Chair of Reservoir Engineering

Master's Thesis

Numerical Simulation of Microbial Interactions in Underground Hydrogen Storage with MRST

Nikola Zoric, BSc

November 2022

MONTANUNIVERSITÄT LEOBEN
www.unileoben.ac.at

AFFIDAVIT

I declare on oath that I wrote this thesis independently, did not use other than the specified sources and aids, and did not otherwise use any unauthorized aids.

I declare that I have read, understood, and complied with the guidelines of the senate of the Montanuniversität Leoben for "Good Scientific Practice".

Furthermore, I declare that the electronic and printed version of the submitted thesis are identical, both, formally and with regard to content.

Date 07.11.2022

Signature Author
Nikola Zoric

MONTANUNIVERSITÄT LEOBEN
www.unileoben.ac.at

BSc
Nikola Zoric
Janka Veselinovica 24
21000 NOVI SAD
SERBIA

To the Dean of graduate Studies of the Montanuniversitaet Leoben

Declaration of Approval for the Digital Publication of Scientific Theses

I am aware that the thesis entitled "Numerical Simulation of Microbial Interactions in Underground Hydrogen Storage with MRST" will be subject to a plagiarism assessment and may be stored by Montanuniversität Leoben for an unlimited period of time.

I agree that the University Library of Montanuniversität Leoben may publish the thesis open access in the World Wide Web. For embargoed theses this will be done after the embargo expires.

Note: in case you refuse the open access publication in the World Wide Web, the thesis will only be published in printed form (after a possible embargo has expired) in the University Library (dissertations also in the Austrian National Library).

I hereby agree with the open access publication of my thesis on the World Wide Web:

☒ Yes

O No

Date 07.11.2022

Signature Author

Author Nikola Zorić

Master Thesis 2022

Petroleum Engineering

Numerical Simulation of Microbial Interactions in Underground Hydrogen Storage with MRST

Supervisor: Dr. Siroos Azizmohammadi

Chair of Reservoir Engineering

dpe

MONTAN UNIVERSITÄT LEOBEN

To

Novica,

Vesna,

and Nastasija

Acknowledgments

I am thankful for the selfless support and guidance provided by Dr. Siroos Azizmohammadi. Also, I would like to thank my colleagues Patrick Jasek and Gerald Stiedl for the fruitful discussions and comments.

Also, my gratitude goes to Dr. Birger Hagemann whose work was the basis for the development of the code and model that is the core of this thesis.

Abstract

As the interest in hydrogen as an energy carrier, for an energy storage, grows, with it grows the effort for a better understanding of the working environment and the conditions that can be encountered. Depleted natural gas reservoirs are so far considered the best option for underground hydrogen storage. Seeing how they bring an option for a large storage volume, whose rocks have a history of being in contact with the gas. An extensive library of collected data and knowledge about the reservoir, throughout the years of production, only adds to the favorability of such storage. In the reservoirs where such implementation has been carried out an interesting phenomenon has been observed, namely various species of microorganisms that thrive at the rough reservoir conditions have been using stored hydrogen as an energy source for their metabolism. The focus of this work is set on a certain species of these microorganisms called *methanogenic archaea*, which are producing methane as they consume hydrogen. In other words, stored hydrogen gas is converted to methane in the presence of these microorganisms. Little is known about the underlying physics and influencing parameters on the microbial conversion of underground hydrogen storage. Therefore, many laboratory tests, as well as simulations, are being done in order to broaden our knowledge and familiarize ourselves with the expected outcomes when it comes to these types of microorganisms and their activity. We intend to implement a developed mathematical model into the *MRST* (*The MATLAB Reservoir Simulation Toolbox*) code and therefore simulate a one-dimensional injection of the nutrient gas mixture into the core sample that has been populated with methanogenic archaea. The model consists of two equations, one describing the population growth of the microbial species and the other describing the effects of metabolic bioreaction (called methanogenesis), those being nutrient (H_2 and CO_2) consumption and methane production. Once developed, this code can serve as an assisting tool for future laboratory experiments and a basis for full reservoir scale simulations that could predict the expected effects in a more realistic environment.

Zusammenfassung

Mit dem wachsenden Interesse an Wasserstoff als Energieträger zur Energiespeicherung, wächst auch der Aufwand für ein besseres Verständnis der Arbeitsumgebung und der anzutreffenden Bedingungen. Erschöpfte Erdgaslagerstätten gelten bisher als beste Option für die unterirdische Wasserstoffspeicherung, da Gesteine die in der Vergangenheit mit Erdgas in Kontakt waren die Möglichkeit bieten enorme Volumina an Gas zu speichern. Eine umfangreiche Bibliothek mit gesammelten Daten und Wissen über eine Lagerstätte durch die Jahre der Produktion hinweg, trägt dazu bei bevorzugt als Lagerstätte für Wasserstoff in Frage zu kommen. In jenen Lagerstätten, mit eben solcher Implementierung, wurde ein interessantes Phänomen beobachtet, nämlich verschiedene Arten von Mikroorganismen, die unter den rauen Lagerstättenbedingungen nicht nur gedeihen, sodern den gespeicherten Wasserstoff als Energiequelle für ihren Stoffwechsel nutzen. Der Fokus dieser Arbeit liegt auf einer bestimmten Spezies dieser Mikroorganismen, den *methanogene Archaeen*, welche Methan produzieren, während sie Wasserstoff verbrauchen. Mit anderen Worten, gespeichertes Wasserstoffgas wird in Gegenwart dieser Mikroorganismen in Methan umgewandelt. Über die zugrunde liegende Physik und die Einflussparameter der mikrobiellen Umwandlung unterirdischer Wasserstoffspeicher ist wenig bekannt. Um unser Wissen daher zu erweitern, werden viele Labortests sowie Simulationen durchgeführt. Ziel ist, uns mit den erwarteten Ergebnissen vertraut zu machen, speziell wenn es um diese Art der Mikroorganismen und ihre Aktivität geht. Wir beabsichtigen, ein mathematisches Modell im *MRST* (*The MATLAB Reservoir Simulation Toolbox)* Code zu implementieren, um damit eine eindimensionale Injektion des Nährgasgemisches in die mit methanogenetischen Archaeen besiedelte Kernprobe zu simulieren. Das Modell besteht aus zwei Gleichungen, von denen die erste das Populationswachstum der mikrobiellen Spezies beschreibt. Die zweite Gleichung beschreibt die Auswirkungen der metabolischen Bioreaktion (=Methanogenese), welche als Verbrauch von Nährstoffen (H_2 und CO_2) und der daraus folgenden Methanproduktion ersichtlich wird. Einmal entwickelt, kann dieser Code als unterstützendes Werkzeug für zukünftige Laborexperimente und als Grundlage für Simulationen im vollen Reservoirmaßstab dienen, um die erwarteten Auswirkungen in einer realistischeren Umgebung vorhersagen zu können.

Table of Contents

Chapter 1

Introduction

As the focus of the energy industry changes towards renewable supply systems hydrogen emerged as the feasible option for the energy carrier role. Even though there have been suggestions in the past regarding hydrogen-based power systems, some even dating to the early years of the past century, it was not until recent years that a new spike in interest for hydrogen, for the mentioned role, really started to show.

When it comes to renewable systems (sun, wind or flowing water being dominating sources), high dependency on the everchanging weather conditions presents itself. The resulting fact is that there are highly fluctuating periods in energy production. When there is an excess in production – energy gets wasted, on the other hand, when there is a lack of optimal weather conditions – there is no production. In both of these scenarios, there is an obvious need for energy storage. In the first, not to lose produced electricity and in the second to have a reserve that can provide continuous supply when the production is lacking (Hagemann, 2017; Zivar et al., 2021).

From different Electrical Energy Storage (EES) options Underground Hydrogen Storage (UHS) shows high potential. From which most interesting are the depleted natural gas reservoirs, considering that they provide high volumes in which gas can be stored, as well as such an environment where gas presence is a familiarity, additionally years of knowledge gathered throughout the life of a natural gas reservoir are available. Another point in favor of UHS is the fact that hydrogen has a high energy density per mass (around 120 MJ/kg), and also it is a 'low-carbon energy carrier' which goes in favor of decarbonization on a global level. To quote United Nations Industrial Development Organization *"Hydrogen is a true paradigm shift in the area of more efficient energy storage, especially for renewable energy on industrial scale…"* (Heinemann et al., 2021)

So far there are two concepts regarding hydrogen storage and its utilization. One is “POWER-to-GAS”, where produced hydrogen is added to the existing natural gas grid. H_2 concentrations in this “mixture” are to be in low (single-digit) percentages. Second is the so-called “POWER-to-GAS-to-POWER”, here pure hydrogen is stored in the subsurface. Different options include depleted gas or oil reservoirs, aquifers and solution-mined salt caverns, from where, dictated by the demand, hydrogen is produced and used as an energy fuel. (Hagemann, Rasoulzadeh et al., 2016)

1.1 UHS

Underground hydrogen storage is a temporal storage of energy-carrying hydrogen in the subsurface formation, having similarities to the natural gas reservoirs. UHS can provide long-term, high-capacity energy storage at a lower cost compared to some other storing options (such as batteries, which are preferred as short-term storage).

In Electrical Energy Storages (EES), energy holds a form characteristic to the technology that is being used, and then in time of need, it is converted to electrical energy (Luo et al., 2015). When it comes to UHS energy is stored as *chemical energy* by utilizing hydrogen as a carrier. Stored hydrogen is later on produced, therefore forming a cycle.

UHS cycles include:

- Electrolysis process, where water is split into hydrogen and oxygen. Energy used comes from the produced electricity excess.
- Compression and transport of the hydrogen to the wells and from there to the subsurface reservoir. As mentioned already when describing the “POWER-to-GAS-to-POWER” concept, H_2 remains until needed in the formation storage.
- Need for energy calls for H_2 production. Once produced it is also processed if necessary (Hagemann, 2017; Hagemann, Rasoulzadeh et al., 2016).

When storing hydrogen in the formation, pre-injection of cushion gas, either nitrogen (N_2) or methane (CH_4), is carried out, after which the injection of hydrogen occurs. The role of the cushion gas is pressure maintenance in order to achieve an optimal production rate of the hydrogen, while the cushion gas itself is not to be produced (nevertheless, some amounts are produced in a mixture with H_2 which then requires separation) (Zivar et al., 2021).

For a formation to be considered as a UHS it needs to have sufficient volume capacity, a good sealing structure so that there are no losses to the surrounding formations and also the necessary injection-production rates must be possible (conditioned by the properties of the rocks such as porosity, permeability etc.).

As of now, possible options include salt caverns, aquifers and depleted gas or oil reservoirs.

Caverns provide no limitations to the injection-production rates, while highly enclosed structure ensures almost none-existing losses. Cushion gas is allowing compression and decompression of the working gas, which is how the cycles are carried out. Limiting sides to this strategy are the available volumes, investment into the cushion gas, as well as the availability of the salt caverns in the domes.

Aquifers and *depleted gas or oil reservoirs*, both consist of a porous rock that serves as the storing medium and a tight sealing rock preventing the losses. Compared to salt caverns properties of the porous rock are limiting the injection-production rates, and due to the much bigger size of the reservoir possibilities for leakages and therefore losses are increased. What else, reactions between hydrogen and rock minerals and also bioreactions are adding to the losses. What separates the two and makes the latter a more favorable option is the much higher storage availability, whereas with the salt caverns volumes of hydrogen gas are around $6 * 10^7 m^3$ and here possible volumes can be higher than $10^9 m^3$. Also, these structures are having bigger availability than what is the case with the salt caverns. The usage of cushion gas as a means of pressure maintenance is a common factor for both. As for the differences between the three (aquifer, gas reservoirs and oil reservoirs), they lie in the type of medium that was present in the pores. With the aquifers, we are talking about brine while with depleted reservoirs up to three different phases (brine, gas, oil) (Hagemann, 2017).

1.2 Bioreactions

Field operations regarding hydrogen storage are based on salt caverns when it comes to pure hydrogen storage, aquifers or depleted reservoirs have so far stored gas mixtures containing hydrogen, such as town gas (a mixture of carbon monoxide and hydrogen). Reports in town gas storages have been showing considerable reductions in gas volumes, what was concluded as the most probable cause was the presence of microbial species which during their metabolism are consuming stored gas. Namely, so far four microbial processes have been marked as most affecting in the UHS scenario (Hagemann, 2017; Zivar et al., 2021):

- Methanogenesis, where organisms are consuming CO_2 and H_2 and are producing methane with water, converting therefore stored hydrogen gas into methane. An example was with the town gas storage in Lobodice, Czech Republic, where with the decrease in H_2, CO_2 and CO an increase in CH_4 was observed (Buzek et al., 1994).

$$CO_2 + 4H_2 \rightarrow CH_4 + 2H_2O$$

- Sulfate-reduction

$$SO_4^{2-} + 5H_2 \rightarrow H_2S + 4H_2O$$

- Acetogenesis

$$2CO_2 + 4H_2 \rightarrow CH_3COOH + 2H_2O$$

- Iron-reduction

$$3Fe_2^{III}O_3 + H_2 \rightarrow 2Fe_3^{III}O_4 + H_2O$$

1.3 Thesis Agenda

Throughout the years there have been projects and pilot tests executed in order to better understand the conditions and environment behind hydrogen underground injection and storage as well as to test possible storage sites throughout the world. Below are the tables summarizing some of these, as presented by Davood Zivar, Sunil Kumar and Jalal Foroozesh in their work *"Underground hydrogen storage: A comprehensive review"* (*Table 1* & *Table 2*). What can be observed is that there is an ongoing interest in further understanding and developing when it comes to the application of UHS.

Table 1: UHS operating sites (Zivar et al., 2021)

Field/project name	Storage type	H_2 (%)	Working condition	Depth (m)	Volume (m^3)	Status
Teesside (UK)	Bedded salt	95	45 bar	365[a]	210,000	Operating
Clemens (USA)	Salt dome	95	70–137 bar	1,000[a]	580,000	Operating
Moss Bluff (USA)	Salt dome		55–152 bar	1,200[a]	566,000	Operating
Spindletop (USA)	Salt dome	95	68–202 bar	1,340[a]	906,000	Operating
Kiel (Germany)	Salt cavern	60	80–100 bar		32,000	Closed
Ketzin (Germany)	Aquifer	62	Not reported	200–250	Not reported	Operating with natural gas
Beynes (France)	Aquifer	50	Not reported	430	3.3×10^8	Operating with natural gas
Lobodice (Czech Republic)	Aquifer	50	90 bar/34 °C	430	Not reported	Operating
Diadema (Argentina)	Depleted gas reservoir	10	10 bar/50 °C	600	Not reported	Not reported
Underground Sun Storage (Austria)	Depleted gas reservoir	10	78 bar/40 °C	1000	Not reported	Operating

[a] Mean depth is reported for the salt caverns.

Table 2: UHS potential sites (Zivar et al., 2021)

Location	Storage type	Properties	Criteria
San Pedro belt, Spain [36]	Saline aquifer	Porosity (ϕ) = 0.2 Permeability (k) = 100 mD	Recovery ratio for seasonal storage of hydrogen produced from wind power
Rough gas storage facility, UK [30]	Depleted natural gas reservoir	Capacity = 48 million m^3 ϕ = 0.2 k = 75 mD Depth = 2743 m Pressure = 50–100 bar Period = 120 days	-Chemical stability -Biological consumption -Leakage -Operational condition
- Ocna Mures - Targu Ocna - Ocnele Mari - Cacica (Romania) [64]	Salt cavern	Not reported	-Geological criteria -Brine availability and consumption -Storage location
- Northern Nordrhein Westfalen - northwest Germany - central Germany [33]	Salt cavern	Capacities = – 2.4 billion m^3 – 4.6 billion m^3 – 1.8 billion m^3	Economical evaluation for switching from hydrocarbon storage to hydrogen storage
Salina B and A2, Ontario, Canada [22]	Salt cavern	-B Depth = 400 m Thickness = 90 m Capacity = 6.4 million m^3 -A2 Depth = 525 m Thickness = up to 45 m Capacity = 9.5 million m^3	-Geological criteria, such as depth and mineralogy. -Switching from hydrocarbon storage to hydrogen storage
Mount Simon aquifer, Ontario, Canada [22]	Saline aquifer	Depth = 800 m ϕ = 5–15% Pressure = 76 bar Salinity = 100 k-300 k mg/l Capacity = 725 million tons of CO_2	Based on previous evaluation of CCS
Midland Valley, UK [24]	Oil resrvoir	k = 60–80 mD Thickness = 100–1000 m	-Geological uncertainty -Storage capacity
- ROGOZNO - DAMASLAWEK - LANIETA - LUBIEN - GOLENIOW - IZBICA KUJAWSKA - DEBINA (Poland) [23]	Salt cavern	Not reported	- Size/area of the salt dome - Depth of the salt mirror - Recognition of salt dome - The complexity of the internal structure - Existing geological reports and salt reserves
Gora region, Poland [65]	Salt structure	Not reported	- Reservoir lithology - Stage of exploration - Type of salt deposit - Reservoir volume - Depth - Geothermal gradient
Chabowo T, Poland [65]	Aquifer	Not reported	- Tectonic activity - Overburden rock lithology - Stage of exploration - Depth - The pore volume of the reservoir
Przemysl, Poland [65]	Natural gas field	Not reported	- Overburden rock lithology - Tectonic activity - Deposit form - The pore volume of the reservoir - Depth - Stage of exploration
Rhaetian, Schleswing-Holstein, Germany [39]	Gas reservoir	ϕ = 13–33% k = 2.1–572.2 mD Depth = 460–490 m Thickness = 5–30 m Pressure = 65 bar	- Feasibility of hydrogen storage into the proven possible natural gas reservoir - Storage performance (deliverability)
A1 (the authors named it A1), Turkey [66]	Salt cavern	Not reported	- Technique characteristics - Costs - Socio-economic characteristics - Risks
Tuz Golu gas storage site, Turkey [67]	Salt cavern	Capacity = 12 × 630,000 m^3 Depth = 1100–1400 m Max. Internal gas pressure = 220 bar	Switching from hydrocarbon storage to hydrogen storage
Lille Thorup, Denmark [68]	Salt cavern	Depth = 1270–1690 m Pressure = 50–100 bar Temperature = 40–50 °C Capacity = 445 × 10^6 m^3	Switching from hydrocarbon storage to hydrogen storage

Besides the ongoing and planned field testing, there are also related research projects that are following. Some are consisting out of numerical simulations, others are combining both field testing and simulating, adding to these laboratory tests as well. Some of these are:

- ***H2STORE*** in Germany (Ganzer et al., 2013);
- ***HyUnder*** in European Union countries (Simon et al., 2015);
- ***Hychico*** in Argentina (Sebastien, 2016);
- ***ANGUS+*** in Germany (Kabuth et al., 2017);
- ***Underground Sun Storage*** in Austria (Bauer and Pichler, 2017);
- ***HyINTEGER*** in Germany (Hagemann, 2017).

Work that will be described in this thesis is focusing on the depleted gas storage conditions and is aiming to numerically simulate the effects of *methanogenic archaea* microorganisms present in underground storages, where as a result of their metabolism stored hydrogen is reduced as a generation of methane occurs in return.

The scenario is set on a core scale, complementing laboratory-planned and ongoing tests. In which injection of a gas mixture containing hydrogen and carbon dioxide is introduced to the environment already pre-populated with methanogenic archaea.

The simulation will be done using *'The MATLAB Reservoir Simulation Toolbox (MRST)'* an open-source library, developed for reservoir simulation and prototyping. With the intent to expand our understanding of the underlying physics and influencing parameters on the microbial conversion of underground hydrogen storage.

Using MRST would be a different approach compared to what was done so far, and is available in the literature, where the majority of cases relied on DuMu^{X} open-source code as the simulating tool.

Chapter 2

Microbes – an Overview

Microorganisms are defined as living organisms that are of microscopic size (micrometers, sometimes nanometers and millimeters), most of them being too small to be seen by the human eye. Currently accepted classification of cellular life forms consists of three domains Archaea, Bacteria and Eucarya. The first two belong to Prokaryotes and the third belongs to Eukaryotes (Talaro and Chess, 2018).

2.1 Prokaryotes' Structure

Prokaryotes are simple, usually single-celled organisms that do not have a nucleus nor membrane-bound cell structures (organelles), while eukaryotes in turn are complex and contain both. While all prokaryotes are microorganisms, only some eukaryotes belong to the microorganisms. Relevant for this work are the ones that can survive in the UHS conditions, those being some archaea and bacterial species.

Necessary parts for all prokaryotes include cell membrane, cytoplasm, ribosomes and chromosomes. Some species have additional parts such as cell wall, surface coating or even appendages used for directional movement or adhesion (*Figure 1)* (Talaro and Chess, 2018).

Although both archaea and bacteria have similar structures and metabolism the key differences are still existing when it comes to genetics and biochemistry (genetic processes of transcription and translation, as well as materials that walls and membranes are consisted of). Archaea are considered to be one of the oldest forms of life, and are capable to survive and even thrive in some of the most extreme conditions, which is why their presence in UHS is not a surprise (Gentry et al., 2015).

