

**Weiterentwicklung und  
Qualitätssicherung von  
Modellierungswerkzeugen  
zur Durchführung und  
Bewertung von  
Sicherheitsanalysen im  
Standortauswahlverfahren**

## Weiterentwicklung und Qualitätssicherung von Modellierungswerkzeugen zur Durchführung und Bewertung von Sicherheitsanalysen im Standortauswahlverfahren

Martin Navarro  
Thomas Beuth  
Guido Bracke  
Jens Eckel  
Gerd Frieling  
Stephan Hotzel  
Ingo Kock  
Holger Seher  
Torben Weyand

Februar 2021

### **Anmerkung:**

Das diesem Bericht zugrunde liegende Forschungsvorhaben wurde mit Mitteln des Bundesministeriums für Umwelt, Naturschutz und nukleare Sicherheit (BMU) unter dem Förderkennzeichen 4718E03260 durchgeführt.

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der GRS.

Der Bericht gibt die Auffassung und Meinung der GRS wieder und muss nicht mit der Meinung des BMU übereinstimmen.

## **Deskriptoren**

Gittergenerierung, Langzeitsicherheit, MARNIE2, SITA, TOUGH2, TOUGHREACT

## **Kurzfassung**

Zur Durchführung und Bewertung von Sicherheitsanalysen des Standortauswahlverfahrens hat die GRS im Vorhaben 4718E03260 „*Weiterentwicklung und Qualitätssicherung von Modellierungswerkzeugen zur Durchführung und Bewertung von Sicherheitsanalysen im Standortauswahlverfahren*“ des Bundesministeriums für Umwelt, Naturschutz und nukleare Sicherheit (BMU) ihre numerischen Rechenprogramme weiterentwickelt. Die Arbeiten fokussierten auf die Programme MARNIE2, TOUGH2-GRS, TOUGH2-MP-GRS, TOUGHREACT und SITA. Für den Code TOUGH2-MP-GRS wurde ein Generator für unstrukturierte Voronoi-Gitter entwickelt.

## **Abstract**

In order to carry out and evaluate safety analyses of the site selection process GRS further developed its numerical codes in project 4718E03260 „*Weiterentwicklung und Qualitätssicherung von Modellierungswerkzeugen zur Durchführung und Bewertung von Sicherheitsanalysen im Standortauswahlverfahren*“ of the Federal Ministry for the Environment, Nature Conservation and Nuclear Safety (BMU). The work focussed on the codes MARNIE2, TOUGH2-GRS, TOUGH2-MP-GRS, TOUGHREACT, and SITA. A grid generator for unstructured Voronoi grids was developed for the code TOUGH2-MP-GRS.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung .....</b>	<b>1</b>
<b>2</b>	<b>Erweiterung des Codes MARNIE um Zweiphasenflussprozesse (AP 1) .....</b>	<b>3</b>
2.1	Zweiphasiger Transport in MARNIE2.....	4
2.2	Verifikationstestfälle .....	6
2.3	Codekonsolidierung und Restrukturierung .....	7
2.4	Relative Permeabilitäten .....	10
2.5	Kapillardruckkurven .....	16
2.6	Erweiterungen in den Randmodulen .....	19
2.7	Phasenwechsel .....	20
2.7.1	Implementierte Einzelprozesse .....	20
2.7.2	Fehlerbeseitigung in der FEBE .....	23
2.7.3	Plausibilitätstests für Umschaltprozesse .....	24
2.8	Ausstehende Arbeiten .....	33
<b>3</b>	<b>Weiterentwicklung des Codes TOUGH2-MP-GRS (AP 2) .....</b>	<b>35</b>
3.1	Die Codes TOUGH2-GRS und TOUGH2-MP-GRS .....	35
3.2	Einführung einer Restartfunktionalität .....	36
3.3	Verbesserung der Plausibilitätskontrollen für die Eingabe.....	39
3.4	Verbesserung des Fehlermanagements .....	41
3.5	Entwicklung von Schnittstellen.....	41
3.6	Verifikationstests.....	42
3.6.1	Modul GCOMP: Änderung der Hauptgaskomponente .....	42
3.6.2	Modul RN: Löslichkeitsbegrenzung.....	46
3.6.3	Restartfunktionen.....	48
3.7	Konsolidierung.....	57
3.8	Dokumentation .....	59

<b>4</b>	<b>Gittergenerierung und Visualisierung für den Code TOUGH2-MP-GRS (AP 3)</b> .....	<b>61</b>
4.1	Voronoi-Gitter .....	61
4.2	Anforderungen an einen Gittergenerator.....	63
4.3	Recherche zu Gittergeneratoren und Programmbibliotheken.....	65
4.4	Wahl der Vergitterungsstrategie.....	66
4.5	Konzept der Gittergenerierung und -visualisierung .....	67
4.6	User Interface .....	70
4.7	Status der Implementierung.....	76
4.8	Neuvergitterung (Remeshing).....	77
4.9	Erstellung von Knotenseeds .....	79
4.9.1	Notwendigkeit der Flächenvereinfachung .....	79
4.9.2	Lokale Knotensysteme und Positionen der Knotenseeds.....	80
4.9.3	Steuerungsroutine .....	83
4.9.4	Kantenorientiertes Spiegelungsverfahren .....	85
4.9.5	Kantenvervollständigung.....	89
4.10	Erstellung von Schutzseeds.....	90
4.11	Erstellung von Körperseeds.....	91
4.12	Erstellung des Voronoi-Gitters .....	92
4.13	Setzung inaktiver Elemente und Export der TOUGH2-Gitterdatei.....	93
4.14	Datenbereitstellung.....	94
4.14.1	Zielsetzung .....	94
4.14.2	Datengrundlage .....	95
4.14.3	Erste Visualisierung .....	97
4.14.4	Selektion einer Beispielschicht.....	98
4.14.5	Datenkonvertierung .....	99
4.14.6	Generierung eines gemeinsamen Oberflächennetzes .....	100
4.14.7	Bereinigung des Oberflächennetzes .....	101
4.15	Entwicklung von Visualisierungswerkzeugen zur Ergebnisanalyse .....	102
4.15.1	Methoden zur Erstellung von Darstellungsvorlagen in ParaView .....	103

4.15.2	Visualisierungsvorlagen .....	104
4.15.3	Weiterführende Arbeiten .....	109
4.16	Fazit.....	110
<b>5</b>	<b>Verwendung von temperaturabhängigen thermodynamischen Daten in TOUGHREACT (AP 4) .....</b>	<b>111</b>
5.1	TOUGH-Codes und THEREDA .....	111
5.2	Benchmark-Tests und Codes.....	113
5.2.1	PHREEQC.....	114
5.2.2	TOUGHREACT .....	116
5.3	Benchmark-01 (System ozeanischer Salze) .....	123
5.3.1	Vorgaben .....	123
5.3.2	Temperaturabhängigkeit.....	125
5.3.3	Ergebnisse.....	131
5.4	Benchmark-02 (System mit Aktiniden).....	136
5.4.1	Vorgaben .....	136
5.4.2	Ergebnisse.....	138
5.5	Benchmark-03 (System mit CO <sub>2</sub> (g)).....	144
5.5.1	Vorgaben .....	144
5.5.2	Ergebnisse.....	145
5.6	Zusammenfassung und Ausblick .....	147
<b>A</b>	<b>Anhang .....</b>	<b>149</b>
A.1	Beschreibung der MOPR-Jumper für TR-BRINE /SPY 19/ .....	149
	<b>Literaturverzeichnis.....</b>	<b>151</b>
	<b>Abbildungsverzeichnis.....</b>	<b>161</b>
	<b>Tabellenverzeichnis.....</b>	<b>167</b>
	<b>Danksagung .....</b>	<b>169</b>



# 1 Einleitung

Ziel des BMU-geförderten Vorhabens „*Weiterentwicklung und Qualitätssicherung von Modellierungswerkzeugen zur Durchführung und Bewertung von Sicherheitsanalysen im Standortauswahlverfahren*“ (Förderkennzeichen 4718E03260), dessen Ergebnisse in diesem Bericht vorgestellt werden, war die Weiterentwicklung der bei der GRS verwendeten Codes zur Durchführung oder Bewertung vorläufiger Sicherheitsuntersuchungen im Rahmen des Standortauswahlverfahrens /STA 20/. Die Arbeiten konzentrierten sich auf die Strömungs- und Transportcodes TOUGH2-GRS, TOUGH2-MP-GRS und MARNIE2, den hydraulisch-chemisch-gekoppelten Code TOUGHREACT und den Code SITA /SEH 16/, der ein Code zur Durchführung automatisierter Verifikationstests für MARNIE2 und die TOUGH2-Codefamilie ist.

Der Code TOUGH2-MP-GRS eröffnet durch seine Fähigkeit, auf mehreren Prozessoren zu laufen, die Möglichkeit zur Simulation komplexer Modellgebiete mit hoher Gitterelementanzahl, wie sie z. B. bei Fernfeldsimulationen für Endlager im Wirtsgestein Tonstein auftreten. Um dies zu ermöglichen, wurde im Vorhaben ein Gittergenerator für unstrukturierte Voronoi-Gitter und zugehörige Pre- und Postprocessing-Werkzeuge für die TOUGH2-Codefamilie entwickelt.

Die Wartbarkeit von Simulationscodes ist angesichts der zeitlichen Länge des Standortauswahlverfahrens von großer Bedeutung. Die Anforderung der Wartbarkeit erfordert, dass der Quelltext der Codes übersichtlich, verständlich und gut dokumentiert ist, damit spätere Korrekturen und Weiterentwicklungen möglich bleiben und der Code langfristig genutzt werden kann. Das schafft die Voraussetzungen dafür, dass Simulationsrechnungen auch noch nach vielen Jahren kontrolliert und nachvollzogen werden können. Die Verbesserung der Wartbarkeit wurde im Vorhaben für die Codes TOUGH2-GRS, TOUGH2-MP-GRS, MARNIE2 und SITA angestrebt und umfasste teilweise umfangreiche Codekonsolidierungen, insbesondere für den Code MARNIE2, der eine langjährige Entwicklungsgeschichte besitzt.

Das Projekt untergliederte sich in fünf fachliche Arbeitspakete.

- AP 1: Weiterentwicklung des Zweiphasenflusses im Code MARNIE
- AP 2: Weiterentwicklung und Qualitätssicherung der Codes TOUGH2-GRS und TOUGH2-MP-GRS
- AP 3: Weiterentwicklung des Pre- und Postprocessings für TOUGH2-MP-GRS
- AP 4: Weiterentwicklung des Codes TOUGHREACT zur Anbindung der THEREDA-Datenbasis für hochsalinare Lösungen
- AP 5: Neukonzeption des Codes SITA

Den Arbeitspaketen 1 bis 4 ist im Folgenden jeweils ein Kapitel gewidmet. Die Neukonzeption des Codes SITA im Arbeitspaket 5 diente der Verbesserung der Erweiterbarkeit und Wartbarkeit des Codes. Diese Arbeiten wurden in einem GRS-internen Bericht /NAV 19/

M. Navarro: *Neukonzeption des Codes SITA, interner Bericht*. GRS, unveröffentlichter Bericht, Dezember 2019.

dokumentiert, da es sich um Informationen handelt, die ausschließlich für SITA-Entwickler mit entsprechenden Codekenntnissen von Interesse sein dürften.

## 2 Erweiterung des Codes MARNIE um Zweiphasenflussprozesse (AP 1)

*Jens Eckel*

Das Rechenprogramm MARNIE /MAR 02/ wurde ursprünglich zur Simulation komplexer Strömungs- und Transportprozesse im Nahfeld eines lösungsgefüllten Endlagers in einer Salzformation entwickelt, kann jedoch mit Einschränkungen auch für andere Wirtsgesteine genutzt werden. MARNIE simuliert wesentliche Prozesse, die im Endlager ablaufen, wie z. B.

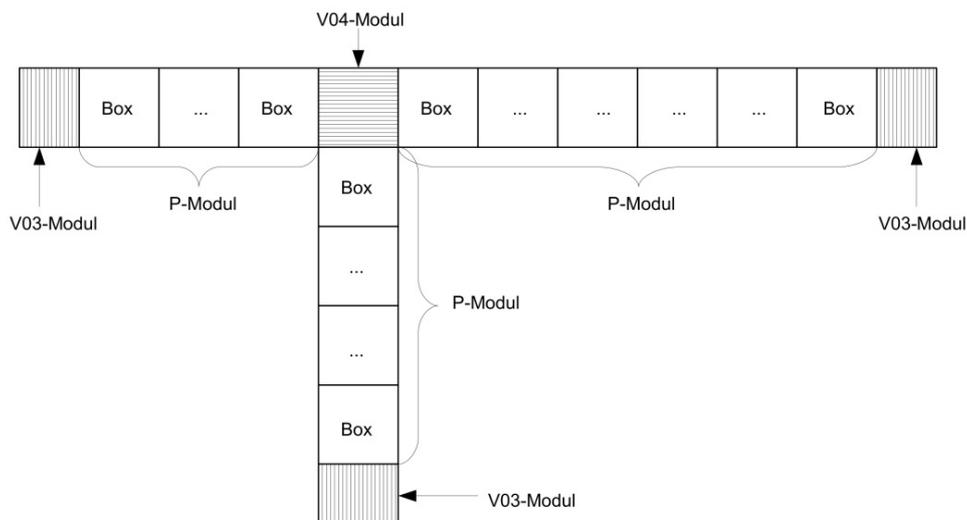
- den advektiven, dispersiven und diffusiven Transport der Lösung und der Radionuklide,
- den Einfluss der erzeugten Gase auf den Nuklidtransport,
- die Konvergenz der Hohlraumvolumina,
- den radioaktiven Zerfall der Radionuklide,
- das Versagen der Behälter,
- die Freisetzung der Nuklide aus den Behältern,
- die Löslichkeitsbegrenzung der Nuklide in der Lösung,
- die Ausfällung von Nukliden aus der Lösung,
- Sorption und Desorption von Nukliden am Feststoff,
- Temperatureinflüsse auf den Nuklidtransport sowie
- Sorption durch nichtlineare Sorptionsisothermen.

Für eine erste Implementierung des Zweiphasenflusses in MARNIE wurde der Quellcode von MARNIE im BMU-Vorhaben 4715E03230 restrukturiert und zunächst um elementare Zweiphasenflussprozesse erweitert /NAV 18a/. Die entwickelte Codeversion erlaubte dann die Simulation von Gas- und Flüssigkeitsströmungen in allen elementaren Gitterstrukturen (V- und P-Module) mit allen Kombinationen von den im ursprünglichen MARNIE zur Verfügung stehenden Randbedingungen. Um über den elementaren zweiphasigen Transport hinaus eine realitätsnahe Modellierung von Zweiphasenflussprozessen mit MARNIE durchzuführen, ist eine Erweiterung um weitere relevante physikalische

Einzelprozesse notwendig geworden. Diese fand im Vorhaben 4718E03260 statt und die wesentlichen Ergebnisse werden im Folgenden vorgestellt.

## 2.1 Zweiphasiger Transport in MARNIE2

Bevor über die im Vorhaben erzielten Ergebnisse berichtet wird, sollen zur besseren Einordnung die dem zweiphasigen Transport zugrundeliegenden Gleichungen und Annahmen gezeigt werden, weitere Details finden sich in /BEA 18/, /NAV 18a/.



**Abb. 2.1** Elemente des Netzwerkes in MARNIE2

MARNIE2 ist als eindimensionales, gekoppeltes, instationäres Netzwerkprogramm konzipiert. Zur Modellierung der Transportprozesse in der eindimensionalen Netzwerkstruktur wird das Endlager in eine Anzahl von Transport- und Verknüpfungsmodulen zerlegt, wobei Transportmodule (P-Module) der Simulation eindimensionaler Transportvorgänge dienen, während Verknüpfungsmodule (V-Module) zur Vorgabe zeitabhängiger Randwerte (V03-Module) für die angrenzenden Transportmodule und zur Modellierung der Kopplung und der Vermischungsvorgänge (V04-Module) an den Knotenpunkten angrenzender P-Module verwendet werden. Die P-Module werden in eine Anzahl von Kontrollvolumina, im Folgenden Boxen genannt, diskretisiert. Diese Boxen sind von 1 bis M indiziert, wobei die Boxen 1 und M jeweils an ein V-Modul grenzen. Zu beachten ist, dass bei geeigneten P-Modulen Box 1 unten ist. Zur Verdeutlichung wird die Anordnung dieser Netzwerkelemente in Abb. 2.1 dargestellt.

Zur numerischen Lösung der Transportgleichungen für skalare Größen wird in MARNIE2 das Finite-Volumen-Verfahren verwendet, d. h. die skalaren Größen werden den Boxmittelpunkten zugeordnet, während die Vektorgrößen auf den Rändern eines finiten Volumens mit Hilfe des Finite-Differenzen-Verfahrens gelöst werden. Das daraus resultierende Differentialgleichungssystem wird mit Hilfe des ODE-Solvers (ODE: **O**rdinary **D**ifferential **E**quations) FEBE (**F**orward **E**uler, **B**ackward **E**uler) gelöst. Der Löser kann in Abhängigkeit von der Problemstellung eine Differentialgleichung implizit und explizit lösen, eine genaue Beschreibung findet sich in /BAR 89/ und den darin enthaltenen Referenzen. Aufgrund der modularen Struktur der FEBE können ohne großen Aufwand zusätzliche Differentialgleichungen für weitere physikalische Prozesse in MARNIE2 umgesetzt werden.

Für jede Box eines P-Moduls wird die Flüssigkeitssättigung  $S_l$  vorgegeben, aus der sich die Gassättigung  $S_g = 1 - S_l$  in dieser Box entsprechend ergibt. Aus den Sättigungen ergeben sich die zur Verfügung stehenden Volumina für beide Phasen  $i = l, g$  zu

$$V_i = S_i V \phi \quad (2.1)$$

mit der Porosität  $\phi$  und dem Gesamtvolumen der Box  $V$ . Die initialen Massen  $m_l$  und  $m_g$  berechnen sich aus den Volumina über die Dichten  $\rho_l$  und  $\rho_g$  der Phasen. Der Massenfluss über die Ränder der Box eines P-Moduls leitet sich aus der eindimensionalen Impulserhaltungsgleichung für die Phase  $i$

$$\frac{\partial}{\partial t}(\rho_i A_{\phi,i} v_i) + \frac{\partial}{\partial x}(v_i^2 \rho_i A_{\phi,i}) = \sum F \quad (2.2)$$

her. Hierbei ist  $A_{\phi,i} = \phi A S_i$  die Kontaktfläche der jeweiligen Phase im Porenraum zwischen zwei Boxen,  $v_i$  die Geschwindigkeit der Phase im Porenraum und  $F$  die Kraft pro Längeneinheit. Mit dem Massenstrom  $G_i = \rho_i v_i A_{\phi,i}$  hat die in MARNIE2 implementierte Darcygleichung für Transport in porösen Medien dann die Form:

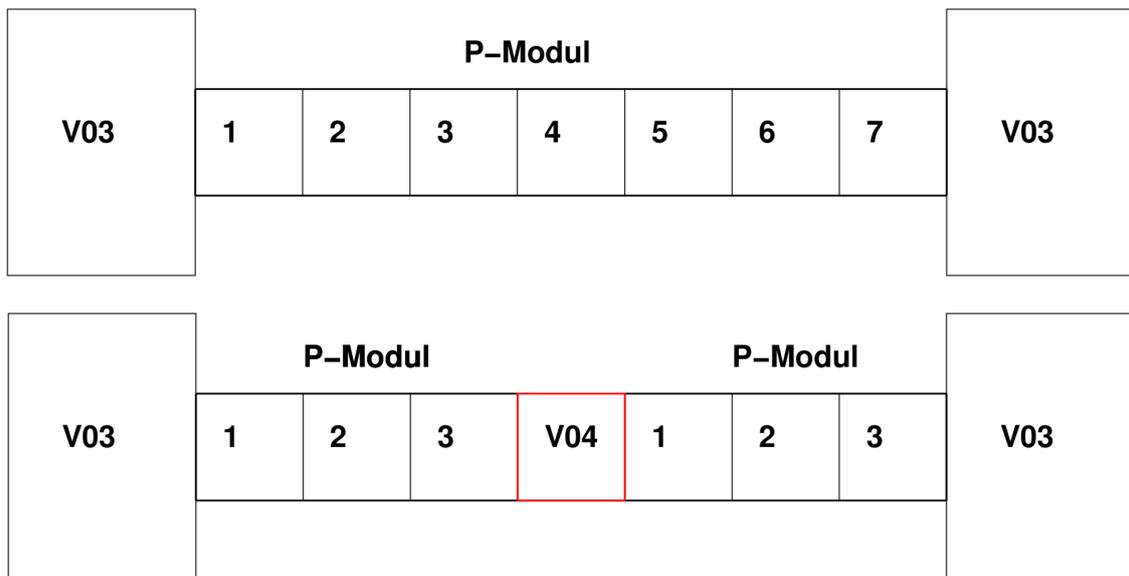
$$\dot{G}_i = -\frac{\partial}{\partial x}(v_i G_i) - \frac{\partial}{\partial x}(p_i A_{\phi,i}) + \rho_i g_x A_{\phi,i} - \frac{G_i \mu_i \phi}{\rho_i \chi_{rel,i} \chi}. \quad (2.3)$$

Hier ist  $p_i$  entweder der Druck in der Gasphase /NAV 18a/

$$\dot{p}_g = \frac{R}{V_g} \left[ T \left( \frac{1}{M_g} \int \dot{m}_g dx + \frac{\dot{m}_{gen}}{M_{gen}} \right) + N_g \dot{T} - \frac{T N_g}{V_g} \left( -k V - \frac{1}{\rho_l} \int \dot{m}_l dx \right) \right], \quad (2.4)$$

bzw. der Druck in der flüssigen Phase  $p_l = p_g - p_{cap}$  mit dem Kapillardruck  $p_{cap}$ .  $g_x = f g$  ist die Komponente der Erdbeschleunigung in Strömungsrichtung, wobei beliebige Neigungen des P-Moduls durch Setzen von  $f$  zwischen 0 und 1 berücksichtigt werden. Für die Gasphase wird die Erdbeschleunigung vernachlässigt. Die relevanten Gleichungen für Massenbilanzen sind in /NAV 18a/ zu finden. Der letzte Term in Gl. (2.3) berücksichtigt die absolute Permeabilität  $\chi$  des porösen Mediums zusammen mit der relativen Permeabilität  $\chi_{rel,i}$  der Phase  $i$ .

## 2.2 Verifikationstestfälle



**Abb. 2.2** Räumliche Diskretisierung für die Verifikationstestfälle zur Entwicklung von MARNIE2.

Um die Implementierung der für zweiphasigen Transport notwendigen Prozesse in MARNIE2 verifizieren zu können, wurde ein Satz an Verifikationstestfällen entwickelt. Der zugrunde gelegte Datensatz baut jeweils immer auf zwei räumlichen Diskretisierungen auf. Zum einen wird ein P-Modul mit sieben Boxen gleichen Volumens berandet von zwei V03-Modulen betrachtet, Abb. 2.2 oben. Um dann den Transport durch ein V04-

Modul direkt vergleichen zu können, wird zum anderen das P-Modul in zwei P-Module aufgeteilt, die mittig durch ein V04-Modul verbunden sind, welches das gleiche Volumen wie die mittlere Box des P-Moduls der ersten Diskretisierung hat, siehe Abb. 2.2 unten. Die Testfälle sind in der Tab. 2.2 zusammengefasst, die Ergebnisse sind in den Unterkapiteln zu den jeweiligen Themen zu finden. Der Satz von Testfällen deckt die meisten elementaren Prozesse in MARNIE2 ab und kann bei jeder Änderung am Quellcode durchgeführt werden, um nicht vorgesehene Veränderungen an den Ergebnissen direkt zu erkennen und so die Qualitätssicherung des Rechenprogramms zu erhöhen. Der Satz an Testfällen sollte für weitere zu implementierende Einzelprozesse schrittweise erweitert werden.

### **2.3 Codekonsolidierung und Restrukturierung**

Die im Vorhaben 4715E03230 entwickelte MARNIE Version /NAV 18a/, welche elementare zweiphasige Transportprozesse beinhaltet, wurde als Basis für die Restrukturierung des Quellcodes herangezogen. Das Ergebnis der Restrukturierung ist dann der Code MARNIE2, alle weiteren Entwicklungen finden ausschließlich in diesem Quellcode statt und sind auf dem Repository als eigener Pfad abgelegt.

Der Versuch, MARNIE2 komplett manuell zu restrukturieren, erwies sich als zeitaufwendig und fehleranfällig, weshalb zur weiteren Arbeit eine Lizenz für das Computerprogramm plusFORT V 7.21 /POL 20/ von Polyhedron durch die GRS erworben wurde. plusFORT ist ein Programmpaket zur Analyse und Verbesserung von Fortran-Programmen und kombiniert Restrukturierung und Neuformatierung mit globaler statischer und dynamischer Analyse. Zur Restrukturierung wurde das Unterprogramm SPAG<sup>1</sup> genutzt, welches Fortran 77-Code entschlüsseln und in modern strukturiertes Fortran umwandeln kann. SPAG verändert dadurch die Art und Weise, wie die Programmlogik aufgeschrieben ist, sodass diese einfacher les- und erweiterbar und dadurch besser wartbar wird. Diese Maßnahmen kommen direkt heutigen Anforderungen an die Qualitätssicherung von Rechenprogrammen zu Gute und sind auch Bestandteil des QA-Handbuchs zu MARNIE2 /ECK 18/.

---

<sup>1</sup> Veralteter, heute schwer lesbarer Quellcode wird gerne als „Spaghetticode“ bezeichnet.

In einem ersten Schritt wurden mit Hilfe von SPAG sämtliche Variablen in MARNIE2 explizit deklariert, indem automatisiert Deklarationen für implizit definierte Variablen hinzugefügt wurden. Die explizite Deklaration von Variablen erlaubt es, während der Übersetzung des Quellcodes Fehler zu erkennen, die ansonsten im Programmablauf nur mit hohem Aufwand zu identifizieren sind. Diese Umstrukturierung wurde auch an den zahlreichen COMMON-Blöcken vorgenommen und ist als Beispiel im Vergleich für den vergleichsweise kompakten Block CAPAR1 in der Abb. 2.3 gezeigt.

```

*      LIST OFF
*      LIST OFF
CH NAME | DIM. | S | SIGNIFICATION
-----|-----|---|-----
!V IAPAR1 |      |   | LENGTH OF COMMON BLOCK
!V JAPAR1 |      |   |
!V CPUTOT |      |   | TOTAL CPU-TIME OF RUN
!V TF     |      |   | TIME OF FEBE MODULE
!V TM     |      |   | MASTER TIME
!V LFEND  |      |   | INDICATES IF FEBE-STEP IS NORMALLY COMPLETED
*      LIST ON
COMMON /CAPAR1/ IAPAR1, JAPAR1
1, CPUTOT, TF , TM , LFEND
LOGICAL LFEND

```

```

!      LIST OFF
!      LIST OFF
!H NAME | DIM. | S | SIGNIFICATION
-----|-----|---|-----
!V IAPAR1 |      |   | LENGTH OF COMMON BLOCK
!V JAPAR1 |      |   |
!V CPUTOT |      |   | TOTAL CPU-TIME OF RUN
!V TF     |      |   | TIME OF FEBE MODULE
!V TM     |      |   | MASTER TIME
!V LFEND  |      |   | INDICATES IF FEBE-STEP IS NORMALLY COMPLETED
!      LIST ON
!
!*** Start of declarations rewritten by SPAG
!
! COMMON variables
!
! REAL(R8KIND) :: CPUTOT , TF , TM
! INTEGER      :: IAPAR1 , JAPAR1
! LOGICAL      :: LFEND
! COMMON /CAPAR1/ IAPAR1 , JAPAR1 , CPUTOT , TF , TM , LFEND
!
!*** End of declarations rewritten by SPAG
!

```

**Abb. 2.3** Restrukturierung der COMMON Blöcke am Beispiel von CAPAR1. Original (oben) und Ergebnis der Bearbeitung durch SPAG (unten)

In einem weiteren, sehr viel umfangreicheren Schritt wurden die einzelnen Routinen des Quellcodes mit SPAG restrukturiert und relabeled. Hierbei wurden z.B. folgende Aspekte berücksichtigt:

- Berechnete GO-TO-Anweisungen werden in SELECT-CASE-Anweisungen umgewandelt.
- Zum Verlassen von Schleifen werden EXIT- und CYCLE-Anweisungen genutzt.
- Kurze Codeteile können automatisiert wiederholt werden, um die Lesbarkeit des Codes zu erhöhen.
- Die Ausgabe von SPAG erzeugt Code, der einheitlich gesetzt ist hinsichtlich Einrückung, Leerzeichen etc., um die Lesbarkeit des Codes zu erhöhen.

Nach der Anwendung von SPAG auf die einzelnen Routinen wurde das erzeugte Ergebnis im Detail mit der ursprünglichen Version verglichen. Hierbei wurden überflüssige Coderelevante Teile aus MARNIE, die in MARNIE2 nicht mehr benötigt werden, entfernt. Dies umfasst beispielsweise alle Teile, die die Einlaufphase bzw. die Übergangsphase zu flüssigkeitsgefüllten P-Modulen in MARNIE behandeln. Diese beiden Zustände werden im zweiphasigen MARNIE2 nicht mehr benötigt und wurden aus dem Quellcode entfernt. Es wurden auch zahlreiche Routinen entfernt, die zwar bei der Erstellung des ausführbaren Programms mit übersetzt werden, aber nicht mehr aufgerufen werden.

Als Ausnahme ist der Differentialgleichungslöser in MARNIE2, FEBE, eine Eigenentwicklung der GRS, nicht durch SPAG modernisiert worden. Die FEBE stammt aus den 80er-Jahren und ist seitdem in ihrer grundlegenden Struktur nicht modernisiert worden; eine Beschreibung findet sich in /BAR 89/. Gleichwohl haben Vergleichsarbeiten /GRA 92/, /HOF 93/, /LUT 96/ gezeigt, dass die Lösung des zugrunde liegenden Differentialgleichungssystems mit guter Effizienz erfolgt. Da für die FEBE keine Weiterentwicklung geplant ist, wurde von einer Restrukturierung abgesehen, vielmehr sollte die FEBE bei einer Übersetzung des Quellcodes in einer nicht debug-fähigen Version als Bibliothek eingebunden werden. Der Quellcode der FEBE ist in MARNIE2 in der Ordnerstruktur vom Rest des Quellcodes getrennt, um diese Unterscheidung klar zu machen.

In der weiteren Arbeit der Restrukturierung und Konsolidierung des Quellcodes von MARNIE2 sind dann noch die Modularisierung des Quellcodes, sowie die dynamische Allokation sämtlicher Felder umzusetzen. Beide Punkte können sukzessive mit SPAG durchgeführt werden und bedürfen einer genauen und schrittweisen Überprüfung der Ergebnisse auf ihre Korrektheit hin. Da die bisher durchgeführten Konsolidierungsmaßnahmen zeitaufwändiger als ursprünglich geplant waren, sind die vorgenannten Arbeiten im Vorhaben nicht mehr umgesetzt worden.

Während der Implementierung zweiphasiger Prozesse wurden auch zusätzlich Eingabeparameter in die MARNIE2-Eingabe eingebaut; nicht mehr benötigte Eingabeparameter wurden entfernt. Die ursprünglich geplante vollständige Bereinigung und Restrukturierung der Eingabeschnittstelle und die Umstellung vom kartenbasierten Format auf ein moderneres Format (z. B. HDF5 oder JSON) wurde in der Priorität zurückgestellt. Es ist sinnvoll, zunächst ein verifiziertes Programm zu entwickeln, welches zweiphasigen Transport und die beteiligten Einzelprozesse abbilden kann. Die Bearbeitung der Eingabeschnittstelle kann dann abschließend erfolgen.

## 2.4 Relative Permeabilitäten

Um zweiphasigen Transport in einem porösen Medium adäquat beschreiben zu können, wird üblicherweise in der Darcygleichung (2.3) eine relative Permeabilität  $\chi_{rel,i}$  für die jeweilige Phase  $i$  (gasförmig oder flüssig) eingeführt. Hierzu wird eine residuale Phasensättigung  $S_{res,i}$  definiert. Sinkt die jeweilige Phasensättigung unter diese residuale Phasensättigung, wird die Phase immobil und es findet jeweils nur noch einphasiger Transport statt. Für das zugrundeliegende Konzept wird angenommen, dass die Permeabilität eine Funktion der Sättigung ist. Damit lässt sich die relative Permeabilität für die Phase  $i$  als dimensionslose Größe

$$\chi_{rel,i} = \frac{\chi_i(S_i)}{\chi_{sat}} \quad (2.5)$$

mit der Permeabilität bei voller Phasensättigung  $\chi_{sat}$  und dem Wertebereich  $0 \leq \chi_{rel,i} \leq 1$  formulieren /BEA 18/.

Dem Konzept liegt letztlich die Annahme zugrunde, dass das poröse Medium für die Phase mit der höheren Sättigung permeabler ist, da diese Phase mehr Porenvolumen ausfüllt als die Phase mit der niedrigeren Sättigung und somit mehr zum Transport beiträgt. Weiterführende Betrachtungen finden sich in /BEA 18/, /LI 06/.

In MARNIE2 sind verschiedene Funktionen für die relativen Permeabilitäten der Gas- und Flüssigphase implementiert worden. Diese werden über die Eingabe für jedes P-Modul ausgewählt. Die dritte Eingabezeile der Karte -----P-MODUL.....NR----- hat jetzt die Form:

Spalte	1	12	13	24	25	36	37	48	49	60	61	72
	KONVER		CHEMKOP		PORAUS		PERMART		PCAPART		TLANG	
Feld	1		2		3		4		5		6	

Neue Eingabeparameter sind hier und im Folgenden farblich hervorgehoben. Mit dem Eingabeparameter PERMART wird die Funktion für die relative Permeabilität ausgewählt. In der 15. Zeile der Karte für das P-Modul werden die residualen Sättigungen sowie bei Bedarf der Parameter M der van Genuchten Funktion vorgegeben, diese Zeile muss für jede Box des P-Moduls wiederholt werden:

Spalte	1	12	13	24	25	36	37	48	49	60	61	72
	SRES_LIQ		SRES_GAS		M		PCAP0					
Feld	1		2		3		4		5		6	

Die aufgeführten Eingabegrößen finden sich ebenfalls in den Eingaben zu den V04-Modulen und können der modifizierten Eingabebeschreibung entnommen werden.

Im Einzelnen stehen in MARNIE2 folgende Funktionen für die relativen Permeabilitäten zur Verfügung:

- PERMART = MOBILE  
Alle Phasen sind sättigungsunabhängig mobil mit  $\chi_{rel,g} = \chi_{rel,l} = 1$ .
- PERMART = LINEAR  
Die relativen Permeabilitäten wechseln jeweils linear zwischen 0 und 1.
- PERMART = COREY  
Die Funktion für die relative Permeabilität geht auf Corey /COR 54/ zurück, eine detaillierte Herleitung findet sich in /LI 06/. Die Corey-Kurve hat die Form

$$\chi_{rel,l} = S_{eff}^4 \quad \text{und} \quad \chi_{rel,g} = (1 - S_{eff})^2 - (1 - S_{eff}^2) \quad (2.6)$$

mit der effektiven Sättigung

$$S_{eff} = \frac{S_l - S_{res,l}}{1 - S_{res,l} - S_{res,g}} \quad (2.7)$$

- PERMART = GRANT  
Die Kurve von Grant /GRA 77/ ist eine Modifikation der Kurve von Corey und hat die Form

$$\chi_{rel,l} = S_{eff}^4 \quad \text{und} \quad \chi_{rel,g} = 1 - \chi_{rel,l}, \quad (2.8)$$

mit  $S_{eff}$  aus Gl. (2.7).

- PERMART = FATT  
Die Funktion von Fatt und Klikoff /FAT 59/ hat die Form

$$\chi_{rel,l} = S_{eff}^3 \quad \text{und} \quad \chi_{rel,g} = (1 - S_{eff})^3, \quad (2.9)$$

mit  $S_{eff} = (S_l - S_{res,l}) / (1 - S_{res,l})$ .

- PERMART = VANGEN

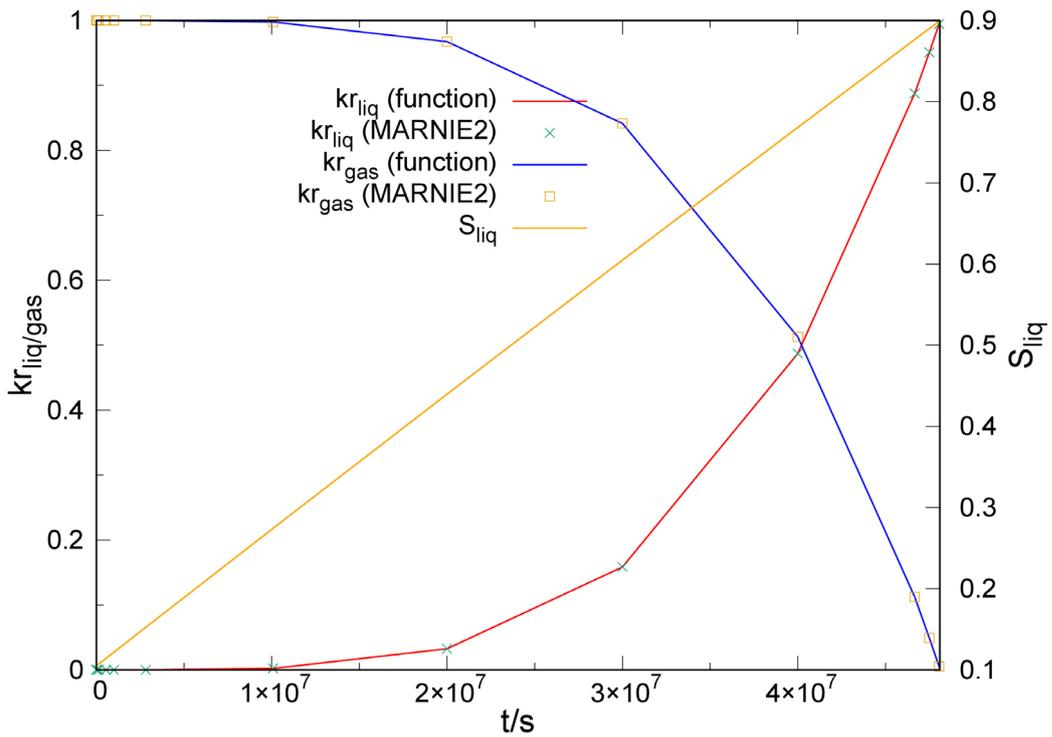
Hiermit wird die Funktion von van Genuchten /GEN 80/ aktiviert. Diese hat die Form

$$\chi_{rel,l} = \sqrt{S_{eff}} \left( 1 - \left( 1 - S_{eff}^{\frac{1}{m}} \right)^m \right)^2 \quad (2.10)$$

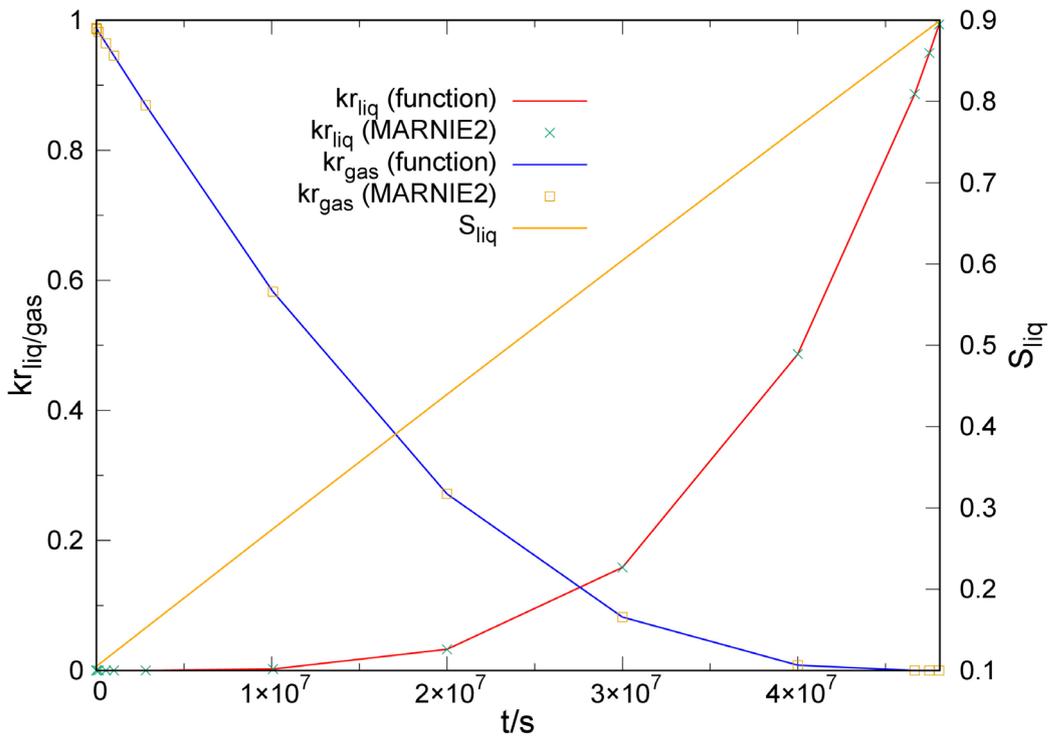
mit  $0 < m < 1$  und der effektiven Sättigung nach Gl. (2.7). Für die relative Permeabilität der Gasphase wird die Funktion nach Corey, Gl. (2.6), für  $S_{res,g} > 0$ , bzw. nach Gl. (2.8) für  $S_{res,g} = 0$ .

In /BEA 18/, /LI 06/ sind weitere Funktionen für die relativen Permeabilitäten zu finden, die wegen der modularen Struktur von MARNIE2 leicht hinzugefügt werden können.

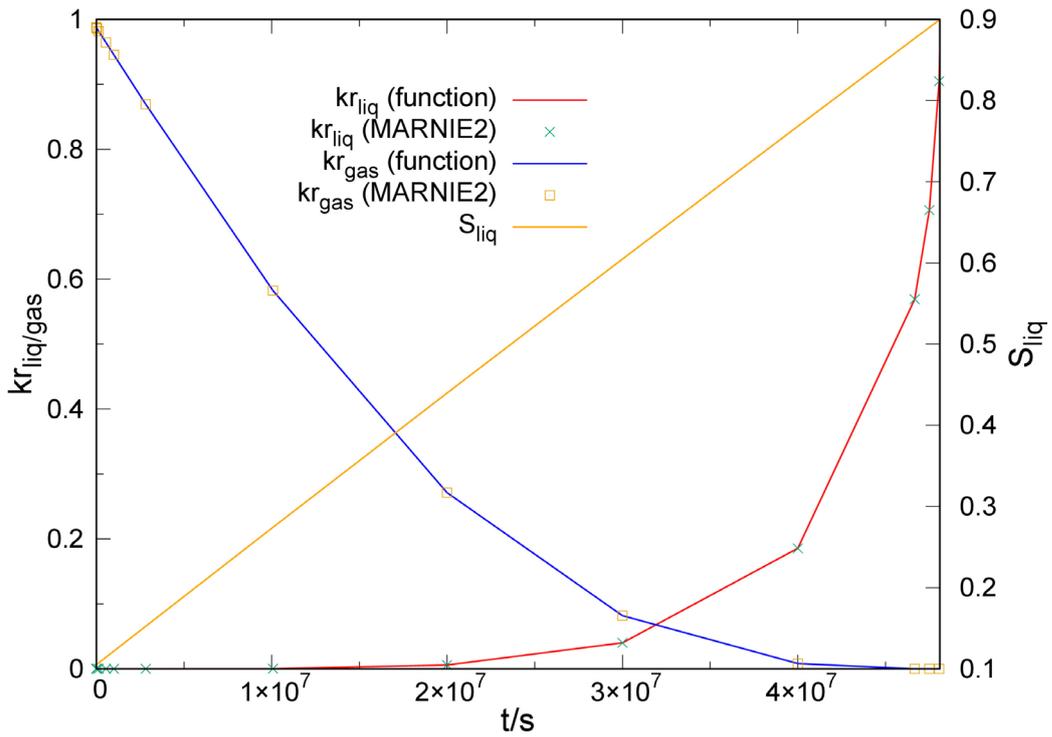
Um zu überprüfen, ob die Funktionen für die relativen Permeabilitäten korrekt implementiert sind, wurde MARNIE2 in der Weise gestartet, dass die Phasensättigung  $S_l$  den vollen Bereich bis zur Sättigung durchläuft. Die in MARNIE2 berechneten relativen Permeabilitäten werden zusammen mit den relativen Permeabilitäten aus den Gl. (2.6) bis (2.10) aufgetragen. Die Abb. 2.4 bis Abb. 2.7 zeigen eine gute Übereinstimmung und die korrekte Implementierung der Funktionen.



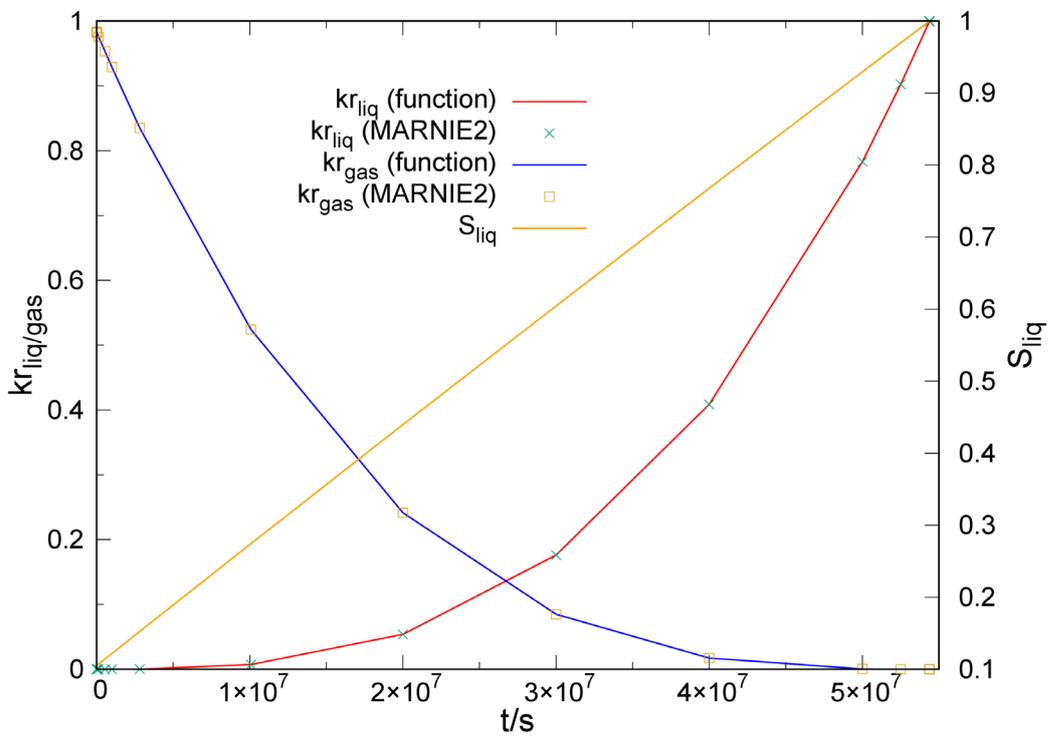
**Abb. 2.4** Funktion nach Grant für die relativen Permeabilitäten; Die Berechnung aus MARNIE2 ist im Vergleich zu Gl. (2.8) gezeigt



**Abb. 2.5** Funktion nach Corey für die relativen Permeabilitäten; Die Berechnung aus MARNIE2 ist im Vergleich zu Gl. (2.6) gezeigt



**Abb. 2.6** Funktion nach Fatt und Klikoff für die relativen Permeabilitäten; Die Berechnung aus MARNIE2 ist im Vergleich zu Gl. (2.9) gezeigt



**Abb. 2.7** Funktion nach van Genuchten für die relativen Permeabilitäten; Die Berechnung aus MARNIE2 ist im Vergleich zu Gl. (2.10) gezeigt

## 2.5 Kapillardruckkurven

Der Druck in der flüssigen Phase ist in MARNIE2 als  $p_l = p_g - p_{cap}$  vorgegeben. Der Kapillardruck  $p_{cap}$  ist eine wichtige Eigenschaft zweiphasigen Transports in porösen Medien und wird über passende Kapillardruckfunktionen vorgegeben /BEA 18/. Da für die Berücksichtigung des Kapillardrucks in MARNIE2 umfassende Änderungen im Quellcode durchzuführen waren, ist zunächst nur die häufig verwendete van Genuchten Beziehung implementiert worden. Weitere Funktionen können aufgrund der modularen Programmierweise leicht hinzugefügt werden.

Die van Genuchten Funktion für den Kapillardruck hat die Form /GEN 80/

$$p_{cap} = p_0 \left( \frac{1}{S_{eff}^{1/m}} - 1 \right)^{1/n} \quad (2.11)$$

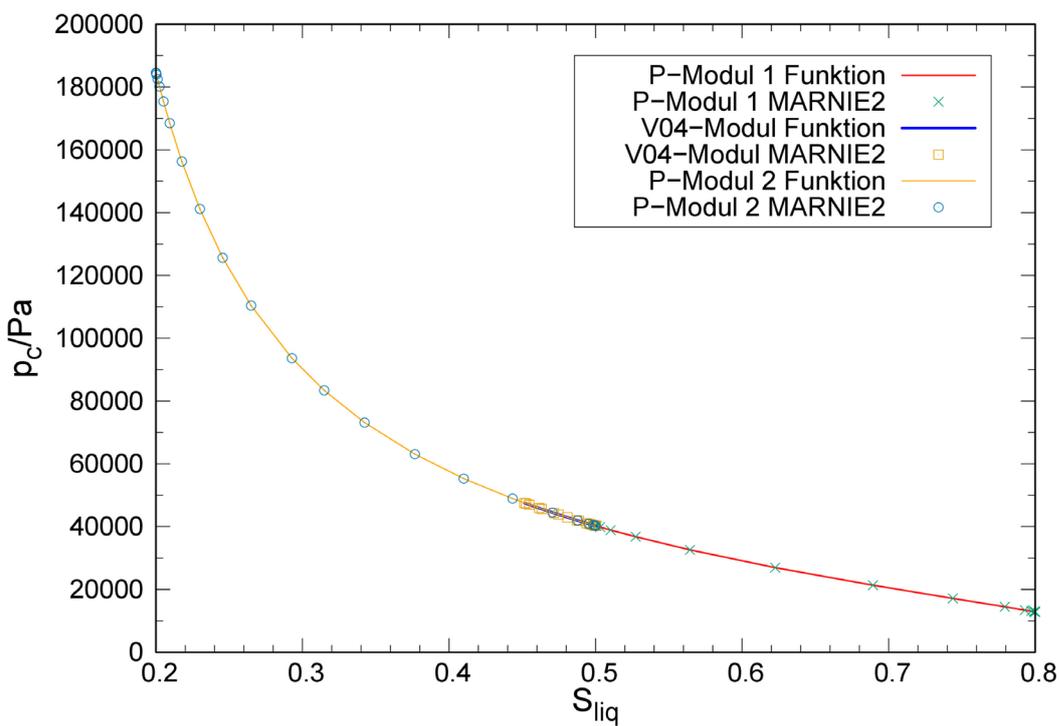
mit  $1/n = 1 - m$  und  $S_{eff}$  aus Gl. (2.7).  $p_0, n$  und  $m$  sind materialabhängige Größen des porösen Mediums /BEA 18/. Die van Genuchten Funktion für den Kapillardruck wird in MARNIE2 über die Eingabe für jedes P-Modul ausgewählt, die dritte Eingabezeile der Karte -----P-MODUL.....NR----- hat jetzt die Form:

Spalte	1	12	13	24	25	36	37	48	49	60	61	72
	KONVER		CHEMKOP		POROUS		PERMART		PCAPART		TLANG	
Feld	1		2		3		4		5		6	

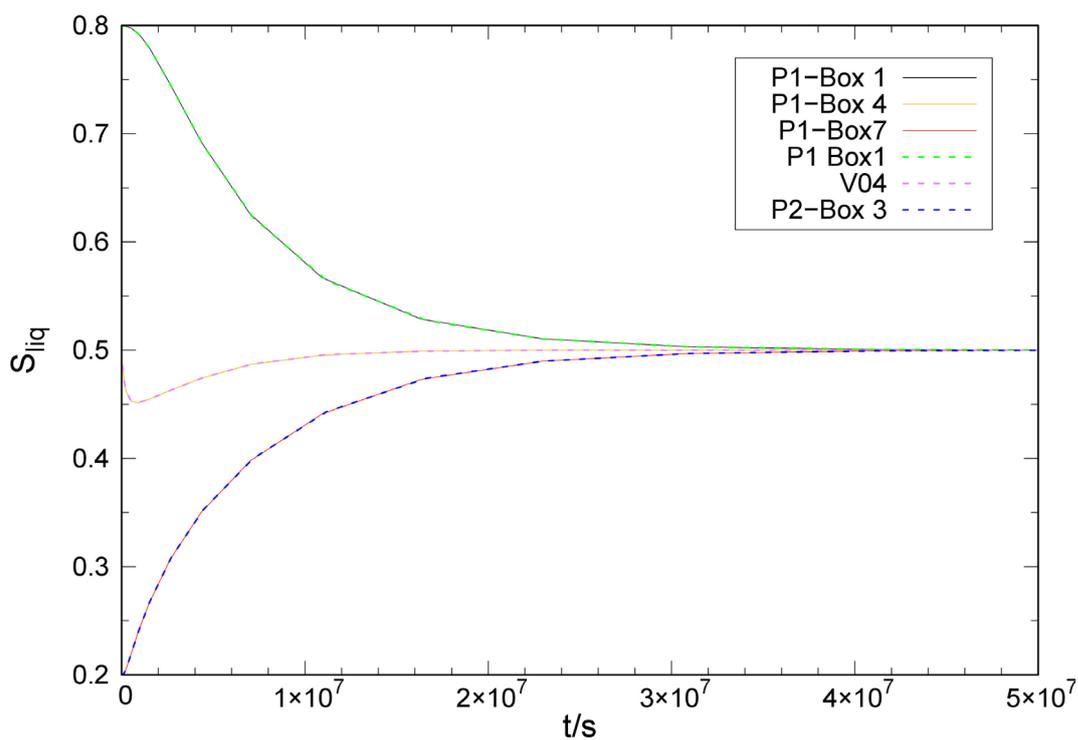
Mit dem Eingabeparameter PCAPART = VANGEN wird die Funktion für die relative Permeabilität ausgewählt.

Um zu testen, ob die Kapillardruckfunktion nach van Genuchten korrekt implementiert wurde, wurde ein Plausibilitätstestfall für ein System entsprechend der Diskretisierung in Kapitel 2.2 betrachtet. Für die Boxen 1 bis 3 des P-Moduls 1 ist initial eine Flüssigkeitssättigung von  $S_l = 0,8$  vorgegeben, für die Box 4 des P-Moduls 1 bzw. des V04-Moduls ist initial  $S_l = 0,5$  und für die Boxen 5 bis 7 des P-Moduls 1 bzw. Boxen 1 bis 3 des P-Moduls 2 ist initial  $S_l = 0,2$  vorgegeben. Über die Ränder der beiden Diskretisierungen ist ausschließlich Gastransport möglich (die Änderungen in der Implementierung der V03-Module sind in Kapitel 2.6 beschrieben). In Abb. 2.8 ist der Kapillardruck nach Gl. (2.11) in Abhängigkeit von der Flüssigkeitssättigung aus MARNIE2 zusammen mit

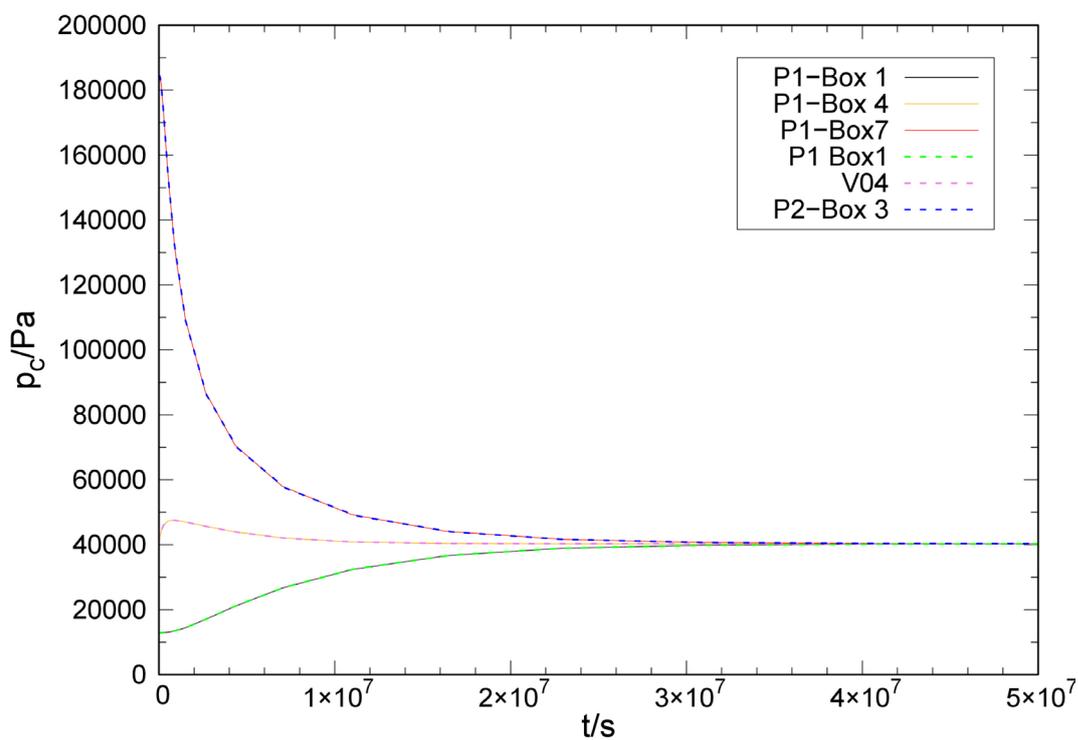
dem in MARNIE2 berechneten Kapillardruck aufgetragen. Das Ergebnis zeigt die korrekte Implementierung der Kapillardruckfunktion nach van Genuchten in MARNIE2. Die zeitliche Entwicklung des Plausibilitätstestfalls ist für die Flüssigkeitssättigung in Abb. 2.9 gezeigt. Wie zu erwarten, gleicht sich die Sättigung über die komplette Länge des Systems im Laufe der Zeit durch die Unterschiede im Kapillardruck, siehe Abb. 2.10, an. Der Kapillardruck ist hierbei die einzige treibende Kraft. Interessanterweise sinkt  $S_l$  aufgrund des nichtlinearen Kapillardruckprofils in der mittleren Box zunächst unter den späteren Gleichgewichtswert, um sich dann dem stationären Zustand anzunähern.



