

Forschungsprogramm
Umgebungs- und Abwärme,
Wärme-Kraft-Kopplung (UAW)

Erweiterung des Programms EWS **für Erdwärmesondfelder**

ausgearbeitet durch

Arthur Huber

Huber Energietechnik,
Ingenieur- und Planungsbüro SIA
Jupiterstrasse 26, 8032 Zürich
huber@igjzh.com

Daniel Pahud

Ecole Polytechnique Fédérale de Lausanne
EPFL – DGC - LASEN
CH – 1015 Lausanne
pahud@dct.supsi.ch

im Auftrag des
Bundesamtes für Energie

Zusammenfassung

Ziel der vorliegenden Arbeit ist es, das validierte Simulationsprogramm EWS für Erdwärmesonden (Huber, Schuler, 1997) so zu ergänzen, dass damit auch Mehrsondenanlagen und Erdwärmesondenfelder über ihre ganze Lebensdauer berechnet werden können. Ausserdem soll eine Schnittstelle zum neuen Programm SwEWS (Leu et al., 1999) erstellt werden, mit dem die geothermischen Eigenschaften im Schweizer Molassebecken direkt eingelesen und für die Simulation verwendet werden können.

Das gestellte Anforderungsprofil wird mit einer dynamischen Simulation der Sole und des Erdreiches im Sonden-Nahbereich (1.5 m - 2 m) mit dem Crank-Nicholson-Verfahren gelöst. Die äusseren Temperatur-Randbedingungen werden mit dimensionslosen Temperatur-Sprungantworten („g-functions“) nach der Theorie von Eskilson (1987) bestimmt. Damit lassen sich die mittleren Temperaturen von Einzelsonden und Erdwärmesondenfeldern auch für beliebig lange Simulationszeiten berechnen. Das Problem des nicht konstanten Wärmeentzugs und der Regeneration der Erde kann elegant durch Superposition von beliebig vielen, zu unterschiedlichen Zeitpunkten beginnenden, konstanten Wärmeentzügen gelöst werden (Faltung).

Mit dem vorliegenden Ansatz können Erdwärmesonden (Doppel-U-Sonden) und Sondenfelder über einen Zeitraum von über 60 Jahren korrekt berechnet werden. Das Wärme-Entzugsprofil kann dabei frei gewählt werden. Es können bis zu 10 verschiedene, vertikale Erdschichten mit unterschiedlichen Stoffparametern dargestellt werden. Das Modell ist geeignet auch das transiente Verhalten der Erdwärmesonde nachzubilden. Der Zeitschritt für die Simulation ist im Bereich zwischen 1 und 60 Minuten wählbar. Die für die Simulation von 30 Jahren nötige Rechenzeit (Pentium Prozessor 130 MHz) liegt im Bereich von 7 Minuten (bei einer vertikalen Schicht).

Zur Validierung der neuen Programmteile wurden 20 Testfälle definiert, die mit dem Programm EWS berechnet und mit dem Programm SBM / TRNSBM (Pahud, Fromentin, Hadorn, 1996) verglichen wurden. Mit TRNSBM können Erdwärmesondenfelder direkt (ohne g-functions) berechnet werden. Das Erdreich ist aber immer homogen und das transiente Verhalten der Erdwärmesonde kann nur mit einem Trick angenähert werden. Die Rechenzeit bei TRNSBM ist mit 2 Erdwärmesonden 4-5 mal länger als bei EWS, mit 50 Erdwärmesonden ist ca. 60 mal mehr Rechenzeit notwendig. Zur Kontrolle wurden ausserdem einige Fälle zusätzlich mit dem Programm EED (Hellström, Sanner, 1994) überprüft.

Die für die EWS-Berechnung nötigen g-functions wurden aus der Publikation von Eskilson (1987) übernommen. Generell kann in all jenen Fällen, bei denen die g-functions direkt übernommen werden konnten, eine sehr gute Übereinstimmung (vergl. Abschnitt 3.3) über die ganze Vergleichsperiode von 60 Jahren festgestellt werden. Etwas grössere Abweichungen treten auf, falls die im Programm EWS implementierte Extrapolation der g-functions verwendet werden. Die Einsatzgrenzen der Extrapolation liegen bei einer Halbierung des Sondenabstandes. Da zur Zeit noch keine g-functions für so kleine Sondenabstände publiziert sind, wie dies bei der saisonalen Wärmespeicherung üblich ist, kann das Programm EWS momentan für solche Fälle noch nicht eingesetzt werden.

Ein leichtes Überschwingen kann ausserdem in einigen Fällen zum Zeitpunkt $\ln(t/t_s) = 4$ beobachtet werden. Der Grund dafür liegt im nicht kontinuierlichen Übergang vom linearen Ansatz auf den Polynomansatz für die g-functions zu diesem Zeitpunkt. Der Fehler ist aber in keinem Fall additiv, d.h. bei einer Fortsetzung der Simulation über diesen Zeitpunkt hinaus verschwindet er wieder. Ganz generell kann festgestellt werden, dass die Abweichungen zwischen den Programmen die Grössenordnung von einigen Prozent kaum je übersteigen.

Im Kapitel 4 sind alle Simulationsparameter und die Ein- und Ausgabefiles im Sinne eines Programm - Manuals beschrieben. Das Programm ist in PASCAL geschrieben und im Anhang

Diese Arbeit ist im Auftrag des Bundesamtes für Energie entstanden. Für den Inhalt und die Schlussfolgerungen sind ausschliesslich die Autoren dieses Berichtes verantwortlich.

vollständig abgedruckt. Das EWS-Modul und ein Testprogramm dazu kann bei der Firma Huber Energietechnik, zum Selbstkostenpreis für Kopieren und Versand bezogen werden.

Résumé

Le programme de simulation EWS a été validé pour le calcul de sondes géothermiques uniques (Huber, Schuler, 1997). Le but de ce travail est d'étendre la capacité de calcul du programme EWS pour des champs de sondes géothermiques arbitraires sur toute leur durée de vie. En outre une connexion avec le nouveau programme SwEWS (Leu et al., 1999) sera établie, de manière à pouvoir importer les propriétés thermiques de la couverture de molasse suisse et les utiliser directement pour les simulations.

Le problème thermique est résolu avec une simulation dynamique du fluide caloporteur et du terrain dans le voisinage de la sonde (1.5 m - 2 m) avec la technique de Crank-Nicholson. Les conditions de température extérieures sont fixées, à choix, par la solution de Werner (1996) ou par la réponse de température sans dimension ("fonction-g") définie par la théorie d'Eskilson (1987). De cette manière les températures moyennes de sondes uniques ou de champ de sondes peuvent être calculées pour des durées de simulation aussi longues que désiré. Le problème de l'extraction de chaleur non constante et de la régénération du terrain est résolu par la technique de superposition.

Avec les solutions retenues le calcul de sondes géothermiques et de champs de sondes peut être correctement effectué sur des périodes dépassant 60 ans. Le profil d'extraction de chaleur peut être choisi librement. Jusqu'à 10 couches de terrain superposées peuvent être définies avec des propriétés thermiques différentes. Le pas de temps utilisé pour la simulation peut être choisi entre 1 et 60 minutes. La simulation de 30 ans nécessite un temps de calcul d'environ 7 minutes pour une couche de terrain (processeur Pentium 130 MHz)

Pour la validation de la nouvelle partie du programme, une vingtaine de cas sont définis et simulés. Les calculs obtenus avec EWS sont comparés à TRNSBM (Pahud et al., 1996), qui n'est autre que SBM intégré dans TRNSYS (Klein et al., 1996), un programme de simulation dynamique de systèmes thermiques. Le temps de calcul est 4 à 5 fois plus grand qu'avec EWS pour 2 sondes géothermiques, et plus de 60 fois plus grand avec 50 sondes. La comparaison avec EED (Hellström, Sanner, 1994) n'est possible que lorsque la puissance thermique est donnée en entrée au programme. Toutefois, une comparaison TRNSBM - EED permet de vérifier que l'utilisation de TRNSBM correspond bien aux conditions fixées pour le calcul des fonction-g relatives à une configuration de sondes géothermiques choisie.

Dans l'ensemble, les comparaisons des résultats de calcul effectués avec EWS et TRNSBM sont très satisfaisantes. Les différences observées sont principalement dues aux valeurs des fonction-g utilisées dans le programme EWS, dont les valeurs ne sont pas exactement celles que l'on obtiendrait avec TRNSBM. (Dans le programme EWS, les fonction-g sont prises des publications d'Eskilson (1987)). Ces différences-là ne sont pas importantes, dans la mesure où les erreurs induites ne dépassent pas quelques pour-cent.

Des différences significatives sont observées lorsque les fonction-g sont extrapolées en-dehors du domaine des fonction-g pré-calculées. Plus précisément, lorsque le rapport B/H est trop petit ou lorsque le temps est trop petit. L'extrapolation des fonction-g devrait toujours se faire à l'intérieur du domaine des fonction-g pré-calculées. L'erreur observée pour les petit temps (lorsque $\ln(t/t_s)=4$) n'est pas additive. Elle disparaît dès que le temps devient plus grand. Dans l'état actuel du programme, le rapport B/H ne devrait pas être inférieur au plus petit rapport utilisé pour les fonction-g pré-définies. Le programme est donc adéquat pour le calcul de systèmes avec extraction de chaleur. En revanche, le calcul de stockage de chaleur n'est pas recommandé, puisque dans ces situations, l'espacement entre les sondes est généralement réduit (trop petite valeur de B/H). Un message d'avertissement devrait être montré à l'utilisateur pour l'avertir quand il désire simuler un cas qui demande l'extrapolation

des fonction-g en dehors de leur domaine de définition. A l'avenir, ce problème peut être résolu en recalculant les fonction-g de manière à couvrir tout le domaine de B/H et de $\ln(t/t_s)$. Il n'y aurait alors plus de problèmes liés à l'extrapolation des fonction-g.

Abstract

The principal aim of this project is to complete the simulation program EWS for vertical borehole heat exchangers (Huber, Schuler, 1997) in a way, that it is enabled to calculate the whole lifetime of fields of boreholes. Additionally, a data exchange to the new program SwEWS (Leu et al., 1999) should allow the use of geothermal properties for typical rocks of the Swiss Molasse Basin.

The given demands are solved with a dynamic simulation of the earth in a radius of about 2 m around the borehole with the Crank-Nicholson algorithm. For the outer boundary conditions, the analytical formula of Werner (1996) or the dimensionless temperature-response functions (g-functions) of Eskilson (1987) can be used. By this, the medium temperatures of single vertical boreholes and fields of boreholes can be calculated for any given period of simulation time. By superposing constant heat extractions, starting at different time steps, it is possible to calculate the temperature profile at the outer boundaries of the simulation area and even to predict the refilling of the temperature sink in the summer properly.

For the validation of the new program EWS, 20 test cases have been defined. The simulation results of these test cases were compared with the results of the program SBM / TRNSBM (Pahud, Fromentin, Hadorn, 1996). As a control, some of the test cases were additionally recalculated with the program EED (Hellström, Sanner, 1994). The needed g-functions in the program EWS were taken out of a report of Eskilson (1987). Generally, in all cases calculated with g-functions directly taken out of Eskilsons report, a very good correspondence was found (cf. chapter 3). Some differences could be observed, when the extrapolation of the g-functions was used that is implemented in the program EWS. This extrapolation is limited to 50% of the borehole distance of the input g-function. Up to now, only g-functions for rather large borehole distances are published. Therefore it is not yet possible to calculate thermal heat storage systems, since they normally have smaller borehole distances.

EWS needs about 7 minutes for the simulation of 30 years (Pentium 130 MHz), the simulation time of TRNSBM is about 4 to 60 times longer, depending on the number of boreholes.

In some other cases there was also a difference at the time $\ln(t/t_s) = 4$. This is due to the incontinuous change from the linear approach to the polynomial approach of the g-functions at this time. The deviation is in no case additional, that means, it disappears while continuing the simulation.

Generally, the differences between the programs are within a range of a few per cent. According to the present state of the program, the calculations are satisfactory when the spacing between the boreholes is large enough (no extrapolation of the g-function). It is thus recommended for heat extraction boreholes. However, duct storage simulation is not recommended as the spacing is normally smaller. A warning message should be issued when the user would like to simulate a case which requires the extrapolation of the g-function outside of its domain of definition. In the future, this problem could be solved by recalculating the g-function for the whole domain of B/H and $\ln(t/t_s)$ values.

Verdankungen

Diese Arbeit wurde im Rahmen des Forschungsprogramms Umgebungs- und Abwärme, Wärme-Kraft-Kopplung (UAW) des Bundesamtes für Energie erstellt. Wir danken Herrn Dr. M.

Zogg, Programmleiter UAW, für seine konstruktiven Hinweise und das Zustandekommen dieser Arbeit. Weiter danken wir Herrn Ernst Rohner, Grundag AG und Herrn Thomas Mégel für Ihre Unterstützung und Mithilfe bei der Erstellung der Schnittstelle zum Programm SwEWS und Herrn Dr. A. Fromentin für die Begleitung des Projektes.

Inhaltsverzeichnis

1	ZIELSETZUNG DER ARBEIT	1
1.1	Einsatzgebiete von Erdwärmesonden	1
1.2	Anforderungen an ein Auslegungsprogramm	3
1.3	Lösungsansatz	4
2	MODELLBESCHRIEB	7
2.1	Simulation des Sonden-Nahbereichs	7
2.1.1	Crank-Nicholson-Ansatz für den Sonden-Nahbereich	7
2.1.2	Lösungsalgorithmus	7
2.1.3	Rechengitter	8
2.1.4	Wärmekapazitäten C	9
2.1.5	Thermische Widerstände R und Leitfähigkeit L	9
2.2	Wärmeübergangskoeffizient Sonde-Sole	11
2.2.1	Wärmeübergangskoeffizient α_1 beim Betrieb der Erdwärmesonde	11
2.2.2	Wärmeübergangskoeffizient α_0 beim Stillstand der Erdwärmesonde	12
2.3	Modellierung der Sole	13
2.4	Äussere Randbedingung mit den „g-functions“	14
2.4.1	Dimensionslose Sprungantwort „g-function“ von Eskilson	14
2.4.2	Superpositionsprinzip	16
2.4.3	Polynom - Ansatz für die g-function	17
2.4.4	Extrapolation der g-functions	18
2.4.5	Verwendete g-functions	19
2.5	Abschätzung der Starttemperaturen	20
2.6	Auslegungs - Lastprofil	21
2.7	Stoffwerte des Erdreichs und Bodentemperaturen	22
3	VALIDATION	24
3.1	Méthode de validation	24
3.2	Définition des cas de validation	25
3.2.1	Définition des paramètres de base	25
3.2.2	Variation des paramètres:	26
3.3	Résultats des validations	27
3.4	Conclusion	50
4	PROGRAMMBESCHRIEB	51
4.1	Flussdiagramme	51
4.1.1	Modulaufruf aus dem Hauptprogramm	51
4.1.2	Simulation des Zeitschrittes	52
4.2	Parameter im EWS-Modul	53
4.2.1	Eingabeparameter	53
4.2.2	Ausgabewerte	62
4.2.3	Parametereingabe im Hauptprogramm und Eingabefile	63
5	SYMBOLVERZEICHNIS	67
6	LITERATURVERZEICHNIS	69
7	ANHANG: PASCAL - PROGRAMM	71

1 Zielsetzung der Arbeit

1.1 Einsatzgebiete von Erdwärmesonden

Erdwärmesonden werden in Einzelsonden oder in Sondenfeldern eingesetzt. Die daraus gewonnene Erdwärme kann direkt (Luftvorwärmung) oder mit Hilfe einer Wärmepumpe genutzt werden. Erdwärmesonden können aber auch zur Kühlung verwendet werden (direkt zur Luft- und Strukturkühlung oder als Rückkühlung für eine Wärmepumpe). Auch Kombinationen von Kühl- und Heizanwendungen sind immer häufiger anzutreffen und bieten den Vorteil einer aktiven, thermischen Regeneration des Erdreiches.

Die bis anhin häufigste Einsatzart ist die Kombination von Heizungs-Wärmepumpen mit Erdwärmesonden als Wärmequelle (Abb 1-1). Bei diesem Sondeneinsatz wird auf eine aktive Regeneration des Erdreiches verzichtet. Deshalb ist die jährliche Wärmeentzugszeit begrenzt. Es ist eine genügend lange Regenerationsphase nötig, in der die Wärme passiv nachströmen und den entstandenen Temperaturtrichter um die Erdwärmesonden wieder auffüllen kann.

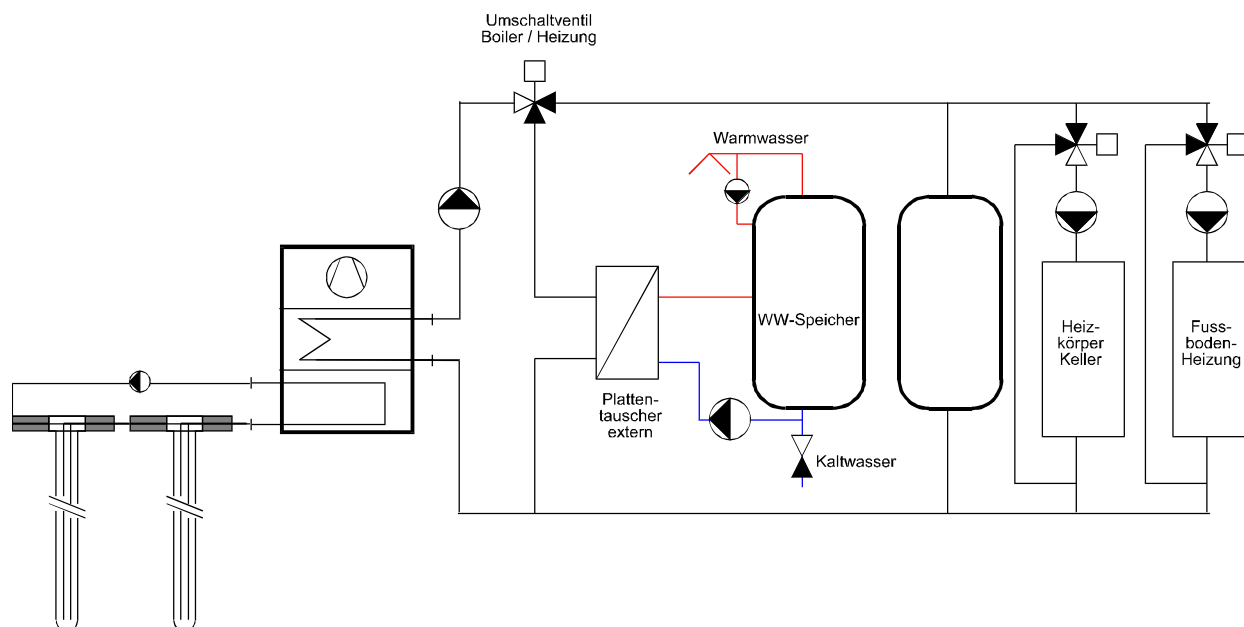


Abb 1-1: Häufigste Anwendung von Erdwärmesonden als Wärmequelle für Wärmepumpen. Keine aktive Regeneration des Erdreiches.

Bedingt durch bessere Kältemittel und Wärmepumpen wird nun immer häufiger neben der Raumheizung von Gebäuden auch das Warmwasser mit der Wärmepumpe erzeugt. Und da die Gebäude immer besser wärmedämmend werden, steigt der anteilmässige Energiebedarf für die Warmwasserproduktion ständig an, wodurch auch die Sondenbelastung stetig zunimmt. Dies bedeutet nun, dass die Regenerationsphase immer kürzer wird und die bisher üblichen Dimensionierungsrichtlinien für Erdwärmesonden versagen.

Bedingt durch den Preiszerfall bei den Erdwärmesonden und Wärmepumpen werden immer häufiger auch ganze Reihensiedlungen mit je einer eigenen Erdwärmesondenanlage ausgerüstet. Dies bedeutet nun aber, dass das Nachströmen der Wärme durch die Nachbarsonden behindert wird, und somit auch hier die klassischen Auslegungsrichtlinien zu kurz greifen.

Primär Kühlung ist das Ziel der folgenden Anlage (Abb 1-2), bei der die Erdwärme direkt, ohne Wärmepumpen, zum Kühlen eingesetzt. Bedingung für die Machbarkeit einer solchen Anlage ist, dass die Quelltemperatur nicht über 17°C - 18°C steigt und die ganze Gebäudestruktur über eine Strukturkühlung miteinbezogen wird. Je etwa 50% der Kühlleistung wird für die Luft- und die Strukturkühlung verwendet. Dank der Luftvorwärmung mit den Erdwärmesonden erfolgt im Winter eine aktive Regeneration des Erdreiches, wodurch auch grosse Erdwärmesondenfelder und somit grosse Kühlleistungen möglich werden. Verwirklicht sind solche Anlagen z.B. beim Neubau des Eidgenössischen Amtes für Messwesen in Wabern oder im neuen Ausbildungszentrum eines Versicherungskonzerns in Rüschlikon mit je 50 Erdwärmesonden. Die Auslegung eines solchen Erdwärmesondenfeldes erfordert eine Simulation, bei der nicht die Entzugsleistung, sondern die Rücklauftemperaturen vorgegeben werden können.

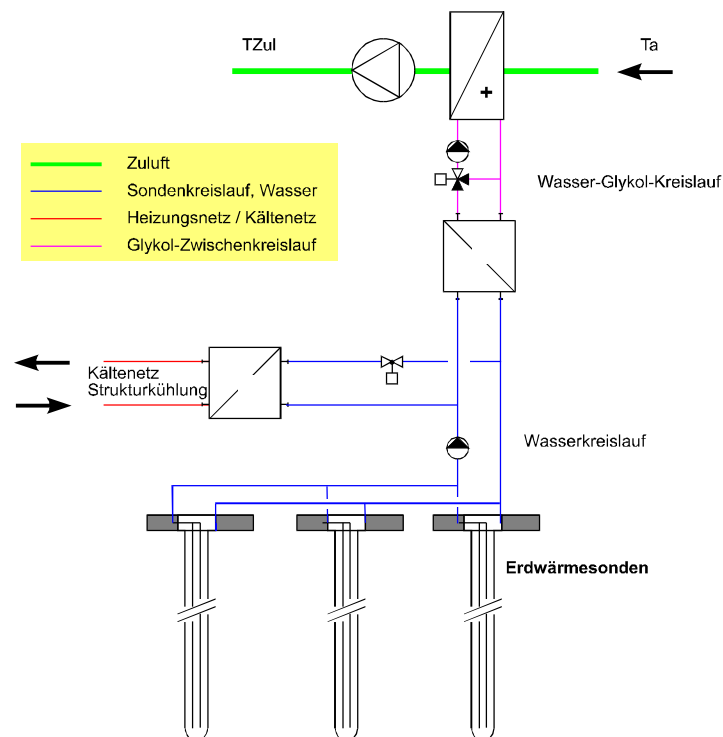


Abb 1-2: Erdwärmesonden zum Kühlen über die Zuluft und eine Strukturkühlung des Gebäudes. Die Luftvorwärmung im Winter regeneriert das Erdreich und ermöglicht den Einsatz grosser Erdwärmesonden - Felder.

Gerade im gewerblichen Bereich (Restaurant, Verkaufsläden, Metzgereien, Bäckereien etc.) wird oft gleichzeitig Wärme und Kälte verlangt. Mit einer Wärmepumpe kann beides geliefert werden (Abb 1-3). Immer wieder gibt es aber Perioden, in denen ein Überschuss an Abwärme aus der Kälteproduktion, oder aber eine zusätzliche Wärmequelle benötigt wird. Hier eignen sich Erdwärmesonden bestens, beide Aufgaben zu übernehmen und die Abwärme bis zu einem späteren Bedarf zwischenzulagern. Dank der Wärmepumpe ist es dabei nicht so entscheidend, dass das Temperaturniveau im Erdreich so tief sinkt, dass es für eine direkte Nutzung nicht mehr gebraucht werden kann. Dank der aktiven Regeneration sind auch bei dieser Anwendung Erdwärmesondenfelder einsetzbar.

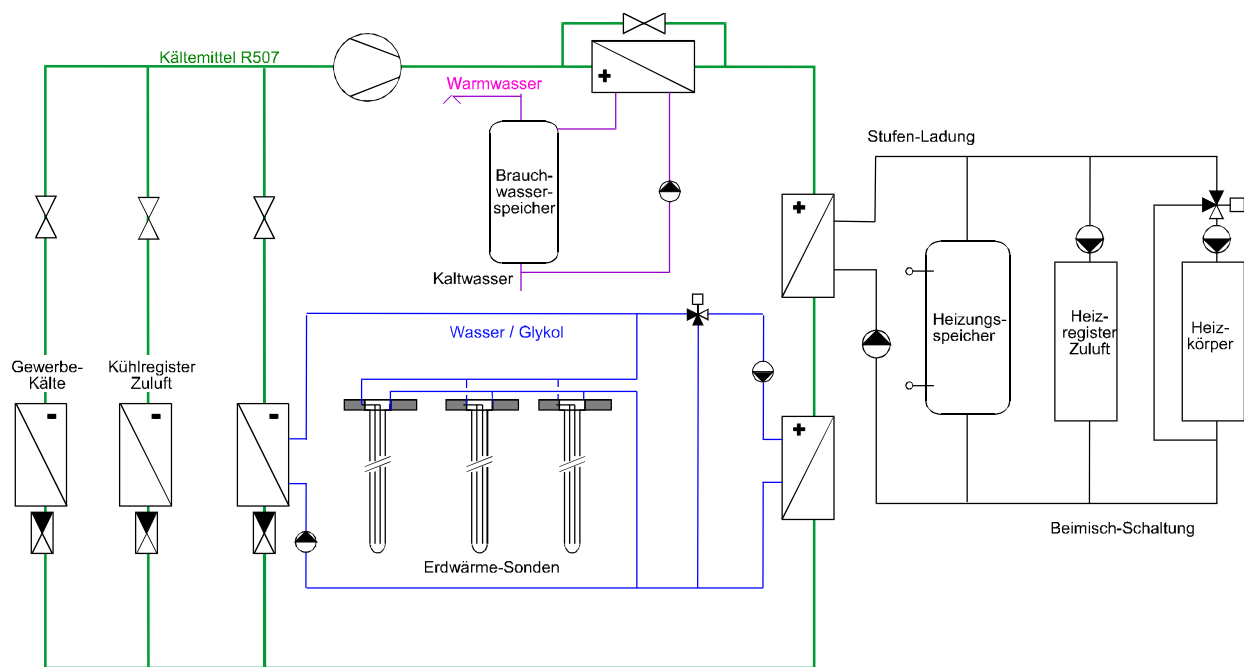


Abb 1-3: Einsatz von Erdwärmesonden als Wärmequelle und Rückkühlung in einer Wärmepumpenanlage für Gewerbekälte und Heizung

Nochmals eine Dimension grösser sind die Erdwärmesondenfelder als Saisonspeicher. Dabei wird teilweise sogar versucht, Überschusswärme aus dem Sommer auf direkt nutzbarem Temperaturniveau in den Winter zu speichern.

1.2 Anforderungen an ein Auslegungsprogramm

Bei all den oben beschriebenen Anwendungen von Erdwärmesonden fällt folgendes auf:

- Es ist unbedingt notwendig, dass neben Einzel auch Sondenfelder berechnet werden können.
- Auch Einzelsondenanlagen sind oft durch Nachbaranlagen mitbeeinflusst und verhalten sich wie Erdwärmesondenfelder mit grossem Sondenabstand.
- Die Regenerationsphase muss ebenso mitberücksichtigt sein wie die Entzugsphase.
- Auch eine aktive Regeneration des Erdreiches muss berechnet werden können.
- Die Temperaturen in einer speziellen Erdwärmesonde eines ganzen Sondenfeldes sind aus der Sicht des Wärmenutzers unwichtig, da die Solekreisläufe in der Regel zusammengefasst und einer gemeinsamen Wärmenutzung zugeführt werden. Wichtig ist einzig die Durchschnittstemperatur aller Erdwärmesonden.
- Die Temperaturen und der Energiebeitrag von einzelnen Erdwärmesonden in einem Sondenfeld lassen sich mit Hilfe der Superpositionstechnik abschätzen. Das Programm muss deshalb nicht in der Lage sein, dies ebenfalls direkt berechnen zu können.
- Damit auch Anlagen ohne Wärmepumpen berechnet werden können, muss auch die Rücklauftemperatur (ohne Kenntnis der Entzugsleistung) als Eingabegrösse zulässig sein.

Die entsprechende Kühl- oder Heizleistung des Erdreiches ist in diesem Fall das Resultat der Berechnung.

- Die Wärmekapazität des Sondenfluids ist nicht vernachlässigbar. Für genaue Berechnungen eines nicht kontinuierlichen Betriebes ist eine instationäre Berechnung erforderlich.
- Die Bohrtiefe ist eine wichtige Auslegungsgröße. Unterschiedliche Gesteinsschichten in verschiedenen Bohrtiefen haben unterschiedliche Stoffwerte und müssen schichtweise berücksichtigt werden können.
- Die Lebensdauer einer Erdwärmesonde sollte mit der Lebensdauer der geplanten Nutzung übereinstimmen. Im Wohn- und Bürobereich betragen diese mehrere Jahrzehnte. Das Rechenwerkzeug muss in der Lage sein, das thermische Verhalten der ganzen geplanten Lebensdauer zu berechnen.
- Die Erdwärmesonde ist immer nur ein Teilelement eines ganzen Systems. Für eine Systemauslegung gehören gleichberechtigt die übrigen Komponenten wie Wärmepumpe und Wärmetauscher. Der Wert eines Auslegungsprogrammes für Erdwärmesonden steht und fällt somit mit der Möglichkeit, weitere Systemkomponenten zu integrieren. Ein modulares Vorgehen ist somit zwingend.
- Die Anwendung des Programmes soll so wenig Simulationskenntnisse wie möglich benötigen. Das Resultat soll nicht durch die Wahl von Simulationsparametern (Wahl des Simulationsgebietes, Temperaturen am Rand des Simulationsgebietes, Wahl des Rechengitters etc.) beeinflusst werden.
- Die Rechenzeit für eine mehrjährige Simulation eines Erdwärmesondenfeldes soll einige Minuten nicht übersteigen.

1.3 Lösungsansatz

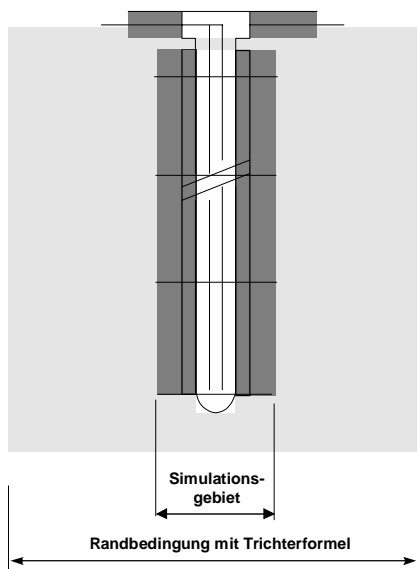


Abb 1-4: Simulation des Erdreiches im Sonden-Nahbereich. Berechnung der Randbe-

dingungen mit dimensionslosen Sprungantworten (g-functions).

Berechnungen von Eskilson (1987) zeigen, dass bei Erdwärmesonden von über 100 m Länge die vertikale Wärmeleitung im Sondennahbereich (< 3 m Abstand von der Sonde) vernachlässigt werden kann. Verzichten wir in diesem Bereich auf die Berechnung der vertikalen Wärmeleitung, so können wir die Wärmeleitungsgleichung in Zylinderkoordinaten schichtweise eindimensional lösen, wodurch bei der Simulation die Rechenzeit stark reduziert wird. Schichtweise sind so unterschiedliche Stoffwerte möglich, so dass auch die häufig vorkommenden, unterschiedlichen Gesteinsschichten im Erdreich mitberücksichtigt werden können. Für die Simulation der Erdreichtemperaturen im

Sonden-Nahbereich (1.5 m – 2 m) soll das werden.
Crank-Nicholson-Verfahren angewendet

Als innere Randbedingung soll die durchschnittliche Soletemperatur der jeweiligen Schicht genommen werden. Um dem gestellten Anforderungsprofil zu genügen, soll die Sole mit einem expliziten Zeitschrittverfahren dynamisch simuliert werden. Dadurch ist es möglich, auch das Anfahrverhalten der Sonde zu berechnen.

Die äusseren Temperatur-Randbedingungen werden aus den dimensionslosen Temperatur-Sprungantworten (g-function) nach der Theorie von Eskilson (1987) berechnet. Das Problem des nicht konstanten Wärmeentzugs und der Regeneration der Erde kann elegant durch Superposition von beliebig vielen, zu unterschiedlichen Zeitpunkten beginnenden, konstanten Wärmeentzügen gelöst werden. Mit diesem Ansatz kann auch der Einfluss von Nachbarsonden mitberücksichtigt werden: Die g-functions geben immer die durchschnittliche Temperaturabsenkung über alle Erdwärmesonden bei einem konstanten Wärmeentzug an. Dementsprechend sind auch die errechneten Soletemperaturen immer Durchschnittstemperaturen aller Erdwärmesonden eines Sondenfeldes.

Mit dem gewählten Ansatz können intern unterschiedliche Zeitschritte verwendet werden: Die kürzesten Zeitschritte werden bei der dynamischen Berechnung der Sole benötigt, für die Crank-Nicholson-Berechnung im Simulationsgebiet kommt ein grösserer Zeitschritt zur Anwendung. Für die Berechnung der Erdreichtemperatur ausserhalb des Simulationsgebietes mit den g-functions reicht gar ein Zeitschritt von einer Woche. Diese unterschiedlichen Zeitschritte sind auch anschaulich einleuchtend: Temperaturstörungen kommen immer von innen (von der Sonde), womit innen auch die kleinsten Zeitschritte nötig sind. Weiter aussen machen sich nur der gemittelte Wärme-Entzug oder -Eintrag bemerkbar. Auf diese Weise gelingt es, ohne einen Verlust an Genauigkeit eine Erdsonde mit wesentlich geringerem Rechenaufwand zu simulieren als mit herkömmlichen Ansätzen.

Als Startwert für die Simulation muss die Temperatur in der Erde vorgegeben werden. Zur Voraussage, wie sich diese Temperatur nach mehreren Jahren Sondenbetrieb um eine Einzelsonde oder in einem Sondenfeld verhält, werden ebenfalls die g-functions verwendet.

Im Projekt „Berechnungsmodul für Erdwärmesonden“ (Huber, Schuler, 1997) wurde ein Programm-Modul entwickelt und validiert, mit dem unbeeinflusste Doppel-U-Sonden (Einzelsonden) berechnet werden können. Dieses Programm wurde im vorliegenden Projekt mit der Möglichkeit ergänzt, auch Sondenfelder berechnen zu können. Über entsprechende, logische Schaltelemente (beschrieben in Kapitel 4) kann im vorliegenden Programm-Modul zwischen dem ersten Ansatz mit der Trichterformel von Werner (1996) und dem neuen Ansatz mit den g-functions als Randbedingung für Sondenfelder gewählt werden. Da es sich beim vorliegenden Programm somit um eine Weiterentwicklung des gut dokumentierten Programmes EWS von 1997 handelt, wurden die bereits im Forschungsbericht von 1997 dokumentierten Programmteile in der vorliegenden Beschreibung nur gekürzt dargestellt, ausführlich wird im folgenden Kapitel dagegen auf die neuen Elemente eingegangen.

Das neue Programm-Modul (EWS-Modul) wurde in der vorliegenden Arbeit durch einen Vergleich mit dem 3-dimensionalen Simulationsprogramm SBM / TRNSBM validiert (Kapitel 3). Damit sollten die Genauigkeit der Methode und die Grenzen des Einsatzgebietes geklärt werden. Das Schwergewicht der Simulation wurde auf die Rechengenauigkeit von Einzelsonden und Sondenfeldern bei Simulationsdauern von mehreren Jahrzehnten gelegt (bis 60 Jahre). Es hat sich dabei gezeigt, dass der Haupteinfluss auf die Rechengenauigkeit bei der Kenntnis der dimensionslosen Sprungantworten (g-fuctions) für die entsprechende Sondengeometrie liegt.

Entsprechend der Philosophie des modularen Programmaufbaus wurde neu eine Eingabeschnittstelle zum Programm SwEWS (Leu et al. 1999) erstellt. Das Programm SwEWS enthält

eine Datenbank, mit der die Stoffwerte für das Molassebecken des Schweizerischen Mittel-landes bestimmbar sind. Diese Stoffwerte können in das Programm EWS importiert werden und ermöglichen so eine Berechnung mit korrekten, schichtweisen Stoffwerten der Erde.

2 Modellbeschreibung

2.1 Simulation des Sonden-Nahbereichs

2.1.1 Crank-Nicholson-Ansatz für den Sonden-Nahbereich

Die eindimensionale Fourierrechnung mit der Temperatur T lautet:

$$\frac{\partial T}{\partial t} = a \frac{\partial^2 T}{\partial x^2} \quad \text{wobei } T = T(t, x) \quad \text{und} \quad a = \frac{\lambda}{\rho c_p} \quad (2-1)$$

Als implizite Differenzengleichung kann sie mit dem Crank-Nicholson-Ansatz geschrieben werden als:

$$\begin{aligned} T_{k+1,j} - \frac{dt}{2} \frac{L_j}{C_j} (T_{k+1,j-1} - T_{k+1,j}) - \frac{dt}{2} \frac{L_{j+1}}{C_j} (T_{k+1,j+1} - T_{k+1,j}) = \\ T_{k,j} + \frac{dt}{2} \frac{L_j}{C_j} (T_{k,j-1} - T_{k,j}) + \frac{dt}{2} \frac{L_{j+1}}{C_j} (T_{k,j+1} - T_{k,j}) \end{aligned} \quad (2-2)$$

Der Index k beschreibt die Zeitkoordinate, der Index j die radiale Ortskoordinate. C ist die Wärmekapazität des Rechenknotens und L die thermische Leitfähigkeit zwischen zwei Rechenknoten. Der thermische Widerstand R ist der Kehrwert der thermischen Leitfähigkeit L.

2.1.2 Lösungsalgorithmus

In Matrix-Schreibweise kann die Gleichung 2-2 am Beispiel mit 3 Masseknoten geschrieben werden als:

$$\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & T_0^{k+1} \\ -dt L_1 & 2C_1 + dt L_1 + dt L_2 & -dt L_2 & 0 & 0 & T_1 \\ 0 & -dt L_2 & 2C_2 + dt L_2 + dt L_3 & -dt L_3 & 0 & T_2 \\ 0 & 0 & -dt L_3 & 2C_3 + dt L_3 + dt L_4 & -dt L_4 & T_3 \\ 0 & 0 & 0 & 0 & 1 & T_4 \end{array} \quad = \quad \begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & T_0^k \\ dt L_1 & 2C_1 - dt L_1 - dt L_2 & dt L_2 & 0 & 0 & T_1 \\ 0 & dt L_2 & 2C_2 - dt L_2 - dt L_3 & dt L_3 & 0 & T_2 \\ 0 & 0 & dt L_3 & 2C_3 - dt L_3 - dt L_4 & dt L_4 & T_3 \\ 0 & 0 & 0 & 0 & 1 & T_4 \end{array} \quad (2-3)$$

Wir wollen im folgenden die obige Gleichung symbolisch darstellen als:

$$[A] \cdot \{T\}_i^{k+1} = [F] \cdot \{T\}_i^k \quad (2-4)$$

Durch Multiplikation dieser Beziehung mit A^{-1} kann die Vektorgleichung einfach gelöst werden:

$$\{T\}_i^{k+1} = [A]^{-1} \cdot [F] \cdot \{T\}_i^k = [B] \cdot \{T\}_i^k \quad (2-5)$$

mit

$$[B] = [A]^{-1} \cdot [F] \quad (2-6)$$

Die Matrix B bleibt während der ganzen Simulation unverändert. Es genügt also, B einmal auszurechnen. Da wir aber für jede Erdschicht eigene Stoffwerte zulassen, muss für jede axiale Schicht eine eigene Matrix B_i berechnet und gespeichert werden.

Um Verwechslungen mit der Soletemperatur zu vermeiden, wird nachfolgend im Simulationsgebiet der Erde die Temperatur T durch TEarth ersetzt.

2.1.3 Rechengitter

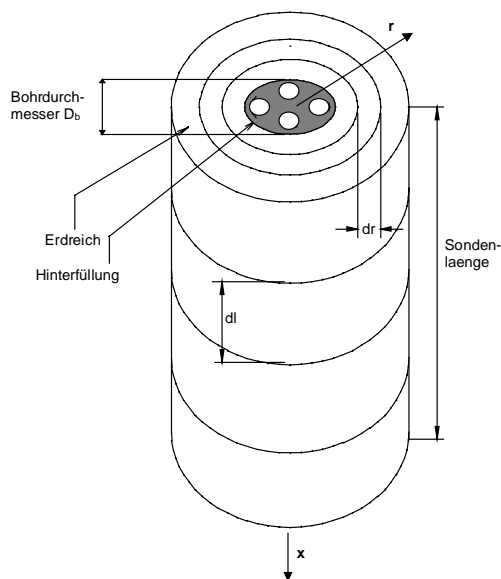


Abb 2-1: Axialsymmetrisches Rechengitter für das Simulationsgebiet

Die Berechnungen werden in einem axialsymmetrischen Rechengitter gemäss Abb. 2-1 durchgeführt. In axialer Richtung wird die Erde in gleiche Teile der Länge dl aufgeteilt.

