

Visualisierung der Dichte im quinären System mariner Evaporite

Studienarbeit

eingereicht bei
Prof. Dr. Kai Hormann
Institut für Informatik
Technische Universität Clausthal

12. April 2006

vorgelegt von
Robert F. P. Hartmann
Matrikelnummer: 302300



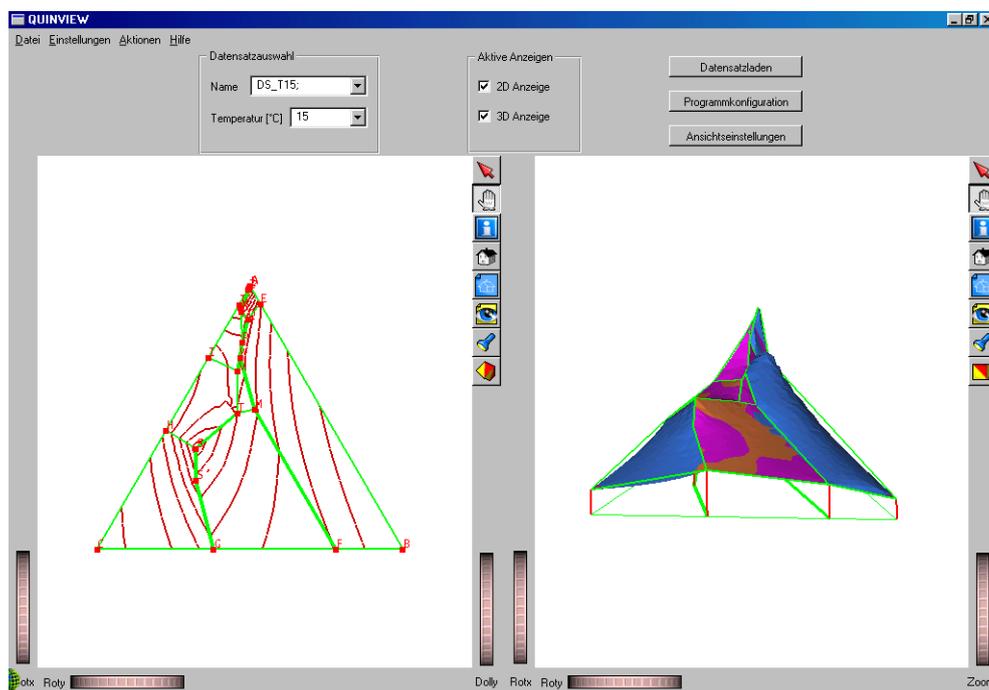
TU Clausthal

Studienarbeit

Visualisierung der Dichte im quinären System mariner Evaporite

Robert F.P. Hartmann

12. April 2006



Screenshot: Programm Quinview

BLAISE PASCAL (1623 - 1662) in **Von der Methode der geometrischen, d.h. methodischen und vollkommenen Beweise**^a:

„Ich will also das, was ein Beweis ist, [...] verständlich machen [...] Vorher muß ich aber die Idee von einer noch höheren und vollkommeneren Methode geben, wohin die Menschen jedoch niemals gelangen können: denn was über die Geometrie geht, geht über uns hinaus; trotzdem ist es notwendig, etwas davon zu sagen, obgleich es unmöglich zu verwirklichen ist. Diese wahre Methode, welche Beweise in höchster Vollendung führen würde, wenn es möglich wäre, sie zu erreichen, würde in zwei Hauptsachen bestehen:

Einmal, keinen Begriff zu verwenden, dessen Sinn man nicht vorher deutlich erklärt hätte, zum anderen, niemals eine Behauptung aufzustellen, die man nicht aus schon bekannten Wahrheiten bewiesen hätte; d.h. mit einem Wort, alle Begriffe zu definieren und alle Behauptungen zu beweisen.“

^a Zitat aus Karl Vorländer: **Geschichte der Philosophie, Band 4, Philosophie der Neuzeit**; Rowohlt Taschenbuchverlag Band 261/62, 1966 München

Das Programm mit dem Quelltext liegt auf CD-Rom
dem Institut für Mineralogie sowie dem Institut für Informatik vor.

Inhaltsverzeichnis

Inhaltsverzeichnis	ii
1 Einleitung, Aufgabenstellung	1
1.1 Einleitende Worte	1
1.2 Aufgabenstellung	2
1.2.1 Institut für Mineralogie, Grundlegende Aufgabenstellung	2
1.2.2 Institut für Informatik, Erweiterung der Aufgabenstellung	2
2 Physiko-chemische Grundlage	3
Konzentrationseinheiten, Konzentration, Aktivität, Aktivitätskoeffizienten	3
Debye-Hückelsches Gesetz	3
Lewis-Randall-Ionenstärke	3
Erweiterte Debye-Hückel-Gleichung	4
Güntelberg-Formel	4
Davies-Formel	4
Thermodynamische Gleichgewichtskonstante	4
Thermodynamisches Löslichkeitsprodukt	5
Stöchiometrisches Löslichkeitsprodukt	5
Zusammenhang zwischen K_s und K_s^0	5
3 Berechnungsverfahren der Lösungsdichte	6
3.1 Definitionen und Annahmen	6
3.2 Drei Verfahren	6
3.2.1 Bestimmung der Datengrundlage	6
3.2.2 Konversion der Datengrundlage	7
3.2.3 D'ANS-Verfahren	9
3.2.4 MONNIN-Verfahren	10
3.2.5 EZROCHI-Verfahren	10
4 Quinäres System als 2D-Projektion	11
4.1 Das Jänecke-Dreieck	12
4.1.1 Baryzentrische Koordinaten	13
5 Quinäres System als 3D-Objekt	14
5.1 Dreieckszerlegung	14

5.1.1	Forderung an eine allgemeine Triangulierung	14
5.1.2	Regelmäßige Dreieckszerlegung	14
5.1.3	Delaunay-Triangulation	15
5.2	Problem: 2D \mapsto 3D	16
5.3	Mögliche Lösungen des Problems	17
5.3.1	Linearer Ansatz	18
5.3.2	Nichtlinearer Ansatz	18
5.3.3	Feinmaschigkeit vs. Stützpunkte	23
5.4	Isolinien aus 3D Oberflächengitternetz	23
5.4.1	Klassifikation von Dreiecken für Isolinien	24
6	Dateiparser	26
6.1	Quinviewkonfigurationsdatei	26
6.2	Monninparameterdatei	26
6.3	Datensatzdatei	27
6.3.1	BNF der Datensatzdatei	27
7	Abschließende Gesichtspunkte	28
7.1	Mineralogische Erkenntnisse	28
7.2	Mathematische Erkenntnisse	29
7.3	Programmiertechnische Erkenntnisse	30
7.4	Ausblick	30
7.5	Schlusswort	31
	Abbildungsverzeichnis	32
	Literaturverzeichnis	34

1 Einleitung, Aufgabenstellung

1.1 Einleitende Worte

Bei der Bildung von marinen Evaporiten durch Eindunstung von Meerwasser entstehen Lösungen, die neben H_2O vor allem Na , Mg , K , Cl und SO_4 enthalten. Wenn die Lösungen neben **Halit** ($NaCl$) an mindestens einem weiteren Mineral gesättigt sind, liegen sie auf den Flächen des quinären Systems der marinen Evaporite, das typischerweise in Form einer Projektion der Flächen ins Dreieck $SO_4, 2K$ und Mg dargestellt wird. Bedingt durch die unterschiedlichen Mengen der gelösten Ionen variieren die Dichten der Lösungen im quinären System im Bereich von 1.2 bis $1.4 \frac{g}{l}$. Diese Dichteunterschiede sind sowohl bei natürlichen als auch bei technischen Prozessen von erheblicher Bedeutung, weil durch Dichte-getriebene Konvektion und die Ausbildung stabiler Schichtungen Reaktionen sowohl beschleunigt als auch gebremst werden können.

Die Lage der Sättigungsflächen im quinären System ist gut bekannt und wird durch die experimentell bestimmten Punkte P_e definiert, in denen neben Halit drei weitere Minerale stabil sind bzw. der Gehalt einer Komponente null ist und zwei weitere Minerale stabil sind.

Ist bei Punkten eine Komponente null, so handelt es sich um Punkte in einem 4-Komponenten-System, das einen Freiheitsgrad weniger hat. Da solche Systeme experimentell einfacher zu handhaben sind, sind die Punkte dieser Randsysteme normalerweise besonders gut bekannt.

Ausgehend von den Punkten P_e können die Zusammensetzungen von Lösungen in jedem Punkt des quinären Systems der marinen Evaporite über Vektoren $\in \mathbb{R}^5$ berechnet werden. Zur Berechnung der Dichten von salinaren, d.h. salzigen, Lösungen existieren verschiedene Methoden (Algorithmen), die auf empirischen Daten basieren und auf dem Prinzip der Interpolation beruhen.

1.2 Aufgabenstellung

1.2.1 Institut für Mineralogie, Grundlegende Aufgabenstellung

Ziel der Studienarbeit ist die Entwicklung eines Computerprogrammes, das die Zusammensetzung von Lösungen in einem feinen Raster berechnet, diese Daten zur Berechnung der Dichte benutzt und dann die Dichten graphisch in der Dreiecks-Projektion des quinären Systems darstellt.

Zur Berechnung sollen die Daten des Systems für verschiedene Temperaturen (15° , 25° , 35° und $55^\circ C$) genutzt werden.

Da es bedingt durch neue Untersuchungen zukünftig zu Änderungen an den Datensätzen kommen kann, sollen diese (z.B. durch Editieren von Text-Dateien) veränderbar sein.

Zur Berechnung der Dichten sollen drei verschiedene Algorithmen verwendet werden, zwischen denen gewechselt werden kann.

Die graphische Darstellung soll in Form von „Höhenlinien“ erfolgen, wobei neben der Darstellung des Gesamtsystems eine Ausschnittsvergrößerung bis Faktor 10 möglich sein sollte.

1.2.2 Institut für Informatik, Erweiterung der Aufgabenstellung

Das zu schreibende Programm soll neben den durch das Institut für Mineralogie geforderten Funktionen zusätzlich noch weitere Möglichkeiten bieten.

Es soll möglich gemacht werden, die Dichteinformationen in einem 3D-Raum über der dreieckigen 2D-Grundfläche des quinären Systems mariner Evaporite wahlweise als gefüllte Oberflächen (surface) oder Netz (surface mesh) anzuzeigen; alternativ sollen auch nur die interpolierten Punkte, die dann für die Oberfläche genutzt werden, anzeigbar sein.

Bei der Dokumentation des Quellcodes soll darauf geachtet werden, dass man mittels **doxygen** eine Klassendokumentation extrahieren kann.

Zur Bewältigung der Programmieraufgabe soll genutzt werden:

Programmiersprache: MS Visual C++ 6.0

GUI-Programmierung: QT-Library der Firma Trolltech
(<http://www.trolltech.com>)

3D-Visualisierung: Coin-Library der Firma Coin3D
(<http://www.coin3d.org>)

Quelldokumentation: doxygen (<http://doxygen.org>)

Es sollen die non-commercial/opensource Versionen der beiden Bibliotheken eingesetzt werden, da der produzierte Quellcode ebenfalls Open Source sein wird.

2 Physiko-chemische Grundlage

Als Konzentrationseinheiten werden bei gelösten Stoffen **Molalität** bzw. **Molarität** (besser nach IUPAC¹, 1970: **Stoffmengenkonzentration** bzw. **Konzentration**) verwendet.

