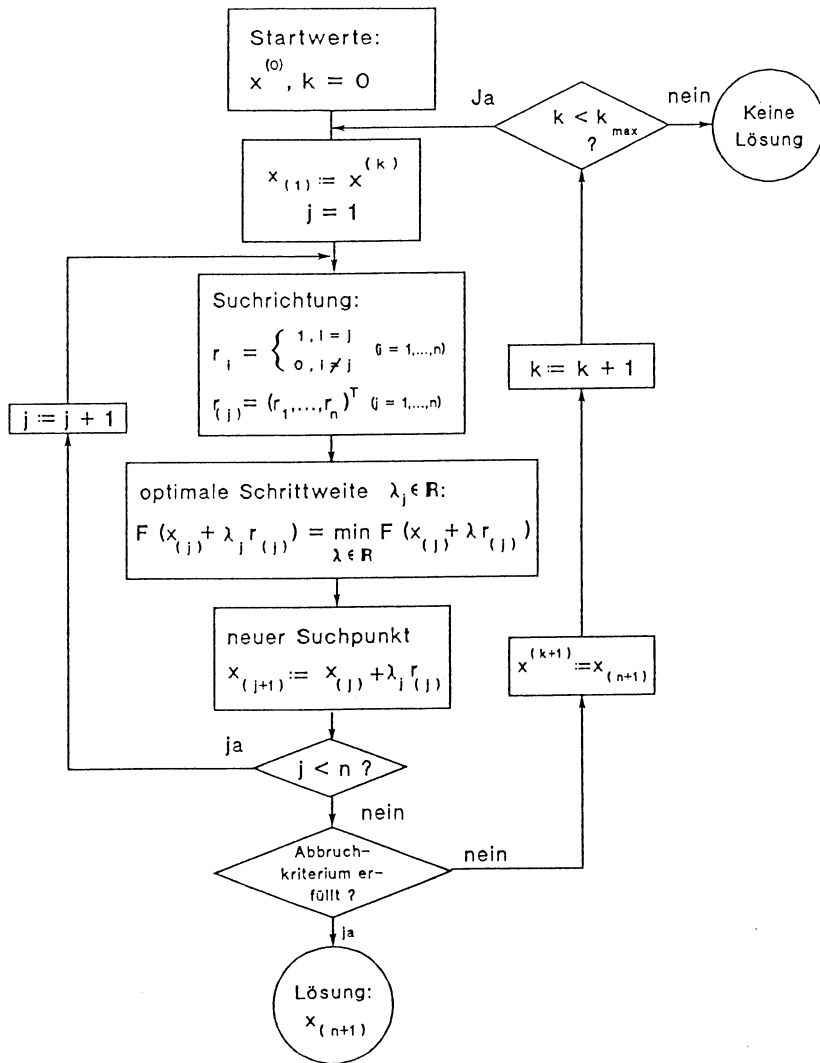
von

Susanne Deyda

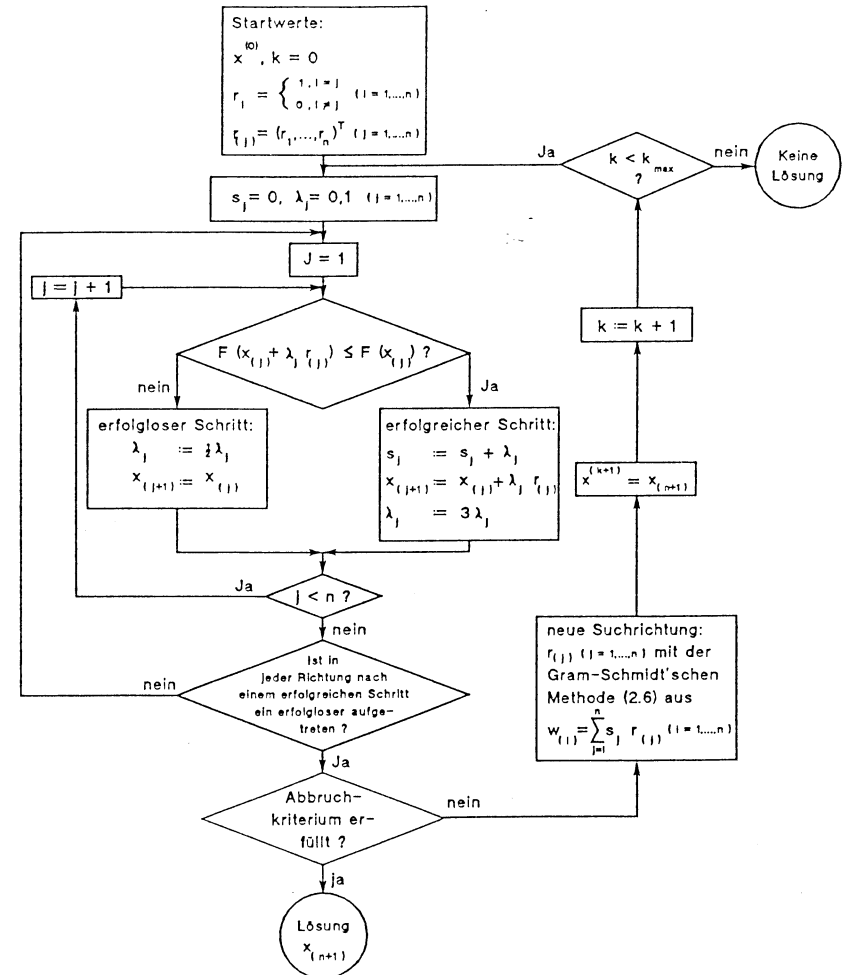
Anhang

- A.1 Flußdiagramme
- A.2 Die Ableitungen der Simulationsgleichungen
- A.3 Sensitivitätsanalyse
- A.4 UDM-Netze
- A.5 Optimierungsläufe
- A.6 Kurzbeschreibung und Quelltexte des Programms
OPTI

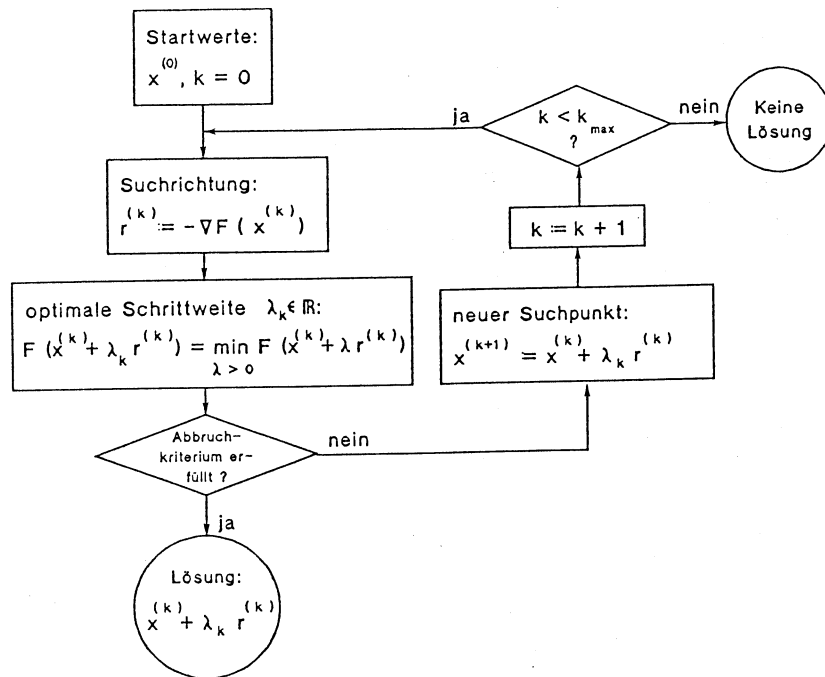
Anhang A.1 Flußdiagramme



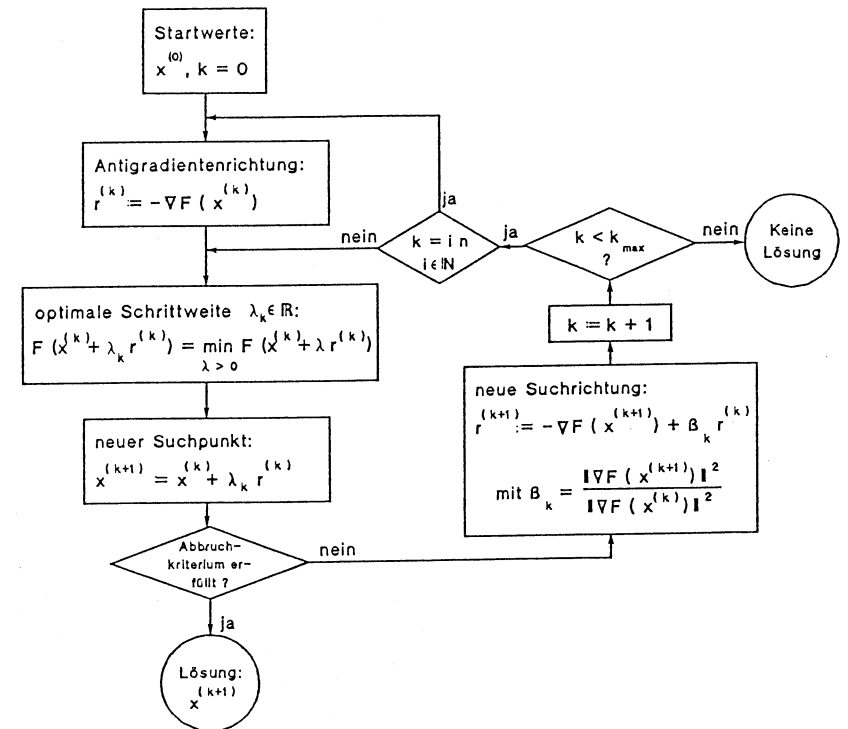
Die sukzessive Variation der Variablen



Die ROSENBRACK-Methode



Die optimale Gradientenmethode



DAS FLETCHER-REEVES-Methode

Anhang A.2 Die Ableitungen der Simulationsgleichungen

Mit

$$\begin{pmatrix} A_a \\ A_e \\ V_m \\ \alpha_u \end{pmatrix}^{(k+1)} = \begin{pmatrix} A_a \\ A_e \\ V_m \\ \alpha_u \end{pmatrix}^{(k)} + \lambda_k \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}^{(k)} \quad (k \in \mathbb{N})$$

folgt in der Prozedur innerhalb eines Iterationsschrittes:

$$\frac{dA_a}{d\lambda} = r_1, \quad \frac{dA_e}{d\lambda} = r_2, \quad \frac{dV_m}{d\lambda} = r_3, \quad \frac{d\alpha_u}{d\lambda} = r_4$$

Hierbei und im weiteren wird vereinfacht auf die Definition von Hilfsfunktionen $\varphi(\lambda)$ wie in Kapitel 2. verzichtet.

(3.1)':

$$\frac{\partial \varepsilon_i}{\partial A_a} = \left(\frac{\partial \varepsilon_{i-1}}{\partial A_a} - (1 - \varepsilon_{i-1}) \frac{N_i}{V_{mr}} \right) e^{-\frac{A_e - A_a}{V_{mr}} N_i}$$

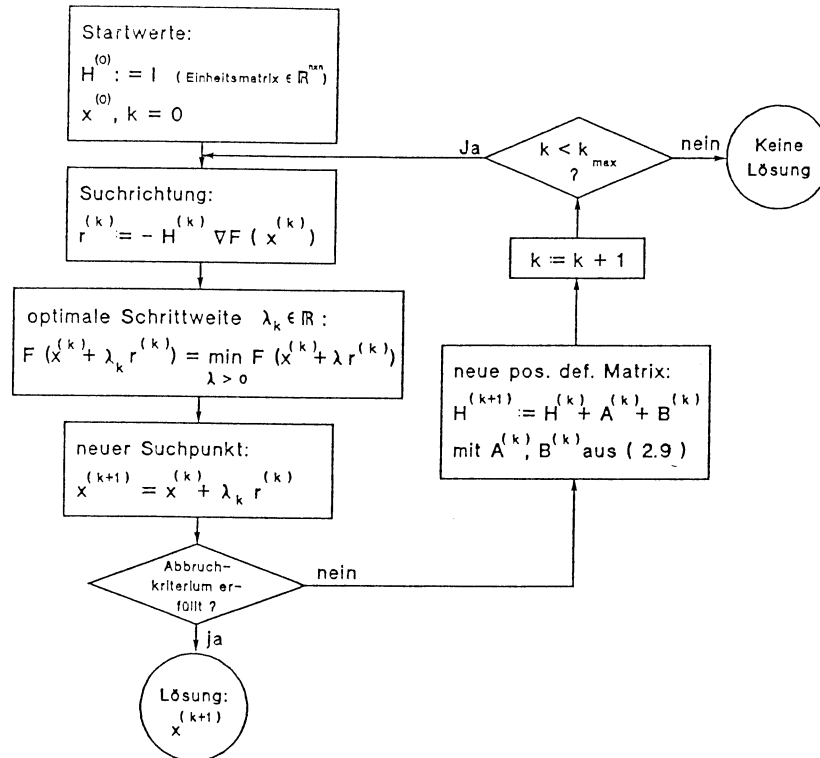
$$\frac{\partial \varepsilon_i}{\partial V_m} = \left(\frac{\partial \varepsilon_{i-1}}{\partial V_m} - (1 - \varepsilon_{i-1}) \frac{A_e - A_a}{(V_{mr})^2} e^{-(-0,0709) N_i} \right) e^{-\frac{A_e - A_a}{V_{mr}} N_i}$$

$$\frac{\partial \varepsilon_i}{\partial A_e} = \left(\frac{\partial \varepsilon_{i-1}}{\partial V_m} + (1 - \varepsilon_{i-1}) \frac{N_i}{V_{mr}} \right) e^{-\frac{A_e - A_a}{V_{mr}} N_i}$$

$$\frac{d\varepsilon_i}{d\lambda} = \left(\frac{d\varepsilon_{i-1}}{d\lambda} - (1 - \varepsilon_{i-1}) \frac{N_i}{V_{mr}} (r_1 + r_2 e^{-(-0,0709) \frac{A_e - A_a}{V_m}} - r_3) \right) e^{-\frac{A_e - A_a}{V_{mr}} N_i}$$

mit

$$\frac{\partial \varepsilon_o}{\partial A_a} = \frac{\partial \varepsilon_o}{\partial V_m} = \frac{\partial \varepsilon_o}{\partial A_e} = \frac{d\varepsilon_{i-1}}{d\lambda} = 0$$



(3.2)':

$$\frac{\partial R_i}{\partial A_a} = -V_{mr} \left(\frac{\partial \epsilon_i}{\partial A_a} - \frac{\partial \epsilon_{i-1}}{\partial A_a} \right)$$

$$\frac{\partial R_i}{\partial V_m} = (\epsilon_{i-1} - \epsilon_i) e^{(-0,07Q_g)} + V_{mr} \left(\frac{\partial \epsilon_{i-1}}{\partial V_m} - \frac{\partial \epsilon_i}{\partial V_m} \right)$$

$$\frac{\partial R_i}{\partial A_e} = N_i - V_{mr} \left(\frac{\partial \epsilon_i}{\partial A_e} - \frac{\partial \epsilon_{i-1}}{\partial A_e} \right)$$

$$\frac{dR_i}{d\lambda} = r_3 N_i - r_2 e^{(-0,07Q_g)} (\epsilon_i - \epsilon_{i-1}) - V_{mr} \left(\frac{d\epsilon_i}{d\lambda} - \frac{d\epsilon_{i-1}}{d\lambda} \right)$$

(3.4)':

$$\frac{\partial R_i}{\partial A_a} = \frac{\partial R_i}{\partial V_m} = 0$$

$$\frac{\partial R_i}{\partial A_e} = N_i$$

$$\frac{dR_i}{d\lambda} = r_3 N_i$$

(3.5)':

$$\frac{\partial t_L}{\partial \alpha_u} = 1$$

(3.9)':

$$\frac{\partial K_1}{\partial \alpha_u} = 0,82$$

(3.10)':

$$\frac{\partial t_{p1}}{\partial \alpha_u} = 0,5$$

(3.11)':

$$\frac{\partial Q_{ps1}}{\partial \alpha_u} = \frac{\frac{\partial k_1}{\partial \alpha_u} \left((6 t_{p1} + 12 k_1) \Delta t - 4 (t_{p1})^2 \right) + \frac{\partial t_{p1}}{\partial \alpha_u} (4 t_{p1} k_1 + (3 t_{p1} + 6 k_1) \Delta t)}{3 (t_{p1} + 2 k_1)^3}$$

(3.12)':

$$\frac{\partial Q_{p1}}{\partial \alpha_u} = \frac{A A_{und}}{0,006} \left(\frac{\frac{\partial Q_{ps1}}{\partial \alpha_u} t_L - Q_{ps1}}{(t_L)^2} \right)$$

(3.13)':

$$\frac{\partial k_2}{\partial \alpha_u} = -\frac{1}{0,99} \left(\frac{A A_{und}}{0,006 (Q_{p1})^2} \frac{\partial Q_{p1}}{\partial \alpha_u} + \frac{1}{2} \frac{\partial t_p}{\partial \alpha_u} \right)$$

(3.14)':

$$\frac{\partial t_p}{\partial \alpha_u} = \frac{t_p}{t_L}$$

(3.15)':

$$\frac{\partial Q_{p2}}{\partial \alpha_u} = -\left(\frac{1}{2} \frac{\partial t_p}{\partial \alpha_u} + 0,99 \frac{\partial k_2}{\partial \alpha_u} \right) \frac{0,006 (Q_{p2})^2}{A A_{und}}$$

(3.16)':

$$\frac{\partial Q_p}{\partial \alpha_u} = \frac{1}{2} \left(\frac{\partial Q_{p1}}{\partial \alpha_u} + \frac{\partial Q_{p2}}{\partial \alpha_u} \right)$$

(3.17)':

$$\frac{\partial Q_{ps1}}{\partial \alpha_u} := \frac{1}{2} \left(\frac{\partial Q_{ps1}}{\partial \alpha_u} + \frac{\partial Q_{ps2}}{\partial \alpha_u} \right)$$

(3.18)':

$$\frac{\partial Q_{e,i}}{\partial \alpha_u} = \left(\frac{\partial Q_p}{\partial \alpha_u} t_p - \frac{\partial t_p}{\partial \alpha_u} Q_p \right) \frac{t_i}{(t_p)^2}$$

$$\frac{dQ_{e,i}}{d\lambda} = r_4 \frac{\partial Q_{e,i}}{\partial \alpha_u} \quad (t_i = i\Delta t; i=1; \dots; \frac{t_p}{\Delta t})$$

(3.19)':

$$\frac{\partial Q_{e,i}^*}{\partial \alpha_u} = \left(\frac{\partial Q_p}{\partial \alpha_u} + \frac{Q_p}{k^2} \left(\frac{\partial t_p}{\partial \alpha_u} k + \frac{\partial k}{\partial \alpha_u} (t_i - t_p) \right) \right) e^{-\frac{t_i - t_p}{k}}$$

$$(t_i = i\Delta t; i = \frac{t_p}{\Delta t} + 1, \dots, \frac{t_g}{\Delta t})$$

(3.20)':

$$\frac{\partial \gamma}{\partial \alpha_u} = \frac{-\frac{1}{2} \frac{\partial Q_p}{\partial \alpha_u} (t_p + \Delta t) + Q_p \frac{\partial t_p}{\partial \alpha_u} + \gamma \sum_{i=\frac{t_p}{\Delta t}+1}^{\frac{t_g}{\Delta t}} \frac{\partial Q_{e,i}^*}{\partial \alpha_u} \Delta t}{\sum_{i=\frac{t_p}{\Delta t}+1}^{\frac{t_g}{\Delta t}} Q_{e,i}^* \Delta t}$$

(3.21)':

$$\frac{\partial Q_{e,i}}{\partial \alpha_u} = \gamma \frac{\partial Q_{e,i}^*}{\partial \alpha_u} + \frac{\partial \gamma}{\partial \alpha_u} Q_{e,i}^*$$

$$\frac{dQ_{e,i}}{d\lambda} = r_4 \frac{\partial Q_{e,i}}{\partial \alpha_u} \quad (i = \frac{t_p}{\Delta t} + 1, \dots, \frac{t_g}{\Delta t})$$

(3.22)':

$$\frac{\partial Q_{z,j}}{\partial A_a} = \sum_{i=1}^{r_{anz}} \frac{\partial R_i}{\partial A_a} Q_{e,j-i+1}, \quad \frac{\partial Q_{z,j}}{\partial V_m} = \sum_{i=1}^{r_{anz}} \frac{\partial R_i}{\partial V_m} Q_{e,j-i+1}$$

$$\frac{\partial Q_{z,j}}{\partial A_e} = \sum_{i=1}^{r_{anz}} \frac{\partial R_i}{\partial A_e} Q_{e,j-i+1}$$

$$\frac{\partial Q_{z,j}}{\partial \alpha_u} = \sum_{i=1}^{r_{anz}} R_i \frac{\partial Q_{e,j-i+1}}{\partial \alpha_u}$$

$$\frac{dQ_{z,j}}{d\lambda} = \sum_{i=1}^{r_{anz}} \left(\frac{dR_i}{d\lambda} Q_{e,j-i+1} + R_i \frac{dQ_{e,j-i+1}}{d\lambda} \right) \quad (0 < j-i+1 < l_{anz})$$