Das Rechengitter in radialer Richtung ist variabel. Es wird durch den Gitterfaktor f definiert:

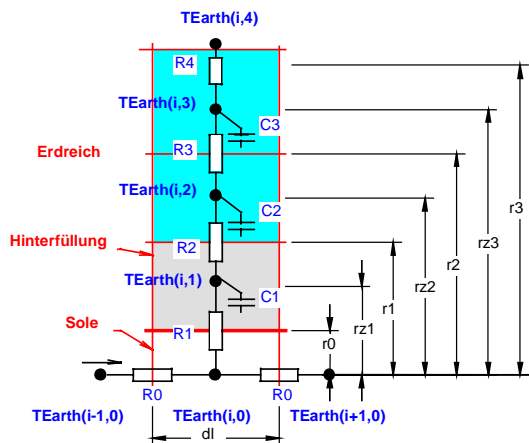
$$\text{Gitterfaktor } f = \frac{r_{j+1} - r_j}{r_j - r_{j-1}} \quad (2-7)$$

Bei Vorgabe des Simulationsgebietes mit dem grössten Rechenradius r_m kann das Gitter wie folgt berechnet werden, wobei m die Anzahl Rechenknoten in radialer Richtung darstellt (d.h. $m = \text{DimRad}$):

$$r_0 = \frac{D_i}{2} = \frac{\text{Innendurchmesser der Sondenrohre}}{2}$$

$$r_1 = \frac{D_b}{2} = \frac{\text{Bohrdurchmesser}}{2}$$

$$\text{für } j \geq 2: \quad r_j = r_{j-1} + (r_m - r_1) \frac{1-f}{1-f^{m-1}} f^{j-2} \quad (2-8)$$



Ein Gitterfaktor von 2 verdoppelt jeweils die Radius-Differenz zwischen zwei Rechen-volumen.

Der für die Bestimmung der thermischen Widerstände wichtige Massenschwerpunkt kann wie folgt berechnet werden:

$$r_{zj} = \sqrt{\frac{(r_j^2 + r_{j-1}^2)}{2}} \quad (2-9)$$

Abb 2-2: Konvention der Bezeichnungen

2.1.4 Wärmekapazitäten C

Die Wärmekapazitäten C_j sind am Beispiel mit 3 Rechenknoten (d.h. DimRad = 3) wie folgt zu berechnen:

$$\begin{aligned} C_1 &= cp_{\text{Fill}} \rho_{\text{Fill}} \pi (r_1^2 - r_0^2) dl \\ C_2 &= cp_{\text{Erde}} \rho_{\text{Erde}} \pi (r_2^2 - r_1^2) dl \\ C_3 &= cp_{\text{Erde}} \rho_{\text{Erde}} \pi (r_3^2 - r_2^2) dl \end{aligned} \quad (2-10)$$

Dabei ist dl die Schichtdicke, wenn die Erde in axialer Sondenrichtung in DimAxi gleiche Elemente unterteilt wird:

$$dl = \frac{\text{Sondenlänge}}{\text{DimAxi}} \quad (2-11)$$

Die Wand des Sondenrohres wird also vernachlässigt (d.h. als Teil der Hinterfüllung gerechnet).

2.1.5 Thermische Widerstände R und Leitfähigkeit L

Der thermische Widerstand R ist der Kehrwert der thermischen Leitfähigkeit L. Dieser ist definiert durch:

$$L = \frac{1}{R} = \frac{\dot{Q}}{\Delta T_{\text{Earth}}} \quad (2-12)$$

Um den thermischen Widerstand an einem Zylinder in radialer Richtung zu berechnen, müssen wir zunächst die Temperaturdifferenz in Funktion des Radius bestimmen. Da der Wärmefluss \dot{Q} zwischen zwei Masseknoten konstant ist, gilt

$$\dot{q} = \frac{dT_{Earth}}{dr} \lambda = \frac{\dot{Q}}{2 \pi r dl} \quad (2-13)$$

Durch Integration zwischen dem Radius r_1 und r_2 lässt sich daraus der Temperaturabfall zwischen zwei Masseknoten berechnen:

$$\Delta T_{Earth} = dT_{Earth} = \frac{\dot{Q}}{2 \pi r dl \lambda} dr = \frac{\dot{Q}}{2 \pi dl \lambda} \ln \frac{r_2}{r_1} \quad (2-14)$$

Der thermische Widerstand zwischen dem Radius r_1 und r_2 beträgt somit

$$L = \frac{1}{R} = \frac{\dot{Q}}{\Delta T_{Earth}} = \frac{2 \pi dl \lambda}{\ln \frac{r_2}{r_1}} \quad (2-15)$$

Die thermischen Widerstände lassen sich wie in der Elektrotechnik addieren. Für Doppel-U-Erdwärmesonden berechnen sich die thermischen Widerstände wie folgt:

$$R_1 = \frac{1}{4} \frac{1}{2 \pi \alpha r_0 dl} + \frac{1}{2 \pi \lambda_{Fill} dl} \ln \frac{r_1 - rz_1}{r_0} \quad (2-16)$$

$$R_2 = \frac{1}{2 \pi dl} \frac{1}{\lambda_{Fill}} \ln \frac{r_1}{rz_1} + \frac{1}{\lambda_{Erde}} \ln \frac{rz_2}{r_1} \quad (2-17)$$

$$R_3 = \frac{1}{2 \pi dl} \frac{1}{\lambda_{Erde}} \ln \frac{rz_3}{rz_2} \quad (2-18)$$

$$R_4 = \frac{1}{2 \pi \lambda_{Erde} dl} \ln \frac{r_3}{rz_3} \quad (2-19)$$

Die Widerstände R_3 und R_4 lassen sich aus der Fourier-Gleichung in Radialkoordinaten exakt herleiten. Bei den Gleichungen für R_1 und R_2 hingegen handelt es sich um einen vereinfachten Ansatz. Es besteht deshalb im Programm auch die Möglichkeit, R_1 als Parameter vorzugeben.

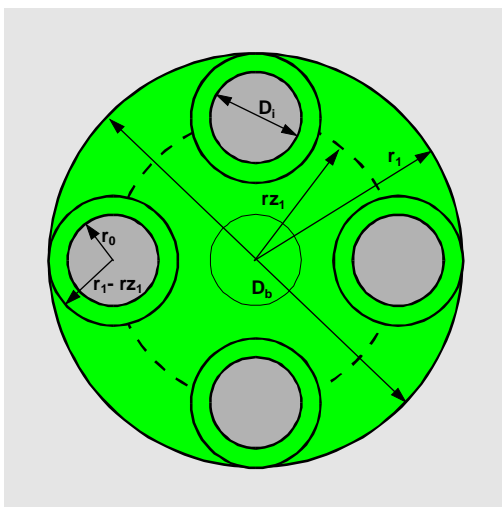


Abb 2-3: Angestrebte, möglichst periphere Lage der 4 Sondenrohre

Anstelle von R_1 und R_2 wird in der Literatur (z.B. Hellström, 1991) auch etwa der „internal thermal resistance“ R_a und der „borehole thermal resistance“ R_b verwendet. R_a ist der thermische Widerstand zwischen der hinunter- und der hinaufströmenden Sole, multipliziert mit der Sondenlänge dl :

$$R_a = 4 dl R_1 \quad (2-20)$$

R_b ist definiert als:

$$R_b = \frac{dl (T_{Sole} - T_b)}{\dot{Q}} \quad (2-21)$$

wobei T_b die „borehole temperature“, die Temperatur beim Radius r_1 ist.
 $T_{\text{Sole}} (=T_{\text{Earth}}(i,0))$ ist die durchschnittliche Soletemperatur der Schicht i , also der Durchschnitt zwischen der hinunter- und hinaufströmenden Sole.

Damit lässt sich die Beziehung zwischen R_1 und R_b herleiten. Diese ist:

$$R_b = R_1 dl + \frac{1}{2 \pi \lambda_{\text{Fill}}} \ln \frac{r_1}{rz_1} \quad (2-22)$$

Ist also R_b bekannt (z.B. durch eine zweidimensionale, numerische Berechnung mit einem Wärmeleitungsprogramm), so kann mit 2-22 auch R_1 berechnet werden. Wird im Programm anstelle von R_1 der Wert für R_b vorgegeben, so berechnet das Programm daraus automatisch mit 2-22 den Wert für R_1 .

Werden andererseits R_a und R_b vorgegeben, so berechnet sich R_1 und R_2 daraus wie folgt:

$$R_1 = \frac{R_a}{4 dl} \quad (2-23)$$

und

$$R_2 = \frac{(R_b - \frac{R_a}{4})}{dl} + \frac{1}{2 \pi dl} \frac{1}{\lambda_{\text{Erde}}} \ln \frac{rz_2}{r_1} \quad (2-24)$$

Auch diese Option ist im Programm EWS möglich, vorausgesetzt dass $R_a < 4R_b$ ist.

2.2 Wärmeübergangskoeffizient Sonde-Sole

2.2.1 Wärmeübergangskoeffizient α_1 beim Betrieb der Erdwärmesonde

Der Wärmeübergangskoeffizient α vom Fluid an die Sondenrohre ist eine Funktion der mit dem Rohrdurchmesser D_i gebildeten Reynoldszahl Re und der Prandtlzahl Pr der Sole. Der Wärmeübergangskoeffizient wird üblicherweise dimensionslos mit der Nusseltzahl Nu für die ausgebildete Rohrströmung angegeben:

$$Nu(Re, Pr) = \frac{\alpha_1 D_i}{\lambda_{\text{Sole}}} \quad (2-25)$$

Je nach Reynoldszahl befinden wir uns im laminaren Bereich ($Re < 2 \cdot 10^3$), dem Übergangsbereich laminar-turbulent ($2 \cdot 10^3 < Re < 10^4$) oder im turbulenten Bereich ($Re > 10^4$).

Für Reynoldszahlen $Re < 2'300$ liegen wir immer im laminaren Bereich, für den Merker (1987) die folgende Nusseltzahl angibt:

$$Nu_{\text{lam}} = 4.36 \quad (2-26)$$

Im turbulenten Bereich $Re > 10^4$ wird im Programm EWS die Petukhov - Formel für den Wärmeübergang im Rohr angewendet:

$$Nu_{turb} = \frac{\frac{\xi}{8} Re Pr}{K_1 + K_2 \sqrt{\frac{\xi}{8}} (Pr^{2/3} - 1)} \quad (2-27)$$

mit

$$K_1 = 1 + 27.2 \frac{\xi}{8}$$

$$K_2 = 11.7 + 1.8 Pr^{-1/3}$$

$$\xi = \frac{1}{(1.82 \log Re - 1.64)^2}$$

Zwischen dem laminaren und dem turbulenten Bereich liegt ein Übergangsbereich ($2'300 < Re < 10'000$), für den bis heute noch keine Untersuchungen bekannt sind. Für die vorliegende Arbeit soll in diesem Bereich ein stetiger Ansatz für die Nusseltzahl gewählt werden, der analog zum turbulenten Bereich von einer exponentiellen Zunahme der Nusseltzahl mit der Reynoldszahl ausgeht.

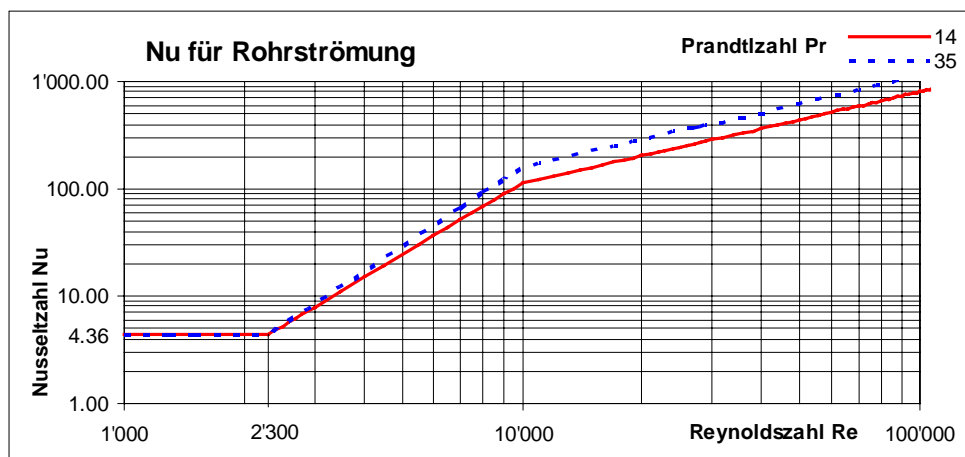


Abb 2-4: Mittlere Nusseltzahl für die Rohrströmung in Abhängigkeit der Reynoldszahl Re

2.2.2 Wärmeübergangskoeffizient α_0 beim Stillstand der Erdwärmesonde

Beim Stillstand der Erdwärmesonde wird davon ausgegangen, dass in der Sole keine Konvektion stattfindet. Die Erwärmung der Sole im Stillstand erfolgt somit ausschliesslich durch Wärmeleitung. Es wird angenommen, dass die Wärmeleitung im Durchschnitt bis zum radialen Massenschwerpunkt zu erfolgen hat. Im Stillstand der Erdwärmesonde gilt somit der folgende Ansatz:

$$\alpha_0 = \frac{\lambda_{Sole}}{\frac{D_i}{2} (1 - \sqrt{0.5})} \quad (2-28)$$

2.3 Modellierung der Sole

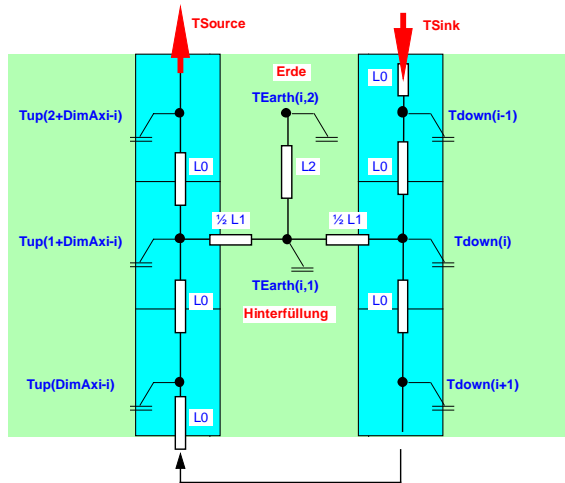


Abb 2-5: Dynamische Berechnung der Sole mit explizitem Zeitschrittverfahren

Nachfolgend ist der Rechengang für die Modellierung der Sole kurz dargestellt. Eine ausführliche Beschreibung ist in (Huber, Schuler, 1997) zu finden.

Für die folgenden Betrachtungen werden die beiden Sondenrohre zusammengefasst, in denen die Sole nach unten fließt und diejenigen, bei denen die Sole hinaufströmt. Dabei berechnet sich die Strömungsgeschwindigkeit v einer Doppel-U-Sonde aus dem Solemassenstrom \dot{m} mit

$$v = \frac{\dot{m}}{2 \pi r_0^2 \rho_{Sole}} \quad (2-29)$$

Analog zu den thermischen Leitfähigkeiten in der Erde kann auch in der Sole die Leitfähigkeit L_0 definiert werden:

$$L_0 = \frac{\dot{Q}_{Ein} - \dot{Q}_{Aus}}{T_{Ein} - T_{Aus}} = cp_{Sole} \dot{m} = 2 \pi r_0^2 v \rho_{Sole} cp_{Sole} \quad (2-30)$$

Wir stellen nun die Bilanz für das Sole-Element der Länge dl auf. Das Bilanzelement (bestehend aus den beiden zusammengefassten Sondenstücken der Doppel-U-Sonde) hat somit eine Masse m von

$$m = 2 \pi r_0^2 dl \rho_{Sole} \quad (2-31)$$

Für die Modellierung der Sole soll ein dynamischer Ansatz gewählt werden. Die Soletemperatur wird dabei mit einem expliziten Zeitschrittverfahren berechnet. Dadurch wird auch eine Erwärmung der Sole beim Stillstand der Solepumpe mitberücksichtigt.

Aus der Energiebilanz für das Bilanzelement i ergibt sich so für die Soletemperatur $T_{down,i}$ der hinunterfließenden Sole:

$$T_{down_{k+1,i}} = T_{down_{k,i}} + \left(T_{down_{k+1,i-1}} - T_{down_{k,i}} \right) \frac{L_0 dt^2}{m cp} + \left(T_{Earth_{k,i,1}} - T_{down_{k,i}} \right) \frac{L_1 dt^2}{2 m cp} \quad (2-32)$$

und der hinaufströmenden Sole mit der Soletemperatur $T_{up,i}$:

$$T_{up_{k+1,i}} = T_{up_{k,i}} + \left(T_{up_{k+1,i-1}} - T_{up_{k,i}} \right) \frac{L_0 dt^2}{m cp} + \left(T_{Earth_{k,1+DimAxi-i,1}} - T_{up_{k,i}} \right) \frac{L_1 dt^2}{2 m cp} \quad (2-33)$$

Um die Rechenzeit zu verkürzen, ist es auch möglich, die Sole stationär zu berechnen. Dazu muss der Parameter *stationaer* = true gesetzt werden.

2.4 Äussere Randbedingung mit den „g-functions“

2.4.1 Dimensionslose Sprungantwort „g-function“ von Eskilson

Sowohl Einzelsonden als auch Sondenfelder weisen bei geometrischer Ähnlichkeit auch ähnliche Temperatur-Sprungantworten (=„g-function“) auf. Der Zeitmassstab muss dabei nach Eskilson (1987) mit der Sondenzeitkonstante t_s

$$t_s = \frac{H^2}{9a} \quad (2-34)$$

dimensionsbefreit werden. Die dimensionslose Zahl E_s von Eskilson

$$E_s = \frac{t \ 9a}{H^2} \quad (2-35)$$

kann somit als dimensionslose Zeit für Sondenfelder und Einzelsonden betrachtet werden.

Nach Eskilson (1987) kann bis zum Zeitpunkt t_s ($E_s = 1$) die vertikale Wärmeleitung im Erdreich (bei einem maximalen Fehler von 7% für eine Einzelsonde) vernachlässigt werden, ab ca. $E_s = 10$ ist der Gleichgewichtszustand im Erdreich beinahe hergestellt. Im Gleichgewichtszustand fliesst im Jahresschnitt die gleiche Energiemenge, die mit der Sonde dem Erdreich entzogen wird, von der Erdoberfläche und zu einem kleineren Teil vom Erdinnern vertikal wieder nach. Es findet somit keine weitere Auskühlung mehr statt.

Die Dimensionsbefreiung der Temperaturabsenkung ΔT bei einer spezifischen Wärmeentzugsleistung \dot{q} erfolgt mit \dot{q} , der Wärmeleitfähigkeit der Erde λ_{Erde} und der Sondenlänge H . Damit kann die dimensionslose Temperaturabsenkung g definiert werden als

$$g \left(E_s, \frac{r_b}{H} \right) = \frac{\Delta T \ 2\pi \ \lambda_{\text{Erde}}}{\dot{q}} \quad (2-36)$$

Die dimensionslose Temperatursprungantwort g ist nach Eskilson (1987) sowohl für Einzelsonden, als auch für Sondenfelder einzig eine Funktion der dimensionslosen Zeit (E_s) und des dimensionslosen Sondenabstandes r_b/H . Es wird dabei von einem konstanten, spezifischen Wärmeentzug pro Sondenlänge \dot{q} ausgegangen. Für eine Einzelsonde im Bereich $5r_b^2/a < t < t_s$ kann die Funktion g bei einem maximalen Fehler von 7% angenähert werden mit

$$g(E_s, r_b / H) = \ln\left(\frac{H}{2r_b}\right) + 0.5 \ln(E_s) \quad (2-37)$$

Für Zeiten grösser als t_s strebt die Einzelsonde dem folgenden Gleichgewichtszustand zu:

$$g(r_b / H) = \ln\left(\frac{H}{2r_b}\right) \quad (2-38)$$

Um die absolute Temperatur im Erdreich zu erhalten, kann nun die Sprungantwort ΔT einfach von der Anfangstemperatur T_m abgezählt werden (Superpositionsprinzip). Die Bohrlochtemperatur T_b z.B. kann aus g und T_m wie folgt berechnet werden:

$$T_b = T_m - \frac{\dot{q}}{2\pi\lambda_{\text{Erde}}} g(Es, r_m / H) \quad (2-39)$$

wobei \dot{q} die spezifische Wärmeentzugsleistung pro Sondenlänge ist:

$$\dot{q} = \frac{\dot{Q}}{\text{Sondenlänge}} \quad (2-40)$$

Was jetzt noch fehlt, ist die Abhängigkeit der g -Funktion vom Sondenabstand r_m . Dafür gilt nach Eskilson der folgende, logarithmische Zusammenhang:

$$g(Es, r_m / H) = g(Es, r_b / H) - \ln(r_m / r_b) \quad (2-41)$$

Diese Formel ist im Sonden-Nahbereich gültig und ermöglicht es uns, mit einer einzigen Sprungantwort g das Temperaturverhalten im ganzen Sonden-Nahbereich zu bestimmen.

In Bild 2-6 sind als Beispiel die dimensionslosen Temperatur-Sprungantworten („ g -functions“) von 2 Erdwärmesonden im Abstand B eingezeichnet. Im Vergleich dazu ist gestrichelt die g -function einer Einzelsonde zu finden.

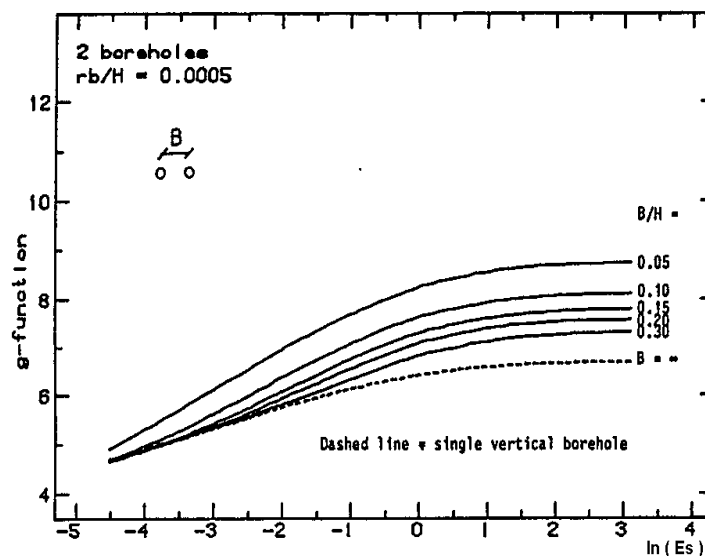


Abb 2-6: Dimensionslose Sprungantworten „ g -functions“ für 2 Erdwärmesonden mit dem Abstand B nach Eskilson (1987). Die Sprungantwort einer Einzelsonde ist gestrichelt eingezeichnet.

Für das vorliegende Programm benötigen wir die Temperaturen einzig am äusseren Rand des Simulationsgebietes, also am Radius $r_m = \text{Rechenradius}$. Die Wärmeentzugsleistung q andererseits muss direkt an der Erdwärmesonde genommen werden. Daraus ergibt sich ein kleiner, dynamischer Fehler, der durch die bloss wöchentliche Neuberechnung der äusseren Temperaturrandbedingung teilweise kompensiert wird.

2.4.2 Superpositionsprinzip

Beim normalen Betrieb einer Erdsonde haben wir keinen konstanten Wärmeentzug. Da die Wärmeleitungsgleichung linear ist, kann aber die aus einer variablen Entzugsleistung resultierende Temperaturverteilung mit Hilfe des Superpositionsprinzips als Summe verschiedener, konstanter Entzugsleistungen berechnet werden:

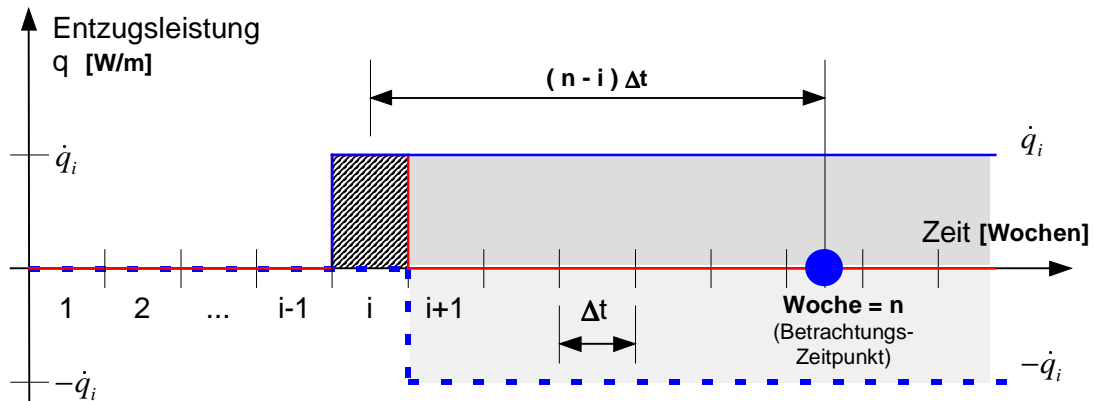


Abb 2-7: Berechnung des Temperaturtrichters aus einem Wärmeentzug q_i in der i -ten Woche durch Superposition zweier zeitlich versetzter Sprungantworten von $+q_i$ und $-q_i$

Ein Wärmeentzug von q_i in der i -ten Woche kann beispielsweise als Superposition einer konstanten Entzugsleistung q_i , beginnend am Anfang der i -ten Woche und einer konstanten Entzugsleistung $-q_i$, beginnend am Ende der i -ten Woche berechnet werden. In der Betrachtungswoche n erhalten wir daraus den folgenden Temperaturtrichter:

$$\Delta T(r, t = n\Delta t) = \frac{q_i}{2\pi\lambda} \left[g(r, t = (n-i+1)\Delta t) - g(r, t = (n-i)\Delta t) \right] \quad (2-42)$$

In der eben beschriebenen Weise kann ein beliebiges Entzugsprofil zusammengesetzt werden, wobei:

$$\Delta T(r, t = n\Delta t) = \sum_{i=1}^n \frac{q_i}{2\pi\lambda} \left[g(r, t = (n-i+1)\Delta t) - g(r, t = (n-i)\Delta t) \right] \quad (2-43)$$

mit

$$g(r, t = 0) = 0 \quad (2-44)$$

In der Gleichung 2-43 muss jedes Glied $g(r, t)$ zweimal berechnet werden. Wir formen diese Gleichung deshalb um in:

$$\Delta T(r, t = n\Delta t) = \sum_{i=1}^n \frac{g(r, t = i\Delta t)}{2\pi\lambda} [q_{n-i+1} - q_{n-i}] \quad (2-45)$$

mit

$$q_0 = 0 \quad (2-46)$$

Die Gleichungen 2-43 und 2-45 sind identisch, Gleichung 2-45 erfordert aber nur halb soviel Rechenaufwand wie 2-43.

Es hat sich gezeigt, dass es für einen typischen Wärmepumpenbetrieb genügt, die äusseren Randbedingungen nur einmal pro Woche neu zu berechnen, sofern das Simulationsgebiet grösser als 1 m ist. Entsprechend ist $\dot{Q}(i, \text{Woche})$ die durchschnittliche Entzugsleistung in der i-ten axialen Schicht in der entsprechenden Woche:

$$\dot{Q}(i, \text{Woche}) = \frac{[T_{\text{Earth}}(i, 1) - T_{\text{Earth}}(i, 0)] L_1 dt}{\text{Woche} \cdot 7 \cdot 24 \cdot 3600s} \quad (2-47)$$

Daraus berechnet sich die spezifische Wärmeentzugsleistung q für diese Woche mit

$$q_{\text{Woche}} = \frac{\dot{Q}(i, \text{Woche})}{H / \text{DimAxi}} \quad (2-48)$$

Damit kann in jeder neuen Woche aus 2-45 und 2-48 die äussere Randbedingung für das Simulationsgebiet berechnet werden:

$$T_{\text{Earth}}(i, \text{DimRad} + 1) = T_0(i) - \Delta T(i, r = r_{\text{DimRad}}, \text{Woche}) \quad (2-49)$$

wobei $T_0(i)$ die Temperatur zum Simulationsbeginn am äusseren Rand des Simulationsgebietes in der i-ten axialen Schicht ist.

2.4.3 Polynom - Ansatz für die g-function

Die dimensionslosen Temperatur-Sprungantworten (g-functions) sind in grosser Zahl publiziert (Eskilson, 1987). Um diese in einem Simulationsprogramm einsetzen zu können, müssen sie zunächst einmal parametrisiert werden. In der vorliegenden Arbeit wurde dazu ein Polynomansatz gewählt. Die g-functions werden im Zeitbereich $-4 < \ln(t/t_s) < 2.5$ durch ein Polynom 5. Grades approximiert. Als Eingabeparameter werden die Funktionswerte bei den Stützstellen $\ln(t/t_s) = -4 / -2 / 0 / 2 / 3$ benötigt. Diese können aus den veröffentlichten Diagrammen herausgelesen werden. Im Bereich $\ln(t/t_s) < -4$ wirken sich Störungen nur noch im Sonden-Nahbereich aus, auch Sondenfelder verhalten sich hier aufgrund der fehlenden gegenseitigen Beeinflussung der einzelnen Sonden kurzfristig wie Einzelsonden. Alle g-functions gehen deshalb in diesem Bereich in den Wert für Einzelsonden über. Dafür hat Eskilson die folgenden Beziehung angegeben:

$$g(\ln(t / t_s) \leq -4) = \ln\left(\frac{H}{2r_b}\right) + 0.5 \ln(t / t_s) \quad (2-50)$$

Ab dem Zeitpunkt $t/t_s = 2.5$ haben die Sprungantworten den stationären Zustand praktisch erreicht, die g-functions bleiben deshalb ab diesem Zeitpunkt unverändert.

$$g(\ln(t / t_s) \geq 2.5) = g(\ln(t / t_s) = 2.5) \quad (2-51)$$

Da bisher g-functions nur für einen Sondenabstand von $r_b / H = 0.0005$ publiziert wurden, sind auch im vorliegenden Programm immer diese Sprungantworten einzulesen.

2.4.4 Extrapolation der g-functions

Um dem Programmbenutzer die Extrapolation zu den publizierten g-functions für verschiedene Sondenabstände zu erleichtern, ist im Programm-Modul EWS ein empirischer Ansatz dazu einprogrammiert. Der Programmbenutzer sollte sich dabei allerdings bewusst bleiben, dass eine solche Extrapolation immer eine zusätzliche Fehlerquelle bedeutet. Im Kapitel 3 werden die Größenordnung dieser Abweichung aufgezeigt. Je mehr der effektive Sondenabstand vom Sondenabstand der Eingabe-g-function abweicht, um so grösser wird dabei auch der Fehler. Die Möglichkeit der Extrapolation sollte deshalb nur bei eher kleinen Abweichungen der effektiven Sondenabstände von den g-function-Sondenabständen verwendet werden.

Das Programm-Modul EWS verwendet die folgende Extrapolation:

$$g_{eff}(\ln(t/t_s) = -4) = 4.82 + \frac{B_{dia}}{B_{eff}} C_A e^{-\frac{B_{eff}}{B_{dia}} C_B^{343}} \quad (2-52)$$

$$g_{eff}(\ln(t/t_s) = -2) = 5.69 + \frac{B_{dia}}{B_{eff}} C_A e^{-\frac{B_{eff}}{B_{dia}} C_B^{125}} \quad (2-53)$$

$$g_{eff}(\ln(t/t_s) = -0) = 6.29 + \frac{B_{dia}}{B_{eff}} C_A e^{-\frac{B_{eff}}{B_{dia}} C_B^{27}} \quad (2-54)$$

$$g_{eff}(\ln(t/t_s) = +2) = 6.57 + \frac{B_{dia}}{B_{eff}} C_A e^{-\frac{B_{eff}}{B_{dia}} C_B} \quad (2-55)$$

$$g_{eff}(\ln(t/t_s) = +3) = 6.6 + \frac{B_{dia}}{B_{eff}} C_A \quad (2-56)$$

mit den Koeffizienten:

$$C_A = g_{dia}(\ln(t/t_s) = 0) - 6.29 \quad (2-57)$$

$$C_B = \frac{1}{27} \ln \frac{g_{dia}(\ln(t/t_s) = 0) - 6.29}{g_{dia}(\ln(t/t_s) = 3) - 6.6} \quad (2-58)$$

Die Werte 4.82 / 5.69 / 6.29 / 6.57 und 6.6 sind die Funktionswerte der g-function bei den Stützstellen $\ln(t/t_s) = -4 / -2 / 0 / 2$ und 3 für Einzelsonden und sind somit die kleinst möglichen Funktionswerte zu diesem Zeitpunkt.

Fehler bei der Abbildung der g-functions wirken sich proportional zum berechneten, langfristigen Temperaturniveau der Bohrlochtemperatur aus. Der Fehler ist aber durch das gewählte Rechenverfahren nicht additiv, d.h. er vergrössert sich nicht ständig. Die Kurzzeiteffekte wirken sich andererseits nur im Simulationsgebiet aus und sind durch eine etwaige Ungenauigkeit der g-function nur wenig beeinflusst. Die Validierung (Kapitel 3) hat gezeigt, dass das hier beschriebene, empirische Extrapolationsverfahren bis zu einem Sondenabstand gültig ist, der halb so gross ist wie der Sondenabstand der eingelesenen g-function. Bei kleineren Sondenabständen wird im Programm-Modul EWS die Fehler-Variable **ErrorMldg** auf 1 gesetzt. Die Rechnung kann dann in solchen Fällen numerisch instabil werden. Ist der Sondenabstand grösser, so sind nur geringfügige Abweichungen zu erwarten.

2.4.5 Verwendete g-functions

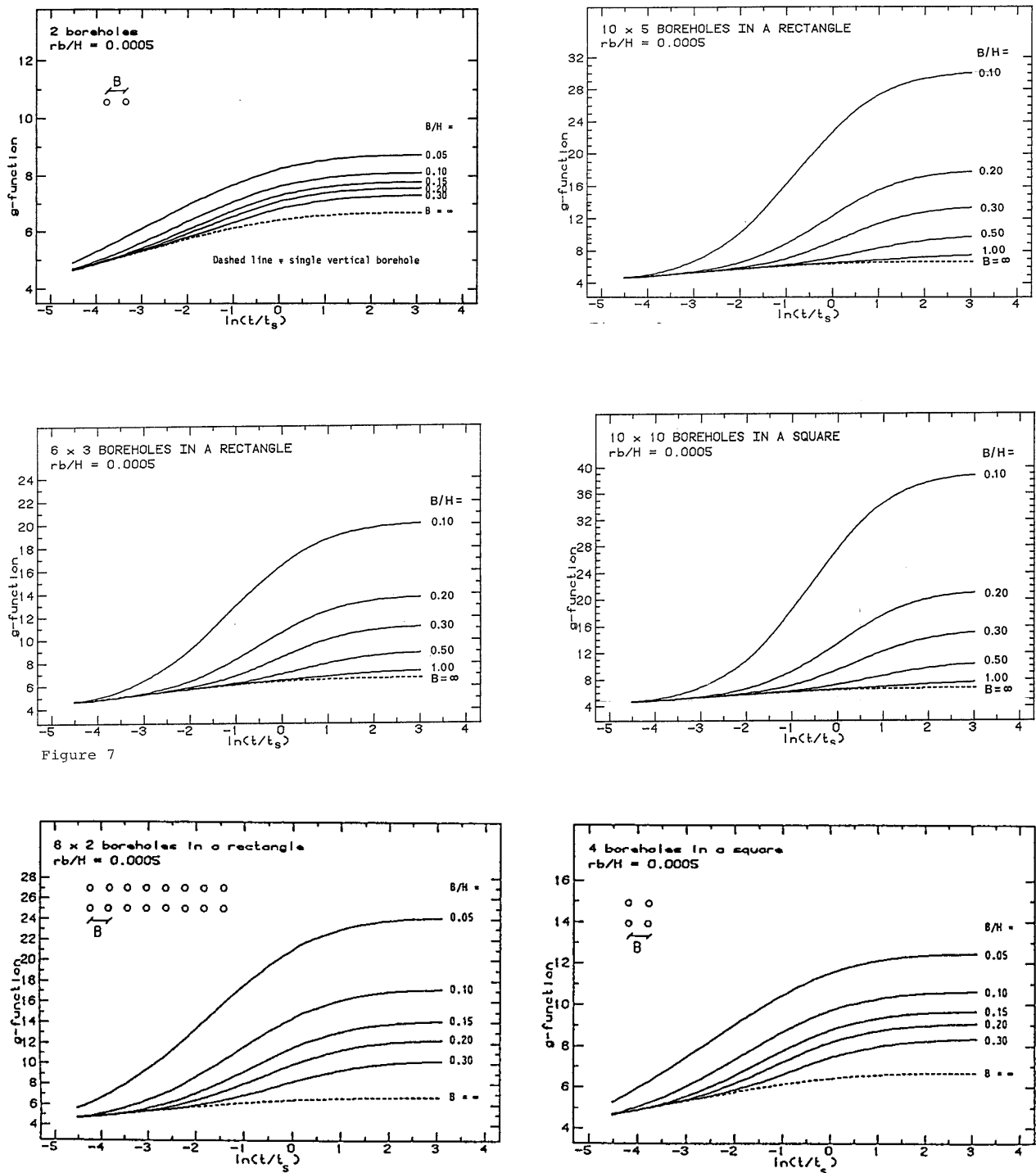


Figure 7

Abb 2-8: g-functions aus Eskilson (1987), die für die Validierung in Kapitel 3 verwendet wurden

2.5 Abschätzung der Starttemperaturen

In der Regel interessiert uns das thermische Verhalten von Sondenfeldern auch nach mehrjährigem Betrieb. Wir haben deshalb die Möglichkeit, entweder all diese Jahre durchzusimulieren, oder wir schätzen mit Hilfe der g-function die Starttemperatur des Sondenfeldes vor dem letzten Simulationsjahr ab.

Für diese Abschätzung muss im Programm Starttemp = true gesetzt werden, das letzte Simulationsjahr wird mit dem Parameter *LastYear* definiert. In diesem Fall werden zunächst die durchschnittlichen, spezifischen Entzugsleistungen $q(i)$ für alle Schichten im ersten Simulationsjahres berechnet (i bezeichnet die axiale Schicht). Mit dieser Entzugsleistung wird dann mit Hilfe der g-function-Theorie nach 2/3 der gewählten Simulationsdauer die Erdreichtemperatur wie folgt abgeschätzt:

$$TEarth_0(i, j) = T_{mo} + (i dl - \frac{dl}{2}) T_{Grad} - q(i) \frac{g(t/t_s, 0.0005) - \ln \frac{rz_j / H}{0.0005}}{2 \pi \lambda_{Erde}} \quad (2-59)$$

wobei für die Zeit t

$$t = \frac{2}{3} \cdot LastYear \cdot 8760 \cdot 24 \cdot 3600 \text{ [s]} \quad (2-60)$$

einzusetzen ist. Mit dieser Starttemperatur wird noch einmal ein Jahr simuliert und erneut die spezifische Entzugleistung $q(i)$ während dieses Jahres berechnet. Danach wird mit (2-59) die Starttemperatur vor dem letzten Simulationsjahr abgeschätzt, wobei für die Zeit t

$$t = (LastYear - 1) \cdot 8760 \cdot 24 \cdot 3600 \text{ [s]} \quad (2-61)$$

einzusetzen ist. Mit dieser neuen Schätzung für die Starttemperaturen vor dem letzten Simulationsjahr wird dann ein letztes Mal ein Jahr lang simuliert.

Mit dieser Methode ist es möglich, mit der Rechenzeit von 3 Simulationsjahren das thermische Verhalten eines Erdsondenfeldes auch nach mehreren Jahrzehnten Betriebszeit zu berechnen.

Diese Rechenmethode eignet sich sowohl bei Entzugsleistungen (Q_{Source}) als Eingabegrößen für die Simulation (Iteration), als auch bei Rücklauftemperaturen (T_{sink}).

2.6 Auslegungs - Lastprofil

Um eine Erdwärmesonde oder ein Sondenfeld dimensionieren zu können, muss ein sinnvolles Lastprofil vorgegeben werden können. Im vorliegenden Programm-Modul EWS kann mit dem Aufruf *Erdsonde(ErzeugeLastprofil)* ein Simulations-Eingabefile mit einem Auslegungs-Lastprofil im Stundenschritt erzeugt werden. Dabei kann für jeden Monat die tägliche Sondenlaufzeit (Anzahl Stunden pro Tag) definiert werden. Diese tägliche Laufzeit kann im Vektor *Laufzeit[1..12]* vorgegeben werden. Ausserhalb der Laufzeit wird der Sondendurchsatz auf Null gesetzt. Bei Anlagen, bei denen die Erdwärmesonde als Wärmequelle für eine Wärmepumpe dient, treten in der Regel im Februar die tiefsten Quelltemperaturen auf. Deshalb ist im Lastprofil Ende Februar ein mehrtägiger Dauerentzug vorgesehen (definiert im Parameter *DauerLastspitze*).

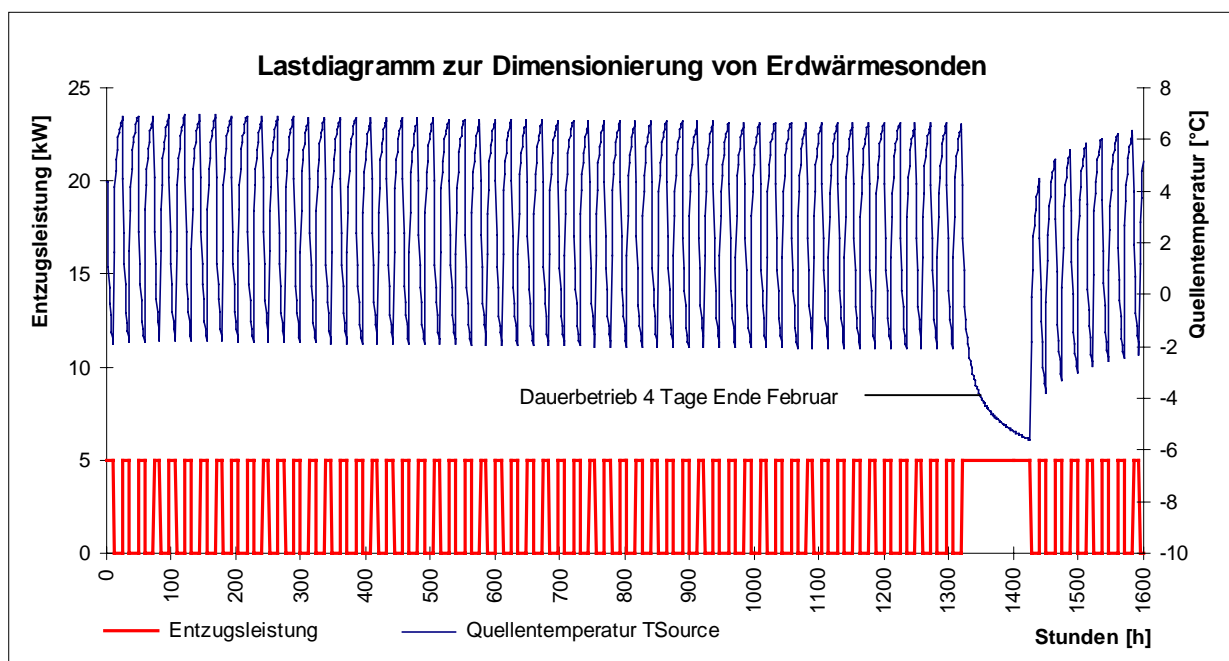


Abb 2-9: Beispiel eines Auslegungs-Lastprofils mit 10 Stunden Betriebszeit im Januar und Februar und einem 4-tägigen Dauerentzug Ende Februar

Beim Rechnen mit diesem Lastprofil werden sodann die minimalen und maximalen Rücklauf- und Quelltemperaturen berechnet ($T_{SourceMin}$, $T_{SourceMax}$, $T_{SinkMin}$, $T_{SinkMax}$). Diese Temperaturen können zur Überprüfung der Sondenauslegung verwendet werden.