Eine **einmolale Lösung** [1*m*] ist durch die Stoffmenge $1 \frac{\text{mol}}{\text{kg Lösungsmittel}}$ und eine **einmolare Lösung** [1*M*] durch die Stoffmenge $1 \frac{\text{mol}}{\text{l Lösung}}$ definiert.

Die in realen Systemen wirksame Konzentration wird als **Aktivität** *a* bezeichnet. Sie ergibt sich aus der **Konzentration** *c* durch Multiplikation mit einem als **Aktivitätskoeffizienten** γ – manchmal auch *f_a* – bezeichneten Korrekturfaktor; also – siehe auch [ST90]:

$$a = \gamma \cdot c \quad (2.1)$$

Aktivitätskoeffizienten von Ionen lassen sich für **verdünnte wässrige Lösungen** ($10^{-3}m \leq xm < 10^{-2}m$), wie in [ST90] beschrieben, theoretisch aus der interionischen Wechselwirkung nach P. DEBYE und E. HÜCKEL ableiten. Hier sind die Aktivitätskoeffizienten eine Funktion der Ionenladung *z* und der Ionenstärke *I*.

Debye-Hückelsches Gesetz [FB72]:

$$\lg \gamma = -A \cdot z_i^2 \cdot \sqrt{I}$$

wobei $A = \frac{1}{2,303} \cdot \frac{e^2}{2 \cdot \varepsilon \cdot k \cdot T} \cdot \sqrt{\frac{8\pi \cdot e^2 \cdot N}{1000 \cdot \varepsilon \cdot k \cdot T}}$ (2.2)

- mit
- e* Ladung des Elektrons
 - N* LOSCHMIDT-Konstante, auch AVOGADRO-Konstante genannt, = $6.022136 \cdot 10^{23} \frac{1}{\text{mol}}$ [BBD+05], [Gär93]
 - k* BOLTZMANN-Konstante = $1.380658 \cdot 10^{-23} \frac{\text{J}}{\text{K}}$ [BBD+05]
 - ε Dielektrizitätskonstante des reinen Lösungsmittels
 - T* absolute Temperatur in Kelvin

Die Ionenstärke *I* kann nach LEWIS und RANDALL berechnet werden [FB72]:

$$I = \frac{1}{2} \sum c_i z_i^2 \quad (2.3)$$

¹International Union of Pure and Applied Chemistry

Für **stärker konzentrierte Lösungen** gibt es die **erweiterte Debye-Hückel-Gleichung** – siehe [FB72] und [Voi90]:

$$\lg \gamma = -A \cdot z_i^2 \cdot \frac{\sqrt{I}}{1 + B \cdot r \cdot \sqrt{I}} + c_i I \quad (2.4)$$

wobei $B = \frac{50,3}{\sqrt{\varepsilon \cdot T}}$

und r der effektive Radius des solvatisierten Ions in Ångström ($1\text{Å} = 10^{-10}\text{Meter} = 0,1\text{nm}$) ist.

In [Voi90] führt VOIGT auf, dass das letzte Glied dieser Formel nach ALEKIN(1962) für Wasser mit einer Mineralisation von $1,2 - 2,4 \frac{g}{kg}$ vernachlässigt werden könnte.

Bei Lösungen mit mehreren Elektrolyten ist die **Güntelberg-Formel** nützlich (siehe [SS94]). GÜNTEMBERG vereinfacht die erweiterte Debye-Hückel-Gleichung, indem $B \cdot r = 1$ gesetzt und das letzte Glied vernachlässigt wurde. Allerdings ist diese Vereinfachung nur gültig bei Ionenstärken bis zu $0,1M$.

Die **Davies-Formel** lässt sich auf Lösungen mit bis zu $0,5M$ anwenden.

$$\lg \gamma = -A \cdot z_i^2 \cdot \left(\frac{\sqrt{I}}{1 + \sqrt{I}} - 0,2I \right) \quad (2.5)$$

Der Korrekturfaktor $0,2$ in der Formel von DAVIES wurde zu einem späteren Zeitpunkt auf $0,3$ gesetzt.

Die Ermittlung von Aktivitätskoeffizienten im Meerwasser erfordert wegen der hohen Ionenkonzentration ($0,723m$) die Berücksichtigung spezifischer Ionenwechselwirkungen, so genannte „Salzeffekte“.

Für **hochmineralisierte Wässer**, z.B. Salzseen, Grundwässer von Trockensteppen und Wüsten sowie Tiefenwässern, ist der Zusammenhang zwischen Aktivitätskoeffizienten und Ionenstärken nicht mehr einfach bestimmbar [ST90]. Diese sind jedoch im Vergleich zu den Lösungen im quinären System verdünnt [TUC05]. Nach Aussage von Herrn Prof. Dr. A. ADAM (Institut für Anorganische und Analytische Chemie, TU Clausthal) im WS2005/06 bei einem persönlichen Gespräch nach seiner Vorlesung „Chemie wässriger Lösungen“ sei der Zusammenhang eigentlich noch nicht wirklich erfasst, und es sei noch viel Grundlagenforschung notwendig bis man ein formeltaugliches Allgemeinverhalten erkennen könne. Allerdings gibt es spezielle Lösungsarten, also Sonderfälle, bei denen man das Verhalten soweit modellieren kann, dass sie konsistent gegenüber den Messungen sind. So sind für die in salinaren Lösungen des quinären Systems die **Pitzer-Gleichungen** als geeignetes Modell angesehen². [Voi01]

²Es ist aktuelle Forschungsarbeit das Pitzer-Modell auszubauen und neue Einsatzbereiche zu finden.

Bei der **elektrolytischen Dissoziation** des Stoffes AB gilt die thermodynamische Gleichgewichtskonstante K_a

$$K_a = \frac{a_{A^+} \cdot a_{B^-}}{a_{AB}} \quad (2.6)$$

$$A_m B_n \rightleftharpoons mA + nB$$

Da die Aktivität von festen Elektrolyten AB definitionsgemäß gleich eins ist ($a_{AB} = 1$), gilt nach [NW72]:

$$a_{A^+}^m \cdot a_{B^-}^n = K_s^0 \quad (2.7)$$

K_s^0 wird **thermodynamisches Löslichkeitsprodukt** genannt und ist nur von der Temperatur abhängig, nicht aber von der Art und Menge der in der Lösung vorhandenen Stoffe.

Allerdings ist das **stöchiometrische Löslichkeitsprodukt** K_s abhängig von Temperatur und Ionenstärke – nach [NW72] gilt:

$$K_s = c_{A^+}^m \cdot c_{B^-}^n \quad (2.8)$$

Um einen direkten Zusammenhang zwischen den beiden Löslichkeitsprodukten aufzuzeigen, kann man diese über die Aktivitäten verknüpfen; dabei entstehen folgende Gleichungen:

$$K_s^0 = a_{A^+}^m \cdot a_{B^-}^n = (\gamma_{A^+} \cdot c_{A^+})^m \cdot (\gamma_{B^-} \cdot c_{B^-})^n \quad (2.9)$$

$$= (\gamma_{A^+}^m \cdot \gamma_{B^-}^n) \cdot (c_{A^+}^m \cdot c_{B^-}^n)$$

$$\frac{K_s^0}{\gamma_{A^+}^m \cdot \gamma_{B^-}^n} = c_{A^+}^m \cdot c_{B^-}^n = K_s \quad (2.10)$$

3 Berechnungsverfahren der Lösungsdichte

3.1 Definitionen und Annahmen

Die Dichte ρ in $[\frac{kg}{l}]$ kann sich je nach Lösungsart schon innerhalb der Lösung unterscheiden, dann wäre sie inhomogen.

Im Folgenden wird angenommen, dass die Lösung homogen ist, d.h. an jeder Position besitzt die Lösung die gleiche Dichte.

Definition:

- Ein **binäres Stoffsystem** oder auch **binäres System** ist ein System bestehend aus einem Lösungsmittel und einem Stoff, welcher in **zwei Ionenarten** dissoziiert. So bildet $NaCl$ gelöst in reinem Wasser das binäre System Na^+Cl^- . Mit binärem System ist hier also nicht das binäre Zahlensystem gemeint.
- Ein **ternäres Stoffsystem** besitzt **drei Ionenarten**, so bilden $NaCl$ und KCl in H_2O gelöst das trinäre System $Na^+K^+Cl^-$.
- Ein **n-äres Stoffsystem** besitzt demnach n verschiedene Ionenarten.

3.2 Drei Verfahren

Es gibt u.a. drei Methoden nach denen man die Dichte der Salzlösungen rechnerisch bestimmen kann. Diese drei Verfahren benutzen unterschiedliche Datengrundlagen und liefern zum Teil verschiedene Ergebnisse, zumindest ab einer bestimmten Stelle nach dem Komma.

3.2.1 Bestimmung der Datengrundlage

Bei gegebener salzhaltiger realer Flüssigkeit, die den Bedingungen „ $NaCl$ gesättigt und weiteres Mineral stabil“ genügt, muss zunächst mit Versuchen die Zusammensetzung bestimmt werden. Die Messung erfolgt mit Verfahren, die Menge pro Volumen oder Gewichte liefern. Bei bekannten Flüssigkeiten lassen sich sicherlich entsprechende Werte in der Literatur finden. Das Programm `Quinview` liest die Anzahl der gelösten Ionen

und deren Häufigkeit in $\frac{mol}{1000molH_2O}$ aus einer Datei, z.B. `original.qrh`, aus und arbeitet mit diesen.

3.2.2 Konversion der Datengrundlage

1. Aus den Häufigkeiten in $[\frac{mol}{1000molH_2O}]$ der Verbindungen und deren Molgewichte lässt sich der prozentuale Massenanteil dieser gelösten Verbindungen rechnerisch bestimmen.
(Implementierung siehe „`void chemieObj::konv_mol_in_mass_VB(void)`“ in `chemie.cpp`)
 2. Aus den Häufigkeiten der Salzverbindungen lassen sich direkt die Häufigkeiten der verschiedenen Ionen ebenfalls in $[\frac{mol}{1000molH_2O}]$ bestimmen.
(Implementierung siehe „`void chemieObj::konv_Verb_Ion_Mol(void)`“ in `chemie.cpp`)
 3. Aus den Häufigkeiten der Ionen lassen sich unter Berücksichtigung der entsprechenden Ionengewichte der prozentuale Massenanteil der Anionen und Kationen in Gewichtsprozent berechnen.
(Implementierung siehe „`void chemieObj::konv_mol_in_mass_AK(void)`“ in `chemie.cpp`)
 4. Aus den Häufigkeiten der Ionen können die Häufigkeiten der (fiktiven) Verbindungen ermittelt werden.
Die Implementierung (in „`void chemieObj::konv_Ion_Verb_Mol(void)`“ in `chemie.cpp`) ist relativ verzweigt und lässt sich übersichtlich und schön im folgenden Diagramm darstellen. (Siehe Abbildung 3.1.)
-