**Abb. 2.8** Kapillardruckfunktion nach van Genuchten; Die Berechnung aus MARNIE2 ist im Vergleich zu Gl. (2.11) gezeigt



**Abb. 2.9** Flüssigkeitssättigung für den Plausibilitätstestfall zum Kapillardruck



**Abb. 2.10** Verlauf des Kapillardrucks für den Plausibilitätstestfall zum Kapillardruck

## 2.6 Erweiterungen in den Randmodulen

Die V03-Module in MARNIE2 sind mit dem Ziel erweitert worden, Randbedingungen (Druck und Fluss) mit verschiedenen Phasenzuständen vorzugeben. Dazu werden zunächst für den Rand die residualen Phasensättigungen  $S_{res\_03,l}$  bzw.  $S_{res\_03,g}$  vorgegeben. Dies geschieht über die erste Eingabezeile der Karte -----V-MODUL.....NR-----. Diese hat jetzt die Form:

Spalte	1	12	13	24	25	36	37	48	49	60	61	72
	IPRINT		IDRUCK		SRES_LIQ03		SRES-GAS03		-		-	
Feld	1		2		3		4		5		6	

Die Sättigung in den V03-Modulen kann dann über eine Tabelle zeitabhängig vorgegeben werden. Dies geschieht mit der dritten Eingabezeile der Karte:

Spalte	1	12	13	24	25	36	37	48	49	60	61	72
	IRAND		NX		IKENRA		-		-		-	
Feld	1		2		3		4		5		6	

Mit IRAND = 3 wird eine Tabelle für die Flüssigkeitssättigung  $S_{l\_03}$  vorgegeben, NX ist die Nummer der entsprechenden Tabelle, IKENRA = 2 ist zu wählen. Wenn in dem jeweiligen V03-Modul keine Flussrandbedingung vorgegeben ist, gilt:

- $S_{l\_03} < S_{res\_03,l}$ : Der Transport auf dem Rand ist einphasig gasförmig.
- $1 - S_{l\_03} = S_{g\_03} < S_{res\_03,g}$ : Der Transport auf dem Rand ist einphasig flüssig.
- $S_{l\_03} > S_{res\_03,l}$  und  $S_{g\_03} > S_{res\_03,g}$ : Der Transport auf dem Rand ist zweiphasig.

Es wird außerdem aus der vorgegebenen Sättigung der zugehörige Kapillardruck berechnet, wozu die Kapillardruckfunktion des angrenzenden P-Moduls ausgewählt wird. Der so ermittelte Kapillardruck geht bei einer Druckrandbedingung in den Druck auf dem Rand für die flüssige Phase mit ein; es ist

$$p_{l\_03} = p_{g\_03} + p_{cap\_03}. \quad (2.12)$$

## 2.7 Phasenwechsel

In vielen Situationen ist es nötig, zusätzlich zu zweiphasigem Transport, einphasig flüssigen oder einphasig gasförmigen Transport und die Übergänge zwischen diesen Zuständen beschreiben zu können. So sind z. B. reale Einzelprozesse wie das Fluten eines Endlagersystems oder Gasproduktion mit Gasmigration in flüssigkeitsgesättigte Bereiche eines Endlagers (Gaseindringen) nur durch Abbildung der genannten Phasenzustände adäquat im Rechenprogramm abbildbar.

Der einphasige Transport wird über die Darcygleichung (2.3) beschrieben; die programmiertechnische Herausforderung besteht darin, in MARNIE2 kontrolliert und effizient zwischen diesen Phasenzuständen umzuschalten. Dazu müssen am Ende eines erfolgreichen Zeitintegrationsschrittes des Differentialgleichungslösers FEBE, Differentialgleichungen aktiviert bzw. deaktiviert und Anfangswerte gesetzt bzw. Werte abgespeichert werden. Hierbei sollen die definierten Umschaltunkte präzise getroffen werden, ohne die Laufzeit des Rechenprogramms unnötig zu vergrößern.

### 2.7.1 Implementierte Einzelprozesse

Die technische Umsetzung des Übergangs zwischen zwei Phasenzuständen soll im Folgenden am Übergang von zweiphasigem Transport zu einphasig flüssigem Transport beispielhaft gezeigt werden. Dieser Übergang findet statt, wenn die Box eines P-Moduls geflutet wird, wodurch die Gasphase verdrängt wird. Zum Zeitpunkt des Phasenübergangs wird

$$S_g = S_{res,g} \quad (2.13)$$

erstmalig erfüllt und der Phasenzustand wird gewechselt. Wenn die Bedingung in Gl. (2.13) bei zweiphasigem Transport innerhalb eines Makrozeitschritts der FEBE erfüllt ist, so wird der Makrozeitschritt verlassen und mit einer kleineren Zeitschrittweite neu gestartet. Ziel ist es, dass am Ende eines Makrozeitschrittes die Bedingung

$$S_{res,g} - 0,01 \cdot S_{res,g} \leq S_g < S_{res,g} \quad (2.14)$$

erfüllt ist. Das in Gl. (2.14) definierte Band dient dazu, den Phasenwechsel effizient zu gestalten und durch das Umschalten bedingte Oszillationen zu vermeiden; wenn die Bedingung (2.13) erfüllt sein müsste, kann dies je nach zeitlichem Verlauf der Gassättigung zu einer Vielzahl von Reduktionen des Makrozeitschritts durch die FEBE führen und so den rechnerischen Aufwand stark erhöhen. Das hier definierte Band bildet nicht eine eventuelle physikalisch bedingte Hysterese ab, sondern hat ausschließlich die vorgenannte Motivation.

Ist die Bedingung (2.14) für die Gassättigung am Ende eines Makrozeitschritts der FEBE erfüllt, so werden die Differentialgleichungen für den Gastransport auf den Rändern der entsprechenden Box des P-Moduls sowie die Differentialgleichung für die Gasmasse in der Boxmitte deaktiviert. Die Gasmasse, die immobil in der Box verbleibt, wird als potentielle Anfangsbedingung für erneuten Gastransport gespeichert. Die Kompression des Gases ist durch zuströmende Flüssigkeit aufgrund entsprechender Druckerhöhung weiter möglich. Der nächste Makrozeitschritt startet dann mit einem aktualisierten Satz an Differentialgleichungen.

Nach der zuvor beschriebenen Vorgehensweise sind in MARNIE2 die in Tab. 2.1 zusammengefassten Prozesse implementiert worden. Für den Fall zweiphasigen Transports sind die internen Flags bei MARNIE2 boxweise auf `FLPHAS = FALSE.` und `GASPHAS = FALSE.` gesetzt. Der Gaseindringdruck ist in Anlehnung an Konrad momentan mit  $p_{ein} = 100$  kPa fest vorgegeben, soll aber in einer zukünftigen Version von MARNIE2 als Eingabeparameter vorgegeben werden können.

**Tab. 2.1** In MARNIE implementierte Umschaltprozesse zusammen mit den jeweiligen Bedingungen und den genutzten logischen Flags

Prozess	Bedingung	Flags in MARNIE2 nach Umschalten
zweiphasig → einphasig flüssig	$S_{res,g} - 0,01 \cdot S_{res,g} \leq S_g < S_{res,g}$	FLPHAS = .TRUE. GASPHAS = .FALSE.
zweiphasig → einphasig gasförmig	$S_{res,l} - 0,01 \cdot S_{res,l} \leq S_l < S_{res,l}$	FLPHAS = .FALSE. GASPHAS = .TRUE.

Prozess	Bedingung	Flags in MARNIE2 nach Umschalten
einphasig gasförmig → Eindringen von Flüssigkeit	$100 \text{ Pa} \leq \Delta p$ , wobei die 100 Pa eingeführt wurden, um einer zu häufigen Verkleinerung des FEBE Makroschrittes vorzubeugen.	FLPHAS = .FALSE. GASPHAS = .TRUE. FLUFL = .TRUE.
Eindringen von Flüssigkeit → zweiphasig	$S_{res,l} \leq S_l < S_{res,l} + 0,01 \cdot S_{res,l}$ , d.h. es ist so viel Flüssigkeit eingedrungen, dass die flüssige Phase mobil wird.	FLPHAS = .FALSE. GASPHAS = .FALSE. FLUFL = .FALSE.
einphasig flüssig → Gaseindringen	<u>Fall a:</u> $p_{ein} \leq \Delta p_g < p_{ein} + 0,01 \cdot p_{ein}$ wobei $\Delta p_g$ die Druckdifferenz zwischen einer Box in der Gastransport möglich ist und der dazu benachbarten einphasig flüssigen Box ist. Entsprechendes gilt für Boxen, die an V-Module angrenzen. <u>Fall b:</u> $p_{ein} \leq \Delta p_g$ , im Gegensatz zu Fall a ist die Gasphase in einer Box oder einem V-Modul im Moment mobil geworden. $\Delta p_g$ kann dann weit ausserhalb des in Fall a angegebenen Bandes sein und es ist sofortiges Gaseindringen in die einphasig flüssige Nachbarbox (oder ein V-Modul) möglich.	FLPHAS = .TRUE. GASPHAS = .FALSE. GASFL = .TRUE.
Gaseindringen → zweiphasig	$S_{res,g} \leq S_g < S_{res,g} + 0,01 \cdot S_{res,g}$ , d.h. es ist so viel Gas eingedrungen, dass die Gasphase mobil wird.	FLPHAS = .FALSE. GASPHAS = .FALSE. GASFL = .FALSE.

## 2.7.2 Fehlerbeseitigung in der FEBE

In der ursprünglichen Version von MARNIE wurden die Rücksprungmöglichkeiten der FEBE zusammen mit einer möglichen Reduktion der Zahl der implizit zu lösenden Differentialgleichungen nicht genutzt. Im Rahmen der Implementierung der vorgenannten Umschaltprozesse in MARNIE2 und der erstmaligen Nutzung dieser Möglichkeiten der FEBE wurden Unstimmigkeiten im Quellcode der FEBE gefunden, die beseitigt wurden:

- Routine `felmhe.f`: Diese Routine reduziert eine vorgegebene Matrix auf die obere Hessenbergform. Hierbei wurde das Indexarray `INT(M)` nicht komplett neu berechnet, da die zugrundeliegende Matrix durch Reduktion der Zahl der impliziten Differentialgleichungen verkleinert wurde. Das Indexarray wird jetzt vor seiner Berechnung komplett zu 0 initialisiert.
- Routine `fdeche.f`: Diese Routine führt die LU-Zerlegung einer Matrix in Hessenbergform durch. Hierbei wurde das Indexarray `IP(N)` nicht komplett neu berechnet, da die zugrundeliegende Matrix durch Reduktion der Zahl der implizit zu lösenden Differentialgleichungen verkleinert wurde. Das Indexarray wird jetzt vor seiner Berechnung komplett zu 0 initialisiert.
- Routine `febe.f`: Nach einem Rücksprung zu Label 150 durch Setzen von `HXX = .TRUE.` musste `NPAC = .FALSE.` gesetzt werden, damit nach dem Rücksprung und der Verkleinerung des Zeitschritts die Jacobimatrix erneut auf Hessenbergform gebracht wird und eine anschließende LU-Zerlegung durchgeführt wird. Beides geschieht in der Routine `fimp.f`, welche die eigentliche Lösung des implizit diskretisierten Differentialgleichungssystems berechnet. Der Rücksprung selbst (durch setzen von `HXX = .TRUE.`) findet außerhalb der FEBE in Abhängigkeit von vorgegebenen Grenzwerten statt, wie z. B. bei den o. g. Umschaltprozessen.

Die in MARNIE2 implementierten Umschaltprozesse sind für den Parameter `IFTRIX = 1` getestet worden und es ist zu prüfen, ob für `IFTRIX = 2` und `IFTRIX = 3` weitere Unstimmigkeiten beseitigt werden müssen. Die gefundenen Unstimmigkeiten fielen bei früheren Rechnungen mit MARNIE nicht ins Gewicht, da die Zahl der Differentialgleichungen nie verändert wurde.

### 2.7.3 Plausibilitätstests für Umschaltprozesse

Die in Tab. 2.2 aufgelisteten Plausibilitätstestfälle wurden entwickelt, um die in MARNIE2 implementierten Umschaltprozesse auf ihre Plausibilität hin überprüfen zu können. Die räumliche Diskretisierung folgt der Darstellung in Kapitel 2.2.

**Tab. 2.2** Plausibilitätstestfälle für die in MARNIE2 implementierten Umschaltprozesse

Name	Anfangsbedingung	Randbedingungen
I-1	In allen Boxen (+ V04-Modul): $p_g = 2 \text{ bar}$ , $S_l = 0,4$ , $\chi_{rel,g} = \chi_{rel,l} = 1$ , kein Kapillardruck, $S_{res,l} = S_{res,g} = 0,1$	V03 links: $G_l = 10^{-5} \text{ m/s}$ auf $10^{-7} \text{ m/s}$ V03 rechts: $p_g = 2 \text{ bar}$
I-2	wie I-1	V03 links: $G_g = 10^{-3} \text{ m/s}$ auf $10^{-5} \text{ m/s}$ V03 rechts: wie I-1
II-1	wie I-1	V03 links: $p_g = 3,5 \text{ bar}$ , $S_{res\_03,l} =$ $S_{res\_03,g} = 0,1$ , $S_{l\_03} = 0,4$ V03 rechts: $p_g = 2 \text{ bar}$ , $S_{res\_03,l} =$ $S_{res\_03,g} = 0,1$ , $S_{l\_03} = 0,4$
II-2	wie I-1	V03 links: $p_g = 3,5 \text{ bar}$ , $S_{res\_03,l} =$ $S_{res\_03,g} = 0,1$ , $S_{l\_03} = 0,04$ V03 rechts: wie II-1
II-3	wie I-1	V03 links: $p_g = 3,5 \text{ bar}$ , $S_{res\_03,l} =$ $S_{res\_03,g} = 0,1$ , $S_{l\_03} = 0,95$ V03 rechts: wie II-1
III-1	In allen Boxen (+ V04-Modul): $p_g = 2 \text{ bar}$ , $S_l = 0,95$ , $\chi_{rel,g} = \chi_{rel,l} = 1$ , kein Kapillardruck, $S_{res,l} = S_{res,g} = 0,1$	V03 links: $G_g = 10^{-3} \text{ m/s}$ V03 rechts: $p_g = 2 \text{ bar}$ , $S_{res\_03,l} =$ $S_{res\_03,g} = 0,1$ , $S_{l\_03} = 0,4$
III-2	wie III-1	V03 links: $p_g = 3,5 \text{ bar}$ , $S_{res\_03,l} =$ $S_{res\_03,g} = 0,1$ , $S_{l\_03} = 0,04$ V03 rechts: wie III-1
III-3	wie III-1	V03 links: $p_g = 5,5 \text{ bar}$ , $S_{res\_03,l} =$ $S_{res\_03,g} = 0,1$ , $S_{l\_03} = 0,04$ V03 rechts: wie III-1
IV-1	In allen Boxen (+ V04-Modul): $p_g = 2 \text{ bar}$ , $S_g = 0,95$ , $\chi_{rel,g} = \chi_{rel,l} = 1$ , kein Kapillardruck, $S_{res,l} = S_{res,g} = 0,1$	V03 links: $G_l = 10^{-5} \text{ m/s}$ V03 rechts: $p_g = 2 \text{ bar}$ , $S_{res\_03,l} =$ $S_{res\_03,g} = 0,1$ , $S_{l\_03} = 0,4$
IV-2	wie IV-1	V03 links: $p_g = 3,5 \text{ bar}$ , $S_{res\_03,l} =$ $S_{res\_03,g} = 0,1$ , $S_{l\_03} = 0,95$ V03 rechts: wie IV-1

Name	Anfangsbedingung	Randbedingungen
V-1	In allen Boxen (+ V04-Modul): $p_g = 2 \text{ bar}$ , $S_l = 0,5$ , $\chi_{rel,g} = \chi_{rel,l} = 1$ , kein Kapillardruck, $S_{res,l} = S_{res,g} = 0,1$	V03 oben: $p_g = 2 \text{ bar}$ , $S_{res_{03,l}} = S_{res_{03,g}} = 0,1$ , $S_{l_{03}} = 0,04$ V03 unten: no flow Randbed.
V-2	wie V-1	V03 oben: $p_g = 2 \text{ bar}$ , $S_{res_{03,l}} = S_{res_{03,g}} = 0,1$ , $S_{l_{03}} = 0,04$ V03 unten: $p_g = 2 \text{ bar}$ , $S_{res_{03,l}} = S_{res_{03,g}} = 0,1$ , $S_{l_{03}} = 0,4$
Va-1	Diskretisierung im Falle zweier P-Module so verändert, dass das obere P-Modul vier Boxen, das untere zwei Boxen hat. Ansonsten wie V-1	wie V-1
Vb-1	Diskretisierung im Falle zweier P-Module so verändert, dass das obere P-Modul zwei Boxen, das untere vier Boxen hat. Ansonsten wie V-1	wie V-1

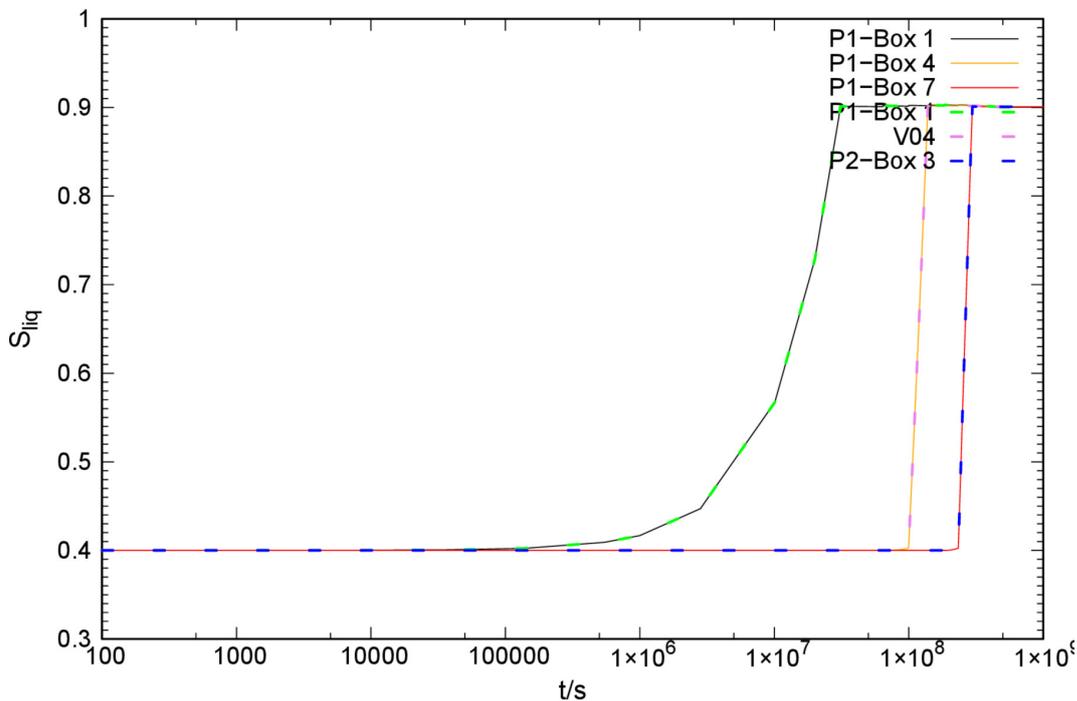
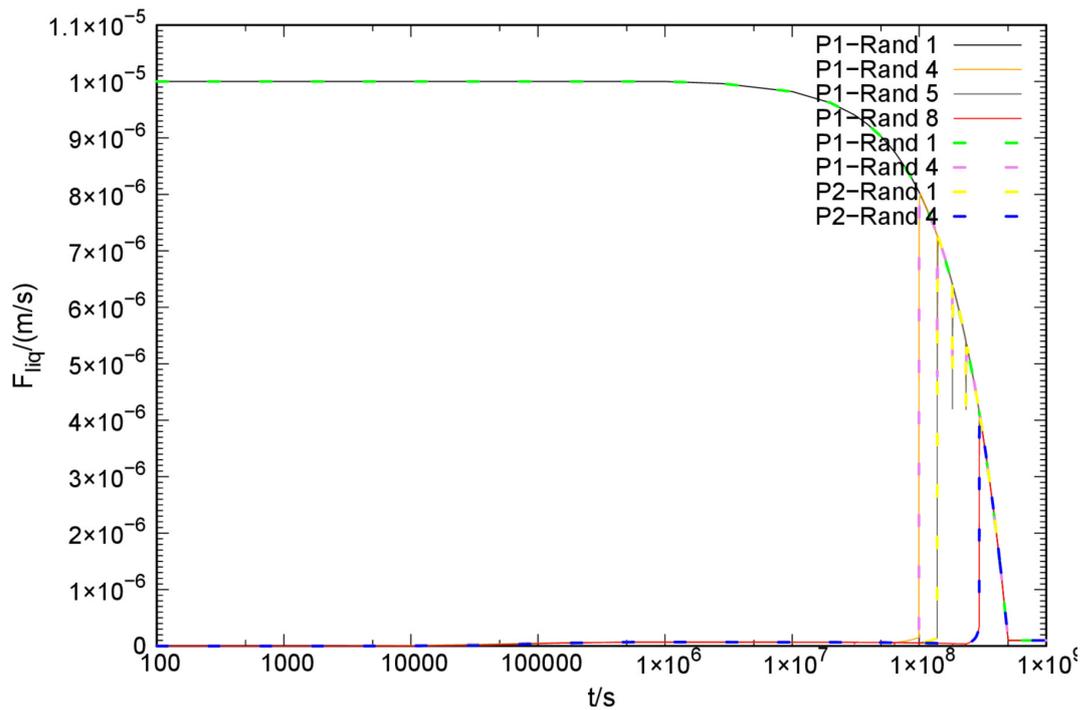


Abb. 2.11 Sättigungsverlauf für den Testfall I-1

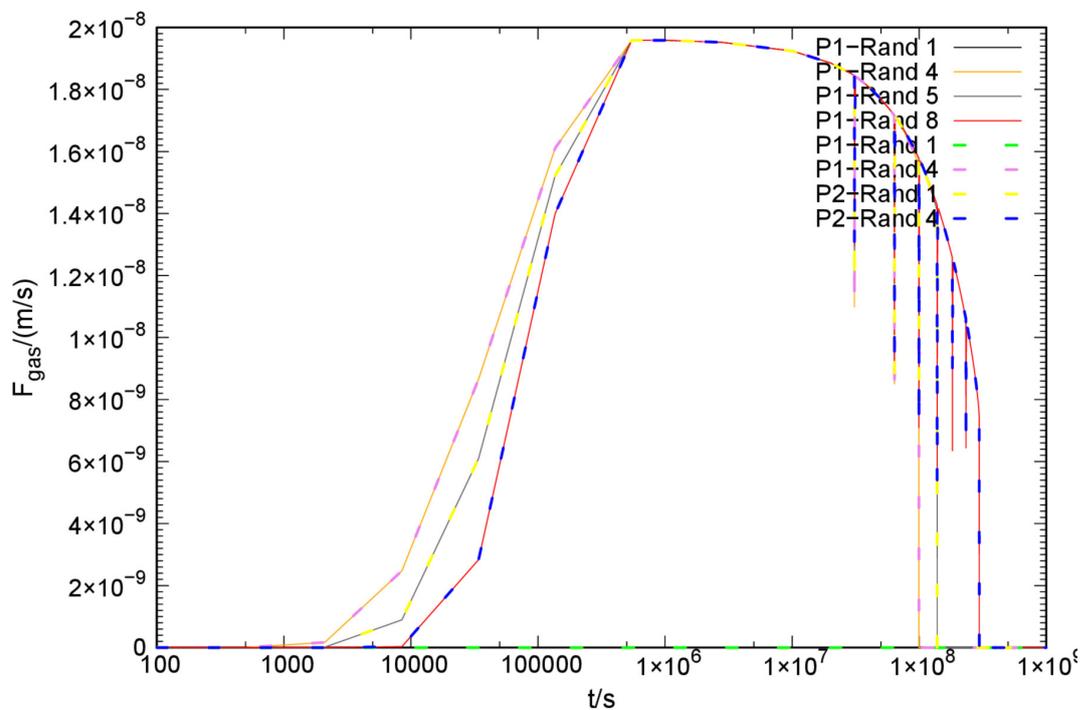
Um die Grundidee der Plausibilitätstestfälle zu verdeutlichen, soll zunächst der Testfall I-1 betrachtet werden. Der Testfall beschreibt den Umschaltprozess von zweiphasigem Transport zu einphasig flüssigem Transport und vernachlässigt bewusst die treibende Kraft des Kapillardrucks für den zweiphasigen Transport, um so eine Fokussierung auf den Umschaltprozess zu erlauben. Damit bildet der Plausibilitätstestfall eine artifizielle Situation ab. Der Verlauf der Flüssigkeitssättigung  $S_l$  ist in der Abb. 2.11 gezeigt. Durch die vorgegebene Randbedingung füllt sich zunächst die angrenzende erste Box des P-Moduls bis zur vollständigen Sättigung bei  $S_l = 0,9$ , dann kommt es zum Umschaltprozess auf einphasig flüssige Bedingungen in der ersten Box; sukzessive werden dann die restlichen Boxen des Systems auf einphasig flüssige Bedingungen umgeschaltet. Die Resultate für die vierte Box des ersten P-Moduls und das V04-Modul für die Diskretisierung liegen übereinander, was zeigt, dass die Umschaltprozesse auch für die Ränder des P-Moduls bzw. für das V04-Modul korrekt implementiert sind. Diese Umschaltprozesse werden in der MARNIE2-Ausgabe notiert und haben die in Abb. 2.12 gezeigte Form; die vorherigen Rücksprünge in der FEBE werden ebenfalls ausgegeben. Der Massenfluss für die flüssige und die gasförmige Phase ist in der Abb. 2.13 und Abb. 2.14 gezeigt. Die Übereinstimmung zwischen beiden räumlichen Diskretisierungen zeigt die korrekte Implementierung in den P- und V04-Modulen. Diese Übereinstimmung war zu Beginn der programmiertechnischen Arbeiten nicht gegeben und wurde durch eine kontinuierliche Überprüfung der Arbeitsfortschritte mit Hilfe der Plausibilitätstestfälle erreicht. Hierbei zeigten sich insbesondere Inkonsistenzen bei der Mittelung skalarer Größen der Boxmittelpunkte auf den Boxrändern zur Berechnung der dortigen vektoriellen Größen, in diesem Fall der Massenflüsse.

HXX in P-Modul 1 Box 1 bei KEYG\$\$ 5 SGAS = 0.98827D-01 < SGASRES = 0.10000D+00 Der Zeitschritt wird wiederholt.
Phasenwechsel: In P-Modul 1 Box 1 wird von zweiphasigen Bedingungen auf einphasig fluessige Bedingungen umgeschaltet. Residuale Gasmasse: 0.11646D+00 kg, t =0.3088541667D+08

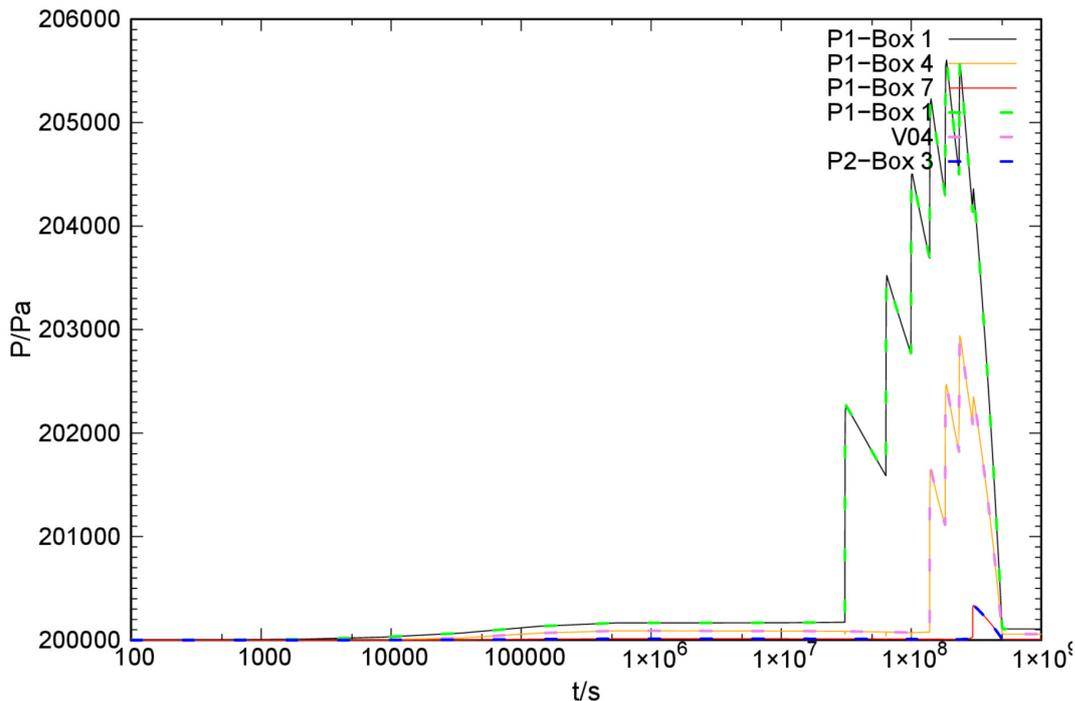
**Abb. 2.12** Ausgabe des Flags für den Rücksprung in der FEBE (oben) und Ausgabe für erfolgreichen Phasenwechsel (unten)



**Abb. 2.13** Massenfluss für die flüssige Phase über die Boxränder. Der Fluss auf dem 1. Rand des P-Moduls (P1-Rand1) entspricht der vorgegebenen Flussrandbedingung



**Abb. 2.14** Massenfluss für die Gasphase auf den Boxrändern

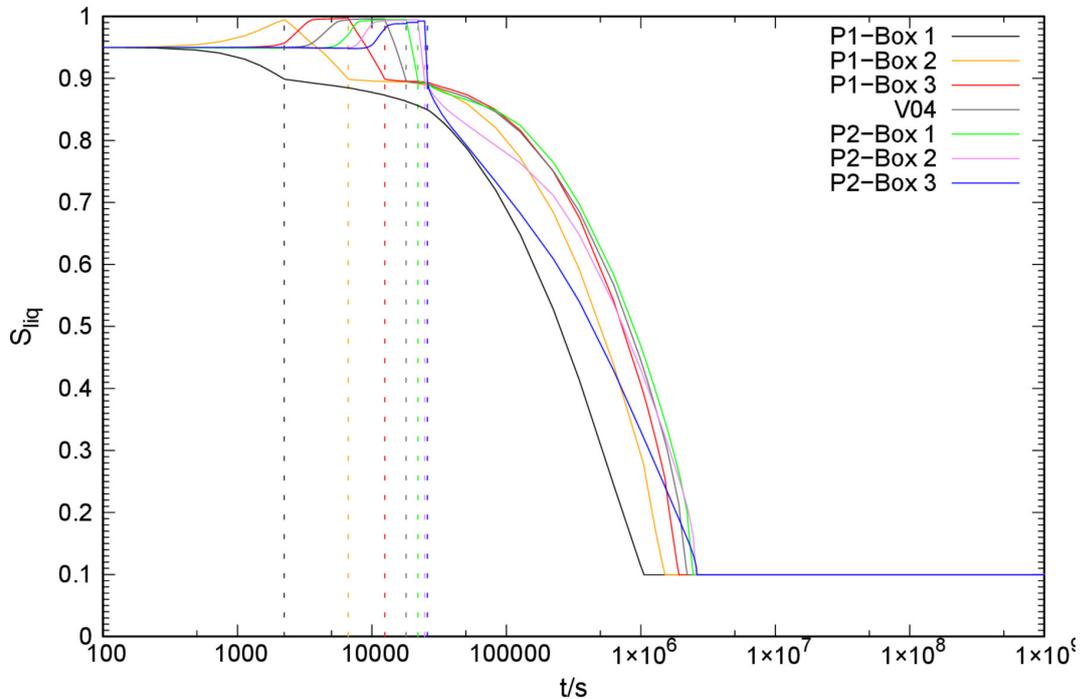


**Abb. 2.15** Druckverlauf für den Plausibilitätstestfall I-1

Auffallend in dem in Abb. 2.11 gezeigten Sättigungsverlauf ist, dass die weiteren Boxen des P-Moduls erst beginnen vollzulaufen, wenn in der Box davor auf einphasig flüssige Bedingungen umgeschaltet wurde. Dies ist eine direkte Folge der eingangs beschriebenen (unphysikalischen) Vernachlässigung des Kapillardrucks. Die einzige treibende Kraft im System ist der Gasdruck, der sich durch die Ausdehnung des Flüssigkeitsvolumens erhöht, da das Gasvolumen in der Folge komprimiert wird. Solange der Massenstrom über einen Boxrand zweiphasig ist, führt diese Druckerhöhung in erster Linie zu Gastransport, da der Darcywiderstand (letzter Term in Gl. (2.3)) der flüssigen Phase sehr viel höher als der der Gasphase ist. Erst wenn in der Box durch den Zustrom von Flüssigkeit auf einphasig flüssige Bedingungen gewechselt wurde und damit kein Gastransport über den Boxrand mehr möglich ist, wird das Gasvolumen durch den weiteren Zustrom von Flüssigkeit so stark komprimiert, dass der Anstieg des Drucks ausreicht, um Flüssigkeit über den Boxrand in die benachbarte zweiphasige Box zu transportieren. Dieses Verhalten zeigt sich im in Abb. 2.15 gezeigten Druckverlauf der einzelnen Boxen.

Das Verhalten der weiteren Testfälle I und II ist ähnlich und soll hier nicht diskutiert werden. Als Beispiel für den Prozess des Gaseindringens soll im Folgenden der Testfall III-1 aus Tab. 2.2 betrachtet werden. Hierzu ist das System initial mit  $S_l = 0,95$ ,  $S_{res,l} = 0,1$  und  $S_{res,g} = 0,1$  in allen Boxen, bzw. bei der zweiten Diskretisierung auch im V04-Modul, vollständig gesättigt. Die Randbedingung wird nun so vorgegeben, dass Gas über den

Rand eindringen kann und die Flüssigkeit verdrängt. Sobald  $S_l < S_{res,l}$  findet zweiphasiger Transport statt, in der Folge wird die Flüssigkeit sukzessive aus den Boxen verdrängt; hierzu müssen entsprechend Differentialgleichungen aktiviert bzw. deaktiviert werden und Anfangsbedingungen korrekt gesetzt werden.

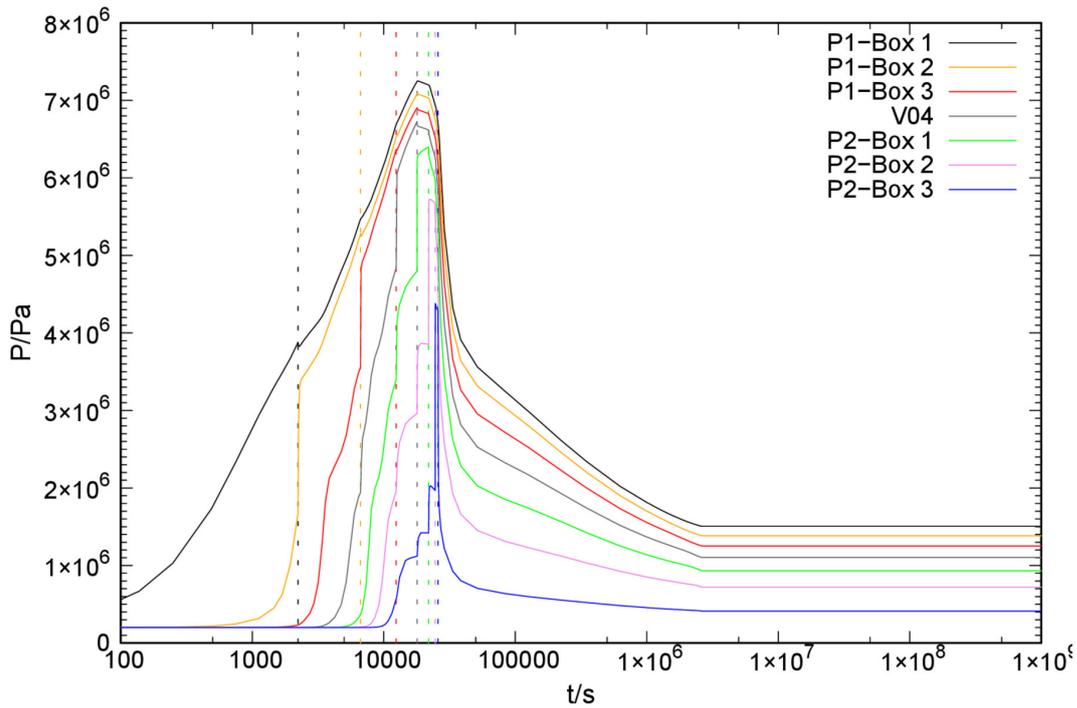


**Abb. 2.16** Verlauf der Flüssigkeitssättigung für den Testfall III-1; Die senkrechten Linien markieren den Übergang von Gaseindringen zu zweiphasigem Transport

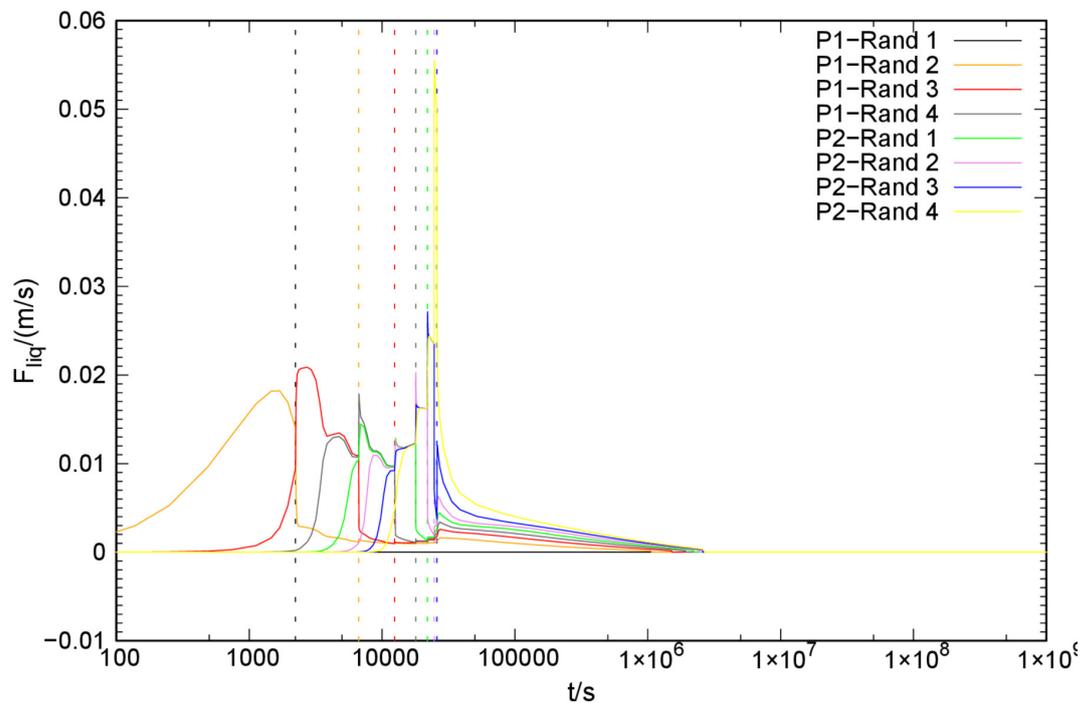
In der Abb. 2.16 ist der beschriebene Verlauf der Flüssigkeitssättigung gezeigt. Das Gas tritt über die vorgegebene Randbedingung zu und verdrängt die flüssige Phase in der Box 1 des ersten P-Moduls in die angrenzende Box 2 des P-Moduls.

Wenn  $S_g = 1 - S_l < S_{res,g}$  in der ersten Box erreicht ist, wird die Gasphase mobil und kann in die zweite Box eindringen und dort weiter die flüssige Phase verdrängen. Dieses schrittweise Ablaufen des Gaseindringens im ganzen System spiegelt die räumliche Diskretisierung des Systems wider, in der Boxgrößen (wie Sättigung und Druck) über die komplette Box homogenisiert sind. Als Konsequenz kann das Gaseindringen nicht als Ausbreiten einer Gasblase in einer Box verstanden werden, vielmehr ist die Gasmasse immer über das gesamte Boxvolumen gleich verteilt. Zum gezeigten Sättigungsverhalten korrespondiert der in Abb. 2.17 gezeigte Druckverlauf im System. Der Druck in den Boxen baut sich, von den zwischenzeitlichen Änderungen im Verlauf aufgrund einzelner Phasenwechselprozesse und damit einhergehender Flüssigkeitsverdrängung abgese-

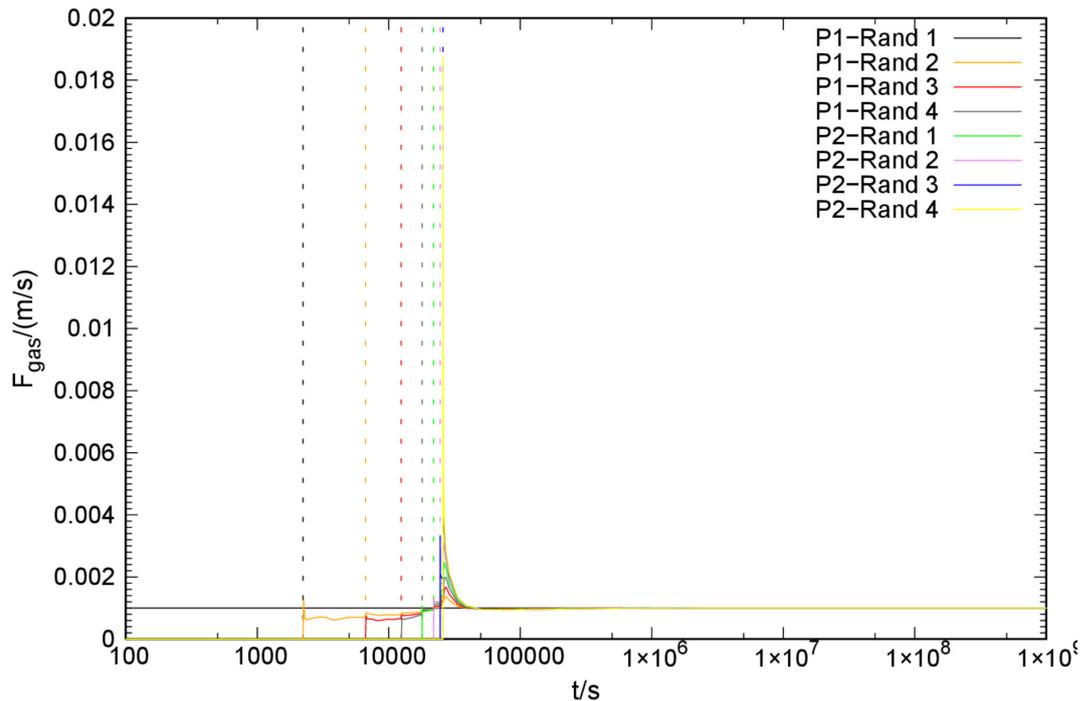
hen, immer weiter auf, bis in der letzten Box des Systems (P2-Box 3) Gastransport möglich wird und das Gas über den rechten Rand abfließen kann.



**Abb. 2.17** Druckverlauf für den Testfall III-1; Die senkrechten Linien markieren den Übergang von Gaseindringen zu zweiphasigem Transport



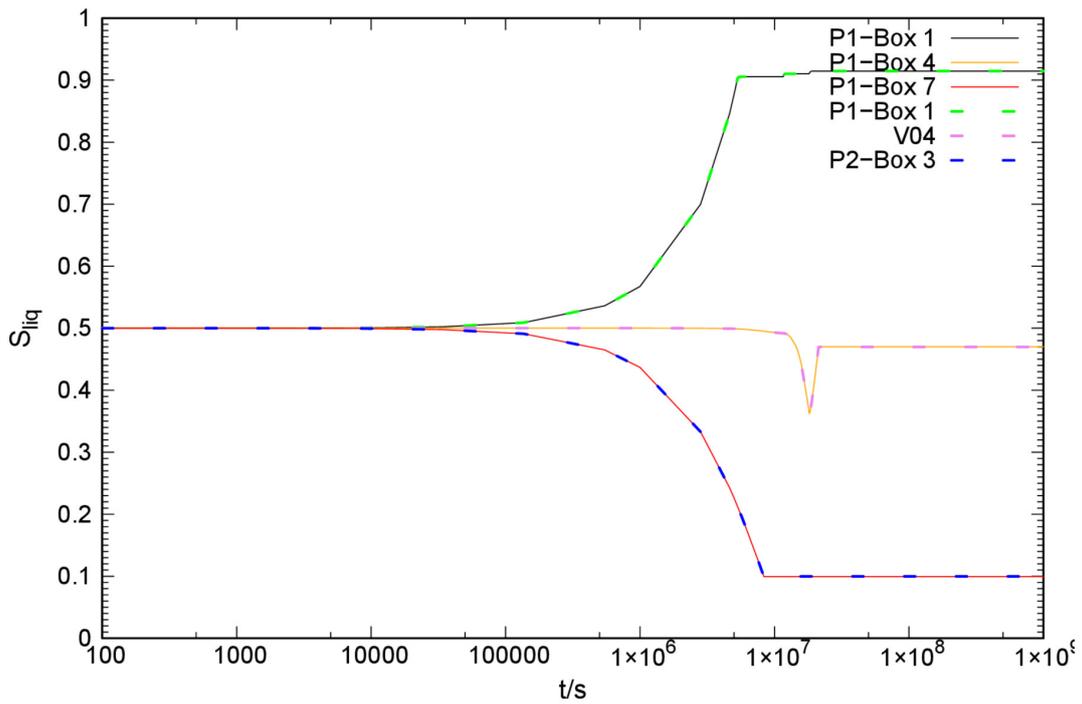
**Abb. 2.18** Massenfluss der flüssigen Phase über die Boxränder für den Testfall III-1; Die senkrechten Linien markieren den Übergang von Gaseindringen zu zweiphasigem Transport



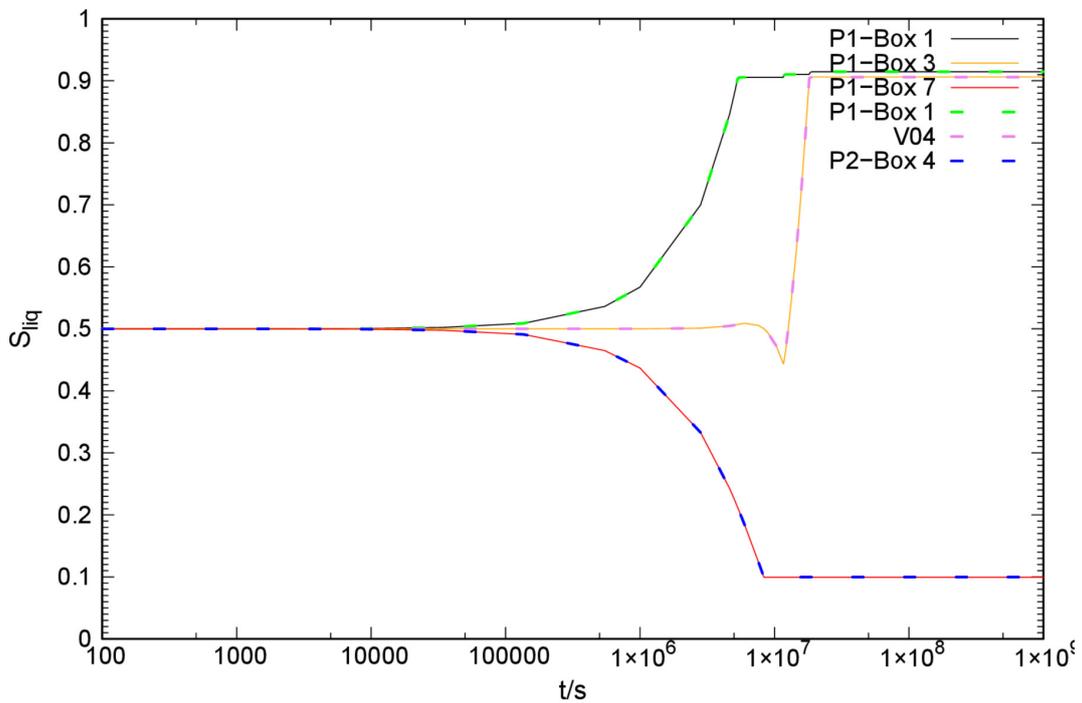
**Abb. 2.19** Massenfluss der gasförmigen Phase über die Boxränder für den Testfall III-1; Die senkrechten Linien markieren den Übergang von Gaseindringen zu zweiphasigem Transport

Die zugehörigen Ströme der Phasen sind in den Abb. 2.18 und Abb. 2.19 gezeigt und passen zum beschriebenen Systemverhalten. So ist beispielsweise der hohe schmale Peak im Gasfluss über den Rand P2-Rand 4 eine direkte Folge der schlagartigen Druckentlastung beim Übergang zum zweiphasigen Transport. Die Höhe des Peaks bestimmt sich durch den aufgestauten Druck über die komplette Zeit der Gasverdrängung.

Um die Wirkung des Gravitationsterms in der Gl. (2.3) zu überprüfen, wird die Entsättigung einer Säule im Testfall V-1 betrachtet. Hierzu werden zunächst die in Abb. 2.2 gezeigte räumliche Diskretisierung um 90 Grad gedreht, sodass eine vertikale Strömung stattfinden kann; hierbei werden die Boxen von unten nach oben durchnummeriert. Die Säule ist initial halb mit Flüssigkeit gesättigt und unten geschlossen (No-Flow- Randbedingung), die weiteren Initial- und Randbedingungen finden sich in Tab. 2.2. Wie erwartet, entsättigt sich der obere Teil der Säule durch die Wirkung der Gravitationskraft auf die Flüssigkeit und der untere Teil der Säule sättigt sich entsprechend auf; die mittlere Box der Säule bzw. das V04 sind hinterher halb gesättigt, wie in der Abb. 2.20 zu sehen. Die entsprechenden Umschaltprozesse haben korrekt stattgefunden.



**Abb. 2.20** Verlauf der Flüssigkeitssättigung für den Testfall V-1 (Entsättigung einer Säule)



**Abb. 2.21** Verlauf der Flüssigkeitssättigung für den Testfall Va-1 (Entsättigung einer Säule)

Um die Diskretisierung mit Bezug zur Wirkung der Gravitation zu überprüfen, wurde die Diskretisierung im Falle zweier P-Module so verändert, dass das obere P-Modul vier Boxen, das untere zwei Boxen hat. Wie zu erwarten und in der Abb. 2.21 zu sehen, wird dann das V04-Modul voll mit Flüssigkeit gefüllt.

## **2.8 Ausstehende Arbeiten**

Die Gasphase besteht in der momentanen Version von MARNIE2 zunächst ausschließlich aus Luft. Um auch Diffusionsprozesse in der Gasphase, sowie Lösung bzw. Ausgasung von Komponenten beschreiben zu können, muss die Gasphase in der Lage sein, analog zu den vorhandenen Möglichkeiten in der flüssigen Phase aus bestimmten Komponenten zusammengesetzt zu sein. Die Arbeiten hierzu befinden sich aktuell in der Umsetzung. Hierzu wurde zunächst der Eingabebereich für die Komponenten in der flüssigen Phase um die Vorgabe der Henrykonstanten für die Komponenten erweitert, so dass ein Übergang zwischen den Phasen möglich ist. Außerdem wurde ein neuer Eingabebereich implementiert, in dem es möglich ist, Komponenten mit ihren Spezifikationen vorzugeben, die ausschließlich in der Gasphase sein können.

In der originalen Version von MARNIE werden die einzelnen Komponenten über Differentialgleichungen zur Änderung ihrer Konzentrationen in der Flüssigkeit beschrieben. Die Änderung dieser Konzentrationen wird dann u. a. bestimmt durch Diffusion, radioaktiven Zerfall oder Sorption. Um dann von den Konzentrationen der einzelnen Komponenten auf die Gesamtmasse der Flüssigkeit zu schließen, wird dann die Dichte der Flüssigkeit berechnet. Dies führt in der Kopplung zu Chemieprogrammen wie ChemApp und Phreeqc zwangsläufig zu Massenbilanzfehlern, die bei hohen Konzentrationen das Rechenergebnis unzulässig verfälschen können. Deshalb ist in MARNIE2 geplant, die Differentialgleichungen für die Komponenten beider Phasen so umzuformulieren, dass die Änderungen der Massen bilanziert werden. Damit ist dann die zugrundeliegende Erhaltungsgröße direkt in den Differentialgleichungen verankert und es können keine inhärenten Bilanzfehler mehr auftreten.

Wenn die vorgenannten Arbeiten erfolgreich durchgeführt worden sind, kann MARNIE2 mit allen implementierten Einzelprozessen einer intensiven Verifizierung durch Vergleich mit passenden analytischen Lösungen sowie experimentellen Daten unterzogen werden. Hierbei sollte auch untersucht werden, inwieweit Laufzeitaspekte bei umfangreicheren räumlichen Diskretisierungen eine Rolle spielen. Die in der FEBE zur Verfügung stehenden Optimierungsoptionen sollten dann getestet werden und auf ihre Funktionalität hin reevaluiert werden. Parallel und in Ergänzung zu diesen Arbeiten wird sich weiterer Entwicklungsbedarf ergeben und es muss der Satz an Plausibilitätstestfällen so erweitert werden, dass als Ziel ein automatisierter Regressionstest steht, der bei jeder Änderung des Quellcodes und dessen Veröffentlichung im Repository ausgeführt wird. Dadurch kann bei Abweichungen sofort eine Rückmeldung an den betreffenden Entwickler gegeben werden. Hiermit ist dann ein hohes Maß an Qualitätssicherung für MARNIE2 erreicht, welches letztlich das Vertrauen in die Qualität der Simulationsergebnisse steigert und das Rechenprogramm z. B. für den Einsatz in Genehmigungsverfahren qualifiziert.

### **3 Weiterentwicklung des Codes TOUGH2-MP-GRS (AP 2)**

*Martin Navarro, Stephan Hotzel*

#### **3.1 Die Codes TOUGH2-GRS und TOUGH2-MP-GRS**

TOUGH2 ist ein Code zur Simulation zweiphasiger Strömungsvorgänge in porösen Medien, der vom *Lawrence Berkeley National Laboratory* entwickelt wurde /PRU 99/. Der Code simuliert die Strömung mehrerer Phasen anhand eines verallgemeinerten Darcy'schen Gesetzes. Durch die sogenannten EOS-Module (EOS: **E**quation-**o**f-**S**tate) werden die zur Verfügung stehenden Phasen, Phasenkomponenten und thermodynamischen Gleichgewichtsprozesse festgelegt.

Die GRS benutzt den Code TOUGH2 seit 1991 im Rahmen von Prozess- und Langzeitsicherheitsanalysen für tiefe geologische Endlager und hat den Code im Laufe der Jahre um verschiedene endlagerrelevante Prozesse erweitert (/JAV 00/, /JAV 06a/, /JAV 06b/, /JAV 06c/, /JAV 08/, /NAV 08/, /NAV 09/). Für eine langfristige Nutzung der Code-Varianten erfolgte in den BMU-Vorhaben 3609R03210 /NAV 13/ und UM13A03400 (ZIESEL) /NAV 16/ eine Integration der Entwicklungen in dem Code TOUGH2-GRS. Dieser baut auf der Version 2.0 des Codes TOUGH2 auf, die 1999 vom DOE's Energy Science and Technology Software Center herausgegeben wurde. TOUGH2-GRS beruht auf einer kompletten Neuimplementierung der GRS-eigenen Entwicklungen, wobei viele Prozessmodelle verbessert und erweitert wurden.

Maßgeblichen Einfluss auf die Konzeption und Implementierung der Prozessmodelle hatte der Einsatz des Codes im Rahmen des BMU-Vorhabens UM10A03200 (Vorläufige Sicherheitsanalyse Gorleben; VSG), bei dem die hydrodynamische Entwicklung eines hypothetischen Endlagers im Steinsalz am Standort Gorleben simuliert wurde. Der resultierende Code orientierte sich somit stark an den praktischen Erfordernissen der Langzeitsicherheitsanalyse für saline Wirtsgesteine und stellte einen Meilenstein für die Behandlung eines Endlagers im Salinar als zweiphasiges System dar. Im BMU-Vorhaben 4715E03230 wurde der Code TOUGH2-GRS um Prozesse für Endlager im Wirtsgestein Tonstein erweitert.