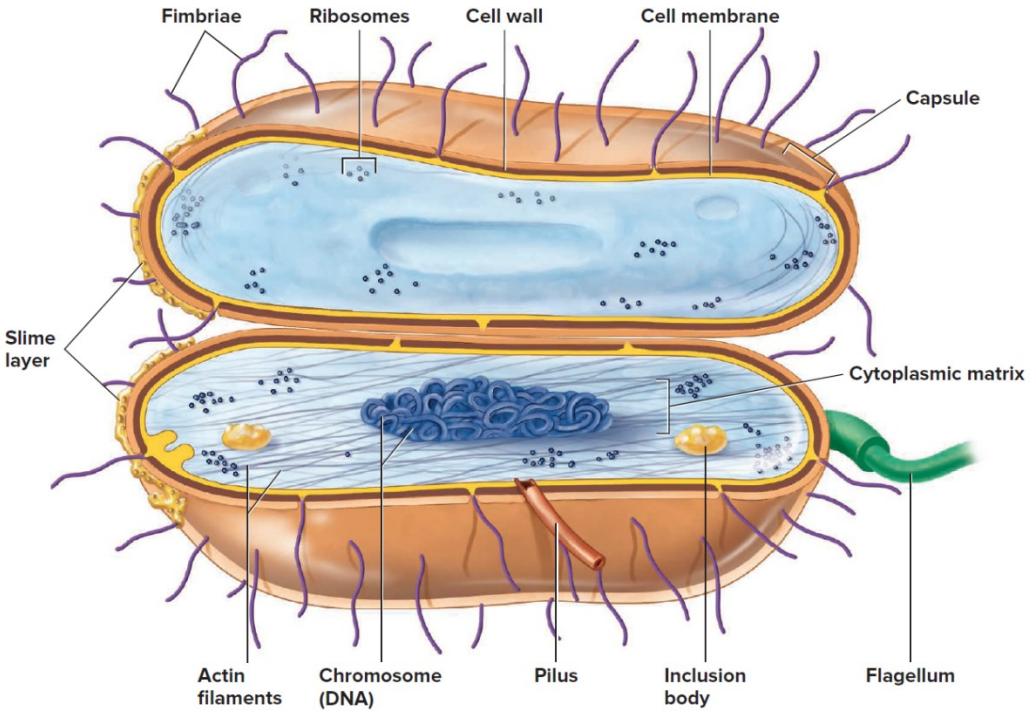

Figure 1: Example of a bacterial cell structure (Talaro and Chess, 2018)

- **Cytoplasm,** or cytoplasmic matrix. Represents the internal filling medium of bacteria, not only does it hold all of the internal parts of the microorganism (chromosome, ribosomes, granules, and actin strands) it is also here that the necessary biochemical activities are carried out. Consisting mostly of water it is a perfect solvent for nutrients.
- **Chromosome**, organism's DNA containing genetic information. As a characteristic of all prokaryotes, there is no nuclear membrane surrounding the chromosome, part of the bacteria where it is located is known as the *nucleoid*.
- **Ribosomes** are distributed throughout the cytoplasm and are responsible for protein synthesis.
- **The cell membrane** located underneath the cell wall is a selectively permeable layer that stretches itself around the cytoplasm and serves as a filter dictating what comes into the cell (nutrients) and what comes out (wastes). The cell wall is a surrounding rigid layer, responsible for the shape and protection of the cell (Gentry et al., 2015; Talaro and Chess, 2018).

2.2 Metabolism and Growth

Microbial metabolism is activated when nutrients and needed surrounding factors are available, a result of metabolic activation is growth. Two types of growth can be differentiated, one where cell size is increased and the other where cell number increases. The latter is a result of cell division. The singular parent cell engages in chromosome duplication as well as in cell wall thinning in the middle as it grows double in size, finally resulting in two daughter cells (*Figure 2*) (Talaro and Chess, 2018).

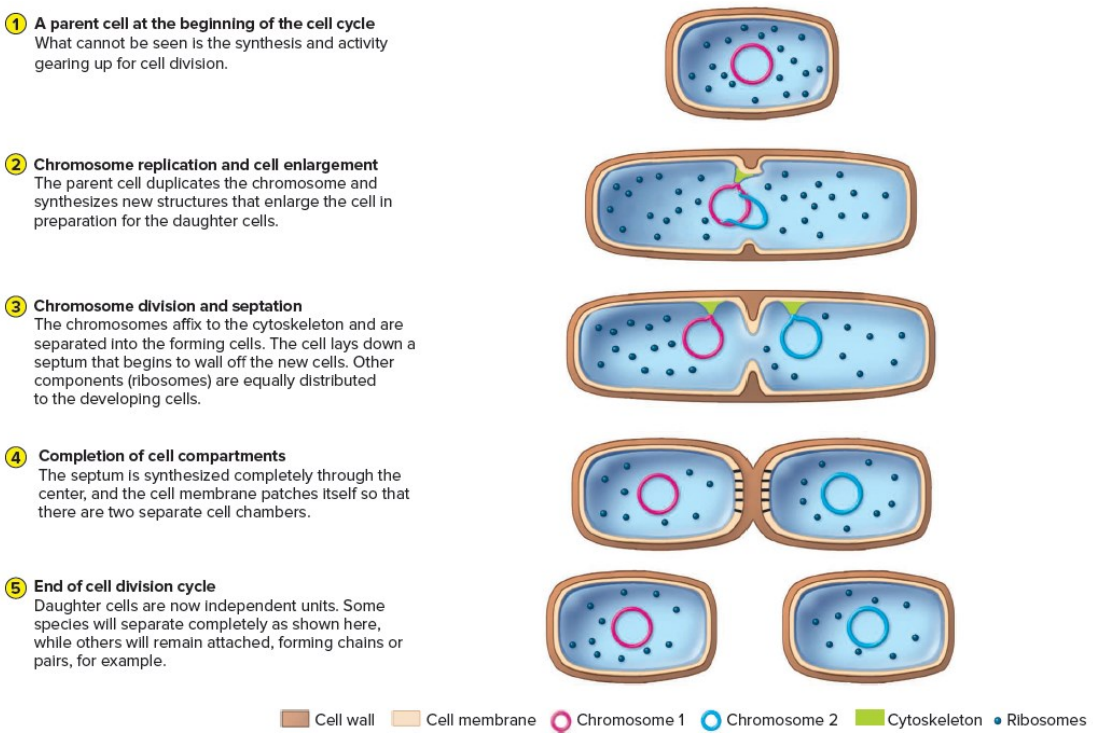

Figure 2:Steps in microbial duplication (Talaro and Chess, 2018)

Another differentiation is present, but now when it comes to metabolism. Namely, besides growth metabolism, whose result was just described, there is also a *non-growing metabolism.* Lacking growth and cell duplication, nutrient consumption serves just as a mean of cell maintenance.

For a functioning metabolism microbes need an *energy source* and a *carbon source*. Depending on outside conditions these are obtained in various ways. The interest here is in the UHS conditions, where the source of energy lies in hydrogen (which is in an abundance in UHS) serving as an electron donor for the redox reaction (hydrogenotrophic metabolism). The second requirement is fulfilled by an inorganic source of carbon found in CO_2 (autotrophy) or from an organic one like glucose (heterotrophy) (Gentry et al., 2015).

2.3 Microbial Population

Described duplication is a continuous process, as long as the required conditions are present each daughter cell becomes the parent, therefore with each new generation number of cells is doubled. The resulting growth is exponential, due to which microorganisms are occurring in a form of a population (having a very high number of individual cells) (Gentry et al., 2015) (*Figure 3*).

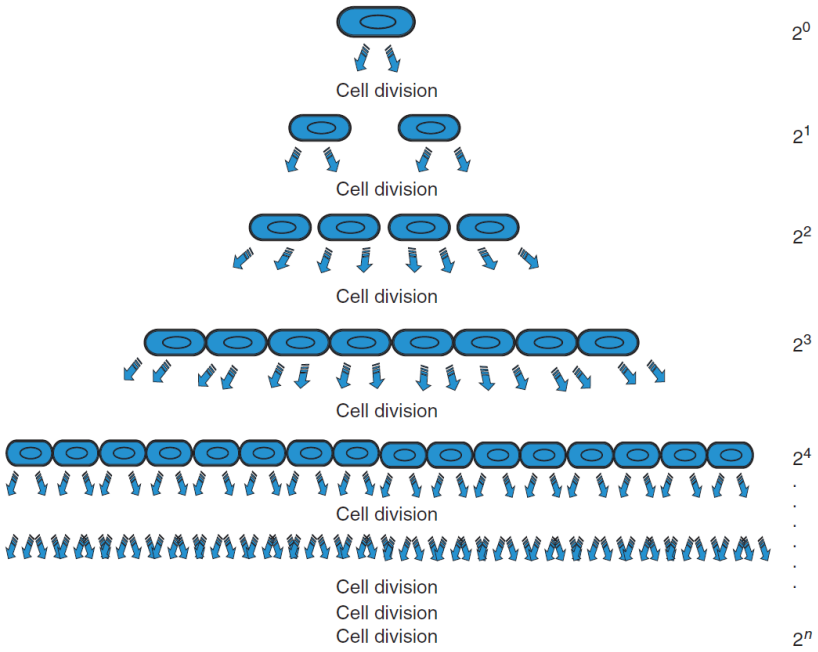

Figure 3: Exponential cell growth (Gentry et al., 2015)

As mentioned, as long as the conditions are favorable population will grow, but this does not go on forever, many different factors are controlling the cells' maximum-rate reproduction. Laboratory batch culture studies have shown that what typically occurs is a growth curve over time (*Figure 4*). Microbial culture is contained in a bottle and fed with nutrients under controlled conditions. Over time, samples are taken and analysis is carried out concerning biomass $[kg]$ (sum of individual microorganisms' mass) or microbial counts $\left[\frac{cells}{ml}\right]$ (or $\left[\frac{cells}{g}\right]$, depending on whether the count is related to a volume or total mass).

The resulting growth curve (semi-log plot) contains different stages – the lag phase, the exponential growth (log) phase, the stationary phase and finally the death phase (*Figure 4*) (Talaro and Chess, 2018).

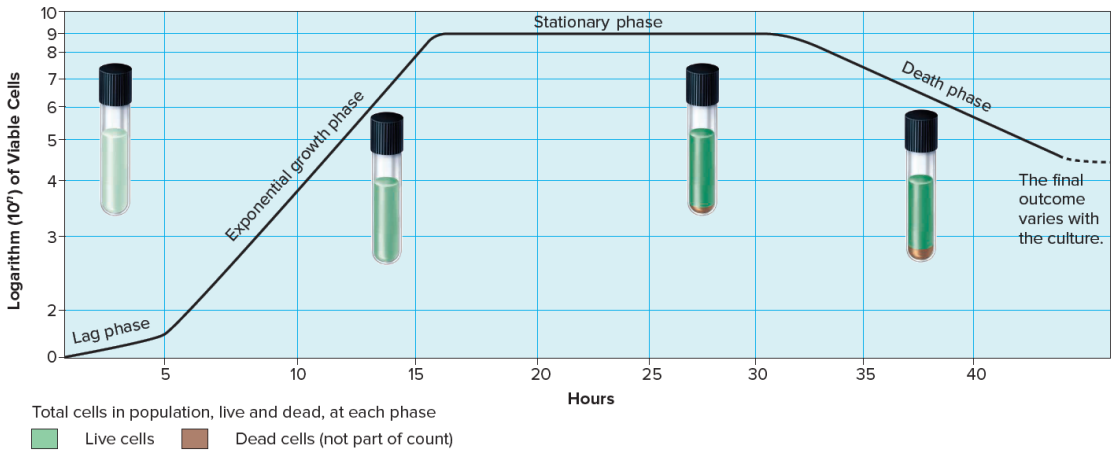

Figure 4: Microbial population growth curve over time (Talaro and Chess, 2018)

The lag phase represents the initial stage of population growth, it is characterized by the fact that either no growth appears to be happening, or it is at much lower rates compared to the exponential rate. This occurs due to the introduced cells' need for a period of adjustment to the new outside conditions. Even though the population itself is not growing, individual cell's metabolism is active as division is in preparation. Under laboratory conditions, this phase lasts from minutes up to some hours (Gentry et al., 2015; Talaro and Chess, 2018).

The exponential growth (log) phase, is the stage at which population growth starts to increase exponentially (*Figure 3*). The average time of growth between generations is 30 minutes to an hour at optimal conditions, while at perfect scenario it takes 5 to 10 minutes, and at unfavorable conditions, it can take days and even months or years. Rates depend on both the type of microorganism as well as outside conditions (Talaro and Chess, 2018). Population growth during this phase can be mathematically expressed by the equation(Gentry et al., 2015):

$$\frac{dn}{dt} = \mu n \tag{2.1}$$

n... number of cells, or mass of cells $\left[\frac{mass}{volume}\right]$ (depending on the representation choice);

t... time $[preferred\ time\ unit]$;

μ... specific growth rate constant $\left[\frac{1}{time}\right]$ (also referred to as the maximum growth rate).

Generation time (cell division time) or the doubling time (t) as well as the specific growth rate (μ) can both be calculated using the provided equation, given that there is an existing growth curve (such as one from *Figure 4*) from which data can be extracted. This would require rearranging and solving equation (2.1):

$$\frac{dn}{n} = \mu dt \tag{2.2}$$

$$\frac{dn}{n} = \mu dt \tag{2.3}$$

$$\ln n - \ln n_0 = \mu t \text{ or } \frac{n}{n_0} = e^{\mu t} \tag{2.4}$$

Where for population growth of one generation (doubling the n_0):

$$\frac{n}{n_0} = 2 \tag{2.5}$$

meaning that:

$$2 = e^{\mu t} \tag{2.6}$$

The Stationary Phase, when reached results in a state of no net growth. This does not mean that cells are not growing and dividing, they indeed do, it is just that this growth is balanced by the number of cells dying. This phase is expressed by:

$$\frac{dn}{dt} = 0 \tag{2.7}$$

Reasons that are bringing the population growth cycle to this phase are commonly found in the depletion of nutrients required for a functioning metabolism – carbon or energy sources are used up. *Endogenous metabolism* is the term used for when population growth occurs from consuming dead cells, this takes place throughout the whole cycle but is best observable in the stationary phase since all other growth causes are eliminated (on some growth curves seen as the small but observable amount of growth in the stationary phase). Another reason for the stationary phase is in the scenario with a high density of microbial cells, where too much waste has been accumulated throughout the exponential growth phase and therefore is preventing further growth or is even toxic to cells (Gentry et al., 2015; Talaro and Chess, 2018).

The death phase comes as conditions for already limited metabolism become even worse resulting in population decline. It is not only that the death of cells occurs, but some cells go into an idle state in which they remain without growing, and some go into starvation by which resistance to the lack of favorable conditions increases. Endogenous metabolism keeps some cells alive even at this phase. Nevertheless, the net population growth is negative and exponential as well. Decay does have a lesser rate than what is the case with the growth in the exponential phase (Gentry et al., 2015; Talaro and Chess, 2018).

Describing equation is:

$$\frac{dn}{dt} = -bn \tag{2.8}$$

b… specific death rate $\left[\frac{1}{time}\right]$.

2.4 UHS Occurring Microbial Species

A major factor in the viability of an UHS are effects caused by microbial population growth. Microbial survival and reproduction are based on their ability to achieve functional metabolism, which is essentially composed of biochemical reactions in which hydrogen is permanently lost as it is an energy source for different microbial species found under UHS conditions. By injecting hydrogen necessary nutrients for microbes are provided in an abundance leading to biodegradations. Losses come in a form of hydrogen conversion into other gases (CH_4, H_2S etc. depending on microbial species) (Heinemann et al., 2021).

Loss of 17% of hydrogen followed by a reduction in CO_2 and generation of CH_4 was reported in Lobodice, Czech Republic in a town gas storage that was rich with hydrogen, the cause for this was found in the presence of methanogenic archaea (Buzek et al., 1994). Another hydrogen consumption examples are Underground Sun.Storage project in Austria as well as the HyChico project in Argentina. In each, microbial population was detected and as a result percentage of injected hydrogen was converted into CH_4 by methanogenesis (Bauer and Pichler, 2017; Sebastien, 2016). Sulfate-reducing microbes were reported as well, in town gas and natural gas storages, as a result, generation of H_2S was detected, and consequences are the same as abiotically formed H_2S – mainly corrosion-caused problems (Kleinitz and Böhling, 2005).

Besides the biodegradation of hydrogen, pore closure is occurring due to pore space microbial overpopulation. Such an effect is an often-met problem in geothermal and CO_2 operations. As a consequence, reduced hydrogen injectivity and flow rates throughout the reservoir emerge.

This is yet to be investigated properly and in detail when it comes to the UHS scenario, mentioned projects (Underground Sun.Storage and HyChico) did not report pore-clogging effects (Heinemann et al., 2021).

Four chemical reactions caused by microbial presence have been so far identified as important for UHS. Besides already mentioned methanogenesis and sulfate reduction, there are reports about the presence of homoacetogenic archaea and iron-reducing bacteria (Cord-Ruwisch et al., 1988).

Methanogenic archaea's metabolic reaction is as follows:

$$CO_2 + 4H_2 \rightarrow CH_4 + 2H_2O \tag{2.9}$$

Where CO_2 is used as a carbon source (autotrophy).

Homoacetogenic archaea:

$$2CO_2 + 4H_2 \rightarrow CH_3COOH + 2H_2O \tag{2.10}$$

The same sources for energy and carbon as with methanogenic archaea are used here. But they can use an organic carbon source as well.

Sulfate-reducing bacteria:

$$SO_4^{2-} + 5H_2 \rightarrow H_2S + 4H_2O \tag{2.11}$$

The most common source of carbon is of an organic origin (heterotrophy), but some types of bacteria use CO_2 as well.

Iron-reducing bacteria:

$$3Fe_2^{III}O_3 + H_2 \rightarrow 2Fe_3^{III}O_4 + H_2O \tag{2.12}$$

Here are also possible both options for a carbon source, with the organic one being what most species use.

What occurs when multiple species coexist is the so-called competition for nutrients. The ones that have the lowest microbial concentration threshold, after which they start consuming hydrogen, are the ones to dominate the competition (Cord-Ruwisch et al., 1988; Lovley and Phillips, 1987). But in the UHS conditions where hydrogen is in high volumes, and can be considered as an always available nutrient, species end up coexisting.

Chapter 3

Mathematical Model and Governing Equations

The basis for this work is the mathematical model developed by Hagemann (Hagemann, 2017; Hagemann, Rasoulzadeh et al., 2016). Model is considering up to two phases (gas and water) saturating the reservoir, it is combining flow and transport with biochemical processes of the microbes. Scale which he used is a continuum one, where an upscaling from discrete scale has been done, to do so representative elementary volume (REV) was used (*Figure 5*). This is the smallest volume that is containing average parameters implemented in the model (porosity, saturation, mole fraction, microbial density).

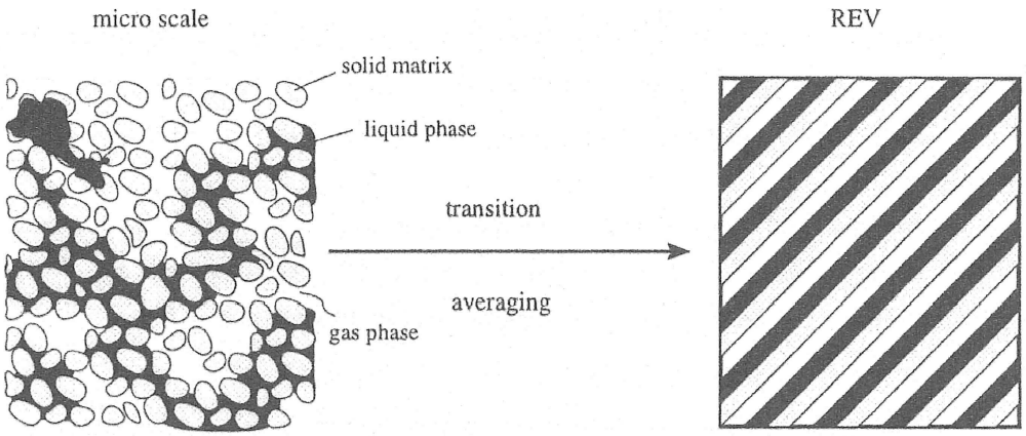

Figure 5: Going from discrete to the continuum scale (Hagemann, 2017)

Porosity was calculated as pore volume divided by total volume; *saturation* as phase volume divided by pore volume; *mole fractions* as a molar volume of a component in a phase divided by the total molar volume of that phase; *microbial density* as the number of microbial cells divided by total volume.

The approach was to first find adequate equations dealing with physio-chemical processes and the ones dealing with biochemical processes before coupling them into a system of differential equations.

3.1 Physio-Chemical Processes

Mass (mol) conservation of chemical components in different phases (water or gas) considering advective and dispersive/diffusive transport (Hagemann, 2017; Hagemann, Rasoulzadeh et al., 2016):

$$\phi \frac{\partial(\hat{\rho}_v c_v^k S_v + \hat{\rho}_l c_l^k S_l)}{\partial t} + \nabla\left(\hat{\rho}_l c_l^k v_l + J_l^k + \hat{\rho}_v c_v^k v_v + J_v^k\right) = q^k \tag{3.1}$$

ϕ … porosity; $\hat{\rho}$… molar density $\left[\frac{mol}{m^3}\right]$;

c… mole fraction; S … saturation;

q … source/sink; k … chemical component;

v… vapor phase; l… liquid phase.

v_i is the Darcy velocity of the phase in $\left[\frac{m}{s}\right]$ (dictating advective transport):

$$v_i = -\frac{K k_{ri}}{\mu_i}(\nabla P_i - \rho_i g) \tag{3.2}$$

$i = v, l;$

K… absolute permeability $[m^2]$; k_r… relative permeability;

μ… dynamic viscosity $[Pa.s]$; ∇P_i… phase pressure gradient $\left[\frac{Pa}{m}\right]$;

ρ_i… phase density $\left[\frac{kg}{m^3}\right]$; g… gravity acceleration $\left[\frac{m}{s^2}\right]$.

J is dispersive/diffusive flux in $\left[\frac{mol}{m.s}\right]$, consisting out of molecular diffusion and mechanical dispersion:

$$J_i^k = -\hat{\rho}_i (D_{diff,i}^k + D_{disp,i}^k)\nabla c_i^k \tag{3.3}$$

$i = v, l;$

$D_{diff,i}^k$… effective molecular diffusion coefficient (of component k in phase i) $\left[\frac{m^2}{s}\right]$;

$D_{disp,i}^k$… effective mechanical dispersion coefficient (of component k in phase i) $\left[\frac{m^2}{s}\right]$.

Molar balance equation is finalized by:

$$S_v + S_l = 1 \tag{3.4}$$

$$\sum_k c_i^k = 1, \quad i = v, l \tag{3.5}$$

3.2 Biochemical Processes

Earlier studies had a structured approach regarding microbial population, in one such study population was divided into a part which is attached (to the solid) biomass called biofilms, and a part which is suspended (in the aqueous phase) biomass (Murphy and Ginn, 1999). Another work divided the population into biomass in an aqueous phase (including both suspended and attached biomass) called biofilm and into biomass located close to the gas-water interface forming a coherent structure this group was referred to as neuston biofilm (Panfilov et al., 2012). For the continuum scale, on which Hagemann developed his model, the decision was made to treat microbial populations without considering different structures that they can form (chosen approach is called the unstructured approach). Leading to only one general structure of microbes at the end. The assumed structure is considered as a penetrable, volumeless component that is not affecting the porosity or permeability of the reservoir no matter its density. Microbes are considered to be located in the aqueous phase therefore only relevant nutrient concentration is the one dissolved in the water.