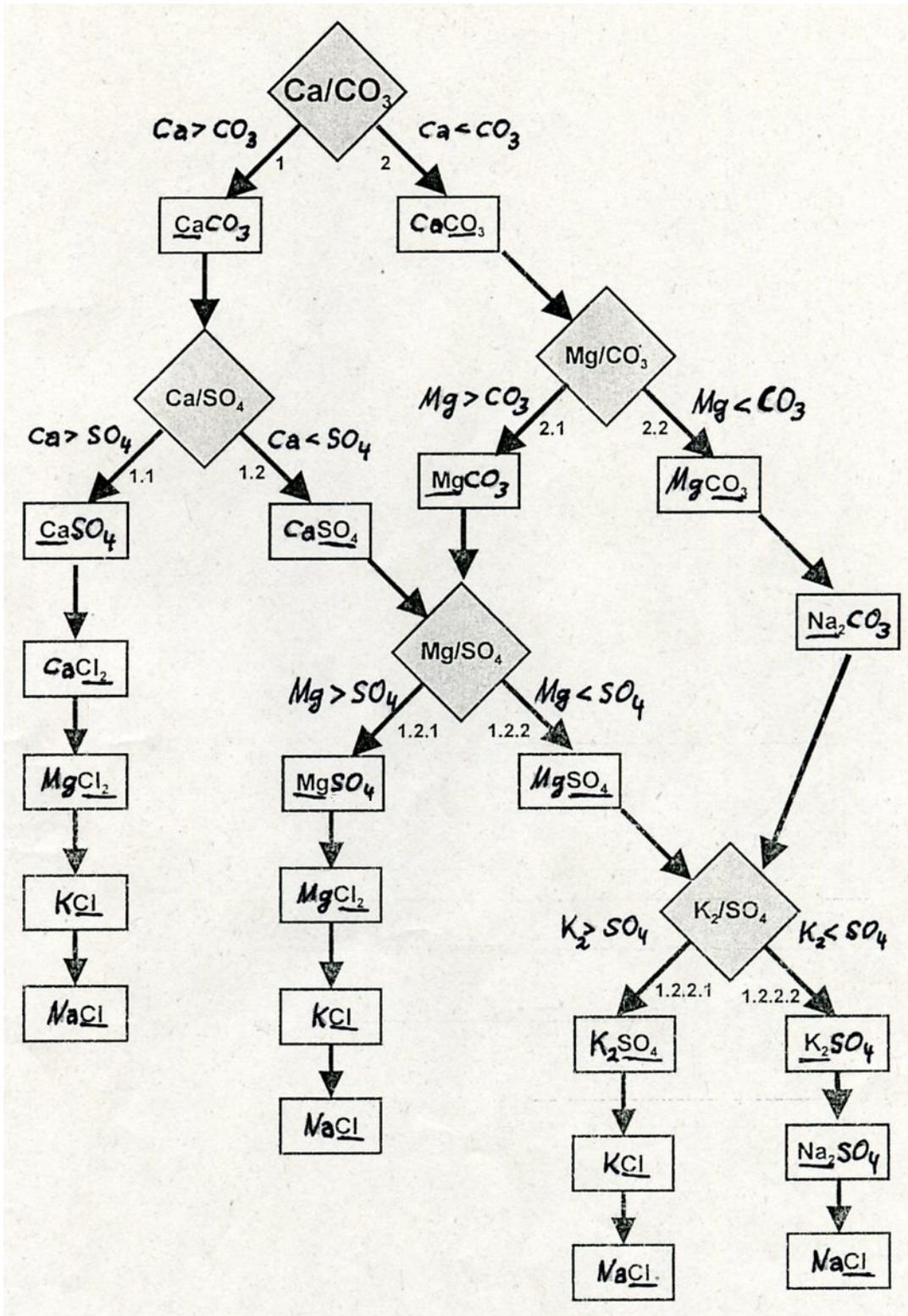


Abbildung 3.1: Flussdiagramm aus [TUC05]: Berechnung von fiktiven Verbindungen aus Anionen/Kationen

3.2.3 D'Ans-Verfahren

Diese Methode, erstmals bei D'ANS (1940) und bei v. BORSTEL (1992) beschrieben, wurde in Details verbessert [TUC05].

Input: Dieses Verfahren benötigt Information über den **prozentualen Massenanteil** der gelösten Salzverbindungen.

Prinzipiell geht man bei der Bestimmung der Gesamtdichte von den Dichten der jeweiligen binären Systeme aus. Zunächst werden die möglichen Verbindungen nach einem Kriterium sortiert. Meiner Meinung nach handelt es sich bei dem Sortierkriterium um die Löslichkeit bzw. die Lösbarkeit. Die Reihenfolge, in der die binären Systeme betrachtet werden sollen, ist in [TUC05] angegeben als:

$CaSO_4 \mapsto K_2SO_4 \mapsto Na_2CO_3 \mapsto Na_2SO_4 \mapsto KCl \mapsto NaCl \mapsto MgSO_4 \mapsto MgCl_2 \mapsto CaCl_2$

Allerdings entfallen bei den in dieser Studienarbeit betrachteten Lösungen $CaSO_4$, K_2SO_4 , Na_2CO_3 , $CaCl_2$, da diese Salze sich laut Aufgabenstellung nicht in der Lösung befinden.

Diese Dichten der binären Systeme wurden in Abhängigkeit der Konzentration und die Konzentration in Abhängigkeit der Dichten in ein Koordinatensystem aufgetragen und gefittet.

Oder anders ausgedrückt:

Das zuerst zu betrachtende binäre System ist das Na_2SO_4 -System und das letzte das $MgCl_2$ -System.

Für jedes System werden zwei Schritte durchgeführt; dabei startet der iterative Berechnungsprozess mit der Dichte von Wasser $\rho = 0.99823$ und mit Null für die Verbindungskonzentration:

- $k_{neu} = f_k(d_{alt}) + \text{MasseanteilVB}$
- $d_{neu} = f_d(k_{neu})$

d_{alt} ist beim ersten binären System die Dichte von Wasser, ansonsten die Dichte, die durch das direkt zuvor betrachtete binäre System bestimmt wurde. f_k gibt die Konzentration einer Verbindung zu einer Dichte d_{alt} aus, d.h. man fragt: „Wieviel ist von der Verbindung des betrachteten binären Systems nötig, um Wasser auf die Dichte d_{alt} zu bringen?“ Die neue Konzentration setzt sich aus der so berechneten „alten“ Konzentration und dem tatsächlichen Masseanteil der Verbindung zusammen. Mit dieser neuen Konzentration lässt sich die neue Dichte bestimmen. Beim Systemwechsel wird diese dann zur alten Dichte.

Achtung: $f_k()$, $f_d()$ sind Polynome bis zum zweiten Grad, die sich aber von System zu System verändern; so müsste man eigentlich noch eine Indizierung der Funktionen durchführen. Wegen der Lesbarkeit habe ich davon abgesehen, da die Funktionen hier nicht weiter betrachtet werden müssen.

(**Implementierung** siehe „double chemieObj::generate_DAns_dichte(void)“ in *chemie.cpp*)

3.2.4 Monnin-Verfahren

Basierend auf [Mon89] hat CHRISTOPHE MONNIN 1993 die Methode in Fortran implementiert. Die Methode stützt sich auf das PITZER Ioneninteraktionsmodell für Volumenanteile („volumetric properties“) von komplizierten Elektrolytlösungen.

Input: Dieses Verfahren benötigt Information über den **prozentualen Massenanteil** der Anionen und Kationen sowie die **Pitzer-Koeffizienten**. Es ist ein temperaturabhängiges Verfahren, d.h., die empirischen Pitzer-Koeffizienten können nur so benutzt werden, wie es getan wird, wenn die **Temperatur** als 25°C konstant angenommen wird.

(**Implementierung** siehe „`double chemieObj::generate_Monnin_dichte(monninparameter* monninparams)`“ in *chemie.cpp*)

Das thermodynamische Pitzer-Modell zu erklären, würde den Rahmen meiner Studienarbeit sprengen, daher verweise ich nur auf die entsprechende nach Jahreszahl sortierte Literatur.

- K. S. Pitzer, J. Phys. Chem. 77, 268 (1973).
- K. S. Pitzer und G. Mayorga, J. Phys. Chem. 77, 2300 (1973).
- K. S. Pitzer, Adv. Mineral. 17, 97 (1987).
- K. S. Pitzer, Activity Coefficients in Electrolyte Solutions 2nd ed., CRC Press, Boca Raton(1991).
- Christophe Monnin, „A thermodynamic model for the solubility of barite and celestite in electrolyte solutions and seawater to 200°C and to 1 kbar“ in Chemical Geology Vol. 153 (1-4) pp. 187-209, 01-January-1999
- Christophe Monnin, Catherine Jeandel, Thierry Cattaldo, Frank Dehairs, „The marine barite saturation state of the world’s oceans“ , Marine Chemistry 65, 253–261, 1999
- W. Voigt, Solubility equilibria in multicomponent oceanic salt systems from $t = 0$ to 200°C . Model parameterization and databases, Pure Appl. Chem., Vol. 73, No. 5, pp. 831–844, 2001.
- Nicolas Papaiconomou, „Thermodynamic modelling of industrial relevant electrolyte solutions“, Promotionsarbeit 2003 an der Universität Regensburg.

3.2.5 Ezrochi-Verfahren

Input: Dieses Verfahren ist temperaturvariabel, und benötigt als solches Information über den **prozentualen Massenanteil** der gelösten Salzverbindungen und die **Temperatur**. Die Ezrochi-Formel ist nach [DKS03] nur gültig im Temperaturbereich $0 - 100^{\circ}\text{C}$.

$$\lg \frac{\varrho}{\varrho_0} = \sum_{k=2} A_k \cdot g_k \quad (3.1)$$

mit ϱ Dichte der Lösung
 ϱ_0 Dichte des reinen Wassers
 A_k temperaturabhängiger Korrelationskoeffizient für die Komponente k
 g_k Massenbruch der Komponente k

(**Implementierung** siehe „`double chemieObj::generate_Ezrochi_dichte(double Temp)`“ in *chemie.cpp*)

4 Quinäres System als 2D-Projektion

Das quinäre System
 $\text{Na} - \text{K} - \text{Mg} - \text{Cl} - \text{SO}_4 - \text{H}_2\text{O}$
 Oder $\text{NaCl} - \text{MgCl}_2 - \text{KCl} - \text{Na}_2\text{SO}_4 - \text{H}_2\text{O}$
 Graphische Darstellung

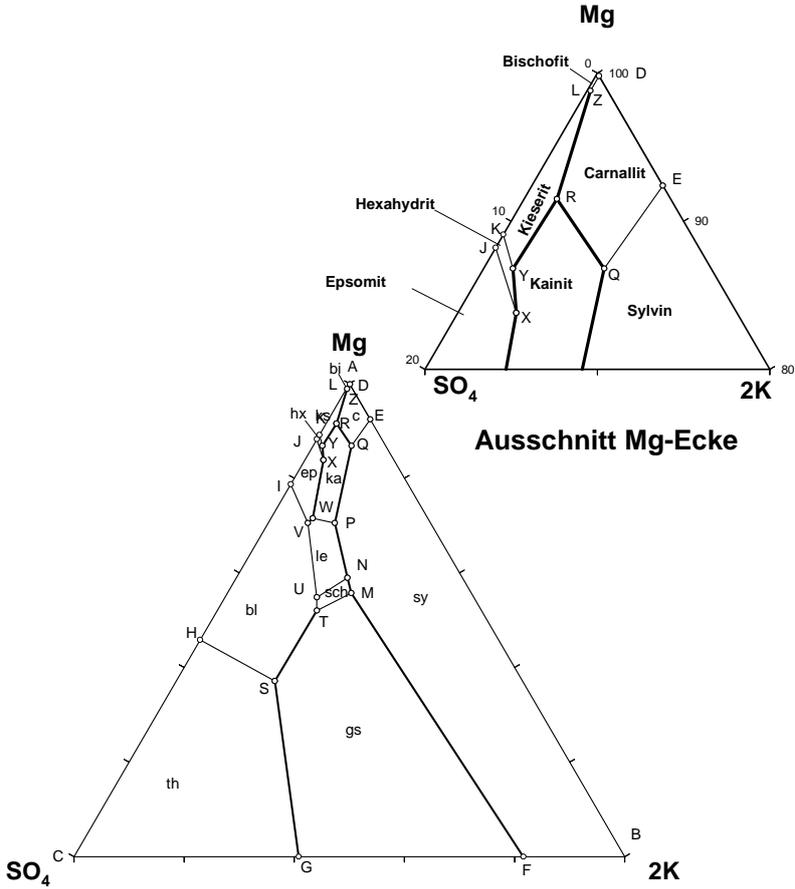


Abbildung 4.1: Jänecke-Dreieck (Graphik aus [Sie])

4.1 Das Jänecke-Dreieck

Das Jänecke-Dreieck stellt als Dreistoffdiagramm die drei Komponenten $SO_4 - 2K - Mg$ des quinären Systems da. Das Jänecke-Dreieck $[SO_4 - 2K - Mg]$ wird häufig als gleichseitiges Dreieck dargestellt. Die Länge einer Dreiecksseite kann mit 2 angenommen werden. Ohne Beschränkung der Allgemeinheit liege der Eckpunkt SO_4 im Ursprung $(0; 0)$ eines karthesischen Koordinatensystems. Der Punkt $2K$ liege bei $(2; 0)$. Der Punkt Mg liege oberhalb der erstgenannten, also bei $(1; \sqrt{3})$.

