JFTS+H: A JULIA-BASED PARALLEL SIMULATOR FOR THE DESCRIPTION OF THE COUPLED FLOW, THERMAL AND GEOCHEMICAL PROCESSES IN HYDRATE-BEARING GEOLOGIC MEDIA

A Thesis

by

KOTARO ANNO

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	George J. Moridis
Co-Chairs of Committee,	Thomas A. Blasingame
Committee Members,	Vivek Sarin
Head of Department,	Jeff Spath

August 2021

Major Subject: Petroleum Engineering

ABSTRACT

The objectives of this study are to develop (a) the Julia Flow and Transport Simulator (JFTS), a serial and parallel, high performance non-isothermal, multi-phase, multi-component general simulator of flow and transport through porous/fractured media, and (b) an associated module that describes quantitatively the Equation-of-State (EOS) of the complete H_2O+CH_4 system by covering all combinations of phase coexistence that are possible in geologic media and including all the regions of the phase diagram that involve CH_4-hydrates. As indicated by the JFTS name, this simulator is written in the Julia programming language and its parallelization is based on the Message Passing Interface (MPI) approach.

The JFTS+H simulator is a fully-implicit, Jacobian-based compositional simulator that describes the accumulation, flow and transport of heat, and up to four mass components (H_2O, CH_4, CH_4-hydrate and a water-soluble inhibitor) distributed among four possible phases (aqueous, gas, hydrate, and ice) in complex 3D geologic systems. The dissociation and formation of CH_4-hydrates can be described using either an equilibrium or a kinetic model.

The JFTS+H code can model the fluid flow, thermal and geochemical processes associated with the formation and dissociation of CH_4-hydrates in geological media, either in laboratory or in natural hydrate accumulations. The JFTS+H results show very good agreement with solutions of standard reference problems, and of large 2D and 3D problems obtained from another well-established and widely used numerical simulator.

The code exploits the speed, computational efficiency and low memory requirements of the Julia programming language. The parallel architecture of JFTS+H addresses the persistent problem of very large computational demands in serial hydrate simulations by using multiple processors to reduce the overall execution time and achieve scalable speedups.

DEDICATION

This thesis is dedicated to my family and friends for all of their help and support.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis (or) dissertation committee consisting of Dr. George J. Moridis and Dr. Thomas A. Blasingame of the Harold Vance Department of Petroleum Engineering and Dr. Vivek Sarin of the Department of Computer Science & Engineering.

Funding Sources

Graduate study was supported by the Texas Engineering Experiment Station and the Department of Petroleum Engineering at Texas A&M University through their start-up funding of the projects of Dr. Moridis.

NOMENCLATURE

A	Surface area, m^2
b	Klinkenberg b-factor accounting for gas slipping effects, Pa
C_R	Heat capacity of the dry rock, $\mathrm{J\ kg^{-1}\ K^{-1}}$
E_p	Efficiency on p processors, Fraction
F_A	Area adjustment factor, Fraction
f_{eq}	Fugacity at equilibrium at temperature T, Pa
f_v	Fugacity in the gas phase at temperature T, Pa
F^κ	Darcy flux vector of component $\kappa, \mathrm{kg\ m^{-2}\ s^{-1}}$
g	Gravitational acceleration vector, $\mathrm{m\ s^{-2}}$
H_β	Specific enthalpy of phase β, $\mathrm{J\ kg^{-1}\ K^{-1}}$
h_β^κ	Specific enthalpy of component κ in phase β, $\mathrm{J\ kg^{-1}\ K^{-1}}$
k	Intrinsic permeability, m^2
$k_{r\beta}$	Relative permeability of the phase β, Fraction
K_0	Intrinsic hydration reaction constant, $\mathrm{kg\ m^{-2}\ Pa^{-1}\ s^{-1}}$
k_θ	Composite thermal conductivity of the rock-fluids ensemble, $\mathrm{W\ m^{-1}\ K^{-1}}$
M^κ	Mass accumulation term of component κ, $\mathrm{kg/m^3}$
N_H	Hydration number
S_p	Speed-up on p processors, Fraction
P_β	Phase pressure, Pa
P_{cGW}	Gas-water capillary pressure, Pa
P_Q	Hydrate quadruple point, Pa
q^κ	Source/sink term of component κ, $\mathrm{kg}\ m^{-3}\ s^{-1}$

Q_H	Mass rate of hydrate dissociation/formation (kg/s)
R	Universal gas constant, J mol^{-1} K^{-1}
S_β	Phase saturation, Fraction
t	Time, sec
T	Temperature, K
T_{eq}	Equilibrium temperature, K
U_β	Specific internal energy of phase β, J kg^{-1}
u_G^κ	Specific internal energy of component κ in the gaseous phase β, J kg^{-1}
V	Volume, m^3
W^κ	Molecular weights of component κ = w, m, h, i (kg/mole)
X_β^κ	Mass fraction of component κ in phase β, kg/kg
α_P	Pore compressibility, 1/Pa
β	Subscript denoting a phase (= A, G, H, I)
Γ_n	Surface area of subdomain, m^2
ΔE_a	Hydration activation energy, J mol^{-1}
ΔH_{dis}	Enthalpy of dissociation/formation, J/kg
κ	Subscript denoting a component (= *w, m, h, i*)
μ_β	Viscosity, Pa s
ρ_β	Density of phase β, $\mathrm{kg\ m}^{-3}$
ρ_R	Rock density, $\mathrm{kg\ m}^{-3}$
ϕ	Porosity, Fraction

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Statement of the Problem

Gas hydrates are ice-like crystalline minerals in which gas molecules (such as methane, ethane, or carbon dioxide) are encased inside the lattice of ice crystals . Gas hydrates naturally occur under low temperature and high-pressure conditions such as those encountered in the permafrost and in deep ocean sediments (Kvenvolden, 1988).

The total worldwide volume of natural gas stored as hydrate is estimated to range between 10^{15} and 10^{18} ST m^3 (Sloan *et al.,* 1998). The enormous magnitude of this estimate makes hydrate reservoirs a potentially significant future energy resource even if only a fraction of the resource is actually producible. While commercial production of gas hydrate is widely viewed as being 20 or more years in the future, their potential provides a strong incentive to evaluate at this early stage the production potential of gas hydrate bearing geological media. The three main suggested methods of gas production from $\mathrm{CH_4}$-hydrate deposits are depressurization, thermal stimulation, and inhibitor injection, as well as their combinations (Makogon, 1997). A general equation describing the hydrate dissociation/formation reaction is

$$G + \ N_H \ H_2O \leftrightarrow G \circ N_H \ H_2O \ + \ \Delta H \tag{1.1}$$

where G is a hydrate-forming gas, $\mathrm{N_H}$ is the hydration number, and ΔH is the enthalpy of the reaction. $\mathrm{CH_4}$ is the most common hydrate former, $\mathrm{CH_4}$-hydrates constitute the overwhelming fraction of the global inventory of natural gas hydrates, and the associated dissociation reaction is strongly endothermic.

1.2 Objectives

The main objectives of this study are:

- To develop JFTS, a Message Passing Interface (MPI)-based parallel simulator written in

the Julia programming language that can provide a general framework for the solution of coupled problems of flow and transport of fluids and heat (and possibly other processes) through complex geologic media evaluate gas hydrate bearing geologic media.

- To develop a JFTS module describing all phase coexistence combinations of the phase diagram of the H_2O+CH_4 system that are possible in geologic media, which results a four-phase, three- or four-component, non-isothermal reservoir simulator capable of analyzing and evaluating all processes associated with the dissociation or formation of CH_4-hydrates.

- To evaluate the computational performance of the parallel simulator through comparisons to the performance of (a) the serial version of the code and (b) other well-established serial simulators that solve the hydrate problem.

