

Entwicklung, Validierung und Anwendung
eines dreidimensionalen, strömungsgekoppelten
finite Differenzen
Wärmetransportmodells

Inaugural-Dissertation
zur Erlangung des Doktorgrades der Naturwissenschaften
der Justus Liebig-Universität Gießen
Fachbereich Geowissenschaften und Geographie

vorgelegt von
Dirk R. Brehm
aus
Singen/Hohentwiel
Gießen 1989

Inhaltsverzeichnis

Vorwort	i
Kurzfassung	ii
Abstract	iii
Nomenklatur	iv
Liste der Abbildungen	vii
Liste der Tabellen	xii
1 Einleitung	1
1.1 Problemstellung	2
1.2 Bisherige Arbeiten	3
2 Numerische Grundlagen	5
2.1 Strömungsberechnung	5
2.2 Mechanismen des Wärmetransports	6
2.2.1 Konduktion	6
2.2.2 Strahlung	8
2.2.3 Konvektion	8
2.3 Konvektion und Diffusion	9
2.4 Diffusion und Dispersion in porösen Medien	11
2.5 Phasenwechsel von Wasser	12
3 Numerische Behandlung der Gleichungen	15
3.1 Konstruktion der finiten Differenzen Approximation ...	16
3.1.1 Der Explizit-Ansatz	17
3.1.2 Der Implizit-Ansatz	19
3.1.3 Der Crank-Nicolson-Ansatz	19

3.2	Möglichkeiten der Diskretisierung	20
3.3	Die Diskretisierungsgleichungen	22
3.4	Iterative Lösungsverfahren	29
3.5	Der Thomas-Algorithmus	37
3.6	Behandlung der Randbedingungen	39
4	Programmentwicklung	41
5	Validierung des Modells	44
5.1	Konduktiver Wärmetransport	44
5.2	Instationärer, eindimensionaler Wärmetransport	45
5.3	Eindimensionales Konvektionsproblem	47
5.4	Brunnenabsenkung (Theis-Problem)	49
5.5	Testsimulation eines Stufenpumpversuches	50
5.6	Leakage aus einem Oberflächengewässer	53
6	Anwendung	58
6.1	Die Forschungsanlage Schwalbach	58
6.1.1	Beschreibung der Anlage	58
6.1.2	Diskretisierung	60
6.1.3	Wärmeentzug mit quasikonstanter Soletemperatur	61
6.1.4	Wärmetransport im Boden	66
6.2	Die Erdsondenanlage Göttingen	69
6.2.1	Beschreibung der Anlage	69
6.2.2	Diskretisierung	70
6.2.3	Wärmentzug mit konstanter Bohrlochtemperatur	73
6.2.4	Wärmentzug mit konstanter Entzugsleistung ...	76

6.3	Temperaturprognose in große Tiefen	79
7	Diskussion	82
8	Literatur	84
	Anhang 1: Physikalische Eigenschaften	A.1.1
	Anhang 2: Programmaufbau und Variablenerklärung	A.2.1
	Anhang 3: Programm-Listing	A.3.1
	Anhang 4: Beispieleingabedaten	A.4.1
	Anhang 5: <i>Basic</i> -Programme zur Validierung	A.5.1
	Anhang 6: Ergebnisse der Validierungsläufe	A.6.1
	Anhang 7: Programmübersicht ähnlicher Modelle	A.7.1
	Anhang 8: Diagramme der Schwalbacher Kalibrierungsläufe	A.8.1

Vorwort

Die Anregung zu der vorliegenden Arbeit verdanke ich Herrn Prof. Dr. K. Knoblich, der auch durch eine Vielzahl von Ratschlägen die theoretische Entwicklung des Programmes erleichterte. Ferner gewährte er mir, als sein wissenschaftlicher Mitarbeiter am Institut für Angewandte Geowissenschaften der Justus-Liebig-Universität Gießen, sehr viel Freiraum für die Fertigstellung der Arbeit.

Den Herrn Prof. Dr. U. Haack, Institut für Geowissenschaften und Lithosphärenforschung, und Prof. Dr. H. Wilhelm, Geophysikalisches Institut der Universität Fridericiana Karlsruhe, danke ich für ihr Interesse und die bereitwillige Übernahme des Korreferats.

Der Helmut Hund GmbH und dem Bundesminister für Forschung und Technologie (BMFT) gebührt große Anerkennung für die finanzielle und materielle Unterstützung, die die Durchführung der vorliegenden Untersuchungen überhaupt erst ermöglichte.

Herrn Dipl.-Geol. B. Sanner, Helmut Hund GmbH, sei für die gute Zusammenarbeit im Rahmen des geförderten Forschungsprojektes gedankt.

Mein Kollege, Herr Dipl.-Geol. M. Einig, erleichterte mir durch die vielen Diskussionen und Tips den Einstieg in die für Geologen etwas ungewohnte Materie der numerischen Methoden des Massen- und Stofftransports.

Dem Hochschulrechenzentrum Gießen schulde ich Dank für die zur Verfügung gestellte Rechenzeit sowie die Benutzung der Peripheriegeräte. Seine Mitarbeiter erleichterten mir mit vielen wertvollen Tips und Tricks den Umgang mit dem Großrechner *CDC 860*.

Bei Herrn Dr. Bentzler, Fa. Mettler - Gießen, möchte ich mich für die Durchführung der temperaturabhängigen Wärmekapazitätsmessungen bedanken.

Kurzfassung

Die vorliegende Arbeit beschreibt die Entwicklung, Validierung und Kalibrierung eines dreidimensionalen, numerischen Modells zur Simulation des instationären Fluid- und Wärmetransports in gesättigten, heterogenen, porösen Medien. Mit diesem Modell sollte der Prozess der Wärmeübertragung im Erdreich, wie er durch den Betrieb von Wärmepumpenanlagen mit vertikalen Erdkollektoren angeregt wird, näher untersucht sowie Möglichkeiten und Grenzen derartiger Systeme aufgezeigt werden. Darüber hinaus werden Simulationsläufe für die Prognose des in größeren Tiefen ($> 1000\text{m}$) herrschenden Temperaturfeldes sowie für den konduktiven Wärmetransport in der ungesättigten Bodenzone vorgestellt.

Das auch als Quelltext vorliegende finite Differenzenmodell, eine modular aufgebautes FORTRAN77 Computer-Programm, löst simultan die Strömungsgleichung und die konduktiv-konvektive Wärmetransportgleichung durch Einsatz zweier alternativer Iterationsverfahren. Es können jedoch auch Teilaspekte wie Grundwasserströmung oder rein konduktiver Wärmetransport simuliert werden.

Abstract

This paper describes the development, validation and calibration of a three-dimensional numerical model which enables the simulation of transient mass and heat transfer in saturated heterogenous porous media. The major purpose was to get a better impression of heat transfer process in the surrounding of vertical earth heat exchangers which are used as a heat source for heat pumps. Additionally some sample computations dealing with forecasting the temperature field in great depth ($> 1000\text{m}$) and the pure conductive heat transfer in soils are presented.

The modular FORTRAN77-computer code solves simultaneously the Darcy - equation and the conductive-convective heat transfer equation with two alternative iteration schemes. The code can also be used as a ground-water model or simple conductive heat transfer model. The source code of the program is given in the appendix.

Nomenklatur

Lateinische Symbole

a	Koeffizienten des linearen Gleichungssystems
A	Fläche, $[L^2]$
b	Quelltermkoeffizient
b_0	Basis des Aquifers, $[L]$
C	Wärmekapazität, $[L^2 t^{-2} T^{-1}]$
C_e	Wärmekapazität von Eis, $[L^2 t^{-2} T^{-1}]$
C_{eff}	effektive Wärmekapazität, $[L^2 t^{-2} T^{-1}]$
C_g	Wärmekapazität von Gestein, $[L^2 t^{-2} T^{-1}]$
C_l	Wärmekapazität von Luft, $[L^2 t^{-2} T^{-1}]$
C_w	Wärmekapazität von Wasser, $[L^2 t^{-2} T^{-1}]$
Co_i	Courant-Zahl in i -Richtung, $[-]$
D_i	Diffusionskoeffizient in i -Richtung, $[M^2 t^{-1}]$
e	Basis der Exponentialfunktion, 2.7182818...
F_i	Konvektionskoeffizient in i -Richtung, $[M^2 t^{-1}]$
g	Erdbeschleunigung, $[L t^{-2}]$
h	Piezometerhöhe, $[L]$
h_0	Ausgangspiezometerhöhe, $[L]$
H_f	Schmelzenthalpie von Wasser, $[L^2 t^{-2}]$
k_f	isotrope hydraulische Leitfähigkeit, $[L t^{-1}]$
k_{xx}	hydraulische Leitfähigkeit in X - Richtung, $[L t^{-1}]$
k_{yy}	hydraulische Leitfähigkeit in Y - Richtung, $[L t^{-1}]$
k_{zz}	hydraulische Leitfähigkeit in Z - Richtung, $[L t^{-1}]$
K	Wärmeleitfähigkeit, $[ML t^{-3} T^{-1}]$
K_o	Wärmeleitfähigkeit bei einer bekannten Temperatur, $[ML t^{-3} T^{-1}]$
K_e	Wärmeleitfähigkeit von Eis, $[ML t^{-3} T^{-1}]$
K_{eff}	effektive Wärmeleitfähigkeit, $[ML t^{-3} T^{-1}]$
K_g	Wärmeleitfähigkeit von Gestein, $[ML t^{-3} T^{-1}]$
K_{korr}	temperaturkorrigierte Wärmeleitfähigkeit, $[ML t^{-3} T^{-1}]$
K_l	Wärmeleitfähigkeit von Luft, $[ML t^{-3} T^{-1}]$
K_{nm}	Wärmeleitfähigkeit von Nichtmetallen, $[ML t^{-3} T^{-1}]$
K_p	Wärmeleitfähigkeit der Porenraumfüllung, $[ML t^{-3} T^{-1}]$
K_w	Wärmeleitfähigkeit von Wasser, $[ML t^{-3} T^{-1}]$
K_{xx}	effektive Wärmeleitfähigkeit in X - Richtung, $[ML t^{-3} T^{-1}]$

K_{yy}	effektive Wärmeleitfähigkeit in Y - Richtung,	$[ML t^{-3} T^{-1}]$
K_{zz}	effektive Wärmeleitfähigkeit in Z - Richtung,	$[ML t^{-3} T^{-1}]$
m	Mächtigkeit,	$[L]$
J	Wärmestrom,	$[M L^2 t^{-3}]$
Pe	Peclet - Zahl,	$[-]$
Pr	Prandtl - Zahl, $Pr = \mu_{dyn} C / K$,	$[-]$
q''	Wärmestromdichte,	$[M t^{-3}]$
q_x	Volumenstrom in X - Richtung,	$[L^3 t^{-1}]$
q_y	Volumenstrom in Y - Richtung,	$[L^3 t^{-1}]$
q_z	Volumenstrom in Z - Richtung,	$[L^3 t^{-1}]$
Q_e	Entnahme oder Zugabe von Energie,	$[ML^{-1} t^{-3}]$
Q_{rad}	radiogene Wärmeproduktion,	$[ML^{-1} t^{-3}]$
Q_w	Entnahme oder Zugabe von Wasser,	$[L^3 t^{-1}]$
r	radiale Entfernung vom Entnahmehrunden,	$[L]$
R_w	Wärmewiderstand,	$[M^{-1} L^{-2} t^3 T]$
Re	Reynolds-Zahl, $Re = v \delta x \rho / \mu_{dyn}$,	$[-]$
S_s	spezifischer Speicherkoeffizient,	$[L^{-1}]$
S'	dimensionsloser Speicherkoeffizient,	$[-]$
T	Temperatur,	$[T]$
T_0	Ausgangstemperatur,	$[T]$
T_1	Ausgangstemperatur,	$[T]$
T^*	Temperatur des vorausgegangenen Zeitschrittes,	$[T]$
t	Zeit,	$[t]$
v_a	Abstandsgeschwindigkeit,	$[L t^{-1}]$
v_f	Filtergeschwindigkeit,	$[L t^{-1}]$
v_x	Filtergeschwindigkeit in X - Richtung,	$[L t^{-1}]$
v_y	Filtergeschwindigkeit in Y - Richtung,	$[L t^{-1}]$
v_z	Filtergeschwindigkeit in Z - Richtung,	$[L t^{-1}]$
X_e	Volumenanteil des gefrorenen Wassers,	$[-]$
X_w	Volumenanteil des ungefrorenen Wassers,	$[-]$
z	Tiefe,	$[L]$

Griechische Symbole

α	Wichtungsfaktor zwischen Explizit- und Implizitverfahren,	$[-]$
α_T	Temperaturleitwert,	$[L^2 t^{-1}]$
β_i	Teilfeld des Thomas-Algorithmus	
γ_i	Teilfeld des Thomas-Algorithmus	

Γ_i	Wärmediffusion in i -Richtung, $[L^2 t^{-1}]$
κ	intrinsische Leitfähigkeit, $[L^2]$
μ_{dyn}	dynamische Viskosität, $[MLt^{-1}]$
μ_{kin}	kinematische Viskosität, $[L^2 t^{-1}]$
ρ	Dichte, $[ML^{-3}]$
ρ_e	Dichte von Eis, $[ML^{-3}]$
ρ_{eff}	effektive Dichte, $[ML^{-3}]$
ρ_g	Dichte von Gestein, $[ML^{-3}]$
ρ_l	Dichte von Luft, $[ML^{-3}]$
ρ_w	Dichte von Wasser, $[ML^{-3}]$
τ	thermische Relaxationszeit, $[t]$
ϕ	unbekannte Zielgröße (Temperatur, Piezometerhöhe ...)
Φ_g	Gesamtporenraum, $[-]$
Φ_n	durchflußwirksamer Porenraum, $[-]$
Ψ	Wassersättigungsgrad des Porenraumes, $[-]$
Ω	Relaxationsfaktor, $[-]$

Indizes

E, e	Osten
I, i	Index der X - Richtung
J, j	Index der Y - Richtung
K, k	Index der Z - Richtung
nb	Sammelindex aller Nachbarknoten des Knotens P
N, n	Norden
O, o	Oben
P	eindimensionaler Index des Knotens (i,j,k)
S, s	Süden
U, u	Unten
W, w	Westen

Liste der Abbildungen

1	Verlauf der Funktion $A(Pe)$ für die in Tab. 1 aufgeführten Verfahren	10
2	Beginn der Phasenumwandlung von Wasser; hier bei 0 °C.	13
3	Berechnete Wärmekapazität von teilgefrorenem Wasser; die Ordinate wurde aus Gründen der Übersichtlichkeit logarithmisch dargestellt	15
4	Knotenzentrierte Anordnung der Zellengrenzen; die Bezeichnungen beziehen sich auf die im Programm verwandten Variablennamen	21
5	Zellenzentrierte Anordnung der Knoten; die Bezeichnungen beziehen sich auf die im Programm verwandten Variablennamen	22
6	Knotenindizierung der Schicht k	23
7	Fluß durch eine dreidimensionale finite Differenzen Zelle	23
8	Indizierung der sechs Nachbarn des Knotens P	24
9	Flußdiagramm der Gauss-Seidel-Iteration	30
10	Flußdiagramm des IADI-Verfahrens	32
11	Stationärer Wärmefluß durch eine Hochofenwand	45
12	Konduktiver Wärmetransport in dem beschriebenen Gesteinskörper. Die durchgezogenen Linien markieren die oben beschriebene analytische Lösung, die Symbole das Simulationsergebnis von TRADIKON-3D.	47
13	Vergleich der analytischen mit der numerischen Lösung des Konvektionsproblem es	48
14	Zeit-Absenkungskurve des numerischen und analytischen Ansatzes	52
15	Eingangsdaten und Diskretisierung der Simulation des Stufenpumpversuches	53

16	<i>TRADIKON-3D Simulation des Stufenpumpversuches</i>	54
17	<i>ASM Simulation des Stufenpumpversuches</i>	54
18	<i>Diskretisierung des ungespannten Aquifers</i>	55
19	<i>TRADIKON-3D Simulation der Leakage aus einem Oberflächen- gewässer</i>	56
20	<i>Lageplan der Erdsondenforschungsanlage Schwalbach im November 1987</i>	59
21	<i>Lage der Bohrungen in dem finite Differenzen Raster</i>	60
22	<i>Temperaturentwicklung in der Entzugsbohrung Z</i>	61
23	<i>Gemessene Regeneration der Erdreichtemperaturen nach 31-tägigem Wärmeentzug.</i>	62
24	<i>Temperaturerniedrigung in Bohrung 2/0</i>	63
25	<i>Temperaturerniedrigung in Bohrung 4/0</i>	63
26	<i>Rein konduktiv berechnete Regeneration nach 31- tägigem Wärmeent- zug (Datensatz 7, Tab. 5)</i>	64
27	<i>Durch die Schmelzenthalpie des Wassers hervorgerufene Verzögerung der Temperaturregeneration in der Entzugsbohrung.</i>	65
28	<i>Gemessene Temperaturen in den obersten 0.5 Meter des Schwalbacher Bodenprofiles</i>	67
29	<i>Simulierte Temperaturen in den obersten 0.5 Meter des Schwalbacher Bodenprofiles</i>	69
30	<i>Lageplan der Erdsondenanlage Göttingen</i>	70
31	<i>Ausschnitt der Diskretisierung der Erdsondenanlage Göttingen; die Distanz zu den Modellrändern beträgt an jeder Seite etwa 33 Meter (jeweils 7 Knotenreihen)</i>	71

32	Gemessener Temperaturverlauf in der Bohrung B1, November / Dezember 1987	72
33	Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 1. Betriebsjahr	73
34	Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 3. Betriebsjahr	74
35	Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 7. Betriebsjahr	74
36	Geglätteter Temperaturverlauf in den Bohrungen B1 - B6 über 7 Betriebsjahre	75
37	Numerische Approximation der verbleibenden Temperaturabsenkung bei vorgegebener Bohrlochtemperatur	75
38	Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 1. Betriebsjahr	76
39	Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 3. Betriebsjahr	77
40	Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 7. Betriebsjahr	77
41	Geglätteter Temperaturverlauf in den Bohrungen B1 - B6 über 7 Betriebsjahre	78
42	Numerische Approximation der verbleibenden Temperaturabsenkung bei vorgegebener Entzugsleistung	79
43	Eindimensionale Temperaturberechnung mit druck- und temperaturabhängiger Wärmeleitfähigkeit	80
A.1.1	Temperaturabhängigkeit der Dichte von luftfreiem Wasser bei 760 Torr	A.1.2
A.1.2	Temperaturabhängigkeit der Wärmeleitfähigkeit von Wasser für die Verfahren nach NEISS(1982) und TOULIKAN(1970)	A.1.3

A.1.3	Temperaturabhängigkeit der Wärmekapazität von Wasser ...	A.1.4
A.1.4	Temperaturabhängigkeit der kinematischen Viskosität	A.1.4
A.1.5	Abhängigkeit der Wärmeleitfähigkeit des Eises von der Temperatur, nach NEISS(1982)	A.1.6
A.1.6	Temperaturkorrektur der Wärmeleitfähigkeit nach dem exponentiellen Ansatz	A.1.9
A.1.7	Temperaturkorrektur der Wärmeleitfähigkeit nach dem polynomischen Ansatz	A.1.10
A.1.8	Temperaturabhängigkeit der Wärmeleitfähigkeit von Granit und ihre numerische Approximation	A.1.11
A.1.9	Druck- und Temperaturabhängigkeit der Wärmeleitfähigkeit von Amphibolit.	A.1.12
A.1.10	Druck- und Temperaturabhängigkeit der Wärmeleitfähigkeit von Disthen-Sillimanit-Gneis.	A.1.12
A.1.11	Näherungsverfahren zur Bestimmung der effektiven Wärmeleitfähigkeit nach /210/	A.1.13
A.1.12	Einfluß der Porenraumverteilung auf die effektive Wärmeleitfähigkeit nach /210/ im Vergleich zum gewichteten arithmetischen Mittel	A.1.14
A.1.13	Temperaturabhängigkeit der Wärmekapazität einiger paläozoischer Sedimentgesteine, Messungen mit dem METTLER Thermoanalytensystem TA3000	A.1.15
A.8.1	Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 1, Tab. 5)	A.8.1
A.8.2	Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 2, Tab. 5)	A.8.2
A.8.3	Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 3, Tab. 5)	A.8.2

A.8.4	Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 4, Tab. 5)	A.8.3
A.8.5	Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 5, Tab. 5)	A.8.3
A.8.6	Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 6, Tab. 5)	A.8.4
A.8.7	Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 7, Tab. 5)	A.8.4
A.8.8	Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 8, Tab. 5)	A.8.5
A.8.9	Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 9, Tab. 5)	A.8.5
A.8.10	Berechnete Regeneration in Bohrung Z nach 31-tägigem Wärmeentzug (alle Datensätze)	A.8.6

Liste der Tabellen

1	Die Wichtungsfunktion $A(Pe)$ für die verschiedene Verfahren nach /151/.	9
2	Absenkung ($h_0 - h$) in Meter für $r = 100m$	51
3	Piezometerhöhen in Meter über NN zu ausgewählten Zeitpunkten	52
4	Berechnete Piezometerhöhen zweier Brunnen in Meter ü. NN . . .	56
5	Eingangsparameter der 9 Kalibrierungsläufe	65
A.1.1	Wärmeleitfähigkeiten wichtiger Erguß- und Tiefengesteine . . .	A.1.7
A.1.2	Wärmeleitfähigkeiten wichtiger metamorpher Gesteine	A.1.8
A.1.3	Wärmeleitfähigkeiten wichtiger Sedimentgesteine	A.1.8
A.1.4	Wärmekapazitäten wichtiger Erguß- und Tiefengesteine	A.1.16
A.1.5	Wärmekapazitäten wichtiger metamorpher Gesteine	A.1.16
A.1.6	Wärmekapazitäten wichtiger Sedimentgesteine	A.1.17
A.1.7	Wärmeproduktion verschiedener Gesteine	A.1.18

1 Einleitung

Die Verknappung der fossilen Brennstoffträger, wachsendes Umweltbewußtsein und nicht zuletzt ökonomische Gesichtspunkte haben in den beiden letzten Jahrzehnten zu einer Vielfalt von Forschungsaktivitäten geführt, die vorhandenen Energiereserven so umweltschonend wie möglich zu nutzen. Dabei wurden große Fortschritte in der besonders umweltfreundlichen Nutzung der Wind- und Solarenergie erzielt. Dennoch können diese Techniken in den dichtbesiedelten Industrieländern mit ihrem hohen Energieverbrauch nur mittelfristig die bisherigen Verfahren ersetzen, sodaß z.B. die Stromerzeugung aus Atomenergie und Kohle noch bis in das nächste Jahrtausend von großer Bedeutung sein wird. Die zunehmende Luftverschmutzung und die damit verbundene Gefahr einer globalen Klimakatastrophe drängt zum Handeln. Auf dem häuslichen Sektor existiert neben den Solarkollektoren bereits eine Technik, mit der man die Verbrennung von fossilen Brennstoffen stark einschränken oder gar vermeiden kann - die erdgekoppelte Wärmepumpe. Mit einer solchen Anlage wird den obersten 20 bis 100m der Erdkruste Wärmeenergie entzogen, die dann in komprimierter Form zur Warmwasserbereitung und Beheizung von Gebäuden dienen kann.

Da Erdsondenwärmepumpenanlagen ein geschlossenes System darstellen, ist auch erdseitig eine gute Umweltverträglichkeit gewährleistet. Speziell in Ballungsräumen mit ihren zum Teil anthropogen erheblich erwärmten Grundwasserleitern wäre schon aus wasserhygienischen Gründen der Betrieb solcher Anlagen geradezu wünschenswert, /82/,/89/. Elektrisch angetriebene Wärmepumpenanlagen sind jedoch unter dem Aspekt des Primärenergieverhältnisses nur dann sinnvoll, wenn ihre Leistungszahl, der Quotient aus der an das Heizungssystem abgegebenen und investierten Energie, den Wert von 3 nach Möglichkeit überschreitet. Für die Leistungszahl sind neben rein technischen Aspekten wie der Auslegung des Heizungssystems und der Anlagensteuerung vor allem die Eigenschaften des Entzugsmediums verantwortlich. Wichtige physikalische Parameter sind hierbei:

- die unbeeinflusste Erdreichtemperatur
- die Wärmeleitfähigkeit des Gesteins oder Bodens
- die Wärmekapazität des Gesteins oder Bodens
- der Anteil des Porenraumes und sein Wassersättigungsgrad
- eventuell das Vorhandensein von bewegtem Grundwasser

- die Eigenschaften des Hinterfüllmaterials

Da man aus Kostengründen und z.T. aus räumlichen Beschränkungen geneigt ist, die vertikalen Wärmetauscher so eng nebeneinander wie möglich zu placieren und die insgesamt einzubauende Sondenlänge zu minimieren, sind für ein optimales Anlagenlayout stets Berechnungen des erdseitigen Wärmetransports erforderlich.

Speziell für überschlägige Berechnungen existieren bereits eine Reihe von PC-Programmen, durch deren Einsatz man den Einfluß der Wärmetauscherlänge, des Rückfüllmaterials, der thermischen Eigenschaften des Gesteins usw. auf die zu erwartende Entzugsleistung abschätzen kann /62/. Der Vorteil dieser Programme liegt in ihrer einfachen Handhabung und hohen Rechengeschwindigkeit. Da sie jedoch durchwegs auf analytischen Lösungen basieren, lassen sich nur relative einfache Symmetrien und Randbedingungen mit ihnen lösen.

Das hier vorgestellte numerische Modell eignet sich wegen der sehr aufwendigen Datenvorbereitung und dem z.T. immensen Rechenzeitbedarf weniger für den Installateurbetrieb, es stellt vielmehr ein Instrument für Parameterstudien an den in die Berechnung einfließenden physikalischen Größen dar.

1.1 Problemstellung

Der erste Teil der vorliegenden Arbeit beschreibt die Entwicklung eines dreidimensionalen, strömungsgekoppelten Wärmetransportmodells auf Basis der finiten Differenzen. Dieses Modell sollte in der Lage sein, den über Erdwärmetauscher angeregten Wärmetransport im Erdreich unter Berücksichtigung des strömenden Grundwassers und des möglichen Phasenwechsels von Wasser hinreichend genau nachzubilden. Strömungsteil und Wärmetransportteil des Programms wurde getrennt von einander gegen analytische Lösungen validiert. Ferner dienten Vergleichssimulationen mit anderen Modellen zur Validierung gegenüber Problemstellungen, zu denen keine analytischen Lösungen existieren.

Im zweiten Teil sollten eine Reihe von Simulationsläufen Aufschluß darüber geben, welche physikalischen Parameter von maßgeblicher Bedeutung für die oben geschilderte Problematik des Wärmeentzugs sind. Ferner werden Simulationsergebnisse des Modells zum rein konduktiven Wärmetransport in Böden und zur Temperaturprognose in übertiefen Bohrungen vorgestellt.

1.2 Bisherige Arbeiten

Die zum Thema Erdsondenwärmepumpenanlagen und Erdwärmespeicherung in Aquiferen existierende Literatur ist, obwohl es sich hierbei noch um eine vergleichsweise junge Technologie handelt, bereits sehr umfangreich /31/. Einen sehr guten Überblick über die Entwicklungsstufen des oben beschriebenen Systems erhält man von /16/.

Die zur Problematik des Wärmetransportes im Erdreich bereits existierenden Computermodelle kann man in zwei Gruppen einteilen: Die erste Gruppe verfügt in der Regel über einen relativ einfachen — meist rein konduktiven — Wärmetransportteil. Allerdings können mit dieser Kategorie von Programmen meistens auch die Wärmetransportvorgänge innerhalb des Wärmetauschers und vom Wärmetauscher zum Entzugsmedium behandelt werden. Ferner gestatten diese Modelle oft auch die Wahl zwischen unterschiedlichen Koordinatensystemen (kartesisch, radial, zylindrisch, sphärisch ...).

Die zweite Kategorie von Programmen behandelt mehr allgemein den Wärme- bzw. Stofftransport in porösen Aquiferen. Eine nahezu vollständige Übersicht der zu dieser Gruppe gehörigen Modelle sowie eine kurze Beschreibung des jeweiligen Leistungsumfanges wird durch /10/, /212/ und /99/ vorgestellt.

Der bei weitem überwiegende Teil der allgemein zugänglichen Programme letzterer Kategorie stammt aus den Vereinigten Staaten von Amerika, und hier im besonderen vom U. S. Geological Survey. Dies hängt zum einen damit zusammen, daß in den USA die Infiltration von Kühlwasser und flüssigen Industrieabfällen in tiefe Porenwasserleiter intensiv diskutiert wurde und noch wird. Ferner wurde nahezu die gesamte Pionierarbeit auf dem Sektor der Modellierung hydrologischer Fragestellungen dort geleistet. Zum anderen wird speziell in Europa bei der Veröffentlichung derartiger Programme wesentlich restriktiver verfahren.

Eine Übersicht der verfügbaren Programme beider Kategorien sowie eine kurze Beschreibung des jeweiligen Leistungsumfanges kann dem Anhang 7 entnommen werden.

Da jedoch nach intensiver Literaturrecherche keines der allgemein verfügbaren Programme in der Lage zu sein schien, die geschilderte Problematik vollständig zu lösen, bestand nur die Möglichkeit, entweder ein vorhandenes Programm um die fehlenden Aspekte zu ergänzen oder ein komplett neues Modell zu schreiben. Der Autor entschied sich

für eine Neuentwicklung, weil sie zum einen dem besseren Verständnis der zu behandelnden Materie diene und zum anderen nur sehr wenige als Quelltext verfügbare Programme so strukturiert sind, daß sich Erweiterungen leicht integrieren lassen, /132/,/209/. Das im folgenden vorgestellte Wärmetransportmodell trägt den Namen TRADIKON-3D, der als Kürzel für Transport von Wärme durch Diffusion und Konvektion in 3 Dimensionen steht.

2 Numerische Grundlagen

2.1 Die Grundwasserströmungsgleichung

Das durch ein dreidimensionales, poröses Medium fließende Grundwasser läßt sich nach /19/ berechnen:

$$\nabla^2 h = \frac{S_s}{k_f} \frac{\delta h}{\delta t} \quad (2.00)$$

wobei:

h	Piezometerhöhe, $[L]$
S_s	spezifischer Speicherkoeffizient, $[L^{-1}]$
k_f	hydraulische Leitfähigkeit, $[Lt^{-1}]$

Bei anisotropen und heterogenen Verhältnissen und nach Einführung des Quellentermes ändert sich Gl.(2.00) zu:

$$\nabla \left(k_{ij} \frac{\delta h}{\delta_j} \right) = S_s \frac{\delta h}{\delta t} + Q_w \quad (2.01)$$

Q_w stellt den Volumenstrom pro Einheitsvolumen dar. Nimmt man ferner an, daß die Achsen des kartesischen Koordinatensystems mit den Achsen der Anisotropie zusammen fallen, so schreibt sich Gl.(2.01) in expandierter Form:

$$\frac{\delta}{\delta x} \left(k_{xx} \frac{\delta h}{\delta x} \right) + \frac{\delta}{\delta y} \left(k_{yy} \frac{\delta h}{\delta y} \right) + \frac{\delta}{\delta z} \left(k_{zz} \frac{\delta h}{\delta z} \right) - Q_w = S_s \frac{\delta h}{\delta t} \quad (2.02)$$

wobei k_{xx} , k_{yy} und k_{zz} die Hauptkomponenten des hydraulischen Leitfähigkeitstensors mit der Einheit $[Lt^{-1}]$ repräsentieren

In dreidimensionalen finiten Differenzen Modellen diskretisiert man gewöhnlich unterschiedliche hydraulische Stockwerke durch eine lagenweise Anordnung von Knoten, /132/,/200/. Multipliziert man eine derartige Einheit mit ihrer Mächtigkeit m , so erhält nach Gl.(2.02):

$$\frac{\delta}{\delta x} \left(T_{xx} \frac{\delta h}{\delta x} \right) + \frac{\delta}{\delta y} \left(T_{yy} \frac{\delta h}{\delta y} \right) + \frac{\delta}{\delta z} \left(mk_{zz} \frac{\delta h}{\delta z} \right) - mQ_w = S' \frac{\delta h}{\delta t} \quad (2.03)$$

wobei T_{xx} und T_{yy} die Hauptkomponenten des Transmissivitätstensors mit $[L^2t^{-1}]$ darstellen und S' für den dimensionslosen Speicherkoeffizienten steht. Die Darcy- oder Filtergeschwindigkeit v_f kann bei bekannter Piezometerhöhenverteilung wie folgt berechnet werden:

$$v_f = k_f \frac{\delta h}{\delta x_i} \quad (2.04)$$

und die Abstandsgeschwindigkeit v_a nach:

$$v_a = \frac{v_f}{\Phi_n} \quad (2.05)$$

2.2 Mechanismen des Wärmetransports

Wärme wird als Energie, die unter einem Temperaturgradienten transportiert wird, definiert. Sie fließt von Regionen höherer Temperatur in Gebiete niedriger Temperatur. Man unterscheidet drei verschiedene Arten des Wärmetransports - Konduktion, Strahlung und Konvektion.

2.2.1 Konduktion

Konduktion oder Wärmeleitung findet innerhalb eines Körpers mit Bereichen unterschiedlicher Temperatur oder zwischen zwei unterschiedlichen Körpern, die in Kontakt miteinander stehen und unterschiedlich temperiert sind, stets entlang des Temperaturgefälles statt. Der physikalische Vorgang der Konduktion spielt sich im molekularen Bereich ab und umfaßt die Übertragung von Energie höherenergetischer Moleküle auf solche, die sich auf einem niedrigeren Energieniveau befinden. Makroskopisch betrachtet ist der Wärmefluß q'' pro Flächeneinheit proportional zum Temperaturgradienten, sodaß man nach dem Fourier Gesetz der Wärmeleitung formulieren kann:

$$q'' = -K \frac{dT}{dx} \quad (2.06)$$

Die Proportionalitätskonstante K wird als Wärmeleitfähigkeitskonstante oder auch kurz als Wärmeleitfähigkeit bezeichnet. Sie ist eine materialspezifische Größe und in der Regel temperaturabhängig. In Kristallen kann

die Wärmeleitfähigkeit für verschiedene kristallographische Richtungen unterschiedliche Werte annehmen, d.h. K wird zum Tensor (*K-Elipsoid*), /38/. Das negative Vorzeichen in (2.06) resultiert aus der Tatsache, daß die Wärme in Richtung des Temperaturgradienten übertragen wird. Berücksichtigt man, daß es sich beim Wärmefluß um einen dreidimensionalen Vorgang handelt, so kann man das *Fourier Gesetz* auch allgemeiner formulieren mit

$$q'' = -K \nabla T \quad (2.07)$$

wobei ∇ für den dreidimensionalen Laplace- Operator steht und es sich bei T um das skalare Temperaturfeld handelt. Betrachtet man den eindimensionalen Wärmefluß senkrecht durch eine ebene Wand in der Richtung x , so liefert die Integration von Gl.(2.06):

$$J = \frac{KA}{\delta x} (T_2 - T_1) \quad (2.08)$$

wobei δx die Wandstärke, T_1 und T_2 die Temperatur an den beiden Rändern und J nach $q'' = J/A$ den Wärmefluß durch eine Fläche bezeichnet. Durch Umformung von Gl.(2.08) und Definition erhält man in Anlehnung an das Ohmsche Gesetz nach:

$$R_w = \frac{\delta x}{KA} \quad (2.09)$$

den Wärmewiderstand. Man kann somit auch formulieren:

$$J = \frac{T_2 - T_1}{\Delta x / KA} = \frac{T_2 - T_1}{R_w} = \frac{\text{thermische Potentialdifferenz}}{\text{thermischer Widerstand}} \quad (2.10)$$

Neben den erwähnten Größen existiert eine weitere, die in enger Beziehung zur Wärmeleitfähigkeit steht und als Temperatur-Leitwert oder thermische Diffusivität α_T bezeichnet wird.

$$\alpha_T = \frac{K}{\rho c} \quad (2.11)$$

Hierbei steht c für die spezifische Wärmekapazität und ρ für die Dichte des betreffenden Körpers oder Stoffes. Der Temperaturleitwert bestimmt die Zeit, die zum Abbau eines Temperaturgefälles über eine Strecke d benötigt wird. Die thermische Relaxationszeit τ berechnet sich somit nach:

$$\tau = \frac{d^2 \rho c}{K} \quad (2.12)$$

Bei vielen Wärmeleitungsprozessen ist die Temperatur nicht konstant, d.h. es fließt in ein Volumenelement mehr Wärme hinein als durch die Ränder entweichen kann. Die Nettobilanz wird durch die Divergenz der Wärmestromdichte beschrieben. Man formuliert daher die allgemeine Wärmeleitungsgleichung nach:

$$\frac{\delta T}{\delta t} = \frac{K}{\rho c} \operatorname{div} \operatorname{grad} T = \frac{K}{\rho c} \nabla^2 T \quad (2.13)$$

2.2.2 Strahlung

Thermische- oder auch Infrarot-Strahlung ist eine elektromagnetische Strahlung, die prinzipiell von jedem Körper als Folge der Wärmebewegung emittiert wird. Sie besitzt also dieselbe Natur wie sichtbares Licht, Röntgen-Strahlen oder Radiowellen. Der Unterschied besteht nur in ihrer Wellenlänge und der Quelle ihrer Entstehung. Während die Wärmeleitung an die Präsenz von Materie gebunden ist, breitet sich thermische Strahlung im Vakuum am besten aus. Für das hier betrachtete Medium und den dazugehörigen Temperaturbereich kann der Wärmetransport über Strahlung als bedeutungslos eingeschätzt werden.

2.2.3 Konvektion

Nimmt man an, daß ein strömendes Fluid keine Wärmeleitfähigkeit besitzt, so hat die rein konvektive Wärmetransportgleichung die Form:

$$-\rho_w C_w q \nabla T + Q_\epsilon = \rho_w C_w \frac{\delta T}{\delta t} \quad (2.14)$$

Strömt dieses Fluid durch ein nichtwärmeleitendes poröses Medium mit einer Porosität Φ , dann erweitert sich Gl.(2.14) zu

$$-\rho_w C_w q \nabla T + Q_\epsilon = [\rho_w \Phi C_w + \rho_g (1 - \Phi) C_g] \frac{\delta T}{\delta t} \quad (2.15)$$

Nimmt man schließlich an, daß der Wärmetransport sowohl konduktiv als auch konvektiv erfolgt und das Fluid sich im Temperaturgleichgewicht mit dem durchströmten Medium befindet, so schreibt man:

$$\nabla \cdot (K \nabla T) - \rho_w C_w q \nabla T + Q_\epsilon = [\rho_w \Phi C_w + \rho_g (1 - \Phi) C_g] \frac{\delta T}{\delta t} \quad (2.16)$$

oder in expandierter Form, wenn die Komponenten des Leitfähigkeitstensors parallel zu den Achsen des kartesischen Koordinatensystems verlaufen:

$$\begin{aligned} & \frac{\delta}{\delta x} \left(K_{xx} \frac{\delta T}{\delta x} \right) + \frac{\delta}{\delta y} \left(K_{yy} \frac{\delta T}{\delta y} \right) + \frac{\delta}{\delta z} \left(K_{zz} \frac{\delta T}{\delta z} \right) - \\ & \rho_w q_x C_w \frac{\delta T}{\delta x} - \rho_w q_y C_w \frac{\delta T}{\delta y} - \rho_w q_z C_w \frac{\delta T}{\delta z} + Q_e = \\ & [\rho_w \Phi C_w + \rho_g (1 - \Phi) C_g] \frac{\delta T}{\delta t} \end{aligned} \quad (2.17)$$

2.3 Konvektion und Diffusion

Der Diffusions- und Konvektionsterm wird numerisch als eine untrennbar mit einander verknüpfte Einheit behandelt. Gemäß /151/ stehen für eine solche Verfahrensweise mehrere Methoden zur Auswahl. Die Grundidee ist die, daß man in Abhängigkeit der lokalen Peclet - Zahl, Konvektion und Diffusion unterschiedlich gewichtet addiert, als einen Term auffaßt. Die dimensionslose Peclet-Zahl Pe stellt das Produkt aus Reynolds-Zahl Re und Prandtl-Zahl Pr bzw. den Quotienten aus Konvektions- und Diffusionskoeffizienten dar. TRADIKON-3D bietet der Vollständigkeit halber die Möglichkeit, alternativ eine der fünf bekannten Wichtungsfunktionen zu berücksichtigen:

Tab. A.1.1 Die Wichtungsfunktion $A(|Pe|)$ für die verschiedene Verfahren nach /151/.

Verfahren	Formel für $A(Pe)$	Referenz
Zentrale Diffenzen	$1 - 0.5 Pe $	
Upwind-Schema	1	COURANT (1952)
Hybrid-Schema	$AMAX1[0, 1 - 0.5 Pe]$	SPALDING (1972)
Power-Law-Schema	$AMAX1[0, (1 - 0.1 Pe)^5]$	PATANKAR (1979)
Exponential-Schema	$ Pe / (e^{ Pe } - 1)$	PATANKAR (1980)

Hierbei bedeuten $|Pe|$ den Absolutbetrag der Peclet-Zahl und die aus der Programmiersprache FORTRAN stammende Funktion $AMAX1$ die größte Zahl der in dem darauffolgenden Klammerausdruck aufgeführten Argumente. Alle Verfahren führen bei kleinen Peclet-Zahlen, d.h. $|Pe| < 2$, zu physikalisch realistischen Ergebnissen. Da die Peclet-Zahl auch eine Funktion des Knotenabstandes ist und man aus Gründen der Ökonomie nicht beliebig fein diskretisieren kann, sollte dem Exponential- oder dem Power-Law-Schema der Vorzug eingeräumt werden, /151/.

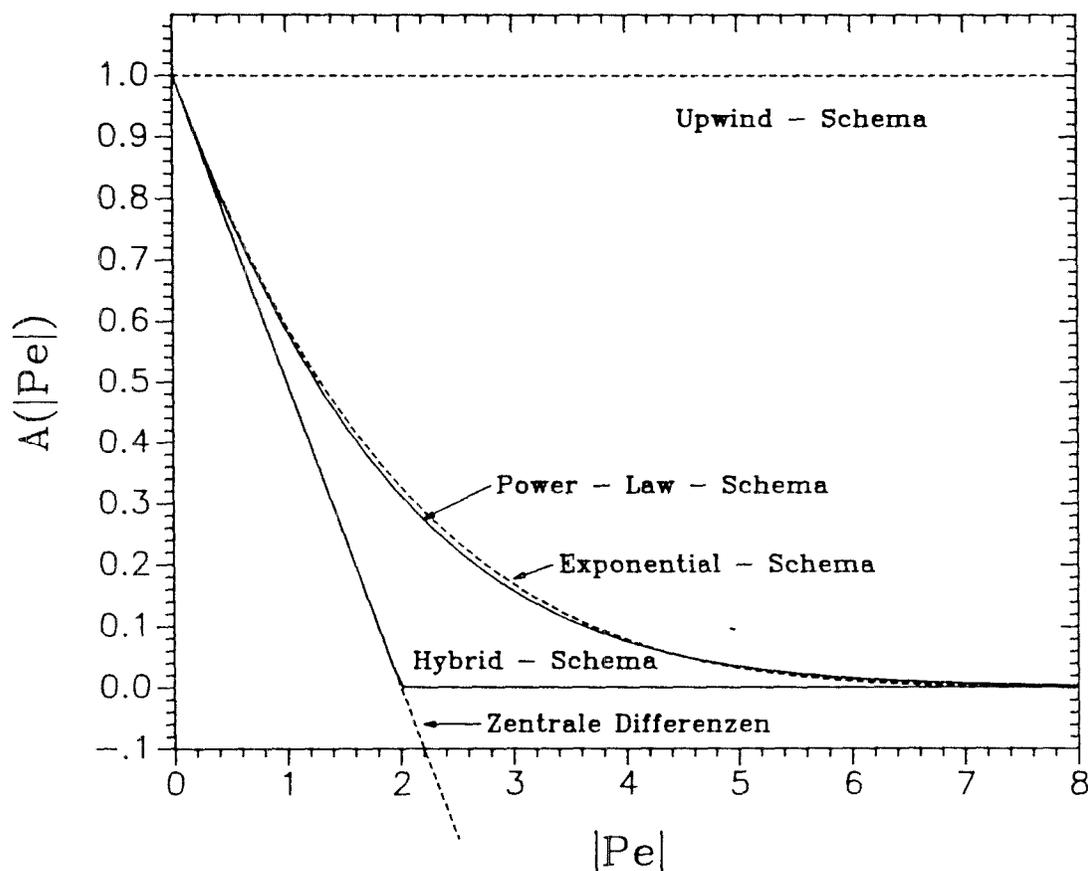


Abb. 1: Verlauf der Funktion $A(|Pe|)$ für die in Tab. 1 aufgeführten Verfahren

Ein speziell bei der Simulation von Schadstofftransportvorgängen auftretender, unangenehmer Begleiteffekt ist das Phänomen der *numerischen Dispersion*, /106/, auch *falsche Diffusion*, /151/, genannt. Sie läßt sich besonders gut bei der Simulation der Ausbreitung eines dispersionsfreien Tracers diagonal durch ein finites Differenzen Raster beobachten. Da-

durch, daß der somit nur rein konvektiv erfolgende Transport verfahrensbedingt nur senkrecht zu den Zellengrenzflächen des Rasters erfolgen kann, kommt es zu einer unrealistischen Aufspreizung der Schadstoffahne, deren Ausmaß seinerseits eine Funktion der Gitter-Peclet-Zahl ist. Bei den hier betrachteten Vorgängen des Wärmetransportes in porösen Medien tritt der Effekt der numerischen Dispersion nur bei sehr hohen Fließgeschwindigkeiten auf. Sofern man in solchen Fällen Kenntnis von dem Strömungsfeld hat, kann er durch eine parallel zur Fließrichtung verlaufenden Orientierung einer Achse des finiten Differenzen - Koordinatensystems minimiert werden.

2.4 Diffusion und Dispersion in porösen Medien

Eine Flüssigkeit bewegt sich in porösen oder klüftigen Medien auf tortuosen Bahnen. Die daraus resultierende makroskopische Vermischung der Flüssigkeit wird als mechanische Dispersion bezeichnet. Dieser Vorgang unterscheidet sich von der molekularen Diffusion, die eine mikroskopische Natur besitzt. Der gesamte Dispersionsvorgang spaltet sich also in eine molekulare Diffusion und eine mechanische Dispersion auf.

Für einen nichtisothermalen Fließvorgang in einem porösen Medium mit einem gegebenen Konzentrationsgradienten läßt sich der effektive hydrodynamische Dispersionskoeffizient D_e und der effektive thermische Dispersionskoeffizient K_e beschreiben mit

$$D_e = D_o + D' \quad \text{bzw.} \quad K_e = K + K' \quad (2.18)$$

Hierbei bedeuten D_o und D' die Massendifusions- und Masserdispersionskoeffizienten, K und K' die thermische Leitfähigkeit bzw. den thermischen Dispersionskoeffizienten des flüssigkeitsgefüllten Mediums.

Der Dispersionsvorgang besitzt eine anisotrope Natur, wobei der longitudinale Dispersionskoeffizient D_L meist ein bis zwei Größenordnungen über dem transversalen Dispersionskoeffizienten D_T liegt, /19/,/180/.

Die Korrelationsgleichung für den longitudinalen Massendispersionskoeffizienten läßt sich nach /41/ folgendermaßen formulieren:

$$\frac{D'_L}{D_o} = C_1 (Pe)^m \quad (2.19)$$

mit $C_1 = 0.5$ und $1 < m < 1.2$ für $Pe < 10^2$ bzw. $C_1 = 1.8$ und $m = 1$ für $10^2 < Pe < 10^5$. Pe entspricht hierbei der molekularen Peclet-Zahl

für den Massentransport mit $Pe = vd/D_o$, d = Korndurchmesser, v = Darcy-Geschwindigkeit und D_o = Massendiffusionskoeffizient.

Es bestehen zwei Hauptunterschiede zwischen dem thermischen Dispersionskoeffizienten und dem Massendispersionskoeffizienten: Erstens erfolgt der Transport von Wärme in porösen Medien auch — in der Regel sogar zum deutlich überwiegenden Anteil — über die Gesteinsmatrix während dies beim Massentransport nicht möglich ist und zweitens rangiert der thermische Diffusionskoeffizient etwa drei Größenordnungen über dem Massendiffusionskoeffizienten, /19/.

Während selbst bei sehr kleinen Fließgeschwindigkeiten Massendispersionseffekte außerordentlich wichtig sind, spielen thermische Dispersionsvorgänge bei kleinen Fließgeschwindigkeiten keine Rolle. Deswegen wird in der vorliegenden Arbeit die thermische Dispersion vernachlässigt.

2.5 Phasenwechsel von Wasser

Da das hier vorgestellte Modell in der Lage sein sollte, die Betriebsbedingungen einer Erdsondenwärmepumpenanlage so wirklichkeitsnah wie möglich zu simulieren, kam der Phasenumwandlung des im Boden oder Gestein enthaltenen Wassers eine gewisse Bedeutung zu. Bei der Phasenumwandlung von Wasser in Eis wird eine Wärmeenergie von 6030 kJ/mol oder $3.336 \cdot 10^5 \text{ J/kg}$ freigesetzt, eine Energie, die ausreicht, eine entsprechende Stoffmenge Wasser von 0 auf etwa $78 \text{ }^\circ\text{C}$ zu erwärmen. Unter Laborbedingungen gefriert reines Wasser erst, wenn diese Energie, die sogenannte Schmelzwärme oder -enthalpie, aufgezehrt wurde. Derartige Bedingungen lassen sich durch einen isothermalen Algorithmus simulieren, bei dem die Temperatur des betreffenden Knotens solange konstant gehalten wird, bis die Nettobilanz aller zu- und abgeflossenen Energieströme zur Regeneration (beim Auftauen) oder zum Aufzehren (beim Gefrieren) der Schmelzenthalpie ausgereicht hat.

Bei der Berechnung eines instationären Temperaturfeldes erhält man durch die Anwendung des obigen Verfahrens stets einen plateauartigen Temperaturverlauf, der sich jedoch beim Gefrieren von Böden nicht beobachten läßt. Dafür sind eine Vielzahl von Gründen verantwortlich von denen hier nur einige erwähnt werden:

- Porenwasser ist stets, wenn auch in unterschiedlichem Maße, mineralisiert, was zu einer Schmelzpunkterniedrigung führt

- der Schmelzpunkt des Porenwassers ist abhängig vom Wassergehalt des Bodens, /145/
- der Boden weist eine Vielzahl von unterschiedlich großen Poren auf, in denen das darin enthaltene Wasser bei unterschiedlichen Temperaturen gefriert; das Haftwasser gefriert z.T. erst deutlich unterhalb von $-10\text{ }^{\circ}\text{C}$
- das nicht drainierbare Porenvolumen gestattet nur eine eingeschränkte Seitenausdehnung des beim Gefrieren sein Volumen vergrößernden Wassers; dies führt zu einer Druckerhöhung und damit zu einer Schmelzpunkterniedrigung.

Ausgehend von der Überlegung, daß eine relativ einfache e -Funktion in der Lage sein müßte, die mit abnehmender Temperatur allmählich stattfindende Umwandlung von Wasser in Eis hinreichend exakt nachzubilden, wurde das im Folgenden diskutierte Verfahren entwickelt:

Grundvoraussetzung ist zunächst eine möglichst genaue Kenntnis von dem Zweiphasendiagramm (Wasser/Eis) des zu berechnenden Entzugsmediums.

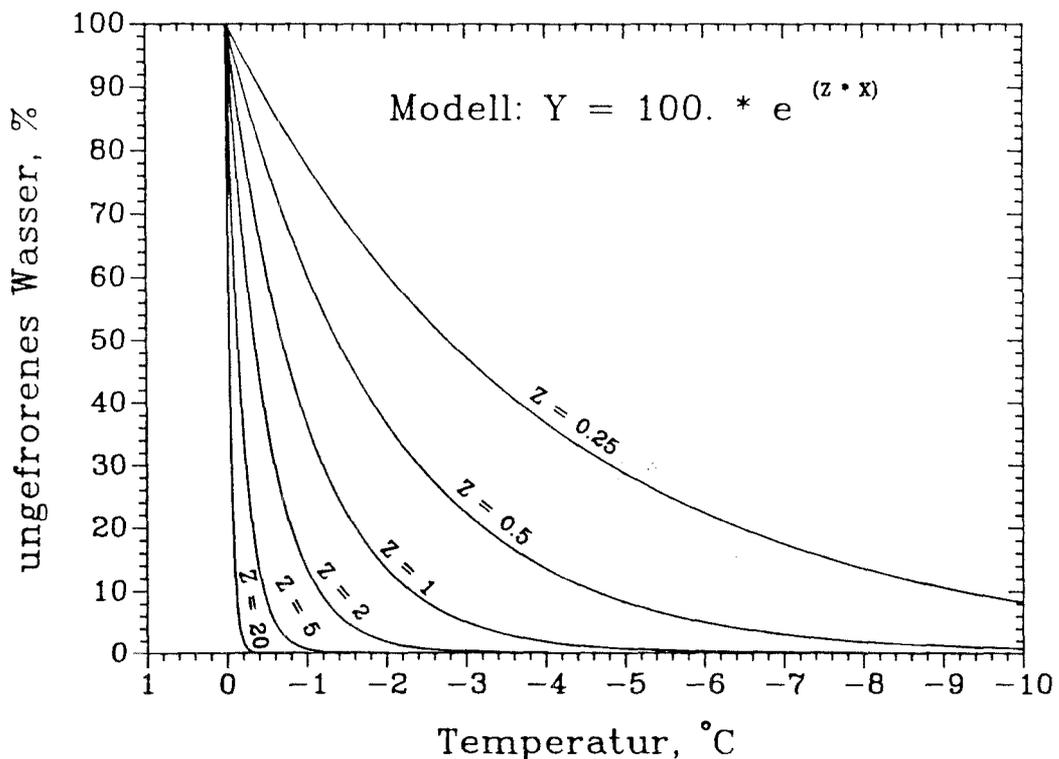


Abb. 2: Beginn der Phasenumwandlung von Wasser; hier bei $0\text{ }^{\circ}\text{C}$.

Die in Abb. 2 dargestellten Kurvenscharen markieren den Anteil des noch ungefrorenen Wassers. Die Kurven mit $Z = 1$ bis $Z = 0.25$ könnten auf einen tonig-schluffigen Boden zutreffen, während die Kurven mit $Z = 20$ bis $Z = 2$ für ein eher grobklastisches Material der Kies- bis Sandfraktion sprechen. Die in der Literatur /64/ dokumentierten Gefrierversuche an Böden scheinen diese Annahme zu bestätigen.

Rechnet man nun die Schmelzenthalpie des noch verbleibenden, ungefrorenen Wasser gemäß:

$$C = C_w * X_w + C_e * X_e + \frac{1}{\Delta T} \int_T^{T+\Delta T} \Delta H_f \frac{\delta x_u}{\delta T} dT \quad (2.17)$$

wobei:

C_w, C_e	Wärmekapazität von Wasser und Eis
X_w, X_e	Volumenanteile von Wasser und Eis
H_f	Schmelzwärme pro Masseneinheit des noch ungefrorenen Wassers
δx_u	Änderung des Volumenanteils von ungefrorenem Wasser über das Temperaturintervall δT

auf die Wärmekapazität des Wasser um, so resultieren die in Abb. 3 dargestellten Kurvenscharen.

Im Modell wird nun so verfahren, daß bei ablaufendem Phasenwechsel nicht nur die Temperatur des betreffenden Knotens sondern auch der Anteil des ungefrorenen Wassers und damit die effektive Wärmekapazität iterativ bestimmt werden muß. Dabei darf die maximale Temperaturdifferenz δT ein bestimmtes Maß nicht überschreiten, da es sonst zu erheblichen Berechnungsfehlern kommt.

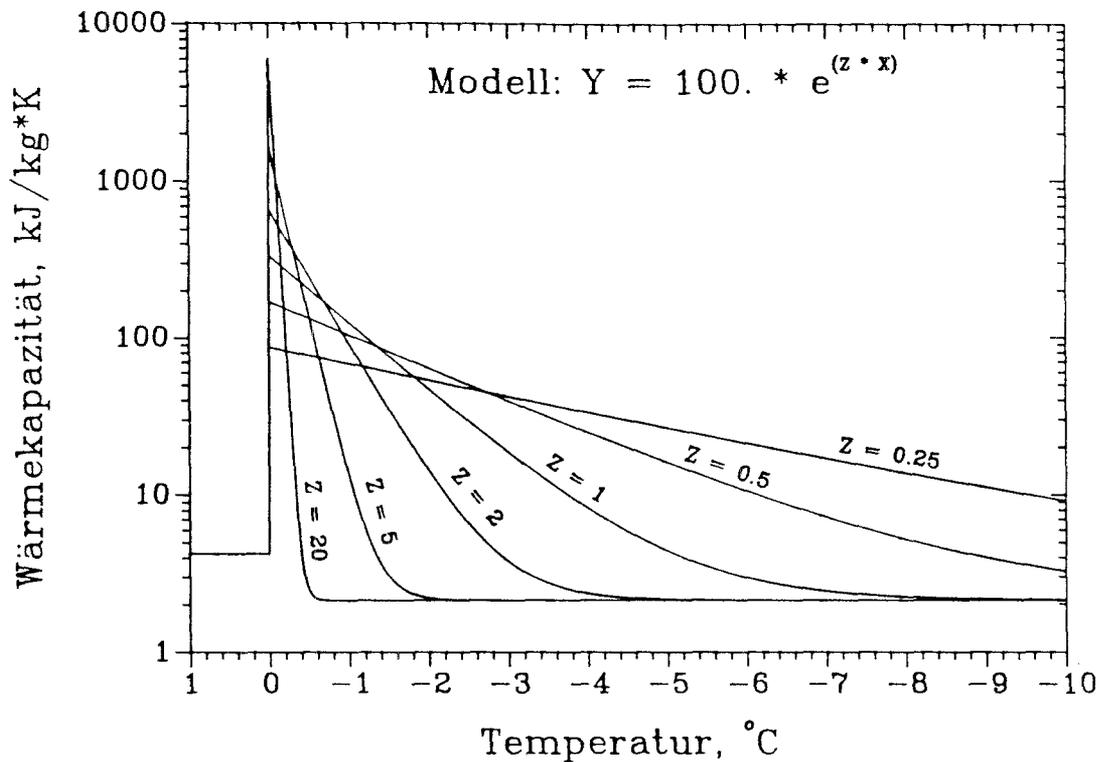


Abb. 3: Berechnete Wärmekapazität von teilgefrorenem Wasser; die Ordinate wurde aus Gründen der Übersichtlichkeit logarithmisch dargestellt.

3 Numerische Behandlung der Gleichungen

Aufgrund der Komplexität der physikalischen Phänomene, die beim Wärmetransport im Erdreich wirksam sind, lassen sich für deren Berechnung nur unter sehr starken Vereinfachungen analytische Lösungen herleiten. Man ist also auf ein leistungsfähiges numerisches Verfahren angewiesen, das in der Lage sein muß, auch komplizierte Anfangs- und Randbedingungen zu bewältigen. Die gebräuchlichsten numerischen Verfahren zur Berechnung der Strömung und des Stofftransportes in Aquiferen sind:

- die finiten Differenzen (FD)
- die finiten Elemente (FE)

Da diese Verfahren speziell bei der Lösung der Transportgleichung einige Nachteile aufweisen (numerische Dispersion, Oszillationen, ...) haben sich hier das auf stochastischen Ansätzen beruhende

- "Random - Walk" - Verfahren /161/

und das von der Berechnung von Druckstoßvorgängen her bekannte und speziell für Fälle, in denen der Konvektionsterm überwiegt, entwickelte

- Charakteristiken - Verfahren /111/

fest etabliert. Da bei der hier betrachteten Thematik, anders als bei der numerischen Behandlung von Schadstofftransportvorgängen, das konvektive Glied im Normalfall nicht dominiert, erschien die Methode der finiten Differenzen als eine brauchbare Methode, die oben erwähnten Gleichungen zu lösen.

3.1 Konstruktion der finiten Differenzen Approximation

Das am häufigsten eingesetzte Verfahren zur Ableitung finiter Differenzen Gleichungen besteht aus der Approximation der Ableitung der Differenzialgleichung über eine gekappte Taylor-Reihe /167/. Für eine Funktion $f(x)$ hat diese Reihenentwicklung in der positiven X -Richtung die Form:

$$f(x + \Delta x) = f(x) + \Delta x \frac{df}{dx} + \frac{(\Delta x)^2}{2!} \frac{d^2 f}{dx^2} + \frac{(\Delta x)^3}{3!} \frac{d^3 f}{dx^3} + \dots \quad (3.00)$$

Durch Auflösen nach df/dx erhält man:

$$\frac{df}{dx} = \frac{f(x + \Delta x) - f(x)}{\Delta x} + \Theta(\Delta x) \quad (3.01)$$

wobei $\Theta(\Delta x)$ den verbleibenden Term der Serie darstellt. Schneidet man diesen Term ab, so resultiert mit:

$$\frac{df}{dx} \approx \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (3.02)$$

der vordere Differenzenquotient. In ähnlicher Weise erhält man für die negative X -Richtung die Taylor Reihe:

$$f(x - \Delta x) = f(x) - \Delta x \frac{df}{dx} + \frac{(\Delta x)^2}{2!} \frac{d^2 f}{dx^2} - \frac{(\Delta x)^3}{3!} \frac{d^3 f}{dx^3} + \dots \quad (3.03)$$

Durch Auflösen nach df/dx erhält man unter Vernachlässigung von $\Theta(\Delta x)$:

$$\frac{df}{dx} \approx \frac{f(x - \Delta x) - f(x)}{\Delta x} \quad (3.04)$$

den hinteren Differenzenquotienten. Subtrahiert man (3.03) von (3.00), so ist das Ergebnis:

$$f(x + \Delta x) - f(x - \Delta x) = 2\Delta x \frac{df}{dx} + \frac{2(\Delta x)^3}{3!} \frac{d^3 f}{dx^3} + \dots \quad (3.05)$$

Schneidet man die Taylor Reihen jeweils nach dem dritten Term ab, so kann aus (3.05) die zentrale Differenzen Approximation:

$$\frac{df}{dx} \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} \quad (3.06)$$

abgeleitet werden. Um eine Näherung für die zweite Ableitung von f zu erhalten, werden die Gleichungen (3.00) und (3.03) addiert:

$$f(x + \Delta x) - f(x - \Delta x) = 2f(x) + \frac{2(\Delta x)^2}{2!} \frac{d^2 f}{dx^2} + \Theta((\Delta x)^4) \quad (3.07)$$

Die gewünschte Approximation resultiert mit einem Fehler von $\Theta((\Delta x)^2)$:

$$\frac{d^2 f}{dx^2} \approx \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{(\Delta x)^2} \quad (3.08)$$

3.1.1 Der Explizit-Ansatz

Angewandt auf ein eindimensionales Randwertproblem kann man nun schreiben:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} + \Theta(\Delta t) = \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2} + \Theta((\Delta x)^2) \quad (3.09)$$

Unter Vernachlässigung der Terme $\Theta(\Delta t)$ und $\Theta((\Delta x)^2)$ ergibt sich:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2} \quad (3.10)$$

Das Symbol j bezeichnet den Zeitschrittindex und i den Ortsindex. Nach u_i^{j+1} aufgelöst erhält man:

$$u_i^{j+1} = u_i^j + \frac{\Delta t}{(\Delta x)^2} \left(u_{i+1}^j - 2u_i^j + u_{i-1}^j \right) \quad (3.11)$$

Demnach werden die neuen Werte von u_i ausschließlich aus solchen des vorherigen Zeitschrittes berechnet. Diese explizite Formulierung wird in der Literatur bisweilen auch als Vorwärtsdifferenzen - Verfahren bezeichnet. Es erfordert einen minimalen Aufwand an Rechenoperationen und Speicherplatz, hat aber den Nachteil, daß eine bestimmte Zeitschrittlänge nicht überschritten werden darf. Bei der Berechnung des instationären, konduktiven Wärmetransports darf z.B. das folgende Kriterium nicht verletzt werden /151/:

$$\Delta t < \frac{\rho c (\Delta x)^2}{2K} \quad (3.12)$$

Für die zweidimensionale Strömungsberechnung lautet das Stabilitätskriterium, /106/:

$$\frac{T_{xy}}{S} \left(\frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2} \right) \leq \frac{1}{2} \quad (3.13)$$

bzw.:

$$\Delta t \leq \frac{S}{2T_{xy} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)} \quad (3.14)$$

Bei der konvektiv-diffundiven Wärmetransportberechnung muß zusätzlich noch das Courant-Kriterium erfüllt sein:

$$Co_x = \left| \frac{\Delta t v_x}{\Delta x} \right| \leq 1 \quad (3.15a)$$

$$Co_y = \left| \frac{\Delta t v_y}{\Delta y} \right| \leq 1 \quad (3.15b)$$

$$Co_z = \left| \frac{\Delta t v_z}{\Delta z} \right| \leq 1 \quad (3.15c)$$

Co steht hierbei für die *Courant - Zahl*. Dieses Kriterium besagt, daß auf konvektiven Wege nicht mehr Wärmeenergie — oder Schadstoffmenge — eine Zelle über den Zeitschritt $[t, t + \Delta t]$ verlassen darf, als zu Beginn dieses Zeitschrittes in der betreffenden Zelle gespeichert war.

Diese Einschränkungen bezüglich der Wahl von Δt führt bei vielen Anwendungen zu unrealistisch kleinen Zeitschrittlängen und damit zu einem enormen Rechenaufwand.

3.1.2 Der Implizit-Ansatz

Bei der impliziten Approximation, auch Rückwärtsdifferenzen - Verfahren genannt, schreibt man:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{(\Delta x)^2} \quad (3.16)$$

oder aufgelöst nach u_i^{j+1} :

$$u_i^{j+1} = \frac{u_i^j + \frac{\Delta t}{(\Delta x)^2} (u_{i+1}^{j+1} + u_{i-1}^{j+1})}{1 + 2\Delta t / (\Delta x)^2} \quad (3.17)$$

Der Hauptvorteil des Implizit-Verfahrens ist darin zu sehen, daß es für beliebige Zeitschrittängen – im Gegensatz zum Explizit-Verfahren – stabil bleibt, /167/. Mit dem Implizit-Ansatz steht ein Verfahren zur Verfügung, mit dem man die Zeitschrittlänge Δt unabhängig von der Maschenweite des finiten Differenzenrasters wählen kann. Da jedoch die Geschwindigkeit, mit der eine finiten Differenzen Approximation zur Konvergenz führt, auch von dem Rundungsfehler abhängt, gibt es eine weitere Verbesserung — den Crank-Nicolson-Ansatz, /53/.

3.1.3 Der Crank-Nicolson Ansatz

Nach den Gleichungen (3.01) bzw. (3.04), den Näherungen für $\delta x / \delta t$ über den vorderen bzw. hinteren Differenzenquotienten, beträgt der Rundungsfehler jeweils $\Theta(\Delta t)$, nach (3.08) für den zentralen Differenzenquotienten jedoch nur $\Theta((\Delta t)^2)$. Die einfachste Integration des zentralen Differenzenquotienten, bei der die Zeitschritte $[t - \Delta t, t]$ und $[t, t + \Delta t]$ die Ableitung nach der Zeit ersetzen, führt zu der Näherung:

$$\frac{u_i^{j+1} - u_i^{j-1}}{2\Delta t} = \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\delta x)^2} \quad (3.18)$$

Dieses Verfahren neigt allerdings sehr schnell zur Instabilität, weshalb es auch nicht eingesetzt werden sollte. Diesen Mangel kann jedoch umgehen, indem man für den Zeitschritt $j + 1/2$ die zentrale Differenzen Approximation $(u_i^{j+1} - u_i^j) / \Delta t$ einführt. Dies läßt sich durch eine gewichtete

Mittelung der Näherung von $\delta^2 u / \delta x^2$ für die Zeitschritte j und $j + 1$ bewerkstelligen. Bezeichnet man den Wichtungsfaktor mit α , wobei $0 \leq \alpha \leq 1$, so kann man wiederum durch Anwendung der Taylor Reihenentwicklung unter Vernachlässigung des Rundungsfehlers herleiten:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{\alpha(u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}) + (1 - \alpha)(u_{i+1}^j - 2u_i^j + u_{i-1}^j)}{(\Delta x)^2} \quad (3.19)$$

Für $\alpha = 1/2$ erhält man schließlich die Crank-Nicolson Approximation:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{0.5 (u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}) + 0.5 (u_{i+1}^j - 2u_i^j + u_{i-1}^j)}{(\Delta x)^2} \quad (3.20)$$

Die explizite Approximation erhält man für $\alpha = 0$, die implizite, indem man $\alpha = 1$ setzt. Abschließend sei noch darauf hingewiesen, daß die Crank-Nicolson Approximation — ebenso wie der Implizit-Ansatz — ungeachtet der gewählten Zeitschrittlänge zwar stabil bleibt, diese Stabilität jedoch noch lange keine Garantie für physikalisch realistische Berechnungsergebnisse bietet /151/, /190/.

3.2 Möglichkeiten der Diskretisierung

Das hier beschriebene Modell benutzt kartesische Koordinaten mit einer variablen Diskretisierung aller drei Raumkoordinaten, wobei — je nach Fragestellung — entweder die Knotenpunkte zellenzentriert oder die Zellengrenzen knotenzentriert angeordnet werden können. Zur Veranschaulichung seien in den Abb. 4 und 5 für eine X - Y - Ebene die Anordnungen für eine $10 * 10$ Knoten Diskretisierung aufgezeigt.

Legt man besonders großen Wert darauf, daß die berechneten Temperaturen oder Potentiale gut durch die Knoten repräsentiert werden, empfiehlt sich die Verwendung von zellenzentrierten Knoten. Mit knotenzentrierten Zellengrenzen lassen sich hingegen exaktere Berechnungen der Wärmeübergänge zwischen den Zellen realisieren, /151/. Es gibt verschiedene Möglichkeiten der Knotenindizierung. Bei der eindimensionalen Knotenindizierung werden die Knoten fortlaufend, z.B. von links oben nach rechts unten durchnumeriert. Dieses Verfahren hat Vorteile bezüglich der Rechengeschwindigkeit und einer effizienten Speicherplatzverwaltung, erfordert allerdings einen

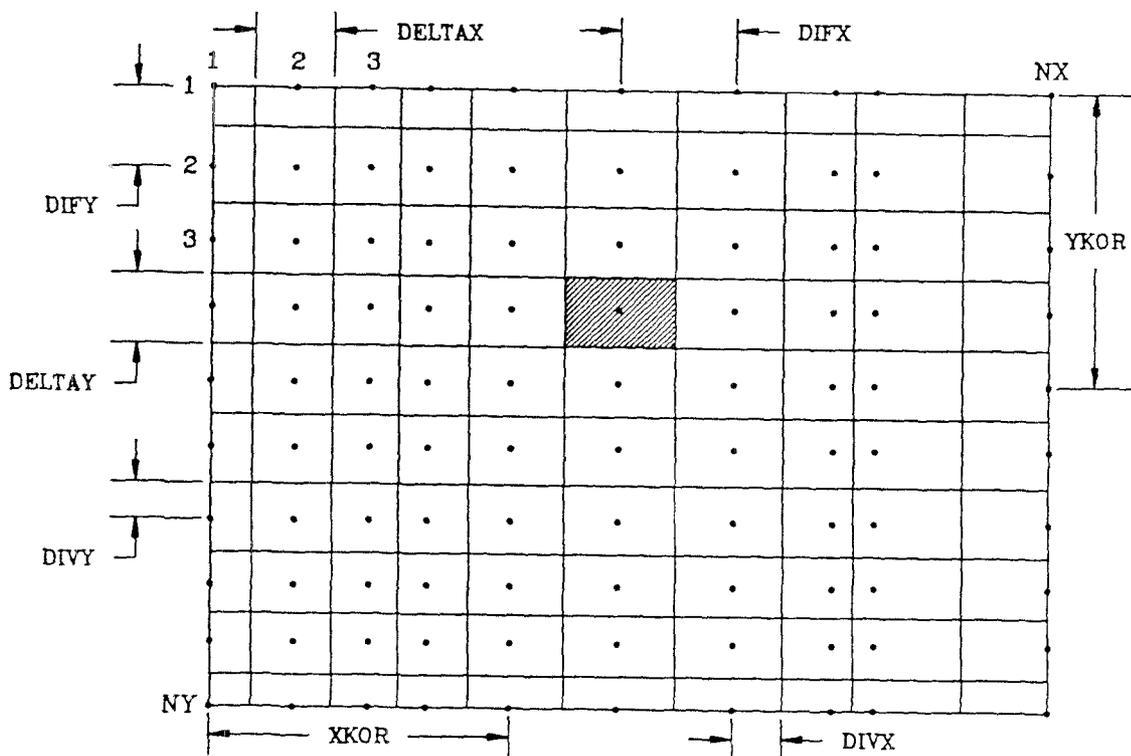


Abb. 4: Knotenzentrierte Anordnung der Zellengrenzen; die Bezeichnungen beziehen sich auf die im Programm verwandten Variablennamen

etwas höheren Programmieraufwand und erschwert außerdem die Fehlersuche während der Programmierphase. Das hier beschriebene Modell benutzt Dreifachindizes, wobei die Numerierung der Knoten entlang eines 180 Grad um die X -Achse auf den Betrachter zugeordneten Koordinatensystems erfolgt.

Vereinbarungsgemäß werden die sechs Grenzflächen der finiten Differenzen Zelle des Knotens $P_{i,j,k}$ mit "halben" Indizes, nämlich $(i - 1/2, j, k)$, $(i + 1/2, j, k)$, $(i, j - 1/2, k)$, $(i, j + 1/2, k)$, $(i, j, k - 1/2)$ und $(i, j, k + 1/2)$, versehen. Sämtliche Flüsse J , sowohl Fluid- als auch Wärmefluß, durch diese Zellengrenzen werden analog indiziert.

Alle Eigenschaften, die sich auf die Zellengrenzflächen beziehen, werden aus dem gewichteten, harmonischen Mittel der Eigenschaften zweier Nachbarzellen berechnet. So gilt z.B. für die hydraulische Leitfähigkeit der

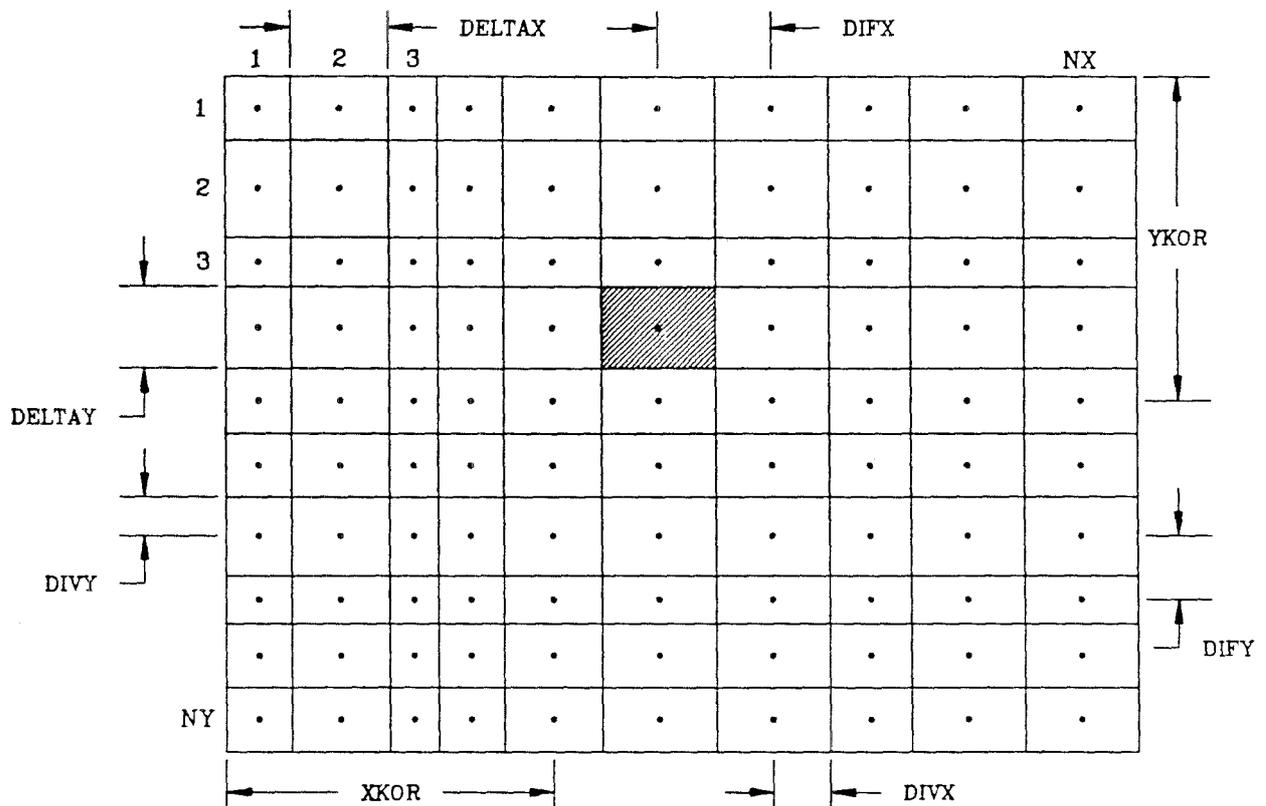


Abb. 5: Zellenzentrierte Anordnung der Knoten; die Bezeichnungen beziehen sich auf die im Programm verwandten Variablennamen

Zellengrenzfläche $(i + 1/2, j, k)$:

$$(k_{xx})_{i+1/2,j,k} = \frac{(k_{xx})_{i,j,k} (k_{xx})_{i+1,j,k} (\delta x_i + \delta x_{i+1})}{(k_{xx})_{i,j,k} \delta x_i + (k_{xx})_{i+1,j,k} \delta x_{i+1}} \quad (3.20)$$

wobei δx_i und δx_{i+1} den Strecken von den Knoten zur betreffenden Zellengrenzfläche entsprechen. Dieses Verfahren bietet gegenüber dem arithmetischen Mittel den Vorteil, daß z.B. zwischen einer durchlässigen Zelle und einer nichtleitenden Zelle kein Fluß erfolgen kann.

3.3 Die Diskretisierungsgleichungen

Bei der Formulierung der gesuchten Diskretisierungsgleichungen erschien es ratsam, für die Nachbarkoeffizienten eine aus der angelsächsischen Literatur, /151/, /190/ bekannte Indizierung einzuführen. Im dreidimensionalen Fall hat jeder Knoten P , abgesehen davon, daß er an den Rändern

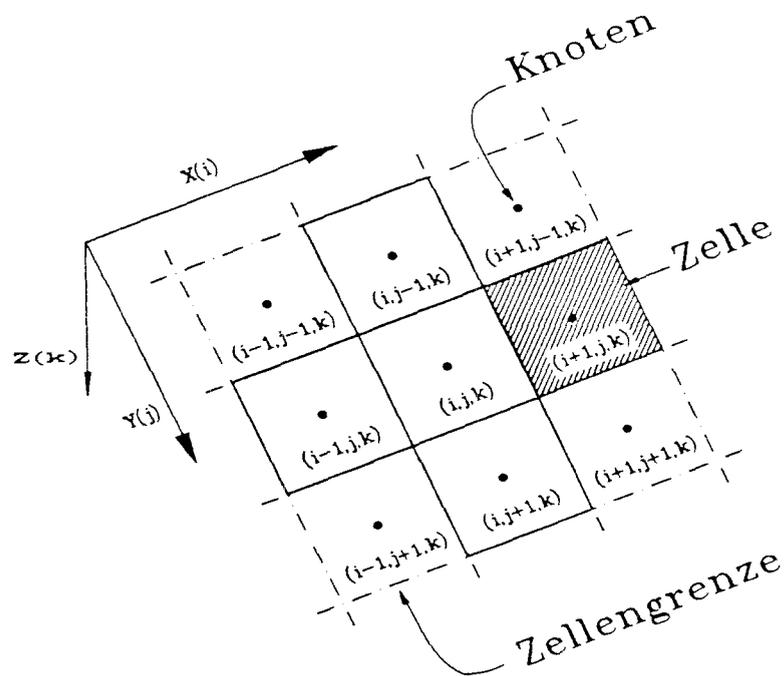


Abb. 6: Knotenindizierung der Schicht k

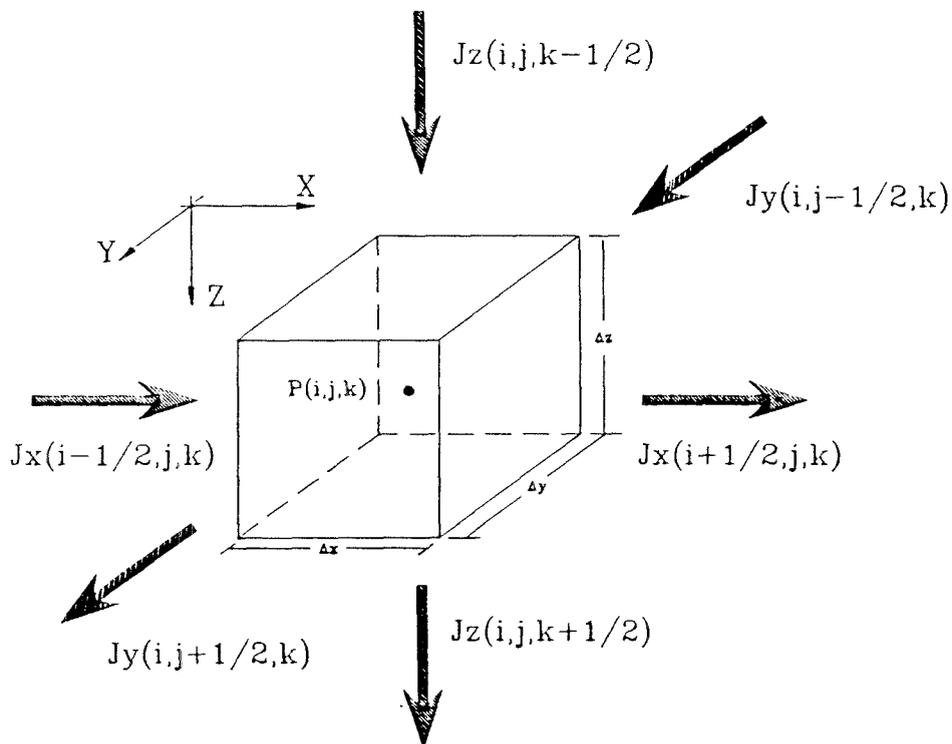


Abb. 7: Fluß durch eine dreidimensionale finite Differenzen Zelle

oder Eckpunkten des Modellquaders liegt, sechs Nachbarn. Den vier Nachbarn in der horizontalen Ebene werden in Analogie zu den Himmelsrichtungen einer Kompassrose je nach der Lage zum Knoten P die Indizes N (Norden), S (Süden), E (Osten) und W (Westen) zugeordnet. Die restli-

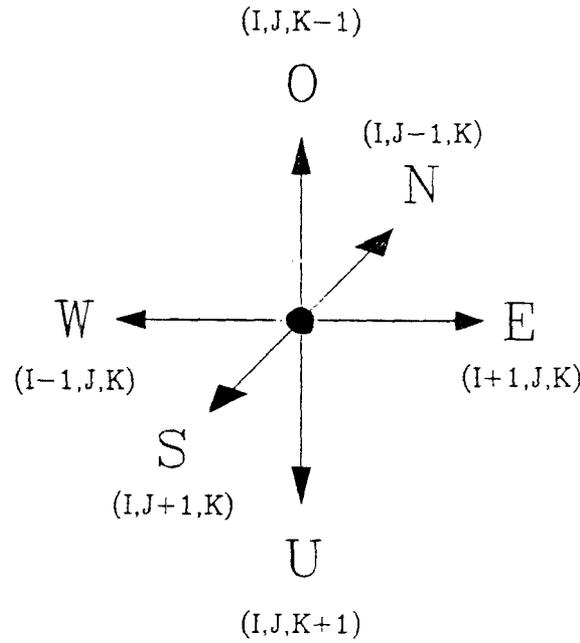


Abb. 8: Indizierung der sechs Nachbarn des Knotens P

chen beiden Knoten erhalten dann die Indizes O (Oben) und U (unten).

Die für den Knoten P aufzulösende Diskretisierungsgleichung des konduktiv - konvektiven Wärmetransportes hat gemäß /151/ die Form:

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S + a_O T_O + a_U T_U + b \quad (3.21)$$

oder:

$$T_P = \frac{a_E T_E + a_W T_W + a_N T_N + a_S T_S + a_O T_O + a_U T_U + b}{a_P} \quad (3.22)$$

wobei:

$$a_W = D_w A(|Pe_w|) + [F_w, 0] \quad (3.22a)$$

$$a_E = D_e A(|Pe_e|) + [-F_e, 0] \quad (3.22b)$$

$$a_N = D_n A(|Pe_n|) + [F_n, 0] \quad (3.22c)$$

$$a_S = D_s A(|Pe_s|) + [-F_s, 0] \quad (3.22d)$$

$$a_O = D_o A(|Pe_o|) + [F_o, 0] \quad (3.22e)$$

$$a_U = D_u A(|Pe_u|) + [-F_u, 0] \quad (3.22f)$$

$$a_P^0 = \frac{\rho_P \Delta x \Delta y \Delta z}{\Delta t} \quad (3.23)$$

$$b = Q_\epsilon \Delta x \Delta y \Delta z + a_P^0 T_P \quad (3.24)$$

$$a_P = a_E + a_W + a_N + a_S + a_O + a_U + a_P^0 \quad (3.25)$$

Die Diffusions- und Konvektionskoeffizienten werden wie folgt definiert:

$$F_w = (\rho v_x)_w \Delta y \Delta z \quad D_w = \frac{\Gamma_w \Delta y \Delta z}{(\delta x)_w} \quad (3.26a)$$

$$F_e = (\rho v_x)_e \Delta y \Delta z \quad D_e = \frac{\Gamma_e \Delta y \Delta z}{(\delta x)_e} \quad (3.26b)$$

$$F_n = (\rho v_y)_n \Delta x \Delta z \quad D_n = \frac{\Gamma_n \Delta x \Delta z}{(\delta y)_n} \quad (3.26c)$$

$$F_s = (\rho v_y)_s \Delta x \Delta z \quad D_s = \frac{\Gamma_s \Delta x \Delta z}{(\delta y)_s} \quad (3.26d)$$

$$F_o = (\rho v_z)_o \Delta x \Delta y \quad D_o = \frac{\Gamma_o \Delta x \Delta y}{(\delta z)_o} \quad (3.26e)$$

$$F_u = (\rho v_z)_u \Delta x \Delta y \quad D_u = \frac{\Gamma_u \Delta x \Delta y}{(\delta z)_u} \quad (3.26f)$$

wobei die dimensionslosen Peclet-Zahlen aus dem Quotienten des konvektiven und diffundiven Gliedes hervorgehen:

$$Pe_e = \frac{F_e}{D_e} \quad Pe_w = \frac{F_w}{D_w} \quad (3.27a)$$

$$Pe_n = \frac{F_n}{D_n} \quad Pe_s = \frac{F_s}{D_s} \quad (3.27b)$$

$$Pe_o = \frac{F_o}{D_o} \quad Pe_u = \frac{F_u}{D_u} \quad (3.27c)$$

Die Funktion $A(|Pe|)$ kann Tabelle 1 entnommen werden. Die für den Knoten P aufzulösende Diskretisierungsgleichung der Strömungsberechnung hat die Form:

$$a_P h_P = a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_O h_O + a_U h_U + b \quad (3.28)$$

oder:

$$h_P = \frac{a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_O h_O + a_U h_U + b}{a_P} \quad (3.29)$$

wobei

$$a_w = T x_w \frac{\Delta y \Delta z}{(\delta x)_w} \quad (3.30a)$$

$$a_e = T x_e \frac{\Delta y \Delta z}{(\delta x)_e} \quad (3.30b)$$

$$a_n = T y_n \frac{\Delta x \Delta z}{(\delta y)_n} \quad (3.30c)$$

$$a_s = T y_s \frac{\Delta x \Delta z}{(\delta y)_s} \quad (3.30b)$$

$$a_o = m k z_o \frac{\Delta x \Delta y}{(\delta z)_o} \quad (3.30e)$$

$$a_u = m k z_u \frac{\Delta x \Delta y}{(\delta z)_u} \quad (3.30f)$$

$$a_P^0 = \frac{S' \Delta x \Delta y}{\Delta t} \quad (3.31)$$

$$b = Q_w + a_P^0 h_P \quad (3.32)$$

$$a_P = a_E + a_W + a_N + a_S + a_O + a_U + a_P^0 \quad (3.33)$$

wobei m der Mächtigkeit der zu berechnenden hydraulischen Einheit entspricht. Die lokalen Transmissivitäten T_i werden aus dem Produkt des geometrischen Mittels der aktuellen Aquiferfermächigkeiten zweier benachbarter Knoten und der entsprechenden Grenzflächenleitfähigkeit ermittelt, /106/. Bei äquidistanten Knotenabständen liest sich dieses Berechnungsschema z.B. für die östliche Zellengrenzfläche, des Knotens $P_{i,j,k}$ mit der Aquiferbasis $bo_{i,j,k}$:

$$T x_e = k x_e \sqrt{(h_{i,j,k} - bo_{i,j,k}) (h_{i+1,j,k} - bo_{i+1,j,k})} \quad (3.34)$$

Die hydraulische Grenzflächenleitfähigkeit kx_e wird durch das harmonische Mittel der Leitfähigkeiten der entsprechenden Nachbarknoten berechnet:

$$kx_e = \frac{2kx_{i,j,k} kx_{i+1,j,k}}{kx_{i,j,k} + kx_{i+1,j,k}} \quad (3.35)$$

Bei der Diskretisierung nichtäquidistanter Knoten verkompliziert sich das Verfahren ein wenig. Nimmt man an, δx_{i_e} sei der Abstand des Knotens $P_{i,j,k}$ zu seiner östlichen Grenzfläche und δx_e der Abstand zu seinem östlichen Nachbarknoten $P_{i+1,j,k}$, dann kann mit

$$f_e \equiv \frac{\delta x_{i_e}}{\delta x_e} \quad (3.36)$$

der Grenzflächenwichtungsfaktor definiert werden. Die Gleichungen (3.34) und (3.35) modifizieren sich dann zu

$$Tx_e = kx_e (h_{i,j,k} - bo_{i,j,k})^{f_e} (h_{i+1,j,k} - bo_{i+1,j,k})^{(1-f_e)} \quad (3.37)$$

bzw.

$$kx_e = \frac{kx_{i,j,k} kx_{i+1,j,k}}{(1-f_e) kx_{i,j,k} + f_e kx_{i+1,j,k}} \quad (3.38)$$

oder

$$kx_e = \left(\frac{1-f_e}{kx_{i,j,k}} + \frac{f_e}{kx_{i+1,j,k}} \right)^{-1} \quad (3.39)$$

Die für den Konvektionsterm der Wärmetransportberechnung benötigten lokalen Fließgeschwindigkeiten können auf zweierlei Wegen — entweder für die Knoten oder für die Zellengrenzflächen — über eine explizite finite Differenzen Form berechnet werden, /111/. Die Filtergeschwindigkeit in X -Richtung am Knoten $P_{i,j,k}$ berechnet man wie folgt:

$$(v_x)_{i,j,k} = (k_{xx})_{i,j,k} \frac{h_{i-1,j,k} - h_{i+1,j,k}}{\delta x_w + \delta x_e} \quad (3.40)$$

δx_w und δx_e bedeuten hierbei die Entfernung des Knotens $P_{i,j,k}$ zu seinem westlichen bzw. östlichen Nachbarn. Für die Filtergeschwindigkeit an der östlichen Grenzfläche des selben Knotens schreibt man:

$$(v_x)_{i+1/2,j,k} = (k_{xx})_{i+1/2,j,k} \frac{h_{i,j,k} - h_{i+1,j,k}}{\delta x_e} \quad (3.41)$$

Wie man erkennt, wird bei äquidistanten Knotenabständen in Gl. (3.40) die Filtergeschwindigkeit über den doppelten Knotenabstand integriert, was gegenüber Gl. (3.41), in die nur der einfache Knotenabstand einfließt, eine gewisse Ungenauigkeit in sich birgt. In dem hier beschriebenen Modell werden daher die Konvektionskoeffizienten über Grenzflächenfiltergeschwindigkeiten berechnet.