In dieses Jänecke-Dreieck wird das quinäre System hinein projiziert. Mathematisch betrachtet geschieht diese Projektion durch Anwendung von vier verschiedenen Abbildungsvorschriften.

$$f_i : \mathbb{R}^5 \rightarrow \mathbb{R}^5; f_i(\vec{v}) = \vec{v} \quad (4.1)$$

$$f_j : \mathbb{R}^5 \rightarrow \mathbb{R}^5; f_j(\vec{v}) = \vec{j} \quad (4.2)$$

$$f_b : \mathbb{R}^5 \rightarrow \mathbb{R}^3; f_b(\vec{j}) = \vec{b} \quad (4.3)$$

$$f_k : \mathbb{R}^3 \rightarrow \mathbb{R}^2; f_k(\vec{b}) = \vec{k} \quad (4.4)$$

Somit gilt

$$\vec{k} = f_k(f_b(f_j(f_i(\vec{v})))) \quad (4.5)$$

Erklärung:

1. f_i ist die mathematische Beschreibung der chemischen Dissoziation, bei der die Verbindungen der fünf Salze – $NaCl$, KCl , $MgCl_2$, $MgSO_4$ und Na_2SO_4 – in $1000molH_2O$ aufgelöst werden.
2. f_j konvertiert die Ionenanzahlen in Jänecke-Einheiten (= Jänecke%), d.h. sie führt eine Normierung der Ionenarten auf die Summe der Magnesiumionen Mg^{2+} , Kaliumionen $2K^+$ und Sulfationen SO_4^{2-} durch. Offensichtlich ist nach dieser Normierung die Summe der drei Ionen (Mg^{2+} , $2K^+$, SO_4^{2-}) 100%. Die Informationen über die Chloridionen sind implizit über die Ionenneutralität gegeben und werden daher durch die Wassermenge, ebenfalls in Jänecke%, ersetzt.
3. f_b bildet die Jänecke-Einheiten auf baryzentrische Dreieckskoordinaten ab.
4. f_k überführt die baryzentrischen Koordinaten in karthesische Koordinaten.

allgemeine Anwendung:

Sei \vec{v} ein beliebiges 5-Tupel der genannten Salze.

$$\begin{aligned}
 \vec{v} &= \left\{ \begin{array}{l} v_1 \frac{2NaCl}{1000molH_2O} \\ v_2 \frac{2KCl}{1000molH_2O} \\ v_3 \frac{MgCl_2}{1000molH_2O} \\ v_4 \frac{MgSO_4}{1000molH_2O} \\ v_5 \frac{Na_2SO_4}{1000molH_2O} \end{array} \right\} \\
 \vec{i} &= \left\{ \begin{array}{l} i_1 \frac{2Na^+}{1000molH_2O} \\ i_2 \frac{2K^+}{1000molH_2O} \\ i_3 \frac{Mg^{2+}}{1000molH_2O} \\ i_4 \frac{SO_4^{2-}}{1000molH_2O} \\ i_5 \frac{Cl^-}{1000molH_2O} \end{array} \right\} = f_i(\vec{v}) = \left\{ \begin{array}{l} v_1 + v_5 \frac{2Na^+}{1000molH_2O} \\ v_2 \frac{2K^+}{1000molH_2O} \\ v_3 + v_4 \frac{Mg^{2+}}{1000molH_2O} \\ v_4 + v_5 \frac{SO_4^{2-}}{1000molH_2O} \\ 2 \cdot (v_1 + v_2 + v_3) \frac{Cl^-}{1000molH_2O} \end{array} \right\} \\
 \vec{j} &= \left\{ \begin{array}{l} j_1 \text{ 2K Jänecke\%} \\ j_2 \text{ Mg Jänecke\%} \\ j_3 \text{ SO}_4 \text{ Jänecke\%} \\ j_4 \text{ 2Na Jänecke\%} \\ j_5 \text{ H}_2\text{O Jänecke\%} \end{array} \right\} = f_j(\vec{i}) = \frac{100}{i_2 + i_3 + i_4} \cdot \left\{ \begin{array}{l} i_2 \text{ 2K Jänecke\%} \\ i_3 \text{ Mg Jänecke\%} \\ i_4 \text{ SO}_4 \text{ Jänecke\%} \\ i_1 \text{ 2Na Jänecke\%} \\ 1000 \text{ H}_2\text{O Jänecke\%} \end{array} \right\} \\
 \vec{b} &= \left\{ \begin{array}{l} \lambda \\ \mu \\ \nu \end{array} \right\} = f_b(\vec{j}) = \frac{1}{100} \cdot \left\{ \begin{array}{l} j_3 \\ j_1 \\ j_2 \end{array} \right\} \\
 \vec{k} &= \left\{ \begin{array}{l} x \\ y \end{array} \right\} = f_k(\vec{b}) = \left\{ \begin{array}{l} (2 \cdot \mu) + \nu \\ \nu \cdot \sqrt{3} \end{array} \right\}
 \end{aligned}$$

4.1.1 Baryzentrische Koordinaten

Jeder Punkt p im Innern eines beliebigen Dreiecks mit den Ecken U, V, W , welches dann auch *Grunddreieck* des baryzentrischen Koordinatensystems genannt wird, kann durch baryzentrische Koordinaten (λ, μ, ν) beschrieben werden. Dazu auch [HL92]

Dabei muss das Zahlentripel die Bedingung $\lambda, \mu, \nu \in \mathbb{R} \geq 0 \wedge \lambda + \mu + \nu = 1$ erfüllen.

Klar ist: Liegt ein Punkt außerhalb des Grunddreiecks, so ist mindestens eine Komponente negativ.

Der **Vorteil** dieser Koordinatenart ist, dass sie invariant unter affinen Transformationen ist. Bezeichnet $p \rightarrow \hat{p} = l \cdot p + t$ eine affine Transformation mit linearem Anteil l und Translationsanteil t , dann besitzt \hat{p} bzgl. des transformierten Dreiecks die gleichen baryzentrischen Koordinaten wie p bzgl. des ursprünglichen.

Aus diesem Grund kann ein einzelnes Dreieck auch als gleichseitiges Dreieck, wie oben, dargestellt werden. Dies ist besonders sinnvoll, wenn Symmetrien in λ, μ, ν auch optisch aufgezeigt werden sollen.

5 Quinäres System als 3D-Objekt

Quinäres System als 3D-Objekt bedeutet, dass über der X-Y-Ebene, die das Jänecke-Dreieck enthält, Punkte anhand ihrer Na -, H_2O - oder ϱ -Werte aufgetragen werden. An der Z-Achse wird somit entweder das Jänecke-Natrium aufgetragen oder das Jänecke-Wasser, falls nicht die Dichte als deren Alternative aufgetragen wurde.

5.1 Dreieckszerlegung

Da in Abschnitt 5.2 Dreieckszerlegungen benutzt werden, ist es sinnvoll zunächst etwas Allgemeines zu diesen Zerlegungen zu sagen.

5.1.1 Forderung an eine allgemeine Triangulierung

Die Dreieckszerlegung (Triangulierung) wird zum Beispiel bei FEM (Finite Element Methode) eingesetzt, und als solche muss sie bestimmten Anforderungen genügen. Eine Zerlegung $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$ von Ω in Dreiecke heißt *zulässig*, wenn folgende Eigenschaften erfüllt sind:

1. $\bar{\Omega} = \bigcup_{i=1}^m T_i$ Das Gebiet lässt sich komplett in Dreiecke unterteilen. Die Unterteilung muss nicht regelmäßig sein.
2. Besteht $T_i \cap T_j$ aus genau einem Punkt, so ist dieser ein Eckpunkt sowohl von T_i als auch von T_j .
3. Besteht $T_i \cap T_j$, $i \neq j$, aus mehr als einem Punkt, so ist $T_i \cap T_j$ eine Kante von T_i und auch von T_j .

5.1.2 Regelmäßige Dreieckszerlegung

Es sei $T = \Delta(ABC)$ das Dreieck mit den Eckpunkten A , B , C und die Seiten des Dreiecks seien $c = \overline{AB}$, $a = \overline{BC}$ und $b = \overline{CA}$

Eine regelmäßige Dreieckszerlegung von T kann durch Parallelverschiebung der Seiten a , b und c erreicht werden.

Zunächst unterteilt man die Seiten in gleichviele äquidistante Teilabschnitte, z.B. in $n \in \mathbb{N}$ Teilabschnitte.

Nummeriere die entstehenden $n - 1$ Punkte wie folgt:

Die Seite c wird von A nach B , die Seite a wird von C nach B und die Seite b wird von A nach C nummeriert. Dabei bekommen die Punkte, die bei der äquidistanten Unterteilung entstanden sind, die Namen a_1, a_2, \dots, a_{n-1} ; b_1, b_2, \dots, b_{n-1} ; c_1, c_2, \dots, c_{n-1} .

Für alle $i \in \{1, \dots, n - 1\}$ verbindet man nun nacheinander die Punkte c_i mit den a_i . Es entstehen somit zur Seite b parallelverschobene neue Seiten.

Genauso werden die Punkte b_i mit den c_i verbunden. Dieses führt zu neuen Seiten, die zur Seite a parallelverschoben sind.

Übrig bleiben nun noch die Parallelen zur Seite c , diese erhält man, indem man die Punkte b_i mit a_i verbindet.

Die neuen Seiten kreuzen sich im Inneren des Dreiecks. Diese Kreuzungspunkte bilden zusammen mit den a_i, b_i, c_i und den Punkten A, B, C die Punktmenge des nun regelmäßigen Dreiecksnetzes, welches aus dem Dreieck T entstanden ist.

5.1.3 Delaunay-Triangulation

Der duale Graph zur Delaunay Triangulierung ist die Dirichlet-Parkettierung (tessellation), die oft auch Thiessen- oder Voronoi-Parkettierung genannt wird.