(3.27)': Mit

$$x_1 := A_a, \quad x_2 := V_m, \quad x_3 := A_e, \quad x_4 := \alpha_u$$

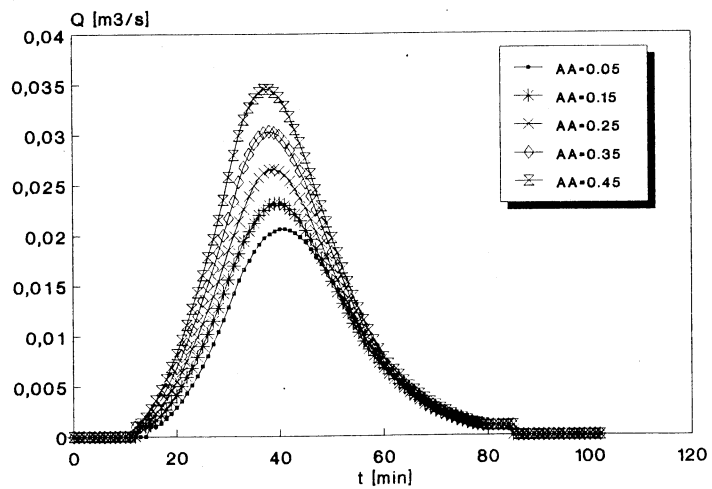
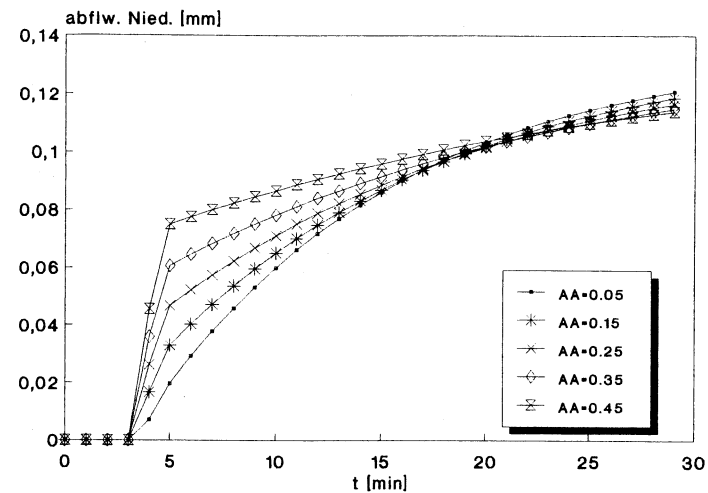
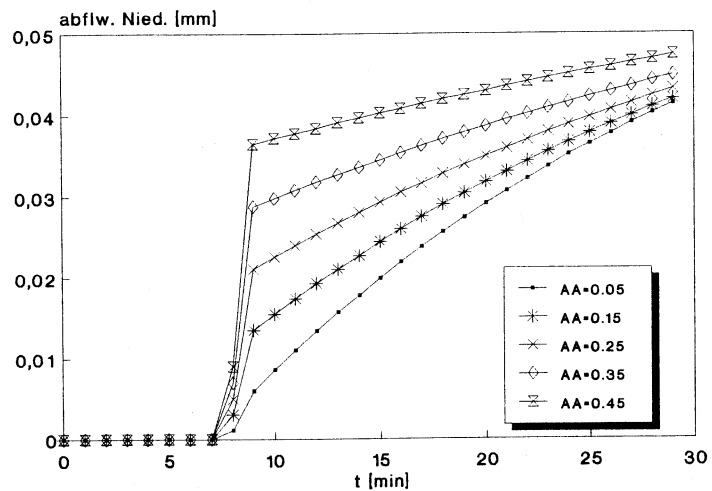
gilt:

$$\frac{\partial Q_l^{i+1}}{\partial x_j} = \sum_{k=1}^m a_{l,u} \frac{\partial Q_k^{(i)}}{\partial x_j} + \sum_{k=n}^m b_{l,k} \frac{\partial u_k^{(i)}}{\partial x_j} + \sum_{k=1}^m c_{l,k} \frac{\partial u_k^{i+1}}{\partial x_j}$$

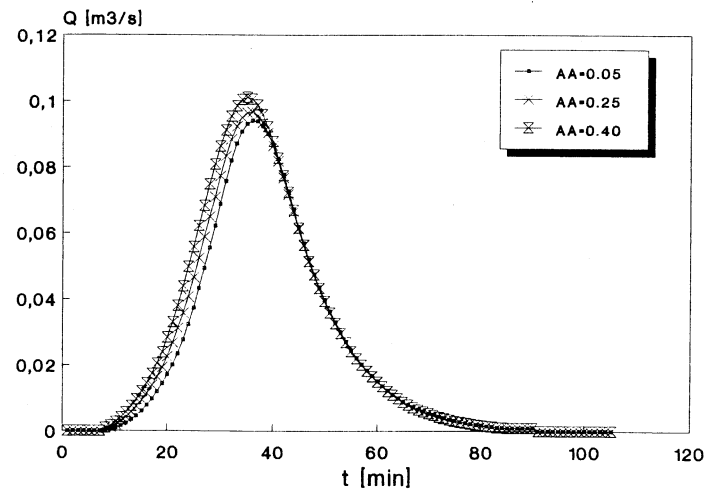
$$\frac{dQ_l^{i+1}}{d\lambda} = \sum_{k=1}^m a_{l,u} \frac{dQ_k^{(i)}}{d\lambda} + \sum_{k=n}^m b_{l,k} \frac{du_k^{(i)}}{d\lambda} + \sum_{k=1}^m c_{l,k} \frac{du_k^{i+1}}{d\lambda}$$

$$(j = 1, \dots, 4; \frac{\partial u_k^{(i)}}{\partial x_j}, \frac{\partial u_k^{(i+1)}}{\partial x_j} \text{ durch (3.22)' bestimmt})$$

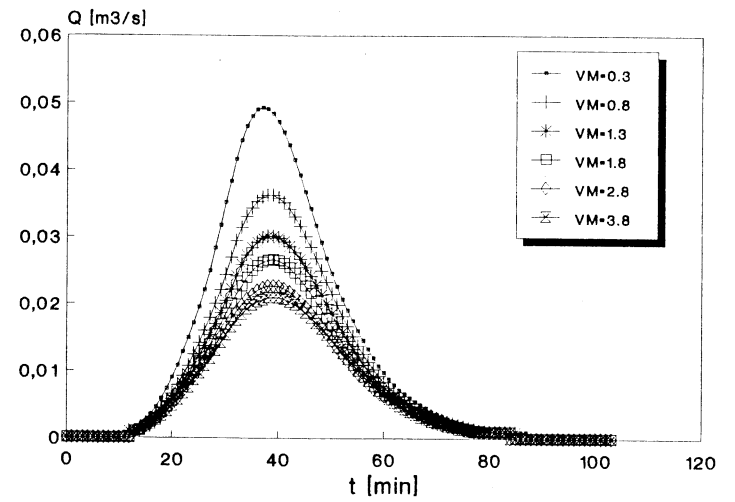
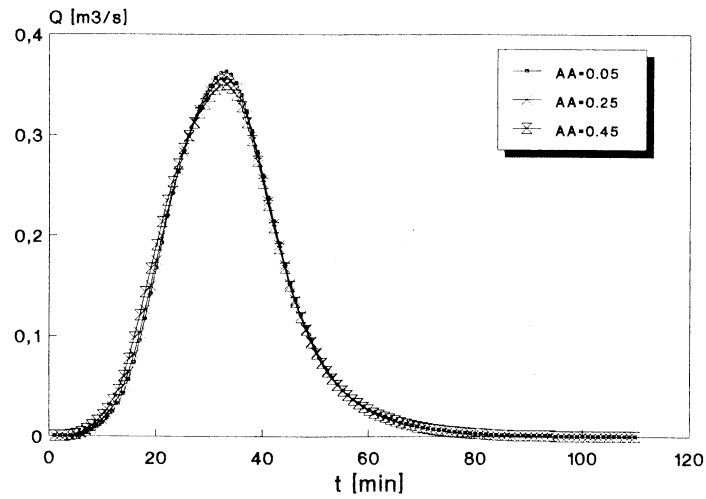
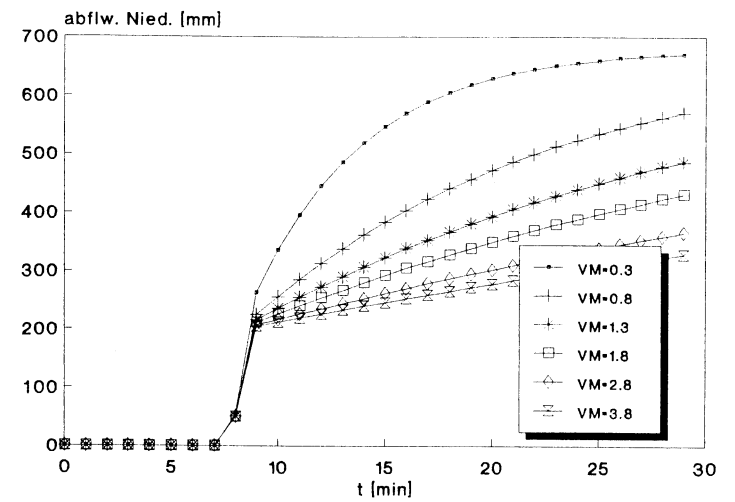
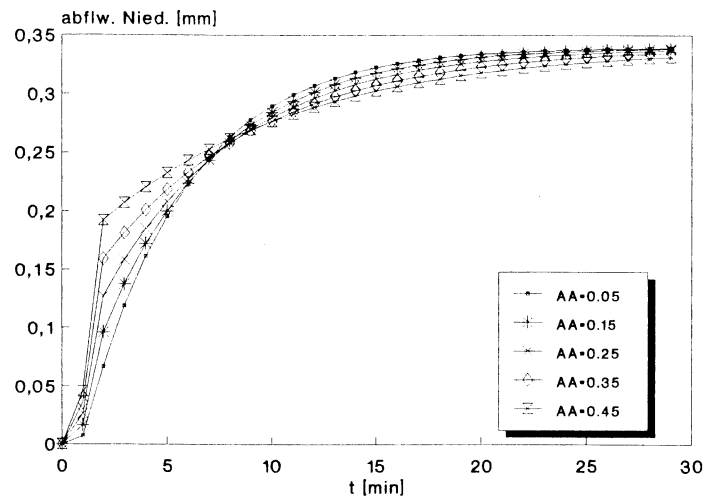
Anhang A.3 Sensitivitätsanalyse



Sensitivität des Parameters A, mit 0.08 mm/min Blockregen

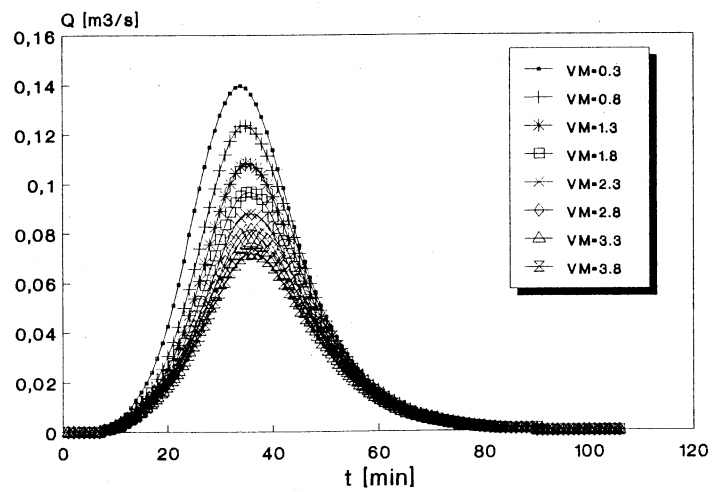
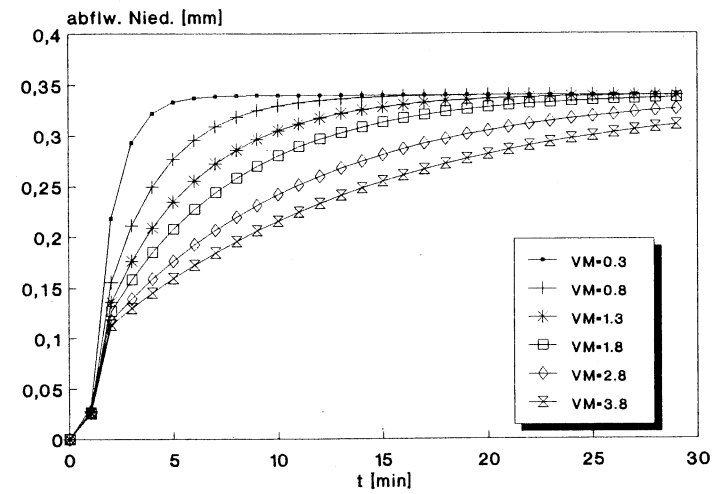
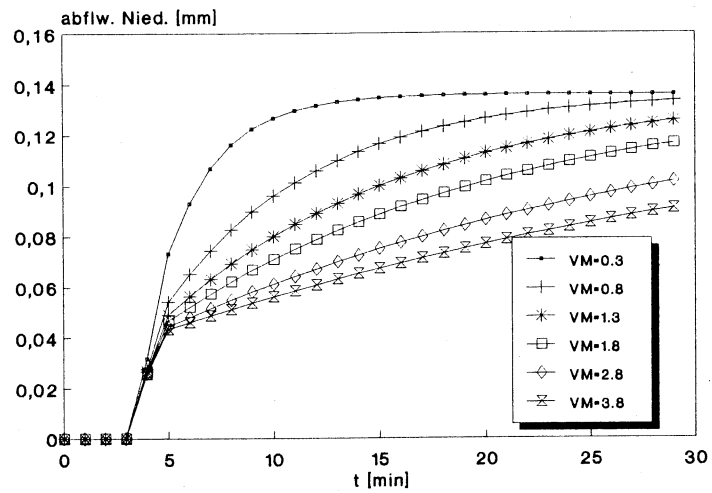


Sensitivität des Parameters A, mit 0.16 mm/min Blockregen

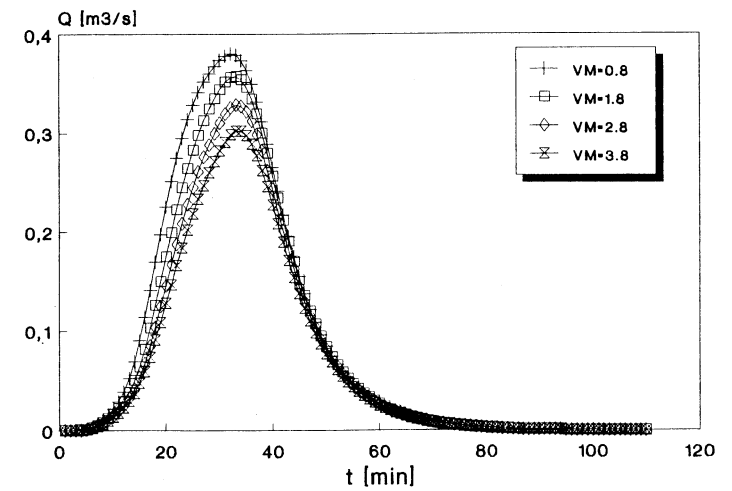


Sensitivität des Parameters A_A mit 0.40 mm/min Blockregen

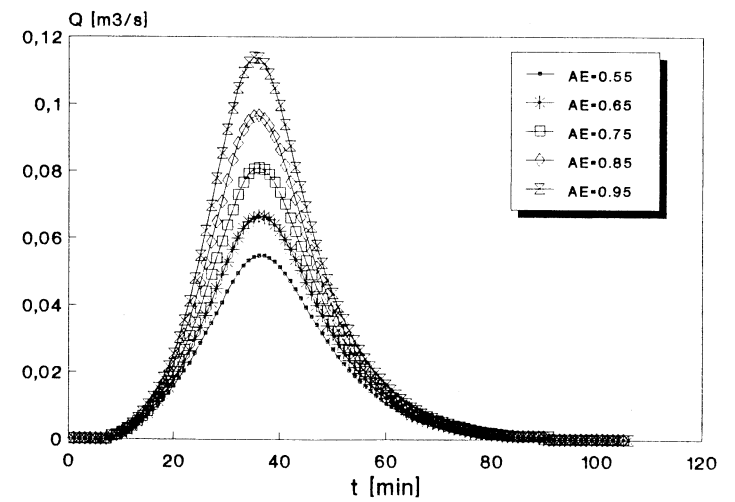
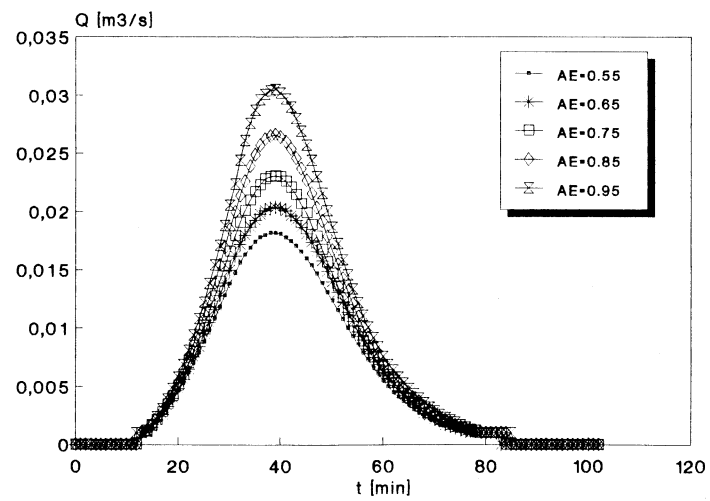
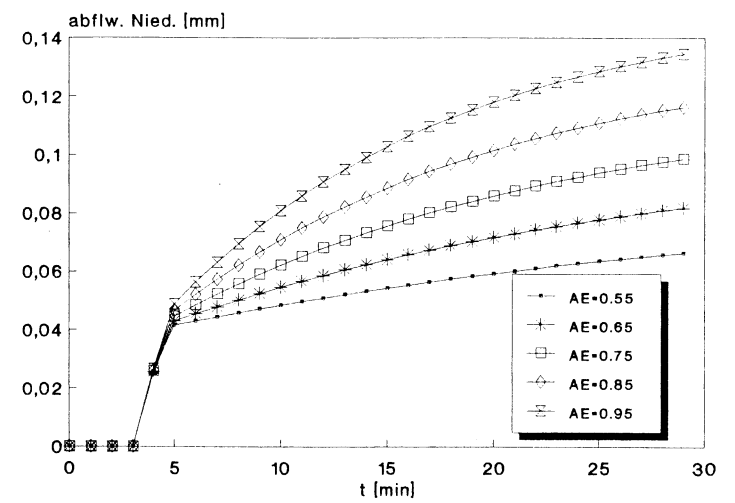
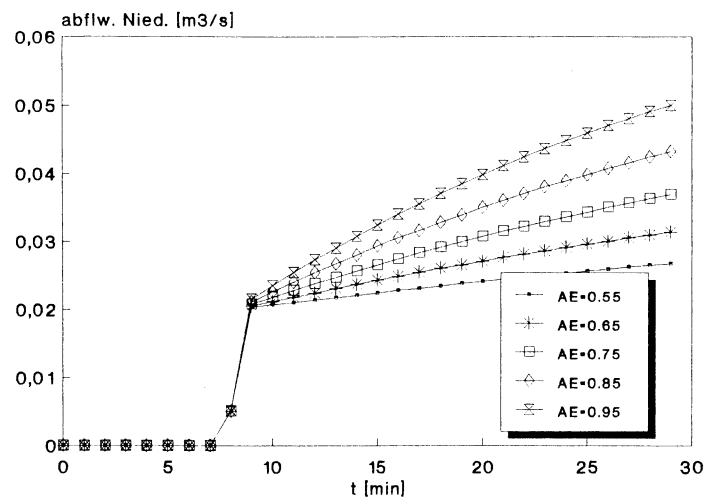
Sensitivität des Parameters V_M mit 0.08 mm/min Blockregen



Sensitivität des Parameters V_m mit 0.16 mm/min Blockregen

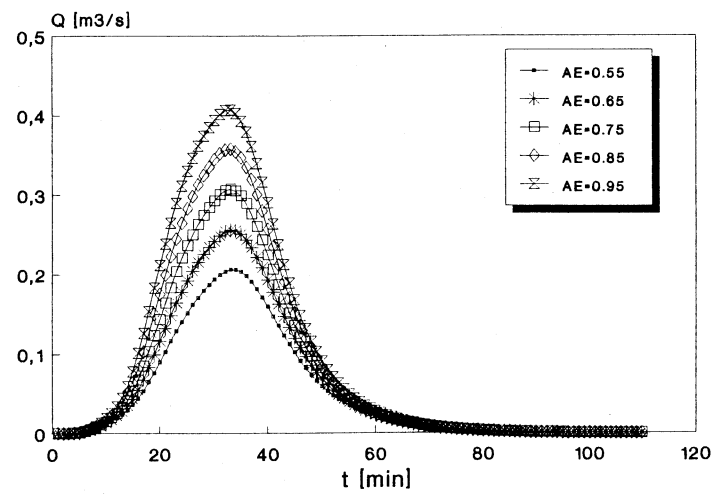
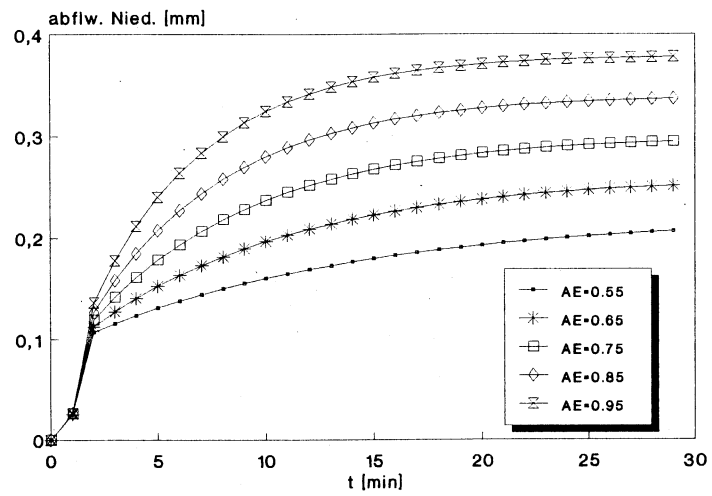