In addition, microbial movement is present as well, including attachment and detachment by which they can interchange between mentioned occurring structures (when talking about a structured approach), this movement is diffusion-like in the sense that it can be perceived as random. Also, chemotaxis is present which is the tendency of microbes to go where nutrient (substrate) concentration is high. For the continuum scale advective transport of microbes and chemotaxis is neglected (Hagemann, 2017; Hagemann, Panfilov, Ganzer, 2016; Hagemann, Rasoulzadeh et al., 2016). Resulting microbial population dynamics equation:

$$\frac{dn}{dt} = S_l \psi^{growth}\left(c_l^{H_2}, c_l^{CO_2}\right) * n - \psi^{decay} * n + \nabla(D_m \nabla n) \tag{3.6}$$

ψ^{growth}… microbial growth function $\left[\frac{1}{s}\right]$. It depends on hydrogen (substrate) concentration ($c_l^{H_2}$) and carbon dioxide (electron acceptor) concentration ($c_l^{CO_2}$) in liquid phase;

ψ^{decay}… microbial decay function $\left[\frac{1}{s}\right]$;

n… microbial density $\left[\frac{1}{m^3}\right]$;

D_m… microbial diffusion coefficient $\left[\frac{m^2}{s}\right]$.

When it comes to methanogenic reactions, as was mentioned in section 2.2, hydrogen is used as a source of energy (hydrogenotrophy) and carbon dioxide as a carbon source (autotrophy). The rate at which biochemical reaction (for energy uptake) is carried out is proportional to microbial growth. Therefore, the source/sink term in equation (3.1) takes the following form:

$$q^k = \phi S_l \Upsilon^k \frac{\psi^{growth}}{Y} n \tag{3.7}$$

Υ… stoichiometric coefficient depending on the reaction. For the methanogenesis (2.9) it is:

$$\Upsilon = \begin{pmatrix} -1 \\ -4 \\ 1 \\ 2 \end{pmatrix}, \text{ representing } CO_2, H_2, CH_4, H_2O;$$

Y… yield coefficient, relating microbial growth to the consumption of hydrogen $\left[\frac{1}{mol(H_2)}\right]$.

3.2.1 Growth and Decay Functions

In section 2.3 growth function of a batch experiment was plotted and explained. Monod (Monod, 1949) developed a model using such an experiment that describes the growth function of a microbial population. In regards to UHS conditions where substrate as well as electron acceptor availability is to be taken into account, Hagemann has used the extended Monod model – "double Monod model". Microbial growth function for the methanogenic reaction:

$$\psi^{growth} = \psi_{max}^{growth} \left(\frac{c_l^{H_2}}{\alpha_{H_2} + c_l^{H_2}} \right) \left(\frac{c_l^{CO_2}}{\alpha_{CO_2} + c_l^{CO_2}} \right) \tag{3.8}$$

ψ_{max}^{growth}… maximum specific rate of microbial growth $\left[\frac{1}{s}\right]$;

α… half-velocity constant $\left[\frac{mol}{mol}\right]$. In his work, Monod defined this as a concentration of a nutrient at which the rate of growth is half of the maximum growth rate (ψ_{max}^{growth}) (Monod, 1949).

The model is not able to present lag and stationary phases (*Figure 4*). It starts with the exponential growth phase and, if the conditions are such, it goes into the death phase. Mentioned competition between different bacterial species for the nutrients (section 2.4) is also not covered by the growth model, but seeing how our work is focused only on methanogenic archaea this does not present a problem.

With the decay rate it all comes down to two options, either having a constant decay rate:

$$\psi^{decay} = b \tag{3.9}$$

b… decay coefficient $\left[\frac{1}{s}\right]$.

Or a decay rate that is linearly dependent on microbial density:

$$\psi^{decay} = b * n \tag{3.10}$$

3.3 Final Governing Equations

After analyzing necessary equations that are covering flow and transport, as well as microbial population growth due to biochemical reactions, the resulting mathematical model for methanogenic archaea is composed of two differential equations:

1. Microbial population dynamics:

$$\frac{dn}{dt} = S_l \left[\psi_{max}^{growth} \left(\frac{c_l^{H_2}}{\alpha_{H_2} + c_l^{H_2}} \right) \left(\frac{c_l^{CO_2}}{\alpha_{CO_2} + c_l^{CO_2}} \right) \right] * n - b * n + \nabla(D_m \nabla n) \tag{3.11}$$

Where the first term on the right side is *microbial growth in the liquid (aqueous) phase*, followed by *the microbial decay* term and the final term that is representing *microbial diffusion-like movement*.

***2. Reactive transport of nutrients (mobile components)**:*

$$\begin{aligned} &\frac{\partial\left(\hat{\rho}_v c_v^k S_v + \hat{\rho}_l c_l^k S_l\right)}{\partial t} \\ &+\nabla\left[-\hat{\rho}_v c_v^k \frac{K k_{rv}}{\mu_v}(\nabla P_v - \rho_v g) - \hat{\rho}_l c_l^k \frac{K k_{rl}}{\mu_l}(\nabla P_l - \rho_l g)\right] \\ &+\nabla\left[-\rho_v\left(D_{diff,v}^k + D_{disp,v}^k\right)\nabla c_v^k - \rho_l\left(D_{diff,l}^k + D_{disp,l}^k\right)\nabla c_l^k\right] \\ &= \phi S_l \Upsilon^k \frac{\psi^{growth}}{Y} n \\ &k = CO_2, H_2, CH_4, H_2O \end{aligned} \tag{3.12}$$

Where the second and third terms represent *advective* and *dispersive/diffusive* transport, while *the source/sink* term is on the right side.

3.4 Parameters

Besides developing the mathematical model Hagemann also collected available data from the literature regarding parameters used in the equations. These are resulting from different laboratory batch experiments on the basis of the Monod model, and are presented in *Table 3* and *Table 4* with the averages at the bottom. Initial values for the simulation runs are to be in the range of the table values.

Table 3: Parameters' values for the mathematical model's equations (Hagemann, 2017)

	$\psi_{max}^{growth}\left[\frac{1}{s}\right]$	$\alpha_{H_2}\left[\frac{mol}{mol}\right]$	$\alpha_{CO_2}\left[\frac{mol}{mol}\right]$	$Y\left[\frac{1}{mol(H_2)}\right]$	$b\left[\frac{1}{s}\right]$
	2.488×10^{-5}	3.240×10^{-7}	5.400×10^{-6}	6.875×10^{11}	1.019×10^{-6}
	1.273×10^{-5}	1.179×10^{-7}	4.140×10^{-6}	4.584×10^{11}	6.944×10^{-7}
	8.912×10^{-6}	1.170×10^{-7}	2.340×10^{-8}	2.483×10^{11}	
	1.447×10^{-5}	1.179×10^{-7}		8.403×10^{11}	
	1.433×10^{-5}	1.188×10^{-7}		1.222×10^{11}	
	1.447×10^{-5}	1.188×10^{-7}		3.056×10^{11}	
	1.528×10^{-5}	8.999×10^{-8}		7.257×10^{11}	
	1.736×10^{-5}	8.999×10^{-8}		1.451×10^{12}	
	3.009×10^{-5}	2.160×10^{-11}		1.093×10^{13}	
	2.373×10^{-5}			3.820×10^{11}	
	1.852×10^{-5}			2.925×10^{11}	
	1.794×10^{-5}			7.697×10^{10}	
	1.472×10^{-5}				
	7.523×10^{-6}				
	2.546×10^{-5}				
	1.505×10^{-6}				
	1.736×10^{-5}				
Mean	1.643×10^{-5}	1.094×10^{-7}	3.188×10^{-6}	1.376×10^{12}	8.565×10^{-7}

Table 4: Methanogenic archaea density in subsurface systems (Hagemann, 2017)

	Depth $[m]$	$n\left[\frac{1}{m^3}\right]$
	330	5.600×10^{13}
	376	4.800×10^{12}
	68–446	3.600×10^{9}
	200-1800	4.400×10^{12}
	200–1800	7.800×10^{12}
	200–1800	1.120×10^{13}
	1264–1742	2.000×10^{8}
	129–1240	1.001×10^{11}
	14–182	1.000×10^{13}
	647	2.000×10^{11}
Mean		9.450×10^{12}

Chapter 4

MRST Implementation

Window of possible simulating tools available for the exploration of UHS processes consists of the commercial software used for the petroleum industry as well as the scientific software and open-source codes. None are able to simulate UHS processes in their original form, without previous adjustments. That is why an option which is allowing access to the source code is preferred (Hagemann, 2017).

Majority of numerical simulations, regarding UHS processes, that have been done so far and are available in literature used DuMu^{X}. Open-source software for scientific purposes, used for flow and transport simulations in porous media. It was developed for different hydrological applications, and one can adjust it relatively easy for the bio-reactive modeling needed for UHS simulations (Flemisch et al., 2011).

In this work, however, for the simulation of methanogenic archaea's metabolic reaction, we are using MRST (MATLAB Reservoir Simulation Toolbox), an open-source library originally developed in order to provide solutions to daily reservoir simulation problems. It was developed by SINTEF and their Computational Geosciences group in the Department of Mathematics and Cybernetics. The library contains core modules for basic data structures and functionality but also a big set of additional modules offering discretization, solvers, physical models and a large range of simulators and workflow tools. Some of these are third-party modules developed by researchers from Heriot-Watt University, NTNU, University of Bergen, TNO, and TU Delft. This code is broadly used for research purposes to investigate new reservoir simulation theories, and since its initial development, it surpassed usage in just the petroleum industry.

MATLAB is one of, if not the most popular *tool* for applied numerical analysis. MRST is also more of a research *tool* than just a simulator for porous media flow processes. As such, it opens up opportunities for the prototyping of new user-made modeling and simulation utilities. This is possible due to the wide range of provided data structures and computational methods which can be modified (LIE, 2019; SINTEF, 2022).

With that in mind, this work is set out to do the same, namely, MRST's *compositional module* will be used (since the black-oil module is more suitable for flow modeling in oil and/or gas reservoirs containing up to three phases). This compositional module will be adjusted for the modeling of flow and transport as well as bio-reactive processes in UHS. As far as it comes to available works in the literature, this will be the first attempt to adapt MRST for the UHS simulations, which could provide yet another option for such simulations, diversifying the UHS numerical simulation approaches.

4.1 Fluid Model Classes

The compositional module of the MRST contains different model classes which are implemented in the module (*Table 5*).

Table 5: Model classes implemented in the compositional module of MRST (LIE and Møyner, 2021)

Group	Class name	Formulation	Note
I	*ThreePhaseCompositionalModel*	—	Virtual base class
	OverallCompositionCompositionalModel	Overall composition	
	NaturalVariablesCompositionalModel	Natural variables	
II	*GenericOverallCompositionModel*	Overall composition	Generic model
	GenericNaturalVariablesModel	Natural variables	Generic model

The first three classes compose a group that was developed using principles set out for the black-oil module. These models are assuming that by default two hydrocarbon phases (vapor and liquid) representing multicomponent phases and an aqueous phase representing immiscible phase are always present. Scenarios, where any of these are not present, are treated as special cases, the gas or the aqueous phase can be excluded but the problem occurs when the oleic phase is not present.

This is exactly why models from *group I* are not useful for this work, in which the UHS conditions that are to be simulated consist of two phases, *water* which will be our liquid phase and *gas* which will be our vapor phase. Therefore, the default conditions that are expected from the models of this group are not fulfillable.

Using the black-oil module's code as a basis for the development of the different compositional model classes turned out not to be an optimal solution resulting in complications when adding new features. This resulted in the development of the second group of compositional model classes called *generic model classes*. The generic approach has no default expectations that should be fulfilled, it constructs the setup of the fluid model as a collection of individual components, where each may belong to certain predefined categories exhibiting specific behavior.

In conclusion, the initially developed compositional model formulations have inherited principles from the black-oil model and therefore are requiring the oil phase in the fluid to work, but the oleic phase is not present for the planned simulation. The generic formulation, on the other hand, was developed later on with the goal of overcoming more complicated scenarios that are coming together with the compositional model approach. This formulation is providing the option of not having an oil phase in the fluid model (LIE and Møyner, 2021).

Normally, the aqueous phase is treated as immiscible and hydrocarbon liquid and vapor phases are governed by an equation of state (EOS). In this work, the oil phase is disabled while the water phase is set as the liquid phase, and the gas phase remained as the vapor phase. In general, the compositional model is relying on the equation of state in order to compute equilibrium compositions and densities of the individual components that are a part of the fluid system.

As can be seen from *Table 5* there are two different formulations that are implemented in the compositional models:

- *The overall composition formulation*, which is using *pressure* and *overall mole fractions* as primary variables.
- *The natural variable formulation,* which is using *phase saturations* and *phase mole fractions* as primary variables.

Mentioned primary variables are used as a basis for solving a system of equations in the compositional model scenario. A two-part system of equations is composed of *the N conservation equations* for *each component in the system* and *the N isofugacity or K-value constraints* for *cells in which phases are present*. The conservation equations are time-dependent and cell residuals depend on the neighboring cells' fluxes. K-values are a result of the EOS flash calculations and are local for each cell.

There are different strategies for solving the system of equations. For each, governing equations are the same as well as the converged solution but the number of iterations before the solution is reached can be much different depending on the strategy. The solution-reaching strategy will significantly differ depending on whether natural variables (in *natural variable formulation*) or overall composition variables (in *overall composition formulation*) are used.

Each component of the fluid system can exist in both *liquid* and *vapor* phases. The *mole fraction* of component *i* in the *liquid phase* is denoted as x_i and in the *vapor phase* as y_i. *The overall mole fraction* of component *i* is denoted as z_i. The liquid *phase mole fraction* of the fluid system is denoted as L while the *vapor phase mole fraction* is denoted as V.

$$x_i L + y_i V = z_i$$

$$\sum_i^N x_i = 1, \quad \sum_i^N y_i = 1, \quad L + V = 1$$

The overall composition formulation uses *pressure (p)* and *N-1 overall mole fractions (z_i)* as primary variables. It assumes that the *flash equations* are more difficult to converge than the *flow equations*. Therefore, flash equations are solved first as a nested nonlinear system using primary variables (p, z_i). Resulting flash outputs are made up of vapor and liquid phase mole fractions, these are used for the determination of saturations. Next, flow equations, in terms of pressure and overall molar compositions, are assembled and solved separately. If they converge simultaneous convergence of both *conservation* and *flash equations* is achieved, and the next time-step calculation initiates.

The natural variable formulation's primary variables consist of saturations and components' phase fractions (x_i, y_i) which are grouped in a natural primary variable *set* together with pressure (unknowns consist out of p, S_i, x_i and y_i):

$$\eta_N = (p, x_1, \ldots, x_{N-1}, S_l, y_1, \ldots, y_{N-1})$$

Contrary to the overall composition formulation, here *flash equations* and *flow equations* are gathered in one system of equations and are solved simultaneously for the unknowns. The natural variable formulation showed better convergence for immiscible displacement cases (LIE and Møyner, 2021).

Compatible for this work is the second group of compositional model classes and from there the *GenericOverallCompositionModel*. *The Natural variables model* had convergence problems where it could not find the solution during iterations.

In the *Appendix B* (*Fluid Model Implementation*) section of this work, part of the code is presented where the implementation of the fluid model is carried out.

4.2 Discretization and Solver

The standard discretization method in MRST is the finite volume method (FVM) and for the solver, MRST's `NonLinearSolver` function is used. This solver is based on the Newton-Raphson method, which together with FVM is described in *Appendix A*. MRST also uses the *Fully Implicit Method* solving scheme (LIE, 2019; LIE and Møyner, 2021).

Therefore, governing partial derivative equations are discretized, linearized and solved providing resulting pressure, saturation, temperature, components' mole fractions as well as phase fractions and phase densities for each time step throughout the grids.

4.3 Microbial Effects Configuration

As stated in *Chapter 3*, the desired process can be described using differential equations for *microbial population dynamics* (3.11) and *reactive transport of nutrients (mobile components)* (3.12). In this study following assumptions and strategies were made:

- Microbes are assumed as immobile (ignoring the diffusion term in (3.11)).
- Equation (3.12) served as a basis for the derivation of a new – *involved in reaction nutrient's molar fraction* (4.2) equation. This has been done in order to enable the combination of results from the MRST's solver with the effects of the methanogenic reaction.

Resulting in:

$$\frac{dn}{dt} = S_l \left[\psi_{max}^{growth} \left(\frac{c_l^{H_2}}{\alpha_{H_2} + c_l^{H_2}} \right) \left(\frac{c_l^{CO_2}}{\alpha_{CO_2} + c_l^{CO_2}} \right) \right] * n - b * n \tag{4.1}$$

$$\Delta c_{r,l}^{k} = \Upsilon^{k} * \frac{\psi^{growth}}{Y * \hat{\rho}_l} * n * \Delta t, \quad k = CO_2, H_2 \tag{4.2}$$

In order to implement the equations, the *sequential explicit approach* is carried out. The approach consists of taking the solver's results for each time-step and updating them by the additional calculations coming from the two equations listed above. In the following detailed description of the entire process is present.

How the approach looks in the code can be seen in the extracted section presented in *Appendix B* (*Microbial Effects*).

As mentioned in the previous chapters, microbes are located in the liquid (aqueous) phase, in it therefore nutrient consumption occurs. Amounts of available nutrients are presented in the form of liquid phase molar fractions of each. The solubility of nutrients is calculated by the implementation of *Henry's law*. Which states that the solubility of a gas in a liquid is proportional to its pressure over the solution (*Figure 6*) (Mills and Coimbra, 2015):

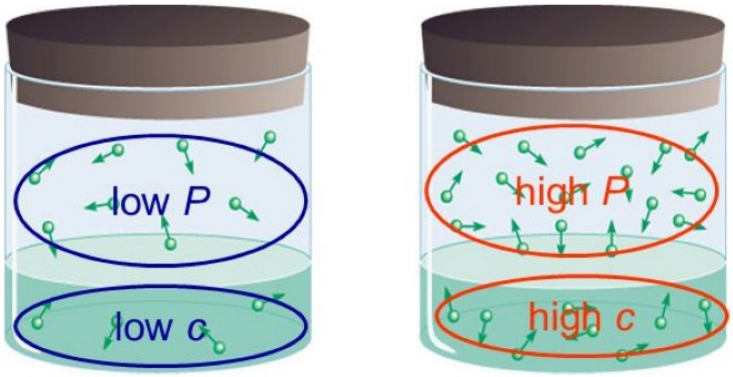

Figure 6: Representation of Henry's law (Azizmohammadi, 2018)

$$p_k = c_l^k * H_k \quad \rightarrow \quad c_l^k = \frac{p_k}{H_k}, \quad k = CO_2, H_2 \tag{4.3}$$

p_k… partial pressure of component *k* above the solution;

c_l^k… molar fraction of component *k* in solution (liquid phase);

H_k… Henry constant of component *k*.

Where:

$$p_k = p * c_v^k \tag{4.4}$$

p… total pressure distribution gained from the solver;

c_v^k… molar fraction of component *k* above solution (in vapor phase).

Values for the Henry constant are provided in *"Basic heat and mass transfer"* by A. F. Mills and C. F. M. Coimbra (*Table 6*).

Table 6: Henry constants for different gases in aqueous solutions (in [bar]) (Mills and Coimbra, 2015)

Solute	290 K	300 K	310 K	320 K	330 K	340 K
H_2S	440	560	700	830	980	1,140
CO_2	1.280	1,710	2,170	2,720	3,220	-
O_2	38,000	45,000	52,000	57,000	61,000	65,000
H_2	67,000	72,000	75,000	76,000	77,000	76,000
CO	51,000	60,000	67,000	74,000	80,000	84,000
Air	62,000	74,000	84,000	92,000	99,000	104,000
N_2	76,000	89,000	101,000	110,000	118,000	124,000

4.3.1 Change in Microbial Density

For *microbial population dynamics* equation (4.1), the microbial density "*n*" of each grid cell is calculated as the number of microbes in the cell divided by the cell's total volume. Next, microbial growth using the double Monod model (equation (3.8)) is calculated where the model is fed with water phase molar fractions of hydrogen and carbon dioxide gained from the implemented *Henry's law*. Derivatives are then replaced by differential formulations in which the time difference has a fixed value of the time-step and the difference in microbial density is equal to the current density value (one being calculated) minus the previous time-step's microbial density (explicit scheme).

The described calculation process is then carried out for each grid cell (i) and time-step (t) as follows:

$$\frac{dn}{dt} = S_l \psi^{growth}\left(c_l^{H_2}, c_l^{CO_2}\right) * n - \psi^{decay} * n$$

$$\frac{dn}{dt} = S_l \left[\psi_{max}^{growth}\left(\frac{c_l^{H_2}}{\alpha_{H_2} + c_l^{H_2}}\right)\left(\frac{c_l^{CO_2}}{\alpha_{CO_2} + c_l^{CO_2}}\right)\right] * n - b * n$$

replacing the calculated value of the equation's right side with "A_1" and derivatives with differential formulations:

$$n_i^t - n_i^{t-1} = A_1 * \Delta t \tag{4.5}$$

$$n_i^t = n_i^{t-1} + A_1 * \Delta t$$

4.3.2 Reaction-induced Compositional Changes

Seeing how what the solver is providing is a result of just an advective transport, the changes caused by the reaction must be calculated as well. With this in mind, we derived a new equation (based on equation (3.12)), that has the goal of providing the *reaction* mole fractions of nutrients in the liquid phase (because this is where the reaction is occurring).

Starting with equation (3.12):

$$\phi \frac{\partial\left(\hat{\rho}_v c_v^k S_v + \hat{\rho}_l c_l^k S_l\right)}{\partial t}$$

$$+\nabla.\left[-\hat{\rho}_v c_v^k \frac{K k_{rv}}{\mu_v}\left(\nabla P_v - \rho_v g\right) - \hat{\rho}_l c_l^k \frac{K k_{rl}}{\mu_l}\left(\nabla P_l - \rho_l g\right)\right]$$

$$+\nabla.\left[-\rho_v\left(D_{diff,v}^k + D_{disp,v}^k\right)\nabla c_v^k - \rho_l\left(D_{diff,l}^k + D_{disp,l}^k\right)\nabla c_l^k\right]$$

$$= \phi S_l \Upsilon^k \frac{\psi^{growth}}{Y} n$$

- From the first term on the left, the vapor phase mole fraction is cut.
- Diffusive and dispersive transport of the chemical component is neglected.
- The advective transport effects are handled by the MRST's solver, therefore the advective term is excluded.