Im Rahmen des Vorhabens ZIESEL /KOC 16/, in welchem der Code TOUGH2-GRS zur Simulation zweiphasiger Strömungen in einem Endlager im Salinar angewandt wurde,

erfolgten umfangreiche Maßnahmen zur Qualitätssicherung des Codes. Diese umfassten die Erstellung eines QS-Handbuchs sowie die Durchführung umfangreicher Verifikationstests mit Hilfe des Codes SITA /SEH 16/. SITA ist ein von der GRS entwickelter Code zur automatisierten Durchführung regelmäßig zu wiederholender Code-Tests.

Die Modellierung von Transportprozessen im Nahfeld eines Endlagers im Salinar mit dem Code TOUGH2-GRS beschränkt sich in der Regel auf das Endlagerbergwerk und lässt das impermeable Wirtsgestein unbeachtet. Nahfeld-Transportrechnungen für Endlager in Tonstein müssen jedoch auch das Wirtsgestein umfassen, was zu Modellgittern mit hoher Elementanzahl führt. Im Vorhaben 4715E03230 wurde deshalb eine Portierung der GRS-eigenen Codeerweiterungen auf den parallelen Code TOUGH2-MP /ZHA 08/ vorgenommen. Der resultierende Code wurde TOUGH2-MP-GRS genannt /NAV 18b/.

Ziel des vorliegenden Vorhabens ist es, die Anwendbarkeit, Wartbarkeit und Robustheit der Codes TOUGH2-GRS und TOUGH2-MP-GRS weiter zu verbessern. Dies soll die Effizienz und Anwendungssicherheit der Codes und ihre langfristige Nutzbarkeit im Standortauswahlverfahren sichern. Darüber hinaus sollen eine Restartfunktion, Plausibilitätskontrollen für Eingabedaten und verbesserte Fehlermanagementroutinen entwickelt werden. Durch Erstellung von Verifikationstests soll die Qualitätssicherung der Codes TOUGH2-GRS und TOUGH2-MP-GRS weitergeführt werden.

### **3.2 Einführung einer Restartfunktionalität**

Die Codes TOUGH2-GRS und TOUGH2-MP-GRS wurden um eine Restart-Funktion erweitert, deren grundlegende Funktionsweise sich der Restart-Funktion von TOUGH2 und TOUGH2-MP anpasst. Zu jedem Ausgabezeitpunkt werden mehrere Restart-Dateien erzeugt, die jeweils einen Teilzustand des Systems beschreiben. Liegen bei Simulationsbeginn solche Restartdateien vor, so werden die betreffenden Teilzustände eingelesen und wiederhergestellt.

Die Restart-Datei SAVE enthält Angaben zur Simulationszeit, zum Zeitschritt, zu den Porositäten und den Primärvariablen. Darüber hinaus erzeugen verschiedene Prozessmodule eigene Restart-Dateien mit den Namen SAVE\_<Modulname>. Um Restart-Dateien verwenden zu können, müssen sie umbenannt werden, indem der Namensbestandteil „SAVE“ durch „INCON“ ersetzt wird. Die Dateien müssen sich im

Simulationsverzeichnis befinden. Ein Restart wurde für die Module RN, FISS, PRLIM, CORFL, COMP, CORRO und DEGRA implementiert und mithilfe von Verifikationstests überprüft (siehe hierzu Kapitel 3.6.3). Die Module GCOMP, RELA, RANGE and CNTRL benötigen keine Restart-Dateien.

Tab. 3.1 zeigt die erzeugten Restart-Dateien und die Zustände, die sie beschreiben. Anwender können entscheiden, welche dieser Dateien sie nutzen.

**Tab. 3.1** Restart-Dateien und gespeicherte Zustände

Restart-Datei	Inhalt
SAVE	Porosität, Primärvariablen, Zeitschritt, Zeit (sec), Iteration
SAVE_COMP	Physikalisches Elementvolumen (m <sup>3</sup> ), Porosität (überschreibt die Porositäten aus SAVE)
SAVE_CORFL	Massendichte der erzeugten Lauge (kg/m <sup>3</sup> )
SAVE_CORRO	Korrosionsgrad, Behälterwassermasse (kg), erzeugtes Gas (mol)
SAVE_FISS	Rissporosität, Druckschwellwert (Pa)
SAVE_PRLIM	Entwichene Gasmasse (kg), Entwichenes Gasvolumen (m <sup>3</sup> )
SAVE_RN	Stoffmenge der Radionuklide (mol)

Wird die Datei `SAVE` unverändert verwendet, so beginnt die wiederaufgenommene Simulation mit der Zeit und dem Zeitschritt der vorangegangenen Simulation. Ein Rücksetzen der Zeit und des Zeitschrittes wird erzwungen, wenn die Zeile „+++“ am Ende der Datei `SAVE` durch Leerzeichen ersetzt wird. Eine Wiederherstellung des Zustandes für das Modul `DEGRA` erfordert lediglich eine Wiederherstellung der Zeitinformation und daher eine unveränderte `SAVE`-Datei.

Auch modulspezifische Restart-Dateien können verändert werden. Es können alle Datenzeilen gelöscht werden, die weder Parameternamen enthalten noch mit 0 beginnen. Zustandsvariablen werden in der Regel in folgender Form gespeichert:

```
<name of parameter>
<global element index> <value>
<global element index> <value>
...
0 0.0
<name of parameter>
<global element index> <value>
<global element index> <value>
...
0 0.0
```

Einige Zustandsvariablen sind jedoch einer Kombination aus Element und Quelle zugeordnet, weshalb sie eine abweichende Datenstruktur nutzen:

```
<name of parameter>
<global element index> <source> <value>
<global element index> <source> <value>
...
0 0 0.0
```

Module wie das RN-Modul speichern zweidimensionale Arrays, die abgewickelt und als Folge eindimensionaler Arrays abgespeichert werden:

```
<name of parameter>-1
...
<name of parameter>-2
...
```

Die Porositäten werden sowohl in der Datei SAVE als auch in der Datei SAVE\_COMP gespeichert. Grund ist, dass das Modul COMP die Porositäten auch dann wiederherstellen muss, wenn die Datei SAVE nicht verwendet wird. (Die Überschreibung von Porositäten erfolgt nun in folgender Reihenfolge: Block ROCKS → Block INCON → Datei SAVE → Datei INCON\_COMP.)

TOUGH2-GRS und TOUGH2-MP-GRS überprüfen nicht, ob Modifikationen von Restart-Dateien physikalisch plausibel oder konsistent sind. Um inkonsistente Zustände zu vermeiden, sollten folgende Daten vor einem Restart niemals verändert werden:

- Die Anzahl der Elemente
- Die Elementindizes (durch die Reihenfolge im ELEM-Block definiert)
- Die Anzahl und Reihenfolge der Quellen des Moduls CORRO
- Die Anzahl und Reihenfolge der Radionuklide des Moduls RN

Folgendes sollte der Nutzer nur ändern, wenn er sicher ist, dass er damit keine inkonsistenten Zustände erzeugt:

- Die Korrosionsmodelle des Moduls CORFL
- Die Rissöffnungsmodelle des Moduls FISS

Es ist stets möglich, Module bei einem Restart auszuschalten, indem man den zugehörigen Datenblock in der Eingabe weglässt. Die zugehörigen Restart-Dateien sind dann wirkungslos.

### **3.3 Verbesserung der Plausibilitätskontrollen für die Eingabe**

Zur Stärkung der Fähigkeit des Codes, Anwendungsfehler in der Eingabedatei zu erkennen und den Anwender auf Fehleingaben hinzuweisen, wurden in einem ersten Schritt die am häufigsten genutzten Eingabevariablen zusammengestellt und zu jeder Variable der numerisch bzw. physikalisch erlaubte bzw. sinnvolle Wertebereich tabelliert. Auch bekannte Wechselbeziehungen zwischen zwei Eingabevariablen wurden aufgenommen sowie aus der Anwendungserfahrung bereits bekannte häufige Eingabefehlerquellen berücksichtigt. Eine Überprüfung der erlaubten Variablenbereiche wurde in den Code eingearbeitet.

Abgesehen von der Komplexität, die durch die Wechselbeziehung zwischen verschiedenen Eingabedaten besteht, ist bei einzelnen Variablen eine Analyse nötig. Wird ein auffälliger Wert erkannt (z. B. Permeabilität=0), dann wird in diesen Fällen nicht nur eine der beiden Aktionen „Warnmeldung“ oder „Fehlermeldung und Programmabbruch“ ausgelöst. Vielmehr wird hier zusätzlich zu der Ausgabe der Warnmeldung geprüft, ob die

möglicherweise vom Anwender modelltechnisch gewollte Eigenschaft im Programmablauf modellmathematisch auch möglich ist – zum Beispiel, dass keine Division durch Null auftritt. Diese Kontrolle erfolgte durch Testrechnungen und Code-Analysen.

Insgesamt werden nun 18 Eingabevariablen zusätzlich überprüft. Bei 11 Variablen geht es primär oder ausschließlich um die Überprüfung, ob der physikalisch und numerisch erlaubte und sinnvolle Wertebereich eingehalten wird. Diese Variablen sind: DROK, PER, CWET, SPHT, COM, EXPAN, CDRY, GK, IRP, ICP, SIGX.

Bei weiteren 7 Variablen geht es primär um die (aufwändigere) Überprüfung einer sinnvollen Verwendung des Parameters im Kontext mit anderen Parametern. Dies umfasst einerseits wiederum den Wertebereich der Variable (nun im Zusammenspiel mit anderen Parametern), andererseits die Verwendung der Variable überhaupt bzw. die Art ihrer Wert-Zuweisung (z. B. zur Vermeidung einer versehentlichen Doppel-Zuweisung). Diese Variablen sind: POR, TORTX, XKD3/4, RP, CP, NB, MOP11.

In allen Fällen wird eine Warn- oder Fehlermeldung mit der Routine `ALARM_NOTE` ausgegeben: Fehlermeldungen (mit Rechenlauf-Abbruch) dann, wenn der numerisch erlaubte Wertebereich nicht eingehalten wird, sodass mit größter Wahrscheinlichkeit der Rechenlauf keine sinnvollen Ergebnisse produzieren würde; Warnmeldungen (ohne Rechenlauf-Abbruch) dann, wenn der physikalisch sinnvolle Wertebereich nicht eingehalten wird, sodass einerseits mit einiger Wahrscheinlichkeit von einem Versehen bei der Eingabe ausgegangen werden kann und somit ein nicht-intendiertes Ergebnis zu befürchten steht – andererseits ein Verlassen der Gültigkeitsbereiche der implementierten physikalischen Modelle und somit ein physikalisch unsinniges Ergebnis zu befürchten steht. Dabei sind die hier implementierten „physikalisch sinnvollen Wertebereiche“ weit gefasst, um What-If-Programm-Anwendungen (z. B. mit deutlich erhöhter Permeabilität) zuzulassen.

### **3.4 Verbesserung des Fehlermanagements**

TOUGH2-Codes lesen Eingabeblocke sowohl aus der Standardeingabe als auch aus Blockdateien, wobei Eingaben durch Blockdateien durch Eingaben aus der Standardeingabe überschrieben werden. Dieser Überschreibungsmechanismus kann in folgenden Fällen zu einer Fehlbedienung führen:

1. Man will einen Block nicht nutzen, übersieht aber die Existenz einer alten Blockdatei, welche diesen Block aktiviert und einliest.
2. Man möchte eine Blockdatei einlesen, übersieht aber, dass es einen entsprechenden Block in der Standardeingabe gibt, der die Daten der Blockdatei überschreibt.
3. Man ist sich über die Überschreibungsregeln nicht im Klaren. (Beispielsweise werden Anfangsbedingungen zuerst durch den PARAM-Block gesetzt, dann durch den INDOM-Block und später durch den INCON-Block überschrieben.)

Punkt 1 kann ohne weitere Interaktion mit dem Anwender nicht abgefangen werden. Um Punkt 2 zu begegnen, wurden Warnungen eingeführt, die anzeigen, ob Blockdateien durch die Standardeingabe überschrieben werden.

Punkt 3 wird jetzt in Bezug auf das Einlesen von Porositäten überprüft. Hier wird gewarnt, wenn die über den ROCKS-Eingabeblock eingelesenen Porositäten durch optionale Porositätsangaben eines INCON-Eingabeblocks überschrieben werden.

Darüber hinaus wurden einige Fehlermeldungen der Routinen EOS, SAT, COWAT, PP und INPUT umformuliert, um Fehlerursachen detaillierter zu erläutern.

### **3.5 Entwicklung von Schnittstellen**

Ziel dieser Arbeiten war die Einbindung der Codes TOUGH2-GRS und TOUGH2-MP-GRS im Rahmen der Gittererstellung und Visualisierung von Simulationsdaten, die im Arbeitspaket 3 entwickelt wurden. Zu diesem Zwecke wurden für den Code TOUGH2-MP-GRS Routinen zur Datenausgabe in einem binären HDF5-Format entwickelt, um das Volumen der Ausgabedaten zu reduzieren und die Datenextraktion zu beschleunigen. Aufgrund der verkürzten Laufzeit des Vorhabens konnten die Arbeiten nicht vollständig abgeschlossen werden.

### 3.6 Verifikationstests

Zur Qualitätssicherung der Codes TOUGH2-GRS und TOUGH2-MP-GRS wurden mit dem Code SITA /SEH 16/ Verifikationstests für folgende Funktionalitäten erstellt:

- Einstellung der Molmasse, Viskosität, inversen Henry-Konstanten und spezifischen Enthalpie für die Hauptgaskomponente durch das Modul GCOMP
- Einstellung einer Löslichkeitsgrenze für Radionuklide durch das Modul RN
- Restartfunktionen für die Module RN, FISS, PRLIM, CORFL, COMP, CORRO und DEGRA

Die Tests werden in den folgenden Unterkapiteln näher erläutert.

#### 3.6.1 Modul GCOMP: Änderung der Hauptgaskomponente

##### Spezifische Wärmekapazität bei konstantem Volumen ( $C_v$ )

In einem geschlossenen, gasgesättigten Gebiet mit einem initialen Gasdruck von 1 MPa wird die Temperatur durch eine Wärmequelle mit einer Leistung von 10 J/s im gesamten Gebiet von 10 °C auf ca. 200 °C erhöht. Wegen der homogenen Wärmeeinspeisung findet kein Wärmetransport statt. Um die Temperaturerhöhung nur im Gas zu erzeugen, wird die Wärmekapazität des Gesteins auf 0 gesetzt. Das Elementvolumen beträgt 0,25 m<sup>3</sup>, die Porosität 0,5.

Der Testfall gibt mit Hilfe des GCOMP-Moduls eine spezifische Wärmekapazität von  $C_v = 500$  J/kg/K vor. Zusätzlich wird die spezifische Wärmekapazität  $C_v$  durch

$$C_v(\vartheta) = \frac{Q t}{m (\vartheta - \vartheta_0)} \quad (3.1)$$

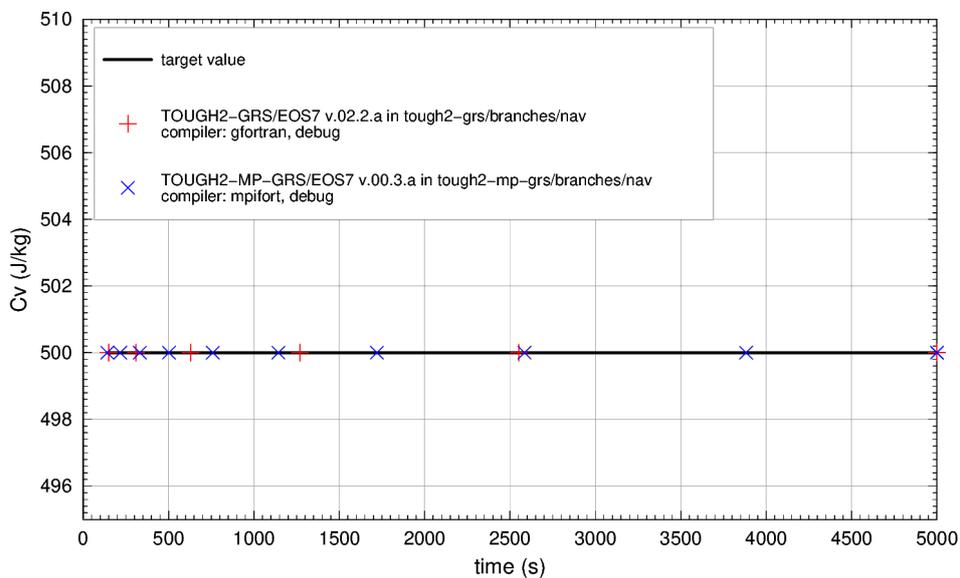
ermittelt, um sie anschließend mit dem Vorgabewert von GCOMP zu vergleichen.  $Q$  bezeichnet hier die Leistung der Wärmequelle von 10 J/s,  $t$  die Zeit,  $m$  die Gasmasse,  $\vartheta$  die aktuelle Temperatur und  $\vartheta_0$  die Starttemperatur von 10 °C. Die Variablen  $m$ ,  $t$  und  $\vartheta$  sind Ausgabeparameter der Simulation. Abb. 3.1 zeigt den Erfolg des Verifikationstests.

## Molmasse

In derselben Simulation wird die Einhaltung der von GCOMP eingestellten Molmasse von 0,01 kg/mol geprüft. Dazu wird die Molmasse mit Hilfe der implementierten universellen Gasgleichung

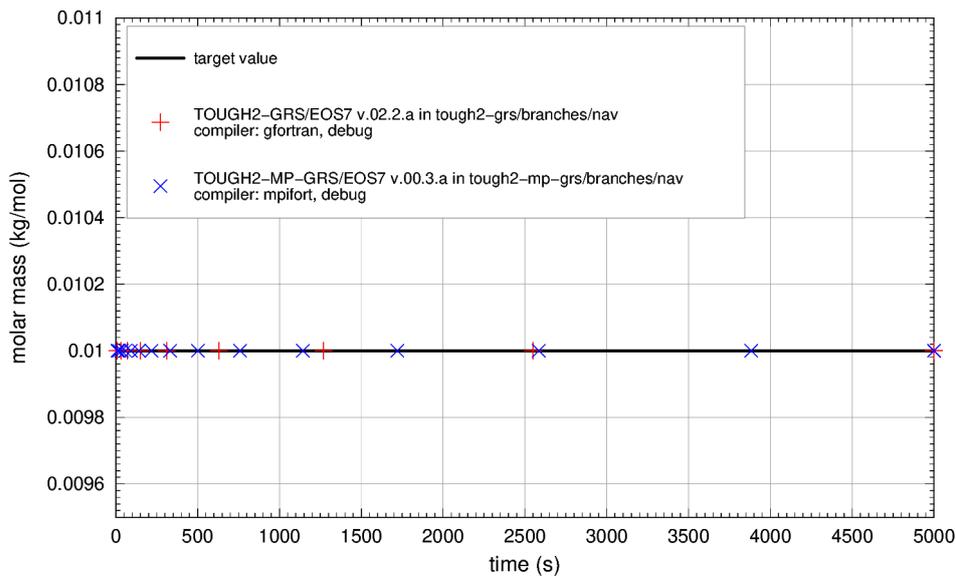
$$\mu = \frac{mRT}{pV} \quad (3.2)$$

ermittelt und mit dem Vorgabewert von 0,01 kg/mol verglichen.  $T$  bezeichnet hier die Temperatur in Kelvin,  $V$  das Porenvolumen von 0,125 m<sup>3</sup>,  $p$  den Gasdruck.  $T$  und  $p$  sind Simulationsergebnisse. Abb. 3.2 zeigt den Erfolg des Verifikationstests.



**Abb. 3.1** Verifikationstests für Modul GCOMP: Spezifische Wärmekapazität  $C_v$  bei konstantem Volumen

Rechenfall: gcomp-onlygas. Die simulierten Werten sollten zu allen Zeiten einen Wert von 500 J/kg/K haben.

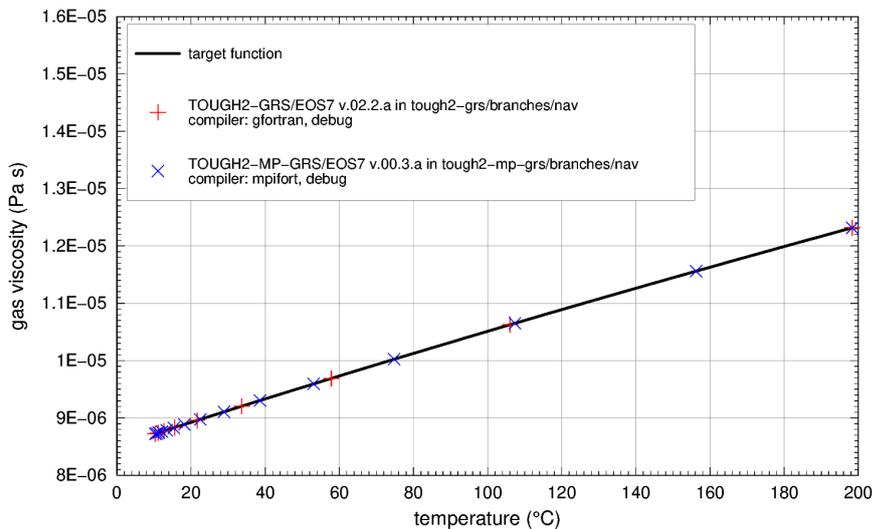


**Abb. 3.2** Verifikationstests für Modul GCOMP: Molare Masse

Rechenfall: gcomp-onlygas. Die simulierten Werten sollten zu allen Zeiten einen Wert von 0,01 kg/mol haben.

### Dynamische Gasviskosität

Wiederum in derselben Simulation wird die korrekte Implementierung der Viskositätsfunktion für ein H<sub>2</sub>-H<sub>2</sub>O-Gemisch übergeprüft, wobei nur der H<sub>2</sub>O-freie Zustand betrachtet wird. Die Gasviskosität ist im Wesentlichen temperaturabhängig und eine direkte Ausgabegröße der Simulation. Die zugrundeliegende, aus /LAN 69/ abgeleitete analytische Viskositätsfunktion ist in /NAV 18b/ dokumentiert. Abb. 3.3 zeigt den Erfolg des Verifikationstests.



**Abb. 3.3** Verifikationstests für Modul GCOMP: Dynamische Gasviskosität

Rechenfall: gcomp-onlygas. Die simulierten Werten sollten mit der Zielfunktion übereinstimmen.

### Inverse Henry-Konstante

Das GCOMP-Modul ermöglicht die Vorgabe einer temperaturabhängigen inversen Henry-Konstante für die Hauptgaskomponente. Zur Überprüfung dieser Funktionalität wird ein geschlossenes Gebiet mit einer Sättigung von 0,5 betrachtet. Aus der Verteilung der Gaskomponente auf die beiden Phasen wird eine inverse Henry-Konstante berechnet und mit der Vorgabefunktion verglichen.

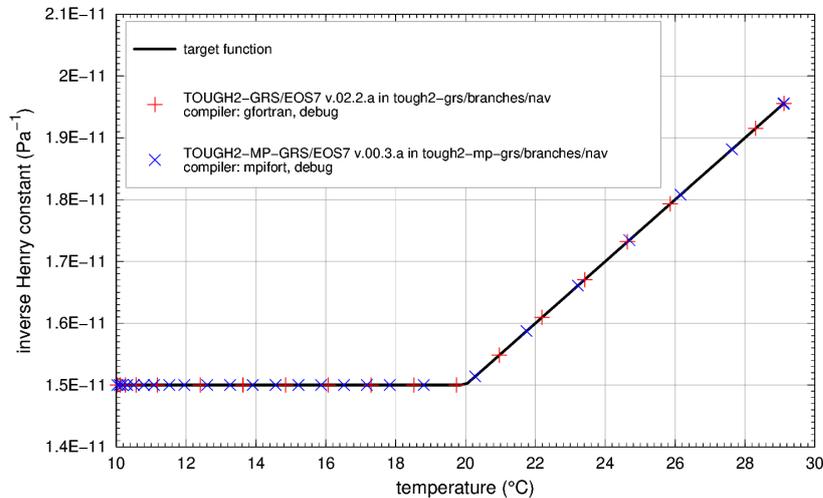
Die Temperatur wird durch eine Wärmequelle von 10 °C auf ca. 30 °C erhöht. Für die temperaturabhängige inverse Henry-Konstante wird die Funktion

$$H_{inv}(\vartheta) = \max (1,5 \cdot 10^{-11} \text{ Pa}^{-1}, 5 \cdot 10^{-12} \text{ Pa}^{-1} + 5 \cdot 10^{-13} \text{ Pa}^{-1} \text{ K}^{-1} \cdot \vartheta) \quad (3.3)$$

vorgegeben, wobei  $\vartheta$  die Temperatur in °C ist. Aus den Simulationsergebnissen wird die inverse Henry-Konstante durch

$$H_{inv}(\vartheta) = \max (1,5 \cdot 10^{-11} \text{ Pa}^{-1}, \frac{m_{\text{air}}^{\text{liq}} V \phi S_{\text{gas}}}{\left( \frac{m_{\text{air}}^{\text{liq}}}{\mu_{\text{air}}} + \frac{m_{\text{water}}^{\text{liq}}}{\mu_{\text{water}}} \right) m_{\text{air}}^{\text{gas}} R T}) \quad (3.4)$$

berechnet.  $\phi$  ist hier die Porosität und  $S_{gas}$  die Gassättigung.  $m_{\kappa}^{\beta}$  bezeichnet die Masse der Komponente  $\kappa$  in der Phase  $\beta$ .  $\mu_{\kappa}$  ist die Molmasse der Komponente. Abb. 3.4 zeigt die erzielte Übereinstimmung.

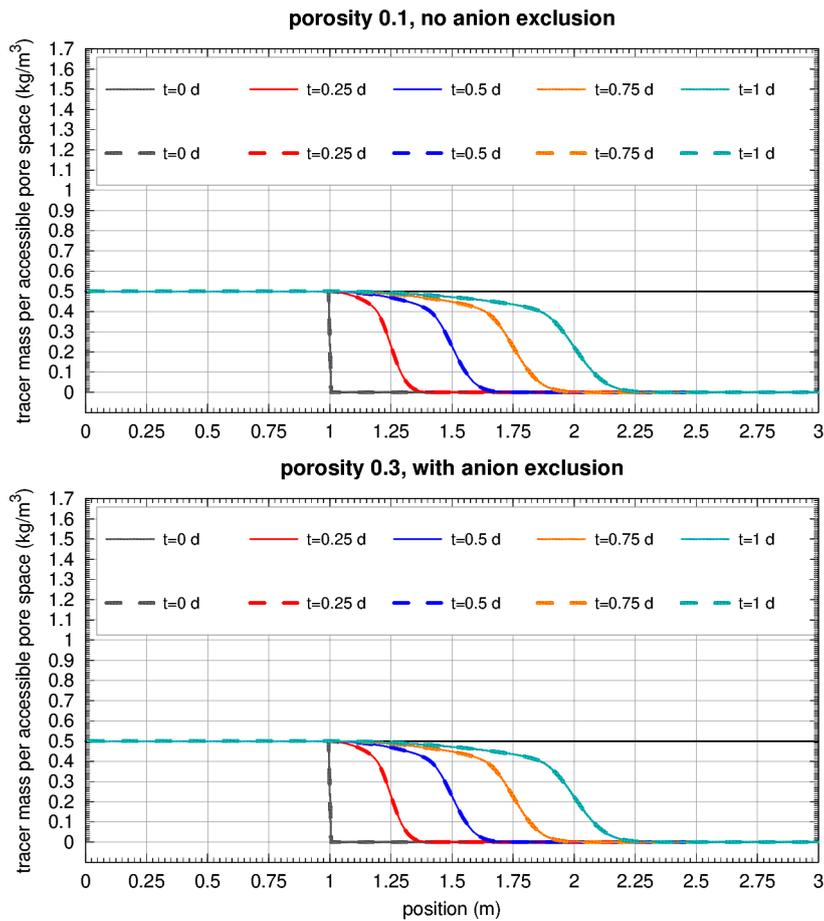


**Abb. 3.4** Verifikationstests für Modul GCOMP: Inverse Henry-Konstante

Rechenfall: gcomp-twophases.

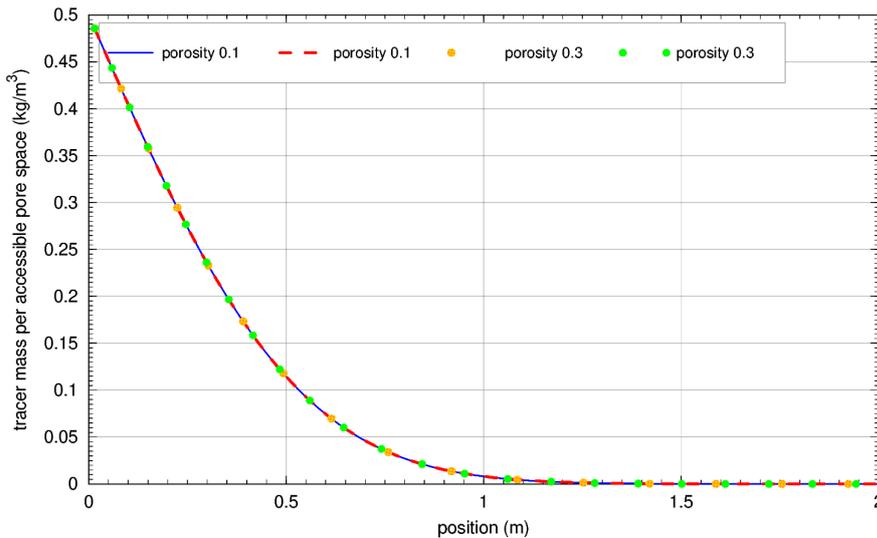
### 3.6.2 Modul RN: Löslichkeitsbegrenzung

Zwei der im Testhandbuch für die Codes TOUGH2-GRS und TOUGH2-MP-GRS /NAV 18c/ beschriebenen Testfälle für den Anionenausschluss wurden unter Vorgabe einer Löslichkeitsgrenze von 0,5 kg/m<sup>3</sup> wiederholt. (Zur Erläuterung der beiden Testfälle sei auf den genannten Bericht verwiesen.) Die berechneten Konzentrationen sollten daher den Konzentrationswert von 0,5 kg/m<sup>3</sup> nicht überschreiten. Die Abb. 3.5 und Abb. 3.6 zeigen, dass die errechneten maximalen Konzentrationen der Löslichkeitsgrenze entsprechen.



**Abb. 3.5** Verifikationstests zur Löslichkeitsbegrenzung des Moduls RN: Konzentration bei rein advektivem Transport, mit und ohne Anionenausschluss

Rechenfall: onlydiffusion-ceq05. Durchgezogene Linien: TOUGH2-GRS. Gestrichelte Linien: TOUGH2-MP-GRS. Der Rechenfall gleicht dem Rechenfall onlydiffusion /NAV 18c/ bis auf die Vorgabe einer Löslichkeitsgrenze von 0,5 kg/m<sup>3</sup>. Die Simulationen zeigen, dass keine Konzentrationen oberhalb der Löslichkeitsgrenze auftreten.

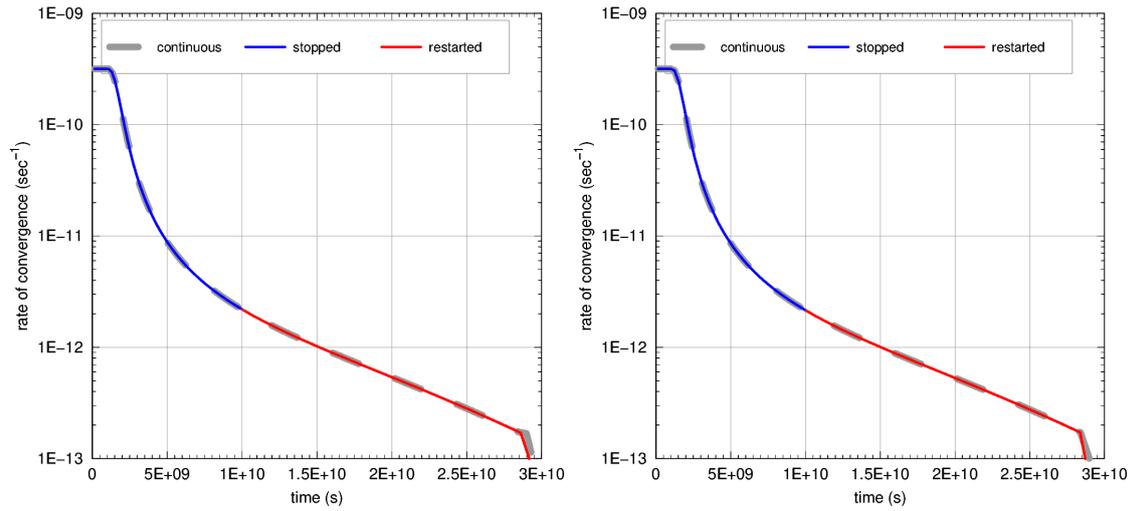


**Abb. 3.6** Verifikationstests zur Löslichkeitsbegrenzung des Moduls RN: Konzentration bei rein diffusivem Transport, mit und ohne Anionenausschluss

Rechenfall: onlyadvection-noflowreduction-ceq05. Der Rechenfall gleicht dem Rechenfall onlyadvection-noflowreduction /NAV 18c/ bis auf Vorgabe einer Löslichkeitsgrenze von  $0,5 \text{ kg/m}^3$ . Die Simulationen zeigen, dass keine Konzentrationen oberhalb dieser Löslichkeitsgrenze auftreten. Blaue Linie und orangefarbene Punkte: TOUGH2-GRS. Rote gestrichelte Linie und grüne Punkte: TOUGH2-MP-GRS. Linien bezeichnen Rechnungen ohne Anionenausschluss, Punkte Rechnungen mit Anionenausschluss.

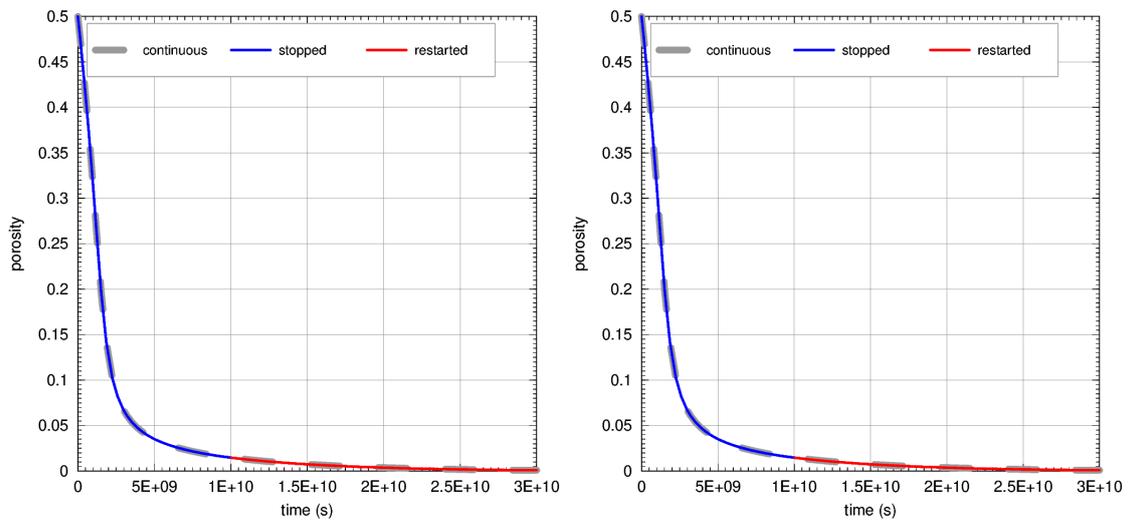
### 3.6.3 Restartfunktionen

Die implementierten Restartfunktionen wurden mit Hilfe existierender Testfälle verifiziert. Hierzu wurden die Testfälle zu festgelegten Simulationszeiten abgebrochen und mit Hilfe der Restart-Dateien wieder neu gestartet. Die Kurvenverläufe der abgebrochenen und wiederaufgenommenen Simulationen sollten sich aneinanderfügen und den Kurvenverläufen der Ursprungsrechenfälle folgen. Die abgebrochenen Simulationen sind in den nachfolgenden Abb. 3.7 bis Abb. 3.23 mit „stopped“ gekennzeichnet, die wiederaufgenommenen Simulationen mit „restarted“ und die Ursprungssimulationen mit „continuous“. Abweichungen im Bereich von Kurvenknicken ergeben sich aufgrund der automatischen Zeitschrittweitensteuerung, die zu einer unterschiedlichen Zeitdiskretisierung der kontinuierlichen und unterbrochenen Rechnungen führen. Zur Erläuterung der verwendeten Testfälle sei auf das Test-Handbuch für die Codes TOUGH2-GRS und TOUGH2-MP-GRS /NAV 18c/ verwiesen.



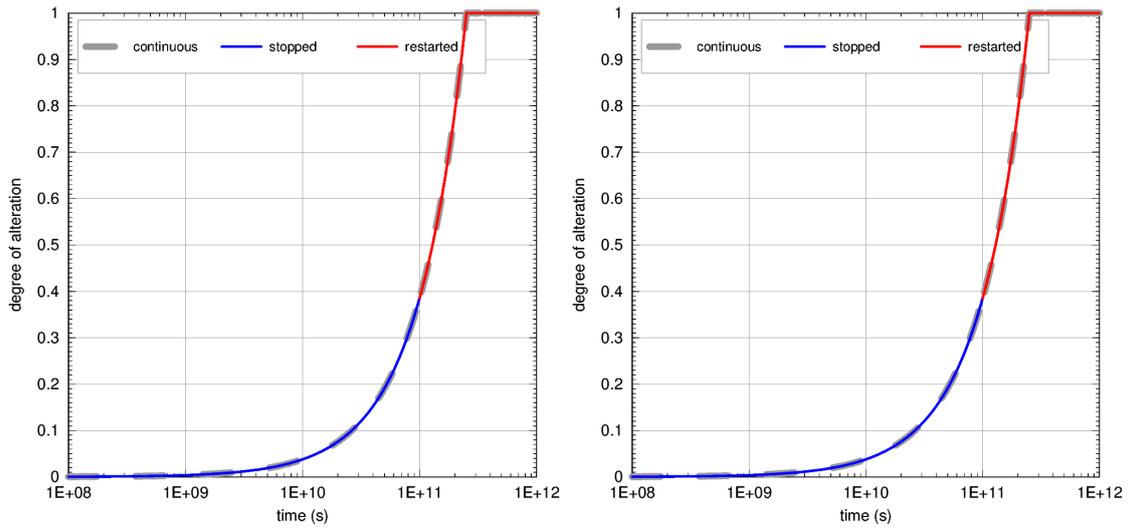
**Abb. 3.7** Verifikationstests des Restarts für Modul COMP: Konvergenzrate

Basisrechenfall: comp-1. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



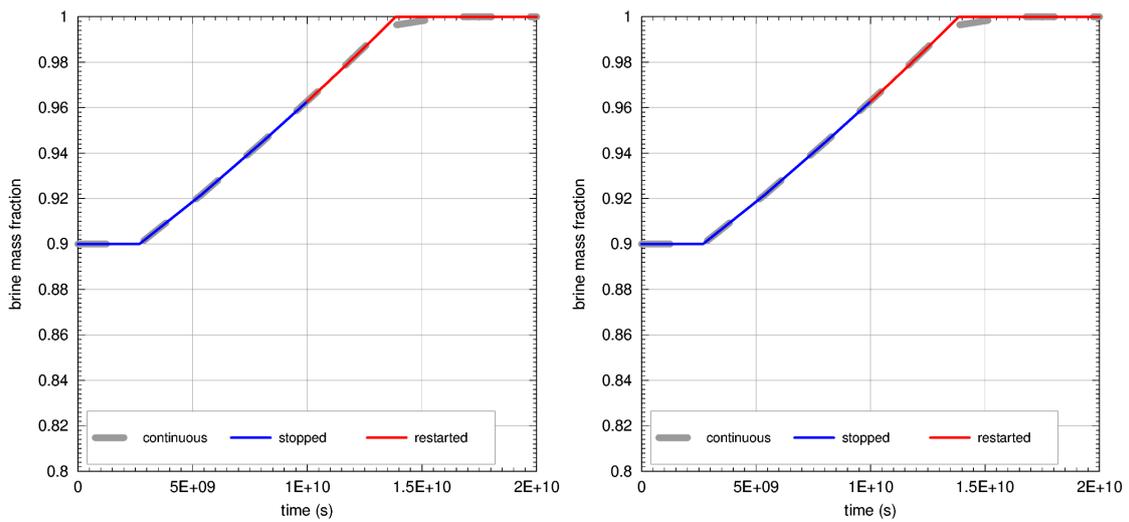
**Abb. 3.8** Verifikationstests des Restarts für Modul COMP: Konvergenzrate

Basisrechenfall: comp-1. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



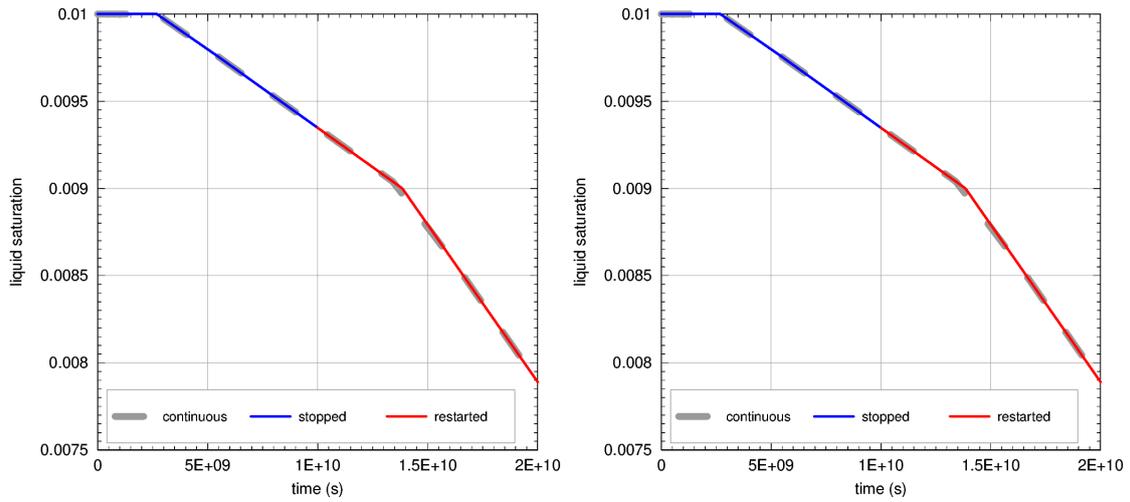
**Abb. 3.9** Verifikationstests des Restarts für Modul CORFL: Alterationsgrad

Basisrechenfall: corflsealperm. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



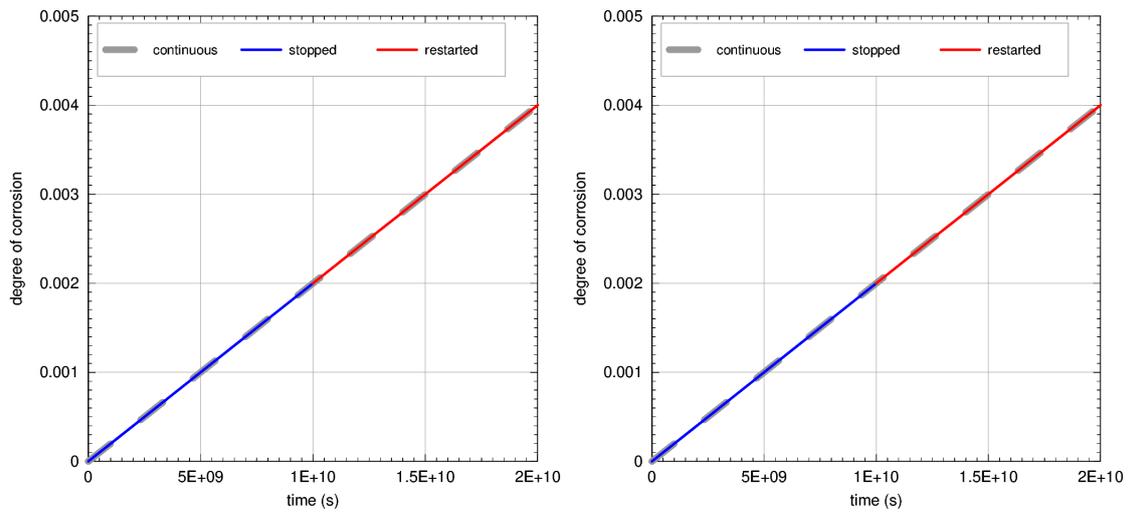
**Abb. 3.10** Verifikationstests des Restarts für Modul CORRO: Laugenanteil

Basisrechenfall: corro-referencecase. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



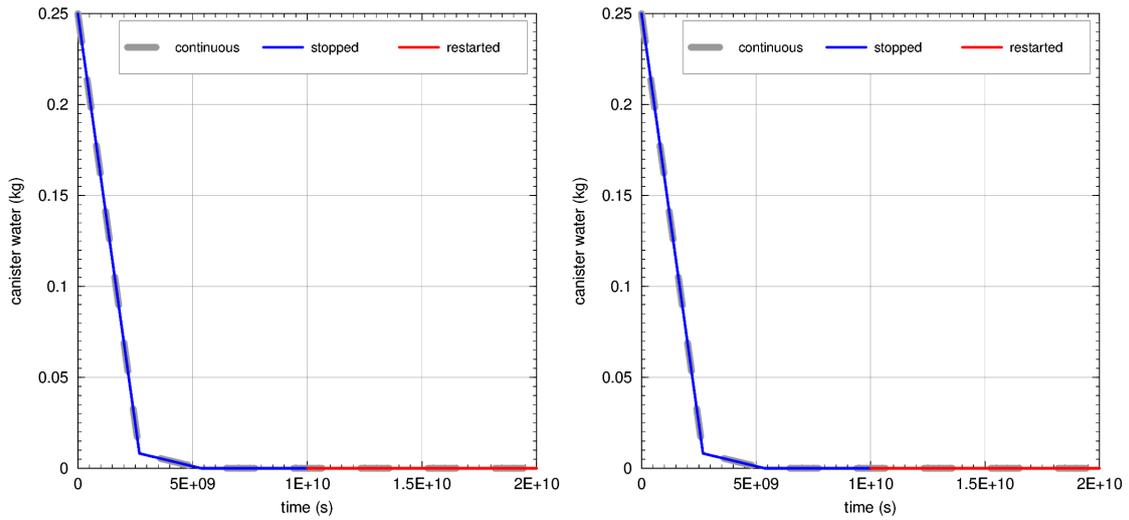
**Abb. 3.11** Verifikationstests des Restarts für Modul CORRO: Flüssigkeitssättigung

Basisrechenfall: corro-referencecase. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



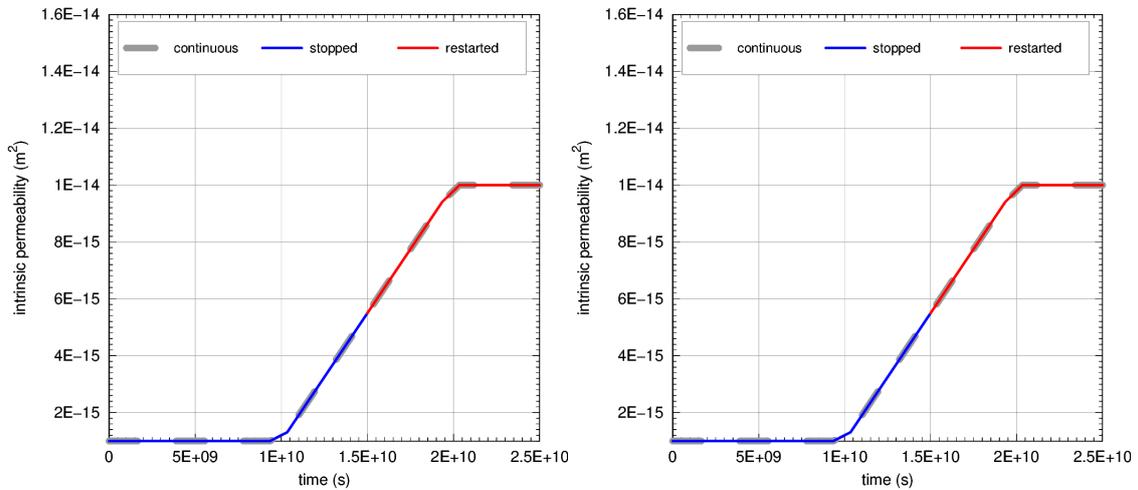
**Abb. 3.12** Verifikationstests des Restarts für Modul CORRO: Korrosionsgrad

Basisrechenfall: corro-referencecase. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



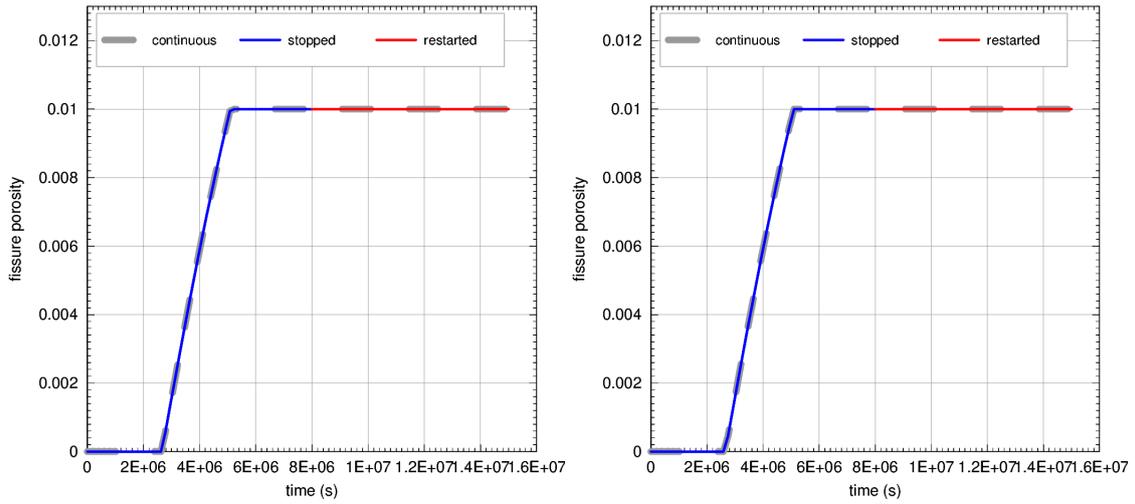
**Abb. 3.13** Verifikationstests des Restarts für Modul CORRO: Behälterwassermasse

Basisrechenfall: corro-referencecase. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



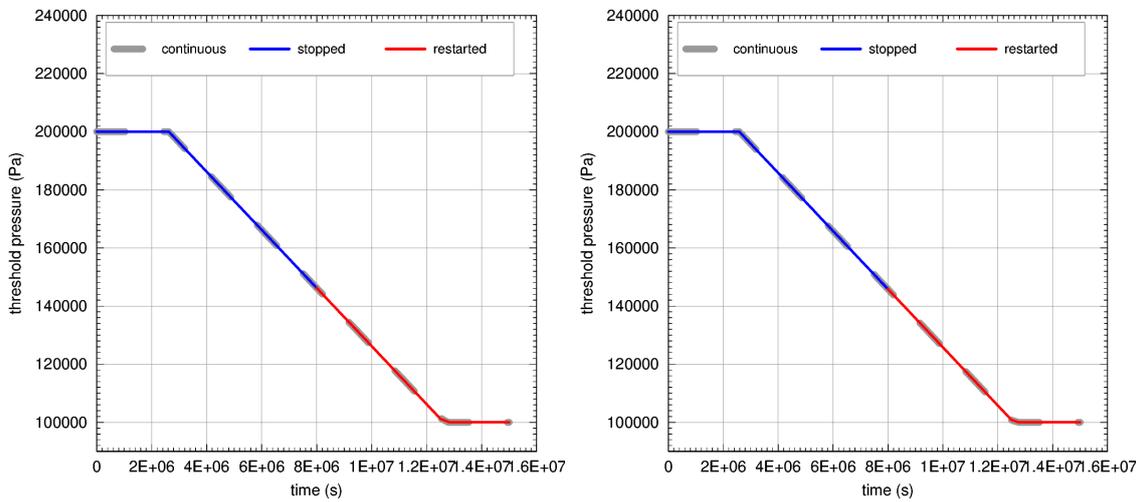
**Abb. 3.14** Verifikationstests des Restarts für Modul DEGRA: Intrinsische Permeabilität

Basisrechenfall: degra-referencecase. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



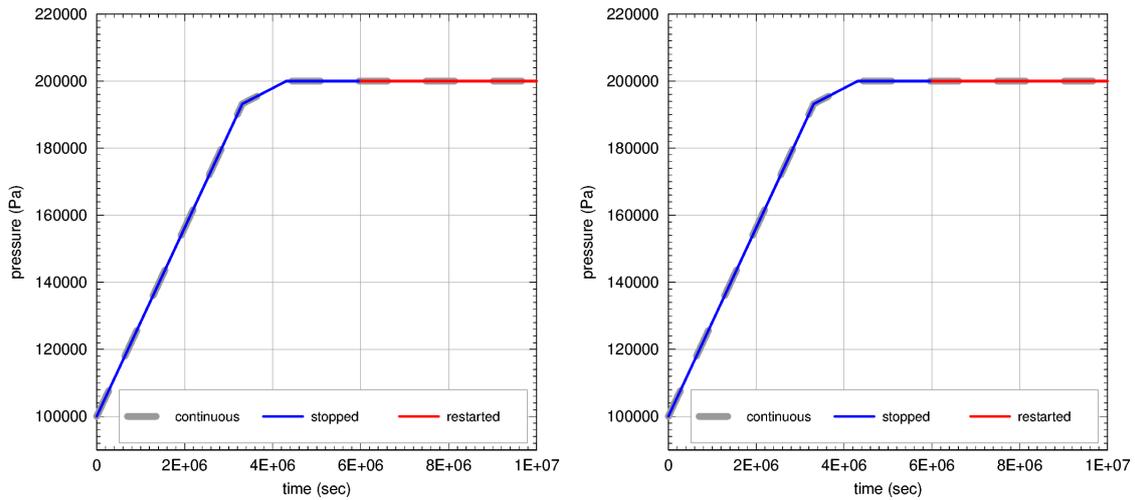
**Abb. 3.15** Verifikationstests des Restarts für Modul FISS: Rissporosität

Basisrechenfall: fiss-ifsoft1-ifpor2-ifper2-b. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



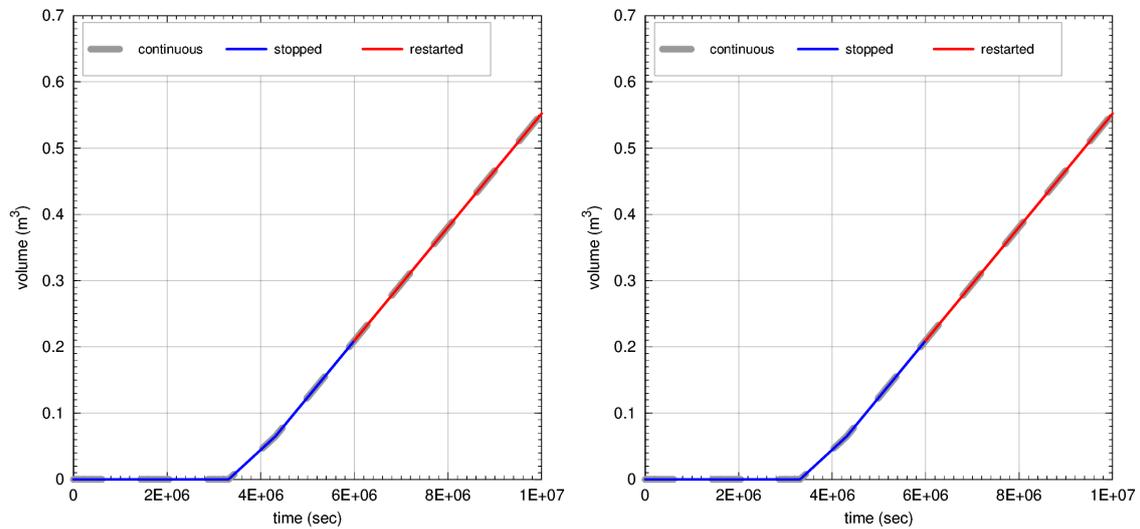
**Abb. 3.16** Verifikationstests des Restarts für Modul FISS: Druckschwellwert

Basisrechenfall: fiss-ifsoft1-ifpor2-ifper2-b. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