3.4 Iterative Lösungsverfahren

Bei den iterativen Lösungsverfahren von finiten Differenzen Gleichungen unterscheidet man "Punkt - für - Punkt" und "Linie - für - Linie" Methoden. Das einfachste "Punkt - für - Punkt" - Verfahren ist die Gauss - Seidel - Iteration, bei der für jeden Knoten der Reihe nach die Differenzengleichung gelöst wird. Für die gesuchte Zielgröße ϕ_P hat diese Gleichung die Form:

$$a_P\phi_P = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + a_O\phi_O + a_U\phi_U + b \quad (3.42)$$

oder kürzer:

$$a_P\phi_P = \sum a_{nb}\phi_{nb} + b \quad (3.43)$$

Hierbei stellt nb für alle Nachbarindizes des Knotens P und b für den Quellen-/Senkenterm. Löst man (3.43) nach ϕ_P auf, so erhält man:

$$\phi_P = \frac{\sum a_{nb}\phi_{nb}^* + b}{a_P} \quad (3.44)$$

Bei ϕ_{nb}^* handelt es sich um die Nachbarwerte von ϕ_P , die entweder bereits berechnet wurden oder noch berechnet werden müssen. Insgesamt wird bei diesem Verfahren immer nur ein Set von den unbekanntem Zielgrößen im Arbeitsspeicher des Rechners gehalten. Bei der iterativen Lösung algebraischer Gleichungen oder der numerischen Behandlung von Nichtlinearitäten kann es bisweilen wünschenswert sein, die maximalen Änderungen pro Iterationsschritt zu vergrößern bzw. zu limitieren. Im ersten Fall kann der Rechenaufwand erheblich reduziert werden, da das Verfahren rascher konvergiert, und im zweiten Fall kann man Oszillation unterdrücken und damit ein Kollabieren der Iteration verhindern. Einen solchen Effekt erreicht man durch die Einführung eines Relaxationsfaktors. ϕ_P^0 sei nun die bereits im vorherigen Iterationsschritt berechnete Zielgröße am Punkt P . Die Addition und gleichzeitige Subtraktion von ϕ_P^0 zur rechten Seite von Gleichung (3.44) führt zu

$$\phi_P = \phi_P^0 + \left(\frac{\sum a_{nb}\phi_{nb} + b}{a_P} - \phi_P^0 \right) \quad (3.45)$$

Die maximalen Änderungen pro Iterationsschritt lassen sich durch die Modifikation von Gleichung (3.45) vergrößern oder limitieren.

$$\phi_P = \phi_P^0 + \Omega \left(\frac{\sum a_{nb}\phi_{nb} + b}{a_P} - \phi_P^0 \right) \quad (3.46)$$

oder

$$\frac{a_P}{\Omega}\phi_P = \sum a_{nb}\phi_{nb} + b + (1 - \Omega)\frac{a_P}{\Omega}\phi_P^0 \quad (3.47)$$

Der Relaxationsfaktor Ω kann Werte zwischen 0 und 2 annehmen. Bei $0 < \Omega < 1$ spricht man von Unterrelaxation und analog, bei $1 < \Omega < 2$, von Überrelaxation. In der angelsächsischen Literatur wird die Überrelaxation in Verbindung mit dem Gauss-Seidel Verfahren auch als **S**uccessive **O**ver **R**elaxation (SOR) bezeichnet. Das Gauss-Seidel-Verfahren läßt sich programmtechnisch sehr einfach realisieren, hat jedoch den Nachteil, daß es relativ langsam konvergiert. Ein Grund für die langsame Konvergenz ist, daß der Informationsgehalt der Randbedingungen nur schrittweise in das Innere des Modellareals transferiert wird.

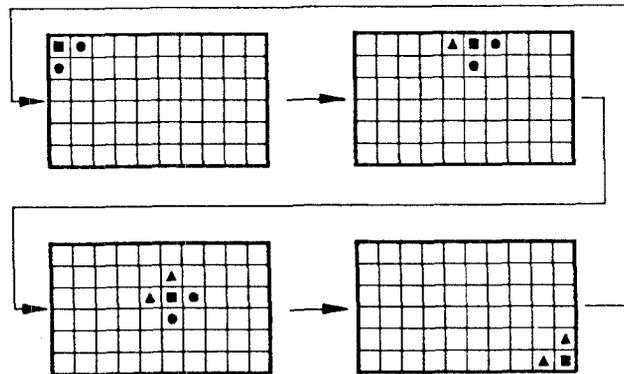


Abb. 9: Flußdiagramm der Gauss-Seidel-Iteration

Das zweidimensionale finite Differenzen Raster in Abb. 9 stellt das unvollständige Flußdiagramm eines Gauss-Seidel-Iterationschrittes dar. Die Quadrate markieren die aktuell zu berechnenden Knoten, die Dreiecke bereits berechnete und die Kreise die aus dem vorherigen Iterationsschritt bekannten oder anfänglich vorbesetzten Knoten. Dadurch, daß die Knoten des linken, oberen Teiles des Rasters in der hier dargestellten Reihenfolge stets zuerst innerhalb eines Iterationsschrittes gelöst werden,

können aus völlig symmetrischen Problemen asymmetrischen Lösungen hervorgehen. Diesen unerwünschten Effekt kann man durch alternierende Iterationsrichtungen unterdrücken. Man beginnt in umgekehrter Richtung die Iteration dort, wo man die vorherige beendet hat.

Der Nachteil des langsamen Transports der Randinformationen in das Innere des Modellareals wird von dem "Linie-für-Linie" Verfahren dadurch umgangen, daß hier alle Knoten einer Reihe in einem Schritt mit ihren Nachbarn ins Bilanzgleichgewicht gebracht werden.

Bei den "Linie-für-Linie" Verfahren unterteilt man in einige Varianten, die sich in ihrer Effizienz nur gering unterscheiden:

1. **Line - Successive - Over - Relaxation (LSOR)**
BJORDAMMAN & COATS(1969)
2. **Sliced - Successive - Over - Relaxation (SSOR)**
PEACEMAN (1977)
3. **Line - Successive - Over - Relaxation with additive Corrections (LSORC)**, PEACEMAN (1977)
4. **Iterative - Alternating - Direction - Implizit (IADI)**
DOUGLAS & RACHFORD (1956)
5. **Strongly - Implizit - Procedure (SIP)**
STONE (1969)

Abb. 10 gibt für ein zweidimensionales Knotenraster das Flußdiagramm des IADI-Verfahrens wieder.

Da in dem hier beschriebenen Modell neben der Gauss-Seidel-Iteration das IADI-Verfahren eingesetzt wurde, sei es an dieser Stelle etwas näher beschrieben. Die Grundidee ist die, tridiagonale Koeffizientenmatrizen zu generieren, die mit Hilfe des äußerst effizienten Thomas-Algorithmus (vergl. Kap. 3.5) direkt gelöst werden können. Ein solches Matrizengleichungssystem hat die Form

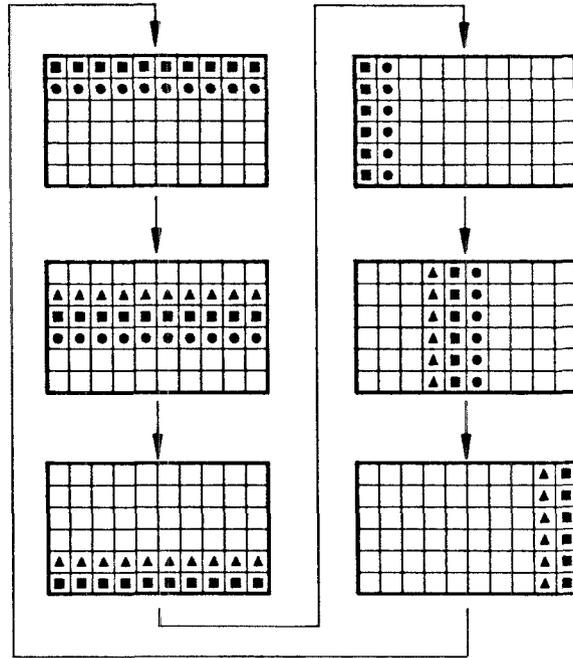


Abb. 10: Flußdiagramm des IADI-Verfahrens

$$\begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{N-1} & b_{N-1} & c_{N-1} & \\ & & & a_N & b_N & \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N-1} \\ d_N \end{bmatrix}.$$

Im dreidimensionalen Fall gliedert sich somit jeder Iterationsschritt in drei Teilschritte, die X -, die Y - und die Z -Schleife. In der X -Schleife werden alle von Westen nach Osten verlaufenden Knotenreihen der Reihe nach berechnet, bei der Y -Schleife alle von Norden nach Süden verlaufenden und bei der Z -Schleife schließlich alle Knotenreihen, die sich von Oben nach Unten erstrecken. Sofern die nichtdiagonalen Komponenten des hydraulischen Leitfähigkeitstensors Null sind, setzt sich die Koeffizientenmatrix der Strömungsberechnung wie folgt zusammen:

$$(a_W)_{i,j,k} = T x_{i-1/2,j,k} \frac{\Delta y_j \Delta z_k}{(\delta x_i)_w} \quad (3.48a)$$

$$(a_E)_{i,j,k} = T x_{i+1/2,j,k} \frac{\Delta y_j \Delta z_k}{(\delta x_i)_e} \quad (3.48b)$$

$$(a_N)_{i,j,k} = T y_{i,j-1/2,k} \frac{\Delta x_i \Delta z_k}{(\delta y_j)_n} \quad (3.48c)$$

$$(a_S)_{i,j,k} = T y_{i,j+1/2,k} \frac{\Delta x_i \Delta z_k}{(\delta y_j)_s} \quad (3.48d)$$

$$(a_o)_{i,j,k} = m_k k z_{i,j,k-1/2} \frac{\Delta x_i \Delta y_j}{(\delta z_k)_o} \quad (3.48e)$$

$$(a_u)_{i,j,k} = m_k k z_{i,j,k+1/2} \frac{\Delta x_i \Delta y_j}{(\delta z_k)_u} \quad (3.48f)$$

für die X -Schleife schreibt man:

$$a_i = -(a_w)_{i,j,k}$$

$$b_i = (a_w)_{i,j,k} + (a_\epsilon)_{i,j,k} + (a_n)_{i,j,k} + (a_s)_{i,j,k} + (a_o)_{i,j,k} + (a_u)_{i,j,k} \\ + \Delta x_i \Delta y_j \frac{S'}{\Delta t}$$

$$c_i = -(a_\epsilon)_{i,j,k}$$

$$d_i = (a_n)_{i,j,k} h_{i,j-1,k} + (a_s)_{i,j,k} h_{i,j+1,k} + (a_o)_{i,j,k} h_{i,j,k-1} + (a_u)_{i,j,k} h_{i,j,k+1} \\ + m_k (Q_w)_{i,j,k} + \Delta x_i \Delta y_j \frac{S'}{\Delta t} h_{i,j,k}^0$$

$$u_i = h_{i,j,k}$$

analog die Y -Schleife:

$$a_j = -(a_n)_{i,j,k}$$

$$b_j = (a_w)_{i,j,k} + (a_\epsilon)_{i,j,k} + (a_n)_{i,j,k} + (a_s)_{i,j,k} + (a_o)_{i,j,k} + (a_u)_{i,j,k} \\ + \Delta x_i \Delta y_j \frac{S'}{\Delta t}$$

$$c_j = -(a_s)_{i,j,k}$$

$$d_j = (a_w)_{i,j,k} h_{i-1,j,k} + (a_\epsilon)_{i,j,k} h_{i+1,j,k} + (a_o)_{i,j,k} h_{i,j,k-1} + (a_u)_{i,j,k} h_{i,j,k+1} \\ + m_k (Q_w)_{i,j,k} + \Delta x_i \Delta y_j \frac{S'}{\Delta t} h_{i,j,k}^0$$

$$u_j = h_{i,j,k}$$

und schließlich die Z -Schleife:

$$a_k = -(a_o)_{i,j,k}$$

$$b_k = (a_w)_{i,j,k} + (a_\epsilon)_{i,j,k} + (a_n)_{i,j,k} + (a_s)_{i,j,k} + (a_o)_{i,j,k} + (a_u)_{i,j,k} \\ + \Delta x_i \Delta y_j \frac{S'}{\Delta t}$$

$$c_k = -(a_u)_{i,j,k}$$

$$d_k = (a_w)_{i,j,k} h_{i-1,j,k} + (a_\epsilon)_{i,j,k} h_{i+1,j,k} + (a_n)_{i,j,k} h_{i,j-1,k} + (a_s)_{i,j,k} h_{i,j+1,k} \\ + m_k (Q_w)_{i,j,k} + \Delta x_i \Delta y_j \frac{S'}{\Delta t} h_{i,j,k}^0$$

$$u_k = h_{i,j,k}$$

Nimmt man an, daß die nichtdiagonalen Komponenten des Wärmeleitfähigkeitstensors ebenfalls Null sind, setzt sich die Koeffizientenmatrix der diffundiv - konvektiven Wärmetransportberechnung in analoger Weise zusammen.

$$(a_\epsilon)_{i,j,k} = (D_\epsilon)_{i,j,k} A(|(Pe_\epsilon)_{i,j,k}|) + [(F_\epsilon)_{i,j,k}] \quad (3.49a)$$

$$(a_w)_{i,j,k} = (D_w)_{i,j,k} A(|(Pe_w)_{i,j,k}|) + [(F_w)_{i,j,k}] \quad (3.49b)$$

$$(a_n)_{i,j,k} = (D_n)_{i,j,k} A(|(Pe_n)_{i,j,k}|) + [(F_n)_{i,j,k}] \quad (3.49c)$$

$$(a_s)_{i,j,k} = (D_s)_{i,j,k} A(|(Pe_s)_{i,j,k}|) + [(F_s)_{i,j,k}] \quad (3.49d)$$

$$(a_o)_{i,j,k} = (D_o)_{i,j,k} A(|(Pe_o)_{i,j,k}|) + [(F_o)_{i,j,k}] \quad (3.49e)$$

$$(a_u)_{i,j,k} = (D_u)_{i,j,k} A(|(Pe_u)_{i,j,k}|) + [(F_u)_{i,j,k}] \quad (3.49f)$$

Die Diffusions- und Konvektionskoeffizienten — D_{nb} bzw. F_{nb} — sowie die lokalen Peclet-Zahlen Pe_{nb} werden hierbei nach den Gleichungen (3.26a-f) bzw. (3.27a-c) berechnet. Die Schleifengleichungen lauten somit:

X -Schleife:

$$a_i = -(a_w)_{i,j,k}$$

$$b_i = (a_w)_{i,j,k} + (a_\epsilon)_{i,j,k} + (a_n)_{i,j,k} + (a_s)_{i,j,k} + (a_o)_{i,j,k} + (a_u)_{i,j,k}$$

$$+\Delta x_i \Delta y_j \Delta z_k \frac{(\rho_{eff})_{i,j,k}}{\Delta t}$$

$$c_i = -(a_\epsilon)_{i,j,k}$$

$$d_i = (a_n)_{i,j,k} T_{i,j-1,k} + (a_s)_{i,j,k} T_{i,j+1,k} + (a_o)_{i,j,k} T_{i,j,k-1} + (a_u)_{i,j,k} T_{i,j,k+1}$$

$$+\frac{(Q_\epsilon)_{i,j,k}}{(C_{eff})_{i,j,k}} + \Delta x_i \Delta y_j \Delta z_k \frac{(\rho_{eff})_{i,j,k}}{\Delta t} T_{i,j,k}^0$$

$$u_i = T_{i,j,k}$$

Y-Schleife:

$$a_j = -(a_n)_{i,j,k}$$

$$b_j = (a_w)_{i,j,k} + (a_\epsilon)_{i,j,k} + (a_n)_{i,j,k} + (a_s)_{i,j,k} + (a_o)_{i,j,k} + (a_u)_{i,j,k}$$

$$+\Delta x_i \Delta y_j \Delta z_k \frac{(\rho_{eff})_{i,j,k}}{\Delta t}$$

$$c_j = -(a_s)_{i,j,k}$$

$$d_j = (a_w)_{i,j,k} T_{i-1,j,k} + (a_\epsilon)_{i,j,k} T_{i+1,j,k} + (a_o)_{i,j,k} T_{i,j,k-1} + (a_u)_{i,j,k} T_{i,j,k+1}$$

$$+\frac{(Q_\epsilon)_{i,j,k}}{(C_{eff})_{i,j,k}} + \Delta x_i \Delta y_j \Delta z_k \frac{(\rho_{eff})_{i,j,k}}{\Delta t} T_{i,j,k}^0$$

$$u_j = T_{i,j,k}$$

Z-Schleife:

$$a_k = -(a_o)_{i,j,k}$$

$$b_k = (a_w)_{i,j,k} + (a_\epsilon)_{i,j,k} + (a_n)_{i,j,k} + (a_s)_{i,j,k} + (a_o)_{i,j,k} + (a_u)_{i,j,k}$$

$$+\Delta x_i \Delta y_j \Delta z_k \frac{(\rho_{eff})_{i,j,k}}{\Delta t}$$

$$c_k = -(a_u)_{i,j,k}$$

$$d_k = (a_w)_{i,j,k} T_{i-1,j,k} + (a_\epsilon)_{i,j,k} T_{i+1,j,k} + (a_n)_{i,j,k} T_{i,j-1,k} + (a_s)_{i,j,k} T_{i,j+1,k}$$

$$+\frac{(Q_\epsilon)_{i,j,k}}{(C_{eff})_{i,j,k}} + \Delta x_i \Delta y_j \Delta z_k \frac{(\rho_{eff})_{i,j,k}}{\Delta t} T_{i,j,k}^0$$

$$u_k = T_{i,j,k}$$

wobei:

$$(C_{eff})_{i,j,k} = (\Phi_g)_{i,j,k} \Psi_k C_w + \left(1 - (\Phi_g)_{i,j,k}\right) (C_g)_{i,j,k} \quad (3.50)$$

bzw. beim Gefrieren von Wasser:

$$\begin{aligned} (C_{eff})_{i,j,k} &= (\Phi_g)_{i,j,k} \Psi_k \left((X_\epsilon)_{i,j,k} C_\epsilon + (X_w)_{i,j,k} C_w \right) \\ &\quad + \left(1 - (\Phi_g)_{i,j,k}\right) (C_g)_{i,j,k} \end{aligned} \quad (3.51)$$

und

$$(\rho_{eff})_{i,j,k} = (\Phi_g)_{i,j,k} \Psi_k \rho_w + \left(1 - (\Phi_g)_{i,j,k}\right) (\rho_g)_{i,j,k} \quad (3.52)$$

bzw. beim Gefrieren von Wasser:

$$\begin{aligned} (\rho_{eff})_{i,j,k} &= (\Phi_g)_{i,j,k} \Psi_k \left((X_\epsilon)_{i,j,k} \rho_\epsilon + (X_w)_{i,j,k} \rho_w \right) \\ &\quad + \left(1 - (\Phi_g)_{i,j,k}\right) (\rho_g)_{i,j,k} \end{aligned} \quad (3.53)$$

Analog zur Gauss-Seidel-Iteration kann das IADI-Verfahren durch die Einführung eines Relaxationsfaktors zum LSOR-Verfahren modifiziert werden. Nachdem durch Anwendung des Thomas-Algorithmus das Feld der gesuchten Zielgrößen u_i berechnet wurde, kann man durch Wiedereinsetzen in das Ausgangsfeld z.B. für die X -Schleife der Temperaturberechnung schreiben:

$$T_{i,j,k} = T_{i,j,k}^0 + \Omega (u_i - T_{i,j,k}^0) \quad (3.54)$$

bzw. für die Strömungsberechnung:

$$h_{i,j,k} = h_{i,j,k}^0 + \Omega (u_i - h_{i,j,k}^0) \quad (3.55)$$

T^0 und h^0 bedeuten hierbei die Temperaturen bzw. Piezometerhöhen des vorausgegangenen Iterationsschrittes. Bei linearen Problemen bewirkt ein Relaxationsfaktor von $1 < \Omega < 2$ eine raschere Konvergenz, während sich bei nichtlinearen Problemen mit $0 < \Omega < 1$ Oszillationen unterdrücken lassen.

3.5 Der Thomas Algorithmus

Viele effiziente, numerische Verfahren generieren Gleichungssysteme mit tridiagonalen Matrizen, /172/. Ein typisches Beispiel für eine solche Matrixgleichung hat die Form:

$$[A]\{u\} = \{d\} \quad (3.56)$$

wobei:

$$A = \begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & a_{N-1} & b_{N-1} & c_{N-1} \\ & & & & a_N & b_N \end{bmatrix},$$

$$\{u\} = \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{Bmatrix}, \quad \{d\} = \begin{Bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N-1} \\ d_N \end{Bmatrix},$$

oder ausgeschrieben:

$$\begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & a_{N-1} & b_{N-1} & c_{N-1} \\ & & & & a_N & b_N \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{Bmatrix} = \begin{Bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N-1} \\ d_N \end{Bmatrix}$$

Die Matrix $[A]$ enthält die Koeffizienten des linearen Gleichungssystems, der Vektor $\{u\}$ die gesuchten Zielgrößen, (Piezometerhöhen, Temperaturen, Konzentrationen etc.) und der Vektor $\{d\}$ die bekannten Eingangsparameter. Der Thomas Algorithmus leitet sich aus der Zerlegung der Koeffizientenmatrix in eine obere und eine untere Dreiecksmatrix ab, sodaß

$$[A] = [L][U] \quad (3.57)$$

wobei $[L]$ und $[U]$ wie folgt definiert werden:

$$[L] = \begin{bmatrix} \beta_1 & & & & & \\ a_2 & \beta_2 & & & & \\ & \ddots & \ddots & & & \\ & & & a_{N-1} & \beta_{N-1} & \\ & & & & a_N & \beta_N \end{bmatrix}$$

und

$$[U] = \begin{bmatrix} 1 & \gamma_1 & & & & \\ & 1 & \gamma_2 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & \gamma_{N-1} \\ & & & & & 1 \end{bmatrix},$$

Durch Multiplikation der unteren Matrix mit der oberen und durch Gleichsetzen der Elemente der Produktmatrix mit der Koeffizientenmatrix erhält man folgende Beziehungen:

$$\begin{aligned} \beta_1 &= b_1 \\ \gamma_i &= \frac{c_i}{\beta_i} \quad i = 1, 2, \dots, N-1 \\ \beta_i &= b_i - a_i \gamma_{i-1} \quad i = 2, 3, \dots, N \end{aligned}$$

Gleichung (3.56) kann man somit auch schreiben:

$$[L][U]\{u\} = \{d\}$$

oder

$$[L]\{s\} = \{d\}$$

wobei $\{s\}$ einen vorübergehend unbekanntem Vektor darstellt, der wie folgt definiert ist:

$$\{s\} = [U]\{u\}$$

Die Elemente von $\{s\}$ können durch Auflösen der Gleichung $[L]\{s\} = \{d\}$ bestimmt werden. Hierbei ergeben sich folgende Beziehungen:

$$\begin{aligned} s_1 &= \frac{d_1}{\beta_1} \\ s_i &= \frac{(d_i - a_i s_{i-1})}{\beta_i}, \quad i = 2, 3, \dots, N \end{aligned}$$

Mit dem nun bekannten Vektor $\{s\}$ wird durch rückwärtiges Einsetzen der gesuchte Vektor $\{u\}$ ermittelt. Dies geschieht durch folgende Beziehungen:

$$\begin{aligned} u_n &= s_N \\ u_i &= s_i - \gamma_i u_{i+1}, \quad i = N-1, N-2, \dots, 1 \end{aligned}$$

Noch einmal zusammengefaßt gliedert sich das Berechnungsverfahren in zwei Schritte: zunächst werden die Felder β_i und s_i berechnet, die darauf - im zweiten Schritt - durch rückwärtiges Einsetzen die unbekannt Zielgrößen liefern. Der Thomas Algorithmus stellt für den eindimensionalen Fall einen direkten Gleichungslöser dar, er kann aber auch sehr effizient für mehrdimensionale Anwendungen, z.B. in Verbindung mit der iterativen ADI - Methode von /57/ eingesetzt werden.

3.6 Behandlung der Randbedingungen

Bei der numerischen Behandlung von partiellen Differentialgleichungen unterscheidet man drei Arten von Randbedingungen:

- Randbedingungen 1. Art - z.B. bekannte Piezometerhöhen, Temperaturen etc.
- Randbedingungen 2. Art - gegebener Randzufluß - z.B. basaler Wärmefluß, bekannter Zu- und Abstrom von Grundwasser über die Modellränder
- Randbedingungen 3. Art, die eine Kombination aus 1. und 2. darstellen, d.h. der Randzufluß hängt von der gesuchten Zielgröße ab

Diese Randbedingungen können in finiten Differenzen Modellen auf sehr vielfältige Weise gelöst werden. Ein häufig eingesetztes Verfahren besteht aus der Generierung virtueller Knotenreihen an den Rändern des Modellareals, kurz *Superposition* genannt. Undurchlässige Ränder lassen sich hierbei durch die Spiegelung der vorletzten Knotenreihe über den Rand auf die virtuelle Knotenreihe realisieren. Randbedingungen zweiter Art werden durch eine 180° - Rotation der vorletzten Knotenreihe um den Modellrand gewährleistet.

Die Superposition virtueller Knotenreihen hat den Vorteil, daß man z.B. für die Simulation eines völlig symmetrischen Absenkungstrichters eines Brunnens bei kartesischer Diskretisierung nur einen Quadranten zu berechnen braucht, (vergl. Kap 5.4). Speziell im dreidimensionalen Fall benötigt man dafür jedoch eine große Anzahl zusätzlicher Knoten, was auf Computern mit eingeschränktem Kernspeicher zu untragbaren Beschränkungen bezüglich der Diskretisierung führen kann. In dem hier vorgestellten Modell werden die Randbedingungen daher wie folgt behandelt. Alle Modellränder werden als wasser- und energieundurchlässig betrachtet. Randbedingungen 1. Art werden in der GAUSS-SEIDEL-Iteration als gelöste Knotengleichung betrachtet, beim IADI-Verfahren wird den Zellen der betreffenden Knoten eine quasiunendliche Speicherkapazität¹ zugeordnet. Der Vollständigkeit halber sei an dieser Stelle erwähnt, daß man zur Lösung der in Kap. 3.3 beschriebenen Gleichungssysteme wenigstens einen Knoten, für den die Randbedingung der 1. Art erfüllt ist, benötigt. Bekannte Randzufflüsse, also Randbedingungen 2. Art, werden in beiden Iterationsverfahren dem Quellen/Senkentem des betreffenden Knotens beaufschlagt. Randbedingungen der 3. Kategorie können mit der hier beschriebenen Programmversion noch nicht behandelt werden, sie lassen sich jedoch relativ einfach in das Modell einbauen.

¹ bei der Strömungsberechnung wird diesen Knoten ein Speicherkoeffizient von $S > 10^{20}$ zugeordnet, bei der Wärmetransportberechnung werden ρ und C auf $> 10^{20}$ gesetzt

4 Programmentwicklung

Die numerische Behandlung der den Prozess der Strömung und des Wärmetransportes beschreibenden partiellen Differentialgleichungen kann je nach Anzahl der Knoten und Zeitschritte in eine außerordentlich rechenintensiven Prozedur ausarten. Man ist daher bei der Modellentwicklung auf eine Programmiersprache angewiesen, die einerseits weite Verbreitung hat, d.h. auch auf Workstations und Großrechnern einsetzbar ist, andererseits aber auch über ein reichhaltiges Angebot vordefinierter Funktionen verfügt. Daher fiel die Wahl auf die Programmiersprache FORTRAN77. Da kein Modell so komplex ist, daß es nicht doch noch um den einen oder anderen Aspekt erweitert werden könnte, bot sich ein modulares Konzept an. So besteht TRADIKON-3D aus einem Hauptprogramm, das lediglich die Aufgabe hat, die Aufrufe der 62 Unterprogramme zu koordinieren und für eine problemabhängige Programmsteuerung zu sorgen. Eine Reihe von Eingabeoptionen erlaubt es, die Problemlösung, sofern physikalisch statthaft, zu vereinfachen und somit die Rechenzeit zu verkürzen.

Reine Konduktionsprobleme und Diffusions-Konvektionsprobleme können z.B. getrennt voneinander behandelt werden. Bei der Berechnung des Wärmetransportes in Gesteinen mit stagnierenden Grundwasserständen kann z.B. die Lösung des Konvektionstermes entfallen, was die Rechenzeit spürbar verkürzt. Ferner läßt sich die Strömungsberechnung vollständig abkoppeln, etwa für den Fall, daß man eine ausreichend genaue Kenntnis von den Fließverhältnissen hat. Über die Auswirkungen des Wärmetransports auf das Strömungsfeld können dann naturgemäß keinerlei Aussagen mehr getroffen werden. Schließlich kann das Programm als eigenständiges — zugegebenermaßen einfaches — Grundwassermodell eingesetzt werden.

Die wichtigsten physikalischen Vereinfachungen des Modells seien an dieser Stelle noch einmal kurz zusammengefaßt:

- das Grundwasser strömt laminar, sodaß die Gültigkeit des Darcy - Gesetzes gewährleistet ist
- die Gesteinsmatrix befindet sich im Temperaturgleichgewicht mit dem strömenden Fluid, /131/
- Wärmetransport durch Strahlung wird vernachlässigt

- das strömende Grundwasser besitzt eine konstante Dichte und ist inkompressibel
- der konvektive Teil des Wärmetransports geht frei von thermischer Dispersion vonstatten, /41/
- Feuchtigkeitsmigration in der ungesättigten Bodenzone in Richtung einer Eisfront oder — bei höheren Temperaturen — entlang des thermischen Gradienten wird aufgeklammert
- die Richtungen der hydraulischen und thermischen Anisotropie verlaufen parallel zu den Achsen des kartesischen Koordinatensystems
- die Wärmekapazität von Wasser und Gestein ist konstant

Das Programm ist so konzipiert, daß folgende physikalische Eingangsparameter für *jeden* Knoten variabel eingegeben werden können:

- Temperatur
- Zufuhr bzw. Entzug von Wärme
- isotrope Wärmeleitfähigkeit des Gesteins
- Wärmekapazität des Gesteins
- Dichte des Gesteins
- Fließgeschwindigkeit in X, Y und Z -Richtung
- Piezometerhöhe
- Zugabe bzw. Entnahme von Wasser
- hydraulische Leitfähigkeit in X, Y und Z -Richtung
- durchflußwirksame oder drainierbare Porosität
- Gesamtporosität
- Speicherkoeffizient

Die Anisotropie der Wärmeleitfähigkeit des Gesteins kann nur über Anisotropiefaktoren, die dann jedoch für das gesamte Modellareal Gültigkeit besitzen müssen, berücksichtigt werden.

$$K_{xx} = aK_{isotrop} \quad (4.00a)$$

$$K_{yy} = bK_{isotrop} \quad (4.00b)$$

$$K_{zz} = cK_{isotrop} \quad (4.00c)$$

wobei a, b und c die Anisotropiefaktoren darstellen. Folgende Eingangsparameter werden Schicht für Schicht erfragt:

- Wassersättigungsgrad
- Schmelztemperatur von Wasser
- Gefrierkurve des Bodens

Zeitlich variable Eingangsparameter, wie z.B. schwankende Förderraten eines Brunnens oder vorgegebene Soletemperaturen im Wärmetauscherkreislauf, können über eine zusätzliche Eingabedatei in die laufende Simulation mit einbezogen werden. Ferner verfügt TRADIKON-3D über eine "Warmstart" - Routine, die es gestattet, mit den Ergebnissen einer vorherigen Simulation einen weiteren Simulationslauf zu starten. Diese Option kann z.B. für die Berechnung der Anfangsbedingungen einer instationären Simulation eingesetzt werden. Da bei aktivierter "Warmstart" - Routine parallel zu jeder Ergebnisausgabe das aktuelle Temperatur- bzw. Strömungsfeld auf eine Datei geschrieben wird, kann man auch nach Systemabstürzen die Simulation fortsetzen.

Eine umfangreiche Erläuterung der wichtigsten, im Programm verwandten Variablen sowie eine Beschreibung der Dateneingabe wird im Anhang 2 vorgestellt, das vollständige Programmlisting befindet sich im Anhang 3.

5 Validierung des Modells

5.1 Konduktiver Wärmetransport

Die einfachste Form der Validierung stellt die Berechnung des eindimensionalen, stationären Wärmeflusses dar, /167/. Eine gedachte Hochofenwand verfüge über folgenden Aufbau:

Schamotte	$dx_1 = 0.30m;$	$K_1 = 3.2 W/mK$
Isolation	$dx_2 = 0.18m;$	$K_2 = 0.1 W/mK$
Ziegel	$dx_3 = 0.27m;$	$K_3 = 0.9 W/mK$

Die Temperatur T_1 an der Oberfläche der Innenwand eines Hochofens betrage konstant $950\text{ }^\circ\text{C}$, die der Außenwand konstant $50\text{ }^\circ\text{C}$ (T_4). Gesucht sind die Temperaturen T_2 und T_3 an der Grenze Schamotte/Isolation bzw. Isolation/Ziegel unter der Voraussetzung, daß sich ein stationärer Wärmefluß eingestellt hat.

Analytische Lösung:

Der Wärmefluß durch die Hochofenwand des oben beschriebenen Aufbaus beträgt nach

$$J = \frac{A(T_1 - T_4)}{dx_1/K_1 + dx_2/K_2 + dx_3/K_3} = 410.26 \text{ Watt}$$

Für T_2 gilt:

$$T_2 = T_1 - \frac{Jdx_1}{K_1A} = 911.54\text{ }^\circ\text{C}$$

Für T_3 gilt:

$$T_3 = T_2 - \frac{Jdx_2}{K_2A} = 173.08\text{ }^\circ\text{C}$$

Finite Differenzen Simulation:

Bei der Berechnung des stationären, rein konduktiven Wärmeflusses wird in Analogie zum Speicherkoeffizienten bei der Strömungsberechnung die Dichte und Wärmekapazität des Transportmediums auf Null gesetzt.

Ferner entfällt die Lösung des Konvektionsterms. Die TRADIKON - Simulation ergab mit dem Gauss-Seidel-Verfahren nach 90 Iterationen (SOR, $\Omega = 1.88$) bei einem Iterationskriterium von $10^{-2} \text{ }^\circ\text{C}$ Temperaturen von $T_2 = 911.34^\circ\text{C}$ und $T_3 = 173.26^\circ\text{C}$. Die geringen Abweichungen von der analytischen Lösung müssen auf die relativ grobe Diskretisierung (11 Knoten) und Rundungsfehler zurückgeführt werden.

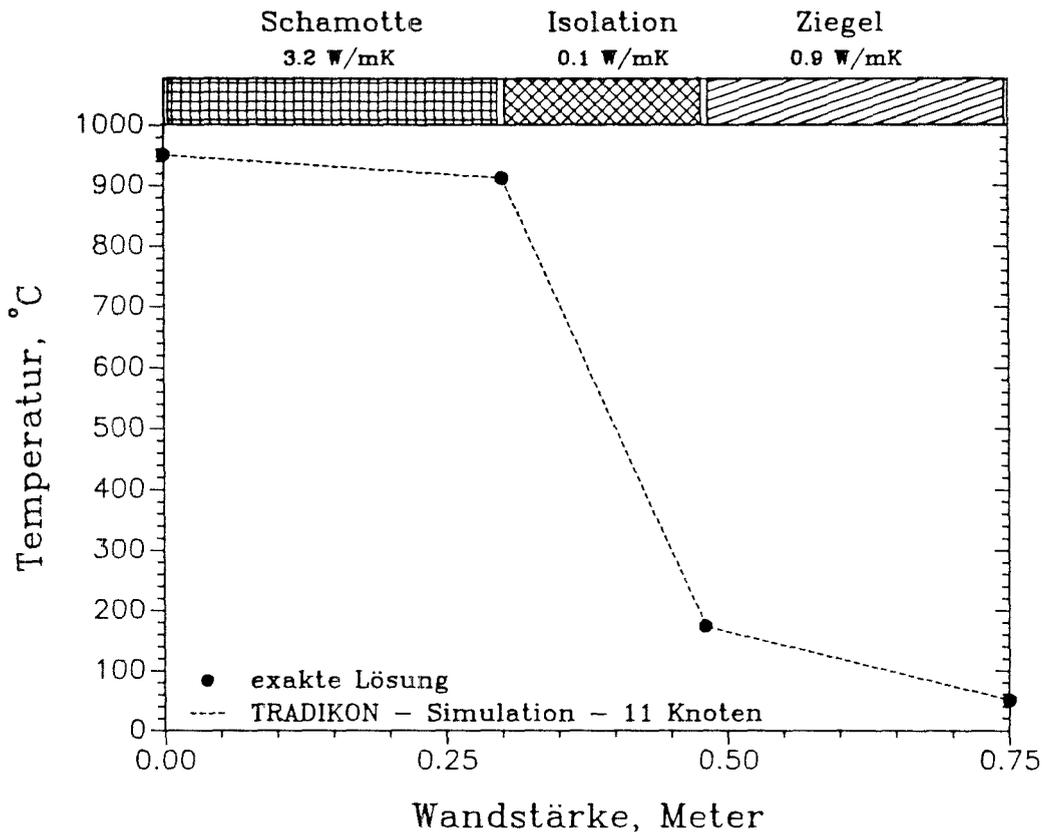


Abb. 11: Stationärer Wärmefluß durch eine Hochofenwand

5.2 Instationärer, eindimensionaler Wärmetransport

Der instationäre, eindimensionale Wärmetransport durch ein isotropes Medium mit dem Temperaturleitwert α_T läßt sich mit der Differentialgleichung:

$$\frac{\delta T}{\delta t} = \alpha_T \frac{\delta^2 T}{\delta x^2} \quad (5.00)$$

berechnen. Handelt es sich um ein homogen temperiertes Medium, das man schlagartig an dem einen Ende abkühlt und dann auf einem konstanten

Temperaturniveau hält, so läßt sich mit Hilfe der komplementären Fehlerfunktion $erfc(x)$ eine sehr gute Näherung erzielen. Das im folgenden vorgestellte Rechenbeispiel wurde /8/ entnommen. T_1 sei die Ausgangstemperatur und T_0 die plötzlich an dem einen Ende induzierte Temperatur. Die Randbedingungen sollen somit lauten:

$$t = 0; \quad x \geq 0; \quad T = T_1$$

$$t > 0; \quad x = 0; \quad T = T_0$$

Das eindimensionale Temperaturfeld unter den gegebenen Bedingungen läßt sich nach

$$\frac{(T - T_0)}{(T_0 - T_1)} = erfc\left(\frac{x}{2\sqrt{\alpha_T t}}\right) \quad (5.01)$$

wobei:

$$erfc(x) = \left(\frac{2}{\sqrt{\pi}}\right) \int_x^\infty e^{-t^2} dt \quad (5.02)$$

mit:

$$erfc(x) = 1 - erf(x) \quad (5.03)$$

berechnen. Die Fehlerfunktion kann durch ein numerisches Näherungsverfahren abgeschätzt werden, /1/

$$erf(x) = 1 - \frac{1}{(1 + a_1 x + a_2 x^2 + \dots + a_6 x^6)^{16}} + E(x) \quad (5.04)$$

mit:

$$\begin{array}{ll} a_1 = 0.0705230784 & a_4 = 0.0001520143 \\ a_2 = 0.0422820123 & a_5 = 0.0002765672 \\ a_3 = 0.0092705272 & a_6 = 0.0000430638 \end{array}$$

$$E(x) < 3 \cdot 10^{-7}$$

Ein Gesteinskörper verfüge über eine Dichte $\rho_g = 2600 \text{ kg/m}^3$, eine spezifische Wärmekapazität $c_g = 850 \text{ J/kgK}$ und eine Wärmeleitfähigkeit $K_{eff} = 3.0 \text{ W/mK}$. Daraus ergibt sich nach Gl. (2.10) ein Temperaturleitwert von $\alpha_T = 0.78 \text{ m}^2/\text{s}$. Gesucht werden die Temperaturprofile nach 0.25, 0.5, 1, 2, 4 und 8 Tagen bis in eine Tiefe von 1.8 m, die sich ausbilden,

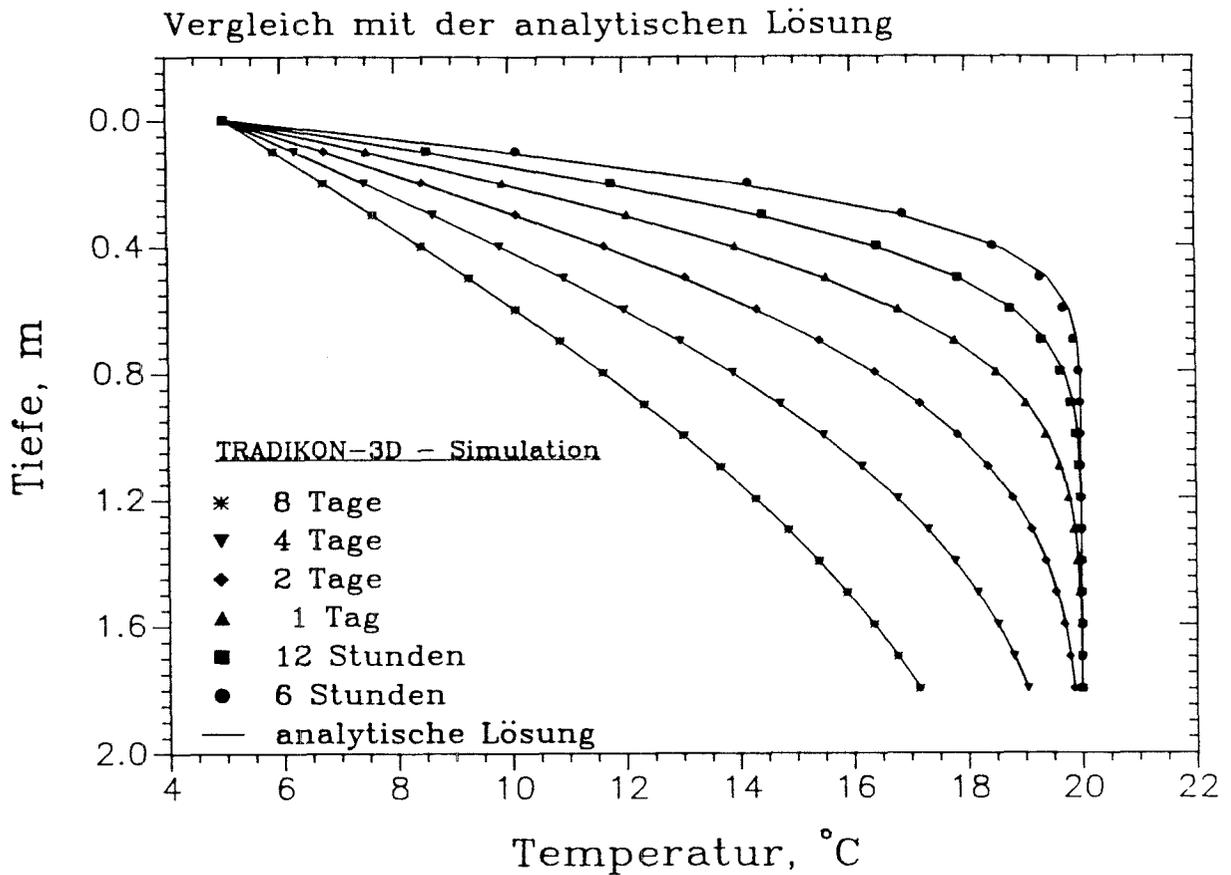


Abb. 12: Konduktiver Wärmetransport in dem beschriebenen Gesteinskörper. Die durchgezogenen Linien markieren die oben beschriebene analytische Lösung, die Symbole das Simulationsergebnis mit TRADIKON-3D.

wenn man den ursprünglich $20\text{ }^{\circ}\text{C}$ warmen Körper schlagartig an dem einen Ende auf $5\text{ }^{\circ}\text{C}$ abkühlt.

Die Ergebnisse des analytischen Ansatzes decken sich nahezu exakt mit denen der numerischen Simulation. Die genauen Zahlenwerte beider Verfahren können den Tabellen im Anhang 6 entnommen werden. Die analytische Lösung läßt sich mit dem in Anhang 5 gelisteten BASIC - Programm berechnen, die TRADIKON-3D Simulation wurde mit dem Datensatz TRADERF in Anhang 4 durchgeführt.

5.3 Eindimensionales Konvektionsproblem

Die numerische Behandlung des Konvektionstermes konnte durch folgendes Beispiel sehr einfach auf ihre Genauigkeit überprüft werden.

Nimmt man von zwei benachbarten Zellen an, daß sie ausschließlich über ihre gemeinsame Grenzfläche Kontakt mit der jeweils anderen Zelle haben und der Wärmefluß durch sie hindurch ausschließlich auf konvektivem Wege erfolgt, das strömende Medium also keine Wärmeleitfähigkeit besitzt, so läßt sich der instationäre Wärmetransport nach /24/ durch die Beziehung (5.05) berechnen.

$$T_2 = (T_I - T_1)e^{\frac{qt}{\rho v}} + T_1 \quad (5.05)$$

Hierbei bedeutet T_I die Ausgangstemperatur von Zelle 2. Sind die beiden Zellen 1 und 2 mit 100 bzw. 200 °C temperiert und erfolgt von Zelle 1 nach Zelle 2 ein konstanter Massenfluß von 10^{-3} kg/s , so ergibt sich der in Abb. 13 dargestellte Temperaturverlauf.

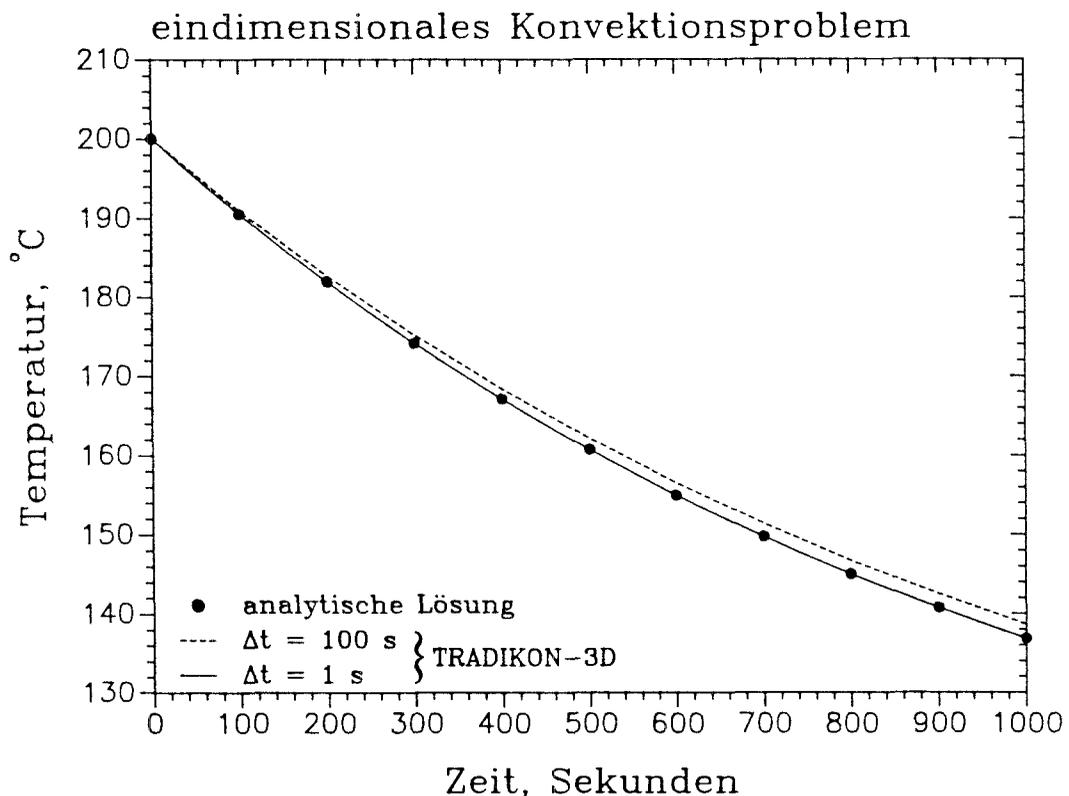


Abb. 13: Vergleich der analytischen mit der numerischen Lösung des Konvektionsproblem

Mit zunehmender Zeitschrittlänge (> 10 Sekunden) kommt es zu einer deutlichen Abweichung von der analytischen Lösung. Bei einer Zeitschritt-

länge von einer Sekunde lagen die Temperaturänderungen pro Zeitschritt bei etwa 0.1 °C und der maximale Berechnungsfehler bei 0.002 °C

5.4 Brunnenabsenkung (Theis - Problem)

Die instationäre Absenkung einer Piezometerhöhe durch den Betrieb eines Brunnens in einem unendlich begrenzten, gespannten Aquifer mit ursprünglich horizontaler Erstreckung der piezometrischen Oberfläche läßt sich nach THEIS(1935) durch die folgende Beziehung ermitteln:

$$h_o - h = \frac{Q}{4\pi T} W(u) \quad (5.06)$$

wobei:

$$W(u) = \int_u^\infty \frac{e^{-u}}{u} du \quad (5.07)$$

und

$$u = \frac{r^2 S}{4Tt} \quad (5.08)$$

$W(u)$ wird hierbei als *Brunnenfunktion* (well function) bezeichnet und ist in den einschlägigen Standardwerken der Hydrogeologie, z. B. /131/, in Form von Tabellenwerten erhältlich. Rechnerisch ermitteln läßt sich diese Funktion durch eine polynomische Approximation, wie sie z.B. /92/ vorschlug:

Für $0 \leq u \leq 1$ gilt:

$$W(u) = -\ln u + C_0 + C_1 u + C_2 u^2 + C_3 u^3 + C_4 u^4 + C_5 u^5 \quad (5.09)$$

Für $0 \leq u < \infty$ gilt:

$$W(u) = \frac{1}{ue^u} \frac{C_6 + C_7 u + u^2}{C_8 + C_9 u + u^2} \quad (5.10)$$

mit:

$C_0 = -0.57721566$	$C_5 = 0.00107857$
$C_1 = 0.99999193$	$C_6 = 0.250621$
$C_2 = -0.24991055$	$C_7 = 2.334733$
$C_3 = 0.05519968$	$C_8 = 1.681534$
$C_4 = -0.00976004$	$C_9 = 3.330657$

WANG & ANDERSON(1982) zogen folgendes Testbeispiel für den Vergleich der numerischen Lösung mit einer analytischen heran: Ein randlich unbegrenzter, gespannter Aquifer mit einer Transmissivität von $T = 300 \text{ m}^2/d$ ($3.47 \cdot 10^{-3} \text{ m}^2/s$) und einem Speicherkoeffizienten von $S' = 0.002$ soll über einen Zeitraum von knapp 2 Wochen (13.12 Tage) mit einer Förderrate von $Q = 2000 \text{ m}^3/d$ ($0.0232 \text{ m}^3/s$) bepumpt werden. Gesucht wird die Zeit-Absenkungskurve eines 100 Meter vom Brunnen entfernten Pegels. Die analytische Lösung des Problems wurde mit dem im Anhang 5 abgedruckten BASIC - Programm gemäß dem oben beschriebenen Verfahren nach /92/ berechnet und den numerischen Lösungen von TRADIKON-3D bzw. WANG & ANDERSON vergleichend gegenübergestellt.

Bei der TRADIKON-Simulation (vergl. Datensatz aus Anhang 4) wurde der Aquifer durch ein äquidistantes Raster von $41 * 41$ Zellen mit einer Kantenlänge von 100 Meter diskretisiert und in dessen Zentrum die Brunnenzelle plaziert (Index 21,21). Als Lösungsschema bot sich ein vollimpliziter Ansatz und ein LSOR - Verfahren mit einem Relaxationsfaktor von 1.2 an. WANG & ANDERSON setzten ein CRANK-NICOLSON Schema in Verbindung mit dem ebenfalls iterativen GAUSS-SEIDEL-Verfahren ein. Bei beiden numerischen Ansätzen betrug die Startzeitschrittlänge 864 Sekunden (0.01 Tage), die nach jedem Zeitschritt um den Faktor 1.5 verlängert wurde.

Wie man den Tabellenwerten entnehmen kann, stimmen die Ergebnisse der numerischen Simulation recht gut mit denen des analytischen Ansatzes überein. Lediglich gegen Ende der Simulation kommt es zu einer größeren Abweichung, die sich jedoch damit erklären läßt, daß der sich ausbildende Absenkungstrichter nach etwa 10 Tagen den Modellrand überschreitet und somit die Annahme eines unendlich ausgedehnten Aquifers modelltechnisch nicht mehr erfüllt ist.

5.5 Testsimulation eines Stufenpumpversuches

Für Problemstellungen mit zeitlich variablen Randbedingungen lassen sich in der Regel keine oder nur sehr schwer analytische Lösungen finden. Daher sollte eine Vergleichssimulation mit dem Aquifersimulationsmodell ASM von KINZELBACH & RAUSCH (1988) dazu dienen, den Strömungsteil von TRADIKON-3D zu testen. ASM ist ein zweidimensionales Strömungsmodell mit variabler Diskretisierungsmöglichkeit sowohl in Raum und

Tab. 2: Absenkung ($h_0 - h$) in Meter für $r = 100m$

Zeit [d]	Theis [m]	TRADIKON [m]	WANG [m]
0.010	0.042	0.064	0.05
0.025	0.211	0.195	-.--
0.047	0.420	0.380	0.41
0.081	0.638	0.595	-.--
0.132	0.856	0.821	0.88
0.208	1.074	1.048	-.--
0.322	1.291	1.273	-.--
0.493	1.508	1.494	1.55
0.749	1.724	1.713	-.--
1.133	1.940	1.930	1.98
1.710	2.156	2.144	-.--
2.575	2.371	2.356	-.--
3.872	2.586	2.566	-.--
5.819	2.802	2.779	2.81
8.738	3.017	3.013	-.--
13.117	3.232	3.294	3.26

Zeit. Das Programm wurde speziell für den Einsatz auf Personal Computern konzipiert und gestattet dem Benutzer ebenfalls die Wahl zwischen zwei unterschiedlichen, iterativen Gleichungslösern.

Als Testbeispiel wurde ein ungespannter Aquifer mit einer horizontalen Ausgangspiezometerhöhe von 50 m und folgenden Formationsparametern berechnet: $k_{xx} = 10^{-4} \text{ m/s}$, Anisotropiefaktor $k_{yy}/k_{xx} = 1.3$, der östliche und westliche Modellrand sollte ein Festpotential repräsentieren, während der nördliche und südliche Modellrand als undurchlässig definiert wurden. Die Aquiferbasis wurde über das gesamte Modellareal mit 0 m definiert. Gesucht wurden über einen Zeitraum von 9 Tagen die Ganglinien zweier Brunnen (I und II) sowie die eines Pegels. Der Brunnen I sollte seine Förderung zu Beginn des 1. Tages mit $0.01 \text{ m}^3/\text{s}$ aufnehmen und darauffolgend um täglich $0.01 \text{ m}^3/\text{s}$ gesteigert werden. Der Brunnen II sollte anfänglich $0.03 \text{ m}^3/\text{s}$ leisten und mit Ablauf des 5. Tages, nachdem er zuvor die gleichen Leistungssteigerungen erfahren hatte, abgeschaltet werden.

Die Zeitschrittlänge betrug vereinbarungsgemäß konstant 120 min. Die Lage der Brunnen und des Pegels wird aus Abb. 15 ersichtlich. Das Modellareal wurde mit einem nichtäquidistanten Raster von $20 * 20$ zellenzentrierten Knoten überzogen, wobei die kleinsten (Brunnen-) Zellen

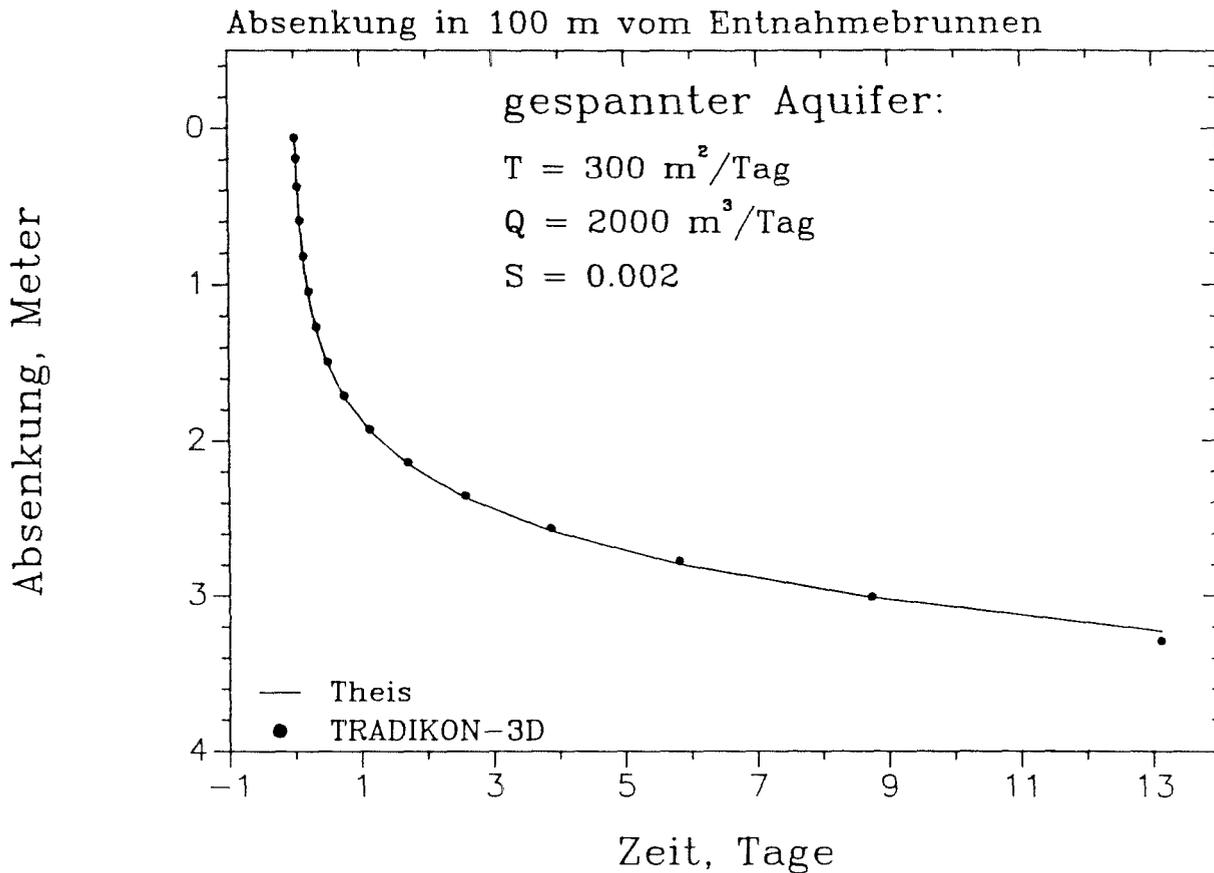


Abb. 14: Zeit-Absenkungskurve des numerischen und analytischen Ansatzes

Kantenlängen von 20 Meter und die größten Zellen Kantenlängen von 50 Meter aufwiesen. Als Iterationskriterium wurde eine Differenz von ± 0.001 Metern vereinbart. Aus den Abbildungen 16 und 17 lassen sich die nahezu deckungsgleichen Verläufe der berechneten Ganglinien ablesen.

Tab. 3: Piezometerhöhen in Meter über NN zu ausgewählten Zeitpunkten nach 5 Tagen:

	ASM	TRADIKON-3D
Brunnen I	43.291	43.257
Brunnen II	40.845	40.781
Pegel	47.087	47.075

nach 9 Tagen:

	ASM	TRADIKON-3D
Brunnen I	37.495	37.376
Brunnen II	48.080	48.075
Pegel	45.993	45.984

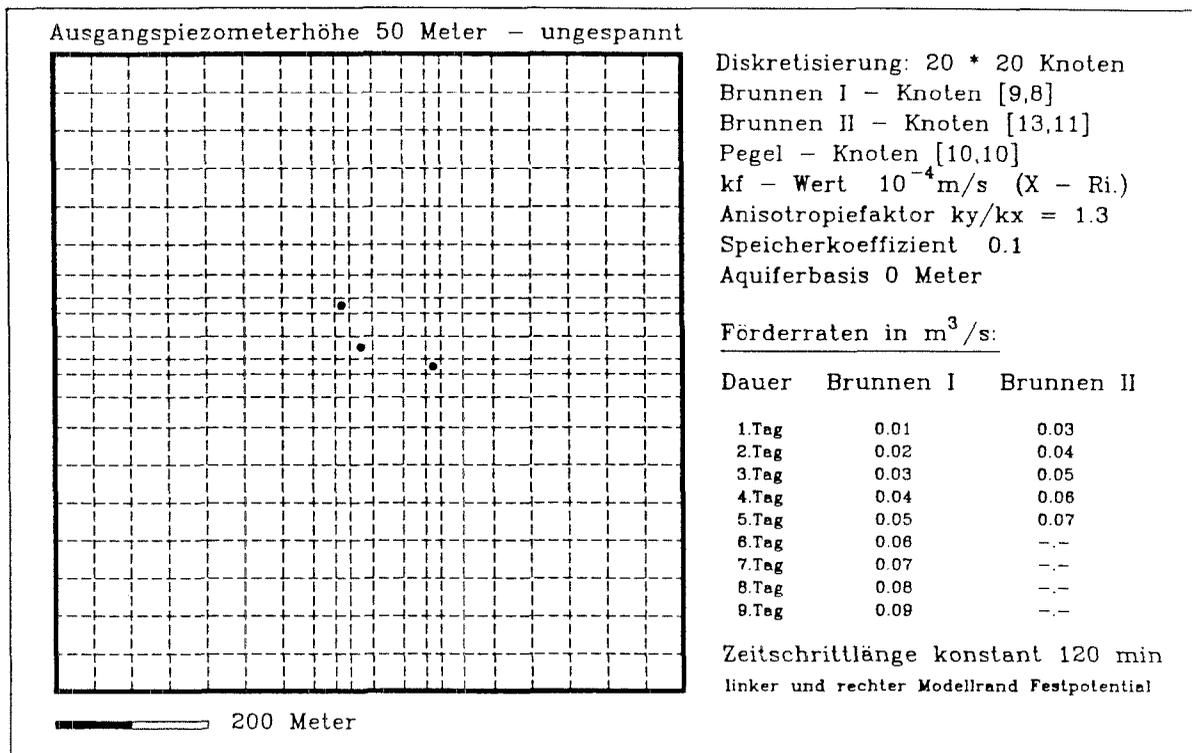


Abb. 15: Eingangsdaten und Diskretisierung der Simulation des Stufenpumpversuches

Die geringen Abweichungen können u.U. durch die unterschiedlichen Verfahren der Berechnung der Transmissivitäten zwischen den Zellen verursacht sein. Während ASM das einfache geometrische Mittel einsetzt, wird in TRADIKON-3D das gewichtete geometrische Mittel benutzt.

5.6 Leakage aus einem Oberflächengewässer

Ein stationärer Lauf sollte Aufschluß über die Richtigkeit der numerischen Behandlung der Leakage aus einem Oberflächengewässer in den darunter liegenden Aquifer liefern. Dazu wurde ein Testbeispiel aus KINZELBACH (1986) S. 80 übernommen, das wie folgt lautet:

Ein isotroper, ungespannter Aquifer mit einer Durchlässigkeit von $3 \cdot 10^{-4}$ m/s sei an seinem Ost- und Westrand durch je ein Festpotential mit einer Piezometerhöhe von 75 bzw. 80 Meter ü. NN begrenzt. Die Auiferbasis soll bei 10 Meter ü. NN liegen. Der Nord- und Südrand sei als

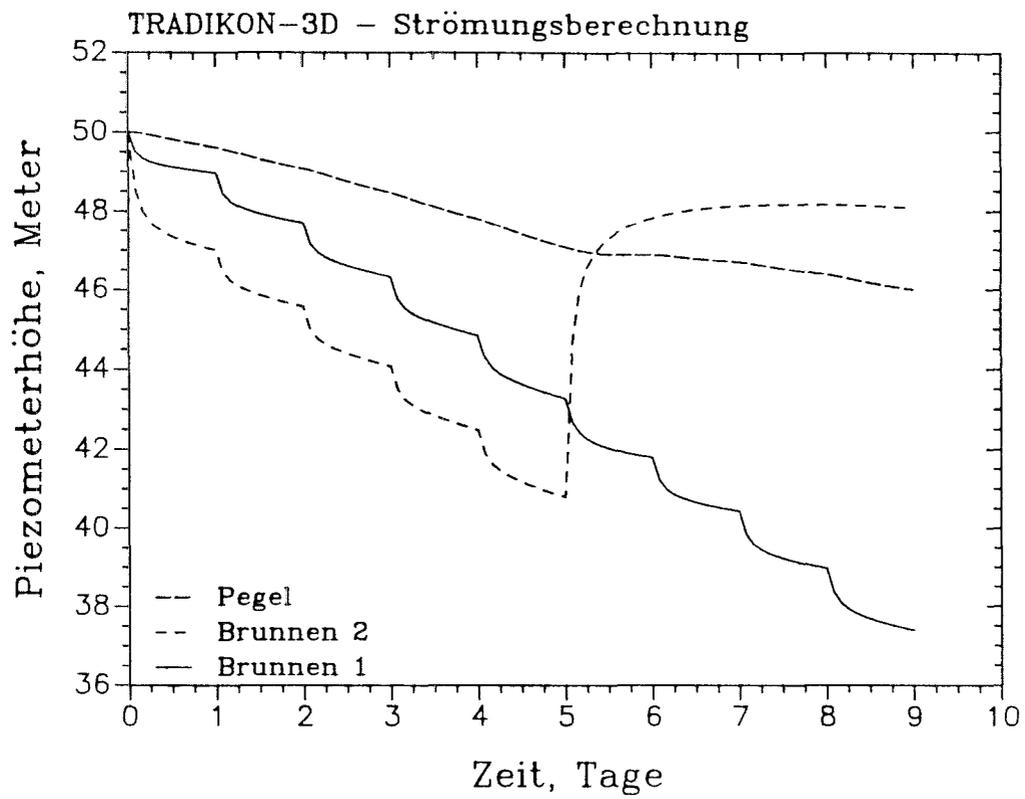


Abb. 16: TRADIKON-3D Simulation des Stufenpumpversuches

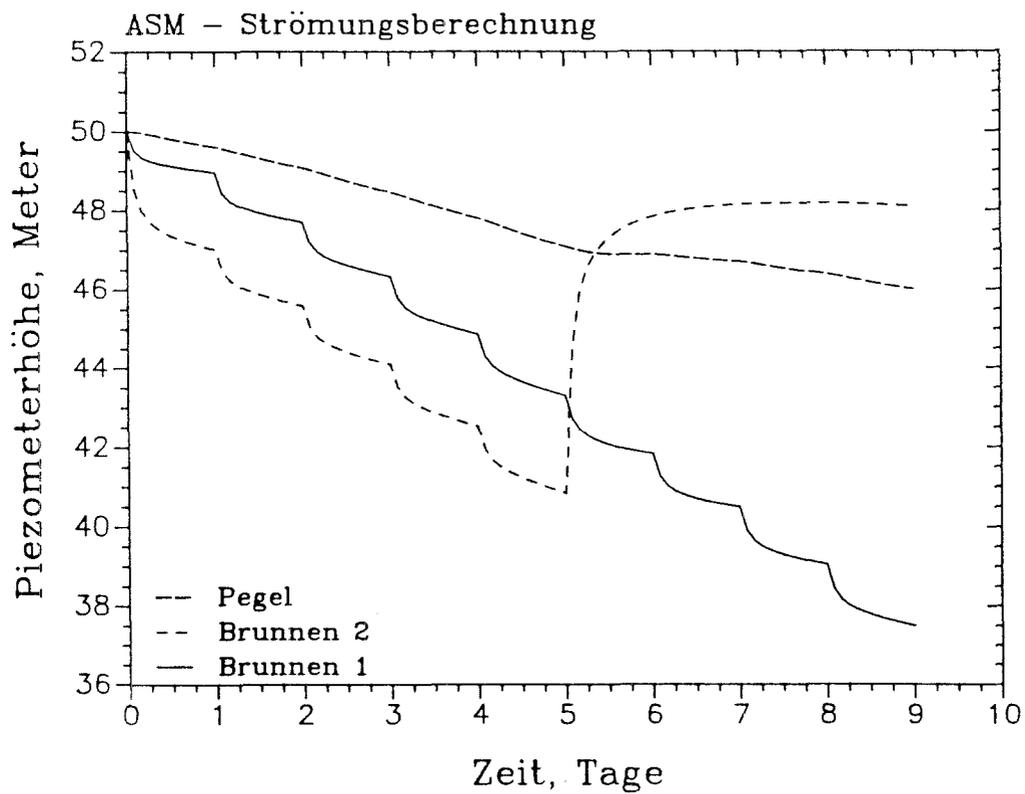


Abb. 17: ASM Simulation des Stufenpumpversuches

undurchlässiger Rand definiert, dessen Verlauf in Abb. 18 durch den schräg schraffierten Bereich markiert wird.

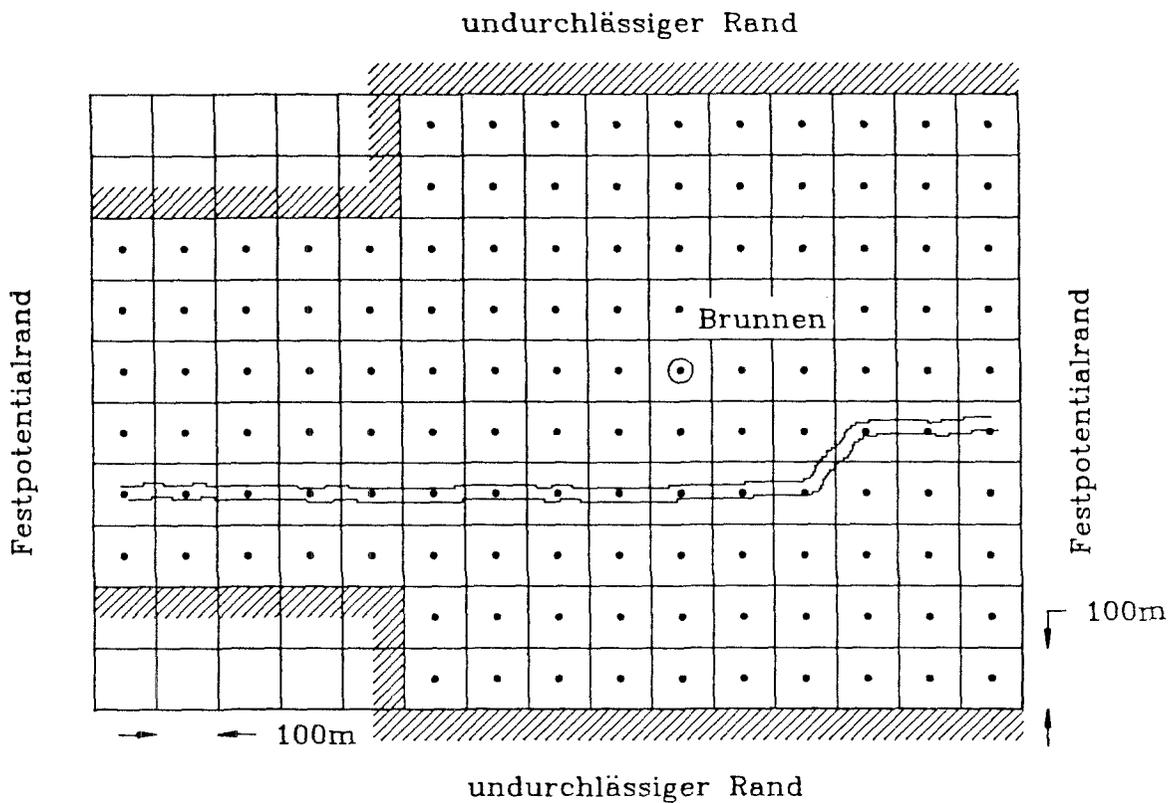


Abb. 18: Diskretisierung des ungespannten Aquifers

Das durch ein äquidistantes Raster von $15 \cdot 10$ zellenzentrierten Knoten diskretisierte Modellareal wird von Westen nach Osten durch einen Fluß gequert, dessen Sohle vom Westrand mit 79 Meter ü. NN nach Osten auf 72 Meter ü. NN mit einem konstanten Gefälle von etwa 3.5 Promille abtaucht. Die Flußsohle weist in ihrer gesamten Erstreckung einen Leakagefaktor von $5 \cdot 10^{-6} s^{-1}$ auf. Der Wasserspiegel des Flusses soll 3 Meter über seiner Sohle stehen. Ein nördlich des Flusses gelegenes Wasserwerk fördert über einen Brunnen konstant $0.1 m^3/s$. Gesucht wird die stationäre Piezometerhöhenverteilung, die sich unter der Annahme einer konstanten Grundwasserneubildungsrate von $3 \cdot 10^{-9} m^3/s/m^2$ einstellt.

Aus dem fischgrätenartigen Verlauf der Äquipotentiallinien erkennt man sehr gut, daß der Fluß auf seiner gesamten Länge in den Aquifer infiltriert und ein Großteil des vom Wasserwerk geförderten Wassers

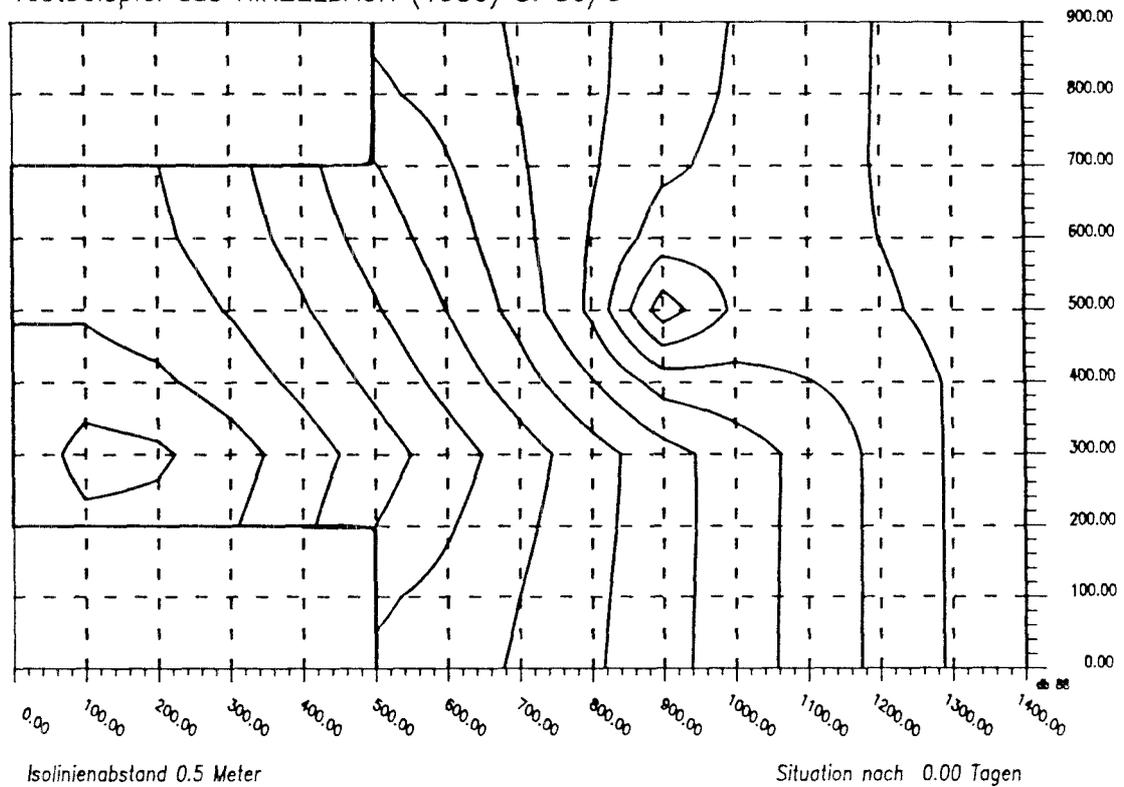


Abb. 19: TRADIKON-3D Simulation der Leakage aus einem Oberflächen-
gewässer

somit indirekt dem Fluß entstammt. Der Ergebnisvergleich mit dem Aquifersimulationsmodell ASM von KINZELBACH & RAUSCH (1988) für zwei exemplarische Knoten ergab folgendes Bild:

Tab. 4: Berechnete Piezometerhöhen zweier Brunnen in Meter ü. NN

	TRADIKON-3D	ASM
Knoten [10, 5]	74.73	74.74
Knoten [2, 7]	80.74	80.74

Diese Knoten repräsentierten zugleich die höchste bzw. niedrigste Piezometerhöhe innerhalb des Modellareals. Bei einem vorgegebenen Abbruchkriterium von 10^{-3} Metern benötigte TRADIKON-3D 8 Iterationen (LSOR, $\Omega = 1.64$) und ASM 14, (IADI). Während, wie aus dem Datensatz LEAK im Anhang 4 ersichtlich, die TRADIKON Simulation als dreidimensionaler Lauf gerechnet wurde, verfährt das zweidimensionale Modell ASM so,

daß zunächst aus der Potentialdifferenz des Oberflächengewässers zu dem unterlagernden Aquifer über die Leakage Faktoren eine Sickerwassermenge berechnet wird, die dann den Quellenterm beaufschlagt. Ungeachtet dessen stimmen die Berechnungsergebnisse nahezu exakt überein.

6 Anwendung

6.1 Die Forschungsanlage Schwalbach

6.1.1 Beschreibung der Anlage

Die etwa 7 km südwestlich der mittelhessischen Stadt Wetzlar gelegene Forschungseinrichtung gestattet den Betrieb von Erdsondenwärmepumpenanlagen unter realitätsnahen Bedingungen. Sie besteht aus zwei Bohrfeldern mit insgesamt 14 Bohrungen und einem Laborgebäude, das die Wärmepumpe sowie die zentrale Datenaufzeichnung beherbergt. Das erste Bohrfeld besteht aus 10 jeweils 50 m tiefen Bohrungen, die radialstrahlig um die zentrale Entzugsbohrung Z angeordnet sind. Während die Bohrungen H1 und H2 hydraulischen Tests und der Entnahme von Grundwasserproben dienen, kann in den übrigen Bohrungen 1/1 bis 5/1 über Meßsonden das aus den Energieentzügen resultierende Temperaturfeld aufgezeichnet werden /175/. In der Entzugsbohrung Z wurden im Laufe eines Forschungsprojektes /176/ unterschiedliche Wärmetauscher, in denen als Wärmeträgerflüssigkeit eine froststabile Sole zirkuliert, getestet.

In die Bohrungen des zweiten Bohrfeldes wurden zu Testzwecken Wärmetauscher eingebracht, in denen das Kältemittel direkt verdampft wird.

Die während der Bohrarbeiten durchhörte Wechselfolge aus Tonschiefer und Grauwacken oberdevonischen bis unterkarbonischen Alters, wird von einer einer 2 m mächtigen Boden- und Fließerdenauflage überdeckt. Die stark tektonisch beanspruchten, paläozoischen Gesteine sind durch junge Hebungsvorgänge des Gebirges geklüftet worden. An geologischen Aufschlüssen der unmittelbaren Umgebung läßt sich eine NW-SE streichende Q-Klüftung und eine variszisch streichende L-Klüftung nachweisen, /175/. Diese Trennflächen stellen zugleich die einzigen wasserwegsamten Hohlräume des Untergrundes dar. Im rechtsrheinischen Schiefergebirge betragen die Auflockerungstiefen nur in Ausnahmefällen mehr als 50 m. Der in einer Tiefe von 15 m unter Geländeoberkante angetroffene Grundwasserleiter ist somit als Klüftgrundwasserleiter einzustufen. Aufgrund der exponierten Lage des Standortes auf einem Bergrücken und dem somit geringen hydraulischen Gradienten sind die Grundwasserfließgeschwindigkeiten relativ gering, was durch einen Tracerversuch auch bestätigt werden konnte /175/.

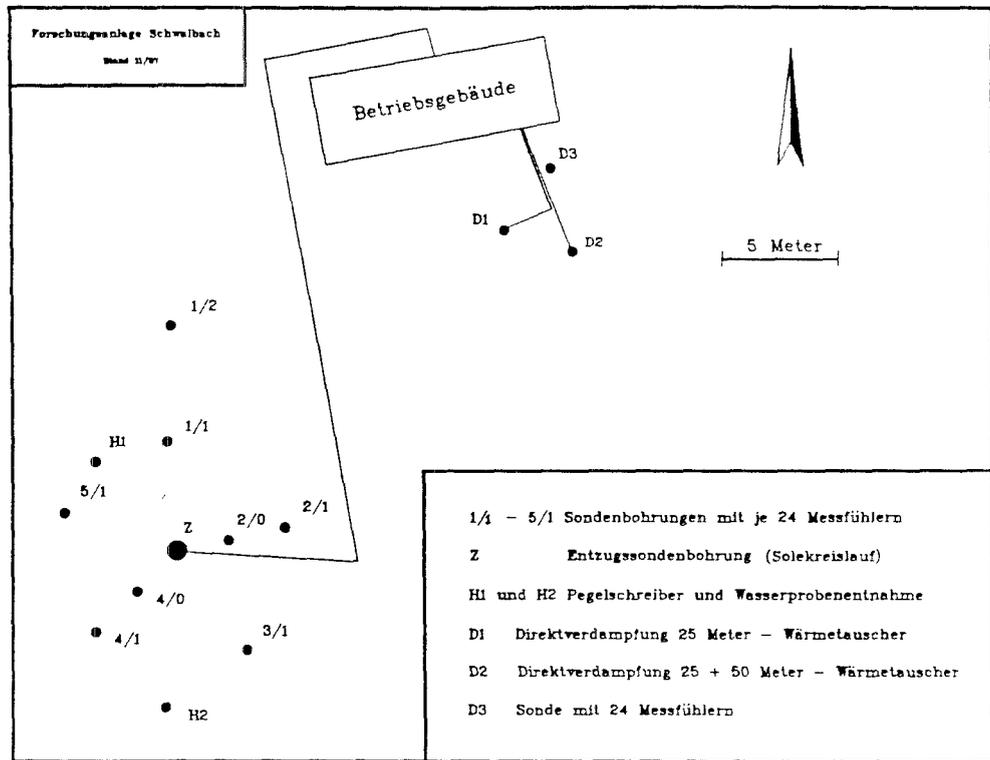


Abb. 20: Lageplan der Erdsondenforschungsanlage Schwalbach im November 1987

Der durch das Wiederanstiegsverfahren nachgewiesenermaßen geringe Speicherkoeffizient des ungespannten Aquifers gilt als Ursache für die relativ hohen jahreszeitlichen Schwankungen des Grundwasserspiegels /175/. Die Transmissivität des Aquifers läßt sich überschlägig mit $6.4 \cdot 10^{-4} m^2/s$ beziffern, was unter Berücksichtigung einer Aquifermächtigkeit von 36 m einem k_f -Wert von $1.8 \cdot 10^{-5} m/s$ entspricht /175/. Die lokale Grundwasserneubildungsrate mit 2 bis $2.5 l/s/km^2$ darf als typisch für diesen Teil des rheinischen Schiefergebirges gelten und ist somit relativ gering.

Die anstehenden Gesteine verfügen über eine thermische Leitfähigkeit von $1.5 W/mK$ (Tonschiefer) bis $3.4 W/mK$ (quarzitische Grauwacke), während die Wärmekapazitäten mit $800 - 850 J/kgK$ naturgemäß ein wesentlich schmaleres Spektrum aufweisen (vergl. Anhang 1). Die Gesteine verfügen mikroskopisch betrachtet über eine unterschiedlich ausgeprägte Anisotropie ihrer Wärmeleitfähigkeit. Da diese sich jedoch innerhalb der Genauigkeit des eingesetzten Meßgerätes bewegt und die geologischen Lagerungsverhältnisse am Projektstandort als nicht restlos geklärt gelten, wurden den weiter unten beschriebenen Kalibrierungsläufen isotrope Leitfähigkeiten zugrundegelegt.

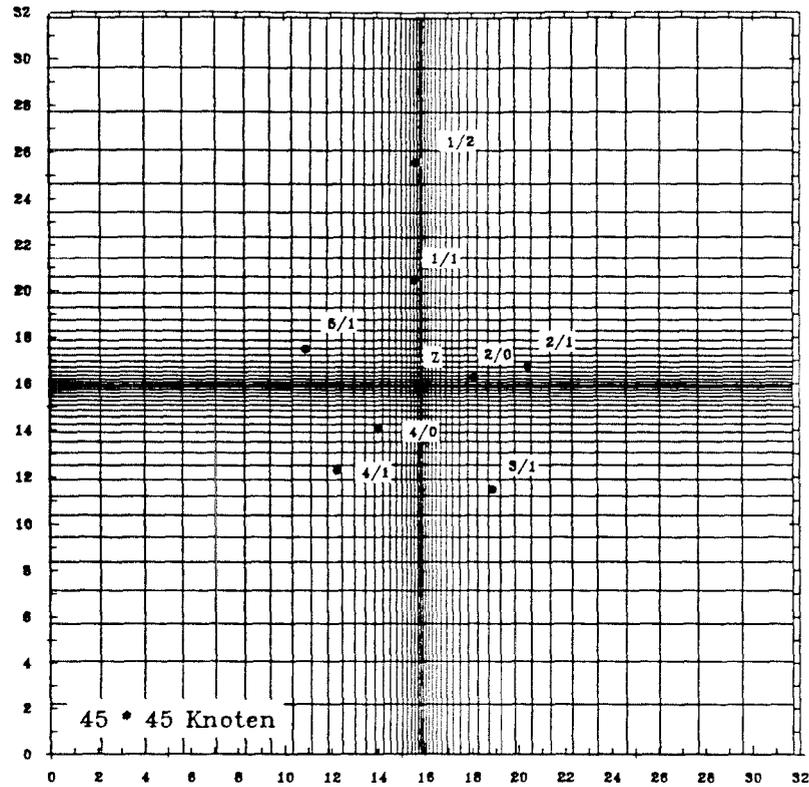


Abb. 21: Lage der Bohrungen in dem finiten Differenzen Raster

Die Anlagenkonzeption gestattet diverse Betriebsmodi der Wärmepumpe /175/. Mehrere Testentzugsläufe mit quasikonstanter Soletemperatur konnten daher dazu herangezogen werden, das Modell auf den Standort zu kalibrieren. Als etwas nachteilig stellte sich dabei jedoch heraus, daß auch relativ weit zurückliegende Wärmeentzüge das Temperaturfeld nachhaltig gestört hatten und somit zu keinem Zeitpunkt, außer zu Beginn des Forschungsvorhabens, stationäre Bedingungen vorlagen. Somit mußten neben den modelltechnisch bedingten Vereinfachungen auch diverse Vereinfachungen bezüglich des Anfangszustandes getroffen werden.