Resulting in:

$$\phi \frac{\partial\left(\hat{\rho}_l c_l^k S_l\right)}{\partial t} = \phi S_l \Upsilon^k \frac{\psi^{growth}}{Y} n, \quad k = \mathrm{CO_2}, \mathrm{H_2} \tag{4.6}$$

What we are left with essentially is the change in component liquid phase molar fraction *only due to the reaction*. Considering that and replacing derivatives with differential formulations:

$$\phi \frac{\hat{\rho}_l S_l \Delta c_{r,l}^k}{\Delta t} = \phi S_l \Upsilon^k \frac{\psi^{growth}}{Y} n, \quad k = \mathrm{CO_2}, \mathrm{H_2} \tag{4.7}$$

where:

$$\Delta c_{r,l}^k = c_{pre-reaction,l}^k - c_{post-reaction,l}^k \tag{4.8}$$

Seeing how the $\Delta c_{r,l}^k$ is the difference between the state of component's molar fraction in the liquid phase before and after the reaction, the resulting value is providing exactly what is needed – the nutrient's molar fraction that is involved in the reaction. This calculation is then carried out for each time-step.

Porosities and saturations from each side of equation (4.7) cancel each other out. With that, equation (4.2), based on carbon dioxide, is having the following look:

$$\Delta c_{r,l}^{CO_2} = \Upsilon^{CO_2} * \frac{\psi^{growth}}{Y * \hat{\rho}_l} * n * \Delta t \tag{4.9}$$

The molar density of the liquid phase is calculated by dividing the density of the liquid phase (resulting from the solver) by the sum of the molar masses of dissolved components, each multiplied by its liquid phase molar fraction.

$$\hat{\rho}_l = \frac{\rho_l}{\sum M_k * c_l^k}, \quad k = \mathrm{CO_2}, \mathrm{H_2}, \mathrm{CH_4}, \mathrm{H_2O} \tag{4.10}$$

Provided in *Chapter 3*, below equation (3.7) were the stoichiometric coefficients (Υ) of components derived from the methanogenesis reaction:

$$\mathrm{CO_2} + 4\mathrm{H_2} \rightarrow \mathrm{CH_4} + 2\mathrm{H_2O}$$

In that mathematical model, Υ for CO_2 was -1 since it is a consumed component in the reaction. This is then used for the reaction term calculation in equation (3.12), once summed up with the advection and diffusion/dispersion terms they would give the *total* molar fraction *changes* of components. But seeing how for equation (4.2) we are only looking for the component *reaction* molar fraction for *each time-step*, a negative sign would not work for Υ_{CO_2}. Therefore, in order to get correct values Υ for CO_2 was set to 1.

Once the liquid phase molar fraction of *CO_2* used in the reaction is gained (Eq. (4.9) it is possible to calculate the actual moles of each component that is consumed or produced in the reaction.

This calculation is done on the basis of *1 mol* due to which molar fractions of both liquid (L) and vapor phase (V), provided by the solver, are actually considered as the amounts of total moles in each phase (n_L, n_V).

First, total moles present in the liquid phase are multiplied with the calculated $\Delta c_{r,l}^{CO_2}$. Resulting in moles of CO_2 consumed in the reaction:

$$n_r^{CO_2} = \Delta c_{r,l}^{CO_2} * n_L \tag{4.11}$$

n… amount of substance [mol], not to be confused with microbial density $\left[\frac{1}{m^3}\right]$ form before;

n_L … total moles in the liquid phase;

$n_r^{CO_2}$… moles of CO_2 involved in the reaction;

$\Delta c_{r,l}^{CO_2}$… CO_2 molar fraction that is involved in the reaction (as a change of CO_2 molar fraction in the liquid phase *before* and *after* the reaction).

Using calculated moles of CO_2 that are consumed in the reaction (4.11) and the stoichiometry of the methanogenesis, moles of the remaining components are gained:

$$n_r^{H_2} = 4 * n_r^{CO_2}, \quad n_r^{CH_4} = n_r^{CO_2}, \quad n_r^{H_2O} = 2 * n_r^{CO_2} \tag{4.12}$$

4.3.3 Combining the Processes

In order to gain a complete picture, a connection must be made between the effects of the advection (injection) as well the ones caused by the methanogenic reaction. In other words, a connection between the results provided by the MRST's solver and the results of our newly derived equation (4.2.

The approach taken can be explained by presenting some ground ideas and from them resulting calculations.

Reaction nutrients (CO_2 and H_2):

- The vapor phase serves as a source for the nutrients that are dissolved in the liquid phase.
- The liquid phase serves as a habitat for the microbes and therefore is the phase in which methanogenic reaction occurs.
- As the reaction is occurring dissolved nutrients are consumed from the liquid phase.
- As the nutrients are consumed, they are replaced by new ones from the vapor phase.
- Nutrient losses per each time-step due to the methanogenic reaction are at the end observed in the vapor phase.
- To present these as a vapor phase molar fraction, moles of each nutrient consumed in the reaction (Eq. (4.11 & (4.12) are used:

$$C^{k}_{reaction,v} = \frac{n^{k}_{r}}{n_V}, \quad k = CO_2, H_2. \tag{4.13}$$

n^{k}_{r}… moles of component "k" involved in the reaction;

n_V … total moles in the vapor phase.

Reaction products (CH_4 and H_2O):

- As the reaction occurs methane and water are produced in the liquid phase.
- The very low solubility of methane in water was assumed to be zero.
- All of the produced CH_4 in the methanogenesis migrates to the vapor phase:

$$C^{CH_4}_{reaction,v} = \frac{n_r^{CH_4}}{n_V}. \tag{4.14}$$

- All of the produced H_2O remains in the liquid phase:

$$C^{H_2O}_{reaction,l} = \frac{n_r^{H_2O}}{n_L}. \tag{4.15}$$

Vapor phase molar fraction changes:

- *Hydrogen and Carbon dioxide:*
  - Each time-step (t) of the simulation **starts** with the set relation between injected nutrients' molar fractions in the vapor phase – $\left(C_{start,v}^{CO_2,H_2}\right)^t$.
  - **During** each time-step these fractions change due to advection – $\left(C_{solver,v}^{CO_2,H_2}\right)^t$ and reaction – $\left(C_{reaction,v}^{CO_2,H_2}\right)^t$.
  - Each time-step **ends** with new molar fractions in the vapor phase of the nutrients:

$$\left(C_{end,v}^{CO_2,H_2}\right)^t = \left(C_{solver,v}^{CO_2,H_2}\right)^t - \left(C_{reaction,v}^{CO_2,H_2}\right)^t. \tag{4.16}$$

  - **Ending** values of a **previous** time-step serve as **starting** values for a **new** time-step:

$$\left(C_{start,v}^{CO_2,H_2}\right)^t = \left(C_{end,v}^{CO_2,H_2}\right)^{t-1}$$

    - The only exception to this is the initial time-step. Since the injection starts with this time-step no reaction occurred before it. The initial time-step *starts* with the values affected *only* by the advection.
- *Methane:*
  - The only source of CH_4 is the methanogenic reaction.
  - There are no advection-caused changes to the vapor phase molar fraction as there were for the injected CO_2 and H_2 molecules.

$$\left(C_{end,v}^{CH_4}\right)^t = \left(C_{pre-reaction,v}^{CH_4}\right)^t + \left(C_{reaction,v}^{CH_4}\right)^t \tag{4.17}$$

Liquid phase molar fraction:

- *Of the nutrients:*
  - At the **start** of each time-step implementation of the mentioned *Henry's law* occurs.
  - Resulting are the molar fractions of the nutrients in the liquid phase (Eq. (4.3), available for the reaction carried out **during** the time-step.
  - Used in Henry's law calculation (Eq. (4.4) are the described **starting** vapor phase molar fractions – $\left(C_{start,v}^{CO_2,H_2}\right)^t$. These are containing changes due to both advection and the reaction that have occurred in the **previous** time-step – $\left(C_{start,v}^{CO_2,H_2}\right)^t = \left(C_{end,v}^{CO_2,H_2}\right)^{t-1}$.
  - Therefore, the amount of available nutrients for each **new** time-step is dictated by the changes occurring in each **previous** time-step.

- The Exception again is with the initial time-step, since no reaction occurs before this time step, it **starts** by using the values provided by the solver.

- *Of the produced water molecules*:
  - Water molecules' increase in the liquid phase is as follows:

$$\left(C_{end,l}^{H_2O}\right)^t = \left(C_{solver,l}^{H_2O}\right)^t + \left(C_{reacton,l}^{H_2O}\right)^t \tag{4.18}$$

Chapter 5

Case Studies

Following are different simulation runs, all replicating laboratory-planned experiments. Dimensions are of a core scale, samples are pre-populated by a certain number of methanogenic archaea and injection of the nutrient mixture is introduced for the duration of the simulation.

The scenario is a one-dimensional (1-D) injection simulation, due to which the resulting distribution of microbial density throughout the sample and in different time-steps is to be shown, as well as the production of methane from the archaea's metabolic reaction.

5.1 Base Case

Domain geometry and grid

Core sample dimensions are 25 cm in the x direction and 5 cm in both y and z directions, resulting in a grid system of 25 grids in x and 1 grid in both y and z directions.

Figure 7 shows a homogenous medium, with constant 20% *porosity* and *permeability* of 250 millidarcy throughout the sample.

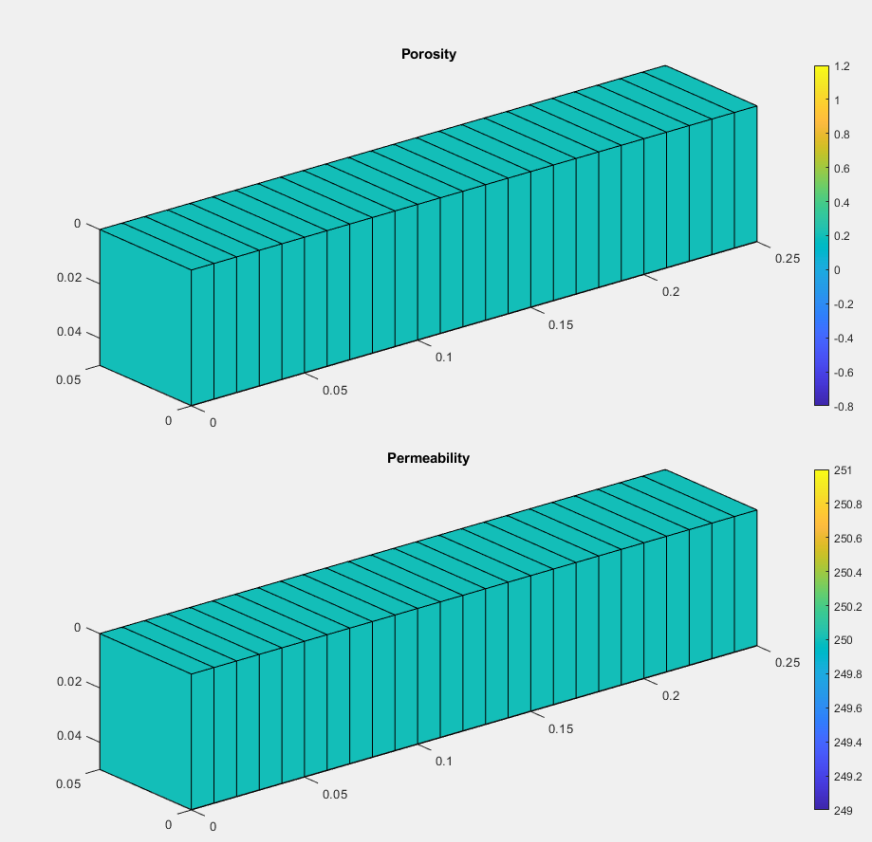

Figure 7: Grid system of a sample with assigned Φ and k

Compositional Fluid Model

As mentioned in *Chapter 4* for this work compositional module of MRST was used, and from there the *GenericOverallCompositionModel* (*Table 5*). Necessary inputs for the model contain:

- Geometry containing rock properties (ϕ and k).
- Description of the flowing fluid's phases, including the number of phases (for our work it is a water-gas system), densities, viscosities and compressibility of both phases.
- List of components for the compositional model present in different fluid phases. This simulation contains initially just water (H_2O component) and nitrogen (N_2) which serves as cushion gas. Once the injection starts added are hydrogen (H_2) and carbon dioxide (CO_2) as well as produced methane (CH_4). Nevertheless, all of them are added to the list and fed to the input argument of the model.

  MRST's function `TableCompostionalMixture` offers the possibility to initiate a multiple-component mixture with each species called by name from stored tables formed from *CoolProp* library. *CoolProp* library represents an open-source database of fluid and humid air properties (Ian H. Bell and the CoolProp Team, 2020).

- As explained in *Chapter 4* model is expecting a three-phase fluid by default (having oil as *the liquid phase* gas as *the vapor phase* and water belonging to the third, so-called, *immiscible phase*). Therefore, as a final input, the default setup is changed in order to serve our scenario. Water is assigned to be *the liquid phase*, while gas remains as *the vapor phase*. The oleic phase is completely removed from the setup.

The setting of the fluid model code is present in *Appendix B* (*Fluid Model Implementation*).

Initial State

In order to formulate an initial state, function `initCompostionalState` is used, having the following inputs:

- Pressure is set to 40 bar (which is the pressure that the cushion gas is maintaining before the injection starts).
- The temperature is 50 °C (323.15 K).
- Equation of State model, which is contained in the output of the used *GenericOverallCompositionModel*.
- Initial fluid composition, containing *overall mole fractions* for each component (z_i).
- When it comes to the saturation calculations (S_l and S_v) they are calculated internally using (LIE and Møyner, 2021):

$$S_l = \frac{Z_l * L}{Z_l * L + Z_v * (1 - L)} \tag{5.1}$$

L… liquid phase molar fraction;

Z… compressibility factor gained as a result of cubic EOS.

As can be seen, saturation is dependent on the molar fractions of the fluid, for this base case, different compositions were set resulting in different saturations, in order to compare the results and find the optimal one. These scenarios can be observed in the *Results* section below.

The resulting initial state is provided in a *structure array (struct)* data type, which groups related data using data containers called *fields*, each field can contain any type of data (MathWorks and Inc., 2022).

Additionally, added to the resulting initial state struct is an initial microbial distribution throughout the grid cells, for this case the same number of microbes is distributed throughout each grid cell.

Boundary Conditions

Both left and right boundaries are controlled by the rates. The left boundary is defined by an injection rate resulting from dividing the pore volumes of the whole sample (sum of pore volumes of each grid block) by the total time of the injection. The right boundary is defined by the production rate which was set equal to the injection rate. Injected gas consists of 80% hydrogen and 20% of carbon dioxide.

Different injection rates were tested in order to compare their effects to the results as well as to find the optimum one, these can be seen below in the *Results* section.

Microbial Equations' Parameters

For the implemented equations (4.1) and (4.2), the following parameters were used:

Table 7: Equations' parameters

$\psi_{max}^{growth}\left[\frac{1}{s}\right]$	$1.643 * 10^{-5}$
$b\left[\frac{1}{s}\right]$	$2.3 * 10^{-6}$
$\alpha_1\left[\frac{mol}{mol}\right]$	$1.1 * 10^{-7}$
$\alpha_2\left[\frac{mol}{mol}\right]$	$3.2 * 10^{-6}$
$Y\left[\frac{1}{mol(H_2)}\right]$	$1.376 * 10^{12}$
Υ_{CO_2}	1
Υ_{H_2}	4
Υ_{CH_4}	1
Υ_{H_2O}	2

All values are in the range of provided literature values presented in *Chapter 3* (*Table 3* & *Table 4*).

Regarding the initial number of microbes, for the base case, it was set to 10, resulting in $n_{initial} = 4 * 10^5\left[\frac{1}{m^3}\right]$. This was done in order to easily observe the changes through the time and space (grid cells) as well as changes due to different water saturations and injection rates.

5.1.1 Injection Rate Analysis

Analyzing different rates, it was found that the 50 PV injection results in the highest average number of microbes throughout the sample. This injection rate was taken as the base one, to which all other tested rates were compared. Each injection lasted for 10 days, which for the optimum injection of 50 PV can be converted into the rate of $2.6 * 10^{-5} \left[\frac{m^3}{h}\right]$ or $26 \left[\frac{cm^3}{h}\right]$.

Tested were two rates higher than 50 PV, the first one being double of it – 100 PV and the second one being 20 times bigger – 1000 PV. What was concluded is that, as the rates go up, injected gas mixture is flushing the liquid phase present in the sample, therefore removing the possibility for microbial life. As can be seen from the figures trend of the flushed liquid phase (*Figure 9*) reflects the microbial population trend (*Figure 8*).

Presented figures are of the last 10th day of the injection, and the initial liquid saturation is 20%.

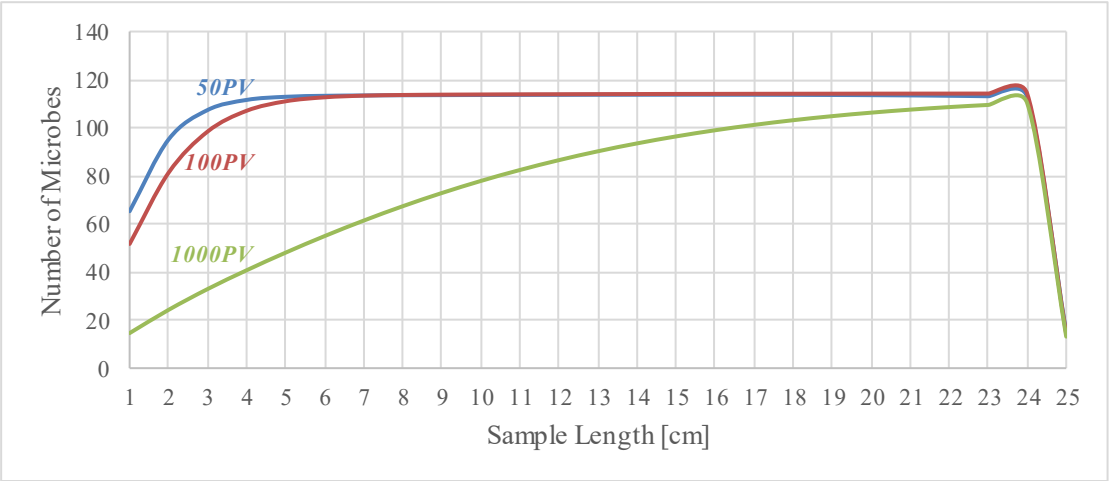

Figure 8: Above 50PV – number of microbes throughout the core sample on the last day of the injection

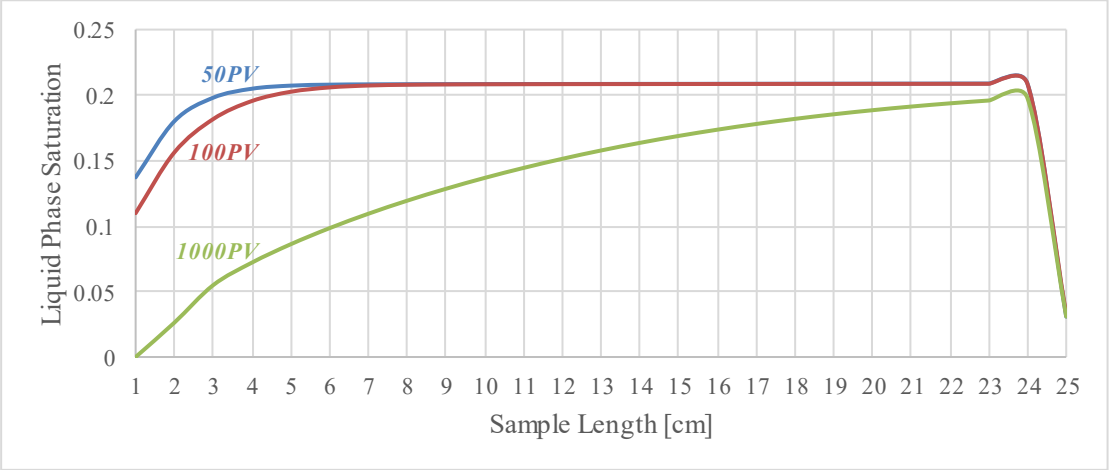

Figure 9: Above 50PV liquid phase gets flushed

Also tested were three rates lower than the 50 PV, namely 1 PV, 10 PV and 25 PV. What occurs here is that if the injection rate is too small the necessary nutrients do not reach the far parts of the core sample. As a result, the microbes' population gets high at the beginning of the sample but further down the length less and less growth is observed.

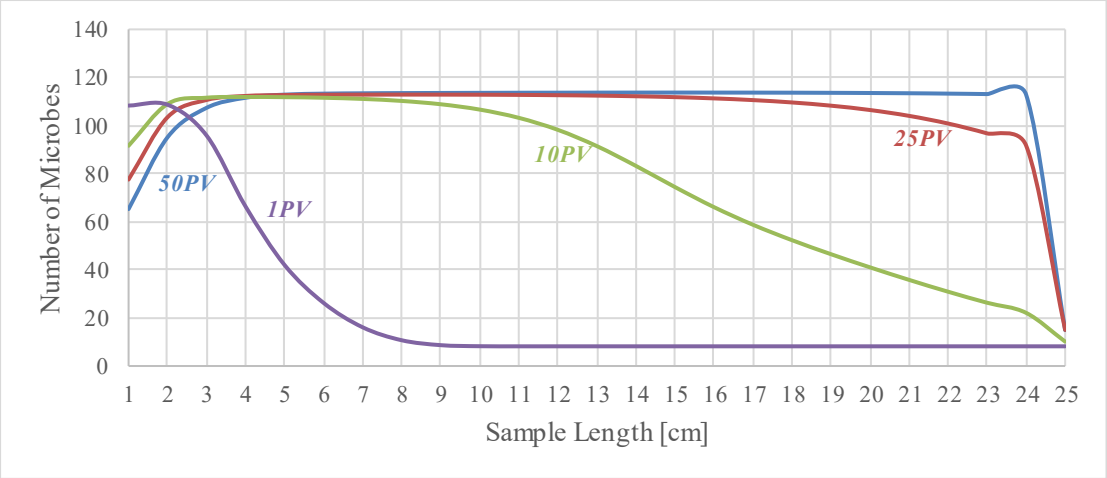

Figure 10: Below 50PV – number of microbes throughout the core sample on the last day of the injection

As a summary, *Table 8* presents an average number of microbes throughout the sample at different rates. As noted at the beginning, after 10 days of injection, the rate of 50 PV results in the highest average number of microbes.