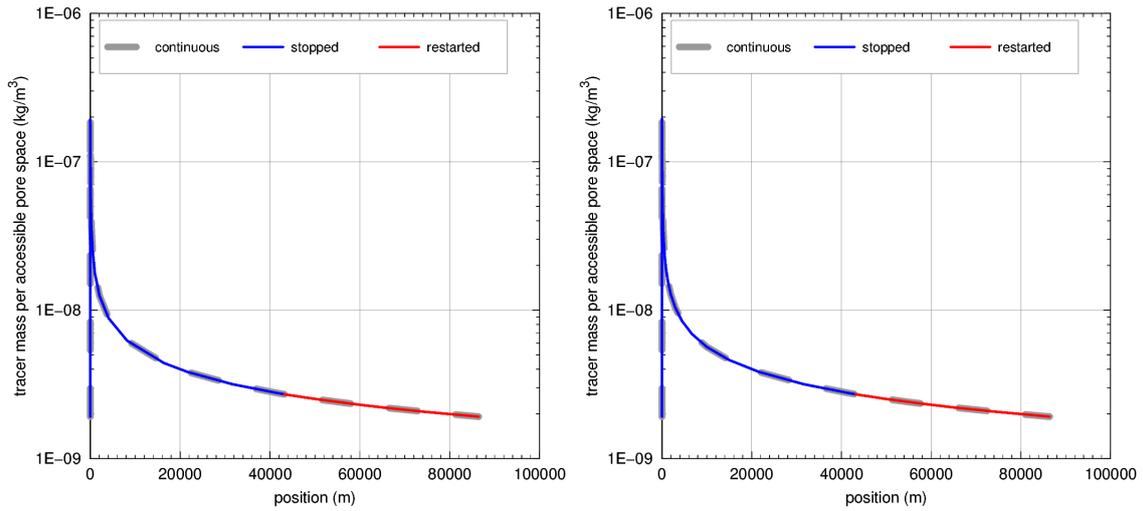
**Abb. 3.17** Verifikationstests des Restarts für Modul PRLIM: Gasdruck

Basisrechenfall: prlim. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



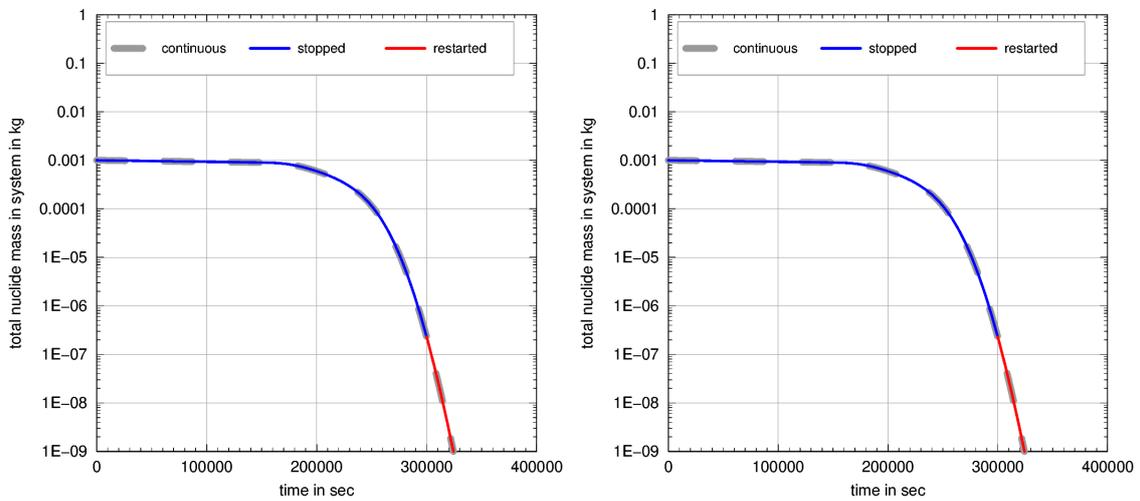
**Abb. 3.18** Verifikationstests des Restarts für Modul PRLIM: Normvolumen des entwickelten Gases

Basisrechenfall: prlim. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



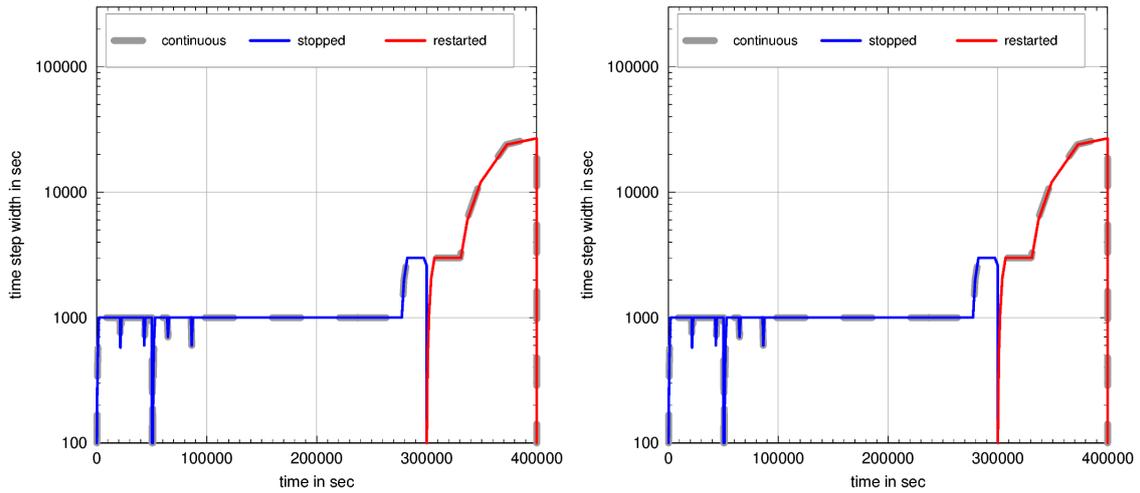
**Abb. 3.19** Verifikationstests des Restarts für Modul RN (Anionenausschluss): Tracerkonzentration im zugänglichen Porenraum

Basisrechenfall: onlydiffusion. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



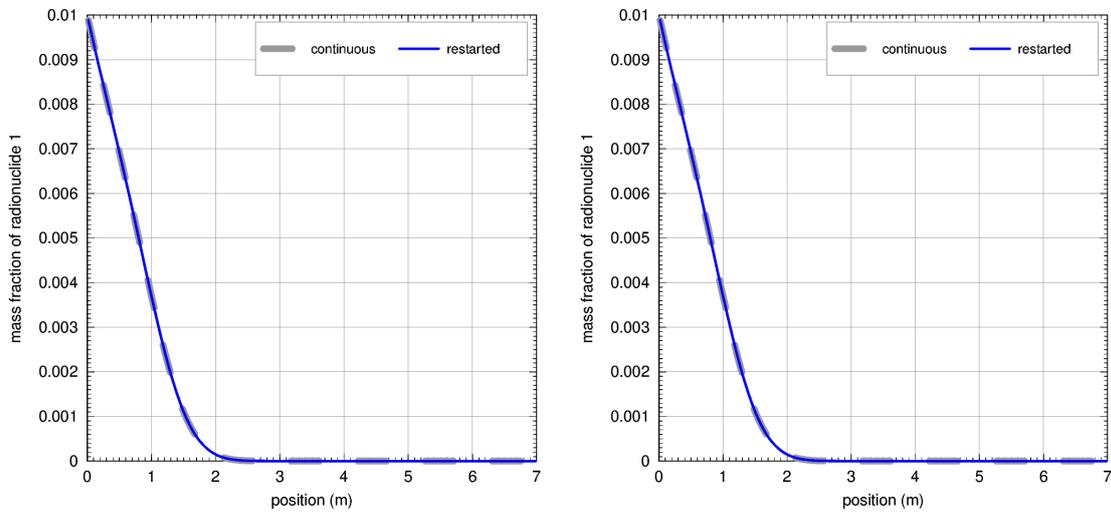
**Abb. 3.20** Verifikationstests des Restarts für Modul RN (Zeitschrittwertenregelung): Nuklidmasse im System

Basisrechenfall: rn\_dt\_limit. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



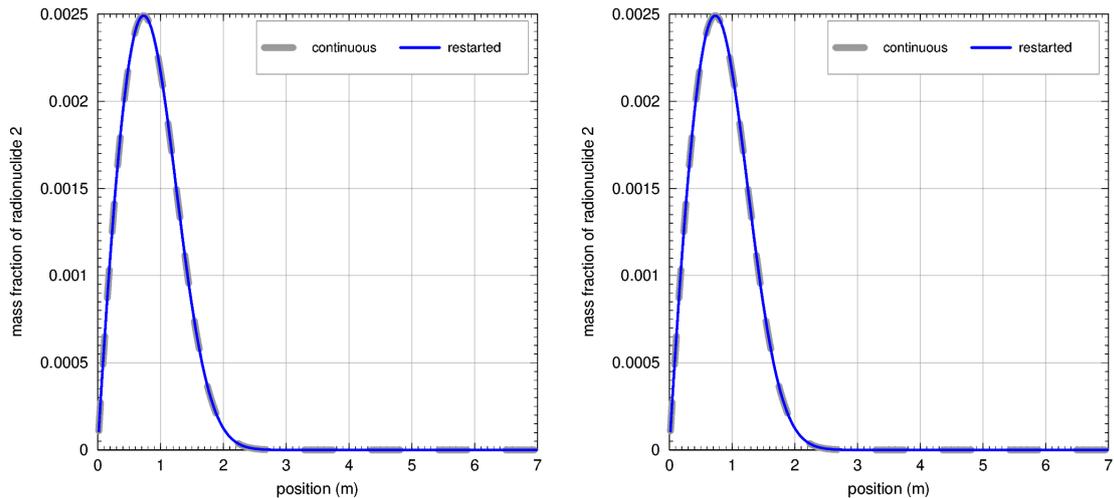
**Abb. 3.21** Verifikationstests des Restarts für Modul RN (Zeitschrittwitenregelung): Zeitschrittwitenentwicklung

Basisrechenfall: rn\_dt\_limit. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



**Abb. 3.22** Verifikationstests des Restarts für Modul RN (Radionuklidtransport): Massenanteil des ersten Radionuklids

Basisrechenfall: SAMR1\_lambda\_20d\_R\_2\_fine\_EOS7R\_RNmod. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.



**Abb. 3.23** Verifikationstests des Restarts für Modul RN (Radionuklidtransport): Massenanteil des ersten Radionuklids

Basisrechenfall: SAMR1\_lambda\_20d\_R\_2\_fine\_EOS7R\_RNmod. Links: TOUGH2-GRS. Rechts: TOUGH2-MP-GRS.

### 3.7 Konsolidierung

Der Quelltext der Codes TOUGH2-GRS und TOUGH2-MP-GRS wurden weiter konsolidiert, um Verständlichkeit, Wartbarkeit und Robustheit der Codes zu verbessern. Es wurden folgende Konsolidierungsmaßnahmen vorgenommen:

- Es erfolgte ein Funktionsabgleich der Codes TOUGH2-MP-GRS und TOUGH2-GRS. Im Codevergleich besaßen einige Module aufgrund ihrer Entwicklungshistorie eine unterschiedliche Struktur. Hier wurden die jeweiligen TOUGH2-GRS-Module durch vereinfachte Versionen der betreffenden TOUGH2-MP-GRS-Module ersetzt, wodurch eine Struktur- und Funktionsgleichheit der Module erreicht wurde.
- Es fanden verschiedene Variablenumbenennungen statt, um ein konsistentes Benennungsschema zu erreichen.
- TOUGH2-MP-GRS-Arrays, die für die Umrechnung zwischen lokalen und globalen Indices genutzt werden, wurden zur besseren Übersicht in ein gesondertes Modul (INDICES) ausgelagert.

- Im Ausgabemodul OUTPUT erfolgt die Ausgabe der Zeitreihen bzw. der Element- und Connectiondaten jetzt zentralisiert mit demselben Algorithmus, wodurch die Fehleranfälligkeit der Ausgaberroutinen reduziert wird.
- In den Codes PORO, MISC, TEST und OUTPUT wurden Codereликte entfernt und der Quelltext übersichtlicher geordnet.
- Die Kommentierungen im Quelltext wurden aktualisiert und erweitert, insbesondere in den Modulen PORO und OUTPUT.
- Die Berechnung des veränderten Gesteinsvolumens durch Porositätsänderung ( $V_{phys}$ ) wurde vollständig aus dem Kompaktionsmodul (COMP) ausgelagert und, thematisch korrekter, im Modul für Porositätsänderungen (PORO) untergebracht.
- Porositäts-Permeabilitäts-Beziehungen und porositätsabhängige Kapillardrücke wurden in ein neues Modul RELA für porositätsabhängige Prozesse ausgelagert.

Darüber hinaus wurden folgende Funktionsverbesserungen vorgenommen:

- In den EOS-Modulen EOS7 und EOS7R wurde die Genauigkeit der Dichtebe-  
rechnung für zweiphasige Zustände verbessert.
- Die Genauigkeit der Flüssigkeitssättigungsberechnung des Mikroriss-Moduls  
FISS wurde verbessert.

Im Rahmen der Verifikationsmaßnahmen wurden drei Codefehler in bislang nicht freigegebenen Funktionen entdeckt und behoben:

- RN-Modul: Bei einer Halbwertszeit von 0 wurde zuvor ein Defaultwert für die Half-  
times verwendet. Jetzt wird die Halbwertszeit auf 0 gesetzt.
- RN-Modul: Bei Annahme eines Anionenausschlusses wurde die Gleichgewichtskon-  
zentration ( $C_{eq}$ ) nicht korrekt eingestellt. Dies wurde behoben.
- GCOMP-Modul: Ein Initialisierungsfehler wurde behoben.

Die Fehlerbehebung wurde mithilfe der vorliegenden Verifikationstests erfolgreich überprüft.

### 3.8 Dokumentation

Die oben genannten Codeerweiterungen wurden im User Manual der beiden Codes /NAV 18b/ aufgenommen. Für die Codeentwicklung relevante Informationen wurden schwerpunktmäßig in Quellcodekommentaren dokumentiert.

Hinweise für Codeentwickler wurden in einem internen Entwicklerhandbuch (Programme Reference Manual) /NAV 20/

Martin Navarro: *Entwicklerhandbuch für TOUGH2-GRS und TOUGH2-MP-GRS*, Gesellschaft für Anlagen- und Reaktorsicherheit gGmbH (GRS), interner Bericht, 2020.

festgehalten.



## **4 Gittergenerierung und Visualisierung für den Code TOUGH2-MP-GRS (AP 3)**

*Martin Navarro, Ingo Kock, Gerd Frieling, Thomas Beuth*

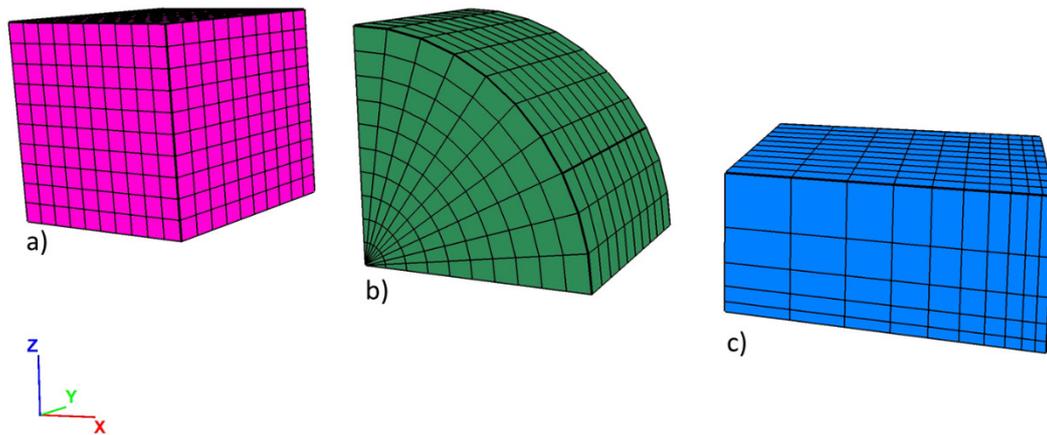
Simulationsrechnungen zum Transport von Radionukliden in Endlagersystemen nutzen zunehmend parallelisierte Rechencodes mit hoher Rechenleistung, was die Verwendung komplex strukturierter, hochauflösender Modellgitter ermöglicht. Die vorliegende Arbeit beschreibt die Entwicklung eines entsprechenden Gittergenerators für den parallelisierten Rechencode TOUGH2-MP-GRS, der in Kapitel 3.1 beschrieben wurde.

Kapitel 4.2 formuliert Anforderungen an den Gittergenerator. Die in Kapitel 4.3 dargelegten Rechercheergebnisse zu verfügbaren Gittergeneratoren zeigten, dass keiner der betrachteten Gittergeneratoren die gestellten Anforderungen erfüllt. Dies gab Anlass zur Entwicklung eines Gittergenerators im vorliegenden Vorhaben, wodurch frühere Entwicklungen im Vorhaben 4715E03230 /NAV 18a/ fortgeführt werden. Konzept und Ausführung dieses Gittergenerators werden in den Kapiteln 4.5 bis 4.13 vorgestellt.

Kapitel 4.14 zeigt auf, wie aus geologischen Daten aus Geoinformationssystemen Eingabedaten für den Gittergenerator erzeugt werden können. Der diesbezügliche Arbeitsablauf wurde anhand geologischer Datensätze des Landesamtes für Bergbau, Energie und Geologie (LBEG) getestet. Kapitel 4.15 beschäftigt sich mit dem Postprocessing, also der Visualisierung des Gitters und der Simulationsdaten.

### **4.1 Voronoi-Gitter**

TOUGH2-Gitter müssen Voronoi-Gitter sein, was bedeutet, dass die Verbindung zwischen zwei benachbarten Elementzentren stets senkrecht auf ihrer gemeinsamen Grenzfläche liegen muss. Abb. 4.1 zeigt Beispiele für Gitter, die zwar Voronoi-Gitter sind, aber wegen ihrer Strukturierung für die Darstellung komplexer Geometrien nur bedingt geeignet sind.



**Abb. 4.1** Beispiele dreidimensionaler strukturierter Voronoi-Gitter: a) kartesisches Gitter b) Gitter mit Radialsymmetrie c) orthogonales Gitter (aus /NAV 18a/)

Den Aufbau unstrukturierter Voronoi-Gitter lässt sich durch das Wachstum der Samen in einer Sonnenblumenblüte erläutern. Die Samen, die mit gleicher Geschwindigkeit wachsen, stoßen irgendwann aneinander und bilden auf diese Weise Grenzflächen aus, die der oben genannten Bedingung genügen (die Verbindung zwischen benachbarten Samenzentren liegt senkrecht auf der gemeinsamen Grenzfläche). Die Zentren von Sphären, durch deren Wachstum ein Voronoi-Gitter entsteht, werden üblicherweise – dem obigen Beispiel entsprechend – *Seeds* genannt. Die Seeds liegen in den Zentren der Voronoi-Zellen. Die Voronoi-Zellen können somit den Grenzen der TOUGH2-Elemente gleichgesetzt werden, während die Seeds deren Zentren entsprechen.

Viele der verfügbaren Generatoren für unstrukturierte Gitter sind für Finite-Elemente-Codes entworfen worden und erzeugen Gitter mit Tetraeder- oder Hexaederelementen, die keine Voronoi-Gitter darstellen. Es ist zwar möglich, aus solchen Gittern Voronoi-Gitter zu erzeugen, indem man die Gitterknoten als Seeds definiert, allerdings liegen dann die Seeds auch auf Materialgrenzen. Voronoi-Elemente, die aus solchen Seeds entstehen, werden daher von Materialgrenzen geschnitten. Es ist also bei der Erzeugung von Voronoi-Gittern wesentlich, Seeds so zu setzen, dass die Elementgrenzen zumindest näherungsweise mit Materialgrenzen zusammenfallen.

## 4.2 Anforderungen an einen Gittergenerator

An einen Gittergenerator für TOUGH2-MP-GRS wurden im Vorhaben folgende Anforderungen gestellt.

### Funktionsanforderungen

1. Es sollen **Voronoi-Gitter** erzeugt werden.
2. Es sollen **3-dimensionale Gitter** erzeugt werden.
3. **Sparsamkeit.** Das erzeugte Gitter soll sparsam sein im Sinne einer geringeren Anzahl von Elementen und Elementverbindungen (bei TOUGH2-Codes werden Elementpaare miteinander verbunden, um Strömung und Transport zu berechnen), insbesondere im Vergleich mit orthogonalen Gittern. Die Gittererstellung soll so sparsam sein, dass das erzeugte Gitter effizienter ist als ein fein diskretisiertes orthogonales Gitter.
4. **Konforme Abbildung von Materialgrenzen.** Materialgrenzen sollen möglichst lagegetreu abgebildet werden. Voronoi-Zellen dürfen nur in unbedeutendem Maße von Materialgrenzen geschnitten werden. (Die Anforderungen 3 und 4 erzeugen einen Zielkonflikt, auf den Anforderung 11 näher eingeht.)
5. **Abschätzbarkeit von Fehlern.** Wenn die erzeugten Gitter die Eigenschaften von Voronoi-Gittern nur näherungsweise erfüllen, muss der dadurch entstehende Fehler abschätzbar sein.
6. **Verwendbarkeit von GIS- und CAD-Daten.** Der Gittergenerator sollte Informationen aus Geoinformationssystemen (GIS) oder CAD-Systemen direkt oder indirekt nutzen können.
7. **Besondere geologische Strukturen.** Typische geologische Strukturen sollten darstellbar sein. Hierzu gehören: Schichten mit variabler Mächtigkeit und Neigung; dünne Schichten, die nur aus einer Elementreihe bestehen oder deren Mächtigkeit aus modelltechnischen Gründen vergrößert werden muss; auskeilende Schichten; Störungen mit permeabler Füllung (fault gouge); Störungen ohne Füllung, die lediglich Materialversätze erzeugen.
8. **Besondere technische Strukturen.** Endlagertypische technische und geotechnische Strukturen sollen abbildbar sein. Hierzu gehören Strecken, Abdichtbauwerke und Firstspalten.
9. **Vorlagen für spezielle Strukturen.** Der Gittergenerator soll Vorlagen für spezielle Strukturen anbieten, wie etwa die Einführung eines Materialgebietes, das nur aus nur einem einzigen Element besteht.

## Anforderungen an Nutzersteuerung

10. **Flexibilität.** Der Gittergenerator soll zur Erstellung problemorientierter Gitter eine hohe Flexibilität besitzen. Gitterauflösung und Anzahl der Elemente hängen von den zu simulierenden Prozessen und den auftretenden Materialkontrasten ab und sollten daher beeinflussbar sein. Auch können unterschiedliche Anforderungen an die Anisotropie oder Regelmäßigkeit der Gitter gestellt werden. Der Gittergenerator sollte in dieser Hinsicht flexibel und erweiterbar sein.
11. **Möglichkeit einer Abwägung zwischen geometrischer Vereinfachung und Genauigkeit.** Wegen begrenzter Rechenressourcen ist die Anzahl der Gitterelemente oft ein limitierender Faktor. Kleinräumige geometrische Strukturen führen oft zu einer starken Erhöhung der Elementanzahl. Der Gittergenerator muss daher nutzergesteuerte geometrische Vereinfachungen ermöglichen, um die Elementanzahl in einem praktikablen Rahmen halten zu können.
12. **Inaktive Elemente.** Für die TOUGH2-Codes müssen zur Definition von Dirichlet-Randbedingungen sogenannte *inaktive Elemente* /PRU 99/ erzeugt werden können. Eine flexible Lokalisierung der inaktiven Elemente ist erforderlich.

## Anforderungen zur Anwendbarkeit und Wartbarkeit

13. **Wartung bzw. Wartbarkeit.** Bei Codes Dritter sollte eine langfristige Wartung gewährleistet sein. Ansonsten sollte der Quellcode vorliegen und der Wartungsaufwand klein sein.
14. **Erweiterbarkeit.** Funktionalitäten, die in diesem Projekt nicht implementiert werden können, sollen auf einfache Weise zu einem späteren Zeitpunkt realisiert werden können.
15. **Anwenderfreundlichkeit.** Der Aufwand zur Erlernung der Bedienung soll gering sein. Die Bedienung soll einfach sein. Der Code sollte ausreichend dokumentiert sein.

## Lizenzrechtliche Anforderungen

16. Sofern bei einer Eigenentwicklung Programme oder Programmbibliotheken Dritter verwendet werden, die nicht bloß aufgerufen werden, sondern mit denen komplexe Datenstrukturen ausgetauscht werden, so dürfen diese keine GPL- oder AGPL-Lizenz besitzen.

### 4.3 Recherche zu Gittergeneratoren und Programmbibliotheken

Ob anforderungsgemäße Gittergeneratoren oder Programmbibliotheken zur Gittergenerierung verfügbar sind, wurde mit Hilfe einer Internetrecherche geklärt. Die Recherche setzte dabei die im Vorhaben 4715E03230 /NAV 18a/ begonnene Recherche zum gleichen Thema fort. Die Rechercheergebnisse sind in Tab. 4.1 zusammengestellt. Programmmerkmale, die nicht den gestellten Anforderungen genügen, wurden hier unterstrichen.

**Tab. 4.1** Kurzbeschreibung vorhandener Gittergeneratoren und Programmbibliotheken zur Gittererstellung

Software zur Gittergenerierung	Beschreibung
LaGriT V3.2 and V3.3 (Los Alamos Grid Toolbox) /LOS 11/	Bibliothek zur Erstellung 3-dimensionaler unstrukturierter Gitter für geologische und technische Geometrien. Eine Nachbearbeitung von Gittern ist möglich. Schwerpunkt liegt auf der Delaunay-Triangulierung. <u>Die Unterstützung von Voronoi-Gittern war zum Zeitpunkt der Recherche noch nicht vollständig entwickelt und kaum dokumentiert.</u> Ein Test war daher nicht möglich. Genutzte Bibliotheken: Exodus II 6, HDF5, NetCDF4.3, zlib-1.2.8. Open Source (BSD-3-Lizenz). Fortlaufende Entwicklung.
TOUGH2-GIS /BER 14/	Bash-Skripte, basierend auf GRASS GIS, zur Erzeugung unstrukturierter 3-dimensionaler Gitter, insbesondere für geologische Strukturen. <u>Keine reinen 3D-Voronoi-Gitter</u> (2D-Voronoi-Strukturen werden entlang der z-Achse repliziert). Open Source (GPL Version 2). Fortlaufende Entwicklung. <u>Dokumentation unzureichend.</u>
VORO2MESH /BON 17/	Gittergenerator zur Erzeugung unstrukturierter 3D-Gitter auf Basis der Bibliothek voro++. Mischung von orthogonalen und unregelmäßigen Elementen möglich. Open Source (GPL Version 2). Fortlaufende Entwicklung. <u>Dokumentation unzureichend.</u>
MeshVoro /FRE 14/	Gittergenerator für unstrukturierte 3-dimensionale Voronoi-Gitter für Codes der TOUGH2-Familie auf Basis der Bibliothek voro++. Fragen der Objektintegrität (Erhalt von Materialgrenzen) werden adressiert. <u>Keine Anwenderdokumentation. Verfügbarkeit und Lizenz unklar</u> (keine Projektseite). <u>Weiterentwicklung unklar.</u>
GRASS GIS /QGI 18/, /GRA 18/	Geoinformationssystem, mit dem <u>nur 2-dimensionale</u> Voronoi-Gitter erzeugt werden können. Open Source (GPL).
PetraSim /THU 15/	Erzeugung von unstrukturierten, 3-dimensionalen Gittern für den Code TOUGH2, insbesondere für geologische Strukturen. <u>Keine reinen 3D-Voronoi-Gitter</u> (2D-Voronoi-Strukturen werden entlang der z-Achse repliziert). Daraus resultierende Fehler sind vermutlich klein, wenn es sich um verhältnismäßig flach lagernde hydrogeologische Einheiten handelt. Dieses Verfahren erscheint <u>ungeeignet zur Abbildung komplizierterer räumlicher Strukturen</u> , wie zum Beispiel technischer Bauwerke. Kommerzielles Programm. <u>Quellcode nicht verfügbar.</u>
Voro++ pyvoro 1.3.2 /RYC 09/	Softwarebibliotheken zur Erstellung von Voronoi-Gittern für gegebener Seedpunkte. Das Problem des Erhalts von Materialgrenzen durch geeignete Seed-Positionierung wird nicht adressiert. Gute Dokumentation. <u>Keine Weiterentwicklung.</u> Open Source (modifizierte BSD-Lizenz). Letzter Release 2013. <u>Ob die Programme weitergepflegt werden, konnte nicht ermittelt werden.</u>

Software zur Gittergenerierung	Beschreibung
Scipy.spatial.Voronoi /SCI 18/	Python-Modul zur Erstellung von Voronoi-Gittern für gegebene Seedpunkte. Das Problem des Erhalts von Materialgrenzen durch geeignete Seed-Positionierung wird nicht adressiert.
CGAL (The Computational Geometry Algorithms Library) /CGA 20/	Umfangreiche Programmbibliothek für geometrische Anwendungen (auch Gittergenerierung und -bearbeitung). Es können <u>nur 2-dimensionale Voronoi-Gitter</u> erzeugt werden. Gute Dokumentation. Quelloffene und kommerzielle Teile (auch <u>GPL</u> ).
TetGen /SI 13/	Gittergenerator mit <u>Schwerpunkt auf der Delaunay-Triangulierung</u> . Wurde in GMSH integriert. <u>Fragen der Objektintegrität (Erhalt von Materialgrenzen)</u> werden nicht adressiert. Gute Dokumentation. Open Source ( <u>AGPL</u> ).
foamyHexMesh /GRE 19/	Gittergenerator für OpenFOAM, mit dem 3-dimensionale Voronoi-Gitter erzeugt werden können. Äußere Materialgrenzen werden approximiert. <u>Ob diese Funktionalität auch bei komplexeren inneren Materialgrenzen greift, ist unklar</u> . Open Source ( <u>GPL</u> ). <u>Wartung und Weiterentwicklung unklar</u> (letzter Release 2014).
VoroCrust v. 1.0 /ABD 20/	Gittergenerator, der aus CAD-Modellen 3-dimensionale unstrukturierte Voronoi-Gitter erstellt. <u>Strömungsmodellierung und geologische Systeme stehen nicht im Vordergrund</u> . <u>Fähigkeit zur Reduktion der Anzahl der Elementverbindungen nicht ersichtlich</u> . Kommerzielles Programm. Eine Testversion war nicht verfügbar.

Die Ergebnisse zeigen, dass bis auf die Programmbibliothek Scipy.spatial.Voronoi keiner der betrachteten Codes die gestellten Anforderungen zufriedenstellend erfüllte. Im vorliegenden Vorhaben wurde daher mit der Entwicklung eines Gittergenerators begonnen. Die Programmbibliothek Scipy.spatial.Voronoi wird dabei genutzt.

#### 4.4 Wahl der Vergitterungsstrategie

Auf Grundlage der formulierten Anforderungen kommen folgende grundsätzliche Strategien der Gittergenerierung in Frage.

##### 1. Überwiegend orthogonale Gitter mit Verfeinerungselementen

Mithilfe pyramiden- und keilförmiger Verbindungselemente können feine und grobe orthogonale Teilgitter so zusammengesetzt werden, dass sich ein konformes Voronoi-Gitter ergibt. Beispiele sind in /NAV 11/ zu finden.

*Vorteile:* Die Gittererstellung ist einfach und erzeugt ein exaktes Voronoi-Gitter.

*Nachteile:* Bei Strukturen, die nicht parallel zu den Hauptachsen liegen, ergeben sich bei geringer Auflösung Darstellungsprobleme.

## 2. Extrudierte unstrukturierte Voronoi-Gitter

Elemente eines 2-dimensionalen, unstrukturierten Voronoi-Gitters (meist in horizontaler Ebene) werden in der dritten Dimension wiederholt, indem das Element auf verschiedene Grenzflächen projiziert wird. Wenn diese Grenzflächen nicht senkrecht auf der Projektionsrichtung stehen, handelt es sich um kein exaktes Voronoi-Gitter. Das Verfahren eignet sich zur Darstellung leicht geneigter geologischer Schichtflächen.

*Vorteile:* Mit PetraSim liegt ein geeigneter Gittergenerator vor.

*Nachteile:* Kein exaktes Voronoi-Gitter. Die Zahl der abbildbaren Strukturen ist beschränkt. (Im Wesentlichen auf geologische Schichten mit geringer Neigung.)

## 3. Unstrukturierte Voronoi-Gitter mit hoher Anzahl von Blockelementen

Unstrukturierte Voronoi-Gitter können auch orthogonale Hexaeder (Blöcke) enthalten, mit denen für TOUGH2-MP-GRS die Anzahl der Elementverbindungen reduziert werden kann. Gittertypen dieser Art sind flexibel und können bei richtiger Auslegung sparsam sein.

*Vorteile:* Keine Einschränkungen bezüglich der darstellbaren Geometrien.

*Nachteile:* Geringe Sparsamkeit, falls nur wenig Blockelemente vorliegen (große Anzahl an Connections). Keine einfache Gittererstellung.

Die Einsatzgebiete der Gittertypen 1 und 2 sind eingeschränkt. Im Rahmen ihres Anwendungsbereiches besitzen diese Gittertypen jedoch deutliche Vorteile (Einfachheit, Sparsamkeit). Gittertyp 3 zeichnet sich durch eine hohe Flexibilität bei der Abbildung schiefwinkliger Strukturen aus. Gittertypen dieser Art werden in der Praxis häufig verwendet. Wegen der gestellten Flexibilitätsanforderung wurde mit der Entwicklung eines Gittergenerators des Typs 3 begonnen. Der Gittergenerator wurde in der Programmiersprache Python 3 entwickelt, um für eine gute Lesbarkeit und Wartbarkeit des Quellcodes zu sorgen.

## 4.5 Konzept der Gittergenerierung und -visualisierung

Das grundlegende Konzept der Gittergenerierung illustriert der nachfolgende Arbeitsablauf. Für die Gittergenerierung wird angenommen, dass geologische Daten aus Geoinformationssystemen vorliegen, und zwar als triangulierte Grenzflächen von Materialien bzw. Gebieten. Daten dieser Art weisen oft noch Gitterfehler auf, die zunächst bereinigt

werden. Danach erfolgt die Verbindung der Material- oder Gebietsgrenzen zu geschlossenen Körpern. Es ergeben sich damit folgende Anfangsschritte:

1. Bereitstellung eines triangulierten Flächengitters für Gebietsgrenzen und geologische Materialgrenzen
2. CAD-Nachbearbeitung: Bereinigung von Gitterfehlern
3. CAD-Nachbearbeitung: Erstellung vollständig geschlossener Körperoberflächen

Die zentrale Aufgabe der Gittergenerierung liegt in einer geeigneten Setzung von Seeds, aus denen später Voronoi-Zellen (die späteren TOUGH2-Gitterelemente) erzeugt werden. Das Seedsetzungsverfahren unterscheidet sich für Körpergrenzen (d. h. Material- oder Gebietsgrenzen) und das Körperinnere. Seeds für Körpergrenzen werden nachfolgend **Flächenseeds** genannt, Seeds innerhalb von Körpern **Körperseeds**. Beide Seedtypen werden weitgehend unabhängig voneinander gesetzt. Während die Position von Körperseeds verhältnismäßig frei gewählt werden kann, ist es die Aufgabe der Flächenseeds, Voronoi-Zellen zu erzeugen, die den Verlauf der Körpergrenzen nachzeichnen.

Flächenseeds werden unter anderem für jeden einzelnen Knoten (Triangulationspunkt) des Oberflächengitters erzeugt, sodass die Dichte der Oberflächenknoten die Anzahl der Flächenseeds steuert (und damit die Zahl der an den Körpergrenzen erzeugten Gitterelemente). Somit kann die Zahl der Flächenseeds über ein Resampling des Oberflächengitters reguliert werden.

Flächenseeds werden in **Knotenseeds** und **Schutzseeds** unterschieden. Knotenseeds werden für jeden Knoten des Oberflächengitters und die an ihm hängenden Dreiecksflächen generiert. Ihre Aufgabe ist es, Voronoi-Zellen zu erzeugen, die das Flächensystem des Knotens ausreichend genau approximieren. Knotenseeds sorgen also für eine Reproduktion des Oberflächengitters in der Nähe der Knoten. Zwischen Oberflächenknoten kann die Oberfläche nun immer noch durch naheliegende Seeds – die zum Beispiel benachbarten Flächen gehören – gestört werden. Hier sorgen Schutzseeds für eine Beseitigung der Störungen. Durch ein Resampling des Gitters sollte dafür gesorgt werden, dass die Zahl der Schutzseeds gegenüber der Zahl der Knotenseeds klein bleibt, da Schutzseeds zu einer Unregelmäßigkeit des Gitters führen und die Anzahl der Gitterelemente erhöhen.

Nach dem Resampling und der Erzeugung der Flächenseeds werden die Körperseeds innerhalb der Körper platziert. Körperseeds, die zu nahe an Oberflächen liegen und diese somit stören könnten, werden eliminiert. Körperseeds können in verschiedenen räumlichen Anordnungen produziert werden. Insbesondere ist eine orthogonale Anordnung möglich, die Hexaederelemente erzeugen und somit die Anzahl der Elementverbindungen klein halten. Der Anschluss der Elemente im Körperinneren an die oberflächennahen Elemente kann über optionale Optimierungsverfahren verbessert werden, die für eine geeignete Verschiebung der Körperseeds sorgen. Für jeden einzelnen Schritt auf dem Weg zum fertigen Gitter werden Daten zur Visualisierung mit ParaView ausgegeben. Diese Arbeiten lassen sich wie folgt zusammenfassen:

#### **4. Seedsetzungsverfahren**

- Resampling von Materialgrenzen zur Steuerung der Knotendichte und Vereinheitlichung der Dreiecksgrößen. Hierdurch wird die Zahl der Knotenseeds gesteuert und die Zahl der Schutzseeds verringert (optional).
- Generierung der Knotenseeds
- Generierung der Schutzseeds
- Generierung der Körperseeds
- Optimierung von Körperseeds zur Verbesserung des Anschlusses an die Flächenseeds

Aus den erzeugten Seeds kann nun ein Voronoi-Gitter erzeugt und als TOUGH2-Gitter exportiert werden. Zuvor können inaktive Elemente positioniert werden, die Dirichlet-Randbedingungen definieren. Beim Export des Gitters als TOUGH2-Gitter kann ein Toleranzwert für die Vernachlässigung kleinflächiger Elementverbindungen angegeben werden. Auf diese Weise wird die Zahl der Elementverbindungen, unter Kontrolle des damit verbundenen Fehlers, reduziert.

#### **5. Gittererzeugung**

- Erzeugung eines Voronoi-Gitters anhand der gesetzten Seeds
- Lokalisierung der inaktiven Elemente zur Einführung von Dirichlet-Randbedingungen (optional)
- Erzeugung der Gittereingabe für Codes der TOUGH2-Familie

Nach den Simulationsrechnungen folgt die Darstellung des Gitters und der Simulationsergebnisse mit ParaView.

6. Simulationsrechnungen
7. Visualisierung der Simulationsdaten im erzeugten Gitter mit ParaView

## 4.6 User Interface

Der Gittergenerator wird über die IPython-Konsole oder im Batch-Betrieb bedient. In der IPython-Konsole wird zunächst mit

```
from voronica import V
```

der Gittergenerator geladen. Die einzelnen Schritte der Gittergenerierung werden durch Befehle gesteuert, die beliebig wiederholt und kombiniert werden können, sofern sie auf den aktuellen Zustand des Gitters anwendbar sind (Abb. 4.2). Der aktuelle Zustand des Gitteraufbaus wird vor jedem Befehl in einer separaten Datei gespeichert, sodass die Gittererstellung im Falle eines Abbruchs an der gleichen Stelle wieder aufgenommen werden kann. In einer Log-Datei wird über die bisherigen Befehle buchgeführt. Die Befehle werden nachfolgend thematisch gegliedert beschrieben.

### Arbeitsfluss

```
V.new_project(<Projekttitel>)
```

beginnt eine neue Gitterkonstruktion. Jeder Befehl, der das Gitter verändert, speichert die aktuelle Gitterkonstruktion in einer separaten Datei. Vorhergehende Gitterkonstruktionen bleiben auf diese Weise erhalten. Jeder Gittergenerierungsbefehl wird zudem in einer Log-Datei mit dem Namen `<Projekttitel>.log` registriert.

```
V.continue_project(<Projekttitel>)
```

liest nach einem Abbruch der Gitterkonstruktion den letzten Zustand ein.

```
V.back()
```

Löscht die Datei mit dem aktuellen Gitter und kehrt zum vorherigen Gitter zurück. Zugleich wird der letzte in `<Projekttitel>.log` gespeicherte Befehl gelöscht.

V.batch(<Log-Datei>)

Führt alle Befehle der Datei <Log-Datei> aus. Dies führt allerdings nur dann zur aktuellen Gitterkonstruktion, wenn die Log-Datei zuvor nicht editiert wurde.

## Gittergenerierung

V.import(<Pfad/Dateiname>)

Importiert ein Oberflächengitter im Format `.inp`. (Ein weitverbreitetes textbasiertes CAD-Format, ursprünglich für die Software ABAQUS entwickelt.)

V.set\_material\_name(<Koordinate>, <Name>)

Der Gitterimport `V.import_grid()` stößt die Identifizierung von Materialgebieten an, für die zunächst nur interne Nummern vergeben werden. Nutzer können mit `V.set_material_name()` angeben, welcher Materialgebietsname das Gebiet erhält, das den Punkt <Koordinate> umschließt.

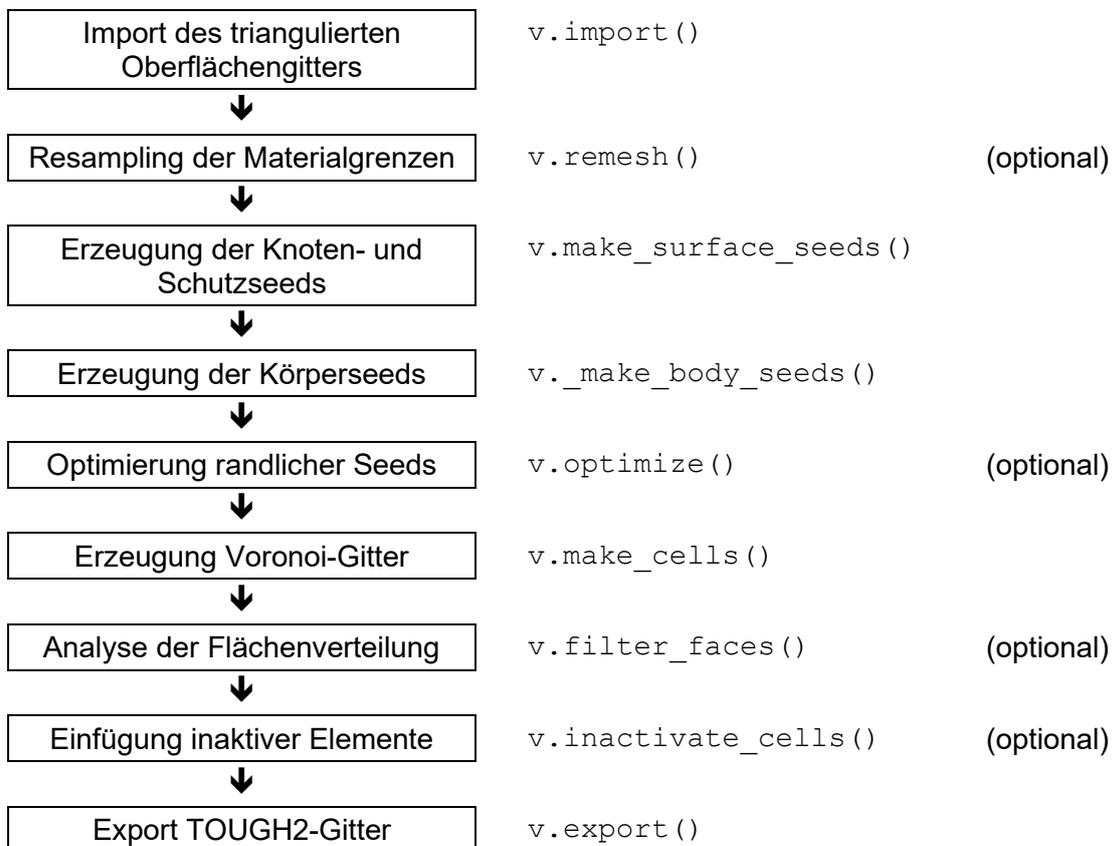


Abb. 4.2 Hauptschritte der Gittergenerierung

```
V.remesh(<Materialname 1>, <Materialname 2>, <Ziellänge>)
```

vergittert die Fläche zwischen den beiden Materialien mit den angegebenen Namen neu. Um die Materialien eindeutig zu identifizieren, empfiehlt sich eine vorherige Namensvergabe per `V.set_material_name`. Die Neuvergitterung betrifft angrenzende Kanten und strebt die Zielkantenlänge `<Ziellänge>` an.

Solange der Nutzer noch nach einer optimalen Ziellänge sucht, wird empfohlen, das alte Remeshing-Ergebnis mit `V.back()` zu löschen, anstatt auf dem aktuellen Remeshing-Ergebnis ein erneutes Remeshing durchzuführen.

```
V.make_surface_seeds(tolerance=<Toleranzwinkel>,  
relative_seed_distance=<Seedabstand>)
```

Erzeugt Flächenseeds für das Oberflächengitter (Knotenseeds und Schutzseeds). Vor diesem Befehl ist in der Regel ein Resampling mit `V.remesh` angezeigt. Danach sollte kein Resampling mehr durchgeführt werden. Der Toleranzwinkel bestimmt die Genauigkeit, mit welcher Körperflächen approximiert werden, und beeinflusst somit auch die Zahl der Körperseeds. Der Seedabstand bestimmt den relativen Abstand (Wertebereich  $0 < \text{RSD} < 0.5$ ) eines Knotenseeds zwischen zwei Knoten.

```
V.make_body_seeds(<Form>, <Ursprung>, <Ende X>, <Ende Y>,  
<Ende Z>, <Verteilung>, <Seeds>)
```

Erzeugt Körperseeds innerhalb des von den Knotenseeds definierten Körpers. Die Funktion kann beliebig oft aufgerufen werden. Die notwendigen Argumente haben die folgenden Bedeutungen:

- `<Form>`: Beschreibt die Körperseedverteilung. Hier sind einfache dreidimensionale Basisformen oder modellspezifische Verteilungen möglich. Werte, die `<Form>` annehmen kann sind als `string`: `cub` für eine quaderförmige Form, `rad` für Formen mit einem kreis- oder ellipsenförmigen Anteil (z. B. Ellipsoid, Zylinder, ...). Wenn `model` oder `material` für `<Form>` gewählt werden, wird das ganze Modell bzw. nur das gewählte Materialgebiet nach den folgenden Spezifikationen mit Körperseeds gefüllt.
- `<Ursprung>`: Die Ursprungsordinate (im Format einer Python-Liste `[x,y,z]`) des Seedmusters bildet die Basis eines rechtshändigen

Koordinatensystems. Wird für die `<Form> model` ignoriert und für die `<Form> material` als Liste von Material-ID interpretiert.

- `<Ende 1>, <Ende 2>, <Ende 3>`: Ausdehnung der zu erstellenden `<Form>` in allen drei Raumrichtungen in 3D-Koordinaten. Die Raumrichtungen können beliebig in Bezug zum `<Ursprung>` gewählt werden, müssen jedoch ein rechtshändiges Koordinatensystem ergeben. Wird für die `<Form> model` und für die `<Form> material` ignoriert.
- `<Verteilung>`: Abstand der Körperseeds zueinander. Folgende String-Werte sind möglich: `equi` für eine einfache äquidistante Verteilung in allen drei Raumrichtungen sowie `log` und `random.log` kann mit Kombinationen der Raumrichtungen 1, 2 und 3 (also z. B. `log1, log23`) weiter spezifiziert werden. Dabei sind `log` und `log123` gleichbedeutend. Eine logarithmische Verteilung in den spezifizierten Raumrichtungen heißt, dass mit zunehmender Entfernung vom `<Ursprung>` der Abstand der Seeds größer wird. In einer nicht spezifizierten Raumrichtung wird eine äquidistante Verteilung angenommen.
- `<Seeds>`: Anzahl der Seeds als ganzzahliger Tupel (`n_x, n_y, n_z`) in allen drei Raumrichtungen.

V. `make_cells()`

erzeugt das Voronoi-Gitter auf Basis der vorher erzeugten Seeds und Materialgebiete.

V. `faces(<Option>, <Prozentsatz>)`

Zeigt statistische Informationen über die Flächen und filtert kleine Flächen aus dem Gitter.

- `<Option>`: kann folgende Werte enthalten:
  - Der String `info` zeigt statistische Informationen der aktuellen Flächenverteilung im Gitter.
  - Der String `filter` führt zur Aussortierung kleiner Flächen bis zu einer Gesamtfläche, die in Prozent angegeben werden muss. Diese Angabe entspricht dem Eingabewert `<Prozentsatz>`, der eine Zahl zwischen 0 und 1 sein kann. Es werden statistische Informationen über die Flächen des Gitters vor und nach dem Aussortieren kleiner Flächen gezeigt.

V.inactivate\_cells(<Topologie>, <inaktive\_Elemente>)

definiert die inaktiven Elemente (also den Rand außerhalb des Modellraums) und ihre Beziehung zu den aktiven Elementen des Modellgebiets. Inaktive Elemente sind großvolumige ( $1E+80 \text{ m}^3$ ) Elemente, die Dirichlet-Randbedingungen einführen. Die optionalen Argumente haben die folgenden Bedeutungen:

- <Topologie>: spezifiziert die modellweite Topologie der inaktiven Randelemente. Standardeinstellung ist `open`, sodass ein einziges inaktives Element erzeugt wird. Jedes aktive Element am Rande des Modellgebiets hat zu diesem inaktiven Element entsprechend seiner Fläche einen „offenen“ Anschluss. Das Gegenteil ist die modellweite Topologie `closed`. Hier erzeugt man ein vollständig geschlossenes Modellgebiet. Bei beiden Varianten wird das 2. Argument <inaktive\_Elemente> nicht benötigt.

Weitere Einstellungen für das erste Argument sind `materials`, `elements`, `anywhere` oder `boundary`. Bei diesen Optionen ist die Nutzung des 2. Arguments als nähere Spezifizierung erforderlich. Bei `materials` erhalten die aktiven Elemente spezifischer Materialgebiete am Rand des Modellgebietes einen offenen Anschluss zum inaktiven Element. Bei `elements` kann der Status eines aktiven Elements auf inaktiv gesetzt werden. Durch die Option `anywhere` wird ein inaktives Element an beliebiger Stelle im Raum mit einer Verbindung zu einem beliebigen aktiven Element erzeugt. Die Option `boundary` erzeugt ein inaktives Element an beliebiger Stelle im Raum mit je einer Verbindung zu zahlreichen aktiven Elementen.

- <inaktive\_Elemente>: Diese Option ist nur erforderlich, falls für <Topologie> `materials`, `elements`, `anywhere` oder `boundary` gewählt wurde. Für `materials` ist die Angabe der Material-ID erforderlich, für `elements` die Angabe der Element-Id. Die Optionen `anywhere` oder `boundary` benötigen eine bzw. mehrere Koordinaten.

- V. export()

erzeugt die TOUGH2-Eingabedatei mit den notwendigen Blöcken „ELEM“ und „CONN“, die .xmf- und .hdf5-Dateien zum Einlesen des Gitters in Paraview sowie eine weitere Datei mit der Zuordnung der TOUGH2-Materialgebiete zu den Originalnamen der Materialgebiete.

Einige Befehle können erst nach anderen Befehlen ausgeführt werden. So kann zum Beispiel `V.remesh()` erst nach `V.import_grid()` genutzt werden. Der Nutzer erhält bei Eingabe von Befehlen, die noch nicht ausführbar sind, einen entsprechenden Hinweis sowie Tipps zu aktuell möglichen Befehlen.

Nach jedem Gitterkonstruktionsbefehl werden folgende Dateien aktualisiert. Dateien mit den Endungen `.xmf` und `.hdf5` dienen der Visualisierung mit ParaView.

`<Projektttitel>.log:`

Log-Datei für den Betrieb mit `V.batch()`

`<Projektttitel>-surfaces (.xmf und .hdf5):`

Triangulierte Oberflächen

`<Projektttitel>-all_seeds (.xmf und .hdf5):`

Alle Seeds

`<Projektttitel>-surface_seeds (.xmf und .hdf5):`

Knotennahe Flächenseeds

`<Projektttitel>-protective_seeds (.xmf und .hdf5):`

Knotenferne Flächenseeds

`<Projektttitel>-body_seeds (.xmf und .hdf5):`

Körperseeds

`<Projektttitel>-markers (.xmf und .hdf5):`

Punkte, an denen Probleme festgestellt wurden.

`<Projektttitel>-cells (.xmf und .hdf5):`

Voronoi-Zellen (Gitterelemente)

## **Ausgabe**

`V.export()`

Erzeugt in der Datei `<Projektttitel>-tough` die Gittereingabe für TOUGH.

## 4.7 Status der Implementierung

Die wesentlichen Funktionen des Gittergenerators konnten im Vorhaben fertiggestellt werden. Die Flächenseederzeugung erwies sich in Bezug auf die Algorithmen und das Resampling in Bezug auf die Datenstrukturverwaltung als komplex. Die betreffenden Entwicklungen waren daher zeitintensiver als ursprünglich geplant, sodass einige untergeordnete Funktionen nicht vollständig implementiert werden. Dies betrifft folgende Punkte.

- `V.optimize`: Grundalgorithmen zur Optimierung der Anbindung zwischen Knotenseeds und Flächenseeds wurden implementiert, jedoch noch nicht zur Anwendungsreife gebracht.
- `V.make_body_seeds`: Erweiterung der Palette an möglichen Körperseedverteilungen, dazu gehört beispielsweise das Einfügen von neuen Materialgebieten.
- Visualisierung der inaktiven Elemente: Die zusätzlichen Elemente mit sehr großem Volumen erscheinen bislang lediglich im exportierten TOUGH2-Gitter, nicht in der grafischen Paraview-Ausgabe.
- Permeable und nichtpermeable tektonische Störungen – also Störungen, die neben ihrem Versatz auch eine Transmissivität besitzen – müssen zurzeit noch als separate Körperoberflächen in den Eingabedaten ausgewiesen werden. Es ist geplant, Routinen zur Erkennung und sparsamen Diskretisierung solcher Störungen zu ergänzen.
- Probleme bei der Abbildung dünner Schichten wurden noch nicht ausreichend untersucht.
- Die Vereinfachungsalgorithmen für Oberflächen sollten in zukünftigen Vorhaben weiterentwickelt werden, da sie die Sparsamkeit der Gitter erheblich beeinflussen.
- Es konnten keine umfangreichen Funktions- und Performancetests mit komplexeren Gittern durchgeführt werden.

## 4.8 Neuvergitterung (Remeshing)

Implementiert wurde ein in /BOT 10/ skizzierter Algorithmus, der ein iteratives Aufspalten, Verschmelzen und Vertauschen von Dreieckskanten vornimmt und dabei eine vorgegebene Zielkantenlänge anstrebt. Bei jeder Iteration werden folgende Operationen durchgeführt:

1. Spaltung langer Kanten
2. Löschung kurzer Kanten
3. Vertauschung von Kanten, um die Abweichungen von den idealen Valenzzahlen (Anzahl der Kanten pro Knoten) zu minimieren
4. Verschiebung von Knoten, um spitze Winkel zu vermeiden
5. Projektion der neuen Knotenlagen auf die Ausgangsfläche

Die Algorithmen werden auf nutzerdefinierte Materialgrenzen angewandt.

Eine Hauptschwierigkeit des Remeshings liegt in seiner Instabilität, wenn sehr spitzwinklige Dreiecke oder sehr hohe Valenzzahlen vorliegen. Auch muss verhindert werden, dass solche Zustände während des Remeshings auftreten. Die oben genannten Operationen dürfen deshalb nur dann ausgeführt werden, wenn spitzwinklige Dreiecke und hohe Valenzzahlen vermieden werden. Die hierzu aus /BOT 10/ entnommenen Kriterien mussten durch weitere Kriterien ergänzt werden, um den Algorithmus zu stabilisieren. Damit ergaben sich insgesamt folgende operationsspezifische Kriterien, von denen alle erfüllt sein müssen, um die betreffende Operation ausführen zu dürfen.

### **Operation: Spaltung langer Kanten**

- Kantenlänge  $> 4/3 \cdot$  Zielkantenlänge

### **Operation: Löschung kurzer Kanten**

*Einer der beiden Knoten wird an den Platz des anderen Knotens geschoben und beide Knoten werden danach miteinander verschmolzen.*

- Kantenlänge  $< 4/5 \cdot$  Zielkantenlänge
- Es gibt mindestens einen Knoten mit 0 oder 2 festen Kanten

- Durch die Knotenverschiebung entsteht keine Kante, die länger ist als  $\frac{4}{3}$  der Zielkantenlänge
- Die Dreiecksnormalen drehen sich nicht um mehr als  $50^\circ$
- Es entstehen keine Dreiecke mit Winkeln  $< 5^\circ$  oder  $> 170^\circ$
- Es dürfen keine Dreiecke aufeinander fallen
- Feste Kanten, die an dem zu löschenden Knoten angeschlossen sind, dürfen sich nicht um mehr als  $25^\circ$  drehen

**Operation: Vertauschung von Kanten**

*Die Knotenvalenz ist die Zahl der Verbindungen zu anderen Knoten. Randknoten besitzen eine Zielvalenz von 4, alle anderen eine Zielvalenz von 6.*

- Die Flächen müssen erhalten bleiben
- Nach Kantentausch dürfen die Dreiecke nicht zu spitze oder zu flache Winkel besitzen
- Betrachtet man die absoluten Differenzen der Knotenvalenz zur jeweiligen Zielvalenz, dann darf sich die Summe dieser Differenzen nicht erhöhen

**Operation: Flächenparallele Knotenverschiebung**

- Der Knoten hat 0 oder 2 feste Kanten
- Die Dreiecksnormalen drehen sich nicht um mehr als  $50^\circ$
- Es entstehen keine Dreiecke mit Winkeln  $< 5^\circ$  oder  $> 170^\circ$

**Operation: Projektion von Knoten auf die Ausgangsfläche**

*Die Projektion von Knoten auf die Ausgangsflächen oder auf feste Kanten erfolgt mit Hilfe von Suchbäumen, in denen nach den nächsten Dreiecken bzw. festen Kanten gesucht wird. Da diese Suche zu falschen Ergebnissen führen kann, müssen die folgenden Kriterien abgefragt werden.*

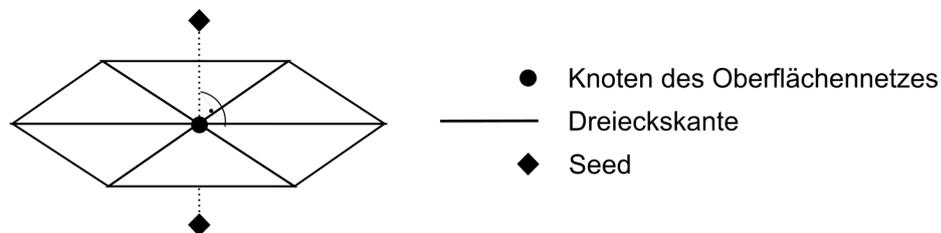
- Die Dreiecksnormalen drehen sich nicht um mehr als  $50^\circ$
- Es entstehen keine Dreiecke mit Winkeln  $< 5^\circ$  oder  $> 170^\circ$

Die Hauptschwierigkeiten der Implementierung lagen in der Identifikation der genannten Kriterien und in der konsistenten Änderung der komplexen Gitterdaten. Diese Schwierigkeiten konnten für einfache Testbeispiele befriedigend gelöst werden, jedoch sind Fehlerfreiheit und Performance noch für größere und kompliziertere Oberflächengitter nachzuweisen.