2.7 Stoffwerte des Erdreichs und Bodentemperaturen

Für die korrekte Berechnung des thermischen Verhaltens von Erdwärmesonden und Sondenfeldern sind die Stoffeigenschaften (Wärmeleitfähigkeit λ_{Erde} , Wärmekapazität $c_{p,\text{Erde}}$, Dichte ρ_{Erde}) und die Temperaturen des ungestörten Erdreiches von entscheidender Bedeutung. Deshalb wurde im Auftrag des Bundesamtes für Energie das Programm SwEWS entwickelt, das auf der Basis von 230 Messungen im Schweizer Molassebecken Vorschlagswerte für diese Stoffeigenschaften in verschiedenen Schichten bis 500 m Tiefe generiert (Leu et al, 1999). Als Eingabeparameter wird die geografische Länge und Breite, die Schichtdicke und die Zuordnung zu einer geologischen Schicht des Molassebeckens verlangt (USM, OMM, USM, Schlamm-Siltstein, Siltstein, Feinsandstein, Mittelsandstein, Grobsandstein und Konglomerat). Diese Angaben sind oft aus Bohrprofilaufnahmen von Erdwärmesonden in der Nähe abschätzbar (erhältlich bei den kantonalen Bewilligungsbehörden, bei den Bohrfirmen oder den lokalen Geologiebüros). Als weitere Hilfestellung kann die nächstgelegene, in der Datenbank enthaltene Bohrung angezeigt werden. Eine gewisse Unsicherheit kommt aus der notwendigen Abschätzung der Schichtdicke des über der Molasse liegenden Schotter- und Moränengebietes.

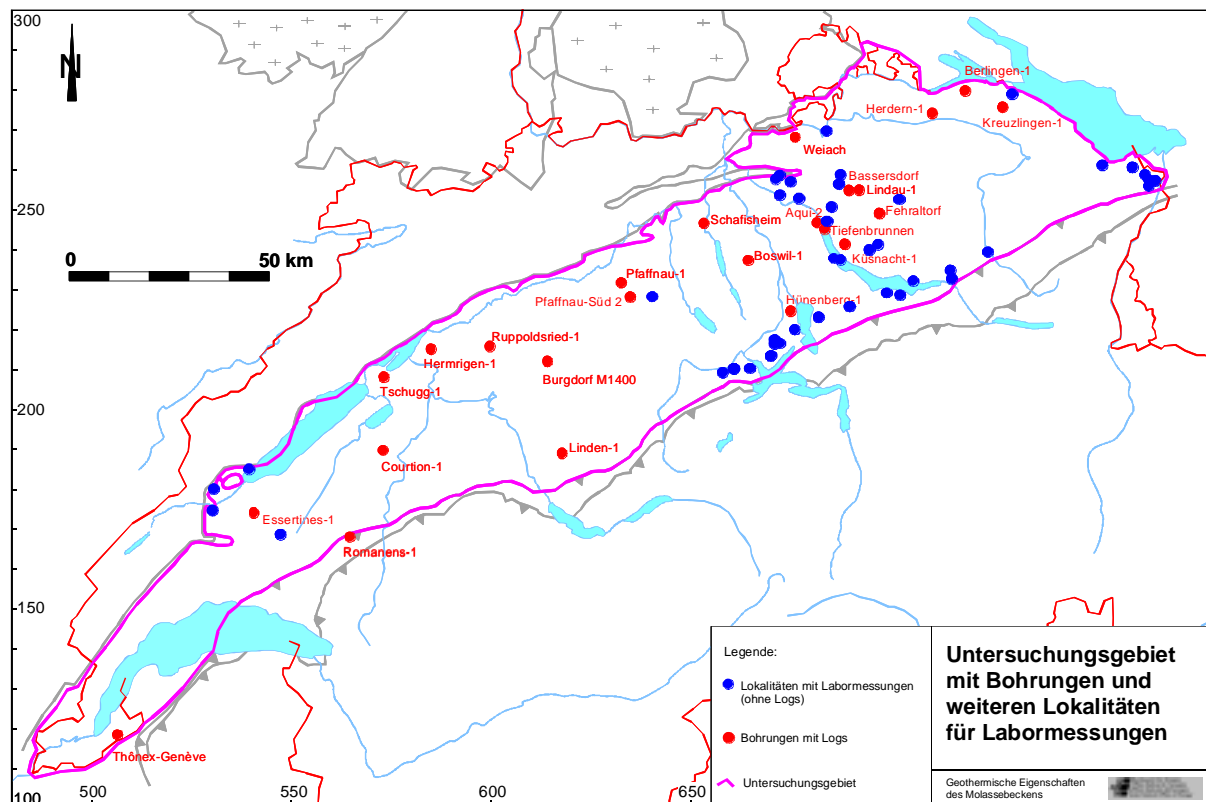


Abb 2-10: Bodeneigenschaft des schweizerischen Mittellandes können aus dem Programm SwEWS-99 (Leu et al. 1999) exportiert und in das Programm EWS importiert werden.

Die mit dem Programm SwEWS schichtweisen bestimmten Stoffwerte und Temperaturen des ungestörten Erdreiches können nun exportiert und mit dem Befehl *Erdsonde(Stoffdaten)* in das Programm EWS importiert werden. Falls die Schichtdicken des Programms EWS nicht mit den geologischen Schichtdicken übereinstimmen, so werden im Programm EWS jeder Rechenschicht eigene, mit der entsprechenden Dicke der geologischen Schicht gewichtete Stoffwerte und Starttemperaturen zugeordnet. Die Anzahl und Dicken der Rechenschichten und der geologischen Schichten müssen also nicht übereinstimmen.

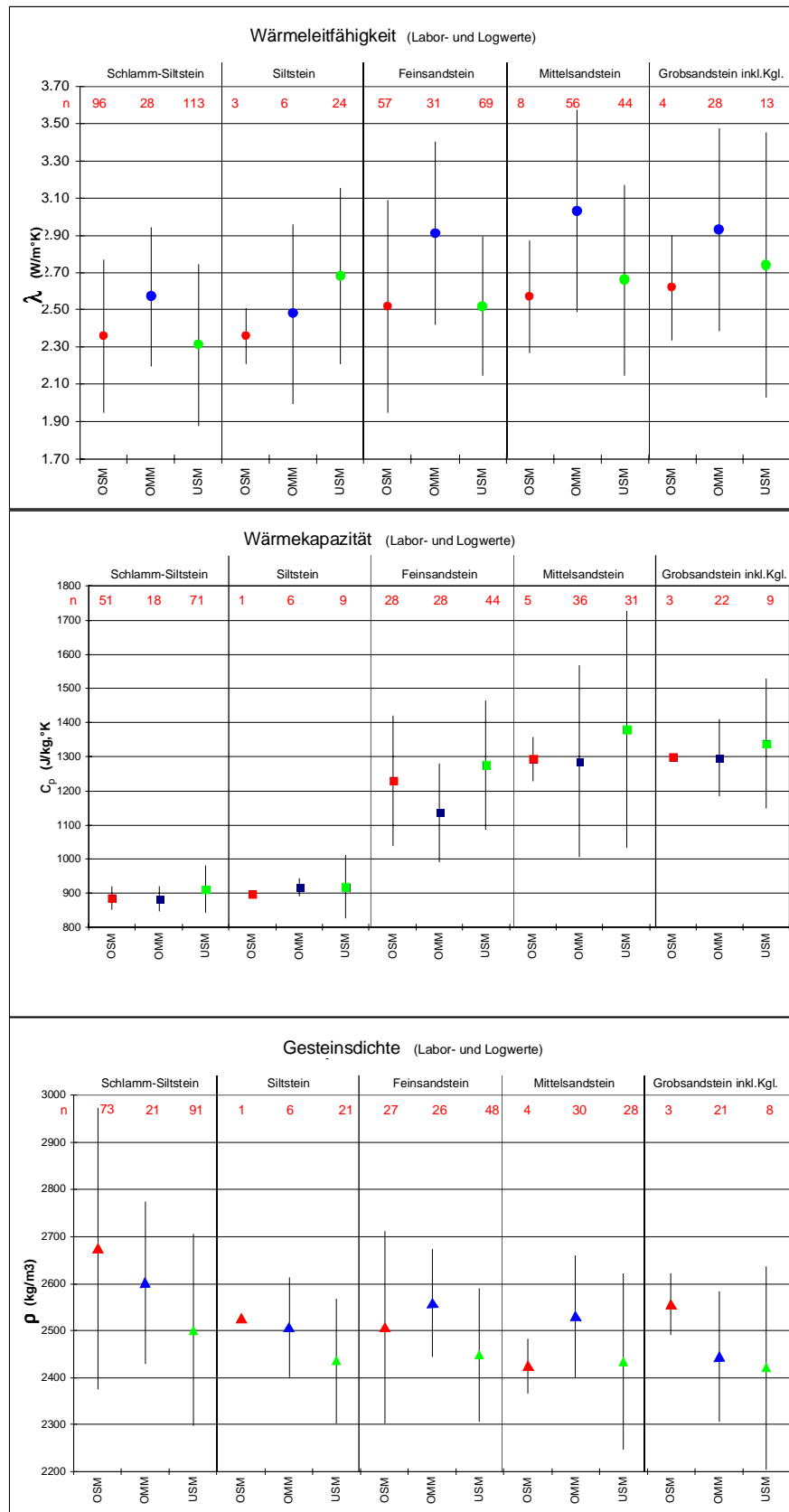


Abb 2-11: *Arithmetischer Mittelwert und Standardabweichung der Stoffwerte des Bodens im Molassebecken des schweizerischen Mittellandes nach Leu et al. (1999).*

3 Validation

3.1 Méthode de validation

Comme le programme EED (Hellström and Sanner, 1994), le programme EWS utilise la technique des fonction-g (Eskilson, 1987) pour déterminer la réponse thermique d'un ensemble de sondes géothermiques soumises à l'extraction ou à l'injection d'une puissance thermique. Le programme EED permet de calculer l'évolution de la température du fluide pour une extraction/injection de chaleur spécifiée par l'utilisateur. Le programme EWS permet aussi, contrairement à EED, de spécifier une évolution de la température du fluide caloporteur et de calculer la puissance thermique transférée par les sondes.

Les fonction-g sont utilisées pour établir la relation entre la température moyenne du fluide circulant dans les sondes et la puissance thermique transférée dans le terrain. Elles prennent en compte les effets à long terme dans le terrain et dépendent de la configuration des sondes, définie par leur nombre et leur disposition. Par exemple 4 sondes dont l'arrangement forme un carré représente une configuration. Exprimée avec des paramètres sans dimension, les fonction-g de cette configuration dépendent principalement de deux paramètres, t/t_s et r_b/H (Eskilson, 1987):

$$T_b(t) = T_m - \frac{q}{2\pi\lambda} g(t/t_s, r_b/H) \quad (3-1)$$

Avec: q : flux de chaleur constant transféré par la ou les sondes (W/m);

λ : conductibilité thermique du terrain (W/mK);

C : capacité thermique volumétrique du terrain (J/m³K);

t : temps comptabilisé à partir du moment où le flux de chaleur q devient non nul (s);

t_s : temps pour obtenir un régime stationnaire (s), défini par $t_s = H^2 / (9a)$;

a : diffusivité thermique du terrain (m²/s), définie par $a = \lambda / (c_{p_{\text{Erde}}} \rho_{\text{Erde}})$;

$T_b(t)$: température moyenne du terrain au rayon r_b mesuré depuis l'axe des sondes (°C);

T_m : température du terrain non perturbé (°C);

H : longueur active de la sonde (m);

B : espacement entre deux sondes (m);

$g(t/t_s, r_b/H)$: fonction-g (-).

Les fonction-g sont calculées pour $r_b/H = 0.0005$. Si le rayon du forage r_1 est différent, alors la relation suivante permet de passer du rayon r_b au rayon r_1 :

$$g(t/t_s, r_1/H) = g(t/t_s, r_b/H) - \ln(r_1/r_b) \quad (3-2)$$

Pour chaque configuration de sondes géothermiques, une fonction-g doit être calculée pour une valeur de l'espacement sans dimension B/H . Une gerbe de fonction-g permet, par interpolation, de couvrir toutes les possibilités offertes par une configuration (il suffit de calculer entre 5 et 10 fonction-g par configuration). Le programme SBM (Eskilson, 1986) est utilisé. Une sonde géothermique de longueur active H transfère de l'énergie thermique entre la profondeur D et $D+H$. Dans le cas d'une sonde unique, l'influence de la longueur D sur les

fonction-g est négligeable avec leurs longueurs usuelles (D: quelques mètres et H environ 100 m). Les fonction-g sont calculées en fixant le rapport D/H à 5/110.

Les calculs obtenus avec EWS sont comparés à TRNSBM (Pahud et al., 1996), qui n'est autre que SBM intégré dans TRNSYS (Klein et al., 1996), un programme de simulation dynamique de systèmes thermiques. La comparaison avec EED n'est possible que lorsque la puissance thermique est donnée en entrée au programme. Toutefois, une comparaison TRNSBM - EED permet de vérifier que l'utilisation de TRNSBM correspond bien aux conditions fixées pour le calcul des fonction-g relatives à une configuration de sondes géothermiques choisie.

3.2 Définition des cas de validation

3.2.1 Définition des paramètres de base

Les paramètres utilisés dans les calculs sont donnés dans la table ci-dessous. La plupart sont fixés pour correspondre à des caractéristiques typiques de terrain et à une sonde en double-U. Ces derniers seront constants pour tous les calculs effectués.

<i>Parameter</i>	<i>Wert</i>	<i>Parameter</i>	<i>Wert</i>
Sondenlänge	verschieden	Wärmeleitfähigkeit Erde λ_{Erde}	2.5 W/mK
Sondenanordnung / Anzahl	verschieden	Dichte ρ_{Erde}	2200 kg/m ³
Sondenabstand B	verschieden	spez. Wärmekapazität $c_{p\text{Erde}}$	1000 J/kgK
Sondendurchmesser innen D_i	26 mm	Wärmeleitfähigkeit Füllung λ_{Fill}	0.81 W/mK
Bohrdurchmesser D_b	120 mm	Dichte Füllung ρ_{Fill}	1180 kg/m ³
therm. Sondenwiderstand R_a	0.35 Km/W	spez. Wärmekap. Füllung $c_{p\text{Fill}}$	3040 J/kgK
therm. Bohrlochwiderstand R_b	0.1 Km/W	kin. Viskosität der Sole ν_{Sole}	4.15 mm ² /s
Solemassenstrom pro Sonde	0.139 kg/s	Wärmeleitfähigkeit Sole λ_{Sole}	0.5 W/mK
Mittlere Erdtemperatur oben	0 °C	Dichte Sole ρ_{Sole}	1053 kg/m ³
Temperaturgradient	0 °C/m	spez. Wärmekapazität $c_{p\text{Sole}}$	3875 J/kgK
Sonden-Rücklauftemperatur	verschieden	Entzugsleistung	verschieden

Tabelle 3-1: *Definition der Grundparameter für die Validierung*

3.2.2 Variation des paramètres:

Une vingtaine de situations (“runs”), numérotées de 1 à 20, sont définies de manière à tester l’implémentation des fonction-g dans EWS pour différentes configurations de sondes géothermiques. Les “runs” 1, 10 et 17 permettent une comparaison directe avec EED. La profondeur et l’espacement des sondes sont volontairement réduits, de manière à tendre plus rapidement vers un régime stationnaire, et ainsi tester le calcul des effets à long terme. Le critère de temps t_s , qui définit le temps nécessaire pour être proche du régime stationnaire, est calculé à environ 8 ans avec une longueur de sonde de 50 m, un espacement de 5 m et avec les propriétés thermiques du terrain données dans le tableau de la section précédente.

Run Nr.	Anzahl Sonden [-]	Sonden-länge H [m]	Sonden-Abstand B [m]	Simulations-Zeit [Jahre]	Entzugsleistung [W/m]	Sonden-Rücklauf [°C]	Entzugsprofil [h/a]	Sonden-Zeitkonst. t_s [Jahre]
1	1	50	-	60	-10	-	8760	7.8
2	1	50	-	60	-	10	8760	7.8
3	1	100	-	60	-	10	8760	31.0
4	1	200	-	60	-	10	8760	124.0
5	2	50	1.25	30	-	10	8760	7.8
6	2	50	2.5	30	-	10	8760	7.8
7	2	50	5	30	-	10	8760	7.8
8	2	50	10	30	-	10	8760	7.8
9	3x6	50	5	30	-	10	8760	7.8
10	10x5	50	5	30	-10	-	8760	7.8
11	10x5	50	5	30	-	10	8760	7.8
12	10x5	100	5	30	-	10	8760	31.0
13	10x5	200	5	30	-	10	8760	124.0
14	10x5	50	2.5	30	-	10	8760	7.8
15	10x5	50	10	30	-	10	8760	7.8
16	10x10	50	5	30	-	10	8760	7.8
17	10x5	50	5	15	-20	-	4380	7.8
18	10x5	50	5	15	50	-	168 im 1.Jahr	7.8
19	10x5	50	5	15	50	-	1680 im 1.Jahr	7.8
20	10x5	50	5	15	50	-	16800 total	7.8

Tabelle 3-2: Definition der Validierungsfälle

3.3 Résultats des validations

Run 1

Il s'agit d'une sonde unique avec extraction constante de chaleur, fixée à 10 W/m de sonde. La figure 3.1 montre l'évolution de la température moyenne du fluide en fonction du temps, sur une durée de 60 ans.

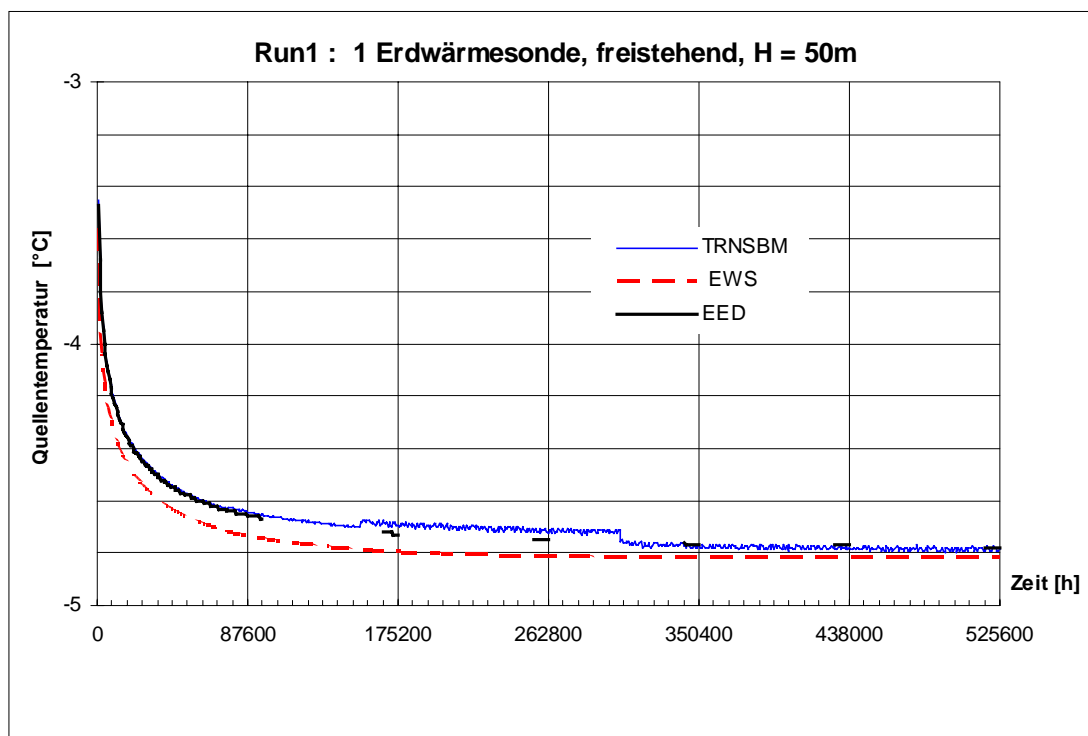


Figure 3.1 Run 1: évolution de la température moyenne du fluide dans un sonde géothermique de 50 m, soumise à une puissance d'extraction constante de 10 W/m, calculée avec les programmes TRNSBM, EWS et EED.

L'évolution de température obtenue avec TRNSBM est proche de celle calculée avec EED. Les irrégularités observées avec TRNSBM proviennent de la manière dont les conditions d'exploitation sont données dans la sonde. La température du fluide est ajustée par itérations successives jusqu'à ce que la puissance d'extraction corresponde à la valeur prescrite. Avec EED (ou SBM), c'est la puissance d'extraction qui est directement donnée comme condition au bord (la sonde).

L'évolution de température obtenue avec EWS diffère légèrement de celle obtenue avec EED ou TRNSBM. La différence est imputée aux valeurs de la fonction-g utilisée (voir le run 10 pour la mise en évidence de cet effet). Toutefois, l'erreur sur l'abaissement de la température du fluide est modeste, de l'ordre de 0.1 K pour un abaissement de température de 4 à 5 K (soit une erreur de 2 à 3%).

Run 2

Le cas n°2 est la même sonde que dans la cas n°1, mais avec des conditions d'exploitation différentes. Ce n'est plus la puissance d'extraction qui est imposée, mais le débit et la température d'entrée du fluide dans la sonde, fixés respectivement à 0.139 kg/s et à 10 °C. La figure 3.2 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 60 ans.

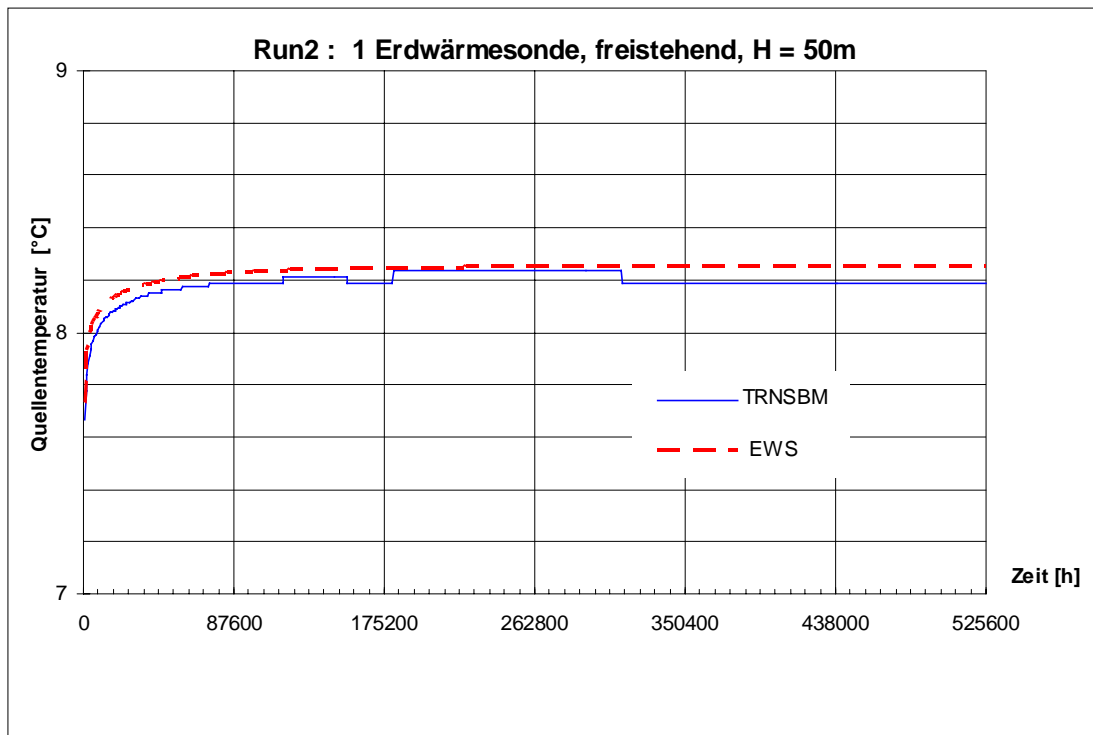


Figure 3.2 Run 2: évolution de la température du fluide à la sortie d'une sonde géothermique de 50 m, soumise à une température et un débit constant à l'entrée, calculée avec les programmes TRNSBM et EWS.

Compte tenu des imprécisions inhérentes aux calculs effectués avec TRNSBM (irrégularités de la courbe), les deux évolutions de température montrées dans la figure 3.2 sont très proches. L'écart de température maximum est inférieur à 0.1 K. La différence sur la puissance transférée est de l'ordre de 3% (0.06K/1.8K).

La simulation de 60 années a pris 9 minutes avec EWS et 38 minutes avec TRNSBM (PC avec processeur Pentium 130 MHz), soit 4 à 5 fois plus de temps. La durée de calcul avec TRNSBM augmente avec un nombre plus grand de forages (voir le cas 11 avec 50 forages).

Run 3

Le cas n°3 est comme le cas n°2, mais avec une sonde de 100 m au lieu de 50 m. Le débit et la température d'entrée du fluide dans la sonde sont fixés à respectivement 0.139 kg/s et 10°C. La figure 3.3 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 60 ans.

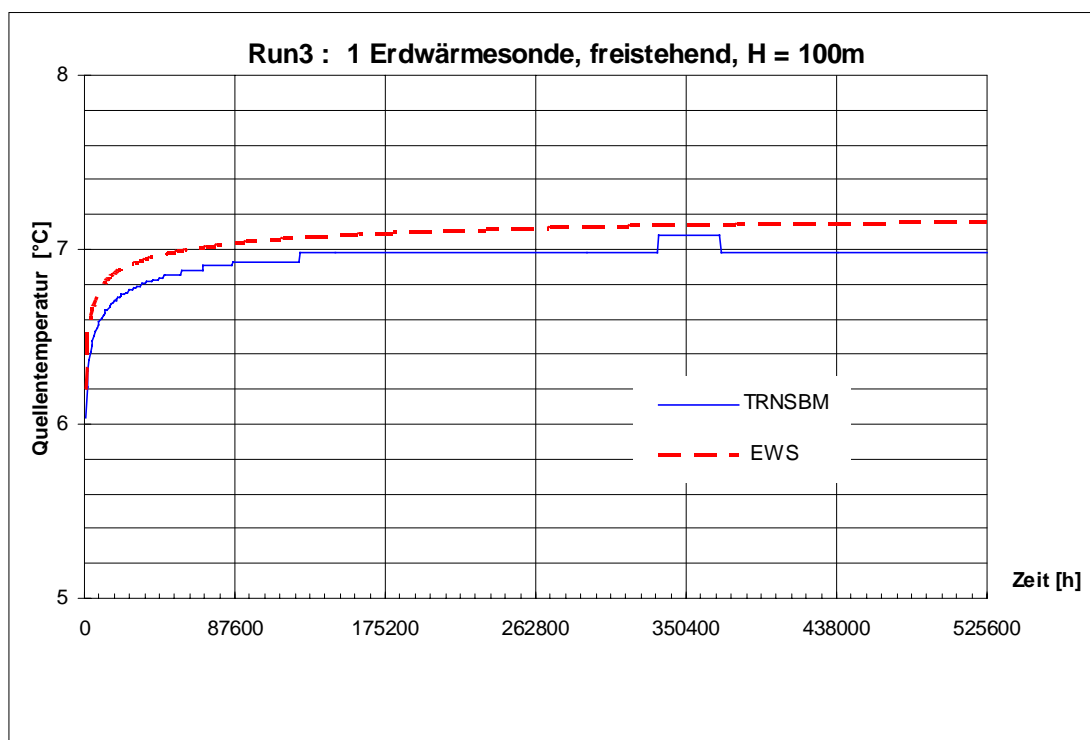


Figure 3.3 Run 3: évolution de la température du fluide à la sortie d'une sonde géothermique de 100 m, soumise à une température et un débit constant à l'entrée, calculée avec les programmes TRNSBM et EWS.

La température de sortie se stabilise à un niveau plus bas que dans le cas n°2, puisque la puissance injectée est plus grande (sonde deux fois plus longue) pour un débit de même valeur (0.139 kg/s).

Comme pour le cas n°2, les différences observées à long terme sont de l'ordre de grandeur des imprécisions inhérentes aux calculs effectués avec TRNSBM. L'écart de température maximum est de l'ordre de 0.2 K les premières années. La différence maximum sur la puissance transférée est alors de l'ordre de 6-7% (0.2K/3K). Ces différences sont également attribuées aux fonction-g utilisées dans EWS (voir cas n°10).

Run 4

Le cas n°4 est comme les cas n°2 et 3, mais avec une sonde de 200 m au lieu de 50 et 100 m. Le débit et la température d'entrée du fluide dans la sonde sont encore fixés à respectivement 0.139 kg/s et 10°C. La figure 3.4 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 60 ans.

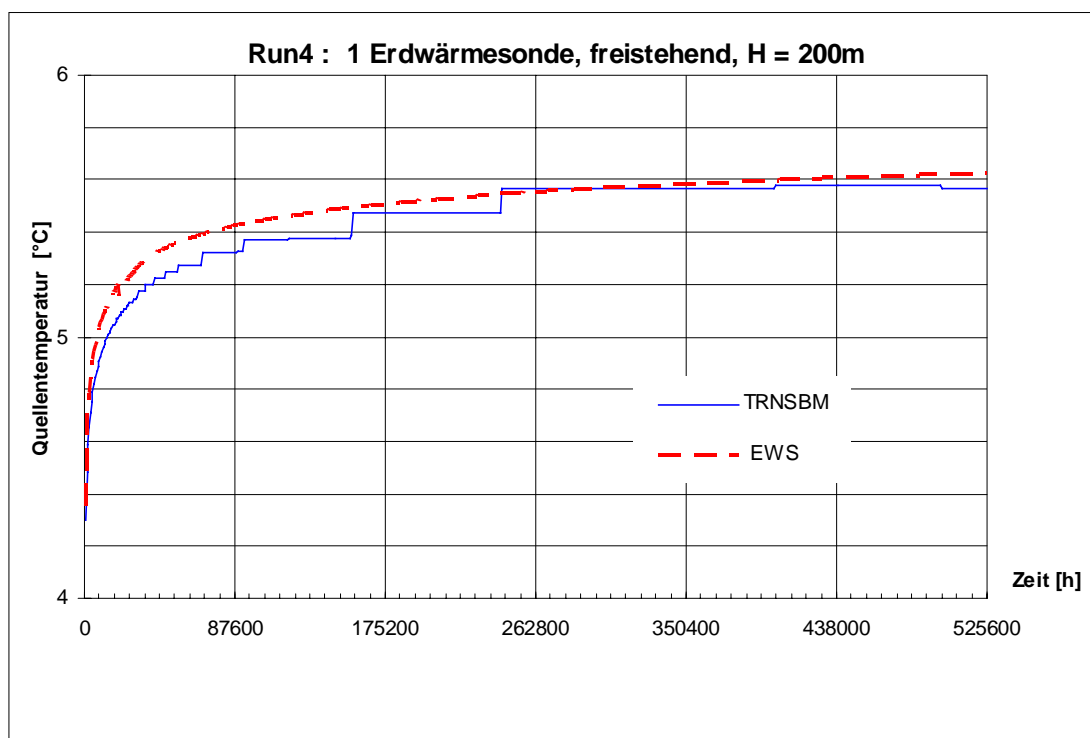


Figure 3.4 Run 4: évolution de la température du fluide à la sortie d'une sonde géothermique de 200 m, soumise à une température et un débit constant à l'entrée, calculée avec les programmes TRNSBM et EWS.

La température de sortie se stabilise à un niveau plus bas que dans le cas n°3, puisque la puissance injectée est plus grande (sonde deux fois plus longue) pour un débit de même valeur (0.139 kg/s).

Les différences observées à long terme sont inférieures aux imprécisions inhérentes aux calculs effectués avec TRNSBM. L'écart de température maximum est de l'ordre de 0.10 - 0.15 K les premières années. La différence maximum sur la puissance transférée est alors de l'ordre de 3% (0.13K/4.6K).

L'évolution de température calculée avec le programme EWS présente une petite dent de scie après 2 - 3 ans de simulation. Il provient du passage de la solution analytique de l'évolution des températures autour de la sonde à la solution donnée par la fonction-g. En effet, les fonction-g utilisées dans le programme EWS sont tirées de la publication d'Eskilson (1987). Ces dernières sont données pour les effets à long terme et ne comprennent pas le domaine des petits temps. Ce saut peut être supprimé en recalculant les fonction-g dans la totalité du domaine temporel, depuis les effets à court-terme jusqu'à l'établissement du régime stationnaire dans le terrain.

Run 5

Il s'agit de 2 sondes de 50 m de longueur et espacées par seulement 1.25 m. Le débit et la température d'entrée du fluide dans chacune des sondes sont fixés à respectivement 0.139 kg/s et 10°C. La figure 3.5 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 30 ans.

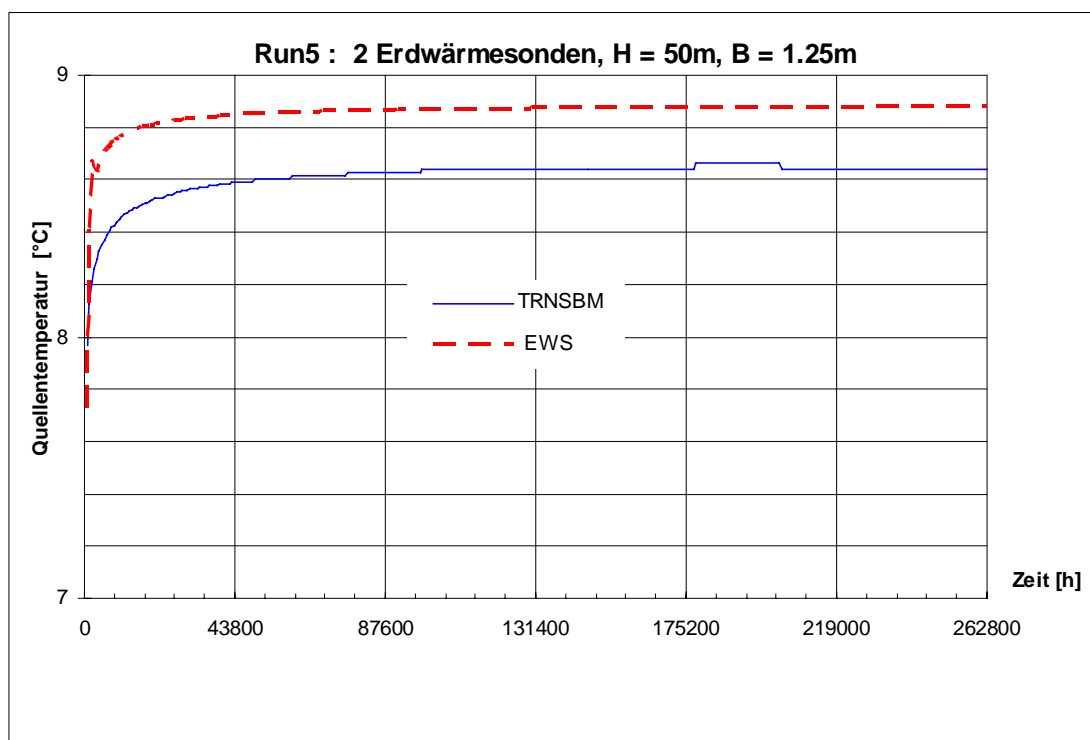


Figure 3.5 Run 5: évolution de la température du fluide à la sortie de 2 sondes géothermiques de 50m et espacées par 1.25m, calculée avec les programmes TRNSBM et EWS. La température du fluide à l'entrée des sondes et le débit sont constants.

L'écart de température entre les calculs effectués avec EWS et TRNSBM n'est pas négligeable, compte tenu de la différence de température entrée - sortie des sondes. L'écart atteint 0.4 K les premières années et se stabilise vers 0.3 K. L'erreur sur la puissance injectée est de l'ordre de 20% (0.3K/1.4K). Toutefois, la puissance d'injection calculée avec EWS est conservatrice, puisqu'elle est plus faible que celle calculée avec TRNSBM.

Les écarts importants observés sont certainement induits par l'extrapolation de la fonction-g en dehors du domaine donné par Eskilson. Pour cette configuration, Eskilson les a calculées pour un rapport B/H qui varie de 0.05 à l'infini. Un espacement B de 1.25 m et une longueur H de 50 m donne un rapport B/H de 0.025.

Run 6

Le cas n°6 est comme le cas n°5, mais avec un espacement entre les deux sondes de 2.5 m plutôt que 1.25 m. Le débit et la température d'entrée du fluide dans chacune des sondes sont fixés comme précédemment à respectivement 0.139 kg/s et 10°C. La figure 3.6 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 30 ans.

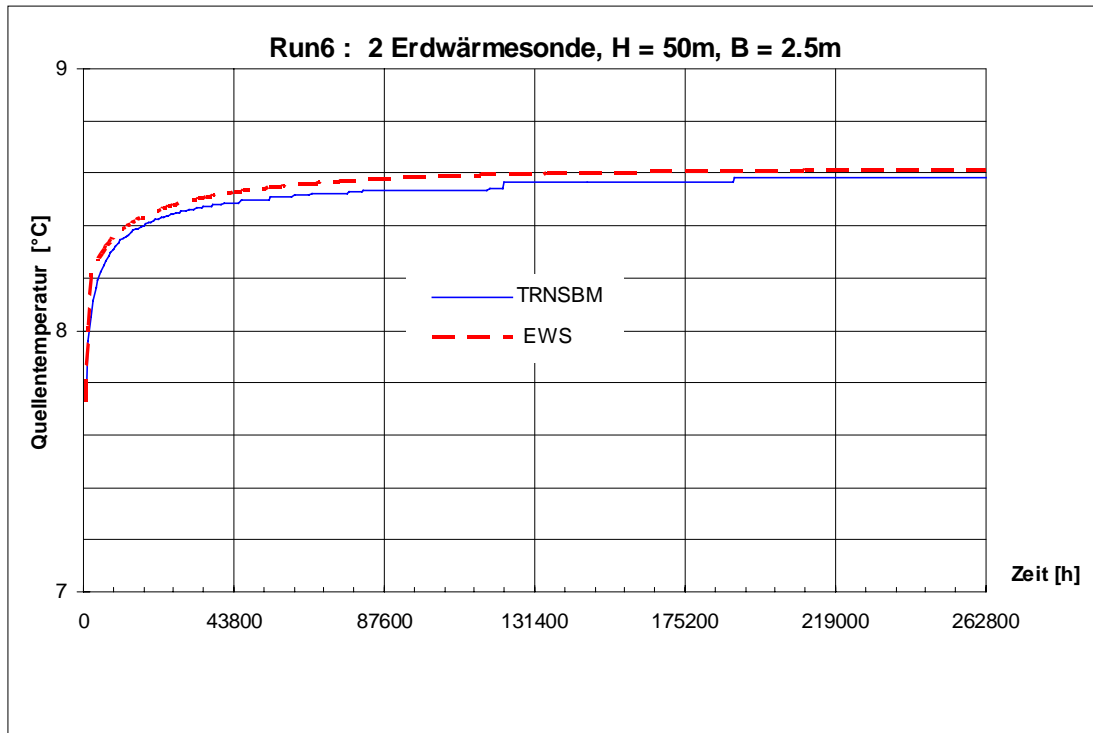


Figure 3.6 Run 6: évolution de la température du fluide à la sortie de 2 sondes géothermiques de 50 m et espacées par 2.5 m, calculée avec les programmes TRNSBM et EWS. La température du fluide à l'entrée des sondes et le débit sont constants.

Contrairement au cas précédent, la concordance entre les deux courbes de la figure 3.6 est tout à fait bonne. Le rapport B/H utilisé pour le calcul des fonction-g se trouve dans le domaine donné par Eskilson. L'écart de température entre les 2 courbes pour les effets à long terme sont de l'ordre de grandeur de la précision des calculs obtenus avec TRNSBM. L'écart de température maximum est d'environ 0.1 K la première année. L'erreur sur la puissance thermique transférée est alors de 5% (0.1K/2K).

Run 7

Le cas n°7 est comme les cas n°6, mais avec un espacement entre les deux sondes de 5 m plutôt que 2.5 m. Le débit et la température d'entrée du fluide dans chacune des sondes sont fixés comme précédemment à respectivement 0.139 kg/s et 10°C. La figure 3.7 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 30 ans.