Table 8: Average number of microbes throughout different rates

Injection Rate	Average Number of Microbes
1 PV	25
10 PV	77
25 PV	104
50 PV	107
100 PV	105
1000 PV	76

5.1.2 Saturation Analysis

As was already mentioned with equation (5.1, saturation is calculated internally by MRST based on the initial fluid composition, set by the user.

As an optimal value, initial liquid phase saturation of 20% was chosen. This results from the set initial molar fraction of $z_0 = [0,0.89,0.11,0,0]$ representing $[H_2, H_2O, N_2, CO_2, CH_4]$. Different comparisons were carried out in order to see the effect of liquid phase saturation on the microbial population dynamics as well as on the produced methane and the injected nutrients.

On all of the figures that are showing temporal changes, presented are per-day-average values throughout the sample. All of the injections are carried out at the rate of 50 PV for the duration of 10 days.

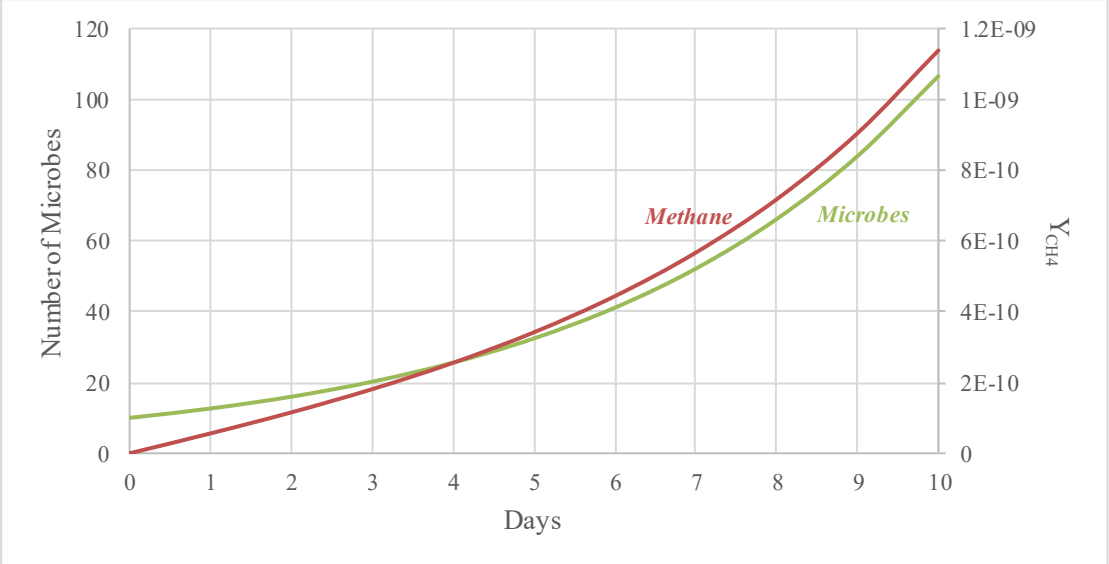

Figure 11: Base case $S_{li} = 20\%$ temporal distribution of the microbial population and produced CH_4

As the liquid phase saturation increases so does the microbial population, seeing how that is the assumed microbial habitat. Naturally, as the microbes are growing the production of methane, due to the metabolic reaction, is larger (*Figure 13*). Therefore, indirectly, saturation increase dictates the generation of methane as well.

Opposite to this, when the liquid phase saturation is reduced so is the population growth and the methane production with it.

Described *liquid saturation-microbes-methane* relation is observable on the below figures. Trend set by liquid phase saturation throughout the core sample (*Figure 12*) is reflected onto microbial population distribution and therefore onto the produced methane as well (*Figure 13*).

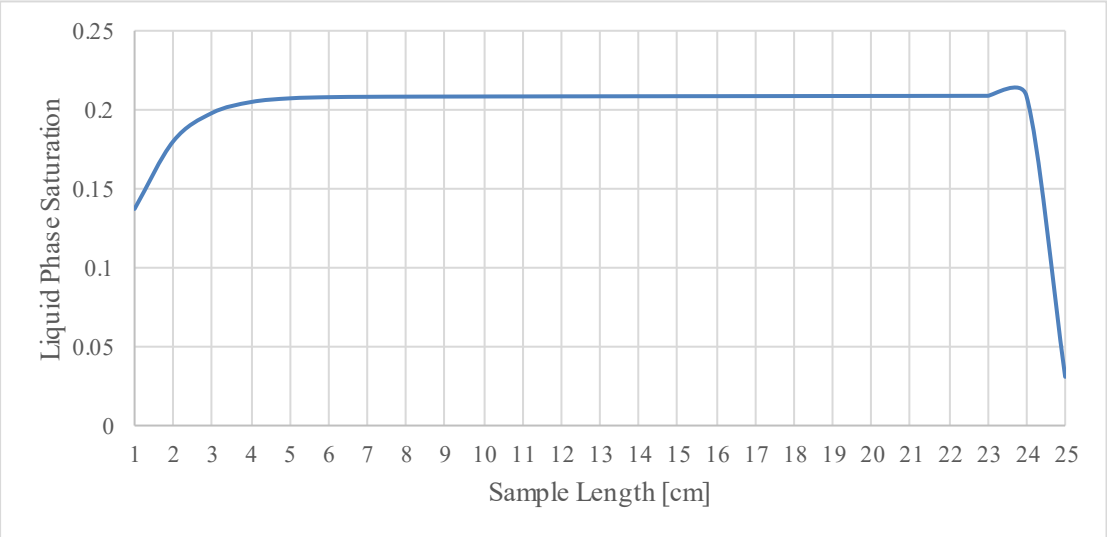

Figure 12: $S_{li} = 20\%$; day 10 of the injection – spatial distribution of liquid phase saturation

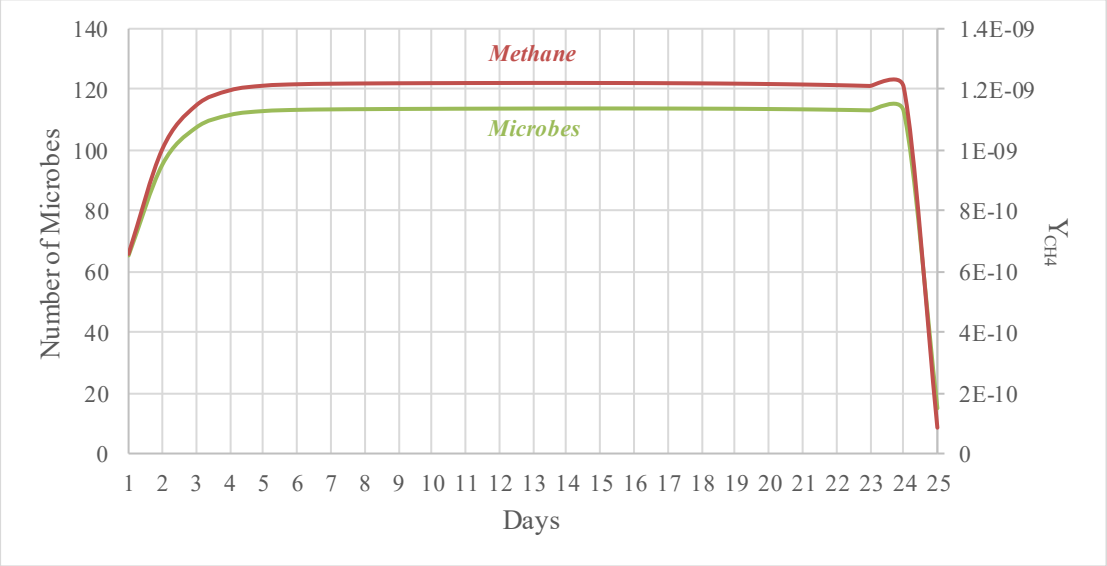

Figure 13: $S_{li} = 20\%$; day 10 of the injection – spatial distribution of microbes and produced CH_4

Compared to the optimal 20% were two larger and two lower saturations. Larger values of the initial liquid phase saturation were set to 38% and 76%.

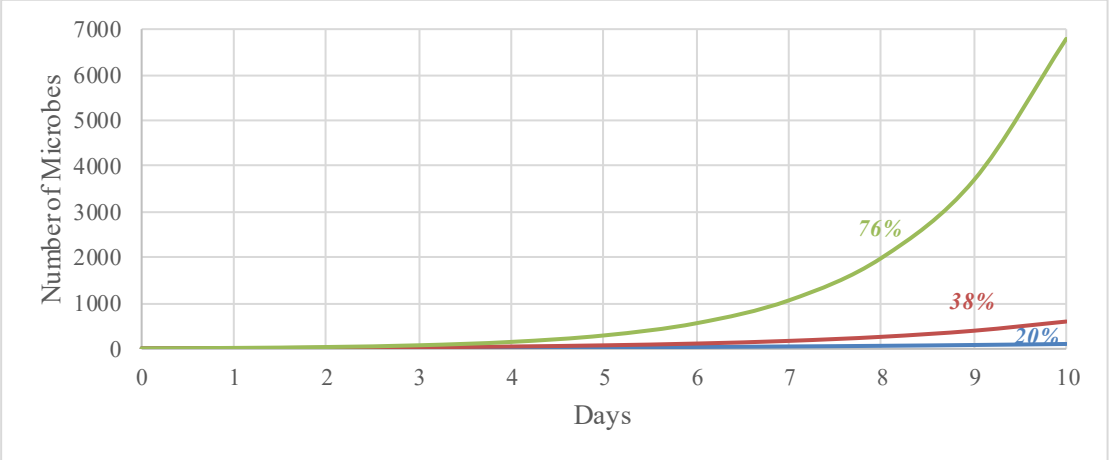

Figure 14: Microbial growth comparison between $S_{li} = 20\%$, $S_{li} = 38\%$ and $S_{li} = 76\%$

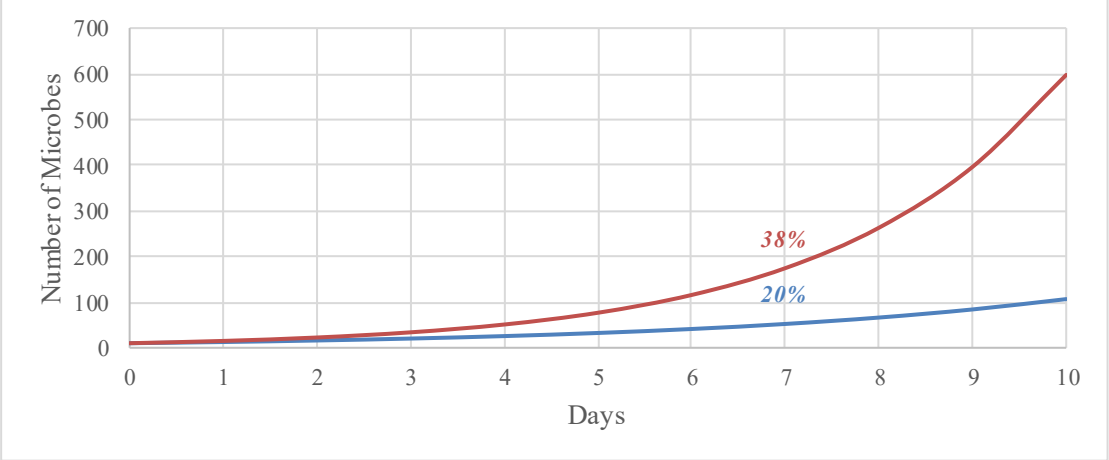

Figure 15: Microbial growth comparison between $S_{li} = 20\%$ and $S_{li} = 38\%$

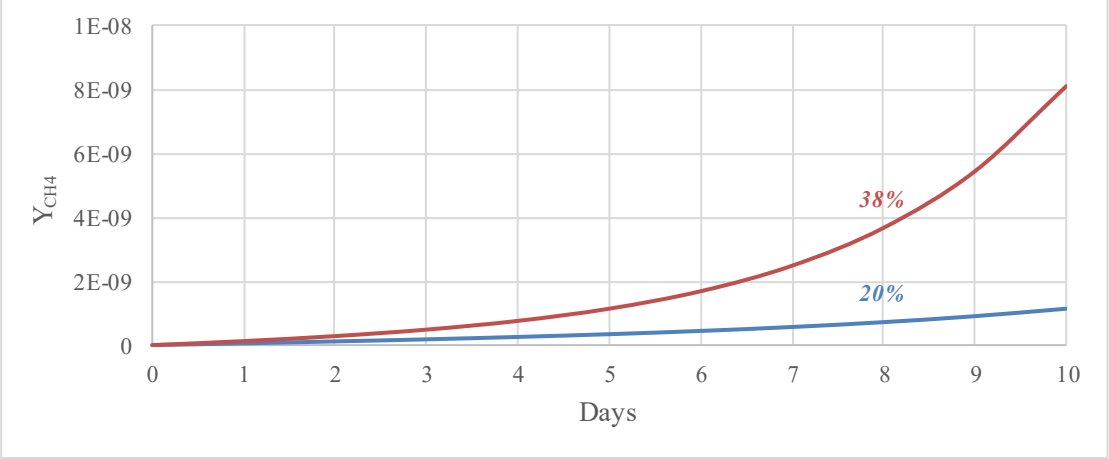

Figure 16: Produced methane comparison between $S_{li} = 20\%$ and $S_{li} = 38\%$

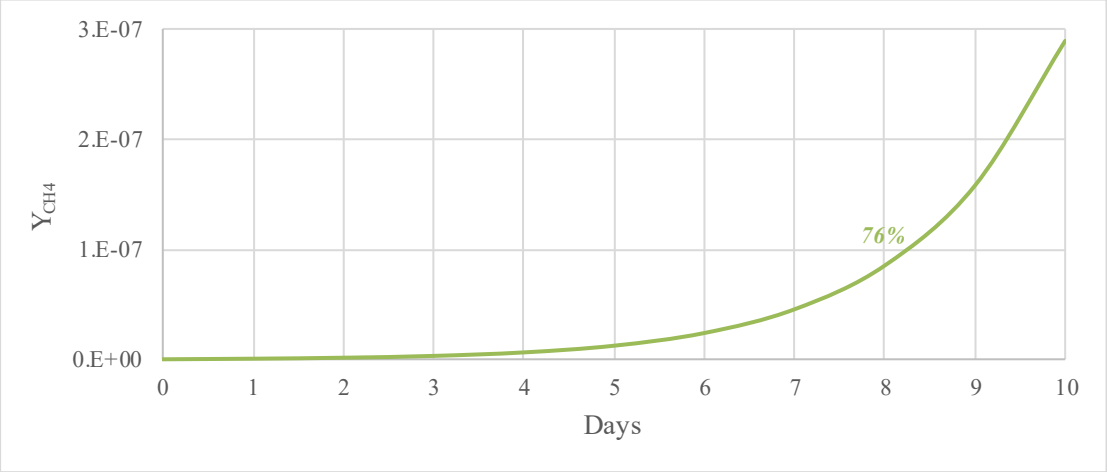

Figure 17: $S_{li} = 76\%$ produced methane

With the $S_{li} = 76\%$ the described (in *Chapter* 2) doubling of the microbe number with each generation is observed (*Table 9*), where the largest exponential growth phase is therefore established. So much so, that a semi-log plot is used to represent the results (*Figure 18*).

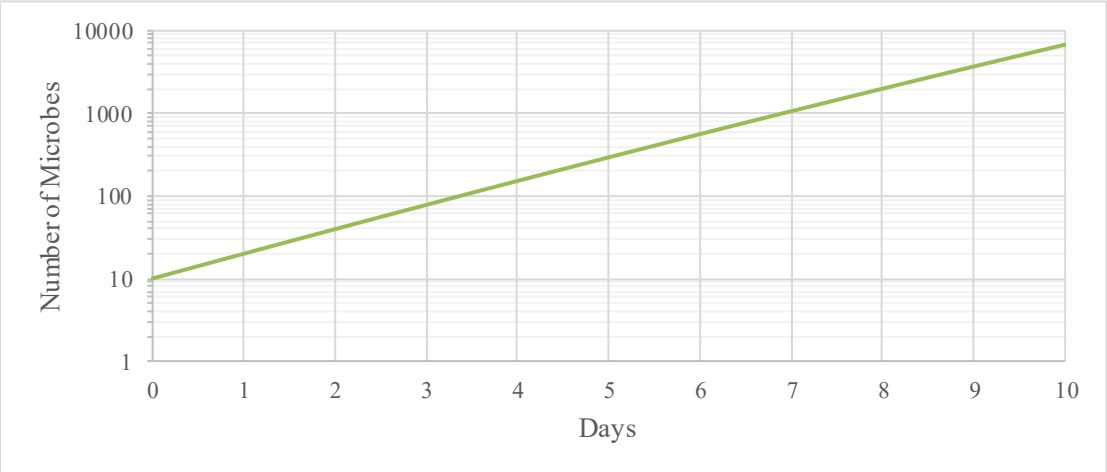

Figure 18: $S_{li} = 76\%$ microbial population distribution throughout 10 days of injection

Table 9: Doubling of the microbial number per each generation at $S_{li} = 76\%$ (injection starts on day 1 and ends on day 10)

Days	Average Number of Microbes
0	10
1	20
2	40
3	78
4	152
5	293
6	561
7	1061
8	1988
9	3690
10	6787

Two lower saturations were introduced as well, $S_{li} = 10\%$ and $S_{li} = 2\%$. From *Figure 19* it can be observed that the initial liquid phase saturation of 2% results in the non-growing metabolism (described in *Chapter 2*).

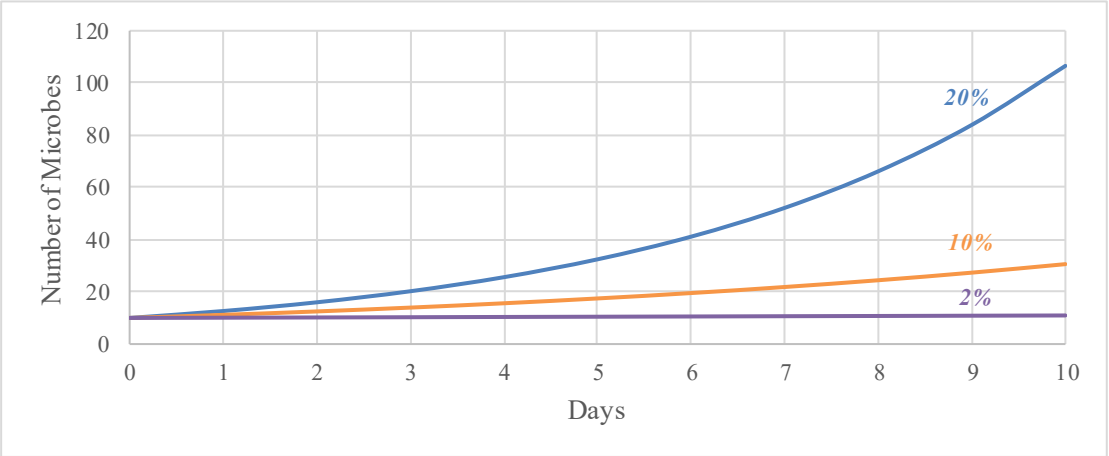

Figure 19: Microbial growth comparison between $S_{li} = 20\%$, $S_{li} = 10\%$ and $S_{li} = 2\%$

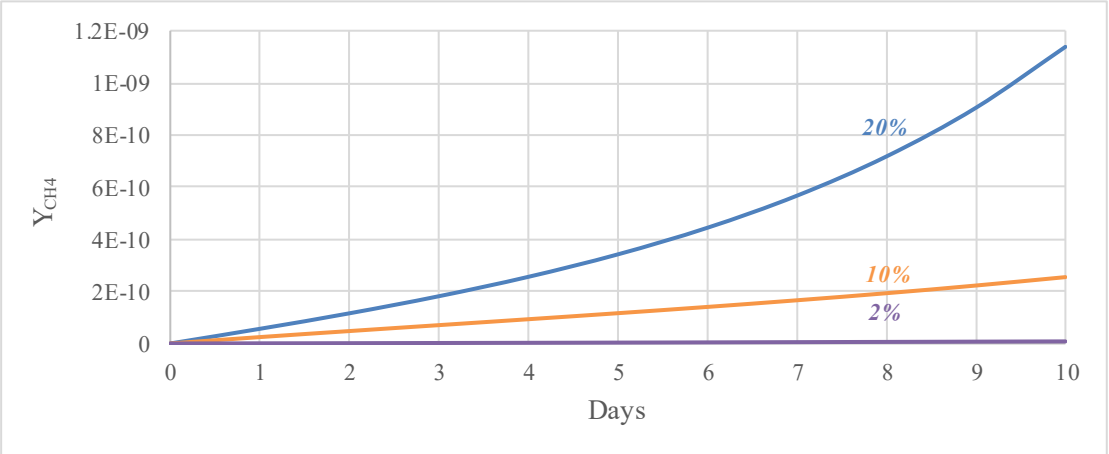

Figure 20: Produced methane comparison between $S_{li} = 20\%$, $S_{li} = 10\%$ and $S_{li} = 2\%$

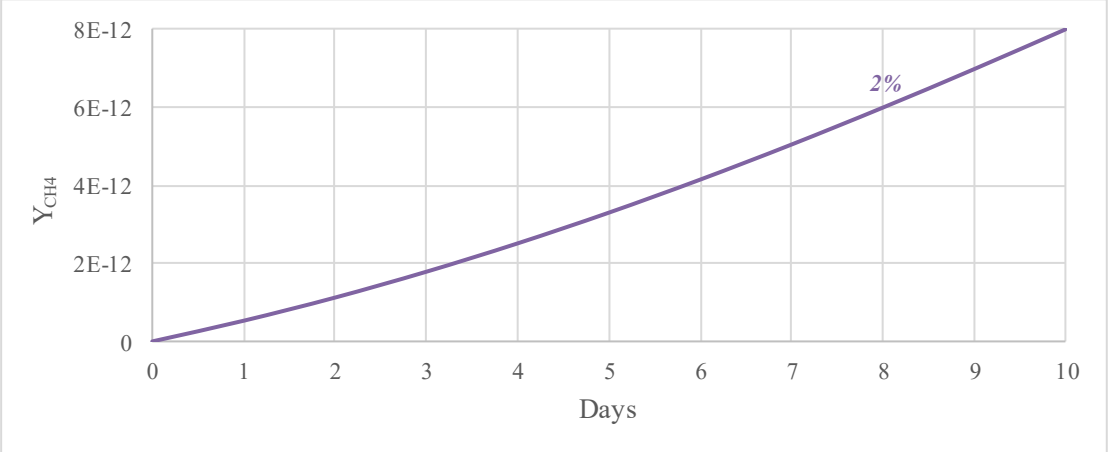

Figure 21: $S_{li} = 2\%$ produced methane

By increasing the *decay coefficient (b)* (Eq. (4.1) under the same initial liquid phase saturation conditions the *death phase* (also described in *Chapter 2*) was reached (*Figure 22*). Production of methane is indicating that the metabolism is ongoing, but it is not resulting in growth. In the first case (*Figure 19)* it is just sufficient enough to maintain the life of the microbes while in the second case it is not.