1.3 Impetus of the Study

Simulators for the prediction of gas production from hydrate accumulations are inevitably complicated because of the need to fully consider the coupled flow, thermal, thermo-dynamic and geochemical processes associated with the hydrate dissociation and formation. A direct result of the complexity and of the related strong non-linearity of the physical and chemical processes is *short time steps*, very large computational requirements and, consequently, very long execution times if serial simulators are used. This study attempts to address this issue by implementing a parallel computational architecture in a Julia-based numerical simulator framework (in addition to the serial version of the code) that works seamlessly on a wide range of computational platforms and evaluates the gas production from hydrates within a fraction (depending on the number of processors) of the computational time required by a serial simulator with no loss of accuracy.

2. LITERATURE REVIEW

2.1 Processes for Methane Production from Gas Hydrate Reservoir

The difficulty of recovering CH_4 from gas hydrates is that this source of energy is in a solid form and thus unsuitable for production using conventional gas and oil techniques. Dissociation-induced release for CH_4 from natural hydrates accumulations prior to production is the only way that this abundant resource can be brought to production in an economic and safe manner.

Hydrate dissociation can be accomplished by one of the following methods (or combinations thereof): depressurization, thermal stimulation, and injection of hydrate inhibitors (Moridis, 2014). Depressurization involves lowering the pressure below the hydration equilibrium pressure at a given temperature, with the dissociation enthalpy of the endothermic dissociation reaction supplied by the system sensible energy. Thermal stimulation involves the addition of heat to raise the hydrate temperature above the hydration equilibrium temperature at a given pressure, providing the necessary enthalpy of dissociation in the process. The use of inhibitors lowers (shift) the equilibrium temperature of the hydration (formation and dissociation) equation. Commonly used inhibitors are alcohols and salt.

Field tests for producing gas from a naturally occurring gas hydrate deposit were conducted in the Daini-Atsumi Knoll in the eastern Nankai Trough area off Honshu island, Japan in 2013 and 2017 (Yamamoto, *et al.,* 2019). The operation was based on the depressurization method and two producer wells were located. The gas production of the first well continued for 12 days with intermittent sand-production events. The operation of the other well continued for a total of 24 days without sand problems. However, the bottomhole pressure was limited because of a higher water production rate than expected. According to the obtained data including gas and water production and the pressure and temperature, the heterogeneous gas hydrate distribution was estimated as the main reason for making the gap between the anticipated and actual behavior.

From the point of view of numerical simulation, the heterogeneity of gas hydrate reservoir

strongly affected the estimations of hydrate behaviors (Yilong *et al.,* 2019). The study investigated the hydrate reservoir at the Eastern Nankai Trough of Japan. The reservoir varied porosity, permeability, and hydrate saturation vertically. According to the obtained results, the hydrate dissociation zone highly depended on the reservoir heterogeneity and showed a unique dissociation front. Two numerical models (i.e. a simplistic model and a sedimentary-complex model) were compared and they revealed a simplistic model leaded to significant underestimation of gas productivity.

2.2 The TOUGH+HYDRATE Code

TOUGH+HYDRATE is a member of the TOUGH+ family of codes which is designed based on object-oriented programming as implementing in Fortran 95. TOUGH+ solves mass and energy balance equations describing fluid and heat flow in general multi-phase, multi-component in the porous media. The HYDRATE properties module is a numerical code that is used for the simulation of the behavior of hydrate-bearing geologic systems (Moridis, 2014). TOUGH+HYDRATE is under the Lawrence Berkeley National Laboratory license and is one of the most widely used numerical simulations for evaluating gas hydrate-bearing reservoirs. Currently, this code has been cited by 350. This study compared the results from TOUGH+HYDRATE v 1.5 for validation.

The MPI-parallel TOUGH+HYDRATE code (pT+H) (Zhang *et al.,* 2008) was developed and applied to field-scale assessment of a large, deep-ocean hydrate reservoir (Reagan *et al.,* 2013). The system in the study is up to 2.5M gridblocks and thousands of supercomputer nodes are required to simulate without loss of accuracy. The simulations show the challenges in gas production from deep, relatively cold systems with extensive water-bearing channels and connectivity of aquifers.

TOUGH+HYDRATE simulator was coupled with Millstone which is a code describing the geomechanical responses such as compaction and subsidence (Reagan *et al.,* 2019). The study demonstrated the capability of the TOUGH+Millstone code and achieved the estimations of the system flow, thermal, and geomechanical processes during gas production from an offshore hydrate deposit.

2.3 Numerical Simulators and Studies for Gas Hydrate

MH21 Research Consortium of Japan evaluated gas hydrate reservoirs in the Eastern Nankai Trough. As part of the project, a numerical simulator (MH21-HYDRES) for predicting dissociation and production of gas hydrate was developed (Kurihara, *et al.,* 2008). The MH21-HYDRES was originally developed by the University of Tokyo, and modified by Japan Oil Engineering Company, the University of Tokyo, Japan National Oil Corporation and National Institute of Advanced Industrial Science and Technology (Masuda et al., 1997; Masuda *et al.,* 1999; Masuda *et al.,* 2002; Kurihara *et al.,* 2005).

Sakamoto *et al.,* (2019) conducted a series of numerical analyses of the effects of acid injection as a secondary gas recovery from hydrate-bearing reservoirs after the depressurization operation using MH21-HYDRES. The objectives of acid injection are (a) utilizing the heat generation resulting from mineral dissociation, (b) modification in permeability resulting from mineral dissolution, and (c) inhibitor effects coming from the injected acid and dissolved minerals. They revealed that a high-temperature zone was formed because of heat generation and the total gas recovery through depressurization method and acid injection was estimated at approximately 90 %.

STOMP (Surface Transport Over Multiple Phases)-HYDE-KE was developed by Pacific Northwest National Laboratory (White, 2012). This simulator is designed for predicting the production of natural gas from hydrate-bearing reservoirs using depressurization, thermal stimulation, inhibitor injection, or guest molecule exchange technologies. The unique feature of the simulator is the capabilities for ternary hydrates of CH_4, CO_2, and N_2 mixtures. Another unique feature is that it tracks components (CH_4, CO_2, and N_2) in mobile and hydrate phases independently(HYDE). The phase transition between mobile and gas hydrate phases occurs via kinetic exchange(KE).

The STOMP-HYDE-KE simulator was used for the interpretation of the collected data obtained from the field test of injecting a mixture of N_2-CO_2 into a CH_4-hydrate accumulation beneath the permafrost on the Alaska North Slope (White *et al.,* 2014). This study demonstrated that parameters for the numerical simulators describing the kinetic processes can be determined from laboratory experiments.

Phale *et al.,* (2006) showed a method of injecting CO_2-microemulsion for Enhanced Gas Hydrate Recovery by using STOMP-HYD simulator. The concepts of this study are (a) CO_2 is thermodynamically favored over CH_4 in hydrate, (b) the heat released from the formation of CO_2 hydrate is greater than the heat needed for dissociation of CH_4-hydrate, (c) refilling pore space with CO_2 hydrate is expected to maintain the mechanical stability of the formation during production, and (d) storing CO_2 in the underground is environmentally friendly.

STARS was developed by CMG (Computer Modeling Group ltd.) which was designed for a three-phase, multi-component thermal and steam additive simulation (CMG, 2019). Gaddipati and Anderson (2012) conducted numerical simulations for the Gulf of Mexico (GOM) Gas Hydrates Joint-Industry-Project (JIP) using CMG STARS. A comprehensive logging-while-drilling dataset was collected, and a 3D reservoir model was created. From the results of these simulations, they revealed that hydrate deposits in GOM could be produced because of the high intrinsic reservoir quality and their proximity to the base of hydrate stability.