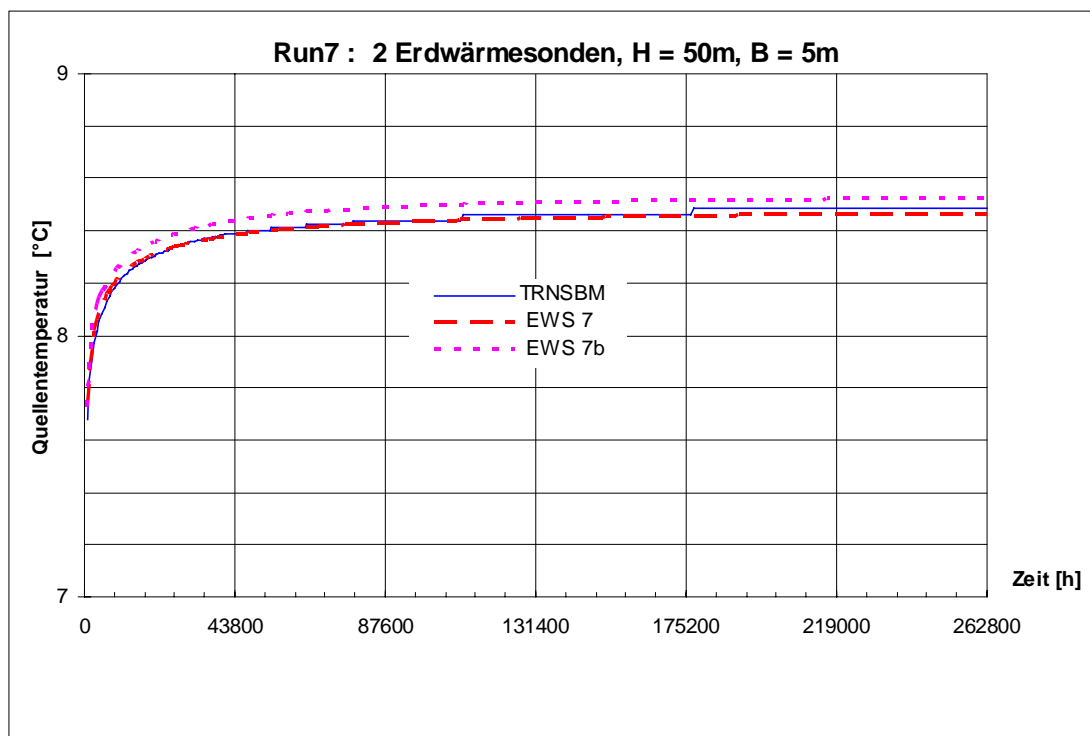


Figure 3.7 Run 7: évolution de la température du fluide à la sortie de 2 sondes géothermiques de 50 m et espacées par 5 m, calculée avec les programmes TRNSBM et EWS. La température du fluide à l'entrée des sondes et le débit sont constants. L'évolution de température calculée avec le programme EWS est montrée avec interpolation (EWS 7) et sans interpolation (EWS 7b).

Pour le cas 7 le rapport B/H vaut 0.1. Dans la figure 3.7, la courbe EWS 7 est calculée en extrapolant la fonction-g définie pour B/H=0.05 avec les relations données dans la section 2.4.4. La courbe EWS 7b est obtenue en prenant directement la fonction-g définie pour B/H=0.1.

La courbe calculée avec extrapolation par le programme EWS est la plus proche de celle calculée avec TRNSBM. L'écart de température est inférieur à 0.04 K. L'erreur sur la puissance thermique transférée est au plus de 2% (0.04K/2K).

Run 8

Le cas n°8 est comme les cas n°7, mais avec un espacement entre les deux sondes de 10 m plutôt que 5 m. Le débit et la température d'entrée du fluide dans chacune des sondes sont fixés comme précédemment à respectivement 0.139 kg/s et 10°C. La figure 3.8 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 30 ans.

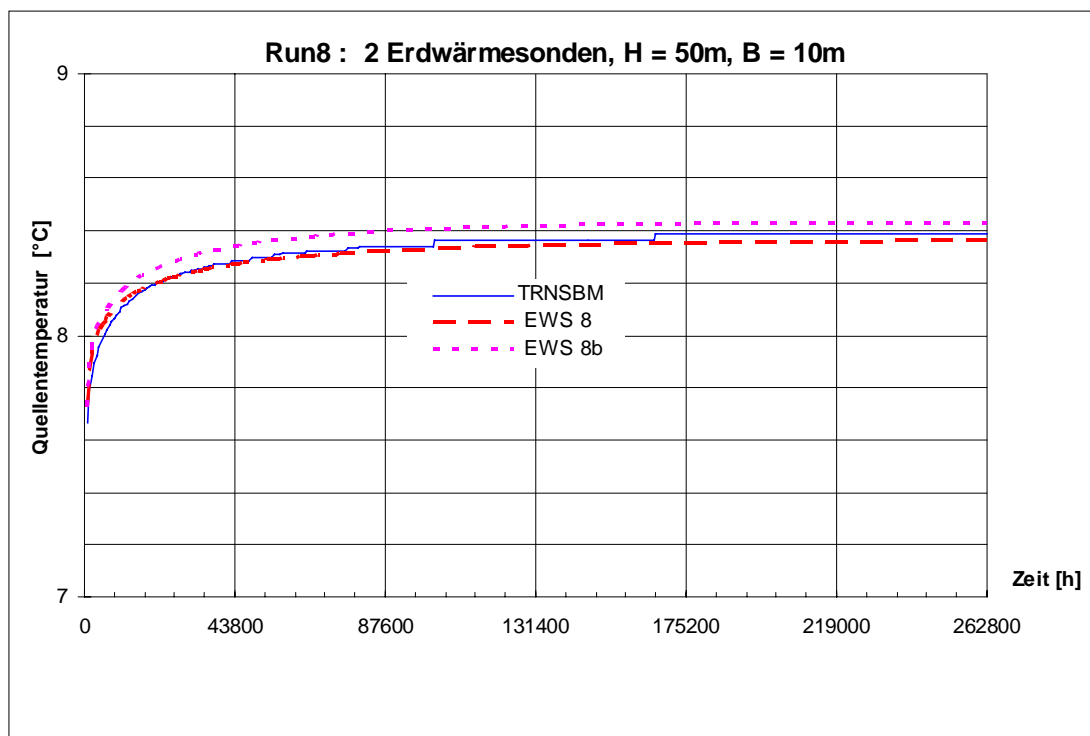


Figure 3.8 Run 8: évolution de la température du fluide à la sortie de 2 sondes géothermiques de 50 m et espacées par 10 m, calculée avec les programmes TRNSBM et EWS. La température du fluide à l'entrée des sondes et le débit sont constants. L'évolution de température calculée avec le programme EWS est montrée avec interpolation (EWS 8) et sans interpolation (EWS 8b).

Pour le cas 8 le rapport B/H vaut 0.2. Comme pour le cas 7, la courbe EWS 8 est calculée en extrapolant la fonction-g définie pour B/H=0.05. La courbe EWS 7b est obtenue en prenant directement la fonction-g définie pour B/H=0.2.

La courbe calculée avec extrapolation par le programme EWS est toujours la plus proche de celle calculée avec TRNSBM. L'écart de température est inférieur à 0.02 K après 2 ans. L'écart maximum est de 0.08 K la première année, et provient certainement de l'extrapolation de la fonction-g pour les petits temps. L'erreur sur la puissance thermique transférée est au plus de 4% (0.08K/2K). A partir de la deuxième année, l'erreur est de l'ordre de 1% (0.02K/1.6K).

Run 9

Le cas 9 est défini par 3 x 6 sondes de 50 m et espacées de 5 m. Il s'agit de 18 sondes disposées en 3 rangées de 6. L'espacement entre les rangées est égal à l'espacement entre les sondes (arrangement quadratique des sondes). Le débit et la température d'entrée du fluide dans chacune des 18 sondes sont fixés à respectivement 0.139 kg/s et 10°C. La figure 3.9 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 30 ans.

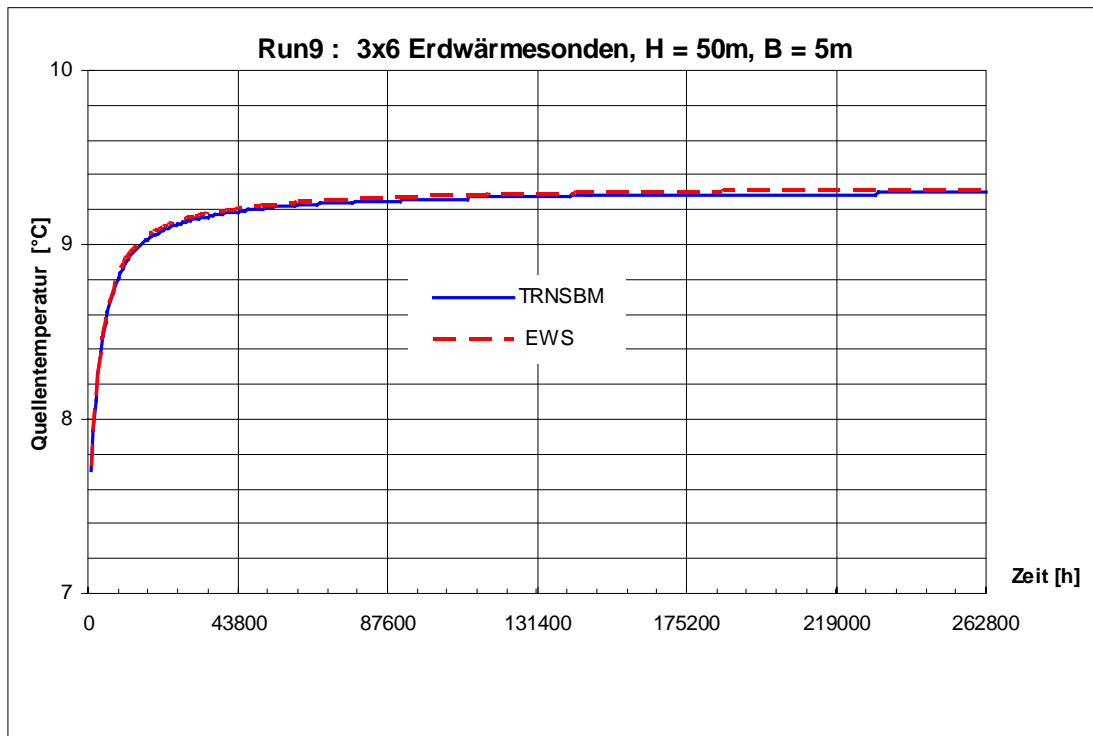


Figure 3.9 Run 9: évolution de la température du fluide à la sortie des 3x6 sondes géothermiques de 50 m et espacées par 5 m, calculée avec les programmes TRNSBM et EWS. La température du fluide à l'entrée des sondes et le débit sont constants.

La concordance entre les 2 courbes de la figure 3.9 est parfaite. Les écarts de température sont au plus de 0.02 K et ne peuvent être imputés qu'aux imprécisions de calculs de TRNSBM (évolution de la température en escalier).

Run 10

Il s'agit de 50 sondes arrangées en 10 rangées de 5, de longueur active 50 m et disposées de manière quadratique avec un espacement de 5 m. Une extraction constante de chaleur, fixée à 10 W/m de sonde, détermine les conditions d'exploitation. La figure 3.10 montre l'évolution de la température moyenne du fluide en fonction du temps, sur une durée de 30 ans.

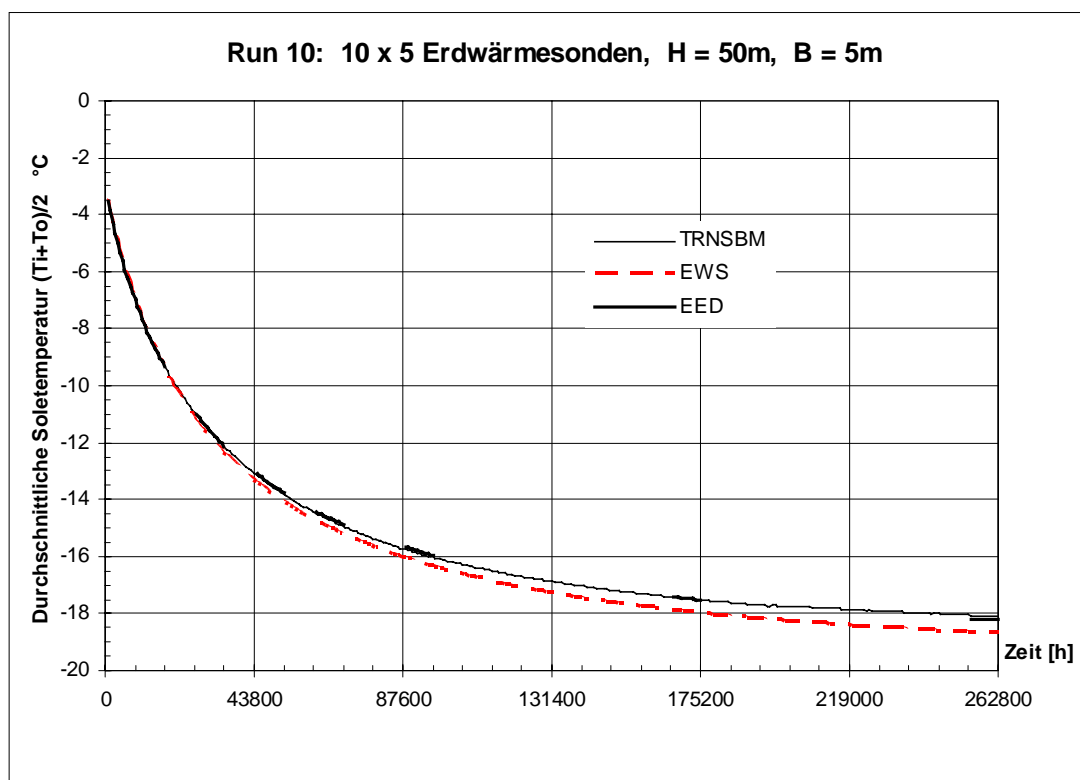


Figure 3.10 Run 10: évolution de la température moyenne du fluide dans 10x5 sondes géothermiques de 50 m et espacées de 5 m, calculée avec les programmes TRNSBM, EWS et EED. Les sondes sont soumises à une puissance d'extraction constante de 10 W/m.

L'évolution de la température obtenue avec TRNSBM est proche de celle calculée avec EED, ce qui valide un usage adéquat de TRNSBM.

L'évolution de température obtenue avec EWS diffère légèrement de celle obtenue avec EED ou TRNSBM. Cette différence s'explique par la fonction-g utilisée dans EWS qui diffère légèrement de celle utilisée dans EED, et aussi par l'erreur de reproduction de la fonction-g dans EWS (voir ci-après). En effet, la fonction-g utilisée dans EWS est basée sur des valeurs de la fonction-g publiée par Eskilson (1987) pour cette configuration. Par ailleurs, les fonctions-g du programme EED ont été recalculées pour élargir leur domaine de validité à des espacements plus petits entre les sondes. La différence provient, dans ce cas, de la résolution du maillage utilisé pour calculer la fonction-g. Lorsqu'un nombre important de sondes est simulé, la résolution du maillage doit être plus fine que celle qui serait utilisée pour un nombre plus restreint de sondes. Cet aspect a été mieux pris en compte lorsque les fonction-g ont été recalculées. Comme les résultats doivent être indépendants de la taille du maillage, un maillage trop grossier conduit à une surestimation de l'abaissement de température induit par

une extraction de chaleur, soit une surestimation de la fonction-g. Toutefois, la différence entre EWS et EED sur l'abaissement de température reste modeste, puisqu'elle ne dépasse pas 2 à 3% de l'abaissement calculé.

Les fonction-g sont généralement représentées en fonction de $\ln(t/t_s)$. La table suivante permet de comparer les valeurs de la fonction-g calculées par EED, TRNSBM et EWS. Les valeurs publiées par Eskilson sont également indiquées.

Ln(t/t _s)	temps t	fonction-g			
		EED	TRNSBM	EWS	ESKILSON
-4	52 jours	5.1	5.1	5.1	5
-2	≈ 1 an	10.4	10.4	10.2	10.2
-1	≈ 3 ans	15.9	16.0	16.2	16.3
0	≈ 8 ans	22.4	22.5	22.75	22.75
1	≈ 21 ans	26.8	26.9	27.5	27.3
1.35	≈ 30 ans	27.8	27.6	28.4	28.2
2	≈ 60 ans			29.2	29.5
3	≈ 160 ans			29.3	30

Tabelle 3-3: Vergleich der verwendeten g-funktionen

On peut remarquer que les valeurs des fonction-g calculées avec EED et TRNSBM deviennent inférieures à celles qui sont données par Eskilson lorsque $\ln(t/t_s)$ devient supérieur à -1. Ces différences restent malgré tout modestes, de l'ordre de 2 %. Elles peuvent facilement s'expliquer par un maillage moins fin dans les calculs d'Eskilson. Quant au programme EWS, l'interpolation des valeurs données par Eskilson introduit également une erreur modeste lorsque $\ln(t/t_s)$ est grand.

Les erreurs des fonction-g se répercutent sur la température du fluide calculée T_f . L'effet reste modeste et l'erreur sur l'abaissement de température est de 2 à 3 % après 30 ans (0.5 K pour un abaissement total de 18.2 K). La table suivante permet de comparer l'erreur sur la température induite par les fonction-g à la différence totale calculée.

Ln(t/t _s)	temps t	Erreur de temp. causée par fonction-g		Différence de température calculée	
		ESKILSON - EED	EWS - EED	EWS - EED	EWS - TRNSBM
-4	52 jours	0.09 K	0.02 K	0.06 K	0.06 K
-2	≈ 1 an	0.11 K	0.11 K	0.09 K	0.09 K
-1	≈ 3 ans	-0.24 K	-0.17 K	-0.21 K	-0.14 K
0	≈ 8 ans	-0.25 K	-0.25 K	-0.30 K	-0.24 K
1	≈ 21 ans	-0.32 K	-0.44 K	-0.49 K	-0.45 K
1.35	≈ 30 ans	-0.26 K	-0.38 K	-0.45 K	-0.57 K

Tabelle 3-4: Abweichung der Resultate

On s'aperçoit que les erreurs induites par les fonction-g suffisent à expliquer les différences de température observées. Une erreur de 0.1 sur la fonction-g se répercute, dans ce cas précis, avec une erreur de 0.06 K sur la température du fluide. On peut en déduire que l'implémentation des fonction-g dans le programme EWS est tout à fait satisfaisante.

Run 11

Le cas n°11 est comme le cas n°10 (10x5 sondes de 50 m et espacées de 5 m), mais avec des conditions d'exploitation différentes. Ce n'est plus la puissance d'extraction qui est imposée, mais le débit et la température d'entrée du fluide dans chaque sonde, fixés respectivement à 0.139 kg/s et à 10 °C. La figure 3.11 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 30 ans.

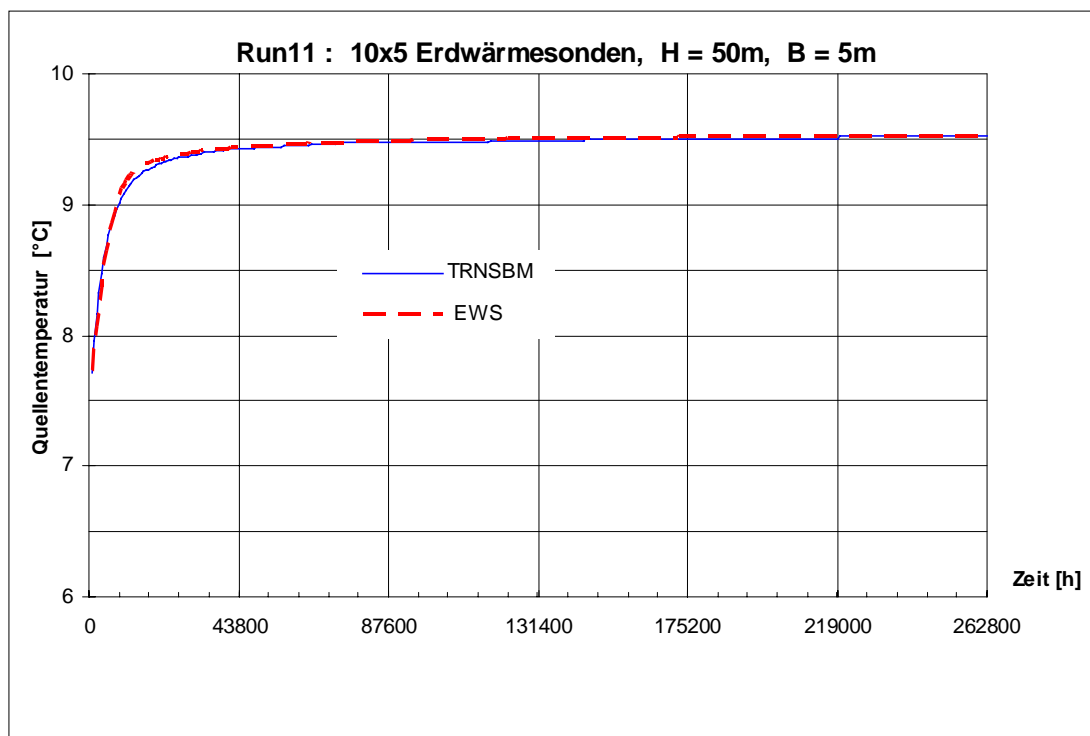


Figure 3.11 Run 11: évolution de la température du fluide à la sortie des 10x5 sondes géothermiques de 50 m et espacées de 5 m, calculée avec les programmes TRNSBM et EWS. Les sondes sont soumises à une température et un débit constant à l'entrée.

La correspondance entre les deux courbes de la figure 3.11 est très bonne, puisque les valeurs de la fonction-g utilisées dans le programme EWS ont été ajustées sur celles de EED, suite à l'analyse du cas 10. L'écart de température au delà de la troisième année est imputé aux imprécisions de calculs induites par TRNSBM (évolution de la température en escalier), ce qui confirme une très bonne concordance entre les résultats des calculs obtenus avec les deux programmes.

La simulation des 30 années a pris 7 - 8 minutes avec EWS et 8h34 minutes avec TRNSBM (PC avec processeur Pentium 130 MHz), soit environ 60 fois plus de temps.

Run 12

Le cas n°12 est comme le cas n°11, mais avec une longueur de sonde fixée à 100 m plutôt que 50 m. Le débit et la température d'entrée du fluide dans chaque sonde sont encore fixés à respectivement 0.139 kg/s et 10 °C. La figure 3.12 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 30 ans.

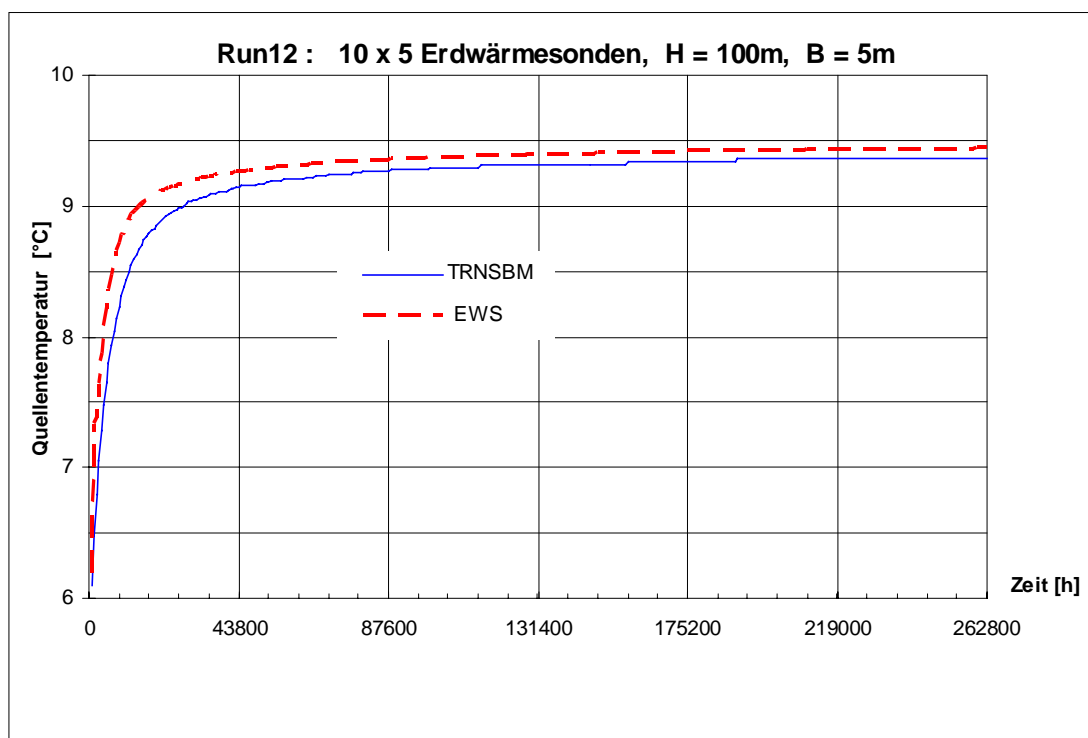


Figure 3.12 Run 12: évolution de la température du fluide à la sortie des 10x5 sondes géothermiques de 100 m et espacées de 5 m, calculée avec les programmes TRNSBM et EWS. Les sondes sont soumises à une température et un débit constant à l'entrée.

L'écart entre les températures calculées par EWS et TRNSBM est d'environ 0.5 K les deux premières années. L'erreur sur la puissance injectée peut atteindre 25% (0.5K/2K). Au delà de la dixième année, l'écart se réduit à environ 0.08 K. Toutefois, compte tenu de la faible différence de température entrée - sortie du fluide caloporteur, l'erreur sur la puissance thermique transférée par les sondes reste supérieure à 10% (0.08K/0.6K).

Les calculs effectués avec TRNSBM sont contrôlés en simulant une extraction de chaleur de 10 W/m de sonde. L'évolution de la température du fluide calculée est ensuite comparée à celle obtenue avec EED (cf. figure 3.13).

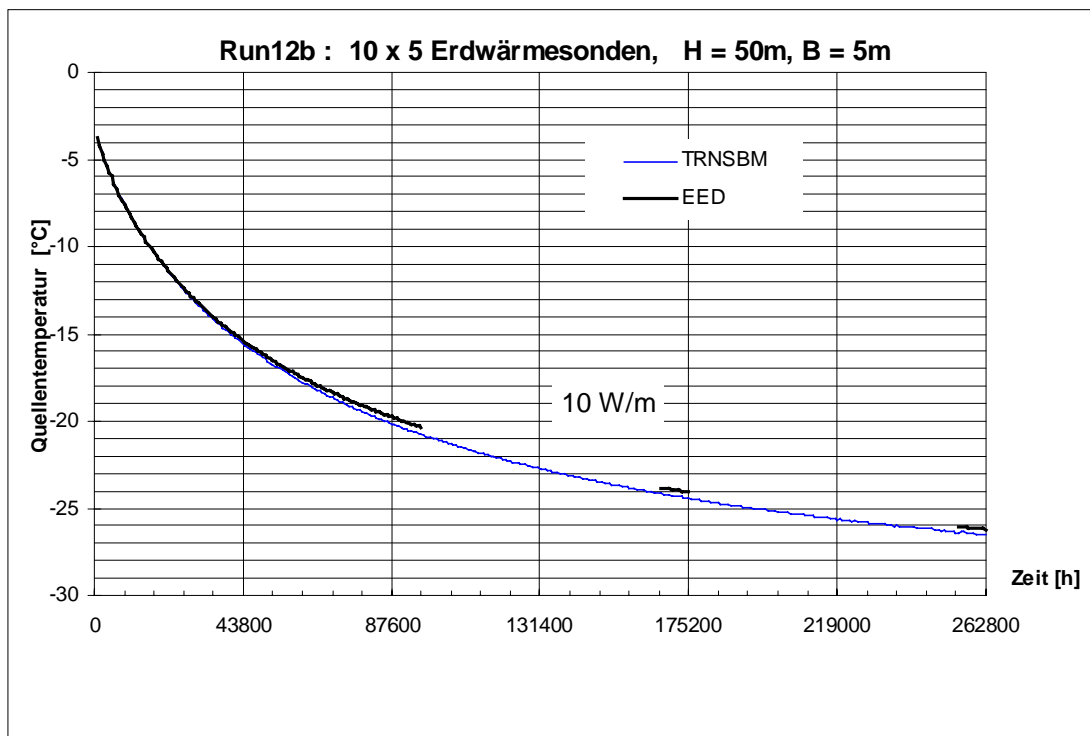


Figure 3.13 Run 12b: évolution de la température moyenne du fluide dans les 10x5 sondes géothermiques de 100 m et espacées de 5 m, calculée avec les programmes TRNSBM et EED. Les sondes sont soumises à une extraction de chaleur constante de 10 W/m.

La concordance entre TRNSBM et EED est satisfaisante. L'écart de température observé entre les deux courbes augmente avec le temps pour atteindre 0.5 K après 10 ans. Comme l'abaissement de température est de 20 K, cet écart de température représente une erreur de 2.5%. L'écart de température diminue ensuite légèrement, et vaut 0.4 K après 30 ans. L'abaissement de température est alors de 26 K, ce qui réduit l'erreur à 1.5%.

Les différences observées dans la figure 3.12, qui montre l'évolution de la température du fluide calculée avec TRNSBM et EWS, est certainement induite par l'extrapolation de la fonction-g en dehors du domaine donné par Eskilson. Pour cette configuration, Eskilson les a calculées pour un rapport B/H qui varie de 0.1 à l'infini. Un espacement B de 5 m et une longueur H de 100 m donne un rapport B/H de 0.05.

Run 13

Le cas n°13 est comme le cas n°12, mais avec une longueur de sonde fixée à 200 m plutôt que 100 m. Le débit et la température d'entrée du fluide dans chaque sonde sont encore fixés à respectivement 0.139 kg/s et 10 °C. La figure 3.14 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 30 ans.

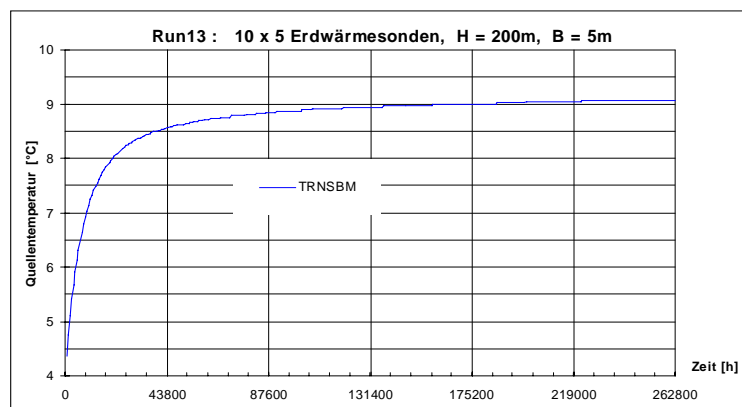


Figure 3.14 Run 13: évolution de la température du fluide à la sortie des 10x5 sondes géothermiques de 200 m et espacées de 5 m, calculée avec les programmes TRNSBM. Les sondes sont soumises à une température et un débit constant à l'entrée.

Le cas n°13 ne pouvait pas être calculer avec EWS, parce que les extrapolations des fonctions g dépassent les limites de la validité des formules 2-52 jusque 2-58. En effet, le rapport B/H est de 0.25, soit deux fois plus petit que celui du cas précédent, qui était déjà lui-même deux fois plus petit que le plus petit rapport utilisé par Eskilson pour le calcul des fonction-g.

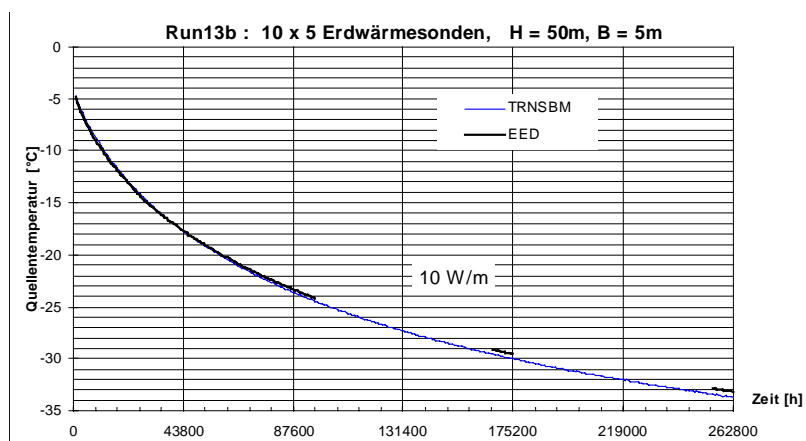


Figure 3.15 Run 13b: évolution de la température moyenne du fluide dans les 10x5 sondes géothermiques de 200 m et espacées de 5 m, calculée avec les programmes TRNSBM et EED. Les sondes sont soumises à une extraction de chaleur constante de 10 W/m.

Les calculs effectués avec TRNSBM sont contrôlés en simulant une extraction de chaleur de 10 W/m de sonde. L'évolution de la température du fluide calculée est ensuite comparée à celle obtenue avec EED (cf. figure 3.15). La concordance des courbes obtenues avec TRNSBM et EED est tout à fait satisfaisante.

Run 14

Le cas n°14 est comme le cas n°11, mais avec un espacement de 2.5 m plutôt que 5 m. Le débit et la température d'entrée du fluide dans chaque sonde sont toujours fixés à respectivement 0.139 kg/s et 10 °C. La figure 3.16 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 30 ans.

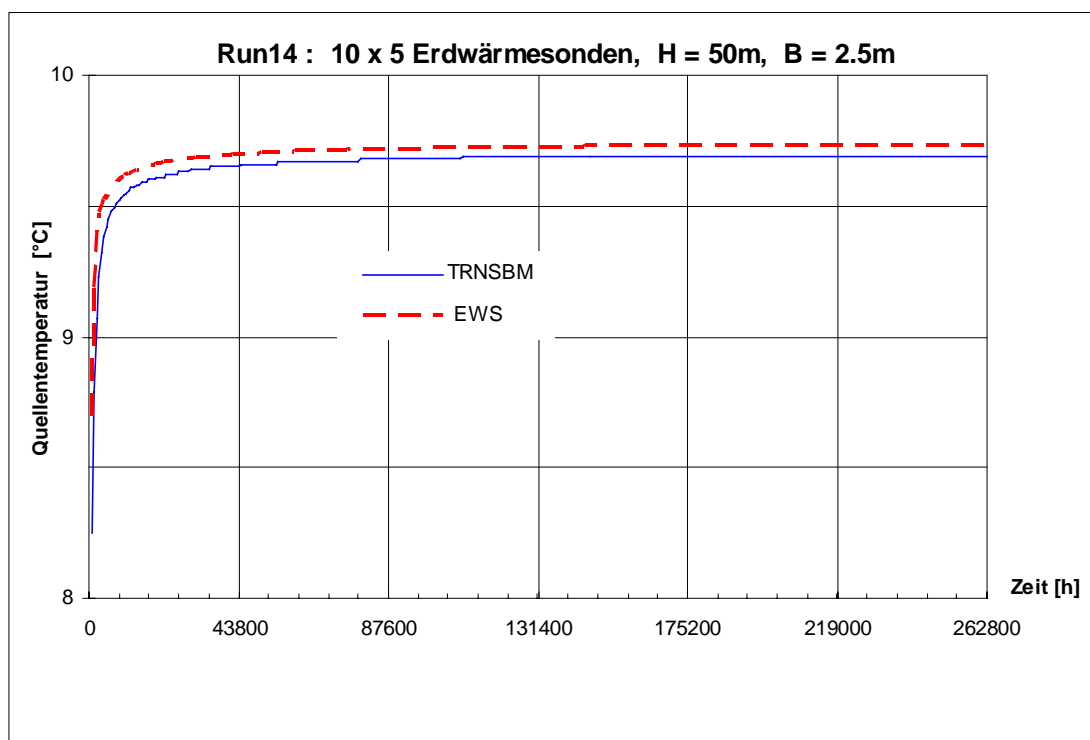


Figure 3.16 Run 14: évolution de la température du fluide à la sortie des 10x5 sondes géothermiques de 50 m et espacées de 2.5 m, calculée avec les programmes TRNSBM et EWS. Les sondes sont soumises à une température et un débit constant à l'entrée.

L'écart de température entre les courbes calculées avec TRNSBM et EWS est d'environ 0.5 K la première année, ce qui induit une erreur sur la puissance injectée d'environ 50% (0.5K/1K). L'écart se réduit avec le temps à 0.05 K, mais l'erreur sur la puissance injectée est de 17% (0.05K/0.3K). Ces différences non négligeable sont très certainement encore induite par l'extrapolation de la fonction-g en dehors du domaine donné par Eskilson. Le rapport B/H est de 0.05, soit deux fois plus petit que le plus petit rapport utilisé par Eskilson pour le calcul des fonction-g.

Run 15

Le cas n°15 est comme le cas n°11, mais avec un espacement de 10 m plutôt que 5 m. Le débit et la température d'entrée du fluide dans chaque sonde sont toujours fixés à respectivement 0.139 kg/s et 10 °C. La figure 3.17 montre l'évolution de la température du fluide à la sortie de la sonde en fonction du temps, sur une durée de 30 ans.

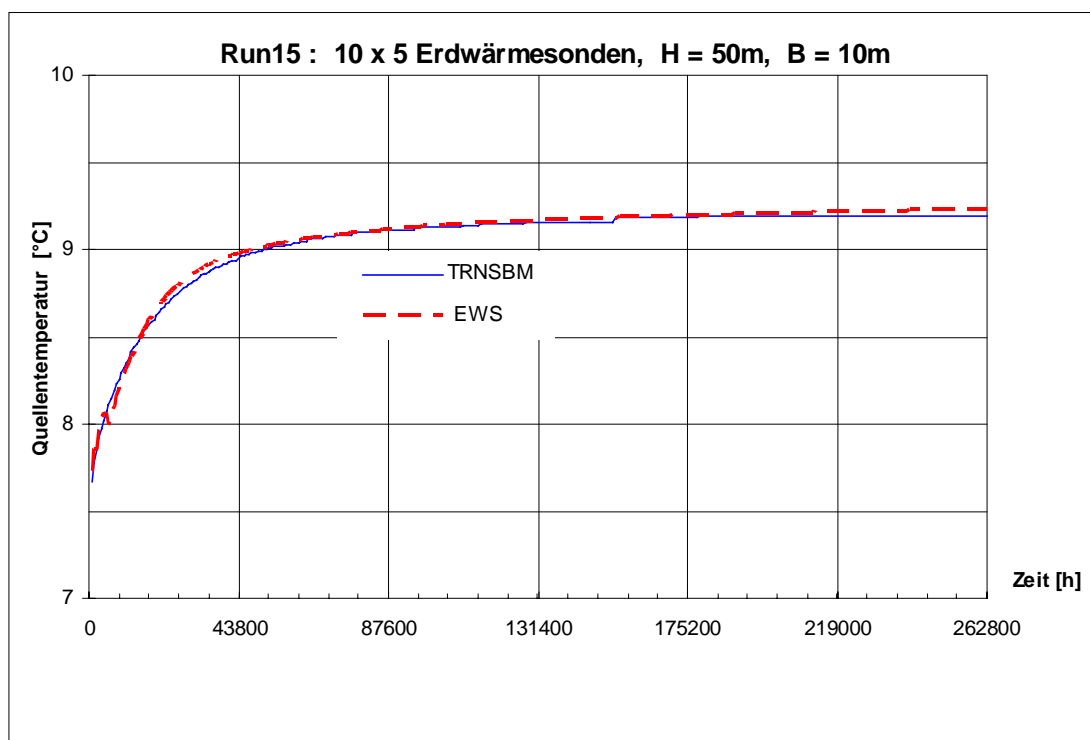


Figure 3.17 Run 15: évolution de la température du fluide à la sortie des 10x5 sondes géothermiques de 50 m et espacées de 10 m, calculée avec les programmes TRNSBM et EWS. Les sondes sont soumises à une température et un débit constant à l'entrée.

La correspondance entre les deux courbes calculées est tout à fait bonne. L'erreur sur la puissance thermique transférée due à l'écart de température entre les deux courbes est inférieure à 5%. Contrairement au cas précédent qui présentait des différences significatives entre les deux courbes calculées, le paramètre B/H est à l'intérieur du domaine donné par Eskilson pour le calcul des fonction-g.

La courbe calculée avec EWS montre encore une petite dent de scie, induite par l'interpolation de la fonction-g pour de petit temps (voir également run4).

Run 16

Il s'agit de 100 sondes arrangées en 10 rangées de 10, de longueur active 50 m et disposées de manière quadratique avec un espacement de 5 m. Le débit et la température d'entrée du fluide dans chaque sonde sont toujours fixés à respectivement 0.139 kg/s et 10 °C. La figure 3.18 montre l'évolution de la température moyenne du fluide en fonction du temps, sur une durée de 30 ans.

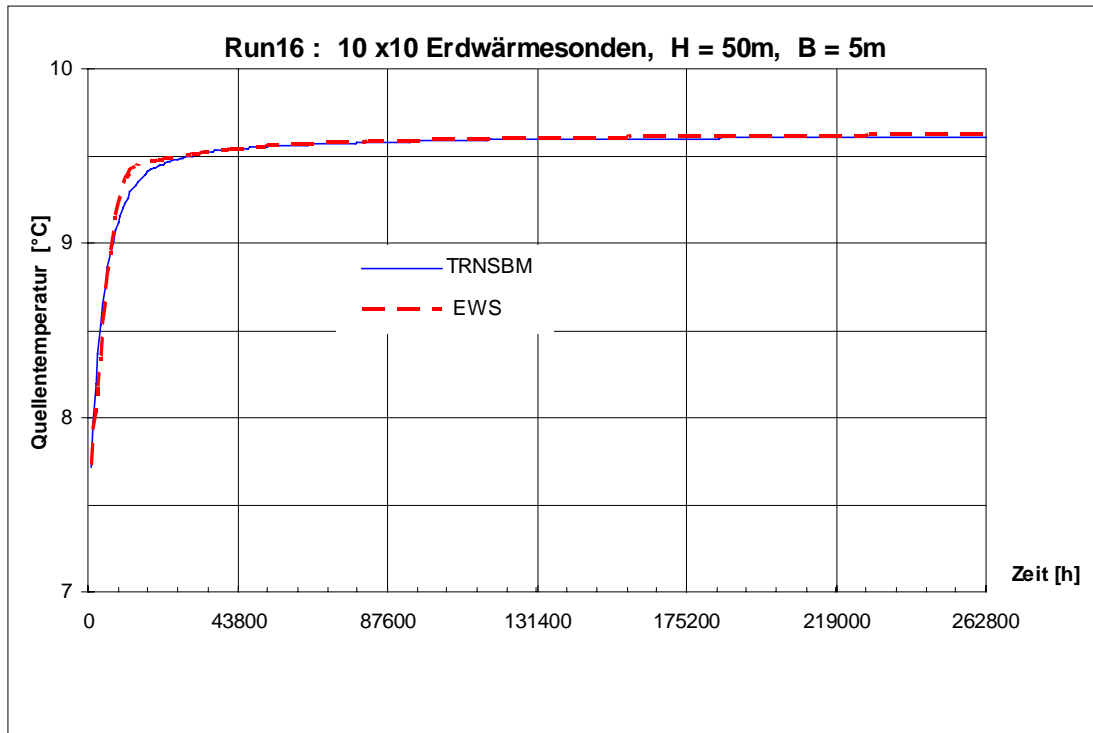


Figure 3.18 Run 16: évolution de la température du fluide à la sortie des 10x10 sondes géothermiques de 50 m et espacées de 5 m, calculée avec les programmes TRNSBM et EWS. Les sondes sont soumises à une température et un débit constant à l'entrée.