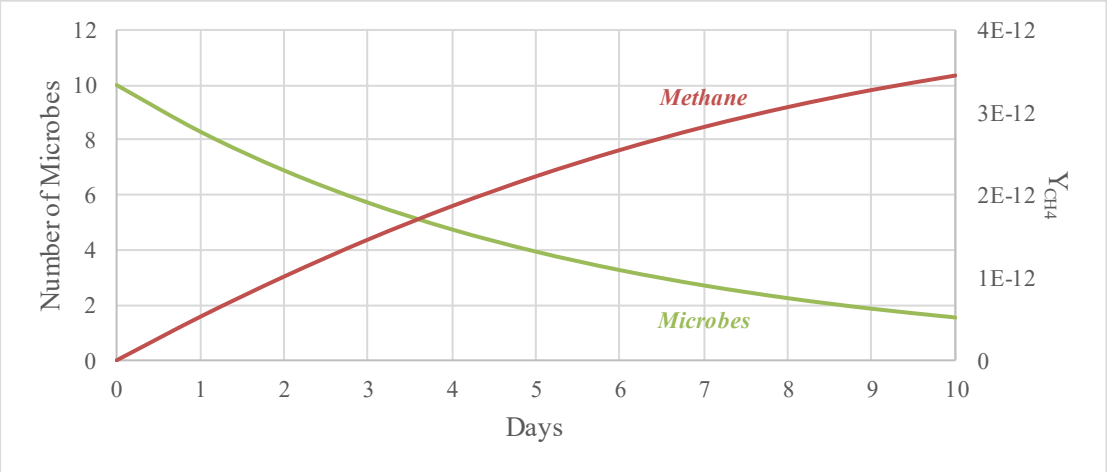

Figure 22: $S_{li} = 2\%$; microbial and produced methane distribution during the death phase

Throughout all of the presented results, a very small amount of produced methane can be seen. The reason for this is that so far simulation runs had only 10 microbes per grid block set as the initial population of the sample. In order to get more of a realistic representation, number of initial microbes at each grid block was changed to the average value calculated from the data available in the literature (*Table 4*).

This being $2.36 * 10^8$ resulting in $n_{initial} = 9.45 * 10^{12}$ $\left[\frac{1}{m^3}\right]$. Reflecting the high increase in the number of the initial microbes, the produced methane is much higher than what it was before (*Figure* 23).

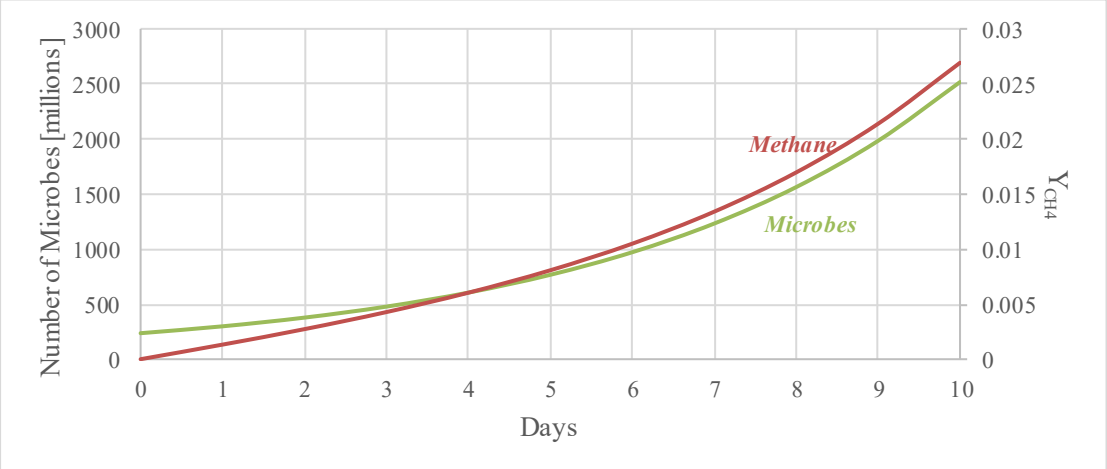

Figure 23: Per-day-average microbial and produced methane distribution

Another interesting observation was that as the saturation of the liquid phase increases in the simulated core sample, so does the injected gas mixture reach the set composition of 80% H_2 and 20% CO_2 faster (see ***Boundary Conditions*** above) (*Figure 24*).

An increase in the liquid phase saturation means that the volume occupied by the liquid phase is larger, therefore it is easier for the injected gas to fill the smaller volume occupied by the vapor phase and therefore get to the set fraction faster.

As an example, on the *Figure 24* one can see that for the liquid phase initial saturation of 76% it takes for about 3 days to reach the set injection fraction between H_2 and CO_2 in the vapor phase of the sample. While for the rest of the lower saturations it takes around 8 to 10 days to do so.

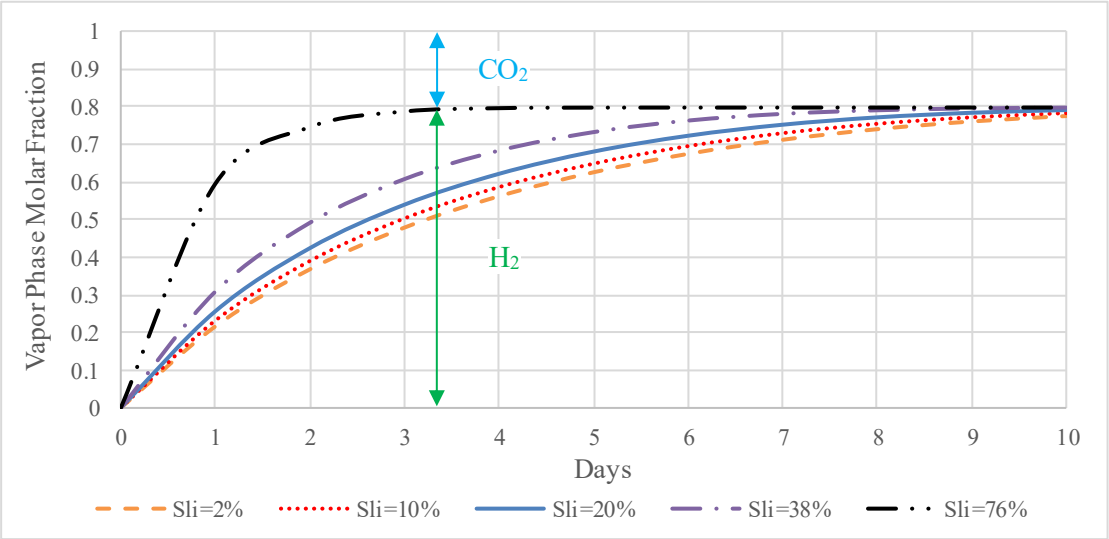

Figure 24: Time needed in order to reach the set injected nutrients' fractions inside of the sample's vapor phase throughout different initial liquid phase saturation conditions

5.2 Realistic Case

Simulations that were done in *Base Case* all had the same number of microbes distributed initially throughout the sample, which is not something that can be expected in reality. That is the reason why here the initial microbial distribution throughout the grid cells was set to random values. Initial liquid phase saturation was 20% and the injection rate was 50 PV with the injection lasting 10 days.

For the *initial microbial distribution* throughout the grid blocks normal (gaussian) distribution was used based on values from *Table 4*:

- Mean value was set to $2.36 * 10^8$;
- Standard Deviation was set to $4.01 * 10^8$;
- Lower and upper values are 5000 and $1.4 * 10^9$.

Table 10: Random distributed initial number of microbes

Grid Block	Initial Number of Microbes	Grid Block	Initial Number of Microbes	Grid Block	Initial Number of Microbes
1	6.78E+08	11	1.26E+08	21	5.07E+08
2	8.29E+08	12	1.04E+09	22	3.01E+07
3	1.02E+08	13	9.75E+08	23	7.27E+08
4	8.46E+08	14	3.67E+08	24	8.98E+08
5	4.86E+08	15	6.59E+08	25	5.28E+08
6	7.97E+07	16	1.14E+08		
7	2.15E+08	17	3.19E+08		
8	4.15E+08	18	8.51E+08		
9	9.76E+08	19	6.49E+08		
10	1.01E+09	20	9.84E+08		

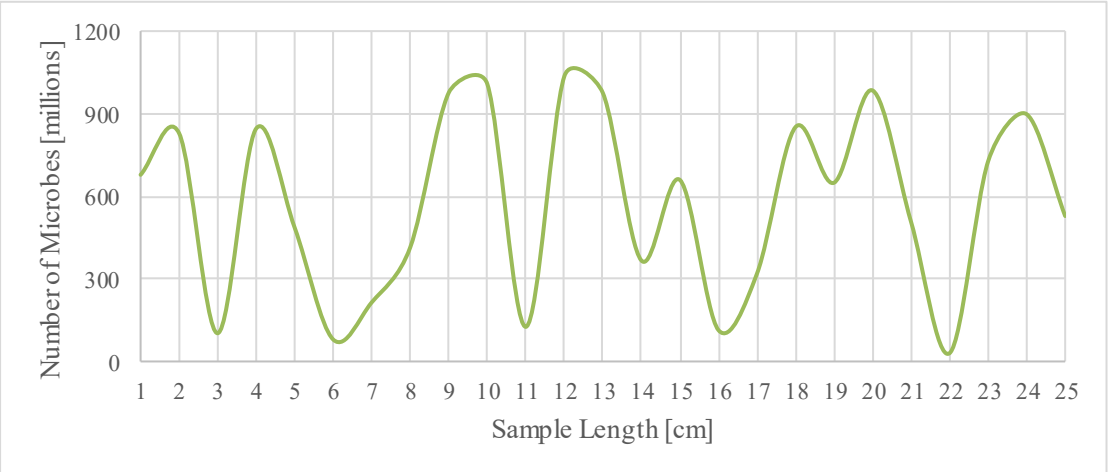

Figure 25: Initial distribution of the microbes

Compared to the *Base Case* simulation that was carried out by initially distributing $2.36 * 10^8$ microbes throughout the sample (*Figure 23*), it can be seen that an average population growth and therefore, as well, the production of methane throughout 10 days of injection are now bigger (*Figure 26* & *Figure 27*).

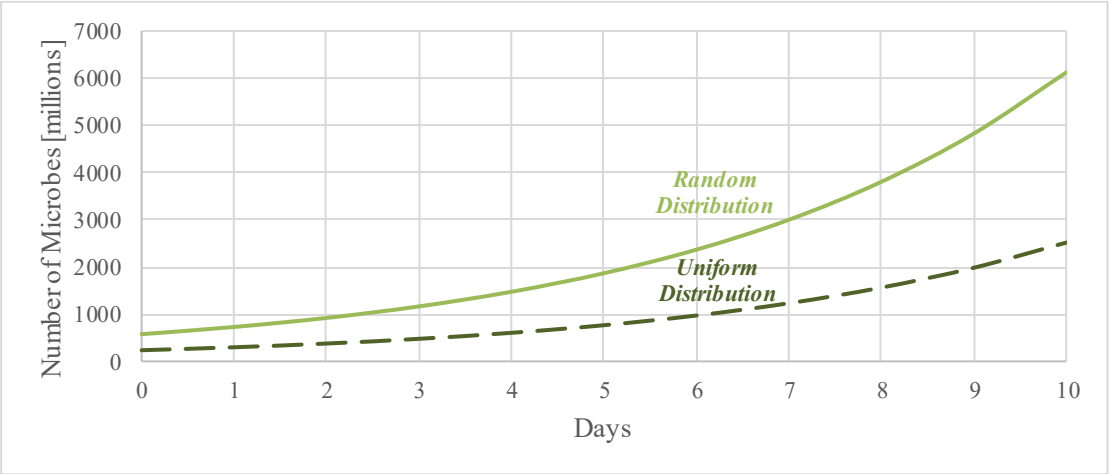

Figure 26: Per-day-average microbial distribution Realistic vs Base Case

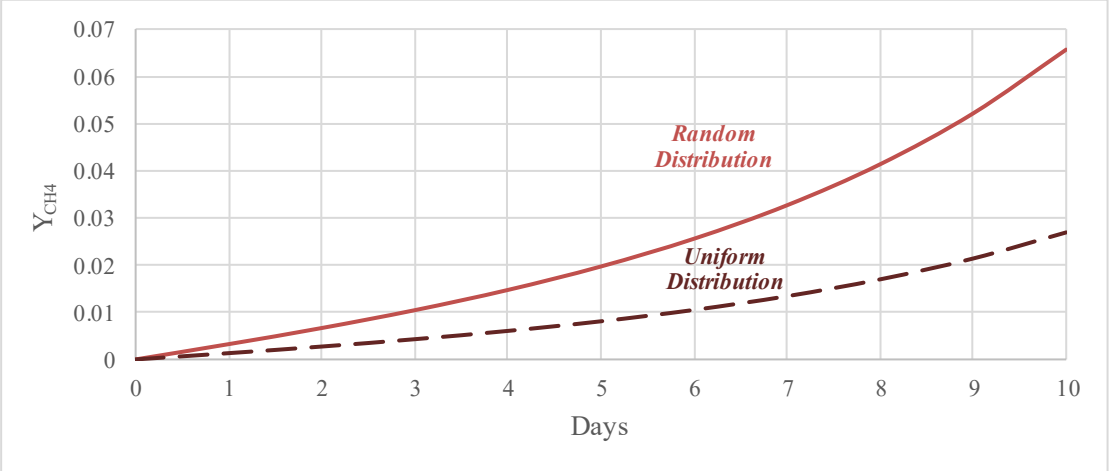

Figure 27: Per-day-average produced methane Realistic vs Base Case

What can also be tested as a controlling factor is a random porosity and permeability distribution. For the case that has been the focus of this work, which is a core sample 25 cm long, this is not something that is realistically possible. Because, when applied, this would mean that throughout such a small length drastic changes in both porosity and permeability would occur after each centimeter (seeing how we have 25 grids). This is therefore more suitable for the simulation of the whole reservoir.

Nevertheless, we present here what such a set-up in MRST would look like:

- Used for the random distribution of *porosity* throughout the model is the `gaussianField` function of MRST that generates a porosity field in which the values are distributed using the normal (gaussian) distribution.

  As an input to the function porosity range as well as a standard deviation should be set (for example, one can say that porosity is to be from 10% to 30% with the standard deviation of 5% – meaning that around 70% of data will deviate $\pm 5\%$ from the mean value of the porosity).

- *Permeability*, being a function of porosity, is calculated using the Carman-Kozeny relation (LIE, 2019):

$$K = \frac{1}{8 * \tau * A_v^2} * \frac{\phi^3}{(1-\phi)^2} \tag{5.2}$$

  ϕ… porosity;
  τ… tortuosity;
  A_v… specific surface area.

  Assuming that the medium is made of uniform spherical grains and that their radius is $r_p = 10\ \mu m$, grain specific area is then $A_v = \frac{3}{r_p}$. Assumed tortuosity is $\tau = 0.81$.

  The resulting form of the Carman-Kozeny equation that would be implemented:

$$K = \frac{1}{72 * \tau} * \frac{\phi^3 * r_p^2}{(1-\phi)^2} \tag{5.3}$$

$$K = \frac{1}{72 * 0.81} * \frac{\phi^3 * (10 * 10^{-6})^2}{(1-\phi)^2}$$

Chapter 6

Alternative Approach

The idea behind this chapter is to present an alternative approach to the mathematical modeling of methanogenic archaea's metabolism. Namely, it is a review of the work done by Muñoz-Tamayo and his colleagues in 2019.

Their work had a focus on the methanogenic microbes present in the mammalian gut, so from that point of view, it is in a completely different area of focus compared to the UHS. On the other hand, the strategy for the modelling approach of the methanogenic reactions and microbial growth itself is something that can be correlated.

They have constructed, what they call, an energetic-based mathematical model of methanogenesis capturing the dynamics of the reaction (Muñoz-Tamayo et al., 2019).

6.1 Mathematical Model of Methanogenesis

To start, in this model not just a single reaction of methanogenesis is used, but a combination of methanogenesis reaction for methane production treated as *a catabolic reaction* (6.1) and *an anabolic reaction* (6.2) representing microbial formation. Where it was assumed that ammonia is the nitrogen source for microbial formation, as well as that microbial biomass has a formula of $C_5H_7O_2N$.

$$CO_2 + 4H_2 \rightarrow CH_4 + 2H_2O \tag{6.1}$$

$$5CO_2 + 10H_2 + NH_3 \rightarrow C_5H_7O_2N + 8H_2O \tag{6.2}$$

This is one of the key points that are differentiating this approach from the other works available in the literature.

The mathematical model consists of five differential equations. The first one is considering the biomass concentration:

$$\frac{dx_{H_2}}{dt} = \mu_{max} * exp\left(-\frac{K_s * V_g}{n_{g,H_2}}\right) * x_{H_2} - k_d * x_{H_2} \tag{6.3}$$

x_{H_2}… biomass concentration $\left[\frac{mol}{L}\right]$;

k_d… cell death rate constant $\left[\frac{1}{h}\right]$;

As much as this equation and the one used in the model of our work (4.1) are similar (both having the microbial growth and the microbial death term), there are also quite some differences in the approach. Namely microbial mass is treated as a concentration of *moles* of biomass per liter of the aqueous solution, while in our model there was microbial density (n), the *number* of microbes per m^3. When it comes to the growth model instead of the popular *Monod* equation (3.8) what is used is *the kinetic rate function* developed by Desmond-Le Quéméner and Bouchez (Desmond-Le Quéméner and Bouchez, 2014):

$$\mu = \mu_{max} * exp\left(-\frac{K_s * V_g}{n_{g,H_2}}\right) \tag{6.4}$$

Where:

$$K_s = \frac{E_M + E_{dis}}{V_{harv} * E_{cat}} \tag{6.5}$$

μ_{max}… maximum specific growth rate constant $\left[\frac{1}{h}\right]$;

V_g… gas (vapor) phase volume $[mL]$;

n_{g,H_2}… amount of hydrogen in gaseous (vapor) phase $[mol]$.

K_s is the *affinity constant* $\left[\frac{mol}{L}\right]$, derived on the energy basis, focusing on the concept of exergy. That is the maximum work that microorganism has available during chemical transformation. From equation (6.5):

E_M… stored exergy during growth $\left[\frac{kJ}{mol}\right]$;

E_{dis}… dissipated exergy during growth $\left[\frac{kJ}{mol}\right]$;

E_{cat}… catabolic exergy of one molecule of energy-limiting substrate $\left[\frac{kJ}{mol}\right]$;

V_{harv}… volume in which each microbe can harvest the chemical energy in the form of substrate molecules $[L]$.

The second equation focuses on the change in the CO_2 concentration in the liquid phase:

$$\frac{ds_{CO_2}}{dt} = -\frac{-Y_{CO_2} * \mu_{max}}{Y} * exp\left(-\frac{K_s * V_g}{n_{g,H_2}}\right) * x_{H_2} \qquad (6.6)$$

$$-k_L a * \left(s_{CO_2} - K_{H,CO_2} * \frac{R*T*n_{g,H_2}}{V_g}\right)$$

s_{CO_2}… CO_2 concentration in the liquid phase $\left[\frac{mol}{L}\right]$;

Y_{CO_2}… CO_2 yield factor $\left[\frac{mol(CO_2)}{mol(H_2)}\right]$;

Y… microbial biomass yield factor $\left[\frac{mol(biomass)}{mol(H_2)}\right]$;

$k_L a$… mass transfer coefficient $\left[\frac{1}{h}\right]$;

K_{H,CO_2}… Henry's law coefficient $\left[\frac{mol}{L*bar}\right]$;

R… ideal gas law constant $\left[\frac{bar*L}{mol*K}\right]$;

T… temperature $[K]$.

The first term on the right side of the equation is relating the yield factors of both carbon dioxide and microbial biomass to the growth model and the biomass concentration. The second term is dedicated to the liquid-gas transfer of CO_2, governed by the mass transfer rate coefficient and the concentration difference of CO_2 in the liquid (s_{CO_2}) and in the vapor (gas) phase. Carbon dioxide concentration in the vapor phase is calculated using the ideal gas law (providing the CO_2 partial pressure in the vapor phase) and Henry's law.

The rest of the governing differential equations are referring to the molar gas phase changes of hydrogen, carbon dioxide and produced methane:

$$\frac{dn_{g,H_2}}{dt} = -\frac{\mu_{max}}{Y} * exp\left(-\frac{K_s * V_g}{n_{g,H_2}}\right) * V_L * x_{H_2} \qquad (6.7)$$

$$\frac{dn_{g,CO_2}}{dt} = V_L * k_L a * \left(s_{CO_2} - K_{H,CO_2} * \frac{R * T * n_{g,H_2}}{V_g}\right) \qquad (6.8)$$

$$\frac{dn_{g,CH_4}}{dt} = \frac{Y_{CH_4} * \mu_{max}}{Y} * exp\left(-\frac{K_s * V_g}{n_{g,H_2}}\right) * V_{L*} x_{H_2} \qquad (6.9)$$

Yield factors for carbon dioxide and methane are calculated using the microbial biomass yield factor for which the values are provided in the literature. In addition to that, the stoichiometry of the two chemical reactions (6.1) and (6.2) is used:

$$Y = \frac{1}{10} * (1 - f) \tag{6.10}$$

With f being the fraction of hydrogen used for the catabolic reaction (6.1), therefore $(1 - f)$ would be the fraction used in the anabolic reaction (6.2). The fraction $\frac{1}{10}$ refers to the stoichiometry from the second equation, relating biomass to hydrogen (since Y is in $\left[\frac{mol(biomass)}{mol(H_2)}\right]$). From equation (6.10) one can calculate the f, and from there the rest of the yield factors are gained:

$$Y_{CO_2} = \frac{1}{4} * f + \frac{5}{10} * (1 - f) \tag{6.11}$$

$$Y_{CH_4} = \frac{1}{4} * f \tag{6.12}$$

6.2 Future Work

The presented model is describing the batch type of laboratory experiments, therefore to be used in a work similar to the one done in this thesis it would have to be optimized for the injection type of experiments. An advective term would be necessary for equation (6.6, besides that the approach for our model was to use molar fractions since that is what the MRST solver is providing, while here emphasis is on the actual mols. Also, the mentioned difference in treating microbes as part of the chemical reaction and therefore not having a microbial density in the system but the microbial concentration should also be addressed.