6.1.2 Diskretisierung

Da die Wärmetauscherlänge von 50 m etwa eine Größenordnung über dem zu erwartenden Einflußradius einer 4-wöchigen Wärmeentzugsperiode liegt, diente zunächst ein zweidimensionaler Lauf für Parameterstudien an den unterschiedlichen Einflußgrößen. Auf halber Länge des Wärmetauschers ist während eines Energieentzuges geometriebedingt der vertikale Wärmefluß gegenüber dem horizontalen vernachlässigbar. Daher wurde ein Teufenschnitt von 28 m unter GOK durch ein zweidimensionales Raster von 45 * 45 zellenzentrierten Knoten diskretisiert.

Daß bei der getroffenen Diskretisierung einige Bohrungen nicht im Zentrum ihrer zugehörigen Zellen liegen, wurde bewußt in Kauf genommen, da sich der Lageplan auf die Vermessung an der Geländeoberfläche bezieht und alle Bohrungen, durch Inklinometermessungen nachgewiesen, mehr oder minder von der Lotrechten abweichen, /175/. - Derartige Messungen weisen naturgemäß einen gewissen Fehler auf, sodaß eine exakte Diskretisierung innerhalb der vorgegebenen Zellengrößen als nicht realisierbar gelten darf.

6.1.3 Wärmeentzug mit quasikonstanter Soletemperatur

Der in den Monaten Juli und August 1988 durchgeführte Wärmeentzugstest mit quasikonstanter Soletemperatur eignete sich für die Kalibrierung des Modells am besten, da nach 31-tägiger Laufzeit der Anlage über einen Zeitraum von mehr als 40 Tagen die Regeneration aufgezeichnet werden konnte.

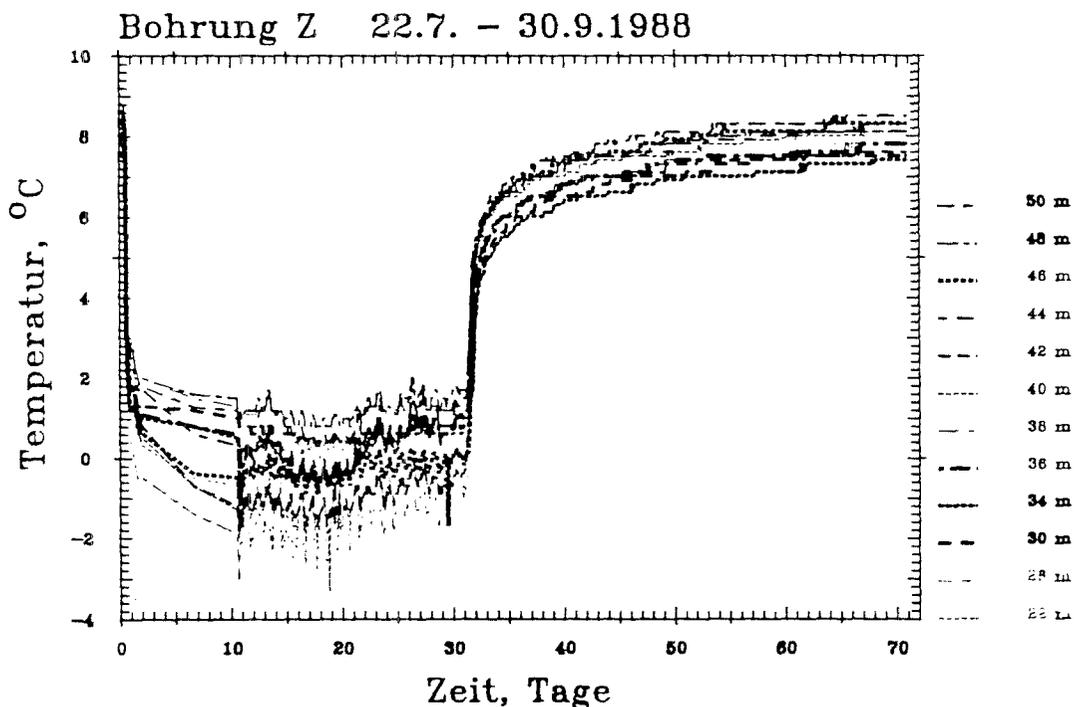


Abb. 22: Temperaturentwicklung in der Entzugsbohrung Z

Da die Meßfühler der Temperatursonde aus technischen Gründen nicht direkt auf den Wärmetauscher aufgeklebt werden konnten, hängen die aufgezeichneten Temperaturen stark von der Entfernung zum Wärmetauscher

ab. Hierin ist die Ursache für die relativ große Temperaturspreizung von etwa 3 K entlang des Bohrloches zu sehen. Vermutlich wiesen einige Fühler auch eine leichte Drift von der Kalibriertemperatur auf. Wie auch immer, ein zu erwartender Effekt kann deutlich registriert werden: Im unteren Bereich des Bohrloches erfolgt die Regeneration des Temperaturfeldes nach Abschalten der Anlage wesentlich spontaner als im darüberliegenden Bereich. Die Ursache hierfür ist im halbkugelartig auf das Wärmetauscherende gerichteten Wärmefluß zu sehen.

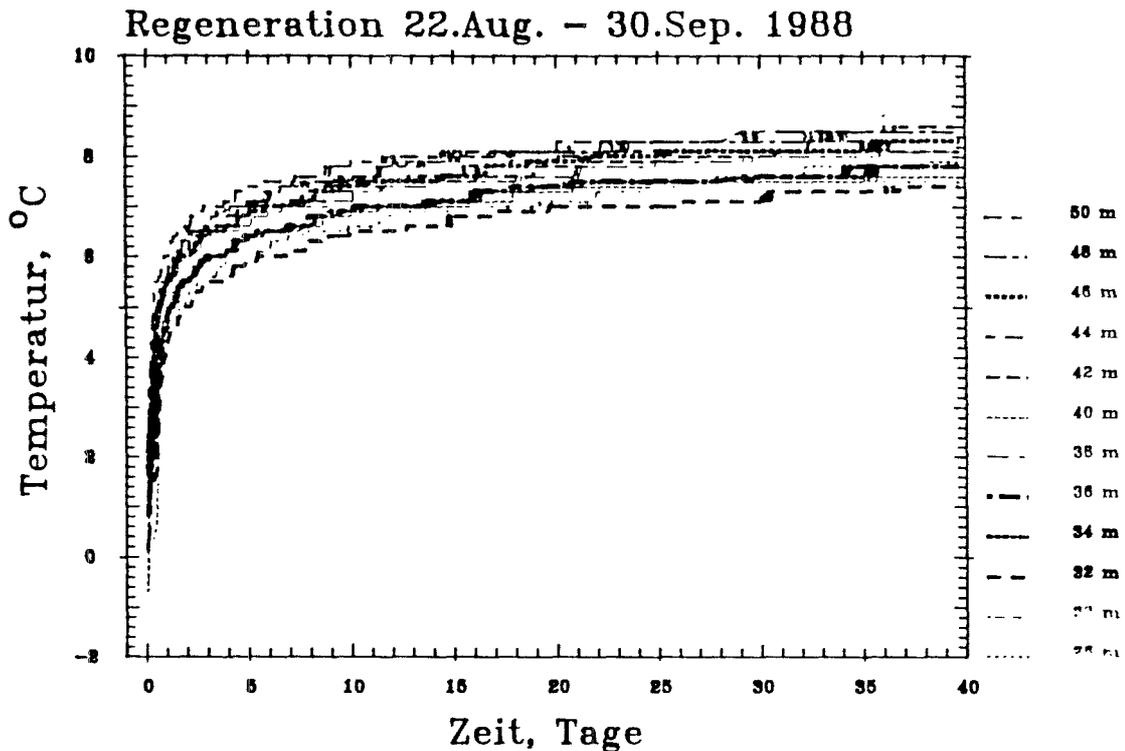


Abb. 23: Gemessene Regeneration der Erdreichtemperaturen nach 31-tägigem Wärmeentzug.

Als Reaktion auf den 31-tägigen Wärmeentzug konnte nur in den Bohrungen 2/0 und 4/0 eine signifikante Temperaturerniedrigung registriert werden. Bei den übrigen Bohrungen reichte die Auflösung der installierten Temperaturfühler von 0.2 K nicht dazu aus, eine eindeutige Reaktion zu messen.

In die TRADIKON-Simulation mußten während der Entzugsperiode die in stündlichen Intervallen gemessenen Daten mit einbezogen werden, da die Soletemperaturen eben doch nur "quasikonstant" gehalten werden konnten. Dazu wurde der zentral gelegene Knoten mit den Indizes (23,23,1) als

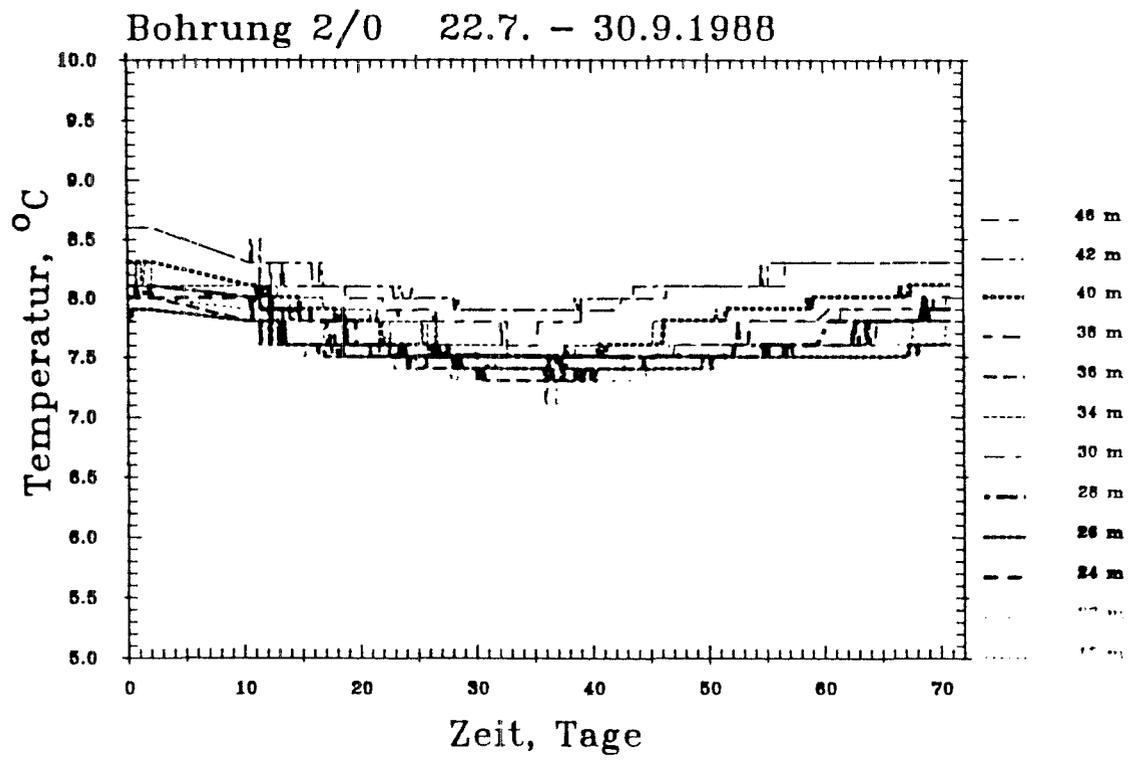


Abb. 24: Temperaturerniedrigung in Bohrung 2/0

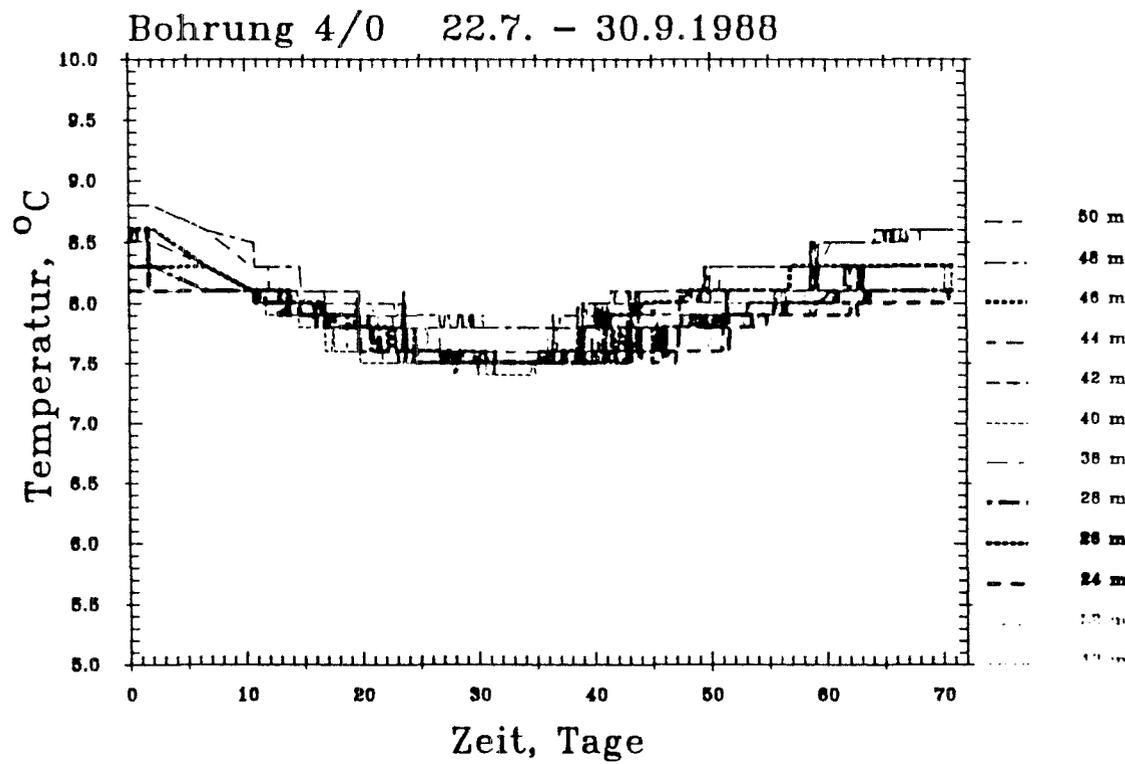


Abb. 25: Temperaturerniedrigung in Bohrung 4/0

Festtemperaturknoten definiert. Nach Abschalten der Wärmepumpe wurde das Datenfile von dem Prozessor entkoppelt und die Temperaturregeneration berechnet. Über den gesamten Simulationszeitraum von 104 Tagen betrug die Zeitschrittlänge konstant 1 Stunde (2500 Zeitschritte). Bereits einer der ersten rein konduktiven Simulationsläufe ergab eine nahezu perfekte Übereinstimmung mit den gemessenen Temperaturwerten.

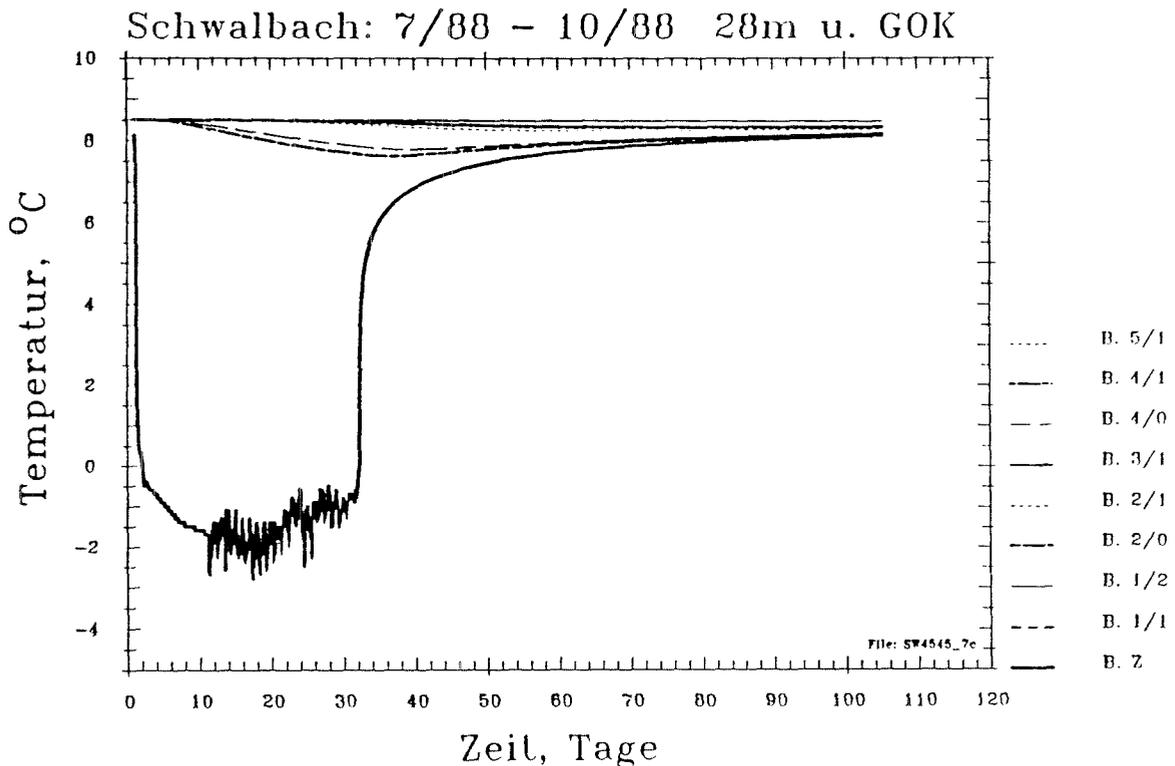


Abb. 26: Rein konduktiv berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 7, Tab. 5)

Bemerkenswert gut war auch die Übereinstimmung in den ersten 24 Stunden nach Abschalten der Anlage. Da sich während des Betriebes der Wärmepumpe die Soletemperatur stets unterhalb des Gefrierpunktes von Wasser lag, mußte sich ein Eispanzer um den Wärmetauscher bilden, der anschließend zu einer verzögerten Regeneration führte. Der in Kap. 2.5 beschriebene Gefrieralgorithmus ist also in der Lage, die gemessenen Vorgänge hinreichend genau zu reproduzieren.

Mit einer Reihe von Simulationsläufen konnte die Bedeutung der in die Berechnung einfließenden physikalischen Parameter abgeschätzt werden. Alle Berechnungen wurden mit dem Großrechner *CDC 860* des Hochschulrechenzentrums (HRZ) Gießen durchgeführt, wobei die

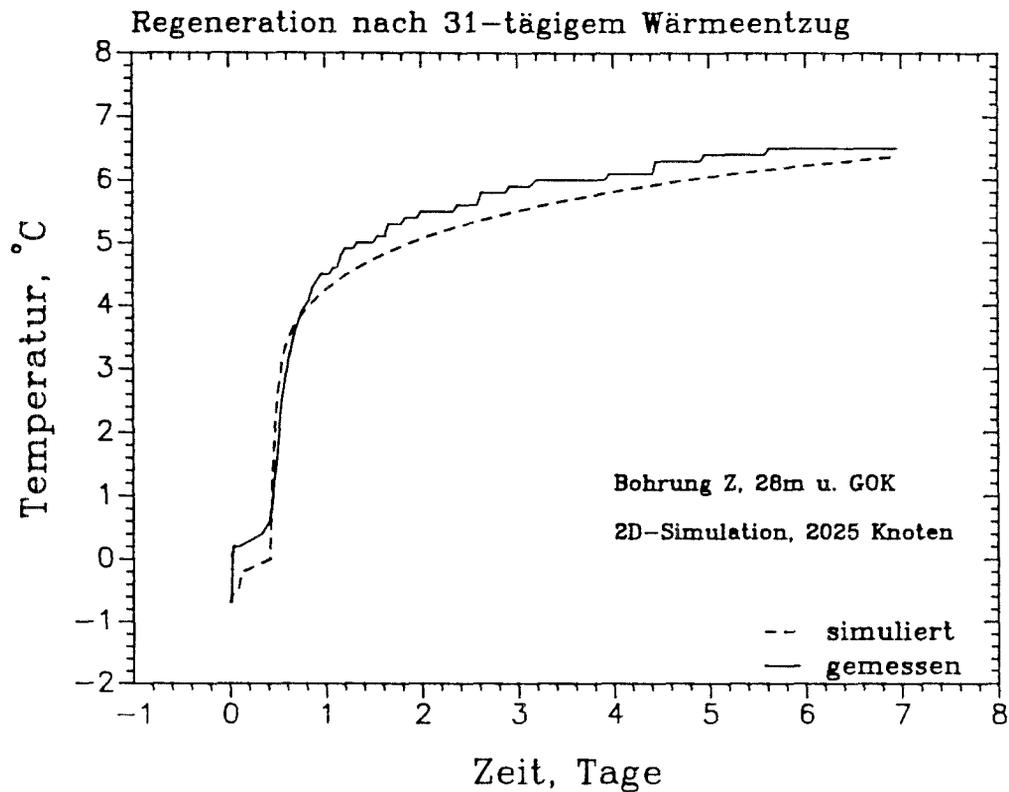


Abb. 27: Durch die Schmelzenthalpie des Wassers hervorgerufene Verzögerung der Temperaturregeneration in der Entzugsbohrung.

pro Simulationslauf benötigte Rechenzeit zwischen 3300 und 6800 CPU-Sekunden schwankte. Die Diagramme der Kalibrierungsläufe können dem Anhang 8 entnommen werden.

Tab. 5: Eingangsparameter der 9 Kalibrierungsläufe

Parameter\Lauf	1	2	3	4	5	6	7	8	9
Ausgangstemperatur	8.8	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5
Wärmeleitfähigkeit des Gesteins	3.0	3.0	3.0	1.5	3.0	3.0	3.0	3.0	3.0
Wärmekapazität des Gesteins	850	850	700	850	850	850	850	850	850
Porosität	3.0	25.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
Diffusion	+	+	+	+	+	+	+	+	+
Konvektion	-	-	-	-	+	+	-	+	-
Fließgeschwindigkeit	0.0	0.0	0.0	0.0	$5 \cdot 10^{-6}$	$5 \cdot 10^{-7}$	0.0	0.0	0.0
Gefrieralgorithmus	-	-	-	-	-	+	+	+	-

den Wärmetransport usw. /66/, /128/, /137/, /157/, /198/ — ist die Kalibrierung eines für den Wärmetransport in der Pedosphäre geeigneten Modells, ohne die oben erwähnten Punkte a priori zum Scheitern verurteilt. Vor diesem Hintergrund ist die nachfolgend vorgestellte, rein konduktive Simulation zu betrachten.

Zunächst wurde eine niederschlagsarme Periode ausgewählt um den Einfluß der Konvektion vernachlässigen zu können. Da sich gleichzeitig große Temperaturvariationen im obersten Bodenkompartment am ehesten für die Kalibrierung des Modells eignen, erschienen die Temperaturkurven des Monats August 1988, in dem nur etwa 12.5 mm Niederschläge fielen, ideal.

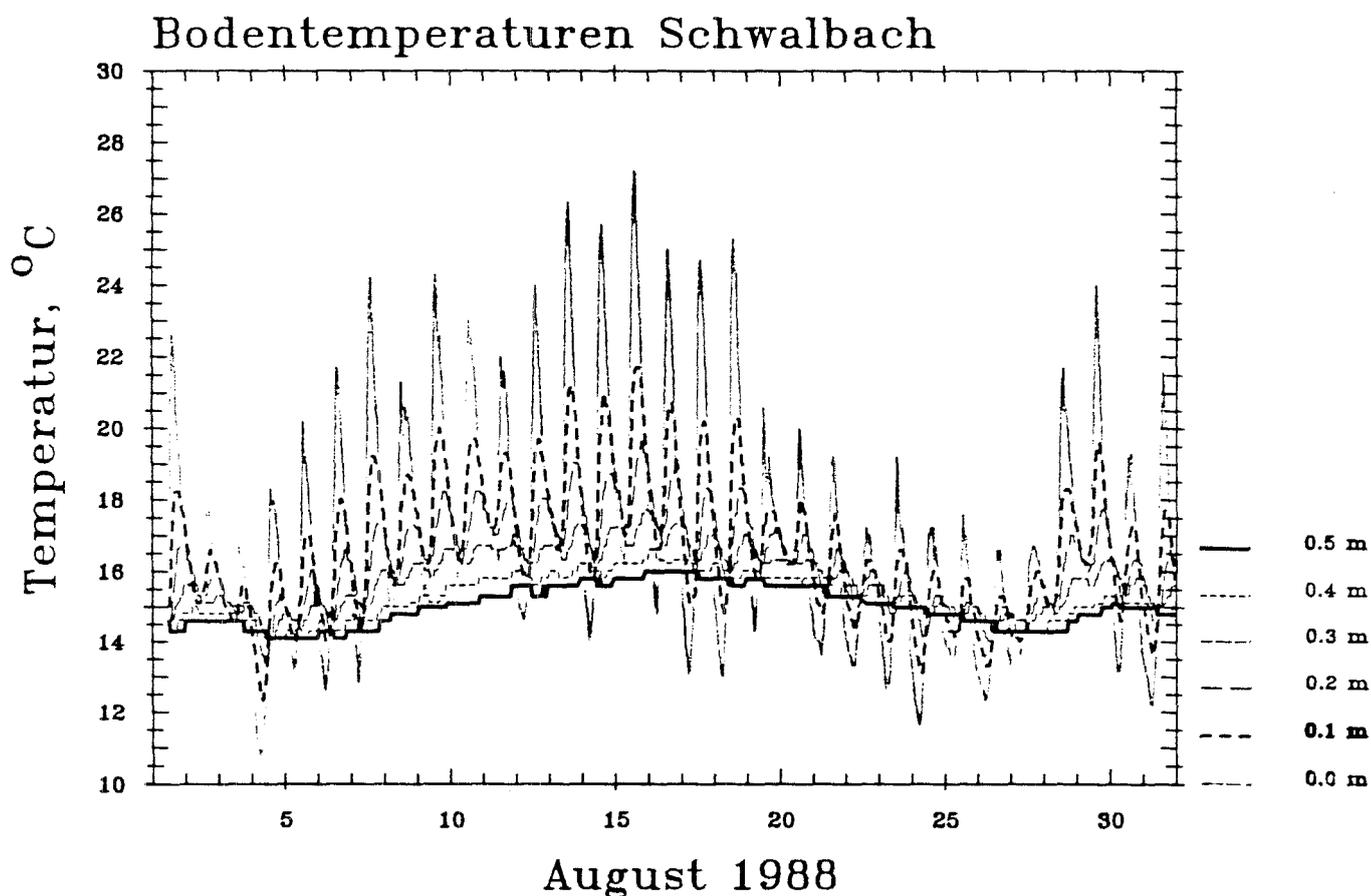


Abb. 28: Gemessene Temperaturen in den obersten 0.5 Meter des Schwalbacher Bodenprofils

Das zu simulierende Boden- und Gesteinsprofil wurde eindimensional durch eine nichtäquidistante Anordnung von 39 zellenzentrierten Knoten

6.1.4 Wärmetransport im Boden

Zur Abschätzung der Auswirkungen der Klimafaktoren auf den Wärmehaushalt des Bodens und der ihn unterlagernden Gesteine wurde am Standort der Forschungsanlage ein aufwendiges Meßprogramm gefahren. Neben den stündlichen Niederschlägen wurden die Lufttemperaturen in verschiedenen Höhen über Geländeoberkante (GOK) und die Windgeschwindigkeit in 3 m Höhe registriert, während das zur Messung der Globalstrahlung installierte Sternpyranometer leider keine interpretationswürdigen Daten lieferte. Das resultierende Temperaturfeld konnte in einer 2.4 m tiefen, etwa 1 m von 1/2 entfernten Bohrung beobachtet werden, die in vertikalen Abständen zu je 10 cm mit insgesamt 24 Temperaturmeßsonden bestückt war. Da ja der Schwerpunkt des Forschungsprojektes eindeutig anders akzentuiert war, konnten die folgenden, wichtigen Einflußgrößen nicht untersucht werden:

- Porositäts - Tiefen - Profil
- hydraulische Leitfähigkeit in verschiedenen Tiefen
- Ganglinie des Wassergehaltes in verschiedenen Tiefen
- Feldkapazität in verschiedenen Tiefen
- mineralogische Zusammensetzung des Bodensubstrates
- Höhe und Dauer der Schneebedeckung im Winter
- biogene und chemische Wärmeproduktion im Boden
- Einfluß der Vegetation
- Einfluß der feuchtigkeitsabhängigen Tönung des unbewachsenen Bodens und der damit verbundenen Adsorptionsfähigkeit von Infrarotstrahlung
- Wärmeübergangskoeffizient an der Boden - Luft - Schnittstelle
- und vieles weitere

Da es sich beim diffundiv - konvektiven Wärmetransport in der ungesättigten Bodenzone um einen im höchsten Maße gekoppelten Prozess handelt — der Wasserhaushalt wird von dem herrschenden Temperaturfeld kontrolliert, die diffundiv - konvektive Feuchtigkeitsmigration beeinflusst

diskretisiert, wobei der unterste Knoten mit einer über die gesamte Simulationsdauer konstanten Temperatur von $8.8\text{ }^{\circ}\text{C}$ 19 Meter unter GOK lag. Die Temperatur des oberen Randkompartiments wurde aus der gemessenen Kurve — 0.0 m in Abb. 28 — vorgegeben. Die Anfangsbedingungen, die maßgeblich die Qualität der Ergebnisse einer solchen Simulation beeinflussen, ergaben sich aus der stationären Wärmetransportberechnung unter Berücksichtigung aller zu diesem Zeitpunkt bekannten Temperaturen, also auch denen der Bohrung $1/2$. Über den gesamten Simulationszeitraum von 31 Tagen betrug die Zeitschrittlänge konstant 60 Minuten. In den selben Intervallen wurden die Temperaturen des obersten Knotens durch die Meßwerte einer externen Datei erneuert. Die thermophysikalischen Eigenschaften des Bodens wurden wie folgt angenommen:

- Wärmekapazität des Festkornes 850 J/kgK
- Wärmeleitfähigkeit: $0 - 5.5\text{ m u.GOK } 1.8\text{ W/mK}$; darunter 2.0 W/mK
- Dichte des Festkornes 2.65 g/m^3

Der Kalibrierungslauf mit der besten Anpassung an die gemessenen Temperaturkurven wies folgende Eingangsparameter auf:

- Gesamtporosität Φ_g von 0.35 allmählich auf 0.05 nach unten hin abnehmend
- Wassersättigungsgrad Ψ von 20% auf 100% nach unten hin zunehmend

Als Kalibriergrößen wurden somit nur die nach /210/ berechneten effektiven Wärmeleitfähigkeiten und die nach Gl. (3.50) berechneten Wärmekapazitäten herangezogen.

Wie aus dem Vergleich der Abb. 28 und 29 deutlich wird, ist die Übereinstimmung der gemessenen mit den berechneten Kurven trotz der starken Vereinfachungen erstaunlich gut. In beiden Fällen paust sich die enorm oszillierende Tagesganglinie je nach Tiefe mehr oder weniger gedämpft und phasenverschoben auf die tieferen Bodenhorizonte durch, bis schließlich bereits 0.5 m unter GOK nur noch ein Großzyklus erkennbar ist.

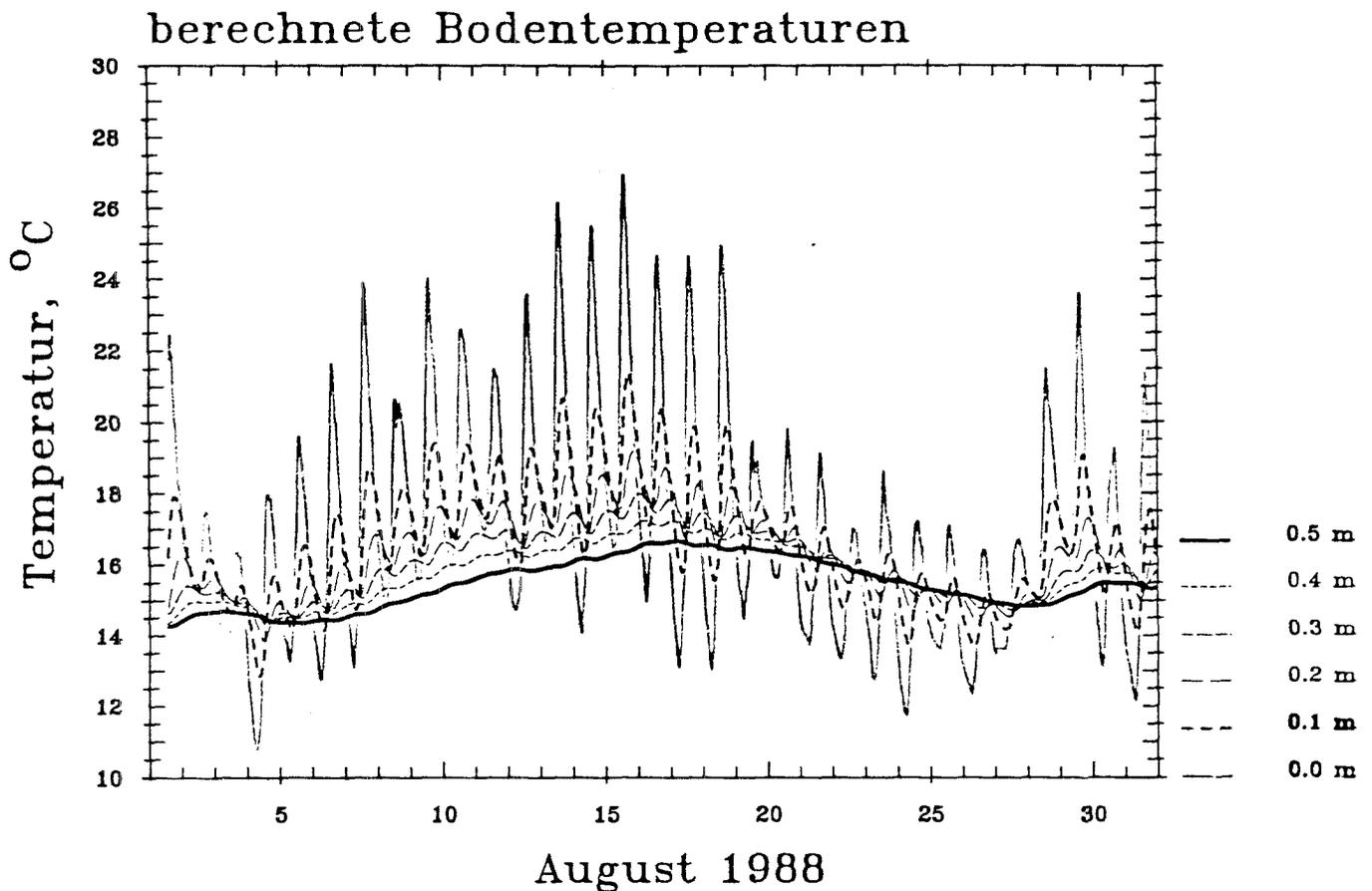


Abb. 29: Simulierte Temperaturen in den obersten 0.5 Meter des Schwabacher Bodenprofils

6.2 Erdsondenanlage Göttingen

6.2.1 Beschreibung der Anlage

Die der Beheizung eines Einfamilienhauses dienende Erdsondenwärmepumpenanlage Göttingen besteht aus 6 je 40 m tiefen Bohrungen und einer Wärmepumpe mit einer Abgabeleistung von 17 kW. Während der Bohrarbeiten wurden die Ceratitenkalke des oberen Muschelkalkes durchörtet. Für diese Gesteine — im wesentlichen Kalke und Mergel — läßt sich überschlägig eine isotrope Wärmeleitfähigkeit von 3.4 W/mK und eine Wärmekapazität von 820 J/kgK abschätzen, (vergl. Anhang 1), während der Gesamtporenraum mit 5% beziffert wurde. Der Grundwasserflurabstand beträgt aufgrund der exponierten Lage auf einem Bergrücken mehr als die erbohrten 40 m, weshalb die nachfolgende Simulationsrechnung rein konduktiv durchgeführt werden konnte. Die Lage der Bohrungen zueinander

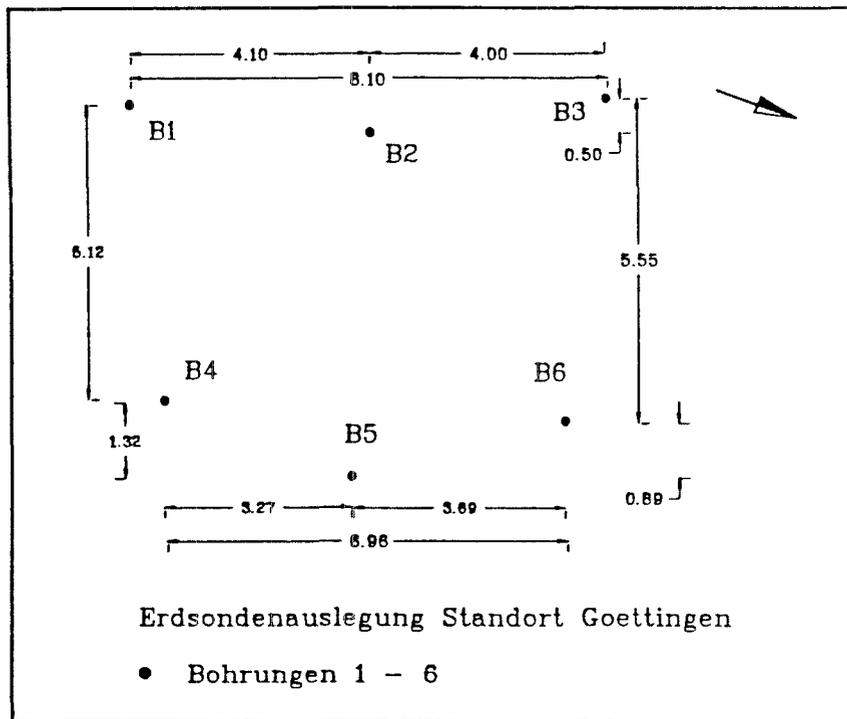


Abb. 30: Lageplan der Erdsondenanlage Göttingen

geht aus Abb. 30 hervor.

In allen Bohrungen wurden Wärmetauscher installiert, während der Testphase waren jedoch nur die Bohrungen B1, B2, B3, B5 und B6 mit der Wärmepumpenanlage verbunden. Die Beobachtungsbohrung B4 sollte der Aufzeichnung des resultierenden Temperaturfeldes dienen, die installierte Temperatursonde versagte jedoch irreparabel kurze Zeit nach Inbetriebnahme der Anlage. Vor dem Hintergrund, daß somit keine Möglichkeit bestand, das Modell auf den Standort zu kalibrieren, sind die nachfolgend beschriebenen Simulationen zu betrachten.

6.2.2 Diskretisierung

Zwei zweidimensionale Simulationen sollten Aufschluß über die mögliche Beeinflussung der Bohrungen untereinander und die Auswirkungen verschiedener Betriebsmodi der Wärmepumpe auf die anschließende Regeneration liefern. Von besonderem Interesse waren auch die langfristigen Auswirkungen der Energieentzüge auf die Erdreichtemperaturen. Dazu wurde das Areal durch ein sehr feinmaschiges Raster von 79×67 zellenzentrierte

Knoten — insgesamt 5293 — diskretisiert. Die kleinsten Zellen hatten eine Kantenlänge von 6 mal 6 cm und die Vergrößerung der Kantenlängen zweier benachbarter Zellen betrug maximal 35%. Für die Modellränder wurde über den gesamten Simulationszeitraum eine konstante Temperatur von 9.0 °C angenommen.

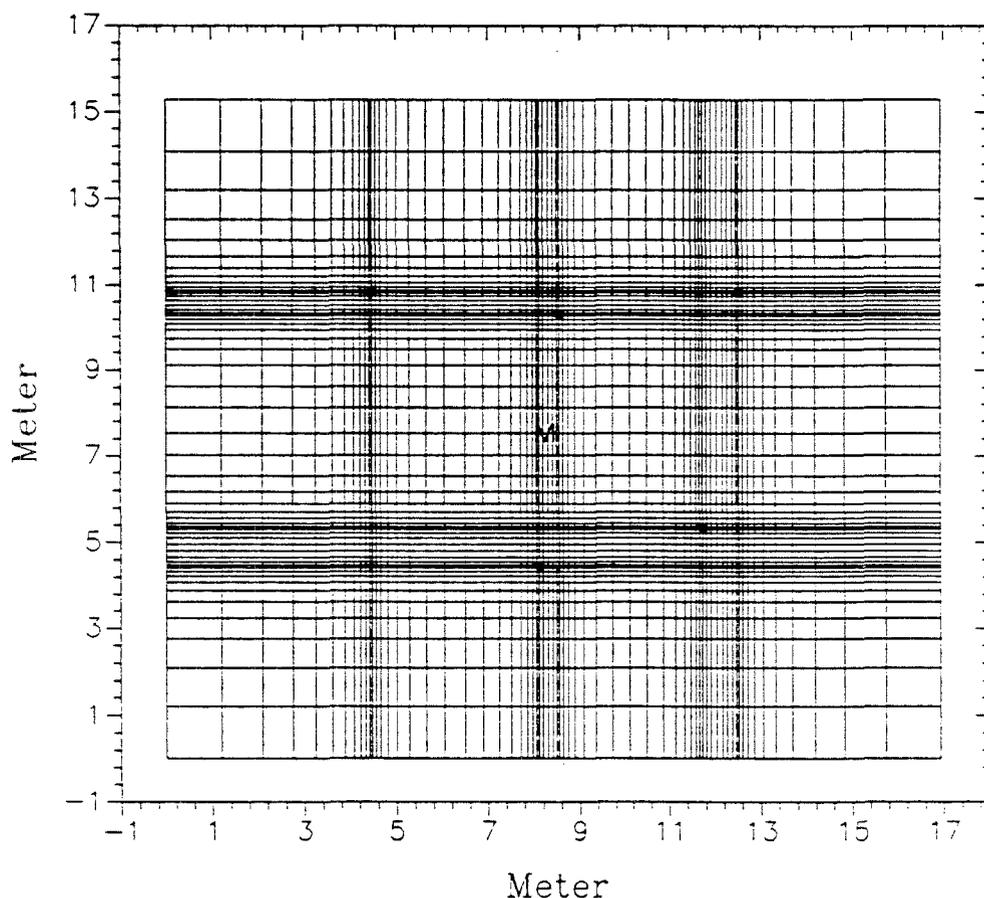


Abb. 31: Ausschnitt der Diskretisierung der Erdsondenanlage Göttingen; die Distanz zu den Modellrändern beträgt an jeder Seite etwa 33 Meter (jeweils 7 Knotenreihen)

Da die Anlage im normalen Heizbetrieb lief, wurden die Betriebszustände der Wärmepumpe primär von den herrschenden Außentemperaturen und der Brauchwasseranforderung kontrolliert. Abb. 32 zeigt exemplarisch die Entwicklung der Erdreichtemperaturen eines 4-wöchigen Zeitraumes der Heizperiode 1987/88 von Bohrung B1. Wie man an dem zackartigen Temperaturverlauf erkennt, wechseln sich zyklisch kurzzeitige Wärmeentzüge und länger anhaltende Regenerationsphasen ab.

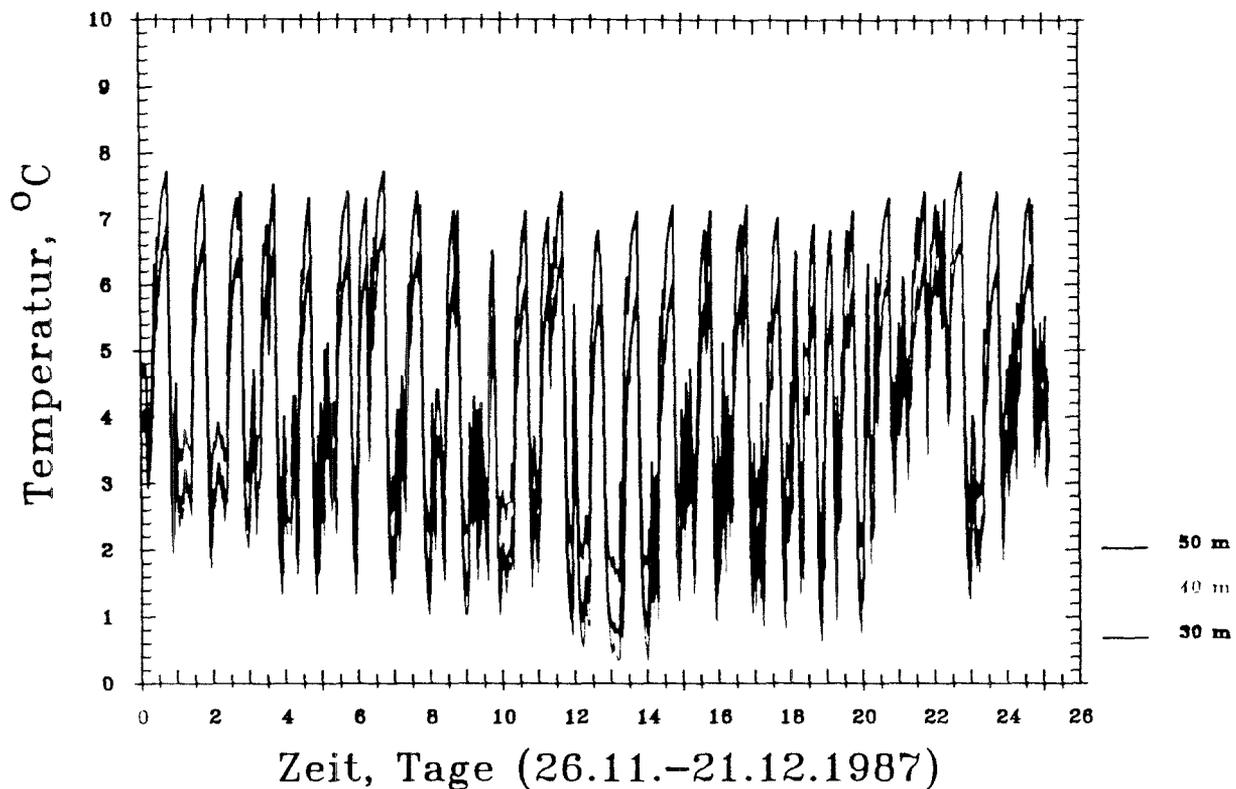


Abb. 32: Gemessener Temperaturverlauf in der Bohrung B1, November / Dezember 1987

Basierend auf der Beobachtung, daß die durchschnittliche Betriebszeit der Wärmepumpe etwa 2000 Stunden im Jahr beträgt und sich dabei die Temperaturen in den Entzugsbohrungen auf etwa 1°C absenken, wurden für beide Simulationen die nachfolgend beschriebenen Vereinfachungen getroffen.

Vom 1.10. - 31.12. (92 Tage) sollte die Anlage täglich 2 mal 4 Stunden, vom 1.1. - 28.2. (59 Tage) 3 mal 4 Stunden und vom 1.3. - 30.4. (61 Tage) 2 mal 4 Stunden pro Tag in Betrieb sein. Daraus ergibt sich an insgesamt 212 Tagen eine jährliche Laufzeit von insgesamt 1932 Stunden.¹ Bei der ersten Simulation wurde die in den Bohrungen während der Betriebsphasen der Wärmepumpe herrschenden Temperaturen mit konstant 1°C fest vorgegeben, bei der zweiten wurden die Entzugsleistungen der Wärmetauscher konstant gehalten.

Diese grobe Vereinfachung ergab sich aus der Notwendigkeit, die An-

¹ In der Heizperiode 1987/88 betrug die Laufzeit der Wärmepumpe 1920 Stunden, in denen die Anlage dem Erdreich — überschlägig abgeschätzt — insgesamt 17.4 MWh Wärmeenergie entzog. Der saisonale Leistungskoeffizient konnte mit 2.53 beziffert werden.

zahl der Zeitschritte wegen des enormen Rechenaufwandes zu limitieren. Die Zeitschrittlänge betrug in beiden Läufen über den gesamten Simulationszeitraum von jeweils 7 Jahren konstant 4 Stunden (15330 Zeitschritte). Trotz alledem benötigte der eingesetzte Großrechner *CDC 860* 24 bzw. 28 CPU-Stunden für die Lösung des gestellten Probleme.

6.2.3 Wärmentzug mit konstanter Bohrlochtemperatur

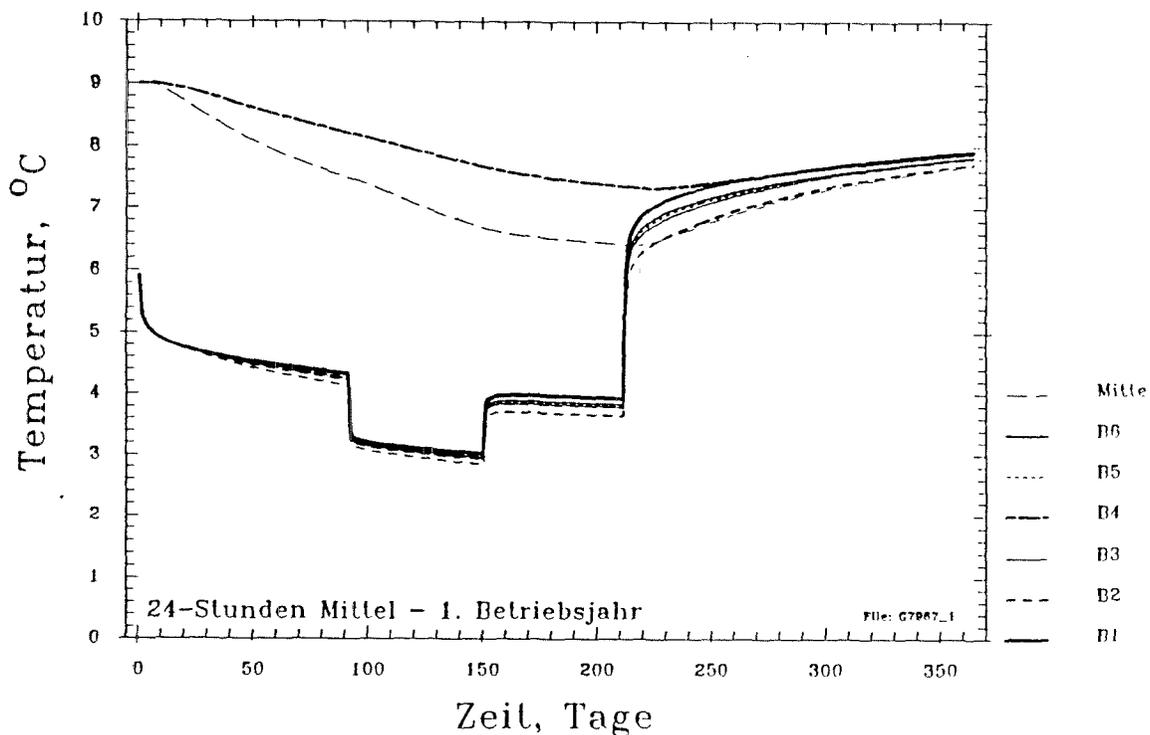


Abb. 33: Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 1. Betriebsjahr

Die in den Abb. 33 - 36 dargestellten Temperaturverläufe lassen eine asymptotische Abnahme der verbleibenden Temperaturabsenkung erkennen, die sich im Mittel durch die folgende Beziehung recht gut reproduzieren läßt:

$$T = -1.804 \cdot (1 - e^{-0.385 \cdot a}) + 8.378 \quad (6.00)$$

wobei:

T mittlere Temperatur in den Entzugsbohrungen zu Beginn des a -ten Betriebsjahres unter der Annahme einer gegebenen Bohrlochtemperatur von 1°C

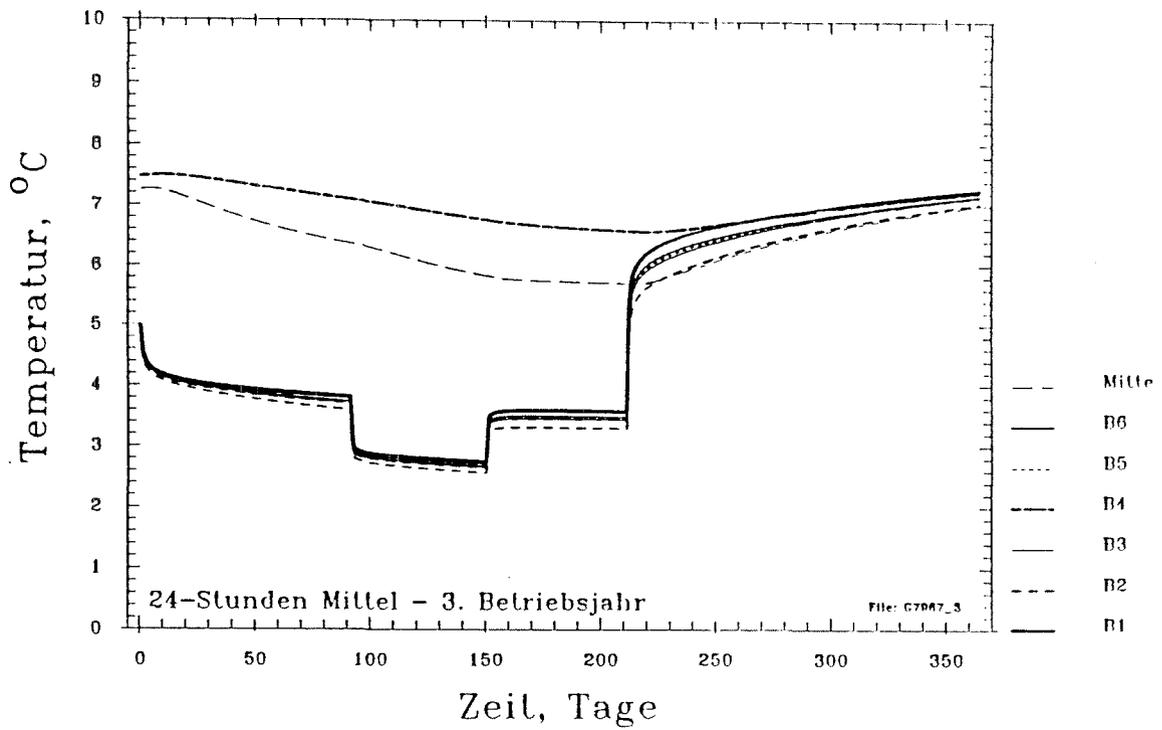


Abb. 34: Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 3. Betriebsjahr

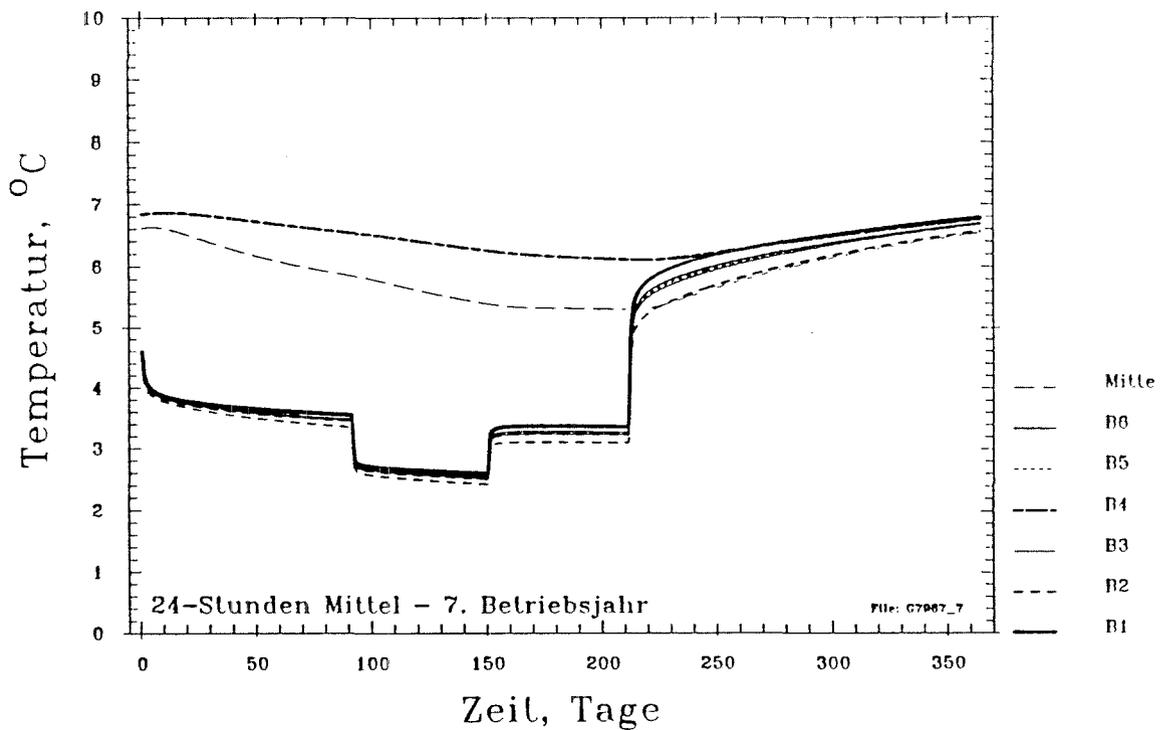


Abb. 35: Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 7. Betriebsjahr

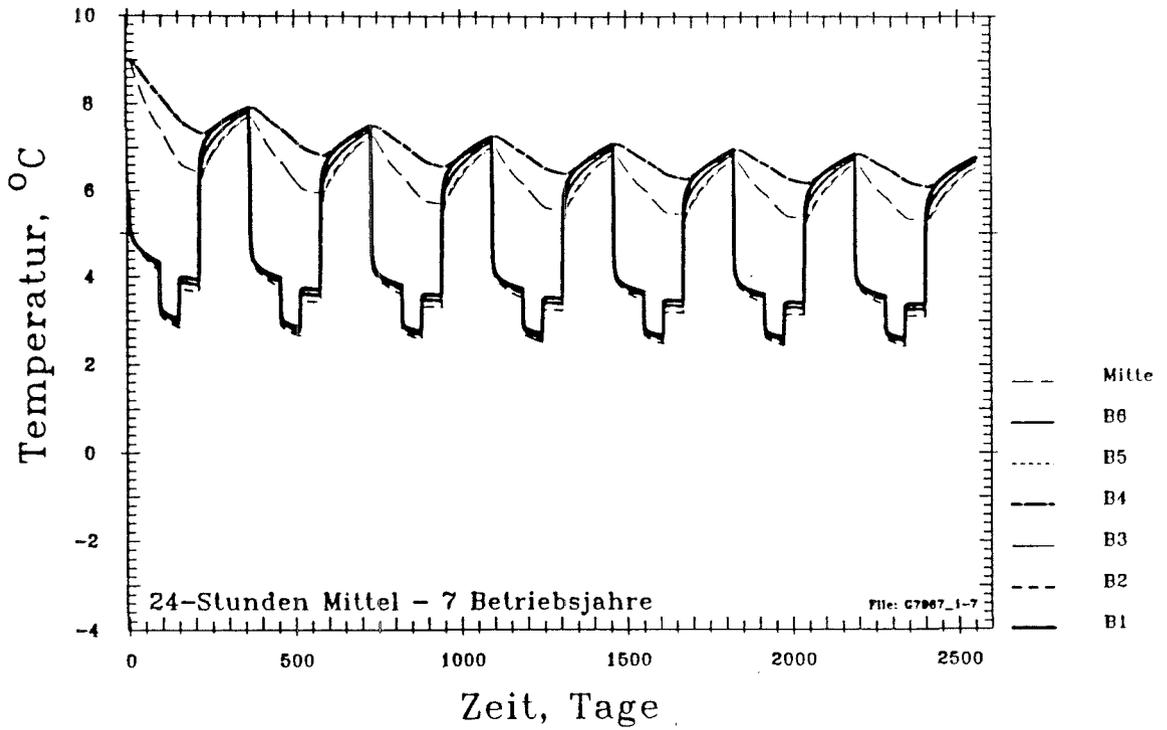


Abb. 36: Geglätteter Temperaturverlauf in den Bohrungen B1 - B6 über 7 Betriebsjahre

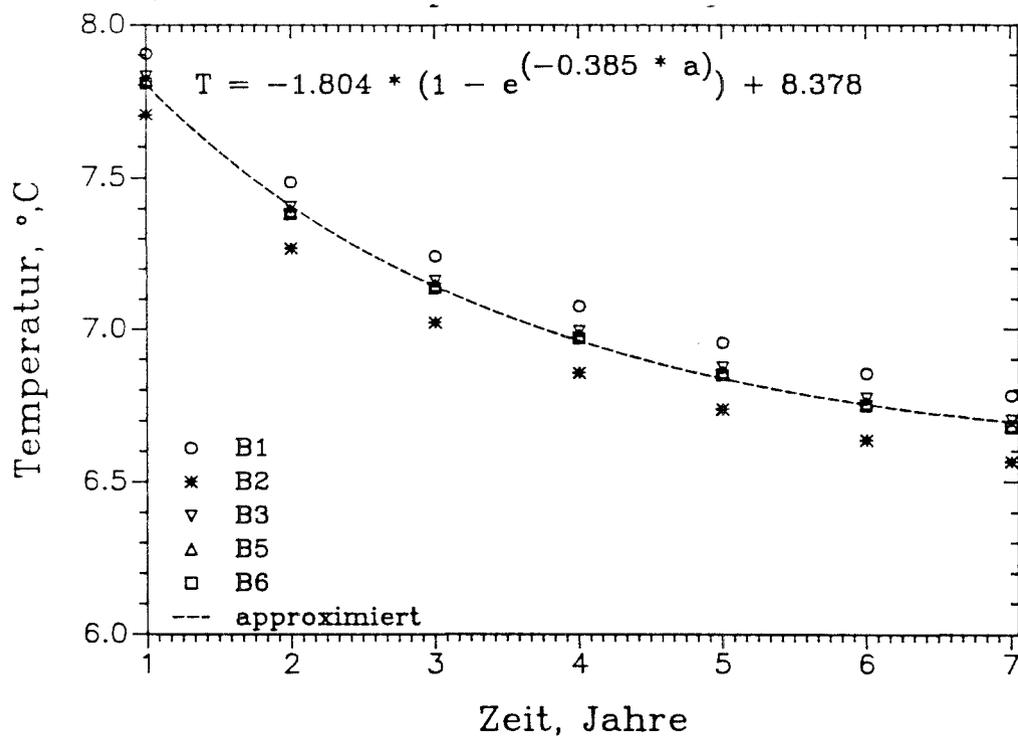


Abb. 37: Numerische Approximation der verbleibenden Temperaturabsenkung bei vorgegebener Bohrlochtemperatur

Demnach würde sich bei einer längerfristigen Prognose eine verbleibende Temperaturabsenkung auf etwa $6.5\text{ }^{\circ}\text{C}$ — -2.5 K von der Ausgangstemperatur von $9.0\text{ }^{\circ}\text{C}$ gerechnet — einstellen. Bei dieser Betriebsart nimmt zwangsläufig die Entzugsleistung der Anlage analog zur verbleibenden Temperaturabsenkung ab.

6.2.4 Wärmentzug mit konstanter Entzugsleistung

Bei der zweiten Simulation wurde in den beschriebenen Zyklen eine konstante Sondenentzugsleistung von 45.8 W/m vorgegeben. Bei einer Sondenlänge von insgesamt 200 m und einer Laufzeit der Wärmepumpe von 1932 h/a ergibt somit eine Einzugsleistung von 17.7 MWh/a .

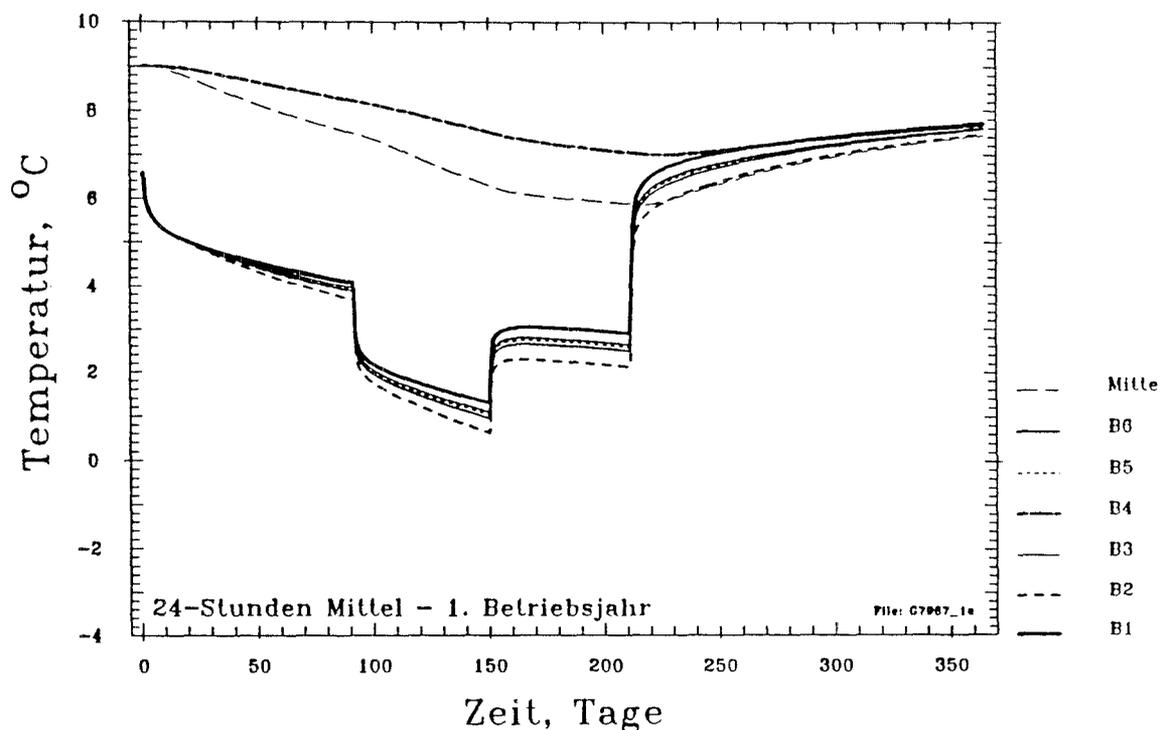


Abb. 38: Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 1. Betriebsjahr

Auch hier läßt sich für den simulierten Zeitraum eine gute Näherung für die jährlich verbleibende, mittlere Temperaturabsenkung finden:

$$T = 3.182 \cdot (e^{-0.2939 \cdot a}) + 5.205 \quad (6.01)$$

wobei:

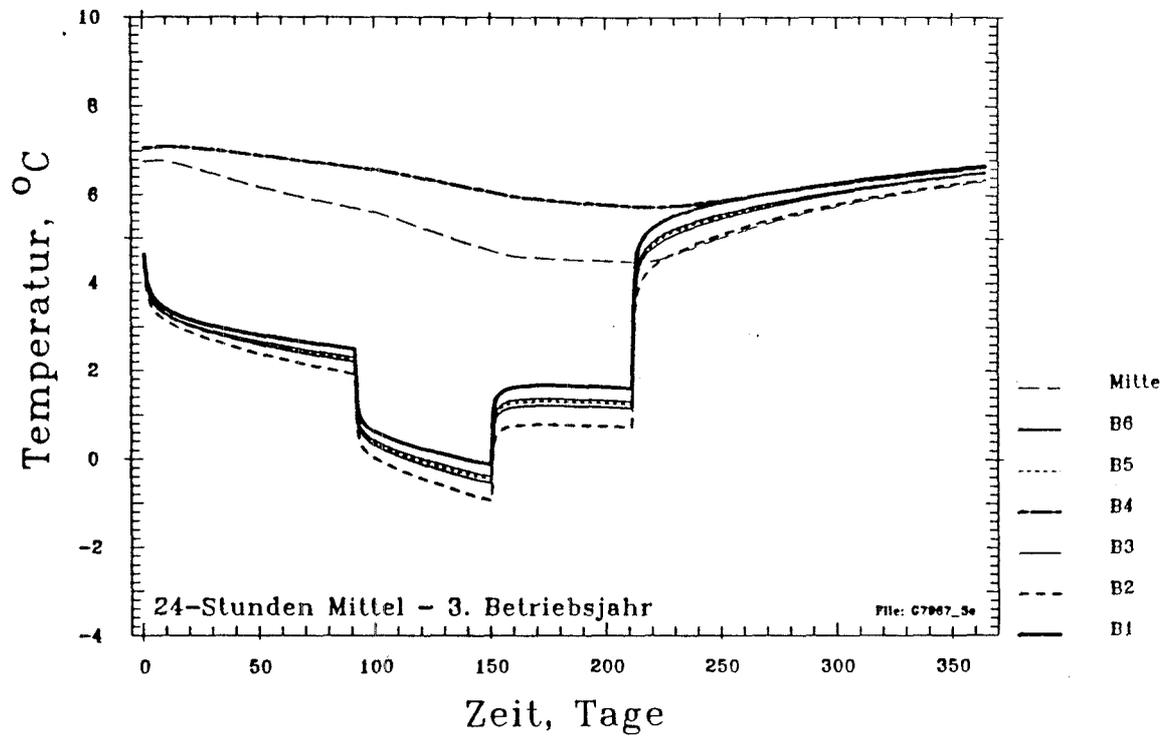


Abb. 39: Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 3. Betriebsjahr

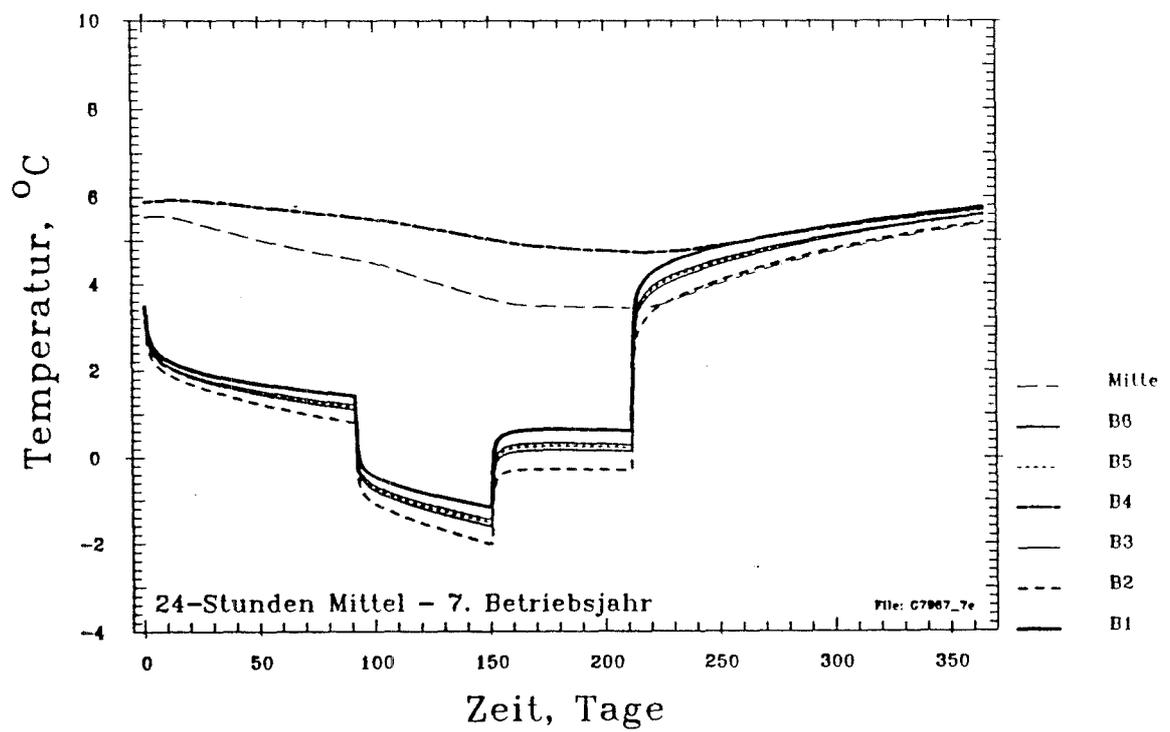


Abb. 40: Geglätteter Temperaturverlauf in den Bohrungen B1 - B6, 7. Betriebsjahr

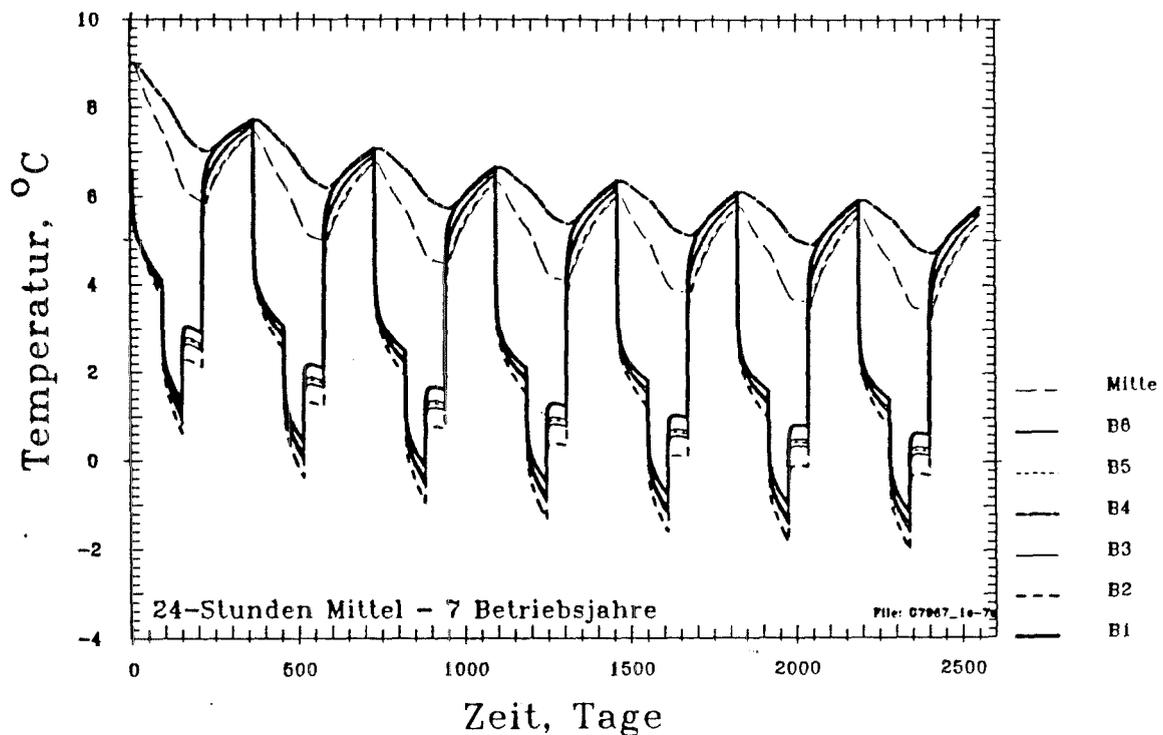


Abb. 41: Geglätteter Temperaturverlauf in den Bohrungen B1 - B6 über 7 Betriebsjahre

T mittlere Temperaturabsenkung in den Entzugsbohrungen zu Beginn des a -ten Betriebsjahres unter der Annahme einer gegebenen Entzugsleistung von 45.8 W/m

Da, wie bereits erwähnt, weder eine Möglichkeit bestand, das Modell auf den Standort zu kalibrieren, noch exakte Wärmemengenmessungen an der Anlage vorlagen, muß die Interpretation der Simulationsläufe einige Fragen offen lassen.

Ungeachtet dessen läßt sich aus den Abb. 37 und 42 zumindest ein zu erwartender Effekt deutlich erkennen: Die Temperaturen in der exponiert gelegenen Bohrung B1 regenerieren sich deutlich besser als die der übrigen Bohrungen. Am schlechtesten erfolgt die Regeneration in der zentral gelegenen Bohrung B2, während die Temperaturverläufe in den übrigen Bohrung ein sehr ähnliches Verhalten zeigen.

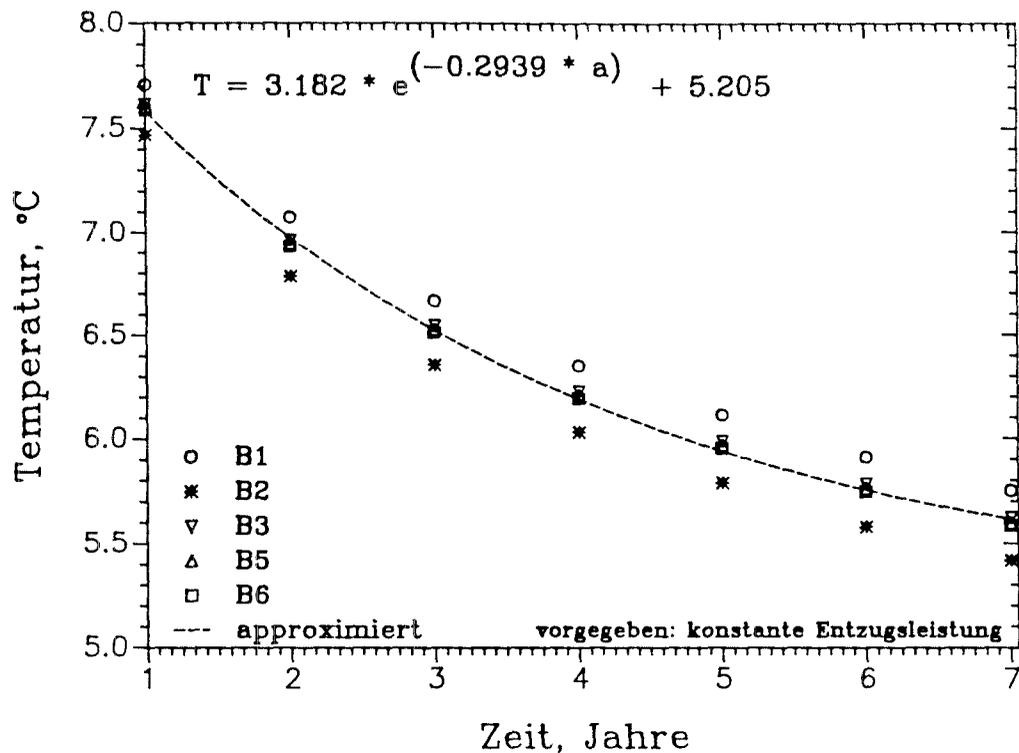


Abb. 42: Numerische Approximation der verbleibenden Temperaturabsenkung bei vorgegebener Entzugsleistung

6.3 Temperaturprognose in große Tiefen

Bei der Temperaturprognose für größere Tiefen ($> 1000 \text{ m}$) können die Druck- und Temperaturabhängigkeit der Wärmeleitfähigkeit das Ergebnis einer Berechnung stark beeinflussen. Im Anhang 1 werden verschiedene Korrekturverfahren vorgestellt, mit deren Hilfe man die im Labor gemessenen Abhängigkeiten modelltechnisch berücksichtigen kann. Zur Abschätzung der Auswirkungen der von /91/ ermittelten P/T-Abhängigkeiten von Disthen-Sillimanit-Gneisen und Amphiboliten der Kontinentalen Tiefbohrung (KTB) wurden 3 Testsimulationen durchgeführt. Dabei wurden folgende Vereinfachungen getroffen:

- $K_o = 3.0 \text{ W/mK}$
- basaler Wärmefluß 100 mW/m^2
- keine interne Wärmeproduktion
- rein konduktiver Wärmetransport

Als obere Randbedingung der eindimensionalen Simulation wurden 10 °C vorgegeben. Durch die P/T-abhängige Wärmeleitfähigkeit erhält das

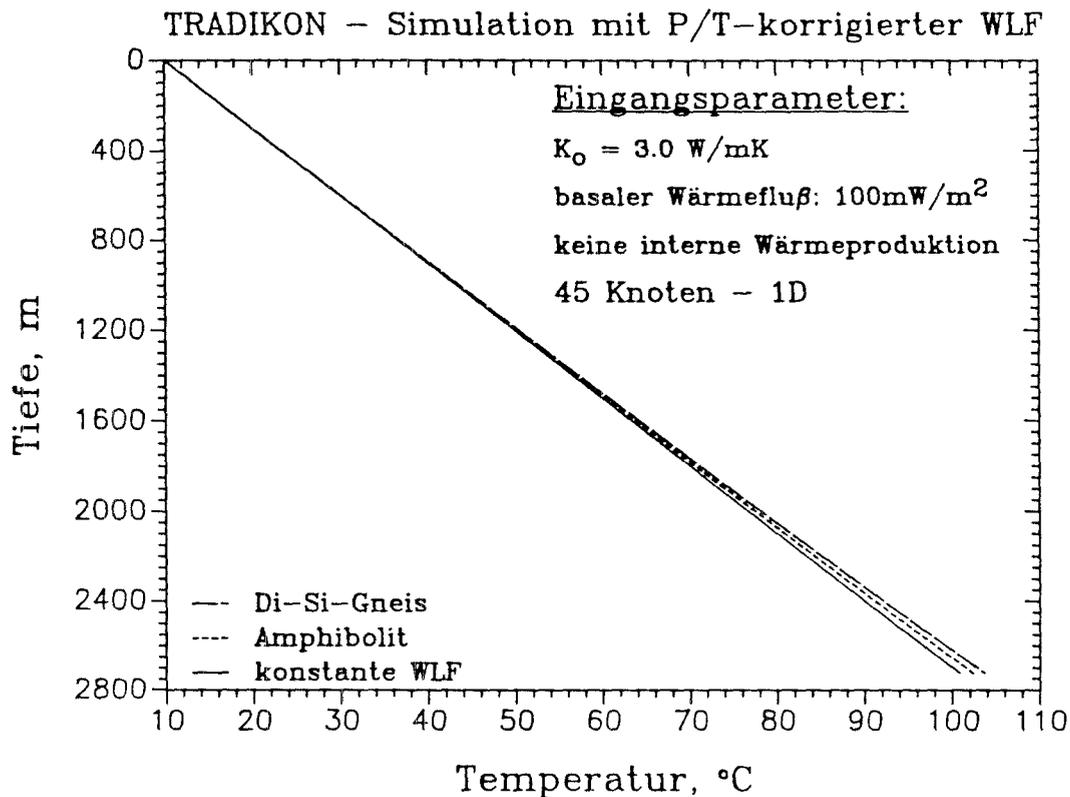


Abb. 43: Eindimensionale Temperaturberechnung mit druck- und temperaturabhängiger Wärmeleitfähigkeit

Problem einen stark nichtlinearen Charakter, sodaß die numerische Lösung nicht nur gegen die gesuchten Temperaturen, sondern auch gegen die sich einstellenden Wärmeleitfähigkeiten konvergieren muß. Die Konvergenz gegen ein Iterationskriterium von 10^{-4} K ließ sich nur durch eine kräftige Unterrelaxation ($\Omega = 0.2$) bewerkstelligen, die eine Anzahl 72 (Amphibolit) bzw. 98 (Di-Si-Gneis) Iterationsschritten nach sich zog.

Die Temperaturen in 2725 m Tiefe betragen bei der unkorrigierten Simulation $100.83 \text{ }^\circ\text{C}$, bei den korrigierten Simulationen 102.37 (Amphibolit) bzw. $103.65 \text{ }^\circ\text{C}$ (Di-Si-Gneis).

Inwieweit man die oben erwähnten Korrekturverfahren in größere Tiefen projizieren darf, ist ungewiß. Setzt man jedoch die Simulation mit denselben Eingangsparametern bis in eine Tiefe von 10 km fort, so resultieren $343.31 \text{ }^\circ\text{C}$ für die konstante WLF, 365.79 (Amph.) bzw. $380.66 \text{ }^\circ\text{C}$ (Di-Si-G.) für die P/T-korrigierten Wärmeleitfähigkeiten. Neben

einer möglichst genauen Abschätzung des basalen Wärmeflusses und der internen Wärmeproduktion sind daher vor allem auch die P/T-Abhängigkeiten der WLF für eine möglichst genaue Temperaturprognose in große Tiefen erforderlich.

7 Diskussion

Das hier vorgestellte finite Differenzen Wärmetransportmodell hat in der Validierung gegenüber analytischen Lösungen und einem anderen Strömungsmodell gezeigt, daß es unter Beachtung aller Diskretisierungsvorschriften in der Lage ist, den konvektiv-diffunden Wärmetransport sowie die Strömung in einem porösen Medium hinreichend genau zu berechnen.

Bei der Kalibrierung des Modells auf den Standort der Erdsondenforschungsanlage Schwalbach haben sich folgende Ergebnisse herausgestellt:

- der konvektive Anteil des Wärmetransports ist an dem gegebenen Standort aufgrund des geringen hydraulischen Gradientens und der niedrigen durchflußwirksamen Porosität vernachlässigbar
- der relativ simple Gefrieralgorithmus ist in der Lage, den Phasenwechsel von Grundwasser im Erdreich hinreichend genau nachzubilden
- die Regeneration der Erdreichtemperatur in der direkten Sondenumgebung wird bei den gegebenen Wärmeentzugsbedingungen nur einige Stunden nach Abschalten der Anlage durch das Schmelzen von Eis verzögert
- der instationäre Wärmetransport in der ungesättigten Bodenzone läßt sich zumindest in niederschlags- und frostfreien Perioden rein konduktiv berechnen

Zwei zweidimensionale Simulationen der Göttinger Erdsondenanlage haben folgendes ergeben:

- ungeachtet des gewählten Betriebsmodus der Anlage — vorgegebene Bohrlochtemperatur oder Entzugsleistung — wird zwischen zwei Heizperioden das Temperaturfeld nicht mehr vollständig regeneriert
- die verbleibende Temperaturabsenkung läßt über den Simulationszeitraum von jeweils 7 Jahren durch e -Funktionen berechnen und innerhalb gewisser Schranken auch in die Zukunft extrapolieren
- unter dem Vorbehalt, daß keine Möglichkeit bestand das Modell auf den Standort zu kalibrieren, scheint die Anlage

bei einer vorgegebenen Entzugsleistung von 17.8 *MWh/a* mit 5 Bohrungen etwas unterdimensioniert zu sein — sowohl was die Abstände der Bohrungen untereinander als auch die installierte Sondenlänge betrifft

- letzteres wird dadurch bestätigt, daß nach Beendigung des Versuchsbetriebes der Anlage durch das Einbeziehen der Bohrung *B4* eine spürbare Verbesserung der Leistungszahl eintrat
- ein Hinweis auf den nach den Ergebnissen der Simulation etwas zu dicht gewählten Bohrlochabstand ist die Tatsache, daß am Ende einer Heizperiode die tiefsten Temperaturen im Mittelpunkt des Bohrfeldes angetroffen werden, (in den gezeigten Diagrammen als "Mitte" bezeichnet).

Die Temperaturprognose in größere Tiefen ergab, daß neben einer möglichst genauen Kenntnis der internen Wärmeproduktionsraten und des basalen Wärmeflusses auch eine möglichst gute Abschätzung der Temperaturabhängigkeit der Wärmeleitfähigkeit über das Zutreffen einer Simulation entscheiden kann.

- /001/ ABRAMOWITZ, M. & STEGUN, I. (1964): Handbook of Mathematical Functions.- Applied Mathematics Series, Vol. 55, National Bureau of Standards.
- /002/ ALGAN, U. (1984): Numerical simulation of heat transfer in free surface aquifers.- Ph.D. Thesis, University of Missouri-Rolla.
- /003/ ALGREN, A. B. (1949): Ground temperatures as affected by weather conditions.- ASHVE Transactions, 55; Minneapolis.
- /004/ ANDERSON, J., THUNVIK, R. (1986): Predicting Mass Transport in Discrete Fracture Networks with the Aid of Geometrical Field Data.- Water Resources Research, Vol. 22, No. 13, pp. 1941-1950.
- /005/ ANDERSON, O. & WIBERG, N. (1983): Computer methods for heat storage problems.- Proc. int. conf. subs. heatstorage, Swed. coun. build. res. 1+2: 481-487; Stockholm.
- /006/ ANDERSON; S. (1983): Heat storage in natural ground water basins.- Proc. int. conf. subs. heat storage, Swed. coun. build. res., 1+2: 762-767; Stockholm.
- /007/ ANGEHRN, P., HANKE, J., LYONS, T. C., RAU, W. (1985): Energetische Grundwasserbewirtschaftung im Kanton Nidwalden unter Verwendung des hydrothermischen Grundwasserströmungsmodell HYD THERM-2.- Wasser-Energie-Luft, 77: 70-74; Baden.
- /008/ ANLAUF, R., KERSEBAUM, K. C., PING, L. Y., NUSKE-SCHÜLER, A., RICHTER, J., SPRINGOB, G., SYRING, K. M., UTERMANN, J. (1988): Modelle für Prozesse im Boden - Programme und Übungen.- Enke-Verlag; Stuttgart.
- /009/ ATHUR, J., MEIXEL, G. D., SHEN, L. S. (1983): Application of numerical methods for predicting energy transport in earth contact systems.- Applied energy, 13: 121-156.
- /010/ BACHMAT, Y., et al.(1985): Groundwater management: the use of numerical models.- 2nd ed., Water Resource Monograph, American Geophysical Union, Washington.
- /011/ BALADI, J. Y., SCHOENHALS, R. J., AYERS, D. L. (1979): Transient heat and mass transfer in soils.- ASME, 78-HT-31; New York (10017).
- /012/ BALKE, K. D. & WERNER, D. (1975): Der Abkühlungsvorgang bei anthropogen verursachten, positiven Temperaturanomalien im oberflächennahen Grundwasser.- Z. dt. geol. Ges., 126: 385-395; Hannover.
- /013/ BALKE, K. D., (1977), Das Grundwasser als Energieträger.- Brennstoff-Wärme-Kraft, 29; 5: 191-194; Düsseldorf.
- /014/ BALKE, K. D. (1979): Die Abkühlung des Untergrundes beim Betrieb von Grundwasser-Wärmepumpen.- Elektrowärme International, 7: 243-249; Essen.