## 4.9 Erstellung von Knotenseeds

### 4.9.1 Notwendigkeit der Flächenvereinfachung

Für jeden Triangulationspunkt der Körperoberflächen (hier auch Knoten genannt) werden Knotenseeds gesucht, die Voronoi-Zellen erzeugen, welche sowohl den Knoten als die an diesen Knoten angeschlossenen Dreiecksbereiche mit ausreichender Genauigkeit abbilden. Im einfachsten Fall liegen die Dreiecke eines Knotens auf einer Ebene, sodass lediglich zwei Knotenseeds zur Darstellung der lokalen Geometrie benötigt werden (Abb. 4.3). Die Verbindungslinie zwischen beiden Seeds steht dann senkrecht auf der Ebene; der Knoten liegt auf der halben Strecke.



**Abb. 4.3** Seedpaar im Falle von Dreiecken, die auf einer Ebene liegen

Je stärker die triangulierte Fläche sich krümmt, faltet oder verzweigt, desto mehr Knotenseeds werden zu ihrer Darstellung benötigt. Die Zahl der Seeds kann dabei so hoch werden, dass eine sparsame Vergitterung scheitert. Aus diesem Grund ist in vielen Fällen eine Vereinfachung der triangulierten Flächen notwendig. Der entwickelte Algorithmus führt deshalb zuerst Flächenvereinfachungen durch, bevor er für diese die exakten Seedlagen berechnet.

Flächenvereinfachungen führen zwangsläufig zu einer Passungenauigkeit der Flächen benachbarter Knoten. Im endgültigen Gitter bilden sich dann kleine Stufen aus. Die durch diese Stufen zusätzlich entstehenden Grenzflächen sind jedoch klein, sodass sie bei Strömungs- und Transportrechnungen vernachlässigt werden können. Da die Größe der vernachlässigten Flächen bekannt ist, kann der damit verbundene Fehler abgeschätzt werden.

#### **4.9.2 Lokale Knotensysteme und Positionen der Knotenseeds**

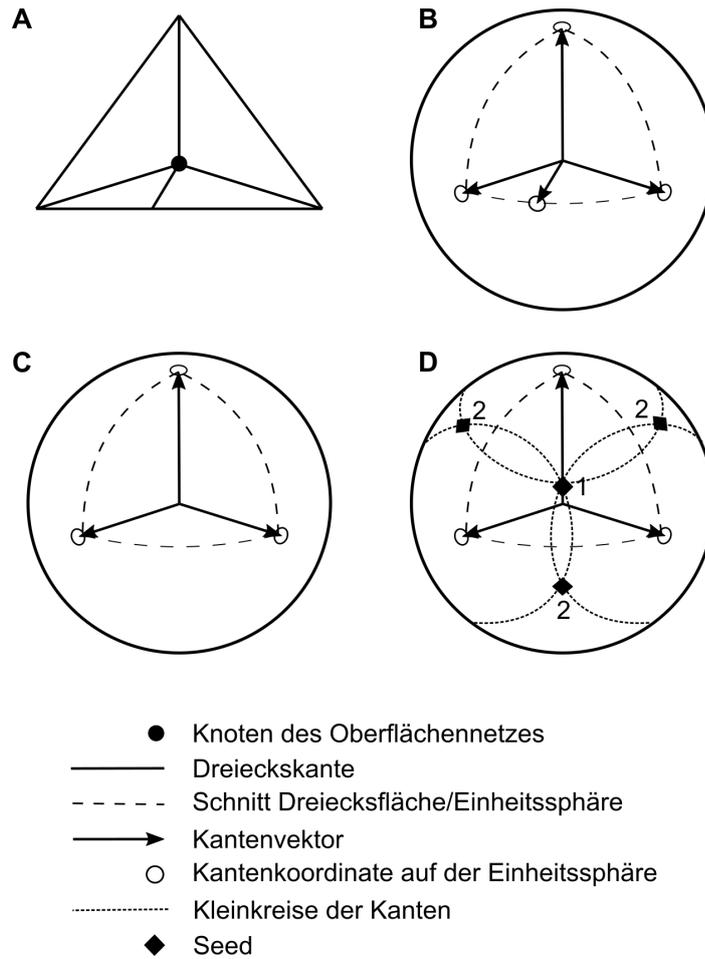
Jeder Knoten des triangulierten Oberflächengitters gehört zu einer Gruppe von Dreiecken, von denen einige über Kanten verbunden sind. Diese Konstellation wird im Folgenden das lokale Knotensystem genannt. Für jedes lokale Knotensystem werden Knotenseeds gesetzt, und zwar so, dass die triangulierte Oberfläche die erzeugten Voronoi-Elemente begrenzt. Die möglichen Lagen der Knotenseeds im dreidimensionalen Raum ergeben sich aus folgenden allgemeinen Feststellungen:

- Alle Knotenseeds liegen auf einer Sphäre um den Knoten des lokalen Knotensystems („Seedsphäre“).
- Dreiecksflächen werden durch ein oder mehrere Seedpaare gebildet, wobei die Seeds eines Paares sich an der Dreiecksfläche spiegeln.
- Seeds, die Dreieckskanten generieren, liegen auf einem Kreis, dessen Achse die Kante ist.

Um die Seedsuche zu vereinfachen, wird das Knotensystem parallel verschoben, und zwar so, dass der Knoten auf dem Nullpunkt zu liegen kommt (Translation). Die Seedsuche erfolgt der Einfachheit halber auf einer Einheitssphäre um den Nullpunkt. Die Verbindungskanten der Dreiecke werden als Kantenvektoren auf der Einheitssphäre dargestellt. Diese sind über die Dreiecksebenen miteinander verbunden (Abb. 4.4 A und B). Annähernd parallele Dreiecksflächen werden nun verschmolzen (Abb. 4.4 C). Nach dieser Flächenvereinfachung erfolgt die Seedsetzung, wobei die Herausforderung darin besteht, die Seedsanzahl dadurch zu verringern, dass Seeds für die Generierung möglichst vieler Kanten und Flächen genutzt werden (Abb. 4.4 C).

Für die Erläuterung des Algorithmus' werden für das auf den Nullpunkt projizierte lokale Knotensystem folgende Begriffe eingeführt:

- Kante: Flächen bilden mit ihren Nachbarn Kanten, die auf der Einheitssphäre von Kantenvektoren der Länge 1 repräsentiert werden.
- Fläche und Flächenwinkel: Dreiecke und miteinander verschmolzene Dreiecke liegen auf Ebenen, die Flächen genannt werden. Jede Fläche wird von zwei Kantenvektoren aufgespannt. Es ist zu beachten, dass Kantenvektoren parallel stehen können. Aus diesem Grund gehört zur Flächeninformation ein Normalenvektor. Weil zwei Kantenvektoren und ein Normalenvektor grundsätzlich zwei Flächen repräsentieren können, werden Flächen durch ein Tupel zweier Kantenvektoren definiert, wobei der erste Kantenvektor die Fläche überstreicht, wenn er um den Normalenvektor im Gegenuhrzeigersinn in Richtung des zweiten Kantenvektors rotiert wird (Rechte- Daumen-Regel). Der Winkel zwischen den beiden Kantenvektoren wird Flächenwinkel genannt.
- Zwickel: Der minimale Winkel zweier Flächen, die an derselben Kante liegen, werden „Zwickel“ genannt.
- Zelle: Die aneinanderhängenden Flächen teilen den Raum in getrennte Bereiche. Diese Bereiche werden Zellen genannt. Jede Seite einer Fläche ist genau einer Zelle zugeordnet.



**Abb. 4.4** Sparsame Seedsetzung für eine Ecke

**A:** Der betrachtete Knoten liegt in einer Ecke, die aus 4 Dreiecken gebildet wird. **B:** Nach der Projektion auf den Nullpunkt: Kantenvektoren mit Einheitslänge bilden das Knotensystem ab. **C:** Parallele Dreiecke werden verschmolzen und Kanten ggf. eliminiert. **D:** Seedlagen liegen auf den Kleinkreisen der Kanten. Überlappende Kleinkreise ermöglichen eine Verringerung der Seedanzahl. Nicht jedes Knotensystem besitzt eine sparsame Lösung.

### 4.9.3 Steuerungsroutine

Tab. 4.2 skizziert die Steuerungsroutine zur Setzung der Knotenseeds. Sie besitzt die folgenden Hauptelemente:

- Transformation eines Knotens und seiner angeschlossenen Flächen auf die Einheitssphäre, auf der später die Seeds gesucht werden.
- Zusammenfassung von Flächen, die im Rahmen eines nutzerdefinierten Toleranzwinkels als parallel gelten können. Falls das nachfolgende Seedsetzungsverfahren scheitert, wird zu diesem Schritt zurückgesprungen und der Toleranzwinkel erhöht.
- Anstoß der Seedsetzung für spezielle Flächensysteme, für die es schnell zu ermittelnde Lösungen gibt (zurzeit nur für Ebenen). Ansonsten Anstoß eines kantenorientierten Seedsetzungsverfahrens für komplexere Flächensysteme (siehe Erläuterung unten).
- Nach erfolgreicher Seedsuche werden die Seeds auf die ursprüngliche Seedsphäre des lokalen Kantensystems projiziert. Deren Radius wird auf ein Drittel des maximalen Radius gesetzt, in dem sich keine weiteren Knoten und keine benachbarten Dreiecksflächen befinden. Hierdurch soll gewährleistet werden, dass benachbarte Knotensysteme oder Flächen nicht gestört werden.

Für die Seedsuche auf der Einheitssphäre werden nahezu parallele Dreiecksflächen zu einer einzigen Fläche zusammengefasst. Durch Angabe eines Toleranzwinkels kann die Genauigkeit dieser Approximation gesteuert werden. Die Flächenvereinfachung ist ein wichtiger Schritt zur Reduktion der Seedanzahl. Insbesondere bei stark gekrümmten Flächen, die nicht rein konkav oder konvex sind, ist es oft nicht möglich, einen ausreichend kleinen Satz an Seeds oder überhaupt eine Lösung zu finden. Die Vereinfachung der ursprünglichen Geometrie trägt wesentlich zur Lösbarkeit des Seedsetzungsproblems bei.

**Tab. 4.2** Algorithmus zur Generierung von Knotenseeds

```
function Generiere Knotenseeds:

  for Knoten in alle Knoten der triangulierten Oberfläche:

    Transferiere Knoten, Kanten und Flächen
    auf knotenzentriertes Bezugssystem

    # SEEDSUCHE IN KNOTENZENTRIERTEM BEZUGSSYSTEM
    # BEI MISSERFOLG WIRD DIE GEOMETRIE SUKZESSIONSVEREINFACHT

    for Toleranzwinkel = 10° to 30° by 5°:

      # GEOMETRISCHE VEREINFACHUNG

      Vereinfache Flächengeometrie durch Zusammenfassung von Flächen, deren
      Winkel kleiner als der Toleranzwinkel ist

      Identifiziere Zellen anhand zusammenhängender Flächenseiten

      # GLATTE TEILSTÜCKE SOLLEN DURCH KONVEXE FORMEN ABGEBILDET WERDEN

      if es liegt eine einzige, fast ebene Zellgrenze vor, die weder konkav
      noch konvex ist:

        continue loop # Vereinfache weiter

      # SEED-SUCHE ANHAND VON ZELL-PRIMITIVEN

      Suche Ähnlichkeit mit Zellprimitiven # (zur Zeit nur Ebene)

      if Ähnlichkeit gefunden:

        Setze entsprechende Seeds auf die Einheitssphäre

        break loop # Seedsuche beendet

      # SEED-SUCHE DURCH SPIEGELUNG VON ANFANGSSEEDS

      Kantenorientiertes Spiegelungsverfahren (siehe folgendes Kapitel)

      if erfolgreich:

        Setze Seeds auf die Einheitssphäre

        break loop # Seedsuche beendet

    end for # vereinfache weiter ...

  Fehlerrückgabe if keine Seeds gefunden

  # TRANSFERIERE SEEDS IN DAS URSPRUNGSSYSTEM

  Ermittle Radius der Seedsphäre

  Verschiebung der Seeds von der Einheitssphäre auf die Seedsphäre

end for
```

Aus Gründen der Performance wird nach der Flächenvereinfachung überprüft, ob das Flächensystem mit einem primitiven Flächensystem übereinstimmt, dessen Knoten-seedpositionen bekannt sind. Zurzeit wird hier lediglich überprüft, ob das Flächensystem eine Ebene bildet. Die Zahl primitiver Flächensysteme sollte zukünftig erweitert werden, um die Performance der Seedsetzung zu verbessern.<sup>2</sup>

Im Rahmen der vorliegenden Arbeit wurde keine allgemeine Lösung zur Identifikation sparsamer Seedsätze für beliebig komplexe Flächenstrukturen gefunden, weshalb für komplexere Flächensysteme verschiedene heuristische Suchansätze entwickelt und getestet wurden. Das zuletzt verwendete Verfahren ist ein kantenorientiertes Spiegelungsverfahren, was bedeutet, dass nacheinander alle Kanten mit Seeds belegt werden. Dieses Verfahren wird im nächsten Kapitel besprochen.

Das kantenorientierte Spiegelungsverfahren sorgt für eine korrekte Flächenreproduktion in der Nähe der Kanten. In Sonderfällen kann es vorkommen, dass die zwischen zwei Kanten liegenden Flächenteile von anderen Seeds gestört werden. Prinzipiell könnte in solchen Fällen ein schützendes Seedpaar für Abhilfe sorgen, jedoch neigt ein solches Verfahren dazu, zu viele Seeds zu generieren. Diese Problematik sollte daher besser durch eine Flächenvereinfachung abgefangen werden.

#### **4.9.4 Kantenorientiertes Spiegelungsverfahren**

Das Seedsetzungsverfahren beginnt mit der Setzung des ersten Seeds. Dieser wird an benachbarten Flächen gespiegelt, und die hieraus hervorgehenden Seeds erzeugen wiederum weitere Seeds, sofern sie keine Spiegelpartner besitzen. Der Erfolg dieses Vorgangs wird maßgeblich bestimmt durch

- die Position des ersten Seeds,
- die Reihenfolge der Flächen, an denen gespiegelt wird,
- die Seedpositionen, die eine Fläche spiegeln (der Spiegelungsbereich der Fläche) sowie

---

<sup>2</sup> Der Versuch, alle Flächen auf primitive Flächensysteme mit bekannter Anfangsseedposition zu reduzieren, erscheint auch aus Gründen der Performance bedenkenswert. Hierzu müsste ein erweiterter Mechanismus entwickelt werden, der die Ähnlichkeit von Flächen mit geometrischen Primitiven ermittelt. Bei einer zukünftigen Weiterentwicklung des Gittergenerators sollte diese Option geprüft werden.

- die Seeds, die zusätzlich gesetzt werden müssen, wenn Flächen in ihrem Spiegelungsbereich keinen Seed besitzen.

Vor allem die ersten beiden Punkte bestimmen, ob die Seedsetzung scheitert und ob der sich ergebende Seedsatz sparsam ist oder geometrisch gleichförmig.

Das hier vorgestellte kantenorientierte Spiegelungsverfahren iteriert über Kanten. Dabei wird für jede Kante versucht, die kantennächsten Seeds an allen Flächen der Kante zu spiegeln. Die Reihenfolge der Flächen, an denen gespiegelt wird, wird demzufolge durch die Reihenfolge bestimmt, in der die Kanten durchlaufen werden. Der Hintergrund für dieses Verfahren ist, dass erfolgversprechende Kandidaten für Anfangsseeds nicht immer im Zentrum von Zellen liegen, sondern manchmal auch in den Zwickeln<sup>3</sup> von Kanten. In diesem Fall sollte zuerst geprüft werden, ob sich die betreffende Kante vollständig mit Seeds belegen lässt. Das kantenorientierte Verfahren lieferte in den betrachteten Testfällen zufriedenstellende Ergebnisse, jedoch kann nicht grundsätzlich ausgeschlossen werden, dass eine andere Flächenreihenfolge in manchen Fällen bessere Ergebnisse liefert.

Das Verfahren untergliedert sich in folgende Schritte:

- Aufstellung erfolgversprechender Kandidaten für den ersten Seed. Die Positionen für den Anfangsseed liegen in den Zentren konkaver Zellen und in spitzen Zwickeln.
- Iteration über die Kanten zur Spiegelung weiterer Seeds, ggf. Hinzufügung zusätzlicher Seeds.
- Abspeichern des Seedsatzes, falls die Seedsetzung erfolgreich war.
- Die Auswahl des günstigsten Seedsatzes.

Der zugehörige Algorithmus ist in Tab. 4.3 skizziert.

Für jeden gesetzten Seed wird verzeichnet, an welchen Flächen er gespiegelt werden sollte. Weist eine Fläche mehrere zu spiegelnde Seeds auf, so wird immer nur derjenige Seed gespiegelt, der der Fläche am nächsten ist.

---

<sup>3</sup> Der Bereich einer Kante, der von zwei Flächen, die an diese Kante grenzen, mit einem Winkel unter 180° begrenzt wird.

Seeds müssen immer dann gespiegelt werden, wenn die Fläche so orientiert ist, dass der Seed sein eigenes Spiegelbild „sieht“ bzw. es im Rahmen eines bestimmten Toleranzwinkels sehen könnte. Der Toleranzwinkel ist problemabhängig und wird auf  $20^\circ$  gesetzt, wenn der Anfangsseed in einem Zellzentrum liegt, und auf  $5^\circ$ , wenn er in einem Zwickel liegt.

Die als nächstes zu verarbeitende Kante wird aus jenen Kanten ausgewählt, deren Flächen zwar ein zu spiegelnder Seed zugeordnet wurde, die aber noch nicht vollständig belegt sind. Unter diesen Kanten wird diejenige mit dem kleinsten Flächenwinkel ausgewählt, da die Belegung kleiner Winkel prioritär erfolgen sollte. (Die Wahrscheinlichkeit, dass von einem kleinen Winkel erfolgreich in einen größeren gespiegelt wird, ist größer als umgekehrt.)

Scheitert die Seedsetzung für die aktuelle Kante, wird die Suche noch einmal mit einem Seed versucht, der näher an der Kante liegt. Auf diese Weise wird die Abhängigkeit von bestehenden Seedlagen um den Preis einer höheren Seedanzahl gelockert. Scheitert auch dieser Versuch, werden die bisher erzeugten Seeds verworfen und die Seedsuche beginnt von vorne mit dem nächsten Kandidaten für den Anfangsseed.

**Tab. 4.3** Seed-Suche durch kantenorientierte Spiegelung

```
function Kantenorientiertes Spiegelungsverfahren:  
  # STELLE ERFOLGSVERSPRECHENDE KANDIDATEN FÜR ANFANGSSEEDS ZUSAMMEN  
  Erstelle Seedkandidaten in Zentren enger Zellen  
  Erstelle Seedkandidaten in engen Zwickeln  
  
  # VERSUCHE FÜR JEDEN KANDIDATEN EINEN SEED-SATZ ZU ERZEUGEN  
  for alle Anfangsseed-Kandidaten:  
  
    # INITIALISIERUNG DES VERSUCHS  
    Ordne den Anfangsseed allen Flächenseiten zu, die nicht zu stark abge-  
    wandt sind  
    # (Seeds werden nur an Flächen gespiegelt, denen sie zugeordnet sind)  
    Wähle unter den betroffenen Kanten diejenige mit dem kleinsten Zwickel  
    als aktuelle Kante  
  
    # NACHEINANDER ALLE KANTEN MIT SEEDS BELEGEN  
    while noch nicht alle Kanten bearbeitet wurden:  
      Flächen der aktuellen Kante mit Seeds belegen (s. nächstes Kapitel)  
      if nicht erfolgreich:  
        Setze einen neuen Seed, der näher an der Kante liegt  
        Flächen der aktuellen Kante mit Seeds belegen  
      if immer noch erfolglos:  
        break # Beende diesen Versuch (Misserfolg)  
      # NÄCHSTE KANTE ERMITTELN  
      Wähle unter den unprozessierten Kanten, die in Nachbarschaft prozes-  
      sierter Kanten liegen, diejenige mit dem kleinsten Zwickel  
  
    end while  
    Speichere erzeugte Seeds if Seeds gefunden  
  end for # nächster Versuch  
  
  # ERGEBNIS DER VERSUCHE ERMITTELN  
  if es gab erfolgreiche Versuche:  
    Wähle die Versuche mit der geringsten Seedanzahl  
    Wähle unter diesen Versuchen denjenigen mit der geringsten Schwankung  
    der minimalen Seedabstände zu den Flächen  
    return Seeds  
  else:  
    return „nicht erfolgreich“
```

#### 4.9.5 Kantenvervollständigung

Die Kantenvervollständigung setzt Seeds für unvollständig besetzte Kanten. Jeder Fläche einer Kante können mehrere zu spiegelnde Seeds zugeordnet worden sein. Es wird jedoch nur derjenige Seed gespiegelt, welcher der Fläche am nächsten ist. Seeds, die über die Nachbarzelle hinausspiegeln, werden aus der Liste der zu spiegelnden Seeds gelöscht. Es wird solange über Flächen iteriert, bis keine neuen Seeds mehr durch Spiegelung gebildet werden. Flächen, die am Ende noch unbesetzt sind, erhalten ein eigenes Seedpaar.

**Tab. 4.4** Kantenvervollständigung

```
function Flächen der aktuellen Kante mit Seeds belegen:

# ERMITTLE SEEDS, DIE GESPIEGELT WERDEN MÜSSEN

Identifiziere den Kegel um den Kantenvektor mit minimalem Öffnungswinkel,
auf dem noch Seeds liegen („Seedkegel“)

Betrachte nur die Seeds, die auf diesem Kegel liegen.

# SEEDS SPIEGELN

do

  for alle Flächen:

    if mindestens einer Flächenseite ist ein Seed zugeordnet:

      Identifiziere den Seed mit dem kleinsten Abstand zur Fläche

      if Seed hat kein Spiegelbild:

        if Seed spiegelt in Nachbarzelle:

          Seed spiegeln

          Seed allen Zellflächen zuordnen, in deren Spiegelungsbereich
          er liegt

        else:

          Seed von der Flächenseite entfernen

    while es wurden Seeds generiert and Anzahl Seeds < 10

# UNBELEGTE FLÄCHEN BELEGEN

for alle unbelegte Flächen:

  Generiere auf dem Seedkegel ein nahes Seedpaar und weise es den Flä-
  chenseiten zu
```

#### 4.10 Erstellung von Schutzseeds

Knotenseeds erzeugen näherungsweise die Dreiecksflächen ihres jeweiligen Knotens und sichern damit die Integrität der knotennahen Flächen. Die Integrität von Dreiecksflächen kann jedoch in knotenfernen Bereichen durch die Seeds benachbarter Flächen gestört werden. Dies tritt insbesondere bei schmalen Körpern auf, deren gegenüberliegende Flächen einander nahekommen. Solche störenden Seeds werden nachfolgend *Störseeds* genannt und Seeds, die gesetzt werden, um diese Störungen zu verhindern, *Schutzseeds*. Seeds, die außerhalb des Umkreises eines Oberflächendreiecks liegen, können dieses Dreieck nicht stören, da sie weiter entfernt sind als die Knotenseeds der Dreiecksknoten. Als Störseeds einer Dreiecksfläche kommen also nur solche Seeds in Frage, die im Umkreis des Dreiecks liegen.

Tab. 4.5 zeigt den Algorithmus zur Setzung von Schutzseeds. Dieser prüft für jedes Dreieck, welche Seeds im Umkreis liegen, und identifiziert unter diesen alle Seedpaare, die das Dreieck schützen. Die restlichen Seeds werden als potenzielle Störseeds klassifiziert. Die Identifikation der schützenden Seedpaare muss tolerant erfolgen, da die Knotenseeds die Dreiecksflächen aufgrund der vorgenommenen Flächenvereinfachungen nicht exakt reproduzieren müssen.

Die Störseeds werden nun nach ihrem Abstand zur Dreiecksfläche sortiert und, beginnend mit dem nächsten Störseed, durchlaufen. Für jeden Störseed wird geprüft, ob er durch die schützenden Seeds, die auf der gleichen Dreiecksseite liegen, von der Dreiecksfläche abgeschirmt wird. Falls nicht, wird der Störseed entweder gespiegelt oder – wenn diese Spiegelung über den gegenüberliegenden Körper hinausspiegelt – ein näheres Seedpaar eingefügt. Im Falle einer Spiegelung wird der Störseed für die anderen potenziellen Störseeds zu einem Schutzseed.

**Tab. 4.5** Algorithmus zur Generierung von Schutzseeds

```
function Erstelle_Schutzseeds:
  do
    for alle Dreiecke:
      Identifiziere alle Seeds im Radius des Dreiecks
      Unterscheide zwischen Seedpaaren, welche die Fläche schützen,
      und potenziellen Störseeds

      # ERMITTLE GESTÖRTE DREIECKSTEILFLÄCHE (POLYGON)
      # Sortiere Störseeds nach Entfernung zur Dreiecksfläche
      for jeden potenziellen Störseed:
        for jeden Schutzseed auf der gleichen Dreiecksseite:
          Verkleinere den potenziell gestörten Teil des Dreiecks (Polygon)
          if Dreieck wird gestört:
            Erzeuge einen Schutzseed durch Spiegelung des Angreifers
            oder füge ein näheres Schutzseedpaar ein, falls die
            Spiegelung in das falsche Materialgebiet fällt

        end for
      while neue Seeds wurden hinzugefügt
```

#### 4.11 Erstellung von Körperseeds

Die Erstellung von Körperseeds erfolgt nach der Erstellung der Schutzseeds, damit die Körperseeds die Anzahl der Schutzseeds nicht unnötigerweise erhöhen. Im Gegensatz zu den Knotenseeds und Schutzseeds werden die Körperseeds nicht paarweise erzeugt, sondern einzeln in regelmäßiger Anordnung, die in dem in Tab. 4.6 dargestellten Algorithmus spezifiziert werden kann. Insbesondere können hier Seedanordnungen gewählt werden, die orthogonale Elemente erzeugen und somit die Anzahl der Elementverbindungen reduzieren.

Technisch gesehen ist die Erzeugung der Körperseeds einfach. Der Aufwand erhöht sich, wenn viele unterschiedliche geometrische Formen für die Körperseeds implementiert werden sollen. Der Algorithmus vermeidet Störungen der vorhandenen Knoten- und Schutzseeds durch die hinzukommenden Körperseeds und eine Erzeugung von Körperseeds außerhalb des Modellraums. Es erfolgt eine Buchhaltung der Körperseeds und ihrer assoziierten Materialgebiete.

**Tab. 4.6** Algorithmus zur Generierung von Körper-Seeds

```
function Erstelle Körperseeds in der Form „Form“  
if „Form“ entspricht einem der möglichen Formen:  
    Prüfe, ob Eingabeparameter konsistent für ein rechtshändiges Koordinatensystem sind  
  
    Erstelle Menge von 3D-Koordinaten basierend auf den Argumenten Form, Ursprung, Endkoordinaten und Abstand  
  
    for jeden Punkt aus Matrix:  
        Unterscheide Zugehörigkeit zum Materialgebiet  
  
        if Punkt außerhalb Modell:  
            Berücksichtige Punkt nicht weiter  
  
        if Punkt potenzieller Störseed:  
            Berücksichtige Punkt nicht weiter  
  
    end for  
  
    Füge verbliebene Punkte den Körperseeds hinzu
```

#### 4.12 Erstellung des Voronoi-Gitters

Die Erstellung des Voronoi-Gitters kann erst erfolgen, nachdem die Seedsetzung abgeschlossen ist. Da äußere Regionen mit Infinite Ridges die grafische Darstellung erheblich beeinträchtigen und auch für die TOUGH2-Modellrechnungen nicht benötigt werden, müssen diese aussortiert werden. Deshalb müssen alle Referenzen, welche untereinander zwischen den Koordinaten der Seeds und Vertices und den Indizes der Regionen, Ridges, Vertices bestehen, aktualisiert werden. Außerdem müssen ggf. einige der zunächst nicht berücksichtigten Regionen später als inaktive Elemente (siehe Kap. 4.6 bzw. 4.13) doch weiter genutzt werden. Im Algorithmus in Tab. 4.7 spielt deshalb die Datenhaltung eine große Rolle.

**Tab. 4.7** Algorithmus zur Erstellung des Voronoi-Gitters

```
function Erstelle_Gitter:  
    Erstelle Voronoi-Objekt aus scipy.spatial.voronoi  
    for jeden Seed aus Voronoi-Objekt:  
        Unterscheide innerhalb oder außerhalb Modell  
        if seed außerhalb Modell:  
            Möglicherweise noch für inaktive Elemente berücksichtigen  
    end for  
    for jedes Seed-Paar, jedes ridges aus Voronoi-Objekt:  
        Unterscheide, ob Seed-Paar innerhalb oder außerhalb des Modell oder  
        Kombination aus innerhalb/außerhalb ist  
        if Seed-Paar innerhalb Modell:  
            Behalte Seed-Paar  
            Aktualisiere alle Referenzen zu seeds, regions, vertices, ridges,  
            ridge_points etc.  
        if Seed-Paar außerhalb Modell:  
            Berücksichtige Seed-Paar nicht weiter  
            Aktualisiere alle Referenzen zu seeds, regions, vertices, ridges,  
            ridge_points etc.  
        if Seed-Paar Kombination aus innerhalb/außerhalb:  
            Sortiere Seed-Paar separat für potenzielle inaktive Elemente  
            Aktualisiere alle Referenzen zu seeds, regions, vertices, ridges,  
            ridge_points etc.  
    end for  
    for jede aktualisierte Region und aktualisierten Ridge:  
        Berechne Volumen der Region  
        Berechne Fläche des Ridge  
        Berechne Abstand des Ridge-Mittelpunkts zu jedem Regionen-Mittelpunkt  
        Berechne Abstandsvektor benachbarter Regionen  
    end for
```

#### 4.13 **Setzung inaktiver Elemente und Export der TOUGH2-Gitterdatei**

Inaktive Elemente setzen in TOUGH2 Dirichlet-Randbedingungen. Inaktive Elemente zeichnen sich dadurch aus, dass sie ein unendlich großes Volumen (z. B.  $1E80 \text{ m}^3$ ) haben (TOUGH2-GRS akzeptiert auch Element mit Nullvolumen). Im TOUGH2-Eingabeformat stehen diese Elemente am Ende der Elementliste. Wie bei aktiven Elementen müssen auch die Verbindungen zwischen aktiven und inaktiven Elementen durch „Connections“ (Elementverbindungen) mit Angabe einer Interfacefläche und der

Entfernung der Interfacefläche zu den Elementzentren spezifiziert werden. Der in Tab. 4.8 dargestellte Algorithmus stellt dazu das Werkzeug bereit. Der Abstand zwischen dem Zentrum eines inaktiven Elementes und einer Interfacefläche wird auf einen kleinen Wert gesetzt, um die durch das inaktive Element definierte Randbedingung nahe an das Interface zu bringen. Verbindungen zwischen inaktiven Elementen kommen nicht vor.

**Tab. 4.8** Algorithmus zur Erstellung der TOUGH2-Eingabe mit inaktiven Elementen

```
function inaktive_Zellen:  
  for jedes Seed-Paar:  
    Prüfe, welche Option für die inaktiven Zellen gewählt wurde  
    Erzeuge TOUGH2 Element-Namen und Paraview-ID  
    Prüfe, ob ein Seed des Paares inaktives Element sein soll  
    if Seed soll inaktiv sein:  
      Gib Element „inaktive“ Eigenschaften  
    if Seed soll aktiv sein:  
      Gib Element „aktive“ Eigenschaften  
    Erzeuge TOUGH2 Connection  
  end for  
  Erzeuge Daten für TOUGH2 und Paraview  
  Schreibe Eingabe-Datei im TOUGH2-Format
```

## 4.14 Datenbereitstellung

### 4.14.1 Zielsetzung

Der vorgestellte Gittergenerator soll unter Verwendung bzw. Einbeziehung realer geologischer Standortdaten eingesetzt werden. Solche Daten werden u. a. von den geologischen Landesämtern erhoben und zur Verfügung gestellt. Die Aufbereitung dieser Daten zur Erstellung der Eingangsdaten für den entwickelten Gittergenerator, ist ein notwendiger Vorverarbeitungsschritt, der im vorliegenden Vorhaben anhand von geologischen Datensätzen des Landesamtes für Bergbau, Energie und Geologie (LBEG) erprobt wurde und für die weiteren Codeentwicklungen durchgeführt wurden. Das wesentliche Ziel ist dabei die Überführung der triangulierten geologischen Schichtflächen, welche im betreffenden Datensatz im GOCAD-Dateiformat abgelegt sind, in geschlossene triangulierte Körperoberflächen, die durch Material- oder Gebietsgrenzen gebildet werden. Zur

Verarbeitung der Daten soll dabei auf frei zugängliche Programme zurückgegriffen werden. In diesem Bericht werden die grundsätzlichen Arbeitsschritte und Ergebnisse dargestellt.

Der generelle Ablauf der Datenverarbeitung ist wie folgt.

- Erste Visualisierung und Datenselektion mit Geoscience Analyst (Version 2.80)
  - Importieren der GOCAD-Dateien
  - Auswahl und Selektion der zu betrachtenden GOCAD-Dateien
- Konvertierung der Datenformate mithilfe der im Vorhaben entwickelten Konvertierungsroutine `conv_ts_to_csv_v03.py` für eine Weiterverarbeitung mit FreeCAD.
  - Konvertierung der selektierten Dateien in ASC-Dateien als Input für FreeCAD
- Erzeugung geschlossener Körper mit FreeCAD (Version 0.18)
  - Importieren der konvertierten Dateien
  - Erstellen von Dreiecksnetzen mit dem im Vorhaben entwickelten FreeCAD-Makro `ImportPointDataMesh_v0_5.FCMacro`
  - Zusammenführung aller Dreiecksnetze zu geschlossenen Körperflächen
  - Exportieren des gesamten Netzes im STL-Format
- MeshLab (Version v2020.03)
  - Bereinigung des Netzes von Gitterfehlern
  - Abspeichern im STL-Format

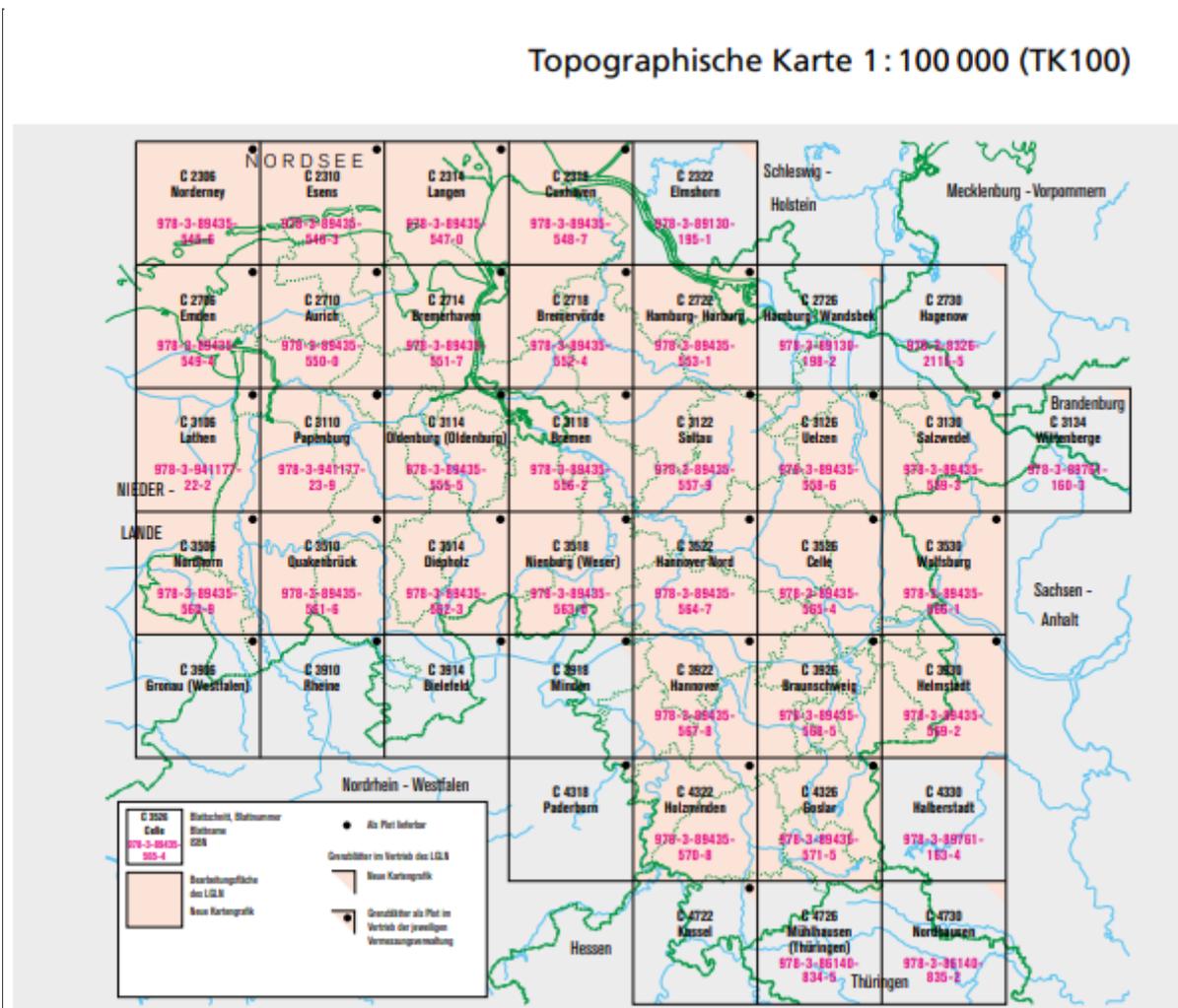
#### **4.14.2 Datengrundlage**

Das LBEG erarbeitet und aktualisiert kontinuierlich das geologische 3D-Modell des tieferen Untergrundes von Niedersachsen. Das Modell basiert auf den Daten des Geotektonischen Atlas von Nordwestdeutschland und dem Deutschen Nordsee-Sektor. Es bildet die 14 stratigraphischen Einheiten des Geotektonischen Atlas ab, wobei als zusätzliche Ebenen die Quartärbasisfläche sowie die Geländeoberfläche integriert sind. Das Modell wird vom LBEG blattschnittweise (TK100) unter anderem als GOCAD-

Objekte bereitgestellt (siehe Abb. 4.5). Das GOCAD-Dateiformat ist ein ASCII-basiertes Format, wobei die Daten des LBEG vom Objekttyp Tsurf mit der Dateiendung `ts` sind. Die zugehörigen Dateinamen folgen der Form

`[praefix]_[kachelid]_[klasse]_[horizontid]_[baueinheit]_[praezisierung].ts`

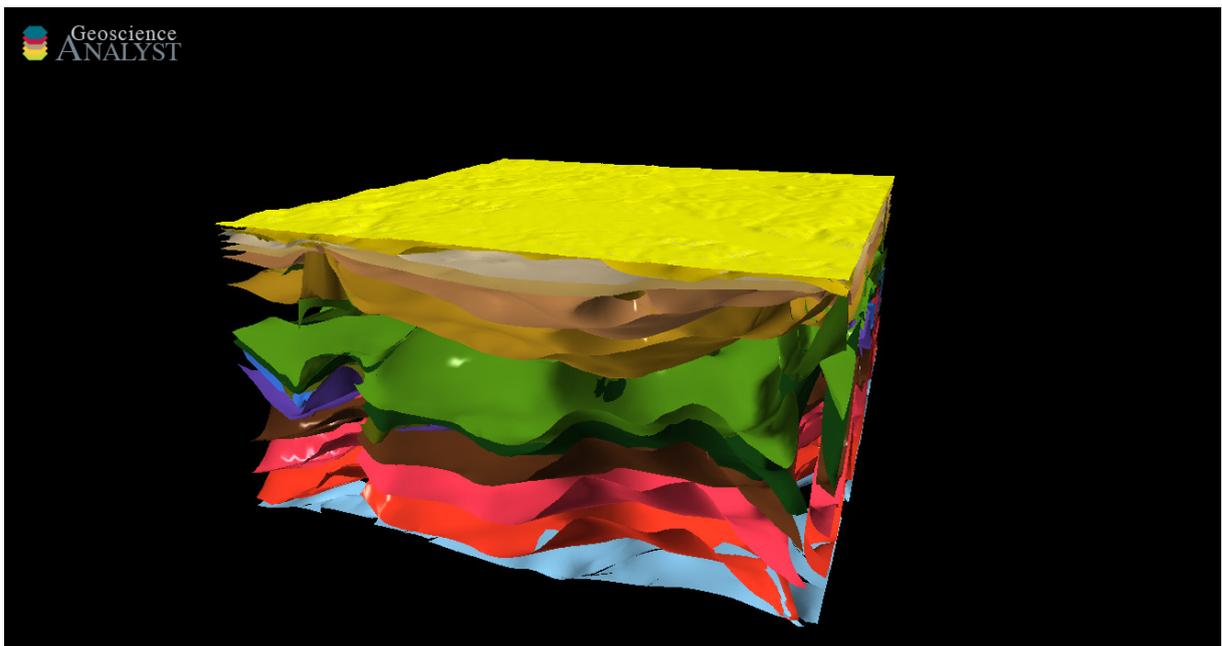
Zur Erprobung des generellen Ablaufs der Datenverarbeitung wurden die 3D-Daten des Blattes C3126 herangezogen. Der Datenumfang beträgt für diesen Blattschnitt ca. 39 MB für 338 Dateien.



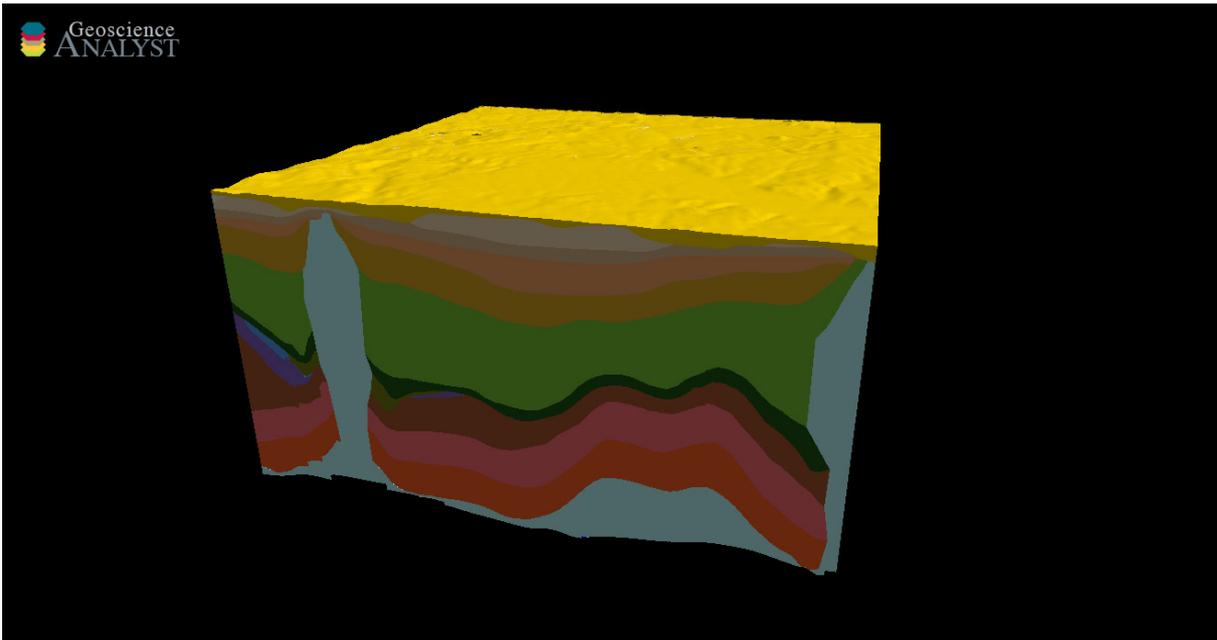
**Abb. 4.5** Darstellung der Blattschnitte (Kacheln) der Topographischen Karte 1:100000 (TK100) von Niedersachsen (entnommen aus /LAN 17/)

#### 4.14.3 Erste Visualisierung

Im Rahmen der Recherche wurde das Programm Geoscience ANALYST von Mira Geoscience Ltd. mit Hauptsitz in Montreal (Kanada) zur Darstellung und ersten Analyse der im GOCAD-Format vorliegenden Daten verwendet. Das Programm enthält ein kostenfreies Modul zur Betrachtung der Daten. Die ersten Anwendungen mit dem freigeschalteten Datenviewer in diesem Programm erfolgten mit der Version 2.80. Die folgenden Darstellungen (Abb. 4.6 bis Abb. 4.7) zeigen die visualisierten Daten des selektierten Blattschnittes.



**Abb. 4.6** Perspektivische Darstellung der Daten „Basisflaechen“ der Kachel C3126 mit 5-facher Überhöhung der z-Koordinaten



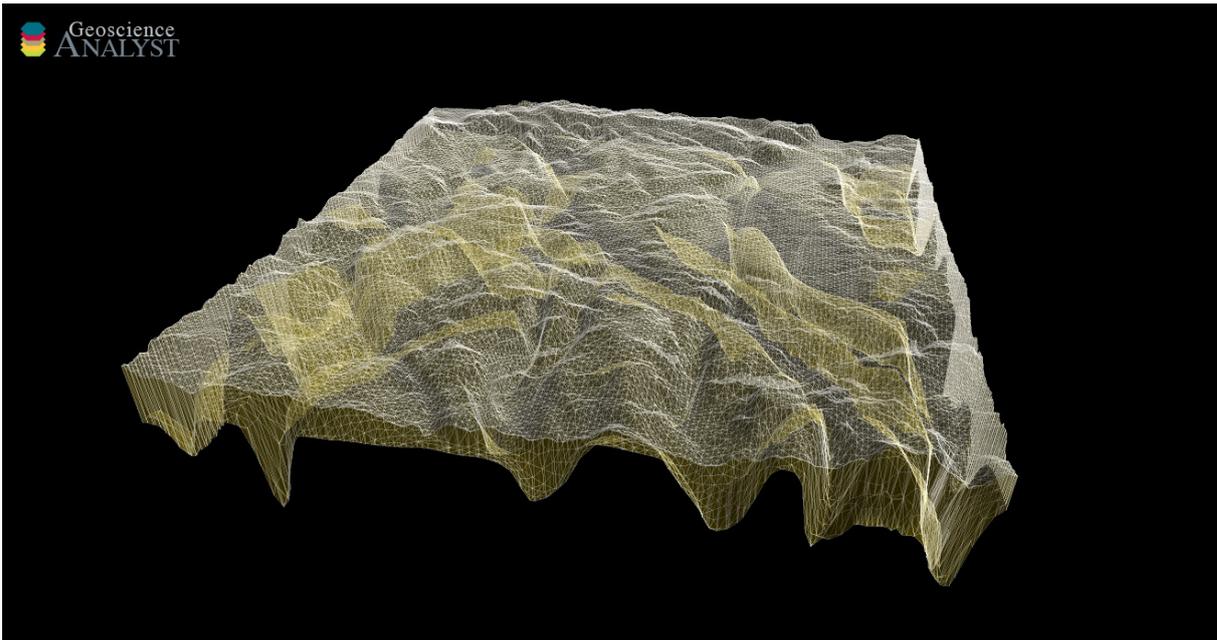
**Abb. 4.7** Perspektivische Darstellung der gesamten Daten der Kachel C3126 mit 5-facher Überhöhung der z-Koordinaten

#### 4.14.4 Selektion einer Beispielschicht

Zu Erprobung des weiteren Verfahrens wurde aus dem Blattschnitt C3126 eine Beispielschicht selektiert. Es wurden bewusst keine Daten zu Störungen und Salzstockumhüllenden berücksichtigt, um zunächst die prinzipielle Machbarkeit an einem weniger komplexen Modell zu zeigen. Das Arbeitsbeispiel ergibt sich aus den folgenden Dateien:

- Ordner: Basisflaechen
  - ts\_C3126\_k2\_01\_DGM\_basis.ts
  - ts\_C3126\_k2\_02\_q\_basis.ts
- Ordner: Waende
  - ts\_C3126\_k2\_02\_q\_TKwand-W.ts
  - ts\_C3126\_k2\_02\_q\_TKwand-E.ts
  - ts\_C3126\_k2\_02\_q\_TKwand-S.ts
  - ts\_C3126\_k2\_02\_q\_TKwand-N.ts

Die Abb. 4.8 zeigt das Dreiecksnetz der Beispielschicht.



**Abb. 4.8** Perspektivische Darstellung vom Dreiecksnetz der Beispielschicht mit 20-facher Überhöhung der z-Koordinaten

#### 4.14.5 Datenkonvertierung

Das für die weitere Datenverarbeitung benötigte Programm FreeCAD (verwendete Version: 0.18) ermöglicht den Import vieler Datenformate, allerdings befindet sich unter diesen nicht das GOCAD-Format. Für die weitere Behandlung der Daten war daher eine Konvertierung der Dateien in ein geeignetes Format erforderlich. Als Zielformat boten sich hier die Formate ASC, PCD und PLY an.

Zur Konvertierung der GOCAD-Dateien wurde ein Skript in der Programmiersprache Python entwickelt. Die Programmiersprache Python wurde gewählt, weil die weiteren Code-Entwicklungsarbeiten in diesem Vorhaben ebenfalls mittels Python realisiert wurden und das Programm FreeCAD eine Schnittstelle zu Python-Skripten vorhält.

Die Konvertierung der GOCAD-Daten legt für jede `ts`-Datei anhand der enthaltenen Schlüsselwörter fünf neue Dateien an:

- `<Dateiname>_bd.asc`  
(`bd` = BORDER, konvertierte Datei mit Daten zu BORDER),
- `<Dateiname>_bs.asc`  
(`bs` = BSTONE, konvertierte Datei mit Daten zu BSTONE),

- `<Dateiname>_gi.asc`  
(gi = General Information, konvertierte Datei mit Daten zum Header),
- `<Dateiname>_tr.asc`  
(tr = TRGL, konvertierte Datei mit Daten zu den Dreiecken)
- `<Dateiname>_vx.asc`  
(vx = VRTX, konvertierte Datei mit Daten zu den Stützstellen (Punkte, Vertices))

Für die weitere Verarbeitung in FreeCAD werden im Prinzip nur die Dateien mit der Endung `_vx.asc` und „`_tr.asc`“ benötigt. Die anderen Dateien wurden zum Zweck möglicher weiterer Entwicklungen, z. B. zur Analyse des Modellrandes, angelegt. Nach Auswahl des Ordners sucht das Konvertierungsskript nach GOCAD-Dateien („`.ts`“), wobei auch Unterordner einbezogen werden, und konvertiert alle gefundenen Dateien.

#### 4.14.6 Generierung eines gemeinsamen Oberflächennetzes

Die durch den Konverter generierten Dateien `_vx.asc` und `_tr.asc` bilden die Grundlage für die Dreiecks-Netzerstellung mit dem Programm FreeCAD. Das Ziel ist nun, diese Dateien in FreeCAD zu importieren und eine sog. MESH-Datei mit den Informationen zu den Dreiecken zu erstellen. Hierzu bietet sich die Import-Funktion von Punktdaten in FreeCAD an. Die `_tr.asc`-Dateien beinhalten zwar keine Punktdaten im Sinne von Koordinaten, aber dennoch wird die angebotene Funktion dazu ausgenutzt, um die Daten zu importieren, da jede Zeile drei Indices enthält und somit der geforderten Struktur genügt. Die Funktion unterscheidet dabei nicht, ob die Datei Koordinaten oder wie in diesem Fall Indices beinhaltet.

Neben der genannten Importfunktion bietet FreeCAD, wie schon erwähnt, eine Schnittstelle zu Python-Skripte (Makros) an. Für den Import der `_vx.asc`- und `_tr.asc`-Dateien und die Erzeugung eines Dreiecksnetzes (MESH) wurde im Vorhaben ein entsprechendes Makro `ImportPointDataMesh_v0_5.FCMacro` entwickelt.

Bei der Eingabeaufforderung des Makros muss die gewünschte `_vx.asc`-Datei ausgewählt werden, die entsprechende `_tr.asc`-Datei wird automatisch eingebunden. Über die Indices der Datei `_tr.asc` wird eine Verbindung zu den Koordinaten der Datei `_vx.asc` hergestellt, um daraus das entsprechende Dreiecksnetz (MESH) zu generieren. Hierzu müssen die Dateien in demselben Verzeichnis abgelegt sein.

Anhand des Makros wurden für das Arbeitsbeispiel sechs Dreiecksnetze bestehend aus den Basisflächen und den Wänden generiert, die die Körperumhüllende bilden. Diese Dreiecksnetze wurden mittels FreeCAD in ein gemeinsames geschlossenes Dreiecksnetz zusammengeführt. Hierzu wählt man in der Oberfläche das Menü „Mesh Design“ aus und wählt aus der Combo-Ansicht die zusammenzuführenden Netze aus. Abschließend wählt man im Menü „Netze“ die Funktion „Zusammenführen“.

Das Gesamtnetz für das Arbeitsbeispiel besteht aus 45787 Punkten, 132507 Kanten und 86726 Flächen. Die Anzahl an Punkten, Kanten und Flächen entspricht den Summen aus den sechs Einzelnetzen (siehe Tab. 4.9).

**Tab. 4.9** Übersicht über Anzahl an Stützstellen (Punkte), Kanten und Flächen der einzelnen Dreiecksnetze und des zusammengeführten Gesamtnetzes

Bezug	Datei	Punkte	Kanten	Flächen
Dach	ts_C3126_k2_01_DGM_basis_mesh	33327	98528	65202
Boden	ts_C3126_k2_02_q_basis_mesh	10038	29147	19110
East	ts_C3126_k2_02_q_TKwand_E_mesh	582	1161	580
North	ts_C3126_k2_02_q_TKwand_N_mesh	548	1093	546
South	ts_C3126_k2_02_q_TKwand_S_mesh	566	1129	564
West	ts_C3126_k2_02_q_TKwand_W_mesh	726	1449	724
Summe	Mesh	45787	132507	86726

Das erzeugte Netz wurde in das Format STL (C3126\_Arbeitsbeispiel.stl) exportiert. Die Dateierweiterung STL steht für Standard Triangulation/Tesselation Language. Mit dem STL-Format lassen sich die Oberflächen von 3D-Körpern anhand von Dreiecksflächen beschreiben. Die Dreiecksflächen werden durch die jeweiligen drei Eckpunkte und die zugehörige Flächennormale des Dreieckes charakterisiert.

#### 4.14.7 Bereinigung des Oberflächennetzes

Das Oberflächennetz musste anschließend bereinigt werden, um Duplikate von Stützpunkten (Vertices) oder Dreiecken, sich selbstschneidende Dreiecke, schlechte Qualität von Dreiecken, zu dicht beieinanderliegenden Stützpunkte und Lücken zu vermeiden.

Für die Bereinigung wurde das frei zugängliche Programm MeshLab (Version v2020.03) verwendet. Das Programm bietet eine Schnittstelle zu Dateien im STL-Format und darüber hinaus eine Vielzahl von Funktionalitäten zur Erstellung, Aufbereitung und Darstellung von Oberflächennetzen.

Zunächst wurde die Datei `C3126_Arbeitsbeispiel.stl` in MeshLab importiert. Das importierte Oberflächennetz wurde daraufhin auf folgende Inkonsistenzen geprüft:

- Prüfung und Eliminierung von sich schneidenden Dreiecken (Ergebnis: 40 Dreiecke und entsprechende Stützstellen gefunden und gelöscht).
- Prüfung und Vereinigung von dicht beieinanderliegenden Stützpunkten (Ergebnis: Es wurden 69 entsprechende Stützstellen gefunden und vereinigt).
- Prüfung auf Duplikate von Stützpunkten (Ergebnis: Es lagen keine Duplikate vor).
- Prüfung auf Duplikate von Dreieckflächen (Ergebnis: Es lagen keine Duplikate vor).
- Prüfung auf Dreieckflächen mit der Fläche Null (Ergebnis: Es lagen keine Dreiecke mit der Fläche Null vor).
- Prüfung auf Lücken (Ergebnis: Es wurden 25 Lücken geschlossen und 35 neue Dreiecke generiert).