Sensitivität des Parameters V_m mit 0.40 mm/min Blockregen

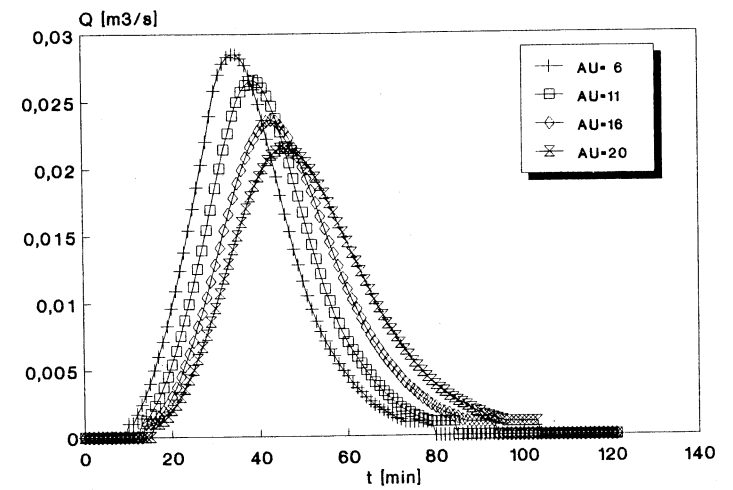


Sensitivität des Parameters A. mit 0.08 mm/min Blockregen

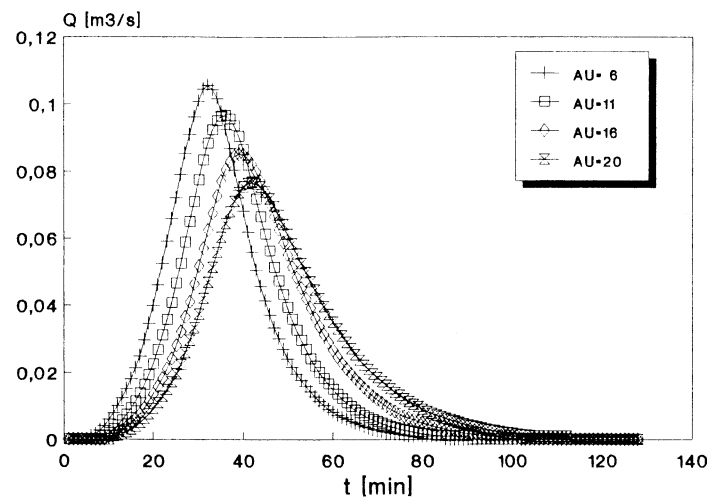
Sensitivität des Parameters A. mit 0.16 mm/min Blockregen



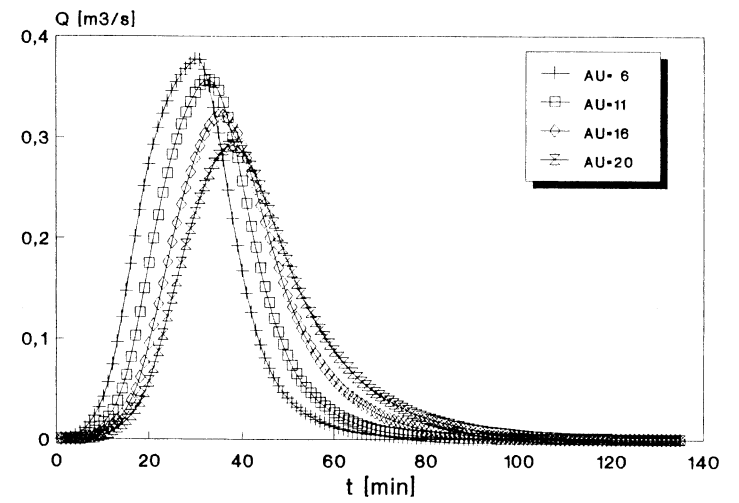
Sensitivität des Parameters A_e mit 0.40 mm/min Blockregen



Sensitivität des Parameters α_u mit 0.08 mm/min Blockregen

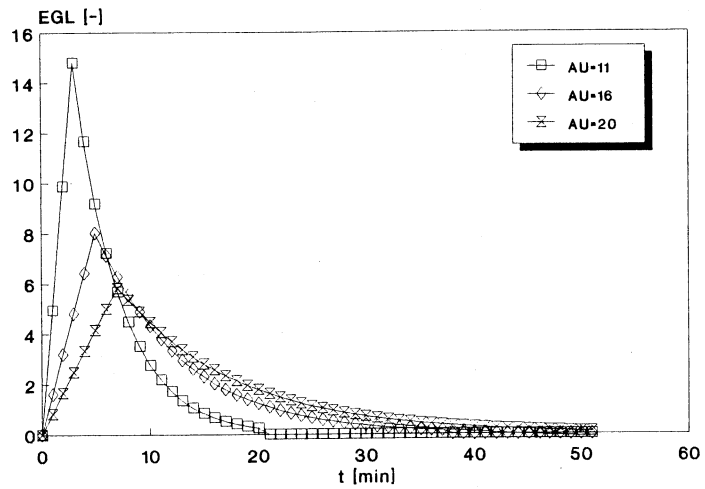
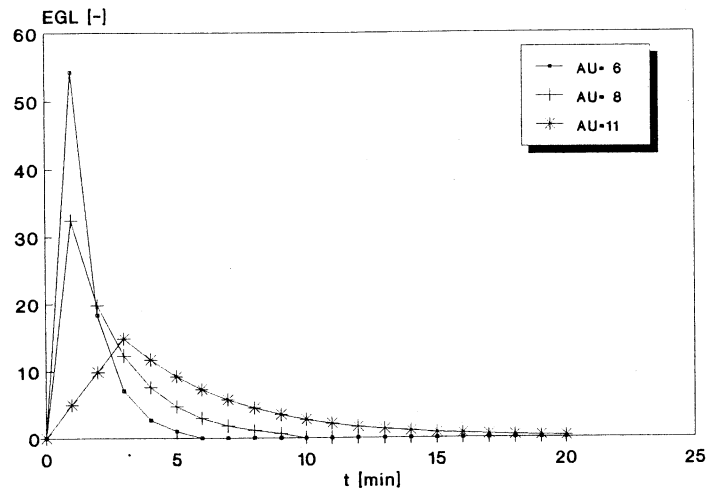


Sensitivität des Parameters α_u mit 0.16 mm/min Blockregen

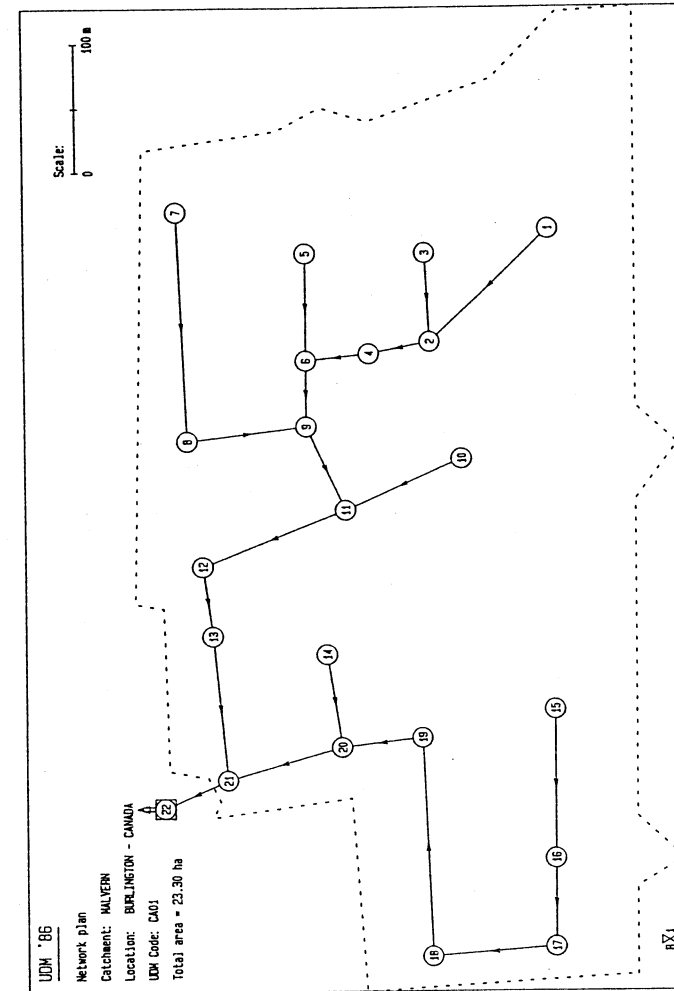


Sensitivität des Parameters α_u mit 0.40 mm/min Blockregen

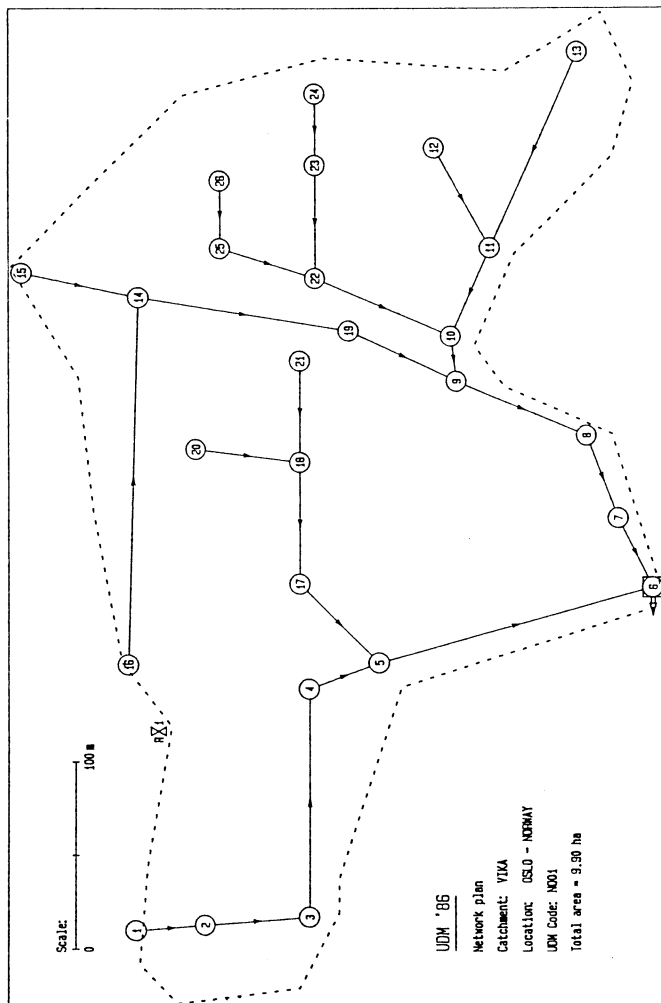
Anhang A.4 UDM-Netze



Sensitivität des Parameters α_u : Einheitsganglinien

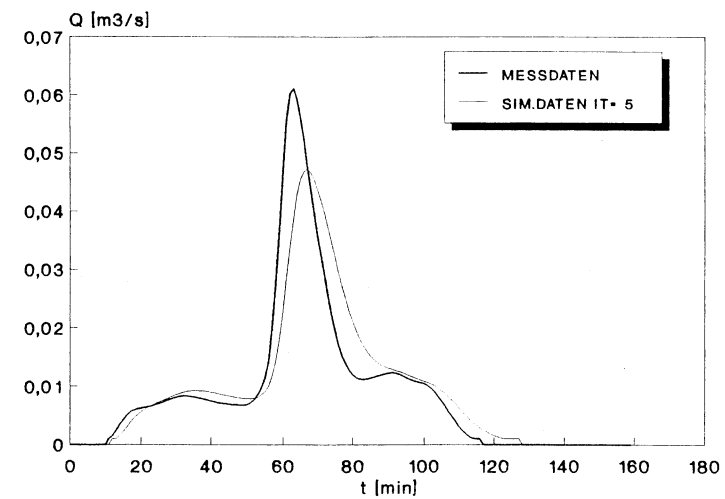
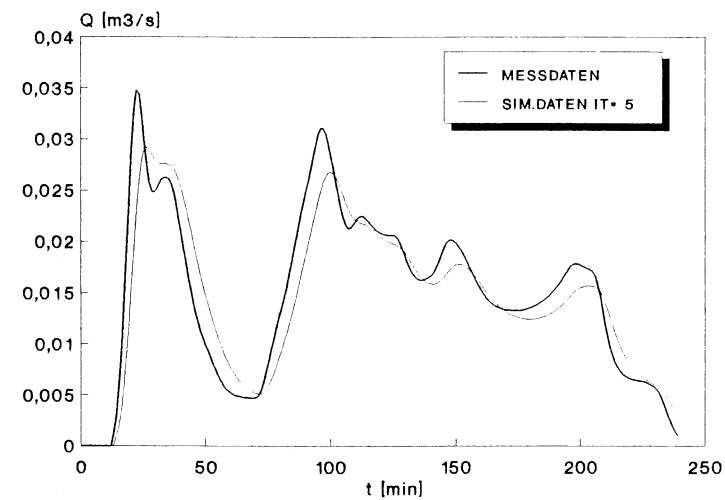


Grundriß des UDM-Netzes CA01

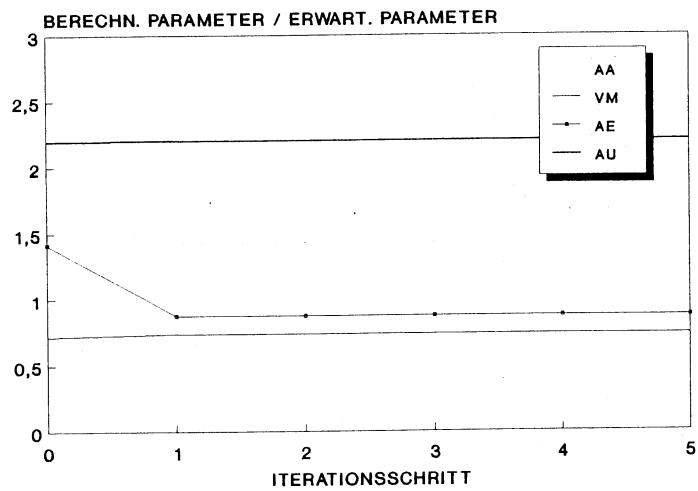
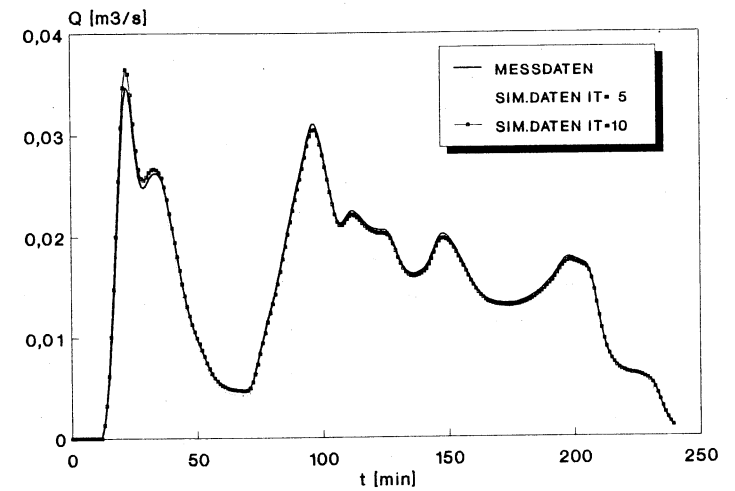
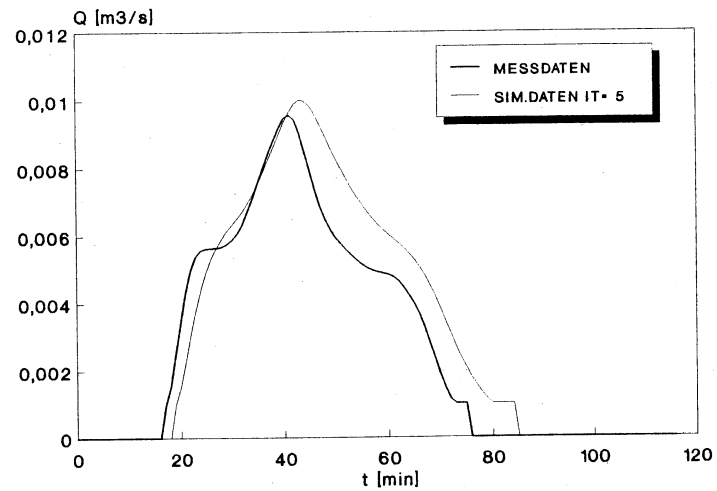


Grundriß des UDM-Netzes N001

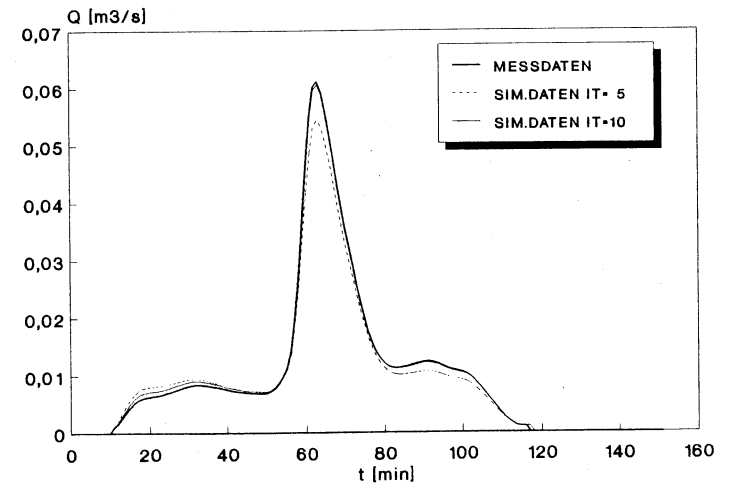
Anhang A.5 Optimierungsläufe



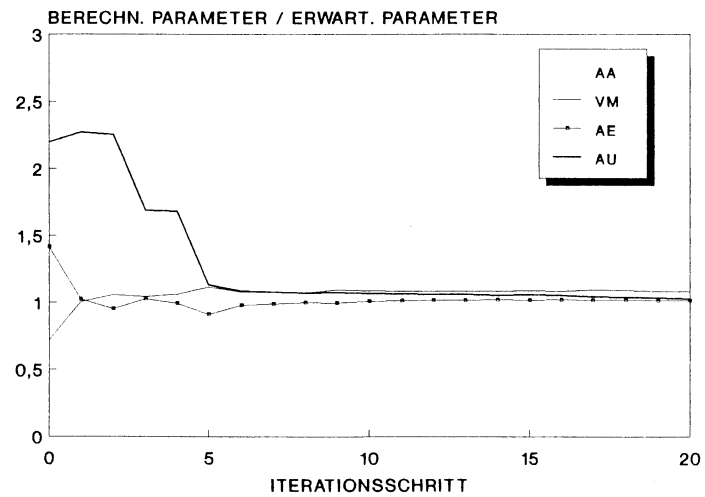
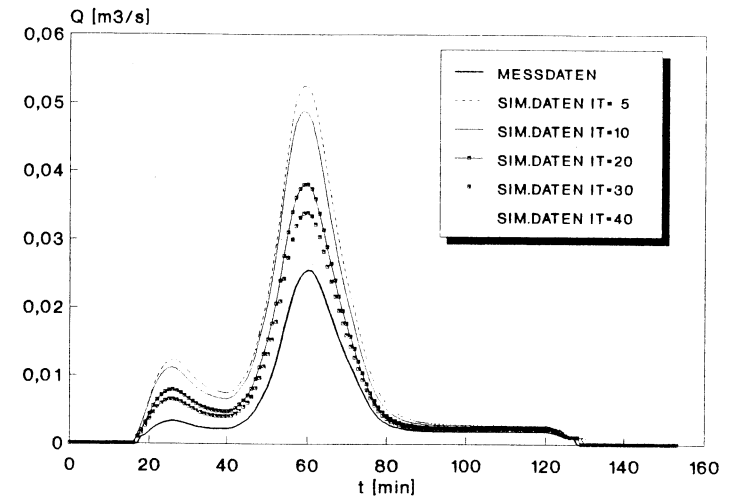
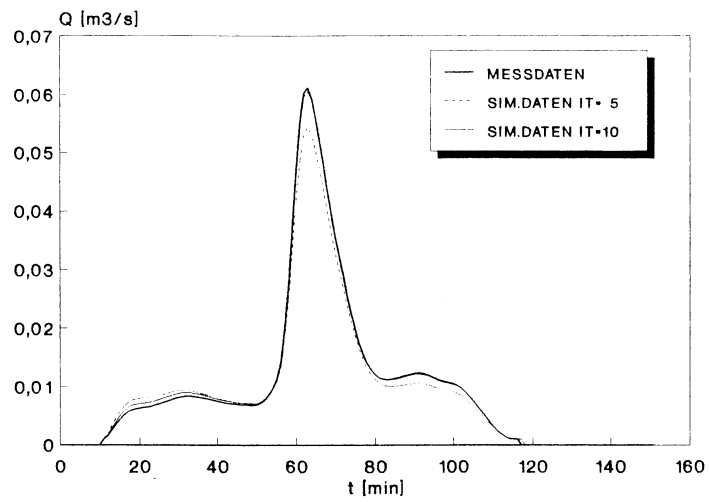
A.5.1 Optimierungslauf 1 : Regen 12 und 13