- /015/ BALKE, K. D., (1981), Möglichkeiten und Grenzen des Einsatzes von Grundwasser-Wärmepumpen und Erdwärmesonden.- Z. dt. geol. Ges., 132: 691-713; Hannover.
- /016/ BALL, A., FISCHER, R., TALBERT, S. (1983): Stat-of-the-Art survey of existing knowledge for the design of ground-source heat pumps.- Ornl/Sub 80-7800/2&06, Battelle Columbus Laboratories; Frankfurt.
- /017/ BATTERMANN, G. (1981): Feldexperimente zur Ausbreitung von Warm- und Kaltwasser im Grundwasserleiter und ihre modellmäßige Erfassung.- Z. dt. geol. Ges., 132: 849-857, 3 Abb.; Hannover.
- /018/ BAU, H. H. (1984): Convective heat losses from a pipe buried in a semi-infinite porous medium.- Int. J. Heat Mass Transfer, Vol.27, No. 11: 2047-2056; Pergamon Press; London.
- /019/ BEAR, J. (1972): Dynamics of fluid in porous media.- American Elsevier; New York.
- /020/ BEAR, J. (1979): Hydraulics of groundwater.- McGraw-Hill Inc.
- /021/ BENNET, J., CLAEISSON, J., HELLSTRÖM, G. (1987): Multipole Method to Compute the Conductive Heat Flows to and between Pipes in a Composite Cylinder.- Notes on Heat Transfer 3-87, Dep. of Building Technology and Mathematical Physics Lund.
- /022/ BJORDAMMAN, J., COATS, K. H. (1969): Comparison of alternating-direction and successive over-relaxation techniques in simulation of reservoir fluid flow.-J. Soc. Pet. Eng. 9, pp. 47-58.
- /023/ BLAU, R. v. (1981): Der Einsatz von Grundwasserwärmepumpen in der Schweiz.- Z. dt. geol. Ges. 132: 727-721; Hannover.
- /024/ BODVARSSON, G. S. (1982): Mathematical modeling of the behavior of geothermal systems under exploitation.- Ph. D. thesis, University of California, Berkeley.
- /025/ BODVARSSON, G. S. (1983): Temperature flow statistics and thermodynamics of low-temperature geothermal systems in Iceland.- J. Volcan. Geotherm. Res., 9: 255-280; Amsterdam.
- /026/ BOTHA, J. F., PINDER, G. F. (1983): Fundamental Concepts in the Numerical Solution of Differential Equations.- Wiley & Sons; New York.
- /027/ BOUMA, J., KOPPENOL, A. (1983): Investigation into a complete earth-to-water heat pump system in a single family dwelling focussing on the application of a vertical vertical subsoil heat exchanger.- EUR 8077 BF 1000, Brüssel (CEC).
- /028/ BOURDET, D., GRINGARTEN, A. C. (1980): Determination of fissure volume and block size in fractured reservoirs by type-curve analysis.- SPE 9239, 15 S.; Hannover.

- /029/ BRAM, K. (1979): Heat flow measurements in the FRG.- in: CERMAK, V., RYBACH, L. ed. (1979): Terrestrial Heat Flow in Europe.- Springer, 191-196; Berlin.
- /030/ BREBBIA, C., GRAY, W., PINDER, G. F. (1978): Finite elements in water resources.- Proc. 2. Int. conf. FE in wat. res. Pentech Press; London.
- /031/ BREHM, D., KNOBLICH, K., SANNER, B. (1986): Schrifttum zum Thema Erdwärmepumpen und Erdwärmespeicher.- Z. angew. Geowiss., No. 7, S. 129-142, Gießen.
- /032/ BRISTOW, Q., CONAWAY, J. G. (1984): Temperature gradient measurements in boreholes using low noise high resolution digital techniques.- Current res. Part B, Geol. Surv. od Canada, Paper 84-1B: 101-108; Canada.
- /033/ BUETTNER, K. (1955): Evaluation of soil heat conductivity with cylindrical test bodies.- Trans. Amer. Geophys. Union, 36/5: 831-837.
- /034/ BUNTEBARTH, G. & RUEFF, P. (1988): Laboratory thermal conductivities applied to crustal conditions.- in HANZA, V. M. et al. (eds.) Geothermics and geothermal energy; Sao Paulo.
- /035/ BUNTEBARTH, G. (1980): Geothermie.- Springer, 156 S; Berlin.
- /036/ BUSCHECK, T. A., DOUGHTY, C., TSANG, C. F. (1983): Prediction and Analysis of a Field Experiment on a Multilayered Aquifer Thermal Energy Storage System With Strong Buoncy Flow.- Water Resources Research, Vol. 19, No. 5, pp. 1307-1315.
- /037/ BÜTOW, E. (1981): Modellrechnung zur Ermittlung der Bedingungen für eine Wärmespeicherung in tiefen Grundwasserleitern.- Z. dt. geol. Ges., 132: 839-847, 2 Abb.; Hannover.
- /038/ CARSLAW, H. S., JAEGER, J. C. (1959): Conduction of Heat in Solids.- 2.ed., Oxford University Press.
- /039/ CERMAK, V., RYBACH, L. (1979): Terrestrial heat flow in Europe.- 151 Abb.; Springer-Verlag, Berlin/Heidelberg/New-York.
- /040/ CHARBENEAU, R. J., STREET, R. L. (1979): Modeling Groundwater Flow Fields Containing Point Singularities: Streamlines, Travel Times, and Breakthrough Curves.- Water Resources Research, Vol. 15, No. 6, pp. 1445-1450.
- /041/ CHENG, P., (1985): Geothermal Heat Transfer.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Applications.- Chapter 11, 1-53, McGraw-Hill Book Company; New York.
- /042/ CHRISTOPHER, P. & MILLY, D. (1982): Moisture and Heat Transport in Hysteretic Inhomogeneous Porous Media: A Matric Head-Based Formulation and a Numerical Model.- Water Resources Research, Vol. 18, No. 3: 489-498.

- /043/ CLAESSON, J., ESKILSON, P: Thermal Analysis of Heat Extraction Boreholes.- Inst. of Technology, 222-227; Lund.
- /044/ CLAESSON, J., DUNAND, A. (1983): Heat extraction from ground by horizontal pipes - A mathematical analysis.- Swed. coun. build. res.; D1:1983; Stockholm.
- /045/ CLARK, D. (1987): Microcomputer programs for groundwater studies.- Elsevier, Amsterdam.
- /046/ CLARK, J. A., NABOZNY, R. L., HEETDERKS, J. L. (1977): ROCKBED: A computer program for thermal storage.- Annual meeting of int. Solar Energy Society, 17/17-17/21.
- /047/ CLARK, P. A. (1985): Thermal Energy Storage.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Applications.- Chapter 8, 1-40, McGraw-Hill Book Company; New York.
- /048/ COATS, K. H. (1977): Geothermal resources modelling.- Paper SPE 6892 52. Tech. conf. of soc. pet. engineers; Denver.
- /049/ CODELL, R. B., KEY, K. T., WHELAN, G. (1982): A Collection of Mathematical Models of Dispersion in Surface Water and Groundwater.- NUREG-0868, U. S. Nuclear Regulatory Commission, Washington.
- /050/ COLLINS, M. W. (1978): Heat transfer by laminar combined convection in a vertical tube-prediction for water.- Heat transfer 1978 v. general papers, National Research Council Canada, 25-29; Ottawa.
- /051/ COMBARNOUS, M. A., BORIES, S. A. (1975): Hydrothermal convection in saturated porous media.- Advances in Hydroscience., Vol. 10, pp 231-307.
- /052/ COURANT, R., ISAACSON, E., REES, M. (1952): On the solution of non-linear hyperbolic differential equations by finite differences.- Comm. Pure Appl. Math., Vol. 5, p. 243.
- /053/ CRANK, J., NICOLSON, P. (1947): A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type.- Proc. Cambridge Philosophical Society, Vol. 43, Cambridge at the University Press.
- /054/ DELISLE, M. A. G. (1977): Modellrechnungen zur Speicherung von Abwärme im flachen Untergrund.- VDI-Berichte, 288: 55-63; Düsseldorf.
- /055/ DINULESCU, H. A., ECKERT, E. R. G. (1979): Analysis of the one dimensional moisture migration caused by temperature gradients in a porous medium.- Int. J. Heat Mass Transfer, Vol. 23, 1069-1078; Pergamon Press, London.
- /056/ DONEA, J., GIULIANI, S. QUARTAPELLE, L. (1980): Finite element solution to transient convective-conductive heat transfer problems.- Nuclear Eng. Design, 61: 131-141; North Holland Publ. Comp.

- /057/ DOUGLAS, J., RACHFORD, H. H. (1956): On the Numerical Solution of Heat Conduction Problems in Two and Three Space Variables.- Trans. Amer. Math. Soc. 82, 421-439
- /058/ DOUGLAS, J. (1962): Alternating Direction Methods for Three Space Variables.- Numerische Mathematik.
- /059/ EDWARDS, A. L. (1972): TRUMP - A Computer program for transient and steady state temperature distributions in multidimensional systems.- UCRL-24754, Lawrence Livermore Laboratories; Livermore.
- /060/ ESKILSON, P. (1986): Superposition Borehole Model.- Department of Mathematical Physics University of Lund, Box 118, S-221 00, Lund (Schweden).
- /061/ ESKILSON, P. (1987): Thermal Analysis of Heat Extraction Boreholes.- Department of Mathematical Physics University of Lund, Box 118, S-221 00, Lund (Schweden).
- /062/ ESKILSON, P. (1987): PC-programs for Dimensioning of Heat Extraction Boreholes.- Notes on Heat Transfer No. 8.
- /063/ ESKILSON, P. (1988): Simulation model for thermally interacting heat extraction boreholes.- Numerical Heat Transfer, vol. 13, pp. 149-165.
- /064/ FAROUKI, O. T. (1986): Thermal Properties of Soils.- Series on Rock and Soil Mechanics Vol. 11; Trans Tech Publications.
- /065/ FAUST, C. R., MERCER, J. W. (1977): Finite-Difference Model of two-dimensional, single-, and two-phase heat transport in porous medium - version I.- U.S. Geological Survey Open-File Report 77-234.
- /066/ FISCHER, R. (1983): Models of simultaneous heat and moisture transfer in soils.- ORNL/SUB/80-7800/1&06, Battelle columbus laboratories Columbus (Ohio).
- /067/ FOGIEL, M. (1984): The heat transfer problem solver.- Research and Education Association, New York.
- /068/ FREEZE, R. A., CHERRY, J. A. (1979): Groundwater.- Prentice-Hall, Englewood Cliffs/N.J.
- /069/ FRIVIK, P. E., COMINI, G. (1981): Seepage and heat flow in soil freezing.- Journal of Heat Transfer, Vol. 104/323.
- /070/ GABOR, J. D., BOTTERILL, J. S. M. (1985): Heat Transfer in Fluidized and Packed Beds.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Applications.- Chapter 6, 1-47, McGraw-Hill Book Company; New York.
- /071/ GANIC, E. N., HARTNETT, J. P., ROHSENOW, W. M. (1985): Basic Concepts of Heat Transfer.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Fundamentals.- Chapter 1, 1-42, McGraw-Hill Book Company; New York.

- /072/ GARTLING, D. K., HICKOX, C. E. (1980): MARIAH - A finite element computer program for incompressible porous flow problems.- Rep. SAND79-1623, Sandia National Laboratories.
- /073/ GILBY, D. J., HOPKIRK, R. J. (1985): MC.TRAD-2D: A multiple coordinate computer code for calculation of transport by diffusion in two dimensions.- Technical Report 85-37; NAGRA Baden/Ch.
- /074/ GILLHAM, R. W., SUDICKY, E. A., CHERRY, J. A., FRIND, E. O. (1984): An Advective-Diffusion Concept for Solute Transport in Heterogeneous Unconsolidated Geological Deposits.- Water Resources Research, Vol. 20, No. 3, pp. 369-378.
- /075/ GLATZEL, W. D., HEISE, K. (1980): Wärmepumpen und Grundwasserschutz - Ökologische Auswirkungen von Wärmepumpen mit Wärmeentzug aus Wasser.- Statusseminar Wärmepumpen und Grundwasserschutz, 382; Berlin (Schmidt).
- /076/ GLAUB, J. E. TREZEK, G. (1982): Heat and mass transfer from a moist porous media of large particle size.- ASME, 8 S.; New York.
- /077/ GOLF - RACHT, T. D. van. (1982): Fundamentals of Fractured Reservoir Engineering.- Developments in Petroleum Science, 12, Elsevier Scientific Publishing Company; Amsterdam.
- /078/ GRIETHE, H. P. (1977): Beitrag zur Bestimmung der Wärmetransporteigenschaften von nichtbindigen Böden unter besonderer Berücksichtigung des teilgesättigten Zustandes.- Mitt. Inst. Wasserbau u. Wasserwirtschaft RWTH Aachen.
- /079/ GRINGARTEN, A. C. (1979): Reservoir lifetime and heat recovery in geothermal aquifers used for urban heating.- Pageoph., 117: 297-308; Birkhäuser, Basel.
- /080/ GRINGARTEN, A. C., BOURDET, A. (1980): Determination of fissure volume and block size in fractured reservoirs by type-curve analysis.- SPE 9293 Transact. AIME; Dallas (Soc. Pet. Eng.).
- /081/ GRINGARTEN, A. C., SAUTY, J. (1975): A theoretical study of heat extraction from aquifers with uniform regional flow.- Jour. geophys. Res. 35: 4956-4962; Washington.
- /082/ HEITFELD, K. H., KRAPP, L., WEILER, A. (1981): Temperaturanomalien im Stadtgebiet Duisburg und deren Auswirkung auf Baumaßnahmen der Stadtbahn.- Z. dt. geol. Ges. 132: 779-797, 22 Abb.; Hannover.
- /083/ HELLSTRÖM, G. (1982): Heat Storage in the Ground - Model of duct storage system manual for computer code.- March 1982, Department of mathematical physics, Univ. of Lund.

- /084/ HELLSTRÖM, G. (1983): Comparison between theoretical models and field experiments for ground heat systems.- Proc. int. conf. subs. heat storag, 1: 102-115; Stockholm (Swed. coun. build. res.).
- /085/ HELLSTRÖM, G. (1983): Thermal performance of ground heat exchangers. Analytic formulas.- Inst. of Technology, 239-244; Lund.
- /086/ HOPKIRK, R. J., RYBACH, L. (1986): Erdwärmesonden für Raumheizungen.- Neue Züricher Zeitung, Separatdruck, Nr. 71; Zürich.
- /087/ HORAI, K. (1971): Thermal conductivity of rock-forming minerals.- J. Geophys. Res., Vol. 76, No. 5, p. 1278-1308.
- /088/ HORAI, K., BALDRIDGE, S. (1971): Thermal conductivity of nineteen igneous rocks - Application of the needle probe method to the thermal conductivity of rocks.- Phys. Earth Planet, 5: 151-157; Amsterdam.
- /089/ HÖTZL, H., MAKURAT, A. (1981): Veränderungen der Grundwassertemperaturen unter dicht bebauten Flächen der Stadt Karlsruhe.- Z. dt. geol. Ges., 132: 767-777; Hannover.
- /090/ HOYA, K. (1981): Vorbereitende Untersuchungen zur Grundwasserwärmennutzung in der "Neuen Stadt Wulfen".- Z. dt. geol. Ges., 132: 733-743, 9 Abb.; Hannover.
- /091/ HUENGES, E., BUNTEBARTH, G., ZOTH, G. (1989): Die Wärmestromdichte in der KTB Oberpfalz VB.- Vorabdruck des Posters anlässlich des KTB Kolloquium Giessen, 15.-17.03.1989.
- /092/ HUNTOON, P. W. (1980): Computationally efficient polynomial approximations used to program the Theis equation.- Groundwater, Vol.18., No. 2, March-April.
- /093/ HUYAKORN, P. S., DOUGHERTY, D. E., FAUST, C. R. (1983): An efficient finite element model for subsurface heat storage.- Proc. int. conf. subs. heat storage, Swed. coun. bild. res., 524-529; Stockholm.
- /094/ HUYAKORN, P. S., SPRINGER, E. P. (1986): A Three-Dimensional Finite-Element Model for Simulating Water Flow in Variably Saturated Porous Media.- Water Resources Research, Vol. 22, No. 13, pp. 1790-1808.
- /095/ INCROPERA, F. P., DeWITT, D. P. (1985): Fundamentals of heat and mass transfer.- John Wiley & Sons; 2nd edition.
- /096/ INGERSOLL, L. ADLER, F., MADISON, W., PLASS, H. J., (1951): Theory of heat exchangers for the heat pump.- , ASHVE Transact., 57.
- /097/ INTERA (1979): Revision of the Documentation for a Model for Calculating Effects of Liquid Waste Disposal in Deep Saline Aquifers.- U.S. Geological Survey Water Resources Investigations 79-96.

- /098/ INTERCOMP (1976): A Model for Calculating Effects of Liquid Waste Disposal in Deep Saline Aquifes.- U.S. Geological Survey Water Resources Investigations 76-61.
- /099/ JAVANDEL, I., DOUGHTY, C., TSANG, C. F. (1984): Groundwater Transport: Handbook of Mathematical Models.- Water Resources Monograph Series 10, American Geophysical Union, Washington.
- /100/ KANGAS, M. T., LUND, P. D. (1987): The simulation of aquifer thermal energy storage systems.- in: LUND, P. ed. (1987): Nordic workshop on computational methods of seasonal storage solar heating systems NBS-workshop, June 12-13,1987
- /101/ KERSTEN, M. (1949): Thermal properties of soil.- Bulletin No. 28, Vol.L11, No 21; Minnesota.
- /102/ KILTY, K., CHAPMAN, D. S. (1980): Convection heat transfer in selected geologic situation.- Groundwater, 18, No. 4, 386-394; Salt Lake City.
- /103/ KINZELBACH, W. (1983): Analytische Lösungen der Schadstofftransportgleichung und ihre Anwendung auf Schadensfälle mit flüchtigen Chlorkohlenwasserstoffen.- In: Mitteilungen des Instituts für Wasserbau, Universität Stuttgart, Heft 54, S. 115-200.
- /104/ KINZELBACH, W., HERZER, J. (1983): Anwendung der Verweilzeitmethode auf die Simulation und Beurteilung von hydraulischen Sanierungsmaßnahmen.- In: Mitteilungen des Instituts für Wasserbau, Universität Stuttgart, Heft 54, S. 201-250.
- /105/ KINZELBACH, W. (1983): Numerische Schadstofftransportmodelle und ihre Anwendung auf einen Schadensfall mit Chlorkohlenwasserstoffen.- In: Mitteilungen des Instituts für Wasserbau, Universität Stuttgart, Heft 54, S. 311-368.
- /106/ KINZELBACH, W. (1986): Groundwater Modelling - An Introduction with Programs in Basic.- Elsevier, Amsterdam.
- /107/ KINZELBACH, W. (1987): Numerische Methoden zur Modellierung des Transports von Schadstoffen im Grundwasser.- Schriftenreihe gwf Wasser - Abwasser; Oldenbourg - Verlag München.
- /108/ KLEY, W., NIESKENS, H. G. (1975): Möglichkeiten der Wärmespeicherung in einem Porengrundwasserleiter und technische Probleme bei der Rückgewinnung der Energie.- Z. dt. geol. Ges., 126: 397-409, 11 Abb.; Hannover.
- /109/ KOBUS, H. (1980): Ausbreitung von abgekühltem Wasser in Grundwasserleitern.- Statusseminar Wärmepumpen und Grundwasserschutz, 35-60; Berlin (Schmidt).
- /110/ KOBUS, H. MEHLHORN, H. (1980): Beeinflussung von Grundwassertemperaturen durch Wärmepumpen.- Gas- und Wasserfach (gwf), 121: 261-268; München.

- /111/ KONIKOW, B. L., BREDEHOEFT, J. D. (1978): Computer model of two-dimensional solute transport and dispersion in groundwater.- Techniques of Water-Resources Investigations of U.S. Geological Survey; Book 7, Chapter C2.
- /112/ KREITH F., RABL, A. (1985): Solar Energy.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Applications.- Chapter 7, 1-37, McGraw-Hill Book Company; New York.
- /113/ KROLEWSKI, H. (1981): Technische und energiepolitische Grundlagen der Heranziehung des Grundwassers für Wärmepumpen.- Z. dt. geol. Ges., 132: 681-690, 3 Abb., 3 Tab.; Hannover.
- /114/ KUIPER, L. K. (1985): Documentation of a numerical code for the simulation of variable density ground-water flow in three dimensions.- U.S. Geological Survey Water Resources Investigations Report 84-4302.
- /115/ KÜHN, P., WEGNER, L. (1984): Die Parabel von Sperenberg - Neuinterpretation der Temperaturmessungen in der Bohrung Sperenberg (1871) und Wärmeflußbestimmung.- Zeitschrift f. ang. Geologie, 30: 84-87; Berlin.
- /116/ KUHRWAHL, H. (1979): Technik der Erschließung der Wärmequelle Grundwasser.- Eta, 37/A5, 236-242, .
- /117/ KUSUDA, T., ACHENBACH, P. R. (1965): Earth temperature and thermal diffusivity at selected stations in the United States.- ASHRAE Semiannual Meeting, Jan. 25-28,1965, No. 1914.
- /118/ KUSUDA, T., (1985): Heat Transfer in Buildings.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Applications.- Chapter 9, 1-55, McGraw-Hill Book Company; New York.
- /119/ LAIBL, J. P., BREBBIA, C. A., GRAY, W., PINDER, G. ed. (1984): Finite Elements in Water Resources - Proceedings of the 5th International Conference, Burlington, Vermont, U.S.A, June 1984.- Springer Verlag Berlin.
- /120/ LANDOLT & BÖRNSTEIN (1967): Zahlenwerte und Funktionen aus Physik, Chemie, Astronomie, Geophysik und Technik.- 6. Aufl. Springer Verlag.
- /121/ LAVAL, H., GIULIANI, S., DONEA, J. (1983): Explicit finite element analysis of convective-conductive heat transfer.- 9 S.; Brüssel (CEC).
- /122/ LEWIS, F. M., VOSS, C. I., RUBIN, J. (1986): Numerical Simulation of Advective-Dispersive Multisolute Transport with Sorption Ion Exchange and Equilibrium Chemistry.- U.S. Geological Survey - Water-Resources Investigations Report 86-4022.
- /123/ LEISMANN, H. M., (1987): Berechnung von Ausbreitungsvorgängen im Grundwasser mit der Methode der Finiten Elemente.- Dissertation Universität Karlsruhe.
- /124/ LILEY, P. E. (1985): Thermophysical Properties.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Fundamentals.- Chapter 3, 1-135, McGraw-Hill Book Company; New York.

- /125/ LIPPMANN, M. J., TSANG, C. F., WITHERSPOON, P. A. (1977): Analysis of the response of geothermal reservoirs under injection and production procedures, paper SPE 6537, presented at 47th Ann. Calif. Regional Meeting of Soc. Petrol. Eng. of AIME, Bakersfield/CA
- /126/ LOTTNER, V. (1986): Die Nutzung von Erdreich und Grundwasser für saisonale Wärmespeicherung und als ganzjährige Wärmequelle für Wärmepumpenheizung.- Z. angew. Geowiss., No. 7, S. 5-20, Gießen.
- /127/ LUDEWIG, H. (1981): Zur Auswertung von Temperaturmessungen im tiefen Grundwasser.- Z. dt. geol. Ges., 132: 799-809; Hannover.
- /128/ LUIKOV, A. V. (1973): Systems of differential equations of heat and mass transfer in capillary porous bodies.- Int. J. Heat Mass Transfer Vol. 18: 1-14, Pergamon Press; London.
- /129/ LUND, P. D., ÖSTMANN, M. B. (1985): A numerical model for seasonal storage of solar heat in the ground by vertical pipes.- Solar Energy Vol. 34, No. 4/5, pp. 351-366, Pergamon Press, London.
- /130/ LUND, P. ed. (1987): Nordic workshop on computational methods of seasonal storage solar heating systems - NBS - Workshop, June 12-13, 1987, Report TKK-F-A612; Dpmt. Tech. Phys. Helsinki.
- /131/ MARSILY, G. de. (1986): Quantitative Hydrogeology.- Groundwater Hydrology for Engineers; Academic Press, Inc. Orlando.
- /132/ McDONALD, M. G., HARBAUGH, A. W. (1984): A modular three-dimensional finite-difference groundwater flow model.- U.S. Geological Survey - Open File Report 83-875.
- /133/ MEI, V. C., FISCHER, S. (1983): Vertical concentric tube ground coupled heat exchangers. ASHRAE Transact. Pt. 2A+B, DC8308, 89: 391-406.
- /134/ MEI, V. C., FISCHER, S. (1984): A theoretical and experimental analysis of vertical concentric-tube ground-coupled heat exchangers.- ORNL, Contract No. DE-ACO5-84OR21400, 68 S.; Springfield.
- /135/ MEI, V. C. (1986): Horizontal ground-coil heat exchanger theoretical and experimental analysis.- ORNL/CON-193.
- /136/ MILLHONE, J. & WILLIS, E., Ed. (1981): New Energy Conservation Technologies.- IEA, Proceedings of an International Conference Berlin; Berlin (Springer).
- /137/ MILLY, P. C. D. (1982): Moisture and Heat Transport in Hysteretic, Inhomogeneous Porous Media: A Matric Head-Based Formulation and a Numerical Model.- Water Resources Research, Vol. 18, No. 3, pp. 489-498.
- /138/ MOGENSEN, P. (1983): Fluid to duct wall heat transfer in duct system heat storages.- Proc. int. conf. subs. heat storage, 652-657; Stockholm, (Swed. coun. build. res.).

- /139/ MOROFSKY, E. (1983): Geotechnical aspects of thermal storage in aquifers.- Proc. int. conf. subs. heat storage, 708-717; Stockholm (Swed. coun. build. res.).
- /140/ MOROFSKY, E. L. (1983): Overview of canadian aquifer thermal energy storage field trials.- Proc. Int. conf. subs. heatstorage, 1+2: 221-231; Stockholm (Swed. coun. build. res.).
- /141/ MULL, R., BATTERMANN, G., BOOCHS, P. (1979): Ausbreitung von Schadstoffen im Grundwasser.- DVWK 13. Seminar.
- /142/ MULL, R. & SCHULZ, H. (1982): Eichung eines Transportmodells zur Beschreibung der Wärmeausbreitung in einem Grundwasserleiter.- Wasserwirtschaft, 72.
- /143/ NARASIMHAN, T. N., WHITHERSPOON, P. A. (1976): An integrated finite difference method for analyzing fluid flow in porous media.- Water resources research, 12, No. 1: 57-64.
- /144/ NEISS, J., WINTER, E. (1976): Analyse der instationären Wärmeleitung zwischen Wärmetauscherrohren einer Wärmepumpe und dem Erdreich.- Wärme- u. Stoffübertragung, 9: 39-48; Berlin (Springer).
- /145/ NEISS, J. (1982): Numerische Simulation des Wärme- und Feuchtetransport und der Eisbildung in Böden.- Fortschr.-Ber. VDI-Z., Reihe 3, Nr. 73
- /146/ NIELSON, D. R., GENUCHTEN, M. TH. VAN, BIGGAR, J. W. (1986): Water Flow and Solute Transport Processes in the Unsaturated Zone.- Water Resources Research, Vol. 22, No. 9, pp. 89S-108S.
- /147/ NIEVERGELD, P. G. M., BRUGH, J. H. A. M. v.d., HORST, J. F. v.d., KOPPENOL, A. D. (1980): Investigation on using the earth as a natural heat source for heat pumps.- EUR 6835 EEN, 75 S.; Brüssel (CEC).
- /148/ PATANKAR, S. V., SPALDING, D. B. (1972): A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows.- Int. J. Heat Mass Transfer, Vol. 15, pp. 1787-1806, Pergamon Press; London.
- /149/ PATANKAR, S. V., BALIGA, B. R. (1978): A New Finite-Difference Scheme for Parabolic Differential Equations.- Numer. Heat Transfer, Vol. 1, pp. 27-37.
- /150/ PATANKAR, S. V. (1979): A calculation procedure for two-dimensional elliptic situations.- Num. Heat Transfer.
- /151/ PATANKAR, S. V. (1980): Numerical heat transfer and fluid flow.- 197 S. New York (McGraw Hill).
- /152/ PEACEMAN, D. W., RACHFORD, H. H. (1955): The numerical solution of Parabolic and elliptic differential equations.- J. Soc. Ind. Appl. Math. 3, 28-41.
- /153/ PEACEMAN, D. W. (1977): Fundamentals of numerical reservoir simulation.- Elsevier Amsterdam.

- /154/ PELKA, W. (1983): Numerical method for calculating heat transport in saturated-unsaturated groundwater flow.- Proc. int. conf. subs. heat stor., 469-475; Stockholm (Swed. coun. build. res.).
- /155/ PELKA, W. (1985): Berechnung einer thermischen Anomalie in der Umgebung des Einleitungsbrunnens einer Grundwasser-Wärmepumpe.- Wasser und Boden, 4.
- /156/ PELKA, W., DANIELS, H. (1986): Die Berechnung der Wärmeausbreitung im Grundwasser.- Z. angew. Geowiss., No. 7, S. 101-128, Gießen
- /157/ PHILIP, J., VRIES de, D. (1957): Moisture movement in porous materials under temperature gradients.- Trans. Amer. Geophys. Union, Vol. 38, No. 2.
- /158/ PINDER, G. F., GRAY, W. G. (1977): Finite element simulation in surface and subsurface hydrology.- Academic Press; New York.
- /159/ POSSON, D. R., HEARNE, G. A., TRACY, J. V., FRENZEL, P. F. (1980): A computer program for simulating geohydrolic systems in three dimension.- U.S. Geological Survey - Open File Report 80-421 .
- /160/ PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A., VETTERLING, W. T. (1987): Numerical Recipes - The Art of Scientific Computing.- Cambridge University Press.
- /161/ PRICKETT, T. A., NAYMIK, T. G., LONNQUIST, C. G. (1981): A "Random-walk" Solute Transport Model for Selected Groundwater Quality Evaluations.- Illinois state water surv. Bull, Dptmt. energy and nat. resour.; Champaign.
- /162/ PRITCHETT, J. W. (1980): Geothermal reservoirs engineering computer code comparison and validation calculations using MUSHRM and CHARGR geothermal reservoir simulators.- System Science and Software SSS-R-81-4749; San Francisco.
- /163/ RAITHBY, G. D., HOLLANDS, K. G. T. (1985): Natural Convection.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Fundamentals.- Chapter 6, 1-94, McGraw-Hill Book Company; New York.
- /164/ RASMUSSEN, R. M., LEVIZZANI, V., PRUPPACHER, H. R. (1982): A numerical study of heat transfer through a fluid layer with recirculating flow between concentric and excentric spheres.- Pure and appl. geophysics, 120: 702-720; Basel.
- /165/ REISENAUER, A. E., KEY, K. T., NARASIMHAN, T. N., NELSON, R. W. (1982): TRUST: A computer program for variably saturated flow in multidimensional, deformable media.- NUREG/CR-2360, PNL-3975, RU, U.S. Nuclear Regulatory Commision; Washington.
- /166/ REED, J. E. (1985): Digital model for simulating steady-state ground-water and heat flow.- U.S. Geological Survey - Water-Resources Investigations Report 85-4248.

- /167/ REMSON, I., HORNBERGER, G. M., MOLZ, F. J. (1971): Numerical Methods in Subsurface Hydrology with an introduction to the finite element method.- Wiley Interscience, New York.
- /168/ ROBINSON, P. C. (1984): Connectivity, Flow and Transport in Network Models of Fractured Media.- Ph. D. thesis Oxford University.
- /169/ ROSENBERG, D. U. von (1969): Methods for the Numerical Solution of Partial Differential Equations.- American Elsevier; New York.
- /170/ ROSS, B., KOPLIK, C. M. (1979): A New Numerical Method for Solving the Solute Transport Equation.- Water Resources Research, Vol. 15, No. 4, pp. 949-955.
- /171/ ROUVE, G., PELKA, W. (1981): Aquifer thermal energy storage: Experimental and theoretical investigations.- Proc. Int. Conf. IEA; Springer, Berlin.
- /172/ RUNCHAL, A. K. (1972): Convergence and Accuracy of Three Finite Difference Schemes for a Two-Dimensional Conduction and Convection Problem.- Int. J. for Numerical Methods in Engineering, Vol. 4, 541-550
- /173/ RYBACH, L. (1973): Wärmeproduktionsbestimmungen an Gesteinen der Schweizer Alpen.- Beiträge zur Geologie der Schweiz, Geotechn. Serie, 51
- /174/ SANFORD, W. E. (1985): A two-constituent solute-transport model for groundwater having variable density.- U.S. Geological Survey - Water-Resources Investigations Report 85-4279.
- /175/ SANNER, B., KNOBLICH, K., BREHM, D., EINIG, M. (1986): Zur geologischen und hydrogeologischen Situation am Standort der Wärmepumpen-Versuchsanlage Schöffengrund-Schwalbach.- Z. angew. Geowiss., No. 7, S. 29-42, Gießen.
- /176/ SANNER, B., BREHM, D., KNOBLICH, K. (1986b): Erstes Betriebsjahr der Erdsonden-Forschungsanlage Schwalbach (1985/1986).- Z. angew. Geowiss., No. 7, S. 43-60, Gießen.
- /177/ SASS, J. H., STONE, C., MONROE, R. J. (1983): Thermal conductivity determination in solid rock - A comparison between a steady-state divided-bar application and a commercial transient line-source device.- Jour. volcan. and geotherm. res., 20.
- /178/ SASS, J. H., LACHENBRUCH, A. H., MONROE, R. J. (1971): Thermal conductivity of rocks from measurements on fragments and its application to heat-flow determination.- Journal of geophysical research, 76, No. 14: 3391-3401.
- /179/ SCHAETZLE, W. (1980): Thermal energy storage in aquifers.- 109 S.; N. Y. (Pergamon).
- /180/ SCHEIDEGGER, A. E. (1974): The physics of flow through porous media.- 3. ed.; University of Toronto Press, Toronto.

- /181/ SCHENK, P. F. (1981): Wärmegewinnung aus dem Grundwasser in Schleswig-Holstein.- Z. dt. geol. Ges., 132: 745-749; Hannover.
- /182/ SCHMIDT, E. (1969): Properties of Water and Steam in SI-Units.- Springer.
- /183/ SCHNEIDER, P. J. (1985): Mathematical Methods.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Fundamentals.- Chapter 2, 1-67, McGraw-Hill Book Company; New York.
- /184/ SCHNEIDER, P. J. (1985): Conduction.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Fundamentals.- Chapter 4, 1-187, McGraw-Hill Book Company; New York.
- /185/ SCHWANNER, I., HOPKIRK, R. (1982): Die vertikale Erdsonde als Energiebeschaffungssystem.- Studie EPA-Nr. 217. 184; Zürich (Bundesamt f. Energiewirtschaft).
- /186/ SCHWILLE, F., ZAUTER, S. (1981): Grundwassertemperaturmessungen im Einflußbereich von Oberflächengewässern.- Z. dt. geol. Ges., 132: 751-765, 20 Abb.; Hannover.
- /187/ SMITH, G., YAMAUCHI, T. (1950): Thermal conductivity of soils for design of heat pump installations.- Transact. Am. Soc. Heating Eng., 56: 355-370.
- /188/ SOCHELNIKOV, V. (1981): Convergence of sequential approximations in the integral equation for a heat field in an inhomogeneous medium.- Earth physics, 17: 629-630.
- /189/ SOMMARUGA, C. (1983): High and low enthalpy geothermal resources exploration: models, strategies, and reality.- EUR 8583 EN, 50 S.; Brüssel (CEC).
- /190/ SPALDING, D. B. (1979): A novel finite difference formulation for differential expressions involving both first and second derivatives.- Int. J. Numer. Methods Engr., Vol. 4, 551-559.
- /191/ SPOSITO, G. (1986): The "Physics" of Soil Water Physics.- Water Resources Research, Vol. 22, No. 9, pp. 83S-88S.
- /192/ STONE, H. L. (1968): Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations.- SIAM J. Numer. Anal., Vol. 5, No. 3.
- /193/ SUN, N.-Z., YEH, W. W.-G. (1983): A Proposed Upstream Weight Numerical Method for Simulating Pollutant Transport in Groundwater.- Water Resources Research, Vol. 19, No. 6, pp. 1489-1500.
- /194/ SVEC, O. J. (1985): R & D needs to realize potential of ground heat source systems.- Proc. 3. Int. Conf. Energy Stor.; Toronto.
- /195/ SVENSSON, G. (1983): Energy-Geology investigations.- Proc. Int. conf. subs. heat storage, 1+2: 29-35; Stockholm (Swed. coun. build. res.).

- /196/ SVENSSON, T. (1983): Swed. research on environmental impact of heat storage and absorption in earth, water and rocks.- Proc. int. conf. subsurf. heatstor, 1: 36-41; Stockholm (Swed. coun. build. res.).
- /197/ SYKES, J., LANTZ, R., PAHWA, S., WARD, D. (1982): Numerical simulation of thermal energy storage experiment conducted by Auburn University.- Ground water, 20: 569-577; Washington.
- /198/ TAYLOR, G., LUTHIN, J. (1978): A model of coupled heat and moisture transfer during soil freezing.- Can. Geotech. J., 15: 548-555.
- /199/ THEIS, C. V. (1935): The relation between the lowering of the piezometric surface and the rate and duration of discharge of a well using groundwater storage.- Trans. Am. Geophys. Union, Ann. Meet., 16th, 519-524.
- /200/ TORAK, L. J. (1982): Modifications and corrections to the finite-difference model for simulation of three-dimensional ground-water flow.- U.S. Geological Survey Water Resources Investigations 82-4025.
- /201/ TORRANCE, K. E. (1985): Numerical Methods in Heat Transfer.- in: ROHSENOW, W. M., HARTNETT, J. P., GANIC, E. N. ed. (1985): Handbook of Heat Transfer Fundamentals.- Chapter 5, 1-85, McGraw-Hill Book Company; New York.
- /202/ TOULOUKIAN, Y. S., LILEY, P. E., SAXENA, S.C. (1970): Thermophysical properties of matter.- Vol. 3 Thermal conductivity - Nonmetallic liquid and gases. IFI-Plenum Press; New York.
- /203/ TRECOTT, P. C. (1975): Documentation of Finite-Difference Model for Simulation of Three-Dimensional Ground-Water Flow.- U.S. Geological Survey Open-File Report 75-438.
- /204/ TRECOTT, P. C., LARSON, S. P. (1976): Documentation of Finite-Difference Model for Simulation of Three-Dimensional Ground-Water Flow.- Supplement to Open-File Report 75-438; U.S. Geological Survey Open-File Report 76-591.
- /205/ TROEDSSON, T. (1983): Ecological effects of soil-heat extraction in soil and vegetation.- Proc. int. conf. subs. heat storage, 685-689; Stockholm (Swed. coun. build. res.).
- /206/ TSANG, C. F. (1983): Aquifer storage simulation in theory and practice.- Proc. int. conf. subs. heat storage, 1: 116-125; Stockholm (Swed. coun. build. res.).
- /207/ VAIL, L., KINCAID, C. (1983): A simple areal flow model.- Proc. int. conf. subs. heat storage, 530-535; Stockholm (Swed. coun. bild. res.).
- /208/ VOIGT, H. HAEFNER, F. (1984): Erkundung und Dimensionierung von Anlagen zur Wärmegegewinnung aus Grund- und Tiefenwasser.- Zeitschrift f. ang. Geologie, 30: 62-68; Berlin.
- /209/ VOSS, C. I. (1984): A finite-element simulation model for saturated-unsaturated, fluid-density-dependent ground-water flow with energy transport or chemi-

cal-reactive single-species solute transport.- U.S. Geological Survey - Water-Resources Investigations

- /210/ WALSH, J. B., DECKER, E. R. (1966): Effect of Pressure and Saturating Fluid on the Thermal Conductivity of Compact Rock.- J. Geophys. Res., Vol. 71, No. 12, 3053-3061.
- /211/ WANG, H. F. Y., ANDERSON, M. P. (1982): Introduction to groundwater modeling.- Freeman and Company; San Francisco.
- /212/ WANG, H. F. Y., TSANG, C.F., STERBENTZ, R. A. (1983): The State of the Art of Numerical Modeling of Thermohydrologic Flow in Fractured Rock Masses.- Environ. Geol., 4,133 - 199.
- /213/ WERNER, D., BALKE, K. (1977): Die Wärmeausbreitung in der Umgebung eines Kühlwasser-Sickerbrunnens.- gwf-wasser, 118: 528-531; München.
- /214/ WIGLEY, T. (1977): WATSPEC: A computer program for determining the equilibrium specification of aqueous solutions.- Techn. Bull. 20; Norwich (British geomorph. res. group).
- /215/ WILLHITE, G. P., WAGNER, J., SIMONPIETRI, F., STOKER, J. (1974): Disposal of heat water through ground water system.- PB-236 302, Vol. I.: Technical and economic feasibility; Kansas Water Resources Research Institute.
- /216/ WILLHITE, G. P., WAGNER, J. (1974): Disposal of heat water through ground water system.- PB-236 303, Vol.II.: User's Manual, Numerical simulation of fluid flow and heat transfer in groundwater system.; Kansas Water Resources Research Institute.
- /217/ YAKI, S., KUNII, D. (1956): Studies on Effektive Thermal Conductivities in Packed Beds.- A. I. Ch. E. Journal Vol. 3, No. 3.; p. 373-381.

Anhang 1:

Physikalische Eigenschaften

Wasser:

Molekulargewicht: $M = 18.0153 \text{ g/mol}$

Dichte:

Die Dichte von Wasser nimmt mit steigender Temperatur ab. Die aus /120/ entnommenen Tabellenwerte lassen sich durch folgende Beziehung approximieren:

$$\rho_w = 0.9998966 + 0.4545388 \cdot 10^{-4} T - 0.7095728 \cdot 10^{-5} T^2 \\ + 0.2760997 \cdot 10^{-7} T^3 \quad [\text{g/cm}^3] \quad (\text{A.1.00})$$

Da im hier hauptsächlich betrachteten Temperaturbereich von 0–15 °C die Dichteunterschiede relativ gering sind, wurde allen Berechnungen eine konstante Dichte von 1000 kg/m³ zugrunde gelegt. Bei geothermischen Fragestellungen oder z.B. bei der Simulation einer Warmwasserinjektion in einen Aquifer ist die Berücksichtigung der Dichteunterschiede wegen des Effektes der gravitativ bedingten Konvektion zwingend erforderlich.¹

Wärmeleitfähigkeit:

Über die Temperaturabhängigkeit der Wärmeleitfähigkeit von Wasser werden in der Literatur unterschiedliche Auffassungen vertreten. Nach /202/ läßt sich die Wärmeleitfähigkeit von Wasser durch die folgende Beziehung berechnen:

$$K_w = \frac{-1390.53 + 15.1937 T - 0.0190398 T^2}{4.1868 \cdot 10^{-4}} \quad [\text{W/mK}] \quad (\text{A.1.01})$$

¹ mit dieser Thematik beschäftigen sich eine Reihe von Publikationen, z.B. /2/, /24/, /36/, /46/, /51/, /100/, /102/, /121/, /163/, /165/, /174/, /209/ und /212/

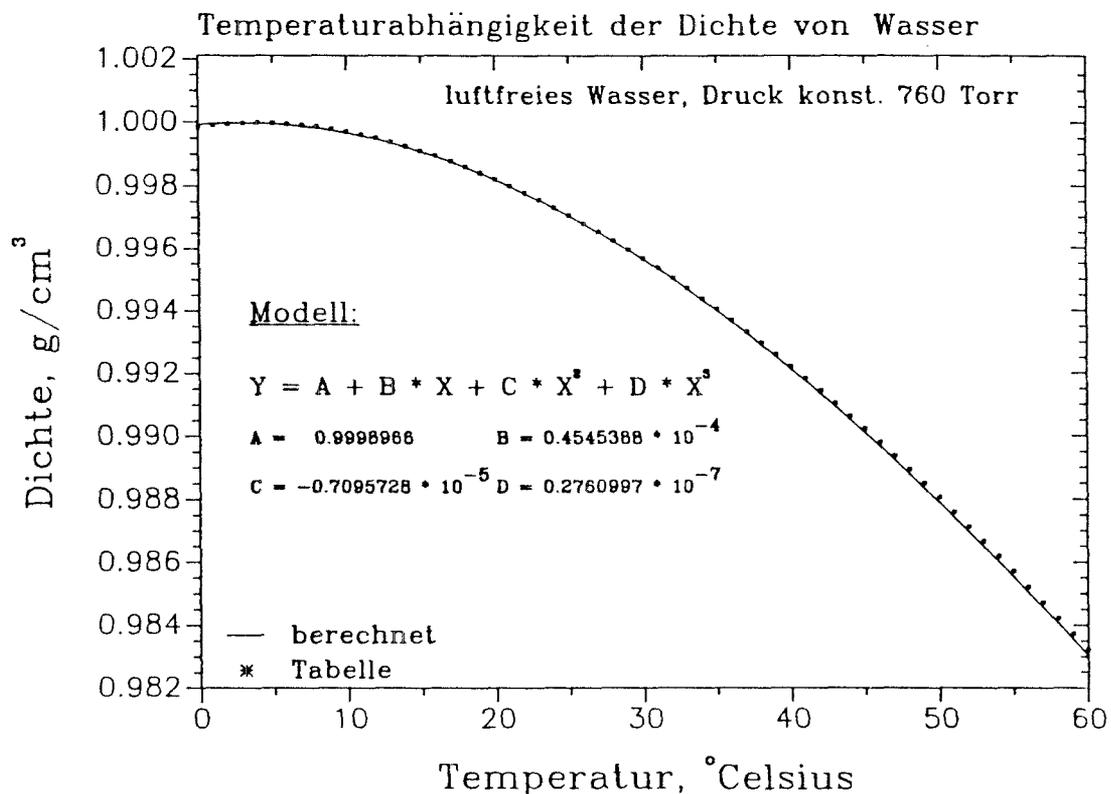


Abb. A.1.1: Temperaturabhängigkeit der Dichte von luftfreiem Wasser bei 760 Torr

wobei T = Temperatur in K , während NEISS(1982) für die Tabellenwerte von SCHMIDT(1969) die parabolische Approximation angibt:

$$K_w = 0.569 + 1.88389 \cdot 10^{-3} T - 7.7222 \cdot 10^{-6} T^2 \quad [W/mK] \quad (A.1.02)$$

wobei T = Temperatur in $^{\circ}C$.

Insgesamt sind die temperaturbedingten Änderungen der Leitfähigkeiten relativ gering, sodaß alle Berechnungen mit einer konstanten Leitfähigkeit von $0.57 W/mK$ durchgeführt wurden.

Wärmekapazität:

Im Temperaturbereich von 0 bis $20^{\circ}C$ nimmt die Wärmekapazität mit steigender Temperatur ab. Die von /120/ angegebenen Tabellenwerte lassen sich durch folgende Beziehung hinreichend genau reproduzieren:

$$C_w = 4216.9 - 3.209 T + 0.0739 T^2 \quad [J/kgK] \quad (A.1.03)$$

Die temperaturbedingte Variation der Wärmekapazität wird im Modell als vernachlässigbar angesehen, weshalb alle Berechnungsergebnisse auf einem Wert von $4200 J/kgK$ basieren.

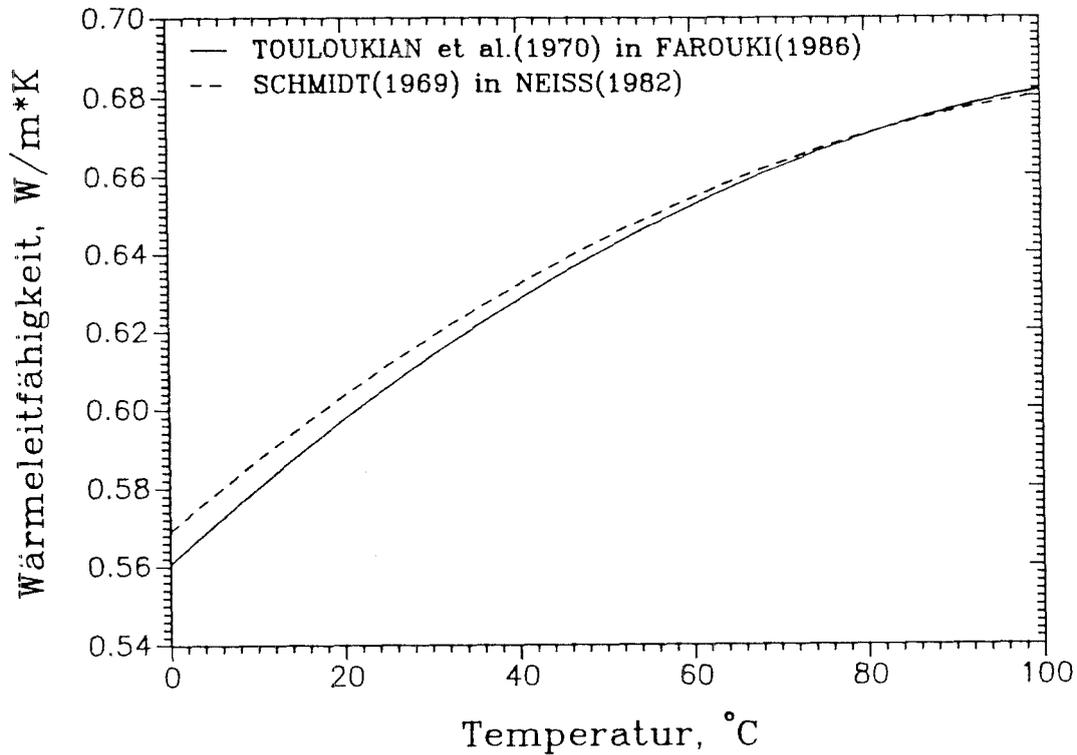


Abb. A.1.2: Temperaturabhängigkeit der Wärmeleitfähigkeit von Wasser für die Verfahren nach NEISS(1982) und TOULIKIAN(1970)

Viskosität:

Die kinematische Viskosität von Wasser ist sehr stark von der Temperatur abhängig. Die Tabellenwerte von /120/ lassen sich die folgende Beziehung recht gut reproduzieren:

$$\mu_{kin} = 0.88977 \left(1 + 1.70623 \cdot e^{-0.03406 T} \right) - 0.635 \quad [mm^2/s] \quad (A.1.04)$$

wobei T die Temperatur in °C.

Die hydraulische Leitfähigkeit eines porösen Mediums verhält proportional zur Fluidichte und umgekehrtproportional zur Viskosität des durchströmenden Fluids, sodaß die intrinsische Permeabilität κ wie folgt definiert werden kann:

$$\kappa = \frac{\mu_{dyn} k}{\rho g} \quad (A.1.05)$$

Die in Gleichung A.1.04 dargestellte Temperaturabhängigkeit kann im Modell zwar berücksichtigt werden, für die Temperaturbereiche der vorgestellten Simulationen schienen die Änderungen im Vergleich zur

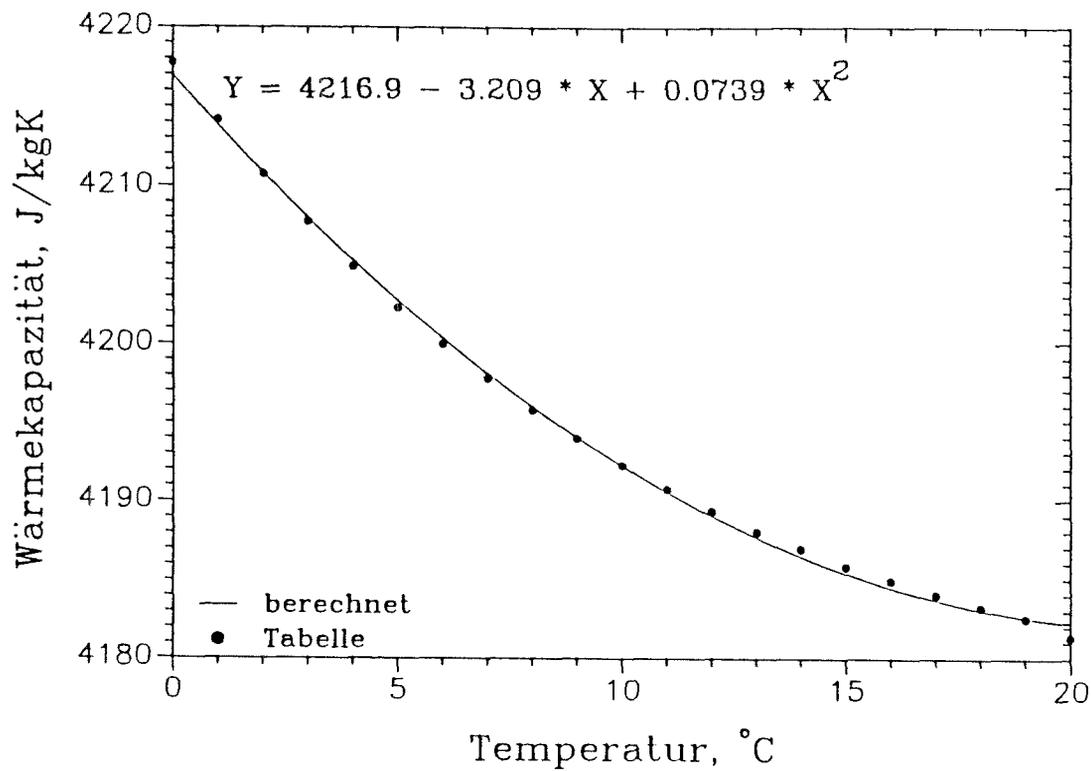


Abb. A.1.3: Temperaturabhängigkeit der Wärmekapazität von Wasser

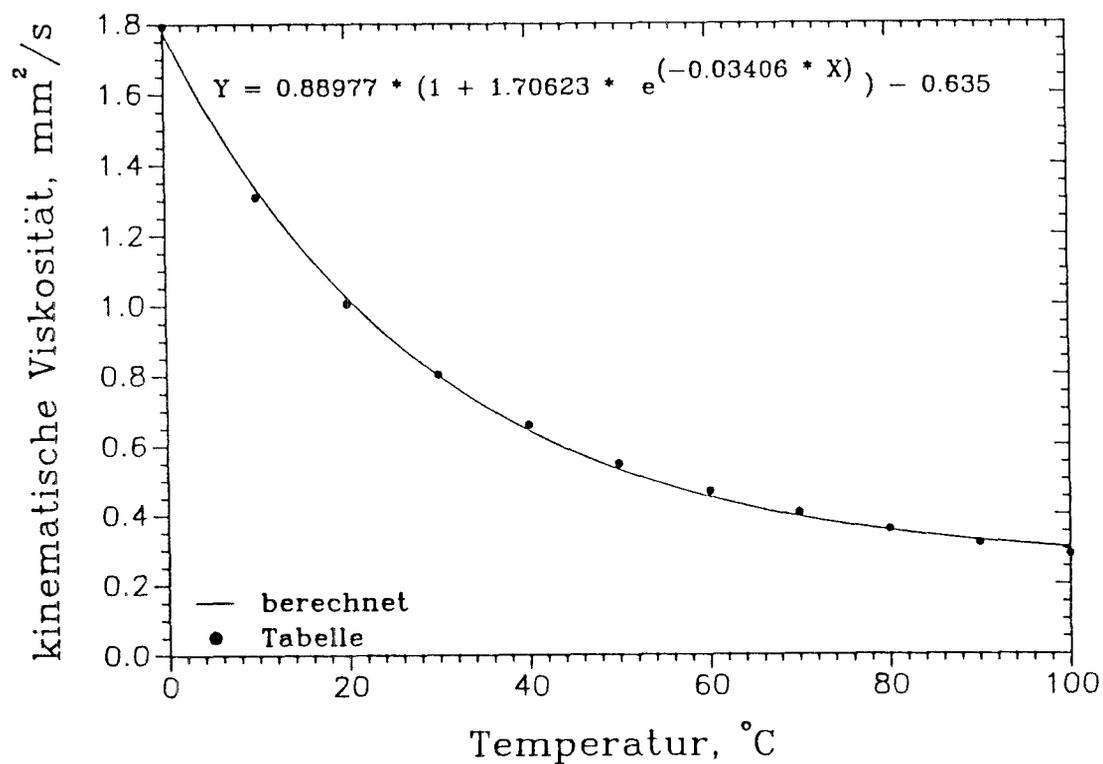


Abb. A.1.4: Temperaturabhängigkeit der kinematischen Viskosität

Genauigkeit, mit der man durch Feld- oder Laborversuche den k_f -Wert bestimmen kann, zu gering, sodaß generell von einem konstanten Durchlässigkeitsbeiwert ausgegangen wurde.

Schmelzwärme:

Die molare Schmelzwärme von Wasser beträgt etwa $6030 J/mol$, /120/, oder — unter Berücksichtigung des oben angeführten Molekulargewichtes von Wasser — etwa $3.336 \cdot 10^5 J/kg$

Eis:

Dichte:

Die Abhängigkeit der Dichte des Eises von der Temperatur wurde im Modell vernachlässigt. Ein Wert von $\rho_e = 917 kg/m^3$ erschien hinreichend genau.

Wärmekapazität:

Die Wärmekapazität von Eis läßt sich nach /145/ durch die Beziehung:

$$C_e = 2110 + 7.79 T \quad [J/kgK] \quad (A.1.06)$$

wobei $T =$ Temperatur in $^{\circ}C$. berechnen. Dieser Ansatz stimmt recht gut mit den Angaben von HODGEMAN(1955)² überein:

$$C_e = 2114 + 7.58 T \quad [J/kgK] \quad (A.1.07)$$

mit $T =$ Temperatur in $^{\circ}C$, überein. Die relativ geringen Änderungen innerhalb des in der vorliegenden Untersuchung betrachteten Temperaturbereiches rechtfertigten jedoch nicht den immensen zusätzlichen Rechenaufwand, sodaß alle Berechnungen mit einer konstanten Wärmekapazität von $2114 W/mK$ durchgeführt wurden.

Wärmeleitfähigkeit:

Die Wärmeleitfähigkeit von vielen kristallinen Nichtmetallen läßt sich durch die Beziehung

$$K_{nm} = \frac{a}{T^n} \quad [W/mK] \quad (A.1.08)$$

² HODGEMAN (1955): Handbook of Chemistry and Physics.

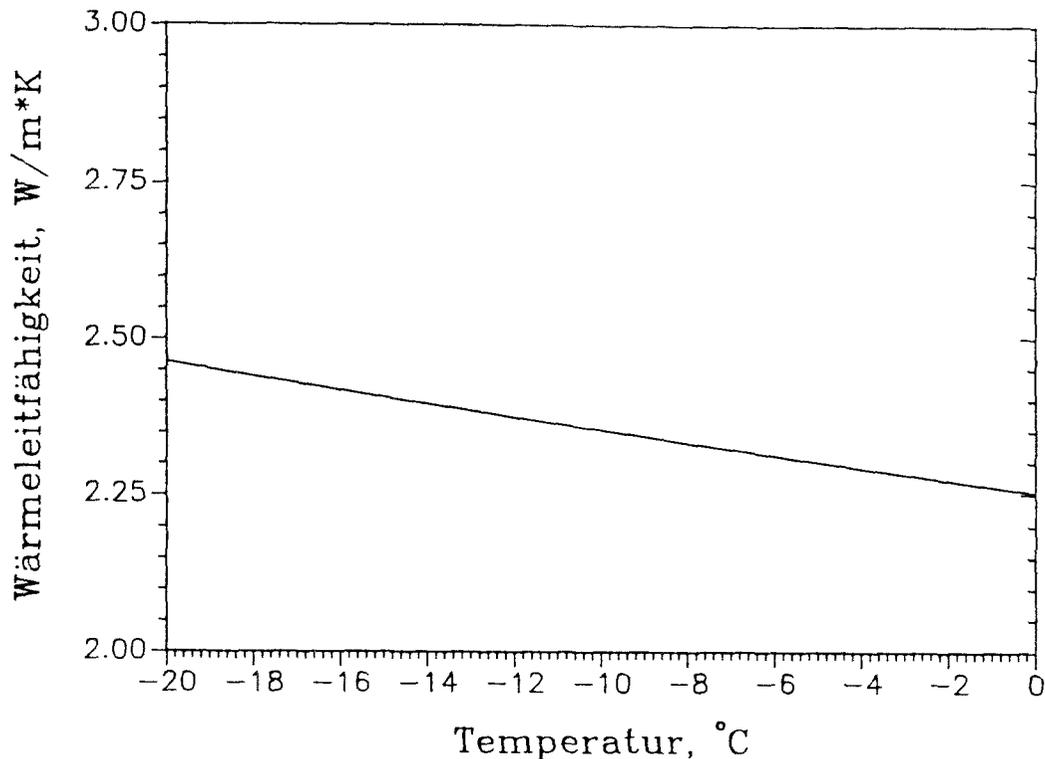


Abb. A.1.5: Abhängigkeit der Wärmeleitfähigkeit des Eises von der Temperatur, nach NEISS(1982)

wobei a und n materialspezifische Größen und T die Temperatur in K bedeuten, abschätzen. Nach /145/ läßt sich demnach für Eis a mit 1631.24 und n mit 1.174 beziffern.

CARSLAW & JAEGER (1959), /38/, geben einen Wert von 2.22 W/mK (bei 0 °C) an. Da die Temperaturabhängigkeit innerhalb des betrachteten Temperaturbereiches relativ gering ist, wurde allen Berechnungen ein Wert von 2.25 W/mK zugrundegelegt.

Gesteine:

Wärmeleitfähigkeit

Die an der Erdoberfläche vorkommenden Gesteine weisen sehr unterschiedliche Wärmeleitfähigkeiten auf. Dies hängt bei porenarmen Gesteinen vornehmlich mit den stark variierenden Quarzgehalten zusammen, da Quarz mit 7.7 W/mK , /64/, zu den am besten wärmeleitenden Mineralen gehört. Die Wärmeleitfähigkeit der Gesteine ist sowohl druck- als auch temperaturabhängig, allerdings in so geringem Maße, daß die Änderungen

Tab. A.1.2 Wärmeleitfähigkeiten wichtiger metamorpher Gesteine

<i>Gestein</i>	<i>Wärmeleitfähigkeit [W/mK]</i>	<i>Quelle</i>
Gneis	1.80-2.81	/A01/
Gneis	2.30-4.00	/A04/
Metabasite	2.40-2.60	/A04/
Glimmerschiefer	2.93	/A01/
Schiefer	2.05	/A01/
Tonschiefer	0.70-1.70	/A03/
Marmor	2.09-3.22	/A01/
Quarzit	5.74	/A01/

Tab. A.1.3 Wärmeleitfähigkeiten wichtiger Sedimentgesteine

<i>Gestein</i>	<i>Wärmeleitfähigkeit [W/mK]</i>	<i>Quelle</i>
Konglomerat	2.09	/A01/
Sandstein	1.46-4.19	/A01/
Sandstein, trocken	0.88	/A02/
schluffiger Sand	0.69	/A02/
Siltstein	0.68	/A02/
Sand, wasserfrei	0.29	/A01/
Sand, 10% Wasser	1.04	/A01/
Sand, fein, trocken	0.63	/A02/
Sand, grob, trocken	0.55	/A02/
Grauwacke, quarzreich	3.00-3.40	/A03/
Kalkstein, kompakt	2.10-3.35	/A01/
Kalkstein, porös	1.05-2.30	/A01/
Kalkstein, (-)	1.70	/A02/
Dolomit	5.00	/A01/
Mergel	0.92-2.22	/A01/
Tone	0.92-1.84	/A01/
Schieferton	0.59-2.76	/A01/
Lehm, lufttrocken	0.25	/A01/
Lehm, wassergesättigt	0.86	/A01/

/A01/

LANDOLT & BÖRNSTEIN (1967): 6. Auflage, Band III

- /A02/ FAROUQ ALI, S. M. (1970): Oil Recovery by Hot Water Flooding.- Producers Publishing Co., Inc. Bradford, PA.
- /A03/ unveröffentliche Messungen am Institut für Angewandte Geowissenschaften der Justus-Liebig-Universität Gießen
- /A04/ internes Arbeitspapier der Arge 4 des Kontinentalen Tiefbohrprogrammes: Dr. Huenges, Feldlabor KTB

Bei bestimmten Fragestellungen kann jedoch die Berücksichtigung der Temperatur und Druckabhängigkeit der Wärmeleitfähigkeit von besonderem Interesse sein. Deshalb wurde eine polynomische, eine exponentielle sowie eine reziproke Korrekturmöglichkeit vorgesehen. Das letztere Verfahren berücksichtigt — im Gegensatz zu den ersten beiden Verfahren — neben der Temperatur auch den Druck, dem ein Gestein ausgesetzt ist:

$$K_{korr} = K_o a e^{(bT)} + K_o c \quad (A.1.09)$$

$$K_{korr} = K_o + K_o \left(T \left(a + T(b + Tc) \right) \right) \quad (A.1.10)$$

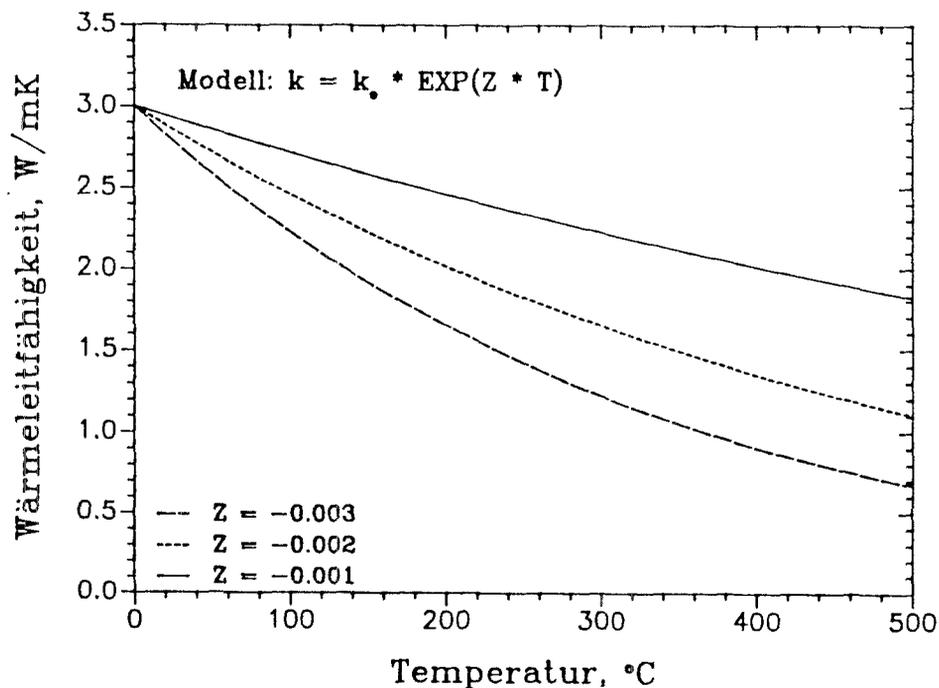


Abb. A.1.6: Temperaturkorrektur der Wärmeleitfähigkeit nach dem exponentiellen Ansatz

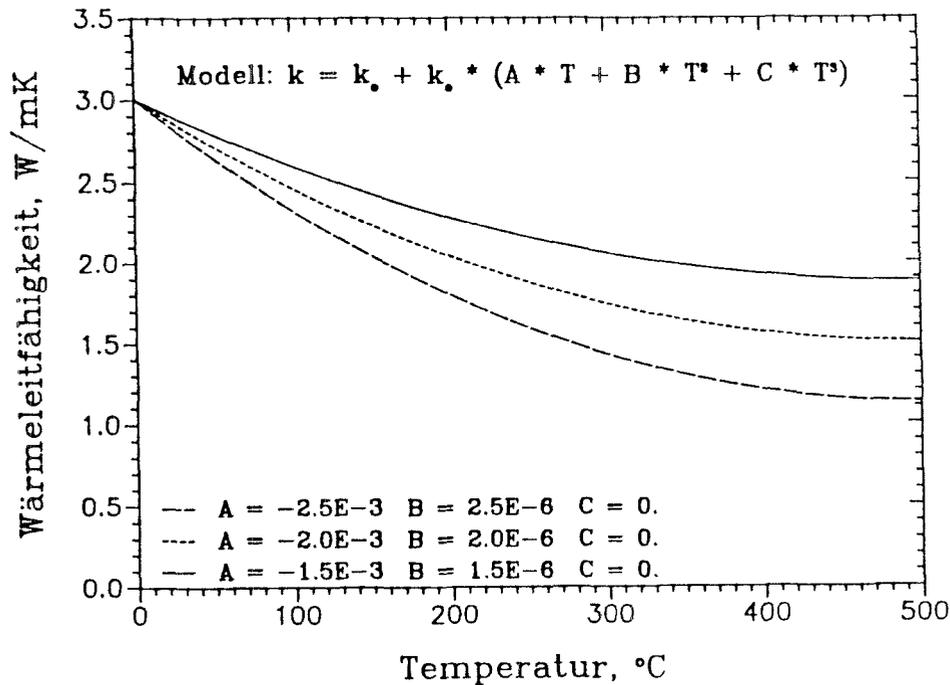


Abb. A.1.7: Temperaturkorrektur der Wärmeleitfähigkeit nach dem polynomi-
schen Ansatz

In der Literatur sind sehr wenige Daten über die Temperaturabhängigkeit dokumentiert. LANDOLT & BÖRNSTEIN (1967) publizierten die Wärmeleitfähigkeiten bei unterschiedlichen Temperaturen von einigen Tiefengesteinen. Für einen Granit lassen sich demnach die beiden Approximationen, wie in Abb. A.1.8 dargestellt, formulieren.

In größeren Tiefen erlangt die Druckabhängigkeit zunehmende Bedeutung. Für einen Temperaturbereich bis 200 °C und einen Druck bis 60 MPa (ca. 2200 m) konnten /91/ mit der bei /34/ beschriebenen Apparatur die Wärmeleitfähigkeiten von 18 Amphibolit- und 11 Disthen - Sillimanit - Gneis Proben bestimmen und für die gemessenen Werte folgende numerische Approximation ermitteln:

$$K_{korr} = \frac{K_o}{1 + K_o \cdot a \cdot (T - 20)} + 0.027 \cdot b \cdot (z - 400) \quad (A.1.11)$$

es bedeuten:

- | | |
|--------|---|
| K_o | Wärmeleitfähigkeit bei 20 °C und 10 MPa |
| z | Tiefe in m unter Geländeoberkante |
| a, b | materialspezifische Größen |

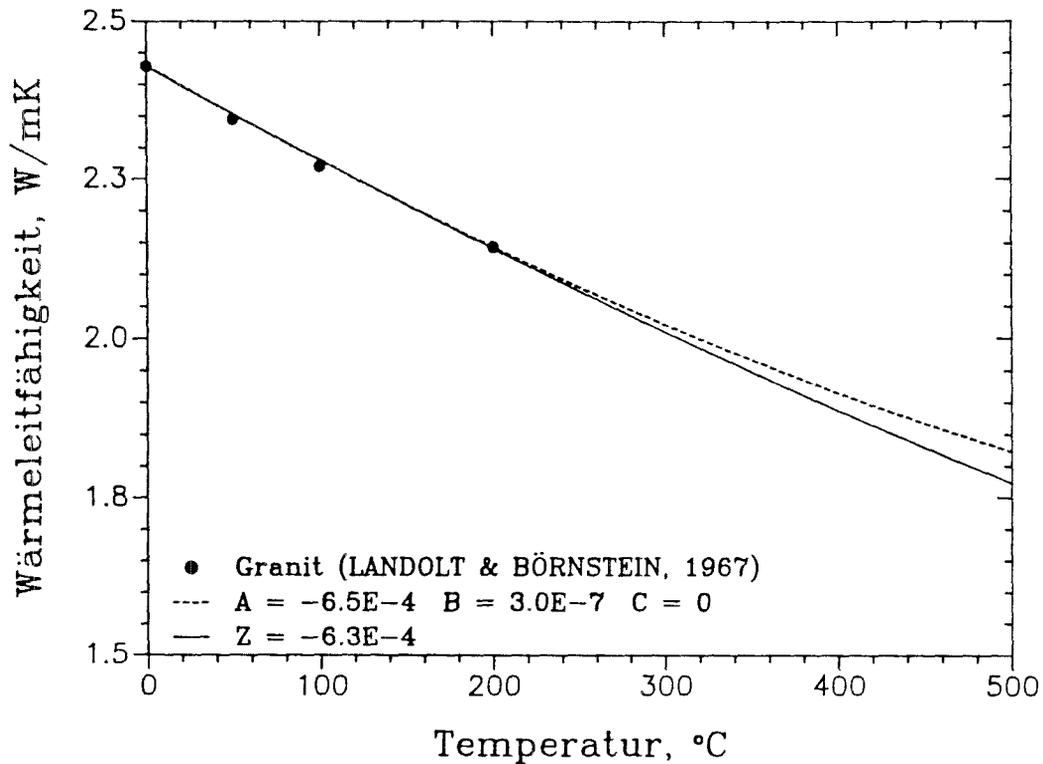


Abb. A.1.8: Temperaturabhängigkeit der Wärmeleitfähigkeit von Granit und ihre numerische Approximation

Nimmt man für einen Amphibolit und einen Disthen-Sillimanit-Gneis eine Wärmeleitfähigkeit (WLF) von jeweils 3 W/mK bei 20 °C und 10 MPa an, so resultieren nach Gl.(A.1.11) die in den Abbildungen A.1.9 und A.1.10 dargestellten Kurven.

Effektive Wärmeleitfähigkeit

Neben einer möglichst genauen Kenntnis der Wärmeleitfähigkeit des Festkornanteiles benötigt man für die Berechnung des Wärmetransportes in Fest- und Lockergesteinen vor allem auch Informationen über Anteil des Gesamtporenraumes und dessen Verteilung. Zu den beiden letztgenannten Einflußgrößen existieren bereits eine Fülle von Detailuntersuchungen, /64/, /70/, /78/, /101/, /210/, /217/.

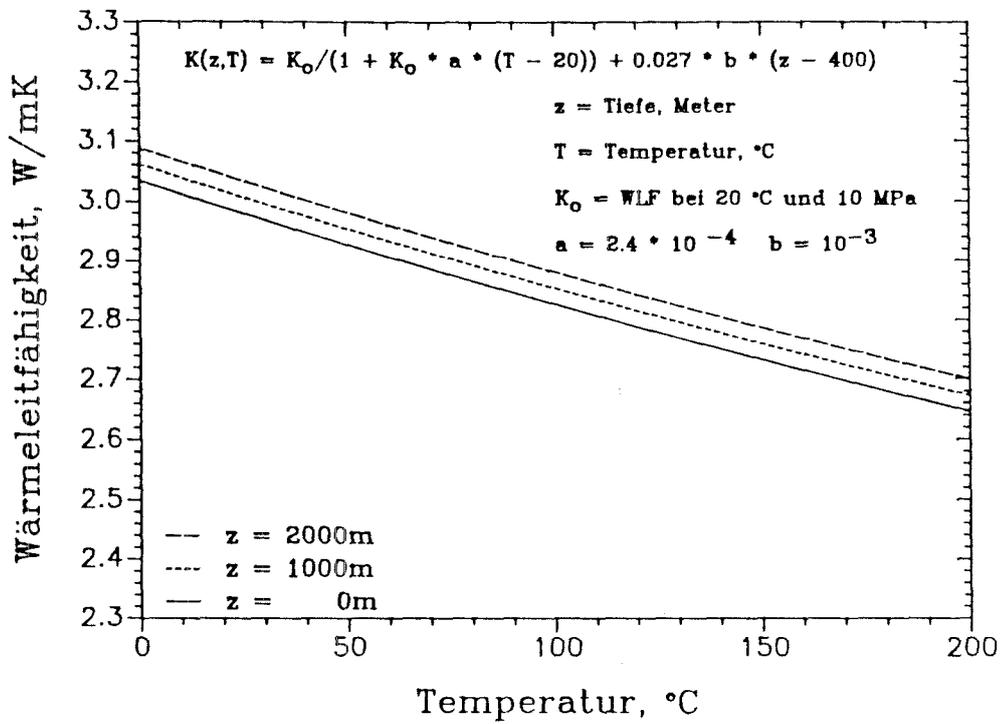


Abb. A.1.9: Druck- und Temperaturabhängigkeit der Wärmeleitfähigkeit von Amphibolit.

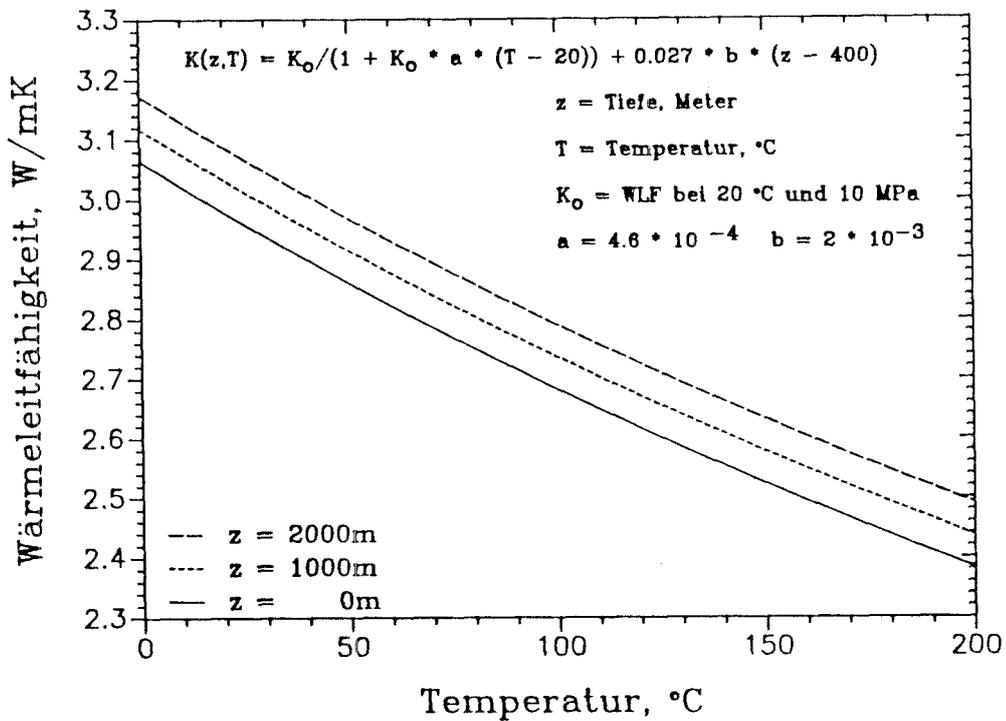


Abb. A.1.10: Druck- und Temperaturabhängigkeit der Wärmeleitfähigkeit von Disthen-Sillimanit-Gneis.

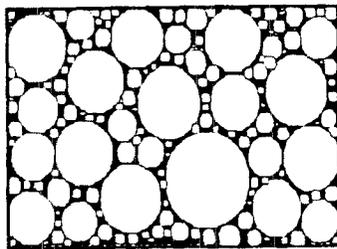
Im hier vorgestellten Modell wurden daher drei Näherungsverfahren zur

Bestimmung der effektiven Wärmeleitfähigkeit berücksichtigt. ¹ Bei dem ersten Verfahren werden die Wärmeleitfähigkeiten der Porenraumfüllung und der Gesteinsmatrix, K_p und K_g , gewichtet, arithmetisch gemittelt:

$$K_{eff} = (1 - \Phi_g) K_g + \Phi_g K_p \quad (A.1.12)$$

Die beiden von /210/ vorgeschlagenen Verfahren tragen der unterschiedlichen Porenraumverteilung Rechnung. Demnach ist bei der Einfluß der Porenraumfüllung auf die effektive Wärmeleitfähigkeit bei Lockergesteinen, z.B. Kiesen und Sanden, am größten. Bei vielen Festgesteinen liegen die Poren als isolierte Hohlräume vor, sodaß sich die effektive Wärmeleitfähigkeit maßgeblich von der der Gesteinsmatrix ableitet.

Modell 1:



$$K_{eff} = K_g \left[1 - \frac{\Phi_g (1 + 2 \frac{K_p}{K_g}) (1 - \frac{K_p}{K_g})}{\Phi_g (1 - \frac{K_p}{K_g}) + 3 \frac{K_p}{K_g}} \right]$$

Modell 2:



$$K_{eff} = K_g \left[1 - \frac{3\Phi_g (1 - \frac{K_p}{K_g})}{2 + \Phi_g + \frac{K_p}{K_g}} \right]$$

Abb. A.1.11: Näherungsverfahren zu Bestimmung der effektiven Wärmeleitfähigkeit nach /210/.

¹ In der Literatur werden natürlich wesentlich mehr Berechnungsverfahren beschrieben. Die meisten Methoden basieren auf — z.T. nicht sehr einfach nachzuvollziehenden — empirischen Annahmen. Der interessierte Leser sei daher auf FAROUKI(1986) verwiesen, der eine nahezu vollständige Übersicht aller dokumentierten Berechnungsmethoden gibt.

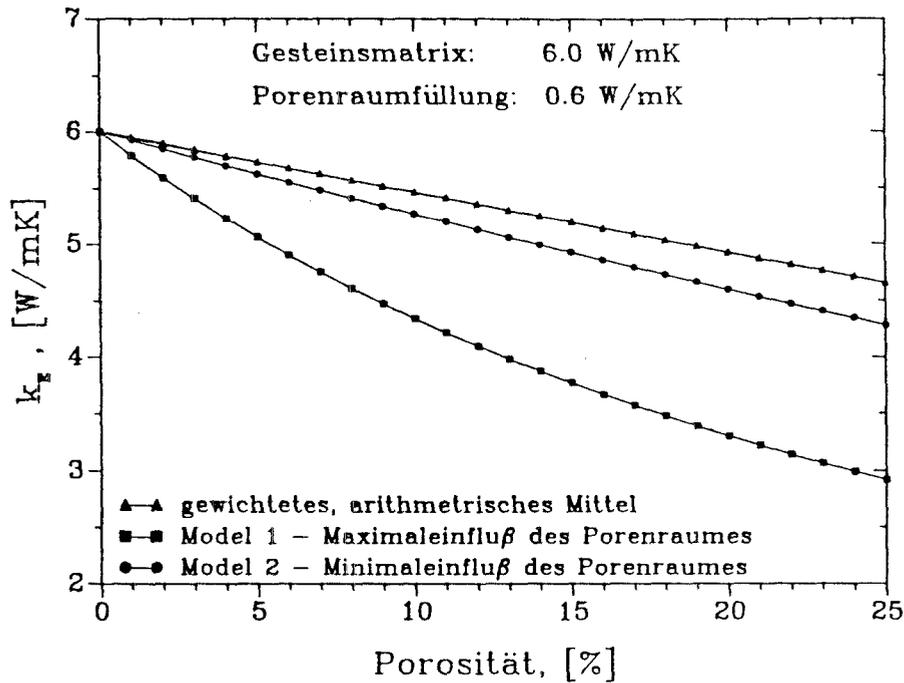


Abb. A.1.12: Einfluß der Porenraumverteilung auf die effektive Wärmeleitfähigkeit nach /210/ im Vergleich zum gewichteten arithmetischen Mittel.

Für den Fall, daß die Porenräume mit einem Zwei- oder Mehrphasengemisch gefüllt sind, wird die Wärmeleitfähigkeit der Porenraumfüllung durch gewichtete Mittelwertbildung berechnet.

Wasser - Luft - Gemisch:

$$K_p = (1 - \Psi)K_l + \Psi K_w \quad (\text{A.1.13})$$

Wasser - Eis - Gemisch:

$$K_p = X_e K_e + X_w K_w \quad (\text{A.1.14})$$

Wasser - Eis - Luft - Gemisch:

$$K_p = (1 - \Psi)K_l + \Psi (X_e K_e + X_w K_w) \quad (\text{A.1.15})$$

Wärmekapazität:

Betrachtet man lediglich den Kornanteil, so variieren die Wärmekapazitäten der häufigsten Gesteinsarten wesentlich geringer als deren Wärmeleitfähigkeiten. Bei sehr porösen Gesteinen spielt jedoch der Gesamtporenraum und der Wassersättigungsgrad eine entscheidende Rolle, da Wasser über eine etwa 5-fach größere Wärmekapazität verfügt. Die Wärmekapazität der meisten Gesteine korreliert positiv mit der Temperaturerhöhung. An Pulverpräparaten kann mittels eines Thermoanalytensystems die Temperaturabhängigkeit der Wärmekapazität recht genau gemessen werden.

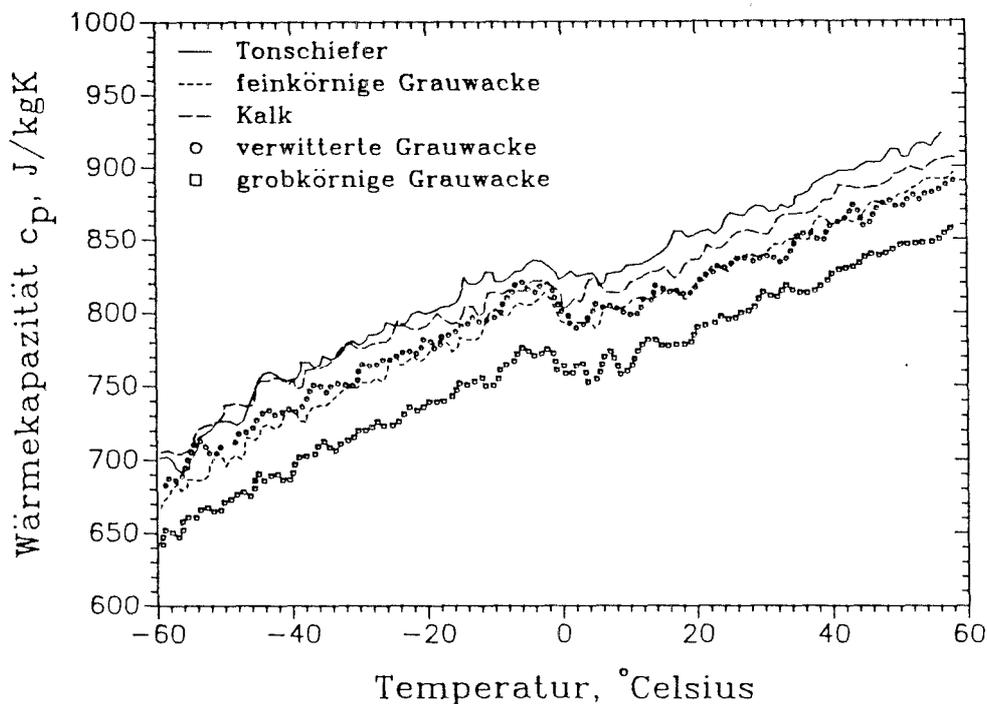


Abb. A.1.13: Temperaturabhängigkeit der Wärmekapazität einiger paläozoischer Sedimentgesteine, Messungen mit dem METTLER Thermoanalytensystem TA3000

Für den in der vorliegenden Arbeit betrachteten Temperaturbereich erschienen die Änderungen im Vergleich zu der Genauigkeit, mit der man den Gesamtporenraum und den Wassersättigungsgrad bestimmen kann, zu gering, sodaß die Wärmekapazität des Gesteins als konstant angenommen wurde. Im folgenden werden für die wichtigsten Magmatite, Metamorphite und Sedimentgesteine die Wärmekapazitäten tabelliert. Die Meßtemperatur beträgt zwischen 0 und 30 °C.