La correspondance entre les deux courbes est très bonne, à l'exception des 3 premières années. Les différences observées au cours de ces 3 ans sont certainement induites par la technique utilisée pour interpoler les fonction-g à de petits temps.

Run 17

Le cas n°17 est comme le cas n°10 (10x5 sondes de 50 m et espacées de 5 m), mais avec une puissance d'extraction cyclique. La puissance d'extraction est fixée à 20 W/m de sonde pendant 6 mois, puis elle est nulle les 6 mois suivants et ainsi de suite. En moyenne la puissance d'extraction est toujours de 10 W/m. La figure 3.19 montre l'évolution de la température moyenne du fluide dans les sondes en fonction du temps, sur une durée de 30 ans.

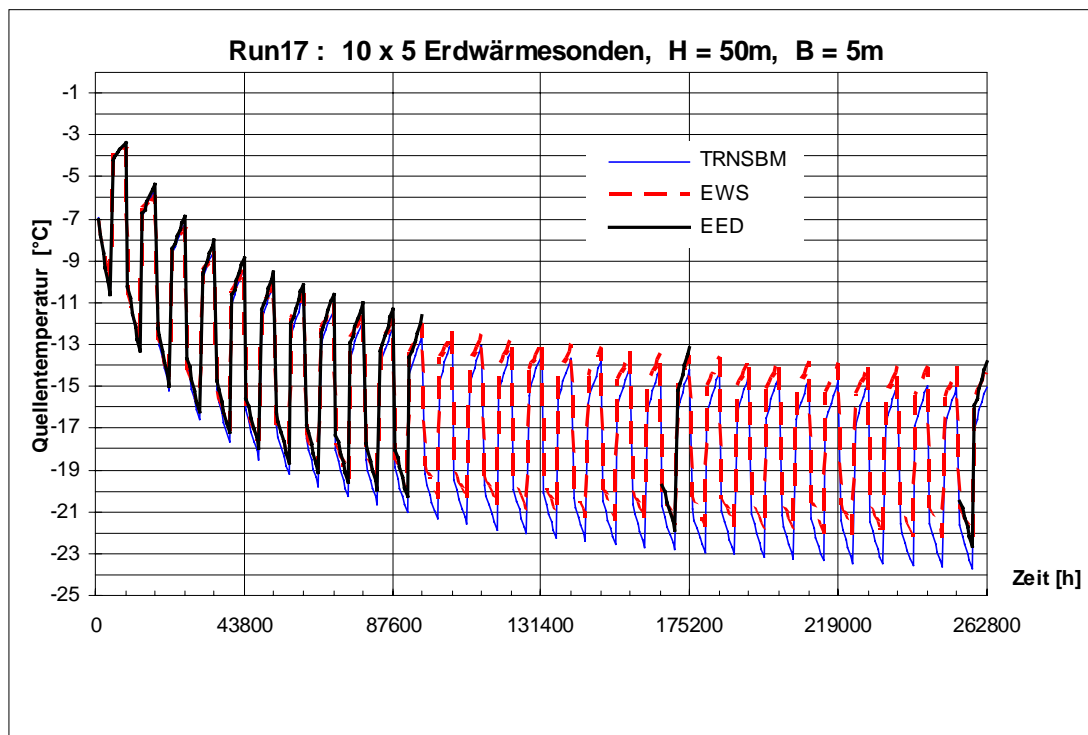


Figure 3.19 Run 17: évolution de la température moyenne du fluide dans 10x5 sondes géothermiques de 50 m et espacées de 5 m, calculée avec les programmes TRNSBM, EWS et EED. Les sondes sont soumises à une puissance d'extraction cyclique.

La courbe calculée avec le programme EWS correspond bien à celle obtenue avec EED, ce qui confirme que la technique de superposition de pulses constants est correctement implémentée dans EWS.

Il est étonnant d'observer la courbe calculée avec TRNSBM. Après une trentaine d'années les températures calculées avec TRNSBM sont 0.5 K plus basses que celles calculées avec EED. Lorsque la puissance d'extraction est constante, la différence entre TRNSBM et EED est beaucoup plus petite (voir le cas 10 pour une extraction de 10 W/m et la figure 3.20 pour une extraction de 20 W/m). Le principe de superposition doit être à l'origine de cet effet, en amplifiant des différences qui sont de signe opposés. Toutefois, l'erreur entre TRNSBM et EED reste modeste et se monte à seulement 2% de l'abaissement de température (0.5K/23K).

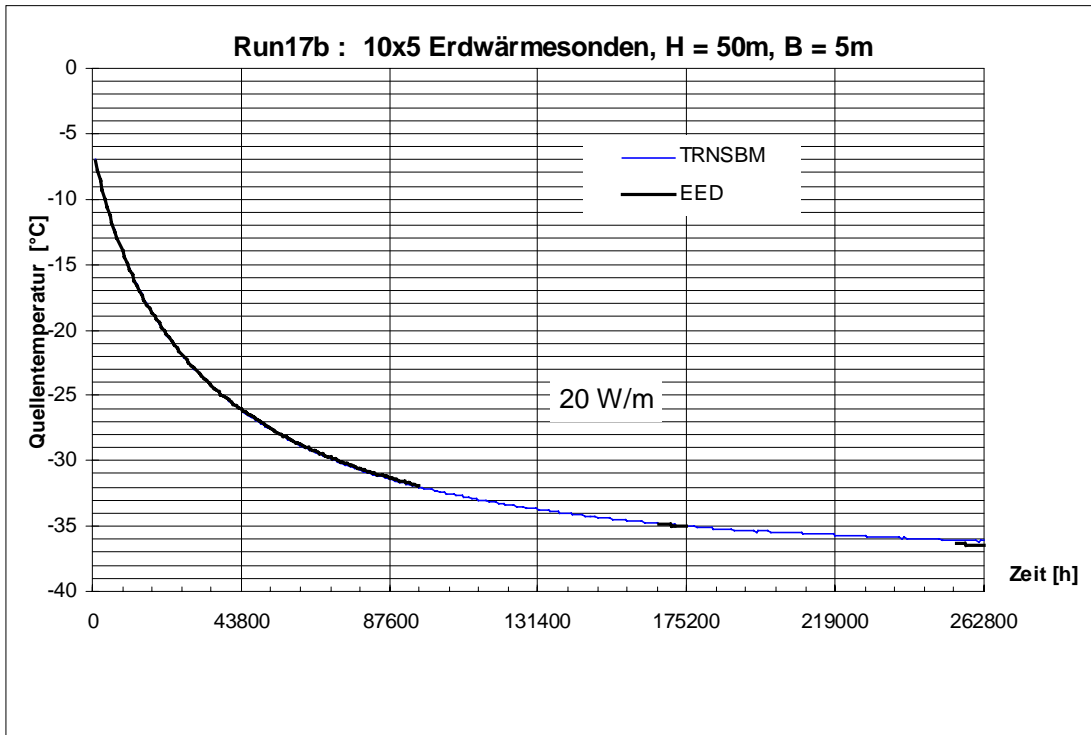


Figure 3.20 Run 17b: évolution de la température moyenne du fluide dans 10x5 sondes géothermiques de 50 m et espacées de 5 m, calculée avec les programmes TRNSBM et EED. Les sondes sont soumises à une puissance d'extraction de 20 W/m.

Run 18

Le cas n°18 utilise la configuration du cas n°10 (10x5 sondes de 50 m et espacées de 5 m). Une puissance d'extraction de 50 W/m de sonde est imposée pendant 168 heures, puis arrêtée. La température du fluide dans les sondes est calculée pendant une durée de 10 ans, et inclut la période de relaxation qui s'ensuit. La figure 3.21 montre l'évolution de la température moyenne du fluide dans les sondes.

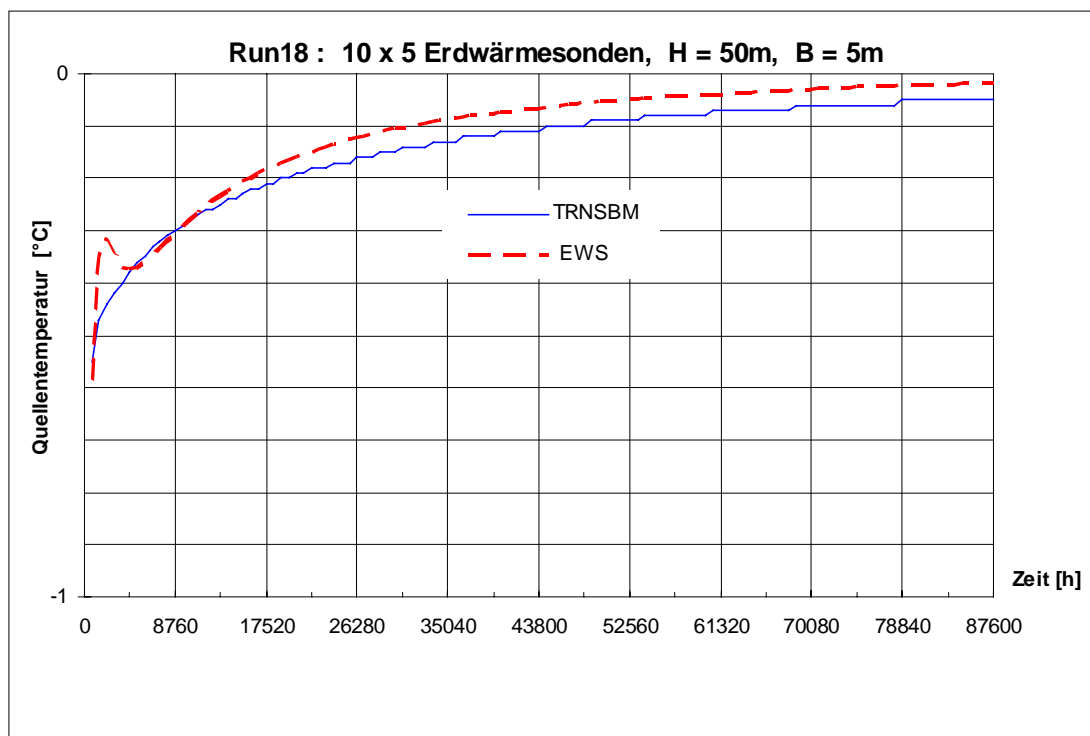


Figure 3.21 Run 18: évolution de la température moyenne du fluide dans 10x5 sondes géothermiques de 50 m et espacées de 5 m, calculée avec les programmes TRNSBM et EWS. Les sondes sont soumises à une puissance d'extraction de 50 W/m pendant 168 heures, puis une période de relaxation s'ensuit.

Mis à part la dent de scie que présente la courbe calculée avec le programme EWS, les allures des deux courbes sont similaires. On ne peut pas quantifier d'erreur sur la puissance thermique transférée, puisque cette dernière est nulle!

Run 19

Le cas n°19 utilise la configuration du cas n°10 (10x5 sondes de 50 m et espacées de 5 m). Une puissance d'extraction de 50 W/m de sonde est imposée pendant 1680 heures, puis arrêtée. La température du fluide dans les sondes est calculée pendant une durée de 10 ans, et inclut la période de relaxation qui s'ensuit. La figure 3.22 montre l'évolution de la température moyenne du fluide dans les sondes.

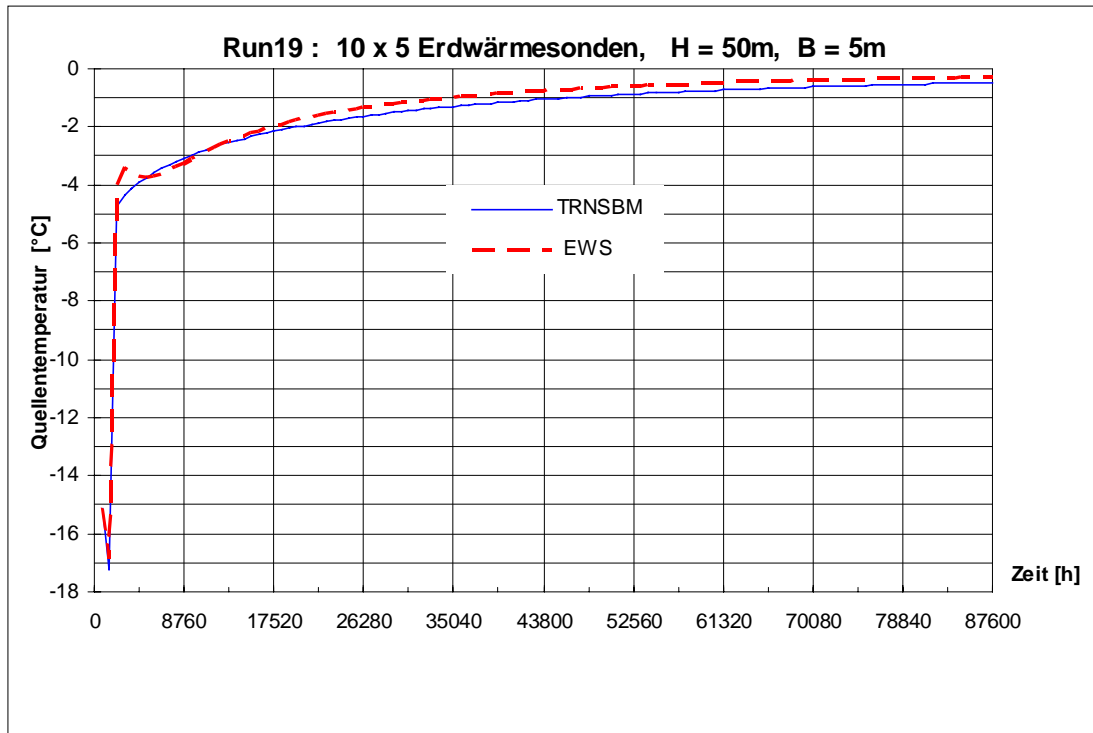


Figure 3.22 Run 19: évolution de la température moyenne du fluide dans 10x5 sondes géothermiques de 50 m et espacées de 5 m, calculée avec les programmes TRNSBM et EWS. Les sondes sont soumises à une puissance d'extraction de 50 W/m pendant 1680 heures, puis une période de relaxation s'ensuit.

La dent de scie de la courbe calculée avec le programme EWS est moins grande que dans le cas précédent. Les allures des deux courbes sont pareilles.

Run 20

Le cas n°20 utilise la configuration du cas n°10 (10x5 sondes de 50 m et espacées de 5 m). Une puissance d'extraction de 50 W/m de sonde est imposée pendant 16800 heures, puis arrêtée. La température du fluide dans les sondes est calculée pendant une durée de 10 ans, et inclut la période de relaxation qui s'ensuit. La figure 3.23 montre l'évolution de la température moyenne du fluide dans les sondes.

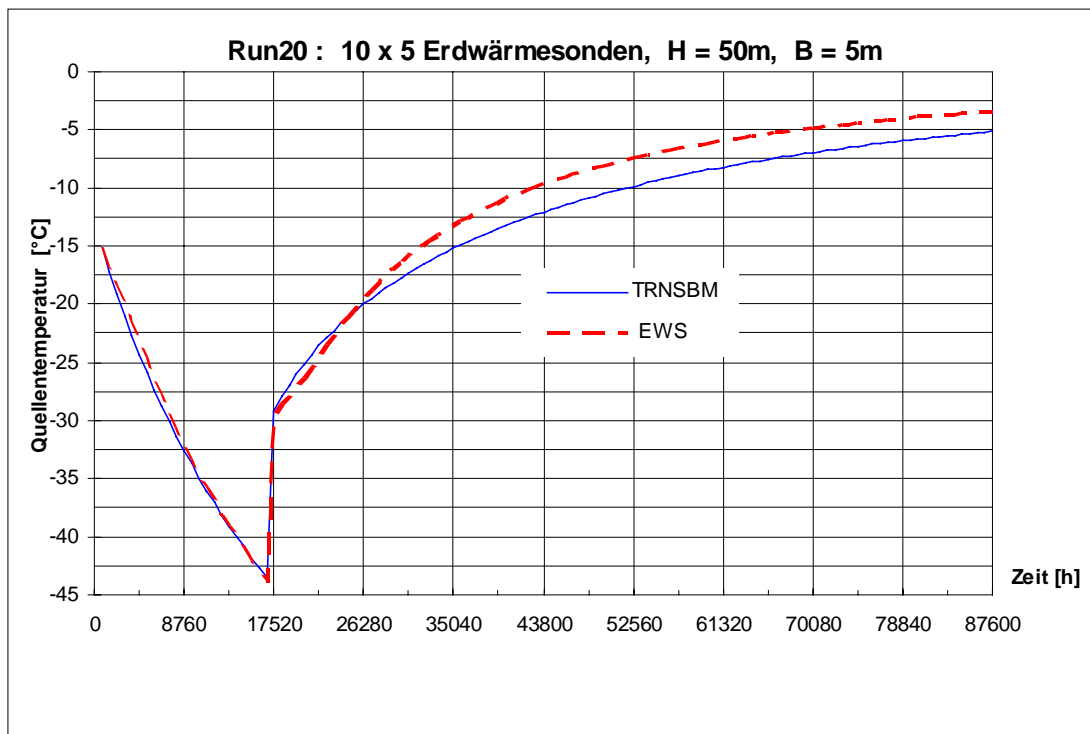


Figure 3.23 Run 20: évolution de la température moyenne du fluide dans 10x5 sondes géothermiques de 50 m et espacées de 5 m, calculée avec les programmes TRNSBM et EWS. Les sondes sont soumises à une puissance d'extraction de 50 W/m pendant 16800 heures, puis une période de relaxation s'ensuit.

La dent de scie de la courbe calculée avec le programme EWS n'apparaît plus. Toutefois, si la concordance des deux courbes est bonne en période d'extraction, elle l'est beaucoup moins durant la période de relaxation. Mais comme déjà mentionné auparavant, l'erreur sur la puissance transférée est nulle pendant la relaxation puisque le système est au repos!

3.4 Conclusion

EWS permet de calculer beaucoup plus rapidement que TRNSBM, surtout quand le nombre de sondes géothermiques est grand. Une trentaine d'années demande environ 7 minutes de calcul avec EWS lorsque une couche de terrain est définie (PC avec processeur Pentium 130 MHz). La durée de calcul avec le programme TRNSBM est 4 à 5 fois plus grande avec 2 sondes géothermiques et environ 60 fois plus grande avec 50 sondes.

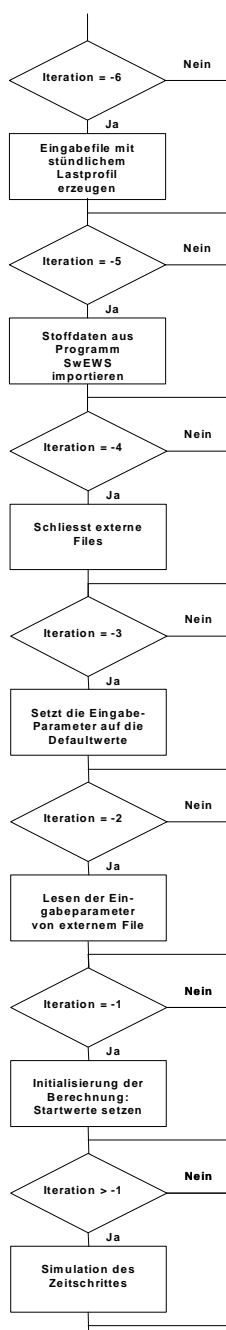
Dans l'ensemble, les comparaisons des résultats de calcul effectués avec EWS et TRNSBM sont très satisfaisantes. Les différences observées sont principalement dues aux valeurs des fonction-g utilisées dans le programme EWS, dont les valeurs ne sont pas exactement celles que l'on obtiendrait avec TRNSBM. (Dans le programme EWS, les fonction-g sont prises des publications d'Eskilson (1987)). Ces différences-là ne sont pas importantes, dans la mesure où les erreurs induites ne dépassent pas quelques pour-cent.

Des différences significatives sont observées lorsque les fonction-g sont extrapolées en-dehors du domaine des fonction-g pré-calculées. Plus précisément, lorsque le rapport B/H est trop petit ou lorsque le temps est trop petit. L'extrapolation des fonction-g devrait toujours se faire à l'intérieur du domaine des fonction-g pré-calculées. L'erreur observée pour les petit temps (lorsque $\ln(t/t_s)=4$) n'est pas additive. Elle disparaît dès que le temps devient plus grand. Dans l'état actuel du programme, le rapport B/H ne devrait pas être inférieur au plus petit rapport utilisé pour les fonction-g pré-définies. Le programme est donc adéquat pour le calcul de systèmes avec extraction de chaleur. En revanche, le calcul de stockage de chaleur n'est pas recommandé, puisque dans ces situations, l'espacement entre les sondes est généralement réduit (rapport B/H trop petit). Un message d'avertissement devrait être montré à l'utilisateur pour l'avertir quand il désire simuler un cas qui demande l'extrapolation des fonction-g en dehors de leur domaine de définition. A l'avenir, ce problème peut être résolu en recalculant les fonction-g de manière à couvrir tout le domaine de B/H et de $\ln(t/t_s)$. Il n'y aurait alors plus de problèmes liés à l'extrapolation des fonction-g.

4 Programmbeschreibung

4.1 Flussdiagramme

4.1.1 Modulaufruf aus dem Hauptprogramm



Das Erdwärmesondenmodul EWS wird mit dem Befehl

```
Erdsonde ( Iteration )
```

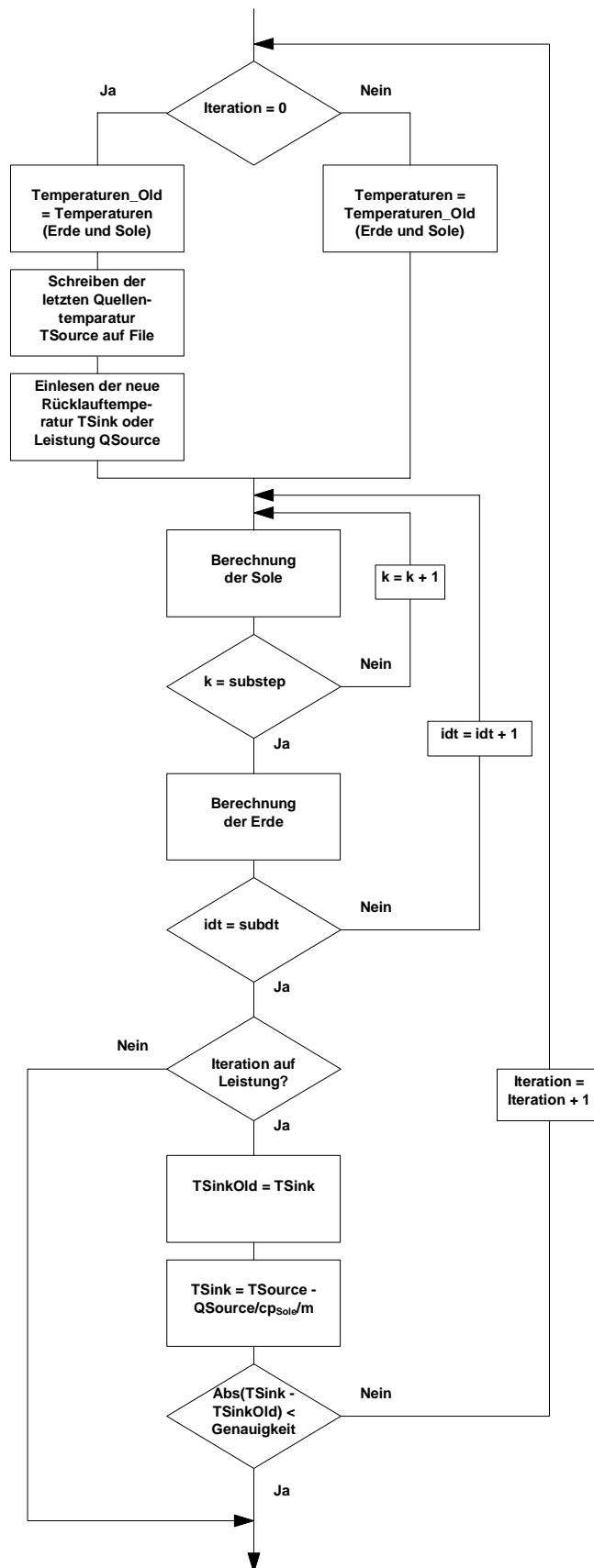
aufgerufen, wobei der Parameter *Iteration* angibt, ob wir uns in der Initialisierungsphase oder in der Simulationsphase des Programms befinden. Dabei kann *Iteration* die folgenden Werte annehmen:

- *Iteration* = -6 Erzeugt ein Eingabefile mit einem Lastprofil
- *Iteration* = -5 Importiert Stoffdaten aus dem Programm SwEWS
- *Iteration* = -4: Schliesst externe Eingabe- und Ausgabefiles
- *Iteration* = -3: Setzt alle Programmparameter auf einen Vorschlagswert (Defaultwert)
- *Iteration* = -2: Liest die wichtigsten Programmparameter aus einem externen Eingabefile
- *Iteration* = -1: Initialisiert die Simulation (Berechnung der Matrizen B, Formel 2-6) und setzt die Anfangswerte für die Simulation
- *Iteration* = 0: Berechnet den nächsten Zeitschritt. Erstmaliger Aufruf des Sondenmoduls während einer Iteration des Hauptprogramms für den aktuellen Zeitschritt.
- *Iteration* > 0: Das Hauptprogramm befindet sich in einer Iterationsschleife. Startwert für die Temperaturberechnung ist der letzte Iterationsschritt des vorherigen Zeitschrittes.

Der Parameter *Iteration* entspricht im TRNSYS-Modul dem Parameter *Info(7)*. Zur besseren Lesbarkeit sind im EWS-Modul die Konstanten *ErzeugeLastprofil* = -6, *Stoffdaten* = -5, *CloseFile* = -4, *Default* = -3,

OpenFile = -2 und *Initial* = -1 definiert. Ein Modulaufruf kann somit auch lauten: `Erdsonde(ErzeugeLastprofil)`, `Erdsonde(Stoffdaten)`, `Erdsonde(Default)`, `Erdsonde(OpenFile)`, `Erdsonde(Initial)` oder `Erdsonde(CloseFile)`.

4.1.2 Simulation des Zeitschrittes



Bei Erdwärmesonden gibt es prinzipiell zwei Betriebsarten:

1. Abkühlung der Quelltemperatur TSource mit einer vorgegebenen Entzugsleistung QSource [kW] in einer Wärmepumpe.
2. Erwärmung oder Abkühlung der Sonde in einem Prozess auf eine vorgegebene Temperatur TSink (z.B. Einsatz der Erdwärmesonde zur Kühlung eines Gebäudes).

Im vorliegenden EWS-Modul sind beide Möglichkeiten vorgesehen. Setzt man den Eingabeparameter *Leistung = true*, so wird auf diese Entzugsleistung hin iteriert, bei *Leistung = false* erfolgt keine interne Iteration, sondern es wird bei einer vorgegebenen Rücklauftemperatur TSink die entsprechende Quelltemperatur TSource berechnet. Selbstverständlich ist es auch möglich, *Leistung = false* zu setzen und die Iteration auf eine Entzugsleistung extern im Hauptprogramm zu realisieren. In diesem Fall muss die Iterationsvariable im ersten Iterationsaufruf im Hauptprogramm auf *Iteration = 0* gesetzt und bei jedem weiteren Iterationsaufruf um eins hinauf gezählt werden (*Iteration = Iteration + 1*).

Die Eingabe der Rücklauftemperatur TSink ist in jedem Fall notwendig, bei *Leistung = true* dient sie allerdings nur als 1. Schätzwert für die Iteration.

Im TRNSYS-Modul ist die Interne Iteration nicht vorgesehen, diese hat dort somit immer extern im Hauptprogramm zu erfolgen.

4.2 Parameter im EWS-Modul

4.2.1 Eingabeparameter

In der folgenden Parameterliste ist Iteration ein Wertparameter, alles andere sind Variablen-Parameter. Vektor ist ein array[0..DimAxi] mit Real-Werten. Die Parameter müssen im Hauptprogramm nicht mehr deklariert werden.

Simulationsparameter:

Iteration	integer [-]	Modul-Steuervariable, beschrieben in Kapitel 5.1
DimAxi	integer [-]	Anzahl Rechenknoten in axialer Richtung Typischer Wert: 1 - 4, maximal 10
DimRad	integer [-]	Anzahl Rechenknoten in radialer Richtung Typischer Wert: 6
Zeitschritt	integer [Min]	Zeitschritt für Erdwärmesonde aus Hauptprogramm Typischer Wert: 3 - 60 Minuten
subdt	integer [-]	Anzahl interner Unterteilungen für den Simulationszeitschritt des Erdreiches für die Crank-Nicholson-Berechnung, wobei $dt = \text{Zeitschritt} * 60 / \text{subdt}$ [s]. Ein Vorschlag für subdt wird bei der Initialisierung gemacht.
substep_run	integer [-]	Anzahl interner Unterteilungen für den Simulationszeitschritt der Sole, wenn die Solepumpe läuft. Der Solezeitschritt dt2 berechnet sich daraus mit $dt2 = dt / \text{substep_run}$ [s]. Ein Vorschlag für substep_run wird bei der Initialisierung gemacht.
substep_stop	integer [-]	Anzahl interner Unterteilungen für den Simulationszeitschritt der Sole, wenn die Solepumpe stillsteht. Der Solezeitschritt dt2 berechnet sich daraus mit $dt2 = dt / \text{substep_stop}$ [s]. Ein Vorschlag für substep_stop wird bei der Initialisierung gemacht.
RepRandbed	integer [Wochen]	Häufigkeit der Neuberechnung der äusseren Randbedingung mit der Trichterformel. Für die üblichen Fälle hat sich $\text{RepRandbed} = 1$ bewährt.
MonitorAxi	integer [-]	Ort des Monitorpunktes in axialer Richtung. Die Erdtemperatur des Monitorpunktes kann in das Ausgabefile hinaus geschrieben werden.
MonitorRad	integer [-]	Ort des Monitorpunktes in radialer Richtung.
numrows	longint [-]	Anzahl der Zeitschritte auf dem Eingabefile.
Jahr	integer [Jahre]	Aktuelles Simulationsjahr. Wenn die Zählvariable $\text{simstep} = \text{numrows}$ ist, dann wird intern die Variable Jahr um eins erhöht und $\text{simstep} = 0$

			gesetzt. Die Variable Jahr kann als Abbruchkriterium für die Simulation verwendet werden.
LastYear	integer	[Jahre]	Anzahl Simulationsjahre. Sofern die Starttemperatur nicht mit <i>Starttemp = true</i> abgeschätzt wird, ist die Anzahl Simulationsjahre durch den Speicherplatz und die Konstante <i>MaxJahre</i> im Programmkopf begrenzt. Werden die Startbedingungen mit <i>Starttemp = true</i> abgeschätzt, so kann <i>LastYear</i> beliebig gross sein. Die Simulation wird üblicherweise im Hauptprogramm mit der folgenden Anweisung abgebrochen: <pre> LastYear := 5; simstep := 0; repeat Simstep := Simstep + 1; Erdsonde(Iteration); until Jahr>Lastyear; </pre>
Rechenradius	real	[m]	Äusserer Rand des Simulationsgebietes und Ort, an dem mit der Trichterformel die Temperaturrandbedingung berechnet wird. Typischer Wert: 1.5 - 2 m.
Gitterfaktor	real	[-]	Faktor zur Bestimmung der Rechenvolumen. Gleichung 2-7. Typischer Wert: 0.7 - 2.0
Sicherheit1	real	[-]	Numerischer Sicherheitsfaktor zur Berechnung des internen Zeitschrittes dt2 für die Soleberechnung. Je grösser Sicherheit1 ist, um so kleiner wird der Simulationszeitschritt dt2. Empfohlener Wert: 4.0
Sicherheit2	real	[-]	Numerischer Sicherheitsfaktor zur Berechnung der internen Zeitschritte dt für die Berechnung der Erdtemperaturen. Je grösser Sicherheit2 ist, um so kleiner wird der Simulationszeitschritt dt. Empfohlener Wert: 2.0.
Genauigkeit	real	[°C]	Abbruchkriterium (Genauigkeit von <i>TSink</i>) für die Iteration der Entzugsleistung. Wird nur benötigt, falls <i>Leistung = true</i> gesetzt ist, d.h. wird intern auf eine vorgegebene Leistung iteriert. Typischer Wert: 0.05 °C
adiabat	real	[-]	Mit <i>adiabat=1.0</i> ist kein Nachströmen von Wärme an den äusseren Rändern des Simulationsgebietes mehr möglich, mit <i>adiabat=0.0</i> wird das Nachströmen nicht behindert und der Wärmefluss berechnet sich nach den g-functions oder der Werner-Formel. Üblicher Wert: adiabat = 0.0

Stoffwerte:

rhoFill	real	[kg/m ³]	Dichte der Hinterfüllung Typischer Wert: 3'000 kg/m ³
cpFill	real	[J/kgK]	spezifische Wärmekapazität der Hinterfüllung Typischer Wert: 1'200 J/kgK
lambdaFill	real	[W/mK]	Wärmeleitfähigkeit der Hinterfüllung Typischer Wert: 0.81 W/mK
rhoSole	real	[kg/m ³]	Dichte der Sole Typischer Wert: 1'050 kg/m ³ Wasser: 1'000 kg/m ³
cpSole	real	[J/kgK]	spezifische Wärmekapazität der Sole Typischer Wert: 3'800 J/kgK Wasser: 4'210 J/kgK
lambdaSole	real	[W/mK]	Wärmeleitfähigkeit der Sole Typischer Wert: 0.5 W/mK Wasser bei 5°C: 0.57 W/mK
nueSole	real	[m ² /s]	kinematische Viskosität der Sole Typischer Wert: 4 - 6 10 ⁻⁶ m ² /s Wasser: 1.8 10 ⁻⁶ m ² /s
cpErde	Vektor	[J/kgK]	Spezifische Wärmekapazität der Erde der jeweiligen axialen Schicht (von oben nach unten). Es kann für jede axiale Schicht, entsprechend der jeweiligen Gesteinsart, ein anderer Wert eingegeben werden. Typischer Wert: 800 - 1'200 J/kgK
rhoErde	Vektor	[kg/m ³]	Dichte der Erde der jeweiligen axialen Schicht (von oben nach unten). Es kann für jede axiale Schicht, entsprechend der jeweiligen Gesteinsart, ein anderer Wert eingegeben werden. Typischer Wert: 2'000 - 2'500 kg/m ³
lambdaErde	Vektor	[W/mK]	Spezifische Wärmekapazität der Erde der jeweiligen axialen Schicht (von oben nach unten). Es kann für jede axiale Schicht, entsprechend der jeweiligen Gesteinsart, ein anderer Wert eingegeben werden. Typischer Wert: 1.5 - 3 W/mK

Sondeneigenschaften:

R1	real	[K/W]	Thermischer Widerstand zwischen Sole und erstem axialen Rechenknoten. Wird $R1=0$ gesetzt, so verwendet das Programm die Formel 2-16 für R_1 .
Ra	real	[Km/W]	„internal thermal resistance“: Widerstand zwischen hinauf und hinunter strömender Sole, multipliziert mit Sondenlänge dl. Wird $Ra=0$ gesetzt, so verwendet das Programm Formel 2-17 für R_2 . Typischer Wert: 0.1 - 0.15 Km/W
Rb	real	[Km/W]	„thermal borehole resistance“: Thermischer Widerstand zwischen Sole und Bohrradius r_1 , multipliziert mit Sondenlänge dl. Wird $Rb=0$ gesetzt, so verwendet das Programm Formel 2-16 für R_1 . Typischer Wert: 0.1 Km/W
AnzahlSonden	integer	[-]	Anzahl der Erdwärmesonden eines Sondenfeldes. Der Massenstrom (<i>Auslegungsmassenstrom</i> und <i>Massenstrom</i>) und die Entzugsleistung (<i>Qsource</i>) beziehen sich jeweils auf die Summe aller Sonden. Typischer Wert: 1 - 100
Sondenlaenge	real	[m]	Länge der Sondenbohrung Typischer Wert: 50 - 200 m
Sondendurchmesser	real	[m]	Innendurchmesser D_i der Sonde Typische Werte: 0.0204 / 0.026 / 0.0326 m
Dicke_Sondenrohr	real	[m]	Dicke der Sondenwand. Wird in der aktuellen Version des Programmes nicht verwendet.
Bohrdurchmesser	real	[m]	Durchmesser der Bohrung Typischer Wert: 0.115 - 0.135 m
Auslegungsmassenstrom	real	[kg/s]	Massenstrom der Sole bei Betrieb der Sondenpumpe. Wird für die Berechnung des Wärmeübergangskoeffizienten α verwendet. Falls nicht spezifiziert, wird der <i>Auslegungsmassenstrom</i> := <i>Massenstrom</i> gesetzt. Typischer Wert: 0.4 - 1.0 kg/s

Temperaturen des ungestörten Erdreiches:

Bodentemp	real	[°C]	Mittlere jährliche Bodentemperatur in der Nähe der Erdoberfläche. Dieser Wert wird für die Bestimmung der Anfangstemperaturen des Erdreiches verwendet. In der Regel ist im ungestörten Fall $Bodentemp \approx$ Jahresdurchschnitts-Lufttemperatur $+0.8^\circ\text{C}$ bis $+3^\circ\text{C}$ (schneereiche Gegend). Verzichtet man auf das Einschwingen der Erdtemperatur über mehrere Jahre, so kann die entsprechende Auskühlung durch eine Reduktion von <i>Bodentemp</i> berücksichtigt werden. Bei richtig dimensionierten
------------------	-------------	-------------	--

Einzelsonden ist diese Reduktion 0.5 - 1.0°C (Burkhart et al., 1989), bei zu klein dimensionierten Sonden und Sondenfeldern kann die entsprechend Reduktion wesentlich grösser werden.

Typische Werte, Einschwingen wird berechnet:

Mittelland: Jahresmittel-Lufttemperatur + 0.8°C

Alpen: Jahresmittel-Lufttemperatur + 2°C

Typische Werte nach mehreren Jahren Entzug:

Mittelland: Jahresmittel-Lufttemperatur

Alpen: Jahresmittel-Lufttemperatur + 1°C

Tgrad	real	[°C/m]	Axialer Temperaturgradient in der ungestörten Erde. Typischer Wert: 0.025 - 0.035 °C/m
--------------	-------------	---------------	---

Betriebsgrössen der Erdwärmesonden:

Massenstrom	real	[kg/s]	Momentaner Solemassenstrom, muss in jedem Zeitschritt neu vorgegeben werden. Beim Stillstand der Sonde muss <i>Massenstrom=0</i> gesetzt werden.
--------------------	-------------	---------------	--

Ohne neue Eingabe wird bei *readfile=false* der Massenstrom während der Simulation nicht mehr geändert.

Mit dem Befehl *Erdsonde(ErzeugeLastprofil)* wird auf dem Inputfile während der Laufzeit der Massenstrom jeweils konstant auf dem gesetzten Wert festgehalten, die übrige Zeit wird der *Massenstrom=0* gesetzt.