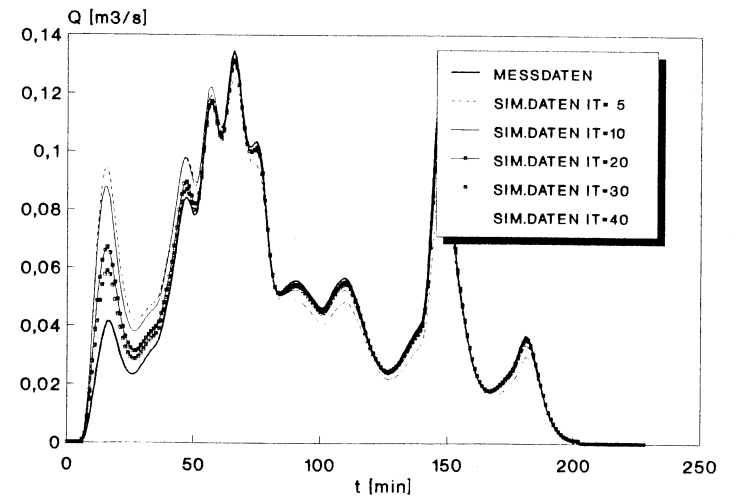
A.5.2 Optimierungslauf 1 : Regen 14



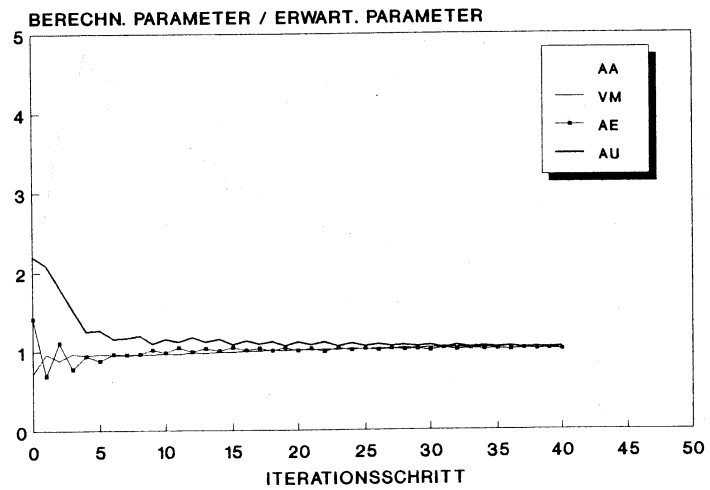
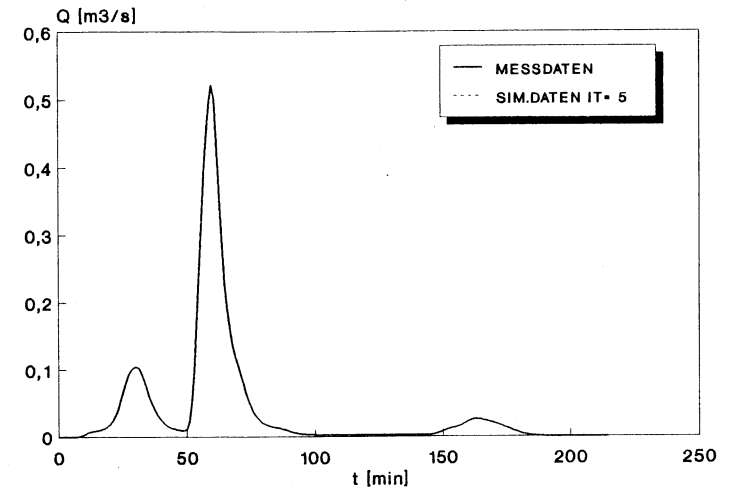
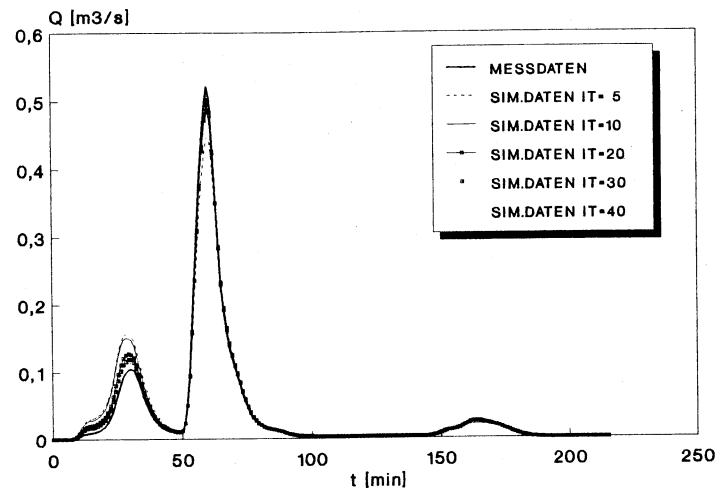
A.5.3 Optimierungslauf 2 : Regen 12 und 13



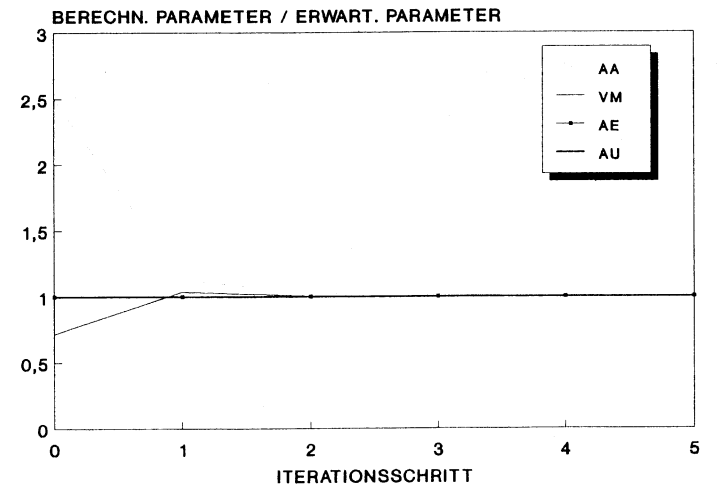
A.5.4 Optimierungslauf 2 : Regen 14



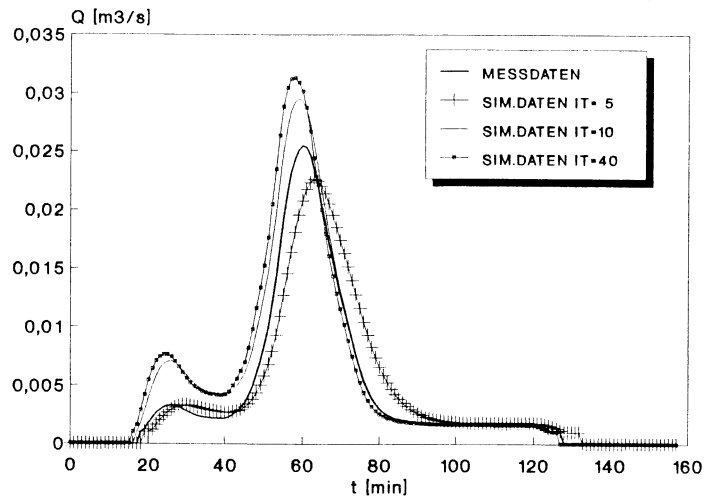
A.5.5 Optimierungslauf 3 : Regen 9 und 10



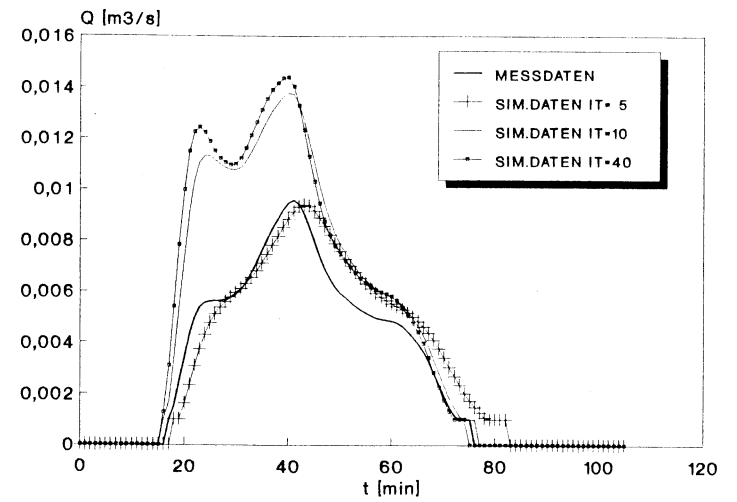
A.5.6 Optimierungslauf 3 : Regen 11



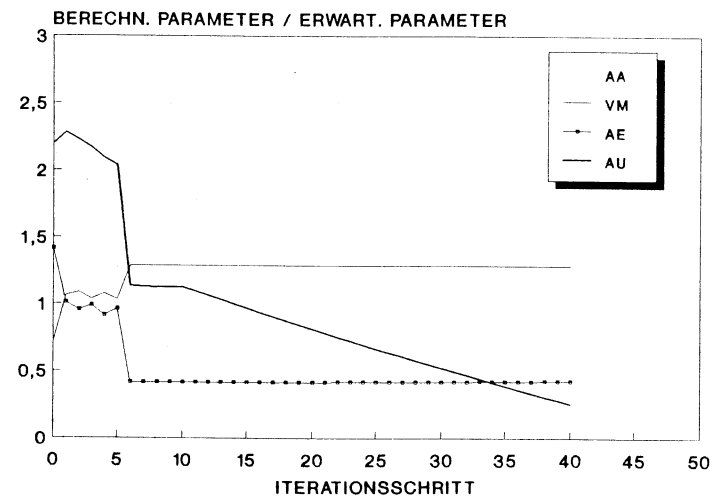
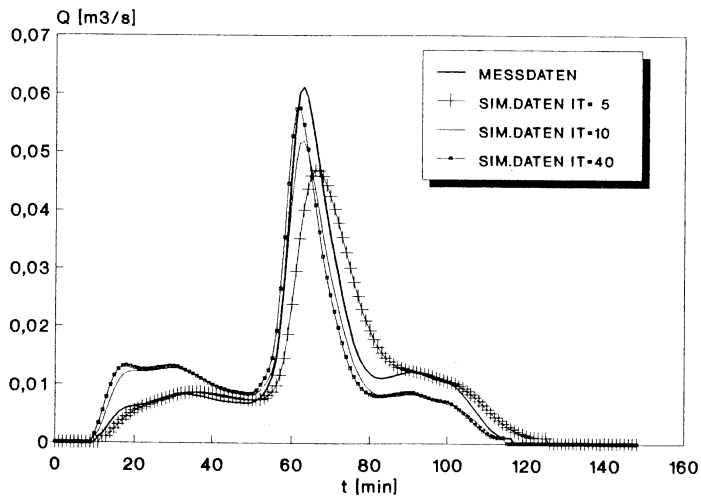
A.5.8 Optimierungslauf 4 : Regen 11

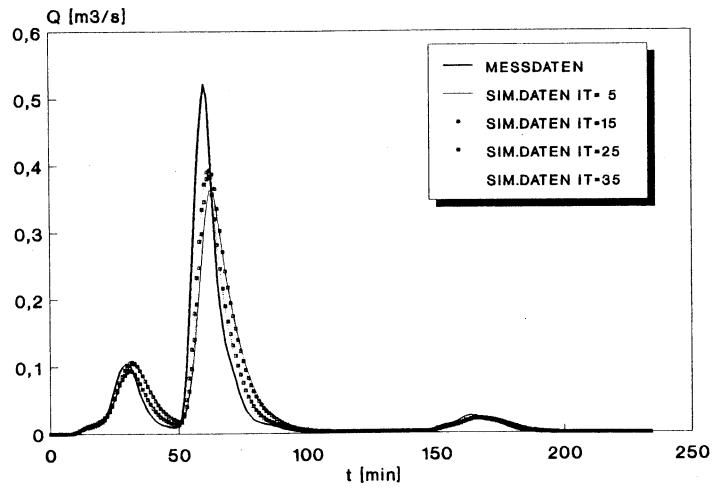
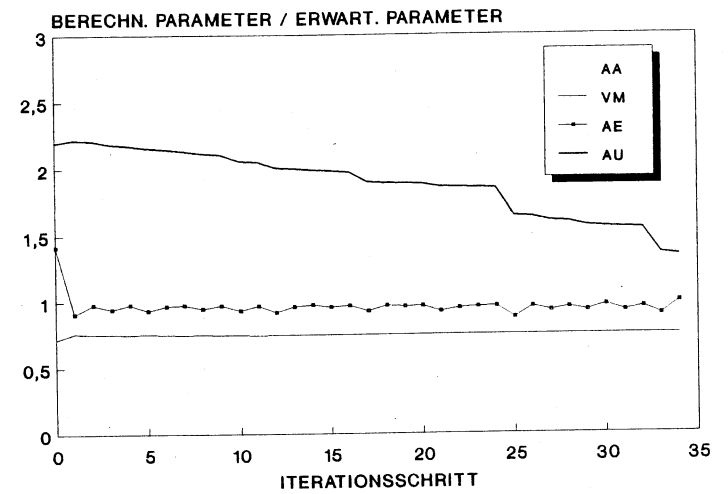
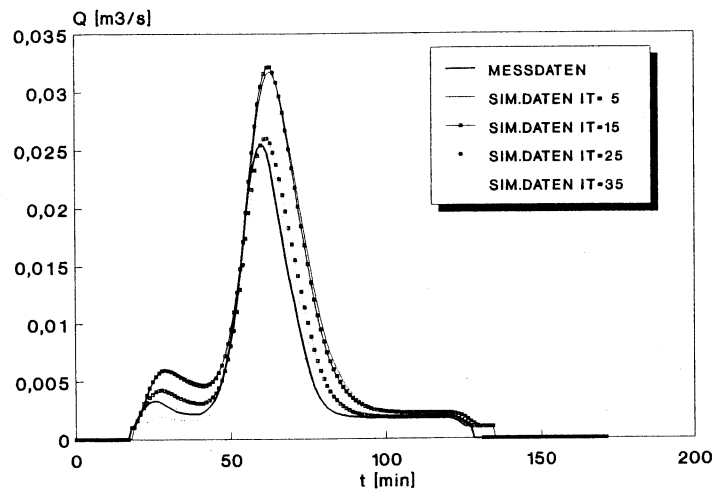


A.5.9 Optimierungslauf 5 : Regen 9 und 13



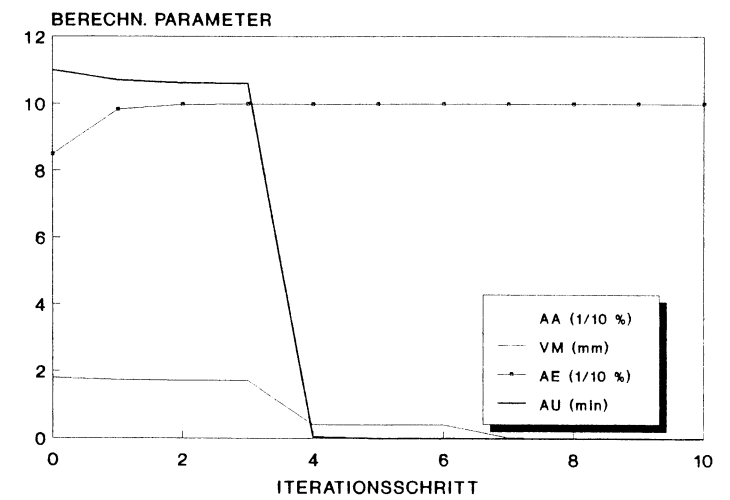
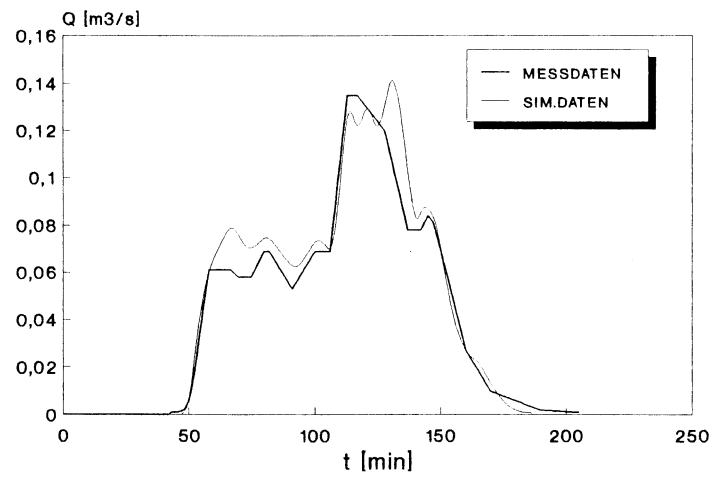
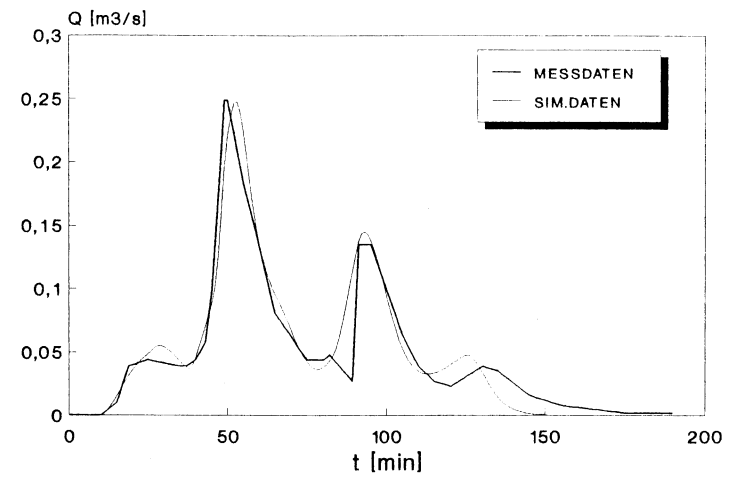
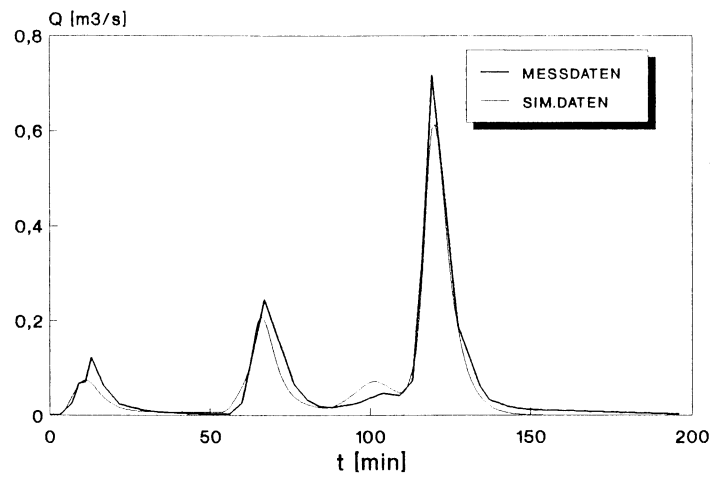
A.5.10 Optimierungslauf 5 : Regen 14





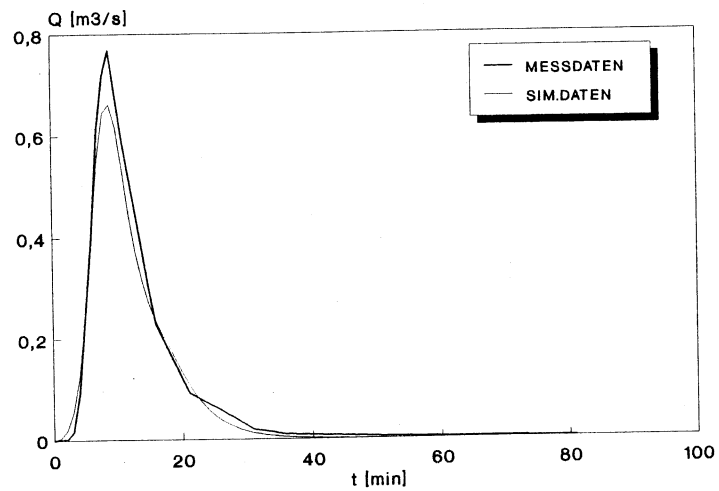
A.5.11 Optimierungslauf 6 : Regen 9 und 11

A.5.12 Optimierungslauf 6

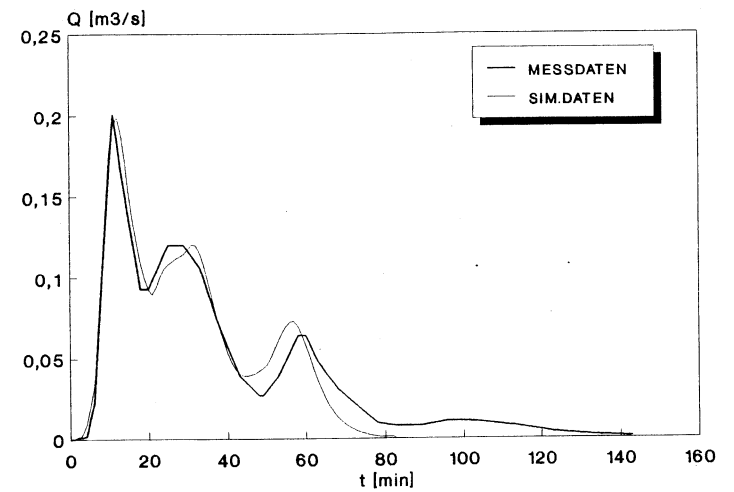


A.5.13 Optimierungslauf 7 : Regen 2 und 8

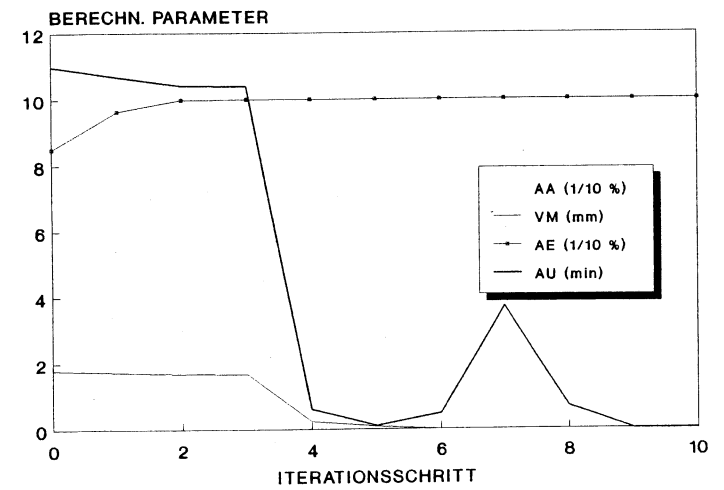
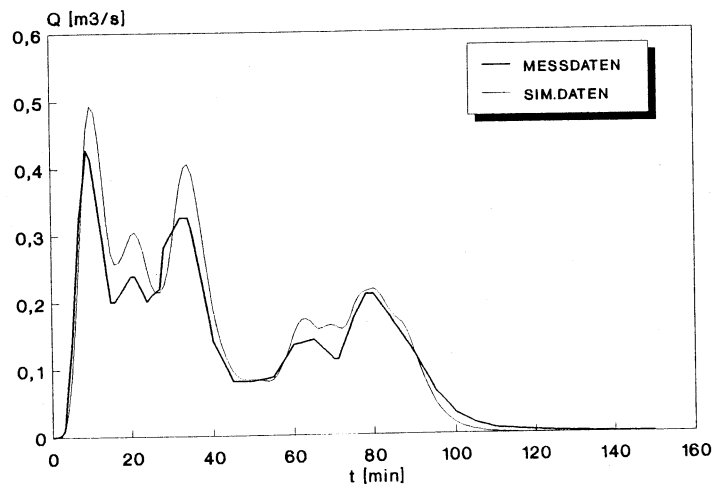
A.5.14 Optimierungslauf 7 : Regen 24



A.5.15 Optimierungslauf 8 : Regen 16 und 21



A.5.16 Optimierungslauf 8 : Regen 23



Anhang A.6 Kurzbeschreibung und Quelltexte des Programms OPTI

Das Hauptprogramm OPTI steuert den Ablauf des Programms und enthält die Schleife der Iterationsschritte.