Tab. A.1.4 Wärmekapazitäten wichtiger Erguß- und Tiefengesteine

<i>Gestein</i>	<i>Wärmekapazität [J/kgK]</i>	<i>Quelle</i>
Granit	800-816	/A01/
Syenit	831	/A01/
Diorit	810	/A01/
Gabbro	720 ¹	/A01/
Porphyr	825	/A01/
Basalt	858	/A01/
Diabas	700	/A01/
Trachyt	870	/A01/
Tuffe	1240	/A01/

Tab. A.1.5 Wärmekapazitäten wichtiger metamorpher Gesteine

<i>Gestein</i>	<i>Wärmekapazität [J/kgK]</i>	<i>Quelle</i>
Gneis	820-895	/A01/
Schiefer	707	/A01/
Tonschiefer	835	/A03/
Marmor	750	/A01/
Quarzit	700	/A01/

¹ aus dem Mineralbestand berechnet

Tab. A.1.6 Wärmekapazitäten wichtiger Sedimentgesteine

Gestein	Wärmekapazität [J/kgK]	Quelle
Konglomerat	875	/A01/
Sandstein	866	/A01/
Sandstein, trocken	766	/A02/
schluffiger Sand	846	/A02/
Siltstein	854	/A02/
Sand, 100 °C getrocknet	850	/A01/
Sand, luftgetrocknet	873	/A01/
Sand, fein, trocken	766	/A02/
Sand, grob, trocken	766	/A02/
Grauwacke, feinkörnig	790	/A03/
Grauwacke, grobkörnig	770	/A03/
Grauwacke, verwittert	800	/A03/
Massenkalk	820	/A01/
Kalkstein	908	/A01/
Kalkstein, (-)	846	/A02/
Mergel	866	/A01/
Tone	800	/A01/
Torf, 100 °C getrocknet	2123	/A01/
Torf, lufttrocken	2215	/A01/

/A01/ LANDOLT & BÖRNSTEIN (1967): 6. Auflage, Band III

/A02/ FAROUQ ALI, S. M. (1970): Oil Recovery by Hot Water Flooding.- Producers Publishing Co., Inc. Bradford, PA.

/A03/ Messungen mit dem Thermoanalysensystem TA3000 der Firma Mettler, Gießen, vergl. Abb. A.1.13.

Sind die Wärmekapazität des Festkornanteiles des Gesteins, der Gesamtporenraum und der Wassersättigungsgrad bekannt, so kann man die effektive Wärmekapazität wie folgt berechnen:

$$C_{eff} = \Phi_g \left((1 - \Psi)C_l + \Psi (X_e C_e + X_w C_w) \right) + (1 - \Phi_g) C_g \quad (A.1.16)$$

Wärmeproduktion

Durch den Zerfall radioaktiver Elemente und chemische Reaktionen produzieren bis auf wenige Ausnahmen nahezu alle Gesteine Wärme. Die Wärme radiogenen Ursprungs läßt sich nach RYBACH (1973), /173/. überschlägig aus den Konzentrationen der radioaktiven Isotope der Elemente U, Th⁵ und K⁶ nach:

$$Q_{rad} = (0.178C_U + 0.193C_{Th} + 0.262C_K) \cdot 0.133\rho_g, \quad [\mu W/m^3] \quad (A.1.17)$$

abschätzen. Während für die Temperaturprognose in größere Tiefen — Beispiel KTB — eine gute Vorstellung über die Wärmeproduktionsraten maßgeblich für die Qualität des Berechnungsergebnisses ist, kann man diesen Effekt bei der Kalkulation des Temperaturfeldes der obersten 100 m der Erdkruste vernachlässigen.

Tab. A.1.7 Wärmeproduktion verschiedener Gesteine

<i>Gestein</i>	<i>Wärmeproduktion</i> [$\mu W/m^3$]	<i>Quelle</i>
Gneis	1.21-1.45	/A01/
Amphibolit	0.37	/A01/
Granit	6-9	/A02/
/A01/	internes Arbeitspapier der Arge 4 des Kontinentalen Tiefbohrprogrammes: Prof. Dr. Hänel, KTB	
/A02/	Falkenstein - Granit/Oberpfalz, persönliche Mitteilung von Dr. Huenges, KTB-Feldlabor	

⁵ U und Th in ppm

⁶ als Gesamtkalium in Prozent

Anhang 2: Programmaufbau und Variablenerklärung

Beschreibung des Programmaufbaus von TRADIKON-3D

TRADIKON-3D baut sich aus einem Hauptprogramm, 62 Unterprogrammen und 5 Funktionen auf. Jedes der Unterprogramme verfolgt einen eng umgrenzten Zweck, sodaß sehr leicht Änderungen und Modifikationen am Quelltext vorgenommen werden können. Das Programm wurde auf einem IBM-XT kompatiblen Personal-Computer in der Programmiersprache FORTRAN 77 unter Einsatz eines MICROSOFT-Fortran-Compilers der Version 4.0 entwickelt und hält sich weitgehend an den Standard ANSI X3.9 von 1978. Es sind daher, wenn überhaupt, nur geringfügige Änderungen notwendig, um den Quelltext auf andere Rechnertypen anzupassen. Vom Standard weichen folgende Sprachelemente ab:

INTEGER*2 2- Byte Integer, FORTRAN verwendet standardmäßig nur 4-Byte Integer

\ Backslash - Editor, unterdrückt das <CR> - Zeichen bei der formatierten Ausgabe

Hauptprogramm

TRAD3D übernimmt die Steuerung des Programmablaufes und Koordination der Unterprogrammaufrufe

Unterprogramme

AUFNUL setzt die Koeffizientenfelder auf Null

DATAUS erzeugt einen kompletten Output der Eingangsparameter zur Kontrolle der Eingabedatei.

DATEIN liest, sofern mit **INPOP** > 0 ein umfangreicher Datensatz gewählt wurde, Schicht für Schicht und Knoten für Knoten die physikalischen Eingangsparameter wie Ausgangstemperatur, Dichte, Wärmekapazität, Wärme-

leitfähigkeit, Nutz- und Gesamtporosität des Gesteins, Wassersättigung in den Zellen, Art der Knoten, Grundwasserfließverhältnisse sowie die geometrischen Daten des Modellgebietes. Falls das Strömungsfeld berechnet werden soll, werden zusätzlich die Ausgangspiezometerhöhen, die hydraulischen Leitfähigkeiten, die Basis des Aquifers, die spezifischen Speicherkoeffizienten und Zugaben bzw. Entnahmen angefordert.

- DELTAH* berechnet, falls mit **IGWF** > 0 die Strömungsberechnung aktiviert wurde, aus den benachbarten Piezometerhöhen und der NN-Höhe der obersten Schicht die Höhe der wassergefüllten Zellengrenzfläche. Bei ungespannten Aquiferen werden während der Iteration die Transmissivitäten derjenigen Zellen neu berechnet, deren vertikale Zellengrenzfläche nicht auf ganzer Länge benetzt sind.
- DKTEST* berechnet für sämtliche Zellen aus der Strömungsgeschwindigkeit und dem Temperaturleitwert die Gitter-Peclet und gibt ggf. die Meldung aus, daß die Diskretisierungsvorschriften verletzt wurden. Wenn **IDSTOP** = 2, wird die Berechnung ggf. abgebrochen.
- DZTEST* überprüft, ob die vorgewählte Zeitschrittlänge zu physikalisch realistischen Ergebnissen führen kann und verkürzt optional (**IDZTST** = 2) die Zeitschrittlänge.
- ENERGI* ermittelt die Energiebilanz für alle Zellen durch Aufruf des Unterprogramms *ZELLEN* und stellt eine Gesamtenergiebilanz bezogen auf den letzten Ausdruck eines Temperaturprofils auf.
- FELDER* Block - Data - Unterprogramm zur Vorbesetzung der COMMON - Blöcke
- FLIESS* ermittelt, falls mit **IGWF** = 1 die Strömungsberechnung aktiviert wurde, aus der Piezometerhöhendifferenz zweier benachbarter Zellen und der hydraulischen Leitfähigkeit ihrer gemeinsamen Grenzfläche die Fließgeschwindigkeit.
- FLUKOF* berechnet, sofern es sich um ein Diffusions-Konvektionsproblem handelt, die Koeffizienten des konvektiven Anteils am Wärmetransport.

<i>FLUSFL</i>	prüft, sofern es sich um ein Konvektion-Diffusionsproblem handelt, anhand der Wassersättigung und des Phasenzustandes ob das Fluid nur in vertikaler Richtung oder überhaupt nicht fließt, (z.B. Eisbildung).
<i>GITTER</i>	liest und berechnet die geometrischen Parameter der Diskretisierung entweder aus den vorgegebenen Knotenabständen (<i>IVOLDF</i> = 1) oder aus den vorgegebenen Zellenkantenlängen (<i>IVOLDF</i> = 0).
<i>HARMON</i>	berechnet den gewichteten, harmonischen Mittelwert der überreichten Argumente. (--> <i>INFINT</i>)
<i>HEADER</i>	Ausgabe des Programmkopfes auf dem Bildschirm und der Ergebnisdatei.
<i>IADIST</i>	löst iterativ, durch Aufruf der Subroutine <i>THOMAS</i> , nach dem <i>IADI</i> -Verfahren das lineare Gleichungssystem der Strömungsberechnung und behandelt die Randbedingungen. Bei ungespannten Aquiferen werden die Transmissivitäten aller teilbenetzten, vertikalen Zellengrenzen laufend durch Aufruf der Subroutine <i>DELTAH</i> aktualisiert.
<i>IADITP</i>	löst iterativ, durch Aufruf der Subroutine <i>THOMAS</i> , nach dem <i>IADI</i> -Verfahren das lineare Gleichungssystem zur Berechnung des Wärmetransports und behandelt die Randbedingungen; ferner wird bei <i>IFROST</i> > 0 eine Phasenwechselkontrolle durchgeführt. Wenn die Temperaturabhängigkeit der Wärmeleitfähigkeit berücksichtigt werden soll, (<i>IWCP</i> > 2) wird während der Iteration gemäß dem gewählten Verfahren die Wärmeleitfähigkeit neu berechnet.
<i>INFINT</i>	ermittelt die Wichtungsfaktoren für die Berechnung der Grenzflächeneigenschaften aus dem gewichteten, harmonischen Mittel zweier benachbarter Zellen. (--> <i>HARMON</i>)
<i>INFORM</i>	liest allgemeine Programminformationen zur Problemstellung wie Problemtitel, Art des Problems, Knotenanzahl, Lage der Knoten in den Zellen, Zeitschrittlänge, Simulationsdauer, Iterationsverfahren usw. ein und gibt entsprechende Meldungen auf der Ergebnisdatei aus

<i>ISOASC</i>	gibt mit den Parametern IPROX , IPROY und IPROZ festzulegende Schnitte der Temperaturdaten, eventuell auch der Piezometerhöhen im ASCII - Format des TRADIKON-3D - Postprozessors aus
<i>ISOBIN</i>	gibt mit den Parametern IPROX , IPROY und IPROZ festzulegende Schnitte der Temperaturdaten, eventuell auch der Piezometerhöhen im MS-Binärformat des TRADIKON-3D - Postprozessors aus
<i>ISODRV</i>	koordiniert den Aufruf der Subroutinen <i>ISOASC</i> bzw. <i>ISOBIN</i>
<i>ITERAH</i>	löst iterativ nach dem gewählten Verfahren - GAUSS - SEIDEL oder SOR - das lineare Gleichungssystem der Strömungsberechnung und behandelt die Randbedingungen. Bei ungespannten Aquiferen werden die Transmissivitäten aller teilbenetzten, vertikalen Zellengrenzen laufend durch Aufruf der Subroutine DELTAH aktualisiert.
<i>ITERAT</i>	löst iterativ nach dem gewählten Verfahren das lineare Gleichungssystem zur Berechnung des Wärmetransports und behandelt die Randbedingungen; ferner wird bei IFROST ≥ 1 eine Phasenwechselkontrolle durchgeführt. Wenn die Temperaturabhängigkeit der Wärmeleitfähigkeit berücksichtigt werden soll, (IWCP > 2) wird während der Iteration gemäß dem gewählten Verfahren die Wärmeleitfähigkeit neu berechnet.
<i>KOEFFH</i>	berechnet je nach gewählter Problemstellung alle Koeffizienten des linearen Gleichungssystems zur Berechnung des Strömungsfeldes.
<i>KOEFFT</i>	berechnet je nach gewählter Problemstellung alle Koeffizienten des linearen Gleichungssystems zur Berechnung des Temperaturfeldes.
<i>KONDUK</i>	ermittelt die hydraulische Leitfähigkeiten der Zellengrenzflächen aus dem gewichteten, harmonischen Mittelwert zweier benachbarter Zellen.
<i>KONINF</i>	ermittelt die Wärmeleitfähigkeit an den Zellengrenzflächen aus dem harmonischen, gewichteten Mittel.
<i>KONKOF</i>	berechnet die Koeffizienten des konduktiven bzw. diffundiven Anteils am Wärmetransport bzw. die Koeffizienten zur Berechnung des Strömungsfeldes.

- KOPIER* kopiert die Temperaturwerte oder Piezometerhöhen des letzten Zeitschrittes in das Feld **T0** bzw. **H0**.
- KURZIN* liest, sofern mit **INPOP** = 0 ein kompakter Datensatz gewählt wurde, in zwei Schritten, Schicht für Schicht, die physikalischen Eingangsparameter wie Ausgangstemperatur, Dichte, Wärmekapazität, Wärmeleitfähigkeit, Nutz- und Gesamtporosität des Gesteins, Wassersättigung in den Zellen, Art der Knoten, Grundwasserfließverhältnisse sowie die geometrischen Daten des Modellgebietes. Wenn **IGWF** = 1, werden auch die Ausgangspiezometerhöhen, spezifischen Speicherkoeffizienten und hydraulische Leitfähigkeiten angefordert. Im ersten Schritt werden allen Knoten einer Schicht die selben Werte zugeordnet. Im zweiten Schritt können bestimmten Knoten, Knotenreihen oder -ebenen über vordefinierte Schlüsselworte abweichende Eigenschaften zugewiesen werden. (--> *UMWAND*)
- OBSDAT* gibt die berechneten Temperaturen, wenn **IGWF** = 1 auch die Piezometerhöhen ausgesuchter Beobachtungsknoten auf der sequentiellen Datei **OBSFIL** aus.
- OMEGAT* testet optional (**ISOR** = 1) den OMEGA - Wert, sofern ein SOR/SUR - Verfahren durchgeführt werden soll; es läßt sich somit prüfen, welcher Relaxationsfaktor bei bestimmten Fragestellungen die Konvergenz beschleunigt. Bei stark nichtlinearen Problemen empfiehlt sich ein SUR - Verfahren.
- PECLET* berechnet je nach gewähltem Verfahren die Peclet- Wichtungsfunktion für den diffundiven und konvektiven Anteil am Wärmetransport.
- PIEDRU* gibt in festzulegenden Intervallen und Schnittlagen Piezometerhöhenprofile auf der Ergebnisdatei aus. Die Profile verlaufen parallel zu Achsen des kartesischen Koordinatensystems
- PHATST* sofern mit **IFROST** > 0 eine Phasenüberprüfung aktiviert wurde, wird je nach gewähltem Gefrieralgorithmus der Anteil des ungefrorenen Fluids berechnet; die verbleibende Schmelzwärme wird auf die Wärmekapazität des Fluids umgerechnet.

<i>PROFIL</i>	gibt optional alle physikalischen Eingabeparameter in Form von festzulegenden Profilschnitten, die parallel zu den Achsen des kartesischen Koordinatensystems verlaufen, aus.
<i>QKOEFT</i>	berechnet die Quellen- oder Senkentermkoeffizienten der Temperaturberechnung.
<i>QKOEFH</i>	berechnet die Quellen- oder Senkentermkoeffizienten der Strömungsberechnung.
<i>RICHTG</i>	bewirkt ein Iterieren in alternierenden Richtungen, sofern IALTER = 1 gesetzt wurde.
<i>ROUTIN</i>	gibt den Namen der gerade aktiven Subroutine auf dem Bildschirm bzw. auf dem JOB - Protokollfile aus, sofern IECHO > 0.
<i>TAUPKT</i>	berechnet den Gefrierpunkt des Fluids in verschiedenen Tiefen.
<i>THOMAS</i>	löst eine tridiagonale Matrix nach dem THOMAS - Algorithmus.
<i>TMPDRU</i>	gibt in festzulegenden Intervallen und Schnittlagen Temperaturprofile auf der Ergebnisdatei aus. Die Profile verlaufen parallel zu Achsen des kartesischen Koordinatensystems
<i>UMWDBL</i>	wandelt die verschlüsselten DOUBLE PRECISION - Variablen in Eingabedaten um.
<i>UMWINT</i>	wandelt die verschlüsselten INTEGER - Variablen in Eingabedaten um.
<i>UMWREA</i>	wandelt die verschlüsselten REAL - Variablen in Eingabedaten um.
<i>UMWAND</i>	koordiniert die Aufrufe der Unterprogramme <i>UMWDBL</i> , <i>UMWINT</i> und <i>UMREA</i>
<i>VDRUCK</i>	gibt, sofern das Strömungsfeld mit IGWF = 0 als konstant angenommen wurde, in tabellarischer Form die Grund- und Sickerwasserfließgeschwindigkeiten auf der Ergebnisdatei aus.
<i>VISKOS</i>	berechnet durch Aufruf der Funktion <i>VISKIN</i> die kinematische Viskosität von Wasser an der Zellengrenzfläche; hierbei wird berücksichtigt, aus

welcher Richtung das Fluid strömt; die berechneten Werte dienen der Berechnung der wirklichen Permeabilität.

- WARMIN* fordert, sofern mit **IWARMS** = 1 ein "Warmstart" vereinbart wurde, sämtliche Temperaturdaten, eventuell auch die Piezometerhöhen einer vorherigen Simulation von der sequentiellen Datei **TRADWS** an.
- WARMOU* schreibt, sofern die Ausgabe mit **IWSDAT** = 1 aktiviert wurde, parallel zu jeder Ergebnisausgabe das aktuelle Temperaturfeld, eventuell auch die Piezometerhöhen und Filtergeschwindigkeiten in die sequentielle Datei **TRADWS**; diese Daten können in für eine spätere Simulation als Eingabedaten dienen ("Warmstart"). Auf die Art kann z.B. mit einem zuvor stationär berechneten Strömungsfeld instationär weitergerechnet werden.
- WASEIS* Berechnung des ungefrorenen Wasseranteils in Abhängigkeit des gewählten Gefriermodells; die hier beschriebene Programmversion verfügt über einen Modellansatz.
- WCPH2O* optionale Berechnung der Wärmekapazität eines Eis-Wasser-Gemisches bei vorgegebenem Mischungsverhältnis.
- WFLUSS* berechnet den Wärmefluss zwischen den mit **IFLAG** = 5 verschlüsselten Zellen und den Nachbarzellen.
- WICHHY* führt die zeitliche Wichtung zwischen alten und neuen Piezometerhöhen durch; d.h. Explizit-, Implizit - oder CRANK - NICHOLSON Verfahren.
- WICHTP* führt die zeitliche Wichtung zwischen alten und neuen Temperatur durch; d.h. Explizit-, Implizit - oder CRANK - NICHOLSON Verfahren.
- ZEITBE* transformiert den aktuellen Simulationszeitpunkt von Sekunden in Wochen, Tage, Stunden und Sekunden.
- ZELLEN* berechnet für die Zelle (**I,J,K**) den Energieinhalt.
- ZUSDAT* liest in zu definierenden Intervallen aus einer zusätzlichen Eingabedatei zeitlich veränderliche Eingabeparameter

Funktionen

<i>CELS</i>	Umrechnung der Temperatur von Kelvin in Grad Celsius
<i>CELV</i>	Umrechnung der Temperatur von Grad Celsius in Kelvin
<i>GRCBCM</i>	Umrechnung der Dichte von Kg/m^3 in g/cm^3
<i>RHOH2O</i>	Berechnung des Dichte des Fluids in Abhängigkeit des Phasenzustandes
<i>VISKIN</i>	Berechnung der kinematischen Viskosität von Wasser bei der betreffenden Temperatur

Beschreibung der Variablen und Felder von TRADIKON-3D

Alle physikalischen Eingabeparameter müssen, sofern nicht ausdrücklich anders erwähnt, in SI - Einheiten eingegeben werden. Die Pfeile in der äußerst linken Druckspalte bedeuten:

-->	zwingend erforderliche Eingaben
=>	alternative Eingaben, z.B. DIFX oder DELTAX
>>	optionale Eingaben, z.B. SK, Q etc.

AA	A - Koeffizientenfeld des tridiagonalen Matrizenalgorithmus; Dim. (IMAX)
>>	ALFAHY Bestimmung des Berechnungsschemas für die Strömungsberechnung ALFAHY = 0 => Explizit Schema 1 > ALFAHY > 0 => Implizit Schema ALFAHY = 0.5 => CRANK - NICOLSON - Schema ALFAHY = 1 => Vollimplizit - Schema Diese Option ist nur in Verbindung mit dem Gauss-Seidel- Verfahren möglich; das IADI-Verfahren rechnet stets vollimplizit
>>	ALFATP Bestimmung des Berechnungsschemas für die Temperatur- berechnung ALFATP = 0 => Explizit Schema 1 > ALFATP > 0 => Implizit Schema ALFATP = 0.5 => CRANK - NICOLSON - Schema ALFATP = 1 => Vollimplizit - Schema

Diese Option ist nur in Verbindung mit dem Gauss-Seidel-Verfahren möglich; das IADI-Verfahren rechnet stets vollimplizit

- > AUS bestimmt den Namen der Ausgabedatei; der Name TERMINAL bewirkt eine Bildschirmausgabe (--> IAUS)
- >> ANISO Anisotropiefaktoren der Wärmeleitfähigkeit des Gesteins in X, Y und Z - Richtung; Dim.(3)
- BB B - Koeffizientenfeld des tridiagonalen Matrizenalgorithmus; Dim. (IMAX)
- CC C - Koeffizientenfeld des tridiagonalen Matrizenalgorithmus; Dim. (IMAX)
- => CPE Wärmekapazität der festen Phase des Fluids (Eis) (--> IWCP)
- >> CPG Wärmekapazität des Gesteins; Dim.(JX, JY, JZ)
- => CPL Wärmekapazität der gasförmigen Phase des Fluids (--> IWCP)
- => CPW Wärmekapazität des Fluids (Wasser) (--> IWCP)
- > DATOP Name der optionalen Eingabedatei für zeitabhängige Eingabedaten; DATOP wird nur geöffnet, wenn IDATOP = 1
- DD Feld der abhängigen Variablen des tridiagonalen Matrizenalgorithmus; Dim. (IMAX)
- => DELTAX Länge der Zellen in X - Richtung; (--> IVOLDF); Dim.(IX)
- => DELTAY Länge der Zellen in Y - Richtung; (--> IVOLDF); Dim.(IY)
- => DELTAZ Länge der Zellen in Z - Richtung; (--> IVOLDF); Dim.(IZ)
- => DIFX Abstände der Knoten in X - Richtung; (--> IVOLDF); Dim.(IX)
- => DIFY Abstände der Knoten in Y - Richtung; (--> IVOLDF); Dim.(IY)
- => DIFZ Abstände der Knoten in Z - Richtung; (--> IVOLDF); Dim.(IZ)
- DIVX Länge der Knoten-Interface-Strecken in X - Richtung; Dim.(IX, 2)
- DIVY Länge der Knoten-Interface-Strecken in Y - Richtung; Dim.(IY, 2)
- DIVZ Länge der Knoten-Interface-Strecken in Z - Richtung; Dim.(IZ, 2)

- >> DTWL Koeffizienten der polynomischen, exponentiellen bzw. reziproken Approximation der Temperaturabhängigkeit der Wärmeleitfähigkeit des Gesteines; wird nur benötigt, wenn IWCP = 3
- => DWAN maximal erlaubte Änderung des ungefrorenen Wasseranteils während einer Iteration; wird nur benötigt, wenn IFROST > 0 und IWCP = 2
- DXYZ Zellenvolumina; Dim.(IX, IY, IZ)
- > DZEIT Zeitschrittlänge in Sekunden, wird aus ISTLAE und ZFLAG berechnet, gegebenenfalls in der Subroutine DZTEST verkürzt (--> IDZTST)
- > DZF Faktor der Zeitschrittlängenvergrößerung; d.h. nach jedem Zeitschritt wird DZEIT um diesen Faktor vergrößert.
- => EIN Name der Haupteingabedatei; TRADIKON-3D sucht zunächst nach dem Programmstart die Datei "TRADAT"; falls die Suche erfolglos bleibt, wird ein Dateiname über die Tastatur angefordert
- => FLAG Art der Zusatzinformation; Dim.(IWERT); (--> INPOP)
 - CPGE => Wärmekapazität des Gesteins (--> CPG)
 - ENDE => beendet die Eingabe von Zusatzinformationen
 - GWNB => Grundwasserneubildung ($m^3/s \cdot m^2$)
 - GWVX => Fließgeschwindigkeit in X - Richtung (--> VX)
 - GWVY => Fließgeschwindigkeit in Y - Richtung (--> VY)
 - GWVZ => Fließgeschwindigkeit in Z - Richtung (--> VZ)
 - HTOT => inaktiver Knoten in der Strömungsberechnung:
TTX = TTY = TTZ = 0 undurchlässig
SK = ZZZZ Speicherkoeffizient unendlich
JFLAG = 2 konstante Piezometerhöhe
H = 999. nur für Postprozessor
 - IFLG => Knotenschlüssel Temperaturen (--> IFLAG)
 - JFLAG => Knotenschlüssel Strömung (--> JFLAG)
 - LEAK => Leakagefaktor (s^{-1})
 - PHIG => Gesamtporosität des Gesteins (--> PHIG)
 - PHIN => durchflußwirksame Porosität (--> PHIN)
 - QUEL => Entnahme oder Zugabe von Fluid (--> Q)
 - RHOG => Dichte des Gesteins in g/cm^3 (--> RHOG)
 - RNZF => Randzufluß in das Modellareal m^3/s (--> Q)
 - SPEI => spezifischer Speicherkoeffizient (--> SK)
 - SOUC => zugeführte oder entzogene Energie(--> SC)
 - TRSX => Durchlässigkeit in X-Richtung (--> TTX)
 - TRSY => Durchlässigkeit in Y-Richtung (--> TTY)

- TRSZ => Durchlässigkeit in Z-Richtung (--> TTZ)
 TTOT => inaktiver Knoten der Temperaturberechnung:
 WLKG = 0 Isolator
 PHIG = 0 keine Porosität
 PHIN = 0 keine Nutzporosität
 CPG = ZZZZ Wärmekapazität unendlich
 IFLAG = 2 konstante Temperatur
 T = 999. nur für Postprozessor
 TXYZ => isotrope Durchlässigkeit (--> TTX,TTY,TTZ)
 TEMP => Temperatur in °C, wird in Kelvin umgerechnet
 (--> T)
 WLKG => Wärmeleitfähigkeit des Gesteins (--> WLKG)
 WPOT => Piezometerhöhe (--> H)
 (--> WERT, IXX, IYY, IZZ, IUFL)
- >> FLURAB Grundwasserflurabstand, wird nur benötigt, wenn IDIF = 1 und IGWF = 0
- FX Interface-Wichtungsfaktoren in X - Richtung; Dim.(IX)
- FY Interface-Wichtungsfaktoren in Y - Richtung; Dim.(IY)
- FZ Interface-Wichtungsfaktoren in Z - Richtung; Dim.(IZ)
- EANF Energieinhalt des Berechnungsgebietes zu Beginn der Simulation
- EALT Energieinhalt des Berechnungsgebietes nach der letzten Energiebilanz
- >> GWV Grundwasserfließgeschwindigkeit in Richtung des mit WINKEL festgelegten Winkel; wird nur benötigt, wenn IDIF = 1 und IGWF = 0; (--> WINKEL)
- >> H aktuelle Piezometerhöhe; wird nur benötigt, wenn IGWF = 1; Dim. (KX, KY, KZ)
- H0 Piezometerhöhe des letzten Zeitschrittes; Dim. (KX, KY, KZ)
- I aktueller Index in X - Richtung
- > IALTER Reihenfolge des Iterationsablaufes
 0 => es wird in aufsteigender Reihenfolge iteriert
 1 => es wird in alternierenden Richtungen iteriert

- IAUS Kanalnummer, auf die die Ergebnisse umgeleitet werden
 1 => Ergebnisse werden am Bildschirm angezeigt
 6 => Ergebnisse werden auf eine Datei geschrieben
 (--> AUS)
- > IAUSOP Wahl des Output - Umfanges
 1 => nur Temperaturprofile drucken
 2 => Temperatur- und Eingangsdatenprofile
 3 => alle Eingangsdaten und Temperaturprofile
- > IBX optionale Eingabe von konstanten Randbedingungen in X - Richtung
 0 => die beiden Ränder in X - Richtung sind sowohl wasser- als auch energieundurchlässig
 1 => die beiden Ränder in X - Richtung behalten eine konstante Temperatur, sind aber wasserundurchlässig
 2 => die beiden Ränder in X - Richtung sind energieundurchlässig, verfügen aber über eine konstante Piezometerhöhe
 3 => die beiden Ränder in X - Richtung verfügen sowohl über eine konstante Temperatur als auch eine konstante Piezometerhöhe
 (--> IFLAG, JFLAG)
- > IBY optionale Eingabe von konstanten Randbedingungen in Y - Richtung
 0 => die beiden Ränder in Y - Richtung sind sowohl wasser- als auch energieundurchlässig
 1 => die beiden Ränder in Y - Richtung behalten eine konstante Temperatur, sind aber wasserundurchlässig
 2 => die beiden Ränder in Y - Richtung sind energieundurchlässig, verfügen aber über eine konstante Piezometerhöhe
 3 => die beiden Ränder in Y - Richtung verfügen sowohl über eine konstante Temperatur als auch eine konstante Piezometerhöhe
 (--> IFLAG, JFLAG)
- > IBZ optionale Eingabe von konstanten Randbedingungen in Z - Richtung
 0 => die beiden Ränder in Z - Richtung sind sowohl wasser- als auch energieundurchlässig
 1 => die beiden Ränder in Z - Richtung behalten eine

- 2 => konstante Temperatur, sind aber wasserundurchlässig
 die beiden Ränder in Z - Richtung sind
 energieundurchlässig, verfügen aber über eine konstante
 Piezometerhöhe
- 3 => die beiden Ränder in Z - Richtung verfügen sowohl über
 eine konstante Temperatur als auch eine konstante
 Piezometerhöhe
 (--> IFLAG, JFLAG)
- > IDATOP optionales Einlesen von zeitlich veränderlichen Daten
 0 => es werden keinen Daten angefordert
 1 => es werden in Abständen von LISDAT - Zeitschritten
 Daten von der Datei DATOP angefordert, die Berech-
 nung wird bei Erreichen des Dateiendes von DATOP
 abgebrochen
- > IDIF bestimmt die Art der Problemstellung
 0 => reines Konduktionsproblem
 1 => Diffusions- und Konvektionsproblem
- >> IDSTOP bestimmt, ob bei Wärmetransportberechnungen trotz verletzter
 Diskretisierungsvorschriften die Berechnung fortgeführt werden
 soll; COURANT - Kriterium zur Reduzierung der numerischen
 Dispersion;
 0 => Diskretisierungsvorschriften werden nicht überprüft
 1 => Diskretisierungsvorschriften werden nur überprüft
 2 => die Berechnung wird ggf. vorzeitig abgebrochen
- > IDRU spezifiziert die Abstände in Zeitschritten, in denen Ergebnisse
 ausgegeben werden und eine Energiebilanzierung durchgeführt
 werden soll
- >> IDZTST ignoriert oder aktiviert die Zeitschrittanpassung bei Wärmetrans-
 portberechnungen; Stabilitätskriterium für das Explizit-Schema
 0 => Zeitschrittlängenüberprüfung wird übersprungen
 1 => es wird nur eine Zeitschrittlängenüberprüfung
 durchgeführt
 2 => es wird eine Zeitschrittlängenüberprüfung durchgeführt,
 eventuell die Zeitschrittlänge verkürzt
- > IECHO optionales Bildschirmprotokoll; diese Option dient vor allem dazu,
 während der Testphase den Programmablauf verfolgen zu können;
 ferner lassen sich einige Rechner nur während der Bildschirm-
 ausgabe stoppen;

- 0 => es wird nur die abgelaufene Zeit angezeigt
 1 => es wird ein Bildschirmprotokoll geführt
- IEN1 Integer - Variable, vor dem ersten Aufruf von Subroutine ENERGI
 1, sonst 0
- >> IENER optionale Energiebilanzierung
 0 => es wird keine Energiebilanzierung durchgeführt
 1 => es wird nach jeder Ergebnisausgabe eine Energiebilanz
 aufgestellt
- > IEIN Länge der Wärmeentzugsperioden in Zeitschritten
- >> IFLAG Verschlüsselung der Knoten - Temperaturberechnung;
 Dim.(IX, IY, IZ)
 1 => variable Temperatur
 2 => konstante Temperatur
 3 => Temperatur wird aus Wärmefluß berechnet
 4 => Wärmefluß wird aus dem Wärmeübergangskoeffizienten
 und der Temperatur des umgebenden Mediums berechnet
 5 => konstante Temperatur, es wird jedoch bei aktivierter
 Energiebilanzierung der Wärmefluß zu den Nachbarzellen
 berechnet(--> IENER)
 (3 & 4 noch nicht verfügbar)
- >> IFROST Berücksichtigung des Phasenwechsels von Wasser
 0 => Phasenwechsel wird ignoriert
 >0 => Phasenwechsel wird überprüft:
 1 => Gefrierpunkt von Wasser konst. 0 °C
 2 => Gefrierpunkt von Wasser wird in Subroutine TAUPKT in
 Abhängigkeit vom herrschenden Druck berechnet (noch
 nicht verfügbar)
 3 => Gefrierpunkt von Wasser wird von der Eingabedatei
 angefordert
- > IGWF legt fest, ob das Strömungsfeld berechnet werden soll.
 0 => Strömungsfeld wird vorgegeben
 1 => Strömungsfeld soll berechnet werden
 2 => ausschließlich Strömungsfeld wird berechnet werden
- > ILOOP maximale Anzahl der Schleifendurchläufe in den Subroutinen
 ITERAT und ITERAH, bzw. IADITP und IADIST

- IMAX maximaler Wert von IX, IY und IZ; wird für die Dimensionierung der Felder AA, BB, CC, DD und UU des tridiagonalen Matrizenalgorithmus benötigt.
- > IMSBIN bestimmt das Ausgabeformat der Daten für den TRADIKON-Postprozessor; diese Option kann nur auf PC's mit dem Betriebssystem MS-DOS genutzt werden.
 0 => es werden ASCII - Daten ausgegeben
 1 => es werden Binärdaten ausgegeben
- > INPOP Art der Eingabedaten
 0 => Daten werden für jede Schicht eingegeben
 1 => Daten werden für jeden Knoten eingegeben
- > IOBS Ausgabe von Ergebnissen auf die sequentielle Datei OBSFIL
 0 => es werden keine Ergebnisse ausgegeben
 >0 => es werden nach jedem Zeitschritt die Temperaturen der mit IXOBS, IYOBS und IZOBS festgelegten Knoten auf die Datei OBSFIL geschrieben; IOBS ist zugleich die Anzahl der Beobachtungsknoten
- IP1 programminterne Dimensionierungsoption:
 0 => Felder zur Temperaturberechnung werden nicht dimensioniert; IQ1 muß (!) auf 1 gesetzt werden
 1 => Felder zur Temperaturberechnung werden mit den Dimensionen (JX, JY, JZ) dimensioniert; IQ1 muß (!) auf 0 gesetzt werden
- IP2 programminterne Dimensionierungsoption:
 0 => Felder zur Strömungsberechnung werden nicht dimensioniert; IQ2 muß (!) auf 1 gesetzt werden
 1 => Felder zur Temperaturberechnung werden mit den Dimensionen (KX, KY, KZ) dimensioniert; IQ2 muß (!) auf 0 gesetzt werden

Anm: Die programminternen Dimensionierungsoptionen IP1, IP2, IQ1 und IQ2 können dazu benutzt werden, die Speicherplatzbedarf des Programms erheblich für den Fall zu reduzieren, falls z.B. nur reine Konduktionsprobleme oder nur Strömung simuliert werden sollen. Dadurch erhöht sich in der Regel auch die Rechengeschwindigkeit. Ferner kann dadurch auf Rechnern mit eingeschränktem Speicherplatz, die nicht die Möglichkeit der virtuellen Speicheradressierung verfügen, jedes Problem für sich rigoros diskretisiert

werden. Nicht benötigte Felder werden jeweils nur mit einem Feldelement dimensioniert z.B. DIMENSION T(1,1,1).

- >> IPECL Berechnungsschema der Diffusions-Konvektions- Wichtungsfunktion
 - 0 => Default (Exponential - Schema)
 - 1 => Zentrale Differenzen
 - 2 => Upwind - Schema
 - 3 => Hybrid - Schema
 - 4 => Power - Law Schema
 - 5 => Exponential - Schema

- > IPROX Knotenreihe des zu druckenden Y-Z - Schnittes
 - 0 => unterdrückt die Ausgabe dieses Schnittes

- > IPROY Knotenreihe des zu druckenden X-Z - Schnittes
 - 0 => unterdrückt die Ausgabe dieses Schnittes

- > IPROZ Knotenreihe des zu druckenden X-Y - Schnittes
 - 0 => unterdrückt die Ausgabe dieses Schnittes

- IQ1 programminterne Dimensionierungsoption; vergl. IP1

- IQ2 programminterne Dimensionierungsoption; vergl. IP2

- > ISOR optionales Testen des OMEGA-Wertes von SOR/SUR
 - 0 => es wird kein Test durchgeführt
 - 1 => es können nach Ausführung des ersten Zeitschrittes neue Werte für ALFATP und OMGT eingegeben werden (--> ALFATP, OMGT); Subroutine ITERAT startet darauf wieder mit den alten Temperaturdaten

- > ISTLAE Zeitschrittlänge in ZFLAG - Einheiten (--> ZFLAG)

- > ISTOP optionales Abbruchkriterium
 - 0 => Berechnung wird bis Ablauf von IZEIT fortgesetzt
 - 1 => Berechnung wird abgebrochen, wenn für den letzten Zeitschritt nur eine Iteration benötigt wurde, und der dabei aufgetretene Berechnungs fehler kleiner FELIT war

- > IUFL Umwandlungs - Flag für Zusatzinformationen; Dim.(IWERT); der Flag verweist auf den Feldbereich, für den die Zusatzinformation gelten soll.
 - 0 => nur die Indizes (IXX, IYY, IZZ

- 1 => die Knotenreihe (1 bis NX, IYY, IZZ)
 - 2 => die Knotenreihe (IXX, 1 bis NY, IZZ)
 - 3 => die Knotenreihe (IXX, IYY, 1 bis NZ)
 - 4 => die Knotenebene (1 bis NX, 1 bis NY, IZZ)
 - 5 => die Knotenebene (1 bis NX, IYY, 1 bis NZ)
 - 6 => die Knotenebene (IXX, 1 bis NY, 1 bis NZ)
 - 7 => alle Knoten (1 bis NX, 1 bis NY, 1 bis NZ)
- > IVIS optionale Berechnung der Fluidviskosität
- 0 => das Fluid verfügt über eine konstante Viskosität
 - 1 => die Viskosität wird in Abhängigkeit der Temperatur nach jedem Zeitschritt neu berechnet
- > IVOLDF bestimmt die Lage der Knoten in den Zellen
- 0 => Knoten liegen im geometrischen Zentrum der Zellen; es werden die Zellenkantenlängen eingelesen; (vergl. DELTAX, DELTAY und DELTAZ)
 - 1 => Zellengrenzflächen liegen zentriert zwischen den Knoten; es werden die Knotenabstände eingelesen; (vergl. DIFX, DIFY und DIFZ)
- > IWARMS legt die Art des Simulationsstartes fest
- 0 => "Kaltstart"; es werden keine Daten einer früheren Simulation gelesen
 - 1 => "Warmstart"; es wird das Temperaturfeld einer vorausgegangenen Simulation als Ausgangsbasis einer neuen Berechnung genutzt; dazu müssen die entsprechenden Werte auf der sequentiellen Datei TRADWS bereitgestellt werden, sonst bricht TRADIKON-3D die Simulation vorzeitig ab
- IWERT maximal mögliche Anzahl der Zusatzinformationen
- > IWCP Festlegung der physikalischen Eigenschaften des Fluids (Wasser)
- 0 => alle physikalischen Eigenschaften werden auf 0 gesetzt
 - 1 => es werden folgende Eigenschaften des Fluids angenommen:
 - CPW = 4200 J/kgK
 - CPE = 2114 J/kgK
 - CPL = 0 J/kgK
 - WLW = 0.598 W/mK
 - WLE = 2.390 W/mK
 - WLL = 0.025 W/mK
 - RHOW = 1000 kg/m³
 - RHOE = 917 kg/m³

RHOL = 0 kg/m³
 SENTHA = 3.336 · 10⁵ J/kg
 DTW = 0.01 K
 DWAN = 0.001

- 2 => alle physikalischen Eigenschaften des Fluids werden von der Eingabedatei angefordert
 - 3 => wie 2, Wärmeleitfähigkeit des Gesteins ist temperaturabhängig (polynomische Approximation) (--> DTWL)
 - 4 => wie 2, Wärmeleitfähigkeit des Gesteins ist temperaturabhängig (exponentielle Approximation)
 - 5 => wie 2, Wärmeleitfähigkeit des Gesteins ist druck- und temperaturabhängig (reziproke Approximation nach HUENGES et al. (1989) (--> DTWL)
- > IWSDAT optionale Ausgabe des jeweils aktuellen Temperaturfeldes auf der sequentiellen Datei TRADWS
- 0 => die Ausgabe wird unterdrückt
 - 1 => die Datei TRADWS wird parallel zu jedem Ergebnisausdruck mit dem aktuellen Temperaturfeld überschrieben; die Datei kann bei einer späteren Simulation als Eingabedatei dienen (--> IWARMS = 1)
- IX maximal erlaubte Anzahl der Knoten in X - Richtung
- > IXISO Y - Z Schnitt für Isolinienplots ignorieren oder aktivieren
- 0 => es wird kein Y - Z Schnitt ausgegeben
 - 1 => es wird parallel zu jedem Ergebnisausdruck ein Y - Z Schnitt der Knotenreihe IPROX auf der Datei TRADYZ ausgegeben; diese Daten können dann nach Ablauf der Simulation mit dem TRADIKON-3D - Postprozessor in Isolinienform dargestellt werden
- > IXOBS Indexfeld der I - Indizes der Observationsknoten; Dim.(10)
- > IXX Indexfeld der I - Indizes der Zusatzinformationen; Dim.(IWERT)
- IY maximal erlaubte Anzahl der Knoten in Y - Richtung
- > IYISO X - Z Schnitt für Isolinienplots ignorieren oder aktivieren
- 0 => es wird kein X - Z Schnitt ausgegeben
 - 1 => es wird parallel zu jedem Ergebnisausdruck ein X - Z Schnitt der Knotenreihe IPROY auf der Datei TRADXZ

ausgegeben; diese Daten können dann nach Ablauf der Simulation mit dem TRADIKON-3D - Postprozessor in Isolinienform dargestellt werden

- > IYOBS Indexfeld der J - Indizes der Observationsknoten; Dim.(10)
- > IYY Indexfeld der J - Indizes der Zusatzinformationen Dim.(IWERT)
- IZ maximal erlaubte Anzahl der Knoten in Z - Richtung
- > IZISO X - Y Schnitt für Isolinienplots ignorieren oder aktivieren
 - 0 => es wird kein X - Y Schnitt ausgegeben
 - 1 => es wird parallel zu jedem Ergebnisausdruck ein X - Y Schnitt der Knotenreihe IPROZ auf der Datei TRADXY ausgegeben; diese Daten können dann nach Ablauf der Simulation mit dem TRADIKON-3D - Postprozessor in Isolinienform dargestellt werden
- > IZOBS Indexfeld der K - Indizes der Observationsknoten; Dim.(10)
- > IZZ Indexfeld der K - Indizes der Zusatzinformationen Dim.(IWERT)
- > IZEIT Gesamtanzahl der zu berechnenden Zeitschritte
- J aktueller Index in Y - Richtung
- >> JFLAG Verschlüsselung der Knoten: Strömungsberechnung
 - 1 => variable Piezometerhöhe
 - 2 => fest vorgegebene Piezometerhöhe
- K aktueller Index in Z - Richtung
- KD programminterne Dimensionierung; gibt an, wieviel Nachbarzellen eine beliebige Zelle maximal haben kann; bei dreidimensionalen Berechnungen 6, bei zweidimensionalen 4 und bei eindimensionalen maximal 2 Nachbarzellen; vergl. IP1, IP2 etc.
- LUMW Anzahl der umzuwandelnden Zusatzdaten
- > MODEL Wahl der Porenraumverteilung zur Berechnung der effektiven Wärmeleitfähigkeit eines Gesteins;
 - 1 => die Poren liegen als isolierte Hohlräume vor
 - 2 => die Poren haben untereinander eine gute Verbindung
 - 3 => zur Berechnung der effektiven Wärmeleitfähigkeit wird das gewichtete, arithmetrische Mittel benutzt

	NIT	Anzahl der benötigten Iterationen für den letzten Zeitschritt
-->	NX	Anzahl der Knoten in X - Richtung
	NX1	NX - 1
-->	NY	Anzahl der Knoten in Y - Richtung
	NY1	NY - 1
-->	NZ	Anzahl der Knoten in Z - Richtung
	NZ1	NZ - 1
-->	OBSFIL	optionale Ausgabedatei für die mit IXOBS, IYOBS und IZOBS vereinbarten Beobachtungsknoten; die maximale Anzahl der Knoten beträgt 10; es werden nach jedem Zeitschritt die Temperaturen dieser Knoten ausgegeben; die Datei wird nur angelegt, wenn IOBS > 0
-->	OMGH	Bestimmung des Iterationsverfahrens bei der Strömungsberechnung OMGH = -1 => IADI - Verfahren 0 > OMGH => IADI - Verfahren mit Relaxation 0 < OMGH = 1 => Gauss - Seidel - Verfahren OMGH > 1 => Successiv - Over - Relaxation OMGH < 1 => Successive - Under - Relaxation (--> ISOR, ALFAHY)
-->	OMGT	Bestimmung des Iterationsverfahrens bei der Temperatur berechnung OMGT = -1 => IADI - Verfahren 0 > OMGT => IADI - Verfahren mit Relaxation OMGT = 1 => Gauss - Seidel - Verfahren OMGT > 1 => Successiv - Over - Relaxation 0 < OMGT < 1 => Successive - Under - Relaxation (--> ISOR, ALFATP)
-->	PHIG	Gesamtporosität des Gesteins in %; Dim.(JX, JY, JZ)
-->	PHIN	Nutzporosität oder durchflußwirksame Porosität in %; Dim.(IX, IY, IZ), Bei teilgespannten Verhältnissen einer Strömungssimulation wird bei Erreichen einer freien Oberfläche der Speicherkoeffizient SK durch PHIN ersetzt.
>>	Q	Entnahme- oder Zugabemengen in m ³ /s; Dim.(KX, KY, KZ)

- > RHOG Dichte des Gesteins (Ein- und Ausgabe in g/cm³, intern in kg/m³); Dim.(JX, JY, JZ)
- > SATU Wassersättigung der Poren in %; Dim.(IZ)
- SENTHA Schmelzenthalpie des Fluids (Wasser) in J/kg
- >> STMP Schmelztemperatur von Wasser; wird je nach Wert von IFROST gesetzt, berechnet oder von der Eingabedatei angefordert; Dim.(JZ)
- >> SK spezifischer Speicherkoeffizient; wird nur benötigt, wenn IGWF = 1; Dim.(KX, KY, KZ)
- SUMSEK abgelaufene Zeit in Sekunden
- >> T aktuelle Temperatur an Knoten; Dim.(JX, JY, JZ); (bei der Ein- und Ausgabe in °C, intern Kelvin)
- T0 Knotentemperaturen nach dem letzten Zeitschritt (Kelvin); Dim.(JX, JY, JZ)
- > TITEL Problemtitel, maximal 80 Zeichen
- >> TTX hydraulische Leitfähigkeit in X - Richtung; wird nur benötigt, wenn IGWF = 1; Dim. (KX, KY, KZ)
- >> TTY hydraulische Leitfähigkeit in Y - Richtung; wird nur benötigt, wenn IGWF = 1; Dim. (KX, KY, KZ)
- >> TTZ hydraulische Leitfähigkeit in Z - Richtung; wird nur benötigt, wenn IGWF = 1; Dim. (KX, KY, KZ)
- UU Feld der unabhängigen Variablen (Temperaturen oder Piezometerhöhen) des tridiagonalen Matrizenalgorithmus; Dim. (IMAX)
- VX Grundwasserfließgeschwindigkeit in X - Richtung; wird aus GWV und WINKEL berechnet, wenn IGWF = 0; wird aus dem hydraulischen Gradienten und der Grenzflächenleitfähigkeit berechnet, wenn IGWF = 1; Dim.(IX, IY, IZ)
- VY Grundwasserfließgeschwindigkeit in Y - Richtung wird aus GWV und WINKEL berechnet, wenn IGWF = 0, wird aus dem hydraulischen Gradienten und der Grenzflächenleitfähigkeit berechnet, wenn IGWF = 1; Dim.(IX, IY, IZ)

>> VZ Sickerwasserfließgeschwindigkeit in Z - Richtung; wird nur benötigt, wenn IDIF = 1 und IGWF = 0; wird aus dem hydraulischen Gradienten berechnet, wenn IGWF = 1; Dim.(IX, IY, IZ)

>> WERT Wert des mit FLAG verschlüsselten Zusatzwertes; diese Eingabeoption wird nur bei kompakten Eingabedaten (INPOP = 0) und beim Einlesen zeitabhängiger Variablen benötigt; Dim.(IWERT)

Beispiel 1: FLAG(L) = CPGE

WERT(L) = 854.

IXX(L) = 7

IYY(L) = 4

IZZ(L) = 6

IUFL(L) = 0

==> CPG(7, 4, 6) = 854. J/kgK

Beispiel 2: FLAG(L) = GWNB

WERT(L) = 3E-9

IXX(L) = 0

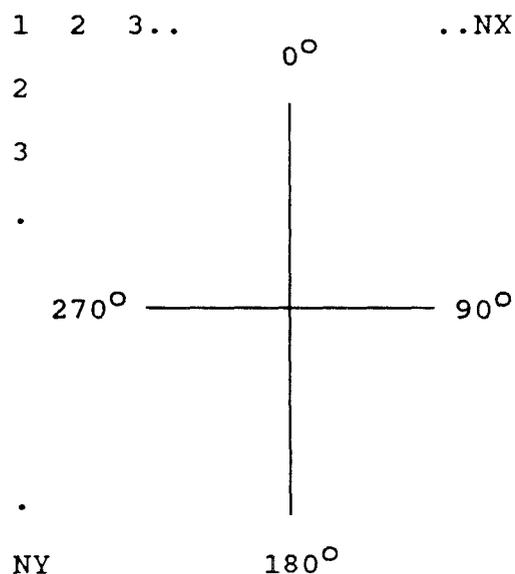
IYY(L) = 0

IZZ(L) = 1

IUFL(L) = 4

==> Grundwasserneubildung in der Schicht 1: $3 \cdot 10^{-9} \text{ m}^3/\text{s} \cdot \text{m}^2$
 (--> FLAG, IXX, IYY, IZZ, IUFL)

=> WINKEL Grundwasserfließrichtung in Grad relativ zur Nordrichtung des Benutzerkoordinatensystems; diese Eingabe wird nur bei Diffusions-Konvektions-Berechnungen mit vorgegebener Strömung benötigt, d.h., wenn IGWF = 0 und IDIF = 1:



- = > WLE Wärmeleitfähigkeit der festen Phase des Fluids (Eis);
(--> IWCP)
- > > WLKG Wärmeleitfähigkeit des Gesteins; Dim.(JX, JY, JZ)
- = > WLL Wärmeleitfähigkeit der gasförmigen Porenraumfüllung (Luft);
(--> IWCP)
- = > WLW Wärmeleitfähigkeit des Fluids (Wasser); (--> IWCP)
- WTI Wärmeleitfähigkeit, Wärmediffusion oder hydraulische Leitfähigkeit
an den Zellengrenzflächen (I+ 1/2, J, K), (I, J+ 1/2, K) und
(I, J, K+ 1/2); Dim.(IX, IY, IZ, KD/2)
- XKOR Absolutkoordinaten der Knoten in X - Richtung; Dim.(IX)
- YKOR Absolutkoordinaten der Knoten in Y - Richtung; Dim.(IY)
- > ZFLAG Einheit der mit ISTLAE - spezifizierten Zeitschrittlänge
 - SEK => Sekunden
 - MIN => Minuten
 - STU => Stunden
 - TAG => Tage
- ZIGNO programminterner Weglasswert für zusätzlich eingelesene Daten;
durch Eingabe dieses Wertes (hier vereinbarungsgemäß -99.9) kann
man ein Umwandeln unterdrücken; diese Option ist vor allem bei
der Eingabe zeitlich variabler Randbedingungen sinnvoll;
(--> WERT, Subroutine UMWAND)
- > Z 'unendlich' große Zahl; dient zur Besetzung des Speicher-
koeffizienten bzw. der Wärmekapazität von Festpotential- oder
-temperaturknoten; Voreinstellung 10^{30} , programmintern hat
hat Z die Bezeichnung ZZZZ
- ZKOR Absolutkoordinaten der Knoten in Z - Richtung; Dim.(IZ)

Beschreibung des kurzen Dateninputs:

Kartenr.

- 1 Problemtitel (maximal 80 Zeichen), Format A80
- 2 Formatkarte für Karte 3, Format A20
- 3 IXISO, IYISO, IZISO, IMSBIN
- 4 Formatkarte für Karte 5, Format A20
- 5 IOBS, IDATOP, LISDAT, XYZFIL, AUS, DATOP
- 6 Formatkarte für Karte 7, Format A20
- 7 IZEIT, ILOOP, IEIN, IAUSOP, INPOP, IDRU, ISTLAE, ZFLAG, DZF, FEHLT, Z
- 8 Formatkarte für Karte 9, Format A20
- 9 FLURAB, MODEL, IENER, ISTOP, IDZTST, IWARMS, IWSDAT, IDSTOP
- 10 Formatkarte für Karte 11, Format A20
- 11 ALFATP, OMTG, IVOLDF, IPECL, IDIF, IFROST, IECHO, IALTER, IWCP, IOMGH
- 11a Formatkarte für Karte 11b, Format A20
- 11b CPW, CPE, CPL, WLW, WLE, WLL, RHOW, RHOE, RHOL, SENTHA, DTW, DWAN
11a und 11b werden nur angefordert, wenn IWCP = 2
- 11c Formatkarte für Karte 11d, Format A20
- 11d DTWL(1), DTWL(2), DTWL(3)
11c und 11d werden nur angefordert, wenn IWCP = > 3
- 12 Formatkarte für Karte 13, Format A20
- 13 BASIS, HOENN, ALFAHY, OMTG, FEHLH, IGWF, IVIS, IWRO
- 14 Formatkarte für Karte 15, Format A20
- 15 IPROX, IPROY, IPROZ
- 15a Formatkarte für Karte 15b, Format A20
- 15b IXOBS, IYOBS, IZOBS; Karten 15a und 15b werden übersprungen, wenn IOBS=0
- 16 Formatkarte für Karte 17, Format A20

- 17 NX, NY, NZ
- 17a Formatkarte für Karte 17b, Format A20
- 17b STMP(NZ); 17a und 17b werden nur angefordert, wenn IFROST = 3
- 17c Formatkarte für Karte 17d, Format A20
- 17d IFRO(NZ); 17c und 17d werden nur angefordert, wenn IFROST > 0
- 17e Formatkarte für die Karten 17f ff., Format A20;
wird nur angefordert, wenn IFROST > 0
- 17f FRO(K,1), FRO(K,2), FRO(K,3);
17f wird nur angefordert, wenn IFRO(K) = 1
- 18 Formatkarte für Karte 19, Format A20
- 19 IBX, IBY, IBZ, ANISOX, ANISOY, ANISOZ (ANISO(3))
- 20 Formatkarte für Karte 21, Format A20
- 21 DELTAX(NX) wenn IVOLDF = 0 oder DIFX(NX-1) wenn IVOLDF = 1
- 22 Formatkarte für Karte 23, Format A20
- 23 DELTAY(NY) wenn IVOLDF = 0 oder DIFY(NY-1) wenn IVOLDF = 1
- 24 Formatkarte für Karte 25, Format A20
- 25 DELTAZ(NZ) wenn IVOLDF = 0 oder DIFZ(NZ-1) wenn IVOLDF = 1
- 26 Formatkarte für Karte 27, Format A20
- 27a T, CPG, RHOG, WLKG, PHIG, PHIN, SATU, SC, WINKEL, GWV, VZ, IFLAG
(wenn IGWF = 0 d.h. das Strömungsfeld wird vorgegeben)
- 27b T, CPG, RHOG, WLKG, PHIG, PHIN, SATU, SC, Q, SK, H, TTX, TTY, TTZ,
SATU, IFLAG, JFLAG
(wenn IGWF = 1 d.h. das Strömungsfeld soll berechnet werden)
- 27c H, TTX, TTY, TTZ, Q, SK, PHIN, JFLAG
(wenn IGWF = 2 d.h. ausschließlich das Strömungsfeld soll berechnet werden)
- 28 Formatkarte für Karte 29, Format A20
- 29 FLAG, WERT, IXX, IYY, IZZ, IUFL

Anhang 3:

Programm-Listing

C&TRAD3D

```
C *****
C      Version 11061987          Programm TRADIKON-3D          Rev. 14/04/1989 *
C *****
C
C      -----
C      TRADIKON-3D
C      -----
C
C      ein stroemungsgekoppeltes, dreidimensionales finite
C      Differenzen Modell zur Berechnung des Waermetransports
C      durch Diffusion und Konvektion in gepannt/ungespannt-
C      ten, anisotropen, heterogenen Porengrundwasserleitern
C
C      Dipl.-Geol. Dirk Brehm
C      Institut fuer Angewandte Geowissenschaften
C      der Justus-Liebig-Universitaet Giessen
C      Diezstrasse 15
C      D-6300 Giessen
C      Tel. 0641/702-8246
C      EARN/BITNET: BrehmDi@DGIHRZ01
C *****
C
```

```
PROGRAM TRAD3D
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
PARAMETER (IWERT=400)
IMPLICIT DOUBLE PRECISION (E)
CHARACTER TITEL*80,FLAG*4
CHARACTER FORMOU*30,FORTXT*48
DOUBLE PRECISION T,T0,STMP
DOUBLE PRECISION UU,AA,BB,CC,DD
INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
INTEGER*2 IXX,IYY,IZZ,IUFL
DIMENSION DELTAX (IX),DELTAY (IY),DELTAZ (IZ),DIFX (IX),DIFY (IY),
&          DIFZ (IZ),XKOR (IX),YKOR (IY),ZKOR (IZ),DIVX (IX,2),
&          DIVY (IY,2),DIVZ (IZ,2),FX (IX),FY (IY),FZ (IZ),
&          DXYZ (IX,IY,IZ)
DIMENSION IFLAG (IX,IY,IZ),JFLAG (KX,KY,KZ),IFLUS (JX,JY,JZ),IFRO (JZ)
DIMENSION UU (IMAX),AA (IMAX),BB (IMAX),CC (IMAX),DD (IMAX)
DIMENSION T (JX,JY,JZ),T0 (JX,JY,JZ),STMP (JZ)
DIMENSION H (KX,KY,KZ),HO (KX,KY,KZ)
DIMENSION CPG (JX,JY,JZ),WLKG (JX,JY,JZ),WTI (IX,IY,IZ,KD2),
&          RHOG (JX,JY,JZ),PHIG (JX,JY,JZ),PHIN (IX,IY,IZ),
&          SATU (IZ),VX (IX,IY,IZ),VY (IX,IY,IZ),VZ (IX,IY,IZ),ANISO (3)
DIMENSION SC (JX,JY,JZ),SP (JX,JY,JZ),FRO (JZ,5)
DIMENSION TTX (KX,KY,KZ),TTY (KX,KY,KZ),TTZ (KX,KY,KZ),
&          SK (KX,KY,KZ),Q (KX,KY,KZ)
DIMENSION AP1 (IX,IY,IZ),APO (IX,IY,IZ),ANB (IX,IY,IZ,KD),
&          FNB (KD),DNB (KD),PECL (KD),B (IX,IY,IZ)
DIMENSION WERT (IWERT),IXX (IWERT),IYY (IWERT),IZZ (IWERT),
```

```

&          FLAG(IWERT), IUFL(IWERT)
DIMENSION IXOBS(10), IYOBS(10), IZOBS(10)
DIMENSION INDX(6), INDY(6), INDZ(6), IADD(4), JADD(4), KADD(4)
DIMENSION FORMOU(8), FORTXT(8)
COMMON /EKAL/EALT, EANF, ESUM
COMMON /ENER/SC, SP, FRO
COMMON /FORM/FORMOU, FORTXT
COMMON /GITT/DELTAX, DELTAY, DELTAZ, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,
&          DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ
COMMON /IADI/UU, AA, BB, CC, DD
COMMON /IFLD/IFLAG, JFLAG, IFLUS, IFRO
COMMON /INCB/INDX, INDY, INDZ, IADD, JADD, KADD
COMMON /INF1/IZEIT, IPECL, IVOLDF, IAUSOP, ILOOP, IEIN, INPOP, IALTER
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
COMMON /INF3/LUMW, IENER, ISTOP, IDATOP, LISDAT, IDZTST
COMMON /INF4/IOBS, IXOBS, IYOBS, IZOBS, IWARMS, IWSDAT, IDSTOP, IGWF
COMMON /INF5/IVIS, IWRO
COMMON /IPRO/IPROX, IPROY, IPROZ
COMMON /ISOP/IXISO, IYISO, IZISO, ISO1, IMSBIN
COMMON /KOEK/AP1, APO, ANB, FNB, DNB, PECL, B
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /PAR1/DZEIT, DZMF, DZF, FLURAB, ALFATP, FEHLT, FEHLER, OMGT, SUMSEK
COMMON /PAR2/BASIS, HOENN, ALFAHY, OMGH, FEHLH, ZZZZ, ZIGNO
COMMON /PAR3/CPW, CPE, CPL, WLW, WLE, WLL, RHOW, RHOE, RHOL, SENTHA
COMMON /PAR4/DTW, DTW2, DWAN, DTWL(3)
COMMON /PHYS/CPG, WLKG, ANISO, WTI, RHOG, PHIG, PHIN, SATU, VX, VY, VZ
COMMON /TEMP/T, TO, STMP
COMMON /TRNS/TTX, TTY, TTZ, Q, SK
COMMON /WPOT/H, HO
COMMON /ZYG1/WERT, IXX, IYY, IZZ, IUFL
COMMON /ZYG2/FLAG
EALT   = 0.
EANF   = 0.
IAN    = -1
IEINS  = 0
IEN1   = 1
ISO1   = 0
IUNIT  = 1
IZAEHL = 0
IZUSD  = 0
LUMW   = 0
NIT    = 0
NPROF  = 1
SUMSEK = 0.
ZZZZ   = 1E30
ZIGNO  = -99.9

```

```

C
C *** Subroutine INFORM Einlesen des Problemtitels, Timesteps etc. *****
C
C      CALL INFORM
C
C      IF (INPOP.GT.0) CALL DATEIN
C
C *** Subroutine KURZIN Einlesen der Daten - kurzer Input *****

```

```

C
  IF (INPOP.EQ.0) THEN
    CALL KURZIN
C
C *** Subroutine UMWAND Umwandeln der Zusatzinformationen *****
C
    IF (LUMW.GE.1) THEN
      DO 40 L = 1, LUMW
        CALL UMWAND (L,IAN)
    40    CONTINUE
      END IF
    END IF
C
C *** Subroutine ZUSDAT Einlesen zusätzlicher, zeitabhängiger Daten **
C
  IF (IDATOP.EQ.1) CALL ZUSDAT (IEINS)
C
C *** Subroutine WARMIN liest das Ergebnis der vorherigen Simulation ***
C
  IF (IWARMS.EQ.1) CALL WARMIN
C
C *** Subroutine KOPIER Kopieren der T oder H-Werte in ein anderes Feld
C
  IF (IGWF.LE.1) CALL KOPIER (1)
  IF (IGWF.GE.1) CALL KOPIER (2)
C
C *** Subroutine VDRUCK Drucken des Grundwasserfließgeschwindigkeiten *
C
  IF (IDIF.EQ.1.AND.IGWF.EQ.0) CALL VDRUCK
C
C *** Subroutine INFINT Berechnung der Interface-Wichtungsfaktoren *****
C
  CALL INFINT
C
C *** Subroutine DATAUS Ausdrucken der Eingangsdaten *****
C
  IF (IAUSOP.EQ.3.AND.IGWF.LE.1) CALL DATAUS
C
C *** Subroutine TAUPKT Gefrierpunktberechnung von Wasser *****
C
  IF (IGWF.LE.1.AND.IFROST.LT.3) CALL TAUPKT (STMP,ZKOR,1,1)
C
C *** Subroutine DZTEST Überprüfung der Zeitschrittlänge *****
C
  IF (IDATOP.EQ.1.AND.IGWF.LE.1.AND.IDZTST.NE.0) THEN
    CALL DZTEST (LISDAT)
    IEIN = IZEIT
  END IF
  IF (IDATOP.EQ.0.AND.IGWF.LE.1.AND.IDZTST.NE.0) CALL DZTEST (IEIN)
C
C *** Subroutine DKTEST Überprüfung der Diskretisierungsvorschriften *
C
  IF (IDIF.EQ.1.AND.IGWF.EQ.0.AND.IDSTOP.NE.0) CALL DKTEST
C

```

```

C *** Subroutine FLUSFL Festlegung der Flowflags *****
C
  IF (IDIF.EQ.1.AND.IAUSOP.EQ.2.AND.IGWF.LE.1) THEN
    CALL FLUSFL (1,1,1,NX,NY,NZ)
  END IF
C
C *** Subroutine PROFIL Drucken der Eingangsdaten in Profilschnitten ***
C
  IF (IAUSOP.EQ.2.AND.IGWF.LE.1) THEN
    CALL PROFIL (CPG,IFLAG,NPROF)
    CALL PROFIL (WLKG,IFLAG,NPROF)
    CALL PROFIL (RHOG,IFLAG,NPROF)
    CALL PROFIL (PHIG,IFLAG,NPROF)
    CALL PROFIL (PHIN,IFLAG,NPROF)
    CALL PROFIL (SC,IFLAG,NPROF)
    CALL PROFIL (SC,IFLAG,NPROF)
    CALL PROFIL (SC,IFLUS,NPROF)
  END IF
C
C *** Subroutine TMPDRU Drucken der berechneten Temperaturprofile *****
C
  IF (IGWF.LE.1) CALL TMPDRU
  IF (IGWF.GE.1) CALL PIEDRU
C
C *** falls das Stroemungsfeld als konstant angenommen wird ... *****
C
  IF (IGWF.EQ.0) THEN
C
C *** Subroutine AUFNUL Initialisieren der Koeffizienten - Felder *****
C
    CALL AUFNUL
C
C *** Subroutine KOEFFT Berechnung der Nachbarkoeffizienten fuer T *****
C
    CALL KOEFFT (1,1,1,NX,NY,NZ)
C
C *** Subroutine QKOEFT Berechnung der Quelltermkoeffizienten *****
C
    CALL QKOEFT
C
C *** Subroutine ENERGI Berechnung der Waermeenergie in den Zellen *****
C
    IMAL = IDRU
    IF (IENER.EQ.1) THEN
      CALL ENERGI
C
C *** Subroutine WFLUSS Berechnung des Waermeflusses zum Waermetauscher
C
    CALL WFLUSS (IMAL)
    END IF
C
  END IF
  IZYKL = IZEIT / IEIN
  IAN = 1

```

```

C
  IF (IGWF.EQ.2) THEN
    CALL AUFNUL
    CALL KOEFFH
    CALL QKOEFFH
  END IF

C
  DO 10 LOOP1 = 1, IZYKL
  DO 20 LOOP2 = 1, IEIN

C
C *** Subroutine OBSDAT Temperaturen auf sequentiellen File schreiben **
C
  IF (IOBS.GT.0) CALL OBSDAT
  IZAEHL = IZAEHL + 1
  IF (IDATOP.EQ.1.AND.IZUSD.EQ.LISDAT) THEN
    CALL ZUSDAT (IEINS)
    IZUSD = 0
    IF (IGWF.EQ.2) THEN
      CALL AUFNUL
      CALL KOEFFH
      CALL QKOEFFH
    END IF
    IF (IGWF.EQ.0) THEN
      CALL AUFNUL
      CALL KOEFFT (1,1,1,NX,NY,NZ)
      CALL QKOEFT
    END IF
  END IF
  IF (IDATOP.EQ.1) IZUSD = IZUSD + 1

C
C *** falls das Stroemungsfeld berechnet werden soll ... *****
C
  IF (IGWF.EQ.1) THEN
    CALL AUFNUL
    CALL KOEFFH
    CALL QKOEFFH
    IF (OMGH.GT.0.) CALL ITERAH
    IF (OMGH.LT.0.) CALL IADIST
    CALL FLIESS
    CALL KOPIER (2)
    CALL AUFNUL
    CALL KOEFFT (1,1,1,NX,NY,NZ)
    CALL QKOEFT
  END IF

C
C *** Subroutine ITERAT iterative Loesung des Temperaturfeldes *****
C
  IF (IGWF.EQ.2) THEN
    IF (OMGH.GT.0.) CALL ITERAH
    IF (OMGH.LT.0.) CALL IADIST
    CALL KOPIER (2)
  END IF

C
  IF (IGWF.LE.1) THEN

```

```
        IF (OMGT.GT.0.) CALL ITERAT
        IF (OMGT.LT.0.) CALL IADITP
        CALL KOPIER (1)
END IF
```

```
C
SUMSEK = SUMSEK + DZEIT
DZEIT = DZEIT * DZF
C
IF (IGWF.EQ.0) THEN
    IF (IFROST.GT.0) CALL KOEFFT (1,1,1,NX,NY,NZ)
    CALL QKOEFT
END IF
IF (IGWF.EQ.2) CALL QKOEFH
C
C *** Subroutine ZEITBE Ausgabe der abgelaufenen Zeit auf IUNIT *****
C
CALL ZEITBE (IUNIT)
C
C *** Ueberpruefung des Abbruchkriteriums wenn ISTOP = 1 *****
C
IF (ISTOP.EQ.1.AND.NIT.EQ.1.AND.FEHLER.LT.FEHLT) THEN
    IF (IGWF.LE.1) CALL TMPDRU
    IF (IGWF.GE.1) CALL PIEDRU
    IF (IWSDAT.EQ.1) CALL WARMOU
    CALL ISODRV
    IF (IENER.EQ.1) THEN
        CALL ENERGI
        CALL WFLUSS (IZAEHL)
    END IF
    IF (IOBS.GT.0) CALL OBSDAT
    WRITE (1,120) FEHLT
    WRITE (IAUS,120) FEHLT
    WRITE (1,130)
    WRITE (IAUS,130)
    STOP
END IF
C
IF (IZAEHL.EQ.IDRU) THEN
    IF (IGWF.LE.1) CALL TMPDRU
    IF (IGWF.GE.1) CALL PIEDRU
    IF (IWSDAT.EQ.1) CALL WARMOU
    CALL ISODRV
    IF (IENER.EQ.1) THEN
        CALL ENERGI
        CALL WFLUSS (IMAL)
    END IF
    IZAEHL = 0
END IF
20 CONTINUE
C
IF (IEIN.EQ.IZEIT) THEN
    IF (IOBS.GT.0) CALL OBSDAT
    WRITE (IAUS,140)
    WRITE (1,140)
```

```

        STOP
    END IF
    IAN = IABS(IAN - 1)
C
    DO 30 L = 1, LUMW
        CALL UMWAND (L, IAN)
30 CONTINUE
C
    IF (IZAEHL.NE.0) THEN
        IF (IGWF.LE.1) CALL TMPDRU
        IF (IGWF.GE.1) CALL PIEDRU
        IF (IWSDAT.EQ.1) CALL WARMOU
        CALL ISODRV
        IF (IENER.EQ.1) THEN
            CALL ENERGI
            CALL WFLUSS (IZAEHL)
        END IF
    END IF
C
    IF (IGWF.LE.1) CALL KOPIER (1)
    IF (IAN.EQ.0) WRITE (IAUS,100)
    IF (IAN.EQ.1) WRITE (IAUS,110)
    IF (IOBS.GT.0) CALL OBSDAT
10 CONTINUE
    WRITE (IAUS,140)
    WRITE (1,140)
    STOP
100 FORMAT(/,T16,' Energieentzug bzw. Energieeingabe gestoppt')
110 FORMAT(/,T16,' Energieentzug bzw. Energieeingabe gestartet')
120 FORMAT(/,T16,' Abbruchkriterium NIT = 1 und FEHLER < ',F8.6,
    & ' erfuehlt')
130 FORMAT(T16,' Berechnung vorzeitig abgebrochen')
140 FORMAT(T16,' TRADIKON-3D Simulation normal beendet')
    END
    BLOCK DATA FELDER
    CHARACTER FORMOU*30,FORTXT*48
    DIMENSION INDX(6),INDY(6),INDZ(6),IADD(4),JADD(4),KADD(4)
    DIMENSION FORMOU(8),FORTXT(8)
    COMMON /FORM/FORMOU,FORTXT
    COMMON /INCB/INDX,INDY,INDZ,IADD,JADD,KADD
    DATA INDX /-1,1,0,0,0,0/, INDY /0,0,-1,1,0,0/, INDZ /0,0,0,0,-1,1/
    DATA IADD /0,1,0,0/, JADD /0,0,1,0/, KADD /0,0,0,1/
    DATA FORMOU/
    &'(F6.2,10F6.0,/,9(6X,10F6.0,/))','(F6.2,10F6.2,/,9(6X,10F6.2,/))',
    &'(F6.2,10F6.2,/,9(6X,10F6.2,/))','(F6.2,10F6.2,/,9(6X,10F6.2,/))',
    &'(F6.2,10F6.2,/,9(6X,10F6.2,/))','(F6.2,10F6.2,/,9(6X,10F6.2,/))',
    &'(F6.2,10I6 ,/,9(6X,10I6 ,/))','(F6.2,10I6 ,/,9(6X,10I6 ,/))'/
    DATA FORTXT/'(/,' Waermekapazitaet des Gesteins:'' )',
    & '(/,' Waermeleitfaehigkeit des Gesteins:'' )',
    & '(/,' Dichte des Gesteins:'' )',
    & '(/,' Gesamtporositaet des Gesteins: (0 - 1)'' )',
    & '(/,' Nutzporositaet des Gesteins: (0 - 1)'' )',
    & '(/,' Waermeproduktion in Watt/cbm:'' )',
    & '(/,' Waermefluss - Flags:'' )',

```

```

&          '(/, ' Grundwasserfluss - Flags: '          )' /
END
C&AUFNUL
C *****
C      Version 12071987  Subroutine AUFNUL
C *****
C
C *** Subroutine AUFNUL Initialisieren der Koeffizienten - Felder *****
C
SUBROUTINE AUFNUL
PARAMETER (IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2)
PARAMETER (IP1=1, IP2=1, IQ1=0, IQ2=0)
PARAMETER (JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2, KY=IY*IP2+IQ2, KZ=IZ*IP2+IQ2)
DIMENSION CPG(JX, JY, JZ), WLKG(JX, JY, JZ), WTI (IX, IY, IZ, KD2),
&          RHOG(JX, JY, JZ), PHIG(JX, JY, JZ), PHIN (IX, IY, IZ),
&          SATU (IZ), VX (IX, IY, IZ), VY (IX, IY, IZ), VZ (IX, IY, IZ), ANISO (3)
DIMENSION AP1 (IX, IY, IZ), AP0 (IX, IY, IZ), ANB (IX, IY, IZ, KD),
&          FNB (KD), DNB (KD), PECL (KD), B (IX, IY, IZ)
COMMON /INF1/IZEIT, IPECL, IVOLDF, IAUSOP, ILOOP, IEIN, INPOP, IALTER
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
COMMON /INF3/LUMW, IENER, ISTOP, IDATOP, LISDAT, IDZTST
COMMON /KOEUF/AP1, AP0, ANB, FNB, DNB, PECL, B
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /PHYS/CPG, WLKG, ANISO, WTI, RHOG, PHIG, PHIN, SATU, VX, VY, VZ
C
C      IF (IECHO.GE.1) CALL ROUTIN ('AUFNUL')
C
DO 10 K = 1, NZ
DO 10 J = 1, NY
DO 10 I = 1, NX
    AP0 (I, J, K) = 0.
    AP1 (I, J, K) = 0.
    B (I, J, K) = 0.
    DO 20 L = 1, KD2
        WTI (I, J, K, L) = 0.
20    CONTINUE
    DO 10 L = 1, KD
        ANB (I, J, K, L) = 0.
10    CONTINUE
    DO 30 L = 1, KD
        FNB (L) = 0.
        DNB (L) = 0.
        PECL (L) = 1.
30    CONTINUE
RETURN
END
C&DATAUS
C *****
C      Version 11061987  Subroutine DATAUS
C *****
C
C *** Subroutine DATAUS Ausdrucken der Eingangsdaten *****
C

```

```

SUBROUTINE DATAUS
PARAMETER (IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2)
PARAMETER (IP1=1, IP2=1, IQ1=0, IQ2=0)
PARAMETER (JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2, KY=IY*IP2+IQ2, KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T, TO, STMP, CELS
INTEGER*2 IFLAG, JFLAG, IFLUS, IFRO
DIMENSION DELTAX (IX), DELTAY (IY), DELTAZ (IZ), DIFX (IX), DIFY (IY),
&          DIFZ (IZ), XKOR (IX), YKOR (IY), ZKOR (IZ), DIVX (IX, 2),
&          DIVY (IY, 2), DIVZ (IZ, 2), FX (IX), FY (IY), FZ (IZ),
&          DXYZ (IX, IY, IZ)
DIMENSION IFLAG (IX, IY, IZ), JFLAG (KX, KY, KZ), IFLUS (JX, JY, JZ), IFRO (JZ)
DIMENSION T (JX, JY, JZ), TO (JX, JY, JZ), STMP (JZ)
DIMENSION CPG (JX, JY, JZ), WLKG (JX, JY, JZ), WTI (IX, IY, IZ, KD2),
&          RHOG (JX, JY, JZ), PHIG (JX, JY, JZ), PHIN (IX, IY, IZ),
&          SATU (IZ), VX (IX, IY, IZ), VY (IX, IY, IZ), VZ (IX, IY, IZ), ANISO (3)
DIMENSION SC (JX, JY, JZ), SP (JX, JY, JZ), FRO (JZ, 5)
COMMON /ENER/SC, SP, FRO
COMMON /GITT/DELTAX, DELTAY, DELTAZ, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,
&          DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ
COMMON /IFLD/IFLAG, JFLAG, IFLUS, IFRO
COMMON /INF1/IZEIT, IPECL, IVOLDF, IAUSOP, ILOOP, IEIN, INPOP, IALTER
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /PHYS/CPG, WLKG, ANISO, WTI, RHOG, PHIG, PHIN, SATU, VX, VY, VZ
COMMON /TEMP/T, TO, STMP
C
IF (IECHO.GE.1) CALL ROUTIN ('DATAUS')
C
DO 10 K = 1, NZ
WRITE (6,100) K, ZKOR(K)
C *** Ausgangstemperatur in Grad Celsius *****
WRITE (6,120) K, ZKOR(K)
WRITE (6,310) (XKOR(I), I = 1, NX)
DO 20 J = 1, NY
WRITE (6,320) YKOR(J), (CELS(T(I,J,K)), I = 1, NX)
20 CONTINUE
C *** Waermekapazitaet des Gesteins in J/kg*K *****
WRITE (6,130) K, ZKOR(K)
WRITE (6,310) (XKOR(I), I = 1, NX)
DO 30 J = 1, NY
30 WRITE (6,330) YKOR(J), (CPG(I,J,K), I = 1, NX)
C *** Dichte des Gesteins in g/cbcm *****
WRITE (6,140) K, ZKOR(K)
WRITE (6,310) (XKOR(I), I = 1, NX)
DO 40 J = 1, NY
40 WRITE (6,320) YKOR(J), (GRCBCM(RHOG(I,J,K)), I = 1, NX)
C *** Waermeleitfaehigkeit des Gesteins in W/m*K *****
WRITE (6,150) K, ZKOR(K)
WRITE (6,310) (XKOR(I), I = 1, NX)
DO 50 J = 1, NY
50 WRITE (6,320) YKOR(J), (WLKG(I,J,K), I = 1, NX)
C *** Definition der Zellen *****
WRITE (6,160) K, ZKOR(K)

```

```

WRITE (6,310) (XKOR(I),I = 1, NX)
DO 60 J = 1, NY
60 WRITE (6,360) YKOR(J), (IFLAG(I,J,K),I = 1, NX)
C *** Gesamtporositaet in den Zellen *****
WRITE (6,170) K,ZKOR(K)
WRITE (6,310) (XKOR(I),I = 1, NX)
DO 70 J = 1, NY
70 WRITE (6,320) YKOR(J), (PHIG(I,J,K),I = 1, NX)
C *** Nutzporositaet in den Zellen *****
WRITE (6,180) K,ZKOR(K)
WRITE (6,310) (XKOR(I),I = 1, NX)
DO 80 J = 1, NY
80 WRITE (6,320) YKOR(J), (PHIN(I,J,K),I = 1, NX)
C *** interne Waermeproduktion in Watt/cbm *****
WRITE (6,190) K,ZKOR(K)
WRITE (6,310) (XKOR(I),I = 1, NX)
DO 90 J = 1, NY
90 WRITE (6,320) YKOR(J), (SC(I,J,K),I = 1, NX)
10 CONTINUE
WRITE (6,210)
WRITE (6,*) ' Knotenabstaende in X-Richtung (Meter):'
WRITE (6,340) (DIFX(I),I = 1, NX1)
WRITE (6,210)
WRITE (6,*) ' Knotenabstaende in Y-Richtung (Meter):'
WRITE (6,340) (DIFY(J),J = 1, NY1)
WRITE (6,210)
WRITE (6,*) ' Knotenabstaende in Z-Richtung (Meter):'
WRITE (6,340) (DIFZ(K),K = 1, NZ1)
100 FORMAT(/,' Eingangsparmeter von Schicht ',I2,
& ' - ',F6.2,' m u. GOK')
120 FORMAT(/,' Ausgangstemperatur in Grad Celsius - Schicht ',I2,
& ' - ',F6.2,' m u. GOK')
130 FORMAT(/,' Waermekapazitaet in J/kg*K - Schicht ',I2,
& ' - ',F6.2,' m u. GOK')
150 FORMAT(/,' thermische Leitfaehigkeit in W/m*K - Schicht ',I2,
& ' - ',F6.2,' m u. GOK')
140 FORMAT(/,' Dichte des Gesteins in g/cbcm - Schicht ',I2,
& ' - ',F6.2,' m u. GOK')
160 FORMAT(/,' T-Flags: 1=var.Tmp. 2=konst.Tmp. 3=Wrmf. 4=Wkof- Schich
&t 5=Wtausch.',I2,' - ',F6.2,' m u. GOK')
170 FORMAT(/,' Ges.Porositaet in den Zellen (0 - 1) - Schicht ',I2,
& ' - ',F6.2,' m u. GOK')
180 FORMAT(/,' Nutzporositaet in den Zellen (0 - 1) - Schicht ',I2,
& ' - ',F6.2,' m u. GOK')
190 FORMAT(/,' interne Waermeproduktion in Watt/cbm - Schicht ',I2,
& ' - ',F6.2,' m u. GOK')
210 FORMAT(/)
310 FORMAT(' Y\X ',10F6.2,/,9(7X,10F6.2,/))
320 FORMAT(F7.2,10F6.2,/,9(7X,10F6.2,/))
330 FORMAT(F7.2,10F6.0,/,9(7X,10F6.0,/))
340 FORMAT(6(10F6.2,/))
360 FORMAT(F7.2,10I6,/,9(7X,10I6,/))
RETURN
END

```

```

C&DATEIN
C *****
C   Version 11061987   Subroutine DATEIN
C *****
C
C *** Subroutine DATEIN Einlesen der Daten - langer Input *****
C
SUBROUTINE DATEIN
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,TO,STMP,CELV
CHARACTER FORMIN*20
INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
DIMENSION IFLAG (IX,IY,IZ),JFLAG (KX,KY,KZ),IFLUS (JX,JY,JZ),IFRO (JZ)
DIMENSION T (JX,JY,JZ),TO (JX,JY,JZ),STMP (JZ)
DIMENSION CPG (JX,JY,JZ),WLKG (JX,JY,JZ),WTI (IX,IY,IZ,KD2),
&          RHOG (JX,JY,JZ),PHIG (JX,JY,JZ),PHIN (IX,IY,IZ),
&          SATU (IZ),VX (IX,IY,IZ),VY (IX,IY,IZ),VZ (IX,IY,IZ),ANISO (3)
DIMENSION H (KX,KY,KZ),HO (KX,KY,KZ)
DIMENSION TTX (KX,KY,KZ),TTY (KX,KY,KZ),TTZ (KX,KY,KZ),
&          SK (KX,KY,KZ),Q (KX,KY,KZ)
DIMENSION SC (JX,JY,JZ),SP (JX,JY,JZ),FRO (JZ,5)
DIMENSION IXOBS (10),IYOBS (10),IZOBS (10)
COMMON /ENER/SC,SP,FRO
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
COMMON /TEMP/T,TO,STMP
COMMON /TRNS/TTX,TTY,TTZ,Q,SK
COMMON /WPOT/H,HO
PI = ATAN(1.) * 4.

C
C   IF (IECHO.GE.1) CALL ROUTIN ('DATEIN')
C
C *** Subroutine GITTER Berechnung der Gitterparameter *****
C
CALL GITTER

C
C *** optionale Eingabe der Anisotropiefaktoren fuer WLKG *****
C
READ (5,100) FORMIN
READ (5,FORMIN) IBX,IBY,IBZ,ANISO(1),ANISO(2),ANISO(3)
DO 10 K = 1, NZ
IF (IGWF.EQ.2) GO TO 500
C *** Ausgangstemperatur in Grad Celsius *****
READ (5,100) FORMIN
DO 20 J = 1, NY
READ (5,FORMIN) (T(I,J,K),I = 1, NX)

```

```

DO 20 I = 1, NX
T(I,J,K) = CELV(T(I,J,K))
20 CONTINUE
C *** Waermekapazitaet des Gesteins in J/kg*K *****
READ (5,100) FORMIN
DO 30 J = 1, NY
30 READ (5,FORMIN) (CPG(I,J,K),I = 1, NX)
C *** Dichte des Gesteins *****
READ (5,100) FORMIN
DO 40 J = 1, NY
READ (5,FORMIN) (RHOG(I,J,K),I = 1, NX)
DO 40 I = 1, NX
RHOG(I,J,K)=RHOG(I,J,K)*1000.
40 CONTINUE
C *** Waermeleitfaehigkeit des Gesteins in W/m*K *****
READ (5,100) FORMIN
DO 50 J = 1, NY
50 READ (5,FORMIN) (WLKG(I,J,K),I = 1, NX)
C *** Definition der Zellen *****
C
C IFLAG = 1 ==> variable Temperatur
C IFLAG = 2 ==> konstante Temperatur
C IFLAG = 5 ==> Waermetauscher
C
READ (5,100) FORMIN
DO 60 J = 1, NY
60 READ (5,FORMIN) (IFLAG(I,J,K),I = 1, NX)
C *** Gesamtporositaet in den Zellen *****
READ (5,100) FORMIN
DO 70 J = 1, NY
70 READ (5,FORMIN) (PHIG(I,J,K),I = 1, NX)
C *** Nutzporositaet in den Zellen *****
READ (5,100) FORMIN
DO 80 J = 1, NY
80 READ (5,FORMIN) (PHIN(I,J,K),I = 1, NX)
C *** Wassersaettigung in den Zellen in Prozent *****
READ (5,100) FORMIN
READ (5,FORMIN) SATU(K)
SATU(K) = SATU(K) / 100.
C *** interne Waermeproduktion in Watt/cbm *****
READ (5,100) FORMIN
DO 110 J = 1, NY
110 READ (5,FORMIN) (SC(I,J,K),I = 1, NX)
IF (IGWF.EQ.0.AND.IDIF.EQ.1) THEN
C *** Fliessgeschwindigkeit in m/sec in X - Richtung *****
READ (5,100) FORMIN
DO 120 J = 1, NY
120 READ (5,FORMIN) (VX(I,J,K),I = 1, NX)
C *** Fliessgeschwindigkeit in m/sec in Y - Richtung *****
READ (5,100) FORMIN
DO 130 J = 1, NY
130 READ (5,FORMIN) (VY(I,J,K),I = 1, NX)
C *** Fliessgeschwindigkeit in m/sec in Z - Richtung *****
READ (5,100) FORMIN