DeltaT	real	[°C]	Temperaturdifferenz zwischen momentaner Rücklauf-temperatur <i>TSink</i> und Quelltemperatur <i>Tsource</i> des letzten Zeitschrittes. Mögliche Methode zur Berechnung der Rücklauf-temperatur <i>TSink</i> . Wird diese Methode nicht angewandt, um <i>TSink</i> zu berechnen, so ist <i>DeltaT = 0</i> zu setzen. Diese Methode wird nur bei kleinem Zeitschritt empfohlen, weil sonst der Fehler zu gross wird.
---------------	-------------	-------------	--

TSink	real	[°C]	Rücklauf-temperatur der Sole. Ist nur dann eine Eingabegrösse, wenn <i>DeltaT = 0</i> gesetzt ist. Ist <i>Leistung = true</i> gesetzt, d.h. wird intern auf eine vorgegebene Leistung iteriert, so ist die Eingabegrösse <i>TSink</i> der Startwert für die Iteration. In diesem Fall wird der Wert während der Iteration laufend angepasst.
--------------	-------------	-------------	--

QSource	real	[kW]	Momentane Entzugsleistung. Beim Einsatz einer Wärmepumpe gleich momentaner Verdampferleistung. Wird nur benötigt, falls <i>Leistung = true</i> gesetzt ist, d.h. wird intern auf eine vorgegebene Leistung iteriert.
----------------	-------------	-------------	--

Randbedingung mit g-functions:

Die folgenden Parameter werden nur benötigt, wenn die äusseren Randbedingungen des Simulationsgebietes mit den dimensionslosen Sprungantworten (g-functions) berechnet werden. Dazu ist *gfunction = true* zu setzen.

g_Sondenabstand	real	[m]	Sondenabstand der eingegebenen g-function. Die g-functions wurden in Eskilson (1987) für verschiedene Sondenabstände publiziert. Für eine g-function mit $B/H = 0.1$ (d.h. mit einem Verhältnis von Sondenabstand zu Sondenlänge von 0.1) und einer Sondenlänge von 100 m muss die Variable <i>g_Sondenabstand</i> = 10 gesetzt werden. Ist der Sondenabstand der g-function <i>g_Sondenabstand</i> nicht gleich gross wie der effektive <i>Sondenabstand</i> , so wird die effektive g-function im Programm interpoliert, wodurch die Rechengenauigkeit reduziert wird.
Sondenabstand	real	[m]	Effektiver Sondenabstand. Typischer Wert: $0.1 * \text{Sondenlänge}$
g1	real	[-]	g-function bei $\ln(t/t_s) = -4$ Typische Werte: 4.82 für Einzelsonde 4.99 für 2 Sonden mit $B/H = 0.1$ 5.30 für 2 Sonden mit $B/H = 0.05$ 5.00 für 3x6 Sonden mit $B/H=0.1$ 5.00 für 5x10 Sonden mit $B/H=0.1$
g2	real	[-]	g-function bei $\ln(t/t_s) = -2$ Typische Werte: 5.69 für Einzelsonde 6.37 für 2 Sonden mit $B/H = 0.1$ 6.92 für 2 Sonden mit $B/H = 0.05$ 9.15 für 3x6 Sonden mit $B/H=0.1$ 10.2 für 5x10 Sonden mit $B/H=0.1$
g3	real	[-]	g-function bei $\ln(t/t_s) = 0$ Typische Werte: 6.29 für Einzelsonde 7.62 für 2 Sonden mit $B/H = 0.1$ 8.20 für 2 Sonden mit $B/H = 0.05$ 16.4 für 3x6 Sonden mit $B/H=0.1$ 22.75 für 5x10 Sonden mit $B/H=0.1$
g4	real	[-]	g-function bei $\ln(t/t_s) = +2$ Typische Werte: 6.57 für Einzelsonde 8.06 für 2 Sonden mit $B/H = 0.1$ 8.67 für 2 Sonden mit $B/H = 0.05$ 19.75 für 3x6 Sonden mit $B/H=0.1$ 29.5 für 5x10 Sonden mit $B/H=0.1$
g5	real	[-]	g-function bei $\ln(t/t_s) = +3$ Typische Werte: 6.60 für Einzelsonde 8.08 für 2 Sonden mit $B/H = 0.1$ 8.71 für 2 Sonden mit $B/H = 0.05$ 20.05 für 3x6 Sonden mit $B/H=0.1$ 30.05 für 5x10 Sonden mit $B/H=0.1$

Erzeugen eines Auslegungs-Lastverlaufs:

Mit dem Befehl *Erdsonde(ErzeugeLastprofil)* kann ein Eingabefile mit einem stündlichen Lastprofil erzeugt werden. Damit sinnvolle Werte im Kopf des Eingabefiles stehen, müssen zunächst den Parametern mit *Erdsonde(Default)* die Defaultwerte aus dem Programmkopf und eventuell mit *fileName := DefaultFile* und *Erdsonde(Openfile)* aus einem externen Default-File zugewiesen werden. Während der Sonden-Laufzeit wird der Sondendurchsatz auf den Wert von *Massenstrom* und die Entzugsleistung auf *QSource* gesetzt, während der übrigen Zeit werden beide Null. Die Sondenrücklauftemperatur wird immer auf *TSink* gesetzt. Als Dimensionierungsperiode wird ausserdem Ende Februar ein Dauerentzug über eine definierte Anzahl Tage vorgegeben.

Die folgenden, täglichen Laufzeiten haben Werte zwischen 0 und 24 h pro Tag:

Laufzeit_Jan	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im Januar
Laufzeit_Feb	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im Februar
Laufzeit_Mar	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im März
Laufzeit_Apr	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im April
Laufzeit_Mai	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im Mai
Laufzeit_Jun	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im Juni
Laufzeit_Jul	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im Juli
Laufzeit_Aug	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im August
Laufzeit_Sep	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im September
Laufzeit_Okt	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im Oktober
Laufzeit_Nov	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im November
Laufzeit_Dez	integer	[h]	Tägliche Laufzeit der Erdwärmesonde im Dezember
QSpitzeFeb	real	[kW]	Dauerentzugsleistung Ende Februar. In der Regel ist <i>QSpitzeFeb := QSource</i> .
DauerLastspitze	integer	[Tage]	Dauer der ununterbrochenen Sondenbelastung Ende Februar in Tagen. Wird als Dimensionierungsperiode verwendet. Typischer Wert: 3 Tage
Massenstrom	real	[kg/s]	Sondendurchsatz aller Sonden während der Laufzeit der Sonden.
QSource	real	[kW]	Entzugsleistung aller Erdwärmesonden während der Laufzeit der Sonden.
TSink	real	[°C]	Rücklauftemperatur, für alle Zeitschritte konstant. Wird bei einer Berechnung mit Entzugsleistung (Iteration, <i>Leistung = true</i> , <i>Leistungsinput = true</i>) als Startwert für die Iteration benötigt.

Logische Schaltelemente:

stationaer	boolean	Bei <i>stationaer = true</i> wird die Sole stationär, bei <i>stationaer = false</i> wird sie instationär berechnet.
readFile	boolean	Bei <i>readFile = true</i> werden die wichtigsten Eingabeparameter (inklusive momentaner Massenstrom, Rücklauftemperatur <i>TSink</i> und die momentane Entzugsleistung <i>QSource</i>) von einem externen Textfile eingelesen.
writeFile	boolean	Bei <i>writeFile = true</i> werden die Ausgabewerte für jeden Zeitschritt (Rücklauftemperatur <i>TSink</i> , Quelltemperatur <i>TSource</i> und Monitortemperatur <i>TMonitor</i>) auf ein externes Textfile geschrieben.
einschwingen	boolean	Solange <i>einschwingen = true</i> ist, werden keine Resultate auf das Ausgabefile geschrieben. Ist die Simulation am Ende des Eingabefiles angelangt, so wird bei <i>einschwingen = true</i> das Eingabefile geschlossen und dann wieder für einen neuen Durchgang geöffnet. Die Variable Jahr wird bei diesem Vorgang um eins erhöht. Aus Speicherplatzgründen muss in der vorliegenden Version spätestens nach 9 Jahren <i>einschwingen = false</i> gesetzt werden.
Leistungsinput	boolean	Bei <i>Leistungsinput = true</i> wird die momentane Entzugsleistung <i>QSource</i> aus einem externen Eingabefile gelesen.
Leistung	boolean	Soll intern auf die Leistung <i>QSource</i> iteriert werden so ist <i>Leistung = true</i> zu setzen.
Druck	boolean	Soll der Druckabfall in der Erdwärmesonde in jedem Rechenschritt neu berechnet und auf das Ausgabefile geschrieben werden, so ist <i>Druck = true</i> zu setzen.
gfunction	boolean	Wird <i>gfunction = true</i> gesetzt, so werden die Randbedingungen mit der g-Funktion von Claeson und Eskilson berechnet. Bei <i>gfunction = false</i> wird an der äusseren Randbedingung mit der Werner-Formel gerechnet.
Allsteps	boolean	Mit <i>Allsteps = true</i> werden auch die Resultate während des Einschwingvorgangs (alle Simulationsjahre) auf das Ausgabefile geschrieben, bei <i>Allsteps = false</i> wird nur das letzte Simulationsjahr auf das Ausgabefile geschrieben.
Starttemp	boolean	Bei <i>Starttemp = true</i> werden die Erdreichtemperaturen mit Hilfe der g-functions aus der Entzugsenergie des ersten Simulationsjahres (bei unveränderter Entzugsleistung für die folgenden Jahre) für das zweitletzte Simulationsjahr abge-

schätzt. Die eigentliche Simulation wird dann nur für das zweitletzte und letzte Simulationsjahr durchgeführt. Die Simulation muss dabei im Stundensschritt erfolgen (d.h. *Zeitschritt = 60, numrows = 8760*). Das letzte Simulationsjahr wird mit *Lastyear* definiert. Mit dieser Methode kann das Temperaturniveau eines Sondenfeldes nach mehrjährigem Entzug ohne lange Simulation abgeschätzt werden.

AlteInputFiles	boolean	Bei <i>AlteInputFiles = true</i> ist das Ein- und Ausgabeformat auf den Eingabe- und Ausgabefiles identisch mit der ersten Version des Programmes EWS, das 1997 publiziert wurde (Huber, Schuler, 1997). Mit <i>AlteInputFiles = false</i> wird ein neues Fileformat verwendet, in dem sämtliche Simulationsparameter der aktuellen Programmversion enthalten sind.
-----------------------	----------------	---

File-Handling:

IFileName	string	Name des Eingabefiles. Einlesen der externen Werte aus einem File mit dem Befehl: <i>Erdsonde (OpenFile)</i> Damit externe Daten gelesen werden können, muss <i>readfile=true</i> gesetzt sein. Typischer Name: Eingabe.txt
OFileName	string	Name des Ausgabefiles. Damit die Resultate in ein Ausgabefile geschrieben werden können, muss <i>writefile=true</i> gesetzt sein. Typischer Name: Ausgabe.txt
swewsName	string	Name des Stoff- und Temperaturdatenfiles des ungestörten Erdreiches, das mit dem Programm SwEWS erzeugt wird (Leu et al., 1999). Einlesen der externen Werte aus einem File mit dem Befehl: <i>Erdsonde (Stoffdaten)</i> Damit werden die Stoffdaten (cpErde, rhoErde, lambdaErde, Jahresmitteltemp, Bodenerwärmung, Tgrad mit den Werten aus dem SwEWS-File überschrieben. Typischer Name: swews.dat
Inputfile	string	Name des Eingabefiles, das mit dem internen Lastprofil-Generator erzeugt wird (Erzeugen des Eingabefiles mit <i>Erdsonde (ErzeugeLastprofil)</i>). Wird im Programmkopf der Unit EWSMOD.PAS festgelegt und kann vom Hauptprogramm nicht geändert werden. Momentaner Name: input.txt

Defaultfile	string	Name eines externen Files mit Default-Werten. Wird in der Regel nicht benötigt. Einlesen der externen Default-Werte mit: <i>IFilename := Defaultfile</i> <i>Erdsonde (OpenFile)</i> Wird im Programmkopf der Unit EWSMOD.PAS festgelegt und kann vom Hauptprogramm nicht geändert werden. Momentaner Name: default.txt
Sprache	string	Wird in der aktuellen Version nicht verwendet.

4.2.2 Ausgabewerte

TSink	real	[°C]	Rücklauftemperatur am Ende des jeweiligen Zeitschrittes. Ist nur dann ein Ausgabewert, wenn intern im EWS-Modul auf eine vorgegebene Leistung iteriert wird, d.h. <i>Leistung = true</i> gesetzt ist.
QSource	real	[kW]	Wärmeentzugsleistung aller Erdwärmesonden. Ist nur dann ein Ausgabewert, wenn <i>Leistung = false</i> gesetzt wurde.
TSource	real	[°C]	Quellentemperatur, Durchschnitt des jeweiligen Zeitschrittes.
TMonitor	real	[°C]	Temperatur im Monitorpunkt des Erdreiches.
p	real	[Pa]	Druckabfall in der Erdwärmesonde. Wird in jedem Zeitschritt neu berechnet, sofern <i>Druck = true</i> gesetzt ist.
laminar	boolean		Gibt an, ob die Reynoldszahl $Re < 2300$ und die Strömung in der Sonde somit wahrscheinlich laminar ist. Wird in jedem Zeitschritt neu berechnet.
Jahr	integer	[Jahre]	Aktuelles Simulationsjahr. Wenn die Zählvariable <i>simstep = numrows</i> ist, dann wird intern die Variable <i>Jahr</i> um eins erhöht und <i>simstep = 0</i> gesetzt. Die Variable <i>Jahr</i> kann als Abbruchkriterium für die Simulation verwendet werden.
Sondenlaufzeit	real	[h/a]	Laufzeit der Erdwärmesonden pro Jahr. Wird nur beim Aufruf <i>Erdsonde (ErzeugeLastprofil)</i> berechnet.
TSinkMin	real	[°C]	Minimale Sonden-Rücklauftemperatur TSink.
TSinkMax	real	[°C]	Maximale Sonden-Rücklauftemperatur TSink.
TSourceMin	real	[°C]	Minimale Quelltemperatur TSource.
TSourceMax	real	[°C]	Maximale Quelltemperatur TSource.
ErrorMldg	integer	[-]	Fehlermeldung: 0 -> Alle in Ordnung 1 -> Sondenabstand zu klein!

4.2.3 Parametereingabe im Hauptprogramm und Eingabefile

Nachfolgend ist ein Beispiel für den Aufruf des EWS-Moduls aus einem Hauptprogramm zu finden. Dabei wird 1 Jahr lang, vorgängig zur eigentlichen Simulation, mit dem vorgegebenen Entzugsprofil eingeschwungen. Änderungen zu den Vorschlagswerten (Defaultwerten) müssen nach dem Aufruf `Erdsonde(Default)` eingegeben werden, da sie ansonsten überschrieben würden. In diesem Beispiel wird intern auf eine vorgegebene Leistung iteriert. Der Wert für die Entzugsleistung wird stündlich aus dem mit `Erdsonde(ErzeugeLastprofil)` erzeugten File *'Input.txt'* ausgelesen.

```
Program EWSTEST;
uses ewsmod;
begin
(* Setzt die Rechenparameter auf die Defaultwerte
   aus dem Programmkopf, zwingende Eingabe *)
  Erdsonde(Default);

(* Setzt die Rechenparameter auf die Defaultwerte
   aus dem externen Default-File, fakultative Eingabe *)
  IfileName := DefaultFile;
  Erdsonde(Openfile);

(* Liest die Rechenparameter aus dem File 'Eingabe.txt' ein.
   Nicht notwendig, falls interner Lastgenerator benutzt wird. *)
  IfileName := 'Eingabe.txt';
  OfileName := 'Ausgabe.txt';
  Erdsonde(Openfile);

(* Liest die Stoffwerte der Erde aus dem
   SwEWS-File 'swews.dat' ein, fakultative Eingabe *)
  swewsInput := true;
  swewsName  := 'swews.dat';
  Erdsonde(Stoffdaten);

(* Erzeugt ein stündliches Lastprofil auf dem Eingabefile 'input.txt' und
   verwendet dieses Eingabefile als stündliche Eingabedaten. Alternative
   Eingabe zum Eingabefile 'Eingabe.txt', das extern erzeugt wurde. *)
  Erdsonde(ErzeugeLastprofil);
  IfileName      := InputFile;
  OfileName      := 'Ausgabe.txt';
  Leistungsinput := true;
  Leistung       := true;
  Lastyear      := 2;
  Erdsonde(Openfile);

(* Initialisiert die Simulation und berechnet die Matrizen B, zwingende *)
  Erdsonde(Initial);

(* Berechnungsroutine, zwingend *)
  Simstep := 0;
  repeat
    if Jahr=Lastyear then Einschwingen:=false else Einschwingen:=true;
    Simstep := Simstep + 1;
    Erdsonde(Iteration);
  until Jahr>Lastyear;

(* Schliesst In- und Outputfile und schreibt letztes Resultat, zwingend *)
  Erdsonde(CloseFile);

end.
```

Das Eingabefile hat die folgende Form, wobei hier nur die ersten 5 Stunden des Simulationsjahres mit 8760 Stunden abgebildet sind. Die stündlichen Eingabedaten beginnen auf der Zeile 202. Als Trennzeichen sind Tabulatoren eingesetzt:

Eingabefile des EWS-Programms

Test		Projektname
A.Huber		Ersteller
		Anmerkung
1		Anzahl Sonden
1.00E+02	m	Sondenlaenge
1.15E-01	m	Bohrdurchmesser
2.60E-02	m	Sondeninnendurchmesser
0.00E+00	m	Dicke Sondenrohr
3.68E-04	K/W	R1
1.47E-01	KmW	Ra
1.00E-01	Km/W	Rb
4.49E-01	W/mK	Lambda Fluid
1.05E+03	kg/m3	Dichte Fluid
3.88E+03	J/kgK	cp Fluid
4.15E-06	m2/s	Viskositaet Fluid
4.00E-01	kg/s	Auslegungsmassenstrom
TRUE		Druckabfall berechnet
4.00E-01	kg/s	Massenstrom
FALSE		Einlesen SWEWS-Daten von File
swews.dat		
2.2	W/mK	Lambda Erde
2000	kg/m3	Dichte Erde
1100	J/kgK	cp Erde
2.2	W/mK	Lambda Erdel
2000	kg/m3	Dichte Erdel
1100	J/kgK	cp Erdel
2.2	W/mK	Lambda Erde2
2000	kg/m3	Dichte Erde2
1100	J/kgK	cp Erde2
2.2	W/mK	Lambda Erde3
2000	kg/m3	Dichte Erde3
1100	J/kgK	cp Erde3
2.2	W/mK	Lambda Erde4
2000	kg/m3	Dichte Erde4
1100	J/kgK	cp Erde4
2.2	W/mK	Lambda Erde5
2000	kg/m3	Dichte Erde5
1100	J/kgK	cp Erde5
2.2	W/mK	Lambda Erde6
2000	kg/m3	Dichte Erde6
1100	J/kgK	cp Erde6
2.2	W/mK	Lambda Erde7
2000	kg/m3	Dichte Erde7
1100	J/kgK	cp Erde7
2.2	W/mK	Lambda Erde8
2000	kg/m3	Dichte Erde8
1100	J/kgK	cp Erde8
2.2	W/mK	Lambda Erde9
2000	kg/m3	Dichte Erde9
1100	J/kgK	cp Erde9
2.2	W/mK	Lambda Erdel0
2000	kg/m3	Dichte Erdel0
1100	J/kgK	cp Erdel0

8.10E-01	W/mK	Lambda Hinterfuellung					
1.18E+03	kg/m3	Dichte Hinterfuellung					
3.04E+03	J/kgK	cp Hinterfuellung					
9.00E+00	_C	Jahresmitteltemperatur Luft					
3.00E-02	_C/m	Temperaturgradient Boden					
8.00E-01	_C	Boden waermer als Luft					
FALSE		Rechnen mit Entzugsleistung					
FALSE		Leistungsinput					
0	y	Beginn Simulation					
8760		numrows					
1	y	Simulationsdauer in Jahren					
8760	h	Simulationszeit					
0.00E+00	_C	DeltaT					
TRUE		stationaere Berechnung Sole					
1		MonitorAxi					
1		MonitorRad					
4.00E+00		Sicherheit1					
2.00E+00		Sicherheit2					
60	Min	Zeitschritt					
1	w	RepRandbed					
5		Radiale Unterteilung					
1		Axiale Unterteilung					
2.00E+00		Gitterfaktor					
1.50E+00		Rechenradius					
5.00E-01	_C	Genauigkeit Iteration					
760		Stepwrite					
TRUE		readfile					
TRUE		writefile					
FALSE		Resultate des Einschwingens aufschreiben					
FALSE		Abschaetzen der Starttemperatur					
0		TSink					
0		TSource					
4.82E+00		g-function bei ln(t/ts) = -4					
5.69E+00		g-function bei ln(t/ts) = -2					
6.29E+00		g-function bei ln(t/ts) = 0					
6.57E+00		g-function bei ln(t/ts) = +2					
6.60E+00		g-function bei ln(t/ts) = +3					
1.00E+01	m	eff. Abstand der Sonden					
1.00E+01	m	Sondenabstand der g-function					
TRUE		Randbedingung mit g-functions					
FALSE		Erdreichtemperatur abschotzen					
12	h/d	Laufzeit Januar					
10	h/d	Laufzeit Februar					
12	h/d	Laufzeit Maerz					
12	h/d	Laufzeit April					
12	h/d	Laufzeit Mai					
12	h/d	Laufzeit Juni					
12	h/d	Laufzeit Juli					
12	h/d	Laufzeit August					
12	h/d	Laufzeit September					
12	h/d	Laufzeit Oktober					
12	h/d	Laufzeit November					
12	h/d	Laufzeit Dezember					
10	kW	Last-Spitze fuer Lastprofil im Februar					
3	d	Dauer der Lastspitze					
....							
Step	M [kg/s]	TSink	Q [kW]	TSource	TMonitor	p [Pa]	laminar?
1	0	3.19423884	0				
2	0	0.38217322	0				
3	0	-1.2628075	0				
4	0.5	-1.28826741	5.404				
5	0.5	-1.28826741	6.14				
....							

Ein Beispiel-Eingabefile mit dem Namen 'input.txt' kann mit folgendem Hilfsprogramm erzeugt werden:

```
Program InputEWS;  
uses ewsmod;  
begin  
  Erdsonde(Default);  
  IfileName:=Inputfile;  
  Erdsonde(ErzeugeLastprofil);  
end.
```

Das Ausgabefile sieht prinzipiell gleich aus, es werden zusätzlich aber in den Stundenwerten hinten noch die Quelltemperatur, die Monitortemperatur, der Druckabfall in der Sonde und der Strömungszustand (laminar = true, turbulent = false) ausgegeben. Ein Ausgabefile kann somit für eine spätere Berechnung wieder als Eingabefile verwendet werden.

5 Symbolverzeichnis

Lateinisches Alphabet

a	m^2/s	Temperaturleitfähigkeit ($\lambda/\rho c_p$)
B	m	Sondenabstand benachbarter Erdwärmesonden
c_p	J/kgK	spezifische Wärmekapazität
C	J/K	Wärmekapazität
D	m	nicht aktiver Teil einer Erdwärmesonde
D_b	m	Bohrdurchmesser
D_i	m	Durchmesser der Sondenrohre (Innen-Mass)
dl	m	Länge eines Rohr-Segmentes (axial)
dt	s	interner Simulationszeitschritt der Erde
dt2	s	interner Simulationszeitschritt der Sole
f		Gitterfaktor, in radialer Richtung
H	m	aktive, vertikale Sondenlänge
i	Woche	Zeitpunkt des Wärmeentzugs der Grösse \dot{q} (Durchschnitt der Woche i)
L	W/K	thermische Leitfähigkeit, $1/R$
L_0	W/K	thermische Leitfähigkeit der fließenden Sole (axial), $2 \pi r_o^2 v \rho_{Sole} c_{pSole}$
L_1	W/K	thermische Leitfähigkeit zwischen Sole (alle 4 Sondenrohre) und Hinterfüllung
m	-	Anzahl Rechenknoten in radialer Richtung
m	kg	Masse zweier Sole-Elemente der Länge dl
\dot{m}	kg/s	Solemassenstrom (Durchsatz durch Pumpe)
n	Woche	Betrachtungszeitpunkt bei Trichterformel
q	W/m	spezifische Wärmeentzugsleistung, \dot{Q}/dl
r	m	radialer Abstand von der Rohrachse
r_o	m	Innen-Radius der Sondenrohre
r_1	m	Radius der Bohrung
r_m	m	Radius des Simulationsgebietes (Ort der äusseren Randbedingung)
rz	m	radialer Massenschwerpunkt eines Rohr-Rings
R	K/W	thermischer Widerstand, $1/L$
R_a	Km/W	internal thermal resistance
R_b	Km/W	borehole thermal resistance
substep	-	Sub-Unterteilung für Soleberechnung (substep = dt / dt2)
t	s	Zeit
t_s	s	Sondenzeitkonstante ($t_s = H^2 / 9a$)
T	°C	Temperatur
T_b	°C	borehole temperature, Temperatur am Bohr-Radius
T_{Grad}	°C/m	vertikaler Temperaturgradient des ungestörten Erdreiches
T_{mo}	°C	Mittlere Jahrestemperatur der Erdoberfläche

TEarth	°C	Temperatur des Erdreiches
TDown	°C	Temperatur der nach unten strömenden Sole
TUp	°C	Temperatur der nach oben strömenden Sole
TSource	°C	Quellentemperatur, Temperatur der aus der Sonde austretenden Sole
TSink	°C	Rücklauftemperatur der Sole, Verdampfer-Austrittstemperatur der Sole
v	m/s	Strömungsgeschwindigkeit der Sole

Griechisches Alphabet

α	W/m ² K	Wärmeübergangskoeffizient
α_0	W/m ² K	Wärmeübergangskoeffizient Sole-Rohr bei Stillstand der Sonde
α_1	W/m ² K	Wärmeübergangskoeffizient Sole-Rohr bei Betrieb der Sonde
λ	W/mK	Wärmeleitfähigkeit
ν	m ² /s	kinematische Viskosität
π	-	Pi, 3.1415..
ρ	kg/m ³	Dichte
ξ	-	Druckverlustkoeffizient

Indizes

dia	Diagramm - Wert, z.B. B _{dia} = Sondenabstand aus dem g-function-Diagramm
eff	effektiver Wert, z.B. B _{eff} = effektiver Sondenabstand
DimAxi	Anzahl Rechenknoten in axialer Richtung
DimRad	Anzahl Rechenknoten in radialer Richtung (inklusive Hinterfüllungsknoten)
i	Ortskoordinate, axial
j	Ortskoordinate, radial
k	Zeitkoordinate
lam	laminar
turb	turbulent
Erde	Erde
Fill	Hinterfüllung
Sole	Sole, generell Wärmeträger-Fluid
Woche	Betrachtungswoche bei Trichterformel

Kennzahlen

Es	t 9a / H ²	Eskilsonzahl, dimensionslose Zeit
Nu	$\alpha D / \lambda$	Nusseltzahl, gebildet mit dem Rohrdurchmesser und λ Fluid
Pr	v / a	Prandtlzahl
Re	D v / ν	Reynoldszahl, gebildet mit dem Rohrdurchmesser

6 Literaturverzeichnis

- Afjei, Th. (1989): YUM. A Yearly Utilization Model for Calculating the Seasonal Performance Factor of Electric Driven Heat Pump Heating Systems. Technical Form. Laboratorium für Energiesysteme, ETH Zürich.
- Afjei, Th.; Wittwer, D. (1995): Yearly Utilization Model YUM WP/Holz. Benutzerhandbuch mit Beispielen. Ein Simulationsprogramm zur Berechnung des Jahresenergieverbrauchs und Nutzungsgrads bivalenter Wärmepumpen-Heizsysteme mit Holzkessel. INFEL und BEW.
- Bigler, R.; Lüthi, F. (1996): Thermoprogramm Erdwärmesonden, Burgdorf. Messdaten des Versuchs 4, Sonde E1300 und Messdaten des Versuchs 6, Sonde E1300. Werner + Partner AG, Ingenieur- und Studienbüro, Burgdorf.
- Burkhart, R.; Hopkirk, R.; Eugster W.; Rybach L. (1989): Erdwärmesonden-Heizanlagen: Durch Messungen und Berechnungen bestimmte Auslegungsgrößen. Bundesamt für Energiewirtschaft, Bern. Studie Nr. 46.
- Claesson, J.; Eskilson, P.: Conductive Heat Extraction by a Deep Borehole. Analytical Studies. Dep. of Mathematical Physics, University of Lund.
- Claesson, J.; Eskilson, P. (1987): Conductive Heat Extraction by a Deep Borehole. Thermal Analysis and Dimensioning Rules. Dep. of Mathematical Physics, University of Lund.
- Eskilson, P. (1986): Superposition Borehole Model. Manual for Computer Code. Department of Mathematical Physics, Lund Institute of Technology, Lund, Sweden.
- Eskilson, P. (1987): Thermal Analysis of Heat Extraction Boreholes. Department of Mathematical Physics, Lund Institute of Technology, Lund, Sweden.
- Eugster, W. (1991): Erdwärmesonden - Funktionsweise und Wechselwirkung mit dem geologischen Untergrund. Feldmessungen und Modellsimulation. Diss. ETH Nr. 9524.
- Eugster, W.; Hopkirk, J.; Rybach, L. (1999): Ist untiefe Geothermie erneuerbar? Forschungsprogramm Geothermie. Bundesamt für Energie (BFE), Bern.
- Grigull U.; Sander H.: Wärmeleitung. Springer Verlag.
- Hässig, W.; Suter, D.; Bigler, R.; Huber, A. (1998): Regeneration von Erdwärmesonden, Phase I. Potentialabschätzung. Forschungsprogramm Umgebungs- und Abwärme, Wärmekraftkopplung (UAW). Bundesamt für Energie (BFE), Bern. ENET-Nr. 9722601/1, 1998.
- Hellström, G. (1991): Ground Heat Storage. Thermal Analyses of Duct Storage Systems. Theory. Dep. of Mathematical Physics, University of Lund, Sweden.
- Hellström, G.; Sanner, B. (1994): PC-Programm zur Auslegung von Erdwärmesonden. Erdgekoppelte Wärmepumpen. IZW-Bericht 1/94. Karlsruhe: Fachinformationszentrum.
- Huber, A.; Müller, C.; Berchtold O.; Eggenberger, H. (1996): Luftvorwärmung für Wärmepumpen in Erdregistern. Phase 1, Kosten-Nutzen-Analyse. Forschungsprogramm Umgebungs- und Abwärme, Wärmekraftkopplung (UAW). Bundesamt für Energie (BFE), Bern. ENET-Nr. 9554499/1, 1996.
- Huber, A.; Schuler, O. (1997): Berechnungsmodul für Erdwärmesonden. Forschungsprogramm Umgebungs- und Abwärme, Wärmekraftkopplung (UAW). Bundesamt für Energie (BFE), Bern. ENET-Nr. 9658807/1, 1997.
- Huber, A.; Pahud, D. (1999): Untiefe Geothermie: Woher kommt die Energie? Forschungsprogramm Geothermie. Bundesamt für Energie (BFE), Bern.

-
- Klein, S.A. et al. (1996): TRNSYS. A Transient System Simulation Program. Solar Energy Laboratory, University of Wisconsin. Madison, USA.
- Knoblich, K.; Sanner, B.; Klugescheid, M. (1993): Erdgekoppelte Wärmepumpen. Energetische, hydrologische und geologische Untersuchungen zum Entzug von Wärme aus dem Erdreich an der Forschungsanlage Schöffengrund-Schwalbach und anderen Standorten. Giessener Geologische Schriften Nr. 49. Lenz-Verlag, Giessen.
- Leu, W.; Keller, G.; Mégel, Th.; Schärli, U.; Rybach, L. (1999): Programm SwEWS-99. Berechnungsprogramm für geothermische Eigenschaften der Schweizer Molasse (0-500m). Bundesamt für Energie, Forschungsprogramm UAW, Bern. ENET-Nr. 9723763.
- Leu, W.; Keller, G.; Matter, A.; Schärli, U.; Rybach, L. (1999): Geothermische Eigenschaften Schweizer Molassebecken (Tiefenbereich 0-500m). Bundesamt für Energie, Programm Geothermie, Bern.
- Pahud, D.; Fromentin, A.; Hadorn J. (1996): The Superposition Borehole Model for TRNSYS (TRNSBM). User Manual for the November 1996 Version. Internal Report. LASEN - DGC - EPFL, Lausanne.
- Pahud, D.; Fromentin, A. (1998): Pieux échangeurs: outils de simulation, KWH - Energieforschung im Hochbau, Status-Seminar ETH Zürich.
- Pahud, D.; Hubbuch, M. (1998): Response-Test für die Energiepfahlanlage Dock Midfield, Zürich Flughafen. Bundesamt für Energie, Forschungsprogramm Wärmespeicherung, Bern.
- Merker, G. (1987): Konvektive Wärmeübertragung. Springer-Verlag.
- Nani, M.; Stalder, M.; Schweizer, A.; Baumgartner, T. (1994): WPcalc. Rechenprogramm zur Auslegung von WP-Anlagen. Bundesamt für Energiewirtschaft, Bern.
- Sanner, B. (1992): Erdgekoppelte Wärmepumpen, Geschichte, Systeme, Auslegung, Installation. In: IZW-Berichte 2/92. Karlsruhe: Fachinformationszentrum.
- SIA (1996): Grundlagen zur Nutzung der untiefen Erdwärme für Heizsysteme. SIA Dok D0136.
- Werner, A.; Bigler, R.; Niederhauser, A. et. al. (1996): Grundlagen für die Nutzung von Wärme aus Boden und Grundwasser im Kanton Bern. Thermoprogramm Erdwärmesonden, Burgdorf. Schlussbericht. Wasser- und Energiewirtschaftsamt des Kt. Bern (WEA).
- Wetter, M.; Huber A. (1997): TRNSYS Type: Vertical Borehole Heat Exchanger, EWS Model. Zentralschweizerisches Technikum ZTL, Luzern.
- Zimmermann, M. et al. (1999): Handbuch der passiven Kühlung. EMPA (ZEN)/ Bundesamt für Energie BFE, Bern. ISBN 3-905594-06-4.


```

Def_Zeitschritt      = 60;          (* Min *)
Def_Simulationsdauer = 8760;       (* h, totale Simulationszeit *)
Def_IFileName       = 'Eingabe.txt'; (* Name des Eingabe-Files *)
Def_OFileName       = 'Ausgabe.txt'; (* Name des Ausgabe-Files *)
Def_swewsName       = 'swews.dat';  (* Name des Stoffwert-Files *)
Defaultfile        = 'default.txt'; (* Externes Defaultwert-File *)
Inputfile           = 'input.txt';  (* Neues Lastprofile-File *)
Def_readFile        = true;
Def_writeFile       = true;
Def_Allsteps        = false;       (* Auch Einschwingen aufschreiben *)
Def_MonitorAxi      = 1;
Def_MonitorRad      = 1;
Def_Einschwingen    = false;
Def_Leistungsinput  = false;       (* Entzugsleistung aus Inputfile *)
Def_Leistung        = false;       (* Rechnen mit Entzugsleistung *)
Def_stationaer      = false;       (* Stationaere Berechnung der Sole *)
Def_Genauigkeit     = 0.05;       (* øC, Iteration Leistungsberech. *)
Def_Sprache         = 'D';
Def_StepWrite       = 1;          (* Ausgabeschritte: 2 fuer jede 2te *)
Def_Jahr            = 1;          (* Beginn der Simulation *)
Def_adiabat         = 0.0;        (* Anteil adiabater Randbed. 0..1 *)
Def_Druck           = true;       (* Wird der Druckabfall berechnet? *)
Def_g1              = 4.82;       (* g-function bei ln(t/ts) = -4 *)
Def_g2              = 5.69;       (* g-function bei ln(t/ts) = -2 *)
Def_g3              = 6.29;       (* g-function bei ln(t/ts) = 0 *)
Def_g4              = 6.57;       (* g-function bei ln(t/ts) = +2 *)
Def_g5              = 6.60;       (* g-function bei ln(t/ts) = +3 *)
Def_gfunction       = true;       (* Randbedingung mit g-functions *)
Def_Sondenabstand   = 10;        (* m, eff. Abstand der Sonden *)
Def_g_Sondenabstand = 10;        (* m, Sondenabstand der g-function *)
Def_Starttemp       = false;      (* Erdreichtemp. abschaetzen *)
Def_LastYear        = 1;          (* Simulationsdauer in Jahren *)
Def_AlteInputfiles  = false;     (* Alter, kurzer Eingabefilekopf *)
Def_swewsInput      = false;     (* Liest Stoffwerte von SWEWS.DAT *)
Def_Laufzeit_Jan    = 12;        (* h, taegliche Sondenlaufziet Jan *)
Def_Laufzeit_Feb    = 11;        (* h, taegliche Sondenlaufziet Feb *)
Def_Laufzeit_Mar    = 9;         (* h, taegliche Sondenlaufziet Mar *)
Def_Laufzeit_Apr    = 7;         (* h, taegliche Sondenlaufziet Apr *)
Def_Laufzeit_Mai    = 3;         (* h, taegliche Sondenlaufziet Mai *)
Def_Laufzeit_Jun    = 2;         (* h, taegliche Sondenlaufziet Jun *)
Def_Laufzeit_Jul    = 2;         (* h, taegliche Sondenlaufziet Jul *)
Def_Laufzeit_Aug    = 2;         (* h, taegliche Sondenlaufziet Aug *)
Def_Laufzeit_Sep    = 3;         (* h, taegliche Sondenlaufziet Sep *)
Def_Laufzeit_Okt    = 7;         (* h, taegliche Sondenlaufziet Okt *)
Def_Laufzeit_Nov    = 9;         (* h, taegliche Sondenlaufziet Nov *)
Def_Laufzeit_Dez    = 11;        (* h, taegliche Sondenlaufziet Dez *)
Def_QSpitzeFeb      = 5;         (* kW, Spitzenentzug im Februar *)
Def_DauerLastspitze = 4;         (* Tage, Dauer der Lastspitzen *)

type Vektor = array[0..DimAxiMax] of real;
Vektor10 = array[0..10] of real;
Vektor12 = array[0..12] of integer;
VektorRad= array[0..DimRadMax+1] of real;
MatrixQ = array[1..DimAxiMax,0..53*MaxJahre] of real;
Matrix = array[0..DimAxiMax, 0..DimRadMax+1] of real;
Matrix6 = array[0..6, 0..6] of real;
KVektor = array[0..DimAxiMax, 0..DimRadMax+1] of real;
KMatrix = array[1..DimAxiMax, 0..DimRadMax+1, 0..DimRadMax+1] of real;
line100 = array[1..202] of string[100];

var Sondenlaenge, Sondendurchmesser, Dicke_Sondenrohr,
Bohrdurchmesser, Massenstrom, cpSole, cpFill, rhoSole,
rhoFill, lambdaSole, lambdaFill, nueSole, dl, TGrad, Rl, Ra, Rb,
Rechenradius, DeltaT, TSource, TSink, mcpSole, Llrun, Llstop,
Gitterfaktor, Auslegungsmassenstrom, Jahresmitteltemp,
Bodenerwaermung, Bodentemp, Sicherheit1, Sicherheit2,
Rechengebiet, p, TSourceOld, TSinkOld, cpErd, rhoErd, lambdaErd,
TMonitor, QSource, Genauigkeit, adiabat, g1, g2, g3, g4, g5,
gpar1, gpar2, gpar3, gpar4, gpar5, gpar6, Sondenabstand,
g_Sondenabstand, uMin, QSpitzeFeb, TSourceMin, TSourceMax,
TSinkMin, TSinkMax, Sondenlaufzeit : real;
Zeitschritt, subdt, substep_run, substep_stop, Reprandbed,
Simulationsdauer, iteration, DimRad, DimAxi, simstepn,
MonitorAxi, MonitorRad, Jahr, StepWrite, LastYear, Beginn,
AnzahlSonden, DauerLastspitze, ErrorMldg : integer;
simstep, numrows, filestep : longint;
TEarth, TEarthOld : Matrix;
T0, TUp, TUpold, TDown, Tdownold, Llast, Q0, Q0old, VektorNull,

```



```

QEntzug                                     : Vektor;
lambdaErde,cpErde,rhoErde                  : Vektor10;
B1,B2                                       : kMatrix;
laminar,readFile,writeFile,Einschwingen,Druck,
Leistungsinput,Leistung,stationaer,gfunction,Starttemp,
AlteInputfiles,SwewsInput,Allsteps        : boolean;
Q                                             : MatrixQ;
Projektname,Ersteller,Anmerkungen,Wort,Wort1,Wort2,Wort3,
Wort4,Wort5,Wort6,Wort7,Wort8,Wort9,Wort10,
IFilename,OFilename,Sprache,swewsName     : String;
{ ifile, ofile, swews                       : textfile; } (* Fuer Delphi-Version *)
{ ifile, ofile, swews                       : text; } (* Fuer Pascal-Version *)
line                                        : line100;
SchichtDicke,SchichtTemp,SchichtCp,SchichtLambda,
SchichtRho                                 : Vektor10;
Laufzeit                                    : Vektor12;

Procedure Erdsonde( iteration : integer);
Procedure EWSmodul( iteration          : integer;
var ErrorMldg,DimAxi,DimRad,Zeitschritt,
subdt,substep_run,substep_stop,RepRandbed,
MonitorAxi,MonitorRad,Jahr,LastYear,Beginn,
AnzahlSonden,DauerLastspitze          : integer;
var numrows                             : longint;
var rhoFill,cpFill,lambdaFill,rhoSole,cpSole,
lambdaSole,nueSole,R1,Ra,Rb,Sondenlaenge,
Sondendurchmesser,Dicke_Sondenrohr,
Bohrdurchmesser,Auslegungsmassenstrom,
Massenstrom,Bodentemp,Tgrad,mcpSole,Llrun,
Llstop,dl,Rechenradius,Rechengebiet,
Gitterfaktor,Sicherheit1,Sicherheit2,
DeltaT,TSourceOld,TSource,TSink,TMonitor,p,
QSource,Genauigkeit,adiabat,g1,g2,g3,g4,g5,
Sondenabstand,g_Sondenabstand,QSpitzeFeb,
TSourceMin,TSourceMax,TSinkMin,TSinkMax,
Sondenlaufzeit                         : real;
var B1,B2                                : KMatrix;
var TEarth,TEarthold                    : Matrix;
var T0,TUp,TUpOld,TDown,TDownOld,Llast  : Vektor;
var cpErde,rhoErde,lambdaErde           : Vektor10;
var Q                                     : MatrixQ;
var Laufzeit                             : Vektor12;
var readFile,writeFile,laminar,
Einschwingen,Leistungsinput,Leistung,
stationaer,gfunction,Starttemp,Druck,
AlteInputFiles,SwewsInput,Allsteps     : boolean;
var IFilename,OFilename,swewsName,Sprache,Wort: string );

implementation

function Min(a,b:real):real;
begin if a<b then Min:=a else Min:=b; end;

function Max(a,b:real):real;
begin if a>b then Max:=a else Max:=b; end;

(*-- Procedure Default -----*)
Procedure DefaultWert(var cpSole,cpFill,rhoSole,rhoFill,lambdaSole,
lambdaFill,nueSole,Rechenradius,
Gitterfaktor,Sondenlaenge,
Sondendurchmesser,Dicke_Sondenrohr,
Bohrdurchmesser,Massenstrom,TGrad,
Jahresmitteltemp,Bodenerwaermung,
DeltaT,Auslegungsmassenstrom,TSource,
Sicherheit1,Sicherheit2,R1,Ra,Rb,
Genauigkeit,adiabat,g1,g2,g3,g4,g5,
Sondenabstand,g_Sondenabstand,TSink,
QSource                                     : real;
var cpErde,rhoErde,lambdaErde              : Vektor10;
var RepRandbed,Zeitschritt,Simulationsdauer,
DimAxi,DimRad,MonitorAxi,MonitorRad,
Jahr,StepWrite,LastYear,AnzahlSonden,
DauerLastspitze,ErrorMldg                 : integer;
var numrows                                : longint;
var readFile,writeFile,
Einschwingen,Leistungsinput,Leistung,
statinaer,gfunction,Starttemp,Druck,
AlteInputfiles,swewsInput,Allsteps       : boolean;