Es sei $d(x, P_k)$ der euklidische Abstand, dann ist die Dirichlet-Parkettierung definierbar wie in [HL92] durch:

$$F_i = \{x \in \mathbb{R}^2 : d(x, P_i) \leq d(x, P_j) \forall j \neq i\} \quad (5.1)$$

Eine Dirichlet-Parkettierung einer gegebenen Punktmenge $\mathcal{P} = \bigcup_i P_i$ lässt sich durch Errichten der Mittelsenkrechten der Verbindungsgeraden der P_i erzeugen. Die Delaunay-Triangulierung der P_i ergibt sich sodann als duale Struktur, d.h. Punkte P_i und P_j werden miteinander verbunden, falls die Fliesen F_i und F_j der zugehörigen Dirichlet-Parkettierung eine gemeinsame Kante besitzen.

Die Delaunay-Triangulierung kann aber auch mit dem **lokalen Umkreiskriterium** erstellt werden. Dieses Kreiskriterium besagt, dass der Umkreis des Dreiecks T_{ijk} mit den Ecken P_i, P_j und P_k einer Vierpunkttriangulierung nicht die gegenüberliegende Ecke P_l des entlang der Seite $P_j P_k$ anschließenden Dreiecks T_{jkl} mit Ecken P_j, P_k und P_l beinhalten darf.

Anders formuliert [She]: Die Delaunay-Triangulierung einer Punktmenge ist eine Dreieckszerlegung mit der Eigenschaft, dass für alle Dreiecke kein Punkt der Punktmenge in das Innere eines Umkreises, das ist der Kreis, der alle drei Ecken des Dreiecks durchläuft, fällt.

Constrained Delaunay-Triangulation

Ein PSLG (Planar Straight Line Graph) ist eine Ansammlung Knoten und Segmente. Segmente sind Kanten, deren Endpunkte Knoten des PSLG sind. Punkte, die auf den Segmenten liegen, gehören somit nicht zum PSLG.

Eine constrained Delaunay-Triangulation eines PSLG ist ähnlich der Delaunay-Triangulation, aber jedes PSLG-Segment ist als eine Dreieckskante in der Dreieckszerlegung übernommen.

Eine constrained Delaunay-Triangulation ist keine echte Delaunay-Triangulation. Denn einige ihrer Dreiecke können gegen das Umkreiskriterium verstoßen.

Constrained conforming Delaunay-Triangulation

Eine constrained conforming Delaunay-Triangulation (CCDT) eines PSLG ist in erster Linie eine constrained Delaunay-Triangulation, jedoch enthält sie Steiner-Punkte. Steiner-Punkte sind Punkte, die PSLG-Segmente unterteilen und so für eine Erhöhung der Dreiecksanzahl sorgen. Dies bewirkt meist ein besseres Dreiecksnetz.

5.2 Problem: 2D \mapsto 3D

Die Frage, „Wie kann man aus einem 2D Objekt eindeutig ein 3D Objekt erzeugen?“, kann nur mit „Gar nicht“ beantwortet werden, wenn man keine Annahmen über die dritte Vektorkomponente macht.

Wie in Kapitel 4 beschrieben, lässt sich das quinäre System durch die Jäneckeinheiten mit baryzentrischen Koordinaten in die Ebene zeichnen.

Erinnerung: Die Funktion f_k bildet baryzentrische Koordinaten in kartesische ab. Diese Funktion ist umkehrbar. Die Umkehrfunktion bildet kartesische Koordinaten in baryzentrische ab, d.h., zu jedem Punkt in der x-y-Ebene lässt sich die Lage im Dreieck mittels baryzentrischen Koordinaten errechnen und angeben.

Diese Umkehrfunktion ist definiert durch:

$$\begin{aligned} \nu &= \frac{y}{\sqrt{3}} \\ \mu &= \frac{1}{2}(x - \nu) = \frac{1}{2} \left(x - \frac{y}{\sqrt{3}} \right) \\ \lambda &= 1 - (\mu + \nu) = 1 - \left(\frac{1}{2}x - \frac{1}{2}\frac{y}{\sqrt{3}} + \frac{y}{\sqrt{3}} \right) = 1 - \frac{1}{2} \left(x + \frac{y}{\sqrt{3}} \right) \\ \vec{f}_k(x, y) = \vec{b} &= \left\{ \begin{array}{c} \lambda \\ \mu \\ \nu \end{array} \right\} = \left\{ \begin{array}{c} 1 - \frac{1}{2} \left(x + \frac{y}{\sqrt{3}} \right) \\ \frac{1}{2} \left(x - \frac{y}{\sqrt{3}} \right) \\ \frac{y}{\sqrt{3}} \end{array} \right\} \end{aligned} \quad (5.2)$$

Werden nun die so erhaltenen Werte für (λ, μ, ν) mit 100 multipliziert, erhält man die Werte für Mg , SO_4 und $2K$, d.h., vom Vektor \vec{j} – siehe Kapitel 4 – sind die drei

Komponenten j_1, j_2, j_3 bekannt. Genauer:

$$j_1 \text{ 2K Jänecke\%} = 100 \cdot \mu \text{ 2K Jänecke\%} \quad (5.3)$$

$$j_2 \text{ Mg Jänecke\%} = 100 \cdot \nu \text{ Mg Jänecke\%} \quad (5.4)$$

$$j_3 \text{ SO}_4 \text{ Jänecke\%} = 100 \cdot \lambda \text{ SO}_4 \text{ Jänecke\%} \quad (5.5)$$

Die letzten zwei Komponenten j_4 (Jäneckewert für $2Na$) und j_5 (Jäneckewert für H_2O) lassen sich also nicht aus dem Dreieck ablesen.

Daher schreibt R. COHEN-ADAD [CBTR02], dass eine Lösung durch zwei Graphen im drei-dimensionalen Raum dargestellt werden muss, um eindeutig definiert zu sein. Dabei ist ein Graph für die Salzkombination und ein anderer für die Wassermenge zuständig.

1. bzw. 2. Fall: An der Z-Achse soll Wasser bzw. Natrium in Jäneckeeinheiten aufgetragen werden

Obwohl wir wissen, dass die Stoffe in 1000 mol H_2O gelöst wurden, ist es mit analytischer Mathematik wegen der Normierung, die die Funktion f_j durchführt, unmöglich die beiden fehlenden Komponenten j_4 (Jäneckewert für $2Na$) und j_5 (Jäneckewert für H_2O) eindeutig zu bestimmen.

Selbst wenn es keine Normierung gäbe, könnte man keine Aussage über den Natriumgehalt machen, wegen des in Kapitel 2 geschilderten Problems mit der chemischen Aktivität.

Zur Erinnerung: Die Menge, die von einem Stoff durch ein Lösungsmittel aufgelöst werden kann, hängt von der Aktivität ab.

Ohne Normierung würde die Fragestellung lauten: „Wieviel $NaCl$ muss in $1000 \text{ mol } H_2O$ aufgelöst werden, um $NaCl$ -gesättigt zu werden, wenn schon eine beliebige aber bekannte Anzahl der Ionen K^+ , Mg^{2+} , SO_4^{2-} und Cl^- sich in der Lösung befinden?“

Wegen der Normierung auf die Summe der Magnesium-, Sulfat- und Kaliumionen verschärft sich die Fragestellung zu: „Wieviel $NaCl$ muss in $x \text{ } H_2O$ Jänecke\% aufgelöst werden, wenn sich in der Lösung $100 \cdot \lambda \text{ Mg Jänecke\%}$, $100 \cdot \mu \text{ SO}_4 \text{ Jänecke\%}$ und $100 \cdot \nu \text{ 2K Jänecke\%}$ befinden, um die Lösung an $NaCl$ zu sättigen?“

3. Fall: An der Z-Achse soll die Dichte aufgetragen werden

Besitzt man genügend Information über die Lösung, d.h., sind die Vektoren \vec{j} oder \vec{i} oder \vec{v} vollständig, dann kann man die Dichte wie in Kapitel 3 bestimmen. Ausgehend von der 2D-Projektion ist diese Information, wie gerade schon aufgezeigt, nicht vollständig.

5.3 Mögliche Lösungen des Problems

Die angewandte Mathematik bietet mit numerischen Methoden der Interpolation eine Möglichkeit ausgehend von den Meßwerten über dem Jänecke-Dreieck 3D-Objekte zu erstellen, bzw. einen möglichen Werteverlauf zwischen den einzelnen Messwerten, die sogenannten Stützstellen, zu bestimmen.

Warnung: Ob der interpolierte Werteverlauf chemisch, mineralogisch sinnvoll ist, kann ich nicht sagen. Wie weit die Interpolationsergebnisse von der Realität abweichen, lässt sich erst durch Messreihen bewerten.¹

Im Programm `Quinview` habe ich drei Interpolationsmethoden implementiert, von denen zwei einen nichtlinearen Ansatz besitzen.

5.3.1 Linearer Ansatz

Eine sehr einfache Interpolation im 2D-Raum ist die **lineare Interpolation**.

Gegeben sind (x, y) -Paare als Stützstellen und zwischen je zwei Stützstellen liegen alle zu interpolierenden y -Werte auf einer Geraden zwischen (x_i, y_i) und (x_{i+1}, y_{i+1}) .

Im 3D-Raum sind die Stützstellen gegeben als (x, y, z) -Tripel. Gesucht sind z -Werte zwischen den Stützstellen, wobei zwischen 2 Stützstellen alle zu interpolierenden Werte auf einer Ebene liegen.

Man weiß, daß eine Ebene durch 3 Punkte definiert werden kann.

Problem des Ansatzes: Es stellt sich das Entscheidungsproblem, welche drei Punkte (=Stützstellen), also Meßwerte, eine Ebene definieren sollen.

Lösung: Die Delaunay-Triangulation² (siehe Abschnitt 5.1.3) löst das Entscheidungsproblem. Für jede Facette, das sind die Stabilitätsfelder im Jänecke-Dreieck, wird eine *DT* durchgeführt. Die Eckpunkte der Delaunay-Dreiecke werden als Definitionspunkte für die einzelnen Ebenen genommen.

Um ein feinmaschiges Netz im 3D-Raum zu erhalten, werden nun die einzelnen Delaunay-Dreiecke nach Abschnitt 5.1.2 regelmäßig unterteilt.

Eventueller Nachteil: Es ist klar, dass es einen „Knick“ in der Facette gibt, wenn jeweils zwei Dreiecksflächen gegeneinander stoßen und sich nicht auf einer Ebene befinden. Stoßen Facetten aneinander, sind die Übergänge ebenfalls „verknickt“, d.h. nicht stetig differenzierbar.

Es gibt also im Allgemeinen keine glatten Facetten und damit auch keine glatte Gesamtoberfläche.

5.3.2 Nichtlinearer Ansatz

Hier werden zwei Interpolationsmethoden diskutiert, die versuchen eine möglichst glatte Oberfläche zu erzeugen, d.h., die zu interpolierenden Werte sollen möglichst auf einer C^1 -stetigen gekrümmten Fläche, das ist also keine Ebene mehr, liegen.

¹Diese Messreihen durchzuführen, lag nicht im Aufgabenbereich meiner Studienarbeit. Ich würde aber gerne erfahren, wie weit die Interpolationsergebnisse sich mit der Realität überdecken.

²Die Delaunay-Triangulation in `Quinview` wird durch Aufrufen von J. R. SHEWCHUKS TRIANGLE (<http://www.cs.cmu.edu/~quake/triangle.html>) erledigt.

Problem des Ansatzes: Damit eine möglichst glatte Oberfläche, eine schöne Gitternetzstruktur, erzeugt werden kann, benötigt man viel mehr Stützstellen als hier alleine durch Messwerte zur Verfügung stehen.