```

```

      DO 140 J = 1, NY
140    READ (5,FORMIN) (VZ(I,J,K),I = 1, NX)
      END IF
500  IF (IGWF.GE.1) THEN
      READ (5,100) FORMIN
      DO 150 J = 1, NY
150    READ (5,FORMIN) (H(I,J,K),I = 1, NX)
C *** Piezometerhoehen in Meter *****
      READ (5,100) FORMIN
      DO 160 J = 1, NY
160    READ (5,FORMIN) (TTX(I,J,K),I = 1, NX)
C *** hydraulische Leitfaehigkeit in X - Richtung *****
      DO 170 J = 1, NY
170    READ (5,FORMIN) (TTY(I,J,K),I = 1, NX)
C *** hydraulische Leitfaehigkeit in Y - Richtung *****
      DO 180 J = 1, NY
180    READ (5,FORMIN) (TTZ(I,J,K),I = 1, NX)
C *** hydraulische Leitfaehigkeit in Z - Richtung *****
      DO 190 J = 1, NY
190    READ (5,FORMIN) (SK(I,J,K),I = 1, NX)
C *** spezifischer Speicherkoeffizient *****
      DO 200 J = 1, NY
200    READ (5,FORMIN) (Q(I,J,K),I = 1, NX)
C *** Entnahmen und Zugaben in cbm/s *****
      END IF
      10 CONTINUE
      100 FORMAT(A20)
      RETURN
      END
C&DELTAH
C *****
C   Version 23031988  Subroutine DELTAH
C *****
C
C *** Subroutine DELTAH Berechnung der benetzten Interfacehoehe *****
C
      SUBROUTINE DELTAH (I,J,K)
      PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
      PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
      PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
      PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
      DIMENSION CPG(JX,JY,JZ),WLKG(JX,JY,JZ),WTI (IX,IY,IZ,KD2),
&              RHOG(JX,JY,JZ),PHIG(JX,JY,JZ),PHIN (IX,IY,IZ),
&              SATU (IZ),VX (IX,IY,IZ),VY (IX,IY,IZ),VZ (IX,IY,IZ),ANISO(3)
      DIMENSION AP1 (IX,IY,IZ),AP0 (IX,IY,IZ),ANB (IX,IY,IZ,KD),
&              FNB(KD),DNB(KD),PECL(KD),B (IX,IY,IZ)
      DIMENSION INDX(6),INDY(6),INDZ(6),IADD(4),JADD(4),KADD(4)
      COMMON /INCB/INDX,INDY,INDZ,IADD,JADD,KADD
      COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
      COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
      COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
      COMMON /KOEFL/AP1,AP0,ANB,FNB,DNB,PECL,B
      COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
      COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK

```

```

COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
DO 10 L = 1, 3
  II = I - IADD(L)
  JJ = J - JADD(L)
  IF (II.LE.0.OR.II.GE.NX) GO TO 10
  IF (JJ.LE.0.OR.JJ.GE.NY) GO TO 10
  CALL KONDUK (II,JJ,K)
10 CONTINUE
  CALL KONKOF (I,J,K)
  AP = 0.
  DO 20 L = 1, KD
    ANB(I,J,K,L) = DNB(L)
    AP = AP + ANB(I,J,K,L)
20 CONTINUE
  AP1(I,J,K) = AP + AP0(I,J,K)
  RETURN
  END
C&DKTEST
C *****
C   Version 19021988  Subroutine DKTEST
C *****
C *** Subroutine DKTEST Ueberpruefung der Diskretisierungsvorschriften *
C
  SUBROUTINE DKTEST
  PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
  PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
  PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
  PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
  DIMENSION DELTAX (IX), DELTAY (IY), DELTAZ (IZ), DIFX (IX), DIFY (IY),
&            DIFZ (IZ), XKOR (IX), YKOR (IY), ZKOR (IZ), DIVX (IX, 2),
&            DIVY (IY, 2), DIVZ (IZ, 2), FX (IX), FY (IY), FZ (IZ),
&            DXYZ (IX, IY, IZ)
  DIMENSION CPG (JX, JY, JZ), WLKG (JX, JY, JZ), WTI (IX, IY, IZ, KD2),
&            RHOG (JX, JY, JZ), PHIG (JX, JY, JZ), PHIN (IX, IY, IZ),
&            SATU (IZ), VX (IX, IY, IZ), VY (IX, IY, IZ), VZ (IX, IY, IZ), ANISO (3)
  DIMENSION IXOBS (10), IYOBS (10), IZOBS (10)
  COMMON /GITT/DELTAX, DELTAY, DELTAZ, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,
&            DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ
  COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
  COMMON /INF4/IOBS, IXOBS, IYOBS, IZOBS, IWARMS, IWSDAT, IDSTOP, IGWF
  COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
  COMMON /PAR3/CPW, CPE, CPL, WLW, WLE, WLL, RHOW, RHOE, RHOL, SENTHA
  COMMON /PHYS/CPG, WLKG, ANISO, WTI, RHOG, PHIG, PHIN, SATU, VX, VY, VZ
C
  IF (IECHO.GE.1) CALL ROUTIN ('DKTEST')
C *****
C
C           ABS (VX) * DX
C   2 =>  -----   wobei:  ALFA = -----
C                   ALFA                               RHO * Cp
C *****

```

```

C
  PKRIT = 0.
  ALFA = WLW / RHOW / CPW
  DO 10 K = 1, NZ1
  DO 10 J = 1, NY1
  DO 10 I = 1, NX1
C
C *** in X - Richtung ... *****
C
  GIPEZ = ABS(VX(I,J,K)) * DIFX(I) / ALFA
  IF (GIPEZ.GT.PKRIT) THEN
    PKRIT = GIPEZ
    II = I
    JJ = J
    KK = K
  END IF
C
C *** in Y - Richtung ... *****
C
  GIPEZ = ABS(VY(I,J,K)) * DIFY(J) / ALFA
  IF (GIPEZ.GT.PKRIT) THEN
    PKRIT = GIPEZ
    II = I
    JJ = J
    KK = K
  END IF
C
C *** in Z - Richtung ... *****
C
  GIPEZ = ABS(VZ(I,J,K)) * DIFZ(K) / ALFA
  IF (GIPEZ.GT.PKRIT) THEN
    PKRIT = GIPEZ
    II = I
    JJ = J
    KK = K
  END IF
10 CONTINUE
C
  IF (PKRIT.GT.2.) THEN
    IF (IECHO.GE.1) WRITE (1,100) II,JJ,KK,PKRIT
    WRITE (6,120)
    WRITE (6,100) II,JJ,KK,PKRIT
    IF (IDSTOP.EQ.2) THEN
      WRITE (1,130)
      WRITE (6,130)
      STOP
    END IF
    RETURN
  END IF
  IF (IECHO.GE.1) WRITE (1,110)
  RETURN
100 FORMAT(' Achtung!!!      Gitter-Peclet-Zahl an Knoten (' ,3I3,
&         ') mit ',F10.3,' zu gross',/,
&         ' Abhilfe: --> Knotenabstaende verkleinern')

```

```

110 FORMAT(T17,'Diskretisierungsvorschriften wurden eingehalten')
120 FORMAT(/)
130 FORMAT(T17,'IDSTOP aktiv - Simulation vorzeitig abgebrochen')
END

```

C&DZTEST

```

C *****
C Version 27101987 Subroutine DZTEST
C *****
C *** Subroutine DZTEST Ueberpruefung der Zeitschrittlaenge *****
C
SUBROUTINE DZTEST (ITEST)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
& DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
& DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
& DXYZ(IX,IY,IZ)
DIMENSION CPG(JX,JY,JZ),WLKG(JX,JY,JZ),WTI(IX,IY,IZ,KD2),
& RHOG(JX,JY,JZ),PHIG(JX,JY,JZ),PHIN(IX,IY,IZ),
& SATU(IZ),VX(IX,IY,IZ),VY(IX,IY,IZ),VZ(IX,IY,IZ),ANISO(3)
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
& DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /PAR3/CPW,CPE,CPL,WLW,WLE,WLL,RHOW,RHOE,RHOL,SENTHA
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ

```

C

```
IF (IECHO.GE.1) CALL ROUTIN ('DZTEST')
```

C

```
C *****
```

C

C

C

C

C

C

C

C

C

```

DTMAX = 1.E10
DZT    = DZEIT
WLKW   = WLW
CPF    = CPW
ROW    = RHOW

```

C

C

C

```
*** fuer den unguenstigsten Fall beim Gefrieren von Wasser gilt ... **
```

C

```

IF (IFROST.GE.1) THEN
  WLKW = WLE
  CPF  = CPE

```

```

      ROW = RHOE
      END IF
C
      DO 10 K = 1, NZ
      CPP = CPF * SATU(K) + CPL * (1. - SATU(K))
      RHOP = ROW * SATU(K) + RHOL * (1. - SATU(K))
      WLP = WLKW * SATU(K) + WLL * (1. - SATU(K))
      DO 10 J = 1, NY
      DO 10 I = 1, NX
      IF (WLKG(I,J,K).EQ.0..AND.(PHIG(I,J,K).EQ.0..OR.SATU(K).EQ.0.))
&      GO TO 10
      POR = PHIG(I,J,K)
      SOL = 1. - POR
      RHOC2K = ((SOL * RHOG(I,J,K) + POR * RHOP) *
&              (SOL * CPG(I,J,K) + POR * CPP)) /
&              (2. * (SOL * WLKG(I,J,K) + POR * WLP))
      ZEIT = RHOC2K * DELTAX(I) * DELTAX(I)
      IF (ZEIT.NE.0..AND.ZEIT.LT.DTMAX) DTMAX = ZEIT
      ZEIT = RHOC2K * DELTAY(J) * DELTAY(J)
      IF (ZEIT.LT.DTMAX) DTMAX = ZEIT
      IF (ZEIT.NE.0..AND.ZEIT.LT.DTMAX) DTMAX = ZEIT
      ZEIT = RHOC2K * DELTAZ(K) * DELTAZ(K)
      IF (ZEIT.LT.DTMAX) DTMAX = ZEIT
      IF (ZEIT.NE.0..AND.ZEIT.LT.DTMAX) DTMAX = ZEIT
10  CONTINUE
      IF (DZT.LT.DZEIT) THEN
      WRITE (1,100) DZEIT/60.,DZT/60.
      WRITE (IAUS,100) DZEIT/60.,DZT/60.
      IF (IDZTST.NE.2) RETURN
      STPALT = ITEST
      ITEST = NINT(ITEST * DZEIT / DZT) + 1
      IZEIT = NINT(ITEST / STPALT * IZEIT)
      DZEIT = DZEIT / ITEST * STPALT
      WRITE (1,110) DZEIT / 60.
      WRITE (IAUS,110) DZEIT / 60.
      RETURN
      END IF
      WRITE (1,120) DTMAX / 60.
100  FORMAT(/,' Zeitschrittlaenge mit ',F8.0,' Min. zu gross - maximal
& ',F5.0,' Min. erlaubt')
110  FORMAT(' Zeitschrittlaenge wurde auf ',F8.1,' Minuten verkuerzt!
& ')
120  FORMAT(' Zeitschrittlaenge darf max. ',F8.1,' Minuten betragen')
      RETURN
      END
C&ENERGI
C *****
C      Version 13071987  Subroutine ENERGI
C *****
C
C *** Subroutine ENERGI Berechnung der Waermeenergie in den Zellen ****
C
      SUBROUTINE ENERGI
      PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)

```

```

PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
IMPLICIT DOUBLE PRECISION (E)
DOUBLE PRECISION T,TO,STMP,TEMP,STEMP
DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
&          DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
&          DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
&          DXYZ(IX,IY,IZ)
DIMENSION T(JX,JY,JZ),TO(JX,JY,JZ),STMP(JZ)
DIMENSION SC(JX,JY,JZ),SP(JX,JY,JZ),FRO(JZ,5)
COMMON /EKAL/EALT,EANF,ESUM
COMMON /ENER/SC,SP,FRO
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
&          DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /TEMP/T,TO,STMP

```

```

C
IF (IECHO.GE.1) CALL ROUTIN ('ENERGI')
C
EGES = 0.
DO 10 K = 1, NZ
STEMP = STMP(K)
DO 10 J = 1, NY
DO 10 I = 1, NX
TEMP = T(I,J,K)
CALL ZELLEN (I,J,K,TEMP,STEMP)
EGES = EGES + ESUM
10 CONTINUE
IF (IEN1.EQ.1) THEN
EANF = EGES
ENEU = EGES
EBIL = 0.
ELSE
ENEU = EGES
EBIL = ENEU - EALT
ETOT = EGES - EANF
END IF
EALT = ENEU
WRITE (IAUS,30) EGES
WRITE (IAUS,40) EBIL
WRITE (IAUS,50) ETOT
IEN1 = 0
30 FORMAT(/,' Energieinhalt gesamt:                ',1PE15.6,
&         ' Joule')
40 FORMAT(' Energiebilanz seit dem letzten Ausdruck:  ',1PE15.6,
&         ' Joule')
50 FORMAT(' Energiebilanz seit Beginn der Simulation:  ',1PE15.6,
&         ' Joule')
RETURN
END

```

C&FLIESS

C *****
C Version 12041988 Subroutine FLIESS
C *****
C
C *** Subroutine FLIESS Berechnung der Fließgeschwindigkeiten *****
C

```
SUBROUTINE FLIESS
PARAMETER (IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2)
PARAMETER (IP1=1, IP2=1, IQ1=0, IQ2=0)
PARAMETER (JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2, KY=IY*IP2+IQ2, KZ=IZ*IP2+IQ2)
DIMENSION H(KX, KY, KZ), H0(KX, KY, KZ)
DIMENSION DELTAX(IX), DELTAY(IY), DELTAZ(IZ), DIFX(IX), DIFY(IY),
& DIFZ(IZ), XKOR(IX), YKOR(IY), ZKOR(IZ), DIVX(IX, 2),
& DIVY(IY, 2), DIVZ(IZ, 2), FX(IX), FY(IY), FZ(IZ),
& DXYZ(IX, IY, IZ)
DIMENSION CPG(JX, JY, JZ), WLKG(JX, JY, JZ), WTI(IX, IY, IZ, KD2),
& RHOG(JX, JY, JZ), PHIG(JX, JY, JZ), PHIN(IX, IY, IZ),
& SATU(IZ), VX(IX, IY, IZ), VY(IX, IY, IZ), VZ(IX, IY, IZ), ANISO(3)
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
COMMON /GITT/DELTAX, DELTAY, DELTAZ, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,
& DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /PHYS/CPG, WLKG, ANISO, WTI, RHOG, PHIG, PHIN, SATU, VX, VY, VZ
COMMON /WPOT/H, H0
```

C IF (IECHO.GE.1) CALL ROUTIN ('FLIESS')

C
C DO 10 K = 1, NZ
C DO 10 J = 1, NY
C DO 10 I = 1, NX
C VX(I, J, K) = 0.
C VY(I, J, K) = 0.
C VZ(I, J, K) = 0.

C *** Fließgeschwindigkeit in X - Richtung *****
C

```
IF (I.EQ.NX) GO TO 20
CALL HARMON (PHIN(I, J, K), PHIN(I+1, J, K), POR, FX(I))
VX(I, J, K) = WTI(I, J, K, 1) * (H(I, J, K) - H(I+1, J, K)) / DIFX(I)
VX(I, J, K) = VX(I, J, K) / POR
```

C *** Fließgeschwindigkeit in Y - Richtung *****
C

```
20 IF (J.EQ.NY) GO TO 30
CALL HARMON (PHIN(I, J, K), PHIN(I, J+1, K), POR, FY(J))
VY(I, J, K) = WTI(I, J, K, 2) * (H(I, J, K) - H(I, J+1, K)) / DIFY(J)
VY(I, J, K) = VY(I, J, K) / POR
```

C *** Fließgeschwindigkeit in Z - Richtung *****
C

```
30 IF (K.EQ.NZ) GO TO 10
CALL HARMON (PHIN(I, J, K), PHIN(I, J, K+1), POR, FZ(K))
```

```

      VZ(I,J,K) = WTI(I,J,K,3) * (H(I,J,K) - H(I,J,K+1)) / DIFZ(K)
      VZ(I,J,K) = VZ(I,J,K) / POR

```

```

10 CONTINUE
   RETURN
   END

```

C&FLUKOF

```

C *****
C   Version 29061987   Subroutine FLUKOF
C *****
C
C *** Subroutine FLUKOF Berechnung der Konvektionskoeffizienten *****
C

```

```

SUBROUTINE FLUKOF (I,J,K)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,TO,STMP,CELS
INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
&          DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
&          DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
&          DXYZ(IX,IY,IZ)
DIMENSION T(JX,JY,JZ),TO(JX,JY,JZ),STMP(JZ)
DIMENSION IFLAG(IX,IY,IZ),JFLAG(KX,KY,KZ),IFLUS(JX,JY,JZ),IFRO(JZ)
DIMENSION CPG(JX,JY,JZ),WLKG(JX,JY,JZ),WTI(IX,IY,IZ,KD2),
&          RHOG(JX,JY,JZ),PHIG(JX,JY,JZ),PHIN(IX,IY,IZ),
&          SATU(IZ),VX(IX,IY,IZ),VY(IX,IY,IZ),VZ(IX,IY,IZ),ANISO(3)
DIMENSION AP1(IX,IY,IZ),APO(IX,IY,IZ),ANB(IX,IY,IZ,KD),
&          FNB(KD),DNB(KD),PECL(KD),B(IX,IY,IZ)
DIMENSION IXOBS(10),IYOBS(10),IZOBS(10)
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
&          DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
COMMON /KOEf/AP1,APO,ANB,FNB,DNB,PECL,B
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /PAR3/CPW,CPE,CPL,WLW,WLE,WLL,RHOW,RHOE,RHOL,SENTHA
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
COMMON /TEMP/T,TO,STMP

```

```

DO 50 L = 1, KD
   FNB(L) = 0.

```

```

50 CONTINUE
   TMP = CELS(T(I,J,K))
   STP = CELS(STMP(K))
   ROW = RHOH2O(TMP,STP,K)
   IF (IGWF.EQ.0) IF (IFLUS(I,J,K)) 20,30,10

```

```

C *** nur horizontaler Fluss *****

```

```

10 IF (I.GT.1) FNB(1) = ROW * VX(I-1,J,K) * DELTAY(J) * DELTAZ(K)
   IF (I.LE.NX) FNB(2) = ROW * VX(I,J,K) * DELTAY(J) * DELTAZ(K)
   IF (J.GT.1) FNB(3) = ROW * VY(I,J-1,K) * DELTAZ(K) * DELTAX(I)

```

```

      IF (J.LE.NY) FNB(4) = ROW * VY(I,J,K) * DELTAZ(K) * DELTAX(I)
      IF (IGWF.EQ.0) RETURN
C
C *** nur vertikaler Fluss *****
C
      20 CONTINUE
      IF (K.GT.1) FNB(5) = ROW * VZ(I,J,K-1) * DELTAX(I) * DELTAY(J)
      IF (K.LE.NZ) FNB(6) = ROW * VZ(I,J,K) * DELTAX(I) * DELTAY(J)
      30 RETURN
      END
C&FLUSFL
C *****
C      Version 10061987 Subroutine FLUSFL
C *****
C *** Subroutine FLUSFL Festlegung der Flowflags *****
C
      SUBROUTINE FLUSFL (I1,J1,K1,I2,J2,K2)
      PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
      PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
      PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
      PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
      INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
      DOUBLE PRECISION T,T0,STMP
      DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
&              DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
&              DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
&              DXYZ(IX,IY,IZ)
      DIMENSION IFLAG(IX,IY,IZ),JFLAG(KX,KY,KZ),IFLUS(JX,JY,JZ),IFRO(JZ)
      DIMENSION T(JX,JY,JZ),T0(JX,JY,JZ),STMP(JZ)
      DIMENSION CPG(JX,JY,JZ),WLKG(JX,JY,JZ),WTI(IX,IY,IZ,KD2),
&              RHOG(JX,JY,JZ),PHIG(JX,JY,JZ),PHIN(IX,IY,IZ),
&              SATU(IZ),VX(IX,IY,IZ),VY(IX,IY,IZ),VZ(IX,IY,IZ),ANISO(3)
      COMMON /GIT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
&              DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
      COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
      COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
      COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
      COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
      COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
      COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
      COMMON /TEMP/T,T0,STMP
C
      IF (IECHO.GE.1) CALL ROUTIN ('FLUSFL')
C
C      IFLUS = -1 ==> nur vertikaler Fluss
C      IFLUS = 0 ==> kein Fluss
C      IFLUS = 1 ==> nur horizontaler Fluss
C
      DO 10 K = K1, K2
      DO 10 J = J1, J2
      DO 10 I = I1, I2
      IFLUS(I,J,K) = 0
      IF (T(I,J,K).GE.STMP(K) .AND. PHIN(I,J,K).GT.0. .AND.

```

```

&      SATU(K).LT.1.) IFLUS(I,J,K) = -1
      IF (T(I,J,K).GE.STMP(K) .AND. PHIN(I,J,K).GT.0.)
&      IFLUS(I,J,K) = 1
      IF (IFLAG(I,J,K).EQ.5) IFLUS(I,J,K) = 0
10 CONTINUE
C
C *** optionale Ausgabe der Flowflags *****
C
      IF (IAUSOP.EQ.3) THEN
      DO 20 K = 1, NZ
        WRITE (IAUS,30) K,ZKOR(K)
        WRITE (IAUS,50) (XKOR(I),I = 1, NX)
        DO 20 J = 1, NY
          WRITE (IAUS,40) YKOR(J), (IFLUS(I,J,K),I = 1, NX)
20      CONTINUE
      END IF
30 FORMAT(/, ' Fluss: -1 => nur VZ 0 => kein 1 => nur VX + VY',
&      ' - Schicht ',I2,' - ',F6.2,' m u. GOK')
40 FORMAT(F7.2,10I6,/,9(7X,10I6,/))
50 FORMAT('  Y\X ',10F6.2,/,9(7X,10F6.2,/))
      RETURN
      END
C&FUNKTI
C *****
C      Version 22061987 Double Precision Function CELS
C *****
C *** Double Precision Function CELS Umrechnung Kelvin/Celsius *****
C
      DOUBLE PRECISION FUNCTION CELS(CELVIN)
      DOUBLE PRECISION CELVIN
      CELS = CELVIN - 273.15D00
      RETURN
      END
C *****
C      Version 18021988 Double Precision Function CELV
C *****
C *** Double Precision Function CELV Umrechnung Celsius/Kelvin *****
C
      DOUBLE PRECISION FUNCTION CELV(CELSIU)
      DOUBLE PRECISION CELSIU
      CELV = CELSIU + 273.15D00
      RETURN
      END
C *****
C      Version 22061987 Function GRCBCM
C *****
C *** Function GRCBCM Umrechnung kg/cbm in g/cbcm *****
C
      FUNCTION GRCBCM(KGCBM)
      REAL KGCBM
      GRCBCM = KGCBM / 1000.

```

```

RETURN
END
C *****
C Version 14071987 Function RHOH2O
C *****
C *** Function RHOH2O Berechnung der Dichte von Wasser *****
C
FUNCTION RHOH2O(TMP,STP,K)
COMMON /PAR3/CPW,CPE,CPL,WLW,WLE,WLL,RHOW,RHOE,RHOL,SENTHA
IF (TMP.GE.STP) THEN
RHOH2O = RHOW
RETURN
END IF
CALL WASEIS (WE,TMP - STP,K)
RHOH2O = WE * RHOW + (1. - WE) * RHOE
RETURN
END
C *****
C Version 07051988 Function VISKIN
C *****
C *** Function VISKIN Berechnung der kinem. Viskositaet von Wasser *****
C
FUNCTION VISKIN(TT)
TT = TT - 273.15
VISKIN = 0.88977 * (1. + 1.70623 * EXP(-0.03406 * TT)) - 0.635
RETURN
END
C&GITTER
C *****
C Version 10061987 Subroutine GITTER
C *****
C *** Subroutine GITTER Berechnung der Gitterparameter *****
C
SUBROUTINE GITTER
PARAMETER (IW=1,IE=2,IN=1,IS=2,IT=1,IB=2)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
CHARACTER FORMIN*20,CARD*3
DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
& DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
& DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
& DXYZ(IX,IY,IZ)
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
& DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
C
IF (IECHO.GE.1) CALL ROUTIN ('GITTER')

```

```

C
C   IVOLDF = 0   => Knotenpunkt liegt im geometrischen Zentrum
C   IVOLDF = 1   => Interfaceflaechen liegen auf halber Strecke
C                   zwischen den Knotenpunkten
C
IF (IVOLDF.EQ.0) THEN
  CARD = '21'
  READ (5,500,END=600) FORMIN
  READ (5,FORMIN,ERR=700,END=600) (DELTAX(I),I = 1, NX)
  CARD = '23'
  READ (5,500,END=600) FORMIN
  READ (5,FORMIN,ERR=700,END=600) (DELTAY(J),J = 1, NY)
  CARD = '25'
  READ (5,500,END=600) FORMIN
  READ (5,FORMIN,ERR=700,END=600) (DELTAZ(K),K = 1, NZ)
  GO TO 50
END IF
IF (IVOLDF.EQ.1) THEN
  NXX = NX1
  NYY = NY1
  NZZ = NZ1
  IF (NX1.EQ.0) NXX = 1
  IF (NY1.EQ.0) NYY = 1
  IF (NZ1.EQ.0) NZZ = 1
  CARD = '21'
  READ (5,500,END=600) FORMIN
  READ (5,FORMIN,ERR=700,END=600) (DIFX(I),I = 1, NXX)
  CARD = '23'
  READ (5,500,END=600) FORMIN
  READ (5,FORMIN,ERR=700,END=600) (DIFY(J),J = 1, NYY)
  CARD = '25'
  READ (5,500,END=600) FORMIN
  READ (5,FORMIN,ERR=700,END=600) (DIFZ(K),K = 1, NZZ)
END IF
C
C *** Berechnung der Zellenbreite *****
C
IF (NX.EQ.1) THEN
  DELTAX(1) = DIFX(1)
  GO TO 15
END IF
DELTAX(1) = DIFX(1)/2.
DELTAX(NX) = DIFX(NX1)/2.
DO 10 I = 2, NX1
  DELTAX(I) = DIFX(I-1) * 0.5 + DIFX(I) * 0.5
10 CONTINUE
C
C *** Berechnung der Zellentiefe *****
C
15 IF (NY.EQ.1) THEN
  DELTAY(1) = DIFY(1)
  GO TO 25
END IF
DELTAY(1) = DIFY(1) / 2.

```

```

    DELTAY(NY) = DIFY(NY1) / 2.
    DO 20 J = 2, NY1
        DELTAY(J) = DIFY(J-1) * 0.5 + DIFY(J) * 0.5
20 CONTINUE
C
C *** Berechnung der Zellenhoehe *****
C
    25 IF (NZ.EQ.1) THEN
        DELTAZ(1) = DIFZ(1)
        GO TO 50
    END IF
    DELTAZ(1) = DIFZ(1) / 2.
    DELTAZ(NZ) = DIFZ(NZ1) / 2.
    DO 30 K = 2, NZ1
        DELTAZ(K) = DIFZ(K-1) * 0.5 + DIFZ(K) * 0.5
    30 CONTINUE
    50 CONTINUE
C
C *** Berechnung der Zellenvolumina *****
C
    DO 40 K = 1, NZ
    DO 40 J = 1, NY
    DO 40 I = 1, NX
        DXYZ(I,J,K) = DELTAX(I) * DELTAY(J) * DELTAZ(K)
    40 CONTINUE
    IF (IVOLDF.EQ.1) GO TO 1000
C
C *** falls Knoten im geometrischen Zentrum *****
C
    DO 80 I = 1, NX1
        DIFX(I) = DELTAX(I) * 0.5 + DELTAX(I+1) * 0.5
    80 CONTINUE
C
C *** Berechnung des Knotenabstaende in Y - Richtung *****
C
    DO 90 J = 1, NY1
        DIFY(J) = DELTAY(J) * 0.5 + DELTAY(J+1) * 0.5
    90 CONTINUE
C
C *** Berechnung des Knotenabstaende in Z - Richtung *****
C
    DO 100 K = 1, NZ1
        DIFZ(K) = DELTAZ(K) * 0.5 + DELTAZ(K+1) * 0.5
    100 CONTINUE
C
C *** Berechnung der Absolutkoordinaten der Knoten *****
C
    1000 CONTINUE
    XKOR(1) = 0.
    YKOR(1) = 0.
    ZKOR(1) = 0.
C
C *** falls Knoten im geometrischen Zentrum *****
C

```

```

IF (IVOLDF.EQ.0) THEN
  XKOR(1) = DELTAX(1) / 2.
  YKOR(1) = DELTAY(1) / 2.
  ZKOR(1) = DELTAZ(1) / 2.
END IF
DO 110 I = 2, NX
  XKOR(I) = XKOR(I-1) + DIFX(I-1)
110 CONTINUE
DO 120 J = 2, NY
  YKOR(J) = YKOR(J-1) + DIFY(J-1)
120 CONTINUE
DO 130 K = 2, NZ
  ZKOR(K) = ZKOR(K-1) + DIFZ(K-1)
130 CONTINUE
C
C *** Berechnung der Knoten-Interface-Strecken *****
C
IF (IVOLDF.EQ.0) GO TO 240
C
IF (NX.EQ.1) GO TO 215
DIVX(1,IW) = 0.
DIVX(1,IE) = DIFX(1) / 2.
DIVX(NX,IW) = DIFX(NX1) / 2.
DIVX(NX,IE) = 0.
C
DO 210 I = 2, NX1
  DIVX(I,IW) = DIFX(I-1) / 2.
  DIVX(I,IE) = DIFX(I) / 2.
210 CONTINUE
C
215 IF (NY.EQ.1) GO TO 225
DIVY(1,IN) = 0.
DIVY(1,IS) = DIFY(1) / 2.
DIVY(NY,IN) = DIFY(NY1) / 2.
DIVY(NY,IS) = 0.
C
DO 220 J = 2, NY1
  DIVY(J,IN) = DIFY(J-1) / 2.
  DIVY(J,IS) = DIFY(J) / 2.
220 CONTINUE
C
225 IF (NZ.EQ.1) GO TO 280
DIVZ(1,IT) = 0.
DIVZ(1,IB) = DIFZ(1) / 2.
DIVZ(NZ,IT) = DIFZ(NZ1) / 2.
DIVZ(NZ,IB) = 0.
C
DO 230 K = 2, NZ1
  DIVZ(K,IT) = DIFZ(K-1) / 2.
  DIVZ(K,IB) = DIFZ(K) / 2.
230 CONTINUE
GO TO 280
240 CONTINUE
C

```

```

C *** falls Knoten im geometrischen Zentrum *****
C
  DO 250 I = 1, NX
    DIVX(I,IW) = DELTAX(I) / 2.
    DIVX(I,IE) = DELTAX(I) / 2.
250 CONTINUE
C
  DO 260 J = 1, NY
    DIVY(J,IN) = DELTAY(J) / 2.
    DIVY(J,IS) = DELTAY(J) / 2.
260 CONTINUE
C
  DO 270 K = 1, NZ
    DIVZ(K,IT) = DELTAZ(K) / 2.
    DIVZ(K,IB) = DELTAZ(K) / 2.
270 CONTINUE
280 CONTINUE
RETURN
600 WRITE (1,610) CARD
STOP
700 WRITE (1,710) CARD
STOP
500 FORMAT(A20)
610 FORMAT(' unerwartet Ende der Eingabedatei erreicht in Karte ',A3)
710 FORMAT(' fehlerhafte Datenstruktur in Karte ',A3)
END

C&HARMON
C *****
C Version 23011988 Subroutine HARMON
C *****
C
C *** Subroutine HARMON Berechnung des gewichteten, harmonischen Mittels
C
SUBROUTINE HARMON (A,B,C,F)
IF (A.EQ.0..OR.B.EQ.0.) THEN
  C = 0.
  RETURN
END IF
C = A * B / ((1. - F) * A + F * B)
RETURN
END

C&HEADER
C *****
C Version 03081988 Subroutine HEADER
C *****
C
C *** Subroutine HEADER Ausgabe des Headers *****
C
SUBROUTINE HEADER (IR)
CHARACTER HEAD1*39,HEAD2*36,HEAD3*28
DIMENSION HEAD1(7),HEAD2(7),HEAD3(8)
DATA HEAD1/'TTTTTTTT RRRRRRR AAA DDDDDD ',
&          'TTTTTTTT RR RR AA AA DD DD ',
&          ' TT RR RR AA AA DD DD',

```

```

&          '   TT      RRRRRR      AAAAAAAAAA DD   DD',
&          '   TT      RR  RR      AAA   AAA DD   DD',
&          '   TT      RR  RR      AA    AA DD   DD ',
&          '   TT      RR    RR      AA    AA DDDDDD '/
DATA HEAD2/'IIIIII KK    KK    OOOO    NN    NN',
&          'IIIIII KK    KK    OO  OO  NNNN    NN',
&          '   II     KK  KK    OO  OO  NN  NN  NN',
&          '   II     KKKKK    OO  OO  NN  NN  NN',
&          '   II     KK  KK    OO  OO  NN    NNNN',
&          'IIIIII KK    KK    OO  OO  NN    NNN',
&          'IIIIII KK    KK    OOOO    NN    NN'/
DATA HEAD3/' 33333          DDDDDD ',
&          '33  33          DD  DD ',
&          '    33          DD  DD',
&          '  333  ===== DD  DD',
&          '    33          DD  DD',
&          '33  33          DD  DD ',
&          ' 33333          DDDDDD ',
&          ' Version 1.2  14/04/1989  '/

```

```

C
  IF (IR.EQ.1) WRITE (IR,50)
  IF (IR.EQ.6) WRITE (IR,75)
  WRITE (IR,250)
  WRITE (IR,125)
  DO 10 I = 1, 7
    WRITE (IR,100) HEAD1(I),HEAD2(I)
10 CONTINUE
  WRITE (IR,150)
  DO 20 I = 1, 7
    WRITE (IR,200) HEAD3(I)
20 CONTINUE
  DO 30 I = 1, IR
    WRITE (IR,125)
30 CONTINUE
  WRITE (IR,200) HEAD3(8)
  WRITE (IR,125)
  IF (IR.EQ.6) WRITE (IR,350)
  WRITE (IR,250)
  WRITE (IR,300)
  WRITE (IR,250)
  IF (IR.EQ.6) WRITE (IR,75)
  50 FORMAT(25(/))
  75 FORMAT(1H1)
100 FORMAT(1H ,A39,2X,A36)
125 FORMAT(1H )
150 FORMAT(/)
200 FORMAT(T27,A28)
250 FORMAT(1H ,78(1H*))
300 FORMAT(' Dipl.-Geol. Dirk Brehm c/o I A G - Uni Giessen Diezstra
&sse 15 D-6300 Giessen')
350 FORMAT(10(/))
  RETURN
  END
C&IADIST

```

```

C *****
C   Version 21071988   Subroutine IADIST
C *****
C
C *** Subroutine IADIST Stroemungsberechnung nach dem IADI-Verfahren ***
C
SUBROUTINE IADIST
PARAMETER (IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2)
PARAMETER (IP1=1, IP2=1, IQ1=0, IQ2=0)
PARAMETER (JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2, KY=IY*IP2+IQ2, KZ=IZ*IP2+IQ2)
DOUBLE PRECISION UU, AA, BB, CC, DD, ERR, FEHL
DIMENSION UU(IMAX), AA(IMAX), BB(IMAX), CC(IMAX), DD(IMAX)
DIMENSION H(KX, KY, KZ), HO(KX, KY, KZ)
DIMENSION DELTAX(IX), DELTAY(IY), DELTAZ(IZ), DIFX(IX), DIFY(IY),
& DIFZ(IZ), XKOR(IX), YKOR(IY), ZKOR(IZ), DIVX(IX, 2),
& DIVY(IY, 2), DIVZ(IZ, 2), FX(IX), FY(IY), FZ(IZ),
& DXYZ(IX, IY, IZ)
DIMENSION AP1(IX, IY, IZ), APO(IX, IY, IZ), ANB(IX, IY, IZ, KD),
& FNB(KD), DNB(KD), PECL(KD), B(IX, IY, IZ)
COMMON /GITT/DELTAX, DELTAY, DELTAZ, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,
& DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ
COMMON /IADI/UU, AA, BB, CC, DD
COMMON /INF1/IZEIT, IPECL, IVOLDF, IAUSOP, ILOOP, IEIN, INPOP, IALTER
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
COMMON /INF3/LUMW, IENER, ISTOP, IDATOP, LISDAT, IDZTST
COMMON /KOEf/AP1, APO, ANB, FNB, DNB, PECL, B
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /PAR1/DZEIT, DZMF, DZF, FLURAB, ALFATP, FEHLT, FEHLER, OMT, SUMSEK
COMMON /PAR2/BASIS, HOENN, ALFAHY, OMT, FEHLH, ZZZZ, ZIGNO
COMMON /WPOT/H, HO
90 OMEGA = - OMT
IF (IECHO.GE.1) CALL ROUTIN ('IADIST')
IAUSOP = 1
NIT = 0
M = 0
IF (IECHO.GE.1) WRITE (1, *)
1000 NIT = NIT + 1
FEHL = 0.
M = M + 1
C
C *** Zeilenberechnung *****
C
IF (NX.EQ.1) GO TO 170
CALL RICHTG (M, KANF, KEND, KPLUS, NZ, 1)
DO 160 K = KANF, KEND, KPLUS
N = K
IF (NZ.EQ.1) N = M
HO = HOENN - ZKOR(K) - DIVZ(K, 2) + DELTAZ(K)
CALL RICHTG (N, JANF, JEND, JPLUS, NY, 1)
DO 160 J = JANF, JEND, JPLUS
DO 100 I = 1, NX
AA(I) = - ANB(I, J, K, 1)
CC(I) = - ANB(I, J, K, 2)

```

```

      BB(I) = AP1(I,J,K)
      DD(I) = 0.
      IF (J.EQ.1) GO TO 120
      DD(I) = H(I,J-1,K) * ANB(I,J,K,3)
120  IF (J.EQ.NY) GO TO 130
      DD(I) = DD(I) + H(I,J+1,K) * ANB(I,J,K,4)
130  IF (K.EQ.1) GO TO 140
      DD(I) = DD(I) + H(I,J,K-1) * ANB(I,J,K,5)
140  IF (K.EQ.NZ) GO TO 150
      DD(I) = DD(I) + H(I,J,K+1) * ANB(I,J,K,6)
150  DD(I) = DD(I) + B(I,J,K)
      UU(I) = H(I,J,K)
100  CONTINUE
      CALL THOMAS (NX)
      DO 160 I = 1, NX
      HALT = H(I,J,K)
      H(I,J,K) = H(I,J,K) + OMEGA * (UU(I) - H(I,J,K))
      HNEU = H(I,J,K)
      ERR = ABS(HNEU - HALT)
      IF (ERR.GT.FEHL) FEHL = ERR
      IF (HALT.LT.HO.AND.HNEU.GT.HO) CALL DELTAH (I,J,K)
      IF (HNEU.LT.HO) CALL DELTAH (I,J,K)
160  CONTINUE
170  CONTINUE

```

C
C *** Spaltenberechnung *****
C

```

      IF (NY.EQ.1) GO TO 270
      CALL RICHTG (M,IANF,IEND,IPLUS,NX,1)
      DO 260 I = IANF, IEND, IPLUS
      CALL RICHTG (I,KANF,KEND,KPLUS,NZ,1)
      DO 260 K = KANF, KEND, KPLUS
      HO = HOENN - ZKOR(K) - DIVZ(K,2) + DELTAZ(K)
      DO 200 J = 1, NY
      AA(J) = - ANB(I,J,K,3)
      CC(J) = - ANB(I,J,K,4)
      BB(J) = AP1(I,J,K)
      DD(J) = 0.
      IF (I.EQ.1) GO TO 220
      DD(J) = H(I-1,J,K) * ANB(I,J,K,1)
220  IF (I.EQ.NX) GO TO 230
      DD(J) = DD(J) + H(I+1,J,K) * ANB(I,J,K,2)
230  IF (K.EQ.1) GO TO 240
      DD(J) = DD(J) + H(I,J,K-1) * ANB(I,J,K,5)
240  IF (K.EQ.NZ) GO TO 250
      DD(J) = DD(J) + H(I,J,K+1) * ANB(I,J,K,6)
250  DD(J) = DD(J) + B(I,J,K)
      UU(J) = H(I,J,K)
200  CONTINUE
      CALL THOMAS (NY)
      DO 260 J = 1, NY
      HALT = H(I,J,K)
      H(I,J,K) = H(I,J,K) + OMEGA * (UU(J) - H(I,J,K))
      HNEU = H(I,J,K)

```

```

ERR = ABS(HNEU - HALT)
IF (ERR.GT.FEHL) FEHL = ERR
IF (HALT.LT.HO.AND.HNEU.GT.HO) CALL DELTAH (I,J,K)
IF (HNEU.LT.HO) CALL DELTAH (I,J,K)
260 CONTINUE
270 CONTINUE

```

```

C
C *** Saeulenberechnung ****
C

```

```

IF (NZ.EQ.1) GO TO 370
CALL RICHTG (M,JANF,JEND,JPLUS,NY,1)
DO 360 J = JANF, JEND, JPLUS
CALL RICHTG (J,IANF,IEND,IPLUS,NX,1)
DO 360 I = IANF, IEND, IPLUS
DO 300 K = 1, NZ
AA(K) = - ANB(I,J,K,5)
CC(K) = - ANB(I,J,K,6)
BB(K) = AP1(I,J,K)
DD(K) = 0.
IF (I.EQ.1) GO TO 320
DD(K) = H(I-1,J,K) * ANB(I,J,K,1)
320 IF (I.EQ.NX) GO TO 330
DD(K) = DD(K) + H(I+1,J,K) * ANB(I,J,K,2)
330 IF (J.EQ.1) GO TO 340
DD(K) = DD(K) + H(I,J-1,K) * ANB(I,J,K,3)
340 IF (J.EQ.NY) GO TO 350
DD(K) = DD(K) + H(I,J+1,K) * ANB(I,J,K,4)
350 DD(K) = DD(K) + B(I,J,K)
UU(K) = H(I,J,K)
300 CONTINUE
CALL THOMAS (NZ)
DO 360 K = 1, NZ
HO = HOENN - ZKOR(K) - DIVZ(K,2) + DELTAZ(K)
HALT = H(I,J,K)
H(I,J,K) = H(I,J,K) + OMEGA * (UU(K) - H(I,J,K))
HNEU = H(I,J,K)
ERR = ABS(HNEU - HALT)
IF (ERR.GT.FEHL) FEHL = ERR
IF (HALT.LT.HO.AND.HNEU.GT.HO) CALL DELTAH (I,J,K)
IF (HNEU.LT.HO) CALL DELTAH (I,J,K)
360 CONTINUE
370 CONTINUE
IF (IECHO.GE.1) WRITE (1,30) NIT,FEHL
IF (FEHL.GT.FEHLH.AND.NIT.LT.ILOOP) GO TO 1000
FEHLER = FELH
IF (ISOR.EQ.2) THEN
ALFA = ALFAHY
CALL OMEGAT(FEHL,OMEGA,ALFA)
IF (ISOR.NE.1) THEN
CALL KOEFFH
OMGH = -OMGH
GO TO 90
END IF
END IF

```

```

30 FORMAT(1H+', Iteration Nummer ==> ',I3,' === Iterationsfehler = ',
& F11.6)
RETURN
END
C&IADITP
C *****
C Version 21071988 Subroutine IADITP
C *****
C *** Subroutine IADITP Temperaturberechnung nach dem IADI-Verfahren ***
C
SUBROUTINE IADITP
PARAMETER (IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2)
PARAMETER (IP1=1, IP2=1, IQ1=0, IQ2=0)
PARAMETER (JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2, KY=IY*IP2+IQ2, KZ=IZ*IP2+IQ2)
PARAMETER (IWERT=400)
DOUBLE PRECISION UU, AA, BB, CC, DD, ERR, FEHL
DOUBLE PRECISION T, TO, STMP, TT, TALT
CHARACTER FLAG*4
INTEGER*2 IFLAG, JFLAG, IFLUS, IFRO
INTEGER*2 IXX, IYY, IZZ, IUFL
DIMENSION WERT(IWERT), IXX(IWERT), IYY(IWERT), IZZ(IWERT),
& FLAG(IWERT), IUFL(IWERT)
DIMENSION UU(IMAX), AA(IMAX), BB(IMAX), CC(IMAX), DD(IMAX)
DIMENSION DELTAX(IX), DELTAY(IY), DELTAZ(IZ), DIFX(IX), DIFY(IY),
& DIFZ(IZ), XKOR(IX), YKOR(IY), ZKOR(IZ), DIVX(IX, 2),
& DIVY(IY, 2), DIVZ(IZ, 2), FX(IX), FY(IY), FZ(IZ),
& DXYZ(IX, IY, IZ)
DIMENSION AP1(IX, IY, IZ), AP0(IX, IY, IZ), ANB(IX, IY, IZ, KD),
& FNB(KD), DNB(KD), PECL(KD), B(IX, IY, IZ)
DIMENSION T(JX, JY, JZ), TO(JX, JY, JZ), STMP(JZ)
DIMENSION IFLAG(IX, IY, IZ), JFLAG(KX, KY, KZ), IFLUS(JX, JY, JZ), IFRO(JZ)
COMMON /GITT/DELTAX, DELTAY, DELTAZ, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,
& DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ
COMMON /IADI/UU, AA, BB, CC, DD
COMMON /IFLD/IFLAG, JFLAG, IFLUS, IFRO
COMMON /INF1/IZEIT, IPECL, IVOLDF, IAUSOP, ILOOP, IEIN, INPOP, IALTER
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
COMMON /INF3/LUMW, IENER, ISTOP, IDATOP, LISDAT, IDZTST
COMMON /KOEf/AP1, AP0, ANB, FNB, DNB, PECL, B
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /PAR1/DZEIT, DZMF, DZF, FLURAB, ALFATP, FEHLT, FEHLER, OMGT, SUMSEK
COMMON /TEMP/T, TO, STMP
COMMON /ZYK1/WERT, IXX, IYY, IZZ, IUFL
COMMON /ZYK2/FLAG
90 OMEGA = - OMGT
IF (IECHO.GE.1) CALL ROUTIN ('IADITP')
IAUSOP = 1
NIT = 0
M = 0
IF (IECHO.GE.1) WRITE (1, *)
1000 NIT = NIT + 1
FEHL = 0.

```

```

M = M + 1
C
C *** Zeilenberechnung ****
C
IF (NX.EQ.1) GO TO 170
CALL RICHTG (M,KANF,KEND,KPLUS,NZ,1)
DO 160 K = KANF, KEND, KPLUS
N = K
IF (NZ.EQ.1) N = M
CALL RICHTG (N,JANF,JEND,JPLUS,NY,1)
DO 160 J = JANF, JEND, JPLUS
DO 100 I = 1, NX
AA(I) = - ANB(I,J,K,1)
CC(I) = - ANB(I,J,K,2)
BB(I) = AP1(I,J,K)
DD(I) = 0.
IF (J.EQ.1) GO TO 120
DD(I) = T(I,J-1,K) * ANB(I,J,K,3)
120 IF (J.EQ.NY) GO TO 130
DD(I) = DD(I) + T(I,J+1,K) * ANB(I,J,K,4)
130 IF (K.EQ.1) GO TO 140
DD(I) = DD(I) + T(I,J,K-1) * ANB(I,J,K,5)
140 IF (K.EQ.NZ) GO TO 150
DD(I) = DD(I) + T(I,J,K+1) * ANB(I,J,K,6)
150 DD(I) = DD(I) + B(I,J,K)
UU(I) = T(I,J,K)
100 CONTINUE
CALL THOMAS (NX)
DO 160 I = 1, NX
TALT = T(I,J,K)
T(I,J,K) = T(I,J,K) + OMEGA * (UU(I) - T(I,J,K))
IF (IFROST.GE.1) CALL PHATST (I,J,K,TALT,T(I,J,K))
ERR = DABS(T(I,J,K) - TALT)
IF (ERR.GT.FEHL) FEHL = ERR
160 CONTINUE
170 CONTINUE
C
C *** Spaltenberechnung ****
C
IF (NY.EQ.1) GO TO 270
CALL RICHTG (M,IANF,IEND,IPLUS,NX,1)
DO 260 I = IANF, IEND, IPLUS
CALL RICHTG (I,KANF,KEND,KPLUS,NZ,1)
DO 260 K = KANF, KEND, KPLUS
DO 200 J = 1, NY
AA(J) = - ANB(I,J,K,3)
CC(J) = - ANB(I,J,K,4)
BB(J) = AP1(I,J,K)
DD(J) = 0.
IF (I.EQ.1) GO TO 220
DD(J) = T(I-1,J,K) * ANB(I,J,K,1)
220 IF (I.EQ.NX) GO TO 230
DD(J) = DD(J) + T(I+1,J,K) * ANB(I,J,K,2)
230 IF (K.EQ.1) GO TO 240

```

```

      DD(J) = DD(J) + T(I,J,K-1) * ANB(I,J,K,5)
240 IF (K.EQ.NZ) GO TO 250
      DD(J) = DD(J) + T(I,J,K+1) * ANB(I,J,K,6)
250 DD(J) = DD(J) + B(I,J,K)
      UU(J) = T(I,J,K)
200 CONTINUE
      CALL THOMAS (NY)
      DO 260 J = 1, NY
      TALT = T(I,J,K)
      T(I,J,K) = T(I,J,K) + OMEGA * (UU(J) - T(I,J,K))
      IF (IFROST.GE.1) CALL PHATST (I,J,K,TALT,T(I,J,K))
      ERR = DABS(T(I,J,K) - TALT)
      IF (ERR.GT.FEHL) FEHL = ERR
260 CONTINUE
270 CONTINUE
C
C *** Saeulenberechnung *****
C
      IF (NZ.EQ.1) GO TO 370
      CALL RICHTG (M,JANF,JEND,JPLUS,NY,1)
      DO 360 J = JANF, JEND, JPLUS
      CALL RICHTG (J,IANF,IEND,IPLUS,NX,1)
      DO 360 I = IANF, IEND, IPLUS
      DO 300 K = 1, NZ
      AA(K) = - ANB(I,J,K,5)
      CC(K) = - ANB(I,J,K,6)
      BB(K) = AP1(I,J,K)
      DD(K) = 0.
      IF (I.EQ.1) GO TO 320
      DD(K) = T(I-1,J,K) * ANB(I,J,K,1)
320 IF (I.EQ.NX) GO TO 330
      DD(K) = DD(K) + T(I+1,J,K) * ANB(I,J,K,2)
330 IF (J.EQ.1) GO TO 340
      DD(K) = DD(K) + T(I,J-1,K) * ANB(I,J,K,3)
340 IF (J.EQ.NY) GO TO 350
      DD(K) = DD(K) + T(I,J+1,K) * ANB(I,J,K,4)
350 DD(K) = DD(K) + B(I,J,K)
      UU(K) = T(I,J,K)
300 CONTINUE
      CALL THOMAS (NZ)
      DO 360 K = 1, NZ
      TALT = T(I,J,K)
      T(I,J,K) = T(I,J,K) + OMEGA * (UU(K) - T(I,J,K))
      IF (IFROST.GE.1) CALL PHATST (I,J,K,TALT,T(I,J,K))
      ERR = DABS(T(I,J,K) - TALT)
      IF (ERR.GT.FEHL) FEHL = ERR
360 CONTINUE
370 CONTINUE
      IF (IWCP.GE.3) CALL KOEFFT (1,1,1,NX,NY,NZ)
      IF (IECHO.GE.1) WRITE (1,30) NIT,FEHL
      IF (FEHL.GT.FEHLT.AND.NIT.LT.ILOOP) GO TO 1000
      FEHLER = FEHL
      IF (ISOR.EQ.1) THEN
        ALFA = ALFATP

```

```

        CALL OMEGAT(FEHL,OMEGA,ALFA)
        IF (ISOR.NE.0) THEN
            OMGT = -OMGT
            GO TO 90
        END IF
    END IF
30  FORMAT(1H+,' Iteration Nummer ==> ',I3,' === Iterationsfehler = ',
    &      F11.6)
    RETURN
    END
C&INFINT
C *****
C      Version 18061987  Subroutine INFINT
C *****
C *** Subroutine INFINT Berechnung der Interface-Wichtungsfaktoren *****
C
    SUBROUTINE INFINT
    PARAMETER (IW=1,IE=2,IN=1,IS=2,IT=1,IB=2)
    PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
    PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
    PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
    PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
    DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
    &          DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
    &          DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
    &          DXYZ(IX,IY,IZ)
    COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
    &          DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
    COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
    COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
    COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
C
    IF (IECHO.GE.1) CALL ROUTIN ('INFINT')
C
    IF (IVOLDF.EQ.0) GO TO 40
C
C *** IVOLDF = 1 => Interfaceflaechen liegen auf halber Strecke
C                               zwischen den Knotenpunkten
C
    DO 10 I = 1, NX1
        FX(I) = 0.5
    10 CONTINUE
    DO 20 J = 1, NY1
        FY(J) = 0.5
    20 CONTINUE
    DO 30 K = 1, NZ1
        FZ(K) = 0.5
    30 CONTINUE
    RETURN
C
C *** IVOLDF = 0 => Knotenpunkt liegt im geometrischen Zentrum **
C
    40 CONTINUE

```

```

DO 50 I = 1, NX1
  FX(I) = DIVX(I,IE) / DIFX(I)
50 CONTINUE
DO 60 J = 1, NY1
  FY(J) = DIVY(J,IS) / DIFY(J)
60 CONTINUE
DO 70 K = 1, NZ1
  FZ(K) = DIVZ(K,IB) / DIFZ(K)
70 CONTINUE
RETURN
END

```

C&INFORM

```

C *****
C   Version 11061987  Subroutine INFORM
C *****
C *** Subroutine INFORM Einlesen des Problemtitels und Timesteps *****
C

```

SUBROUTINE INFORM

```

PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,TO,STMP,CELV
CHARACTER ZEIT*3,ZFLAG*3,FORMIN*20,TITEL*80,TEXT*40,FORM*9
CHARACTER EIN*20,AUS*20,OBSFIL*20,DATOP*20,CARD*3
INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
DIMENSION ZEIT(4)
DIMENSION T(JX,JY,JZ),TO(JX,JY,JZ),STMP(JZ)
DIMENSION SC(JX,JY,JZ),SP(JX,JY,JZ),FRO(JZ,5)
DIMENSION IFLAG(IX,IY,IZ),JFLAG(KX,KY,KZ),IFLUS(JX,JY,JZ),IFRO(JZ)
DIMENSION IXOBS(10),IYOBS(10),IZOBS(10)
COMMON /ENER/SC,SP,FRO
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
COMMON /INF5/IVIS,IWRO
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /IPRO/IPROX,IPROY,IPROZ
COMMON /ISOP/IXISO,IYISO,IZISO,ISO1,IMSBIN
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /PAR2/BASIS,HOENN,ALFAHY,OMGH,FEHLH,ZZZZ,ZIGNO
COMMON /PAR3/CPW,CPE,CPL,WLW,WLE,WLL,RHOW,RHOE,RHOL,SENTHA
COMMON /PAR4/DTW,DTW2,DWAN,DTWL(3)
COMMON /TEMP/T,TO,STMP
DATA ZEIT/'SEK','MIN','STU','TAG'/
AUS = 'CON'
EIN = 'TRADAT'
OPEN(1,FILE=AUS,STATUS='OLD')
CALL HEADER(1)
GO TO 2000
1000 WRITE (1,220)

```

```

READ (*,20) EIN
2000 OPEN(5,FILE=EIN,STATUS='OLD',ERR=1000)
C
C   UNIT  1  Bildschirmausgabe
C   UNIT  5  Eingabedatei
C   UNIT  6  Ergebnisdatei
C   UNIT  7  optionale Ausgabedatei fuer Beobachtungsknoten
C   UNIT  8  optionale Eingabedatei der Zusatzdaten
C   UNIT  9  optionale Ein-/Ausgabedatei fuer einen Warmstart
C
CARD = '1'
READ (5,10) TITEL
CARD = '3'
READ (5,20,END=900) FORMIN
READ (5,FORMIN,ERR=800,END=900) IXISO,IYISO,IZISO,IMSBIN
CARD = '5'
READ (5,20,END=900) FORMIN
READ (5,FORMIN,ERR=800,END=900) IOBS,IDATOP,LISDAT,OBSFIL,
&      AUS,DATOP
C
C   IOBS = 0    ==> OBSFIL wird nicht eroeffnet
C   IOBS > 0    ==> OBSFIL wird mit Knotentemperaturen beschrieben
C
C   IDATOP = 0  ==> es werden keine zusaetzlichen Daten angefordert
C   IDATOP = 1  ==> es werden Daten von File DATOP angefordert
C
C   LISDAT gibt an, in welchem Intervall zusaetzliche Daten von File
C       DATOP angefordert werden.
C       z.B. LISDAT = 1 ==> nach jedem Zeitschritt
C       LISDAT = 9 ==> nach jedem neunten Zeitschritt usw.
C
IAUS = 6
IF (IDATOP.EQ.1) THEN
  OPEN(8,FILE=DATOP,STATUS='OLD',ERR=500)
  GO TO 510
500  WRITE (1,610) DATOP
  STOP
  END IF
510  IF (IOBS.GT.0) OPEN(7,FILE=OBSFIL)
C
C *** Eingabe von TERMINAL fuer File AUS bewirkt Bildschirmausgabe ***
C
IF (AUS.EQ.'TERMINAL') THEN
  IAUS = 1
  GO TO 250
  END IF
OPEN(IAUS,FILE=AUS,STATUS='OLD',ERR=210)
GO TO 250
210 OPEN(IAUS,FILE=AUS,STATUS='NEW')
250 CARD = '7'
READ (5,20,END=900) FORMIN
READ (5,FORMIN,ERR=800,END=900) IZEIT,ILOOP,IEIN,IAUSOP,INPOP,
&      IDRUI,ISTLAE,ZFLAG,DZF,FEHLT,Z
IF (Z.GT.0.) ZZZZ = Z

```

```

C
C   IAUSOP = 1   ==> nur Temperaturprofile
C   IAUSOP = 2   ==> Temperatur- und Eingangsdatenprofile
C   IAUSOP = 3   ==> alle Eingangsdaten + Temperaturprofile
C
C   CARD = '9'
C   READ (5,20,END=900) FORMIN
C   READ (5,FORMIN,ERR=800,END=900) FLURAB,MODEL,IENER,ISTOP,IDZTST,
&                                     IWARMS,IWSDAT,IDSTOP
C
C   MODEL = 1   ==> Maximaleinfluss des Porenraumes
C   MODEL = 2   ==> Minimaleinfluss des Porenraumes
C
C   IDZTST = 0  ==> Zeitschrittlänge wird nur getestet
C   IDZTST = 1  ==> Zeitschrittlänge wird getestet, ggf. verkürzt
C
C   IWARMS = 0  ==> "Kaltstart"
C   IWARMS = 1  ==> "Warmstart", d.h. es werden die bereits
C                   berechneten Temperaturen des letzten Ergebnis-
C                   druckes einer vorherigen Simulation aus Aus-
C                   gangsbasis benutzt
C
C   IWSDAT = 0  ==> "Warmstartdaten" - Ausgabe wird unterdrückt
C   IWSDAT = 1  ==> es werden parallel zu jedem Ergebnisausdruck
C                   alle berechneten Temperaturen auf die Datei
C                   TRADWS geschrieben; diese Datei wird bei einer
C                   darauffolgenden Simulation gelesen, sofern
C                   IWARMS = 1
C
C   IDSTOP = 0  ==> Simulation wird auch bei verletzten Dis-
C                   kretisierungsvorschriften fortgesetzt
C   IDSTOP = 1  ==> Simulation wird bei verletzten Dis-
C                   kretisierungsvorschriften vorzeitig abgebrochen
C
C   IOBS       ==> Anzahl der Beobachtungsknoten
C
C   CARD = '11'
C   READ (5,20,END=900) FORMIN
C   READ (5,FORMIN,ERR=800,END=900) ALFATP,OMGT,IVOLDF,IPECL,IDIF,
&                                     IFROST,IECHO,IALTER,IWCP,ISOR
C   IF (IWCP.EQ.0) THEN
C       CPW      = 0.
C       CPE      = 0.
C       CPL      = 0.
C       WLW      = 0.
C       WLE      = 0.
C       WLL      = 0.
C       RHOW     = 0.
C       RHOE     = 0.
C       RHOL     = 0.
C       SENTHA   = 0.
C       DTW      = 0.1
C       DWAN     = 0.001
C   END IF

```

```

IF (IWCP.EQ.1) THEN
  CPW   = 4200.
  CPE   = 2114.
  CPL   = 0.
  WLW   = 0.57
  WLE   = 2.25
  WLL   = 0.025
  RHOW  = 1000.
  RHOE  = 917.
  RHOL  = 0.
  SENTHA = 6030. / 18.0153 * 1000.
  DTW   = 0.1
  DWAN  = 0.001

```

```

C
C Waermekapazitaet von Luft:          0 J/kg*K
C Waermekapazitaet von Wasser:       4200 J/kg*K
C Waermekapazitaet von Eis:          2114 J/kg*K
C Waermeleitfaehigkeit von Luft:    0.025 W/m*K
C Waermeleitfaehigkeit von Wasser:  0.570 W/m*K
C Waermeleitfaehigkeit von Eis:     2.250 W/m*K
C Dichte von Luft:                    0 kg/cbm
C Dichte von Wasser:                  1000 kg/cbm
C Dichte von Eis:                     917 kg/cbm
C Schmelzenthalpie von reinem Wasser: 6030 J/mol
C Molekulargewicht von Wasser         18.0153 g/mol
C DTW bei Neuberechnung der Waerme-
C kapazitaet von teilgefrorenem Wasser 0.1 K
C DWAN maximale Aenderung des unge-
C frorenen Wasseranteiles             0.1 %
C
C vergl. Anhang 1 der Dokumentation

```

```

END IF
IF (IWCP.GE.2) THEN
  CARD = '11b'
  READ (5,20,END=900) FORMIN
  READ (5,FORMIN,ERR=800,END=900) CPW,CPE,CPL,WLW,WLE,WLL,RHOW,
&                                RHOE,RHOL,SENTHA,DTW,DWAN

```

```

C
C *** Modelle fuer die P/T - Abhaengigkeit der WLF *****
C

```

```

  IF (IWCP.GE.3) THEN
    CARD = '11d'
    READ (5,20,END=900) FORMIN
    IDT = 2
    IF (IWCP.LT.5) IDT = 3
    READ (5,FORMIN,ERR=800,END=900) (DTWL(I),I=1,IDT)
  END IF
END IF

```

```

C
C *** NB: SENTHA in Joule/kg eingeben *****
C
  DTW2 = DTW / 2.

```

```

C *** Bestimmung des Loesungsschema der Temperaturberechnung *****
C
C   ALFATP = 0           ==>   Explizit - Schema
C   1 > ALFATP > 0     ==>   Implizit - Schema
C   ALFATP = 0.5       ==>   Crank - Nicolson - Schema
C   ALFATP = 1         ==>   Vollimplizit - Schema
C *** Iterationsverfahren der Temperaturberechnung *****
C
C   OMGT = 0.0          ==>   Default - (IADI - Verfahren)
C   OMGT < -1.0        ==>   Line - Successive Over-Relaxation
C   0.0 > OMGT > -1.0 ==>   Line - Successive Under-Relaxation
C   OMGT = -1.0        ==>   IADI - Verfahren
C   OMGT = 1.0         ==>   Gauss - Seidel - Verfahren
C   OMGT > 1.0         ==>   Successive Over-Relaxation
C   0.0 < OMGT < 1.0  ==>   Successive Under-Relaxation
C
C *** Berechnungsschema der PECLET - Wichtungsfunktion *****
C
C   IPECL = 0          ==>   Default - (Exponential - Schema)
C   IPECL = 1          ==>   Zentrale Differenzen
C   IPECL = 2          ==>   Upwind - Schema
C   IPECL = 3          ==>   Hybrid - Schema
C   IPECL = 4          ==>   Power law - Schema
C   IPECL = 5          ==>   Exponential - Schema
C
C
C   IDIF = 0           ==>   reine Konduktion
C   IDIF = 1           ==>   Diffussion und Konvektion
C
C   IFROST = 0         ==>   Phasenaenderung wird ignoriert
C   IFROST > 0         ==>   Phasenaenderung wird beruecksichtigt:
C   IFROST = 1         ==>   STMP(K) = 0 Grad Celsius
C   IFROST = 2         ==>   STMP(K) wird in Subroutine TAUPKT
C                           berechnet (noch nicht moeglich)
C   IFROST = 3         ==>   STMP(K) wird von der Eingabedatei
C                           angefordert (Karten 11a und 11b)
C
C   IECHO = -1         ==>   kein Bildschirmprotokoll
C   IECHO = 0          ==>   nur Zeit wird ausgegeben
C   IECHO = 1          ==>   Bildschirmprotokoll aktiviert, kompakt
C   IECHO = 2          ==>   Bildschirmprotokoll aktiviert, komplett
C
C   IALTER = 0         ==>   Iteration in aufsteigender Reihenfolge
C   IALTER = 1         ==>   Iteration in alternierender Reihenfolge
C
C   IWCP = 0           ==>   Fluideigenschaften alle 0
C   IWCP = 1           ==>   Fluideigenschaften gegeben, s.o.
C   IWCP = 2           ==>   Fluideigenschaften werden eingelesen
C                           wie 2, jedoch WLKG temperaturabhaengig
C                           (polynomische Approximation)
C   IWCP = 3           ==>   wie 2, jedoch WLKG temperaturabhaengig
C                           (exponentielle Approximation)
C
C   ISOR = 0           ==>   OMEGA - Wert wird nicht getestet

```

```

C      ISOR      = 1          ==>      OMEGA - Wert fuer OMGT wird getestet
C
C      IENER      = 0          ==>      Energibilanzierung wird ignoriert
C      IENER      = 1          ==>      Energibilanzierung wird aktiviert
C
C      ISTOP      = 0          ==>      Berechnung nicht abbrechen
C      ISTOP      = 1          ==>      Berechnung beim Gleichgewichtszustand
C                                     abbrechen (NIT = 1 und FEHLER < FEHLT)
C
C *** Bestimmung der Lage des Knotenpunktes innerhalb der Zellen *****
C
C      IVOLDF      = 0          =>      Knotenpunkt liegt im geometrischen Zentrum
C      IVOLDF      = 1          =>      Interfaceflaechen liegen auf halber Strecke
C                                     zwischen den Knotenpunkten
C
C      CARD = '13'
C      READ (5,20,END=900) FORMIN
C      READ (5,FORMIN,ERR=800,END=900) BASIS,HOENN,ALFAHY,OMGH,FEHLH,
C      &                                     IGWF,IVIS,IWRO
C
C *** falls Stroemungsfeld aus der Durchlaessigkeit und der Potential-
C      differenz berechnet werden soll ...
C
C      IGWF        = 0          =>      Stroemungsfeld gegeben
C      IGWF        = 1          =>      Stroemungsfeld wird berechnet
C      IGWF        = 2          =>      es wird nur das Stroemungsfeld berechnet
C      BASIS                                     Aquiferbasis in Meter ueber NN
C      HOENN                                     Hoehe der Erdoberflaechen in Meter ueber NN
C      IVIS        = 0          =>      Viskositaetsaenderung des Wassers ignorieren
C      IVIS        = 1          =>      Viskositaetsaenderung des Wassers wird in
C                                     Abhaengigkeit der Temperatur berechnet
C      IWRO        = 0          =>      Dichte des Wassers konstant
C      IWRO        = 1          =>      Dichte des Wassers wird in Abhaengigkeit
C                                     der Temperatur berechnet
C
C *** Bestimmung des Loesungsschema der Stroemungsberechnung *****
C
C      ALFAHY = 0          ==>      Explizit - Schema
C      1 > ALFAHY > 0      ==>      Implizit - Schema
C      ALFAHY = 0.5        ==>      Crank - Nicolson - Schema
C      ALFAHY = 1          ==>      Vollimplizit - Schema
C
C *** Iterationsverfahren der Stroemungsberechnung *****
C
C      OMGH = 0.0          ==>      Default - (IADI - Verfahren)
C      OMGH < -1.0         ==>      Line - Successive Over-Relaxation
C      0.0 > OMGH > -1.0   ==>      Line - Successive Under-Relaxation
C      OMGH = -1.0         ==>      IADI - Verfahren
C      OMGH = 1.0          ==>      Gauss - Seidel - Verfahren
C      OMGH > 1.0          ==>      Successive Over-Relaxation
C      0.0 < OMGH < 1.0    ==>      Successive Under-Relaxation
C
C      UNIT 11 optionale Ausgabedatei fuer Y - Z Isolinienplots (T)
C      UNIT 12 optionale Ausgabedatei fuer X - Z Isolinienplots (T)

```

```

C      UNIT 13  optionale Ausgabedatei fuer X - Y Isolinienplots (T)
C      UNIT 21  optionale Ausgabedatei fuer Y - Z Isolinienplots (H)
C      UNIT 22  optionale Ausgabedatei fuer X - Z Isolinienplots (H)
C      UNIT 23  optionale Ausgabedatei fuer X - Y Isolinienplots (H)
C
C      IMSBIN = 0          ==>      formatierte Ausgabe fuer Postprozessor
C      IMSBIN = 0          ==>      MS - Binaerformat fuer Postprozessor
C
FORM = 'FORMATTED'
IF (IMSBIN.EQ.1) FORM = 'BINARY'
IF (IGWF.LE.1) THEN
  IF (IXISO.EQ.1) OPEN(11,FILE='TEMPYZ',FORM=FORM)
  IF (IYISO.EQ.1) OPEN(12,FILE='TEMPXZ',FORM=FORM)
  IF (IZISO.EQ.1) OPEN(13,FILE='TEMPXY',FORM=FORM)
END IF
IF (IGWF.GE.1) THEN
  IF (IXISO.EQ.1) OPEN(21,FILE='PIEYZ',FORM=FORM)
  IF (IYISO.EQ.1) OPEN(22,FILE='PIEZYZ',FORM=FORM)
  IF (IZISO.EQ.1) OPEN(23,FILE='PIEZXY',FORM=FORM)
END IF
C
C *** Bestimmung der bei IAUSOP < 3 zu druckenden Profilschnitte *****
C
CARD = '15'
READ (5,20,END=900) FORMIN
READ (5,FORMIN,ERR=800,END=900) IPROX,IPROY,IPROZ
IF (ZFLAG.EQ.ZEIT(1)) DZMF = 1.
IF (ZFLAG.EQ.ZEIT(2)) DZMF = 60.
IF (ZFLAG.EQ.ZEIT(3)) DZMF = 3600.
IF (ZFLAG.EQ.ZEIT(4)) DZMF = 86400.
DZEIT = ISTLAE * DZMF
IF (DZF.LE.0.) DZF = 1.
DSIM = DZEIT / 3600.
DZT = DSIM
DO 690 I = 1, IZEIT - 1
  DZT = DZT * DZF
  DSIM = DSIM + DZT
690 CONTINUE
DMIN = DZEIT / 60.
IF (IOBS.GT.0) THEN
  CARD = '15b'
  READ (5,20,END=900) FORMIN
  DO 700 I = 1, IOBS
    READ (5,FORMIN,ERR=800,END=900) IXOBS(I),IYOBS(I),IZOBS(I)
700 CONTINUE
  END IF
  CARD = '17'
  READ (5,20,END=900) FORMIN
  READ (5,FORMIN,ERR=800,END=900) NX,NY,NZ
  IF (NX.GT.IX.OR.NY.GT.IY.OR.NZ.GT.IZ.OR.MINO(NX,NY,NZ).LT.1) THEN
    WRITE (1,175)
    WRITE (1,180) NX,NY,NZ
    WRITE (1,185) IX,IY,IZ
    STOP ' Bitte Dimensionierung anpassen !'
  
```

```

END IF
IF ((IGWF.GE.1.AND.IP2.EQ.0).OR.(IGWF.LE.1.AND.IP1.EQ.0)) THEN
  IF (IP1.EQ.0) STOP ' Nur Stroemungsberechnung moeglich IP1 =0)'
  IF (IP2.EQ.0) STOP ' Nur Temperaturberechnung moeglich IP2 =0)'
END IF
IF (IFROST.EQ.1.AND.IGWF.LE.1) THEN
C
C *** Schmelztemperatur des Fluids einheitlich konstant 0 Grad Celsius *
C
      DO 61 K = 1, NZ
          STMP(K) = CELV(0.0D00)
61    CONTINUE
      END IF
C
C *** Einlesen der Schmelztemperaturen *****
C
      IF (IFROST.EQ.3.AND.IGWF.LE.1) THEN
          READ (5,20,END=900) FORMIN
          CARD = '17b'
          READ (5,FORMIN,ERR=800,END=900) (STMP(K),K = 1, NZ)
          DO 70 K = 1, NZ
              STMP(K) = CELV(STMP(K))
70    CONTINUE
          END IF
C
C *** Einlesen der Gefriermodelle ... *****
C
      IF (IFROST.GT.0.AND.IGWF.LE.1) THEN
          READ (5,20,END=900) FORMIN
          CARD = '17d'
          READ (5,FORMIN,ERR=800,END=900) (IFRO(K),K = 1, NZ)
          READ (5,20,END=900) FORMIN
C
C      Gefriermodell 1:  Y = FRO(1) * EXP(FRO(2) * X) + FRO(3)          *
C      -----
C      es bedeuten:          Y = Anteil de ungefrorenen Wassers
C                          X = Temperatur in K unter Schmelztemperatur
C
      DO 71 K = 1, NZ
          IF (IFRO(K).EQ.1) THEN
              CARD = '17f'
              READ (5,FORMIN,ERR=800,END=900) (FRO(K,L),L = 1, 3)
          END IF
71    CONTINUE
      END IF
      NX1 = NX - 1
      NY1 = NY - 1
      NZ1 = NZ - 1
C
C *** Ausgabe der globalen Eingabeparameter *****
C
      IF (IAUS.EQ.6) CALL HEADER(IAUS)
      WRITE (IAUS,15) TITEL
      WRITE (IAUS,600)

```

```

WRITE (IAUS,180) NX,NY,NZ
WRITE (IAUS,185) IX,IY,IZ
IF (DZF.EQ.1.) WRITE (IAUS,30) IZEIT,DMIN,DSIM
IF (DZF.NE.1.) WRITE (IAUS,35) IZEIT,DMIN,DZF,DSIM
IF (IOBS.GT.0) THEN
  WRITE (IAUS,50) IOBS
  DO 710 I = 1, IOBS
    WRITE (IAUS,55) IXOBS(I),IYOBS(I),IZOBS(I)
710  CONTINUE
  WRITE (IAUS,60) OBSFIL
END IF
IF (IGWF.EQ.2) GO TO 520
TEXT = 'Temperatur'
WRITE (IAUS,400) TEXT
IF (OMGT.EQ.0.) OMGT = -1.
IF (OMGT.EQ.1.) WRITE (IAUS,420)
IF (OMGT.GT.0..AND.OMGT.NE.1.) WRITE (IAUS,430) OMGT
IF (OMGT.EQ.-1.) WRITE (IAUS,440)
IF (OMGT.LT.0..AND.OMGT.NE.-1.) WRITE (IAUS,450) -OMGT
TEXT = 'Grad Kelvin'
WRITE (IAUS,160) FEHLT,TEXT
TEXT = 'ITERAT'
IF (OMGT.LT.0.) TEXT = 'IADITP'
WRITE (IAUS,170) ILOOP,TEXT
IF (OMGT.GT.0.AND.IALTER.EQ.0) WRITE (IAUS,340)
IF (OMGT.GT.0.AND.IALTER.EQ.1) WRITE (IAUS,350)
IF (ALFATP.EQ.0.) TEXT = 'Explizit Schema'
IF (ALFATP.GT.0..AND.ALFATP.NE.1..AND.ALFATP.NE.0.5)
& TEXT = 'Implizit Schema'
IF (ALFATP.EQ.0.5) TEXT = 'Crank - Nicolson - Schema'
IF (ALFATP.EQ.1.) TEXT = 'Vollimplizit Schema'
WRITE (IAUS,40) TEXT,ALFATP
520 IF (IGWF.EQ.0) GO TO 525
TEXT = 'Stroemungs'
WRITE (IAUS,400) TEXT
IF (OMGH.EQ.0.) OMGH = -1.
IF (OMGH.EQ.1.) WRITE (IAUS,420)
IF (OMGH.GT.0..AND.OMGH.NE.1.) WRITE (IAUS,430) OMGH
IF (OMGH.EQ.-1.) WRITE (IAUS,440)
IF (OMGH.LT.0..AND.OMGH.NE.-1.) WRITE (IAUS,450) -OMGH
TEXT = 'Meter'
WRITE (IAUS,160) FEHLH,TEXT
TEXT = 'ITERAH'
IF (OMGH.LT.0.) TEXT = 'IADIST'
WRITE (IAUS,170) ILOOP,TEXT
IF (OMGH.GT.0.AND.IALTER.EQ.0) WRITE (IAUS,340)
IF (OMGH.GT.0.AND.IALTER.EQ.1) WRITE (IAUS,350)
IF (ALFAHY.EQ.0.) TEXT = 'Explizit Schema'
IF (ALFAHY.GT.0..AND.ALFAHY.NE.1..AND.ALFAHY.NE.0.5)
& TEXT = 'Implizit Schema'
IF (ALFAHY.EQ.0.5) TEXT = 'Crank - Nicolson - Schema'
IF (ALFAHY.EQ.1.) TEXT = 'Vollimplizit Schema'
WRITE (IAUS,40) TEXT,ALFAHY
525 IF (IGWF.EQ.2) GO TO 535

```

```

IF (IDIF.EQ.0) GO TO 530
WRITE (IAUS,310)
IF (IPECL.EQ.0) IPECL = 5
IF (IPECL.EQ.1) TEXT = 'durch Zentrale Differenzen'
IF (IPECL.EQ.2) TEXT = 'nach dem Upwind - Schema'
IF (IPECL.EQ.3) TEXT = 'nach dem Hybrid - Schema'
IF (IPECL.EQ.4) TEXT = 'nach dem Power law - Schema'
IF (IPECL.EQ.5) TEXT = 'nach dem Exponential - Schema'
WRITE (IAUS,110) TEXT
530 IF (IDIF.EQ.0) WRITE (IAUS,300)
IF (IFROST.EQ.0) TEXT = 'ignoriert'
IF (IFROST.GE.1) TEXT = 'beruecksichtigt'
WRITE (IAUS,320) TEXT
WRITE (IAUS,360) CPW,CPE,CPL,WLW,WLE,WLL,RHOW,RHOE,RHOL,SENTHA,
&
DTW,DWAN * 100.
535 IF (IVOLDF.EQ.0) WRITE (IAUS,90)
IF (IVOLDF.EQ.1) WRITE (IAUS,100)
RETURN
800 WRITE (1,*) 'fehlerhaftes Datenformat in Karte ',CARD
STOP 'Bitte Datei ueberpruefen!'
900 WRITE (1,*) 'unerwartet Dateieinde erreicht nach Karte ',CARD
STOP 'Bitte Datei ergaenzen!'
10 FORMAT(A80)
15 FORMAT(1H ,A80)
20 FORMAT(A20)
30 FORMAT(/,I4,' Zeitschritte ',F7.1,' Minuten Zeitschrittlaenge ',
&
F6.1,' Stunden Simulation')
35 FORMAT(/,I4,' Zeitschritte ',F8.2,' Minuten Anfangszeitritt',
&
'laenge',/,F7.4,' Zeitschrittfaktor ',F7.1,' Stunden ',
&
'Simulation')
40 FORMAT(/,' Loesung der DGL nach dem ',A26,
&
' - Wichtungsfaktor => ',F4.2)
50 FORMAT(/,' Die Temperaturen bzw. Piezometerhoeehen folgender ',
&
I2,' Knoten:',//,T20,' X Y Z ',//,
&
T20,21(1H-))
55 FORMAT(T17,3I7)
60 FORMAT(T20,21(1H-),//,' werden auf der Datei ',A14,' ausgegeben.')
90 FORMAT(/,' Knotenpunkte liegen im geometrischen Zentrum der Zellen
&')
100 FORMAT(/,' Zellengrenzen liegen auf halber Strecke zwischen den Kn
&oten')
110 FORMAT(/,' Berechnung der Peclet-Wichtungsfunktion ',A30)
160 FORMAT(/,' erlaubter Iterationsfehler +/- ',F9.7,1X,A11)
170 FORMAT(/,' jedoch maximal ',I4,' Durchlaeufer von Subroutine ',A6)
175 FORMAT(' Konflikt in der Knotendimensionierung:',/,1H ,38(1H=))
180 FORMAT(/,' Anzahl der Knoten: NX ==> ',I4,' NY ==> ',I4,
&' NZ ==> ',I4,/)
185 FORMAT(/,' Programmdimensionen: IX ==> ',I4,' IY ==> ',I4,
&' IZ ==> ',I4,/)
220 FORMAT(' Eingabedatenfile eingeben ==> ',\ )
300 FORMAT(/,' reines Konduktionsproblem')
310 FORMAT(/,' Diffusion und Konvektion werden beruecksichtigt')
320 FORMAT(/,' Phasenaenderung von Wasser wird ',A40)
340 FORMAT(/,' Iteration erfolgt in aufsteigender Reihenfolge')

```

```

IF (IGWF.LE.1) THEN
  IF (IXISO.EQ.1) CALL ISOASC (11,IY,IZ,NY,NZ,DIFY,DIFZ)
  IF (IYISO.EQ.1) CALL ISOASC (12,IX,IZ,NX,NZ,DIFX,DIFZ)
  IF (IZISO.EQ.1) CALL ISOASC (13,IX,IY,NX,NY,DIFX,DIFY)
END IF
IF (IGWF.GE.1) THEN
  IF (IXISO.EQ.1) CALL ISOASC (21,IY,IZ,NY,NZ,DIFY,DIFZ)
  IF (IYISO.EQ.1) CALL ISOASC (22,IX,IZ,NX,NZ,DIFX,DIFZ)
  IF (IZISO.EQ.1) CALL ISOASC (23,IX,IY,NX,NY,DIFX,DIFY)
END IF
ISO1 = 1
RETURN
10 IF (IGWF.LE.1) THEN
  IF (IXISO.EQ.1) CALL ISOBIN (11,IY,IZ,NY,NZ,DIFY,DIFZ)
  IF (IYISO.EQ.1) CALL ISOBIN (12,IX,IZ,NX,NZ,DIFX,DIFZ)
  IF (IZISO.EQ.1) CALL ISOBIN (13,IX,IY,NX,NY,DIFX,DIFY)
END IF
IF (IGWF.GE.1) THEN
  IF (IXISO.EQ.1) CALL ISOBIN (21,IY,IZ,NY,NZ,DIFY,DIFZ)
  IF (IYISO.EQ.1) CALL ISOBIN (22,IX,IZ,NX,NZ,DIFX,DIFZ)
  IF (IZISO.EQ.1) CALL ISOBIN (23,IX,IY,NX,NY,DIFX,DIFY)
END IF
ISO1 = 1
RETURN
END
C&ISOASC
C *****
C   Version 02011988   Subroutine ISOASC
C *****
C   *** Subroutine ISOASC erzeugt Eingangsdaten fuer den Postprozessor ***
C
SUBROUTINE ISOASC (ISOUN,II,JJ,NI,NJ,DIFI,DIFJ)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,T0,STMP,CELS
DIMENSION DIFI(II),DIFJ(JJ)
DIMENSION T(JX,JY,JZ),T0(JX,JY,JZ),STMP(JZ)
DIMENSION H(KX,KY,KZ),H0(KX,KY,KZ)
COMMON /TEMP/T,T0,STMP
COMMON /WPOT/H,H0
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /IPRO/IPROX,IPROY,IPROZ
COMMON /ISOP/IXISO,IYISO,IZISO,ISO1,IMSBIN
C
IF (IECHO.GE.1) CALL ROUTIN ('ISOASC')
C
IF (ISO1.EQ.0) THEN
  WRITE (ISOUN,100) NI,NJ
  WRITE (ISOUN,110) (DIFI(I),I = 1, NI-1)

```

```

        WRITE (ISOUN,110) (DIFJ(J),J = 1, NJ-1)
    END IF
    WRITE (ISOUN,120) SUMSEK
    IF (ISOUN.EQ.11) THEN
        DO 10 J = 1, NJ
            WRITE (ISOUN,110) (CELS(T(IPROX,I,J)),I = 1, NI)
10     CONTINUE
    END IF
    IF (ISOUN.EQ.12) THEN
        DO 20 J = 1, NJ
            WRITE (ISOUN,110) (CELS(T(I,IPROY,J)),I = 1, NI)
20     CONTINUE
    END IF
    IF (ISOUN.EQ.13) THEN
        DO 30 J = 1, NJ
            WRITE (ISOUN,110) (CELS(T(I,J,IPROZ)),I = 1, NI)
30     CONTINUE
    END IF
    IF (ISOUN.EQ.21) THEN
        DO 40 J = 1, NJ
            WRITE (ISOUN,110) (H(IPROX,I,J),I = 1, NI)
40     CONTINUE
    END IF
    IF (ISOUN.EQ.22) THEN
        DO 50 J = 1, NJ
            WRITE (ISOUN,110) (H(I,IPROY,J),I = 1, NI)
50     CONTINUE
    END IF
    IF (ISOUN.EQ.23) THEN
        DO 60 J = 1, NJ
            WRITE (ISOUN,110) (H(I,J,IPROZ),I = 1, NI)
60     CONTINUE
    END IF
100  FORMAT(2I5)
110  FORMAT(6(10F7.2,/))
120  FORMAT(F12.2)
    RETURN
    END

```

C&ISOBIN

```

C *****
C   Version 12091988  Subroutine ISOBIN
C *****

```

```

C *** Subroutine ISOBIN erzeugt Eingangsdaten fuer den Postprozessor ***
C

```

```

SUBROUTINE ISOBIN (ISOUN,II,JJ,NI,NJ,DIFI,DIFJ)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,T0,STMP,CELS
DIMENSION DIFI(II),DIFJ(JJ)
DIMENSION T(JX,JY,JZ),T0(JX,JY,JZ),STMP(JZ)
DIMENSION H(KX,KY,KZ),H0(KX,KY,KZ)

```

```

COMMON /TEMP/T,TO,STMP
COMMON /WPOT/H,H0
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /IPRO/IPROX,IROY,IPOZ
COMMON /ISOP/IXISO,IYISO,IZISO,ISO1,IMSBIN

```

C

```
IF (IECHO.GE.1) CALL ROUTIN ('ISOBIN')
```

C

```

IF (ISO1.EQ.0) THEN
  WRITE (ISOUN) 'BINARY'
  WRITE (ISOUN) NI,NJ
  WRITE (ISOUN) (DIFI(I),I = 1, NI-1)
  WRITE (ISOUN) (DIFJ(J),J = 1, NJ-1)
END IF
WRITE (ISOUN) SUMSEK
IF (ISOUN.EQ.11) THEN
  DO 10 J = 1, NJ
    WRITE (ISOUN) (REAL(CELS(T(IPROX,I,J))),I = 1, NI)
10  CONTINUE
END IF
IF (ISOUN.EQ.12) THEN
  DO 20 J = 1, NJ
    WRITE (ISOUN) (REAL(CELS(T(I,IROY,J))),I = 1, NI)
20  CONTINUE
END IF
IF (ISOUN.EQ.13) THEN
  DO 30 J = 1, NJ
    WRITE (ISOUN) (REAL(CELS(T(I,J,IPOZ))),I = 1, NI)
30  CONTINUE
END IF
IF (ISOUN.EQ.21) THEN
  DO 40 J = 1, NJ
    WRITE (ISOUN) (H(IPROX,I,J),I = 1, NI)
40  CONTINUE
END IF
IF (ISOUN.EQ.22) THEN
  DO 50 J = 1, NJ
    WRITE (ISOUN) (H(I,IROY,J),I = 1, NI)
50  CONTINUE
END IF
IF (ISOUN.EQ.23) THEN
  DO 60 J = 1, NJ
    WRITE (ISOUN) (H(I,J,IPOZ),I = 1, NI)
60  CONTINUE
END IF
RETURN
END