2.4 Matrix Solvers

Matrix solving is one of the most time-consuming parts of numerical simulations. To achieve faster computation, Conjugate Gradient (CG) can be used. CG is an effective matrix solver for symmetric positive definite systems (Saad, 1994). The method processes by generating vector sequences of iterates and residuals, and search directions used in updating the iterates and residuals. In every iteration of the method, two inner products are performed to compute scalars that are defined to make the sequences satisfy certain orthogonality conditions.

The CG method is not suitable for non-symmetric systems, because the residual vectors cannot be made orthogonal. Then, Bi Conjugate Gradient (BiCG) is another approach (Saad, 1994). BiCG requires computing a matrix-vector product $Ap^{(p)}$ and a transpose product $A^T\tilde{p}^{(k)}$. Few theoretical results show the irregular behavior of convergence. Bi Conjugate Gradient Stabilized (BiCG-stab) (Van der Vorst, 1992) was developed to solve non-symmetric linear systems while avoiding irregular convergence.

3. PARALLEL COMPUTING

3.1 Types of Parallel Computing Architecture

Parallel computing is a form of computation where many calculations are carried out simultaneously. There are three types of parallel computing models: (a) Single instruction, multiple data (SIMD), in which the same computations are carried out simultaneously on multiple data in parallel; (b) multiple instruction, multiple data (MIMD), in which different processors execute different operations on different data and (c) single program, multiple data(SPMD), a subcategory of MIMD, in which tasks are split up and executed simultaneously on multiple processors and on different data to achieve faster performance. SPMD combines independent threads of execution with global collective communication and synchronization operations and is the most common style of parallel programming (Barney, 2021).

In this study, the SPMD approach (single program, multi data) is employed in the JFTS+H simulator to achieve better/faster computational performance. A simplified sketch of the SPMD architecture is shown in **Fig. 3.1**. Tasks are split and distributed to multiple processors to run simultaneously. Each processor processes different input data, but the program/instruction is the same. SPMD usually involves message passing functions on a distributed memory computer architecture, which consists of a set of processors interconnected by a high-speed network. Each node can access only its local memory, and message passing systems (interfaces) are used to move data among processors.

3.2 Message Passing Interface - MPI

MPI (Message Passing Interface) (Blaise, 2019) is the standardized and portable library specification for message passing involved in the JFTS+H design, implementation and architecture. This architecture was designed for high performance computing in supercomputer environments. The objectives (and advantages) of MPI are practicality, portability, efficiency and flexibility. The MPI specification can be categorized into three functions: (a) the environment identifier that de-

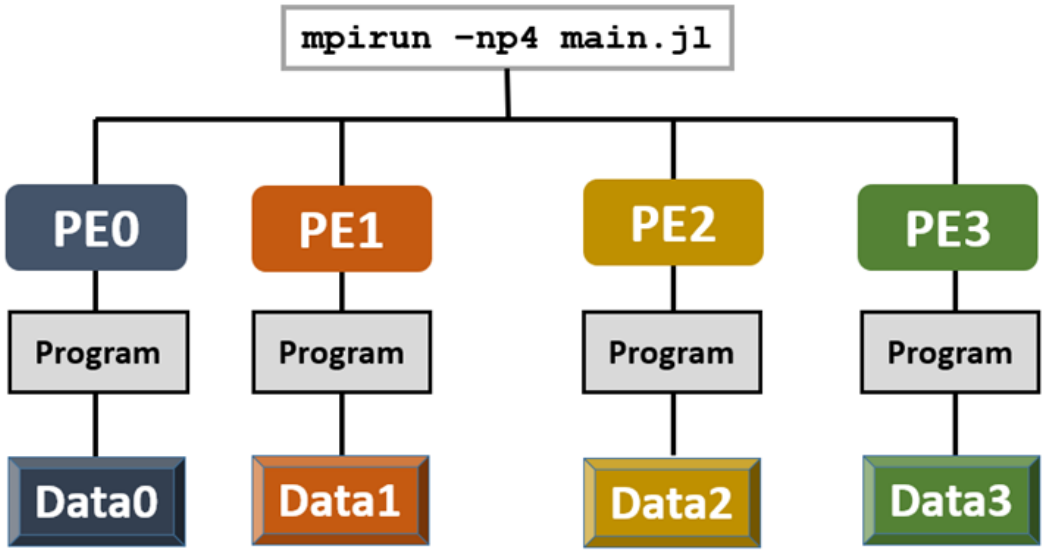

Figure 3.1: Visualized image of single program, multi data architecture.

termines the number of processors, as well as the rank and communicator of each processor; (b) point-to-point communications, which send and receive data between two processors; and (c) collective communications, barriers, broadcast, and reduction operation (Gropp *et, al.*, 1999). The MPI codes were originally written in FORTRAN, C, and C++, which, for the needs of this project, necessitated the development and implementation of a wrapper that enabled its function within the Julia language framework of the JFTS+H simulator.

3.2.1 Point-to-Point Communication

Point-to-Point (P-to-P) is a message-passing communication that involves two processors. The heat conduction problem discussed immediately below can be used as an example of (P-to-P) communications. The domain of the problem (shown in **Fig. 3.2**) is subdivided into 6 gridblocks and the left-hand boundary is kept at 100 °C. At first, the domain is divided and distributed to two processors. As shown in **Fig. 3.3**, processor 0 has and processes the data associated with gridblocks 0 to 3 and processor1 has gridblocks 4 to 6. To compute heat conduction between grid 3 and grid 4, processor 0 needs to get (and include in its computations) the data at gridblock (element) #4, processor 1 needs to incorporate in its computations the data associated with gridblock # 3. In the

P-to-P communication operation, processor 0 sends to Processor 1 data on element 3 and receives from Processor 1 data associated with element 4. Processor 1 sends to Processor 0 data on element 4 and receives from Processor 0 data associated with element 3 (shown in **Fig. 3.4**).

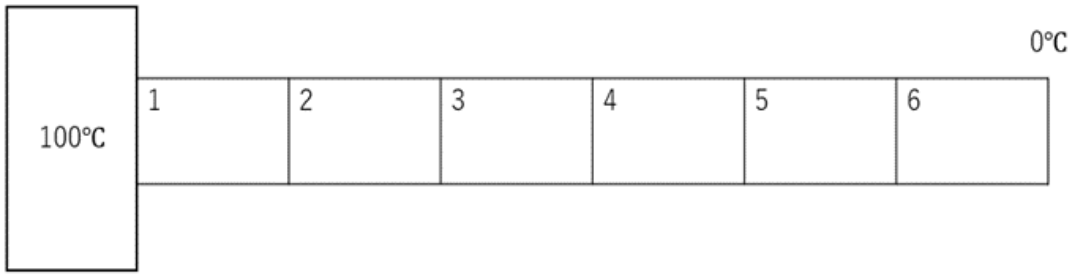

Figure 3.2: Point-to-Point communication in a heat conduction problem.

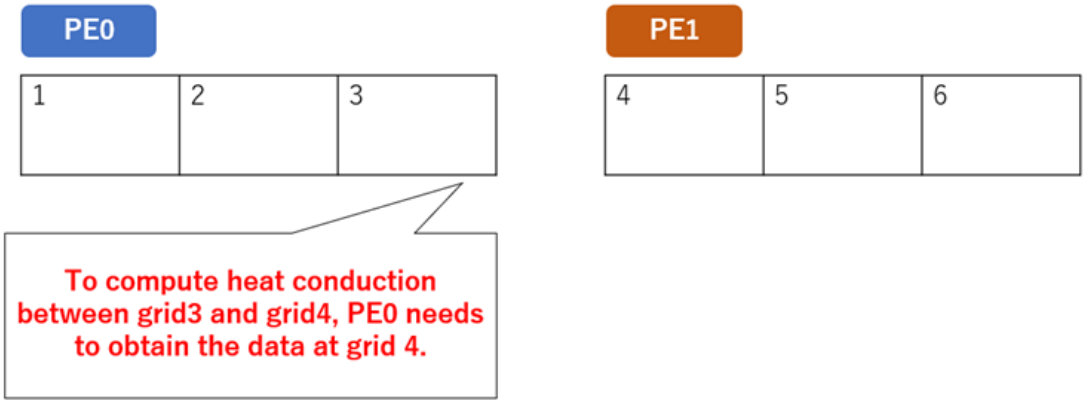

Figure 3.3: Point-to-Point communication: gridblock distribution of heat conduction problem.