Das Unterprogramm INITIA initialisiert die Variablen.

Das Unterprogramm OPTLES liest die Datei OPTPAR.DAT ein, die benötigte Dateinamen, Steuerparameter, und Anfangswerte enthält.

Das Unterprogramm NETLES liest alle benötigten Daten aus der Kanalnetzdatei ein.

Das Unterprogramm BLOCK ruft alle Programme zur Bestimmung der Modellwerte und deren Gradienten auf. Von hier aus wird das Modell HYSRAD/KMROUT aufgerufen.

Das Unterprogramm SEGL berechnet die Gradienten zu der Standardeinheitsganglinie.

Das Unterprogramm EPARAM berechnet die Parameter der Standardeinheitsganglinie mit den Gradienten.

Das Unterprogramm HYSPAR erstellt eine Parameterdatei für das Modell HYSRAD mit den berechneten Parameterwerten.

Das Unterprogramm RECH berechnet die Fehlerfunktion und deren Gradienten.

Das Unterprogramm REGLES liest den Kopf der Regendatei ein.

Das Unterprogramm PROLES liest den Kopf der Protokolldatei von KMROUT ein. In der Protokolldatei stehen die berechneten Retentionskonstanten der einzelnen Zeitschritte.

Das Unterprogramm NEUMAT sucht die Retentionskonstanten zu verschiedenen Zeitschritten aus der Protokolldatei heraus.

Das Unterprogramm KMAT berechnet die Matrizen mit den Kalinin-Miljukov-Konstanten.

Das Unterprogramm GRADI berechnet die Gradienten zu dem Modell HYSRAD/KMROUT.

Das Unterprogramm DKMROU berechnet die Gradienten zu dem Teilmodell KMROUT.

Das Unterprogramm NEUPAR berechnet die Richtungsvektoren und einen neuen Parametersatz für einen Iterationsschritt.

Das Unterprogramm STEP1 berechnet die optimale Schrittweite mit dem Bisektionsverfahren.

Das Unterprogramm STEP2 berechnet die optimale Schrittweite mit der kubischen Interpolation.

Das Unterprogramm KUBI berechnet die Nullstelle des Kubischen Interpolationspolynoms.

```

PARAMETER (NNR=300,NNE=100,NNC=50)
PARAMETER (N1=1,N2=2,N3=3,N4=4,N5=5,N6=6,N7=7,N8=8)

C   INTEGER      AR,FEHLER,SICMET,IRS(NNC),PGRENZ,NGRENZ,FEHFKT
    INTEGER      WCON

C   REAL          KRUECK,KREG,KMCON1,KMCON2,NORTOL

C   LOGICAL*1     ALMOD,AUSGEB,FEST,HYDABR

C   CHARACTER*1   STERN
    CHARACTER*10  ICOND,JUNC1L,NRHALT,JUN,REGFMT,IDUM,MCON
    CHARACTER*12  NETNAM,REGDA,REGNAM,MESSDA,MDAT,OPKMO*10
    CHARACTER*15  RGANF,RGEN

C   COMMON /INTEG / ISZ,ISMAX,ISMOD,FEHLER,IPRO,KR,AR,NC,IT,IEND,
+   ICYC,IPREND,MESS,NR,MRE,NGRENZ(4),PGRENZ(4),
+   KMT,FEHFKT,ITG,NTOL(4),WCON
+   COMMON /CHARA / NETNAM,REGDA(6),MESSDA(6),RGANF(6),RGEN(6),
+   NRHALT(NNC),JUN(NNC),REGFMT,OPKMO
+   COMMON /OPTIM / EPS,PAR(4),DQ(4),R(4),ALMOD,ALPHAM,QAL,DQDAL,
+   RDIAG(4),PDIAG(NNR,6),AUSGEB,QALALT,QDIFF1,
+   EPSTEP,ALTPAR(4),PARM(4),TOL,FEST(4),QDIFF2
+   COMMON /ABBRU / DIFTOL,NORTOL,DEVSTO,DEVMT0,VOLTOL,QMAX(6),
+   QVOL(6),CDEVS(6),CDEVMT(6),DEVS(6),DEVM(6),
+   VOLD(6),HYDABR
+   COMMON /HYSPA / DOWNJC,IDEFLT,JWAHL,JWELLE,JTABEL,SICMET,
+   IEGL,VBENUG,VMULDG,WNULL,ISOIL,RKSTD,ALPHDG,
+   AEDG,VBENDG,ADG,VERDFK,IMMER,EPsiUG,EPsiDG,
+   KRUECK,KREG,FK(6),VBUG(6),EPSUG(6)
+   COMMON /NETZ / VBU(NNC),EXQG(NNC),AGES(NNC),AUND(NNC),
+   HLEN(NNC),XLFU(NNC),ADDI(NNC),NJUNC1(NNC)
+   COMMON /REGDAT/ NANZ,IB,NRS(NNC),NNRS(NNC),RE(NNC),RI(NNC,0:NNE)
+   COMMON /SEINHG/ LGU(NNC),LGUN(NNC),QI(NNC,NNE),DQIDA(NNC,NNE)
+   COMMON /HYSRA1/ DEPSAL(NNC),DEPSAA(NNC),DEPSVM(NNC),DEPSAE(NNC),
+   DRDAL(NNC,0:NNE),DRDAA(NNC,0:NNE),
+   DRDVM(NNC,0:NNE),DRDAE(NNC,0:NNE),EPSIU(NNC)
+   COMMON /HYSRA2/ QTOT,DZADAL(NNC),DZADAA(NNC),DZADVM(NNC),
+   DZADAE(NNC),DZADAU(NNC),DZNDAL(NNC),DZNDAA(NNC),
+   DZNDVM(NNC),DZNDAE(NNC),DZNDAU(NNC)
+   COMMON /HYSRA3/ DMNDAL(NNC),DMNDAU(NNC),DMNDAA(NNC),DMNDVM(NNC),
+   DMNDAE(NNC),DMADAL(NNC),DMADAU(NNC),DMADAA(NNC),
+   DMADVMT(NNC),DMADAE(NNC)
+   COMMON /KMROU / ICHAZU(NNC,8),ICHA,ICHE,KMCON1(NNC),KMCON2(NNC),
+   QMAT(NNC,NNC),UMAT(NNC,NNC+1),DQNDAL(NNC),
+   DQNDAA(NNC),DQNDVM(NNC),DQNDAE(NNC),DQNDAU(NNC),
+   DQADAA(NNC),DQADVMT(NNC),DQADAE(NNC),DQADAU(NNC),
+   DQADAL(NNC)
C   common /testkm/ qzu(nnc),qzualt(nnc),qabalt(nnc),qab(nnc),
C   +   qzm(nnc),qzmalt(nnc)

```

```

PROGRAMM OPTI
*****
*
*   Identifikation der Paramter aus dem Programm HYSRAD mit
*   anschliessender Berechnung mit KMROUT
*
*****
$INCLUDE:'OPTI.CMN'
*
**** Initialisierung und Einlesen der Parameter und Daten ****
*
    CALL INITIA
    CALL OPTLES
    CALL NETLES
    OPEN(N6,FILE='OPT.PRO')
    IF (ipro.eq.1) OPEN(N7,file='opt.gra')
*
**** Iterationsschrittzaeher ****
*
    DO 100 ISZ=1,ISMAX
        WRITE(N6,*) ' ITERATIONSSCHRITT :',ISZ
        AUSGEB = .FALSE.
*
**** Bestimmung der Simulationswerte aus den Programmen ****
**** HYSRAD und KMROUT sowie der Gradienten ihrer Parameter ****
*
        AA = PAR(1)
        VM = PAR(2)
        AE = PAR(3)
        AU = PAR(4)
        CALL BLOCK(AA,VM,AE,AU)
*
**** Parameterloesung gefunden ? ****
*
        IF (HYDABR) THEN
*
**** Abbruch nach hydrologischen Kriterien ****
*
            DO 10 KR=1,AR
                IF (DEVS(KR).GT.DEVSTO) GOTO 15
                IF (DEVM(KR).GT.DEVMT0) GOTO 15
                IF (VOLD(KR).GT.VOLTOL) GOTO 15
            10    CONTINUE
                GOTO 900
            ENDIF
*
**** Abbruch bei kleinen Gradienten der Fehlerfunktion ****
*
            15    CONTINUE
                IF (FEHFKT.GE.0 .AND. NORTOL.NE.0.) THEN
                    QNORM2 = 0.
                    DO 20 L=1,4
                        QNORM2 = QNORM2 + DQ(L) * DQ(L)
                    20    CONTINUE
                    QNORM = SQRT(QNORM2)
                    IF (ISZ.EQ.1) THEN
                        NORTOL = NORTOL * QNORM
                        WRITE(N6,*) ' NORTOL(rel.)',NORTOL
                    ENDIF

```

```

      IF (IPRO.EQ.1 .or. ipro.eq.0)
+      WRITE(N6,1000) ISZ,QNORM
1000  FORMAT(' ITERATIONSSCHRITT: ',I2,' QNORM: ',F12.6)
      IF (QNORM.LT.NORTOL) THEN
          GOTO 900
      ELSE
          GOTO 90
      ENDIF
      ELSE IF (FEHFKT.LT.0 .AND. DIFTOL.NE.0.) THEN
*
***** Abbruch bei kleinen Differenzen der Fehlerfunktionswerte *****
*
      IF (IPRO.EQ.1 .or. ipro.eq.0)
+      WRITE(N6,1200) ISZ,QDIFF1,QDIFF2
1200  FORMAT(' ITERATIONSSCHRITT: ',I2,' QDIFF1/2: ',2F12.6)
      IF (QDIFF1.LT.DIFTOL .AND. QDIFF2.LT.DIFTOL) GOTO 900
      ENDIF
*
***** neue Parameter bestimmen *****
*
90    CONTINUE
      CALL NEUPAR
      IF (IPRO.EQ.1 .or. ipro.eq.0)
+      WRITE(N6,1100) (PAR(L),L=1,4),(R(L),L=1,4),ALPHAM
1100  FORMAT(' PARAMETER(1-4) : ',4F12.4,/,
+          ' RICHTUNGEN(1-4) : ',4F12.6,' ALPHAM : ',F12.4)
100  CONTINUE
*
***** keine Loesung gefunden *****
*
      IF (IPRO.EQ.1 .or. ipro.eq.0) THEN
          WRITE(N6,'(A)') ' KEINE LOESUNG!'
      ENDIF
      GOTO 999
900  CONTINUE
*
***** Loesung gefunden *****
*
      IF (IPRO.EQ.1 .or. ipro.eq.0) THEN
          WRITE(N6,'(A)') ' LOESUNG GEFUNDEN!'
      ENDIF
999  CONTINUE
      CLOSE(N6)
      if (ipro.eq.1) close(n7)
C
      END
C
C
C
      SUBROUTINE INITIA
*****
*      Initialisierung der festen Parameter
*
*****
$INCLUDE:'OPTI.CMN'
*
      DO 4 L=1,4
          RDIAG(L) = 1.0
          NGRENZ(L) = 0

```

```

      NTOL(L) = -1
      FEST(L) = .FALSE.
4    CONTINUE
      DO 6 K=1,6
          DO 5 J=1,NNR
              PDIAG(J,K) = 1.0
5    CONTINUE
          QMAX(K) = 0.
          QVOL(K) = 0.
6    CONTINUE
      do 8 j=1,nnr
C      qzu(j) = 0.
C      qzualt(j) = 0.
C      qabalt(j) = 0.
C      qab(j) = 0.
C 8    continue
      DOMNJC = 0.
      IDEFLT = 0
      JWAHL = 0
      JWELLE = 0
      JTABEL = 1
      SICMET = 0
      IEGL = 0
      VBENUG = .7
      VMULDG = 5.
      WNULL = 10.
      ISOIL = 2
      RKSTD = 4.
      ALPHDG = 2.3
      AEDG = 0.
      VBENDG = 0.
      AADG = AEDG
      VERDFK = 0.
      IMMER = 0
      EPSIUG = 0.
      EPSIDG = 0.
      KRUECK = .060
      KREG = .008
*
      FEHLER = 0
      FEHFKT = 0
      EPS = .000001
      TOL = .02
      ALMOD = .FALSE.
      HYDABR = .FALSE.
      QDIFF1 = 1.
      RETURN
      END
C
C
C
      SUBROUTINE OPTLES
*****
*
*      Einlesen der Parameter und Filenamen aus der Datei OPTPAR
*
*****
$INCLUDE:'OPTI.CMN'
*
      OPEN(N1,FILE='OPTPAR.DAT',STATUS='OLD',MODE='READ')

```

```

READ(N1,'()')
READ(N1,'(1X,F8.6,2(2X,I4),8X,I2)') EPSTEP,ISMAX,KMDT,FEHFKT
KMT = 60 / KMDT
READ(N1,'(9X,I1,7X,I2,8X,I4)') IPRO,ISMOD,WCON
READ(N1,'(11X,A12)') NETNAM
READ(N1,'(8X,I2)') AR
READ(N1,'(8X,6(3X,A12))') (REGDA(KR),KR=1,AR)
READ(N1,'(8X,6(3X,A12))') (MESSDA(KR),KR=1,AR)
READ(N1,'(8X,6(A15))') (RGANF(KR),KR=1,AR)
READ(N1,'(8X,6(A15))') (RGEND(KR),KR=1,AR)
READ(N1,1111) ( FK(KR),KR=1,AR)
READ(N1,1111) ( VBUG(KR),KR=1,AR)
READ(N1,1111) ( EPSUG(KR),KR=1,AR)
1111 FORMAT(8X,6(10X,F5.2))
READ(N1,'()')
READ(N1,'(3(1X,F5.2),2X,F5.2)') (PAR(L),L=1,4)
PAR(1) = PAR(1) / 100.
PAR(3) = PAR(3) / 100.
READ(N1,'(3(1X,F5.2),1X,F6.2)') (RDIAG(L),L=1,4)
READ(N1,'()')
READ(N1,'()')
READ(N1,1112) DEVSTO,DEVMT0,VOLTOL,DIFTOL,NORTOL
1112 FORMAT(3(4X,F4.0),2(1X,F7.6))
IF (DEVSTO.NE.0. .OR. DEVMT0.NE.0. .OR. VOLTOL.NE.0.)
+ HYDABR = .TRUE.
READ(N1,'()')
DO 10 KR=1,AR
IANF = 1
5 READ(N1,'(2X,I4,1X,F5.2)') IPANZ,PWERT
DO 6 I=IANF,IPANZ+IANF-1
PDIAG(I,KR) = PWERT
6 CONTINUE
IANF = IPANZ + IANF
IF (IPANZ.NE.0) GOTO 5
10 CONTINUE
CLOSE(N1)
RETURN
END
C
C
C
SUBROUTINE NETLES
*****
* Lesen aus der Kanalnetzdatei und Berechnung der ersten *
* Gradienten *
*****
$INCLUDE:'OPTI.CMN'
*
REAL CU(4,5),QUERGE(5)
*
DATA CU /4.,8.,12.,8., 2.,3.,4.,2., 3.,4.,5.,4.,
+ 4.,5., 6.,6., 2.,4.,6.,4./
DATA QUERGE /.005,.025,.07,.12,.2/
*
OPEN(N1,FILE=NETNAM,STATUS='OLD',MODE='READ')
READ(N1,'()')
READ(N1,7000) KARTEN
50 READ(N1,7010) ICOND

```

```

IF (ICOND.NE.'*****') GOTO 50
DO 55 J=1,KARTEN
READ(N1,'()')
55 CONTINUE
I = 0
NC = 0
NJ = 0
100 CONTINUE
I = I + 1
READ(N1,7020) ICOND,JUNC1L,HLENL
IF (ICOND.EQ.'*****') GOTO 200
READ(N1,7030) AGESL,AUNDL,NEIGKL,ICS,ICW
READ(N1,'()')
READ(N1,7040) NRS(I)
DO 120 J=5,KARTEN
READ(N1,'()')
120 CONTINUE
IF (AGESL.LE.0.0001 .OR. NRS(I).LE.0) THEN
C I = I - 1
C GOTO 100
LGU(I) = 0.
ENDIF
EXQG(I) = EXP(-7.*QUERGE(NEIGKL))
AGES(I) = AGESL
AUND(I) = AUNDL
HLEN(I) = HLENL
NRHALT(I) = ICOND
DO 150 MM=1,NJ
IF (JUNC1L.EQ.JUN(MM)) THEN
NJUNC1(I) = MM
GOTO 160
ENDIF
150 CONTINUE
NJ = NJ + 1
JUN(NJ) = JUNC1L
NJUNC1(I) = NJ
160 CONTINUE
IF (AUNDL.LE.0.0001) GOTO 100
C = CU(ICS,ICW)
BS = AGESL / HLENL * 10000.
QL = HLENL * .5
QBS = C * BS * .0625
XLFU(I) = SQRT(QL*QL + QBS*QBS)
IF (AUNDL.LE.0.25) THEN
ADDI(I) = .87 * ALOG(AUNDL)
ELSE
ADDI(I) = 7.33 * AUNDL**.306 - 6.
ENDIF
IF (I.LT.NNC) GOTO 100
200 CONTINUE
CLOSE(N1)
NC = I - 1
RETURN
*
***** FORMATANWEISUNGEN *****
*
7000 FORMAT(I1)
7010 FORMAT(A10)
7020 FORMAT(2A10,30X,F5.2)
7030 FORMAT(20X,2F5.2,7X,3I1)

```

```

7040 FORMAT(10X,I5)
      END
C
C
C
      SUBROUTINE BLOCK(AA,VM,AE,AU)
*****
*
*   Berechnet alle benoetigten Ergebnisse aus den Programmen
*   HYSRAD und KMROUT
*
*****
$INCLUDE:'OPTI.CMN'
*
      QAL   = 0.
      DQDAL = 0.
      ITG   = 0
      DO 10 L=1,4
        DQ(L) = 0.
10    CONTINUE
*
***** Parameter runden fuer das Format von HYSRAD *****
*
      AA = (NINT(AA*10000.)) / 10000.
      AE = (NINT(AE*10000.)) / 10000.
      VM = (NINT(VM*100.)) / 100.
      AU = (NINT(AU*100.)) / 100.
*
***** Berechnung der Standardeinheitsganglinie *****
***** mit den dazugehoerigen Gradienten *****
*
      CALL SEGL(AU)
      DO 100 KR=1,AR
*
***** HYSRAD zum Starten von HYSRAD erstellen *****
*
      CALL HYSRAD(AA,VM,AE,AU)
*
***** Simulationsergebnisse mit HYSRAD *****
***** und KMROUT erstellen lassen *****
*
      CALL HYSRAD
      CALL KMROUT
*
***** Bestimmung der Gradienten *****
*
      CALL RECH(AA,VM,AE)
100    CONTINUE
      IF (FEHFKT.EQ.0 .OR. ABS(FEHFKT).GE.3) RETURN
      IF (almod) THEN
        QAL = QAL / ITG
        DQDAL = DQDAL / ITG
      ELSE
        DO 200 L=1,4
          IF (FEST(L)) GOTO 200
          DQ(L) = DQ(L) / ITG
200    CONTINUE
      ENDIF
      RETURN
      END

```

```

C
C
C
      SUBROUTINE SEGL(AU)
*****
*
*   Berechnung der Standardeinheitsganglinien und der
*   dazugehoerigen Gradienten
*
*****
$INCLUDE:'OPTI.CMN'
*
      DO 200 I=1,NC
*
***** KEINE FLAECHE AN DIE HALTUNG ANGESCHLOSSEM ? *****
*
        IF (AUND(I).LT.0.0001) GOTO 200
*
***** Laufzeit *****
        TLU = AMAX1(AU + ADDI(I) - 3.*HLEN(I)/XLFU(I) , 1.)
***** Gradienten nach Schrittweite ALPHA *****
        IF (almod) THEN
          DTLUDA = R(4)
***** Gradienten nach Parameter AU *****
        ELSE
          DTLUDA = 1.
        ENDIF
        CALL EPARAM(AUND(I),TLU,DTLUDA,QP,DQPDA,TP,DTPDA,SK,DSKDA,
+         LG,DLGDA,LGMAX,CO,DCODA,.99)
        LGU(I) = LG
*
***** Einheitsganglinien bestimmen *****
***** DQIDA ist der Gradient nach ALPHA bzw. ALPHAU *****
*
        DO 170 J=1,LG
          TI = FLOAT(J)
          IF (TI.LE.TP) THEN
            QI(I,J) = QP * TI / TP
            DQIDA(I,J) = (DQPDA * TP - QP * DTPDA) * TI/(TP*TP)
          ELSE
            POT = (TP-TI) / SK
            EXPO = EXP(POT)
            QI(I,J) = QP * EXPO * CO
            DPOTDA = (DTPDA*SK + (TI-TP)*DSKDA) / (SK*SK)
            DQIDA(I,J) = (DQPDA*CO + QP*CO*DPOTDA + QP*DCODA) * EXPO
          ENDIF
          IF (i.eq.21) THEN
            IF (almod) THEN
              write(n6,*) ' dqidal(21,',j,')',dqida(21,j)
            ELSE
              IF (ipro.eq.1) write(n7,*) ' dqidau(21,',j,')',dqida(21,j)
            ENDIF
          ENDIF
170    CONTINUE
200    CONTINUE
      RETURN
      END
C
C
C