Bei der Anwendung dieser Aufbereitungsfunktionen kann die Reihenfolge auf das Ergebnis eine Rolle spielen. Darüber hinaus zeigte sich, dass auch eine mehrmalige hintereinander geschaltete Ausführung von bestimmten Funktionen Einfluss auf das Aufbereitungsergebnis haben kann. Eine Optimierung der Aufbereitungstechnik könnte eine Aufgabe in zukünftigen Vorhaben sein.

Die aufbereitete Datei `C3126_Arbeitsbeispiel.stl` wurde unter dem Namen `C3126_Arbeitsbeispiel_optimiert.stl` abgespeichert und dient nun als Grundlage für die Weiterverarbeitung.

#### **4.15 Entwicklung von Visualisierungswerkzeugen zur Ergebnisanalyse**

Zur Visualisierung von Gittern und Simulationsergebnissen von TOUGH2-GRS und TOUGH2-MP-GRS wurde das bereits im Vorhaben 4715E03230 angewendete Programm ParaView verwendet. Für ParaView wurden in diesem Vorhaben Makros zur automatisierten Erstellung verschiedener Darstellungstypen entwickelt, um Anwender bei der Analyse von Simulationsergebnissen zu unterstützen. Neben den Möglichkeiten zur Visualisierung der allgemeinen Simulationsergebnisse wurden Möglichkeiten zur Identifizierung skalarer oder vektorieller Extremwerte erkundet. Für die Entwicklung eines

entsprechenden Postprocessings für TOUGH2-MP-GRS wurde ein bereits existierendes, im Vorhaben 4715E03230 erstelltes Gittermodell mit rechteckigen Elementen wiederverwendet, um die Entwicklungsarbeiten parallel zur Entwicklung des Gittergenerators betreiben zu können.

#### 4.15.1 Methoden zur Erstellung von Darstellungsvorlagen in ParaView

Für die Auswertung von Rechenergebnissen ist es gängige Praxis, zunächst die zeitliche Entwicklung einzelner Zustände im Gesamtmodell zu visualisieren. Hiermit wird zunächst ein allgemeiner Überblick erlangt. Hiervon ausgehend können weitere, detailliertere Untersuchungen, zum Beispiel lokaler Zustände in einzelnen Elementen, im Modellgitter erfolgen. Für detailliertere Untersuchungen eignen sich z. B. Linienplots.

In ParaView erfolgt eine Visualisierung von Daten durch das sogenannte Pipeline-Prinzip. Durch Zusammenstellung sogenannter Quellen (Sources), Filter und Senken (Sinks) durch den Nutzer ergeben sich sogenannte Pipelines, die vielfältige Arten der Datenverarbeitung ermöglichen. Durch Sources werden die Daten in das System eingelesen (Reader). Sinks wandeln Daten in Grafiken um, damit sie auf einem Computerbildschirm dargestellt werden können (Rendering bzw. Render-View). Filter sind Algorithmen, die zwischen Sources und Sinks gesetzt werden können /AYA 19/. Es können mehrere Filter hintereinandergeschaltet werden.

Durch die Erstellung von Makros kann die Erstellung häufig verwendeter Darstellungen bzw. Visualisierungspipelines vereinfacht werden. Makros sind in ParaView im Wesentlichen kleine Python-Skripte, welche einzelne Visualisierungsschritte mit nur einem einfachen Aufruf ausführen können. ParaView bzw. dessen GUI (graphical user interface) kann komplett über Python-Skripte gesteuert werden.

ParaView bietet mehrere Möglichkeiten zur automatisierten Makroerstellung an:

- Zunächst kann ein Makro über die sog. Trace-Funktion aufgezeichnet werden. Die Aufzeichnung wird unter *Tools* → *Start Trace* gestartet und nach Beendigung der Aufgaben über *Stop Trace* (ebenfalls im Menü *Tools*) geschlossen. Dadurch wird ein Python-Skript mit den entsprechenden Funktionen erstellt, das die in der GUI ausgeführten Aktionen rekonstruiert. Die Python-Skripte können in der Python Konsole von ParaView (unter *View* → *Python Shell*) mit dem Befehl *Run Script* ausgeführt werden. Zudem besteht auf der Konsole die Möglichkeiten, das Skript über *Save as*

*Macro* als Makro abzuspeichern. Das Makro ist jetzt in der Makro-Symbolleiste der ParaView-GUI zu sehen.

- Eine weitere Möglichkeit zur Erstellung von Makros besteht darin, eine bereits erstellte Visualisierung als ParaView-Status abzuspeichern. Unter *File* → *Save State* wird die aktuelle Darstellung als pvm-Statusdatei gespeichert. Auf diese Weise erhält man eine XML-Datei.
- Eine weitere Möglichkeit zur Automatisierung von Visualisierungen in ParaView besteht in der Programmierung von Plugins, also z. B. von Windows ausführbaren dll-Dateien und werden in C++ programmiert. Hierdurch gibt es mehr Möglichkeiten auf die C++ Funktionen von VTK (Visualization Toolkit) zuzugreifen. Die GUI von ParaView basiert auf VTK und ist eine Open-Source-Software zur Bearbeitung und Anzeige wissenschaftlicher Daten.

Die erste Möglichkeit zur Makroerstellung über die Trace Funktion bietet die Möglichkeit, den Grundstatus der jeweiligen Visualisierung als Python-Skript mit den entsprechenden Funktionsaufrufen zu speichern. Dieses Skript kann nun entsprechend adaptiert werden. Die Makroerstellung durch Python-Skripte wurde im Vorhaben als die geeignetste Vorgehensweise angesehen. Die Speicherung des Visualisierungsstatus als XML-Datei über die Funktion *Save State* erscheint aufgrund des unübersichtlichen XML-Codes als weniger zielführend. Diese Funktion wurde jedoch trotzdem getestet. Die letzte Möglichkeit über die Programmierung durch Plugins ist sehr aufwendig und wurde daher in diesem Vorhaben nicht weiter untersucht.

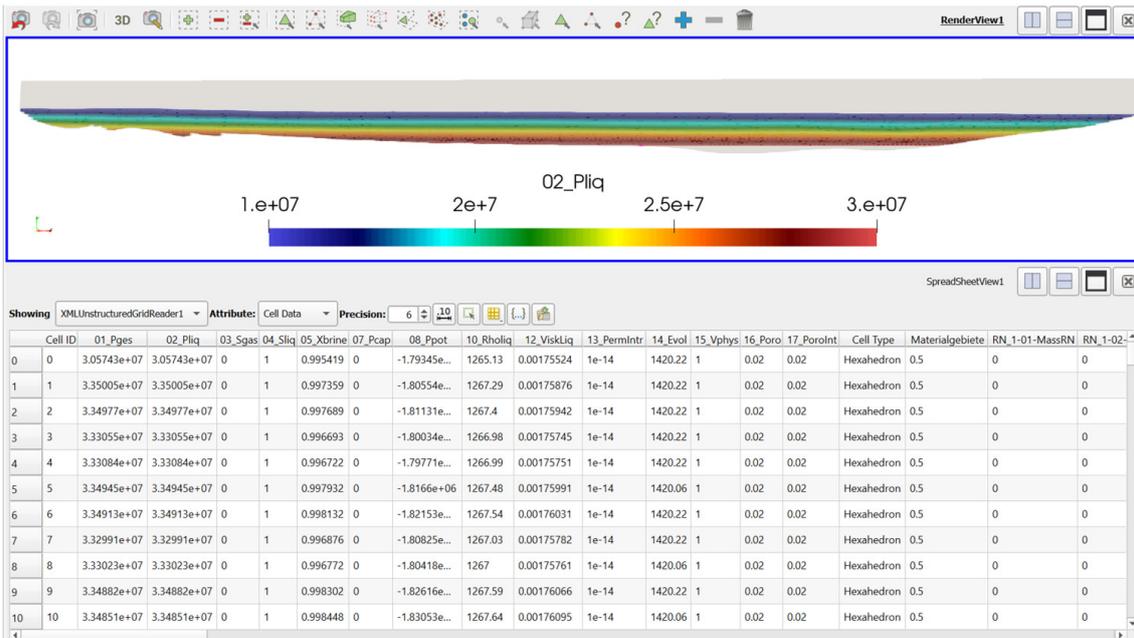
## **4.15.2 Visualisierungsvorlagen**

Mithilfe von ParaView-Makros wurden automatische Visualisierungen von Simulationsergebnissen zur Darstellung einzelner Modellparameter, wie z. B. von Fluidsättigungen (Gas, Flüssigkeit) und Geschwindigkeitsvektoren erstellt. Dabei sollten auch skalare und vektorielle Extremwerte identifiziert werden.

### **4.15.2.1 Darstellung von Zellen mit einem vorgegebenen Wertebereich**

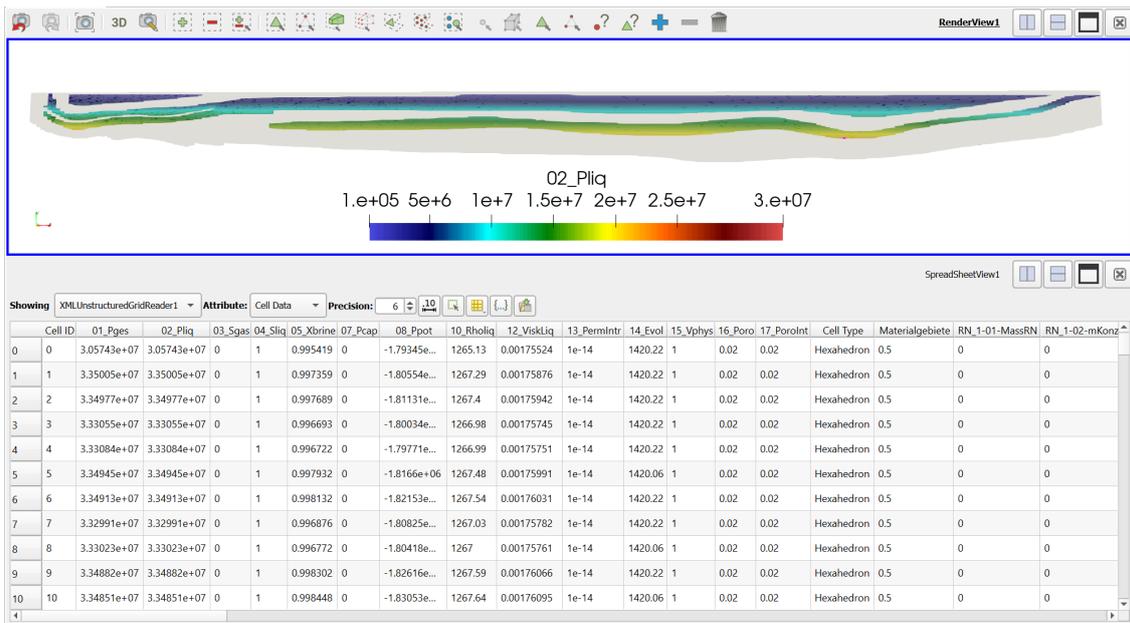
Bei der Visualisierung von Simulationsergebnissen ist es häufig von Interesse, in welchen Zellen Zustandsvariablen hohe Werte annehmen. Die in Abb. 4.9 gezeigte sogenannte Render-Ansicht, bietet die Möglichkeit, über den sog. Threshold-Filter nur solche Zellen anzuzeigen, bei denen die dargestellte Variable in einem vorgegebenen

Datenbereich liegt. In Abb. 4.9 werden z. B. nur Zellen mit einem Druck zwischen  $1 \cdot 10^7$  Pa und  $3 \cdot 10^7$  Pa angezeigt. In Abhängigkeit vom Datenbereich werden die Zellen in der Render-Ansicht, sowie in der darunter angezeigten Datentabelle reduziert (Abb. 4.9 unten). Diese Ansicht ermöglicht eine schnelle Analyse relevanter Daten, sowohl visualisiert (Render-Ansicht) als auch in Tabellenform. Wenn in der Tabelle einzelne Zelldaten angeklickt werden, werden diese Zellen farblich in der Render-Ansicht hervorgehoben.



**Abb. 4.9** Render-Ansicht und Datentabelle bei aktiviertem Threshold-Filter für einen vorgegebenen Datenbereich (grauer Bereich liegt außerhalb des Datenbereiches)

Eine weitere Möglichkeit der Datenselektion kann über die Materialgebiete erfolgen. Unter *Properties* → *Scalars* → *<Materialgebiete>* werden über *Minimum* und *Maximum* die jeweilige Materialgebiete eingegeben. In der Render-Ansicht werden dann nur die entsprechenden Zellen des ausgewählten Materialgebietes angezeigt (Abb. 4.10). Unter *Display* → *Coloring* können die Variablen geändert werden, für die eine Einfärbung der Zellen erfolgen soll. Wie oben schon beschrieben, werden in der Datentabelle nur die Werte des ausgewählten Materialgebietes angezeigt.



**Abb. 4.10** Render-Ansicht mit aktiviertem Threshold-Filter zur Auswahl der Materialgebiete (grauer Bereich liegt außerhalb des Datenbereiches)

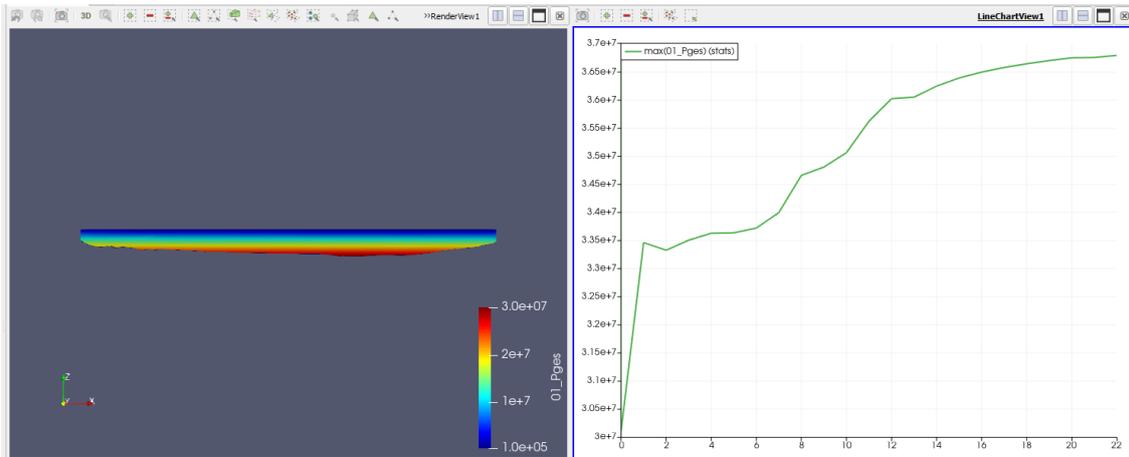
#### 4.15.2.2 Identifikation maximaler Zustände sowie Darstellung der zeitlichen Entwicklung als Kurvendiagramm

Eine häufige Aufgabe bei der Analyse von Simulationsergebnissen ist die Identifizierung maximaler Zustände und die Darstellung zeitlicher Entwicklungen in Kurvendiagrammen. Das erstellte Makro liest zunächst über einen Reader für unstrukturierte Gitter die erforderlichen Daten ein. Dann wird über *Find Selection* die Zelle gesucht, die den höchsten Druck aufweist. Die Zelle wird in der Render-Ansicht farblich hervorgehoben.

Für jeden Zeitschritt kann der maximale Zustand bzw. die Zelle, in der dieser Zustand auftritt, unterschiedlich sein. Wenn also ParaView den zeitlichen Verlauf der maximalen Zustände als Linienplot darstellt, kann sich dieser Plot bei jedem Zeitschritt auf eine andere Zelle beziehen. Wenn zu einem bestimmten Zeitpunkt eine Zelle mit einem Maximalwert identifiziert wurde und dann zu jedem Zeitschritt der Zustand genau dieser Zelle dargestellt werden soll, muss die Option zum Einfrieren der Auswahl (*Freeze Selection*) gewählt werden (siehe auch /MOR 18/).

In dem hier erstellten Makro wurde beispielhaft nach dem maximalen Druck für den ersten Zeitschritt gesucht (Abb. 4.11 links). Anhand der ermittelten Zelle wurde ein Linienplot erstellt, der die zeitliche Druckentwicklung für diese Zelle zeigt (Abb. 4.11 rechts).

Optional könnte, wie in Abb. 4.9 bzw. Abb. 4.10, zusätzlich noch die Datentabelle unterhalb der Ansicht gezeigt werden.

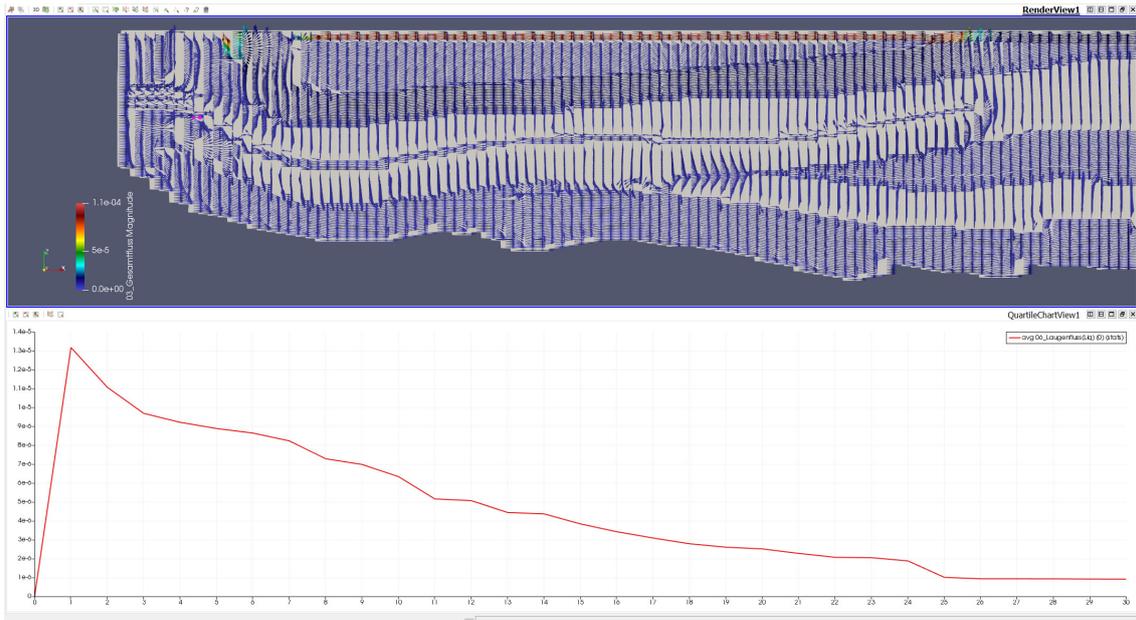


**Abb. 4.11** Render-Ansicht mit einem Linienplot der zeitlichen Entwicklung des Druckes für die Zelle, in der zum Zeitpunkt Null der maximale Druck gemessen wurde

#### 4.15.2.3 Vektordarstellung von Flüssen

Eine weitere Möglichkeit zur Visualisierung von Daten in ParaView ist die Darstellung von Fluss- oder Geschwindigkeitsvektoren durch Pfeile. In ParaView wird hierfür der *Glyph Filter* angewendet. Zur Optimierung der Darstellung können verschiedene Einstellungen erfolgen. So können die Vektoren bzw. deren Dimensionen (Pfeillängen) durch Skalar- oder Vektordaten skaliert werden. Die Anzahl der Vektoren kann für eine schnellere und übersichtlichere Darstellung durch die Auswahlmöglichkeiten *All Points*, *Every Nth Point*, oder *Uniform spatial Distribution* reduziert werden.

Das Makro liest zunächst über einen Reader für unstrukturierte Gitter die erforderlichen Vektordaten ein und visualisiert die Strömungsvektoren (Render-Ansicht, Abb. 4.12 oben). Dann wird über die Funktion *Find Selection* die Zelle bzw. Zellverbindung gesucht, welche die höchste Strömungsgeschwindigkeit aufweist. Die Zellenauswahl ist wie in Abb. 4.11 ebenfalls abhängig vom gewählten Zeitschritt. Dann wird ein Linienplot mit der zeitlichen Entwicklung des maximalen Geschwindigkeitsvektors erstellt (Abb. 4.12 unten).

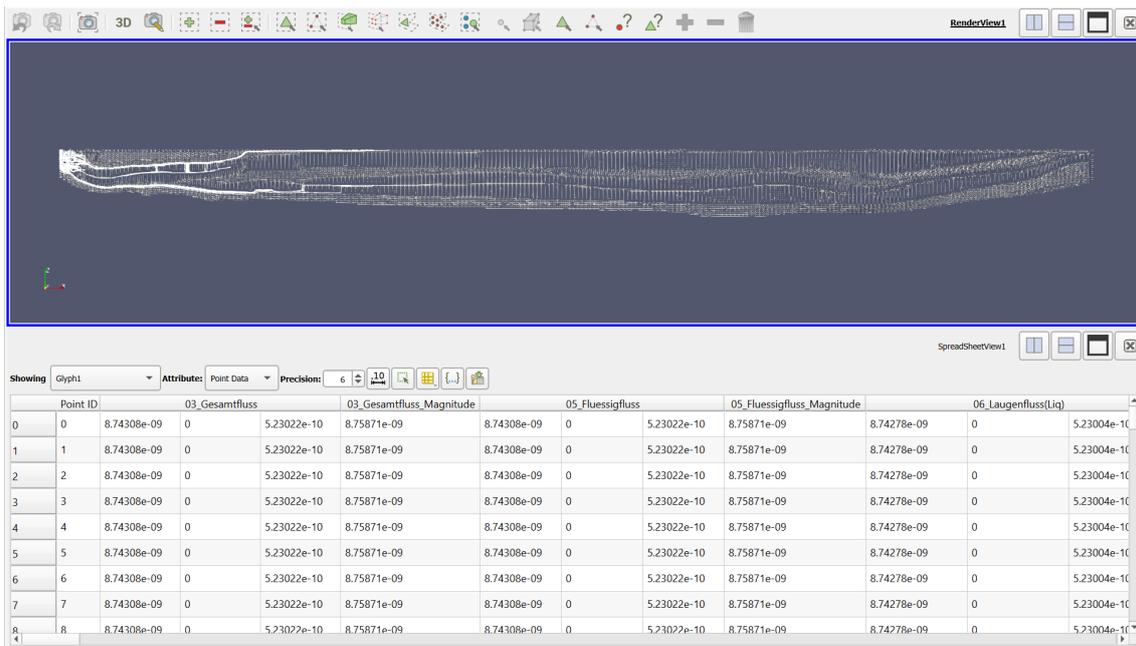


**Abb. 4.12** Render-Ansicht mit Vektoren und Linienplot für die zeitliche Entwicklung der Strömungsgeschwindigkeit in der Elementzelle mit der maximalen Geschwindigkeit

#### 4.15.2.4 Stromliniendarstellung

Einer Vektoransicht können zudem Stromlinien hinzugefügt werden (Abb. 4.13 oben). Die Stromlinien werden gebildet, indem zunächst Ausgangspositionen für virtuelle Tracer bestimmt werden, die dann entlang der Strömungsvektoren weiterverfolgt werden und somit einen Strömungspfad (Stromlinie) bilden. Die Ausgangspositionen der Tracer können entweder durch eine Linie oder durch eine Sphäre festgelegt werden. In Abb. 4.13 wurde am linken Rand eine Sphäre angenommen. Von dieser Sphäre ausgehend werden die abhängigen Vektoren als Stromlinien verfolgt.

Das Problem bei der Automatisierung der Darstellung von Stromlinien ist, dass der Bereich, für den Stromlinien erstellt werden sollen, für jede Geometrie unterschiedlich sein kann. Dies ist insbesondere schwierig für Netzwerksysteme, wie Endlagerbergwerke. Hier muss die Auswahl der Strecken, für die Stromlinien erstellt werden sollen, sehr individuell erfolgen. Deswegen erscheint eine Automatisierung der Darstellung von Stromlinien durch Makros nicht zielführend.



**Abb. 4.13** Darstellung der Stromlinien für Vektoren, ausgehend von einem sphärischen Bereich am linken Modellrand

### 4.15.3 Weiterführende Arbeiten

Die in den vorherigen Kapiteln erstellten Ansichtsvorlagen zeigen das große Potential von ParaView, Daten zu analysieren und Visualisierungen zu automatisieren. Die erstellten Visualisierungsvorlagen zeigen nur einige Beispiele für gängige Visualisierungen, schöpfen jedoch bei weitem nicht das große Potential von ParaView aus, Datenanalysen zu bewerkstelligen. Die Python-Schnittstelle zur Erstellung von Makros wird fortlaufend weiterentwickelt und bietet viele Möglichkeiten zur Steuerung von ParaView.

Die automatische Makroerstellung besitzt jedoch noch gewisse Nachteile. Ein Problem ist die geringe Flexibilität der mit dem Trace-Recorder erstellten Makros. Wenn zur Erstellung von Darstellungsvorlagen der Trace-Recorder verwendet wird, ist das resultierende Python-Skript nur für dieses spezielle Modellgitter und dessen räumlichen Dimensionen anwendbar. Der Trace-Recorder bzw. die Python-Schnittstelle greift auf die C++-Funktionen von ParaView (VTK) zu. Die C++-Funktionen ermitteln dann zum Beispiel die Gitterdimensionen und berechnen den Mittelpunkt des Modellgitters, um die Achsenansicht richtig einzustellen. Somit enthält das Python-Skript modellspezifische Dimensionen, was eine Anwendung des Skriptes auf Modellgitter mit anderen Dimensionen nicht mehr möglich macht. Günstiger wäre es, wenn durch die Python-Schnittstelle nur der Funktionsaufruf der entsprechenden C++-Funktionen erfolgen würde und die

modellspezifischen Informationen nicht in das Pythonskript geschrieben würden. Zur Behebung des Problems gibt es zwei Vorgehensweisen:

- Die Ermittlung der Modelldimensionen bevor das eigentliche Python-Skript bzw. Makro aufgerufen wird.
- Ein direkter Zugriff auf die C++-Funktionen in ParaView. Hierzu müsste voraussichtlich ein Plugin programmiert werden. Es muss noch geprüft werden, ob u. U. auch ein direkter Zugriff auf die C++-Funktionen durch die Pythonschnittstelle möglich ist.

Weiterführende Arbeiten müssen sich deshalb mit der Flexibilisierung der Makroerstellung befassen. Aus den oben genannten Gründen kann die automatische Erstellung von Makros durch den Makrorekorder nur dazu dienen, eine Makrogrundlage zu erstellen. Basierend auf dieser Grundlage muss der Code für die erforderlichen Visualisierungen adaptiert werden, um z. B. die Modelldimensionen zu ermitteln und Benutzerabfragen zu tätigen.

Es wäre zudem hilfreich, wenn eine größere Flexibilität bei der automatischen Erstellung verschiedener Ansichten durch Makros möglich wäre. So könnten etwa die Anzahl der Ansichtsfenster (Renderansichten) und die darin angezeigten Parameter nach dem Aufruf eines Makros ausgewählt werden. Das Gleiche gilt für die Gegenüberstellung einer Render-Ansicht und eines Linienplots oder Histogramms. Hier könnte vor der Erstellung von Plotlinien eine Auswahl einzelner Zellen und der entsprechenden Ergebnisparameter erfolgen.

#### **4.16      Fazit**

Der im Vorhaben entwickelte Gittergenerator für unstrukturierte Voronoi-Gitter besitzt zum Ende des Vorhabens einen fortgeschrittenen Entwicklungsstand. Die Abläufe zur Erstellung der Eingabedaten aus geologischen Daten aus Geoinformationssystemen sowie mögliche Visualisierungsmethoden wurden aufgezeigt. Im Detail sind noch verschiedene Erweiterungen des Gittergenerators notwendig, wie sie in Kapitel 4.7 aufgezeigt wurden. Auch muss der Gittergenerator intensiveren Tests unterzogen werden.

## **5 Verwendung von temperaturabhängigen thermodynamischen Daten in TOUGHREACT (AP 4)**

*Torben Weyand, Holger Seher, Guido Bracke*

### **5.1 TOUGH-Codes und THEREDA**

Der Code TOUGH2 /PRU 99/ wird von der GRS zur Berechnung des mehrphasigen Stofftransports innerhalb von Endlagersystemen benutzt und als modifizierter Code TOUGH2-GRS bzw. TOUGH2-MP-GRS qualitätskontrolliert weiterentwickelt (/NAV 13/, /NAV 16/, /NAV 18b/ sowie aktuelle Arbeiten in diesem Vorhaben).

Der Code TOUGHREACT /XU 06/ aus der TOUGH-Familie erweitert TOUGH2 um die thermodynamische Modellierung geochemischer Prozesse. Diese Prozesse sind die aquatische Speziation und Komplexbildung, Adsorption, Gaslösung und Ausgasung, Mineralauflösung und -ausfällung sowie Kationenaustausch. Die Mineralauflösung und -ausfällung wird in der Regel unter der Annahme berechnet, dass sich abhängig von den Ionenaktivitäten ein lokales Gleichgewicht einstellt. Für einige dieser Prozesse kann eine Reaktionskinetik für die Gleichgewichtseinstellung berücksichtigt werden und mit der Änderung der Porosität, Permeabilität und des Kapillardrucks verbunden werden. Druck- und Temperaturänderungen finden ebenfalls Eingang in die Berechnungen der Gleichgewichte.

Für die thermodynamische Modellierung von hochsalinaren Lösungen – wie sie auch in Endlagersystemen in Deutschland erwartet werden – müssen die starken Wechselwirkungen der Ionen in der flüssigen Phase berücksichtigt werden /SEH 17/. Die notwendige Berechnung der Ionenaktivitäten kann nach dem Pitzer-Ansatz /PIT 91/ erfolgen.

Die aktuelle Programmversion TOUGHREACT v4 wurde vom Lawrence Berkeley National Laboratory (LBNL) um den Pitzer-Ansatz, wie er bisher nur in TOUGHREACT v1.21 /ZHA 06/ zur Verfügung stand, ergänzt. Eine Testversion von TOUGHREACT v4 wurde der GRS zur internen Verwendung zur Verfügung gestellt, da zum Zeitpunkt der Bearbeitung dieses Arbeitspaketes noch keine veröffentlichte Version des Codes existierte. Basis dieser TOUGHREACT-Version ist die Version v3.32 /XU 17/. Mit der Vorgängerversion TOUGHREACT v1.21 /ZHA 06/ sind in den Arbeiten von /SEH 17/ numerische Probleme bei den Modellierungen mit TOUGHREACT mit temperaturabhängigen,

thermodynamischen Datensätzen aus THEREDA bei der Berücksichtigung von zweiwertigen Ionen für Temperaturen, abweichend von 25 °C, identifiziert worden. Bei der Berechnung von Benchmark-Tests (Gleichgewichtsberechnung zwischen einer hochsalinaren Lösung mit zweiwertigen Ionen und Festphasen) traten bei hohen Temperaturen numerische Oszillationen auf, die zu einer deutlichen Abweichung zwischen den berechneten Konzentrationen aquatischer Spezies mit TOUGHREACT und validierten Referenzdaten führten. In der aktuellen Version TOUGHREACT v4 traten diese Probleme nicht wieder auf.

TOUGHREACT wurde im Projekt „*Übergangsphase*“ (FKZ: 3610R03230) zur Modellierung des reaktiven Stofftransports innerhalb der Bentonitbarriere eines Endlagers im Tonstein verwendet /BRA 14/. Im Vorhaben „*Forschung und Entwicklung zu Methoden und Instrumenten des Langzeitsicherheitsnachweises*“ (EMIL, FKZ: 3614R03200) wurde die Bildung von sekundären Eisenphasen in verschiedenen Laborexperimenten nachvollzogen /SEH 17/. In dem aktuell in Bearbeitung befindlichen Vorhaben „*Modellierung relevanter Prozesse und möglicher Endlagerstandorte zur Berechnung des Transports von Radionukliden aus einem Endlager*“ (FKZ: 4719E03230) soll der reaktive zweiphasige Stofftransport von <sup>14</sup>C mit TOUGHREACT v4 modelliert werden /WEY 20/.

Die für die Berechnungen der Ionenwechselwirkungen im Pitzer-Ansatz benötigten Parameter wurden im Vorhaben ***Thermodynamische Referenz-Datenbasis*** (THEREDA) /MOO 15/ qualitätsgesichert erstellt und in einer konsistenten thermodynamischen Datenbasis den Nutzern zur Verfügung gestellt. THEREDA stellt vorformatierte Datensätze für die Programme PHREEQC, ChemApp, Geochemist's Workbench, EQ 3/6 und im JSON-Format zur Verfügung.

Im Vorhaben EMIL wurde der Quellcode von TOUGHREACT v1.21 dahingehend erweitert, dass zwei in TOUGHREACT v1.21 fehlende Koeffizienten für die Berechnung der temperaturabhängigen Pitzer-Parameter, die in den Datensätzen von THEREDA vorhanden sind, berücksichtigt werden können /SEH 17/. Die entsprechenden Änderungen an den verwendeten Funktionen zur Abbildung der Temperaturabhängigkeit der log K-Werte und der Pitzer-Parameter sind in Kap. 5.3.2 näher beschrieben. Da das Format der Eingabedateien von TOUGHREACT v1.21 nicht von THEREDA unterstützt wird, wurde im Vorhaben EMIL ein Konverter entwickelt, der die Parameterwerte aus dem THEREDA-JSON-Format in das Format für TOUGHREACT v1.21 konvertiert. Benchmark-Rechnungen der konvertierten Eingabedateien mit TOUGHREACT v1.21 zeigten im Vergleich zu Simulationsergebnissen mit dem Code PHREEQC eine sehr gute

Übereinstimmung /SEH 17/. Die genannten Entwicklungen wurden vom LBNL bei der Entwicklung von TOUGHREACT v4 umgesetzt.

TOUGHREACT v4 verwendet für die Eingabe der thermodynamischen Daten im Vergleich zu TOUGHREACT v1.21 nur noch eine einzelne Eingabedatei. Darin enthalten sind die Parameter für die temperaturabhängige Berechnung der log K-Werte sowie die Pitzer-Parameter. Das im Vorhaben EMIL entwickelte Programm zur Konvertierung der THEREDA-Eingabedaten in das Eingabeformat von TOUGHREACT v1.21 wurde daher im vorliegenden Vorhaben dahingehend umgeschrieben, dass die Ausgabe nur noch in eine Datei erfolgt, die mit TOUGHREACT v4 eingelesen werden kann. Mit diesem Konverter wurden mehrere thermodynamische Datenbasen aus THEREDA in ein mit TOUGHREACT v4 lesbares Dateiformat konvertiert, um die ab Kap. 5.3 beschriebenen Benchmark-Tests durchzuführen. Diese sollen verifizieren, dass der Konverter ein für TOUGHREACT-BRINE lesbares Datenformat generiert und die Benchmark-Tests mit TOUGHREACT-BRINE die gleichen Ergebnisse erzielen wie mit PHREEQC.

## 5.2 Benchmark-Tests und Codes

Die Parameterdateien<sup>4</sup> von THEREDA wurden mit dem im Vorhaben EMIL entwickelten und im Rahmen dieses Vorhabens weiterentwickelten Konverter-Tool `convert.pl` in ein durch TOUGHREACT lesbares Format gebracht. Um die korrekte Funktion dieses Konverter-Tools und somit dessen Qualität abzusichern, erfolgten sogenannte Benchmark-Tests für ausgewählte Parameterdateien unterschiedlicher geochemischer Systeme. In den Benchmark-Tests werden ausschließlich Gleichgewichtsreaktionen innerhalb eines Gitterelementes und keine Transportrechnungen mit mehreren Gitterelementen durchgeführt. Die geochemischen Systeme wurden so gewählt, dass mehrere Kombinationen der Pitzer-Parameter abgedeckt wurden. Hierzu wurden die in THEREDA existierenden Benchmark-Tests verwendet, die mit den geochemischen Codes CHEMAPP /ERI 03/, PHREEQC /PAR 13/, Geochemist's Workbench (GWB) /BET 15/ und EQ3/6 /WOL 03/ qualitätsgesichert sind.

---

<sup>4</sup> Eine Parameterdatei enthält die thermodynamischen Daten (u. a. log K-Werte, Pitzer-Koeffizienten) für das geochemische System des entsprechenden Releases von THEREDA.

Benchmark-Tests wurden zu den geochemischen Systemen:

- K, Mg, Ca - Cl, SO<sub>4</sub> - H<sub>2</sub>O(l) (polytherm) /MOO 11a/ (THEREDA-Release 01),
- Am(III), Cm(III), Nd(III) - Na, Mg, Ca - Cl, OH - H<sub>2</sub>O(l) (25°C) /ALT 12a/, /ALT 12b/ (THEREDA-Release 02),
- Und Na, K, Mg, Ca - Cl, SO<sub>4</sub> - HCO<sub>3</sub>/CO<sub>2</sub>(g) - H<sub>2</sub>O(l) /VOI 12/ (THEREDA-Release 03)

durchgeführt. Parameterdatei 01 beschreibt das geochemische System der ozeanischen Salze und besitzt temperaturabhängige Gleichgewichtskonstanten sowie Pitzer-Parameter. Parameterdatei 02 fokussiert auf die Bildung und Löslichkeit von Aktiniden und Parameterdatei 03 kann für den Radionuklidtransport von Kohlenstoff (z. B. <sup>14</sup>C) in Lösung und Gasphase eingesetzt werden. Anschließend wurden die Benchmark-Tests mit dem Code TOUGHREACT v4 nachgerechnet und die Ergebnisse zwischen den eigenen Berechnungen mit PHREEQC und TOUGHREACT v4 verglichen.

Im Folgenden werden die Rechenfälle (Benchmark-Tests) und deren Eingabe in PHREEQC und TOUGHREACT v4 beschrieben.

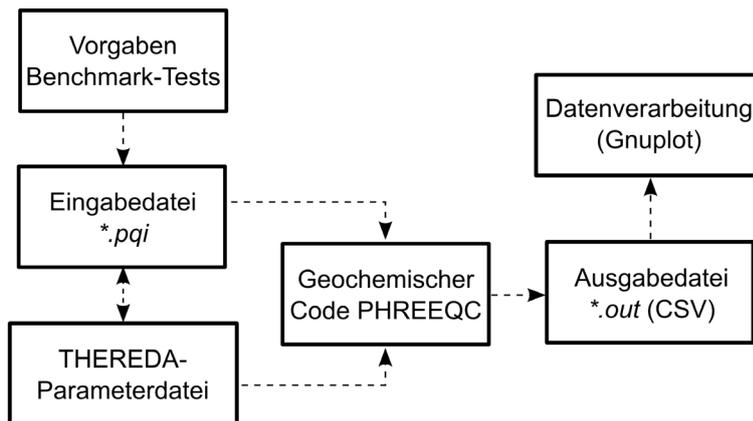
### 5.2.1 PHREEQC

Zur Verifizierung des entwickelten Konverter-Tools `convert.pl` werden die Ergebnisse der Benchmark-Tests mit TOUGHREACT v4 mit Ergebnissen der Benchmark-Tests mit PHREEQC verglichen. Für die Berechnung der Benchmark-Tests mit PHREEQC wird die Programmversion PHREEQC 3.5.0 /PAR 19/ eingesetzt.

PHREEQC ermittelt den thermodynamisch stabilsten Zustand aquatischer Systeme im Gleichgewicht mit Festphasen über alle vorgegebenen Reaktionen des Systems anhand von Gleichgewichtskonstanten /MER 08/. PHREEQC benötigt eine Eingabedatei mit Simulationsparametern sowie eine Eingabedatei mit thermodynamischen Daten. Zur Ergebnisdarstellung mit Gnuplot in der Version 5.2 /WIL 18/ wird eine Ausgabedatei (CSV-Format)<sup>5</sup> generiert (Abb. 5.1).

---

<sup>5</sup> Verwendet wird der Ausgabebefehl `SELECTED_OUTPUT`



**Abb. 5.1** Workflow zur Durchführung der Benchmark-Tests mit PHREEQC

Die Eingabedateien für PHREEQC werden entsprechend der Vorgaben der Benchmark-Tests von THEREDA erstellt (Release 01: /MOO 11a/; Release 02: /ALT 12a/, /ALT 12b/; Release 03: /VOI 12/). Die Vorgaben umfassen

- die initiale Lösungszusammensetzung,
- die in das Gleichgewicht mit der Lösung zu setzenden Mineral- und Gasphasen
- und die physikalisch-chemischen Parameter der Lösung (Temperatur, Druck, pH-Wert, Redoxpotential).

Ein Beispiel der Eingabedatei und eine kurze Beschreibung der verwendeten Funktionen aus PHREEQC (sog. Keywords) werden im Folgenden gegeben. Eine detaillierte Beschreibung der PHREEQC-Kywords befindet sich in /PAR 13/.

Mit dem PHREEQC-Kyword `SOLUTION` wird die initiale Lösungszusammensetzung (hier beispielsweise für die Ionen  $K^+$ ,  $Mg^{2+}$  und  $SO_4^{2-}$ ) mit Angabe der Stoffmenge pro Wassermasse (in  $mmol/kg_{H_2O}$ ), die sich aus den Vorgaben zu den Benchmark-Tests ergibt. Weiterhin werden die Lösungsmenge (in kg Wasser), die Dichte der Lösung (in  $g/m^3$ ) sowie physikalisch-chemische Parameter (Temperatur, pH, pe) festgelegt.

Das PHREEQC-Kyword `EQUILIBRIUM_PHASES` setzt die unter `SOLUTION` definierte Lösung in Sättigung (d. h. hier ist der Sättigungsindex<sup>6</sup> (SI) gleich 0 /MER 08/) mit den

<sup>6</sup> Der Sättigungsindex ist der Quotient aus dem Ionenaktivitätsprodukt (Molalitäten/Aktivitäten der an einer Reaktion beteiligten Ionen in Lösung) und dem Löslichkeitsprodukt (entsprechend der in der thermodynamischen Datenbasis definierten Reaktion).

Mineral- und Gasphasen (hier: Epsomit und Pikromerit), sodass ein stabiles thermodynamisches Gleichgewicht erzielt wird. Die zweite Eingabe nach dem Sättigungsindex gibt die zur Verfügung stehende Stoffmenge der entsprechenden Mineral- oder Gasphase an. Diese wird in den Benchmark-Tests ausreichend hoch definiert, dass für ein Kilogramm Wasser eine Gleichgewichtssättigung erzielt wird (hier 10 mol).

Das PHREEQC-Keyword `PITZER` definiert spezifische Einstellungen des verwendeten Pitzer-Ansatzes zur Berücksichtigung der interionaren Wechselwirkungen in hochsalinaren Lösungen (Abb. 5.2). Die Einstellungen werden entsprechend der Vorgaben aus THEREDA übernommen. Eine Beschreibung der Einstellungsoptionen befindet sich in /PAR 13/, die Begründung der gewählten Einstellungsoptionen in den jeweiligen Beschreibungen zu den Releases und Benchmark-Tests (Release 01: /MOO 11a/, Release 02: /ALT 12b/, /ALT 12a/, Release 03: /VOI 12/).

```
PITZER
-MacInnes    false
-use_etheta  true
-redox       false
```

**Abb. 5.2** PHREEQC-Keyword `PITZER`

Mit dem PHREEQC-Keyword `SELECTED_OUTPUT` wird eine \*.csv-Datei generiert, die Grundlage für die Auswertung der Ergebnisse mit Gnuplot und den Vergleich der Simulationsergebnisse mit TOUGHREACT v4 ist. Die Befehle in `SELECTED_OUTPUT` (`true`, `false`) werden so gesetzt, dass nur die Informationen exportiert werden, die auch bei den THEREDA-Benchmarks miteinander verglichen werden. Hierzu gehören pH-Wert, Molalitäten der aquatischen Spezies und Gesamtkonzentrationen der betrachteten Ionen.

### 5.2.2 TOUGHREACT

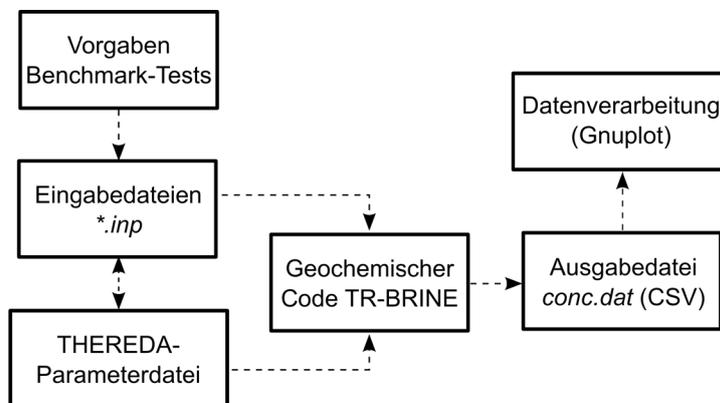
Verwendet wird die Codeversion TOUGHREACT v4, die noch nicht veröffentlicht ist und von den Entwicklern des LBNL der GRS als ein Executable mit dem EOS-Modul<sup>7</sup> EOS1 zu Testzwecken zur Verfügung gestellt wurde. Diese Codeversion wird verwendet, da in dieser Codeversion der Pitzer-Ansatz zur Berechnung der Wechselwirkungen von Ionen in hochsalinaren Lösungen /PIT 91/ aus TOUGHREACT v1.21 /ZHA 06/ in die neuere

---

<sup>7</sup> Equation of State /PRU 99/.

TOUGHREACT v3.2 /XU 17/ implementiert wurde. Dies erlaubt es, die überarbeiteten Funktionalitäten der TOUGHREACT v3.2 (z. B. bezüglich erweiterter temperaturabhängiger Prozesse) in Kombination mit dem Pitzer-Ansatz für hochsalinare Lösungen zu nutzen. Dies ist zur Berechnung der Benchmark-Tests und einen Vergleich der Ergebnisse aus TOUGHREACT mit PHREEQC notwendig, da PHREEQC ebenfalls den Pitzer-Ansatz berücksichtigt.

Die Eingabe der Parameter erfolgt für TOUGHREACT über mehrere Eingabe-Dateien: `flow.inp`, `solute.inp` und `chemical.inp` /XU 06/ sowie eine Datei mit den thermodynamischen Daten. Der Workflow zur Durchführung der Benchmark-Tests mit TOUGHREACT ist in Abb. 5.3 dargestellt.



**Abb. 5.3** Workflow zur Durchführung der Benchmark-Tests mit TOUGHREACT

Die Eingabedatei `flow.inp` beinhaltet transport- und strömungsrelevante Parameter und Informationen zum Gittermodell. Da in den Benchmark-Tests ausschließlich Gleichgewichtsberechnungen zwischen Lösung, Feststoff- und Gasphase durchgeführt werden, sind in `flow.inp` nur wenige Einträge relevant wie z. B. Feststoffvolumen, Sättigung, Temperatur, Druck. Die transportrelevanten Parameter sind vernachlässigbar. Eine vollständige Auflistung der Einstellungen wird im entsprechenden Unterkapitel zu `flow.inp` beschrieben. Modellrelevant sind weiterhin die Einstellungsoptionen zum Pitzer-Ansatz und zur Berechnung des thermodynamischen Gleichgewichtes, die über sog. MOPR-Schalter geregelt werden.

`solute.inp` beinhaltet allgemeine Informationen z. B. zu den Randbedingungen des Modells, zur numerischen Simulation und Struktur der Ausgabedateien. Diese werden einheitlich für alle Benchmark-Tests verwendet.

Die Eingaben zur Lösungszusammensetzung, der physikalisch-chemischen Parameter und der ins Gleichgewicht zu setzenden Mineral- und Gasphasen erfolgen in der Eingabedatei `chemical.inp`. Diese wird spezifisch für jeden Rechenfall entsprechend der Vorgaben der Benchmark-Tests erstellt.

### **Executable mit dem EOS-Modul EOS1**

TOUGH2 kann den Transport verschiedener Phasen simulieren (i. d. R. Flüssigkeits- und Gasphase). Diese Phasen werden im Code als jeweils einzelne Komponenten, die advektiv und diffusiv transportiert werden, berechnet. Komponenten können auch durch Prozesse wie Kondensation, Evaporation, Diffusion, Lösung oder Entgasung ihre jeweilige Phase ändern. Die Festlegung der Phasen und Komponenten erfolgt in TOUGH in sogenannten EOS-Modulen. Die Ermittlung der Phasen- und Komponentenverteilung erfolgt über die Berechnung der thermodynamischen Gleichgewichte und das EOS-Modul stellt die abgeleiteten physikalischen Größen aus den Primärvariablen bereit.

Seitens der Entwickler des LBNL wurde ein Executable für das EOS-Modul EOS1 zur Verfügung gestellt, das geeignet und ausreichend für die Benchmark-Tests ist<sup>8</sup>.

### **Eingabedatei `flow.inp`**

Die Eingabedatei `flow.inp` legt die Gesteinsparameter, Simulationsparameter sowie Initial- und Randbedingungen fest /XU 06/.

In den Benchmark-Tests wird innerhalb eines Gitterelementes die Lösungszusammensetzung durch Herstellen des Gleichgewichtes zwischen Lösung und einer oder mehreren Mineralphasen berechnet. Die hierfür in TOUGHREACT relevanten Simulationsparameter werden im Eingabeblock `REACT` eingestellt.

Der Eingabeblock `REACT` besitzt 20 allgemeine /XU 17/ (MOPR 1 – 20 aus TOUGHREACT-v.3.32, s. /XU 17/) und vier pitzerspezifische (MOPR 21 – 24 aus TOUGHREACT v4, s. Anhang A.1) Einstellungen. Da die allgemeinen Einstellungen zu TOUGHREACT (MOPR 1 – 20) überwiegend für den reaktiven Stofftransport relevant

---

<sup>8</sup> Auch wenn in Release 03 die Gasphase  $\text{CO}_2(\text{g})$  betrachtet wird, ist EOS1 ausreichend, da keine Strömungsrechnung erfolgt, sondern nur die thermodynamische Gleichgewichtsreaktionen berechnet werden.

sind, werden hier ausschließlich die in der Codeversion TOUGHREACT v4 neu hinzugefügten MOPR (21 – 24) beschrieben /SPY 19/ (s. Anhang A.1):

- MOPR-21 bestimmt, ob das erweiterte Debye-Hückel-Modell (MOPR-21 = 0) oder der Pitzer-Ansatz (MOPR-21 = 1) berücksichtigt werden soll. Da in den Benchmark-Tests Gleichgewichte für hochsalinare Lösungen berechnet werden, wird der Pitzer-Ansatz verwendet.
- MOPR-22 ermöglicht es, die Berechnung von bestimmten ionaren Wechselwirkungen des Pitzer-Ansatzes auszuschalten, um Rechenzeit zu sparen (z. B. Neutral-Kation-Wechselwirkungen o. ä.). Da die Rechenzeit für die Durchführung der Benchmark-Tests nicht relevant ist und ein möglichst genaues Ergebnis erzielt werden soll, werden keine Vereinfachungen eingestellt (MOPR-22 = 0).
- MOPR-23 beschreibt die Berücksichtigung der sog. „mixing-terms“. Diese werden aktiviert, da ein möglichst genaues Ergebnis erzielt werden soll und die Rechenzeit der Benchmark-Tests sehr gering ist (MOPR-23 = 0).
- Mit MOPR-24 können zusätzliche Gleichungen für die Evaporation von Salzen auf den Dampfdruck berücksichtigt werden. Da dies für die Benchmark-Tests nicht relevant ist, wird diese Option deaktiviert (MOPR-24 = 0).

Weiterhin werden in der `flow.inp` die Eigenschaften des Gitterelementes definiert, in welchem die Gleichgewichtsreaktionen bestimmt werden. Hierzu gehören

- der Druck von 1 bar,
- die Temperatur (Release 01: 25 °C bis 100 °C; Release 02 + 03: 25 °C),
- die Lösungssättigung von 1,
- sowie die Porosität von 0,0625.

Druck und Temperatur entsprechen hierbei den Vorgaben der Benchmark-Tests. Die Lösungssättigung wird auf 1 gesetzt, um eine Vergleichbarkeit mit einphasigen geochemischen Codes (hier PHREEQC) zu ermöglichen. Die Porosität wird so gering angesetzt, dass die Feststoffmenge im Gitterelement im Verhältnis zur Lösungsmenge ausreichend ist, um ein thermodynamisches Gleichgewicht zu erzielen. Ist die Feststoffmenge im Verhältnis zur Lösungsmenge zu gering, kann u. U. nicht die notwendige Menge eines Minerals gelöst werden, um eine mit diesem Mineral gesättigte Lösung zu

berechnen. Testrechnungen haben ergeben, dass eine Porosität von 0,0625 ausreichend gering, d. h. die Feststoffmenge ausreichend groß ist.

Der Eingabeblock `MULTI` richtet sich nach dem gewählten EOS-Modul `EOS1`. Zu berücksichtigen ist, dass für die Benchmark-Tests die Temperatur eine Primärvariable und damit veränderlich ist<sup>9</sup>.

### **Eingabedatei `solute.inp`**

`solute.inp` beinhaltet Parameter für den reaktiven Transport wie Diffusionskoeffizienten und Toleranzgrenzen für die Iterationsverfahren sowie die Definierung der Ausgabedateien `/XU 17/`. Die Eingabedateien sind für alle Benchmarks mit Ausnahme der Anzahl der Spezies in der Ausgabedatei identisch. Relevant für die Benchmark-Tests sind die Einstellungsoptionen des Eingabeblocks „constraints of chemical solver“. Die minimale Lösungssättigung beträgt  $10^{-05}$  (`SL1MIN`) und die maximale Ionenstärke 44 mol/kg<sub>H<sub>2</sub>O</sub> (`STIMAX`).

Weiterhin werden in `solute.inp` die Dateinamen für die Parameterdatei mit thermodynamischen Daten sowie die für die Ausgabe benannt. Kriterien zur Konvergenz des Gleichungslösers werden aus `/XU 17/` übernommen. In die Ausgabedatei werden die Konzentrationen im \*.csv-Format (Tecplot `/XU 17/`) sämtlicher berechneten Spezies und Komponenten ausgegeben, um einen Vergleich mit anderen geochemischen Codes (hier PHREEQC) im Postprocessing zu ermöglichen.

### **Eingabedatei `chemical.inp`**

Die Eingabedatei `chemical.inp` definiert das geochemische System, z. B. Typ und Anzahl an aquatischen Spezies, Mineral- und Gasphasen (Abb. 5.4). Außerdem werden die geochemische Zusammensetzung initialer Lösungen sowie die sog. Mineral- und Gaszonen definiert. Bei einer Zone handelt es sich in TOUGHREACT um eine bestimmte Mineral- oder Gaszusammensetzung. Anschließend können in `solute.inp` diese Zonen einzelnen Gitterelementen zugeordnet werden.

---

<sup>9</sup> Gewählt wird die Einstellungskombination (NK, NEQ, NPH, NB) 2 3 2 6. Bei der Eingabe ist der Hinweis auf S. 31 in `/PRU 99/` zu beachten, dass bei einer Temperatur ab 100 °C die Reihenfolge der Primärvariablen (Druck, Temperatur) intern im Code getauscht wird.

```

THEREDA R01-01
#-----'
#'DEFINITION OF THE GEOCHEMICAL SYSTEM'
#'PRIMARY AQUEOUS SPECIES'
'h2o'          0
'h+'           0
'Mg+2'         0
'K+'           0
'SO4-2'        0
'**'           0

# AQUEOUS KINETICS
*

# AQUEOUS COMPLEXES
*

#'MINERALS'
'Mg(SO4):7H2O(cr)'  0 0 0 0
0. 0. 0.
'K2Mg(SO4)2:6H2O(cr)'  0 0 0 0
0. 0. 0.
'**'

#'GASES'
'**'

#'SURFACE COMPLEXES'
'**'

#'species with Kd and decay      decay constant(1/s)'
'**'

#'EXCHANGEABLE CATIONS'
#          master      convention      ex. coef.'
'**'

#-----'
#'INITIAL AND BOUNDARY WATER TYPES'
1 0 0 !niwtype, nbwtype = number of initial and boundary waters
1 25. 0 !iwtype initial, temp (C)
#          icon          guess          ctot          constrain'
'h2o'      1          1.000e+00      1.0000e+00      ' ' 0
'h+'       3          1.0000E-07      1.0000E-07      ' ' 0
'K+'       1          1.000e-10      1.000e-10      ' ' 0
'Mg+2'     1          1.000e-10      1.000e-10      ' ' 0
'SO4-2'    1          1.000e-10      1.000e-10      ' ' 0
'**'       0          0.0          0.0          ' ' 0

#-----'
#'INITIAL MINERAL ZONES'
1
1
# 'mineral          vol.frac.'          !MATRIX
Mg(SO4):7H2O(cr)    0.5 0
K2Mg(SO4)2:6H2O(cr) 0.5 0
'**'                0.0 0

#-----'
#'INITIAL gas ZONES'
'**'                0.0

```

**Abb. 5.4** Ausschnitt aus einer exemplarischen Eingabedatei von chemical.inp (THEREDA-Parameterdatei 01, Benchmark-Test 01)

Unter `PRIMARY AQUEOUS SPECIES` werden die primären Masterspezies (i. d. R. Ionen wie  $\text{Na}^+$ ,  $\text{Cl}^-$  usw.) definiert, aus denen weitere aquatische Spezies sowie Mineral- und Gasphasen im berücksichtigten System gebildet werden. Es werden alle sekundären aquatischen Spezies aus der thermodynamischen Datenbasis berücksichtigt, die aus den angegebenen primären Masterspezies gebildet werden können. Die berücksichtigten sekundären aquatischen Spezies werden in der Ausgabedatei `chemical.out` genannt.