Lösung: Die geochemische, mineralogische Modellvorstellung besagt, dass sich im 3D-Raum die z -Werte auf den Kanten (oder Segmenten) der Facetten, d.h. Kristallisationsbahnen und Abgrenzungslinien, linear verändern.

Durch dieses Modell gelingt es die Kanten zu unterteilen (z.B. äquidistant). Diese Unterteilungspunkte werden als Stützstellen benutzt. Diese die Segmente unterteilenden Punkte können als Steiner-Punkte (Abschnitt 5.1.3) angesehen werden.

Definition: Die Menge aller Unterteilungspunkte heiße UP .

Wenn nun eine CDT³ (siehe Abschnitt 5.1.3) durchgeführt wird, um die (x, y) -Paare zu definieren, deren z -Wert es zu interpolieren gilt, hat man händisch eine CCDT (siehe Abschnitt 5.1.3) durchgeführt, mit dem großen Vorteil, dass die Steiner-Punkte von zwei verschiedenen Facetten identisch sind, sobald das Segment, die Kante, zu beiden Facetten gehört.

Falls man facettenweise direkt eine CCDT durchführen würde, wäre es nicht gesichert, dass die einzelnen Netze auch über gemeinsame Steiner-Punkte verbunden werden.

Durch das „manuelle“ Setzen der Steiner-Punkte ist also hier eine gute Verbindung der einzelnen Facetten-Triangulationen untereinander gewährleistet.

Achtung: Je dichter das Dreiecksnetz unter Beachtung von bestimmten Restriktionen (siehe Abschnitt 5.3.3) ist, desto besser kann die Oberfläche rekonstruiert werden.

Shepard Interpolation

Eine bekannte Interpolationsmethode, häufig nach [HL92] in Geologie und Metrologie benutzt, ist das Verfahren von SHEPARD.

Jede Stützstelle hat direkten Einfluß auf jeden zu interpolierenden Wert. Das bedeutet prinzipiell, dass dieses Verfahren über die Facettengrenzen hinweg schaut, und somit in die Kategorie der **globalen Verfahren** fällt.

Eine **lokale Variante** lässt sich dadurch erreichen, dass jede der Facetten – das waren die Stabilitätsfelder – einzeln abgearbeitet werden, so dass nur Punkte auf den Facettenrändern Einfluss auf die Punkte im Innern der entsprechenden Facette haben.

Es sei $S = S_0(x, y)$ der z -Wert, der zu einem Punkt $p = (x, y)$ berechnet werden soll.

Der euklidische Abstand zwischen zwei Punkten $p = (x, y)$ und $p_i = (x_i, y_i)$ ist definiert als:

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (5.6)$$

³Die constrained Delaunay-Triangulation übernimmt J. R. SHEWCHUKS TRIANGLE (<http://www.cs.cmu.edu/~quake/triangle.html>).

Sei w_i eine inverse Potenz des euklidischen Abstands:

$$w_i = \frac{1}{r_i^q} \quad (5.7)$$

und für q gelte die Bedingung

$$q \in \mathbb{N} \mid 0 < q \quad (5.8)$$

Außerdem sei z_i der z -Wert der i -ten Stützstelle.

Dann ergibt sich das Shepard-Verfahren mit w_i als Gewichtung bei n Stützstellen:

$$S = S_0(x, y) = \frac{\sum_{i=1}^n w_i z_i}{\sum_{i=1}^n w_i} = \frac{\sum_{i=1}^n \frac{1}{r_i^q} z_i}{\sum_{i=1}^n \frac{1}{r_i^q}} \quad (5.9)$$

Zähler und Nenner lassen sich im Wesentlichen in $O(n)$ Schritten berechnen. Formal korrekt lässt sich (5.9) umformen zu:

$$S = S_0(x, y) = \sum_{i=1}^n \left(z_i \cdot \frac{\frac{1}{r_i^q}}{\sum_{i=1}^n \frac{1}{r_i^q}} \right) \quad (5.10)$$

Allerdings ist hier die Zeitkomplexität $O(n^2)$.

In [Spä91] wurde mit der Begründung der numerischen Stabilität noch eine weitere Potenzhöhung in der Zeitkomplexität durchgeführt, indem dieses Verfahren in Fortran implementiert wurde als:

$$S = S_0(x, y) = \sum_{i=1}^n \left(z_i \cdot \frac{\prod_{j \neq i} r_j^q}{\sum_{k=1}^n \prod_{j \neq k} r_j^q} \right) \quad (5.11)$$

Offensichtlich ist es absolute Zeitverschwendung für den späteren Anwender, einen $O(n)$ Algorithmus als $O(n^3)$ zu implementieren.

Die numerische Stabilität, d.h. die Verhinderung von Unter- und Überläufen, sowie die Minimierung von Rundungsfehlern, lässt sich auch ganz anders erreichen, ohne dass dabei die Zeitkomplexität wesentlich erhöht werden würde.

Meine Modifikation: In $O(n)$ lässt sich die q -te Potenz des maximalen Abstandes herausfinden:

$$r_{max}^q = \max\{r_i^q \mid \forall i \in \{1, \dots, n\}\} \quad (5.12)$$

Sei A eine Konstante die definiert ist durch:

$$A = \prod_{i=1}^n \left(\frac{r_i^q}{r_{max}^q} \right) = \left(\frac{1}{r_{max}^q} \right)^n \cdot \prod_{i=1}^n (r_i^q) \quad (5.13)$$

Die Berechnung von A besitzt die Zeitkomplexität von $O(n)$. A ist eine Normierung auf die q -te Potenz des maximalen Abstands und liegt immer zwischen Null und Eins, da jedes Produktglied $\frac{r_i^q}{r_{max}^q}$ zwischen Null und Eins liegt.

Mit dieser Konstanten A lässt sich das Shepard-Verfahren formulieren als:

$$\begin{aligned} \tilde{S}_0(x, y) &= 1 \cdot S_0(x, y) = \frac{A}{A} \cdot S_0(x, y) \\ &= \frac{A \cdot \sum_{i=1}^n (z_i \frac{1}{r_i^q})}{A \cdot \sum_{i=1}^n (\frac{1}{r_i^q})} = \frac{\sum_{i=1}^n (z_i \cdot A \cdot \frac{1}{r_i^q})}{\sum_{i=1}^n (A \cdot \frac{1}{r_i^q})} \end{aligned} \quad (5.14)$$

Bei sehr kleinen r_i^q verhindert A durch die Normierung einen Überlauf, der durch das sehr große $\frac{1}{r_i^q}$ entstünde. Ist A so nahe an Null, dass der Exponent für den Datentyp `double` zu klein ist, kann A getrost gekürzt sein, denn die Gefahr eines Überlaufes von $\frac{1}{r_i^q}$ ist nicht gegeben.

Beweis: Um zu zeigen, dass mein $\tilde{S}_0(x, y)$ dasselbe leistet wie die $O(n^3)$ -Form, werde ich die $O(n^3)$ -Form in mein $\tilde{S}_0(x, y)$ umformen.

$$\begin{aligned} S &= S_0(x, y) = \sum_{i=1}^n \left(z_i \cdot \frac{\prod_{j \neq i} r_j^q}{\sum_{k=1}^n \prod_{j \neq k} r_j^q} \right) = \frac{\sum_{i=1}^n \left(z_i \cdot \prod_{j \neq i} r_j^q \right)}{\sum_{k=1}^n \left(\prod_{j \neq k} r_j^q \right)} \\ &= \frac{\sum_{i=1}^n \left(z_i \cdot \prod_{j \neq i} r_j^q \right)}{\sum_{i=1}^n \left(z_i \cdot \frac{1}{r_i^q} \cdot \prod_{j=1}^n (r_j^q) \right)} = \frac{\prod_{j=1}^n (r_j^q) \cdot \sum_{i=1}^n \left(\frac{z_i}{r_i^q} \right)}{\prod_{j=1}^n (r_j^q) \cdot \sum_{i=1}^n \left(\frac{1}{r_i^q} \right)} \\ &= \frac{\left(\frac{1}{r_{max}^q} \right)^n \prod_{j=1}^n (r_j^q) \cdot \sum_{i=1}^n \left(\frac{z_i}{r_i^q} \right)}{\left(\frac{1}{r_{max}^q} \right)^n \prod_{j=1}^n (r_j^q) \cdot \sum_{i=1}^n \left(\frac{1}{r_i^q} \right)} = \frac{A \cdot \sum_{i=1}^n \left(\frac{z_i}{r_i^q} \right)}{A \cdot \sum_{i=1}^n \left(\frac{1}{r_i^q} \right)} \\ &= \frac{\sum_{i=1}^n \left(z_i \cdot A \cdot \frac{1}{r_i^q} \right)}{\sum_{i=1}^n \left(A \cdot \frac{1}{r_i^q} \right)} \\ &= \tilde{S}_0(x, y) \end{aligned}$$

Genauere Betrachtung des Wertes q :

Für $q = 1$ existieren rechts- und linksseitige partielle Ableitungen der v_i in P_i , deren Werte im Allgemeinen verschieden sind;

für $q < 1$ existieren keine partiellen Ableitungen.

Für die häufig empfohlene Wahl $q = 2$ gilt:

$$q = 2 \Rightarrow \left. \frac{\partial S_0}{\partial x} \right|_{P_i} = \left. \frac{\partial S_0}{\partial y} \right|_{P_i} = 0 \quad (5.15)$$

Diese Eigenschaft kann die Umgebung der Stützstellenpunkte P_i flachgedrückt erscheinen lassen. Die Ursache dafür ist letztlich, dass S_0 – und auch \tilde{S}_0 – nur die Größe der Entfernung von P und P_i nicht aber die Richtung der entsprechenden Strecke beachtet.

Dieser Effekt lässt sich beheben, wenn man geschätzte Werte für die ersten partiellen Ableitungen verfügbar hat oder macht. Die Funktion $S_1(x, y)$ hat dieselben Eigenschaften wie S_0 , nimmt jedoch die vorgegebenen Werte für die partiellen Ableitungen an den Punkten P_i an und lässt sich wie \tilde{S}_0 in $O(n)$ berechnen.

$S_1(x, y)$ werde definiert durch:

$$z_{x_i} = \left. \frac{\partial S_0}{\partial x} \right|_{P_i} \quad \forall i \in \{1 \dots n\} \quad (5.16)$$

$$z_{y_i} = \left. \frac{\partial S_0}{\partial y} \right|_{P_i} \quad \forall i \in \{1 \dots n\} \quad (5.17)$$

$$S = S_1(x, y) = \frac{\sum_{i=1}^n \left([z_i + (x - x_i)z_{x_i} + (y - y_i)z_{y_i}] \cdot A \cdot \frac{1}{r_i^q} \right)}{\sum_{i=1}^n \left(A \cdot \frac{1}{r_i^q} \right)} \quad (5.18)$$

Mean Value Interpolation

Die Mean Value Interpolation [Hor05] hat deutliche Vorteile gegenüber der Shepard-Interpolation. Zum Einen ist sie deutlich schneller als Shepard, zum anderen wesentlich glatter. Sie ist glatter, da sie überall in der Fläche C^∞ stetig ist ausgenommen die Stützstellen. Sie liefert an den Kanten zwischen zwei Stützstellen lineare Verläufe.

Die Implementierung in `Quinview` folgt dem Pseudocode aus [Hor05]. Allerdings unterscheidet sich meine Implementierung in `double geometObj::MeanValue_interpolation(const double &x, const double &y, const vector<Vector3D> &inputpoints)` vom Pseudocode dahingehend, dass die Methode direkt verlassen wird, sollte eine Stützstelle interpoliert werden. Hier ist natürlich der Wert der Funktion direkt durch die Stützstelle gegeben.