```

```

*****
SUBROUTINE EPARAM (F,TL,DTLDA,QP,DQPD,TP,DTPDA,SK,DSKDA,LGL,
+ DLGLDA,LGMAX,CORR,DCORDA,ENDW)
*****
*
* Bestimmung der Parameter der EGL über die SEGL
*
* ENDW bestimmt, wann die EGL enden soll:
* = 0.99 heisst, dass bei 1% von QP abgebrochen wird,
* = 0.95 heisst, dass bei 5% von QP abgebrochen wird.
*
*****
C
REAL K
C
DQPSDA(K,DK,TP,DTP) = (DK*(6.*TP+12.*K-4.*TP*TP) +
+ DTP*(4.*TP*K+3.*TP+6.*K)) /
+ (3.*(TP+2.*K)**3)
C
DKDA(QP,DQP,DTP) = - (F1*DQP/(QP*QP) + DTP*.5) / ENDW

F1 = F/0.006
LNC = ALOG(1.-ENDW)
TP1 = 0.5 * TL
TP = ANINT(TP1)
SK = 0.82 * TL
QPS1 = 1.0176-1./(2.13*TL)

C
DTP1DA = .5 * DTLDA
DTPDA = DTP1DA
DSKDA = .82 * DTLDA
DQPS1 = DQPSDA(SK,DSKDA,TP,DTPDA)

C
20 CONTINUE
QP1=QPS1*F1/TL
SK1=(F1/QP1-TP1*.5)/ENDW
QP2=F1/(TP*.5+ENDW*SK1)
QP=(QP1+QP2)*.5
SK=(F1/QP-TP*.5)/ENDW
HELP=1.+2.*SK/TP
QPS2=1.+1./(3.*HELP*HELP)-1./(TP*HELP)

C
DQP1DA = F1 * (DQPS1*TL - QPS1*DTLDA) / (TL*TL)
DSK1DA = DKDA(QP1,DQP1DA,DTPDA)
DQP2DA = - (DTPDA*.5 + ENDW*DSK1DA) * (QP2*QP2)/F1
DQPD = (DQP1DA + DQP2DA) * .5
DSKDA = DKDA(QP,DQPD,DTPDA)
DQPS2 = DQPSDA(SK,DSKDA,TP,DTPDA)

C
IF (ABS(QPS1-QPS2).GT.0.01) THEN
QPS1 = (QPS1 + QPS2) * .5
DQPS1 = (DQPS1 + DQPS2) * .5
GOTO 20
ENDIF

C
END = TP-LNC*SK
LGL = NINT(END)
LGMAX = MAX0(LGMAX,LGL)

C
DENDDA = DTPDA - LNC*DSKDA

```

```

DLGLDA = DENDDA
C
C***** KORREKTURFAKTOR ZUM FLAECHENAUSGLEICH
C***** NACH DER SPITZE ANGEWENDET
C
C***** ZU ERREICHENDES VOLUMEN NACH DER SPITZE
C
VSOLL = F1 - QP*(TP+1)*.5
DVSDA = -.5 * (DQPD*(TP+1) + QP*DTPDA)
C
C***** ERREICHTES VOLUMEN NACH DER SPITZE
C
VIST = 0.
DVIDA = 0.
JEND = LGL - INT(TP)
DO 30 J=1,JEND
TI = FLOAT(J)
QQI = QP * EXP(-TI/SK)
VIST = VIST + QQI
DQQIDA = (DQPD + QP*TI*DSKDA/(SK*SK)) * EXP(-TI/SK)
DVIDA = DVIDA + DQQIDA
30 CONTINUE
C
C***** KORREKTURFAKTOR
C
CORR = VSOLL / VIST
DCORDA = (DVSDA*VIST - VSOLL*DVIDA) / (VIST*VIST)
C
RETURN
END
C
C
C
SUBROUTINE HYPSPAR(AA,VM,AE,AU)
*****
*
* Erstellt eine HYPSPAR-Parameterdatei fuer das Programm HYSRAD
*
*****
$INCLUDE:'OPTI.CMN'
*
CHARACTER*40 PARFMT
*
***** Format fuer die Parameterausgabe erstellen *****
*
IF (AE.EQ.1.) THEN
IF (AA.EQ.AE) THEN
PARFMT = '(2F5.2,2F5.1,3F5.2,I5,F5.0,F5.1,F5.0)'
ELSE
PARFMT = '(3F5.2,F5.1,3F5.2,I5,F5.0,F5.1,F5.0)'
ENDIF
ELSE
PARFMT = '(7F5.2,I5,F5.0,F5.1,F5.0)'
ENDIF
VMULUG = VM
AAUG = AA * 100.
AEUG = AE * 100.
ALPHUG = AU
FAKTOR = FK(KR)
IF ( VBUG(KR).NE.0.) VBENUG = VBUG(KR)

```

```

      IF (EPSUG(KR).NE.0.) EPSIUG = EPSUG(KR)
      OPEN(N1,FILE='HYSPAR')
      WRITE(N1,4999)
      WRITE(N1,'()')
      WRITE(N1,5000) NETNAM
      WRITE(N1,5000) REGDA(KR)
      DO 4 J=1,4
        WRITE(N1,'()')
4      CONTINUE
      WRITE(N1,5025)
      WRITE(N1,5030) RGANF(KR),RGEND(KR),DOWNJC,IDEFLT,JWAHL,JWELLE,
+      JTABEL,SICMET,IEGL
      WRITE(N1,5035)
      WRITE(N1,PARFMT) VBENUG,VMULUG,AAUG,AEUG,ALPHUG,VMULDG,
+      WNULL,ISOIL,RKSTD,ALPHDG,AEDG
      WRITE(N1,5045)
      WRITE(N1,5050) VBENDG,AADG,VERDFK,IMMER,EPSIUG,
+      EPSIDG,KRUECK,KREG,FAKTOR
      CLOSE(N1)
      RETURN
*
***** FORMATANWEISUNGEN *****
*
4999 FORMAT('4 DATEINAMEN (OUTHYS,NETNAM,REGNAM,WELNAM),',
+      '3 KOMM.ZEILEN (USER, ALPHA(1..2))')
5000 FORMAT(A12)
5025 FORMAT(' TMMJJJJ HHMM TMMJJJJ HHMM DOWN DFLT',
+      ' WAHL WELL ITAB SICM IEGL')
5030 FORMAT(2A15,F5.0,6I5)
5035 FORMAT(' VBN VMUU ANTA ANTE ALPU VMUD WNUL',
+      ' SOIL KSTD ALPD ANTD')
5040 FORMAT(7F5.2,I5,F5.0,F5.1,F5.0)
5045 FORMAT(' VBD AAD VFK IMR EPSU EPSD KRUE',
+      ' KREG RFAK')
5050 FORMAT(3F5.0,I5,2F5.0,2F5.3,10X,F5.1)
      END
C
C
C
*****
      SUBROUTINE RECH(AA,VM,AE)
*****
*      Bestimmung der Fehlerfunktion und der Gradienten
*
*****
$INCLUDE:'OPTI.CMN'
*
      REAL      REG(NNC),DQ1(4)
*
***** erstes Einlesen aus Regen-, Mess-, *****
***** Ausgabe- und Protokolldatei *****
*
      REGNAM = REGDA(KR)
      CALL REGLES(REGNAM)
      MDAT = MESSDA(KR)
      OPEN(N4,FILE=MDAT,STATUS='OLD',MODE='READ')
      READ(N4,'()')
      READ(N4,'(19X,A10)') MCOND
      I4 = 1

```

```

      DO 5 J=1,NC
      IF (MCOND.EQ.NRHALT(J)) THEN
        MESS = J
        GOTO 6
      ENDIF
5      CONTINUE
6      CONTINUE
      READ(N4,'()')
      OPEN(N5,FILE='OUTKM.DAT',STATUS='OLD',MODE='READ')
      I5 = 1
      DO 7 J=1,3
        READ(N5,'()')
7      CONTINUE
*
***** Evtl. Zwischenergebnisse (KMROUT-Ausgabe) *****
***** in einer Extradatei merken *****
*
      IF (MOD(ISZ,ISMOD).EQ.0) THEN
        AUSGEB = .TRUE.
        IF (ISZ.LT.10) THEN
          WRITE(OPKMO,1221) ISZ,KR
        ELSE
          WRITE(OPKMO,1222) ISZ,KR
        ENDIF
1221      FORMAT('OP',I1,'R',I1,'.KMO')
1222      FORMAT('OP',I2,'R',I1,'.KMO')
        OPEN(N8,FILE=OPKMO)
        WRITE(N8,1223) MCOND
1223      FORMAT(1X,79('-',)/' HALTUNG ',A10,/,1X,79('-',))
      ENDIF
*
      CALL PROLES
*
***** Benetzungsverluste und Muldenauffuellgrad *****
***** auf Anfangswert setzen *****
*
      DO 8 J=1,NC
        VBU(J) = VBENUG
        EPSIU(J) = EPSIUG
        LGUN(J) = LGU(J)
*
***** Gradienten auf Null setzen *****
      if (almod) then
        DZNDAL(J) = 0.
        DMNDAL(J) = 0.
        DQNDAL(J) = 0.
      else
        DZNDAA(J) = 0.
        DMNDAA(J) = 0.
        DQNDAA(J) = 0.
        DZNDVM(J) = 0.
        DMNDVM(J) = 0.
        DQNDVM(J) = 0.
        DZNDAE(J) = 0.
        DMNDAE(J) = 0.
        DQNDAE(J) = 0.
        DZNDVDM(J) = 0.
        DMNDVDM(J) = 0.
        DQNDVDM(J) = 0.
        DZNDVDM(J) = 0.
        DMNDVDM(J) = 0.
        DQNDVDM(J) = 0.
      endif

```

```

8   CONTINUE
   QALL = 0.
   DQDAL1 = 0.
   ESUM = 0.
   QSSUM = 0.
   DO 9 L=1,4
     DQ1(L) = 0.
9   CONTINUE
   IT = 0
   IEND = 0
*
***** SCHLEIFE REGENDATEN INTERVALLWEISE EINLESEN *****
*
100  CONTINUE
     READ(N2,REGFMT) (REG(J),J=1,NANZ)
     DO 10 I=1,NANZ
       REG(I) = REG(I) * FK(KR)
10   CONTINUE
*
***** Regendaten den Haltungen zuordnen
*
     DO 20 NR=1,NC
       IF (LGR(NR).EQ.0) GOTO 20
       I = NNRS(NR)
       IF (REG(I).LE.0.) THEN
         RE(NR) = 0.
       ELSE
         RE(NR) = REG(I) / IB
       ENDIF
20  CONTINUE
120  CONTINUE
     IF (IEND.EQ.1 .AND. QTOT.LT.EPS) GOTO 500
     IF (ISZ.EQ.1 .AND. .NOT.ALMOD) THEN
       QMAXAK = 1.
     ELSE
       QMAXAK = QMAX(KR)
     ENDIF
*
***** SCHLEIFE UEBER DIE ZEIT EINES REGENINTERVALLES *****
*
     DO 400 IBB=1,IB
       IT = IT + 1
*
***** Gesamtwertezaehler *****
*
       ITG = ITG + 1
*
***** Lesen von Messwert und Simulationswert des Zeitschrittes *****
*
       IF (I4.NE.0) READ(N4,3000,END=410) QABM
*
***** Maximum, Mittelwert und Volumen *****
***** der Messganglinie bestimmen *****
*
       IF (ISZ.EQ.1 .AND. .NOT.ALMOD) THEN
         IF (QABM.GT.QMAX(KR)) QMAX(KR) = QABM
         QVOL(KR) = QVOL(KR) + QABM
       ENDIF
       GOTO 420
*

```

```

***** Ende bei der Messdatei erreicht *****
*
410  CONTINUE
     CLOSE(N4)
     QABM = 0.
     I4 = 0
420  CONTINUE
     IF (I5.NE.0) THEN
       READ(N5,3100) STERN
       IF (STERN.NE.'*') THEN
         BACKSPACE(N5)
         READ(N5,3000) QABS
*
***** Ausgabe des Simulationswertes auf die Extradatei *****
       IF (AUSGEB) THEN
         WRITE(N8,1224) IT,QABS
       ENDIF
     ELSE
       I5 = 0
       CLOSE(N5)
       QABS = 0.
     ENDIF
     ELSE
       QABS = 0.
     ENDIF
     QSSUM = QSSUM + QABS
     E = (QABM - QABS)
     WFAKT = QABM / QMAXAK
     ESUM = ESUM + ABS(E) * WFAKT
*
***** neue Konstanten bei KMROUT ? *****
     IF ((ICYC+2) .EQ. IT) CALL NEUMAT
     QTOT = 0.
*
***** alte Gradienten der Zuflussganglinie merken *****
*
     DO 295 NR=1,NC
       if (almod) then
         DZADAL(NR) = DZNDAL(NR)
         DMADAL(NR) = DMNDAL(NR)
         DQADAL(NR) = DQNDAL(NR)
       else
         DZADAU(NR) = DZNDAU(NR)
         DMADAU(NR) = DMNDAU(NR)
         DQADAU(NR) = DQNDAU(NR)
         DZADAA(NR) = DZNDAA(NR)
         DMADAA(NR) = DMNDAA(NR)
         DQADAA(NR) = DQNDAA(NR)
         DZADV(NR) = DZNDV(NR)
         DMADV(NR) = DMNDV(NR)
         DQADV(NR) = DQNDV(NR)
         DZADAE(NR) = DZNDAE(NR)
         DMADAE(NR) = DMNDAE(NR)
         DQADAE(NR) = DQNDAE(NR)
       endif
       C   qzualt(nr) = qzu(nr)
       C   qzmalt(nr) = qzm(nr)
       C   qabalt(nr) = qab(nr)
295  CONTINUE
*

```

***** SCHLEIFE UEBER DIE HALTUNGEN *****

```

*
  DO 300 NR=1,NC
*
***** Gradienten aus HYSRAD und KMROUT bestimmen *****
*
  CALL GRADI(AA,VM,AE)
  300 CONTINUE
*
***** Bestimmung von DELTAQ *****
*
  PD = PDIAG(IT,KR)
  IF (FEHFKT.EQ.0 .OR. FEHFKT.EQ.-5) WFAKT = 1.
*
***** Fehlerfunktion ist die Summe der Fehlerquadrate *****
*

```

```

  IF (FEHFKT.GE.0) THEN
    if (almod) then
      QAL1 = QAL1 + E * PD * E * WFAKT
      DQDAL1 = DQDAL1 - 2.* DQNDAL(MESS) * PD * E * WFAKT
C      write(n6,*) ' dqndal(21,'it,')',dqndal(21)
C      write(n6,*) ' dqdal(21,'it,')',dqdal
    else
      DQ1(1) = DQ1(1) - 2.* DQNDAA(MESS) * PD * E * WFAKT
      DQ1(2) = DQ1(2) - 2.* DQNDVM(MESS) * PD * E * WFAKT
      DQ1(3) = DQ1(3) - 2.* DQNDAAE(MESS) * PD * E * WFAKT
      DQ1(4) = DQ1(4) - 2.* DQNDAAU(MESS) * PD * E * WFAKT
      if (ipro.eq.1) then
        write(n7,*) ' dqndaa(21,'it,')',dqndaa(21)
        write(n7,*) ' dqndvm(21,'it,')',dqndvm(21)
        write(n7,*) ' dqndae(21,'it,')',dqndae(21)
        write(n7,*) ' dqndau(21,'it,')',dqndau(21)
        write(n7,*) ' dqdaa(21,'it,')',dq(1)
        write(n7,*) ' dqdvm(21,'it,')',dq(2)
        write(n7,*) ' dqdae(21,'it,')',dq(3)
        write(n7,*) ' dqdau(21,'it,')',dq(4)
      endif
C      if (isz.eq.1.or.isz.eq.2.or.isz.eq.10.or.isz.eq.20)
C      + write(n6,*) ' IT',IT,' PD',PD,' E',E
    endif
  ELSE
    if (almod) then
      IF (E.EQ.0.) THEN
        VZQABM = 0.
      ELSE IF (E.LT.0.) THEN
        VZQABM = - WFAKT
      ELSE
        VZQABM = WFAKT
      ENDIF
      QAL1 = QAL1 + E * VZQABM
      DQDAL1 = DQDAL1 - DQNDAL(MESS) * VZQABM
    else
      IF (E.EQ.0.) THEN
        VZQABM = 0.
      ELSE IF (E.LT.0.) THEN
        VZQABM = - WFAKT
      ELSE
        VZQABM = WFAKT
      ENDIF
      DQ1(1) = DQ1(1) - DQNDAA(MESS) * VZQABM

```

```

      DQ1(2) = DQ1(2) - DQNDVM(MESS) * VZQABM
      DQ1(3) = DQ1(3) - DQNDAAE(MESS) * VZQABM
      DQ1(4) = DQ1(4) - DQNDAAU(MESS) * VZQABM
    endif
  ENDIF
400 CONTINUE
  IF (IEND.EQ.1) GOTO 120
  READ(N2,'(A1)') STERN
  IF (STERN.NE.'**') THEN
    GOTO 100
  ELSE
    IEND = 1
    CLOSE(N2)
    GOTO 120
  ENDIF
500 CONTINUE
  ITW = IT + 1
*
***** Lesen der restlichen Mess- und Simul.daten *****
***** (abfallender Ast nach Ende des Regenzuflusses) *****
*
590 CONTINUE
  IF (I4.NE.0) THEN
    READ(N4,3000,END=600) QABM
    IF (ISZ.EQ.1 .AND. .NOT.ALMOD) THEN
      IF (QABM.GT.QMAX(KR)) QMAX(KR) = QABM
      QVOL(KR) = QVOL(KR) + QABM
    ENDIF
  ENDIF
  GOTO 605
600 CONTINUE
  CLOSE(N4)
  QABM = 0.
  I4 = 0
605 CONTINUE
  IF (I5.NE.0) THEN
    READ(N5,3100,END=610) STERN
    IF (STERN.NE.'**') THEN
      BACKSPACE(N5)
      READ(N5,3000) QABS
    ENDIF
*
***** Ausgabe des Simulationswertes auf die Extradatei *****
    IF (AUSGEB) THEN
      WRITE(N8,1224) ITW,QABS
    ENDIF
    GOTO 650
  ENDIF
610 CONTINUE
  CLOSE(N5)
  IF (AUSGEB) CLOSE(N8)
  QABS = 0.
  I5 = 0
  ENDIF
650 CONTINUE
  IF (I4.NE.0 .OR. I5.NE.0) THEN
    QSSUM = QSSUM + QABS
    E = (QABM - QABS)
    WFAKT = QABM / QMAXAK
    ESUM = ESUM + ABS(E) * WFAKT

```

```

      if (almod) then
        IF (FEHFKT.EQ.0 .OR. FEHFKT.EQ.-5) WFAKT = 1.
        IF (FEHFKT.GE.0) THEN
          QAL1 = QAL1 + E * E * WFAKT
        ELSE
          IF (E.GE.0.) THEN
            QAL1 = QAL1 + E * WFAKT
          ELSE
            QAL1 = QAL1 - E * WFAKT
          ENDIF
        ENDIF
      endif
      ITW = ITW + 1
      ITG = ITG + 1
      GOTO 590
    ENDIF

*
***** Mittelwert bestimmen *****
*
      IF (ISZ.EQ.1 .AND. .NOT.ALMOD) THEN
        QMIT = QVOL(KR) / ITW
        CDEVS(KR) = WCON / QMAX(KR)
        CDEVN(KR) = WCON / QMIT
      ENDIF

*
***** Wichtungsfaktoren bestimmen *****
*
      IF (ABS(FEHFKT).EQ.1) THEN
        WICHT = CDEVS(KR)
      ELSE IF (ABS(FEHFKT).EQ.2) THEN
        WICHT = CDEVN(KR)
      ELSE IF (ABS(FEHFKT).EQ.3) THEN
        WICHT = WCON
      ENDIF
      IF (FEHFKT.EQ.0 .OR. FEHFKT.EQ.-5) THEN
        WICHT = 1.
      ELSE IF (ISZ.EQ.1 .AND. .NOT.ALMOD) THEN
        WICHT = WICHT / QMAX(KR)
      ENDIF

*
***** Aufsummieren fuer die gesamten Ereignisse *****
*
      if (almod) then
        QAL = QAL + QAL1 * WICHT
        DQDAL = DQDAL + DQDAL1 * WICHT
      else
        DO 700 L=1,4
          IF (FEST(L)) GOTO 700
          DQ(L) = DQ(L) + DQ1(L) * WICHT
        700 CONTINUE
      end

*
***** hydrologische Fehlermasse berechnen *****
*
      DEVS(KR) = ESUM * CDEVS(KR) * 100. / ITW
      DEVN(KR) = ESUM * CDEVN(KR) * 100. / ITW
      IF (ISZ.EQ.1 .AND. .NOT.ALMOD) THEN
        DEVS(KR) = DEVS(KR) / QMAX(KR)
        DEVN(KR) = DEVN(KR) / QMAX(KR)
      ENDIF
      VOLD(KR) = 100. * QSSUM / QVOL(KR) - 100.