```

```

var IFileName, OFileName, swewsName, Sprache : string);
var i, k : integer;
    sim : longint;
begin
  DimRad := Def_DimRad;
  DimAxi := Def_DimAxi;
  cpSole := Def_cpSole;
  cpFill := Def_cpFill;
  rhoSole := Def_rhoSole;
  rhoFill := Def_rhoFill;
  lambdaSole := Def_lambdaSole;
  lambdaFill := Def_lambdaFill;
  nueSole := Def_nueSole;
  cpErd := Def_cpErd;
  rhoErd := Def_rhoErd;
  lambdaErd := Def_lambdaErd;
  for i:=1 to 10 do begin
    cpErd[i] := Def_cpErd;
    rhoErd[i] := Def_rhoErd;
    lambdaErd[i] := Def_lambdaErd;
  end;
  RepRandbed := Def_RepRandbed;
  Rechenradius := Def_Rechenradius;
  Gitterfaktor := Def_Gitterfaktor;
  AnzahlSonden := Def_AnzahlSonden;
  Sondenlaenge := Def_Sondenlaenge;
  Sondendurchmesser := Def_Sondendurchmesser;
  Dicke_Sondenrohr := Def_Dicke_Sondenrohr;
  Bohrdurchmesser := Def_Bohrdurchmesser;
  R1 := Def_R1;
  Ra := Def_Ra;
  Rb := Def_Rb;
  Massenstrom := Def_Massenstrom;
  Auslegungsmassenstrom := Def_Massenstrom;
  TGrad := Def_TGrad;
  Zeitschritt := Def_Zeitschritt;
  Simulationsdauer := Def_Simulationsdauer;
  Jahresmitteltemp := Def_Jahresmitteltemp;
  Bodenerwaermung := Def_Bodenerwaermung;
  Bodentemp := Jahresmitteltemp + Bodenerwaermung;
  DeltaT := Def_DeltaT;
  TSource := Def_TSource;
  TSink := Def_TSink;
  QSource := Def_QSource;
  Sicherheit1 := Def_Sicherheit1;
  Sicherheit2 := Def_Sicherheit2;
  IFileName := Def_IFileName;
  OFileName := Def_OFileName;
  swewsName := Def_swewsName;
  readFile := Def_readFile;
  writeFile := Def_writeFile;
  MonitorAxi := Def_MonitorAxi;
  MonitorRad := Def_MonitorRad;
  sim := Simulationsdauer;
  numrows := sim * 60 div Zeitschritt;
  Einschwingen := Def_Einschwingen;
  Leistungsinput := Def_Leistungsinput;
  Leistung := Def_Leistung;
  stationaer := Def_stationaer;
  Genauigkeit := Def_Genauigkeit;
  Sprache := Def_Sprache;
  StepWrite := Def_StepWrite;
  Jahr := Def_Jahr;
  adiabat := Def_adiabat;
  Druck := Def_Druck;
  g1 := Def_g1;
  g2 := Def_g2;
  g3 := Def_g3;
  g4 := Def_g4;
  g5 := Def_g5;
  gfunction := Def_gfunction;
  Sondenabstand := Def_Sondenabstand;
  g_Sondenabstand := Def_g_Sondenabstand;
  Starttemp := Def_Starttemp;
  LastYear := Def_LastYear;
  AlteInputfiles := Def_AlteInputfiles;
  swewsInput := Def_swewsInput;
  Allsteps := Def_Allsteps;

```

```

Laufzeit[1]      := Def_Laufzeit_Jan;
Laufzeit[2]      := Def_Laufzeit_Feb;
Laufzeit[3]      := Def_Laufzeit_Mar;
Laufzeit[4]      := Def_Laufzeit_Apr;
Laufzeit[5]      := Def_Laufzeit_Mai;
Laufzeit[6]      := Def_Laufzeit_Jun;
Laufzeit[7]      := Def_Laufzeit_Jul;
Laufzeit[8]      := Def_Laufzeit_Aug;
Laufzeit[9]      := Def_Laufzeit_Sep;
Laufzeit[10]     := Def_Laufzeit_Okt;
Laufzeit[11]     := Def_Laufzeit_Nov;
Laufzeit[12]     := Def_Laufzeit_Dez;
QSpitzeFeb       := Def_QSpitzeFeb;
DauerLastSpitze  := Def_DauerLastSpitze;
Sondenlaufzeit   := 0;
ErrorMldg        := 0;

```

(***** nachfolgend neuer Ausgabefile - Kopf ab Mai 1999 *****)

```

if not AlteInputfiles then begin
  i      := 1;
  line[i] := 'Ausgabefile';           i:=i+1;
  line[i] := '';                       i:=i+1;
  line[i] := tab + 'Projektname';     i:=i+1;
  line[i] := tab + 'Ersteller';       i:=i+1;
  line[i] := tab + 'Anmerkung';       i:=i+1;
  line[i] := tab + 'Anzahl Sonden';   i:=i+1;
  line[i] := 'm' + tab + 'Sondenlaenge'; i:=i+1;
  line[i] := 'm' + tab + 'Bohrdurchmesser'; i:=i+1;
  line[i] := 'm' + tab + 'Sondeninnendurchmesser'; i:=i+1;
  line[i] := 'm' + tab + 'Dicke Sondenrohr'; i:=i+1;
  line[i] := 'K/W' + tab + 'Rl';       i:=i+1;
  line[i] := 'KmW' + tab + 'Ra';      i:=i+1;
  line[i] := 'Km/W' + tab + 'Rb';     i:=i+1;
  line[i] := '';                       i:=i+1;
  line[i] := '';                       i:=i+1;
  line[i] := '';                       i:=i+1;
  line[i] := '';                       i:=i+1;
  line[i] := '';                       i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Fluid'; i:=i+1;
  line[i] := 'kg/m3' + tab + 'Dichte Fluid'; i:=i+1;
  line[i] := 'J/kgK' + tab + 'cp Fluid';   i:=i+1;
  line[i] := 'm2/s' + tab + 'Viskositaeet Fluid'; i:=i+1;
  line[i] := 'kg/s' + tab + 'Auslegungsmassenstrom'; i:=i+1;
  line[i] := tab + 'Druckabfall berechnet'; i:=i+1;
  line[i] := 'kg/s' + tab + 'Massenstrom'; i:=i+1;
  line[i] := '';                       i:=i+1;
  line[i] := '';                       i:=i+1;
  line[i] := '';                       i:=i+1;
  line[i] := tab + 'SWEWS-Inputdaten einlesen'; i:=i+1;
  line[i] := '';                       i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Erde'; i:=i+1;
  line[i] := 'kg/m3' + tab + 'Dichte Erde'; i:=i+1;
  line[i] := 'J/kgK' + tab + 'cp Erde';   i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Erdel'; i:=i+1;
  line[i] := 'kg/m3' + tab + 'Dichte Erdel'; i:=i+1;
  line[i] := 'J/kgK' + tab + 'cp Erdel';   i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Erde2'; i:=i+1;
  line[i] := 'kg/m3' + tab + 'Dichte Erde2'; i:=i+1;
  line[i] := 'J/kgK' + tab + 'cp Erde2';   i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Erde3'; i:=i+1;
  line[i] := 'kg/m3' + tab + 'Dichte Erde3'; i:=i+1;
  line[i] := 'J/kgK' + tab + 'cp Erde3';   i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Erde4'; i:=i+1;
  line[i] := 'kg/m3' + tab + 'Dichte Erde4'; i:=i+1;
  line[i] := 'J/kgK' + tab + 'cp Erde4';   i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Erde5'; i:=i+1;
  line[i] := 'kg/m3' + tab + 'Dichte Erde5'; i:=i+1;
  line[i] := 'J/kgK' + tab + 'cp Erde5';   i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Erde6'; i:=i+1;
  line[i] := 'kg/m3' + tab + 'Dichte Erde6'; i:=i+1;
  line[i] := 'J/kgK' + tab + 'cp Erde6';   i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Erde7'; i:=i+1;
  line[i] := 'kg/m3' + tab + 'Dichte Erde7'; i:=i+1;
  line[i] := 'J/kgK' + tab + 'cp Erde7';   i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Erde8'; i:=i+1;
  line[i] := 'kg/m3' + tab + 'Dichte Erde8'; i:=i+1;
  line[i] := 'J/kgK' + tab + 'cp Erde8';   i:=i+1;
  line[i] := 'W/mK' + tab + 'Lambda Erde9'; i:=i+1;

```

```

line[i] := 'kg/m3' + tab + 'Dichte Erde9'; i:=i+1;
line[i] := 'J/kgK' + tab + 'cp Erde9'; i:=i+1;
line[i] := 'W/mK' + tab + 'Lambda Erde10'; i:=i+1;
line[i] := 'kg/m3' + tab + 'Dichte Erde10'; i:=i+1;
line[i] := 'J/kgK' + tab + 'cp Erde10'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := 'W/mK' + tab + 'Lambda Hinterfuellung'; i:=i+1;
line[i] := 'kg/m3' + tab + 'Dichte Hinterfuellung'; i:=i+1;
line[i] := 'J/kgK' + tab + 'cp Hinterfuellung'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := '°C' + tab + 'Jahresmitteltemperatur Luft'; i:=i+1;
line[i] := '°C/m' + tab + 'Temperaturgradient Boden'; i:=i+1;
line[i] := '°C' + tab + 'Boden waermer als Luft'; i:=i+1;
line[i] := tab + 'Rechnen mit Entzugsleistung'; i:=i+1;
line[i] := tab + 'Leistungsinput'; i:=i+1;
line[i] := 'y' + tab + 'Beginn Simulation'; i:=i+1;
line[i] := tab + 'numrows'; i:=i+1;
line[i] := 'y' + tab + 'Simulationsdauer in Jahren'; i:=i+1;
line[i] := 'h' + tab + 'Simulationszeit'; i:=i+1;
line[i] := '°C' + tab + 'DeltaT'; i:=i+1;
line[i] := tab + 'stationaere Berechnung Sole'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := tab + 'MonitorAxi'; i:=i+1;
line[i] := tab + 'MonitorRad'; i:=i+1;
line[i] := tab + 'Sicherheit1'; i:=i+1;
line[i] := tab + 'Sicherheit2'; i:=i+1;
line[i] := 'Min' + tab + 'Zeitschritt'; i:=i+1;
line[i] := 'w' + tab + 'RepRandbed'; i:=i+1;
line[i] := tab + 'Radiale Unterteilung'; i:=i+1;
line[i] := tab + 'Axiale Unterteilung'; i:=i+1;
line[i] := tab + 'Gitterfaktor'; i:=i+1;
line[i] := tab + 'Rechenradius'; i:=i+1;
line[i] := '°C' + tab + 'Genauigkeit Iteration'; i:=i+1;
line[i] := tab + 'Stepwrite'; i:=i+1;
line[i] := tab + 'readfile'; i:=i+1;
line[i] := tab + 'writefile'; i:=i+1;
line[i] := tab + 'Einschwingen aufschreiben'; i:=i+1;
line[i] := tab + 'Starttemperatur abschaetzen'; i:=i+1;
line[i] := '°C' + tab + 'Senkentemperatur (falls nicht von File)'; i:=i+1;
line[i] := 'kW' + tab + 'Entzugsleistung (falls nicht von File)'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := tab + 'g-function bei ln(t/ts) = -4'; i:=i+1;
line[i] := tab + 'g-function bei ln(t/ts) = -2'; i:=i+1;
line[i] := tab + 'g-function bei ln(t/ts) = 0'; i:=i+1;
line[i] := tab + 'g-function bei ln(t/ts) = +2'; i:=i+1;
line[i] := tab + 'g-function bei ln(t/ts) = +3'; i:=i+1;
line[i] := 'm' + tab + 'eff. Abstand der Sonden'; i:=i+1;
line[i] := 'm' + tab + 'Sondenabstand der g-function'; i:=i+1;
line[i] := tab + 'Randbedingung mit g-functions'; i:=i+1;
line[i] := tab + 'Erdreichtemperatur abschaetzen'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := tab + 'Laufzeit Januar'; i:=i+1;
line[i] := tab + 'Laufzeit Februar'; i:=i+1;
line[i] := tab + 'Laufzeit Maerz'; i:=i+1;
line[i] := tab + 'Laufzeit April'; i:=i+1;
line[i] := tab + 'Laufzeit Mai'; i:=i+1;
line[i] := tab + 'Laufzeit Juni'; i:=i+1;
line[i] := tab + 'Laufzeit Juli'; i:=i+1;
line[i] := tab + 'Laufzeit August'; i:=i+1;
line[i] := tab + 'Laufzeit September'; i:=i+1;
line[i] := tab + 'Laufzeit Oktober'; i:=i+1;
line[i] := tab + 'Laufzeit November'; i:=i+1;
line[i] := tab + 'Laufzeit Dezember'; i:=i+1;
line[i] := 'kW' + tab + 'Spitzenentzug Februar'; i:=i+1;
line[i] := 'd' + tab + 'Dauer des Spitzenentzugs'; i:=i+78;
line[i] := 'Step' + tab + 'M [kg/s]' + tab + 'TSink' + tab + 'Q [kW]' +
tab + 'TSource' + tab + 'TMonitor' + tab + 'p [Pa]' + tab + 'laminar?';
end;
(***** Ende neuer Ausgabefile - Kopf *****)

(***** Nachfolgend alter Ausgabefile - Kopf *****)
if AlteInputfiles then begin
i := 1;
line[i] := 'Ausgabefile fuer EWS-Programm'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := tab + 'numrows'; i:=i+1;
line[i] := 'Min' + tab + 'Zeitschritt'; i:=i+1;
line[i] := ''; i:=i+1;

```

```

line[i] := 'Erdwaermesonde'; i:=i+1;
line[i] := 'm' + tab + 'Sondendurchmesser (innen)'; i:=i+1;
line[i] := 'm' + tab + 'Sondenlaenge'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := 'Hinterfuellung'; i:=i+1;
line[i] := 'm' + tab + 'Bohrdurchmesser'; i:=i+1;
line[i] := 'W/mK' + tab + 'lambda'; i:=i+1;
line[i] := 'kg/m3' + tab + 'Dichte'; i:=i+1;
line[i] := 'J/kgK' + tab + 'cp'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := 'Fluid'; i:=i+1;
line[i] := 'W/mK' + tab + 'lambda'; i:=i+1;
line[i] := 'kg/m3' + tab + 'Dichte'; i:=i+1;
line[i] := 'J/kgK' + tab + 'cp'; i:=i+1;
line[i] := 'm2/s' + tab + 'kinem. Viskositaet'; i:=i+1;
line[i] := 'kg/s' + tab + 'Massenstrom'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := 'Boden'; i:=i+1;
line[i] := 'W/mK' + tab + 'lambda'; i:=i+1;
line[i] := 'kg/m3' + tab + 'Dichte'; i:=i+1;
line[i] := 'J/kgK' + tab + 'cp'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := 'Bodentemperaturen'; i:=i+1;
line[i] := '°C' + tab + 'Jahresmittel'; i:=i+1;
line[i] := '°C/m' + tab + 'Gradient'; i:=i+1;
line[i] := ''; i:=i+1;
line[i] := 'Step' + tab + 'M [kg/s]' + tab + 'TSink' + tab + 'Q [kW]' +
tab + 'TSource' + tab + 'TMonitor' + tab + 'p [Pa]' + tab + 'laminar?';
end;
(***** Ender alter Ausgabefile - Kopf *****)
end;
(*-- Procedure Default -----*)
(*--Input und Output-Files -----*)
procedure OpenSwewsFile; (* Procedure liest die Stoffwerte der Erde aus
der Datei SWEWS.DAT ein *)
var i,j,k : integer;
t : string;
ch : char;
aa,bb,Anteil: real;
LL : Vektor10;
function Zahl: real;
var t : string;
ch: char;
i : integer;
z : real;
begin
t:='';
repeat begin
read(swews,ch); t:=t + ch;
end until ch='.';
read(swews,ch); t:=t + ch;
read(swews,ch); t:=t + ch;
val(t,z,i);
Zahl := z;
read(swews,ch);
end;
begin (* procedure OpenSwewsFile *)
k:=0;
assign(swews,swewsName);
reset(swews);
if IORESULT <> 0 then begin
writeln ('File ',swewsName,' nicht gefunden. ');
writeln ('Kopiere das File ',swewsName,' in den aktuellen Katalog !');
halt(1);
end;
readln(swews);
read(swews,ch);
while ch='' do begin
k:=k+1;
repeat read(swews,ch) until ch='';
repeat read(swews,ch) until ch='';
SchichtDicke[k] := Zahl;
SchichtTemp[k] := Zahl;
SchichtLambda[k]:= Zahl;
SchichtCp[k] := Zahl;

```

```

    SchichtRho[k] := Zahl;
    read(swews,ch);
    read(swews,ch);
end;
SchichtDicke[k+1] := 2000;
SchichtTemp[k+1] := SchichtTemp[k];
SchichtLambda[k+1]:= SchichtLambda[k];
SchichtCp[k+1] := SchichtCp[k];
SchichtRho[k+1] := SchichtRho[k];

dl := Sondenlaenge / DimAxi;
for i:=1 to DimAxi do begin
    lambdaErde[i] := 0; rhoErde[i] := 0; cpErde[i] := 0;
end;
LL[0]:=0; for i:=1 to k+1 do LL[i]:=LL[i-1]+Schichtdicke[i];
for i:=1 to k+1 do begin
    for j:=1 to DimAxi do begin
        aa:=Min((LL[i]-LL[i-1]),Max(dl*j-LL[i-1],0));
        bb:=Max(0,Min((LL[i]-dl*j+dl),(LL[i]-LL[i-1])));
        Anteil := 0;
        if (aa>0.0001) and (bb>0.0001) then
            if aa<dl then Anteil:=aa/dl else if bb<dl then Anteil:=bb/dl
            else Anteil:=1;
        lambdaErde[j] := lambdaErde[j] + Anteil * SchichtLambda[i];
        rhoErde[j] := rhoErde[j] + Anteil * SchichtRho[i];
        cpErde[j] := cpErde[j] + Anteil * SchichtCp[i];
    end;
end;
(* Durchschnittswerte : *)
LambdaErd := 0; RhoErd := 0; cpErd := 0;
for i:=1 to DimAxi do begin
    LambdaErd := LambdaErd + lambdaErde[i]/DimAxi;
    RhoErd := RhoErd + rhoErde[i]/DimAxi;
    CpErd := CpErd + cpErde[i]/DimAxi;
end;
if k>1 then begin
    TGrad := (SchichtTemp[k]-SchichtTemp[1])/(LL[k]-Schichtdicke[k]/2-LL[1]/2);
    Bodenerwaermung := 0;
    Jahresmitteltemp:= SchichtTemp[1] - TGrad * SchichtDicke[1]/2;
    Bodentemp := Jahresmitteltemp;
end;
close(swews);
end; (* OpenSwewsFile *)

procedure OpenInputFile ( Inputfilename : string;
    var Zeitschritt: integer; var numrows : longint);
var i,j,k : integer;
    Wort,t : string;
    Buchstabe : char;

function Entscheidung(var word:string): boolean;
begin
    word:='';
    repeat read(ifile,Buchstabe);
        if Buchstabe = ' ' then Break;
        word := word + Buchstabe;
    until Buchstabe = ' ';
    readln(ifile,line[j]); j:=j+1;
    if (word='true') or (word='TRUE') then Entscheidung:=true else Entscheidung:=false;
end;

function Bemerkung : string;
var word : string;
begin
    word := '';
    repeat read(ifile,Buchstabe);
        if Buchstabe = ' ' then Break;
        word := word + Buchstabe;
    until Buchstabe = ' ';
    Bemerkung:=word;
    readln(ifile,line[j]); j:=j+1;
end;

begin (* OpenInputFile *)
    (***** nachfolgend neuer Eingabefile - Kopf ab Mai 1999 *****)
    if not AlteInputfiles then begin
        j:=1;

```

```

assign(ifile,Inputfilename);
reset(ifile);
readln(ifile);
readln(ifile,line[j]);
Projektname := Bemerkung;
Ersteller := Bemerkung;
Anmerkungen := Bemerkung;
readln(ifile,AnzahlSonden,line[j]);
readln(ifile,Sondenlaenge,line[j]);
readln(ifile,Bohrdurchmesser,line[j]);
readln(ifile,Sondendurchmesser,line[j]);
readln(ifile,Dicke_Sondenrohr,line[j]);
readln(ifile,R1,line[j]);
readln(ifile,Ra,line[j]);
readln(ifile,Rb,line[j]);
for k:=1 to 5 do begin readln(ifile,line[j]);
readln(ifile,lambdaSole,line[j]);
readln(ifile,rhoSole,line[j]);
readln(ifile,cpSole,line[j]);
readln(ifile,nueSole,line[j]);
readln(ifile,Auslegungsmassenstrom,line[j]);
Druck := Entscheidung(Wort1);
readln(ifile,Massenstrom,line[j]);
for k:=1 to 3 do begin readln(ifile,line[j]);
swewsInput := Entscheidung(Wort8);
readln(ifile,swewsName,line[j]);
readln(ifile,lambdaErd,line[j]);
readln(ifile,rhoErd,line[j]);
readln(ifile,cpErd,line[j]);
For k:=1 to 10 do begin
cpErde[k] := cpErd; (* J/kgK *)
rhoErde[k] := rhoErd; (* kg/m3 *)
lambdaErde[k] := lambdaErd; (* W/mK *)
end;
for k:=1 to 10 do begin
readln(ifile,lambdaErde[k],line[j]);
readln(ifile,rhoErde[k],line[j]);
readln(ifile,cpErde[k],line[j]);
if cpErde[k] < 500 then cpErde[k] := cpErd;
if rhoErde[k] < 500 then rhoErde[k] := rhoErd;
if lambdaErde[k] < 0.5 then lambdaErde[k]:= lambdaErd;
end;
readln(ifile,line[j]);
readln(ifile,lambdaFill,line[j]);
readln(ifile,rhoFill,line[j]);
readln(ifile,cpFill,line[j]);
readln(ifile,line[j]);
readln(ifile,Jahresmitteltemp,line[j]);
readln(ifile,TGrad,line[j]);
readln(ifile,Bodenerwaermung,line[j]);
Leistung := Entscheidung(Wort1);
Leistungsinput := Entscheidung(Wort2);
readln(ifile,Beginn,line[j]);
readln(ifile,numrows,line[j]);
readln(ifile,Lastyear,line[j]);
readln(ifile,Simulationsdauer,line[j]);
readln(ifile,DeltaT,line[j]);
stationaer := Entscheidung(Wort3);
readln(ifile,line[j]);
readln(ifile,MonitorAxi,line[j]);
readln(ifile,MonitorRad,line[j]);
readln(ifile,Sicherheit1,line[j]);
readln(ifile,Sicherheit2,line[j]);
readln(ifile,Zeitschritt,line[j]);
readln(ifile,RepRandbed,line[j]);
readln(ifile,DimRad,line[j]);
readln(ifile,DimAxi,line[j]);
readln(ifile,Gitterfaktor,line[j]);
readln(ifile,Rechenradius,line[j]);
readln(ifile,Genauigkeit,line[j]);
readln(ifile,Stepwrite,line[j]);
readfile := Entscheidung(Wort4);
writefile:= Entscheidung(Wort5);
Allsteps := Entscheidung(Wort9);
Starttemp:= Entscheidung(Wort10);
readln(ifile,TSink,line[j]);
readln(ifile,QSource,line[j]);
readln(ifile,line[j]);

```

```

readln(ifile,g1,line[j]);          j:=j+1;
readln(ifile,g2,line[j]);          j:=j+1;
readln(ifile,g3,line[j]);          j:=j+1;
readln(ifile,g4,line[j]);          j:=j+1;
readln(ifile,g5,line[j]);          j:=j+1;
readln(ifile,Sondenabstand,line[j]); j:=j+1;
readln(ifile,g_Sondenabstand,line[j]); j:=j+1;
gfunction := Entscheidung(Wort6);
Starttemp := Entscheidung(Wort7);
readln(ifile,line[j]);          j:=j+1;
for i:=1 to 12 do begin readln(ifile,Laufzeit[i],line[j]); j:=j+1; end;
readln(ifile,QSpitzeFeb,line[j]); j:=j+1;
readln(ifile,Dauerlastspitze,line[j]); j:=j+1;
repeat
  readln(ifile);          j:=j+1;
until j=202;
end;
(***** Ende neuer Eingabefile - Kopf *****)
(***** Nachfolgend alter Eingabefile - Kopf *****)
if AlteInputfiles then begin
  j:=1;
  assign(ifile,Inputfilename);
  reset(ifile);
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,numrows,line[j]);  j:=j+1;
  readln(ifile,Zeitschritt,line[j]); j:=j+1;
  readln(ifile,DeltaT,line[j]);   j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,Sondendurchmesser,line[j]); j:=j+1;
  readln(ifile,Sondenlaenge,line[j]); j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,Bohrdurchmesser,line[j]); j:=j+1;
  readln(ifile,lambdaFill,line[j]); j:=j+1;
  readln(ifile,rhoFill,line[j]);   j:=j+1;
  readln(ifile,cpFill,line[j]);    j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,lambdaSole,line[j]); j:=j+1;
  readln(ifile,rhoSole,line[j]);   j:=j+1;
  readln(ifile,cpSole,line[j]);    j:=j+1;
  readln(ifile,nueSole,line[j]);   j:=j+1;
  readln(ifile,Auslegungsmassenstrom,line[j]); j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,lambdaErd,line[j]); j:=j+1;
  readln(ifile,rhoErd,line[j]);   j:=j+1;
  readln(ifile,cpErd,line[j]);    j:=j+1;
  For i:=1 to DimAxi do begin
    cpErde[i]      := cpErd;      (* J/kgK *)
    rhoErde[i]    := rhoErd;      (* kg/m3 *)
    lambdaErde[i] := lambdaErd;   (* W/mK *)
  end;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,Bodentemp,line[j]); j:=j+1;
  readln(ifile,TGrad,line[j]);   j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
  readln(ifile,line[j]);          j:=j+1;
end;
(***** Ende alter Eingabefile - Kopf *****)

end;

procedure OpenOutputFile (      Outputfilename : string;
                             var Zeitschritt: integer; var numrows : longint);
var i,j,k : integer;
begin
(***** nachfolgend neuer Ausgabefile - Kopf ab Mai 1999 *****)
if not AlteInputfiles then begin
  j := 2;
  assign(ofile,Outputfilename);
  rewrite(ofile);
  writeln(ofile,'Ausgabefile des EWS-Programms');
  writeln(ofile,line[j]);          j:=j+1;

```



```

writeln(ofile,Projektname,tab,line[j]);      j:=j+1;
writeln(ofile,Ersteller,tab,line[j]);      j:=j+1;
writeln(ofile,Anmerkungen,tab,line[j]);   j:=j+1;
writeln(ofile,AnzahlSonden,tab,line[j]);  j:=j+1;
writeln(ofile,Sondenlaenge,tab,line[j]);  j:=j+1;
writeln(ofile,Bohrdurchmesser,tab,line[j]); j:=j+1;
writeln(ofile,Sondendurchmesser,tab,line[j]); j:=j+1;
writeln(ofile,Dicke_Sondenrohr,tab,line[j]); j:=j+1;
writeln(ofile,Rl,tab,line[j]);            j:=j+1;
writeln(ofile,Ra,tab,line[j]);            j:=j+1;
writeln(ofile,Rb,tab,line[j]);            j:=j+1;
for k:=1 to 5 do begin writeln(ofile,line[j]); j:=j+1; end;
writeln(ofile,lambdaSole,tab,line[j]);    j:=j+1;
writeln(ofile,rhoSole,tab,line[j]);      j:=j+1;
writeln(ofile,cpSole,tab,line[j]);       j:=j+1;
writeln(ofile,nueSole,tab,line[j]);      j:=j+1;
writeln(ofile,Auslegungsmassenstrom,tab,line[j]); j:=j+1;
writeln(ofile,Druck,tab,line[j]);        j:=j+1;
writeln(ofile,Massenstrom,tab,line[j]);  j:=j+1;
for k:=1 to 3 do begin writeln(ofile,line[j]); j:=j+1; end;
writeln(ofile,swewsinput,tab,line[j]);  j:=j+1;
writeln(ofile,swewsname);               j:=j+1;
writeln(ofile,lambdaErd,tab,line[j]);    j:=j+1;
writeln(ofile,rhoErd,tab,line[j]);      j:=j+1;
writeln(ofile,cpErd,tab,line[j]);       j:=j+1;
for k:=1 to 10 do begin
  writeln(ofile,lambdaErde[k],tab,line[j]); j:=j+1;
  writeln(ofile,rhoErde[k],tab,line[j]);  j:=j+1;
  writeln(ofile,cpErde[k],tab,line[j]);   j:=j+1;
end;
writeln(ofile,line[j]);                  j:=j+1;
writeln(ofile,lambdaFill,tab,line[j]);   j:=j+1;
writeln(ofile,rhoFill,tab,line[j]);      j:=j+1;
writeln(ofile,cpFill,tab,line[j]);      j:=j+1;
writeln(ofile,line[j]);                  j:=j+1;
writeln(ofile,Jahresmitteltemp,tab,line[j]); j:=j+1;
writeln(ofile,TGrad,tab,line[j]);       j:=j+1;
writeln(ofile,Bodenerwaermung,tab,line[j]); j:=j+1;
writeln(ofile,Leistung,tab,line[j]);    j:=j+1;
writeln(ofile,Leistungsinput,tab,line[j]); j:=j+1;
writeln(ofile,Beginn,tab,line[j]);      j:=j+1;
writeln(ofile,numrows,tab,line[j]);     j:=j+1;
writeln(ofile,LastYear,tab,line[j]);    j:=j+1;
writeln(ofile,Simulationsdauer,tab,line[j]); j:=j+1;
writeln(ofile,DeltaT,tab,line[j]);      j:=j+1;
writeln(ofile,stationaer,tab,line[j]);  j:=j+1;
writeln(ofile,line[j]);                  j:=j+1;
writeln(ofile,MonitorAxi,tab,line[j]);  j:=j+1;
writeln(ofile,MonitorRad,tab,line[j]);  j:=j+1;
writeln(ofile,Sicherheit1,tab,line[j]);  j:=j+1;
writeln(ofile,Sicherheit2,tab,line[j]);  j:=j+1;
writeln(ofile,Zeitschritt,tab,line[j]); j:=j+1;
writeln(ofile,RepRandbed,tab,line[j]);  j:=j+1;
writeln(ofile,DimRad,tab,line[j]);      j:=j+1;
writeln(ofile,DimAxi,tab,line[j]);      j:=j+1;
writeln(ofile,Gitterfaktor,tab,line[j]); j:=j+1;
writeln(ofile,Rechenradius,tab,line[j]); j:=j+1;
writeln(ofile,Genauigkeit,tab,line[j]);  j:=j+1;
writeln(ofile,StepWrite,tab,line[j]);   j:=j+1;
writeln(ofile,readfile,tab,line[j]);    j:=j+1;
writeln(ofile,writefile,tab,line[j]);   j:=j+1;
writeln(ofile,Allsteps,tab,line[j]);    j:=j+1;
writeln(ofile,Starttemp,tab,line[j]);   j:=j+1;
writeln(ofile,TSink,tab,line[j]);      j:=j+1;
writeln(ofile,QSource,tab,line[j]);     j:=j+1;
writeln(ofile,line[j]);                  j:=j+1;
writeln(ofile,g1,tab,line[j]);          j:=j+1;
writeln(ofile,g2,tab,line[j]);          j:=j+1;
writeln(ofile,g3,tab,line[j]);          j:=j+1;
writeln(ofile,g4,tab,line[j]);          j:=j+1;
writeln(ofile,g5,tab,line[j]);          j:=j+1;
writeln(ofile,Sondenabstand,tab,line[j]); j:=j+1;
writeln(ofile,g_Sondenabstand,tab,line[j]); j:=j+1;
writeln(ofile,gfunction,tab,line[j]);   j:=j+1;
writeln(ofile,Starttemp,tab,line[j]);   j:=j+1;
writeln(ofile,line[j]);                  j:=j+1;
for i:=1 to 12 do begin writeln(ofile,Laufzeit[i],tab,line[j]); j:=j+1; end;
writeln(ofile,QSpitzeFeb,tab,line[j]);  j:=j+1;

```

```

        writeln(ofile,DauerLastSpitze,tab,line[j]);      j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        repeat
            writeln(ofile,line[j]);                    j:=j+1;
        until j=202
    end;
    (***** Ende neuer Ausgabefile - Kopf *****)
    (***** nachfolgend alter Ausgabefile - Kopf *****)
    if AlteInputFiles then begin
        j := 2;
        assign(ofile,Outputfilename);
        rewrite(ofile);
        writeln(ofile,'Ausgabefile des EWS-Programms');
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,numrows,tab,line[j]);           j:=j+1;
        writeln(ofile,Zeitschritt,tab,line[j]);       j:=j+1;
        writeln(ofile,DeltaT,tab,line[j]);            j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,Sondendurchmesser,tab,line[j]);  j:=j+1;
        writeln(ofile,Sondenlaenge,tab,line[j]);     j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,Bohrdurchmesser,tab,line[j]);   j:=j+1;
        writeln(ofile,lambdaFill,tab,line[j]);        j:=j+1;
        writeln(ofile,rhoFill,tab,line[j]);           j:=j+1;
        writeln(ofile,cpFill,tab,line[j]);            j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,lambdaSole,tab,line[j]);        j:=j+1;
        writeln(ofile,rhoSole,tab,line[j]);           j:=j+1;
        writeln(ofile,cpSole,tab,line[j]);            j:=j+1;
        writeln(ofile,nueSole,tab,line[j]);           j:=j+1;
        writeln(ofile,Auslegungsmassenstrom,tab,line[j]); j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,lambdaErd,tab,line[j]);         j:=j+1;
        writeln(ofile,rhoErd,tab,line[j]);           j:=j+1;
        writeln(ofile,cpErd,tab,line[j]);            j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,Bodentemp,tab,line[j]);        j:=j+1;
        writeln(ofile,TGrad,tab,line[j]);            j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
        writeln(ofile,line[j]);                        j:=j+1;
    end;
    (***** Ende alter Ausgabefile - Kopf *****)
end;

Procedure ReadStep (var filestep: longint; var Massenstrom,TSink,QSource: real);
begin
    if Leistungsinput then readln(ifile,filestep,Massenstrom,TSink,QSource)
    else readln(ifile,filestep,Massenstrom,TSink);
end;

Procedure WriteStep (var filestep: longint; var Massenstrom,TSink,QSource,
                    TSource,TMonitor,p: real; laminar : boolean );
const tab = chr(9);
var i,j: integer;
begin
    write(ofile,filestep,tab,Massenstrom,tab,TSink,tab,QSource,tab,TSource,
          tab,TMonitor,tab,p,tab,laminar);
    writeln(ofile);
end;

(* Erzeugen eines Inputfiles mit stuendlichem Lastprofil *)
Procedure InputFileErzeugen (Inputfile          : string;
                             Laufzeit          : Vektor12;
                             Massenstrom,Sondenleistung,
                             TSink,QSpitzeFeb : real;
                             var Sondenlaufzeit : real;
                             DauerLastSpitze   : integer );
var k      : integer;
    l      : string;
    t,t1   : text;

```

```

Function Monatslaenge(Month : integer):integer;
var i: integer;
begin
  case Month of 1 : i:= 31; 2 : i:= 28; 3 : i:= 31; 4 : i:= 30;
                5 : i:= 31; 6 : i:= 30; 7 : i:= 31; 8 : i:= 31;
                9 : i:= 30; 10 : i:= 31; 11 : i:= 30; 12 : i:= 31; end;
  Monatslaenge:=i; (* Tage *)
end;

Function Monatsbeginn(Month : integer):integer;
var i: integer;
begin
  case Month of 1 : i:= 0; 2 : i:= 31; 3 : i:= 59; 4 : i:= 90;
                5 : i:= 120; 6 : i:= 151; 7 : i:= 181; 8 : i:= 212;
                9 : i:= 243; 10 : i:= 273; 11 : i:= 304; 12 : i:= 334; end;
  Monatsbeginn:=1+24*i; (* Stunden *)
end;

procedure Entzugsprofil (var t:Text; Monat: integer; Laufzeit: vektor12; m,Q: real);
var Tag,j : integer;
begin
  for Tag:=1 to Monatslaenge(Monat) do begin (* Januar *)
    if Laufzeit[Monat]>24 then Laufzeit[Monat]:=24;
    for j:=1 to round(Laufzeit[Monat]) do begin
      if (Monat=2) then
        if Tag>(Monatslaenge(Monat)-Dauerlastspitze) then
          writeln(t,Monatsbeginn(Monat)-1+24*(Tag-1)+j,tab,m,tab,TSink,tab,QSpitzeFeb)
        else writeln(t,Monatsbeginn(Monat)-1+24*(Tag-1)+j,tab,m,tab,TSink,tab,Q)
        else writeln(t,Monatsbeginn(Monat)-1+24*(Tag-1)+j,tab,m,tab,TSink,tab,Q);
      end;
      if Laufzeit[Monat]<24 then for j:=round(Laufzeit[Monat])+1 to 24 do begin
        if (Monat=2) then
          if Tag>(Monatslaenge(Monat)-Dauerlastspitze) then
            writeln(t,Monatsbeginn(Monat)-1+24*(Tag-1)+j,tab,m,tab,TSink,tab,QSpitzeFeb)
          else writeln(t,Monatsbeginn(Monat)-1+24*(Tag-1)+j,tab,0.0,tab,TSink,tab,0.0)
          else writeln(t,Monatsbeginn(Monat)-1+24*(Tag-1)+j,tab,0.0,tab,TSink,tab,0.0);
        end;
      end;
    end;
  end;

begin (* Procedure InputFileErzeugen *)
  Leistunginput := true;
  Leistung := true;
  Sondenlaufzeit := 0;
  for k:=1 to 12 do Sondenlaufzeit := Sondenlaufzeit + Laufzeit[k]*Monatslaenge(k)
                  + DauerLastspitze * (24 - Laufzeit[2]);
  OpenOutputFile(Inputfile,Zeitschritt,numrows);
  for k:=1 to 12 do Entzugsprofil(ofile,k,Laufzeit,Massenstrom,Sondenleistung);
  close(ofile);
end; (* Procedure InputFileErzeugen *)

(*--Input und Output-Files -----*)

(*-- Procedure InitEWS -----*)
procedure InitEWS (
  StartJahr : integer;
  QSonde : Vektor;
  var rhoFill,cpFill,lambdaFill,rhoSole,
      cpSole,lambdaSole,nueSole,
      Sondenlaenge,Sondendurchmesser,
      Dicke_Sondenrohr,Bohrdurchmesser,
      Rl,Ra,Rb,Massenstrom,Bodentemp,
      Tgrad,Rechenradius,Gitterfaktor,adiabat: real;
  var Zeitschritt,subdt,substep_run,
      substep_stop,Jahr : integer;
  var B1,B2 : KMatrix;
  var TEarth : Matrix;
  var T0,TUp,TUold,TDown,TDownold,Llast : Vektor;
  var cpErde,RhoErde,lambdaErde : Vektor10;
  var Q : MatrixQ;
  var mcpSole,Llrun,Llstop,dl,TSource : real;
  DimAxi,DimRad : integer );

var
  i, j, k : integer;
  h1, h2, h3, h4, h5, alpha,
  Di, Faktor, L, dt, Rechengebiet, LmMin, L0 : real;
  R2 : Vektor;