5.3.3 Feinmaschigkeit vs. Stützpunkte

Um unschöne Artefaktbildung am Rand der Objekte, oder hier im Speziellen am Rand der Facetten, zu vermeiden, sollte die Feinmaschigkeit des Triangulierungsnetzes über den Flächeninhalt der zu generierenden Dreiecke optimiert werden. Das Optimierungskriterium besagt, dass alle neuen Dreiecke einer CD oder CCD annähernd den gleichen Flächeninhalt aufweisen und annähernd gleichseitig sein sollen. Mit neuen Dreiecken sind solche gemeint, deren Kanten nicht Element des PSLG sind.

Idealfall: Alle Dreiecke sind gleichseitig und haben gleichen Flächeninhalt.

Es seien $v_1, v_2 \in UP$ zwei benachbarte segmentunterteilende Punkte; diese sind hier manuelle Steinerpunkte, welche Facettenkanten unterteilen. Und es sei $e = r(v_1, v_2)$ der euklidische Abstand zwischen v_1 und v_2 .

Die Höhe des gleichseitigen Dreiecks mit den zwei fixen Eckpunkten v_1, v_2 berechnet sich nach PYTHAGORAS:

$$h^2 + \left(\frac{e}{2}\right)^2 = e^2 \quad (5.19)$$

$$h^2 = e^2 - \left(\frac{e}{2}\right)^2 = \frac{4e^2}{4} - \frac{e^2}{4} = \frac{3e^2}{4} \quad (5.20)$$

$$h = \frac{\sqrt{3}}{2} \cdot e \quad (5.21)$$

Der geforderte Flächeninhalt f berechnet sich dann aus dem Abstand e und der Höhe h wie folgt:

$$f = \left(\frac{1}{2} \cdot e\right) \cdot h = \left(\frac{1}{2} \cdot e\right) \cdot \left(\frac{\sqrt{3}}{2} \cdot e\right) = e^2 \cdot \frac{\sqrt{3}}{4} \quad (5.22)$$

Anmerkung: Bei der Berechnung des geforderten Flächeninhaltes der Dreiecke eines Dreiecksnetzes ist e der **durchschnittliche euklidische Abstand** zwischen den Punkten.

5.4 Isolinien aus 3D Oberflächengitternetz

Eine Isolinie ist definiert durch zusammenhängende Geraden, Kurven und Punkten mit gleichem Wert. Die Elemente einer Isolinie besitzen beispielsweise denselben z-Wert, z.B. Höhenlinien auf einer Wanderkarte sind solche Isolinien.

Eine Isolinie aus einem 3D-Objekt zu extrahieren ist ganz einfach: Man schneidet das Objekt mit einer Ebene. In *Quinview* werden Ebenen benutzt, die konstanten z-Wert besitzt.

Gibt es Schnitte, so gibt es mindestens eine Isolinie für diesen Wert. Analog gibt es genau dann keine Isolinie für den Wert, wenn es keine Schnitte gibt. Die Oberfläche des

3D-Objektes kann durch ein Dreiecksnetz (triangulation net) im 3D-Raum approximiert werden.

Es sei TN das Dreiecksnetz, das die Oberfläche eines 3D-Objekts modelliert. Schneidet man nun TN mit einer Ebene, so kann jedes Dreieck T des Netzes einzeln betrachtet werden.

5.4.1 Klassifikation von Dreiecken bezüglich einer z-iso-Fläche F

Da es offensichtlich sehr unterschiedliche Möglichkeiten geben kann, wie ein Dreieck von einer Ebene, hier im Speziellen eine z-iso-Fläche F , geschnitten wird, kann man die Dreiecke dahingehend klassifizieren.

- a+b) Alle Punkte liegen entweder über oder unter F
 - c) Zwei Punkte unter und ein Punkt über F
 - d) Ein Punkt unter und zwei Punkte über F
- e+f) Ein Punkt unter und ein Punkt über sowie ein Punkt in der F
 - g) Ein Punkt liegt in und zwei Punkte liegen über F
 - h) Ein Punkt liegt in und zwei Punkte liegen unter F
 - i) Zwei Punkte liegen in und einer über F
 - j) Zwei Punkte liegen in und einer unter F
 - k) Alle drei Punkte liegen in F

Der Isoliniengenerator, der in **Quinview** benutzt wird, arbeitet nach folgendem Prinzip:

1. Teste, ob Dreieck überhaupt betrachtet werden muss (d.h. Fall a+b liegt nicht vor)!
2. Falls ja, sortiere aufsteigend die Dreieckspunkte nach ihrer z-Komponente!
3. Klassifiziere das Dreieck in die verbliebenden Möglichkeiten und generiere den Klassen entsprechend ein Stück der Isolinie!

Genauere Betrachtung des dritten Schrittes:

- Gibt es zwei Schnittpunkte, so wird zwischen diesen eine Linie gezogen.

-
- Gibt es nur einen Schnittpunkt, so wird er ignoriert, da dieser dann von anderen Dreiecken berücksichtigt wird. Falls der Punkt ein lokales Extremum ist, so kann er ebenfalls nicht mit einem anderen Punkt verbunden werden, und wird darum ignoriert.
 - Liegt die Klasse `k` vor, so ist es ein offenes Problem für die automatische Konstruktion, da man ohne topologische Eigenschaften (Nachbarschaftsbeziehungen zwischen den einzelnen Dreiecken) nicht entscheiden kann wie die Isolinie laufen muss.

Allerdings scheint die Klasse `k` in den von `Quinview` ausgewerteten Datensätzen nicht aufzutreten. Dies erkennt man daran, dass Isolinien nicht unterbrochen erscheinen, auch wenn die momentane Implementierung Dreiecke der Klasse `k` ignoriert.

(Implementierung siehe: `void Gui::TriangleSetIsoline(Vector3D v1, Vector3D v2, Vector3D v3, double zIsoValue, int zVektorPosition)` in `generateIsolinesSlot.cpp`)

6 Dateiparser

Es werden im Programm „Quinview“ drei verschiedene Dateiparser benutzt, die im Folgenden kurz beschrieben werden sollen.

6.1 Quinviewkonfigurationsdatei

Im Hauptprogramm wird, bevor die graphische Oberfläche aufgebaut ist, die Konfigurationsdatei `quinview.ini` – falls diese existiert – ausgelesen. Ihr formaler Aufbau ist sehr simpel:

```
<Ort der Monninparameterdatei>  
<Ort der Datensatzdatei>
```

Da der Aufbau so einfach ist, habe ich keine formale Grammatik für den Parser gebaut, sondern direkt das einfache `ifstream` von C++ benutzt.

6.2 Monninparameterdatei

Da sich die Pitzer-Koeffizienten ändern können war es notwendig, dass diese aus einer Datei `monnin.par` eingelesen werden müssen. Ihr formaler Aufbau ist einfach.

Die erste Zeile:

```
<Anzahl Kationen> <Anzahl Anionen>
```

Die Summe der Anzahlen der Anionen und Kationen ergibt die Anzahl der nächsten Zeilen mit der folgenden Form:

```
<Name des Ions> <Ladung> <Molar-Gewicht>
```

Die Multiplikation der Anzahlen der Anionen und Kationen ergibt die Anzahl der nächsten Zeilen mit der folgenden Form:

```
<V0> <β0> <β1> <β2> <cv>
```

Wobei V_0 das Standardvolumen der Salze ist. Die dann folgenden Größen sind die zugehörigen Pitzer-Koeffizienten.

6.3 Datensatzdatei

Da in der Datensatzdatei deutlich mehr Informationen gespeichert sind als in den beiden anderen Dateien, habe ich zunächst eine formale Grammatik $G = (N, T, S, P)$ festgelegt, um dann darauf bauend den Parser zu schreiben. N ist die Menge der Nichtterminalsymbole, T ist die Menge der Terminalsymbole, $S \in N$ ist das Startsymbol und P ist die Menge der grammatikalischen Regeln, die so genannte Menge der Produktionen.

Jede gültige Datensatzdatei hat die Dateiendung `.qrh` und ist ein Wort $w \in L(G)$, wobei $L(G)$ die durch die Grammatik G definierte Sprache ist.

Die Grammatik G , und damit der formale Aufbau der Datei, lässt sich in der **Backus-Naur-Form** darstellen. Die **BNF** ist benannt nach J. BACKUS und P. NAUR, die diese Form zur Definition der Programmiersprache ALGOL 60 verwendeten. Die Nichtterminalsymbole der Grammatik werden in der BNF durch spitze Klammern `< ... >` gekennzeichnet. Die linke und rechte Seite einer BNF-Regel werden durch das Symbol `::=` getrennt. Gibt es mehrere rechte Seiten zu einem Nichtterminalsymbol, so werden diese durch senkrechte Striche getrennt hintereinander geschrieben, ohne dass die linke Seite neu geschrieben wird.

6.3.1 BNF der Datensatzdatei

```

<Datei> ::= <Datensatz>+<EoF>
<Datensatz> ::= <BeginToken><Daten><EndToken>
<BeginToken> ::= Begin_Data;<Zeilenende>*
<EndToken> ::= End_Data;<Zeilenende>*
<Zeilenende> ::= <EoL>|<Kommentar><EoL>
<EoL> ::= <CR><LF>|<LF>
<Daten> ::= <Name><Temp><Knoten><Kanten>|<Temp><Name><Knoten><Kanten>
<Temp> ::= #Temp[°]:<Float>;<Zeilenende>*
<Name> ::= #Name:<String>;<Zeilenende>*
<Knoten> ::= #Knotendef: { {<Knotendata><Zeilenende>*}^2 }*
<Knotendata> ::= (<Knotenname>,<2NaCl>,<2KCl>,<MgCl2>,<MgSO4>,<Na2SO4>)
<Knotenname> ::= <String>
<2NaCl> ::= <Float>   <2KCl>   ::= <Float>   <MgCl2> ::= <Float>
<MgSO4> ::= <Float>   <Na2SO4> ::= <Float>
<Kanten> ::= #Kantendef: { (<Knoten1>,<Knoten2>,<Linienstärke>)<Zeilenende>* }+
<Knoten1> ::= <String>
<Knoten2> ::= <String>
<Linienstärke> ::= <Int>
<Kommentar> ::= {<String>|<Float>|<Int>}*
<String> ::= "String"
<Float> ::= "Float"   <Int> ::= "Int"
<CR> ::= "CR"   <LF> ::= "LF"   <EoF> ::= "EoF"

```

7 Abschließende Gesichtspunkte

7.1 Mineralogische Erkenntnisse

Die mineralogische Modellvorstellung, dass sich die Zahlenwerte der Dichte an den Facettenkanten linear verändert, wenn gleichzeitig die Änderung der Jänecke-Werte von H_2O und Na als linear angenommen wird, wurde von **Quinview** widerlegt.

Die Inkonsistenz dieser Modellvorstellung zeigt das folgende Bild.