This chapter is here to serve as a potential starting point for future work that could implement the presented mathematical model in a numerical simulation of the UHS, or simulations of laboratory experiments mimicking UHS conditions. Where then results of these simulations could be compared with the ones provided in this thesis or any other work available in the literature for that matter.

Chapter 7

Conclusions

Our main goal set for this work was fulfilled. We managed to provide a tool that can present both methanogenic archaea's population change in the presence of nutrients, as well as the changes to the gas composition due to their metabolism.

The available mathematical model finalized by Mr. Hagemann has been further adapted for the needs of this work. Namely, the second equation describing the reactive transport of nutrients was used as a basis for the development of the new equation dealing only with the effects of the reaction, seeing how advective effects were provided by the MRST's solver.

The 1-D simulation results have provided the microbial population growth phases that have been observed in the laboratory batch experiments described in the literature. Whether it be the exponential growth phase including the duplication of the cells after each generation, or the stationary and death phases. Both microbial, as well as the produced methane, dependencies on the liquid phase saturation were described as well as the effect of different injection rates of the nutrient gas mixture.

As it is, the developed code described in this thesis can be used as a tool for the laboratory core injection experiments, but also, we have provided a possible basis for future work that can be carried out.

Further development of this code could include a 3-D reservoir scale model and the effects that methanogenic archaea, as well as other microbial species, would have on the stored hydrogen. Also, possible would be the complete UHS cycle simulation, including the injection phase, resting phase (where hydrogen sits stored in the reservoir) and finally withdrawal phase. Simulations with the same set-up could then be run in both MRST as well as in the popular DuMu^{X} and therefore provide a variety of prediction sources.

Also, the approach presented in *Chapter 6* can be adapted to the UHS scenario seeing how the mathematical model there treats bioreaction members in a different way than the one used in this work, providing a possible upgrade to what has been done here and so far in other works.

References

Azizmohammadi, S. (2018) Reservoir Fluids: Vapor-Liquid Equilibrium.

Azizmohammadi, S. (2022) Reservoir Simulation Methods 2 Practical: 1st Lecture: Introduction.

Azizmohammadi, S. and Rath, A. (2021) Reservoir Simulation Methods 1 Practical: Matlab Programming.

Bauer, S. and Pichler, M. (2017) 'Underground Sun Storage: Ein Projekt zur Erforschung der Wasserstoffverträglichkeit von Erdgasporenspeichern', energie | wasser-praxis, pp. 64–69.

Ben-Israel, A. (1965) 'A modified newton-raphson method for the solution of systems of equations', Israel Journal of Mathematics, vol. 3, no. 2, pp. 94–98.

Buzek, F., Onderka, V., Vančura P. and Wolf, I. (1994) 'Carbon isotope study of methane production in a town gas storage reservoir', Fuel, vol. 73, no. 5, 747-752.

C. G., M. and D. M., C. (2010) Introductory Finite Difference Methods for PDEs, Ventus Publishing ApS.

Cord-Ruwisch, R., Seitz, H.-J. and Conrad, R. (1988) 'The capacity of hydrogenotrophic anaerobic bacteria to compete for traces of hydrogen depends on the redox potential of the terminal electron acceptor', Archives of Microbiology, vol. 149, no. 4, pp. 350–357.

Desmond-Le Quéméner, E. and Bouchez, T. (2014) 'A thermodynamic theory of microbial growth', The ISME journal, vol. 8, no. 8, pp. 1747–1751.

Flemisch, B., Darcis, M., Erbertseder, K., Faigle, B., Lauser, A., Mosthaf, K., Müthing, S., Nuske, P., Tatomir, A., Wolff, M. and Helmig, R. (2011) 'DuMux: DUNE for multi-{phase,component,scale,physics,…} flow and transport in porous media', Advances in Water Resources, vol. 34, no. 9, pp. 1102–1112.

Ganzer, L. (2021a) Reservoir Simulation Methods 1: 6th lecture: Discretization-Gridding.

Ganzer, L. (2021b) Reservoir Simulation Methods 1: 7th lecture: Discretization-Solution.

Ganzer, L., Reitenbach, V., Pudlo D., Panfilov, M., Albrecht D. and Gaupp R. (2013) 'The H2STORE project-experimental and numerical simulation approach to investigate processes in underground hydrogen reservoir storage', EAGE Annual Conference & Exhibition incorporating SPE Europec.

Gentry, T. J., Gerba, C. P. and Pepper, I. L. (2015) Environmental microbiology, 3rd edn, Amsterdam, Elsevier.

Hagemann, B. (2017) Numerical and Analytical Modeling of Gas Mixing and Bio-Reactive Transport during Underground Hydrogen Storage, Dissertation, Clausthal University of Technology and Université de Lorraine [Online]. Available at https://tel.archives-ouvertes.fr/tel-01735019.

Hagemann, B., Panfilov, M. and Ganzer, L. (2016) 'Modelling Bio-reactive Transport in Underground Hydrogen Storages: Spatial Separation of Gaseous Components', ECMOR XV - 15th European Conference on the Mathematics of Oil Recovery.

Hagemann, B., Rasoulzadeh, M., Panfilov, M., Ganzer, L. and Reitenbach, V. (2016) 'Hydrogenization of underground storage of natural gas', Computational Geosciences, vol. 20, no. 3, pp. 595–606.

Heinemann, N., Alcalde, J., Miocic, J. M., Hangx, S. J. T., Kallmeyer, J., Ostertag-Henning, C., Hassanpouryouzband, A., Thaysen, E. M., Strobel, G. J., Schmidt-Hattenberger, C., Edlmann, K., Wilkinson, M., Bentham, M., Stuart Haszeldine, R., Carbonell, R. and Rudloff, A. (2021) 'Enabling large-scale hydrogen storage in porous media: The scientific challenges', Energy & Environmental Science, vol. 14, no. 2, pp. 853–864.

Ian H. Bell and the CoolProp Team (2020) Welcome to CoolProp — CoolProp 6.4.1 documentation [Online]. Available at http://www.coolprop.org/contents.html (Accessed 26 September 2022).

Kabuth, A., Dahmke, A., Beyer, C., Bilke, L., Dethlefsen, F., Dietrich, P., Duttmann, R., Ebert, M., Feeser, V., Görke, U.-J., Köber, R., Rabbel, W., Schanz, T., Schäfer, D., Würdemann, H. and Bauer, S. (2017) 'Energy storage in the geological subsurface: dimensioning, risk analysis and spatial planning: the ANGUS+ project', Environmental Earth Sciences, vol. 76, no. 1.

Kleinitz, W. and Böhling, E. (2005) 'Underground Gas Storage in Porous Media - Operating Experience with Bacteria on Gas Quality', SPE Europec/EAGE Annual Conference.

Qian, L., Mingham, C. G. and Causon, D. M. (2011) Introductory Finite Volume Methods for PDEs, bookboon.com.

LIE, K.-A. (2019) An Introduction to Reservoir Simulation Using MATLAB/GNU Octave, Cambridge University Press.

LIE, K.-A. and Møyner, O. (2021) Advanced Modeling with the MATLAB Reservoir Simulation Toolbox, Cambridge University Press.

Lovley, D. R. and Phillips, E. J. (1987) 'Competitive mechanisms for inhibition of sulfate reduction and methane production in the zone of ferric iron reduction in sediments', Applied and environmental microbiology, vol. 53, no. 11, pp. 2636–2641.

Luo, X., Wang, J., Dooner, M. and Clarke, J. (2015) 'Overview of current development in electrical energy storage technologies and the application potential in power system operation', Applied Energy, vol. 137, pp. 511–536.

MathWorks, T. and Inc. (2022) MATLAB Programming Fundamentals.

Mills, A. F. and Coimbra, C. (2015) Basic heat and mass transfer, 3rd edn, San Diego, CA, Temporal Publishing.

Monod, J. (1949) 'THE GROWTH OF BACTERIAL CULTURES', Annual Review of Microbiology, vol. 3, no. 1, pp. 371–394.

Muñoz-Tamayo, R., Popova, M., Tillier, M., Morgavi, D. P., Morel, J.-P., Fonty, G. and Morel-Desrosiers, N. (2019) 'Hydrogenotrophic methanogens of the mammalian gut: Functionally similar, thermodynamically different-A modelling approach', PloS one, vol. 14, no. 12, e0226243.

Murphy, M. E. and Ginn, R. T. (1999) 'Modeling microbial processes in porous media', Hydrogeology Journal, no. 8.

Nag, S. (1998) 'The Archimedes principle and Gauss's divergence theorem', Resonance, vol. 3, no. 11, pp. 18–29.

Panfilov, M., Panfilova, I., Toleukhanov, A. and Kaltayev, A. (2012) 'Bio-reactive two-phase transport and population dynamics in underground storage of hydrogen: Natural self-organisation', ECMOR 2012 - 13th European Conference on the Mathematics of Oil Recovery.

Sebastien, D. (2016) 'Patagonia Wind - Hydrogen Project: Underground Storage and Methanation', 21st World Hydrogen Energy Conference.

Simon, J., Ferriz, A. M. and Correas, L. C. (2015) 'HyUnder – Hydrogen Underground Storage at Large Scale: Case Study Spain', Energy Procedia, vol. 73, pp. 136–144.

SINTEF (2022) MRST [Online]. Available at https://www.sintef.no/projectweb/mrst/ (Accessed 8 September 2022).

Talaro, K. P. and Chess, B. (2018) Foundations in microbiology: Basic principles, New York, NY, McGraw-Hill Education.

Zivar, D., Kumar, S. and Foroozesh, J. (2021) 'Underground hydrogen storage: A comprehensive review', International Journal of Hydrogen Energy, vol. 46, no. 45, pp. 23436–23462.

Appendix A

Workflow of a Numerical Simulator

To simulate a reservoir, or in our case a specific phenomenon that is occurring in the reservoir under the presence of different fluids, defined steps need to be taken to get from the problem's setup to the solution. What we have is a computer program that solves equations which are describing the processes of interest from the physical world.

Steps include *Formulation* (containing non-linear partial differential equations), *Discretization* (going from PDEs to non-linear algebraic equations), *Linearization* (resulting in linear algebraic equations), and *Solution*.

A.1 Formulation

The mathematical model described in *Chapter 3* and his adaptation described in *Chapter 4* represent the numerical simulation formulation of this work. Such formulation, in general, consists of listed equations that are going to be used to cover all of the necessary physics. The usual types of equations that can be found as a framework of a reservoir simulator are (Ganzer, 2021a):

- Balance equations or rather conservation laws (mass balance, mole balance (used in our work), momentum balance (Darcy's law) etc.);
- Equations of state;
- Rock and fluid models;
- Constraints (mole constraints, saturation constraints);
- Saturation functions (k_r, P_c);
- Source/sink equations;
- Boundaries and initial conditions.

All in all, the formulation of mathematical constraints for the simulation is set in this initial step. These equations are in form of *non-linear partial differential equations*.

A.2 Discretization

To solve these equations numerically, first, the geometrical domain gets subdivided into several grid cells that can be regular or irregular in shape and size. These cells consist of *grid cell volumes* as well as *grid cell points*. Points are located at the center of the cell and are controlling the volume that is constructed around it. The property assigned at the grid point location should represent the average for the entire grid cell volume.

Second, the spatial and time discretization of governing equations is necessary, since there is no analytical solution for the problem. During the process of discretization, equations are converted from the *non-linear partial differential* to *non-linear algebraic equations*

When it comes to spatial discretization – instead of finding a solution that is valid for every point in space (which would result in infinite solutions) solutions are reduced to finite points (one solution per grid point) (Azizmohammadi and Rath, 2021; Ganzer, 2021a).

So far there is a wide range of methods used to discretize space. The most common discretization techniques are (Azizmohammadi and Rath, 2021; C. G. and D. M., 2010; L. et al., 2011):

- Finite Difference Method (FDM), the oldest method;
- Finite Element Method (FEM) is, a newer, flexible method;
- Finite Volume Method (FVM) incorporates what is called a *control volume approach.*

With FEM one can generate very precise shapes which are not necessary when it comes to reservoir simulation where the rough approximation is good enough, which is available with FDM method. That makes it very efficient when it comes to computation compared to FEM (which requires more time and computational power). That is why most commercial reservoir simulators are using FDM. FVM incorporates strong points of the other two methods – it is reasonably fast and also reasonably flexible. Therefore, FVM is what new reservoir simulation projects are following. Our work in MRST is using exactly this method.

The first step in the equation's discretization of space and time is the replacement of *partial derivatives* from the conservation laws. Partial derivatives are replaced either by *differential* or *integral formulation*. When it comes to space, formulation depends on the used discretization method- the integral formulation is suitable for FEM and FVM approach, while the differential formulation is more suitable for the FDM. Also used for time discretization is the differential formulation.

A.2.1 Differential Formulation

The basis of the differential approach is the Taylor expansion series. Taylor expansion scheme for some property *f* can be written as a forward difference (Azizmohammadi and Rath, 2021; C. G. and D. M., 2010):

$$f(x+\Delta x) = f\,(x) + \left(\frac{\partial f}{\partial x}\right)\frac{\Delta x}{1!} + \left(\frac{\partial^2 f}{\partial x^2}\right)\frac{\Delta x^2}{2!} + \left(\frac{\partial^3 f}{\partial x^3}\right)\frac{\Delta x^3}{3!} + \cdots \tag{A.1}$$

Or as a backward difference:

$$f(x-\Delta x) = f\,(x) - \left(\frac{\partial f}{\partial x}\right)\frac{\Delta x}{1!} - \left(\frac{\partial^2 f}{\partial x^2}\right)\frac{\Delta x^2}{2!} - \left(\frac{\partial^3 f}{\partial x^3}\right)\frac{\Delta x^3}{3!} - \cdots \tag{A.2}$$

Both schemes represent the change of property *f* as distance increases (Δx) by its derivatives. In order to replace partial derivatives of first-order Taylor series is considered only up to the first-order derivative and the rest of the terms are cut off. Accuracy will be higher the further the terms that are considered in Taylor expansion, therefore after a cut-off, there is a certain error called *truncation error*. From the above Taylor expansion schemes, different *differential formulation schemes* for both space and time are derived, these are represented in *Table 11* below.

Table 11: Difference schemes for space and time (Azizmohammadi, 2022)

Scheme	Spatial	Time
Forward difference	$\left(\frac{\partial p}{\partial x}\right)_i^n = \frac{p_{i+1}^n - p_i^n}{\Delta x} + O(\Delta x)$	$\left(\frac{\partial p}{\partial t}\right)_i^n = \frac{p_i^{n+1} - p_i^n}{\Delta t} + O(\Delta t)$
Backward difference	$\left(\frac{\partial p}{\partial x}\right)_i^n = \frac{p_i^n - p_{i-1}^n}{\Delta x} + O(\Delta x)$	$\left(\frac{\partial p}{\partial t}\right)_i^{n+1} = \frac{p_i^{n+1} - p_i^n}{\Delta t} + O(\Delta t)$
Central difference	$\left(\frac{\partial p}{\partial x}\right)_i^n = \frac{p_{i+1}^n - p_{i-1}^n}{2\Delta x} + O(\Delta x^2)$	$\left(\frac{\partial p}{\partial t}\right)_i^n = \frac{p_i^{n+1} - p_i^{n-1}}{2\Delta t} + O(\Delta t^2)$
Second order derivative	$\left(\frac{\partial^2 p}{\partial x^2}\right)_i^n = \frac{p_{i+1}^n - 2p_i^n + p_{i-1}^n}{\Delta x^2} + O(\Delta x^2)$	

In the table as an example for the property *pressure* is taken, also, superscript "*n*" refers to the time increment and subscript "*i*" represents spatial increment. Term that is added at the end of each scheme ($0(\Delta x)$ or $0(\Delta t)$) represents mentioned *truncation error* (due to cut-off). First order derivative's forward or backward difference scheme is followed by the first-order error while the central difference scheme and second-order derivative have it squared since the cut-off has been done further into the Taylor expansion.

- Forward difference scheme – forward changes to the function.
- Backward difference scheme – backward changes to the function. When it comes to the time difference in the table, the notation for the current time step is "*n+1*" and for the previous it is "*n*". It could have been also written as:

$$\left(\frac{\partial p}{\partial t}\right)_i^n = \frac{p_i^n - p_i^{n-1}}{\Delta t}$$

- Central difference scheme – represents the combination of both

Explicit and Implicit Schemes Implemented in FDM

Taylor series provides the needed connection between derivation and difference terms required for the FDM as well as for the time discretization. With the FDM approach changes of properties in the grids are represented as the property differences between *grid cells* at *distance Δx* between *grid points* (*Figure 28*). The advantage of FDM is its easy implementation for geometrically simple domains (C. G. and D. M., 2010).

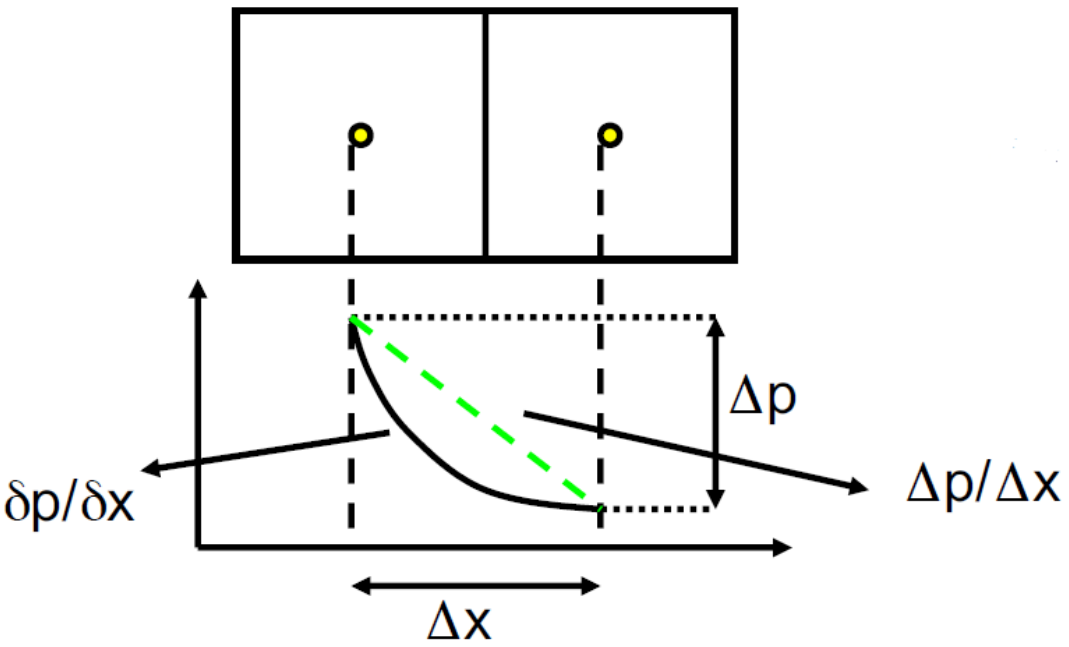

Figure 28: (top) Two grid cells at distance Δx between the grid points; (bottom) Pressure (as an example for cell property) difference between two cells (green line is the linearization of the actual behavior) (Ganzer, 2021a)

The forward difference scheme is based on the next neighboring cell (on the right side – "positive" direction from the current cell), and backward difference evaluates the difference between the current and the previous (in the "negative" direction) cell property value. For the time discretization, on the other hand, the same idea is applied but now differences are being calculated between cell properties at the current and future or rather previous *time-step*.

The central difference scheme simplifies the relation, the derivative of the current grid cell property is evaluated by using both "next" and "previous" cell values, with the distance then being doubled, and therefore the value of the current point itself is not used. Normally central difference is not used for time discretization.

Combining the provided difference schemes will lead to different sets of combined schemes, two of such schemes are relevant (C. G. and D. M., 2010):

Explicit scheme is formed if FTCS (forward time central space) combination is implemented, meaning that for the time *forward difference scheme* and for the space *central difference scheme* is used. The property value of the future time-step (n+1) would be calculated using the known property values of the current time-step (n). At which, property values at the current grid cell position and both grid cells at the neighboring positions are combined (to the left and right) (*Figure 29*).

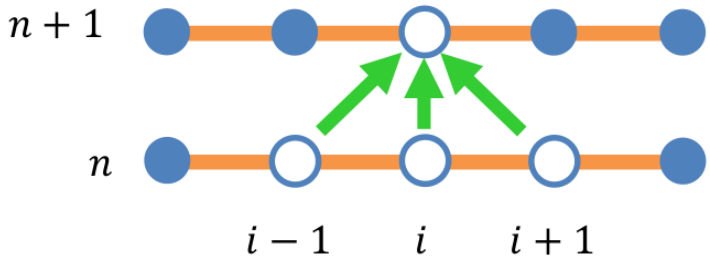

Figure 29: Graphical description of explicit scheme (Azizmohammadi, 2022)

Implicit scheme is formed if BTCS (backward time central space) combination is implemented, meaning that for time *backward difference scheme* is used and for the space *central difference scheme*. The property value of the current time-step (n+1) is unknown and it would be calculated using the known property value of the previous time-step (n) at the same position. Combined with that are the property values of both grid cells at the neighboring positions (to the left and right) of the current grid cell's position, but this time both of the neighboring property values are unknown as well (*Figure 30*).