```

```

      WRITE(N6,1225) KR,DEVS(KR)
      WRITE(N6,1226) KR,DEVN(KR)
      WRITE(N6,1227) KR,VOLD(KR)
    endif
    RETURN

*
***** FORMATANWEISUNGEN *****
*
      1224 FORMAT(' KMR',I5,10X,F10.6)
      1225 FORMAT(' REGEN NR: ',I2,' DEVS: ',F6.2,' %')
      1226 FORMAT(' REGEN NR: ',I2,' DEVN: ',F6.2,' %')
      1227 FORMAT(' REGEN NR: ',I2,' VOLD: ',F6.2,' %')
      3000 FORMAT(19X,F10.6)
      3100 FORMAT(A1)
    END

C
C
C
      SUBROUTINE REGLES(REGNAM)
*****
*
*   Erstes Lesen aus der Regendatei
*
*****
$INCLUDE:'OPTI.CMN'
*
      OPEN(N2,FILE=REGNAM,STATUS='OLD',MODE='READ')
      READ(N2,'()')
      READ(N2,'()')
      READ(N2,'()')
      READ(N2,9005) NANZ,IB,IDIM
      IB = MAX0(IB,1)
      WRITE(REGFMT,1201) -IDIM
      1201 FORMAT('(16F5.',I1,')')

*
***** Schachtzuordnung *****
*
      READ(N2,9010) (IRS(I),I=1,NANZ)
      DO 10 I=1,NANZ
        DO 5 NR=1,NC
          IF (LGU(NR).EQ.0) GOTO 5
          IF (NRS(NR).EQ.IRS(I)) NNRS(NR) = I
        5 CONTINUE
      10 CONTINUE
      READ(N2,'()')
      RETURN

*
***** FORMATANWEISUNGEN *****
*
      9000 FORMAT(2A15)
      9005 FORMAT(20X,3I5)
      9010 FORMAT(16I5)
    END

C
C
C
      SUBROUTINE PROLES
*****
*
*   Erstes Lesen aus der Protokolldatei von KMRROUT
*
*****

```

```

*
*****
$INCLUDE:'OPTI.CMN'
*
      OPEN(N3,FILE='KM.PRO',STATUS='OLD',MODE='READ')
      READ(N3,'()')
*
***** Verknuepfungstabelle *****
      DO 20 I=1,NC
        READ(N3,6000) (ICHAZU(I,J),J=1,8)
      20  CONTINUE
*
***** KM-Matrizen fuer den Anfangszustand *****
      30  READ(N3,'(1X,A10)') IDUM
          IF (IDUM.NE.'  HALTUNG') GOTO 30
          READ(N3,'()')
          DO 40 I=1,NC
            READ(N3,6100) RCONS
            KMCON1(I) = 1. - EXP(-1./RCONS)
            KMCON2(I) = 1. - RCONS*KMCON1(I)
          40  CONTINUE
              ICHA = 1
              ICHE = NC
              IPREND = 0
*
***** erste Konstantenveraenderung heraussuchen *****
      CALL NEUMAT
      RETURN
*
***** FORMATANWEISUNGEN *****
*
      6000 FORMAT(5X,8I4)
      6100 FORMAT(61X,F12.8)
      END
C
C
C
      SUBROUTINE NEUMAT
*****
*
*      Berechnung der neuen Matrizen bei Konstantenveraenderungen
*
*****
$INCLUDE:'OPTI.CMN'
*
      CHARACTER*10      IDUM2
*
      CALL KMAT
      IF (IPREND.EQ.1) RETURN
*
***** nach den naechsten Konstantenveraenderungen suchen *****
      50  READ(N3,6200,END=100) IDUM
          IF (IDUM.NE.'** Haltung') GOTO 50
          BACKSPACE(N3)
      60  READ(N3,6400,END=100) I,RCONS
          KMCON1(I) = 1. - EXP(-1./RCONS)
          KMCON2(I) = 1. - RCONS*KMCON1(I)
      65  READ(N3,6250,END=100) IDUM,IDUM2
          IF (IDUM2.EQ.'neue KM-Ko') THEN
            BACKSPACE(N3)

```

```

      READ(N3,6300,END=100) ICYC,IKNEU,ICHA,ICHE
      IF (MOD(ICYC,KMT).NE.0) GOTO 50
      ELSE IF (IDUM.EQ.'** Haltung') THEN
        BACKSPACE(N3)
        GOTO 60
      ELSE
        GOTO 65
      ENDIF
      RETURN
*
      100  CONTINUE
          IPREND = 1
          CLOSE(N3)
          RETURN
*
***** FORMATANWEISUNGEN *****
*
      6200 FORMAT(4X,A10)
      6250 FORMAT(4X,A10,12X,A10)
      6300 FORMAT(13X,I5,2X,I5,38X,I5,4X,I5)
      6400 FORMAT(28X,I5,13X,F12.8)
      END
C
C
C
      SUBROUTINE KMAT
*****
*
*      PROGRAMM ZUR BERECHNUNG DER MATRIZEN FUER DAS KALININ-
*      MILJUKOV-VERFAHREN
*
-----
*      QMAT - MATRIX FUER Q ZUM ZEITPKT T
*      UMAT - MATRIX FUER U ZUM ZEITPKT T
*      UMAT1- MATRIX FUER U ZUM ZEITPKT T + 1
*
*****
C
$INCLUDE:'OPTI.CMN'
C
-----
C
      INTEGER      IKONTR(NNC)
*
***** Diagonale besetzen *****
      DO 25 I=1,NC
        QMAT(I,I) = 1. - KMCON1(I)
        UMAT(I,I) = KMCON1(I) - KMCON2(I)
        IU = NC - I + 1
        UMAT(IU,IU+1) = KMCON2(I)
      25  CONTINUE
C
      DO 100 IZ=ICHA,ICHE
        DO 111 IZAEHL=1,NNC
          IKONTR(IZAEHL) = 0
        111  CONTINUE
*
***** Berechnung der Matrizenzeilen fuer den angegebenen *****
***** Beeinflussungsbereich ICHA - ICHE einer Haltung IZ *****
*
      IF (ICHAZU(IZ,1).EQ.IZ) GOTO 100

```

```

*
***** keine Anfangshaltung - Zeilen & Spalten Index Bestimmung *****
*
      DO 40 IC=1,8
      IF (IC .EQ. 1) THEN
*
***** erster Zufluss zur Haltung IZ *****
      IZA = ICHAZU(IZ,1)
      ELSE
*
***** zusätzlicher Zufluss zur Haltung IZ *****
      IZA = ICHAZU(IZ,IC)
      IF (IZA .LE. 0) GOTO 100
      END IF
*
***** Anfangshaltung herausuchen *****
      ISPA = ICHAZU(IZA,1)
10      IF (ISPA.NE.ICHAZU(ISPA,1)) THEN
      ISPA = ICHAZU(ISPA,1)
      GOTO 10
      ENDIF
C
      KE = IZA
      IZO = NC - IZA + 1
C
      DO 30 K=ISPA,KE
      IF (IKONTR(K).EQ.1 .OR. QMAT(IZA,K).EQ.0.) GOTO 30
      QMAT(IZ,K) = KMCON2(IZ) * QMAT(IZA,K)
      IF (K.EQ.IZA) QMAT(IZ,K) = KMCON1(IZ)-KMCON2(IZ)+QMAT(IZ,K)
30      CONTINUE
C
      DO 33 K=ISPA,KE
      IF (IKONTR(K) .EQ. 1) GOTO 33
      UMAT(IZ,K) = KMCON2(IZ) * UMAT(IZA,K)
33      CONTINUE
C
      IZDO = NC - IZ + 1
      DO 36 K=ISPA,KE
      ISP = NC - K + 2
      IF (IKONTR(K) .EQ. 1) GOTO 36
      IKONTR(K) = 1
      UMAT(IZDO,ISP) = KMCON2(IZ) * UMAT(IZO,ISP)
36      CONTINUE
C
40      CONTINUE
C
100     CONTINUE
C
      RETURN
      END
C
C
C
      SUBROUTINE GRADI(AA,VM,AE)
*****
*
* Berechnung der Gradienten aus HYSRAD und KMROUT
*
*****
$INCLUDE:'OPTI.CMN'

```

```

*
***** Vom Quergefaelle abhaengigen Muldenverlust berechnen *****
*
      EXPQG = EXPQG(NR)
      VMQG = VM * EXPQG
      IF (LGUN(NR).EQ.0) THEN
      if (almod) then
      DZNDAL(NR) = 0.
      else
      DZNDAL(NR) = 0.
      DZNDAA(NR) = 0.
      DZNDVM(NR) = 0.
      DZNDAE(NR) = 0.
      endif
      GOTO 290
      ENDIF
      IF (IEND.EQ.1) GOTO 150
*
***** Benetzungsverluste abziehen
*
      IF (VBU(NR).GT.0.) THEN
      VBU(NR) = VBU(NR) - RE(NR)
      IF (VBU(NR).GT.0.) THEN
      RE(NR) = 0.
      ELSE
      RE(NR) = - VBU(NR)
      VBU(NR) = 0.
      ENDIF
      ENDIF
*
***** Gradienten zur Abflussbildung *****
*
      MT = MOD(IT, LGU(NR))
      IF (RE(NR).LE.EPS) THEN
      RI(NR,MT) = 0.
      if (almod) then
      DRDAL(NR,MT) = 0.
      else
      DRDAA(NR,MT) = 0.
      DRDVM(NR,MT) = 0.
      DRDAE(NR,MT) = 0.
      endif
      ELSE IF (VMQG.LE.0.01) THEN
      RI(NR,MT) = AE * RE(NR)
      if (almod) then
      DRDAL(NR,MT) = R(3) * RE(NR)
      else
      DRDAA(NR,MT) = 0.
      DRDVM(NR,MT) = 0.
      DRDAE(NR,MT) = RE(NR)
      endif
      ELSE
      CEXPU = (AE - AA) / VMQG
      IF (CEXPU.LE.EPS) THEN
      WRITE(*,*) ' SCHWELLWERTMETHODE NICHT EINGEBAUT'
      STOP
      ELSE
      EXPO = EXP(-CEXPU*RE(NR))
      X = EPSIU(NR)
      EPSIU(NR) = 1. - (1. - X) * EXPO

```

```

      RI(NR,MT) = AE * RE(NR) - VMQG * (EPSIU(NR)-X)
***** Gradienten *****
      RV      = RE(NR) / VMQG
      if (almod) then
        DXDAL      = DEPSAL(NR)
        DEPSAL(NR)  = (DXDAL - (1.-X) * RV
          +          * (R(1)+R(2)*EXPQG*CEXPV-R(3))) * EXPO
        DRDAL(NR,MT) = R(3) * RE(NR) - R(2)*EXPQG*(EPSIU(NR)-X)
          +          - VMQG * (DEPSAL(NR)-DXDAL)
      C      if (nr.eq.21) write(n6,*) ' drdal(21,'mt,')',drdal(21,mt)
      else
        DXDAA      = DEPSAA(NR)
        DXDVM      = DEPSVM(NR)
        DXDAE      = DEPSAE(NR)
        DEPSAA(NR)  = (DXDAA - (1.-X) * RV) * EXPO
        DEPSVM(NR)  = (DXDVM - (1.-X) * CEXPV*EXPQG*RV) * EXPO
        DEPSAE(NR)  = (DXDAE + (1.-X) * RV) * EXPO
        DRDAA(NR,MT) = - VMQG * (DEPSAA(NR) - DXDAA)
          +          + VMQG * (DXDVM - DEPSVM(NR))
        DRDVM(NR,MT) = (X - EPSIU(NR)) * EXPQG +
          +          VMQG * (DXDVM - DEPSVM(NR))
        DRDAE(NR,MT) = RE(NR) - VMQG * (DEPSAE(NR) - DXDAE)
      if (ipro.eq.1) then
        if (nr.eq.21) write(n7,*) ' drdaa(21,'mt,')',drdaa(21,mt)
        if (nr.eq.21) write(n7,*) ' drdvm(21,'mt,')',drdvm(21,mt)
        if (nr.eq.21) write(n7,*) ' drdae(21,'mt,')',drdae(21,mt)
      endif
      endif
    ENDIF
  ENDIF
150  CONTINUE
*
***** Gradienten der Abflusskonzentration *****
*
      IF (IEND.EQ.0) MRE = IT
      QSUMA = 0.
      QSUMAA = 0.
      QSUMVM = 0.
      QSUMAE = 0.
      QSUM = 0.
      DO 250 I=1,MRE
        J = IT - I + 1
        MI = MOD(I, LGU(NR))
        IF (J.GT.LGU(NR)) GOTO 250
        QINRJ = QI(NR,J)
        if (almod) then
          TERMA = DRDAL(NR,MI)*QINRJ + RI(NR,MI)*DQIDA(NR,J)
          QSUMA = QSUMA + TERMA
        else
          TERMA = RI(NR,MI) * DQIDA(NR,J)
          TERMAA = DRDAA(NR,MI) * QINRJ
          TERMVM = DRDVM(NR,MI) * QINRJ
          TERMAE = DRDAE(NR,MI) * QINRJ
          QSUMA = QSUMA + TERMA
          QSUMAA = QSUMAA + TERMAA
          QSUMVM = QSUMVM + TERMVM
          QSUMAE = QSUMAE + TERMAE
        endif
        TERM = RI(NR,MI) * QINRJ
        QSUM = QSUM + TERM
250  CONTINUE

```

```

      if (almod) then
        DZNDAL(NR) = QSUMA * .001
        if (nr.eq.21) write(n6,*) ' dzndal(21,'it,')',dzndal(21)
      else
        DZNDAA(NR) = QSUMA * .001
        DZNDAA(NR) = QSUMAA * .001
        DZNDVM(NR) = QSUMVM * .001
        DZNDAE(NR) = QSUMAE * .001
        if (ipro.eq.1) then
          if (nr.eq.21) write(n7,*) ' dzndaa(21,'it,')',dzndaa(21)
          if (nr.eq.21) write(n7,*) ' dzndvm(21,'it,')',dzndvm(21)
          if (nr.eq.21) write(n7,*) ' dzndae(21,'it,')',dzndae(21)
          if (nr.eq.21) write(n7,*) ' dzndau(21,'it,')',dzndau(21)
        endif
      endif
      QQI = QSUM * 0.001
      qzu(nr) = qqi
      QTOT = QTOT + QQI
      IF (IEND.EQ.1 .AND. QQI.LT.EPS) LGUN(NR) = 0
*
***** Berechnung der Gradienten aus dem Programm KMROUT *****
*
290  CONTINUE
*
***** mittlere Zufluesse fuer KMROUT *****
*
      if (almod) then
        DMNDAL(NR) = (DZNDAL(NR) + DZADAL(NR))*0.5
        if (nr.eq.21) write(n6,*) ' dmndal(21,'it,')',dmndal(21)
      else
        DMNDAU(NR) = (DZNDAA(NR) + DZADAA(NR))*0.5
        DMNDAA(NR) = (DZNDAA(NR) + DZADAA(NR))*0.5
        DMNDVM(NR) = (DZNDVM(NR) + DZADVM(NR))*0.5
        DMNDAE(NR) = (DZNDAE(NR) + DZADAE(NR))*0.5
        if (ipro.eq.1) then
          if (nr.eq.21) write(n7,*) ' dmndaa(21,'it,')',dmndaa(21)
          if (nr.eq.21) write(n7,*) ' dmndvm(21,'it,')',dmndvm(21)
          if (nr.eq.21) write(n7,*) ' dmndae(21,'it,')',dmndae(21)
          if (nr.eq.21) write(n7,*) ' dmndau(21,'it,')',dmndau(21)
        endif
      endif
      endif
      qzm(nr) = (qzu(nr) + qzualt(nr))*0.5
      CALL DKMROU(NR)
      RETURN
      END
C
C
C
      SUBROUTINE DKMROU(INDEX)
*****
*
*   Berechnung der Gradienten fuer das Programm KMROUT
*
*****
$INCLUDE: 'OPTI.CMN'
*
      KN      = NC - INDEX + 1
      QSUMA = 0.
      QSUMAA = 0.
      QSUMVM = 0.
      QSUMAE = 0.

```

```

C      qsum = 0.
      DO 100 I=1,INDEX
*
***** Matrizen *****
      QMINI = QMAT(INDEX,I)
      UMINI = UMAT(INDEX,I)
      K      = NC - I + 2
      UMKNK = UMAT(KN,K)
*
***** Gradienten *****
*
      if (almod) then
        TERMA = QMINI*DQADAL(I) + UMINI*DMADAL(I) + UMKNK*DMNDAL(I)
        QSUMA = QSUMA + TERMA
      else
        TERMA = QMINI*DQADAU(I) + UMINI*DMADAU(I) + UMKNK*DMNDAU(I)
        TERMAA = QMINI*DQADAA(I) + UMINI*DMADAA(I) + UMKNK*DMNDAA(I)
        TERMVM = QMINI*DQADV(I) + UMINI*DMADV(I) + UMKNK*DMNDV(I)
        TERMAE = QMINI*DQADAE(I) + UMINI*DMADAE(I) + UMKNK*DMNDAE(I)
        QSUMA = QSUMA + TERMA
        QSUMAA = QSUMAA + TERMAA
        QSUMVM = QSUMVM + TERMVM
        QSUMAE = QSUMAE + TERMAE
      endif
      term = qmini*qabalt(i) + umini*qzmalt(i) + umknk*qzm(i)
      qsum = qsum + term
100  CONTINUE
C      qab(index) = qsum
C      if(.not.almod.and. index.eq.21) write(n6,*) ' qab(21)',qsum,
C      + ' qabalt',qabalt(21),' qzm',qzm(21),' qzmalt',qzmalt(21)
      if (almod) then
        DQNDAL(INDEX) = QSUMA
      else
        DQNDAL(INDEX) = QSUMA
        DQNDAA(INDEX) = QSUMAA
        DQNDVM(INDEX) = QSUMVM
        DQNDAE(INDEX) = QSUMAE
      endif
      RETURN
END
C
C
C      SUBROUTINE NEUPAR
*****
*      Berechnung der neuen Parameter
*
*****
$INCLUDE:'OPTI.CMN'
      REAL      VZR(4)
*
***** Berechnung der Richtungsvektoren *****
*
      DO 10 L=1,4
        ALTR = R(L)
        R(L) = - RDIAG(L) * DQ(L)
        VZR(L) = ALTR * R(L)
10  CONTINUE
*

```

```

***** Bestimmung der Schrittweite *****
*
      ALMOD = .TRUE.
      CALL STEP
      ALMOD = .FALSE.
      IF (ALPHAM.LE.0.005) FEHLER = FEHLER + 1
      IF (FEHLER.GT.10) THEN
        WRITE(*,*) ' SCHRITTWEITEN ZU KLEIN! STOP!'
        IF (IPRO.EQ.1 .or.ipro.eq.0)
          + WRITE(N6,'(A)') ' SCHRITTWEITEN ZU KLEIN!'
      C      CALL AUSKL
        STOP
      ENDIF
*
***** neue Parameter *****
*
      DO 50 L=1,4
*
***** Wird ein Parameter festgehalten ? *****
*
        IF (FEST(L)) GOTO 50
*
***** alte Parameter *****
*
        ALTPAR(L) = PAR(L)
*
***** neue Parameter *****
*
        PAR(L) = PAR(L) + ALPHAM * R(L)
        WRITE(N6,'(F10.5)') PAR(L)
        IF (PAR(L).LT.0.) THEN
          PAR(L) = 0.
          WRITE(N6,*) ' PARAMETER NICHT MEHR',
          + ' IN DEN GRENZEN (nach STEP)'
          WRITE(N6,*) ' PAR(1-4)',(PAR(J),J=1,4)
        ENDIF
50  CONTINUE
        IF (NGRENZ(1).GE.3) PAR(1) = MIN(PAR(3)-.0001,0.9999)
        IF (PAR(3).GT.1.0) THEN
          PAR(3) = 1.0
          WRITE(N6,*) ' PARAMETER NICHT MEHR',
          + ' IN DEN GRENZEN (nach STEP)'
          WRITE(N6,*) ' PAR(1-4)',(PAR(L),L=1,4)
        ENDIF
        IF (NGRENZ(1).LT.3 .AND. PAR(1).GT.(PAR(3)-.0001)) THEN
          IF (PAR(1).LE.0.0001) THEN
            PAR(1) = 0.
            PAR(3) = 0.0001
          ELSE
            PAR(1) = PAR(3) - .0001
          ENDIF
          WRITE(N6,*) ' PARAMETER NICHT MEHR',
          + ' IN DEN GRENZEN (nach STEP)'
          WRITE(N6,*) ' PAR(1-4)',(PAR(L),L=1,4)
        ENDIF
        DO 70 L=1,4
*
***** wird ein Parameter festgehalten ? *****
*
          IF (FEST(L)) THEN

```

```

      GOTO 70
    ELSE
      IF (NGRENZ(L).GE.3 .OR. NGRENZ(L).LE.-3
+      .OR. NTOL(L).GE.3) FEST(L) = .TRUE.
    ENDIF
  *
  ***** oszilliert ein Parameter um einen Wert ? *****
  *
    IF (NTOL(L).GE.3) THEN
      PAR(L) = PARM(L)
      GOTO 70
    ENDIF
  *
  ***** zuviele Grenzüberschreitungen ? *****
  *
    IF (NGRENZ(L).LE.-3) THEN
      PAR(L) = 0.
      GOTO 70
    ENDIF
    IF (NGRENZ(L).GE.3) THEN
      IF (L.EQ.3) PAR(3) = 1.
      GOTO 70
    ENDIF
  *
  ***** Grenzüberschreitungen zaehlen *****
  *
    IF (PGRENZ(L).LT.0) NGRENZ(L) = NGRENZ(L) - 1
    IF (PGRENZ(L).GT.0) NGRENZ(L) = NGRENZ(L) + 1
  *
  ***** Oszillation zaehlen *****
  ***** (nur bei Vorzeichenwechsel der Richtungsvektoren) *****
  *
    IF (NTOL(L).GE.0) THEN
      IF (VZR(L).LT.0.) THEN
        PARDIF = ABS(PARM(L)-PAR(L))
        TOLA = TOL * RDIAG(L) / 2.
        IF (PARDIF.LT.TOLA) THEN
          NTOL(L) = NTOL(L) + 1
        ELSE
          NTOL(L) = -1
        ENDIF
      ELSE
        NTOL(L) = -1
      ENDIF
    ENDIF
    IF (VZR(L).LT.0.) THEN
      PARDIF = ABS(ALTPAR(L)-PAR(L))
      TOLA = TOL * RDIAG(L)
      IF (PARDIF.LT.TOLA) THEN
        IF (NTOL(L).LT.0) THEN
          NTOL(L) = 0
          PARM(L) = (ALTPAR(L) + PAR(L)) / 2.
        ENDIF
      ELSE
        NTOL(L) = -1
      ENDIF
    ENDIF
  70 CONTINUE
C
  RETURN
END