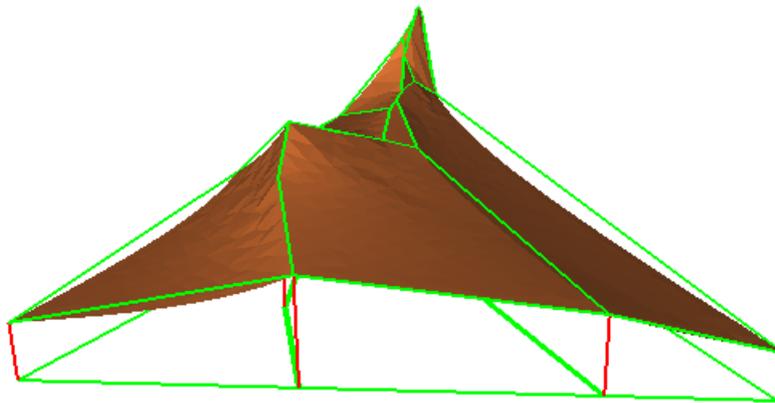


Abbildung 7.1: Meanvalue-Dichteverlauf nach D'Anz bei 15° C

Bilderklärung: Die Meanvalue-Interpolation wurde genutzt, um zwischen den Stützstellen die Jänecke-Werte für Wasser und auch Natrium zu interpolieren. Mit diesen Werten wurde dann der Dichteverlauf nach D'Anz berechnet, das Ergebnis ist die braune Fläche. Die grünen Linien in der dritten Dimension ($z \neq 0$) zeigen den gedachten Verlauf der Dichte zwischen zwei Messpunkten. Vor allem am linken und rechten Rand des Objektes wird der Unterschied zwischen Modell und Berechnung deutlich.

Folgerung: Entweder ändert sich an den Facettenkanten die Dichte linear oder aber die Natrium und Wasserkomponente.

Vermutung: Ich allerdings nehme an, dass sich keine der Größen linear verändert. Doch hierfür müssten deutlich mehr Messungen gemacht werden.

7.2 Mathematische Erkenntnisse

Unterschiedliche Interpolationsverfahren liefern im Endeffekt auch unterschiedliche Isolinien.

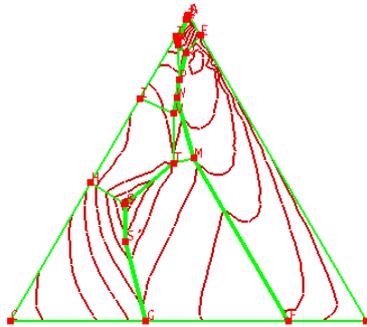


Abbildung 7.2: Isolinien der Durchschnittsdichte bei 15° C mit globalem Shepard-Verfahren

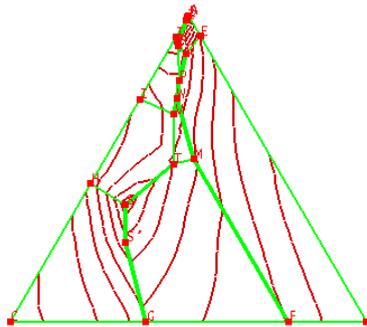


Abbildung 7.3: Isolinien der Durchschnittsdichte bei 15° C mit lokalem Shepard-Verfahren

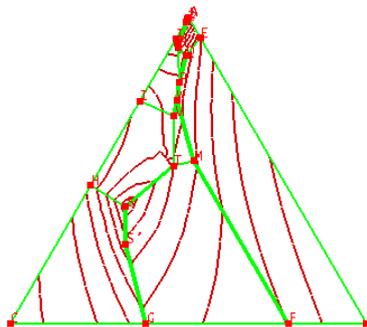


Abbildung 7.4: Isolinien der Durchschnittsdichte bei 15° C mit Meanvalue-Interpolation

7.3 Programmiertechnische Erkenntnisse

Der Microsoft Visual 6.0 C++ Compiler hält sich nicht immer an die Norm, daher kommt es schon beim Versionswechsels des Compilers zu sehr vielen Warnungen und auch Compiler-Fehlern. So ist in VC6 die Variable i in `for (int i=0; i<10 ; i++) { ... }` auch noch außerhalb der Schleife gültig. So ist nicht verwunderlich, wenn ein anderer Compiler bei einer späteren For-Schleife den Fehler gibt, dass die Zählvariable nicht definiert ist.

7.4 Ausblick

Als ich mit dem Programmieren von **Quinview** angefangen hatte, war Qt-Version 4 noch nicht freigegeben, und die Version 3 gab es nicht als freie kostenlose Version für Windows. Diese dritte Version war nur für Linux frei; da aber das Programm unter Windows laufen sollte, habe ich die Version 2.30 non-commercial benutzt.

Mit Qt2.x arbeitet von Coin3D die Bibliotheksversion coin-v1.0.2 mit der Schnittstelle SoQt-0.10.0 zusammen.

Wem es Freude bereitet, der kann gerne **Quinview** auf die neueren Versionen portieren:

1. QT 4
2. Coin 2.4.4
3. SoQt 1.3.0

Hinweis: Die neueren Bibliotheksversionen sind nicht direkt abwärtskompatibel, daher wird es nicht reichen einfach die Bibliotheksheader auszutauschen und neu zu kompilieren.

Bei einem Redesign der Klassenstrukturen könnten für die verschiedenen Dichteberechnungen Threads – QT4 kann hierfür eine betriebssystemunabhängige Grundlage liefern – benutzt werden; dies sollte die Dichteberechnung prinzipiell etwas beschleunigen. Hier wäre auch Potential für komplett verteiltes Rechnen z.B. im Cluster; ein möglicher Compiler wäre dann **mpiCC**.

Da die Inhalte des 2D- und 3D-Viewers separat schon als Bilder abgespeichert werden können, sollte es möglich sein, dass diese auch auf ein Druckermedium geschrieben werden können. Ob dieses aber betriebssystemunabhängig implementiert werden kann, weiß ich nicht.

Das Problem bei der Isoliniengenerierung mit den Dreiecken der Klasse **K**, deren Punkte in einer iso-z-Ebene liegen, könnte vielleicht annähernd gelöst werden, wenn das Dreiecksnetz auch topologische Informationen bereit stellt, d.h. Information darüber, welches

Dreieck mit welchem benachbart ist. Um eine Isolinie mit Lücke durch solche Dreiecke der Klasse K zu schließen, könnte man einen Punkt des Dreiecks um ε anheben und einen anderen absenken. Welche Punkte und welche Richtung dann richtig sind, dass müsste durch die Topologie des Netzes bestimmt werden.

7.5 Schlusswort

Prof. KURT MENGEL vom Institut für Mineralogie und Mineralische Rohstoffe der TU Clausthal fand das Programm `Quinview` sehr gelungen, und möchte es auch in der Lehre einsetzen.

Hinweis für den Anwender:

`Quinview` kann wegen der verschiedenen Modellvorstellungen, die eingesetzt werden, nur als Hilfsinstrument, als Schätzer oder Rater, benutzt werden. `Quinview` als Wahrheitsfinder zu interpretieren und zu nutzen, ist falsch und je nach Einsatzgebiet, z.B. Atommüll-Endlagerstätten-Modellierung, vielleicht gefährlich.

Abbildungsverzeichnis

3.1	Datenkonversion: Anionen/Kationen in fiktive Verbindungen	8
4.1	Jänecke-Dreieck	11
7.1	Meanvalue-Interpolation: 3D Dichteverlauf	28
7.2	Isolinien mit globalem Shepard-verfahren	29
7.3	Isolinien mit lokalem Shepard-verfahren	29
7.4	Isolinien mit Meanvalue-Interpolation	29

Literaturverzeichnis

- [BBD⁺05] BECKER, Frank-Michael ; BOORTZ, Gunter ; DIETRICH, Prof. V. ; ENGELMANN, Dr. L. ; ERNST, Dr. C. ; FANGHÄNEL, Dr. G. ; HÖHNE, Heinz ; LENERTAT, Rudi ; LIESENBERG, Dr. G. ; MEYER, Prof. L. ; PEWS-HOCKE, Dr. C. ; SCHMIDT, Dr. Gerd-Dietrich ; STAMM, Dr. R. ; WEBER, Prof. K.: *Formeln und Tabellen für die Sekundarstufen I und II*. 12. Aufl. Berlin : Duden Paetec Schulbuchverlag, 2005 **3**
- [CBTR02] COHEN-ADAD, R. ; BALAREW, Chr. ; TEPAVITCHAROVA, S. ; RABADJIEVA, D.: Sea-water solubility phase diagram. Application to an extractive process. In: *Pure Appl. Chem.* Vol. 74 (2002), Nr. 10, S. 1811–1821 **17**
- [DKS03] DIETRICH, Dr. A. ; KEIDEL, Dipl.-Ing R. ; STAHL, Prof. I.: Stoffdatensammlung relevanter Verbindungen in der Kali- und Steinsalzindustrie. In: *Kali und Steinsalz* (2003), Nr. 03, S. 18 **10**
- [FB72] FLUCK, Ekkehard ; BECKE-GOEHRING, Margot: *Einführung in die Theorie der quantitativen Analyse*. Dresden : Verlag Theodor Steinkopff, 1972, S. 16 **3, 4**
- [Gär93] GÄRTNER, Harald: *Compact Handbuch Chemie*. München : Compact Verlag, 1993 **3**
- [HL92] HOSCHECK ; LASSER: *Grundlagen der geometrischen Datenverarbeitung*. 2. neubearbeitete und erweiterte Aufl. Stuttgart : B.G. Teubner, 1992 **13, 15, 19**
- [Hor05] HORMANN, Kai: Barycentric Coordinates for Arbitrary Polygons in the Plane / Institut für Informatik, Technische Universität Clausthal. 2005 (IfI-05-05). – Forschungsbericht **22**
- [Mon89] MONNIN, Christophe: An ion interaction model for the volumetric properties of natural waters: Density of the solution and partial molal volumes of electrolytes to high concentrations at 25°C. In: *Geochimica et Cosmochimica Acta* Vol. 53 (1989), 06, Nr. 6, S. 1177–1188 **10**
- [NW72] NYLÉN, Paul ; WIGREN, Nils: *Einführung in die Stöchiometrie*. 16. Aufl. Darmstadt : Steinkopff, 1972, S. 156 **5**

-
- [She] SHEWCHUK, Jonathan R. *Triangle: A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator & Triangle: Definitions.* <http://www.cs.cmu.edu/~quake/triangle.html> & <http://www.cs.cmu.edu/~quake/triangle.defs.html> 15
- [Sie] SIEMANN. *Vorlesungsskript Geochemie III: Marine Evaporite - Teil 3: Salinare Mehrstoffsysteme - Vom binären zum quinären System* 11
- [Spä91] SPÄTH, H.: *Zweidimensionale Spline-Interpolations-Algorithmen.* München : Oldenbourg Verlag, 1991 20
- [SS94] SIGG, Laura ; STUMM, Werner: *Aquatische Chemie: Einführung in die Chemie wässriger Lösungen und natürlicher Gewässer.* 3. vollständig überarbeitete und erweiterte Aufl. Verlag für Fachvereine (Zürich) und Teubner (Stuttgart), 1994 4
- [ST90] SEIM, Rolf (Hrsg.) ; TISCHENDORF, Gerhard (Hrsg.): *Grundlagen der Geochemie.* 1. Aufl. Leipzig : VEB Deutscher Verlag für Grundstoffindustrie, 1990, S. 111f 3, 4
- [TUC05] *Zur Verfügung gestellt durch das Institut für Mineralogie und Mineralische Rohstoffe, Technische Universität Clausthal.* 2005 4, 8, 9
- [Voi90] VOIGT, Hans-Jürgen: *Hydrogeochemie.* Leipzig : Springer Verlag und VEB Deutscher Verlag für Grundstoffindustrie, 1990 4
- [Voi01] VOIGT, W.: Solubility equilibria in multicomponent oceanic salt systems from $t = 0$ to 200°C . Model parameterization and databases. In: *Pure Appl. Chem.* Vol. 73 (2001), Nr. No. 5, S. 831–844 4
-