```

```

r, rz                                     : VektorRad;
L1, L2, C                                 : Matrix;
A1, A2, Ainv1, Ainv2, F1, F2             : KMatrix;

Procedure invertieren (var A,Ainv : kMatrix; DimAxi,DimRad : integer);
{ Diese Procedure invertiert die n x n Matrix A nach der Diagonalen-Methode,
  ***** Ainv = 1/A und n = DimRad+1 ***** }
var
  pivot      : real;
  g,i,j,k    : integer;
begin
  for k:=1 to DimAxi do begin
    for i:=0 to DimRad+1 do for j:=0 to DimRad+1 do Ainv[k,i,j]:=A[k,i,j];
    for g:=0 to DimRad+1 do begin
      pivot:=Ainv[k,g,g];
      for j:=0 to DimRad+1 do Ainv[k,g,j]:=Ainv[k,g,j] * (-1) / pivot;
      for i:=0 to DimRad+1 do begin
        for j:=0 to DimRad+1 do
          if (i <> g) and (j <> g) then
            Ainv[k,i,j]:=Ainv[k,g,j] * Ainv[k,i,g] + Ainv[k,i,j];
            Ainv[k,i,g]:=Ainv[k,i,g] / pivot;
          end;
          Ainv[k,g,g]:=1 / pivot ;
        end;
      end;
    end;
  end;

Function Max (x1,x2 : real) : real;
begin
  if x1>x2 then Max:=x1 else Max:=x2;
end;

Procedure Polynom ( g1,g2,g3,g4,g5,Sondenabstand,g_Sondenabstand : real;
  var gpar1,gpar2,gpar3,gpar4,gpar5,gpar6 : real );
var
  pivot,BH,g10,g20,g30,g40,g50,ExA,ExB : real;
  g,i,j                                 : integer;
  A,Ainv                                 : Matrix6;
  x,y,w                                  : Vektor10;
begin
  g10 := g1; g20 := g2; g30 := g3; g40 := g4; g50 := g5;
  if abs(Sondenabstand - g_Sondenabstand) > 0.05 then
    begin (* Extrapolation der g-Function *)
      BH := Sondenabstand / g_Sondenabstand;
      if BH < 0.4 then ErrorMldg := 1;
      UMin:= -4;
      ExA := g5 - 6.29;
      ExB := -ln((g3-6.29)/(g5-6.6))/27;
      g10 := Max(4.82 + ExA/BH*exp(-BH*ExB*343),4.82);
      if abs(g10-4.82)<1.5 then g10:=4.82 else g10:=(g10+4.82)/2;
      g20 := Max(5.69 + ExA/BH*exp(-BH*ExB*125),5.69);
      g30 := Max(6.29 + ExA/BH*exp(-BH*ExB*27),6.29);
      g40 := Max(6.57 + ExA/BH*exp(-BH*ExB),6.57);
      g50 := Max(6.6 + ExA/BH,6.6);
    end; (* Ende Extrapolation g-Function *)
  { Berechnet die g-function aus 4 Stuetztstellen g1,g2,g3,g4 }
  x[1]:=-4; x[2]:=-2; x[3]:=0; x[4]:=2.5; x[5]:=3;
  x[6]:=min(-4.5,(-4 - (g10-4.82)/2));
  UMin := max((x[6] + 0.5),-6);
  y[1]:=g10; y[2]:=g20; y[3]:=g30; y[4]:=g40; y[5]:=g50*0.99;
  y[6] := (ln(0.5/0.0005) + 0.5 * x[6]) * 0.95;
  y[4] := (y[4]+y[5])/2*0.99;
  for i:=1 to 6 do begin
    A[i,1] := 1;
    for j:=2 to 6 do A[i,j] := A[i,j-1] * x[i];
  end;
  { invertiert die 6 x 6 Matrix A nach der Diagonalen-Methode }
  for i:=1 to 6 do for j:=1 to 6 do Ainv[i,j]:=A[i,j];
  for g:=1 to 6 do begin
    pivot:=Ainv[g,g];
    for j:=1 to 6 do Ainv[g,j]:=Ainv[g,j] * (-1) / pivot;
    for i:=1 to 6 do begin
      for j:=1 to 6 do
        if (i <> g) and (j <> g) then
          Ainv[i,j]:=Ainv[g,j] * Ainv[i,g] + Ainv[i,j];
          Ainv[i,g]:=Ainv[i,g] / pivot;
        end;
        Ainv[g,g]:=1 / pivot ;
      end;
    end;
  end;
end;

```

```

end; { Ende invertieren }
{ w = Ainv * y }
for i:=1 to 6 do w[i]:=0;
for i:=1 to 6 do begin
  for j:=1 to 6 do w[i]:=w[i]+Ainv[i,j]*y[j];
end;
gpar1 := w[1]; gpar2 := w[2]; gpar3 := w[3]; gpar4 := w[4];
gpar5 := w[5]; gpar6 := w[6];
end;

Procedure multiMatrix (var Ainv,F,B : kmatrix; DimAxi,DimRad : integer );
{ ***** B = Ainv x F ***** }
var i,j,k,l : integer;
begin
  for l:=1 to DimAxi do begin
    for k:=0 to DimRad+1 do begin
      for i:=0 to DimRad+1 do begin
        B[l,k,i] := 0;
        for j:= 0 to DimRad+1 do B[l,k,i]:=B[l,k,i]+Ainv[l,k,j]*F[l,j,i];
end;
end;
end;
end;

Function hoch (a, b: real): real; (* a^b : real *)
begin
  hoch := exp(b * ln(abs(a)));
end;

Function alpha0 ( lambdaSole,Sondendurchmesser : real );
(* Waermeuebergang, wenn Pumpe steht *)
begin
  alpha0 := lambdaSole/(Sondendurchmesser/2*(1-sqrt(0.5)));
end;

Function alphas(NueSole,rhoSole,cpSole,lambdaSole,Massenstrom,
  Sondendurchmesser,Dicke_Sondenrohr : real) : real;
(* Function alphas: Waermeuebergang Sole-Hinterfuellung, wenn Pumpe laeuft *)
var x,Geschw,Re,Pr,Nu_turbulent,Nu_laminar,
  Nu0,Nu,Di,St,Xi0,Xi,K1,K2,K10,St0 : real;
begin
  Di := Sondendurchmesser - 2 * Dicke_Sondenrohr;
  Geschw := 2*Massenstrom/rhoSole/sqr(Di)/Pi;
  Re := Geschw * Di / nueSole; (* Reynoldszahl *)
  Pr := NueSole * rhoSole * cpSole / lambdaSole; (* Prandtlzahl *)
  (* Xi = Druckverlustkoeffizient nach Petukhov (1970) *)
  Xi := 1 / sqr(1.82*ln(Re)/ln(10) - 1.64);
  (* Stantonzahl nach Petukhov (1970), gueltig fuer turbulenten Bereich *)
  K1 := 1 + 27.2 * Xi / 8;
  K2 := 11.7 + 1.8 / hoch(Pr,1/3);
  St := Xi/8 / ( K1 + K2 * sqrt(Xi/8) * (hoch(Pr,2/3)-1)); (* Stantonzahl *)
  (* Stantonzahl nach Petukhov an der Grenze turbulent-Uebergangszone *)
  Xi0:= 0.031437;
  K10:= 1.106886;
  ST0:= Xi0/8 / ( K10 + K2 * sqrt(Xi0/8) * (hoch(Pr,2/3)-1));
  Nu0:= St0*10000*Pr; (* Nusseltzahl beim Uebergang turbulent-Uebergangszone *)
  Nu_turbulent := St * Re * Pr; (* Nusseltzahl turbulente Zone *)
  Nu_laminar := 4.36; (* Nusseltzahl laminare Zone *)
  if Re >= 10000 then Nu := Nu_turbulent; (* turbulent *)
  if Re <= 2300 then Nu := Nu_laminar (* laminar *)
  else if Re < 10000 then Nu := Nu_laminar (* Uebergangszone laminar/turbulent *)
    * exp(ln(Nu0/Nu_laminar) / ln(10000/2300) * ln(Re/2300));
  x := Nu * lambdaSole / Di;
  alphas := x;
end;

Procedure Anfangstemp( TMittel, TGrad, dl : real;
  qEntzug : Vektor;
  var TEarth : Matrix;
  var T0,TUp,TUpold,TDown,TDownold : Vektor;
  var TSource : real;
  DimAxi,DimRad,StartJahr : integer);
(* Procedure Anfangstemp: Ungestoerte Erde *)
var i,j : integer;
  ts,u : real;
  go,gl : real;

```

```

    g,Rq    : VektorRad;
begin
    ts := sqr(dl*DimAxi) /9 / lambdaErd * rhoErd * cpErd;
    if StartJahr>0 then u:=ln(StartJahr*365)+ln(24*3600)-ln(ts) else u:=0;
    if u>2.5 then u:=2.5;
    go := 0.5 * u + 6.907755;
    if u<uMin then gl := go
        else gl := gpar1 + gpar2*u + gpar3*sqr(u) + gpar4*sqr(u)*u +
            gpar5*sqr(sqr(u)) + gpar6*sqr(sqr(u))*u;
    if u<-2 then if (go-0.3)>gl then gl := go;
    for j:=0 to DimRad+1 do begin
        g[j] := gl - ln(rz[j]/dl/DimAxi/0.0005);
        if StartJahr=0 then Rq[j]:=0 else Rq[j]:= g[j] / 2 / Pi / lambdaErd;
    end;
    for i:=1 to DimAxi do begin
        for j := 0 to DimRad + 1 do
            TEarth[i,j] := TMittel + (i*dl-dl/2)*TGrad - qEntzug[i]/Sondenlaenge*DimAxi*Rq[j];
        T0[i] := TEarth[i,DimRad+1];
        TUp[DimAxi-i+1] := TEarth[i,0];
        TUpold[DimAxi-i+1] := TEarth[i,0];
        TDown[i] := TEarth[i,0];
        TDownold[i] := TEarth[i,0];
    end;
end;

Procedure AnfangsQ(var Q: MatrixQ);
var i,j : integer;
begin
    for i:=1 to DimAxi do begin
        for j:=0 to 53*MaxJahre do Q[i,j] := 0;
        Q0[i] := 0;
        Q0old[i] := 0;
    end;
end;

Procedure DefMatrixA (var L,C : Matrix; var dt : real; var A : kMatrix;
    DimAxi,DimRad : integer);
var i,j,k : integer;
begin
    for i:=1 to DimAxi do begin
        for j:=0 to DimRad+1 do for k:=0 to DimRad+1 do A[i,j,k]:=0;
        A[i,0,0] := 1; A[i,DimRad+1,DimRad+1] := 1;
        for j:=1 to DimRad do begin
            A[i,j,j] := 2 * C[i,j] + dt * ( L[i,j] + L[i,j+1] );
            A[i,j,j-1] := -dt * L[i,j];
            A[i,j,j+1] := -dt * L[i,j+1];
        end;
    end;
end;

Procedure DefMatrixF ( var L,C : Matrix; var dt : real; var F : kMatrix;
    DimAxi,DimRad : integer);
var i,j,k : integer;
begin
    for i:=1 to DimAxi do begin
        for j:=0 to DimRad+1 do for k:=0 to DimRad+1 do F[i,j,k]:=0;
        F[i,0,0] := 1; F[i,DimRad+1,DimRad+1] := 1;
        for j:=1 to DimRad do begin
            F[i,j,j] := 2 * C[i,j] - dt * ( L[i,j] + L[i,j+1] );
            F[i,j,j-1] := dt * L[i,j];
            F[i,j,j+1] := dt * L[i,j+1];
        end;
    end;
end;

Function Optimaler_Zeitfaktor (L,C : Matrix) : integer;
var i,Optfak : integer;
    Mindx,Mindt,Mindt2 : real;
begin
    Mindt := 3600;
    for i:=1 to DimAxi do begin
        Mindt2 := C[i,1] / L[i,2] / Sicherheit2;
        if Mindt2<Mindt then Mindt := Mindt2;
        Mindt2 := C[i,2] / L[i,2] / Sicherheit2;
        if Mindt2<Mindt then Mindt := Mindt2;
        if not stationaer then Mindt2 := C[i,1] / L[i,1] / Sicherheit2
        else Mindt2 := C[i,1] / L[i,1] / Sicherheit2 * 1.0;
        if Mindt2<Mindt then Mindt := Mindt2;
    end;
end;

```

```

end;
Optfak := trunc(Zeitschritt*60/Mindt);
if Optfak=0 then Optfak := 1;
Optimaler_Zeitfaktor := Optfak;
end;

begin (* InitEWS *)
if DimAxi>DimAxiMax then writeln('DimAxi > DimAxiMax !');
if DimRad>DimRadMax then writeln('DimRad > DimAxiRad !');
TSourceMin := 1000; TSourceMax := -1000;
TSinkMin := 1000; TSinkMax := -1000;
For i:=0 to DimAxi do VektorNull[i] := 0;
Polynom(g1,g2,g3,g4,g5,Sondenabstand,g_Sondenabstand,
gpar1,gpar2,gpar3,gpar4,gpar5,gpar6); (* g-function *)
Rechengebiet := Rechenradius - Bohrdurchmesser / 2;
dl := Sondenlaenge/DimAxi; (* m *)
mcpSole := 2*cpSole*rhoSole*Pi*dl*sqr(Sondendurchmesser/2);(* kJ/K *)
rhoErd := 0; for i:=1 to DimAxi do rhoErd := rhoErd + rhoErde[i]/DimAxi;
cpErd := 0; for i:=1 to DimAxi do cpErd := cpErd + cpErde[i]/DimAxi;
lambdaErd := 0; for i:=1 to DimAxi do lambdaErd:= lambdaErd+lambdaErde[i]/DimAxi;
r[0] := Sondendurchmesser / 2;
r[1] := Bohrdurchmesser / 2;
Faktor:= Rechengebiet*(1-Gitterfaktor) / (1-hoch(Gitterfaktor,DimRad-1));
for i := 2 to DimRad do r[i] := r[i-1] + Faktor * hoch(Gitterfaktor, i-2);
for i := 1 to DimRad do rz[i] := sqrt((sqr(r[i])+sqr(r[i-1]))/2);
rz[0] := r[0]; rz[DimRad+1] := r[DimRad];
alpha := alphas(NueSole,rhoSole,cpSole,lambdaSole,Massenstrom/AnzahlSonden,
Sondendurchmesser,Dicke_Sondenrohr);
L0 := cpSole * Massenstrom / AnzahlSonden;

(* Widerstaende im Bohrloch *)
if (( Ra > 0 ) and ( Rb > 0 )) then begin
R1 := Ra/4/dl;
for i:= 1 to DimAxi do
R2[i] := (Rb - Ra/4)/dl + 1/2/Pi/dl*ln(rz[2]/r[1])/lambdaErde[i];
Llrun := 1 / R1;
Llstop:= 1 / (R1 - 1/8/Pi/alpha/r[0]/dl
+ 1/8/Pi/alpha0(lambdaSole,Sondendurchmesser)/r[0]/dl);
end else if Rb > 0 then begin
R1 := Rb/dl - 1/2/Pi/dl*ln(r[1]/rz[1])/lambdaFill;
for i:=1 to DimAxi do
R2[i] := 1/2/Pi/dl*(ln(r[1]/rz[1])/lambdaFill+ln(rz[2]/r[1])/lambdaErde[i]);
Llrun := 1 / R1;
Llstop:= 1 / (R1 - 1/8/Pi/alpha/r[0]/dl
+ 1/8/Pi/alpha0(lambdaSole,Sondendurchmesser)/r[0]/dl);
Ra := R1 * 4 * dl;
end else if R1 > 0 then begin
for i:=1 to DimAxi do
R2[i] := 1/2/Pi/dl*(ln(r[1]/rz[1])/lambdaFill+ln(rz[2]/r[1])/lambdaErde[i]);
Llrun := 1 / R1;
Llstop:= 1 / (R1 - 1/8/Pi/alpha/r[0]/dl
+ 1/8/Pi/alpha0(lambdaSole,Sondendurchmesser)/r[0]/dl);
Ra := R1 * 4 * dl;
Rb := R1*dl + 1/2/Pi*ln(r[1]/rz[1])/lambdaFill;
end else begin
R1 := 1/8/Pi/dl * (1/alpha/r[0] + ln((r[1]-rz[1])/r[0])/lambdaFill);
for i:=1 to DimAxi do
R2[i] := 1/2/Pi/dl*(ln(r[1]/rz[1])/lambdaFill+ln(rz[2]/r[1])/lambdaErde[i]);
Llrun := 1 / R1;
Llstop:= 8/(1/Pi/alpha0(lambdaSole,Sondendurchmesser)/r[0]/dl
+ ln((r[1]-rz[1])/r[0])/Pi/lambdaFill/dl);
Ra := R1 * 4 * dl;
Rb := R1*dl + 1/2/Pi*ln(r[1]/rz[1])/lambdaFill;
end;

(* Pumpe laeuft *)
for i := 1 to DimAxi do begin
L1[i,1] := Llrun;
L1[i,2] := 1 / R2[i];
for j := 3 to DimRad do L1[i,j] := 1/(ln(rz[j]/rz[j-1])/2/Pi/lambdaErde[i]/dl);
Llast[i] := (1-adiabat)/(1/2/Pi/dl/lambdaErde[i]*ln(r[DimRad]/rz[DimRad]));
L1[i,DimRad+1] := Llast[i];
end;

(* Pumpe laeuft nicht *)
for i := 1 to DimAxi do begin
L2[i,1] := Llstop;

```

```

for j := 2 to DimRad+1 do L2[i,j] := L1[i,j];
end;

(* Waermekapazitaeten *)
for i:=1 to DimAxi do begin
  C[i,1] := cpFill * rhoFill * Pi * (sqr(r[1]) - 4*sqr(r[0])) * dl;
  for j:=2 to DimRad do C[i,j]:=cpErde[i]*rhoErde[i]*Pi*(sqr(r[j])-sqr(r[j-1]))*dl;
end;
(* Bestimmen der optimalen Zeitschrittlaege *)
subdt := Optimaler_Zeitfaktor(L1,C);
dt := Zeitschritt * 60 / subdt; (* s *)

(* Definition der Matrizen, wobei A * Tneu = F * Talt *)
DefMatrixA(L1,C,dt,A1,DimAxi,DimRad); (* Pumpe laeuft *)
DefMatrixF(L1,C,dt,F1,DimAxi,DimRad);
invertieren(A1,Ainv1,DimAxi,DimRad);
Multimatrix(Ainv1,F1,B1,DimAxi,DimRad);

DefMatrixA(L2,C,dt,A2,DimAxi,DimRad); (* Pumpe laeuft nicht *)
DefMatrixF(L2,C,dt,F2,DimAxi,DimRad);
invertieren(A2,Ainv2,DimAxi,DimRad);
Multimatrix(Ainv2,F2,B2,DimAxi,DimRad);

Anfangstemp(Bodentemp,TGrad,dl,QSonde,TEarth,T0,TUp,TUpold,TDown,
TDownOld,TSource,DimAxi,DimRad,Startjahr);
if not Starttemp then AnfangsQ(Q)
else if Jahr=1 then AnfangsQ(Q)
else for i:=1 to DimAxi do Q[i,0]:=Q[i,52];

(* Berechnung der Anzahl substeps *)
LmMin := mcpSole / L0;
if (mcpSole/Llrun) < LmMin then LmMin := (mcpSole/Llrun);
if stationaer then begin
  substep_run := 1;
  substep_stop:= 1;
end else begin
  substep_run := trunc(Sicherheit1*dt/LmMin)+1;
  substep_stop:= trunc(Sicherheit1*dt/mcpSole*Llstop)+1;
end;
end;
(*-- Procedure InitEWS -----*)

(*-- Procedure EWSmodul -----*)
Procedure EWSmodul( iteration : integer;
var ErrorMldg,DimAxi,DimRad,Zeitschritt,
subdt,substep_run,substep_stop,RepRandbed,
MonitorAxi,MonitorRad,Jahr,LastYear,Beginn,
AnzahlSonden,DauerLastspitze : integer;
var numrows : longint;
var rhoFill,cpFill,lambdaFill,rhoSole,cpSole,
lambdaSole,nueSole,R1,Ra,Rb,Sondenlaenge,
Sondendurchmesser,Dicke_Sondenrohr,
Bohrdurchmesser,Auslegungsmassenstrom,
Massenstrom,Bodentemp,Tgrad,mcpSole,Llrun,
Llstop,dl,Rechenradius,Rechengebiet,
Gitterfaktor,Sicherheit1,Sicherheit2,
DeltaT,TSourceOld,TSource,TSink,TMonitor,p,
QSource,Genauigkeit,adiabat,g1,g2,g3,g4,g5,
Sondenabstand,g_Sondenabstand,QSpitzeFeb,
TSourceMin,TSourceMax,TSinkMin,TSinkMax,
Sondenlaufzeit : real;
var B1,B2 : KMatrix;
var TEarth,TEarthold : Matrix;
var T0,TUp,TUpOld,TDown,TDownOld,Llast : Vektor;
var cpErde,rhoErde,lambdaErde : Vektor10;
var Q : MatrixQ;
var Laufzeit : Vektor12;
var readFile,writeFile,laminar,
Einschwingen,Leistungsinput,Leistung,
stationaer,gfunction,Starttemp,Druck,
AlteInputfiles,SwewsInput,Allsteps : boolean;
var IFileName,OFileName,swewsName,Sprache,Wort: string );

var
Gefriergrenze,L0,Summe_TSource,TRT: real;
i,j,k,Woche,idt : integer;
Tneu,Tadneu : Matrix;
QWand : Vektor;

```



```

Pumpelauff : boolean;

Procedure multiplizieren (var M : kmatrix; var w,y : Matrix;
                          k,DimRad : integer);
{ ***** y = M x w ***** }
var i,j : integer;
begin
  y[k,0] := w[k,0]; y[k,DimRad+1] := w[k,DimRad+1];
  for i:=1 to DimRad do begin
    y[k,i] := 0;
    for j:=0 to DimRad+1 do y[k,i]:=y[k,i] + M[k,i,j] * w[k,j];
  end;
end;

Function Fakultae(x:integer):comp;
var i,y : longint;
begin
  y:=1;
  for i:=1 to x do y:=y*i;
  Fakultae:=y;
end;

Function hoch (a: real; b: longint): extended; (* a^b : real *)
begin
  if a>0 then hoch:=exp(b * ln(a))
  else if odd(b) then hoch:=-exp(b * ln(-a))
  else hoch:=exp(b * ln(-a));
end;

Function Druckabfall(NueSole,RhoSole,Massenstrom,Sondendurchmesser,
                     Dicke_Sondenrohr:real; var laminar:boolean) : real;
var Di, wi, Re, Xi, nue : real;
begin
  Di := Sondendurchmesser - 2 * Dicke_Sondenrohr;
  wi := Massenstrom / 2 / Pi / sqrt(Di/2) / RhoSole;
  Re := wi * Di / nueSole;
  laminar := false;
  if Re>0 then begin
    if Re < 2300 then begin
      Xi := 64 / Re;
      laminar := true;
    end {else if Re < 1000 then Xi := 0.316 / sqrt(sqrt(Re)) (* Bose, 1985*)}
    else Xi := 1 / sqrt(1.82*ln(Re)/ln(10) - 1.64); (* Petukhov, 1970 *)
  end else xi:=0;
  Druckabfall := Sondenlaenge * Xi / Di * RhoSole * sqrt(wi) / 2;
end;

Procedure RandAussen (var TRT : real;
                      k,Woche,Zeitschritt : integer;
                      simstep : longint;
                      RepRandbed : integer;
                      Q : MatrixQ;
                      cpErd,rhoErd,lambdaErd,
                      Rechenradius,Sondenlaenge : real;
                      DimAxi : integer);
{ Diese Procedure berechnet die Randbedingung nach der Trichterformel
  von Werner ***** }
var u,u0,W,W_alt,STrt,STrt0 : extended;
    i : integer;
    j : integer;
begin
  u0:=sqrt(Rechenradius)*cpErd*rhoErd/(4*lambdaErd);
  STrt0:=4*Pi*lambdaErd*Sondenlaenge/DimAxi;
  STrt:= 0;
  for i:= 1 to Woche do begin
    u:=u0/(i*604800*RepRandbed);
    if u > 1 then STrt := 0
    else begin
      W:=-0.5772-ln(u);
      j:=1;
      W_alt:=W-hoch(-1,j)*hoch(u,j)/(j*Fakultae(j));
      W:=W_alt;
      repeat
        W_alt:=W;
        j:=j+1;
        W:=W-hoch(-1,j)/j*hoch(u,j)/(Fakultae(j));
      until (abs(1-W/W_alt) < 0.01);
    end;
  end;
end;

```

```

        STrt := STrt+(-Q[k,Woche-i+1]+Q[k,Woche-i])/STrt0*W;
    end;
    end;
    Trt := STrt;
end;

Procedure RandAussen_gfunc (var TRT                : real;
                           k,Woche,Zeitschritt    : integer;
                           simstep               : longint;
                           RepRandbed           : integer;
                           Q                    : MatrixQ;
                           cpErd,rhoErd,lambdaErd,
                           Rechenradius,Sondenlaenge,
                           gpar1,gpar2,gpar3,gpar4,
                           gpar5,gpar6         : real;
                           DimAxi              : integer);
{ Diese Procedure berechnet die Randbedingung mit der g-Function
***** }
var u,STrt,ts,g,go,Rq      : extended;
    i,j                    : integer;
begin
    ts := sqr(Sondenlaenge) / 9 / lambdaErd * rhoErd * cpErd;
    STrt:= 0;
    for i:= 1 to Woche do begin
        u := ln( i * 604800 *RepRandbed / ts );
        if u>2.5 then u:=2.5;
        go := 0.5*u + 6.907755;
        if u<uMin then g := go
            else g := gpar1 + gpar2*u + gpar3*sqr(u) + gpar4*u*sqr(u) +
                gpar5*sqr(sqr(u)) + gpar6*u*sqr(sqr(u));
        if u<-2 then if (go-0.3)>g then g:=go;
        g := g - ln(Rechenradius/Sondenlaenge/0.0005);
        Rq := g / 2 / Pi / lambdaErd;
        STrt := STrt + (-Q[k,Woche-i+1]+Q[k,Woche-i])/Sondenlaenge*DimAxi*Rq;
    end;
    Trt := STrt;
end;

Function TBRINE(var T                : Matrix;
                var TDown,Tup        : Vektor;
                TSink,L0,L           : real;
                Zeitschritt,subdt,substep : integer;
                var QWand             : Vektor;
                mcpSole              : real;
                DimAxi               : integer;
                stationaer           : boolean ) :real;
var i,k                            : integer;
    TOut,dt2,Lm0,Lm1,LmMin         : real;
    L0mcpdt,L1mcpdt,Nichtad        : real;
    Td,Tu,SummeT                   : Vektor;
begin
    dt2 := Zeitschritt*60/subdt/substep; (* [s] *)
    TDown[0]:= TSink;
    TOut := 0;
    L0mcpdt := L0 / mcpSole * dt2;
    L1mcpdt := L / 2 / mcpsole * dt2;
    for i:=1 to DimAxi do SummeT[i] := 0;
    for k := 1 to Substep do begin
        for i := 1 to DimAxi do begin
            Td[i] := (T[i,1] - TDown[i]) * L1mcpdt;
            if stationaer then TDown[i]:=(L/2*T[i,1] + L0*TDown[i-1])/(L/2+L0)
            else TDown[i] := TDown[i] + (TDown[i-1] - TDown[i]) * L0mcpdt + Td[i];
            Td[i] := (T[i,1] - TDown[i]) * L1mcpdt;
        end;
        TUp[0] := TDown[DimAxi];
        for i := 1 to DimAxi do begin
            Tu[i] := (T[1+DimAxi-i,1] - TUp[i]) * L1mcpdt;
            if stationaer then TUp[i]:=(L/2*T[1+DimAxi-i,1] + L0*TUp[i-1])/(L/2+L0)
            else TUp[i] := TUp[i] + (TUp[i-1] - TUp[i]) * L0mcpdt + Tu[i];
            Tu[i] := (T[1+DimAxi-i,1] - TUp[i]) * L1mcpdt;
        end;
        for i:= 1 to DimAxi do SummeT[i] := SummeT[i] + Td[i] + Tu[1+DimAxi-i];
        TOut := TOut + TUp[DimAxi];
    end;
    for i:=1 to DimAxi do QWand[i] := SummeT[i] * mcpSole;
    TOut := TOut / Substep;
    TBRINE := TOut;
end;

```

```

end;      (* Function TBRINE *)

Procedure calculateEWS (var iteration : integer);
var i,j,idt : integer;
begin
  if Iteration=0 then begin
    for i:=1 to DimAxi do begin
      for j:=1 to DimRad do TEarthOld[i,j] := TEarth[i,j];
      TupOld[i] := Tup[i];
      TDownOld[i] := TDown[i];
      Q0Old[i] := Q0[i];
    end
  end else begin
    for i:=1 to DimAxi do begin
      for j:=1 to DimRad do TEarth[i,j] := TEarthOld[i,j];
      Tup[i] := TupOld[i];
      TDown[i] := TDownOld[i];
      Q0[i] := Q0Old[i];
    end;
  end;

  if iteration=0 then begin
    if not readfile then filestep:=simstep-1;
    if simstep>1 then if ((not einschwingen) or Allsteps) then begin
      if TSource < TSourceMin then TSourceMin := TSource;
      if TSource > TSourceMax then TSourceMax := TSource;
      if TSink < TSinkMin then TSinkMin := TSink;
      if TSink > TSinkMax then TSinkMax := TSink;
      if writefile then
        if ((simstep-1) mod StepWrite)=0 then begin
          if Leistung then if Massenstrom>0.0001 then
            TSink := TSource-QSource*1000/Massenstrom/cpSole
          else TSink := TSource
          else QSource:=(TSource-TSink)*Massenstrom*cpSole/1000;
          WriteStep(filestep,Massenstrom,TSink,QSource,TSource,
            TEarth[MonitorAxi,MonitorRad],p,laminar);
        end;
      end;
      if readfile then ReadStep(filestep,Massenstrom,TSink,QSource);
      if DeltaT<>0 then TSink := TSourceOld - DeltaT;
    end;

    if iteration >-1 then begin
      Summe_TSource := 0;
      if not Starttemp then begin
        Woche := (simstep+numrows*(Jahr-1)-1) div (10080 div Zeitschritt * RepRandbed) + 1;
        simstepn:= (simstep+numrows*(Jahr-1)) mod (10080 div Zeitschritt * RepRandbed);
      end else begin
        Woche := (simstep-1) div (10080 div Zeitschritt * RepRandbed) + 1;
        simstepn:= (simstep) mod (10080 div Zeitschritt * RepRandbed);
      end;
      if simstepn = 0 then simstepn := 10080 div RepRandbed div Zeitschritt;
      Rechengebiet := Rechenradius - Bohrdurchmesser / 2;
      L0 := cpSole * Massenstrom / AnzahlSonden;
      if Massenstrom>0.00001 then Pumpelauft := true else Pumpelauft := false;
      for idt := 1 to subdt do begin
        (* calculate brine Temperature *)
        if Pumpelauft then begin
          TSource := TBRINE(TEarth,TDown,TUp,TSink,L0,Llrun,Zeitschritt,subdt,
            substep_run,QWand,mcpSole,DimAxi,stationaer);
          for i:= 1 to DimAxi do TEarth[i,0]:=TEarth[i,1]
            - QWand[i]/Llrun/Zeitschritt/60*Subdt;
        end else begin
          TSource := TBRINE(TEarth,TDown,TUp,0,0,Llstop,Zeitschritt,subdt,
            substep_stop,QWand,mcpSole,Dimaxi,stationaer);
          for i:= 1 to DimAxi do TEarth[i,0]:=TEarth[i,1]
            - QWand[i]/Llstop/Zeitschritt/60*Subdt;
        end;
        for i:=1 to DimAxi do begin
          (* calculate temperature at outer boudary condition at each RepRandBed *)
          Q0[i] := (Q0[i] * ((simstepn-1)*subdt + (idt-1))
            + QWand[i]*subdt/Zeitschritt/60)/((simstepn-1)*subdt+idt);
          if iteration=0 then begin
            if idt=subdt then begin
              if ((simstep+numrows*(Jahr-1))*Zeitschritt) mod (60*24*7*RepRandbed) = 0 then begin
                Q[i,Woche] := Q0[i];
                if gfunction then

```

```

        RandAussen_gfunc(TRT,i,Woche,Zeitschritt,(simstep+numrows*(Jahr-1)),
            RepRandbed,Q,cpErde[i],rhoErde[i],lambdaErde[i],Rechenradius,
            Sondenlaenge,gpar1,gpar2,gpar3,gpar4,gpar5,gpar6,DimAxi)
    else
        RandAussen(TRT,i,Woche,Zeitschritt,(simstep+numrows*(Jahr-1)),RepRandbed,
            Q,cpErde[i],rhoErde[i],lambdaErde[i],Rechenradius,Sondenlaenge,DimAxi);
        TEarth[i,DimRad+1] := T0[i] + TRT;
    end;
end;
end;
end;
(* calculate earth temperature *)
    if Pumpelauf then multiplizieren(B1,TEarth,Tneu,i,DimRad)
    else multiplizieren(B2,TEarth,Tneu,i,DimRad);
    for j:=1 to DimRad do TEarth[i,j] := Tneu[i,j];
    end;
    Summe_TSource := Summe_TSource + TSource;
end;
TSource := Summe_TSource / subdt;      (* avarage source temperature *)
TSourceOld := TSource;
TMonitor := TEarth[MonitorAxi,MonitorRad];
end;
end; (* Procedure calculateEWS *)

Procedure Starttemperatur;
var i,j : integer;
begin
    if LastYear > 3 then begin
        if Jahr=2 then begin
            if not Leistung then begin
                (* Startbedingung fuer 2/3 der Simulationsdauer *)
                for i:=1 to DimAxi do begin
                    QEntzug[i] := 0;
                    for j:=1 to 52*RepRandbed do QEntzug[i] := QEntzug[i] + Q[i,j]/52/RepRandbed;
                end;
                Jahr := LastYear div 3 * 2;
                InitEWS(Jahr-1,QEntzug,rhoFill,cpFill,lambdaFill,rhoSole,cpSole,lambdaSole,NueSole,
                    Sondenlaenge,Sondendurchmesser,Dicke_Sondenrohr,Bohrdurchmesser,
                    Rl,Ra,Rb,Auslegungsmassenstrom,Bodentemp,Tgrad,Rechenradius,
                    Gitterfaktor,adiabat,Zeitschritt,subdt,substep_run,
                    substep_stop,Jahr,B1,B2,TEarth,T0,TUp,TUpold,TDown,
                    TDownold,Llast,cpErde,rhoErde,lambdaErde,Q,mcpSole,
                    Llrn,Llstop,dl,TSource,DimAxi,DimRad);
                for i:=1 to DimAxi do begin
                    for j:=1 to DimRad do TEarthOld[i,j] := TEarth[i,j];
                    Tup[i] := TupOld[i];
                    TDown[i] := TDownOld[i];
                end;
                Jahr := LastYear - 1;
            end else Jahr := LastYear;
        end; (* Ende Berechnung 2/3 der Simulationsdauer *)
        (* Startbedingung fuer letztes Jahr *)
        if Jahr = Lastyear then begin
            for i:=1 to DimAxi do begin
                QEntzug[i] := 0;
                for j:=1 to 52*RepRandbed do QEntzug[i] := QEntzug[i] + Q[i,j]/52/RepRandbed;
            end;
            InitEWS(Jahr-1,QEntzug,rhoFill,cpFill,lambdaFill,rhoSole,cpSole,lambdaSole,NueSole,
                Sondenlaenge,Sondendurchmesser,Dicke_Sondenrohr,Bohrdurchmesser,
                Rl,Ra,Rb,Auslegungsmassenstrom,Bodentemp,Tgrad,Rechenradius,
                Gitterfaktor,adiabat,Zeitschritt,subdt,substep_run,
                substep_stop,Jahr,B1,B2,TEarth,T0,TUp,TUpold,TDown,
                TDownold,Llast,cpErde,rhoErde,lambdaErde,Q,mcpSole,
                Llrn,Llstop,dl,TSource,DimAxi,DimRad);
            for i:=1 to DimAxi do begin
                for j:=1 to DimRad do TEarthOld[i,j] := TEarth[i,j];
                Tup[i] := TupOld[i];
                TDown[i] := TDownOld[i];
            end;
        end; (* Ende letztes Jahr *)
    end;
end; (* Procedure Starttemperatur *)

begin (* Procedure EWSmodul *)

    if iteration=-6 then begin
        InputFileErzeugen(Inputfile,Laufzeit,Massenstrom,QSource,
            TSink,QSpitzeFeb,Sondenlaufzeit,DauerLastSpitze);
    end;
end;

```

```

if iteration=-4 then begin
  if not readfile then filestep:=numrows;
  if writefile then if ((not einschwingen) or Allsteps) then begin
    if Leistung then if Massenstrom>0.0001 then
      TSink := TSource-QSource*1000/Massenstrom/cpSole
      else TSink := TSource
      else QSource:= (TSource-TSink)*Massenstrom*cpSole/1000;
    WriteStep(filestep,Massenstrom,TSink,QSource,TSource,
      TEarth[MonitorAxi,MonitorRad],p,laminar);
  end;
  if readfile then close(ifile);
  if writefile then close(ofile);
end;

if iteration=-3 then
  DefaultWert(cpSole,cpFill,rhoSole,rhoFill,lambdaSole,lambdaFill,nueSole,
    Rechenradius,Gitterfaktor,Sondenlaenge,Sondendurchmesser,
    Dicke_Sondenrohr,Bohrdurchmesser,Massenstrom,TGrad,
    Jahresmitteltemp,Bodenerwaermung,DeltaT,Auslegungsmassenstrom,
    TSource,Sicherheit1,Sicherheit2,Rl,Ra,Rb,Genauigkeit,adiabat,
    g1,g2,g3,g4,g5,Sondenabstand,g_Sondenabstand,TSink,QSource,
    cpErde,rhoErde,lambdaErde,RepRandbed,Zeitschritt,
    Simulationsdauer,DimAxi,DimRad,MonitorAxi,MonitorRad,Jahr,
    StepWrite,LastYear,AnzahlSonden,DauerLastspitze,ErrorMldg,numrows,
    readfile,writefile,Einschwingen,Leistungsinput,Leistung,
    stationaer,gfunction,Starttemp,Druck,AlteInputfiles,SwewsInput,
    Allsteps,IFileName,OFileName,swewsName,Sprache);

if iteration=-2 then begin
  if readfile then OpenInputFile (IFileName,Zeitschritt,numrows);
end;

if iteration=-5 then begin
  if SwewsInput then OpenSwewsFile; (* Einlesen Stoffdaten von SWEWS.DAT *)
end;

if iteration=-1 then begin
  InitEWS(0,VektorNull,rhoFill,cpFill,lambdaFill,rhoSole,cpSole,lambdaSole,
    NueSole,Sondenlaenge,Sondendurchmesser,Dicke_Sondenrohr,Bohrdurchmesser,
    Rl,Ra,Rb,Auslegungsmassenstrom,Bodentemp,Tgrad,Rechenradius,Gitterfaktor,
    adiabat,Zeitschritt,subdt,substep_run,substep_stop,Jahr,B1,B2,TEarth,
    T0,TUp,TUpold,TDown,TDownold,Llast,cpErde,rhoErde,lambdaErde,Q,mcpSole,
    Llrn,Llstop,dl,TSource,DimAxi,DimRad);
  for i:=1 to DimAxi do begin
    for j:=1 to DimRad do TEarthOld[i,j] := TEarth[i,j];
    Tup[i] := TupOld[i];
    TDown[i] := TDownOld[i];
  end;
  if writefile then OpenOutputFile(OFileName,Zeitschritt,numrows);
end;

if iteration > -1 then begin
  if Leistung then begin
    iteration:=0;
    TSink := 0;
    for i:= 1 to DimAxi do TSink:=TSink + TEarth[i,1]/DimAxi;
    if Massenstrom >0.00001 then TSink := TSink -
      (cpSole*Massenstrom/AnzahlSonden+Llrn)/Llrn/cpSole/Massenstrom*QSource*1000;
  end;
  calculateEWS(iteration);
  if Leistung then begin
    if Massenstrom>0.00001 then repeat begin
      Iteration := Iteration + 1;
      calculateEWS(Iteration);
      TSinkOld := TSink;
      TSink := TSource - QSource * 1000 / cpSole / Massenstrom;
      if Abs(TSink)>100 then begin TSink:=-1; TSinkOld:=1; end;
    end until Abs(TSink-TSinkOld)<Genauigkeit;
    if Massenstrom < 0.00001 then begin
      TSink := TEarth[1,1];
      TSource := TSink;
    end;
  end else QSource:= (TSource-TSink)*Massenstrom*cpSole/1000;

  if Starttemp then if Jahr < Lastyear then Einschwingen := true
    else Einschwingen := false;

  if (simstep=numrows) then begin

```

```

simstep := 0;
Jahr := Jahr + 1;
if starttemp then Starttemperatur;
if Einschwingen then begin
  if not readfile then filestep:=numrows;
  if readfile then close(ifile);
  if writefile then if not Allsteps then close(ofile);
  if readfile then begin
    assign(ifile,IFileName);
    reset(ifile);
    if AlteInputFiles then for i:=1 to 32 do readln(ifile)
                        else for i:=1 to 201 do readln(ifile);
  end;
  if writefile then if not Allsteps then OpenOutputFile(OFileName,Zeitschritt,numrows);
  if writefile then if Allsteps then begin
    if Leistung then if Massenstrom>0.0001 then
      TSink := TSource-QSource*1000/Massenstrom/cpSole
    else TSink := TSource
    else QSource:= (TSource-TSink)*Massenstrom*cpSole/1000;
    WriteStep(filestep,Massenstrom,TSink,QSource,TSource,
      TEarth[MonitorAxi,MonitorRad],p,laminar);
  end;
end;
end;
end;
if Druck then p:=Druckabfall(NueSole,RhoSole,Massenstrom/AnzahlSonden,Sondendurchmesser,
  Dicke_Sondenrohr,laminar) else p:=0; (* [Pascal] *)
end;
(*-- Procedure EWSmodul -----*)
(*-- Procedure Erdsonde -----*)
Procedure Erdsonde( iteration : integer);
begin
  EWSmodul(iteration,ErrorMldg,DimAxi,DimRad,Zeitschritt,subdt,substep_run,substep_stop,
    RepRandbed,MonitorAxi,MonitorRad,Jahr,LastYear,Beginn,AnzahlSonden,
    DauerLastspitze,numrows,rhoFill,cpFill,lambdaFill,rhoSole,cpSole,
    lambdaSole,nueSole,R1,Ra,Rb,Sondenlaenge,Sondendurchmesser,
    Dicke_Sondenrohr,Bohrdurchmesser,Auslegungsmassenstrom,Massenstrom,
    Bodentemp,Tgrad,mcpSole,Llrun,Llstop,d1,Rechenradius,Rechengebiet,
    Gitterfaktor,Sicherheit1,Sicherheit2,DeltaT,TSourceOld,TSource,
    TSink,TMonitor,p,QSource,Genauigkeit,adiabat,g1,g2,g3,g4,g5,
    Sondenabstand,g_Sondenabstand,QSpitzeFeb,TSourceMin,TSourceMax,
    TSinkMin,TSinkMax,Sondenlaufzeit,B1,B2,TEarth,TEarthold,T0,TUp,
    TUpOld,TDown,TDownOld,Llast,cpErde,rhoErde,lambdaErde,Q,Laufzeit,
    readfile,writefile,laminar,Einschwingen,Leistungsinput,Leistung,
    stationaer,gfunction,Starttemp,Druck,AlteInputFiles,SwewsInput,
    Allsteps,IFileName,OFileName,swewsName,Sprache,Wort);
end;
(*-- Procedure Eredsonde -----*)
end. { der Unit EWSMOD }

```