3.2.2 Collective Communication

Collective communication is an operation of communication that involves all processors. Message-passing occurs from one processor to all processors or from all processors to all processors. As an

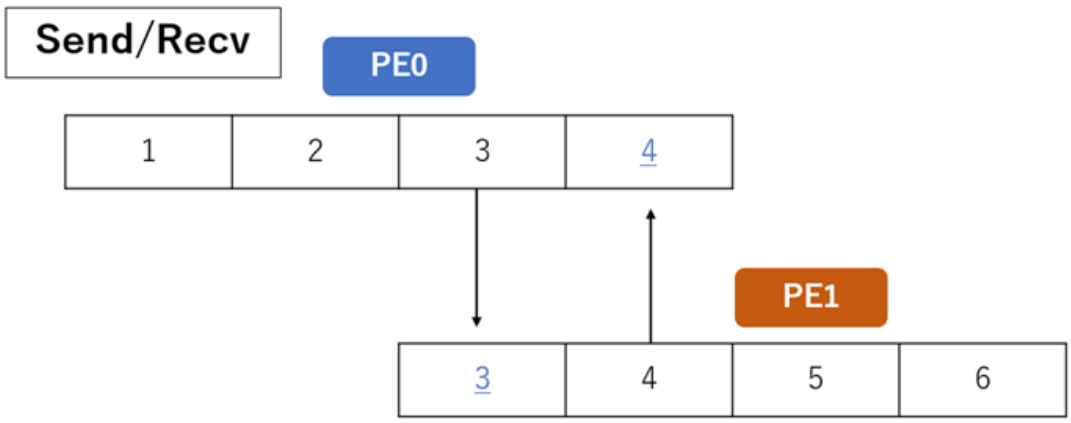

Figure 3.4: Point-to-Point communication: Sending and receiving (send/recv) operations.

example, the *Allreduce* operation is shown in **Fig. 3.5**. In this operation, all processors share the data and find the maximum value.

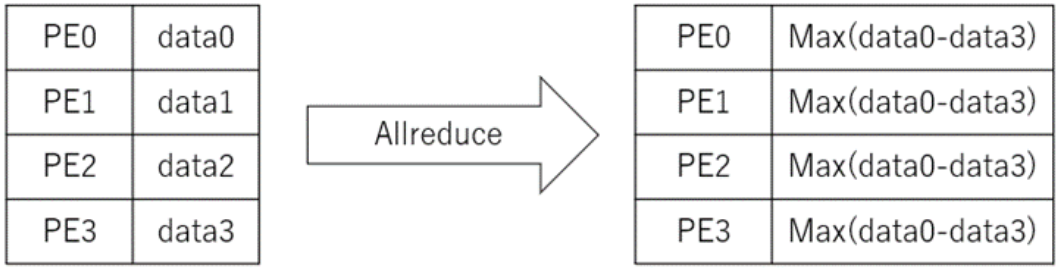

Figure 3.5: Collective communication (Allreduce).

3.3 MPI Related Libraries

The JFTS+H code developed in the course of this study is a MPI-based parallel simulator and uses two libraries to enhance its computational performance: the Library of Iterative Solvers (LIS) for linear system (Nishida, 2010) to solve the set of simultaneous equations of mass and heat balance, and METIS (Karypis, 2013) for a computationally efficient domain decomposition. LIS (Nishida, 2010) is a parallel software library for solving discretized linear equations using

iterative (preconditioned conjugate gradient) methods. Both MPI and openMP (Barney, 2021) versions of the available solvers are supported in this library, which offers multiple matrix solver, preconditioner, and matrix storage format options.

3.4 Analysis of Parallel Computing

Speed-up and efficiency are widely used metrics to analyze the performance of parallel computing. These variables are defined as:

$$S_P = \frac{T_1}{T_p} \tag{3.1}$$

$$E_p = \frac{S_P}{p} \tag{3.2}$$

where $\mathrm{S_p}$ is the speed-up on p processors [-], $\mathrm{E_p}$ is the efficiency on p processors [-], $\mathrm{T_1}$ is the execution time on a single processor [sec], $\mathrm{T_p}$ is the execution time on p processors [sec], and p is the number of processors. The speed-up and the efficiency are used to provide an analysis of how well a parallelized code sped up and generate a plot of the relationship between them and the number of processors to evaluate the behavior of the parallelized code.

4. THE JULIA PROGRAMMING LANGUAGE

Julia is new programming language that was developed by the Massachusetts Institute of Technology (MIT) and is designed for scientific computing. This programming language is dynamically typed like other modern programming languages such as Python, R, and MATLAB. This genre of dynamically-typed programming languages generally do not require the explicit declaration of variables before execution. This feature allows programmers to develop simple high-level codes without the need to specify types like integer or float that are declared in statically-typed languages such as C and Fortran. On the other hand, static languages have better performance at run-time because they compile and check types before execution, thus avoiding the need to check types dynamically while executing. In general, dynamically-typed programming languages are not designed for maximal performance. On the other hand, statically-typed programming languages are not designed for optimal productivity. The developers of Julia aimed to address this gap by designing a programming language that combines enhanced productivity with high performance (Bezanson, *et al*., 2017).

In addition to its performance and productivity, Julia is easily integrated with C/C++ and Fortran because of its built-in capability to call/invoke existing libraries and packages written in these languages. In this project, I integrated into my code the matrix solver library (LIS) that is written in Fortran.

Many dynamic languages like Python were not designed for high performance computing and have a "two-language problem". This is a type of problem that arises when researchers often develop a code with algorithms in a user-friendly language such as Python, but they need to rewrite the parts of the code that involve intensive computations in a faster language such as C/C++ or Fortran (Perkel, 2019). To solve this problem, libraries and all basic functionalities are implemented in Julia. This feature allows programmers to write high-level generic code that is as fast as static language.

4.1 Declaration and Recognition of Types

A program is composed of data and operations. The performance of operations is improving as the computer knows more about the data. Statically-typed programming languages are an extreme case, where all types are defined and statically determined while compiling the code. This leads to excellent performance. Dynamically-typed programming languages do not require type definition, thus leading to improved/enhanced productivity but lower performance, as the compiler checks the type definitions during runtime.

To overcome this issue of lower performance, Julia recognizes the data types more efficiently. An important component of Julia's ability to achieve a combination of performance and productivity is its implementation of dataflow type inference. The dataflow type inference algorithm allows codes to be automatically define data types without explicit specification by the programmers. In a most dynamic-typed programming languages, vectorization of DO loops is applied to achieve a better performance, but the unique feature of Julia language is that it achieves a faster "FOR loops" because of an efficient type inference system (Bezanson, *et al*., 2017).

Another distinguishing feature of Julia is that its user-defined types are treated as built-in types. Many dynamic languages for numerical computing show better performance with built-in types than with user-defined types. Conversely, there is no significant distinction between user-defined and built-in types in Julia.

Julia supports an enumerated type (Enum) which can hold one of a list of possible values. In this project, I used Enum types to describe phase identifiers. This feature enables type stability and fast computation.

4.2 Multiple Dispatch

Multiple dispatch is the capability to select a function implementation based on the type of each argument of the function. This contributes to a simpler code without a long list of case statements and, consequently, to the enhanced performance of Julia language. Multiple dispatch makes it easy to express many object-oriented and functional programming entities. This feature is

achieved by using type inference and just-in-time (JIT) compilation based on the implementation of the low-level-virtual-machine (LLVM) compiler infrastructure.