Unter `MINERALS` werden alle Festphasen definiert, die die Simulation berücksichtigen soll. In diesem Eingabeblock können Parameterwerte zur Beschreibung der Prozesskinetik definiert werden. Die Benchmark-Tests berücksichtigen keine Reaktionskinetik. Um die Gleichgewichtsreaktionen instantan erfolgen zu lassen, werden diese Werte auf Null gesetzt.

Unter `GASES` werden alle Gasphasen definiert, die die Simulation berücksichtigt. Dies betrifft die Parameterdatei-03 von THEREDA zum geochemischen System mit  $\text{CO}_2(\text{g})$ .

Weiterhin werden in `chemical.inp` Mineral- und Gaszonen definiert, die als Randbedingung in `solute.inp` dem Gitterelement zugeordnet werden. Für die Benchmark-Tests wird für das betrachtete Gitterelement, entsprechend der jeweiligen Vorgaben, die Mineralzusammensetzung des Feststoffs angegeben. Diese wird über einen Anteil am Feststoffvolumen des Gitterelementes an den jeweils an den Gleichgewichtsreaktionen Mineralphasen angegeben.

Unter `INITIAL AND BOUNDARY WATER TYPES` wird die Zusammensetzung der Lösung angegeben. In den Benchmark-Tests wird jeweils nur eine Lösungszusammensetzung angegeben, die in TOUGHREACT als Wasserzone (`INITIAL WATER TYPE`) abgebildet wird. Für jede Wasserzone wird die Temperatur definiert (Variable `TEMP` in  $^{\circ}\text{C}$ ), die sich aus den Vorgaben der Benchmark-Tests ergibt. Für jede primäre Spezies wird hier die Molalität (`icon = 1`) bzw. im Falle der  $\text{H}^+$ -Konzentration die Aktivität (`icon = 3`) angegeben, da sich diese aus dem pH-Wert ergibt. Weitere Details zur Struktur der Eingabedatei `chemical.inp` sind in /XU 17/ beschrieben.

### **5.3 Benchmark-01 (System ozeanischer Salze)**

#### **5.3.1 Vorgaben**

Die Parameterdatei 01 der Datenbank THEREDA ist eine thermodynamische Datenbasis für das System der ozeanischen Salze Na-K-Mg-Ca-Cl-SO<sub>4</sub>-H<sub>2</sub>O(l). Die thermodynamischen Daten (log K-Werte und Pitzer-Parameter) sind temperaturabhängig bis maximal 250 °C angegeben (die konkrete Temperatur ergibt sich aus den berücksichtigten Ionen, Spezies und Mineralphasen<sup>10</sup>).

Die Eingabeparameter aus /MOO 11a/ werden für die Vergleichbarkeit der numerischen Ergebnisse für die Benchmark-Tests mit TOUGHREACT übernommen. Hierzu gehören die geochemische Lösungszusammensetzung, die Auswahl der zu berücksichtigenden Festphasen sowie die physikalisch-chemischen Parameter (Temperatur, Druck, pH-Wert). Aus der nachfolgenden Beschreibung der geochemischen Zusammensetzung, inkl. variierender Ionenstärke und berücksichtigter Festphasen (in Abhängigkeit der Temperatur sind unterschiedliche Mineralphasen thermodynamisch stabil), ergeben sich acht Testfälle, die jeweils für zwei bis drei unterschiedliche Temperaturen gerechnet werden. Insgesamt sind es 21 Kombinationen (Tab. 5.1).

---

<sup>10</sup> Für weitere Details s. Release Notes /MOO 11a/ zur Parameterdatei 01.

**Tab. 5.1** Eingabeparameter der Benchmark-Tests

Die „Benchmark-Test-Nr.“ entspricht der fortlaufenden Nummerierung der Benchmark-Tests der Parameterdatei /MOO 11a/. Derzeit werden im Vorhaben BGE-45181017 „Fortführung und Weiterentwicklung der thermodynamischen Referenzdatenbasis (THEREDA)“ die einzelnen Parameterdateien in ein umfassendes einzelnes Release zusammengeführt. Entsprechend ändert sich zukünftig die Nummerierung der Benchmark-Tests, die hier unter der Spalte „THEREDA-Test-Nr.“ aufgeführt ist, um eine bessere Zuordnung zu ermöglichen.

Benchmark	THEREDA-Test-Nr.	Primäre Spezies	Festphasen	T in °C
1	001	$K^+, Mg^{2+}, SO_4^{2-}$	Epsomit + Pikromerit	25
2	002	$K^+, Mg^{2+}, SO_4^{2-}$	Hexahydrit + Leonit	50
3	003	$K^+, Mg^{2+}, SO_4^{2-}$	Kieserit + Langbeinit	100
4	004	$K^+, Mg^{2+}, SO_4^{2-}$	Arkanit + Pikromerit	25
5	005	$K^+, Mg^{2+}, SO_4^{2-}$	Arkanit + Leonit	50
6	006	$K^+, Mg^{2+}, SO_4^{2-}$	Arkanit + Langbeinit	100
7	007	$K^+, Na^+, SO_4^{2-}$	Arkanit + Glaserit	25
8	008	$K^+, Na^+, SO_4^{2-}$	Arkanit + Glaserit	50
9	009	$K^+, Na^+, SO_4^{2-}$	Arkanit + Glaserit	100
10	010	$K^+, Na^+, SO_4^{2-}$ -	Glaserit + Mirabilit	25
11	011	$K^+, Na^+, SO_4^{2-}$	Glaserit + Thenardit	50
12	012	$K^+, Na^+, SO_4^{2-}$	Glaserit + Thenardit	100
13	013	$Ca^{2+}, Cl^-$	$Ca_4Cl_2(OH)_6 \cdot 13H_2O(cr)$ + Portlandit	25
14	014	$Ca^{2+}, Cl^-$	$Ca_2Cl_2(OH)_2 \cdot H_2O(cr)$ + Portlandit	50
15	015	$Ca^{2+}, Cl^-$	$Ca_2Cl_2(OH)_2 \cdot H_2O(cr)$ + Portlandit	100
16	016	$K^+, SO_4^{2-}$	Arkanit + $K_3(HSO_4)(SO_4)(cr)$	25
17	017	$K^+, SO_4^{2-}$	Arkanit + $K_3(HSO_4)(SO_4)(cr)$	50
18	018	$K^+, SO_4^{2-}$	$K_3(HSO_4)(SO_4)(cr)$ + Misenit	25
19	019	$K^+, SO_4^{2-}$	$K_3(HSO_4)(SO_4)(cr)$ + Misenit	50
20	020	$K^+, SO_4^{2-}$	Mercallit + Misenit	25
21	021	$K^+, SO_4^{2-}$	Mercallit + Misenit	50

Die Benchmark-Tests mit der Parameterdatei 01 lassen sich in vier unterschiedliche geochemische Systeme unterteilen:

- Sechs für das System K-Mg-SO<sub>4</sub>,
- sechs für K-Na-SO<sub>4</sub>,
- drei für Ca-Cl und
- sechs für K-SO<sub>4</sub>.

Das Gleichgewicht zwischen destilliertem Wasser mit den Mineralphasen (SI = 0) wird für das entsprechende System für die Temperaturspanne  $T = 25\text{ °C} - 100\text{ °C}$  bei  $p = 1\text{ atm}$  und  $\text{pH} = 7$  berechnet. Es werden in den Benchmark-Tests diejenigen Mineralphasen betrachtet, die unter den gegebenen physikalisch-chemischen Randbedingungen thermodynamisch stabil sind.

Die Spalte „Benchmark-Test-Nr.“ der Tab. 5.1 entspricht der fortlaufenden Nummerierung der Benchmark-Tests der verwendeten Parameterdatei /MOO 11a/. Derzeit werden im Vorhaben BGE-45181017<sup>11</sup> „Fortführung und Weiterentwicklung der thermodynamischen Referenzdatenbasis (THEREDA)“ die Benchmark-Tests weiterentwickelt und erweitert. Diese Arbeiten umfassen die Zusammenführung aller einzelnen thermodynamischen Daten aus den einzelnen Parameterdateien in eine einzige umfassende Parameterdatei. Entsprechend ändert sich im Rahmen des genannten Vorhabens auch die Nummerierung der Benchmark-Tests, die in der Spalte „THEREDA-Test-Nr.“ zur besseren Zuordnung angegeben wird.

### 5.3.2 Temperaturabhängigkeit

Der von der Parameterdatei 01 abgedeckte Temperaturbereich liegt zwischen  $25\text{ °C}$  und  $100\text{ °C}$ . Somit werden temperaturabhängige thermodynamische Daten verwendet. Im Folgenden werden die Gleichungen beschrieben, mit welchen TOUGHREACT in den Versionen TOUGHREACT v1.21 und TOUGHREACT v4 die Gleichgewichtskonstanten und Pitzer-Parameter temperaturabhängig berechnet. Weiterhin wird dargestellt, wie das aus dem Vorhaben EMIL vorhandene Konverter-Tool `convert.pl` weiterentwickelt wurde, um die Temperaturabhängigkeit der  $\log K$ -Werte und Pitzer-Parameter aus

---

<sup>11</sup> Hierbei handelt es sich um die Kennziffer „BGE Bestellnummer 45181017“.

THEREDA für TOUGHREACT v4 umzuformatieren. Im Folgenden wird die Berechnung der Temperaturabhängigkeit für TOUGHREACT näher beschrieben, da die Benchmark-Tests für das Release 01 temperaturabhängig sind.

### 5.3.2.1 Gleichgewichtskonstante

TOUGHREACT v.1.21 berechnet den Logarithmus der temperaturabhängigen Gleichgewichtskonstante  $K(T)$  über die Gleichung

$$\text{TOUGHREACT: } \log K(T) = a \ln T + b + cT + \frac{d}{T} + \frac{e}{T^2} \quad (5.1)$$

anhand von fünf Koeffizienten (a bis e) /XU 06/. Zur Angabe der Temperaturabhängigkeit der log K-Werte und Pitzer-Parameter stehen in THEREDA die folgenden Gleichungen zur Verfügung /MOO 11b/.

$$\text{Pitzer-Funktion: } X(T) = \frac{a}{T} + b + c \ln T + d \cdot T + e \cdot T^2 + \frac{f}{T^2} \quad (5.2)$$

$$\text{NEA: } X(T) = a + bT + cT^2 + \frac{d}{T} + \frac{e}{T^2} \quad (5.3)$$

$$\text{Maier-Kelley: } X(T) = a + bT + \frac{c}{T^2} \quad (5.4)$$

$$\text{NEA-extended: } X(T) = a + bT + c T \ln T + dT^2 + eT^3 + \frac{f}{T^2} \quad (5.5)$$

$$\text{NEA-transformed: } X(T) = a + bT + cT^2 + \frac{d}{T} + \frac{e}{T^2} + f \ln T \quad (5.6)$$

$$\text{EQ3/6: } X(T) = a_1 + a_2 \left( \frac{1}{T} - \frac{1}{T_0} \right) + a_3 \ln \left( \frac{T}{T_0} \right) + a_4(T - T_0) \quad (5.7)$$

$$\begin{aligned} \text{Pitzer-PHREEQC: } X(T) = & a_1 + a_2 \left( \frac{1}{T} - \frac{1}{T_0} \right) + a_3 \ln \left( \frac{T}{T_0} \right) + a_4(T - T_0) \\ & + a_5(T^2 - T_0^2) + a_6 \left( \frac{1}{T^2} - \frac{1}{T_0^2} \right) \end{aligned} \quad (5.8)$$

Es hat sich gezeigt, dass zur Angabe der log K-Werte in THEREDA hauptsächlich die Gleichung „NEA-transformed“ (5.6) und in Ausnahmen die Gleichung „Pitzer-Funktion“ (5.2) Verwendung findet.

Im Vergleich der verwendeten Gleichungen von TOUGHREACT und THEREDA fällt auf, dass in der Gleichung aus TOUGHREACT (5.1) ein Term  $f \cdot T^2$  gegenüber der Gleichung Pitzer-Funktion (5.2) fehlt. Um die temperaturabhängigen log K-Werte der Datenbank

THEREDA in TOUGHREACT verwenden zu können, muss eine genaue Umrechnung der Koeffizienten (a bis f) aus den Gleichungen (5.2) bzw. (5.6) ermöglicht werden. Dafür wurde vom LBNL in der Codeversion TOUGHREACT v4 der fehlende Term in Gleichung (5.1) eingefügt:

$$\log K(T) = a \ln T + b + cT + \frac{d}{T} + \frac{e}{T^2} + f \cdot T^2 \quad (5.9)$$

In TOUGHREACT v4 können daher zur Berechnung der temperaturabhängigen log K-Werte beide Gleichungen (5.1) und (5.9) verwendet werden. Standardmäßig werden fünf Koeffizienten (a bis e) aus Gleichung (5.1) erwartet. In der Datei für die thermodynamischen Parameter können nach den Koeffizienten der log K-Werte AKCOES(i) fünf Stützstellen zur Berechnung der Volumenänderung AKCOP(i) angegeben werden /XU 17/.

```
'Name', (AKCOES(i), i=1,5), (AKCOP(i), i=1,5)
```

Wird nun zur Verwendung von Gleichung (5.9) der sechste Koeffizient f benötigt, so müssen in der Zeile sechs Koeffizienten a bis f eingelesen werden. Da jedoch optional nach den AKCOES(i) noch die AKCOP(i) eingelesen werden, muss TOUGHREACT v4 bekannt sein, wie viele Stützstellen für AKCOES(i) eingelesen werden müssen. Um dieses Problem zu lösen, wurde für TOUGHREACT v4 direkt nach der Header Information eine zusätzliche Eingabe eingefügt. Hier wird die Anzahl der Koeffizienten angegeben.

```
!end-of-header      Do not remove this record!
'Log(K) coefficients:' 5|6
'Temperature points:'
```

Die Koeffizienten der Gleichgewichtskonstante werden in unterschiedlichen Zeilen verschiedener Angaben (Sec.-3, Mineral-3, Gas-3 und Surface-3 in /XU 17/) erwartet. Ein Beispiel für eine Zeile der Eingabe für Sec.-3 ist

```
'Ca(SO4)' 0.0000 1.1252798E+01 -3.48222147E-02 0.0000 0.0000
```

Mit sechs Koeffizienten entspricht diese Zeile dann

```
'Ca(SO4)' 0.0000 1.1252798E+01 -3.48222147E-02 0.0000 0.0000 0.0000
```

## Ausgabe der THEREDA Daten für TOUGHREACT

Die Bedeutung der Koeffizienten a bis f in Gleichung (5.9) von TOUGHREACT v4 ist eine andere als in den Gleichungen (5.2) bzw. (5.6) von THEREDA. Daher müssen die Koeffizienten vom Konverter-Tool `convert.pl` nach dem Einlesen der Koeffizienten aus THEREDA in einer anderen Reihenfolge ausgegeben werden (siehe dazu Tab. 5.2).

**Tab. 5.2** Reihenfolge der Koeffizienten  $i$  in `AKCOES (i)` in der thermodynamischen Eingabedatei von TOUGHREACT

THEREDA „Pitzer-Funktion“ (5.2)	THEREDA „NEA-transformed“ (5.6)	TOUGHREACT (5.9)	$i$ in <code>AKCOES (i)</code>
c	f	a	1
b	a	b	2
d	b	c	3
a	d	d	4
f	e	e	5
e	c	f	6

### 5.3.2.2 Pitzer-Parameter

Die Temperaturabhängigkeit der Pitzer-Parameter wird in TOUGHREACT v1.21 mit den vier Koeffizienten  $a_1$  bis  $a_4$  angegeben /ZHA 06/

$$X(T) = a_1 + a_2 \cdot \left( \frac{1}{T} - \frac{1}{T_0} \right) + a_3 \ln \left( \frac{T}{T_0} \right) + a_4 (T - T_0) \quad (5.10)$$

Zur Berücksichtigung weiterer Koeffizienten wurde vom LBNL die Gleichung (5.10) in TOUGHREACT v4 auf sechs Koeffizienten  $a_1$  bis  $a_6$  erweitert.

$$X(T) = a_1 + a_2 \cdot \left( \frac{1}{T} - \frac{1}{T_0} \right) + a_3 \ln \left( \frac{T}{T_0} \right) + a_4 (T - T_0) + a_5 (T^2 - T_0^2) + a_6 \left( \frac{1}{T^2} - \frac{1}{T_0^2} \right) \quad (5.11)$$

Wie bereits im Kap. 5.3.2.1 erwähnt, erfolgt die Umschaltung zwischen den beiden Gleichungen anhand der Zeile

```
'Log(K) coefficients:' 5 (oder 6)
```

in der thermodynamischen Eingabe von TOUGHREACT v4.

Im Gegensatz zu TOUGHREACT v1.21 sind in TOUGHREACT v4 alle Angaben zu den thermodynamischen Parametern (Gleichgewichtskonstante und Pitzer-Parameter) in einer Datei enthalten. Die Pitzer-Parameter finden sich nach den Oberflächenkomplexen in dieser Datei (siehe Beispiel in Abb. 5.5). Vor der Spezifizierung der Pitzer-Parameter werden allgemeine Angaben gemacht. Dazu gehören die Temperatur, für die die Parameter gültig sind, die Temperaturstützstellen, für die die Gleichgewichtskonstante bereits ausgerechnet wurden, sowie der Debye-Hückel-Parameter  $a_{phi}$  für die Temperaturstützstellen.

```
# #####
# Surface complexes #
# #####
'null'

# #####
# Pitzerion interaction parameters #
# #####
#!!! Note: 'Miscellaneous' below is a needed flag !!!!

+-----+
Miscellaneous parameters
+-----+
Temperature limits (degC)
          0.      120.
temperatures
          0.0001  25.0000  60.0000  100.0000
          150.0000 200.0000 250.0000 300.0000
debye huckel aphi
          0.3767  0.3914  0.4190  0.4606
          0.5295  0.6228  0.7535  0.9595
```

**Abb. 5.5** Struktur der thermodynamischen Eingabe von TOUGHREACT v4 in der die Pitzer-Parameter nach den Oberflächenkomplexen eingefügt werden

Die Struktur der Eingabe der Stützstellen zur Berechnung der temperaturabhängigen Pitzer-Parameter ist in TOUGHREACT v4, wie auch in TOUGHREACT v1.21, die Struktur der temperaturabhängigen Eingabe von EQ3/6 Version 8.0 /WOL 03/ mit den in Gleichung (5.10) verwendeten Stützstellen  $a_1$  bis  $a_4$ . Da in der in TOUGHREACT v4 verwendeten Gleichung (5.11) sechs Koeffizienten für die Temperaturabhängigkeit der Pitzer-

Parameter benötigt werden, wurde vom LBNL die Eingabe um die beiden Stützstellen  $a_5$  und  $a_6$  erweitert (siehe Beispiel in Abb. 5.6).

```
+-----+
ca combinations: beta(n) (ca) and Cphi (ca) [optional: alpha(n) (ca)]
+-----+
Ca++          Cl-
alpha(1) =          2.
alpha(2) =          0.
beta(0):
  a1 =      0.3065449714
  a2 =     523608.18638425
  a3 =     2228.557995242
  a4 =      -4.1971890565
  a5 =      0.00147522
  a6 = -22971638.5982979983
beta(1):
  a1 =      1.7081116892
  a2 =          0.
  a3 =          0.
  a4 =     -0.0154168146
  a5 =     0.0000317906
  a6 =          0.
beta(2):
  a1 =          0.
  a2 =          0.
  a3 =          0.
  a4 =          0.
  a5 =          0.
  a6 =          0.
Cphi:
  a1 =      0.0022244526
  a2 =    -49450.01852654
  a3 =   -210.9536535162
  a4 =      0.3980639533
  a5 =     -0.0001401891
  a6 =  2165261.2860198002
+-----+
```

**Abb. 5.6** Eingabestruktur der Stützstellen  $a_1$  bis  $a_6$  für die Berechnung der temperaturabhängigen Pitzer-Parameter in TOUGHREACT v4

## Ausgabe der THEREDA-Daten für TOUGHREACT

Da die Gleichung (5.11) der Gleichung „Pitzer-PHREEQC“ (5.8) aus THEREDA entspricht, können die Stützstellen, die in THEREDA mit dieser Gleichung abgelegt wurden, direkt in TOUGHREACT v4 übernommen werden (siehe Tab. 5.3).

Die Pitzer-Parameter werden in THEREDA auch mit den Gleichungen „Pitzer-Funktion“ (5.2) und „NEA“ (5.3) abgelegt. Die Werte für die Koeffizienten  $a_1$  bis  $a_6$  aus Gleichung (5.11) für TOUGHREACT v4 müssen unter Verwendung der Referenztemperatur  $T_0$  von 298.15 K aus den Werten der Koeffizienten a bis f aus den Gleichungen (5.2) und (5.3) umgerechnet werden (siehe Tab. 5.3).

**Tab. 5.3** Berechnung der Koeffizienten  $a_1 - a_6$  in TOUGHREACT v4 mit einer Referenztemperatur  $T_0$  von 298.15 K aus den Koeffizienten a bis f aus THEREDA

TOUGHREACT v4 (5.11)	„Pitzer-Funktion“ (5.2)	„NEA“ (5.3)	„Pitzer-PHREEQC“ (5.8)
$a_1$	$b + \frac{a_2}{T_0} + a_3 \ln T_0 + a_4 T_0 + a_5 T_0^2 + \frac{a_6}{T_0^2}$	$a + \frac{a_2}{T_0} + a_4 T_0 + a_5 T_0^2 + \frac{a_6}{T_0^2}$	a
$a_2$	a	d	b
$a_3$	c	0	c
$a_4$	d	b	d
$a_5$	e	c	e
$a_6$	f	e	f

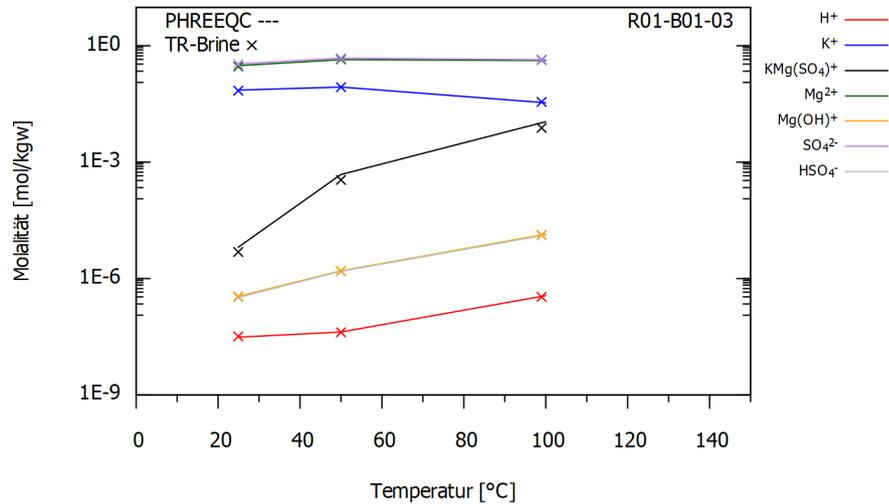
Die genaue Herleitung der Umrechnung der Koeffizienten aus THEREDA für die Verwendung in TOUGHREACT ist in /SEH 17/ beschrieben.

### 5.3.3 Ergebnisse

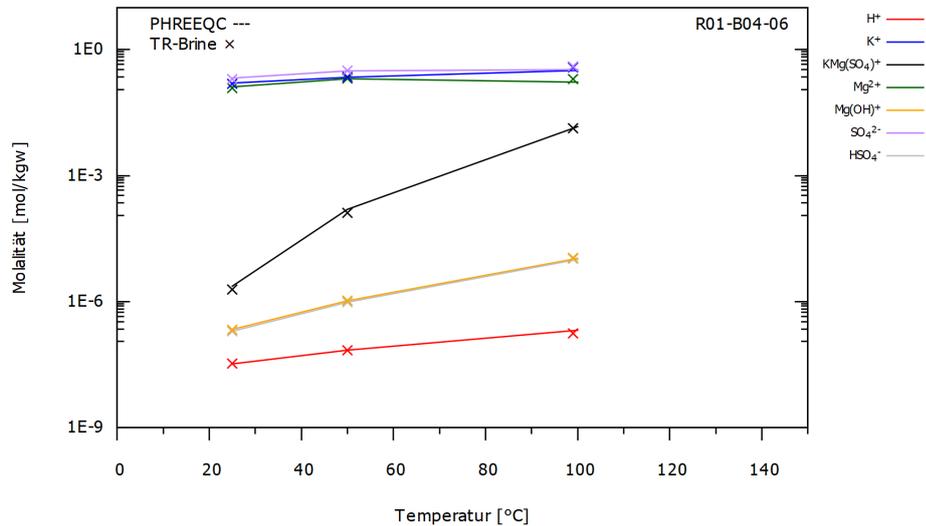
Im Folgenden werden die Ergebnisse der Benchmark-Tests für das Release 01 des geochemischen Systems ozeanischer Salze Na-K-Mg-Ca-Cl-SO<sub>4</sub>-H<sub>2</sub>O(l) mit PHREEQC und TOUGHREACT v4 graphisch dargestellt (Abb. 5.7 – Abb. 5.14).

Die Ergebnisse der Benchmark-Tests mit PHREEQC und TOUGHREACT v4 zeigen für Release 01 eine sehr hohe Übereinstimmung, Unterschiede in der Konzentration der Spezies liegen deutlich unter einem Prozent. Dies gilt über den gesamten betrachteten Temperaturbereich zwischen 25 °C und 100 °C und auch für die betrachteten

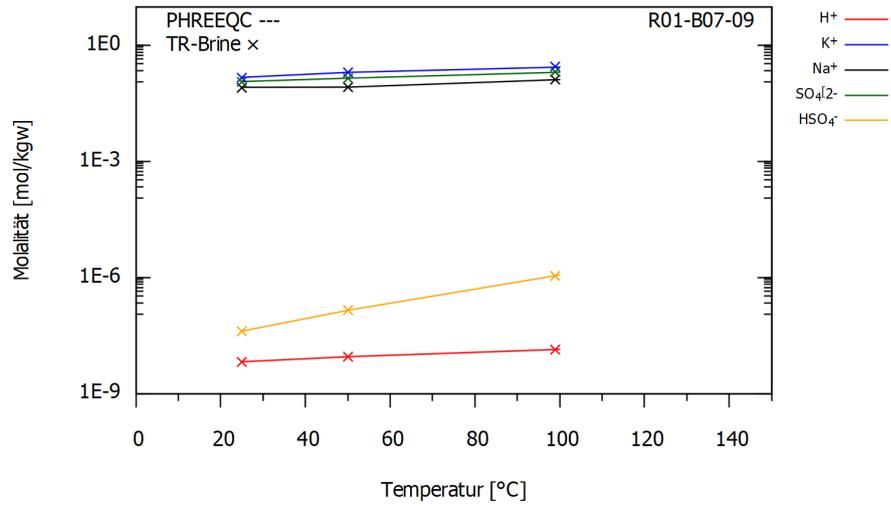
Kombinationen aus unterschiedlichen einwertigen und zweiwertigen Ionen. Da sämtliche Benchmark-Tests Lösungen mit hohen Ionenstärken berücksichtigen, werden die eingesetzten temperaturabhängigen Pitzer-Funktionen mit TOUGHREACT v4 für das System ozeanischer Salze (in Übereinstimmung mit PHREEQC) richtig wiedergegeben.



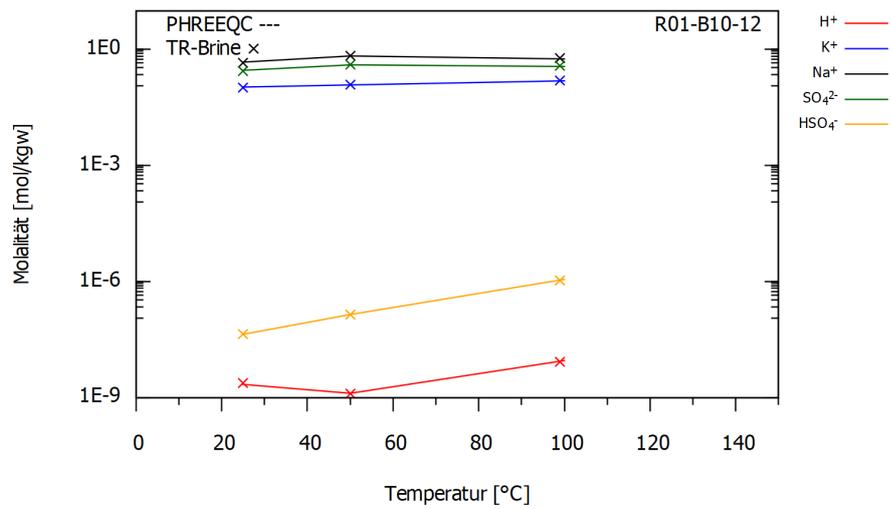
**Abb. 5.7** Molalitäten der aquatischen Spezies für das System K-Mg-SO<sub>4</sub> (Benchmark-Tests 01-03 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)



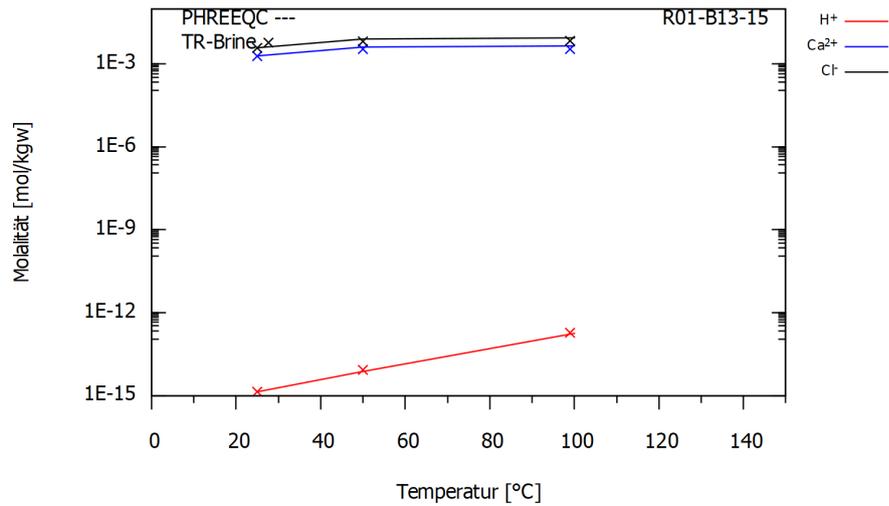
**Abb. 5.8** Molalitäten der aquatischen Spezies für das System K-Mg-SO<sub>4</sub> (Benchmark-Tests 04-06 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)



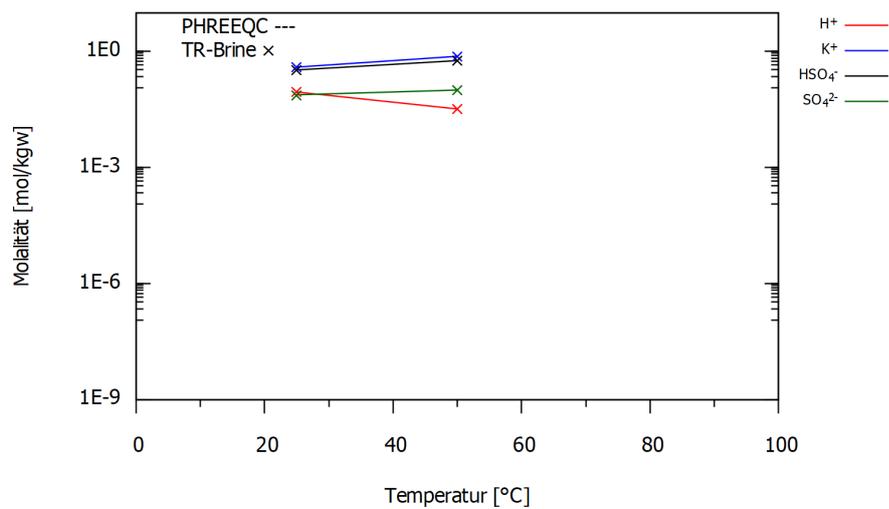
**Abb. 5.9** Molalitäten der aquatischen Spezies für das System K-Na-SO<sub>4</sub> (Benchmark-Tests 07-09 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)



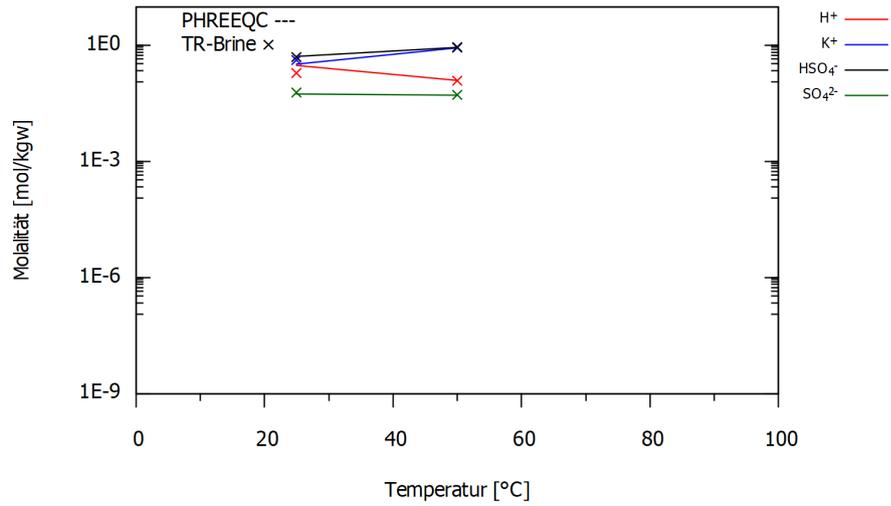
**Abb. 5.10** Molalitäten der aquatischen Spezies für das System K-Na-SO<sub>4</sub> (Benchmark-Tests 10-12 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)



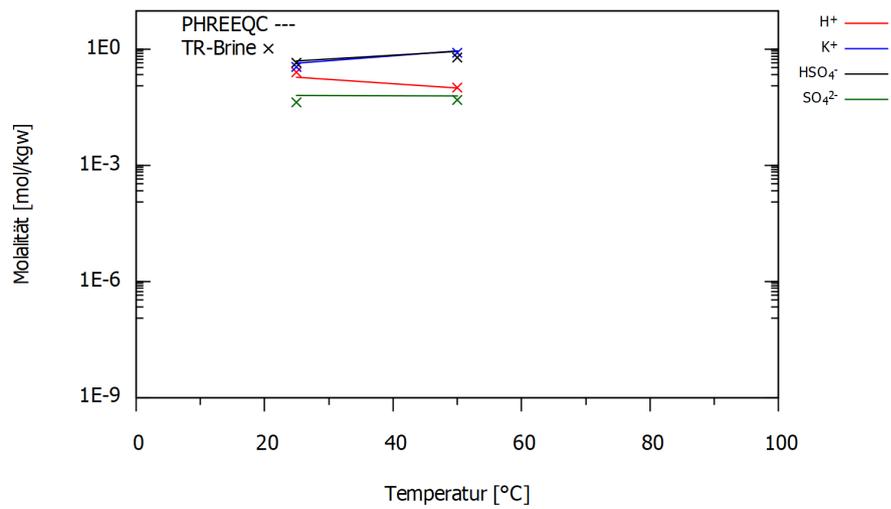
**Abb. 5.11** Molalitäten der aquatischen Spezies für das System Ca-Cl (Benchmark-Tests 13-15 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)



**Abb. 5.12** Molalitäten der aquatischen Spezies für das System K-SO<sub>4</sub> (Benchmark-Tests 16-17 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)



**Abb. 5.13** Molalitäten der aquatischen Spezies für das System K-SO<sub>4</sub> (Benchmark-Tests 18-19 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)



**Abb. 5.14** Molalitäten der aquatischen Spezies für das System K-SO<sub>4</sub> (Benchmark-Tests 20-21 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)

## 5.4 Benchmark-02 (System mit Aktiniden)

### 5.4.1 Vorgaben

Mit der Parameterdatei 02 hat THEREDA eine thermodynamische Datenbasis für ein geochemisches System mit Aktiniden zur Verfügung gestellt: Am-Cm-Nd-Na-Mg-Ca-Cl-OH-H<sub>2</sub>O(l). Für dieses System wurde die Löslichkeit von Festphasen mit Am(III), Cm(III) und Nd(III) in 0,1 und 5,6 molaler NaCl-Lauge bzw. 0,25 und 3,86 molaler CaCl<sub>2</sub>- oder MgCl<sub>2</sub>-Lauge bei unterschiedlichen pH-Werten wie im Folgenden ausgeführt berechnet.

Die Benchmark-Tests mit dem Release 02 berücksichtigen NaCl- /ALT 12a/ und MgCl<sub>2</sub>- bzw. CaCl<sub>2</sub>-haltige Lösungen /ALT 12b/. Die numerischen Ergebnisse der in /ALT 12a/, /ALT 12b/ eingesetzten geochemischen Codes wurden in diesen beiden Publikationen mit experimentellen Daten zur Löslichkeit von /NEC 09/ verglichen. Es zeigt sich, dass alle geochemischen Codes hinreichend gute Ergebnisse erzielen. Zur Überprüfung der Rechnungen mit TOUGHREACT wurden die PHREEQC-Berechnungen aus /ALT 12a/, /ALT 12b/ mit der neueren Codeversion 3.5 /PAR 19/ wiederholt und werden hier, da sich keine Abweichungen gegenüber den alten Rechnungen zeigten, für einen Vergleich herangezogen.

Die Eingabeparameter der Benchmark-Tests aus /ALT 12a/ sowie /ALT 12b/ wurden für die Vergleichbarkeit der numerischen Ergebnisse für die Benchmark-Tests mit TOUGHREACT verwendet. Zu den Eingabeparametern gehören die geochemische Lösungszusammensetzung, die Auswahl der zu berücksichtigenden Festphasen sowie die physikalisch-chemischen Parameter (Temperatur, Druck, pH-Wert). Jeder Benchmark-Test ermittelt für T = 25 °C und p = 1 atm das Lösungsgleichgewicht (SI = 0) einer niedrig- (0,1 bzw. 0,25 mol/l) oder hochsalinaren (3,86 bzw. 5,6 mol/l) Lösung (NaCl, MgCl<sub>2</sub> oder CaCl<sub>2</sub>) mit Nd(OH)<sub>3</sub>(am) für drei verschiedene pH-Werte (siehe Tab. 5.4).

/SEH 17/ zeigte, dass sich die beiden Festphasen Am(OH)<sub>3</sub>(am) und Cm(OH)<sub>3</sub>(am) erwartungsgemäß identisch zu Nd(OH)<sub>3</sub>(am) verhalten, da die verwendeten Koeffizienten und Stützstellen im Pitzer-Ansatz und in der Datenbasis gleich sind. Die Benchmark-Tests aus /ALT 12a/ und /ALT 12b/ für Am(OH)<sub>3</sub>(am) und Cm(OH)<sub>3</sub>(am) werden hier daher nicht durchgeführt und dargestellt. Aus der in Tab. 5.4 dargestellten Beschreibung der geochemischen Zusammensetzung inkl. variierender Ionenstärke und berücksichtigter Festphasen ergeben sich für die entsprechenden pH-Werte insgesamt 18 Berechnungen.

**Tab. 5.4** Eingabeparameter und Nummerierung der Benchmark-Tests zu Release 02

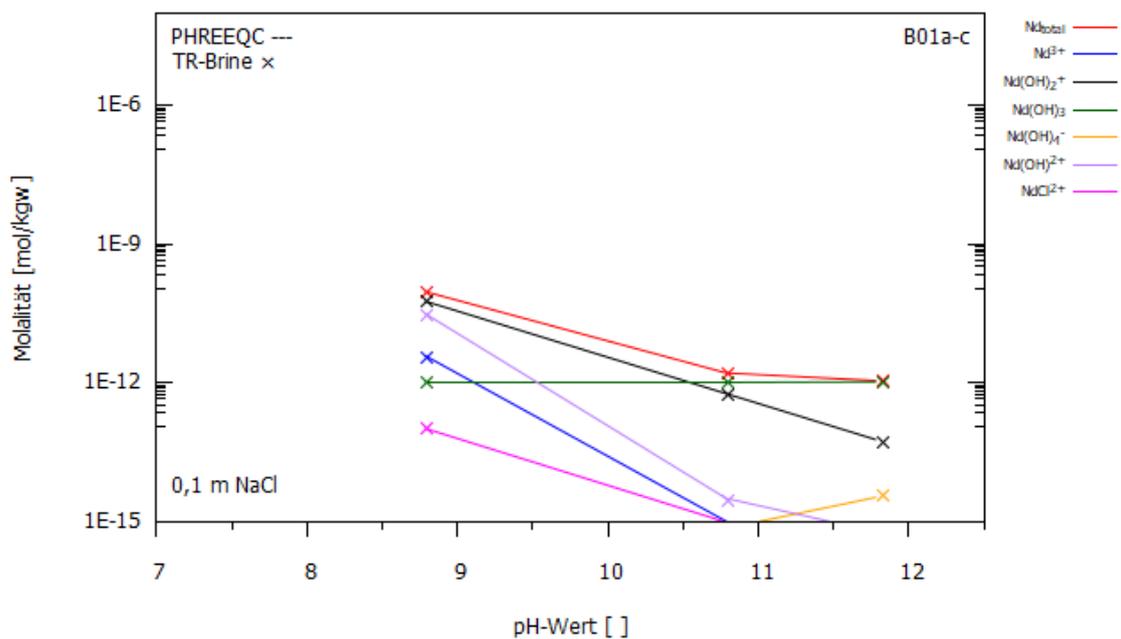
Die „Benchmark-Test-Nr.“ entspricht der fortlaufenden Nummerierung der Benchmark-Tests der Parameterdatei /ALT 12a/, /ALT 12b/. Derzeit werden im Vorhaben BGE-45181017 „Fortführung und Weiterentwicklung der thermodynamischen Referenzdatenbasis (THEREDA)“ die einzelnen Parameterdateien in ein umfassendes einzelnes Release zusammengeführt. Entsprechend ändert sich zukünftig die Nummerierung der Benchmark-Tests, die hier unter der Spalte „THEREDA-Test-Nr.“ aufgeführt ist, um eine bessere Zuordnung zu ermöglichen.

Benchmark-Test-Nr.	THEREDA-Test-Nr.	Festphase	Lösungszusammensetzung	NaOH [mol]
1a	022	Nd(OH) <sub>3</sub> (am)	0,1 M NaCl	10 <sup>-05</sup>
1b	023			10 <sup>-03</sup>
1c	024			10 <sup>-02</sup>
4a	031	Nd(OH) <sub>3</sub> (am)	5,6 M NaCl	10 <sup>-04</sup>
4b	032			10 <sup>-03</sup>
4c	033			10 <sup>-02</sup>
7a	040	Nd(OH) <sub>3</sub> (am)	0,25 M MgCl <sub>2</sub>	10 <sup>-05</sup>
7b	041			5·10 <sup>-05</sup>
7c	042			10 <sup>-04</sup>
8a	043	Nd(OH) <sub>3</sub> (am)	0,25 M CaCl <sub>2</sub>	10 <sup>-05</sup>
8b	044			10 <sup>-04</sup>
8c	045			10 <sup>-02</sup>
13a	058	Nd(OH) <sub>3</sub> (am)	3,86 M MgCl <sub>2</sub>	10 <sup>-04</sup>
13b	059			10 <sup>-03</sup>
13c	060			4·10 <sup>-03</sup>
14a	061	Nd(OH) <sub>3</sub> (am)	3,86 M CaCl <sub>2</sub>	10 <sup>-05</sup>
14b	062			10 <sup>-03</sup>
14c	063			5·10 <sup>-02</sup>

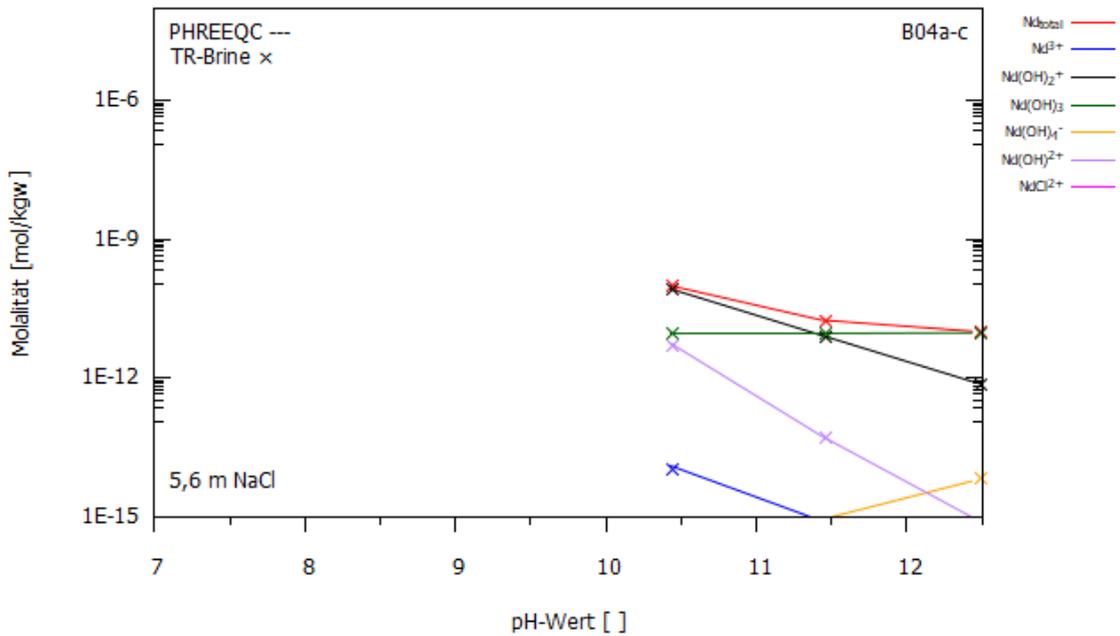
## 5.4.2 Ergebnisse

Im Folgenden werden die Ergebnisse der Benchmark-Tests für die Parameterdatei 02 des geochemischen Systems mit Aktiniden Am-Cm-Nd-Na-Mg-Ca-Cl-OH-H<sub>2</sub>O(l) mit PHREEQC und TOUGHREACT v4 graphisch dargestellt (Abb. 5.15 – Abb. 5.20).

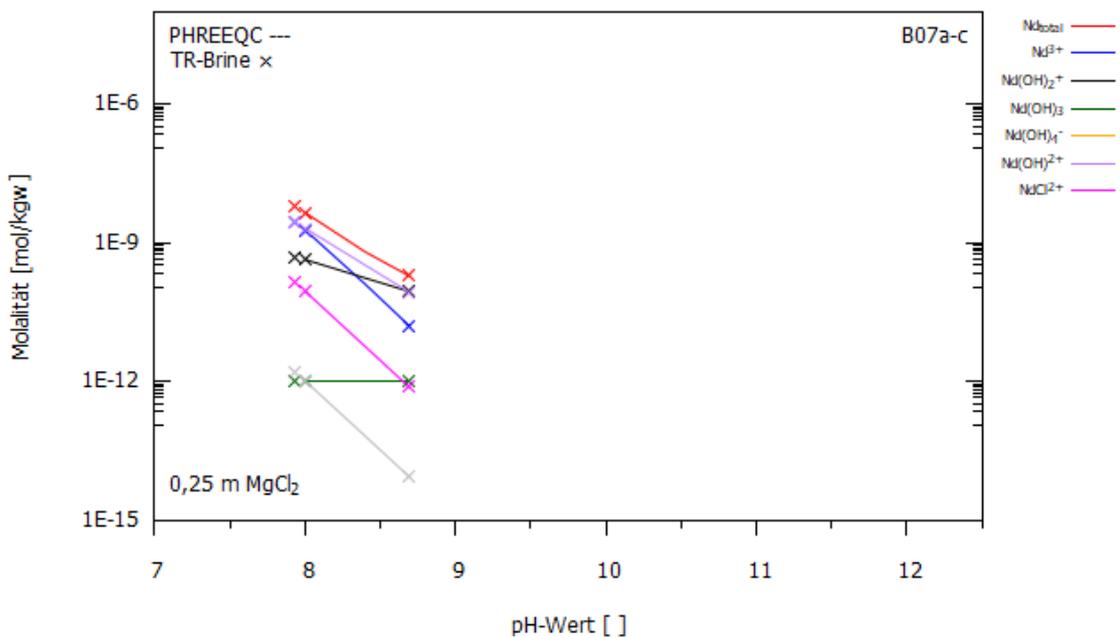
Die überwiegenden Benchmark-Tests zeigen eine sehr hohe Übereinstimmung zwischen den beiden eingesetzten Codes PHREEQC und TOUGHREACT v4. Hierzu gehören die Benchmark-Tests B01 – B13 (Abb. 5.15 – Abb. 5.18).



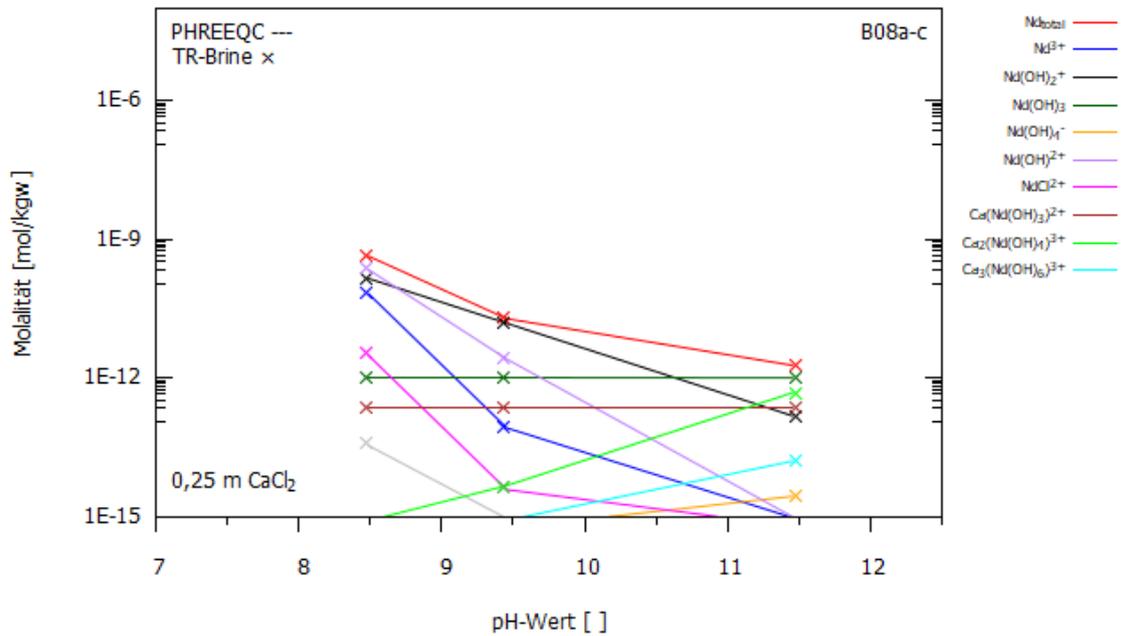
**Abb. 5.15** Molalitäten der aquatischen Spezies von Nd in einer 0,1 molalen NaCl-Lösung (Benchmark-Tests 01a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH)<sub>3</sub>(am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)



**Abb. 5.16** Molalitäten der aquatischen Spezies von Nd in einer 5,6 molalen NaCl-Lösung (Benchmark-Tests 04a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH)<sub>3</sub>(am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)

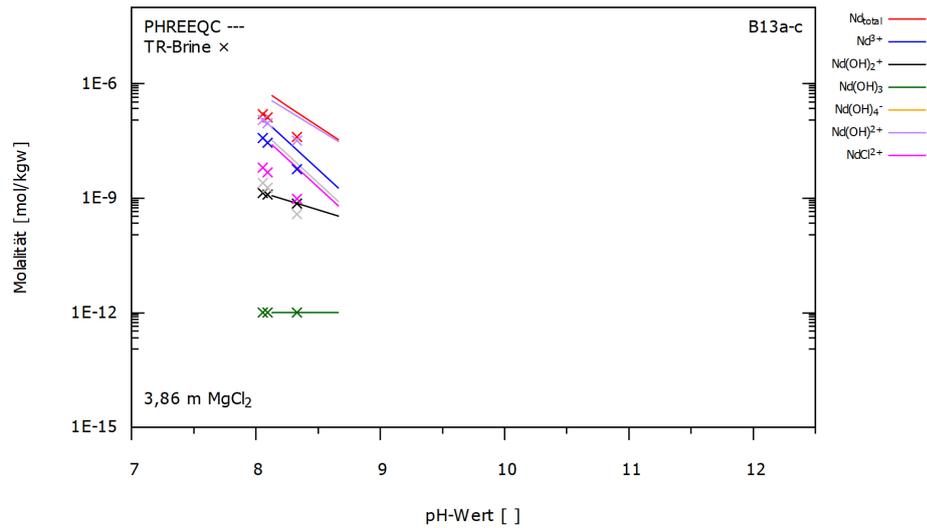


**Abb. 5.17** Molalitäten der aquatischen Spezies von Nd in einer 0,25 molalen MgCl<sub>2</sub>-Lösung (Benchmark-Tests 07a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH)<sub>3</sub>(am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)

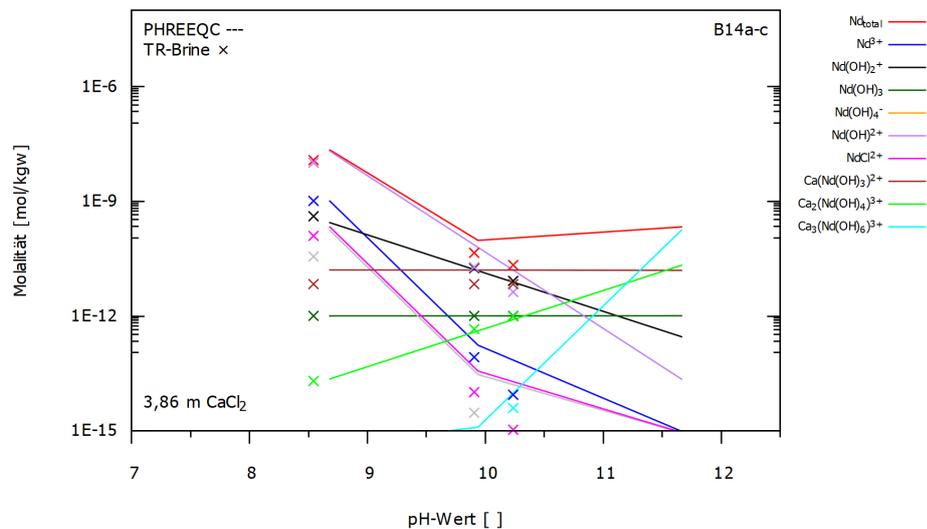


**Abb. 5.18** Molalitäten der aquatischen Spezies von Nd in einer 0,25 molalen  $\text{CaCl}_2$ -Lösung (Benchmark-Tests 08a-c der Parameterdatei 02) im Gleichgewicht mit  $\text{Nd}(\text{OH})_3(\text{am})$  berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)

Zwei der Benchmark-Tests mit hochsalinaren Lösungen mit zweiwertigen Ionen zeigen deutliche Unterschiede zwischen den beiden Codes PHREEQC und TOUGHREACT v4 (Abb. 5.19, Abb. 5.20). Hierbei stimmen die mit PHREEQC erzielten Ergebnisse mit den Ergebnissen aus /ALT 12b/ (offizielle Benchmark-Ergebnisse von THEREDA) überein. Daher liegt nahe, dass die Ergebnisse von TOUGHREACT v4 einem abweichenden Rechenansatz unterliegen. Daher werden Änderungen in der Eingabe bzw. in der Methodik bei der Erstellung der Benchmark-Tests durchgeführt, um diese Abweichung genauer zu untersuchen. Dies wird im Folgenden beschrieben.



**Abb. 5.19** Molalitäten der aquatischen Spezies von Nd in einer 3,86 molalen MgCl<sub>2</sub>-Lösung (Benchmark-Tests 13a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH)<sub>3</sub>(am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)



**Abb. 5.20** Molalitäten der aquatischen Spezies von Nd in einer 3,86 molalen CaCl<sub>2</sub>-Lösung (Benchmark-Tests 14a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH)<sub>3</sub>(am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)

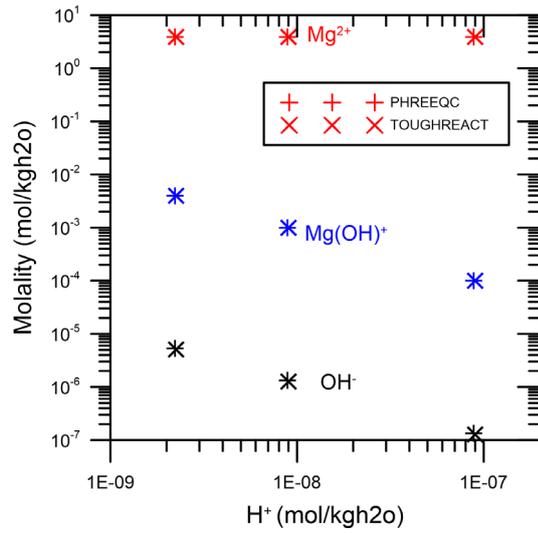
Wie eingangs in Kap. 5.4 erwähnt, bestehen die Benchmark-Tests aus zwei unterschiedlichen Schritten: Zunächst wird die initiale Lösungszusammensetzung bestimmt und der pH-Wert anhand der NaOH-Konzentration berechnet, anschließend erfolgt die Berechnung der Gleichgewichtsreaktion mit der Festphase  $\text{NdOH}_3(\text{s})$ . PHREEQC berechnet diese beiden Vorgänge nacheinander, TOUGHREACT simultan. PHREEQC erlaubt diese Vorgehensweise, da eine Titration mit NaOH programmtechnisch berechnet werden kann, TOUGHREACT erfordert eine manuelle Vorgabe der  $\text{H}^+$ -Konzentration, d. h. der pH-Wert muss für die Berechnung manuell angegeben werden.