```

```

SUBROUTINE STEP1
*****
*
*   Bestimmung der optimalen Schrittweite
*
*****
$INCLUDE:'OPTI.CMN'
C
  REAL      PARA(4),ALPH(0:2),QA(0:2),DQDA(0:2)
C
  IF (IPRO.EQ.1)
C
+  WRITE(N6,'(A,I4,A)') ' ISZ: ',ISZ,' STEP AUFGERUFEN!'
  I1 = 0
  I2 = 1
  ALPH(0) = 0.
  ALPH(1) = 1.
  ISTEP = 0
  DO 3 K=1,4
    PGRENZ(K) = 0
  3 CONTINUE
  DO 4 L=0,2
    QA(L) = 0.
    DQDA(L) = 0.
  4 CONTINUE
  5 CONTINUE
  ISTEP = ISTEP + 1
  IF (ISTEP.GT.10) GOTO 99
  DO 15 I=I1,I2
  7 CONTINUE
  DO 10 J=1,4
    PARA(J) = PAR(J) + R(J) * ALPH(I)
    IF (PARA(J).LT.0.) THEN
      IF (ALPH(I).LT.0.1) PGRENZ(J) = -1
      ALPH(I) = 0.75 * ALPH(I)
      WRITE(N6,*) ' PARAMETER NICHT MEHR IN DEN GRENZEN'
      WRITE(N6,*) ' PAR(1-4)',(PARA(L),L=1,4)
      GOTO 7
    ENDIF
  10 CONTINUE
  IF (NGRENZ(1).GE.3) PARA(1) = MIN(PARA(3)-.0001,0.9999)
  IF (PARA(3).GT.1.0) THEN
    IF (ALPH(I).LT.0.1) THEN
      PGRENZ(3) = 1
      IF (PARA(1).GT.0.9999) PGRENZ(1) = 1
    ENDIF
    GOTO 11
  ELSE IF (NGRENZ(1).LT.3 .AND. PARA(1).GT.(PARA(3)-.0001)) THEN
    IF (ALPH(I).LT.0.1) PGRENZ(1) = 1
    GOTO 11
  ENDIF
  GOTO 12
  11 CONTINUE
  ALPH(I) = 0.75 * ALPH(I)
  WRITE(N6,*) ' PARAMETER NICHT MEHR IN DEN GRENZEN'
  WRITE(N6,*) ' PAR(1-4)',(PARA(L),L=1,4)
  GOTO 7
  12 CONTINUE
  *
  ***** Schrittweite durch Grenzüberschreitungen zu klein ? *****
  *

```

```

IF (I.EQ.2 .AND. ALPH(2).LE.ALPH(1)) THEN
  WRITE(N6,*) ' SCHRITTWEITE DURCH GRENZUEBERSCHREITUNG',
+           ' ZU KLEIN ! '
  ALPHAM = ALPH(1)
  RETURN
ENDIF
AA = PARA(1)
VM = PARA(2)
AE = PARA(3)
AU = PARA(4)
CALL BLOCK(AA,VM,AE,AU)
DQDA(I) = DQDAL
QA(I) = QAL
IF (ALPH(I).EQ.0.) THEN
  IF (ISZ.EQ.1) THEN
    DIFTOL = DIFTOL * QAL
    WRITE(N6,*) ' DIFTOL(rel.)',DIFTOL
    EPSR = EPSTEP * QAL
    EPSRD = EPSTEP * ABS(DQDAL)
    WRITE(N6,*) ' EPSQAL',EPSR,' EPSDQDAL',EPSRD
  ENDIF
  IF (FEHFKT.LT.0) THEN
    QDIFF2 = QDIFF1
    QDIFF1 = ABS(QAL - QALALT)
    QALALT = QAL
  ENDIF
ENDIF
15 CONTINUE
IF (IPRO.EQ.1 .or.ipro.eq.0) THEN
  WRITE(N6,'(A,3(1X,F12.6))') ' ALPH(0-2) : ',
+   (ALPH(L),L=0,2)
  WRITE(N6,'(A,3(1X,F12.6))') ' QA(0-2) : ',
+   (QA(L),L=0,2)
  WRITE(N6,'(A,3(1X,F12.6))') ' DQDA(0-2) : ',
+   (DQDA(L),L=0,2)
ENDIF
IF (I2.NE.2) THEN
*
***** Bei Grenzüberschreitung evtl. schon fertig *****
*
  IF (ALPH(1).NE.1. .AND. QA(1).LE.QA(0)) THEN
    ALPHAM = ALPH(1)
    RETURN
  ENDIF
  IF (QA(1).GT.QA(0) .AND. (QA(1)-QA(0)).GT.EPSR) THEN
    QA(2) = QA(1)
    DQDA(2) = DQDA(1)
    ALPH(2) = ALPH(1)
    ALPH(1) = .5 * (ALPH(0) + ALPH(1))
  IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL2'
    I1 = -1
    GOTO 20
  ELSE
    ALPH(2) = 2. * ALPH(1)
    I1 = 2
    I2 = 2
    GOTO 5
  ENDIF
ENDIF
IF (.NOT.(QA(0).GT.QA(1) .AND. QA(1).LT.QA(2))) THEN

```

```

IF (ABS(QA(0)-QA(1)).LT.EPSR) THEN
  IF (ABS(QA(2)-QA(1)).LT.EPSR) THEN
    IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL3a'
    ALPHAM = ALPH(1)
    RETURN
  ELSE IF (QA(2).GT.QA(1)) THEN
    GOTO 20
  ENDIF
ENDIF
QA(0) = QA(1)
QA(1) = QA(2)
DQDA(0) = DQDA(1)
DQDA(1) = DQDA(2)
ALPH(0) = ALPH(1)
ALPH(1) = ALPH(2)
ALPH(2) = 2. * ALPH(2)
IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL3'
  GOTO 5
ENDIF
20 CONTINUE
*
***** Bei grossem Alpha mit dem 1. gefundenen *****
***** lokalen Minimum aufhoeren *****
*
  IF (ALPH(1).GT.10.) THEN
    ALPHAM = ALPH(1)
    RETURN
  ENDIF
  EPS1 = EPS3 * MAXDQ
  C WRITE(N2,*) 'EPS1:',EPS1,'MAXDQ:',MAXDQ
  C DO 30 M=1,5
  IF (I1.EQ.-1) THEN
    I1 = -2
    GOTO 22
  ENDIF
  IF (DQDA(1).GT.-EPSRD .AND. DQDA(1).LT.EPSRD) THEN
    IF (I1.EQ.-2 .AND. QA(1).GT.QA(0)) THEN
      QA(2) = QA(1)
      DQDA(2) = DQDA(1)
      ALPH(2) = ALPH(1)
      ALPH(1) = .5 * (ALPH(0) + ALPH(1))
    IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL4a'
    ELSE
      IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL4'
      ALPHAM = ALPH(1)
      RETURN
    ENDIF
  ELSE IF (DQDA(1).GT.0.) THEN
    QA(2) = QA(1)
    DQDA(2) = DQDA(1)
    ALPH(2) = ALPH(1)
    ALPH(1) = .5 * (ALPH(0) + ALPH(1))
  IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL5'
  ELSE
    QA(0) = QA(1)
    DQDA(0) = DQDA(1)
    ALPH(0) = ALPH(1)
    ALPH(1) = .5 * (ALPH(1) + ALPH(2))
  IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL6'
  ENDIF

```

```

22 CONTINUE
23 CONTINUE
DO 25 J=1,4
  PARA(J) = PAR(J) + R(J) * ALPH(1)
  IF (PARA(J).LT.0.) THEN
    IF (ALPH(1).LT.0.1) PGRENZ(J) = -1
    ALPH(1) = 0.75 * ALPH(1)
    WRITE(N6,*) ' PARAMETER NICHT MEHR IN DEN GRENZEN'
    WRITE(N6,*) ' PAR(1-4)',(PARA(L),L=1,4)
    GOTO 23
  ENDIF
25 CONTINUE
IF (NGRENZ(1).GE.3) PARA(1) = MIN(PARA(3)-.0001,0.9999)
IF (PARA(3).GT.1.0) THEN
  IF (ALPH(1).LT.0.1) THEN
    PGRENZ(3) = 1
    IF (PARA(1).GT.0.9999) PGRENZ(1) = 1
  ENDIF
  GOTO 27
ELSE IF (NGRENZ(1).LT.3 .AND. PARA(1).GT.(PARA(3)-.0001)) THEN
  IF (ALPH(1).LT.0.1) PGRENZ(1) = 1
  GOTO 27
ENDIF
GOTO 28
27 CONTINUE
ALPH(1) = 0.75 * ALPH(1)
WRITE(N6,*) ' PARAMETER NICHT MEHR IN DEN GRENZEN'
WRITE(N6,*) ' PAR(1-4)',(PARA(L),L=1,4)
GOTO 23
28 CONTINUE
AA = PARA(1)
VM = PARA(2)
AE = PARA(3)
AU = PARA(4)
CALL BLOCK(AA,VM,AE,AU)
DQDA(1) = DQDAL
QA(1) = QAL
IF (IPRO.EQ.1 .or. ipro.eq.0) THEN
  WRITE(N6,'(A,3(1X,F12.6))') ' ALPH(0-2) : ',
+    (ALPH(L),L=0,2)
  WRITE(N6,'(A,3(1X,F12.6))') ' QA(0-2) : ',
+    (QA(L),L=0,2)
  WRITE(N6,'(A,3(1X,F12.6))') ' DQDA(0-2) : ',
+    (DQDA(L),L=0,2)
ENDIF
IF (QA(0).GT.QA(1) .AND. QA(1).LT.QA(2)) THEN
IF (IPRO.EQ.1 .or. ipro.eq.0) WRITE(N6,*) 'FALL7'
  ALPHAM = ALPH(1)
  RETURN
ENDIF
30 CONTINUE
99 CONTINUE
IF (IPRO.EQ.1 .or. ipro.eq.0)
+  WRITE(N6,*) ' keine Loesung bei STEP erreicht!'
ALPHAM = ALPH(1)
RETURN
END

```

C
C
C

```

SUBROUTINE STEP2
*****
*
*   Bestimmung der optimalen Schrittweite
*
*****
$INCLUDE:'OPTI.CMN'
C
  REAL      PARA(4),ALPH(0:2),QA(0:2),DQDA(0:2)
C
C   IF (IPRO.EQ.1)
C   +  WRITE(N6,'(A,I4,A)') ' ISZ: ',ISZ,' STEP AUFGERUFEN!'
  I1      = 0
  I2      = 1
  ALPH(0) = 0.
  ALPH(1) = 1.
  ISTEP   = 0
  DO 3 K=1,4
    PGRENZ(K) = 0
3  CONTINUE
  DO 4 L=0,2
    QA(L) = 0.
    DQDA(L) = 0.
4  CONTINUE
5  CONTINUE
  ISTEP = ISTEP + 1
  IF (ISTEP.GT.10) GOTO 99
  DO 15 I=I1,I2
7  CONTINUE
  DO 10 J=1,4
    PARA(J) = PAR(J) + R(J) * ALPH(I)
    IF (PARA(J).LT.0.) THEN
      IF (ALPH(I).LT.0.1) PGRENZ(J) = -1
      ALPH(I) = 0.75 * ALPH(I)
      WRITE(N6,*) ' PARAMETER NICHT MEHR IN DEN GRENZEN'
      WRITE(N6,*) ' PAR(1-4)',(PARA(L),L=1,4)
      GOTO 7
    ENDIF
10 CONTINUE
  IF (NGRENZ(1).GE.3) PARA(1) = MIN(PARA(3)-.0001,0.9999)
  IF (PARA(3).GT.1.0) THEN
    IF (ALPH(1).LT.0.1) THEN
      PGRENZ(3) = 1
      IF (PARA(1).GT.0.9999) PGRENZ(1) = 1
    ENDIF
    GOTO 11
  ELSE IF (NGRENZ(1).LT.3 .AND. PARA(1).GT.(PARA(3)-.0001)) THEN
    IF (ALPH(1).LT.0.1) PGRENZ(1) = 1
    GOTO 11
  ENDIF
  GOTO 12
11 CONTINUE
  ALPH(1) = 0.75 * ALPH(1)
  WRITE(N6,*) ' PARAMETER NICHT MEHR IN DEN GRENZEN'
  WRITE(N6,*) ' PAR(1-4)',(PARA(L),L=1,4)
  GOTO 7
12 CONTINUE
*
***** Schrittweite durch Grenzüberschreitungen zu klein ? *****
*

```

```

IF (I.EQ.2 .AND. ALPH(2).LE.ALPH(1)) THEN
  WRITE(N6,*) ' SCHRITTWEITE DURCH GRENZUEBERSCHREITUNG',
  + ' ZU KLEIN ! '
  ALPHAM = ALPH(1)
  RETURN
ENDIF
AA = PARA(1)
VM = PARA(2)
AE = PARA(3)
AU = PARA(4)
CALL BLOCK(AA,VM,AE,AU)
DQDA(I) = DQDAL
QA(I) = QAL
IF (ALPH(I).EQ.0.) THEN
  IF (ISZ.EQ.1) THEN
    DIFTOL = DIFTOL * QAL
    WRITE(N6,*) ' DIFTOL(rel.)',DIFTOL
    EPSR = EPSTEP * QAL
    EPSRD = EPSTEP * ABS(DQDAL)
    WRITE(N6,*) ' EPSQAL',EPSR,' EPSDQDAL',EPSRD
  ENDIF
  IF (FEHFKT.LT.0) THEN
    QDIFF2 = QDIFF1
    QDIFF1 = ABS(QAL - QALALT)
    QALALT = QAL
  ENDIF
ENDIF
15 CONTINUE
IF (IPRO.EQ.1 .or.ipro.eq.0) THEN
  WRITE(N6, '(A,3(1X,F12.6))' ) ' ALPH(0-2) : ',
  + (ALPH(L),L=0,2)
  WRITE(N6, '(A,3(1X,F12.6))' ) ' QA(0-2) : ',
  + (QA(L),L=0,2)
  WRITE(N6, '(A,3(1X,F12.6))' ) ' DQDA(0-2) : ',
  + (DQDA(L),L=0,2)
ENDIF
IF (I2.NE.2) THEN
  *
  ***** Bei Grenzüberschreitung evtl. schon fertig *****
  *
  IF (ALPH(1).NE.1. .AND. QA(1).LE.QA(0)) THEN
    ALPHAM = ALPH(1)
    RETURN
  ENDIF
  IF (QA(1).GT.QA(0) .AND. (QA(1)-QA(0)).GT.EPSR) THEN
    QA(2) = QA(1)
    DQDA(2) = DQDA(1)
    ALPH(2) = ALPH(1)
  IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL2'
    I1 = -1
    GOTO 20
  ELSE
    ALPH(2) = 2. * ALPH(1)
    I1 = 2
    I2 = 2
    GOTO 5
  ENDIF
ENDIF
IF (.NOT.(QA(0).GT.QA(1) .AND. QA(1).LT.QA(2))) THEN
  IF (ABS(QA(0)-QA(1)).LT.EPSR) THEN

```

```

IF (ABS(QA(2)-QA(1)).LT.EPSR) THEN
  IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL3a'
  ALPHAM = ALPH(1)
  RETURN
ELSE IF (QA(2).GT.QA(1)) THEN
  GOTO 20
ENDIF
ENDIF
QA(0) = QA(1)
QA(1) = QA(2)
DQDA(0) = DQDA(1)
DQDA(1) = DQDA(2)
ALPH(0) = ALPH(1)
ALPH(1) = ALPH(2)
ALPH(2) = 2. * ALPH(2)
IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL3'
GOTO 5
ENDIF
20 CONTINUE
*
***** Bei grossem Alpha mit dem 1. gefundenen *****
***** lokalen Minimum aufhoeren *****
*
C IF (ALPH(1).GT.10.) THEN
C ALPHAM = ALPH(1)
C RETURN
C
C ENDIF
C EPS1 = EPS3 * MAXDQ
C WRITE(N2,*) 'EPS1:',EPS1,'MAXDQ:',MAXDQ
DO 30 M=1,5
  IF (I1.EQ.-1) THEN
    I1 = -2
    GOTO 22
  ENDIF
  IF (DQDA(1).GT.-EPSRD .AND. DQDA(1).LT.EPSRD) THEN
    IF (I1.EQ.-2 .AND. QA(1).GT.QA(0)) THEN
      QA(2) = QA(1)
      DQDA(2) = DQDA(1)
      ALPH(2) = ALPH(1)
    IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL4a'
    ELSE
    IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL4'
      ALPHAM = ALPH(1)
      RETURN
    ENDIF
  ELSE IF (DQDA(1).GT.0.) THEN
    QA(2) = QA(1)
    DQDA(2) = DQDA(1)
    ALPH(2) = ALPH(1)
  IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL5'
  ELSE
    QA(0) = QA(1)
    DQDA(0) = DQDA(1)
    ALPH(0) = ALPH(1)
  IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL6'
  ENDIF
22 CONTINUE
23 CALL KUBI(ALPH(0),ALPH(2),QA(0),QA(2),DQDA(0),DQDA(2),ALPH(1))
CONTINUE
DO 25 J=1,4

```

```

      PARA(J) = PARA(J) + R(J) * ALPH(1)
      IF (PARA(J).LT.0.) THEN
        IF (ALPH(1).LT.0.1) PGRENZ(J) = -1
        ALPH(1) = 0.75 * ALPH(1)
        WRITE(N6,*) ' PARAMETER NICHT MEHR IN DEN GRENZEN'
        WRITE(N6,*) ' PAR(1-4)',(PARA(L),L=1,4)
        GOTO 23
      ENDIF
25    CONTINUE
      IF (NGRENZ(1).GE.3) PARA(1) = MIN(PARA(3)-.0001,0.9999)
      IF (PARA(3).GT.1.0) THEN
        IF (ALPH(1).LT.0.1) THEN
          PGRENZ(3) = 1
          IF (PARA(1).GT.0.9999) PGRENZ(1) = 1
        ENDIF
        GOTO 27
      ELSE IF (NGRENZ(1).LT.3 .AND. PARA(1).GT.(PARA(3)-.0001)) THEN
        IF (ALPH(1).LT.0.1) PGRENZ(1) = 1
        GOTO 27
      ENDIF
      GOTO 28
27    CONTINUE
      ALPH(1) = 0.75 * ALPH(1)
      WRITE(N6,*) ' PARAMETER NICHT MEHR IN DEN GRENZEN'
      WRITE(N6,*) ' PAR(1-4)',(PARA(L),L=1,4)
      GOTO 23
28    CONTINUE
      AA = PARA(1)
      VM = PARA(2)
      AE = PARA(3)
      AU = PARA(4)
      CALL BLOCK(AA,VM,AE,AU)
      DQDA(1) = DQDAL
      QA(1) = QAL
      IF (IPRO.EQ.1 .or.ipro.eq.0) THEN
        WRITE(N6,'(A,3(1X,F12.6))') ' ALPH(0-2) : ',
+      (ALPH(L),L=0,2)
        WRITE(N6,'(A,3(1X,F12.6))') ' QA(0-2) : ',
+      (QA(L),L=0,2)
        WRITE(N6,'(A,3(1X,F12.6))') ' DQDA(0-2) : ',
+      (DQDA(L),L=0,2)
      ENDIF
      IF (QA(0).GT.QA(1) .AND. QA(1).LT.QA(2)) THEN
      IF (IPRO.EQ.1 .or.ipro.eq.0) WRITE(N6,*) 'FALL7'
        ALPHAM = ALPH(1)
        RETURN
      ENDIF
30    CONTINUE
99    CONTINUE
      IF (IPRO.EQ.1 .or.ipro.eq.0)
+      WRITE(N6,*) ' keine Loesung bei STEP erreicht!'
      ALPHAM = ALPH(1)
      RETURN
      END
C
C
C
      SUBROUTINE KUBI(X0,X1,F0,F1,DF0,DF1,XL)
***** HERMITE-INTERPOLATION : an zwei Stuetzstellen sind Funktions- **

```

```

***** werte und 1. Ableitungen gegeben, dazwischen wird mit einem **
***** Polynom 3. Grades interpoliert und das Minimum bestimmt **
*****
C
***** Werte der verallgemeinerten dividierten Differenzen der *****
***** HERMITE-INTERPOLATION fuer kubische Polynome berechnen *****
C
      XD = X1 - X0
      B = DF0
      B1 = (F1 - F0) / XD
      C = (B1 - B) / XD
      C1 = (DF1 - B1) / XD
      D = (C1 - C) / XD
C
***** Koeffizienten der 1. Ableitung des kubischen Polynoms *****
C
      AA = 3 * D
      BB = 2*C - 2*D * (2*X0 + X1)
      CC = (2*D*X1 + D*X0 - 2*C) * X0 + B
C
***** Nullstellenberechnung der 1. Ableitung *****
C
      BD2A = BB / AA / 2
      SUM = BD2A * BD2A - CC / AA
      WURZ = SQRT(SUM)
      XL1 = WURZ - BD2A
C
***** hinreichendes Minimumkriterium mit der 2. Ableitung pruefen *****
C
      DDFX = 2*AA * XL1 + BB
      IF (DDFX.GT.0.) THEN
        XL = XL1
      ELSE
        XL = - WURZ - BD2A
      ENDIF
      RETURN
      END

```