4.3 Libraries for Improving Computational Performance

4.3.1 MPI.jl (Schnetter, 2021)

This is a wrapper designed for the portable Message Passing Interface (MPI) (Blaise, 2019) based on the C MPI API. This wrapper calls MPI in the Intel Cluster Suite on the Ada cluster of the Texas A&M Supercomputer Center and openMPI on the MacPro used in this study.

4.3.2 Profiling.jl (Nash, 2019)

The profiling library provides metrics of the performance of a code through time measurements of the running code, detecting the time spent on individual lines and code segments. These data help developers improve the performance of the code and to identify "bottlenecks" in the programs. Profiling results can be produced as a figure by using a plotting library available in Julia. **Fig. 4.1** shows an example of profiling results. The horizontal axis shows the execution time, and the vertical axis shows the depth of function. The red color at the top layers show the codes which needs to be modified. In this study, I checked this type of profiling results and found bottlenecks.

4.4 Features for Better Readability

4.4.1 YAML (Jones, 2013)

YAML is a flexible data format designed to enable easy/intuitive reading and writing. Julia can obtain any data type, such as integer, float, string, array, and bool, from the YAML format. **Fig. 4.2** shows an example of input data using the YAML format.

4.4.2 Unicode

Unicode characters can be entered in the Julia REPL (read-eval-print-loop). In this project, I used this feature to describe thermophysical properties like density (ρ) and viscosity (μ).

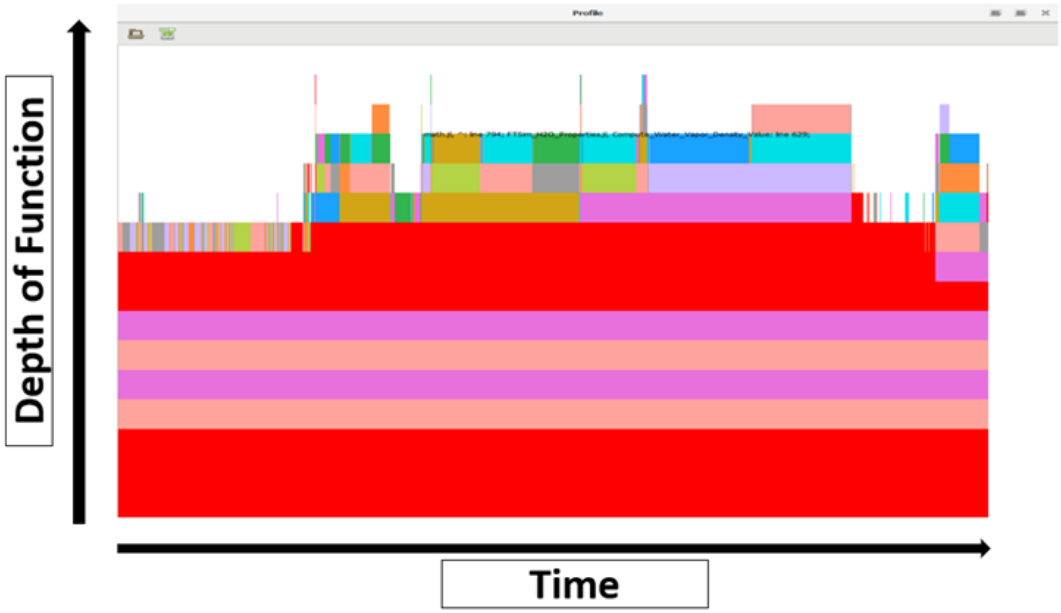

Figure 4.1: Profiling results. In this example, the input data is Test_2D of Moridis (2014).

```
MEDIA_PROEPRTIES:
    MEDIA1:
        Name:                   L01HH
        DensG:                  2650.0
        Poros:                  0.45
        Perm:                   [1.784e-13, 1.784e-13, 1.784e-13]
        KThrW:                  1.45
        SpcHt:                  800.0
        PoMedRGrain:            0.0
        Compr:                  1.0e-8
        Expan:                  0.0
        KThrD:                  1.0
```

Figure 4.2: Input data in the YAML format.

5. THE SIMULATOR AND ITS VALIDATION

The JFTS+H hews closely to the approach and architecture of the TOUGH+ family codes (Moridis, *et al.,* 2014; Moridis, 2014), which has been used for a wide range of applications in problems of flow and transport of fluids and heat through porous and fractured geologic media (Moridis and Freeman, 2014; Lee *et al.,* 2018a. b; Moridis *et al.,* 2019a,b; Reagan *et al.,* 2015; 2019; Zhang and Moridis, 2011). For equilibrium hydrate dissociation/formation, the simulator describes the accumulation and transport of heat and two or three mass components: $\mathrm{H_2O}$, $\mathrm{CH_4}$ and a water-soluble inhibitor (optional). If the hydrate dissociation/formation reaction is described by a kinetic process, the number of components increases by one by the addition of the $\mathrm{CH_4}$-hydrate as a distinct, separate component (as opposed to the equilibrium reaction, in which the $\mathrm{CH_4}$-hydrate is not a component but is just a state of the $\mathrm{H_2O}$-$\mathrm{CH_4}$ system). The mass components in JFTS are distributed among 4 possible phases, that is, hydrate, aqueous, gaseous, and ice. The number of primary variables does not change during the simulation, but the thermophysical quantities used as primary variables can change to allow for the seamless consideration of the emergence or disappearance of phases occurring as the thermodynamic states at different parts of the domain change in response to changes in pressure and/or temperature (Moridis, 2014; Moridis *et al.,* 2019a).

5.1 Mass and Energy Balance Equation

The mass balance equation of any component κ in any domain or subdomain of arbitrary volume and geometry is described by the general integrodifferential equation

$$\frac{d}{dt}\int_{V_n} M^\kappa dV = \int_{\Gamma_n} \mathbf{F}^\kappa \cdot \mathbf{n} d\widetilde{A} + \int_{V_n} q^\kappa dV \tag{5.1}$$

in which V_n is the volume of subdomain [m^3], M^κ is the mass accumulation term of component κ [kg/m^3], A is the surface area [m^2], Γ_n is the surface area of the subdomain [m^2], F^κ is the Darcy flux vector of component κ [$kg\ m^{-2}s^{-1}$] through the domain boundaries, n is the inward unit normal vector at any point on the outer surface of the domain, q^κ is the source/sink term of

component κ [$kg\ m^{-3}\ s^{-1}$], and t is the time [sec].

5.1.1 Mass Accumulation Terms

JFTS can describe hydrate formation and dissociation using either an equilibrium model or a kinetic model. The equilibrium model assumes that hydration formation or dissociation reaction occurs instantaneously (i.e., without a kinetic delay) and equilibrium is attained immediately. In other words, this model assumes that the rates of reactions are extremely fast, which allows the CH_4-hydrate to be treated not as a separate species (chemical compound) but as just a state of the H_2O+CH_4 system.

Under equilibrium condition, the mass accumulation term of any component κ is given by

$$\sum_{\beta=A,\ G,\ I,\ H} \phi S_\beta \rho_\beta X_\beta^\kappa, \quad \kappa = w,\ m,\ i \tag{5.2}$$

where ϕ is the porosity [-], ρ_β is the density of phase β [$kg\ m^{-3}$], and X_β^κ is the mass fraction of component κ in phase β [kg/kg]. The possible components κ are H_2O, CH_4 and inhibitor (denoted by w, m and i, respectively), distributed among four possible phases β: aqueous, gas, ice and hydrate (denoted by A, G, I and H, respectively). Because earlier studies have shown conclusively that hydrate formation or dissociation are very fast reactions, application of the equilibrium model is a valid approach in long-term studies (Kowalsky and Moridis, 2007).