```

C&ITERAH

```

C *****
C   Version 22031988   Subroutine ITERAH
C *****
C

```

C *** Subroutine ITERAH iterative Loesung des Stroemungsfeldes *****

```
C
SUBROUTINE ITERAH
PARAMETER (IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2)
PARAMETER (IP1=1, IP2=1, IQ1=0, IQ2=0)
PARAMETER (JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2, KY=IY*IP2+IQ2, KZ=IZ*IP2+IQ2)
INTEGER*2 IFLAG, JFLAG, IFLUS, IFRO
DOUBLE PRECISION FEHL
DIMENSION H(KX, KY, KZ), HO(KX, KY, KZ)
DIMENSION DELTAX(IX), DELTAY(IY), DELTAZ(IZ), DIFX(IX), DIFY(IY),
&          DIFZ(IZ), XKOR(IX), YKOR(IY), ZKOR(IZ), DIVX(IX, 2),
&          DIVY(IY, 2), DIVZ(IZ, 2), FX(IX), FY(IY), FZ(IZ),
&          DXYZ(IX, IY, IZ)
DIMENSION IFLAG(IX, IY, IZ), JFLAG(KX, KY, KZ), IFLUS(JX, JY, JZ), IFRO(JZ)
DIMENSION AP1(IX, IY, IZ), APO(IX, IY, IZ), ANB(IX, IY, IZ, KD),
&          FNB(KD), DNB(KD), PECL(KD), B(IX, IY, IZ)
DIMENSION INDX(6), INDY(6), INDZ(6), IADD(4), JADD(4), KADD(4)
COMMON /GITT/DELTAX, DELTAY, DELTAZ, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,
&          DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ
COMMON /IFLD/IFLAG, JFLAG, IFLUS, IFRO
COMMON /INCB/INDX, INDY, INDZ, IADD, JADD, KADD
COMMON /INF1/IZEIT, IPECL, IVOLDF, IAUSOP, ILOOP, IEIN, INPOP, IALTER
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
COMMON /INF3/LUMW, IENER, ISTOP, IDATOP, LISDAT, IDZTST
COMMON /KOEUF/AP1, APO, ANB, FNB, DNB, PECL, B
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /PAR1/DZEIT, DZMF, DZF, FLURAB, ALFATP, FEHLT, FEHLER, OMGT, SUMSEK
COMMON /PAR2/BASIS, HOENN, ALFAHY, OMGH, FEHLH, ZZZZ, ZIGNO
COMMON /WPOT/H, HO
```

```
C
90 IF (IECHO.GE.1) CALL ROUTIN ('ITERAH')
```

```
C
OMEGA = OMGH
LMAX = KD
IF (NZ.EQ.1) LMAX = 4
IF (NZ.EQ.1.AND.NY.EQ.1) LMAX = 2
IAUSOP = 1
NIT = 0
M = 0
IF (IECHO.GE.1) WRITE (1,*)
100 NIT = NIT + 1
FELALT = 0.
M = M + 1
CALL RICHTG (M, KANF, KEND, KPLUS, NZ, IALTER)
DO 10 K = KANF, KEND, KPLUS
IF (NZ.EQ.1) N = M
CALL RICHTG (N, JANF, JEND, JPLUS, NY, IALTER)
HU = HOENN - ZKOR(K) - DIVZ(K, 2)
HO = HU + DELTAZ(K)
DO 10 J = JANF, JEND, JPLUS
CALL RICHTG (J, IANF, IEND, IPLUS, NX, IALTER)
DO 10 I = IANF, IEND, IPLUS
```

```

        GO TO (1000,10) JFLAG(I,J,K)
C
C *** IFLAG = 1 ==> variables Potential *****
C
1000 AH = 0.
      DO 20 L = 1, LMAX
      II = I + INDX(L)
      JJ = J + INDY(L)
      KK = K + INDZ(L)
      IF (II.EQ.0.OR.II.EQ.NX+1) GO TO 20
      IF (JJ.EQ.0.OR.JJ.EQ.NY+1) GO TO 20
      IF (KK.EQ.0.OR.KK.EQ.NZ+1) GO TO 20
      CALL WICHHY (II,JJ,KK,HH)
      AH = AH + ANB(I,J,K,L) * HH
20 CONTINUE
      HALT = H(I,J,K)
      H(I,J,K) = HALT + OMEGA * ((AH + B(I,J,K)) / AP1(I,J,K) - HALT)
      HNEU = H(I,J,K)
      IF (HALT.LT.HO.AND.HNEU.GT.HO) CALL DELTAH (I,J,K)
      IF (HNEU.LT.HO) CALL DELTAH (I,J,K)
      FELNEU = ABS(HALT - HNEU)
      IF (FELNEU.GT.FELALT) FELALT = FELNEU
10 CONTINUE
      IF (IECHO.GE.1) WRITE (1,30) NIT,FELALT
      IF (FELALT.GT.FEHLH.AND.NIT.LT.ILOOP) GO TO 100
      FEHLER = FELALT
      FEHL = FELALT
      IF (ISOR.EQ.2) THEN
        ALFA = ALFAHY
        CALL OMEGAT(FEHL,OMEGA,ALFA)
        IF (ISOR.NE.1) THEN
          CALL KOEFFH
          GO TO 90
        END IF
      END IF
30 FORMAT(1H+,' Iteration Nummer ==> ',I3,' === Iterationsfehler = ',
&      F11.6)
      RETURN
      END
C&ITERAT
C *****
C      Version 08071987 Subroutine ITERAT
C *****
C *** Subroutine ITERAT iterative Loesung des Temperaturfeldes *****
C
      SUBROUTINE ITERAT
      PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
      PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
      PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
      PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
      PARAMETER (IWERT=400)
      DOUBLE PRECISION T,T0,TALT,TNEU,TT,STMP,FELALT,FELNEU,AT
      INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO

```

```

INTEGER*2 IXX,IYY,IZZ,IUFL
CHARACTER FLAG*4
DIMENSION IFLAG(IX,IY,IZ),JFLAG(KX,KY,KZ),IFLUS(JX,JY,JZ),IFRO(JZ)
DIMENSION T(JX,JY,JZ),TO(JX,JY,JZ),STMP(JZ)
DIMENSION AP1(IX,IY,IZ),APO(IX,IY,IZ),ANB(IX,IY,IZ,KD),
& FNB(KD),DNB(KD),PECL(KD),B(IX,IY,IZ)
DIMENSION WERT(IWERT),IXX(IWERT),IYY(IWERT),IZZ(IWERT),
& FLAG(IWERT),IUFL(IWERT)
DIMENSION INDX(6),INDY(6),INDZ(6),IADD(4),JADD(4),KADD(4)
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /INCB/INDX,INDY,INDZ,IADD,JADD,KADD
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /KOEK/AP1,APO,ANB,FNB,DNB,PECL,B
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /TEMP/T,TO,STMP
COMMON /ZYK1/WERT,IXX,IYY,IZZ,IUFL
COMMON /ZYK2/FLAG

```

```

C
100 IF (IECHO.GE.1) CALL ROUTIN ('ITERAT')
C
OMEGA = OMGT
LMAX = KD
IF (NZ.EQ.1) LMAX = 4
IF (NZ.EQ.1.AND.NY.EQ.1) LMAX = 2
IAUSOP = 1
NIT = 0
M = 0
IF (IECHO.GE.1) WRITE (1,*)
5000 NIT = NIT + 1
FELALT = 0.
M = M + 1
CALL RICHTG (M,KANF,KEND,KPLUS,NZ,IALTER)
DO 10 K = KANF, KEND, KPLUS
IF (NZ.EQ.1) N = M
CALL RICHTG (N,JANF,JEND,JPLUS,NY,IALTER)
DO 10 J = JANF, JEND, JPLUS
CALL RICHTG (J,IANF,IEND,IPLUS,NX,IALTER)
DO 10 I = IANF, IEND, IPLUS
C
GO TO (1000,10,10,10,10) IABS(IFLAG(I,J,K))
C
C *** IFLAG = 1 ==> variable Temperatur *****
C
1000 AT = 0.
DO 20 L = 1, LMAX
II = I + INDX(L)
JJ = J + INDY(L)
KK = K + INDZ(L)
IF (II.EQ.0.OR.II.EQ.NX+1) GO TO 20
IF (JJ.EQ.0.OR.JJ.EQ.NY+1) GO TO 20
IF (KK.EQ.0.OR.KK.EQ.NZ+1) GO TO 20

```

```

CALL WICHTP (II,JJ,KK,TT)
AT = AT + ANB(I,J,K,L) * TT
20 CONTINUE
TALT = T(I,J,K)
T(I,J,K) = TALT + OMEGA * ((AT + B(I,J,K)) / AP1(I,J,K) - TALT)
IF (IFROST.GE.1) CALL PHATST (I,J,K,TALT,T(I,J,K))
FELNEU = DABS(T(I,J,K) - TALT)
IF (FELNEU.GT.FELALT) FELALT = FELNEU
10 CONTINUE
IF (IWCP.GE.3) CALL KOEFFT (1,1,1,NX,NY,NZ)
IF (IECHO.GE.1) WRITE (1,30) NIT,FELALT
IF (FELALT.GT.FEHLT.AND.NIT.LT.ILOOP) GO TO 5000
FEHLER = FELALT
IF (ISOR.EQ.1) THEN
  ALFA = ALFATP
  CALL OMEGAT(FELALT,OMEGA,ALFA)
  IF (ISOR.NE.0) GO TO 100
END IF
30 FORMAT(1H+,' Iteration Nummer ==> ',I3,' == Iterationsfehler = ',
&      F11.6)
RETURN
END
C&KOEFFH
C *****
C      Version 22031988  Subroutine KOEFFH
C *****
C
C *** Subroutine KOEFFH Berechnung der Nachbarkoeffizienten fuer H *****
C
SUBROUTINE KOEFFH
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DIMENSION AP1 (IX,IY,IZ),APO (IX,IY,IZ),ANB (IX,IY,IZ,KD),
&      FNB(KD),DNB(KD),PECL(KD),B (IX,IY,IZ)
DIMENSION IXOBS(10),IYOBS(10),IZOBS(10)
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
COMMON /KOE/ AP1,APO,ANB,FNB,DNB,PECL,B
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
C
IF (IECHO.GE.1) CALL ROUTIN ('KOEFFH')
C
IF (IGWF.LE.1) CALL FLUSFL(1,1,1,NX,NY,NZ)
DO 10 K = 1, NZ
DO 10 J = 1, NY
DO 10 I = 1, NX
CALL KONDUK (I,J,K)
10 CONTINUE
IF (IECHO.GE.2) WRITE (1,*)
DO 20 K = 1, NZ
IF (IECHO.EQ.1) WRITE (1,100) K
DO 20 J = 1, NY

```

```

DO 20 I = 1, NX
CALL KONKOF (I,J,K)
DO 20 L = 1, KD
ANB(I,J,K,L) = DNB(L)
20 CONTINUE
100 FORMAT(1H+,T17,'***** Schicht ',I4,' wird berechnet ****')
RETURN
END
C&KOEFFT
C *****
C Version 30061987 Subroutine KOEFFT
C *****
C
C *** Subroutine KOEFFT Berechnung der Nachbarkoeffizienten fuer T *****
C
SUBROUTINE KOEFFT (I1,J1,K1,I2,J2,K2)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,T0,STMP,CELS
INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
DIMENSION DELTAX (IX),DELTAY (IY),DELTAZ (IZ),DIFX (IX),DIFY (IY),
& DIFZ (IZ),XKOR (IX),YKOR (IY),ZKOR (IZ),DIVX (IX,2),
& DIVY (IY,2),DIVZ (IZ,2),FX (IX),FY (IY),FZ (IZ),
& DXYZ (IX,IY,IZ)
DIMENSION T (JX,JY,JZ),T0 (JX,JY,JZ),STMP (JZ)
DIMENSION CPG (JX,JY,JZ),WLKG (JX,JY,JZ),WTI (IX,IY,IZ,KD2),
& RHOG (JX,JY,JZ),PHIG (JX,JY,JZ),PHIN (IX,IY,IZ),
& SATU (IZ),VX (IX,IY,IZ),VY (IX,IY,IZ),VZ (IX,IY,IZ),ANISO (3)
DIMENSION SC (JX,JY,JZ),SP (JX,JY,JZ),FRO (JZ,5)
DIMENSION AP1 (IX,IY,IZ),AP0 (IX,IY,IZ),ANB (IX,IY,IZ,KD),
& FNB (KD),DNB (KD),PECL (KD),B (IX,IY,IZ)
DIMENSION IXOBS (10),IYOBS (10),IZOBS (10)
DIMENSION IFLAG (IX,IY,IZ),JFLAG (KX,KY,KZ),IFLUS (JX,JY,JZ),IFRO (JZ)
COMMON /ENER/SC,SP,FRO
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
& DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
COMMON /KOEK/AP1,AP0,ANB,FNB,DNB,PECL,B
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /PAR2/BASIS,HOENN,ALFAHY,OMGH,FEHLH,ZZZZ,ZIGNO
COMMON /PAR3/CPW,CPE,CPL,WLW,WLE,WLL,RHOW,RHOE,RHOL,SENTHA
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
COMMON /TEMP/T,T0,STMP
C
IF (IECHO.GE.1) CALL ROUTIN ('KOEFFT')
C
LMAX = KD2

```

```

      IF (NZ.EQ.1) LMAX = 2
      IF (NZ.EQ.1.AND.NY.EQ.1) LMAX = 1
      LM2 = LMAX * 2
      DO 80 K = K1, K2
      DO 80 J = J1, J2
      DO 80 I = I1, I2
        CALL KONINF (I,J,K)
80 CONTINUE
      IF (IDIF.EQ.1) THEN
C *** Diffusion und Konvektion *****
        IF (IGWF.EQ.0) CALL FLUSFL (I1,J1,K1,I2,J2,K2)
        IF (IECHO.GE.2) WRITE (1,*)
        DO 10 K = K1, K2
        IF (IECHO.GE.2) WRITE (1,100) K
        DO 10 J = J1, J2
        DO 10 I = I1, I2
          CALL KONKOF (I,J,K)
          CALL FLUKOF (I,J,K)
          CALL PECLLET (LM2)
          LL = 0
          DO 10 L = 1, LMAX
            LL = LL + 1
            ANB(I,J,K,LL) = DNB(LL) * PECL(LL) + AMAX1( FNB(LL), 0.)
            LL = LL + 1
            ANB(I,J,K,LL) = DNB(LL) * PECL(LL) + AMAX1(-FNB(LL), 0.)
10 CONTINUE
        ELSE
C *** nur Diffusion *****
          IF (IECHO.GE.2) WRITE (1,*)
          DO 70 K = K1, K2
          IF (IECHO.GE.2) WRITE (1,100) K
          DO 70 J = J1, J2
          DO 70 I = I1, I2
            CALL KONKOF (I,J,K)
            DO 70 L = 1, LM2
              ANB(I,J,K,L) = DNB(L)
70 CONTINUE
        END IF
C
        DO 20 K = K1, K2
        STP = CELS(STMP(K))
        DO 20 J = J1, J2
        DO 20 I = I1, I2
C
C *** konstante Temperatur *****
C
        IF (IFLAG(I,J,K).EQ.2.OR.IFLAG(I,J,K).EQ.5) THEN
          APO(I,J,K) = ZZZZ
        ELSE
          TMP = CELS(T(I,J,K))
          ROG = RHOG(I,J,K)
          POR = PHIG(I,J,K)
          SOL = 1. - PHIG(I,J,K)
          ROP = RHOH2O(TMP,STP,K) * SATU(K) + RHOL * (1. - SATU(K))

```

```

C
C *** nur Diffusion *****
C
      IF (IDIF.EQ.0) THEN
      IF (TMP.GE.STP) THEN
        CPP = CPW * SATU(K) + CPL * (1. - SATU(K))
      ELSE
        CALL WCPH20 (CPF,TMP-STP,K)
        CPP = CPF * SATU(K) + CPL * (1. - SATU(K))
      END IF
      ROPHI = CPP * ROP * POR + CPG(I,J,K) * ROG * SOL
      APO(I,J,K) = DXYZ(I,J,K) * ROPHI / DZEIT
    ELSE
C
C *** Diffusion und Konvektion *****
C
      ROPHI = ROP * POR + ROG * SOL
      APO(I,J,K) = DXYZ(I,J,K) * ROPHI / DZEIT
    END IF
  END IF
20 CONTINUE
C
C *** Addieren aller Nachbarkoeffizienten *****
C
  DO 40 K = K1, K2
  DO 40 J = J1, J2
  DO 40 I = I1, I2
    AP = 0.
    DO 30 L = 1, LM2
      AP = AP + ANB(I,J,K,L)
    30 CONTINUE
    AP1(I,J,K) = AP + APO(I,J,K)
  40 CONTINUE
  RETURN
100 FORMAT(1H+,T17,'***** Schicht ',I4,' wird berechnet ****')
  END
C&KONDUK
C *****
C   Version 21031988   Subroutine KONDUK
C *****
C *** Subroutine KONDUK berechnet die Interfacekonduktivitaet *****
C
  SUBROUTINE KONDUK (I,J,K)
  PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
  PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
  PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
  PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
  INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
  DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
&           DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
&           DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
&           DXYZ(IX,IY,IZ)
  DIMENSION IFLAG(IX,IY,IZ),JFLAG(KX,KY,KZ),IFLUS(JX,JY,JZ),IFRO(JZ)

```

```

DIMENSION H(KX,KY,KZ),HO(KX,KY,KZ)
DIMENSION CPG(JX,JY,JZ),WLKG(JX,JY,JZ),WTI(IX,IY,IZ,KD2),
&          RHOG(JX,JY,JZ),PHIG(JX,JY,JZ),PHIN(IX,IY,IZ),
&          SATU(IZ),VX(IX,IY,IZ),VY(IX,IY,IZ),VZ(IX,IY,IZ),ANISO(3)
DIMENSION TTX(KX,KY,KZ),TTY(KX,KY,KZ),TTZ(KX,KY,KZ),
&          SK(KX,KY,KZ),Q(KX,KY,KZ)
DIMENSION IXOBS(10),IYOBS(10),IZOBS(10)
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
&          DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
COMMON /INF5/IVIS,IWRO
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /PAR2/BASIS,HOENN,ALFAHY,OMGH,FEHLH,ZZZZ,ZIGNO
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
COMMON /TRNS/TTX,TTY,TTZ,Q,SK
COMMON /WPOT/H,HO

```

```

C
C   TREST = 1E-5
C
C *** Restpermeabilitaet, falls die Zellen leerlaufen ... *****
C
C   HU = HOENN - ZKOR(K) - DIVZ(K,2)
C   HO = HU + DELTAZ(K)
C
C *** Berechnung der benetzten, vertikalen Zellengrenzhoehe *****
C
C   TKX1 = TTX(I,J,K)
C   TKY1 = TTY(I,J,K)
C   TKZ1 = TTZ(I,J,K)
C   IF (K.LT.NZ) THEN
C     TKZ2 = TTZ(I,J,K+1)
C     IF (IFLUS(I,J,K+1).EQ.0.AND.IGWF.LE.1) TKZ2 = 0.
C   END IF
C   IF (IFLUS(I,J,K).EQ.0.AND.IGWF.LE.1) THEN
C     TKX1 = 0.
C     TKY1 = 0.
C     TKZ1 = 0.
C   END IF
C   IF (I.EQ.NX) THEN
C     TRX = 0.
C     GO TO 10
C   END IF
C   TKX2 = TTX(I+1,J,K)
C   CALL HARMON (TKX1,TKX2,TKX,FX(I))
C   IF (H(I,J,K).LT.HO.OR.H(I+1,J,K).LT.HO) THEN
C
C *** gewichtetes, geometrisches Mittel ... *****
C
C   IF (H(I,J,K).GT.HU) THEN

```

```

      DHX = (H(I,J,K) - HU) ** (1. - FX(I)) *
&      MIN(ABS(H(I+1,J,K) - HU), DELTAZ(K)) ** FX(I)
      TRX = TKX * DHX / DELTAZ(K)
      ELSE
        TRX = TREST
      END IF
    ELSE
      TRX = TKX
    END IF
10 IF (J.EQ.NY) THEN
      TRY = 0.
      GO TO 20
    END IF
      TKY2 = TTY(I,J+1,K)
      CALL HARMON (TKY1,TKY2,TKY,FY(J))
      IF (H(I,J,K).LT.HO.OR.H(I,J+1,K).LT.HO) THEN
C
C *** gewichtetes, geometrisches Mittel ... *****
C
      IF (H(I,J,K).GT.HU) THEN
        DHY = (H(I,J,K) - HU) ** (1. - FY(J)) *
&        MIN(ABS(H(I,J+1,K) - HU), DELTAZ(K)) ** FY(J)
        TRY = TKY * DHY / DELTAZ(K)
      ELSE
        TRY = TREST
      END IF
    ELSE
      TRY = TKY
    END IF
20 CALL HARMON (TKZ1,TKZ2,TRZ,FZ(K))
      VISX = 1.
      VISY = 1.
      VISZ = 1.
      IF (IVIS.EQ.1) CALL VISKOS (I,J,K,VISX,VISY,VISZ)
      WTI(I,J,K,1) = TRX/VISX
      WTI(I,J,K,2) = TRY/VISY
      WTI(I,J,K,3) = TRZ/VISZ
      RETURN
      END
C&KONINF
C *****
C   Version 13061987  Subroutine KONINF
C *****
C
C *** Subroutine KONINF Berechnung der Interface-Diffusion *****
C
      SUBROUTINE KONINF (I,J,K)
      PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
      PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
      PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
      PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
      DOUBLE PRECISION T,T0,STMP,CELS
      INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
      DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),

```

```

&          DIFZ (IZ) , XKOR (IX) , YKOR (IY) , ZKOR (IZ) , DIVX (IX, 2) ,
&          DIVY (IY, 2) , DIVZ (IZ, 2) , FX (IX) , FY (IY) , FZ (IZ) ,
&          DXYZ (IX, IY, IZ)
DIMENSION T (JX, JY, JZ) , TO (JX, JY, JZ) , STMP (JZ)
DIMENSION CPG (JX, JY, JZ) , WLKG (JX, JY, JZ) , WTI (IX, IY, IZ, KD2) ,
&          RHOG (JX, JY, JZ) , PHIG (JX, JY, JZ) , PHIN (IX, IY, IZ) ,
&          SATU (IZ) , VX (IX, IY, IZ) , VY (IX, IY, IZ) , VZ (IX, IY, IZ) , ANISO (3)
DIMENSION AP1 (IX, IY, IZ) , AP0 (IX, IY, IZ) , ANB (IX, IY, IZ, KD) ,
&          FNB (KD) , DNB (KD) , PECL (KD) , B (IX, IY, IZ)
DIMENSION WLK (4) , CC (4)
DIMENSION INDX (6) , INDY (6) , INDZ (6) , IADD (4) , JADD (4) , KADD (4)
DIMENSION IFLAG (IX, IY, IZ) , JFLAG (KX, KY, KZ) , IFLUS (JX, JY, JZ) , IFRO (JZ)
COMMON /GITT/DELTA X, DELTA Y, DELTA Z, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,
&          DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ
COMMON /INCB/INDX, INDY, INDZ, IADD, JADD, KADD
COMMON /IFLD/IFLAG, JFLAG, IFLUS, IFRO
COMMON /INF1/IZEIT, IPECL, IVOLDF, IAUSOP, ILOOP, IEIN, INPOP, IALTER
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
COMMON /INF3/LUMW, IENER, ISTOP, IDATOP, LISDAT, IDZTST
COMMON /KOE/ AP1, AP0, ANB, FNB, DNB, PECL, B
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /PAR2/BASIS, HOENN, ALFAHY, OMGH, FEHLH, ZZZZ, ZIGNO
COMMON /PAR3/CPW, CPE, CPL, WLW, WLE, WLL, RHOW, RHOE, RHOL, SENTHA
COMMON /PAR4/DTW, DTW2, DWAN, DTWL (3)
COMMON /PHYS/CPG, WLKG, ANISO, WTI, RHOG, PHIG, PHIN, SATU, VX, VY, VZ
COMMON /TEMP/T, TO, STMP
BRUCH = 0.
DO 20 L = 1, 4
  II = I + IADD (L)
  JJ = J + JADD (L)
  KK = K + KADD (L)
  IF (II.GT.NX.OR.JJ.GT.NY.OR.KK.GT.NZ) THEN
    WLK (L) = 0.
    GO TO 20
  END IF
  TDIF = T (II, JJ, KK) - STMP (KK)
  SAT = SATU (KK)
  IF (TDIF.GT.0.) THEN
    IF (IDIF.EQ.1) CPP = CPW * SAT + CPL * (1. - SAT)
    RK1 = WLW * SAT + WLL * (1. - SAT)
  ELSE
    CALL WASEIS (WE, TDIF, KK)
    IF (IDIF.EQ.1) THEN
      CALL WCPH20 (CPF, TDIF, KK)
      CPP = CPF * SAT + CPL * (1. - SAT)
    END IF
    RK1 = (WE * WLW + (1.- WE) * WLE) * SAT + WLL * (1. - SAT)
  END IF
C
C *** falls die WLF temperaturabhaengig ist ... *****
C
  IF (IWCP.GE.3) THEN
    RK = WLKG (II, JJ, KK)
    TT = CELS (T (II, JJ, KK))

```

```

C
C *** polynomisches Modell *****
C
      IF (IWCP.EQ.3)  RK2 = RK + TT * RK * (DTWL(1) + TT *
&                    (DTWL(2) + TT * DTWL(3)))
C
C *** exponentielles Modell *****
C
      IF (IWCP.EQ.4)  RK2 = RK * DTWL(1) * EXP(DTWL(2) * TT) +
&                    DTWL(3) * RK
C
C *** BUNTEBARTH & RUEFF (1988) *****
C
      IF (IWCP.EQ.5)  RK2 = RK / (1. + RK * DTWL(1) *
&                    (TT - 20.)) + 0.027 * DTWL(2) *
&                    (ZKOR(K) - 400.)

      ELSE
        RK2 = WLKG(II,JJ,KK)
      END IF
      PHI = PHIG(II,JJ,KK)
      IF (MODEL.EQ.1 .OR. MODEL.EQ.2) THEN
        IF (RK2.EQ.0.) RK2 = 1. / ZZZZ
        RK12 = RK1 / RK2
        RK12M1 = 1. - RK12
      END IF

C
C Berechnung der effektiven Waermeleitfaehigkeit bei von unter-
C schiedlichen Porenraumverteilungen nach WALSH & DECKER (1966)
C
C MODEL = 1  ==> Maximaleinfluss des Porenraumes
C MODEL = 2  ==> Minimaleinfluss des Porenraumes
C MODEL = 3  ==> arithmetisches Mittel
C
      IF (MODEL.EQ.2) THEN
        BRUCH = (3. * PHI * RK12M1) / (2. + PHI + RK12)
        RM1MB = RK2 * (1. - BRUCH)
      END IF

C
      IF (MODEL.EQ.1) THEN
        BRUCH = (PHI * (1. + 2. * RK12) * RK12M1) /
&              (PHI * RK12M1 + 3. * RK12)
        RM1MB = RK2 * (1. - BRUCH)
      END IF
      IF (RM1MB.LT.1E-20) RM1MB = 0.

C
      IF (MODEL.NE.1.AND.MODEL.NE.2)
&        RM1MB = PHI * RK1 + (1. - PHI) * RK2

C
      WLK(L) = RM1MB
      IF (IDIF.EQ.1) CC(L) = (1. - PHI) * CPG(II,JJ,KK) + PHI * CPP
20 CONTINUE
C
C *** Interface-Leitfaehigkeit - gewichtetes, harmonisches Mittel *****

```

```

C
IF (I.LT.NX) THEN
  CALL HARMON (WLK(1)*ANISO(1),WLK(2)*ANISO(1),WTI(I,J,K,1),FX(I))
  IF (IDIF.EQ.1) THEN
    CALL HARMON (CC(1),CC(2),CCI,FX(I))
    IF (CCI.EQ.0.) CCI = 1.
    WTI(I,J,K,1) = WTI(I,J,K,1) / CCI
  END IF
END IF
IF (J.LT.NY) THEN
  CALL HARMON (WLK(1)*ANISO(2),WLK(3)*ANISO(2),WTI(I,J,K,2),FY(J))
  IF (IDIF.EQ.1) THEN
    CALL HARMON (CC(1),CC(3),CCI,FY(J))
    IF (CCI.EQ.0.) CCI = 1.
    WTI(I,J,K,2) = WTI(I,J,K,2) / CCI
  END IF
END IF
IF (K.LT.NZ) THEN
  CALL HARMON (WLK(1)*ANISO(3),WLK(4)*ANISO(3),WTI(I,J,K,3),FZ(K))
  IF (IDIF.EQ.1) THEN
    CALL HARMON (CC(1),CC(4),CCI,FZ(K))
    IF (CCI.EQ.0.) CCI = 1.
    WTI(I,J,K,3) = WTI(I,J,K,3) / CCI
  END IF
END IF
RETURN
END

```

C&KONKOF

```

C *****
C   Version 20061987  Subroutine KONKOF
C *****
C
C *** Subroutine KONKOF Berechnung der Konduktionskoeffizienten *****
C

```

```

SUBROUTINE KONKOF (I,J,K)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DIMENSION DELTAX (IX),DELTAY (IY),DELTAZ (IZ),DIFX (IX),DIFY (IY),
&          DIFZ (IZ),XKOR (IX),YKOR (IY),ZKOR (IZ),DIVX (IX,2),
&          DIVY (IY,2),DIVZ (IZ,2),FX (IX),FY (IY),FZ (IZ),
&          DXYZ (IX,IY,IZ)
DIMENSION CPG (JX,JY,JZ),WLKG (JX,JY,JZ),WTI (IX,IY,IZ,KD2),
&          RHOG (JX,JY,JZ),PHIG (JX,JY,JZ),PHIN (IX,IY,IZ),
&          SATU (IZ),VX (IX,IY,IZ),VY (IX,IY,IZ),VZ (IX,IY,IZ),ANISO (3)
DIMENSION AP1 (IX,IY,IZ),APO (IX,IY,IZ),ANB (IX,IY,IZ,KD),
&          FNB (KD),DNB (KD),PECL (KD),B (IX,IY,IZ)
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
&          DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
COMMON /KOEf/AP1,APO,ANB,FNB,DNB,PECL,B
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
DO 10 L = 1, KD

```

```

        DNB(L) = 0.
10 CONTINUE
    IF (I.GT.1)
    &   DNB(1) = WTI(I-1,J,K,1) * DELTAY(J) * DELTAZ(K) / DIFX(I-1)
    IF (I.LT.NX)
    &   DNB(2) = WTI(I,J,K,1) * DELTAY(J) * DELTAZ(K) / DIFX(I)
    IF (J.GT.1)
    &   DNB(3) = WTI(I,J-1,K,2) * DELTAX(I) * DELTAZ(K) / DIFY(J-1)
    IF (J.LT.NY)
    &   DNB(4) = WTI(I,J,K,2) * DELTAX(I) * DELTAZ(K) / DIFY(J)
    IF (K.GT.1)
    &   DNB(5) = WTI(I,J,K-1,3) * DELTAX(I) * DELTAY(J) / DIFZ(K-1)
    IF (K.LT.NZ)
    &   DNB(6) = WTI(I,J,K,3) * DELTAX(I) * DELTAY(J) / DIFZ(K)
    RETURN
    END
C&KOPIER
C *****
C   Version 26061987 Subroutine KOPIER
C *****
C
C *** Subroutine KOPIER Kopieren der T oder H-Werte in ein anderes Feld
C
    SUBROUTINE KOPIER( IKOP )
    PARAMETER ( IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2 )
    PARAMETER ( IP1=1, IP2=1, IQ1=0, IQ2=0 )
    PARAMETER ( JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1 )
    PARAMETER ( KX=IX*IP2+IQ2, KY=IY*IP2+IQ2, KZ=IZ*IP2+IQ2 )
    DOUBLE PRECISION T, TO, STMP
    DIMENSION T( JX, JY, JZ ), TO( JX, JY, JZ ), STMP( JZ )
    DIMENSION H( KX, KY, KZ ), HO( KX, KY, KZ )
    COMMON /WPOT/H, HO
    COMMON /TEMP/T, TO, STMP
    COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
    COMMON /INF2/IDIF, IDRUG, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
C
    IF ( IECHO.GE.1 ) CALL ROUTIN ( 'KOPIER' )
C
    IF ( IKOP.EQ.1 ) THEN
        DO 10 K = 1, NZ
        DO 10 J = 1, NY
        DO 10 I = 1, NX
            TO( I, J, K ) = T( I, J, K )
10 CONTINUE
        ELSE
        DO 20 K = 1, NZ
        DO 20 J = 1, NY
        DO 20 I = 1, NX
            HO( I, J, K ) = H( I, J, K )
20 CONTINUE
        END IF
    RETURN
    END
C&KURZIN

```

```

C *****
C   Version 09081987   Subroutine KURZIN
C *****
C
C *** Subroutine KURZIN Einlesen der Daten - kurzer Input *****
C
SUBROUTINE KURZIN
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
PARAMETER (IWERT=400)
DOUBLE PRECISION T,TO,STMP,CELV,TT
CHARACTER FORMIN*20,FLAG*4,CARD*3
INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
INTEGER*2 IIFLAG,JJFLAG
INTEGER*2 IXX,IYY,IZZ,IUFL
DIMENSION IFLAG (IX,IY,IZ),JFLAG (KX,KY,KZ),IFLUS (JX,JY,JZ),IFRO (JZ)
DIMENSION T (JX,JY,JZ),TO (JX,JY,JZ),STMP (JZ)
DIMENSION H (KX,KY,KZ),HO (KX,KY,KZ)
DIMENSION CPG (JX,JY,JZ),WLKG (JX,JY,JZ),WTI (IX,IY,IZ,KD2),
&          RHOG (JX,JY,JZ),PHIG (JX,JY,JZ),PHIN (IX,IY,IZ),
&          SATU (IZ),VX (IX,IY,IZ),VY (IX,IY,IZ),VZ (IX,IY,IZ),ANISO (3)
DIMENSION TTX (KX,KY,KZ),TTY (KX,KY,KZ),TTZ (KX,KY,KZ),
&          SK (KX,KY,KZ),Q (KX,KY,KZ)
DIMENSION SC (JX,JY,JZ),SP (JX,JY,JZ),FRO (JZ,5)
DIMENSION WERT (IWERT),IXX (IWERT),IYY (IWERT),IZZ (IWERT),
&          FLAG (IWERT),IUFL (IWERT)
DIMENSION IXOBS (10),IYOBS (10),IZOBS (10)
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /TEMP/T,TO,STMP
COMMON /WPOT/H,HO
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
COMMON /TRNS/TTX,TTY,TTZ,Q,SK
COMMON /ENER/SC,SP,FRO
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
COMMON /ZYK1/WERT,IXX,IYY,IZZ,IUFL
COMMON /ZYK2/FLAG
PI = ATAN(1.) * 4.

C
C   IF (IECHO.GE.1) CALL ROUTIN ('KURZIN')
C
C *** optionale Eingabe der Randbedingungen und Anisotropiefaktoren ***
C
CARD = '19'
READ (5,100,END=200) FORMIN
READ (5,FORMIN,ERR=300,END=200) IBX,IBY,IBZ,ANISO(1),
&          ANISO(2),ANISO(3)
C
C *** Subroutine GITTER Berechnung der Gitterparameter *****

```

```

C
CALL GITTER
READ (5,100,ERR=300,END=200) FORMIN
DO 10 K = 1, NZ
C
C *** Karte 27a *** Temperaturberechnung mit vorgegebener Stroemung ****
C
IF (IGWF.EQ.0) THEN
CARD = '27a'
READ (5,FORMIN,ERR=300,END=200) TT,CCPG,RRHOG,WWLKG,
& PPHIG,PPHIN,SSATU,SSC,
& WINKEL,GWV,VVZ,IIFLAG
C
C *** Fließgeschwindigkeit in m/sec und Richtung *****
C
VVX = 0.
VVY = 0.
IF (SSATU.GE.100.) THEN
VVX = SIN((WINKEL) * (PI / 180.)) * GWV
VVY = COS((WINKEL + 180.) * (PI / 180.)) * GWV
VVZ = 0.
END IF
TT = CELV(TT)
RRHOG = RRHOG * 1000.
PPHIG = PPHIG / 100.
PPHIN = PPHIN / 100.
SATU(K) = SSATU / 100.
END IF
C
C *** Karte 27b *** Temperatur- und Stroemungsberechnung gekoppelt ****
C
IF (IGWF.EQ.1) THEN
CARD = '27b'
READ (5,FORMIN,ERR=300,END=200) TT,CCPG,RRHOG,WWLKG,
& PPHIG,PPHIN,SSC,QQ,SSK,HH,
& TTTX,TTTY,TTTZ,
& SSATU,IIFLAG,JJFLAG
TT = CELV(TT)
RRHOG = RRHOG * 1000.
PPHIG = PPHIG / 100.
PPHIN = PPHIN / 100.
SATU(K) = SSATU / 100.
END IF
C
C *** Karte 27c *** nur Stroemungsberechnung *****
C
IF (IGWF.EQ.2) THEN
CARD = '27c'
READ (5,FORMIN,ERR=300,END=200) HH,TTTX,TTTY,TTTZ,QQ,SSK,PPHIN,
& JJFLAG
PPHIN = PPHIN / 100.
END IF
C
C *** Vorbesetzung der Felder *****

```

```

C
DO 10 J = 1, NY
DO 10 I = 1, NX
  IF (IGWF.EQ.0) THEN
    VX(I,J,K) = VVX
    VY(I,J,K) = VVY
    VZ(I,J,K) = VVZ
    PHIN(I,J,K) = PPHIN
  END IF
  IF (IGWF.LE.1) THEN
    T(I,J,K) = TT
    CPG(I,J,K) = CCPG
    RHOG(I,J,K) = RRHOG
    WLKG(I,J,K) = WWLKG
    SC(I,J,K) = SSC
    PHIG(I,J,K) = PPHIG
    IFLAG(I,J,K) = IIFLAG
  END IF
  IF (IGWF.GE.1) THEN
    H(I,J,K) = HH
    TTX(I,J,K) = TTTX
    TTY(I,J,K) = TTTY
    TTZ(I,J,K) = TTTZ
    Q(I,J,K) = QQ
    SK(I,J,K) = SSK
    PHIN(I,J,K) = PPHIN
    JFLAG(I,J,K) = JJFLAG
  END IF
10 CONTINUE
  READ (5,100,END=200) FORMIN
  LUMW = 0
  CARD = '29'
  DO 20 L = 1, IWERT
    READ (5,FORMIN,ERR=300,END=400) FLAG(L),WERT(L),IXX(L),IYY(L),
& IZZ(L),IUFL(L)
    IF (FLAG(L).EQ.'ENDE') GO TO 120
    LUMW = LUMW + 1
  20 CONTINUE
  WRITE (1,110)
  STOP
C *** Besetzung des IFLAG - Feldes, falls IBX,IBY,IBZ = 1 *****
120 CLOSE(5)
  IF (IBZ.GT.0) THEN
    DO 50 K = 1, NZ, NZ1
    DO 50 J = 1, NY
    DO 50 I = 1, NX
      IF ((IBZ.EQ.1.OR.IBZ.EQ.3).AND.IGWF.LT.2) IFLAG(I,J,K) = 2
      IF ((IBZ.EQ.2.OR.IBZ.EQ.3).AND.IGWF.GE.1) JFLAG(I,J,K) = 2
  50 CONTINUE
  END IF
  IF (IBY.GT.0) THEN
    DO 60 J = 1, NY, NY1
    DO 60 K = 1, NZ
    DO 60 I = 1, NX

```

```

        IF ((IBY.EQ.1.OR.IBY.EQ.3).AND.IGWF.LT.2) IFLAG(I,J,K) = 2
        IF ((IBY.EQ.2.OR.IBY.EQ.3).AND.IGWF.GE.1) JFLAG(I,J,K) = 2
60    CONTINUE
    END IF
    IF (IBX.GT.0) THEN
        DO 70 I = 1, NX, NX1
        DO 70 K = 1, NZ
        DO 70 J = 1, NY
            IF ((IBX.EQ.1.OR.IBX.EQ.3).AND.IGWF.LT.2) IFLAG(I,J,K) = 2
            IF ((IBX.EQ.2.OR.IBX.EQ.3).AND.IGWF.GE.1) JFLAG(I,J,K) = 2
70    CONTINUE
    END IF
    RETURN
200 WRITE (1,210) CARD
    STOP
300 WRITE (1,310) CARD
    STOP
400 WRITE (1,410) CARD
    STOP
100 FORMAT(A20)
110 FORMAT(' Felddimension IWERT muss groesser gewaehlt werden')
210 FORMAT(' unerwartet Ende der Eingabedatei erreicht in Karte ',A3)
310 FORMAT(' fehlerhafte Datenstruktur in Karte ',A3)
410 FORMAT(' fehlende ENDE - Anweisung in Karte ',A3)
    END
C&OBSDAT
C *****
C    Version 24011988  Subroutine OBSDAT
C *****
C
C *** Subroutine OBSDAT Knotentemperaturen auf File OBSFIL schreiben ***
C
    SUBROUTINE OBSDAT
    PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
    PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
    PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
    PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
    DOUBLE PRECISION T,T0,STMP,CELS
    DIMENSION T(JX,JY,JZ),T0(JX,JY,JZ),STMP(JZ)
    DIMENSION H(KX,KY,KZ),H0(KX,KY,KZ)
    DIMENSION IXOBS(10),IYOBS(10),IZOBS(10)
    COMMON /TEMP/T,T0,STMP
    COMMON /WPOT/H,H0
    COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
    COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
    COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
C
    IF (IECHO.GE.1) CALL ROUTIN ('OBSDAT')
C
    ZEIT = SUMSEK / DZMF
    IF (IGWF.EQ.0)
&    WRITE (7,100) ZEIT,(CELS(T(IXOBS(I),IYOBS(I),IZOBS(I))),
&    I = 1, IOBS)
    IF (IGWF.EQ.1)

```

```

& WRITE (7,200) ZEIT,(CELS(T(IXOBS(I),IYOBS(I),IZOBS(I))),
& H(IXOBS(I),IYOBS(I),IZOBS(I)),I = 1, IOBS)
IF (IGWF.EQ.2)
& WRITE (7,200) ZEIT,(H(IXOBS(I),IYOBS(I),IZOBS(I)),I = 1, IOBS)
100 FORMAT(F12.3,10F8.3)
200 FORMAT(F12.3,10F8.3,/,9X,10F8.3)
RETURN
END

```

C&OMEGAT

```

C *****
C Version 05111987 Subroutine OMEGAT
C *****
C
C *** Subroutine OMEGAT Testen des OMEGA - Wertes bei SOR/SUR *****
C

```

```

SUBROUTINE OMEGAT(FEHL,OMEGA,ALFA)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
CHARACTER ANTW*1, TEXT*10
DOUBLE PRECISION T,TO,STMP,FEHL
DIMENSION T(JX,JY,JZ),TO(JX,JY,JZ),STMP(JZ)
DIMENSION H(KX,KY,KZ),HO(KX,KY,KZ)
DIMENSION TEXT(2)
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /PAR1/DZEIT, DZMF, DZF, FLURAB, ALFATP, FEHLT, FEHLER, OMGT, SUMSEK
COMMON /PAR2/BASIS, HOENN, ALFAHY, OMGH, FEHLH, ZZZZ, ZIGNO
COMMON /TEMP/T, TO, STMP
COMMON /WPOT/H, HO
TEXT(1) = 'Temperatur'
TEXT(2) = 'Stroemungs'

```

C

```
IF (IECHO.GE.1) CALL ROUTIN ('OMEGAT')
```

C

```

WRITE (1,100) OMEGA,NIT,FEHL,ALFA
30 WRITE (1,150)
READ (*,160) ANTW
IF (ANTW.EQ.'n'.OR.ANTW.EQ.'N') GO TO 20
IF (ANTW.NE.'j'.AND.ANTW.NE.'J') GO TO 30
40 WRITE (1,110)
READ (*,*,ERR=40) OMEGA
50 WRITE (1,130)
READ (*,*,ERR=50) ALFA
IF (ISOR.EQ.1) THEN
DO 10 K = 1, NZ
DO 10 J = 1, NY
DO 10 I = 1, NX
T(I,J,K) = TO(I,J,K)
10 CONTINUE
OMGT = OMEGA
ALFATP = ALFA
END IF

```

```

      IF (ISOR.EQ.2) THEN
        DO 15 K = 1, NZ
        DO 15 J = 1, NY
        DO 15 I = 1, NX
          H(I,J,K) = H0(I,J,K)
15      CONTINUE
        OMGH = OMEGA
        ALFAHY = ALFA
      END IF
      RETURN
20     WRITE (IAUS,170) TEXT(ISOR)
        WRITE (IAUS,120) OMEGA
        WRITE (IAUS,140) ALFA
        ISOR = ISOR - 1
      RETURN
100    FORMAT(' Bei OMEGA = ',F5.3,' ergab sich nach ',I2,
      &      ' Iterationen ein Fehler von ',F8.6,/,
      &      ' Wichtungsfaktor ALFA = ',F5.2)
110    FORMAT(T16,' Bitte neuen Wert fuer OMEGA eingeben ==> ',\ )
120    FORMAT(' OMEGA - Wert wurde auf ',F4.2,' geaendert')
130    FORMAT(T16,' Bitte neuen Wert fuer ALFA eingeben ==> ',\ )
140    FORMAT(' Wichtungsfaktor ALFA wurde auf ',F4.2,' geaendert')
150    FORMAT(' Werden neue Werte fuer OMEGA und ALFA gewuenscht? (J/N)
      & ==> ',\ )
160    FORMAT(A1)
170    FORMAT(/,' Aenderung in der ',A10,'berechnung:',/,1H ,38(1H-),/)
      END
C&PECLET
C *****
C   Version 29061987  Subroutine PECLET
C *****
C
C *** Subroutine PECLET Berechnung der Peclet - Wichtungsfunktion *****
C
      SUBROUTINE PECLET(LMAX)
      PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
      PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
      PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
      PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
      DIMENSION AP1 (IX,IY,IZ),APO (IX,IY,IZ),ANB (IX,IY,IZ,KD),
      &      FNB(KD),DNB(KD),PECL(KD),B (IX,IY,IZ)
      COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
      COMMON /KOEf/AP1,APO,ANB,FNB,DNB,PECL,B
C
      GO TO (100, 200, 300, 400, 500) IPECL
C
C *** Zentrale Differenzen *****
C
100    CONTINUE
      DO 10 L = 1, LMAX
        IF (DNB(L).EQ.0.) THEN
          PECL(L) = 1.
        ELSE
          PECL(L) = 1. - 0.5 * ABS(FNB(L) / DNB(L))

```

```

        END IF
10 CONTINUE
RETURN
C
C *** Upwind - Schema      COURANT, ISAACSON & REES (1952) *****
C
200 CONTINUE
    DO 20 L = 1, LMAX
        PECL(L) = 1.
    20 CONTINUE
RETURN
C
C *** Hybrid - Schema      SPALDING (1972) *****
C
300 CONTINUE
    DO 30 L = 1, LMAX
        IF (DNB(L).EQ.0.) THEN
            PECL(L) = 1.
        ELSE
            PECL(L) = AMAX1(0., (1. - 0.5 * ABS(FNB(L) / DNB(L))))
        END IF
    30 CONTINUE
RETURN
C
C *** Power law - Schema   PANTAKAR (1979) *****
C
400 CONTINUE
    DO 40 L = 1, LMAX
        IF (DNB(L).EQ.0.) THEN
            PECL(L) = 1.
        ELSE
            PECL(L) = AMAX1(0., (1. - 0.1 * ABS(FNB(L) / DNB(L))) ** 5)
        END IF
    40 CONTINUE
RETURN
C
C *** Exponential - Schema *****
C
500 CONTINUE
    DO 50 L = 1, LMAX
        IF (DNB(L).EQ.0..OR.FNB(L).EQ.0.) THEN
            PECL(L) = 1.
        ELSE
            P = FNB(L) / DNB(L)
            PECL(L) = ABS(P) / (EXP(ABS(P)) - 1.)
        END IF
    50 CONTINUE
RETURN
END
C&PIEDRU
C *****
C      Version 24011989 Subroutine PIEDRU
C *****
C

```

```

C *** Subroutine PIEDRU Drucken der berechneten Piezometerhoehen *****
C
SUBROUTINE PIEDRU
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DIMENSION H(KX,KY,KZ),H0(KX,KY,KZ)
DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
& DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
& DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
& DXYZ(IX,IY,IZ)
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
& DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /IPRO/IPROX,IPROY,IPROZ
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /WPOT/H,H0

C
IF (IECHO.GE.1) CALL ROUTIN ('PIEDRU')

C
CALL ZEITBE (IAUS)
WRITE (IAUS,120) NIT
IF (IPROY.EQ.0.OR.IPROY.GT.NY) GO TO 45
C *** X - Z Schnitt *****
WRITE (IAUS,20) YKOR(IPROY),IPROY
WRITE (IAUS,30) (XKOR(I),I = 1, NX)
DO 40 K = 1, NZ
WRITE (IAUS,50) ZKOR(K), (H(I,IPROY,K), I = 1, NX)
40 CONTINUE
45 IF (IPROX.EQ.0.OR.IPROX.GT.NX) GO TO 85
C *** Y - Z Schnitt *****
WRITE (IAUS,60) XKOR(IPROX),IPROX
WRITE (IAUS,70) (YKOR(J),J = 1, NY)
DO 80 K = 1, NZ
WRITE (IAUS,50) ZKOR(K), (H(IPROX,J,K), J = 1, NY)
80 CONTINUE
85 IF (IPROZ.EQ.0.OR.IPROZ.GT.NZ) GO TO 115
C *** X - Y Schnitt *****
WRITE (IAUS,90) ZKOR(IPROZ),IPROZ
WRITE (IAUS,100) (XKOR(I),I = 1, NX)
DO 110 J = 1, NY
WRITE (IAUS,50) YKOR(J), (H(I,J,IPROZ), I = 1, NX)
110 CONTINUE
20 FORMAT(/,T20,'X - Z Piezometerhoehen Y bei ',F5.2,
& ' Meter - Spalte ',I2,/)
30 FORMAT(' Z\X ',10F6.2,/,9(10X,10F6.2,/)
50 FORMAT(F10.2,10F6.2,/,9(10X,10F6.2,/)
60 FORMAT(/,T20,'Y - Z Piezometerhoehen X bei ',F5.2,
& ' Meter - Reihe ',I2,/)
70 FORMAT(' Z\Y ',10F6.2,/,9(10X,10F6.2,/)
90 FORMAT(/,T20,'X - Y Piezometerhoehen Z bei ',F5.2,

```

```

      &      ' Meter - Schicht ',I2,/)
100 FORMAT('  Y\X      ',10F6.2,/,9(10X,10F6.2,/)
120 FORMAT(/,' Konvergenz seit dem letzten Zeitschritt nach ',I2,
      &      ' Iterationen')
115 RETURN
      END
C&PHATST
C *****
C      Version 11111987  Subroutine PHATST
C *****
C
C *** Subroutine PHATST ueberprueft den Phasenwechsel *****
C
SUBROUTINE PHATST (I,J,K,TALT,TNEU)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,T0,STMP,TALT,TNEU,STP,TDIF,cels
INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
DIMENSION IFLAG (IX,IY,IZ),JFLAG (KX,KY,KZ),IFLUS (JX,JY,JZ),IFRO (JZ)
DIMENSION T (JX,JY,JZ),T0 (JX,JY,JZ),STMP (JZ)
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
COMMON /PAR3/CPW,CPE,CPL,WLW,WLE,WLL,RHOW,RHOE,RHOL,SENTHA
COMMON /PAR4/DTW,DTW2,DWAN,DTWL(3)
COMMON /TEMP/T,T0,STMP

C
      IF (IECHO.GE.2) CALL ROUTIN ('PHATST')
C
      IF (IFLAG(I,J,K).NE.1) RETURN
      IF (TALT.GE.STMP(K).AND.TNEU.GE.STMP(K)) RETURN
      TDIF = TNEU - TALT
      IF (TDIF) 10, 20, 30
C
C *** es wird kaelter ... *****
C
10 IF (TALT.GE.STMP(K)) THEN
      DT1 = - DTW2
      CALL WASEIS (WE1,DT1,K)
      DWE = 1. - WE1
      IF (DWE.GT.DWAN) THEN
          DDT = DTW * DWAN / DWE
      ELSE
          DDT = DTW
      END IF
      T(I,J,K) = STMP(K) - DDT
  ELSE
      DT1 = TALT - STMP(K) + DTW2
      IF (DT1.GT.0) DT1 = 0.
      CALL WASEIS (WE1,DT1,K)

```

```

DT2 = TALT - STMP(K) - DTW2
CALL WASEIS (WE2,DT2,K)
DWE = WE1 - WE2
IF (DWE.GT.DWAN) THEN
  DDT = (DT1 - DT2) * DWAN / DWE
ELSE
  DDT = DT1 - DT2
END IF
T(I,J,K) = TALT - DDT
END IF
GO TO 100

```

```

C
C *** es hat sich nichts geaendert ... *****
C

```

```

20 RETURN

```

```

C
C *** es wird waermer ... *****
C

```

```

30 DT1 = TALT - STMP(K) - DTW2
CALL WASEIS (WE1,DT1,K)
DT2 = DT1 + DTW
IF (DT2.GT.0.) DT2 = 0.
CALL WASEIS (WE2,DT2,K)
DWE = WE2 - WE1
IF (DWE.GT.DWAN) THEN
  DDT = (DT2 - DT1) * DWAN / DWE
ELSE
  DDT = DT2 - DT1
END IF
T(I,J,K) = TALT + DDT
100 I1 = MAX0 ( 1, I-1)
J1 = MAX0 ( 1, J-1)
K1 = MAX0 ( 1, K-1)
I2 = MIN0 (NX, I+1)
J2 = MIN0 (NY, J+1)
K2 = MIN0 (NZ, K+1)
CALL KOEFFT (I1,J1,K1,I2,J2,K2)
RETURN
END

```

```

C&PROFIL

```

```

C *****
C Version 24061987 Subroutine PROFIL
C *****
C *** Subroutine PROFIL Drucken der Eingangsdaten in Profilschnitten ***
C

```

```

SUBROUTINE PROFIL(PROF,IPROF,NPROF)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
INTEGER*2 IPROF
CHARACTER FORMOU*30,FORTXT*48
DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),

```

```

&          DIFZ (IZ) , XKOR (IX) , YKOR (IY) , ZKOR (IZ) , DIVX (IX , 2) ,
&          DIVY (IY , 2) , DIVZ (IZ , 2) , FX (IX) , FY (IY) , FZ (IZ) ,
&          DXYZ (IX , IY , IZ)
DIMENSION PROF (IX , IY , IZ) , IPROF (IX , IY , IZ)
DIMENSION FORMOU (8) , FORTXT (8)
COMMON /GITT/DELTA X , DELTA Y , DELTA Z , DIFX , DIFY , DIFZ , XKOR , YKOR , ZKOR ,
&          DIVX , DIVY , DIVZ , FX , FY , FZ , DXYZ
COMMON /NDIM/NX , NY , NZ , NX1 , NY1 , NZ1
COMMON /INF1/IZEIT , IPECL , IVOLDF , IAUSOP , ILOOP , IEIN , INPOP , IALTER
COMMON /INF2/IDIF , IDRU , IFROST , IAUS , IECHO , MODEL , NIT , IWCP , ISOR , IEN1
COMMON /INF3/LUMW , IENER , ISTOP , IDATOP , LISDAT , IDZTST
COMMON /IPRO/IPROX , IPROY , IPROZ
COMMON /FORM/FORMOU , FORTXT

C
IF (IECHO.GE.1) CALL ROUTIN ('PROFIL')

C
WRITE (IAUS , FORTXT (NPROF))
IF (IPROY.LE.0.OR.IPROY.GT.NY) GO TO 35
C *** X - Z Schnitt *****
WRITE (IAUS , 10) YKOR (IPROY) , IPROY
WRITE (IAUS , 20) (XKOR (I) , I = 1 , NX)
DO 30 K = 1 , NZ
IF (NPROF.EQ.3)
& WRITE (IAUS , FORMOU (NPROF)) ZKOR (K) , (GRCBCM (PROF (I , IPROY , K)) ,
& I = 1 , NX)
IF (NPROF.LE.6.AND.NPROF.NE.3)
& WRITE (IAUS , FORMOU (NPROF)) ZKOR (K) , (PROF (I , IPROY , K) , I = 1 , NX)
IF (NPROF.GE.7)
& WRITE (IAUS , FORMOU (NPROF)) ZKOR (K) , (IPROF (I , IPROY , K) , I = 1 , NX)
30 CONTINUE
35 IF (IPROX.LE.0.OR.IPROX.GT.NX) GO TO 65
C *** Y - Z Schnitt *****
WRITE (IAUS , 40) XKOR (IPROX) , IPROX
WRITE (IAUS , 50) (YKOR (J) , J = 1 , NY)
DO 60 K = 1 , NZ
IF (NPROF.EQ.3)
& WRITE (IAUS , FORMOU (NPROF)) ZKOR (K) , (GRCBCM (PROF (IPROX , J , K)) ,
& J = 1 , NY)
IF (NPROF.LE.6.AND.NPROF.NE.3)
& WRITE (IAUS , FORMOU (NPROF)) ZKOR (K) , (PROF (IPROX , J , K) , J = 1 , NY)
IF (NPROF.GE.7)
& WRITE (IAUS , FORMOU (NPROF)) ZKOR (K) , (IPROF (IPROX , J , K) , J = 1 , NY)
60 CONTINUE
65 IF (IPROZ.LE.0.OR.IPROZ.GT.NZ) GO TO 95
C *** X - Y Schnitt *****
WRITE (IAUS , 70) ZKOR (IPROZ) , IPROZ
WRITE (IAUS , 80) (XKOR (I) , I = 1 , NX)
DO 90 J = 1 , NY
IF (NPROF.EQ.3)
& WRITE (IAUS , FORMOU (NPROF)) YKOR (J) , (GRCBCM (PROF (I , J , IPROZ)) ,
& I = 1 , NX)
IF (NPROF.LE.6.AND.NPROF.NE.3)
& WRITE (IAUS , FORMOU (NPROF)) YKOR (J) , (PROF (I , J , IPROZ) , I = 1 , NX)
IF (NPROF.GE.7)

```

```

& WRITE (IAUS,FORMOU(NPROF)) YKOR(J), (IPROF(I,J,IPROZ), I = 1, NX)
90 CONTINUE
10 FORMAT(/, ' X - Z - Schnitt Y bei ', F5.2, ' Meter - Spalte ', I2)
20 FORMAT(' Z\X ', 10F6.2, /, 9(7X, 10F6.2, /))
40 FORMAT(/, ' Y - Z - Schnitt X bei ', F5.2, ' Meter - Reihe ', I2)
50 FORMAT(' Z\Y ', 10F6.2, /, 9(7X, 10F6.2, /))
70 FORMAT(/, ' X - Y - Schnitt Z bei ', F5.2, ' Meter - Schicht ', I2)
80 FORMAT(' Y\X ', 10F6.2, /, 9(7X, 10F6.2, /))
95 NPROF = NPROF + 1
RETURN
END

```

C&QKOEFSH

```

C *****
C Version 13051988 Subroutine QKOEFSH

```

```

C *****
C *** Subroutine QKOEFSH Berechnung des Quelltermkoeffizienten Piez. ***
C

```

SUBROUTINE QKOEFSH

PARAMETER (IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2)

PARAMETER (IP1=1, IP2=1, IQ1=0, IQ2=0)

PARAMETER (JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1)

PARAMETER (KX=IX*IP2+IQ2, KY=IY*IP2+IQ2, KZ=IZ*IP2+IQ2)

INTEGER*2 IFLAG, JFLAG, IFLUS, IFRO

DIMENSION DELTAX (IX), DELTAY (IY), DELTAZ (IZ), DIFX (IX), DIFY (IY),

& DIFZ (IZ), XKOR (IX), YKOR (IY), ZKOR (IZ), DIVX (IX, 2),

& DIVY (IY, 2), DIVZ (IZ, 2), FX (IX), FY (IY), FZ (IZ),

& DXYZ (IX, IY, IZ)

DIMENSION H (KX, KY, KZ), HO (KX, KY, KZ)

DIMENSION TTX (KX, KY, KZ), TTY (KX, KY, KZ), TTZ (KX, KY, KZ),

& SK (KX, KY, KZ), Q (KX, KY, KZ)

DIMENSION CPG (JX, JY, JZ), WLKG (JX, JY, JZ), WTI (IX, IY, IZ, KD2),

& RHOG (JX, JY, JZ), PHIG (JX, JY, JZ), PHIN (IX, IY, IZ),

& SATU (IZ), VX (IX, IY, IZ), VY (IX, IY, IZ), VZ (IX, IY, IZ), ANISO (3)

DIMENSION AP1 (IX, IY, IZ), AP0 (IX, IY, IZ), ANB (IX, IY, IZ, KD),

& FNB (KD), DNB (KD), PECL (KD), B (IX, IY, IZ)

DIMENSION IFLAG (IX, IY, IZ), JFLAG (KX, KY, KZ), IFLUS (JX, JY, JZ), IFRO (JZ)

COMMON /GITT/DELTAX, DELTAY, DELTAZ, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,

& DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ

COMMON /IFLD/IFLAG, JFLAG, IFLUS, IFRO

COMMON /INF2/IDIF, IDRUG, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1

COMMON /KOEFSH/AP1, AP0, ANB, FNB, DNB, PECL, B

COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1

COMMON /PAR1/DZEIT, DZMF, DZF, FLURAB, ALFATP, FEHLT, FEHLER, OMGT, SUMSEK

COMMON /PAR2/BASIS, HOENN, ALFAHY, OMGH, FEHLH, ZZZZ, ZIGNO

COMMON /PHYS/CPG, WLKG, ANISO, WTI, RHOG, PHIG, PHIN, SATU, VX, VY, VZ

COMMON /TRNS/TTX, TTY, TTZ, Q, SK

COMMON /WPOT/H, HO

C

IF (IECHO.GE.1) CALL ROUTIN ('QKOEFSH')

C

C *** Addieren aller Nachbarkoeffizienten *****

C

DO 10 K = 1, NZ

```

HO = HOENN - ZKOR(K) - DIVZ(K,2) + DELTAZ(K)
DO 10 J = 1, NY
DO 10 I = 1, NX
SPK = SK(I,J,K)
IF (HO.GT.H(I,J,K)) SPK = PHIN(I,J,K)
IF (JFLAG(I,J,K).EQ.2) SPK = ZZZZ
APO(I,J,K) = DELTAX(I) * DELTAY(J) * SPK / DZEIT
AP = 0.
DO 20 L = 1, KD
AP = AP + ANB(I,J,K,L)
20 CONTINUE
AP1(I,J,K) = AP + APO(I,J,K)
B(I,J,K) = Q(I,J,K) + APO(I,J,K) * HO(I,J,K)
10 CONTINUE
RETURN
END

```

C&QKOEFT

```

C *****
C   Version 30081987   Subroutine QKOEFT
C *****
C
C *** Subroutine QKOEFT Berechnung des Quelltermkoeffizienten Temp. ***
C

```

```

SUBROUTINE QKOEFT
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,TO,STMP,CELS
INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
&          DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
&          DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
&          DXYZ(IX,IY,IZ)
DIMENSION IFLAG(IX,IY,IZ),JFLAG(KX,KY,KZ),IFLUS(JX,JY,JZ),IFRO(JZ)
DIMENSION T(JX,JY,JZ),TO(JX,JY,JZ),STMP(JZ)
DIMENSION CPG(JX,JY,JZ),WLKG(JX,JY,JZ),WTI(IX,IY,IZ,KD2),
&          RHOG(JX,JY,JZ),PHIG(JX,JY,JZ),PHIN(IX,IY,IZ),
&          SATU(IZ),VX(IX,IY,IZ),VY(IX,IY,IZ),VZ(IX,IY,IZ),ANISO(3)
DIMENSION SC(JX,JY,JZ),SP(JX,JY,JZ),FRO(JZ,5)
DIMENSION AP1(IX,IY,IZ),APO(IX,IY,IZ),ANB(IX,IY,IZ,KD),
&          FNB(KD),DNB(KD),PECL(KD),B(IX,IY,IZ)
COMMON /ENER/SC,SP,FRO
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
&          DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /KOEFT/AP1,APO,ANB,FNB,DNB,PECL,B
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
COMMON /PAR2/BASIS,HOENN,ALFAHY,OMGH,FEHLH,ZZZZ,ZIGNO
COMMON /PAR3/CPW,CPE,CPL,WLW,WLE,WLL,RHOW,RHOE,RHOL,SENTHA

```

```

COMMON /TEMP/T,TO,STMP
C
C IF (IECHO.GE.1) CALL ROUTIN ('QKOEFT')
C
C DO 10 K = 1, NZ
C DO 10 J = 1, NY
C DO 10 I = 1, NX
C
C *** konstante Temperatur - Knoten *****
C IF (IFLAG(I,J,K).EQ.2.OR.IFLAG(I,J,K).EQ.5) THEN
C B(I,J,K) = APO(I,J,K) * TO(I,J,K)
C ELSE
C
C *** nur Diffusion *****
C IF (IDIF.EQ.0) THEN
C B(I,J,K) = SC(I,J,K) * DXYZ(I,J,K) + APO(I,J,K) * TO(I,J,K)
C ELSE
C
C *** Diffusion und Konvektion *****
C TDIF = T(I,J,K) - STMP(K)
C IF (TDIF.GT.0.) THEN
C CPP = CPW * SATU(K) + (1. - SATU(K)) * CPL
C ELSE
C CALL WCPH2O (CPF,TDIF,K)
C CPP = CPF * SATU(K) + (1. - SATU(K)) * CPL
C END IF
C CC = PHIG(I,J,K) * CPP + (1. - PHIG(I,J,K)) * CPG(I,J,K)
C IF (CC.EQ.0.) CC = 1.
C B(I,J,K) = SC(I,J,K) / CC * DXYZ(I,J,K) + APO(I,J,K) *
& TO(I,J,K)
C END IF
C END IF
C 10 CONTINUE
C RETURN
C END
C&RICHTG
C *****
C Version 20101987 Subroutine RICHTG
C *****
C *** Subroutine RICHTG Bestimmung der Iterationsreihenfolge *****
C
C SUBROUTINE RICHTG (L,LANF,LEND,LPLUS,LW,L10)
C L01 = IABS(L10 - 1)
C LL = MOD(L+1,2) * (LW - 1)
C LANF = (LL + 1) * L10 + L01
C LEND = (LW - LL) * L10 + LW * L01
C LPLUS = (MOD(L,2) - MOD(L+1,2)) * L10 + L01
C RETURN
C END
C&ROUTIN

```

```

C *****
C   Version 02081988   Subroutine ROUTIN
C *****
C
C *** Subroutine ROUTIN Ausgabe der aktiven Subroutine *****
C
C   SUBROUTINE ROUTIN (SUBROU)
C   CHARACTER SUBROU*6
C   WRITE (1,100) SUBROU
C   RETURN
C 100 FORMAT(T17,'***** Subroutine ',A6,' *****')
C   END
C&TAUPKT
C *****
C   Version 24061987   Subroutine TAUPKT
C *****
C
C *** Subroutine TAUPKT Gefrierpunktberechnung von Wasser *****
C
C   SUBROUTINE TAUPKT(STMP,ZKOR,I,J)
C   PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
C   PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
C   PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
C   PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
C   DOUBLE PRECISION STMP
C   DIMENSION ZKOR(IZ),STMP(IZ)
C   DIMENSION H(KX,KY,KZ),H0(KX,KY,KZ)
C   DIMENSION IXOBS(10),IYOBS(10),IZOBS(10)
C   COMMON /WPOT/H,H0
C   COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
C   COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
C   COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
C   COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
C   COMMON /PAR2/BASIS,HOENN,ALFAHY,OMGH,FEHLH,ZZZZ,ZIGNO
C
C   IF (IECHO.GE.1) CALL ROUTIN ('TAUPKT')
C
C *** exakte Loesung noch nicht vorhanden *****
C
C   FLUR = FLURAB
C   DO 10 K = 1, NZ
C   IF (IGWF.EQ.1) FLUR = HOENN - H(I,J,K)
C   IF (ZKOR(K).LE.FLUR) STMP(K) = 273.15
C   IF (ZKOR(K).GT.FLUR) STMP(K) = 272.15
C 10 CONTINUE
C   RETURN
C   END
C&THOMAS
C *****
C   Version 21071988   Subroutine THOMAS
C *****
C
C *** Subroutine THOMAS Tridiagonaler Matrizzenalgorithmus nach THOMAS *
C

```

```

SUBROUTINE THOMAS (N)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
DOUBLE PRECISION UU,AA,BB,CC,DD,BETA,GAMMA,S,U
DIMENSION UU(IMAX),AA(IMAX),BB(IMAX),CC(IMAX),DD(IMAX)
DIMENSION GAMMA(0:IMAX),S(0:IMAX)
COMMON /IADI/UU,AA,BB,CC,DD
DO 10 I = 1, N
  BETA = BB(I) - AA(I) * GAMMA(I-1)
  IF (BETA.EQ.0.) THEN
    GAMMA(I) = 0.
    S(I) = UU(I)
  ELSE
    GAMMA(I) = CC(I) / BETA
    S(I) = (DD(I) - AA(I) * S(I-1)) / BETA
  END IF
10 CONTINUE
C
C *** Wiedereinsetzen *****
C
  UU(N) = S(N)
  DO 20 I = N-1, 1, -1
    UU(I) = S(I) - GAMMA(I) * UU(I+1)
  20 CONTINUE
  RETURN
  END
C&TMPDRU
C *****
C   Version 14061987 Subroutine TMPDRU
C *****
C
C *** Subroutine TMPDRU Drucken der berechneten Temperaturprofile *****
C
SUBROUTINE TMPDRU
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,T0,STMP,CELS
DIMENSION DELTAX (IX),DELTAY (IY),DELTAZ (IZ),DIFX (IX),DIFY (IY),
&          DIFZ (IZ),XKOR (IX),YKOR (IY),ZKOR (IZ),DIVX (IX,2),
&          DIVY (IY,2),DIVZ (IZ,2),FX (IX),FY (IY),FZ (IZ),
&          DXYZ (IX,IY,IZ)
DIMENSION T (JX,JY,JZ),T0 (JX,JY,JZ),STMP (JZ)
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
&          DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /TEMP/T,T0,STMP
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /IPRO/IPROX,IPROY,IPROZ
COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
C
  IF (IECHO.GE.1) CALL ROUTIN ('TMPDRU')

```

```

C      CALL ZEITBE (IAUS)
      WRITE (IAUS,120) NIT
      IF (IPROY.EQ.0.OR.IPROY.GT.NY) GO TO 45
C *** X - Z Schnitt *****
      WRITE (IAUS,20) YKOR(IPROY),IPROY
      WRITE (IAUS,30) (XKOR(I),I = 1, NX)
      DO 40 K = 1, NZ
      WRITE (IAUS,50) ZKOR(K), (CELS(T(I,IPROY,K))),I = 1, NX)
      40 CONTINUE
      45 IF (IPROX.EQ.0.OR.IPROX.GT.NX) GO TO 85
C *** Y - Z Schnitt *****
      WRITE (IAUS,60) XKOR(IPROX),IPROX
      WRITE (IAUS,70) (YKOR(J),J = 1, NY)
      DO 80 K = 1, NZ
      WRITE (IAUS,50) ZKOR(K), (CELS(T(IPROX,J,K))),J = 1, NY)
      80 CONTINUE
      85 IF (IPROZ.EQ.0.OR.IPROZ.GT.NZ) GO TO 115
C *** X - Y Schnitt *****
      WRITE (IAUS,90) ZKOR(IPROZ),IPROZ
      WRITE (IAUS,100) (XKOR(I),I = 1, NX)
      DO 110 J = 1, NY
      WRITE (IAUS,50) YKOR(J), (CELS(T(I,J,IPROZ))),I = 1, NX)
      110 CONTINUE
      20 FORMAT(/,T20,'X - Z Temperaturverteilung Y bei ',F5.2,
      &      ' Meter - Spalte ',I2,/)
      30 FORMAT('  Z\X      ',10F6.2,/,9(10X,10F6.2,/)
      50 FORMAT(F10.2,10F6.2,/,9(10X,10F6.2,/)
      60 FORMAT(/,T20,'Y - Z Temperaturverteilung X bei ',F5.2,
      &      ' Meter - Reihe ',I2,/)
      70 FORMAT('  Z\Y      ',10F6.2,/,9(10X,10F6.2,/)
      90 FORMAT(/,T20,'X - Y Temperaturverteilung Z bei ',F5.2,
      &      ' Meter - Schicht ',I2,/)
      100 FORMAT('  Y\X      ',10F6.2,/,9(10X,10F6.2,/)
      120 FORMAT(/,' Konvergenz seit dem letzten Zeitschritt nach ',I2,
      &      ' Iterationen')
      115 RETURN
      END
C&UMWAND
C *****
C      Version 24091987  Subroutine UMWAND
C *****
C
C *** Subroutine UMWAND Umwandeln der Zusatzinformationen *****
C
      SUBROUTINE UMWAND (L,IAN)
      PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
      PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
      PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
      PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
      PARAMETER (IWERT=400)
      DOUBLE PRECISION T,T0,STMP,CELV
      CHARACTER FLAG*4,FLG*4
      INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO

```

```

INTEGER*2 IXX,IYY,IZZ,IUFL
DIMENSION IFLAG(IX,IY,IZ),JFLAG(KX,KY,KZ),IFLUS(JX,JY,JZ),IFRO(JZ)
DIMENSION T(JX,JY,JZ),T0(JX,JY,JZ),STMP(JZ)
DIMENSION H(KX,KY,KZ),H0(KX,KY,KZ)
DIMENSION CPG(JX,JY,JZ),WLKG(JX,JY,JZ),WTI(IX,IY,IZ,KD2),
&      RHOG(JX,JY,JZ),PHIG(JX,JY,JZ),PHIN(IX,IY,IZ),
&      SATU(IZ),VX(IX,IY,IZ),VY(IX,IY,IZ),VZ(IX,IY,IZ),ANISO(3)
DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
&      DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
&      DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
&      DXYZ(IX,IY,IZ)
DIMENSION TTX(KX,KY,KZ),TTY(KX,KY,KZ),TTZ(KX,KY,KZ),
&      SK(KX,KY,KZ),Q(KX,KY,KZ)
DIMENSION SC(JX,JY,JZ),SP(JX,JY,JZ),FRO(JZ,5)
DIMENSION WERT(IWERT),IXX(IWERT),IYY(IWERT),IZZ(IWERT),
&      FLAG(IWERT),IUFL(IWERT)
COMMON /ENER/SC,SP,FRO
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
&      DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
COMMON /PAR2/BASIS,HOENN,ALFAHY,OMGH,FEHLH,ZZZZ,ZIGNO
COMMON /TEMP/T,T0,STMP
COMMON /TRNS/TTX,TTY,TTZ,Q,SK
COMMON /WPOT/H,H0
COMMON /ZYK1/WERT,IXX,IYY,IZZ,IUFL
COMMON /ZYK2/FLAG
CELVIN = 273.15
ZAHL = WERT(L)
FLG = FLAG(L)

```

C

```

IF (IAN.GT.-1) GO TO 10
IF (FLG.EQ.'CPGE') CALL UMWREA (CPG,L,ZAHL,0,0)
IF (FLG.EQ.'GWNB') CALL UMWREA (Q,L,ZAHL,1,1)
IF (FLG.EQ.'GWVX') CALL UMWREA (VX,L,ZAHL,0,0)
IF (FLG.EQ.'GWVY') CALL UMWREA (VY,L,ZAHL,0,0)
IF (FLG.EQ.'GWVZ') CALL UMWREA (VZ,L,ZAHL,0,0)
IF (FLG.EQ.'HTOT') THEN
  CALL UMWREA (TTX,L,0.,0,0)
  CALL UMWREA (TTY,L,0.,0,0)
  CALL UMWREA (TTZ,L,0.,0,0)
  CALL UMWREA (SK,L,ZZZZ,0,0)
  CALL UMWREA (PHIN,L,ZZZZ,0,0)
  CALL UMWREA (H,L,999.,0,0)
  CALL UMWREA (H0,L,999.,0,0)
  CALL UMWINT (JFLAG,L,2.)
END IF
IF (FLG.EQ.'LEAK') THEN
  DELZ = DIFZ(IZZ(L))
  CALL UMWREA (TTZ,L,ZAHL * DELZ,0,0)
  IZZ(L) = IZZ(L) + 1
  CALL UMWREA (TTZ,L,ZAHL * DELZ,0,0)
  IZZ(L) = IZZ(L) - 1

```

```

END IF
IF (FLG.EQ.'PHIG') CALL UMWREA (PHIG,L,ZAHL / 100.,0,0)
IF (FLG.EQ.'PHIN') CALL UMWREA (PHIN,L,ZAHL / 100.,0,0)
IF (FLG.EQ.'RHOG') CALL UMWREA (RHOG,L,ZAHL,0,0)
IF (FLG.EQ.'RNZF') CALL UMWREA (Q,L,ZAHL,1,0)
IF (FLG.EQ.'SPEI') CALL UMWREA (SK,L,ZAHL,0,0)
IF (FLG.EQ.'TEMP') THEN
  CALL UMWDBL (T,L,ZAHL+CELVIN)
  CALL UMWDBL (TO,L,ZAHL+CELVIN)
END IF
IF (FLG.EQ.'TXYZ') THEN
  CALL UMWREA (TTX,L,ZAHL,0,0)
  CALL UMWREA (TTY,L,ZAHL,0,0)
  CALL UMWREA (TTZ,L,ZAHL,0,0)
END IF
IF (FLG.EQ.'TRSX') CALL UMWREA (TTX,L,ZAHL,0,0)
IF (FLG.EQ.'TRSY') CALL UMWREA (TTY,L,ZAHL,0,0)
IF (FLG.EQ.'TRSZ') CALL UMWREA (TTZ,L,ZAHL,0,0)
IF (FLG.EQ.'TTOT') THEN
  CALL UMWREA (WLKG,L,0.,0,0)
  CALL UMWREA (PHIG,L,0.,0,0)
  CALL UMWREA (PHIN,L,0.,0,0)
  CALL UMWREA (CPG,L,ZZZZ,0,0)
  CALL UMWDBL (T,L,999.+CELVIN)
  CALL UMWDBL (TO,L,999.+CELVIN)
  CALL UMWINT (IFLAG,L,2.)
END IF
IF (FLG.EQ.'WLKG') CALL UMWREA (WLKG,L,ZAHL,0,0)
IF (FLG.EQ.'WPOT') THEN
  CALL UMWREA (H,L,ZAHL,0,0)
  CALL UMWREA (HO,L,ZAHL,0,0)
END IF
10 IF (IABS(IAN).GT.0) THEN
  IF (FLG.EQ.'IFLG') CALL UMWINT (IFLAG,L,ZAHL)
  IF (FLG.EQ.'JFLG') CALL UMWINT (JFLAG,L,ZAHL)
  IF (FLG.EQ.'QUEL') CALL UMWREA (Q,L,ZAHL,0,0)
  IF (FLG.EQ.'SOUC') CALL UMWREA (SC,L,ZAHL,0,0)
  IF (FLG.EQ.'TEMP') THEN
    CALL UMWDBL (T,L,ZAHL+CELVIN)
    CALL UMWDBL (TO,L,ZAHL+CELVIN)
  END IF
END IF
IF (IAN.EQ.0.AND.IDATOP.EQ.0) THEN
  IF (FLG.EQ.'QUEL') CALL UMWREA (Q,L,-ZAHL,1,0)
  IF (FLG.EQ.'SOUC') CALL UMWREA (SC,L,0.,0,0)
  IF (FLG.EQ.'IFLG') CALL UMWINT (IFLAG,L,1.)
  IF (FLG.EQ.'JFLG') CALL UMWINT (JFLAG,L,1.)
END IF
RETURN
END

```

C&UMWREA

```

C *****
C   Version 20051988   Subroutine UMWREA
C *****

```

```

C
C *** Subroutine UMWREA Umwandeln der REAL - Felder *****
C
SUBROUTINE UMWREA (FELD,L,ZAHL,IFAK1,IFAK2)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
PARAMETER (IWERT=400)
CHARACTER FLAG*4
INTEGER*2 IXX,IYY,IZZ,IUFL
DIMENSION FELD(IX,IY,IZ)
DIMENSION WERT(IWERT),IXX(IWERT),IYY(IWERT),IZZ(IWERT),
& FLAG(IWERT),IUFL(IWERT)
DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
& DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
& DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
& DXYZ(IX,IY,IZ)
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
& DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /ZYG1/WERT,IXX,IYY,IZZ,IUFL
COMMON /ZYG2/FLAG

C
GO TO (100,200,300,400,500,600,700,800) IUFL(L) + 1

C
100 ZAHL1 = ZAHL * (DELTAX(IXX(L)) * DELTAY(IYY(L))) ** IFAK2
FELD(IXX(L),IYY(L),IZZ(L)) = FELD(IXX(L),IYY(L),IZZ(L)) * IFAK1
& + ZAHL1
GO TO 1000
200 DO 20 I = 1, NX
ZAHL1 = ZAHL * (DELTAX(I) * DELTAY(IYY(L))) ** IFAK2
FELD(I,IYY(L),IZZ(L)) = FELD(I,IYY(L),IZZ(L)) * IFAK1 + ZAHL1
20 CONTINUE
GO TO 1000
300 DO 30 J = 1, NY
ZAHL1 = ZAHL * (DELTAX(IXX(L)) * DELTAY(J)) ** IFAK2
FELD(IXX(L),J,IZZ(L)) = FELD(IXX(L),J,IZZ(L)) * IFAK1 + ZAHL1
30 CONTINUE
GO TO 1000
400 ZAHL1 = ZAHL * (DELTAX(IXX(L)) * DELTAY(IYY(L))) ** IFAK2
DO 40 K = 1, NZ
FELD(IXX(L),IYY(L),K) = FELD(IXX(L),IYY(L),K) * IFAK1 + ZAHL1
40 CONTINUE
GO TO 1000
500 DO 50 J = 1, NY
DO 50 I = 1, NX
ZAHL1 = ZAHL * (DELTAX(I) * DELTAY(J)) ** IFAK2
FELD(I,J,IZZ(L)) = FELD(I,J,IZZ(L)) * IFAK1 + ZAHL1
50 CONTINUE
GO TO 1000
600 DO 60 I = 1, NX
ZAHL1 = ZAHL * (DELTAX(I) * DELTAY(IYY(L))) ** IFAK2
DO 60 K = 1, NZ

```

```

        FELD(I,IYY(L),K) = FELD(I,IYY(L),K) * IFAK1 + ZAHL1
60 CONTINUE
    GO TO 1000
700 DO 70 J = 1, NY
    ZAHL1 = ZAHL * (DELTAX(IXX(L)) * DELTAY(J)) ** IFAK2
    DO 70 K = 1, NZ
    FELD(IXX(L),J,K) = FELD(IXX(L),J,K) * IFAK1 + ZAHL1
70 CONTINUE
    GO TO 1000
800 DO 80 J = 1, NY
    DO 80 I = 1, NX
    ZAHL1 = ZAHL * (DELTAX(I)) * DELTAY(J) ** IFAK2
    DO 80 K = 1, NZ
    FELD(I,J,K) = FELD(I,J,K) * IFAK1 + ZAHL1
80 CONTINUE
1000 CONTINUE
    RETURN
    END

```

C&UMWDBL

```

C *****
C   Version 20051988   Subroutine UMWDBL
C *****
C
C *** Subroutine UMWDBL Umwandeln der DOUBLE PRECISION - Felder *****
C

```

```

SUBROUTINE UMWDBL (FELD,L,ZAHL)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
PARAMETER (IWERT=400)
DOUBLE PRECISION FELD
CHARACTER FLAG*4
INTEGER*2 IXX,IYY,IZZ,IUFL
DIMENSION FELD(IX,IY,IZ)
DIMENSION WERT(IWERT),IXX(IWERT),IYY(IWERT),IZZ(IWERT),
& FLAG(IWERT),IUFL(IWERT)
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /ZYK1/WERT,IXX,IYY,IZZ,IUFL
COMMON /ZYK2/FLAG
GO TO (100,200,300,400,500,600,700,800) IUFL(L) + 1
100 FELD(IXX(L),IYY(L),IZZ(L)) = ZAHL
    GO TO 1000
200 DO 20 I = 1, NX
    FELD(I,IYY(L),IZZ(L)) = ZAHL
20 CONTINUE
    GO TO 1000
300 DO 30 J = 1, NY
    FELD(IXX(L),J,IZZ(L)) = ZAHL
30 CONTINUE
    GO TO 1000
400 DO 40 K = 1, NZ
    FELD(IXX(L),IYY(L),K) = ZAHL
40 CONTINUE

```

```

        GO TO 1000
500 DO 50 J = 1, NY
    DO 50 I = 1, NX
        FELD(I,J,IZZ(L)) = ZAHL
    50 CONTINUE
        GO TO 1000
600 DO 60 I = 1, NX
    DO 60 K = 1, NZ
        FELD(I,IYY(L),K) = ZAHL
    60 CONTINUE
        GO TO 1000
700 DO 70 J = 1, NY
    DO 70 K = 1, NZ
        FELD(IXX(L),J,K) = ZAHL
    70 CONTINUE
        GO TO 1000
800 DO 80 K = 1, NZ
    DO 80 J = 1, NY
    DO 80 I = 1, NX
        FELD(I,J,K) = ZAHL
    80 CONTINUE
1000 CONTINUE
        RETURN
        END

```

C&UMWINT

```

C *****
C   Version 20051988   Subroutine UMWINT
C *****
C
C *** Subroutine UMWINT Umwandeln der INTEGER - Felder *****
C

```

```

SUBROUTINE UMWINT (IFELD,L,ZAHL)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
PARAMETER (IWERT=400)
CHARACTER FLAG*4
INTEGER*2 IFELD
INTEGER*2 IXX,IYY,IZZ,IUFL
DIMENSION IFELD(IX,IY,IZ)
DIMENSION WERT(IWERT),IXX(IWERT),IYY(IWERT),IZZ(IWERT),
& FLAG(IWERT),IUFL(IWERT)
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /ZYK1/WERT,IXX,IYY,IZZ,IUFL
COMMON /ZYK2/FLAG
GO TO (100,200,300,400,500,600,700,800) IUFL(L) + 1
100 IFELD(IXX(L),IYY(L),IZZ(L)) = INT(ZAHL)
GO TO 1000
200 DO 20 I = 1, NX
    IFELD(I,IYY(L),IZZ(L)) = INT(ZAHL)
    20 CONTINUE
GO TO 1000
300 DO 30 J = 1, NY

```

```

    IFELD(IXX(L),J,IZZ(L)) = INT(ZAHL)
30 CONTINUE
    GO TO 1000
400 DO 40 K = 1, NZ
    IFELD(IXX(L),IYY(L),K) = INT(ZAHL)
40 CONTINUE
    GO TO 1000
500 DO 50 J = 1, NY
    DO 50 I = 1, NX
    IFELD(I,J,IZZ(L)) = INT(ZAHL)
50 CONTINUE
    GO TO 1000
600 DO 60 I = 1, NX
    DO 60 K = 1, NZ
    IFELD(I,IYY(L),K) = INT(ZAHL)
60 CONTINUE
    GO TO 1000
700 DO 70 J = 1, NY
    DO 70 K = 1, NZ
    IFELD(IXX(L),J,K) = INT(ZAHL)
70 CONTINUE
    GO TO 1000
800 DO 80 K = 1, NZ
    DO 80 J = 1, NY
    DO 80 I = 1, NX
    IFELD(I,J,K) = INT(ZAHL)
80 CONTINUE
1000 CONTINUE
    RETURN
    END

```

C&VISKOS

```

C *****
C   Version 06051988   Subroutine VISKOS
C *****
C *** Subroutine VISKOS Berechnung der kinem. Viskositaet von Wasser ***
C
    SUBROUTINE VISKOS (I,J,K,VISX,VISY,VISZ)
    PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
    PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
    PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
    PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
    DOUBLE PRECISION T,T0,STMP
    DIMENSION T(JX,JY,JZ),T0(JX,JY,JZ),STMP(JZ)
    DIMENSION CPG(JX,JY,JZ),WLKG(JX,JY,JZ),WTI(IX,IY,IZ,KD2),
&           RHOG(JX,JY,JZ),PHIG(JX,JY,JZ),PHIN(IX,IY,IZ),
&           SATU(IZ),VX(IX,IY,IZ),VY(IX,IY,IZ),VZ(IX,IY,IZ),ANISO(3)
    DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
&           DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
&           DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
&           DXYZ(IX,IY,IZ)
    COMMON /TEMP/T,T0,STMP
    COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
    COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,

```

```

&          DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
VISX = 1.
VISY = 1.
VISZ = 1.
T1 = REAL(T(I,J,K))
C
C *** ... des in X - Richtung fliessenden Wassers *****
C
IF (I.EQ.NX) GO TO 10
T2 = REAL(T(I+1,J,K))
IF (VX(I,J,K).EQ.0.) THEN
CALL HARMON (T1,T2,TT,FX(I))
ELSE
IF (VX(I,J,K).GT.0.) THEN
TT = T1
ELSE
TT = T2
END IF
END IF
VISX = VISKIN(TT)
10 CONTINUE
C
C *** ... des in Y - Richtung fliessenden Wassers *****
C
IF (J.EQ.NY) GO TO 20
T2 = REAL(T(I,J+1,K))
IF (VY(I,J,K).EQ.0.) THEN
CALL HARMON (T1,T2,TT,FY(J))
ELSE
IF (VY(I,J,K).GT.0.) THEN
TT = T1
ELSE
TT = T2
END IF
END IF
VISY = VISKIN(TT)
20 CONTINUE
C
C *** ... des in Z - Richtung fliessenden Wassers *****
C
IF (K.EQ.NZ) GO TO 30
T2 = REAL(T(I,J,K+1))
IF (VZ(I,J,K).EQ.0.) THEN
CALL HARMON (T1,T2,TT,FZ(K))
ELSE
IF (VZ(I,J,K).GT.0.) THEN
TT = T1
ELSE
TT = T2
END IF
END IF
VISZ = VISKIN(TT)
30 CONTINUE