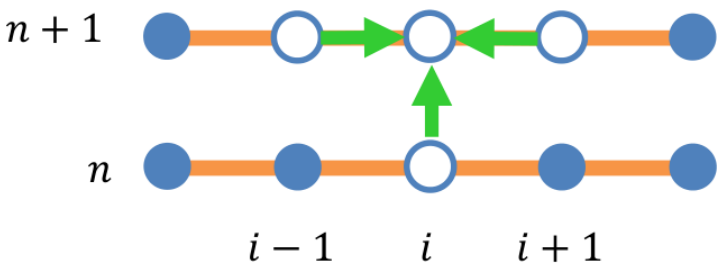

Figure 30: Graphical description of implicit scheme (Azizmohammadi, 2022)

After writing the equations for all of the nodes one would end up with a system of equations that is to be solved, simultaneously for all of the unknowns. This system is sorted out in a form of a linear equation system consisting of three matrices (C. G. and D. M., 2010):

$$A * x = b \tag{A.3}$$

A…coefficient matrix;

x…matrix containing unknown properties;

b…inhomogeneous part of the equation- matrix containing known values.

With the explicit scheme, there is no need for the system of equations which makes it simpler for implementation and is requiring less computational power and time. But it is also less stable when compared to the implicit scheme. The type of problems that are to be solved will dictate the decision which scheme to use (hyperbolic type of equations such as transport or saturation equations will be more suitable for the implicit scheme, while for parabolic, analytic type of equations explicit scheme would work very well).

Taking now for an illustrative example one-dimensional (1-D) pressure diffusion equation (which is indeed PDE) (Azizmohammadi, 2022; Azizmohammadi and Rath, 2021):

$$\frac{\partial^2 p}{\partial x^2} = \frac{1}{\alpha}\frac{\partial p}{\partial t} \tag{A.4}$$

where α is hydraulic diffusivity.

After applying to it FDM, the equation would take following forms:

- *Explicit scheme*

$$\left(\frac{\partial^2 p}{\partial x^2}\right)_i^n = \frac{p_{i+1}^n - 2p_i^n + p_{i-1}^n}{\Delta x^2}, \quad \left(\frac{\partial p}{\partial t}\right)_i^n = \frac{p_i^{n+1} - p_i^n}{\Delta t}$$

$$\frac{p_i^{n+1} - p_i^n}{\Delta t} = \alpha \frac{p_{i+1}^n - 2p_i^n + p_{i-1}^n}{\Delta x^2}$$

after rearranging:

$$p_i^{n+1} = p_i^n + \frac{\alpha \Delta t}{\Delta x^2}(p_{i+1}^n - 2p_i^n + p_{i-1}^n) \tag{A.5}$$

- *Implicit scheme*

$$\left(\frac{\partial^2 p}{\partial x^2}\right)_i^{n+1} = \frac{p_{i+1}^{n+1} - 2p_i^{n+1} + p_{i-1}^{n+1}}{\Delta x^2}, \quad \left(\frac{\partial p}{\partial t}\right)_i^{n+1} = \frac{p_i^{n+1} - p_i^n}{\Delta t}$$

$$\frac{p_i^{n+1} - p_i^n}{\Delta t} = \alpha \frac{p_{i+1}^{n+1} - 2p_i^{n+1} + p_{i-1}^{n+1}}{\Delta x^2}$$

after rearranging in such a way that on the left side of the equation are all of the unknown parameters and on the right side are known values:

$$p_{i-1}^{n+1} - \left(2 + \frac{\Delta x^2}{\alpha \Delta t}\right) p_i^{n+1} + p_{i+1}^{n+1} = -\frac{\Delta x^2}{\alpha \Delta t} p_i^n \tag{A.6}$$

To illustrate the mentioned system of equations (A.3), that is occurring when the implicit formulation of the problem is taken, let us take equation (A.6) and write it as:

$$a_{i-1} p_{i-1}^{n+1} + a_i p_i^{n+1} + a_{i+1} p_{i+1}^{n+1} = b_i \tag{A.7}$$

Each node (grid point) has such an equation, for each of the unknown parameters (in this example it is pressure) which results in an equation system. The coefficient matrix "*A*" consists of coefficients standing in front of each parameter value (in equation (A.7) those are a_{i-1}, $a_{i,}$ a_{i+1}). The size of the matrix depends on the number of unknowns per grid – its width responds to the number of unknowns and its height responds to the number of equations. Matrix "*x*" would contain all of the unknown properties (in equation (A.7) those are all three pressures). And finally, all of the known values that were set on the right side of the equations make up the matrix "b_i".

A.2.2 Integral Formulation

FVM is implementing the integral formulation, what that means is that the volume integral is applied over the individual grid cells to calculate the values of different properties (L. et al., 2011).

To illustrate integral formulation, let's take the general representing partial differential equation for the conservation law (a generalized form of the equations listed in *Chapter 3*) (Hagemann, 2017):

$$\frac{\partial}{\partial t} u + \nabla \cdot f(u) = g(u) \tag{A.8}$$

u… vector representing values of cell properties;

f… flux function;

g… source/sink function.

As already stated, the geometric domain is subdivided into finite non-overlapping control volumes, the conservation law that is represented above is valid for each of these control volumes. Having an integral form of:

$$\frac{\partial}{\partial t}\int_{V_i} u\, dV_i + \int_{V_i} f(u)\, dV_i = \int_{V_i} g(u)\, dVi \tag{A.9}$$

V_i… control volume of a grid cell;

i… grid cell index.

For the volumetric integral of the flux function (f) Gauss-Divergence theorem is applied. Which declares that the change of a property in a control volume is equal to the net change of that property caused over all cell faces. Transforming a volumetric integral into a surface integral (Nag, 1998):

$$\int_{V_i} f(u)\, dV_i = \int_{A_i} f(u) * n\, dA_i \tag{A.10}$$

A_i… total surface area of a grid cell;

n… unit normal vector.

A surface integral is equal to the sum over all sub-surfaces connecting the grid block to its neighbors (taking into account the parameter values of the current (i) and neighboring (j) grid cell):

$$\int_{A_i} f(u) * n\, dA_i = \sum_{j=1}^{m} k_{ij}(u_i, u_j) \tag{A.11}$$

m… number of neighboring cells;

k_{ij}… new flux function, taking now additionally into account geometrical quantities (distance between grid cells' points and cell interface area) as well as average rock and fluid properties.

Since for FVM *average values* of the grid cell properties are contained at the grid cell point (at the center of the grid) we can state that:

$$u_i = \frac{1}{V_i}\int_{V_i} u\, dV_i \tag{A.12}$$

and also:

$$g(u_i) = \frac{1}{V_i}\int_{V_i} g(u)\, dVi \tag{A.13}$$

Dividing the whole equation (A.9) by control volume "V_i", and placing in terms from equations (A.11), (A.12), (A.13):

$$\frac{\partial u_i}{\partial t} + \frac{1}{V_i}\sum_{j=1}^{m} k_{ij}(u_i, u_j) = g(u_i) \tag{A.14}$$

For the time discretization backwards difference scheme is used (*Table 11)*:

$$\left(\frac{\partial u}{\partial t}\right)_i^{n+1} = \frac{u_i^{n+1} - u_i^n}{\Delta t} \tag{A.15}$$

Finally, the discretization of the presented PDE (A.8), having the implicit formulation, looks like this:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{1}{V_i}\sum_{j=1}^{m} k_{ij}(u_i^{n+1}, u_j^{n+1}) = g(u_i^{n+1}) \tag{A.16}$$

Rearranging so that the current time-step's (*n+1*) unknown property values are on the same side:

$$u_i^{n+1} + \Delta t\left(\frac{1}{V_i}\sum_{j=1}^{m} k_{ij}(u_i^{n+1}, u_j^{n+1}) - g(u_i^{n+1})\right) = u_i^n \tag{A.17}$$

A.3 Linearization

After discretization we are left with the *non-linear algebraic equations* before the solution comes into play, equations need to be linearized. To do so Newton-Raphson method is used, this method is seeking zero points of the equation system. Equations are linearized using first-order derivatives (Ben-Israel, 1965).

If we are, for example, talking about the linearization of a single equation, Newton-Raphson's general form is derived in the following way:

Taylor series expansion:

$$f(x + \Delta x) = f(x) + \left(\frac{\partial f}{\partial x}\right)\frac{\Delta x}{1!} + \left(\frac{\partial^2 f}{\partial x^2}\right)\frac{\Delta x^2}{2!} + \left(\frac{\partial^3 f}{\partial x^3}\right)\frac{\Delta x^3}{3!} + \cdots$$

Cutting off after the first derivative and ignoring the truncation error:

$$f(x + \Delta x) = f(x) + \left(\frac{\partial f}{\partial x}\right)\Delta x \tag{A.18}$$

Newton-Raphson is valid for the setup where *f(x) = 0*, it is an equation-zero-point-seeking method, therefore:

$$f(x + \Delta x) = 0 \tag{A.19}$$

Resulting in (from (A.18) & (A.19)):

$$\Delta x = -\frac{f(x)}{f^{'}(x)}, \quad \Delta x = x^{n+1} - x^{n} \tag{A.20}$$

Arriving at Newton-Raphson's general form:

$$x^{n+1} = x^{n} - \frac{f(x^{n})}{f^{'}(x^{n})} \tag{A.21}$$

Equation (A.21) can be rearranged so that it shows the linearized form:

$$\Delta x * f'(x^{n}) = -f(x^{n}) \tag{A.22}$$

With the derivative being the slope of the function.

If we have a function and take the first-order derivative of it, we would result in a tangent of that function. With the Newton-Raphson method, this tangent will look to replace the original function (the approximated new look of the function). The smaller the variable changes per timestep the more will approximated linear function be true to the actual function.

When it comes to the application of the Newton-Raphson method in the numerical simulator, not an exact zero is found, but rather the best approximation. This is carried out by calculating the absolute relative approximate error (Ganzer, 2021b):

$$|\varepsilon| = \left|\frac{x^{n+1} - x^{n}}{x^{n+1}}\right| \tag{A.23}$$

The iterations will stop as the stopping criteria is reached (defined value of the ε).

Newton-Raphson method works on the same principle when it comes to the system of equations, namely it has the following look (Hagemann, 2017):

$$X^{n+1} = X^{n} - J(X^{n})^{-1} * F(X^{n}) \tag{A.24}$$

$$\Delta X^{n} J(X^{n}) = -F(X^{n}) \tag{A.25}$$

X^{n+1}… matrix containing unknown parameters of the future timestep;

X^{n}… matrix containing known parameters of the current timestep;

J… Jacobian matrix.

As can be seen, equations (A.24) and (A.25) are analogous to (A.21) and (A.22). Difference is in the fact that now matrices are used.

After linearization, the equations are now *linear algebraic.*

In the case of our work, these will be sorted in the linear equation system, since we are in the implicit formulation (A.3).

A.4 Solution

The solution technique of the linear equation system depends on the type of governing equations (integral vs. differential formulation) and spatial-time discretization in the end. For our work, using MRST, space is discretized by FVM while time discretization is carried out using a backward difference scheme, leading to the implicit formulation of the problem (LIE, 2019).

Two possible solution methods can be applied to the resulting system of linearized algebraic equations (Ganzer, 2021b):

- *Direct Methods* have the advantage of finding an exact solution, but the problem is that they are limited only to small models.
- *Iterative Methods* do not provide the exact solution of the equations but an approximation with a small error. But they can handle big and complex systems, therefore they are mostly used. The method starts with an initial guess, after which a sequence of approximations follows. These iterations terminate if an approximate solution is "close enough" to the true solution (stopping criteria) or the iteration maximum is exceeded.

All in all, if a discretized domain contains a small number of grid cells direct solution will provide exact and therefore ideal solutions to the unknown parameters' system of equations. While as soon as the number of grid cells gets larger (which is mostly the case) the iterative solution method would give closer values to the ideal solutions.

Linear solvers are what most of the CPU time is spent during the simulation, therefore the aim is to go for a memory-efficient solution. The biggest memory requirement is in storing the coefficient matrix "*A*" (Eq. (A.3)). With that in mind different solution methods, when it comes to reservoir simulators have been developed (Ganzer, 2021b):

- *Fully Implicit Method (FIM)*, all of the variables are solved implicitly. It is the most stable solution but large memory is required.

  If there would be for example 3 unknown property variables (pressure, water saturation, gas saturation) per grid cell, which would be the case for *Black Oil* model (for the *Compositional* this number is much bigger) and that there are 1 million grid cells, we would result in 3 million equations and unknowns. Matrix A would be a 3×10^6 by 3×10^6 matrix containing 9×10^{12} entries.

- *Implicit Pressure Explicit Saturation (IMPES)*, saturation is based on the last time-step while the pressure is calculated for the new time-step. Taking the same example, now there would be only one variable per grid block (pressure) matrix A would be a 1×10^6 by 1×10^6 matrix containing 10^{12} entries, which is less with a factor of 9 compared to the FIM. It is less memory-demanding, but at the same time, it has stability issues.
- *Adaptive Implicit Method (AIM)*, combines the strengths of the other two. High stability of the solution at the computational cost comparable to the IMPES method. Each grid block can change its implicit/explicit formulation status at any time-step. Grids, where variable changes are expected to be high, are using FIM and where the changes are not expected as often are set to IMPES method.

Appendix B

Parts of Code

B.1 Fluid Model Implementation

```
%% Compositional fluid model

compFluid =
TableCompositionalMixture({'Hydrogen','Water','Nitrogen','CarbonDioxide',
'Methane'}, {'H2', 'H2O', 'N2', 'CO2', 'CH4'});


flowFluid = initSimpleADIFluid('phases', 'WG', ... % water-gas system
                               'blackoil', false, ...
                               'rho', [1000, 1], ...
                               'mu', [1, 0.1]*centi*poise, ...
                               'n', [2, 1], ...
                               'c', [0, 1e-4]/barsa);
arg = {G, rock, ...
       flowFluid, ... % flow fluid
       compFluid,... % compositional mixture
       'water', true, 'oil', false, 'gas', true,... % water-gas system
       'liquidPhase', 'W', 'vaporPhase', 'G'}; % water=liquid, gas=vapor

gravity reset on

overall = GenericOverallCompositionModel(arg{:}); % overall mole fractions
```

B.2 Microbial Effects

The current time-step notation is “i+1” while the previous time-step is referred to as “i”.

```
while time < totalTime
    i = i + 1;

    schedule = simpleSchedule(dt, 'bc', bc);
    ctrl = schedule.control(schedule.step.control(1));
    [forces, fstruct] = overall.getDrivingForces(ctrl);
    overall = overall.validateModel(fstruct);
    state = overall.validateState(state);
    currControl = schedule.step.control(1);
    state = solver.solveTimestep(state, dt, overall, 'bc', bc);
    states = horzcat(states,state');

%1# Henry's law_____________________________________________________________
    xH2 = []; xN2 = []; xCO2 = [];
    p = states{i+1}.pressure;

    if i == 1
        yH2pr = states{i+1}.y(:,1);
        yN2 = states{i+1}.y(:,3);
        yCO2pr = states{i+1}.y(:,4);
        yCH4pr = states{i}.y(:,5);
        xH2O = states{i+1}.x(:,2);
    else
        yH2pr = states{i}.y(:,1);
        yN2 = states{i+1}.y(:,3);
        yCO2pr = states{i}.y(:,4);
        yCH4pr = states{i}.y(:,5);
        xH2O = states{i}.x(:,2);
    end

% ypr (pre-reaction) what is there at the beginning of the time step in
% the gaseous phase, before the reaction occurs

    pH2 = p.* yH2pr;
    xH2 = pH2./(76000*10^5);

    pN2 = p .* yN2 ;
    xN2 = pN2./(110000*10^5);

    pCO2 = p .* yCO2pr ;
    xCO2 = pCO2./(2720*10^5);

    states{i+1}.x(:,1) = xH2;
    states{i+1}.x(:,3) = xN2;
    states{i+1}.x(:,4) = xCO2;

% 2# 1st Equation; I) listing parameters that are changing with Time Steps
              %   II)Calculating new 'n'

   %I)_____________________________________________________________________
    n = (states{i}.microbes ./ CellVolume); % microbial density [1/m3]
    states{i}.microbialdensity = n;
```

```
    Sw = states{i+1}.s(:,1);

   %II)_____________________________________________________________________
    microgrowth = (microgrowthmax * (xH2 ./ (alphah2 + xH2)) .*
                  (xCO2 ./ (alphaco2 + xCO2))); %[1/s]

    microdecay = b; %[1/s]

    dndt = ( Sw.* (microgrowth .* n)) - (microdecay .* n); %[1/m3*s]
    dn = dndt * dt;
    nnew = dn + n;

    states{i+1}.microbialdensity = nnew;
    states{i+1}.microbes = (states{i+1}.microbialdensity .* CellVolume);

% 3# 2nd Equation;_____________________________________________________________

    rhow = states{i+1}.PVTProps.Density{1} ./
          (compFluid.molarMass(1) *xH2 + compFluid.molarMass(2) * xH2O +
           compFluid.molarMass(3) * xN2 + compFluid.molarMass(4) * xCO2);
    % molar density [mol/m3]
    states{i+1}.rhow = rhow;

   %CO2_____________________________________________________________________
    Xco2r = (((gammaCO2 * microgrowth/Y) .* n) * dt)./states{i+1}.rhow;
    nco2r = Xco2r .* states{i+1}.L; % [mol] amount in moles of CO2 consumed
.                                     in the reaction

% 4# Combining the Processes_____________________________________________
% Calculating new Ys, after the reaction

    V = 1 - states{i+1}.L;
    states{i+1}.V = V;

    Yco2 = (states{i+1}.y(:,4)) - (nco2r ./ states{i+1}.V);
% What is used in moles of CO2 in the reaction, in the liquid phase is
added from the gas phase. Therefore, the loss is calculated in the gas
phase. Plus, what was added by advection during this ts that is why the
solver fraction is used (since it is considering advection).

    states{i+1}.y(:,4) = Yco2;

    %H2_____________________________________________________
    nh2r = nco2r * gammaH2;
    Yh2 = (states{i+1}.y(:,1)) - (nh2r ./ states{i+1}.V);

    states{i+1}.y(:,1) = Yh2;




    %CH4_____________________________________________________
    nch4 = nco2r * gammaCH4;
    Ych4 = yCH4pr + (nch4 ./ states{i+1}.V);
    YCH4current = nch4 ./ states{i+1}.V;

    states{i+1}.YCH4current = YCH4current;
    states{i+1}.y(:,5) = Ych4;
```

```
    %H20_____________________________________________
    nh2o = nco2r * gammaH20;
    Xh2o = (states{i+1}.x(:,2)) + (nh2o ./ states{i+1}.L);

    states{i+1}.x(:,2) = Xh2o;

    time = time + dt;

end
```

List of Figures

List of Tables

Nomenclature

Φ	porosity	[-]
c	component mole fraction	[-]
S	saturation	[-]
$\hat{\rho}$	molar density	$[\frac{mol}{m^3}]$
q	source/sink	$[\frac{mol}{s.m^3}]$
k	chemical component	[-]
v	vapor phase	[-]
l	liquid phase	[-]
v_i	darcy velocity	$[\frac{m}{s}]$
K	absolute permeability	$[m^2]/[milidarcy]$
k_r	relative permeability	[-]
μ	dynamic viscosity	$[Pa.s]$
P_i	phase pressure	$[Pa]$
ρ_i	phase density	$[\frac{kg}{m^3}]$
g	gravity acceleration	$[\frac{m}{s^2}]$
l	length	$[m]$
J	dispersive/diffusive flux	$[\frac{mol}{m.s}]$
D_{diff}	effective molecular diffusion coefficient	$[\frac{m^2}{s}]$
D_{disp}	effective mechanical dispersion coefficient	$[\frac{m^2}{s}]$
n	microbial density	$[\frac{1}{m^3}]$
Ψ^{growth}	microbial growth function	$[\frac{1}{s}]$

Ψ^{decay}	microbial decay function	$[\frac{1}{s}]$
Ψ_{max}^{growth}	maximum specific rate of microbial growth	$[\frac{1}{s}]$
D_m	microbial diffusion coefficient	$[\frac{m^2}{s}]$
Υ	stoichiometric coefficient	[-]
Y	yield coefficient	$[\frac{1}{mol(H_2)}]$
α	half-velocity constant	$[\frac{mol}{mol}]$
b	decay coefficient	$[\frac{1}{s}]$
A	coefficient matrix	
x	matrix containing unknown properties	
b	inhomogeneous part of the equation-matrix containing known values	
J	jacobian matrix	
L	liquid phase mole fraction	[-]
V	vapor phase mole fraction	[-]
z_i	overall mole fraction of component *i*	[-]
p_k	partial pressure of component k above the solution	$[Pa]$
H_k	henry constant of component k	$[Pa]$
n_k	Amount of substance k	$[mol]$
Z	compressibility factor gained as a result of cubic EOS	

x_{H_2}	biomass concentration	$[\frac{mol}{L}]$
k_d	cell death rate constant	$[\frac{1}{h}]$
μ_{max}	maximum specific growth rate constant	$[\frac{1}{h}]$
V_g	gas (vapor) phase volume	$[mL]$
n_{g,H_2}	amount of hydrogen in gaseous (vapor) phase	$[mol]$
K_s	affinity constant	$[\frac{mol}{L}]$
E_M	stored exergy during growth	$[\frac{kJ}{mol}]$
E_{dis}	dissipated exergy during growth	$[\frac{kJ}{mol}]$
E_{cat}	catabolic exergy of one molecule of energy-limiting substrate	$[\frac{kJ}{mol}]$
V_{harv}	volume in which each microbe can harvest the chemical energy in form of substrate molecules	$[L]$
s_{CO_2}	CO_2 concentration in the liquid phase	$[\frac{mol}{L}]$
Y_{CO_2}	CO_2 yield factor	$[\frac{mol(CO_2)}{mol(H_2)}]$
Y	microbial biomass yield factor	$[\frac{mol(biomass)}{mol(H_2)}]$
k_La	mass transfer coefficient	$[\frac{1}{h}]$
K_{H,CO_2}	CO_2 henry's law coefficient	$[\frac{mol}{L.bar}]$
R	ideal gas law constant	$[\frac{bar.L}{mol.K}]$
T	temperature	$[K]/[°C]$

f	catabolic reaction's hydrogen fraction	[-]
τ	tortuosity	[-]
A_v	specific surface area	$[\frac{1}{m}]$

Abbreviations

EES	Electrical Energy Storages
UHS	Underground Hydrogen Storages
MRST	The MATLAB Reservoir Simulation Toolbox
REV	Representative Elementary Volume
PDE	Partial Differential Equation
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
FTCS	Forward Time Central Space
BTCS	Backward Time Central Space
FIM	Fully Implicit Method
IMPES	Implicit Pressure Explicit Saturation
AIM	Adaptive Implicit Method
EOS	Equation of State
PV	Pore Volume