In the equilibrium model, the following conditions and constraints apply:

$$\beta = G: \quad X_G^i = 0 \tag{5.3}$$

$$\beta = H: X_H^w = \frac{W^m}{W^h}, \quad X_M^m = 1 - X_H^w, \quad X_H^i = 0 \tag{5.4}$$

$$\beta = I: X_I^m = X_I^i = 0, \quad X_I^w = 1 \tag{5.5}$$

where W^m and W^h [kg/mol] are the molecular weights of the CH_4-hydrate and H_2O, respectively.

The kinetic model is more conceptually correct and is applicable to any short- and long-term processes but is also more complex to apply and more computationally demanding. Under kinetic

conditions, the CH_4-hydrate is not a mere state of the H_2O+CH_4 system but is considered as a separate component. The mass accumulation term is then given by

$$\sum_{\beta=A,\ G,\ I,\ H} \phi S_\beta \rho_\beta X_\beta^\kappa, \quad \kappa = w,\ m,\ h,\ i \tag{5.6}$$

In Eq. 5.6, the term κ = h denotes the CH_4-hydrate species, i.e., the hydrate is both a chemical compound and a solid phase. In the kinetic model, the following conditions and constraints apply:

$$\beta = A: \quad X_A^h = 0 \tag{5.7}$$

$$\beta = G: \quad X_G^h = X_G^i = 0 \tag{5.8}$$

$$\beta = H: X_H^w = X_H^m = X_H^i = 0, \quad X_H^h = 1 \tag{5.9}$$

$$\beta = I: X_I^m = X_I^h = X_I^i = 0, \quad X_I^w = 1 \tag{5.10}$$

The equation describing the kinetic behavior of the hydrate formation or decomposition reaction is described by the model of Kim *et al.* (1987) as

$$Q_H = \frac{\partial M}{\partial t} = -K_0 \exp\left(\frac{\Delta E_a}{RT}\right) F_A A \left(f_{eq} - f_v\right) \tag{5.11}$$

where K_0 is the intrinsic hydration reaction constant [$kg\ m^{-2}\ Pa^{-1}\ s^{-1}$], ΔE_a is the hydration activation energy [$J\ mol^{-1}$], R is the universal gas constant [$J\ mol^{-1}\ K^{-1}$], T is the temperature [K], F_A is the area adjustment factor [dimensionless], A is the surface area participating in the reaction [m^2], f_{eq} is the fugacity [Pa] at the equilibrium temperature T_{eq} corresponding to the gas pressure P, and f_v is the fugacity of the gas phase at the actual temperature T [Pa].

The hydrate reactive area A is computed using the equation of Moridis (2014) as

$$A = f_A N_v \left(4\pi r_p^2\right) S_H^{\frac{2}{3}} \tag{5.12}$$

where S_H is the hydration saturation and N_v is the number of voids computed from

$$N_v = \frac{(1-\phi)}{V_p} \tag{5.13}$$

and r_p is the average radius of the porous medium grains, estimated using the Kozeny-Carman equation (Bear, 1972) as:

$$r_p = \left[45k\frac{(1-\phi)^2}{\phi^3}\right]^{\frac{1}{2}} \tag{5.14}$$

with k the intrinsic permeability [m^2].

5.1.2 Heat Accumulation Terms

The heat accumulation term includes contributions from all the phases, the rock matrix and from the enthalpy of the reaction, and is given by

$$M^\theta = \int_{T_0}^{T} (1-\phi)\,\rho_R C_R dT + \sum_{\beta=A,\ G,\ H,\ I} \phi S_\beta \rho_\beta X_\beta U_\beta + Q_{diss} \tag{5.15}$$

where ρ_R is the rock density [$kg\ m^{-3}$], C_R is the heat capacity of the dry rock [$J\ kg^{-1}\ K^{-1}$], and U_β is the specific internal energy of phase β [$J\ kg^{-1}$].

The specific internal energy of the gaseous phase is given by

$$U_G = \sum_{\kappa=w,\ m} X_G^\kappa u_G^\kappa + U_{dep} \quad \left(= H_G - \frac{P}{\rho_G}\right) \tag{5.16}$$

where u_G^κ is the specific internal energy of component κ in the gaseous phase, and U_{dep} is the specific internal energy departure of the gas mixture [$J\ kg^{-1}$]. The enthalpy of the hydrate dissociation/formation reaction is computed as (Moridis, 2014)

$$\Delta H_{diss} = \begin{cases} \Delta\left(\phi\rho_H S_H\right)\Delta H^0 & \text{for equilibrium formation/dissociation} \\ Q_H \Delta H^0 & \text{for kinetic formation/dissociation} \end{cases} \tag{5.17}$$

where ρ_H is the density of the hydrate phase and ΔH^0 is the specific enthalpy of hydrate dissociation/formation. Under the corresponding three-phase conditions (involving coexistence of CH_4-hydrate, gas and aqueous or ice phases), ΔH^0 is computed from the simple equation of Kamath [1984] as

$$\Delta H^0 = C_f \left(C_1 + \frac{C_2}{T} \right) \tag{5.18}$$

In Eq. 5.18, the conversion factor is $C_f = 33.72995 \left(\frac{\text{J}}{\text{kg}} \right) / (\frac{\text{cal}}{\text{gmol}})$.

In this study, the following equation is used to evaluate the temperature depression induced by the inhibitor, (Moridis, 2014):

$$\Delta T_D = \Delta\, T_{D,r} \frac{\ln\left(1 - Y_A^i\right)}{\ln\left(1 - Y_{A,r}^i\right)} \tag{5.19}$$

where Y_A^i is the mole fraction of the inhibitor in the aqueous phase [-], $Y_{A,r}^i$ is the reference mole fraction of the inhibitor in the aqueous phase[-], ΔT_D is the inhibitor-induced temperature depression [K], and $\Delta T_{D,r}$ is the temperature depression at the reference mole fraction [K].

The specific internal energy of the gaseous phase is given by

$$U_G = \sum_{\kappa=w,\, m} X_G^\kappa u_G^\kappa + U_{dep} \quad \left(= H_G - \frac{P}{\rho_G} \right) \tag{5.20}$$

where u_G^κ is the specific internal energy of component κ in the gaseous phase and U_{dep} is the specific internal energy departure of the gas mixture [$J\ kg^{-1}$].

The internal energy of the aqueous phase is given by

$$U_A = X_A^w u_A^w + X_A^m \left(u_A^m + u_{sol}^m \right) + X_A^i \left(u_A^i + u_{sol}^i \right) \tag{5.21}$$

where u_A^w, u_A^m, and u_A^i are the specific internal energy of H_2O, CH_4 and the inhibitor, respectively, and u_{sol}^m and u_{sol}^i are the specific internal energies of dissolution of CH_4 and of the inhibitor in the aqueous phase, respectively [$J\ kg^{-1}$].

The terms u_A^i and U_H are determined from

$$u_A^i = h_A^i - \frac{P}{\rho_i} = \int_{T_0}^{T} C_i dT - \frac{P}{\rho_i} \tag{5.22}$$

$$U_H = H_H - \frac{P}{\rho_H} = \int_{T_0}^{T} C_H dT - \frac{P}{\rho_H} \tag{5.23}$$

where T_0 is a reference temperature [K], h_A^i and H_H are the specific enthalpies of H_2O and hydrate (a phase in equilibrium model, and both a phase and a component in kinetic model), respectively, and C_i and C_H are the temperature-dependent heat capacities of the inhibitor and the gas hydrate, respectively [$J\ kg^{-1}\ K^{-1}$].

5.1.3 Flux Terms

The mass fluxes of F^κ of H_2O, CH_4 and inhibitor in the aqueous and gaseous phases are represented as

$$F^\kappa = \sum_{\beta=A,\ G} X_\beta^\kappa F_\beta, \quad \kappa = w,\ m,\ i \tag{5.24}$$

For obvious reasons, the contributions of the two immobile solid phases ($\beta = H,\ I$) to the fluid fluxes are zero, as is the mass flux of the hydrate component ($\kappa = h$) in the kinetic model.