Den Ionenausgleich bestimmt PHREEQC aufgrund der Vorgabe aus den Benchmark-Tests für beide Teilschritte über den pH-Wert. In TOUGHREACT kann dies innerhalb einer einzigen Berechnung programmtechnisch nicht erfolgen, da

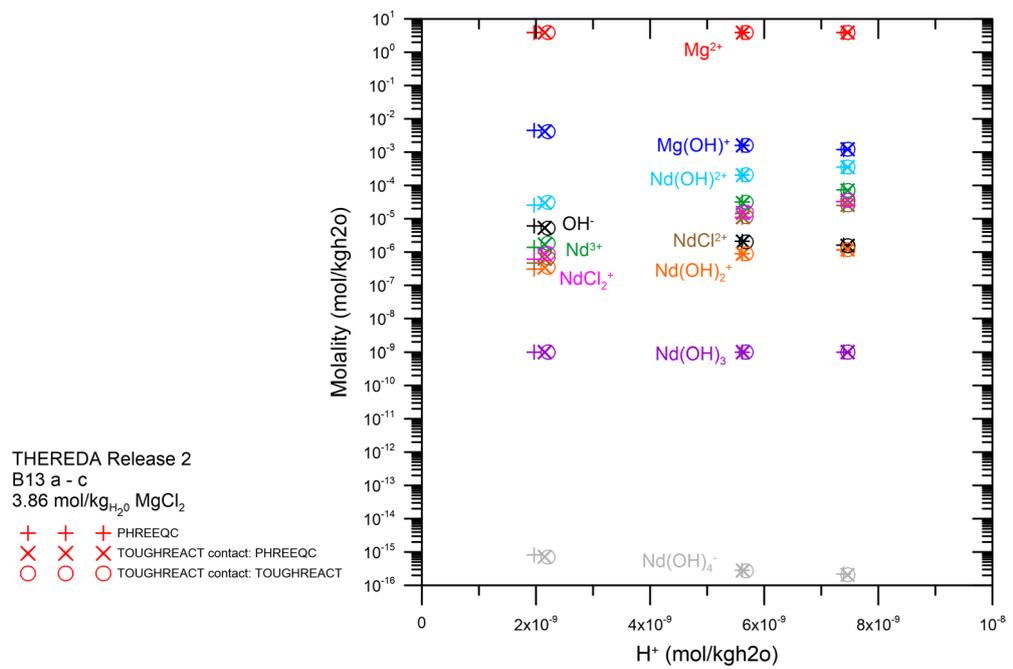
- entweder der Ionenausgleich über den pH-Wert ( $\text{H}^+$ -Konzentration) vorgegeben wird
- oder aber die  $\text{H}^+$ -Konzentration aufgrund der NaOH-Vorgabe ( $\text{pOH} + \text{pH} = 14$ ) angegeben wird.

Daher wurde der Benchmark-Test 13 in zwei einzelne Berechnungen unterteilt und erneut durchgeführt: In der ersten Berechnung erfolgt die Vorgabe der Lösungszusammensetzung und die Berechnung des pH-Wertes (Abb. 5.21), in der zweiten die Festphasenreaktion mit  $\text{NdOH}_3(\text{s})$  mit Vorgabe des pH-Wertes aus der ersten Berechnung (Abb. 5.22).

Diese Vorgehensweise erzeugt für den Benchmark-Test 13 eine hohe Übereinstimmung der Ergebnisse mit den beiden verwendeten Codes PHREEQC und TOUGHREACT. Grundsätzlich sollte zur Verbesserung der Ergebnisse diese Vorgehensweise auf die anderen Benchmark-Tests übertragen werden. Allerdings erscheint aufgrund der sehr guten Ergebnisse in den vorangegangenen Benchmark-Tests diese Vorgehensweise erst bei hochsalinaren Lösungen wie bei Benchmark-Test 13 als erforderlich, um eine hohe Übereinstimmung zu erzielen. Daher wird von einer Umsetzung dieser Methodik auf sämtliche hier durchgeführten Benchmark-Tests abgesehen.



**Abb. 5.21** Ergebnis der Vorrechnung des wiederholten Benchmark-Tests 13 aus Release 02



**Abb. 5.22** Ergebnis der Berechnung der Festphasenreaktion des wiederholten Benchmark-Tests 13 aus Release 02

## 5.5 Benchmark-03 (System mit CO<sub>2</sub>(g))

### 5.5.1 Vorgaben

Mit der Parameterdatei 03 hat THEREDA eine thermodynamische Datenbasis für ein System mit CO<sub>2</sub> zur Verfügung gestellt: Na-K-Mg-Ca-Cl-SO<sub>4</sub>-HCO<sub>3</sub>/CO<sub>2</sub>(g)-H<sub>2</sub>O(l). Die thermodynamischen Daten (log K-Werte und Pitzer-Parameter) werden wie auch in Parameterdatei 02 für eine Temperatur von 25 °C angegeben /VOI 12/. Im Rahmen des Vorhabens 3610R03230 „Forschung und Entwicklung zu gekoppelten THMC-Prozessen in der Übergangsphase nach Verschluss eines Endlagers für Wärme entwickelnde radioaktive Abfälle“ wurde dieser Datensatz für ein Anwendungsbeispiel zum einphasigen Transport von CO<sub>2</sub> mit dem Code PHREEQC eingesetzt /WEY 14/, /WEY 15/ und wird im Vorhaben 4719E03230 zur Modellierung des zweiphasigen Transportes von CO<sub>2</sub> mit dem Code TOUGHREACT verwendet /WEY 20/.

Die Benchmark-Tests für Release 03 verwenden die Eingabeparameter aus /VOI 12/. Hierzu gehören die geochemische Lösungszusammensetzung, die Auswahl der zu berücksichtigenden Festphasen sowie die physikalisch-chemischen Parameter (Temperatur, Druck, pH-Wert). Das Gleichgewicht zwischen destilliertem Wasser mit Mineralphasen (SI = 0) wird für das entsprechende System für T = 25 °C bei p = 1 atm und pH = 7 durchgeführt. Aus der nachfolgenden Beschreibung der geochemischen Zusammensetzung inkl. variierender Ionenstärke und berücksichtigter Festphasen ergeben sich elf Benchmark-Tests (s. Tab. 5.5).

Betrachtet werden geochemische Systeme mit der primären Spezies CO<sub>3</sub><sup>2-</sup> und zusätzlich, je nach Benchmark-Test, Na<sup>+</sup>, K<sup>+</sup>, Ca<sup>2+</sup>, Cl<sup>-</sup> und/oder SO<sub>4</sub><sup>2-</sup>. Sechs der Benchmark-Tests berücksichtigen Kohlenstoff ausschließlich in gelöster Form, fünf Benchmark-Tests zusätzlich Kohlenstoff als gasförmiges CO<sub>2</sub>(g). Für diese fünf Benchmark-Tests wird CO<sub>2</sub>(g) jeweils in eine Untersättigung gesetzt und hierfür ein bestimmter negativer Sättigungsindex angegeben. Der Sättigungsindex (z. B. von CO<sub>2</sub>(g)) in der Gasphase entspricht hierbei dem dekadischen Logarithmus seines Partialdruckes.

**Tab. 5.5** Eingabeparameter der Benchmark-Tests zur Parameterdatei 03

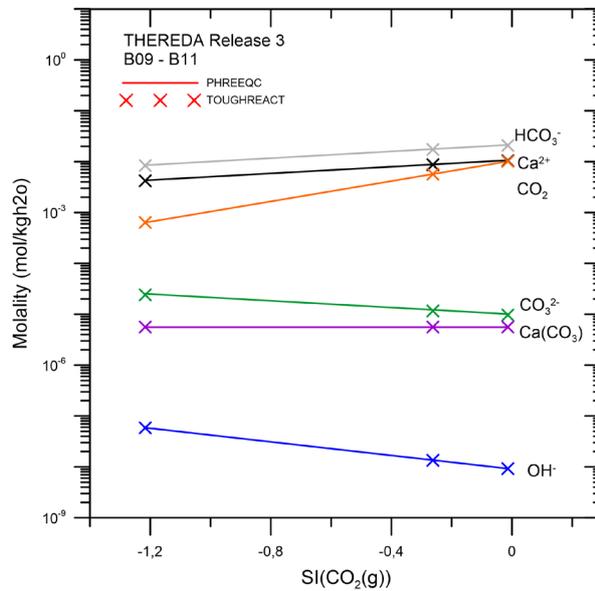
Die „Benchmark-Test-Nr.“ entspricht der fortlaufenden Nummerierung der Benchmark-Tests der Parameterdatei /VOI 12/. Derzeit werden im Vorhaben BGE-45181017 „Fortführung und Weiterentwicklung der thermodynamischen Referenzdatenbasis (THEREDA)“ die einzelnen Parameterdateien in ein umfassendes einzelnes Release zusammengeführt. Entsprechend ändert sich zukünftig die Nummerierung der Benchmark-Tests, die hier unter der Spalte „THEREDA-Test-Nr.“ aufgeführt ist, um eine bessere Zuordnung zu ermöglichen.

Benchmark	THEREDA-Test-Nr.	Primäre Spezies	Festphasen	SI(CO <sub>2</sub> (g))
1	76	K <sup>+</sup> , Cl <sup>-</sup> , CO <sub>3</sub> <sup>2-</sup>	Sylvin	-0,01396764
2	77	6 mol CaCl <sub>2</sub>	/	-0,01396764
3	78	K <sup>+</sup> , Cl <sup>-</sup> , CO <sub>3</sub> <sup>2-</sup>	K <sub>2</sub> (CO <sub>3</sub> ):1.5H <sub>2</sub> O(cr) + Sylvin	/
4	79	K <sup>+</sup> , Cl <sup>-</sup> , CO <sub>3</sub> <sup>2-</sup>	Kalicinit + Sylvin	/
5	80	Na <sup>+</sup> , Cl <sup>-</sup> , SO <sub>4</sub> <sup>2-</sup> , CO <sub>3</sub> <sup>2-</sup>	Mirabilit + Nahcolit	/
6	81	Na <sup>+</sup> , Cl <sup>-</sup> , CO <sub>3</sub> <sup>2-</sup>	Halit + Natron + Na <sub>2</sub> (CO <sub>3</sub> ):7H <sub>2</sub> O(cr)	/
7	82	Na <sup>+</sup> , Cl <sup>-</sup> , CO <sub>3</sub> <sup>2-</sup>	Halit + Na <sub>2</sub> (CO <sub>3</sub> ):7H <sub>2</sub> O(cr)	/
8	83	Na <sup>+</sup> , Cl <sup>-</sup> , CO <sub>3</sub> <sup>2-</sup>	Halit + Nahcolit	/
9	84	Na <sup>+</sup> , Cl <sup>-</sup> , Ca <sup>2+</sup> , CO <sub>3</sub> <sup>2-</sup>	Calcit + Halit	-1,21609642
10	85	Na <sup>+</sup> , Cl <sup>-</sup> , Ca <sup>2+</sup> , CO <sub>3</sub> <sup>2-</sup>	Calcit + Halit	-0,26201267
11	86	Na <sup>+</sup> , Cl <sup>-</sup> , Ca <sup>2+</sup> , CO <sub>3</sub> <sup>2-</sup>	Calcit + Halit	-0,01412464

### 5.5.2 Ergebnisse

Im Folgenden werden die Ergebnisse der Benchmark-Tests für die Parameterdatei 03 des geochemischen Systems mit CO<sub>2</sub>(g) mit PHREEQC und TOUGHREACT v4 dargestellt. Die Benchmark-Tests 09 bis 11 werden graphisch dargestellt, da diese die gleiche Lösungs- und Mineralzusammensetzung bei unterschiedlichem Partialdruck von CO<sub>2</sub> (SI von CO<sub>2</sub>(g)) verwenden. Damit kann der Zusammenhang zwischen dem Partialdruck von CO<sub>2</sub> auf die resultierende Speziation graphisch gezeigt werden (Abb. 5.23). Die Ergebnisse der Benchmark-Tests 01 bis 08 werden tabellarisch dargestellt, da diese jeweils eigene Lösungs- und Mineralzusammensetzungen verwenden und somit für diese Auswertung nicht graphisch vergleichbar sind (Tab. 5.6).

Die Ergebnisse der Benchmark-Tests B09 – B11 des Release 03 zeigen eine sehr hohe Übereinstimmung zwischen den Codes PHREEQC und TOUGHREACT (Abb. 5.23). Die Simulationsergebnisse aus /VOI 12/ zur Abhängigkeit zwischen Speziation von kohlenstoffhaltigen aquatischen Spezies und dem Partialdruck von  $\text{CO}_2(\text{g})$  werden mit TOUGHREACT daher in sehr guter Übereinstimmung wiedergegeben.



**Abb. 5.23** Molalitäten der aquatischen Spezies mit C (Benchmark-Tests 09 - 11 der Parameterdatei 03) zu unterschiedlichen  $\text{SI}_{\text{CO}_2(\text{g})}$  berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze)

Tab. 5.6 stellt die Ergebnisse zur Speziation für die Benchmark-Tests 1 bis 8 mit PHREEQC und TOUGHREACT v4 dar. Die Ergebnisse zwischen den beiden Codes zeigen überwiegend eine sehr gute Übereinstimmung und liegen bei jeder Spezies in der gleichen Größenordnung.

**Tab. 5.6** Molalitäten der aquatischen kohlenstoffhaltigen Spezies als Ergebnis der Benchmark-Berechnungen für Release 03 mit PHREEQC (PRQ) und TOUGHREACT v4 (TR)

Benchmark-Test-Nr. / THEREDA-Test-Nr.	CO <sub>2</sub> (aq) [mol/kgw]	HCO <sub>3</sub> <sup>-</sup> (aq) [mol/kgw]	CO <sub>3</sub> <sup>2-</sup> (aq) [mol/kgw]	CaCO <sub>3</sub> (aq) [mol/kgw]
1 / 76 - TR	2,0847 · 10 <sup>-2</sup>	1,1789 · 10 <sup>-4</sup>	1,5611 · 10 <sup>-10</sup>	-
1 / 76 - PRQ	2,0900 · 10 <sup>-2</sup>	1,189 · 10 <sup>-4</sup>	1,614 · 10 <sup>-10</sup>	-
2 / 77 - TR	4,0710 · 10 <sup>-3</sup>	4,3102 · 10 <sup>-7</sup>	7,0736 · 10 <sup>-8</sup>	2,7752 · 10 <sup>-6</sup>
2 / 77 - PRQ	4,0970 · 10 <sup>-3</sup>	4,334 · 10 <sup>-7</sup>	7,328 · 10 <sup>-8</sup>	2,594 · 10 <sup>-6</sup>
3 / 78 - TR	3,1876 · 10 <sup>-11</sup>	1,8703 · 10 <sup>-2</sup>	6,8580 · 10 <sup>0</sup>	-
3 / 78 - PRQ	1,440 · 10 <sup>-11</sup>	9,475 · 10 <sup>-3</sup>	8,039 · 10 <sup>0</sup>	-
4 / 79 - TR	5,0928 · 10 <sup>-2</sup>	2,6711 · 10 <sup>0</sup>	1,3242 · 10 <sup>-2</sup>	-
4 / 79 - PRQ	2,0740 · 10 <sup>-2</sup>	1,674 · 10 <sup>0</sup>	2,074 · 10 <sup>-2</sup>	-
5 / 80 - TR	7,2551 · 10 <sup>-3</sup>	6,4331 · 10 <sup>-1</sup>	7,2553 · 10 <sup>-3</sup>	-
5 / 80 - PRQ	7,131 · 10 <sup>-3</sup>	6,401 · 10 <sup>-1</sup>	7,131 · 10 <sup>-3</sup>	-
6 / 81 - TR	6,630 · 10 <sup>-10</sup>	4,3638 · 10 <sup>-3</sup>	2,3467 · 10 <sup>0</sup>	-
6 / 81 - PRQ	9,686 · 10 <sup>-10</sup>	5,904 · 10 <sup>-3</sup>	2,627 · 10 <sup>0</sup>	-
7 / 82 - TR	9,5550 · 10 <sup>-10</sup>	5,7733 · 10 <sup>-3</sup>	2,2632 · 10 <sup>0</sup>	-
7 / 82 - PRQ	8,722 · 10 <sup>-10</sup>	5,809 · 10 <sup>-3</sup>	2,395 · 10 <sup>0</sup>	-
8 / 83 - TR	2,6631 · 10 <sup>-3</sup>	1,8427 · 10 <sup>-1</sup>	2,6628 · 10 <sup>-3</sup>	-
8 / 83 - PRQ	2,720 · 10 <sup>-3</sup>	1,869 · 10 <sup>-1</sup>	2,719 · 10 <sup>-3</sup>	-

## 5.6 Zusammenfassung und Ausblick

Im Rahmen des Arbeitspaketes 4 des Vorhabens wurde das im Vorhaben EMIL entwickelte Konvertierungstool /SEH 17/ erweitert, sodass die thermodynamischen Daten aus THEREDA mit der neuen Version TOUGHREACT v4 inkl. der Pitzer-Parameter verwendet werden können.

Zur Validierung des Konvertierungstools im Umgang mit dem Code TOUGHREACT v4 und zur Verifizierung von Codeteilen von TOUGHREACT wurden Benchmark-Tests durchgeführt, die auch schon im Vorhaben THEREDA mit unterschiedlichen geochemischen Codes (z. B. PHREEQC, Geochemist's Workbench oder ChemApp) berechnet wurden. Diese Benchmark-Tests wurden für drei geochemische Systeme (Ozeanische Salze, Aktinide, CO<sub>2</sub>(g)) durchgeführt (Kap. 5.3 – 5.5). Die erzielten Ergebnisse mit

TOUGHREACT v4 erzielten sehr gute Übereinstimmungen mit den im Vorhaben THEREDA erzielten, qualitätsgesicherten Ergebnissen des geochemischen Codes PHREEQC.

Die hier durchgeführten Arbeiten bilden eine Grundlage, dass mit TOUGHREACT v4 künftig mit den thermodynamischen Daten aus THEREDA Modellrechnungen zum reaktiven Transport in hochsalinaren Lösungen durchgeführt werden können. Ein Anwendungsbeispiel zum reaktiven Stofftransport von  $^{14}\text{C}$  mit TOUGHREACT ist derzeit im Rahmen des Vorhabens 4719E03230 in Erarbeitung /WEY 20/. Hierbei kommt das Release 03 von THEREDA zur Anwendung, für das im Rahmen dieser Studie Benchmark-Tests durchgeführt wurden. Hierdurch können die Wechselwirkungen zwischen  $\text{CO}_2$  und hochsalinaren Lösungen während des Transportes detaillierter beschrieben werden. Fernziel dieses Anwendungsbeispiels ist es den bislang in Langzeitsicherheitsanalysen eingesetzten Henry-Ansatz zur Berechnung des Gas-Gleichgewichtes mit dem in TOUGHREACT verwendeten thermodynamischen Ansatzes vergleichen zu können. Mit den thermodynamischen Daten aus Release 03 von THEREDA, die nun mit TOUGHREACT v4 kompatibel sind, bilden eine technische Grundlage zur Durchführung solcher Modellrechnungen.

Weiterhin können nun die temperaturabhängigen thermodynamischen Daten aus THEREDA für hochsalinare Lösungen (Release 01) mit TOUGHREACT v4 verwendet werden. In zukünftigen Forschungs- und Entwicklungsvorhaben können diese daher zum Einsatz kommen, um Fragestellungen zu temperaturabhängigen, geochemischen Prozessen zu bearbeiten. Hierzu zählt beispielsweise der Einfluss eines sich zeitlich ändernden Temperaturfeldes infolge der Wärmeabgabe von beladenen Endlagerbehältern mit hoch radioaktiven Abfällen auf Mineralumwandlungsprozesse, die das Transportverhalten von Radionukliden beeinflussen können.

## A Anhang

### A.1 Beschreibung der MOPR-Jumper für TR-BRINE /SPY 19/

#### REACT input block - Parameter choices for reactive transport simulation

MOPR(21) This flag is used to turn the Pitzer activity coefficient model on or off.

=0: Default HKF extended Debye-Hückel activity coefficient model (same model as in previous code versions).

>0: Turns on the Pitzer activity coefficient model. Care must be taken to use a Pitzer thermodynamic database consistent with this option (with input file name specified in file solute.inp)

When using this option, users should make sure to use an appropriate Pitzer thermodynamic database, with ion-interaction parameters located at the bottom of the database, and making sure that secondary species (ion pairs) accounted for by the Pitzer ion-interaction parameters are not “double-counted” by secondary species. See Section 3.3 for important considerations regarding this matter.

MOPR(22) Only active when MOPR(21) > 0

=0: Pitzer model without simplifications

=1: nca terms omitted (only ca, cc, aa, cca, and caa terms)

=2: nca,cca, and caa terms omitted (only ca, cc, and aa terms)

=3: nca, cca, caa, cc, and aa terms omitted (only ca terms)

(n, c, and a stand for neutral species, cations, and anions, respectively)

MOPR(23) Only active when MOPR(21) > 0

=0: Consider ionic strength in mixing terms

=1: Ignore ionic strength in mixing terms

Default values for MOPR(22) and MOPR(23) are 0. It is the responsibility of the user to ensure that the selected level of simplification is appropriate for the intended use (Appendix A, Section A.2). Typical values for MOPR(22) and MOPR(23) are as follows:

For  $I < 5$  mMOPR (22) = 3 and MOPR (23) = 1

For  $I = 5-10$  mMOPR (22) = 2 and MOPR (23) = 0

For  $I > 10$  mMOPR (22) = 1 or 0, and MOPR (23) = 0

MOPR(24) This flag is used to enable water vapor-pressure lowering due to dissolved salts (with either the HKF or Pitzer models; Equations 2.3 and 2.7). This is not to be confused with vapor-pressure lowering due to capillary pressure, which is enabled when using the EOS4 module; in this case, the salt effect on vapor pressure is added to the capillary pressure effect).

= 0: vapor-pressure lowering is neglected.

= 1: vapor-pressure lowering is enabled.

## Literaturverzeichnis

- /ABD 20/ Abdelkader, A., Bajaj, C. L., Ebeida, M. S., Mahmoud, A. H., Mitchell, S. A., Owens, J. D., Rushdi, A. A.: VoroCrust: Voronoi Meshing Without Clipping. *ACM Transactions on Graphics*, Bd. 39, Nr. 3, S. 1–16, DOI 10.1145/3337680, 2020.
- /ALT 12a/ Altmaier, M., Bok, F., Bube, C., Marquardt, C., Moog, H. C.: Am(III), Cm(III), Nd(III) in NaCl solutions at 298.15K, Benchmark number 02, Revision 3.0. Thermodynamische Referenzdatenbasis (THEREDA), THEREDA - Benchmark calculation, 51 S., 25. Mai 2012.
- /ALT 12b/ Altmaier, M., Bok, F., Bube, C., Marquardt, C., Moog, H. C.: Am(III), Cm(III), Nd(III) in MgCl<sub>2</sub> and CaCl<sub>2</sub> solutions at 298.15K, Benchmark number 02, Revision 3.0. Thermodynamische Referenzdatenbasis (THEREDA), THEREDA - Benchmark calculation, 84 S., 25. Mai 2012.
- /AYA 19/ Ayachit, U.: The ParaView Guide, Updated for ParaView version 5.6. Kitware, 260 S.: Los Alamos National Labs (LANL), 21. Januar 2019.
- /BAR 89/ Barcus, M.: Beschreibung und Diskussion des FEBE-Algorithmus. Gesellschaft für Reaktorsicherheit (GRS) mbH, BAM-89-01: Garching, Juni 1989.
- /BEA 18/ Bear, J.: Modeling Phenomena of Flow and Transport in Porous Media. *Theory and Applications of Transport in Porous Media*, Bd. 31, 742 S., ISBN 978-3-319-72825-4, DOI 10.1007/978-3-319-72826-1, Springer International Publishing: Cham, 2018.
- /BER 14/ Berry, P., Bonduá, S., Bortolotti, V., Cormio, C., Vasini, E. M.: A GIS-based open source pre-processor for georesources numerical modeling. *Environmental Modelling & Software*, Bd. 62, S. 52–64, DOI 10.1016/j.envsoft.2014.08.011, 2014.
- /BET 15/ Bethke, C. M., Yeakel, S.: The Geochemist's Workbench® Release 10.0, Reference Manual. Hrsg.: Aqueous Solutions, L. L.C., 446 S.: Champaign, Illinois, USA, 15. Juni 2015.

- /BON 17/ Bonduà, S., Battistelli, A., Berry, P., Bortolotti, V., Consonni, A., Cormio, C., Geloni, C., Vasini, E. M.: 3D Voronoi grid dedicated software for modeling gas migration in deep layered sedimentary formations with TOUGH2-TMGAS. *Computers and Geosciences*, Bd. 108, S. 50–55, DOI 10.1016/j.cageo.2017.03.008, 2017.
- /BOT 10/ Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., Lévy, B.: *Polygon Mesh Processing*. ISBN 978-1-56881-426-1, 2010.
- /BRA 14/ Bracke, G., Fischer, H., Frieling, G., Hansmeier, C., Hotzel, S., Kock, I., Seher, H., Weyand, T.: *Entwicklungen und Untersuchungen zu (T)HMC-Prozessen eines Endlagers für Wärme entwickelnde radioaktive Abfälle*. GRS-350, 73 S., ISBN 9783944161303, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH: Köln, Oktober 2014.
- /CGA 20/ CGAL: CGAL - The Computational Geometry Algorithms Library. Stand von 2020, erreichbar unter <https://www.cgal.org/index.html>, abgerufen am 25. Mai 2020.
- /COR 54/ Corey, A. T.: The Interrelation Between Gas and Oil Relative Permeabilities. *Producers Monthly*, Bd. 19, S. 38–41, 1954.
- /ECK 18/ Eckel, J.: *QS-Handbuch für das Rechenprogramm MARNIE2*. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, GRS-P-9, Vol. 4 Rev. 0: Köln, April 2018.
- /ERI 03/ Eriksson, G.: *ChemApp - The Thermochemistry Library for your Software*. GTT-Technologies: Herzogenrath, Germany, 2003.
- /FAT 59/ Fatt, I., Klikoff, W. A.: Effect of Fractional Wettability on Multiphase Flow Through Porous Media. *Transactions of the American Institute of Mining, Metallurgical and Petroleum Engineers (AIME)*, Bd. 216, S. 246, 1959.
- /FRE 14/ Freeman, C. M., Boyle, K. L., Reagan, M., Johnson, J., Rycroft, C., Moridis, G. J.: MeshVoro, A three-dimensional Voronoi mesh building tool for the TOUGH family of codes. *Computers and Geosciences*, Bd. 70, S. 26–34, DOI 10.1016/j.cageo.2014.05.002, 2014.

- /GEN 80/ Genuchten, M. T. van: A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. Soil Science Society of America Journal, Bd. 44, Nr. 5, S. 892–898, DOI 10.2136/sssaj1980.03615995004400050002x, 1980.
- /GRA 77/ Grant, M. A.: Permeability Reduction Factors at Wairakei, Paper 77-HT-52. In: American Society of Mechanical Engineers (ASME) (Hrsg.): AIChE-ASME Heat Transfer Conference. AIChEASME Heat Transfer Conference, Salt Lake City, Utah, 15. - 17. August 1977, 1977.
- /GRA 92/ Graf, U., Luther, W., Romstedt, P., Scheurer, M.: ATHLET-Entwicklung, Arbeitspunkt 6: Lösungsverfahren. Hrsg.: Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS-A-Bericht, GRS-A-1939, Mai 1992.
- /GRA 18/ GRASS Development: GRASS GIS 7.5 Reference Manual. Stand vom 24. Januar 2018, erreichbar unter <https://grass.osgeo.org/grass73/manuals/index.html>, abgerufen am 24. Januar 2018.
- /GRE 19/ Greenshields, C. J.: OpenFOAM User Guide, Version 7. OpenFOAM Foundation Ltd., 237 S., Juli 2019.
- /HOF 93/ Hofer, E., Luther, W., Romstedt, P.: ATHLET-Entwicklung, Arbeitspunkt 6: Zeitintegration. Hrsg.: Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS-A-Bericht, GRS-A-2081, November 1993.
- /JAV 00/ Javeri, V.: Scoping analysis of combined gas and nuclide transport including variable brine fraction and rock convergence in a two dimensional repository. In: Roth, A. (Hrsg.): Disposal technologies and concepts 2000, Proceedings. ISBN 9783980641531, Kontec Gesellschaft für Technische Kommunikation: Hamburg, 2000.

- /JAV 06a/ Javeri, V.: Scoping analyses of combined gas and nuclide transport in a deep repository considering coupled hydro-geomechanical processes. In: Varona, P., Hart, R. D. (Hrsg.): FLAC and numerical modeling in geomechanics 2006, Proceedings of the 4th international FLAC symposium, 29-31 May 2006, Madrid, Spain. International Conference on Numerical Methods in Geomechanics, ISBN 9780976757702, Itasca Consulting Group, Inc: Minneapolis, MN, 2006.
- /JAV 06b/ Javeri, V.: Three dimensional analyses of combined gas and nuclide transport in a repository considering coupled hydro-mechanical processes. In: Finsterle, S., Oldenburg, C. M. (Hrsg.): TOUGH Symposium 2006, Technical Program and Abstracts. Lawrence Berkeley National Laboratory (LBNL), TOUGH Symposium 2006, Berkley, 15. - 17. Mai 2006, LBNL-60087, S. 46: Berkeley, California, USA, 2006.
- /JAV 06c/ Javeri, V.: Three dimensional analyses of coupled gas, heat and nuclide transport in a repository including rock salt convergence. In: Finsterle, S., Oldenburg, C. M. (Hrsg.): TOUGH Symposium 2006, Technical Program and Abstracts. Lawrence Berkeley National Laboratory (LBNL), TOUGH Symposium 2006, Berkley, 15. - 17. Mai 2006, LBNL-60087, S. 80: Berkeley, California, USA, 2006.
- /JAV 08/ Javeri, V.: Three dimensional analysis of combined gas, heat and nuclide transport in a repository in clay rock including coupled thermo-hydro-geomechanical processes. Physics and Chemistry of the Earth – Clays in Natural and Engineered Barriers for Radioactive Waste Confinement, Bd. 33, Supplement 1, S. S252-S259, DOI 10.1016/j.pce.2008.10.038, 2008.
- /KOC 16/ Kock, I., Frieling, G., Navarro, M.: Zweiphasenfluss in einem salinaren Endlager am Beispiel des ERAM, Abschlussbericht. GRS-444, 49 S., ISBN 978-3-946607-26-7, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH: Köln, 2016.
- /LAN 17/ Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGLN): Aktuelle Landeskarten von Niedersachsen und Bremen Kartenverzeichnis 2017/2018. Hannover, erreichbar unter , Stand vom Oktober 2017.

- /LI 06/ Li, K., Horne, R. N.: Comparison of methods to calculate relative permeability from capillary pressure in consolidated water-wet porous media. *Water Resources Research*, Bd. 42, Nr. 6, S. 61, DOI 10.1029/2005WR004482, 2006.
- /LOS 11/ Los Alamos National Laboratory (LANL): LaGriT - Los Alamos Grid Toolbox. Erreichbar unter <http://lagrit.lanl.gov/index.shtml>, abgerufen am 24. Januar 2018.
- /LUT 96/ Luther, W., Romstedt, P.: Verbesserung der Zeitintegration und des Sparse Matrix Pakets in ATHLET. Hrsg.: Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS-A-Bericht, GRS-A-2356, Mai 1996.
- /MAR 02/ Martens, K.-H., Fischer, H., Romstedt, P.: Beschreibung des Rechenprogrammes MARNIE. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS-A-3027, 135 S.: Köln, 1. Januar 2002.
- /MER 08/ Merkel, J. B., Planer-Friedrich, B.: Grundwasserchemie, Praxisorientierter Leitfaden zur numerischen Modellierung von Beschaffenheit, Kontamination und Sanierung aquatischer Systeme. 2. Aufl., 242 S., Springer-Verlag: Berlin / Heidelberg, 2008.
- /MOO 11a/ Moog, H. C., Wilhelm, S.: Polythermal solubilities in selected ternary solutions with regard to the system of oceanic salts, Benchmark Release 1. Thermodynamische Referenzdatenbasis (THEREDA), THEREDA - Benchmark calculation, 43 S., 30. Juni 2011.
- /MOO 11b/ Moog, H. C.: Technical Documentation of THEREDA, Databank, Revision 3.5.0.0. Thermodynamische Referenzdatenbasis (THEREDA), THEREDA Technical Paper, 103 S., 24. Juni 2011.
- /MOO 15/ Moog, H. C., Bok, F., Marquardt, C. M., Brendler, V.: Disposal of nuclear waste in host rock formations featuring high-saline solutions – Implementation of a thermodynamic reference database (THEREDA). *Applied Geochemistry*, Bd. 55, S. 72–84, DOI 10.1016/j.apgeochem.2014.12.016, 2015.

- /MOR 18/ Moreland, K.: The ParaView Tutorial, Version 5.6. Sandia National Laboratories, 151 S.: Los Alamos, New Mexico, USA, 2018.
- /NAV 08/ Navarro, M., Baltés, B., Beuth, T., Bracke, G., Fischer, H., Fischer-Appelt, K., Hotzel, S., Javeri, V., Kindt, A., Lambers, L., Larue, P.-J., McStocker, B., Oppermann, U., Schrödl, E.: Verfolgung und Bewertung der Fortentwicklung des Standes von Wissenschaft und Technik beim Nachweis der Langzeitsicherheit von Endlagern, Abschlussbericht zum Vorhaben SR 2548. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS-A-3418: Köln, 2008.
- /NAV 09/ Navarro, M.: Simulating the migration of repository gases through argillaceous rock by implementing the mechanism of pathway dilation into the code TOUGH2 (TOUGH2-PD), PAMINA project, public milestone 3.2.14. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, Community Research, 59 S., European Commission (EC): Brüssel, September 2009.
- /NAV 11/ Navarro, M., Seher, H.: Nicht-kartesische Gitter für TOUGH2. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS-A-3593: Köln, 2011.
- /NAV 13/ Navarro, M.: Handbuch zum Code TOUGH2-GRS.00a, Erweiterungen des Codes TOUGH2 zur Simulation von Strömungs- und Transportprozessen in Endlagern. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS-310, 75 S., ISBN 978-3-939355-89-2: Köln, 2013.
- /NAV 16/ Navarro, M., Eckel, J.: TOUGH2-GRS, Version 1, User Manual. GRS-403, 87 S., ISBN 978-3-944161-84-6, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH: Köln, Juli 2016.
- /NAV 18a/ Navarro, M., Eckel, J., Fischer, H., Frieling, G., Hotzel, S., Kock, I., Kri-scher, A., Seher, H., Uhlmann, S., Weyand, T. (Hrsg.): Entwicklung von Methoden zum Nachweis der Betriebs- und Langzeitsicherheit von Endlagern, Abschlussbericht des Vorhabens 4715E03230. GRS-514, ISBN 978-3-946607-99-1, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH: Köln, September 2018.

- /NAV 18b/ Navarro, M.: User Manual, TOUGH2-GRS Version 2, TOUGH2-MP-GRS Version 0. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, 2018.
- /NAV 18c/ Navarro, M., Eckel, J., Fischer, H., Hotzel, S., Kock, I.: Test Handbook, TOUGH2-GRS Version 2, TOUGH2-MP-GRS Version 0. GRS-402, ISBN 978-3-944161-83-9, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH: Köln, 2018.
- /NAV 19/ Navarro, M.: Neukonzeption des Codes SITA, interner Bericht. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, 2019.
- /NAV 20/ Navarro, M.: Entwicklerhandbuch für TOUGH2-GRS und TOUGH2-MP-GRS, interner Bericht. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, 2020.
- /NEC 09/ Neck, V., Altmaier, M., Rabung, T., Lützenkirchen, J., Fanghänel, T.: Thermodynamics of trivalent actinides and neodymium in NaCl, MgCl<sub>2</sub>, and CaCl<sub>2</sub> solutions, Solubility, hydrolysis, and ternary Ca-M(III)-OH complexes. Pure and Applied Chemistry, Bd. 81, Nr. 9, S. 1555–1568, DOI 10.1351/PAC-CON-08-09-05, 2009.
- /PAR 13/ Parkhurst, D. L., Appelo, C. A. J.: Description of input and examples for PHREEQC version 3, A computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations. U.S. Geological Survey Techniques and Methods 6-A43, 497 S., U.S. Department of the Interior, U.S. Geological Survey: Denver, 2013.
- /PAR 19/ Parkhurst, D. L., Appelo, C. A. J.: PHREEQC, A Computer Program for Speciation, Batch-Reaction, One-Dimensional Transport and Inverse Geochemical Calculations. Version 3.5.0, Windows, U.S. Geological Survey (USGS): Reston, VA, U.S.A., 14. Februar 2019.
- /PIT 91/ Pitzer, K. S.: Ion interaction approach, Theory and data correlation. In: Pitzer, K. S. (Hrsg.): Activity coefficients in electrolyte solutions. 2. Aufl., S. 75–153, ISBN 0-8493-5415-3, CRC Press, Inc.: Boca Raton, Florida, USA, 1991.

- /POL 20/ Polyhedron Solutions: plusFORT. Erreichbar unter <https://polyhedron.com/?product=plusfort>, abgerufen am 13. November 2020.
- /PRU 99/ Pruess, K., Oldenburg, C., Moridis, G.: TOUGH2 User's Guide, Version 2.0. Lawrence Berkeley National Laboratory (LBNL), LBNL-43134, 198 S.: Berkeley, California, USA, 1. November 1999, revised September 2012.
- /QGI 18/ QGIS: QGIS User Guide, Release 2.18. 514 S., 23. Januar 2018.
- /RYC 09/ Rycroft, C. H.: Voro++: a three-dimensional Voronoi cell library in C++. Lawrence Berkeley Laboratory (LBL), 16 S., 2009.
- /SCI 18/ SciPy.org: SciPy.org — SciPy.org. Stand vom 22. Januar 2018, erreichbar unter <https://www.scipy.org/>, abgerufen am 24. Januar 2018.
- /SEH 16/ Seher, H., Navarro, M.: SITA, Version 0, A simulation and code testing assistant for TOUGH2 and MARNIE. GRS-400, 78 S., ISBN 978-3-944161-81-5, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH: Köln, Juni 2016.
- /SEH 17/ Seher, H., Weyand, T., Bracke, G.: Anwendung thermodynamischer Daten aus THEREDA mit TOUGHREACT. GRS-435, 95 S., ISBN 978-3-946607-17-5, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH: Köln, Februar 2017.
- /SI 13/ Si, H.: TetGen, A Quality Tetrahedral Mesh Generator and 3D Delaunay Triangulator, Version 1.5, User's Manual. Institute for Applied Analysis and Stochastics (WIAS), WIAS Technical Report, Nr. 13, 104 S.: Berlin, 2013.
- /SPY 19/ Spycher, N.: Description of MOPR-Jumpers for the test version of TOUGHREACT-BRINE. An Weyand, T., E-Mail, 12. August 2019.
- /STA 20/ Gesetz zur Suche und Auswahl eines Standortes für ein Endlager für hochradioaktive Abfälle (Standortauswahlgesetz – StandAG) in der Fassung vom 5. Mai 2017 (BGBl. I 2017, Nr. 26, S. 1074-1100), zuletzt geändert 19. Juni 2020 (BGBl. I 2020, Nr. 29, S. 1328–1370).

- /THU 15/ Thunderhead Engineering: PetraSim 2015, User Manual. 119 S.: Manhattan, USA, 2015.
- /VOI 12/ Voigt, W., Sukhanov, D., Wilhelm, S., Bok, F., Bube, C.: Solubility in selected ternary salt solutions with regard to the system of oceanic salts in presence of CO<sub>2</sub> and carbonates, Benchmark calculation, THEREDA Release 3. 41 S., 17. Februar 2012.
- /WEY 14/ Weyand, T., Bracke, G., Reichert, B.: PHREEQC – Modellierung des Transportes von CO<sub>2</sub> in hochsalinaren Lösungen eines Endlagers. GRS-323, 99 S., ISBN 978-3-944161-03-7, DOI 10.13140/2.1.2596.4489, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH: Köln, März 2014.
- /WEY 15/ Weyand, T., Bracke, G., Reichert, B.: A one-dimensional transport model of CO<sub>2</sub> in highly saline solutions in a final repository for radioactive waste. Goldschmidt Conference Abstracts, Nr. 3304: Prag, Tschechien, August 2015.
- /WEY 20/ Weyand, T., Bracke, G., Eckel, J.: Modelling the reactive transport of <sup>14</sup>C in a disposal for radioactive waste. Goldschmidt Conference Abstracts, DOI 10.46427/gold2020.2850: Honolulu, Juni 2020.
- /WIL 18/ Williams, T., Kelley, C.: Gnuplot 5.2, An Interactive Plotting Program. 5. April 2018.
- /WOL 03/ Wolery, T. W., Jarek, R. L.: Software User's Manual EQ 3/6, Version 8.0. U.S. Department of Energy (DOE), 10813-UM-8.0-00, 118 S.: Las Vegas, 1. Januar 2003.
- /XU 06/ Xu, T., Sonnenthal, E., Spycher, N., Pruess, K.: TOUGHREACT User's Guide, A Simulation Program for Nonisothermal Multiphase Reactive Geochemical Transport in Variably Saturated Geologic Media. Earth Sciences Division, Lawrence Berkeley National Laboratory, University of California, LBNL-55460, 195 S.: Berkeley, 1. Dezember 2006.

/XU 17/ Xu, T., Sonnenthal, E., Spycher, N., Zheng, L.: TOUGHREACT V3.32 Reference Manual, A Parallel Simulation Program for Non-Isothermal Multiphase Geochemical Reactive Transport. Lawrence Berkeley National Laboratory (LBNL), LBNL-DRAFT, März 2017.

/ZHA 06/ Zhang, G., Spycher, N., Xu, T., Sonnenthal, E., Steefel, C. I.: Reactive Geochemical Transport Modeling of Concentrated Aqueous Solutions, Supplement to TOUGHREACT User's Guide for the Pitzer Ion-Interaction Model. Earth Sciences Division, Lawrence Berkeley National Laboratory, University of California, LBNL-62718, 141 S.: Berkeley, 1. Juni 2006.

## Abbildungsverzeichnis

Abb. 2.1	Elemente des Netzwerkes in MARNIE2.....	4
Abb. 2.2	Räumliche Diskretisierung für die Verifikationstestfälle zur Entwicklung von MARNIE2. ....	6
Abb. 2.3	Restrukturierung der COMMON Blöcke am Beispiel von CAPAR1. Original (oben) und Ergebnis der Bearbeitung durch SPAG (unten) .....	8
Abb. 2.4	Funktion nach Grant für die relativen Permeabilitäten. Die Berechnung aus MARNIE2 ist im Vergleich zu Gl. (2.8) gezeigt. ....	14
Abb. 2.5	Funktion nach Corey für die relativen Permeabilitäten: Die Berechnung aus MARNIE2 ist im Vergleich zu Gl. (2.6) gezeigt. ....	14
Abb. 2.6	Funktion nach Fatt und Klikoff für die relativen Permeabilitäten: Die Berechnung aus MARNIE2 ist im Vergleich zu Gl. (2.9) gezeigt. ....	15
Abb. 2.7	Funktion nach van Genuchten für die relativen Permeabilitäten: Die Berechnung aus MARNIE2 ist im Vergleich zu Gl. (2.10) gezeigt. ....	15
Abb. 2.8	Kapillardruckfunktion nach van Genuchten: Die Berechnung aus MARNIE2 ist im Vergleich zu Gl. (2.11) gezeigt. ....	17
Abb. 2.9	Flüssigkeitssättigung für den Plausibilitätstestfall zum Kapillardruck.....	18
Abb. 2.10	Verlauf des Kapillardrucks für den Plausibilitätstestfall zum Kapillardruck.....	18
Abb. 2.11	Sättigungsverlauf für den Testfall I-1.....	25
Abb. 2.12	Ausgabe des Flags für den Rücksprung in der FEBE (oben) und Ausgabe für erfolgreichen Phasenwechsel (unten).....	26
Abb. 2.13	Massenfluss für die flüssige Phase über die Boxränder. Der Fluss auf dem 1. Rand des P-Moduls (P1-Rand1) entspricht der vorgegebenen Flussrandbedingung.....	27
Abb. 2.14	Massenfluss für die Gasphase auf den Boxrändern.....	27
Abb. 2.15	Druckverlauf für den Plausibilitätstestfall I-1.....	28
Abb. 2.16	Verlauf der Flüssigkeitssättigung für den Testfall III-1. Die senkrechten Linien markieren den Übergang von Gaseindringen zu zweiphasigem Transport.....	29
Abb. 2.17	Druckverlauf für den Testfall III-1. Die senkrechten Linien markieren den Übergang von Gaseindringen zu zweiphasigem Transport. ....	30

Abb. 2.18	Massenfluss der flüssigen Phase über die Boxränder für den Testfall III-1. Die senkrechten Linien markieren den Übergang von Gaseindringen zu zweiphasigem Transport. ....	30
Abb. 2.19	Massenfluss der gasförmigen Phase über die Boxränder für den Testfall III-1. Die senkrechten Linien markieren den Übergang von Gaseindringen zu zweiphasigem Transport. ....	31
Abb. 2.20	Verlauf der Flüssigkeitssättigung für den Testfall V-1 (Entsättigung einer Säule) .....	32
Abb. 2.21	Verlauf der Flüssigkeitssättigung für den Testfall Va-1 (Entsättigung einer Säule). ....	32
Abb. 3.1	Verifikationstests für Modul GCOMP: Spezifische Wärmekapazität $C_v$ bei konstantem Volumen .....	43
Abb. 3.2	Verifikationstests für Modul GCOMP: Molare Masse .....	44
Abb. 3.3	Verifikationstests für Modul GCOMP: Dynamische Gasviskosität .....	45
Abb. 3.4	Verifikationstests für Modul GCOMP: Inverse Henry-Konstante.....	46
Abb. 3.5	Verifikationstests zur Löslichkeitsbegrenzung des Moduls RN: Konzentration bei rein advektivem Transport, mit und ohne Anionenausschluss.....	47
Abb. 3.6	Verifikationstests zur Löslichkeitsbegrenzung des Moduls RN: Konzentration bei rein diffusivem Transport, mit und ohne Anionenausschluss.....	48
Abb. 3.7	Verifikationstests des Restarts für Modul COMP: Konvergenzrate.....	49
Abb. 3.8	Verifikationstests des Restarts für Modul COMP: Konvergenzrate.....	49
Abb. 3.9	Verifikationstests des Restarts für Modul CORFL: Alterationsgrad .....	50
Abb. 3.10	Verifikationstests des Restarts für Modul CORRO: Laugenanteil .....	50
Abb. 3.11	Verifikationstests des Restarts für Modul CORRO: Flüssigkeitssättigung .....	51
Abb. 3.12	Verifikationstests des Restarts für Modul CORRO: Korrosionsgrad .....	51
Abb. 3.13	Verifikationstests des Restarts für Modul CORRO: Behälterwassermasse .....	52
Abb. 3.14	Verifikationstests des Restarts für Modul DEGRA: Intrinsische Permeabilität.....	52

Abb. 3.15	Verifikationstests des Restarts für Modul FISS: Rissporosität.....	53
Abb. 3.16	Verifikationstests des Restarts für Modul FISS: Druckschwellwert.....	53
Abb. 3.17	Verifikationstests des Restarts für Modul PRLIM: Gasdruck .....	54
Abb. 3.18	Verifikationstests des Restarts für Modul PRLIM: Normvolumen des entwichenen Gases .....	54
Abb. 3.19	Verifikationstests des Restarts für Modul RN (Anionenausschluss): Tracerkonzentration im zugänglichen Porenraum.....	55
Abb. 3.20	Verifikationstests des Restarts für Modul RN (Zeitschrittweitenregelung): Nuklidmasse im System .....	55
Abb. 3.21	Verifikationstests des Restarts für Modul RN (Zeitschrittweitenregelung): Zeitschrittweitenentwicklung.....	56
Abb. 3.22	Verifikationstests des Restarts für Modul RN (Radionuklidtransport): Massenanteil des ersten Radionuklids.....	56
Abb. 3.23	Verifikationstests des Restarts für Modul RN (Radionuklidtransport): Massenanteil des ersten Radionuklids.....	57
Abb. 4.1	Beispiele dreidimensionaler strukturierter Voronoi-Gitter: a) kartesisches Gitter b) Gitter mit Radialsymmetrie c) orthogonales Gitter (aus /NAV 18a/) .....	62
Abb. 4.2	Hauptschritte der Gittergenerierung.....	71
Abb. 4.3	Seedpaar im Falle von Dreiecken, die auf einer Ebene liegen .....	79
Abb. 4.4	Sparsame Seedsetzung für eine Ecke .....	82
Abb. 4.5	Darstellung der Blattschnitte (Kacheln) der Topographischen Karte 1:100000 (TK100) von Niedersachsen (entnommen aus /LAN 17/) .....	96
Abb. 4.6	Perspektivische Darstellung der Daten „Basisflaechen“ der Kachel C3126 mit 5-facher Überhöhung der z-Koordinaten.....	97
Abb. 4.7	Perspektivische Darstellung der gesamten Daten der Kachel C3126 mit 5-facher Überhöhung der z-Koordinaten .....	98
Abb. 4.8	Perspektivische Darstellung vom Dreiecksnetz der Beispielschicht mit 20-facher Überhöhung der z-Koordinaten .....	99
Abb. 4.9	Render-Ansicht und Datentabelle bei aktiviertem Threshold-Filter für einen vorgegebenen Datenbereich (grauer Bereich liegt außerhalb des Datenbereiches).....	105

Abb. 4.10	Render-Ansicht mit aktiviertem Threshold-Filter zur Auswahl der Materialgebiete (grauer Bereich liegt außerhalb des Datenbereiches)...	106
Abb. 4.11	Render-Ansicht mit einem Linienplot der zeitlichen Entwicklung des Druckes für die Zelle, in der zum Zeitpunkt Null der maximale Druck gemessen wurde .....	107
Abb. 4.12	Render-Ansicht mit Vektoren und Linienplot für die zeitliche Entwicklung der Strömungsgeschwindigkeit in der Elementzelle mit der maximalen Geschwindigkeit .....	108
Abb. 4.13	Darstellung der Stromlinien für Vektoren, ausgehend von einem sphärischen Bereich am linken Modellrand.....	109
Abb. 5.1	Workflow zur Durchführung der Benchmark-Tests mit PHREEQC.....	115
Abb. 5.2	PHREEQC-Keyword <code>PITZER</code> .....	116
Abb. 5.3	Workflow zur Durchführung der Benchmark-Tests mit TOUGHREACT .....	117
Abb. 5.4	Ausschnitt aus einer exemplarischen Eingabedatei von <code>chemical.inp</code> (THEREDA-Parameterdatei 01, Benchmark-Test 01) .....	121
Abb. 5.5	Struktur der thermodynamischen Eingabe von TOUGHREACT v4 in der die Pitzer-Parameter nach den Oberflächenkomplexen eingefügt werden.....	129
Abb. 5.6	Eingabestruktur der Stützstellen $a_1$ bis $a_6$ für die Berechnung der temperaturabhängigen Pitzer-Parameter in TOUGHREACT v4 .....	130
Abb. 5.7	Molalitäten der aquatischen Spezies für das System K-Mg-SO <sub>4</sub> (Benchmark-Tests 01-03 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	132
Abb. 5.8	Molalitäten der aquatischen Spezies für das System K-Mg-SO <sub>4</sub> (Benchmark-Tests 04-06 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	132
Abb. 5.9	Molalitäten der aquatischen Spezies für das System K-Na-SO <sub>4</sub> (Benchmark-Tests 07-09 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	133
Abb. 5.10	Molalitäten der aquatischen Spezies für das System K-Na-SO <sub>4</sub> (Benchmark-Tests 10-12 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	133

Abb. 5.11	Molalitäten der aquatischen Spezies für das System Ca-Cl (Benchmark-Tests 13-15 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	134
Abb. 5.12	Molalitäten der aquatischen Spezies für das System K-SO <sub>4</sub> (Benchmark-Tests 16-17 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	134
Abb. 5.13	Molalitäten der aquatischen Spezies für das System K-SO <sub>4</sub> (Benchmark-Tests 18-19 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	135
Abb. 5.14	Molalitäten der aquatischen Spezies für das System K-SO <sub>4</sub> (Benchmark-Tests 20-21 der Parameterdatei 01) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	135
Abb. 5.15	Molalitäten der aquatischen Spezies von Nd in einer 0,1 molalen NaCl-Lösung (Benchmark-Tests 01a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH) <sub>3</sub> (am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	138
Abb. 5.16	Molalitäten der aquatischen Spezies von Nd in einer 5,6 molalen NaCl-Lösung (Benchmark-Tests 04a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH) <sub>3</sub> (am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	139
Abb. 5.17	Molalitäten der aquatischen Spezies von Nd in einer 0,25 molalen MgCl <sub>2</sub> -Lösung (Benchmark-Tests 07a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH) <sub>3</sub> (am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	139
Abb. 5.18	Molalitäten der aquatischen Spezies von Nd in einer 0,25 molalen CaCl <sub>2</sub> -Lösung (Benchmark-Tests 08a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH) <sub>3</sub> (am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	140
Abb. 5.19	Molalitäten der aquatischen Spezies von Nd in einer 3,86 molalen MgCl <sub>2</sub> -Lösung (Benchmark-Tests 13a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH) <sub>3</sub> (am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	141
Abb. 5.20	Molalitäten der aquatischen Spezies von Nd in einer 3,86 molalen CaCl <sub>2</sub> -Lösung (Benchmark-Tests 14a-c der Parameterdatei 02) im Gleichgewicht mit Nd(OH) <sub>3</sub> (am) berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	141

Abb. 5.21	Ergebnis der Vorrechnung des wiederholten Benchmark-Tests 13 aus Release 02.....	143
Abb. 5.22	Ergebnis der Berechnung der Festphasenreaktion des wiederholten Benchmark-Tests 13 aus Release 02 .....	143
Abb. 5.23	Molalitäten der aquatischen Spezies mit C (Benchmark-Tests 09 - 11 der Parameterdatei 03) zu unterschiedlichen $SI_{CO_2(g)}$ berechnet mit PHREEQC (Linien) und TOUGHREACT v4 (Kreuze) .....	146

## Tabellenverzeichnis

Tab. 2.1	In MARNIE implementierte Umschaltprozesse zusammen mit den jeweiligen Bedingungen und den genutzten logischen Flags .....	21
Tab. 2.2	Plausibilitätstestfälle für die in MARNIE2 implementierten Umschaltprozesse .....	24
Tab. 3.1	Restart-Dateien und gespeicherte Zustände.....	37
Tab. 4.1	Kurzbeschreibung vorhandener Gittergeneratoren und Programmbibliotheken zur Gittererstellung .....	65
Tab. 4.2	Algorithmus zur Generierung von Knotenseeds.....	84
Tab. 4.3	Seed-Suche durch kantenorientierte Spiegelung .....	88
Tab. 4.4	Kantenvervollständigung.....	89
Tab. 4.5	Algorithmus zur Generierung von Schutzseeds .....	91
Tab. 4.6	Algorithmus zur Generierung von Körper-Seeds.....	92
Tab. 4.7	Algorithmus zur Erstellung des Voronoi-Gitters.....	93
Tab. 4.8	Algorithmus zur Erstellung der TOUGH2-Eingabe mit inaktiven Elementen .....	94
Tab. 4.9	Übersicht über Anzahl an Stützstellen (Punkte), Kanten und Flächen der einzelnen Dreiecksnetze und des zusammengeführten Gesamtnetzes .....	101
Tab. 5.1	Eingabeparameter der Benchmark-Tests .....	124
Tab. 5.2	Reihenfolge der Koeffizienten $i$ in AKCOES( $i$ ) in der thermodynamischen Eingabedatei von TOUGHREACT.....	128
Tab. 5.3	Berechnung der Koeffizienten $a_1 - a_6$ in TOUGHREACT v4 mit einer Referenztemperatur $T_0$ von 298.15 K aus den Koeffizienten $a$ bis $f$ aus THEREDA.....	131
Tab. 5.4	Eingabeparameter und Nummerierung der Benchmark-Tests zu Release 02 .....	137
Tab. 5.5	Eingabeparameter der Benchmark-Tests zur Parameterdatei 03.....	145
Tab. 5.6	Molalitäten der aquatischen kohlenstoffhaltigen Spezies als Ergebnis der Benchmark-Berechnungen für Release 03 mit PHREEQC (PRQ) und TOUGHREACT v4 (TR).....	147

## **Danksagung**

Wir danken dem Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit für die Förderung des Vorhabens.

**Gesellschaft für Anlagen-  
und Reaktorsicherheit  
(GRS) gGmbH**

Schwertnergasse 1  
**50667 Köln**  
Telefon +49 221 2068-0  
Telefax +49 221 2068-888

Forschungszentrum  
Boltzmannstraße 14  
**85748 Garching b. München**  
Telefon +49 89 32004-0  
Telefax +49 89 32004-300

Kurfürstendamm 200  
**10719 Berlin**  
Telefon +49 30 88589-0  
Telefax +49 30 88589-111

Theodor-Heuss-Straße 4  
**38122 Braunschweig**  
Telefon +49 531 8012-0  
Telefax +49 531 8012-200

[www.grs.de](http://www.grs.de)