```

```

RETURN
END
C&VDRUCK
C *****
C   Version 12101987  Subroutine VDRUCK
C *****
C *** Subroutine VDRUCK Drucken des Grundwasserfliessgeschwindigkeiten *
C
SUBROUTINE VDRUCK
PARAMETER (IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2)
PARAMETER (IP1=1, IP2=1, IQ1=0, IQ2=0)
PARAMETER (JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2, KY=IY*IP2+IQ2, KZ=IZ*IP2+IQ2)
DIMENSION DELTAX (IX), DELTAY (IY), DELTAZ (IZ), DIFX (IX), DIFY (IY),
&          DIFZ (IZ), XKOR (IX), YKOR (IY), ZKOR (IZ), DIVX (IX, 2),
&          DIVY (IY, 2), DIVZ (IZ, 2), FX (IX), FY (IY), FZ (IZ),
&          DXYZ (IX, IY, IZ)
DIMENSION CPG (JX, JY, JZ), WLKG (JX, JY, JZ), WTI (IX, IY, IZ, KD2),
&          RHOG (JX, JY, JZ), PHIG (JX, JY, JZ), PHIN (IX, IY, IZ),
&          SATU (IZ), VX (IX, IY, IZ), VY (IX, IY, IZ), VZ (IX, IY, IZ), ANISO (3)
COMMON /GITT/DELTAX, DELTAY, DELTAZ, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,
&          DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ
COMMON /PHYS/CPG, WLKG, ANISO, WTI, RHOG, PHIG, PHIN, SATU, VX, VY, VZ
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
C
IF (IECHO.GE.1) CALL ROUTIN ('VDRUCK')
C
WRITE (IAUS, 100)
WRITE (IAUS, 110)
WRITE (IAUS, 120)
DO 10 K = 1, NZ
WRITE (IAUS, 130) ZKOR(K), SATU(K) * 100.,
&                VX(1, 1, K), VY(1, 1, K), VZ(1, 1, K)
10 CONTINUE
100 FORMAT(' Grundwasserfliessgeschwindigkeit in m/s:',/)
110 FORMAT(' Tiefe (m)      Saettigung      X-Richtung      Y-Richtung
&      Z-Richtung')
120 FORMAT(1H , 70(1H-))
130 FORMAT(F8.2, 8X, F7.2, 6X, 3(1PE11.3, 4X))
RETURN
END
C&WARMIN
C *****
C   Version 15021988  Subroutine WARMIN
C *****
C *** Subroutine WARMIN liest das Ergebnis der vorherigen Simulation ***
C
SUBROUTINE WARMIN
PARAMETER (IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2)
PARAMETER (IP1=1, IP2=1, IQ1=0, IQ2=0)
PARAMETER (JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1)

```

```

PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,T0,STMP,CELV
DIMENSION T(JX,JY,JZ),T0(JX,JY,JZ),STMP(JZ)
DIMENSION H(KX,KY,KZ),H0(KX,KY,KZ)
DIMENSION IXOBS(10),IYOBS(10),IZOBS(10)
COMMON /TEMP/T,T0,STMP
COMMON /WPOT/H,H0
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
C
C IF (IECHO.GE.1) CALL ROUTIN ('WARMIN')
C
OPEN(9,FILE='TRADWS',STATUS='OLD',ERR=100)
IF (IGWF.LE.1) THEN
DO 10 K = 1, NZ
DO 10 J = 1, NY
READ (9,400,ERR=200,END=300) (T(I,J,K),I = 1, NX)
DO 10 I = 1, NX
T(I,J,K) = CELV(T(I,J,K))
10 CONTINUE
END IF
IF (IGWF.GE.1) THEN
DO 20 K = 1, NZ
DO 20 J = 1, NY
READ (9,500,ERR=200,END=300) (H(I,J,K),I = 1, NX)
20 CONTINUE
END IF
CLOSE(9)
RETURN
100 PRINT *,'Datei TRADWS nicht vorhanden - Simulation abgebrochen'
STOP
200 PRINT *,'Fehler in der Datenstruktur der Datei TRADWS'
STOP
300 PRINT *,'Unerwartet das Ende der Datei TRADWS erreicht'
STOP
400 FORMAT(6(10F8.3,/))
500 FORMAT(6(10F8.2,/))
END
C&WARMOU
C *****
C Version 15021988 Subroutine WARMOU
C *****
C *** Subroutine WARMOU schreibt Zwischenergebnis fuer einen Warmstart *
C
SUBROUTINE WARMOU
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T,T0,STMP,CELS
DIMENSION T(JX,JY,JZ),T0(JX,JY,JZ),STMP(JZ)
DIMENSION H(KX,KY,KZ),H0(KX,KY,KZ)

```

```

DIMENSION CPG(JX,JY,JZ),WLKG(JX,JY,JZ),WTI(IX,IY,IZ,KD2),
&          RHOG(JX,JY,JZ),PHIG(JX,JY,JZ),PHIN(IX,IY,IZ),
&          SATU(IZ),VX(IX,IY,IZ),VY(IX,IY,IZ),VZ(IX,IY,IZ),ANISO(3)
DIMENSION IXOBS(10),IYOBS(10),IZOBS(10)
COMMON /TEMP/T,TO,STMP
COMMON /WPOT/H,HO
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
COMMON /NDIM/NX,NY,NZ,NX1,NY1,NZ1
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
C
IF (IECHO.GE.1) CALL ROUTIN ('WARMOU')
C
OPEN(9,FILE='TRADWS')
IF (IGWF.LE.1) THEN
  DO 10 K = 1, NZ
  DO 10 J = 1, NY
  WRITE (9,100) (CELS(T(I,J,K)),I = 1, NX)
10  CONTINUE
END IF
C
C *** Piezometerhoehen ... *****
C
IF (IGWF.GE.1) THEN
  DO 20 K = 1, NZ
  DO 20 J = 1, NY
  WRITE (9,200) (H(I,J,K),I = 1, NX)
20  CONTINUE
  IF (IGWF.EQ.1) THEN
C
C *** Fliessgeschwindigkeit in X - Richtung *****
C
  DO 30 K = 1, NZ
  DO 30 J = 1, NY
  WRITE (9,300) (VX(I,J,K),I = 1, NX)
30  CONTINUE
C
C *** Fliessgeschwindigkeit in Y - Richtung *****
C
  DO 40 K = 1, NZ
  DO 40 J = 1, NY
  WRITE (9,300) (VY(I,J,K),I = 1, NX)
40  CONTINUE
C
C *** Fliessgeschwindigkeit in Z - Richtung *****
C
  DO 50 K = 1, NZ
  DO 50 J = 1, NY
  WRITE (9,300) (VZ(I,J,K),I = 1, NX)
50  CONTINUE
END IF
END IF
CLOSE(9)
RETURN

```

```

100 FORMAT(6(10F8.3,/))
200 FORMAT(6(10F8.2,/))
300 FORMAT(6(10(1PE9.2),/))
END
C&WASEIS
C *****
C   Version 19121988  Subroutine WASEIS
C *****
C *** Subroutine WASEIS Berechnung der Wasser/Eis - Verhaeltnisses *****
C
SUBROUTINE WASEIS (WE,TDIF,K)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
INTEGER*2 IFLAG,JFLAG,IFLUS,IFRO
DIMENSION IFLAG(IX,IY,IZ),JFLAG(KX,KY,KZ),IFLUS(JX,JY,JZ),IFRO(JZ)
DIMENSION SC(JX,JY,JZ),SP(JX,JY,JZ),FRO(JZ,5)
COMMON /ENER/SC,SP,FRO
COMMON /IFLD/IFLAG,JFLAG,IFLUS,IFRO
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST

C
C   IF (IECHO.GE.2) CALL ROUTIN ('WASEIS')
C
C   GO TO (10) IFRO(K)
C   WE = 1.
C   RETURN

C
C *** Gefriermodell 1:  Y = FRO(1) * EXP(FRO(2) * X) + FRO(3) *****
C
C   10 WE = FRO(K,1) * EXP(FRO(K,2) * TDIF) + FRO(K,3)
C   RETURN
C   END
C&WCPH2O
C *****
C   Version 19121988  Subroutine WCPH2O
C *****
C *** Subroutine WCPH2O Berechnung der Waermekapazitaet von Wasser *****
C
SUBROUTINE WCPH2O (CPWE,TDIF,K)
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /PAR3/CPW,CPE,CPL,WLW,WLE,WLL,RHOW,RHOE,RHOL,SENTHA
COMMON /PAR4/DTW,DTW2,DWAN,DTWL(3)

C
C   IF (IECHO.GE.2) CALL ROUTIN ('WCPH2O')
C
C   IF (TDIF + DTW2.LE.0.) THEN
C     CALL WASEIS (WE1,TDIF + DTW2,K)
C     CALL WASEIS (WE2,TDIF - DTW2,K)
C     DT = DTW

```

```

ELSE
  WE1 = 1.
  CALL WASEIS (WE2,-DTW2,K)
  DT = DTW2
END IF
DWE = WE1 - WE2
DWEM = (WE1 + WE2) / 2.
CPWE = CPW * DWEM + CPE * (1. - DWEM) + SENTHA * DWE / DT
RETURN
END

```

C&WFLUSS

```

C *****
C   Version 21111987 Subroutine WFLUSS
C *****
C
C *** Subroutine WFLUSS Berechnung des Waermeflusses zum Waermetauscher
C
SUBROUTINE WFLUSS (IMAL)
PARAMETER (IX=45, IY=45, IZ=1, IMAX=45, KD=6, KD2=KD/2)
PARAMETER (IP1=1, IP2=1, IQ1=0, IQ2=0)
PARAMETER (JX=IX*IP1+IQ1, JY=IY*IP1+IQ1, JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2, KY=IY*IP2+IQ2, KZ=IZ*IP2+IQ2)
DOUBLE PRECISION T, TO, STMP
INTEGER*2 IFLAG, JFLAG, IFLUS, IFRO
DIMENSION DELTAX (IX), DELTAY (IY), DELTAZ (IZ), DIFX (IX), DIFY (IY),
&          DIFZ (IZ), XKOR (IX), YKOR (IY), ZKOR (IZ), DIVX (IX, 2),
&          DIVY (IY, 2), DIVZ (IZ, 2), FX (IX), FY (IY), FZ (IZ),
&          DXYZ (IX, IY, IZ)
DIMENSION CPG (JX, JY, JZ), WLKG (JX, JY, JZ), WTI (IX, IY, IZ, KD2),
&          RHOG (JX, JY, JZ), PHIG (JX, JY, JZ), PHIN (IX, IY, IZ),
&          SATU (IZ), VX (IX, IY, IZ), VY (IX, IY, IZ), VZ (IX, IY, IZ), ANISO (3)
DIMENSION IFLAG (IX, IY, IZ), JFLAG (KX, KY, KZ), IFLUS (JX, JY, JZ), IFRO (JZ)
DIMENSION T (JX, JY, JZ), TO (JX, JY, JZ), STMP (JZ)
COMMON /GITT/DELTAX, DELTAY, DELTAZ, DIFX, DIFY, DIFZ, XKOR, YKOR, ZKOR,
&          DIVX, DIVY, DIVZ, FX, FY, FZ, DXYZ
COMMON /IFLD/IFLAG, JFLAG, IFLUS, IFRO
COMMON /TEMP/T, TO, STMP
COMMON /PHYS/CPG, WLKG, ANISO, WTI, RHOG, PHIG, PHIN, SATU, VX, VY, VZ
COMMON /NDIM/NX, NY, NZ, NX1, NY1, NZ1
COMMON /INF2/IDIF, IDRU, IFROST, IAUS, IECHO, MODEL, NIT, IWCP, ISOR, IEN1
COMMON /PAR1/DZEIT, DZMF, DZF, FLURAB, ALFATP, FEHLT, FEHLER, OMT, SUMSEK

```

C

```

IF (IECHO.GE.1) CALL ROUTIN ('WFLUSS')

```

C

```

IDIFF = IDIF
WATT = 0.
DO 10 K = 1, NZ
DO 10 J = 1, NY
DO 10 I = 1, NX
IF (IFLAG(I,J,K).EQ.5) THEN
  IF (IDIFF.EQ.1) THEN
    IDIF = 0
    CALL KONINF (I,J,K)
    CALL KONINF (I-1,J,K)
  
```

```

        CALL KONINF (I,J-1,K)
        CALL KONINF (I,J,K-1)
    END IF
    IF (I.GT.1.AND.IFLAG(I-1,J,K).EQ.1)
&      WATT = WATT + WTI(I-1,J,K,1) * DELTAY(J) * DELTAZ(K) *
&          (T(I-1,J,K) - T(I,J,K)) / DIFX(I-1)
    IF (I.LT.NX.AND.IFLAG(I+1,J,K).EQ.1)
&      WATT = WATT + WTI(I,J,K,1) * DELTAY(J) * DELTAZ(K) *
&          (T(I+1,J,K) - T(I,J,K)) / DIFX(I)
    IF (J.GT.1.AND.IFLAG(I,J-1,K).EQ.1)
&      WATT = WATT + WTI(I,J-1,K,2) * DELTAX(I) * DELTAZ(K) *
&          (T(I,J-1,K) - T(I,J,K)) / DIFY(J-1)
    IF (J.LT.NY.AND.IFLAG(I,J+1,K).EQ.1)
&      WATT = WATT + WTI(I,J,K,2) * DELTAX(I) * DELTAZ(K) *
&          (T(I,J+1,K) - T(I,J,K)) / DIFY(J)
    IF (K.GT.1.AND.IFLAG(I,J,K-1).EQ.1)
&      WATT = WATT + WTI(I,J,K-1,3) * DELTAX(I) * DELTAY(J) *
&          (T(I,J,K-1) - T(I,J,K)) / DIFZ(K-1)
    IF (K.LT.NZ.AND.IFLAG(I,J,K+1).EQ.1)
&      WATT = WATT + WTI(I,J,K,3) * DELTAX(I) * DELTAY(J) *
&          (T(I,J,K+1) - T(I,J,K)) / DIFZ(K)
    END IF
10  CONTINUE
    IDIF = IDIFF
    WATPH = 3600. * WATT / DZEIT * IMAL
    WRITE (IAUS,110) WATT
    WRITE (IAUS,120) WATPH
110  FORMAT(' Waermefluss zum Waermetauscher           ',1PE15.6,
&          ' Watt')
120  FORMAT(' Waermefluss pro Zeiteinheit           ',1PE15.6,
&          ' Watt/Stunde')
    RETURN
    END
C&WICHTP
C *****
C   Version 03081988  Subroutine WICHTP
C *****
C
C *** Subroutine WICHTP Wichtung zwischen Explizit- und Implizit-Schema
C
    SUBROUTINE WICHTP (I,J,K,TT)
    PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
    PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
    PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
    DOUBLE PRECISION T,T0,STMP,TT
    DIMENSION T(JX,JY,JZ),T0(JX,JY,JZ),STMP(JZ)
    COMMON /TEMP/T,T0,STMP
    COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
    TT = ALFATP * T(I,J,K) + (1. - ALFATP) * T0(I,J,K)
    RETURN
    END
C&WICHHY
C *****
C   Version 03081988  Subroutine WICHHY

```

```

C *****
C
C *** Subroutine WICHHY Wichtung zwischen Explizit- und Implizit-Schema
C
  SUBROUTINE WICHHY (I,J,K,HH)
  PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
  PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
  PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
  DIMENSION H(KX,KY,KZ),H0(KX,KY,KZ)
  COMMON /PAR2/BASIS,HOENN,ALFAHY,OMGH,FEHLH,ZZZZ,ZIGNO
  COMMON /WPOT/H,H0
  HH = ALFAHY * H(I,J,K) + (1. - ALFAHY) * H0(I,J,K)
  RETURN
  END
C&ZEITBE
C *****
C   Version 13081987 Subroutine ZEITBE
C *****
C *** Subroutine ZEITBE Ausgabe der abgelaufenen Zeit auf IUNIT *****
C
  SUBROUTINE ZEITBE (IUNIT)
C
  COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
  COMMON /PAR1/DZEIT,DZMF,DZF,FLURAB,ALFATP,FEHLT,FEHLER,OMGT,SUMSEK
  IF (IECHO.GE.1) CALL ROUTIN ('ZEITBE')
C
  ZSEC = SUMSEK
  ZMIN = ZSEC / 60.
  ZSTU = ZMIN / 60.
  ZTAG = ZSTU / 24.
  ZWOC = ZTAG / 7.
  ISEC = INT(ZSEC) - INT(ZMIN) * 60
  IMIN = INT(ZMIN) - INT(ZSTU) * 60
  ISTU = INT(ZSTU) - INT(ZTAG) * 24
  ITAG = INT(ZTAG) - INT(ZWOC) * 7
  IWOC = INT(ZWOC)
  IF (IUNIT.GT.1) WRITE (IUNIT,10) IWOC,ITAG,ISTU,IMIN,ISEC
  IF (IUNIT.EQ.1.AND.IECHO.GE.1)
  & WRITE (IUNIT,20) IWOC,ITAG,ISTU,IMIN,ISEC
  IF (IUNIT.EQ.1.AND.IECHO.EQ.0)
  & WRITE (IUNIT,30) IWOC,ITAG,ISTU,IMIN,ISEC,NIT
10 FORMAT(/,' Temperaturverteilung nach ',I3,' Wochen ',I2,' Tagen ',
  & I2,' Stunden ',I2,' Minuten ',I2,' Sekunden',/)
20 FORMAT(' Abgelaufene Zeit: ',I3,' Wochen ',I2,' Tagen ',
  & I2,' Stunden ',I2,' Minuten ',I2,' Sekunden')
30 FORMAT(1H+,' Abgelaufene Zeit: ',I3,' Wochen ',I2,' Tagen ',
  & I2,' Stunden ',I2,' Min. ',I2,' Sek. NIT = ',I3)
  RETURN
  END
C&ZELLEN
C *****
C   Version 18081987 Subroutine ZELLEN
C *****

```

```

C
C *** Subroutine ZELLEN Berechnung der Waermeenergie der Zelle (I,J,K) *
C
SUBROUTINE ZELLEN (I,J,K,TEMP,STEMP)
PARAMETER (IX=45,IY=45,IZ=1,IMAX=45,KD=6,KD2=KD/2)
PARAMETER (IP1=1,IP2=1,IQ1=0,IQ2=0)
PARAMETER (JX=IX*IP1+IQ1,JY=IY*IP1+IQ1,JZ=IZ*IP1+IQ1)
PARAMETER (KX=IX*IP2+IQ2,KY=IY*IP2+IQ2,KZ=IZ*IP2+IQ2)
IMPLICIT DOUBLE PRECISION (E)
DOUBLE PRECISION TEMP,STEMP
DIMENSION DELTAX(IX),DELTAY(IY),DELTAZ(IZ),DIFX(IX),DIFY(IY),
&          DIFZ(IZ),XKOR(IX),YKOR(IY),ZKOR(IZ),DIVX(IX,2),
&          DIVY(IY,2),DIVZ(IZ,2),FX(IX),FY(IY),FZ(IZ),
&          DXYZ(IX,IY,IZ)
DIMENSION CPG(JX,JY,JZ),WLKG(JX,JY,JZ),WTI(IX,IY,IZ,KD2),
&          RHOG(JX,JY,JZ),PHIG(JX,JY,JZ),PHIN(IX,IY,IZ),
&          SATU(IZ),VX(IX,IY,IZ),VY(IX,IY,IZ),VZ(IX,IY,IZ),ANISO(3)
DIMENSION SC(JX,JY,JZ),SP(JX,JY,JZ),FRO(JZ,5)
COMMON /GITT/DELTAX,DELTAY,DELTAZ,DIFX,DIFY,DIFZ,XKOR,YKOR,ZKOR,
&          DIVX,DIVY,DIVZ,FX,FY,FZ,DXYZ
COMMON /ENER/SC,SP,FRO
COMMON /PHYS/CPG,WLKG,ANISO,WTI,RHOG,PHIG,PHIN,SATU,VX,VY,VZ
COMMON /PAR3/CPW,CPE,CPL,WLW,WLE,WLL,RHOW,RHOE,RHOL,SENTHA
COMMON /EKAL/EALT,EANF,ESUM
VOLW = DXYZ(I,J,K) * PHIG(I,J,K) * SATU(K)
VOLL = DXYZ(I,J,K) * PHIG(I,J,K) * (1. - SATU(K))
VOLG = DXYZ(I,J,K) * (1. - PHIG(I,J,K))
IF (TEMP - STEMP) 10,10,20

C
C *** ... gefroren oder teilgefroren *****
C
10 TDIF = TEMP - STEMP
CALL WASEIS (WE,TDIF,K)
SENT = WE * SENTHA * VOLW * RHOW
ESUM = TEMP * (VOLW * (WE * CPW * RHOW + (1. - WE) * CPE * RHOE)
&          + VOLL * CPL * RHOL
&          + VOLG * RHOG(I,J,K) * CPG(I,J,K))
&          + SENT
RETURN

C
C *** ... fluessig *****
C
20 SENT = SENTHA * VOLW * RHOW
ESUM = TEMP * (VOLW * RHOW * CPW
&          + VOLL * RHOL * CPL
&          + VOLG * RHOG(I,J,K) * CPG(I,J,K))
&          + SENT
RETURN
END

C&ZUSDAT
C *****
C      Version 30111987 Subroutine ZUSDAT
C *****
C

```

```

C *** Subroutine ZUSDAT Einlesen zusaetzlicher, zeitabhaengiger Daten **
C
SUBROUTINE ZUSDAT(IEINS)
PARAMETER (IWERT=400)
CHARACTER FLAG*4,FORMIN*20
INTEGER*2 IXX,IYY,IZZ,IUFL
DIMENSION WERT(IWERT),IXX(IWERT),IYY(IWERT),IZZ(IWERT),
&          FLAG(IWERT),IUFL(IWERT)
DIMENSION IXOBS(10),IYOBS(10),IZOBS(10)
COMMON /INF1/IZEIT,IPECL,IVOLDF,IAUSOP,ILOOP,IEIN,INPOP,IALTER
COMMON /INF2/IDIF,IDRU,IFROST,IAUS,IECHO,MODEL,NIT,IWCP,ISOR,IEN1
COMMON /INF3/LUMW,IENER,ISTOP,IDATOP,LISDAT,IDZTST
COMMON /INF4/IOBS,IXOBS,IYOBS,IZOBS,IWARMS,IWSDAT,IDSTOP,IGWF
COMMON /PAR2/BASIS,HOENN,ALFAHY,OMGH,FEHLH,ZZZZ,ZIGNO
COMMON /ZYK1/WERT,IXX,IYY,IZZ,IUFL
COMMON /ZYK2/FLAG

C
IF (IECHO.GE.1) CALL ROUTIN ('ZUSDAT')

C
IF (IEINS.EQ.0) THEN
READ (8,100) FORMIN
READ (8,FORMIN) LDAT
IF (LUMW+LDAT.GT.IWERT) THEN
WRITE (1,110) LUMW + LDAT
STOP
END IF
READ (8,100) FORMIN
DO 10 L = LUMW + 1, LUMW + LDAT
READ (8,FORMIN) FLAG(L),IXX(L),IYY(L),IZZ(L),IUFL(L)
10 CONTINUE
READ (8,105) FORMIN
IEINS = 1
END IF
READ (8,FORMIN,ERR=200,END=300) (WERT(LUMW+L), L = 1, LDAT)
DO 20 L = 1, LDAT
IF (WERT(LUMW+L).NE.ZIGNO) CALL UMWAND (LUMW+L,-1)
20 CONTINUE
RETURN
200 WRITE (1,120)
STOP
300 IF (IGWF.LE.1) CALL TMPDRU
IF (IGWF.GE.1) CALL PIEDRU
CALL ISODRV
IF (IENER.EQ.1) CALL ENERGI
IF (IWSDAT.EQ.1) CALL WARMOU
WRITE (1,130)
STOP
100 FORMAT(A20)
105 FORMAT(A20,/)
110 FORMAT(' Es sollen ',I4,' Werte umgewandelt werden - Feld IWERT',
&         ' zu klein dimensioniert')
120 FORMAT(' Fehler in der Datenstruktur der Zusatzdaten')
130 FORMAT(' Ende der optionalen Eingabedatei erreicht -',
&         ' Bearbeitung abgebrochen')

```

END

→

Anhang 4:

Beispieleingabedaten

Beispiel Kapitel 5.1: Datensatz ZIEGEL

Berechnung der eindim. Waermeausbreitung in einer Hoehofenwand 11 Knoten

```
(415)          IXISO IYISO IZISO IMSBIN
  0  0  1  0
(315,5X,3A20) IOBS IDATOP LISDAT XYZFIL AUS DATOP
  0  0  0      OBSDAT          ERGEB          ZUSDAT
(715,2X,A3,3F10.0) IZEIT ILOOP IEIN IAUSOP INPOP IDRU ISTLAE ZFLAG DZF FEHLT Z
  1 100  1  3  0  1  1 SEK          1.0  .01000  1E30
(F5.0,715)      FLURAB MODEL IENER ISTOP IDZTST IWARMS IWSDAT IDSTOP
  0.  0  0  0  0  0  0  0
(2F5.2,815)     ALFATP OMGT IVOLDF IPECL IDIF IFROST IECHO IALTER IWCP ISOR
  1.00 1.88  1  1  0  0  1  1  0  0
(5F10.2,315)    BASIS HOENN ALFAHY OMGH FEHLH IGWF IVIS IWRO
  0.00      0.00  0.00  0.00  0.00  0  0  0
(315)          IPROX IPROY IPROZ
  0  0  1
(315)          NX NY NZ
  11  1  1
(315,3F5.2)     IBX IBY IBZ ANISOX ANISOY ANISOZ
  1  0  0  1.00 1.00 1.00
(10F6.4)       DELTAX(NX) IVOLDF=0 oder DIFX(NX-1) IVOLDF=1
  .1000 .1000 .0999 .0002 .0899 .0899 .0002 .0899 .0900 .0900
( 1F5.2)       DELTAY(NY) IVOLDF=0 oder DIFY(NY-1) IVOLDF=1
  1.
( 1F5.2)       DELTAZ(NZ) IVOLDF=0 oder DIFZ(NZ-1) IVOLDF=1
  1.
(11F7.0,12)    T CPG RHOG WLKG PHIG PHIN SATU SC WINKEL GWV VZ IFLAG
  0.00 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  1
(A4,6X,F10.0,415) FLAG WERT IXX IYY IZZ IUFL
WLKG      3.200  1  1  1  0
WLKG      3.200  2  1  1  0
WLKG      3.200  3  1  1  0
WLKG      3.200  4  1  1  0
WLKG      0.100  5  1  1  0
WLKG      0.100  6  1  1  0
WLKG      0.100  7  1  1  0
WLKG      0.900  8  1  1  0
WLKG      0.900  9  1  1  0
WLKG      0.900 10  1  1  0
WLKG      0.900 11  1  1  0
TEMP      950.000  1  1  1  0
TEMP      50.000  11  1  1  0
ENDE
```

Beispiel Kapitel 5.2: Datensatz TRADERF

```

Testdatensatz 25 * 1 instationaerer Waermetransport in einem Gesteinskoeper
(415)          IXISO IYISO IZISO IMSBIN
      0   0   1   0
(315,5X,3A20)  IOBS IDATOP LISDAT XYZFIL AUS DATOP
      0   0   0   OBSDAT          TRADERG          ZUSDAT
(715,2X,A3,3F10.0) IZEIT ILOOP IEIN IAUSOP INPOP IDRU ISTLAE ZFLAG DZF FEHLT Z
      192  1 192  1   0   6   1  STU          1.0  0.0001  1E30
(F5.0,715)     FLURAB MODEL IENER ISTOP IDZTST IWARMS IWSDAT IDSTOP
      0.   0   0   0   0   0   0   0
(2F5.2,815)    ALFATP OMGT IVOLDF IPECL IDIF IFROST IECHO IALTER IWCP ISOR
      1.0 -1.00  0   0   1   0   0   0   1   0
(5F10.2,315)   BASIS HOENN ALFAHY OMGH FEHLH IGWF IVIS IWRO
      0.00    0.00    0.00    0.00    0.00    0   0   0
(315)          IPROX IPROY IPROZ
      0   0   1
(315)          NX NY NZ
      30  1   1
(315,3F5.2)    IBX IBY IBZ ANISOX ANISOY ANISOZ
      1   0   0  1.0  1.0  1.0
(3(10F7.0,/)) DELTAX(NX) IVOLDF=0 oder DIFX(NX-1) IVOLDF=1
      0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1
      0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1
      0.1  0.1  0.2  0.4  0.8  1.6  3.2  6.4  12.8  25.6
(1F7.1)        DELTAY(NY) IVOLDF=0 oder DIFY(NY-1) IVOLDF=1
      1.0
(1F7.1)        DELTAZ(NZ) IVOLDF=0 oder DIFZ(NZ-1) IVOLDF=1
      1.0
(11F7.0,I2)    T CPG RHOG WLKG PHIG PHIN SATU SC WINKEL GWV VZ IFLAG
      20.00 850.00  2.60  3.00  0.00  0.00  0.00  0.00  0.00  0.0  0.0  1
(A4,6X,F10.0,415) FLAG WERT IXX IYY IZZ IUFL
TEMP          5.0000  1   1   1   0
ENDE

```

Beispiel Kapitel 5.3: Datensatz BODVAR2

Konvektive Waermeausbreitung nach BODVARSSON(1982) --- dt = 1 Sekunde

```

(415)          IXISO IYISO IZISO IMSBIN
      0   0   1   0
(315,5X,3A20)  IOBS IDATOP LISDAT XYZFIL AUS DATOP
      1   0   0   OBSDAT          ERGEB          ZUSDAT
(715,2X,A3,3F10.7) IZEIT ILOOP IEIN IAUSOP INPOP IDRU ISTLAE ZFLAG DZF FEHLT Z
      1000  1 1000  3   0  100  1  SEK          1.0  .000010  1E30
(F5.0,715)     FLURAB MODEL IENER ISTOP IDZTST IWARMS IWSDAT IDSTOP
      0.   0   0   0   0   0   0   0
(2F5.2,815)    ALFATP OMGT IVOLDF IPECL IDIF IFROST IECHO IALTER IWCP ISOR
      0.00 1.0  0   4   1   0   0   0   2   0
(6F10.0,/6F10.0) CPW CPE CPL WLW WLE WLL / RHOW RHOE RHOL SENTHA DTW DWAN
      0.   0.   0.   0.000  0.000  0.000
      1.   0.   0.   0.   0.000  0.000

```

```

(5F10.2,315)      BASIS HOEN ALFAHY OMGH FEHLH IGWF IVIS IWRO
      0.00      0.00      0.00      0.00      0.00      0      0      0
(315)            IPROX IPROY IPROZ
      0      0      1
(315)            IXOBS IYOBS IZOBS
      2      1      1
(315)            NX NY NZ
      2      1      1
(315,3F5.2)      IBX IBY IBZ ANISOX ANISOY ANISOZ
      0      0      0 1.00 1.00 1.00
( 2F5.0)         DELTAX(NX) IVOLDF=0 oder DIFX(NX-1) IVOLDF=1
      1.      1.
( 1F5.0)         DELTAY(NY) IVOLDF=0 oder DIFY(NY-1) IVOLDF=1
      1.
( 1F5.0)         DELTAZ(NZ) IVOLDF=0 oder DIFZ(NZ-1) IVOLDF=1
      1.
(11F7.0,12)      T CPG RHOG WLKG PHIG PHIN SATU SC WINKEL GWV VZ IFLAG
      0.00      0.00      0.000      0.00 100.00 100.00 100.00      0.00 90.00      1E-3      0.00 1
(A4,6X,F10.0,4I5) FLAG WERT IXX IYY IZZ IUFL
TEMP          100.000      1      1      1      0
TEMP          200.000      2      1      1      0
ENDE

```

Beispiel Kapitel 5.4: Datensatz WANG

WANG & ANDERSON (1982) S. 79ff Entnahme an Knoten 21,21 - gespannt

```

(415)            IXISO IYISO IZISO IMSBIN
      0      0      1      1
(315,5X,3A20)    IOBS IDATOP LISDAT XYZFIL AUS DATOP
      2      0      0      WANGOBS          NUL          NUL
(715,2X,A3,3F10.0) IZEIT ILOOP IEIN IAUSOP INPOP IDRU ISTLAE ZFLAG DZF FEHLT Z
      16 200 16      1      0      1 864 SEK          1.5      0.001      1E30
(F5.0,7I5)       FLURAB MODEL IENER ISTOP IDZTST IWARMS IWSDAT IDSTOP
      0.      0      0      0      0      0      1      0
(2F5.2,8I5)      ALFATP SORT IVOLDF IPECL IDIF IFROST IECHO IALTER IWCP ISOR
      0.00 0.00      0      0      0      0      1      0      0      0
(5F10.0,315)     BASIS HOENN ALFAHY SORH FEHLH IGWF IVIS IWRO
      0.00      0.00      1.00      -1.20      0.001      2      0      0
(315)            IPROX IPROY IPROZ
      0      0      1
(315)            IXOBS IYOBS IZOBS
      21 22      1
      21 21      1
(315)            NX NY NZ
      41 41      1
(315,3F10.0)     IBX IBY IBZ ANISOX ANISOY ANISOZ (ANISO(3))
      0      0      0      1.000      1.000      1.000
(5(10F7.0,/))    DELTAX(NX) IVOLDF=0 oder DIFX(NX-1) IVOLDF=1
      100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0
      100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0
      100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0
      100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0

```

```

100.0
(5(10F7.0,/))      DELTAY(NY) IVOLDF=0 oder DIFY(NY-1) IVOLDF=1
100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0
100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0
100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0
100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0
100.0
(1F7.0)           DELTAZ(NZ) IVOLDF=0 oder DIFZ(NZ-1) IVOLDF=1
1.0
(7F10.0,15)      H TTX TTY TTZ Q SK PHIN JFLAG
0.003.47222E-33.47222E-3 0.00 0.00 0.002 2.00 1
(A4,6X,F10.0,4I5) FLAG WERT IXX IYY IZZ IUFL
QUEL -.02314819 21 21 1 0
ENDE

```

Beispiel Kapitel 5.5: Datensatz INST2

```

TESTDATENSATZ 20 * 20 INSTATIONAER - 2 BRUNNEN - 9 PUMPSTUFEN UNGESPANNT
(4I5)           IXISO IYISO IZISO IMSBIN
0 0 1 1
(3I5,5X,3A20)  IOBS IDATOP LISDAT XYZFIL AUS DATOP
3 1 12 INSTMES NUL INSTZUS
(7I5,2X,A3,3F10.0) IZEIT ILOOP IEIN IAUSOP INPOP IDRU ISTLAE ZFLAG DZF FEHLT Z
108 200 108 12 0 1 120 MIN 1.0 0.001 1E30
(F5.0,7I5)     FLURAB MODEL IEENER ISTOP IDZTST IWARMS IWSDAT IDSTOP
0. 1 0 0 0 0 0 0
(2F5.2,8I5)    ALFATP SORT IVOLDF IPECL IDIF IFROST IECHO IALTER IWCP ISOR
0.00 0.00 0 0 0 0 0 1 0 0
(5F10.0,3I5)   BASIS HOENN ALFAHY SORH FEHLH IGWF IVIS IWRO
0.00 50.00 1.00 -1.64 0.00010 2 0 0
(3I5)         IPROX IPROY IPROZ
0 0 2
(3I5)         IXOBS IYOBS IZOBS
9 8 1
13 11 1
10 10 1
(3I5)         NX NY NZ
20 20 1
(3I5,3F10.4)  IBX IBY IBZ ANISOX ANISOY ANISOZ (ANISO(3))
2 0 0 1.000 1.000 1.000
(2(10F7.3,/)) DELTAX(NX) IVOLDF=0 ODER DIFX(NX-1) IVOLDF=1
50. 50. 50. 50. 50. 50. 40. 30. 20. 30.
40. 30. 20. 30. 40. 50. 50. 50. 50. 50.
(2(10F7.3,/)) DELTAY(NY) IVOLDF=0 ODER DIFY(NY-1) IVOLDF=1
50. 50. 50. 50. 50. 40. 30. 20. 30. 30.
20. 30. 40. 50. 50. 50. 50. 50. 50. 50.
(1F7.0)       DELTAZ(NZ) IVOLDF=0 ODER DIFZ(NZ-1) IVOLDF=1
50.
(7F10.0,15)   H TTX TTY TTZ Q SK PHIN JFLAG
50.00 1E-4 13E-5 0.00 0.00 0.100 10.00 1

```

(A4,6X,F10.4,4I5) FLAG WERT IXX IYY IZZ IUFL
 ENDE

Beispiel Kapitel 5.5: Datensatz INSTZUS

```
(15)          LDAT (9 Pumpstufen von 2 Brunnen)
  2
(A4,6X,4I5)   FLAG IXX IYY IZZ IUFL
QUEL         9  8  1  0
QUEL        13 11  1  0
(2F10.0)
Brunnen 1  Brunnen 2
-0.0100  -0.0300
-0.0200  -0.0400
-0.0300  -0.0500
-0.0400  -0.0600
-0.0500  -0.0700
-0.0600   0.0000
-0.0700   0.0000
-0.0800   0.0000
-0.0900   0.0000
```

Beispiel Kapitel 5.6: Datensatz LEAK

```
Testdatensatz 15 * 10 -0.10 cbm/s Entnahme an Knoten 10,5 - ungespannt
(4I5)          IXISO IYISO IZISO IMSBIN
  0  0  1  1
(3I5,5X,3A20) IOBS IDATOP LISDAT XYZFIL AUS DATOP
  2  0  0  OBSDAT          ERGEB          NUL
(7I5,2X,A3,3F10.7) IZEIT ILOOP IEIN IAUSOP INPOP IDRU ISTLAE ZFLAG DZF FEHLT Z
  1 200  1  1  0  1  60 SEK      1.0  0.001  1E30
(F5.0,7I5)     FLURAB MODEL IEINER ISTOP IDZTST IWARMS IWSDAT IDSTOP
  0.  1  0  0  0  0  1  0
(2F5.2,8I5)    ALFATP SORT IVOLDF IPECL IDIF IFROST IECHO IALTER IWCP ISOR
  0.00 0.00  0  0  0  0  1  0  0  0
(5F10.0,3I5)   BASIS HOENN ALFAHY SORH FEHLH IGWF IVIS IWRO
  10.00  80.00  1.00  -1.64  0.0010  2  0  0
(3I5)          IPROX IPROY IPROZ
  0  0  2
(3I5)          IXOBS IYOBS IZOBS
  2  7  2
  10 5  2
(3I5)          NX NY NZ
  15 10  2
(3I5,3F10.4)   IBX IBY IBZ ANISOX ANISOY ANISOZ (ANISO(3))
  2  0  0  1.000  1.000  1.000
(2(10F7.3,/) ) DELTAX(NX) IVOLDF=0 oder DIFX(NX-1) IVOLDF=1
  100. 100. 100. 100. 100. 100. 100. 100. 100. 100. 100.
  100. 100. 100. 100. 100.
(10F7.3)       DELTAY(NY) IVOLDF=0 oder DIFY(NY-1) IVOLDF=1
```

```

100. 100. 100. 100. 100. 100. 100. 100. 100. 100.
(2F7.0) DELTAZ(NZ) IVOLDF=0 oder DIFZ(NZ-1) IVOLDF=1
1E-6 70.00
(7F10.0,15) H TTX TTY TTZ Q SK PHIN JFLAG
0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 2
77.50 3E-4 3E-4 0.0 0.0 0.0 0.0 1
(A4,6X,F10.0,415) FLAG WERT IXX IYY IZZ IUFL
QUEL -0.100 10 5 2 0
GWNB 3E-9 0 0 2 4
WPOT 80.000 1 0 2 2
WPOT 75.000 15 0 2 2
HTOT 0.0 2 1 2 0
HTOT 0.0 3 1 2 0
HTOT 0.0 4 1 2 0
HTOT 0.0 5 1 2 0
HTOT 0.0 1 2 2 0
HTOT 0.0 2 2 2 0
HTOT 0.0 3 2 2 0
HTOT 0.0 4 2 2 0
HTOT 0.0 5 2 2 0
HTOT 0.0 1 9 2 0
HTOT 0.0 2 9 2 0
HTOT 0.0 3 9 2 0
HTOT 0.0 4 9 2 0
HTOT 0.0 5 9 2 0
HTOT 0.0 1 10 2 0
HTOT 0.0 2 10 2 0
HTOT 0.0 3 10 2 0
HTOT 0.0 4 10 2 0
HTOT 0.0 5 10 2 0
WPOT 82.0 1 7 1 0
WPOT 81.5 2 7 1 0
WPOT 81.0 3 7 1 0
WPOT 80.5 4 7 1 0
WPOT 80.0 5 7 1 0
WPOT 79.5 6 7 1 0
WPOT 79.0 7 7 1 0
WPOT 78.5 8 7 1 0
WPOT 78.0 9 7 1 0
WPOT 77.5 10 7 1 0
WPOT 77.0 11 7 1 0
WPOT 76.5 12 7 1 0
WPOT 76.0 13 6 1 0
WPOT 75.5 14 6 1 0
WPOT 75.0 15 6 1 0
LEAK 5E-6 1 7 1 0
LEAK 5E-6 2 7 1 0
LEAK 5E-6 3 7 1 0
LEAK 5E-6 4 7 1 0
LEAK 5E-6 5 7 1 0
LEAK 5E-6 6 7 1 0
LEAK 5E-6 7 7 1 0
LEAK 5E-6 8 7 1 0

```

LEAK	5E-6	9	7	1	0
LEAK	5E-6	10	7	1	0
LEAK	5E-6	11	7	1	0
LEAK	5E-6	12	7	1	0
LEAK	5E-6	13	6	1	0
LEAK	5E-6	14	6	1	0
LEAK	5E-6	15	6	1	0
ENDE					

Anhang 5:

Basic - Programme zur Validierung

Die folgenden drei Basic - Programme dienen der Validierung des Modells gegenüber den in Kapitel 5 beschriebenen analytischen Lösungen. Das Programm KOND.BAS berechnet den rein konduktiven, eindimensionalen Wärmetransport mit Hilfe der komplementären Fehlerfunktion $\text{erfc}(x)$, die ihrerseits über die numerischen Approximation von ABRAMOWITZ & STEGUN (1964) ermittelt wird. Die analytische Lösung des eindimensionalen, rein konvektiven Wärmetransportes nach BODVARSSON (1982) kann mit dem Programm KONV.BAS ermittelt werden. Das Programm THEIS.BAS schließlich erlaubt die Berechnung der Brunnenabsenkung in einem gespannten Aquifer unter Einsatz der Theis-Funktion. Die Theis-Funktion wird hierbei über das polynomische Verfahren von HUNTOON (1980) approximiert. Alle drei Programme wurden in dem Basic - Dialekt der Fa. BORLAND (Turbo BASIC) verfaßt.

```
REM ----- Programm KOND.BAS -----
REM      analytische Lösung für den instationären, eindimen-
REM      sionalen, Wärmetransport in einem homogenen Medium

      DEFDBL A-H, O-Z
      DIM A(6), T(20), TP(20, 10), TSTEP(20), TZ(20)
      OPEN "ERDAT" FOR OUTPUT AS #1
      DATA 0.0705230784, 0.0422820123, 0.0092705272
      DATA 0.0001520143, 0.0002765672, 0.0000430638
      IP = 1
      IScrn = 9
      FOR I = 1 TO 6
        READ A(I)
      NEXT I
      WLKG = 3!           ' Wärmeleitfähigkeit des Gesteins in W/mK
      CPG = 850!         ' Wärmekapazität des Gesteins in J/kgK
      RHOG = 2600!       ' Dichte des Gesteins in kg/cbm
      T1 = 20!           ' Ausgangstemperatur in Grad Celsius
      T0 = 5!            ' Temperatur der obersten Schicht
      Z = 2!             ' Endtiefe in Metern
      DZ = .1            ' Schrittweite in Metern
      ZEIT = .25         ' gewünschter Anfangszeitpunkt in Tagen
      ALPHA = WLKG / CPG / RHOG
      ZEIT = ZEIT / 2! * 86400!
      ISTEP = 6
      FOR K = 1 TO ISTEP
        ZEIT = ZEIT * 2
        TSTEP(IP) = ZEIT / 86400!
```

```

N = INT(Z / DZ)
T(1) = T0
FOR I = 2 TO N
  TZ(I) = DZ * (I - 1)
  XX = TZ(I) / (2 * SQR(ALPHA * ZEIT))
  YY = 1!
  FOR J = 1 TO 6
    YY = YY + A(J) * XX ^ J
  NEXT J
  YY = 1! / YY ^ 16
  T(I) = YY * (T0 - T1) + T1
NEXT I
SCREEN IScrn
WINDOW (T1, Z)-(T0, 0!)
PSET (T0, 0)
TP(1, IP) = T0
FOR I = 2 TO N
  LINE -(T(I), (I - 1) * DZ)
  TP(I, IP) = T(I)
NEXT I
IP = IP + 1
NEXT K
DO
LOOP UNTIL INKEY$ <> ""
FORM$ = " ###.###"
SCREEN 0
CLS
LOCATE 1, 1
PRINT #1, SPACE$(10);
FOR I = 1 TO ISTEP
  PRINT #1, USING FORM$; TSTEP(I);
NEXT I
PRINT #1, : PRINT #1,
FOR K = 1 TO N
  PRINT #1, USING FORM$; TZ(K);
  FOR I = 1 TO ISTEP
    PRINT #1, USING FORM$; TP(K, I);
  NEXT I
  PRINT #1,
NEXT K
END

```

```

REM ----- Programm KONV.BAS -----
REM      analytische Lösung des eindimensionalen, konvektiven
REM      Wärmetransports, BODVARSSON(1982) p. 38
      CLS
      DEFDBL A-H, O-Z, L
      TI      = 200.
      T1      = 100.
      Q       = 0.001
      Rho     = 1.
      V       = 1.

```

```

TS      = 100.
FORM$ = " #### ##.####"
LOCATE 10,2
PRINT "Zeit      T2"
T = 0.
FOR I = 1 TO 10
  INCR T, TS
  QtRhoV = Q * T / Rho / V
  T2 = (T1 - T1) * EXP (-QtRhoV) + T1
  PRINT USING FORM$; T; T2
NEXT I
DO
LOOP UNTIL INSTAT
END

```

```

REM ----- Programm THEIS.BAS -----
REM      analytische Lösung mit Hilfe der THEIS - Brunnen -
REM      Funktion, vergl. WANG & ANDERSON (1982) pp. 79

```

```

DEFDBL D
CLS
Pi = ATN(1.) * 4
T = 300
Q = 2000
DT1 = 0.01
R = 100
S = 0.002
DT = DT1
LOCATE 3,20: PRINT " Tage   Absenkung"
FOR I = 1 TO 16
  Du = R * R * S / 4 / T / DT
  Wu = FNTheis# (Du)
  H = - Q / 4 / Pi / T * Wu
  LOCATE I + 4, 20
  PRINT USING "###.##    ##.###"; DT; H
  DT1 = DT1 * 1.5
  DT = DT + DT1
NEXT I
END

```

```

DEF FNTheis# (Du)
LOCAL D(), I, D1
REM      polynomisches Näherungsverfahren nach HUNTOON(1980)
DIM D(0:9)
D(0) = -0.57721566
D(1) = 0.99999193
D(2) = -0.24991055
D(3) = 0.05519968
D(4) = -0.00976004
D(5) = 0.00107857
D(6) = 0.250621
D(7) = 2.334733
D(8) = 1.681534

```

```

D(9) = 3.330657
IF Du < 0 THEN
  BEEP
  LOCATE 10,10
  PRINT "Achtung - U < 0!"
  END
END IF
IF Du > 0 AND Du < 1 THEN
  D1 = -LOG(Du) + D(0)
  FOR I = 1 TO 5
    D1 = D1 + D(I) * Du ^ I
  NEXT I
  FNTheis# = D1
  EXIT DEF
END IF
IF Du => 1 THEN
  D1 = D(6) + D(7) * Du + Du ^ 2
  FNTheis# = D1/((Du * EXP(Du))*(D(8) + D(9)* Du+Du ^ 2))
  EXIT DEF
END IF
END DEF

```

Anhang 6:

Ergebnisse der Validierungsläufe

Instationärer, eindimensionaler Wärmetransport

Ergebnis des Programms KOND.BAS:

	Tage					
Tiefe\Zeit	0.2500	0.5000	1.0000	2.0000	4.0000	8.0000
	0.0000	5.0000	5.0000	5.0000	5.0000	5.0000
	0.1000	9.8053	8.4457	7.4537	6.7412	5.8729
	0.2000	13.8670	11.6116	9.8053	8.4457	6.7412
	0.3000	16.7689	14.2845	11.9654	10.0792	7.6002
	0.4000	18.5213	16.3578	13.8670	11.6116	8.4457
	0.5000	19.4158	17.8356	15.4715	13.0191	9.2733
	0.6000	19.8016	18.8033	16.7689	14.2845	10.0792
M	0.7000	19.9423	19.3857	17.7745	15.3982	10.8597
e	0.8000	19.9857	19.7076	18.5213	16.3578	11.6116
t	0.9000	19.9970	19.8712	19.0530	17.1672	12.3322
e	1.0000	19.9995	19.9475	19.4158	17.8356	13.0191
r	1.1000	19.9999	19.9802	19.6530	18.3758	13.6703
	1.2000	20.0000	19.9931	19.8016	18.8033	14.2845
	1.3000	20.0000	19.9978	19.8909	19.1345	14.8606
	1.4000	20.0000	19.9993	19.9423	19.3857	15.3982
	1.5000	20.0000	19.9998	19.9707	19.5721	15.8972
	1.6000	20.0000	20.0000	19.9857	19.7076	16.3578
	1.7000	20.0000	20.0000	19.9933	19.8040	16.7809
	1.8000	20.0000	20.0000	19.9970	19.8712	17.1672

Ergebnis des Programms TRADIKON-3D:

		Tage					
Tiefe\Zeit	0.2500	0.5000	1.0000	2.0000	4.0000	8.0000	
	0.0000	5.0000	5.0000	5.0000	5.0000	5.0000	
	0.1000	10.1200	8.5600	7.5000	6.7600	5.8700	
	0.2000	14.1900	11.7900	9.8800	8.4700	6.7400	
	0.3000	16.9000	14.4500	12.0600	10.1200	7.6100	
	0.4000	18.4700	16.4500	13.9500	11.6600	8.4500	
	0.5000	19.2900	17.8500	15.5400	13.0700	9.2800	
	0.6000	19.6900	18.7600	16.8100	14.3300	10.0900	
M	0.7000	19.8700	19.3200	17.7900	15.4300	12.9800	
e	0.8000	19.9500	19.6400	18.5100	16.3900	13.8900	
t	0.9000	19.9800	19.8200	19.0300	17.1800	14.7300	
e	1.0000	19.9900	19.9100	19.3800	17.8400	15.4900	
r	1.1000	20.0000	19.9600	19.6200	18.3700	16.1700	
	1.2000	20.0000	19.9800	19.7700	18.7900	16.7800	
	1.3000	20.0000	19.9900	19.8600	19.1200	17.3100	
	1.4000	20.0000	20.0000	19.9200	19.3700	17.7800	
	1.5000	20.0000	20.0000	19.9600	19.5500	18.1800	
	1.6000	20.0000	20.0000	19.9800	19.6900	18.5200	
	1.7000	20.0000	20.0000	19.9900	19.7900	18.8000	
	1.8000	20.0000	20.0000	19.9900	19.8600	19.0400	

Anhang 7:

Programmübersicht ähnlicher Modelle

Modelle zum Anlagendesign

HELLSTRÖM, G. (1982)

Modellname: DST
Modelltyp: Finite Differenzen in Verbindung mit der Superposition Technik
Art der Diskretisierung: zweidimensional
Einsatzzweck: saisonale Wärmespeicherung über Rohrleitungssystemen im Erdreich
Referenz: /83/

MEI, V. C. & FISCHER, S. K. (1984)

Modellname: TRANS
Modelltyp: Finite Differenzen
Art der Diskretisierung: zweidimensional, radial
Einsatzzweck: Berechnung der Wärmeübergänge in einem koaxialen Wärmetauscher unter Berücksichtigung des konduktiven Wärmetransports im Erdreich
Referenz: /134/

GILBY, D. J. & HOPKIRK, R. J. (1985)

Modellname: MC.TRAD-2D
Modelltyp: Finite Differenzen
Art der Diskretisierung: zweidimensional, kartesisch, zylindrisch, polar
Einsatzzweck: Berechnung des konduktiven Wärmetransports bzw. des diffundiven Massentransports
Besonderheit: Achsen der Anisotropie der Wärmeleitfähigkeit können von den Achsen des kartesischen Koordinatensystems abweichen.
Referenz: /73/

ESKILSON, P. (1986)

Modellname:	SBMB
Modelltyp:	Superposition Technik
Art der Diskretisierung:	dreidimensional, variabel
Einsatzzweck:	Berechnung des konduktiven Wärmetransports im Nahfeld von Erdwärmetauschern mit unterschiedlicher Anordnung und Neigung zu einander
Referenz:	/60/

ESKILSON, P. (1987)

PC-Programme für die Dimensionierung von Wärmeentzugsbohrungen

TFSING	Berechnung der höchsten und niedrigsten Temperatur der Wärmetauscherflüssigkeit in einem einzelnen Bohrloch. Die Wärmeentzugsmenge kann vorgegeben werden.
TFMULT TFSTEP	wie TFSING, jedoch mit mehreren Bohrlöchern Berechnung der Tauscherflüssigkeitstemperatur einer einzelnen oder mehrerer Bohrungen zu einem willkürlichen Zeitpunkt. Die Entzugsleistung kann in 12 Schritten variiert werden.
DIM	Berechnet iterativ die benötigte Bohrlochlänge für ein Ein- oder Mehrfachbohrlochsystem. Die vorgegebene Entzugsleistung kann in 12 Schritten variiert und zyklisch wiederholt werden.
INOUT	Berechnung der Vor- und Rücklauf Temperatur der Wärmeträgerflüssigkeit bei vorgegebener, mittlerer Tauschertemperatur (kann durch TFSING und TFMULT berechnet werden).
QSURF	Berechnet den Wärmefluß durch die Erdoberfläche im Radius r vom Zentrum einer freistehenden Bohrung.
QTOTSURF	Berechnet den gesamten Wärmefluß durch die Erdoberfläche für eine einzelne Bohrung
GRWATER	Berechnet die stationäre Temperatur an einer Bohrlochwand unter Berücksichtigung des strömenden Grundwassers.
Referenz:	/62/

Modelle zum Wärmetransport in porösen Aquiferen

WILLHITE, G. P. & WAGNER, J. (1974)

Modellname:	KANSASHEAT
Modelltyp:	Finite Differenzen
Art der Diskretisierung:	dreidimensional, kartesisch
Einsatzzweck:	Strömungsberechnung, Temperaturberechnung, Energie- und Massenbilanz
Aquifertyp:	anisotrop, heterogen, halbgespannt, mehrlagig
Leistungsumfang:	
a) Strömung	dreidimensional, instationär
b) Wärmetransport:	Konvektion, Konduktion; unabhängig vom Strömungsmodell
Referenz:	/215/, /216/

INTERCOMP (1976)

Modellname:	SWIP
Modelltyp:	Finite Differenzen
Art der Diskretisierung:	dreidimensional, kartesisch oder radial
Einsatzzweck:	Massen- und Energietransport in tiefen salinen Aquiferen
Aquifertyp:	anisotrop, heterogen, gespannt, mehrlagig
Leistungsumfang:	
a) Strömung	instationär, temperatur-, druck- und konzentrationsabhängige Änderung der Fluidviskosität
b) Wärmetransport:	Konvektion, Konduktion, Dispersion verknüpft mit der Strömungsberechnung
c) Stofftransport:	Konvektion, Diffusion, Dispersion verknüpft mit der Strömungsberechnung
Referenz:	/98/

FAUST, C. R. & MERCER, J. W. (1979)

Modellname:	GEOETHER
Modelltyp:	Finite Differenzen
Art der Diskretisierung:	multidimensional
Einsatzzweck:	Druck- und Temperaturberechnung
Aquifertyp:	anisotrop, heterogen, gespannt und/ oder halbgespannt, mehrlagig
Leistungsumfang:	
a) Strömung	stationär/instationär, kompressibles Fluid, Zweiphasenmischung (Wasserdampf und Wasser)
b) Wärmetransport:	Konvektion, Konduktion, Kondensation, Dispersion verknüpft mit der Strömungsberechnung
Referenz:	/65/

INTERA (1979)

Modellname:	SWAP
Modelltyp:	Finite Differenzen
Art der Diskretisierung:	dreidimensional, kartesisch oder radial
Einsatzzweck:	Massen- und Energietransport in tiefen salinen Aquiferen
Aquifertyp:	anisotrop, heterogen, gespannt/ ungespannt, mehrlagig
Leistungsumfang:	
a) Strömung	instationär, temperatur-, druckabhängige Änderung der Fluidviskosität, temperatur-, konzentrations- und druckabhängige Änderung der Fluidichte
b) Wärmetransport:	Konvektion, Konduktion, Dispersion; verknüpft mit der Strömungsberechnung
c) Stofftransport:	Konvektion, Diffusion, Dispersion, Adsorption, Zerfall; verknüpft mit der Strömungsberechnung; lineare Adsorptionsisotherme als Funktion des Matrixtypes
Bemerkung:	Erweiterung und Verbesserung des INTERCOMP - Modells
Referenz:	/97/

BODVARSSON G.S. (1982)

Modellname:	PT
Modelltyp:	Finite Differenzen
Art der Diskretisierung:	dreidimensional, kartesisch, radial, sphärisch oder völlig asymmetrisch
Einsatzzweck:	Energie und Massentransport in heterogenen, porösen oder geklüfteten Medien
Aquifertyp:	anisotrop, gespannt/ ungespannt
Leistungsumfang:	
a) Strömung	instationär, temperatur-, druckabhängige Änderung der Fluidviskosität, temperatur- und druckabhängige Änderung der Fluidichte
b) Wärmetransport:	Konvektion, Konduktion; verknüpft mit der Strömungsberechnung
Sonstige Merkmale:	zusätzlich eindimensionale Kompaktion nach der Theorie von TERZAGHI (1925)
Bemerkung:	Weiterentwicklung des Modells CCC von LIPPMANN et al. (1977)
Referenz:	/24/

ALGAN, U. (1984)

Modellname:	TESPA
Modelltyp:	Finite Differenzen
Art der Diskretisierung:	zweidimensional, kartesisch
Einsatzzweck:	Energietransport in ungespannten Aquiferen
Aquifertyp:	anisotrop, gespannt/-ungespannt
Leistungsumfang:	
a) Strömung	instationär, temperatur-, druckabhängige Änderung der Fluidviskosität, temperatur- und druckabhängige Änderung der Fluidichte
b) Wärmetransport:	Konvektion, Konduktion; verknüpft mit der Strömungsberechnung
Referenz:	/2/

VOSS, C. I. (1984)

Modellname:	SUTRA
Modelltyp:	Finite Elemente
Art der Diskretisierung:	zweidimensional, pseudodreidimensional, kartesisch, radial
Einsatzzweck:	gesättigt/ungesättigter Stoff- und Energietransport in gespannten/ ungespannten Aquiferen
Aquifertyp:	anisotrop, gespannt/ungespannt
Leistungsumfang:	
a) Strömung	instationär, temperatur-, druckabhängige Änderung der Fluidviskosität, temperatur- und druckabhängige Änderung der Fluidichte
b) Wärmetransport:	Konvektion, Konduktion; verknüpft mit der Strömungsberechnung
Referenz:	/209/

REED, J. E. (1985)

Modellname:	HOTWTR
Modelltyp:	Finite Differenzen
Art der Diskretisierung:	dreidimensional, kartesisch
Einsatzzweck:	stationärer Grundwasser- und Wärmefluß in einem porösen Medium
Aquifertyp:	isotrop, heterogen, gespannt/ungespannt
Leistungsumfang:	
a) Strömung	stationär, temperaturabhängige Änderung der Fluidviskosität, konstante Fluidichte
b) Wärmetransport:	Konvektion, Konduktion; verknüpft mit der Strömungsberechnung
Referenz:	/166/

KANGAS, M. T. & LUND, P. D. (1987)

Modellname:	THETA
Modelltyp:	Finite Differenzen
Art der Diskretisierung:	dreidimensional, kartesisch
Einsatzzweck:	instationärer Grundwasser- und Wärmefluß in einem porösen Medium
Aquifertyp:	anisotrop, heterogen, gespannt/ungespannt
Leistungsumfang:	
a) Strömung	instationär, temperaturabhängige Änderung der Fluidviskosität und Fluidichte
b) Wärmetransport:	Konvektion, Konduktion; verknüpft mit der Strömungsberechnung
Referenz:	/100/

Anhang 8:

Diagramme der Schwalbacher Kalibrierungsläufe

Die folgenden Diagramme präsentieren die Ergebnisse der Schwalbacher Kalibrierungsläufe, die zugleich eine Parameterstudie über die in die Berechnung einfließenden, physikalischen Größen darstellen. Die berechnete Regeneration erfolgte erwartungsgemäß bei einer Wärmeleitfähigkeit von $1.5W/mK$ am langsamsten und einer Fließgeschwindigkeit von $5 \cdot 10^{-6} m/s$ ¹ am schnellsten. Bei allen Simulationsläufen mit Konvektion wurde eine konstante, auf den Modellostern (vergl. Kap. 3.3) gerichtete Grundwasserfließrichtung angenommen.

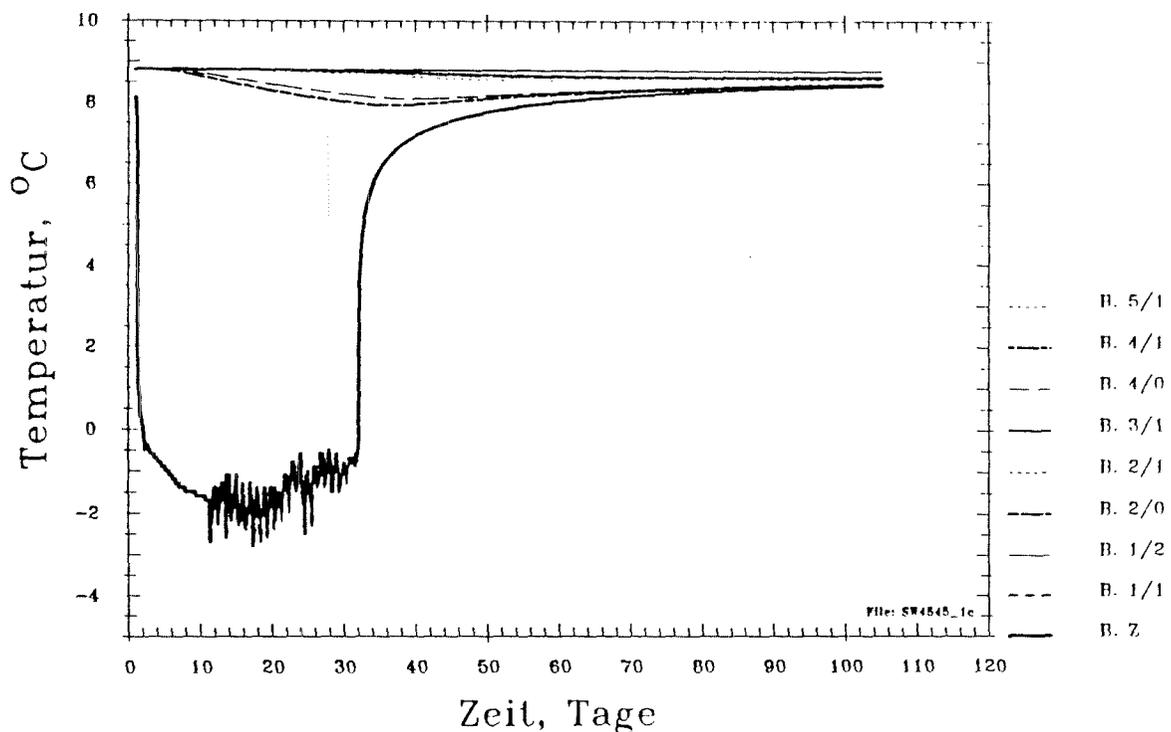


Abb. A.8.1: Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 1, Tab. 5)

¹ entspricht bei einer durchflußwirksamen Porosität von 3 % einer Abstandsgeschwindigkeit von $14.4 m/d$

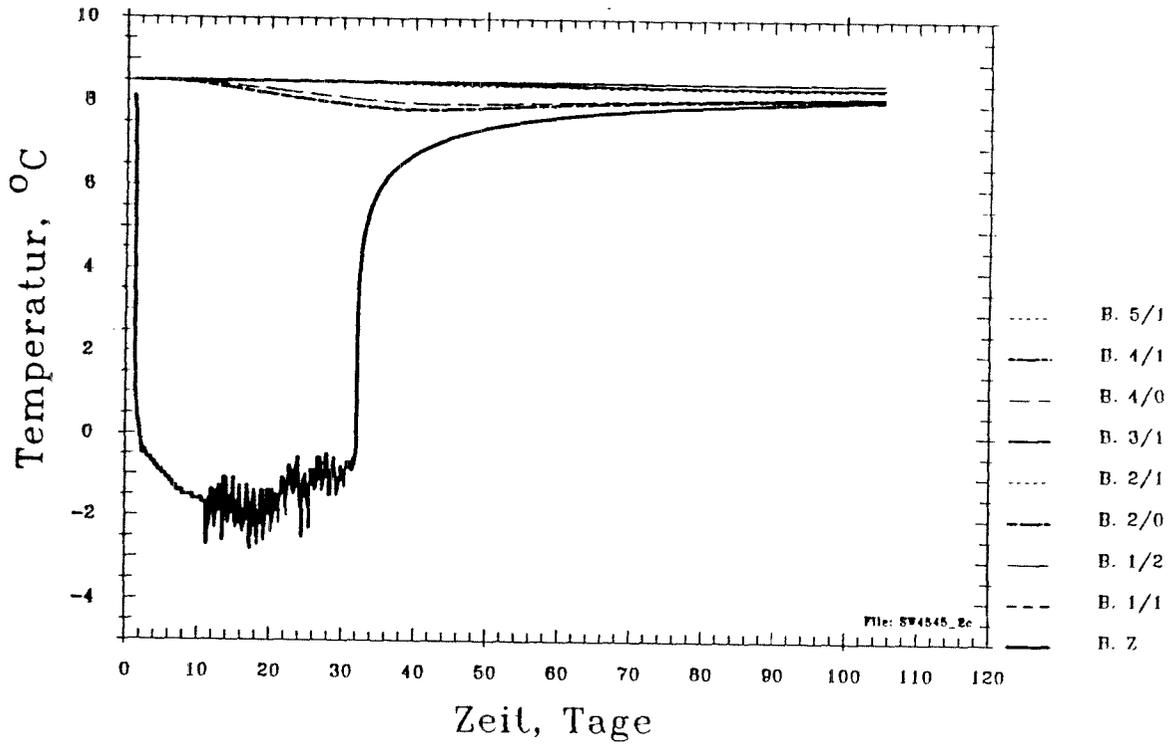


Abb. A.8.2: Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 2, Tab. 5)

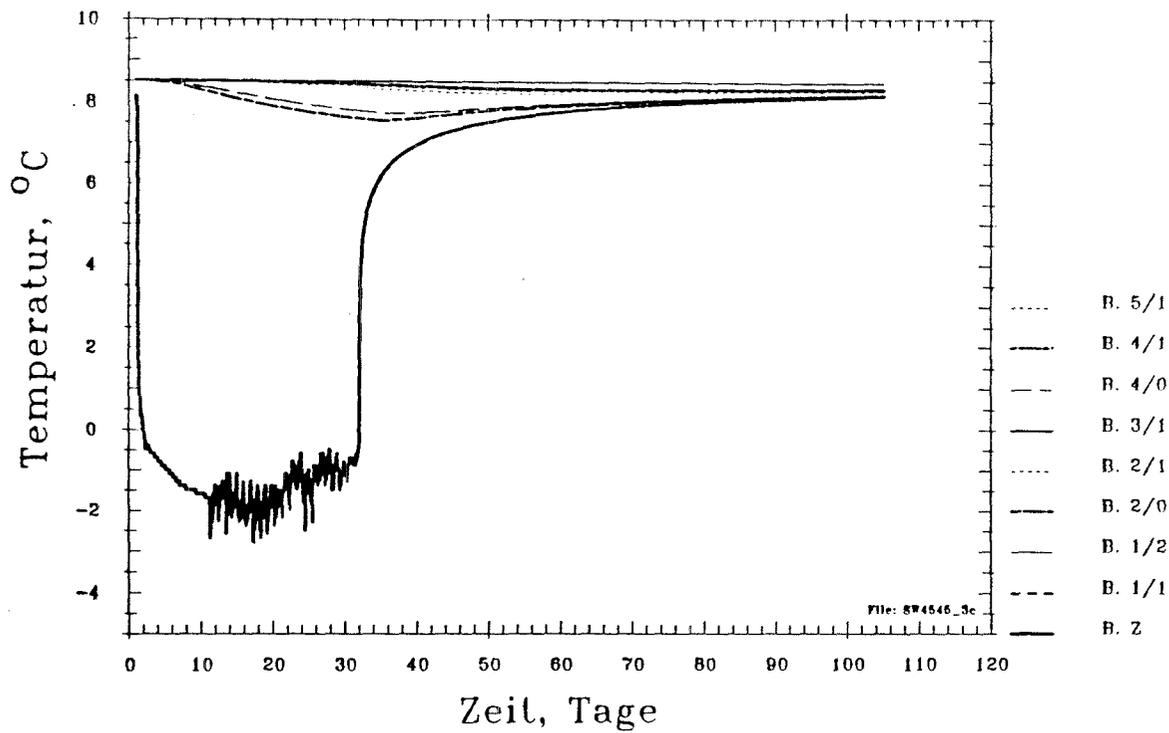


Abb. A.8.3: Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 3, Tab. 5)

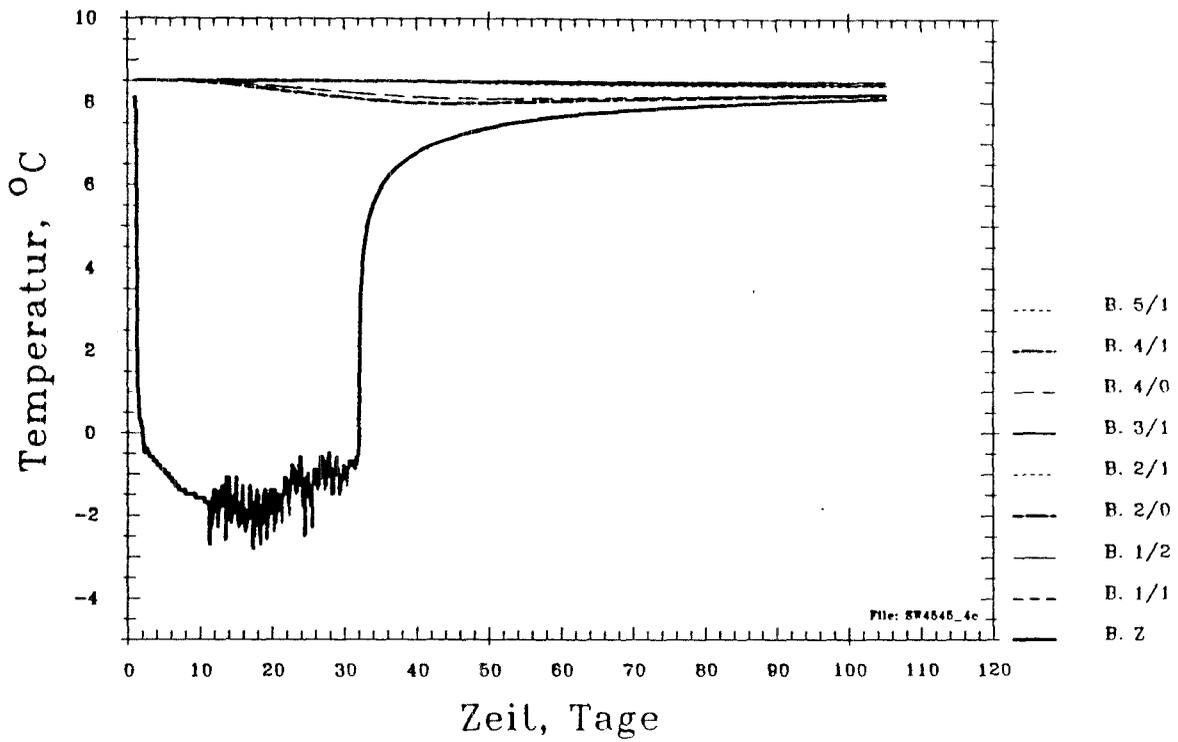


Abb. A.8.4: Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 4, Tab. 5)

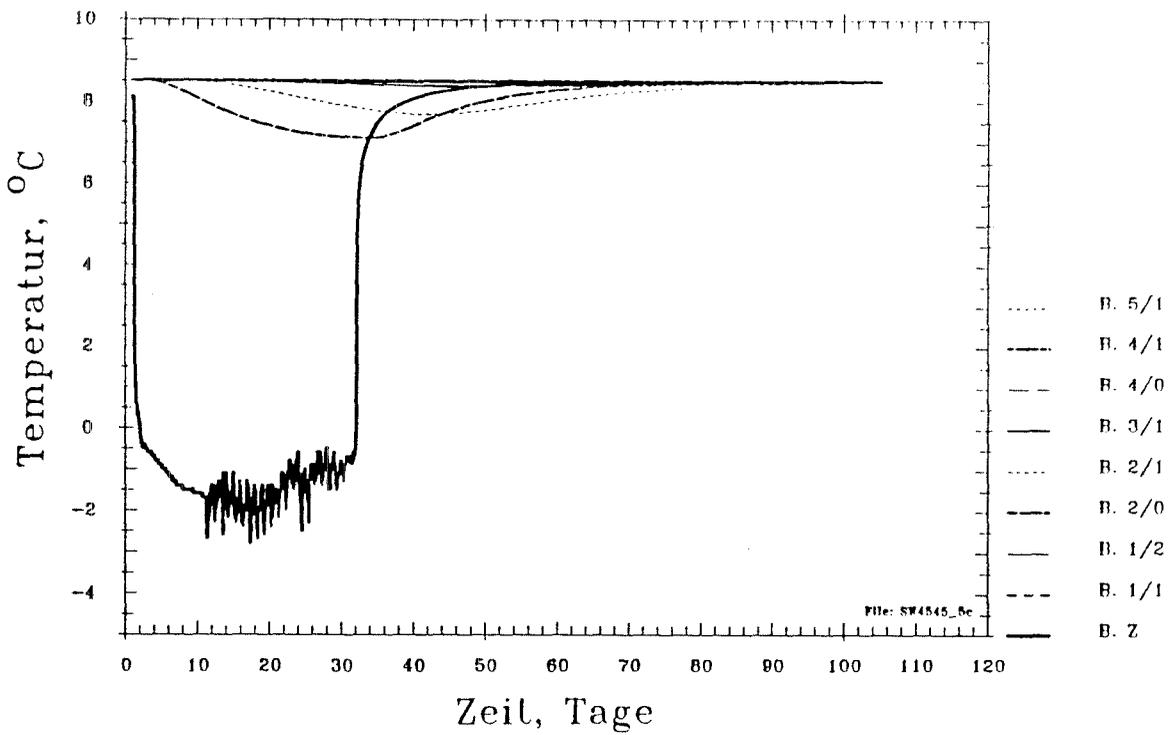


Abb. A.8.5: Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 5, Tab. 5)

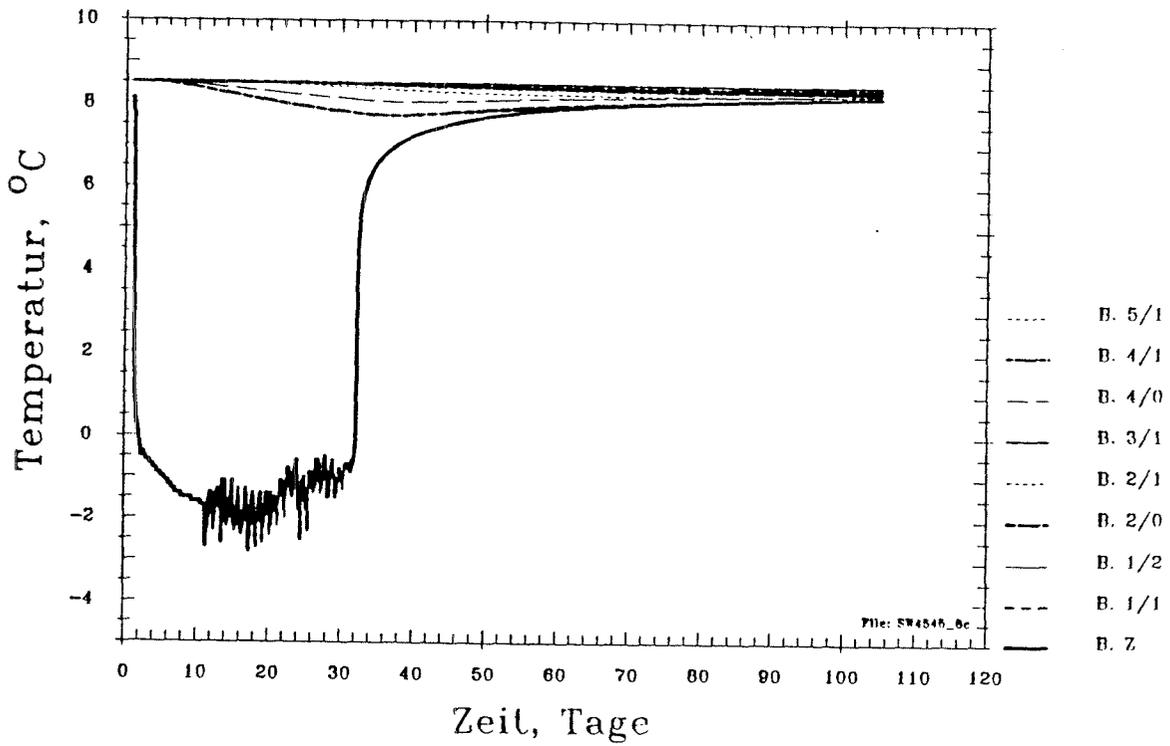


Abb. A.8.6: Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 6, Tab. 5)

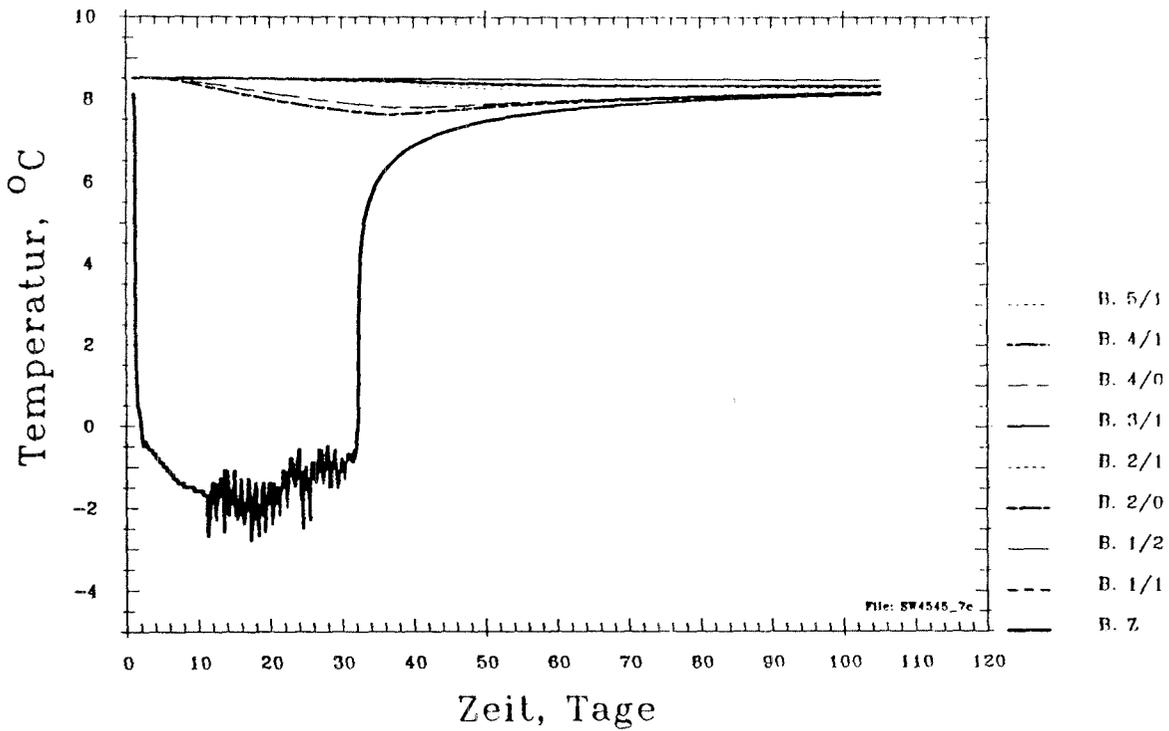


Abb. A.8.7: Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 7, Tab. 5)

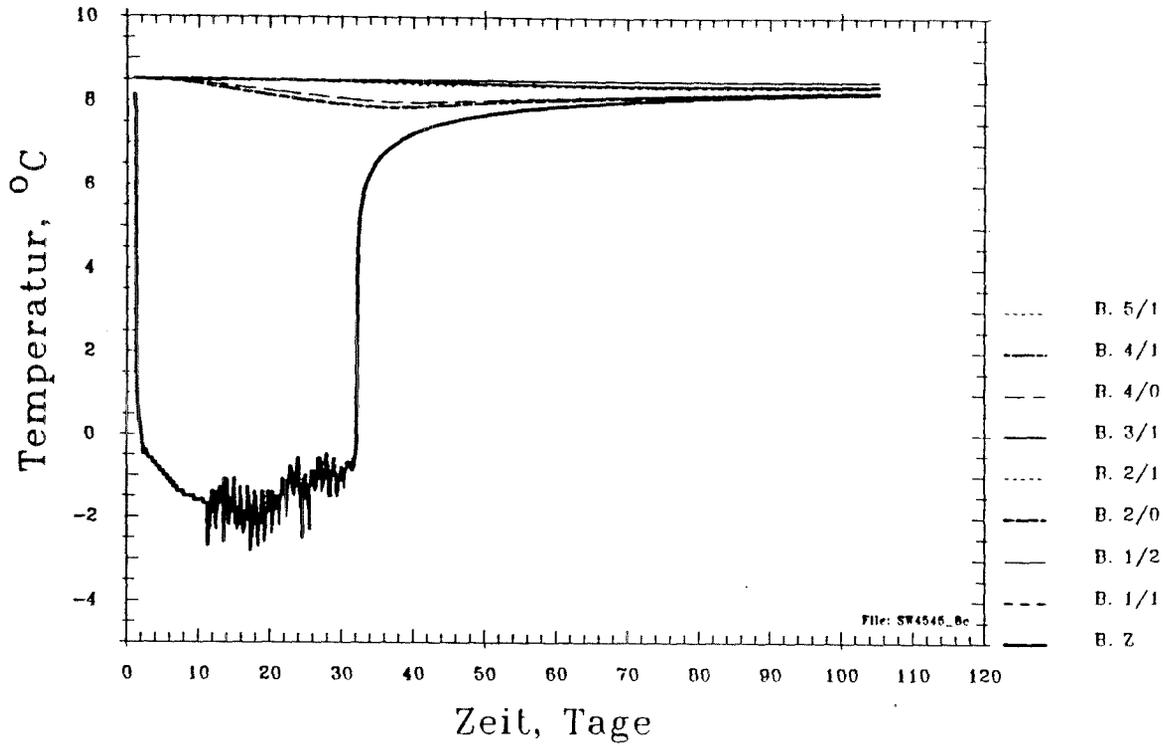


Abb. A.8.8: Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 8, Tab. 5)

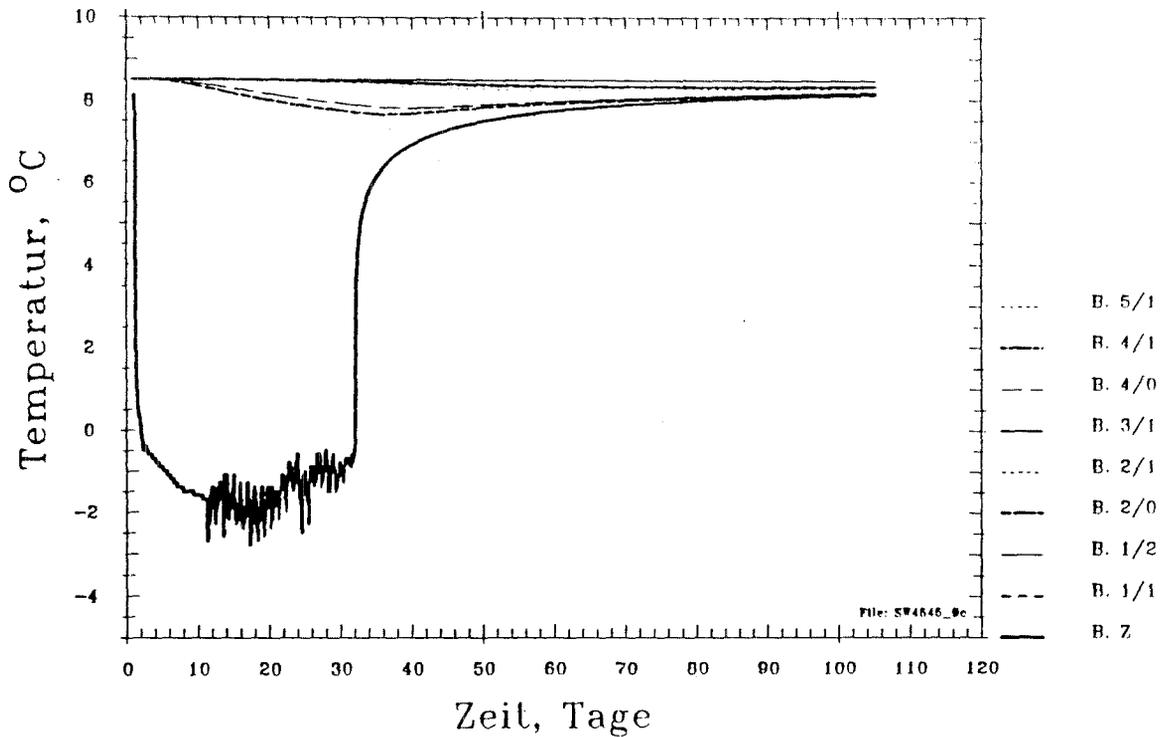


Abb. A.8.9: Berechnete Regeneration nach 31-tägigem Wärmeentzug (Datensatz 9, Tab. 5)

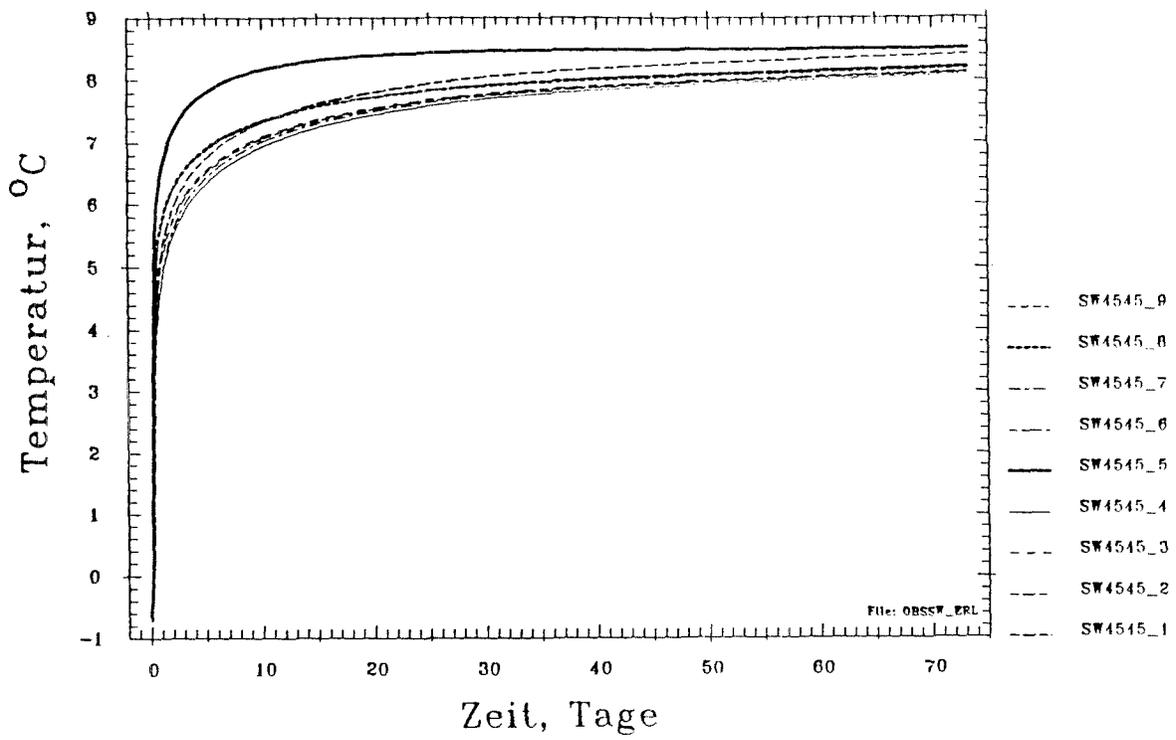


Abb. A.8.10: Berechnete Regeneration in Bohrung Z nach 31-tägigem Wärmeentzug (alle Datensätze)