The phase flux F_A of the aqueous phase is described by Darcy's law

$$F_A = \rho_A \left[-\frac{k k_{rA}}{\mu_A} (\nabla P_A - \rho_A g) \right] \tag{5.25}$$

where k_{rA} is the relative permeability of the aqueous phase [dimensionless], P_A is the phase pressure [Pa], μ_A is the aqueous phase viscosity [Pa.s], and g is the gravitational acceleration vector [$m\ s^{-2}$].

The aqueous pressure P_A is related to the gas pressure P_G through the equation

$$P_A = P_G - P_{cCW} \tag{5.26}$$

where P_{cGW} is the gas-water capillary pressure [Pa], and the gas pressure P_G [Pa] is the sum of the partial pressures of the CH_4 and of the H_2O vapor in the gas phase. The CH_4 solubility in the aqueous phase is described by Henry's law as

$$P_G^m = H^m\,(T)\,Y_A^m \tag{5.27}$$

where $H^m\ (T)$ is the T- and salinity- dependent Henry's coefficient.

The gas phase flux F_G of the gas phase is described by

$$F_G^\kappa = -k_o\left(1+\frac{b}{P_G}\right)\frac{k_{rG}\rho_G}{\mu_G}X_G^\kappa\left(\nabla P_G-\rho_G g\right)+J_G^\kappa,\quad \kappa\equiv\ w,\ m \tag{5.28}$$

where k_o is the absolute permeability at large gas pressures or in liquid flow [m^2], b is the Klinkenberg [1941] b-factor that accounts for gas slippage effects [Pa], k_{rG} is the relative permeability of the gas phase [dimensionless], μ_G is the gas phase viscosity [Pa.s] and J_G^κ is the diffusive flux of component κ in the gas phase (Moridis, 2014).

The heat flux equation accounts for conduction, advection, and radiative heat transfer, and is given by

$$F^\theta = -\bar{k}_\theta\nabla T+\sum_{\beta=A,\ G}h_\beta F_\beta \tag{5.29}$$

where $\bar{k}_\theta$ is the composite thermal conductivity of the rock-fluids ensemble [$W\ m^{-1}K^{-1}$] and h_β is the specific enthalpy of phase β [$J\ kg^{-1}$]. The term $\bar{k}_\theta$ in hydrate-bearing media is estimated from the equation in (Moridis, 2014) as

$$\bar{k}_\theta = k_{\theta d}+\left(\sqrt{S_A}+\sqrt{S_H}\right)\left(k_{\theta w}-k_{\theta d}\right)+\phi\ S_I k_{\theta I} \tag{5.30}$$

where $k_{\theta I}$ is the thermal conductivity of the ice, $k_{\theta w}$ is the thermal conductivity of the water-saturated medium, and $k_{\theta d}$ is the thermal conductivity of the dry porous medium.

The specific enthalpy of the gas phase is computed as

$$H_G = \sum_{\kappa=w,\ m} X_G^\kappa h_G^\kappa + H_{dep} \tag{5.31}$$

where h_G^κ is the specific enthalpy of component κ in the gaseous phase and H_{dep} is the departure enthalpy of the gas mixture [$J\ kg^{-1}$].

The specific enthalpy of the aqueous phase is computed in a manner entirely analogous to that of Eq. 5.21 as

$$H_A = X_A^w h_A^w + X_A^m \left(h_A^m + h_{sol}^m\right) + X_A^i \left(h_A^i + h_{sol}^i\right) \tag{5.32}$$

where h_A^κ is the specific enthalpy of component $\kappa(= w, m, i)$ in the aqueous phase [$J\ kg^{-1}$] and h_{sol}^κ is the enthalpy of solution of component κ in the aqueous phase.

5.1.4 Sinks and Source Terms

In sources (production wells) with a specified mass production rate, the withdrawal of mass component κ is described by

$$\hat{q}^\kappa = \sum_{\beta=A,\ G} X_\beta^\kappa q_\beta, \quad \kappa = w,\ m \tag{5.33}$$

where q_β is the mass production rate of the mobile phase β [kg/m^3].

In sink terms (injection wells), the addition of a mass component κ occurs at desired (and known) rates that are used as input data. For the kinetic model of the hydrate reaction, additional sink/source terms corresponding to the hydrate formation or dissociation (and release of $\mathrm{CH_4}$ and $\mathrm{H_2O}$) are included in each subdomain (element) into which the domain is subdivided. Then, the source term for $\mathrm{CH_4}$ becomes $\hat{q}^m = Q^m$, where the production rate Q^m [$kg\ m^{-3}s^{-1}$] of $\mathrm{CH_4}$ is computed as

$$Q^m = -\frac{W^m}{W^h} Q_H \tag{5.34}$$

Similarly, the hydrate-related release of water Q^w is determined from the stoichiometry equation.

$$Q^w = -\frac{N_H W^w}{W^h} Q_H \tag{5.35}$$

Under equilibrium conditions, the rate of heat removal or addition includes contributions of fluid removal or additions, as well as direct heat inputs or withdraws q_d

$$\hat{q}^\theta = q_d + \sum_{\beta=A,\ G} h_\beta q_\beta \tag{5.36}$$

Under kinetic conditions, the rate of heat removal or addition is determined from

$$\hat{q}^\theta = q_d + \sum_{\beta=A,\ G} h_\beta q_\beta + Q_H \Delta H^0 \tag{5.37}$$

5.2 Thermophysical Properties

The main thermophysical properties of pure water in any state are described by the fast regression equations of IAPWS97 (2007; 2008; 2009; 2011a,b; 2012), and of Duan and Mao (2009). These equations provide accurate estimates of density, viscosity, enthalpy and thermal conductivity of any state of the H_2O substance (liquid, vapor or ice) as a function of *P* and *T*, in addition to the equations describing the 3-phase coexistence lines in the H_2O+CH_4 phase diagram.

For brines, the density and enthalpy are computed using the relationships of Drisner and Heinrich (2009), and the vapor pressure is provided by Atkinson (2002) and Hass (1976). The thermal conductivity of a brine is computed from the equation of Aleksandrov *et al.* (2013) as a function of molality, *P* and *T*.

The main properties of the gas phase are computed by the Peng and Robinson (1976) cubic equation of state, which provides inputs for the computation of the gas viscosity and thermal conductivity by the methods of Chung *et al.* (1988). Gas solubility in water or brine is computed by a set of temperature-dependent Henry's coefficients.

The specific enthalpy of the solid hydrate is computed from the general equation

$$H_H = \int_{T_{0H}}^{T} C_H dT \tag{5.38}$$

where T_{0H} is a reference temperature and $\mathrm{C_H}$ is the heat capacity of the $\mathrm{CH_4}$ hydrate.

The density of hydrate is provided by the Ballard equation (2002)

$$\rho_H = \left[v_s \exp\left(\alpha_1 \Delta T + \alpha_2 \Delta T^2 + \alpha_3 \Delta T^3 + \alpha_4 \Delta P\right)\right]^{-1} \tag{5.39}$$

where ΔT is $T - T_0$ (T_0 is the reference temperature), ΔP is $P - P_0$ (P_0 is the reference pressure), v_s is the hydrate specific volume = $\dfrac{1000 M_H}{22.712 N_H}$ and M_H is the molecular weight of the hydrate [kg/mol]. The coefficients of Eq. 5.39 are $\alpha_1 = 3.38496 \times 10^{-4}$, $\alpha_2 = 5.40099 \times 10^{-7}$, $\alpha_3 = -4.76946 \times 10^{-11}$, and $\alpha_4 = 1.0 \times 10^{-10}$.

5.3 The $\mathrm{H_2O} + \mathrm{CH_4}$ Phase Diagram and Hydrate Phase Coexistence

Using all available data from literature sources, the *P vs. T* relationships along the 3-phase A-H-V (H_{curve}) and I-H-V (L_{curve}) state equilibrium lines are calculated by following equations:

$$H_{curve} = H_1 + H_2 T_c + H_3 T_C^2 + H_4 T_C^3 + H_5 T_C^4 + H_6 T_C^5 + H_7 T_C^6 \tag{5.40}$$

$$L_{curve} = L_1 + L_2 T_c + L_3 T_C^2 + L_4 T_C^3 + L_5 T_C^4 + L_6 T_C^5 + L_7 T_C^6 + L_8 T_C^7 \tag{5.41}$$

where $T_c = T - 274.13$ is the temperature in °C. The coefficients of Eqs. 5.40 and 5.41 are given in **Table 5.1**.

To provide continuity of the two distinctly different curves and eliminate the problems posed by the presence of an inflection point at their intersection at the quadruple point and the associated inevitable discontinuity of the derivatives at this point, the vicinity of the junction of the two functions was smoothed by means of the following equation that involves the properties of the

hyperbolic tangents:

$$f(T) = 0.50542(1.0 + \tanh\left(0.4(273.16 - T)\right)\ )\ L_{curve}\ (T) \\ + 0.5(1.0 + \tanh\left(0.4(T - 273.16)\right)\ )\ H_{curve}\ (T) \quad (5.42)$$

The resulting continuous equation yields the correct pressure and temperature at the quadruple point and ensures continuity of the derivatives, a necessity in the Jacobian-based formulation of the JFTS simulator. The I-A-H 3-phase state line in JFTS+H is based on the IAPWS (2007) equation for the I-A 2-phase state line for pure water, and is described by the following relationship:

$$P = P_Q - 6.26 \times 10^5 \left(1.0 - T_d^{-3}\right) + 1.97135 \times 10^5 \left(1.0 - T_d^{21}\right) \quad (5.43)$$

where T_d is T/273.15 [K] and $\mathrm{P_Q}$ is the pressure at the $\mathrm{CH_4}$-hydrate quadruple point. The I-A-V 3-phase state line is described by a constant temperature line at $\mathrm{T_Q}$ = 273.16 K. The complete hydrate phase diagram is shown in **Fig. 5.1**.

Table 5.1: Coefficients for the A-H-V line and I-H-V line (Eqs. 5.40 and 5.41).

H_1	$9.7115763331500005\times10^{-1}$
H_2	$9.5219324937155264\times10^{-2}$
H_3	$4.9359488568130128\times10^{-4}$
H_4	$-1.5393095521139555\times10^{-5}$
H_5	$3.9601127401297287\times10^{-6}$
H_6	$-1.6415280580462842\times10^{-7}$
H_7	$1.8011013154736077\times10^{-9}$
L_1	$9.7115763331500005\times10^{-1}$
L_2	$5.7051179211755162\times10^{-2}$
L_3	$-9.0738509254725073\times10^{-3}$
L_4	$1.5404236043787813\times10^{-4}$
L_5	$-1.1481265601708788\times10^{-6}$
L_6	$4.5036705257558079\times10^{-9}$
L_7	$-9.1461922856893920\times10^{-12}$
L_8	$7.6164185888971483\times10^{-15}$

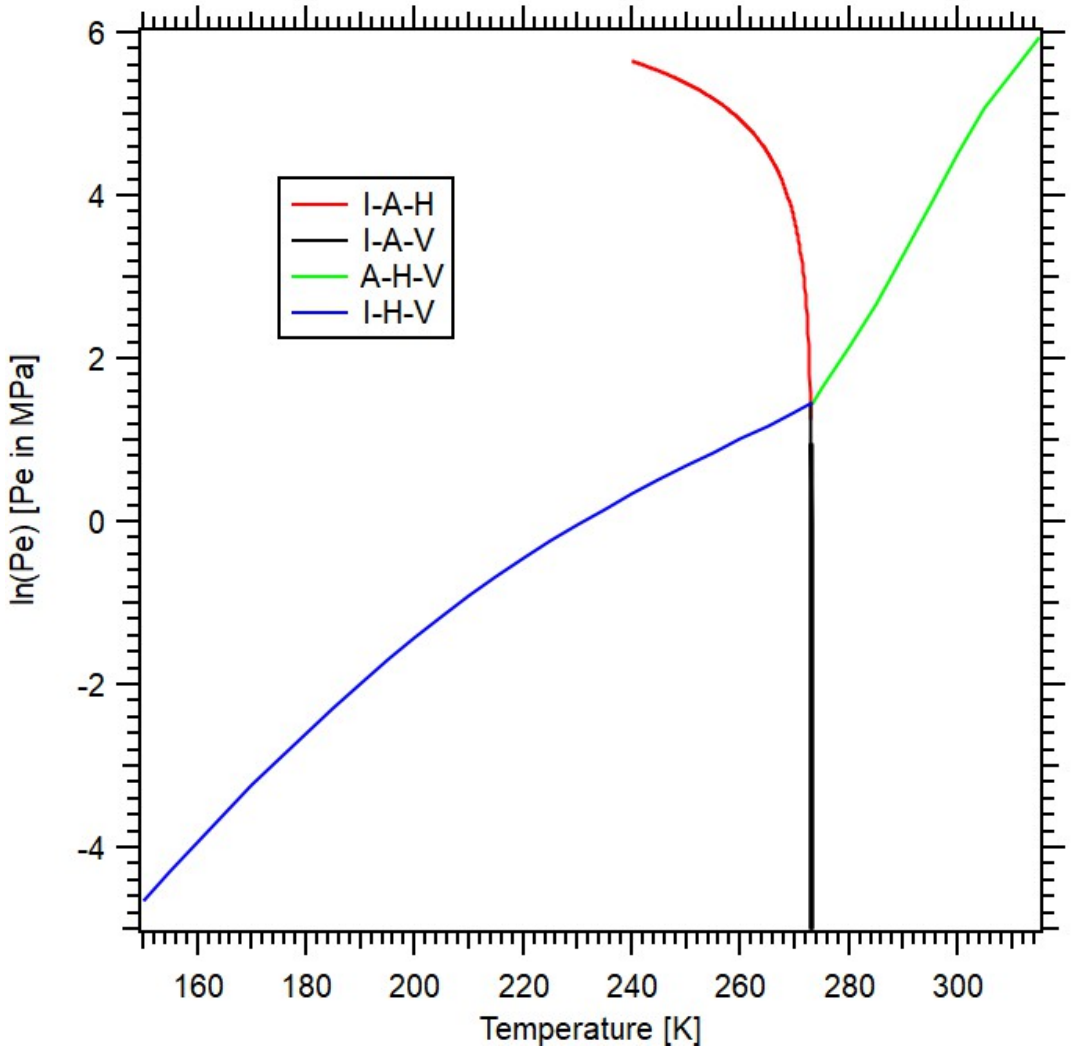

Figure 5.1: Phase diagram of the $H_2O + CH_4$ system, and P-T equilibrium relationships of the 3-phase coexistence states.

5.4 Effects of Solid Phase on Porosity and Permeability

JFTS+H accounts for the effects of the solid phases (such as hydrate and ice) on the porosity and permeability of the porous medium. The porosity is computed as

$$\phi = \phi_o F_{PT} \tag{5.44}$$

where ϕ_o is the original porosity [dimensionless].

$$F_{PT} = \frac{\phi}{\phi_0} = \exp\left[\alpha_P \Delta P + \alpha_T \Delta T\right] \approx 1 + \alpha_P \Delta P + \alpha_T \Delta T \tag{5.45}$$

α_P is the pore compressibility [1/Pa] and α_T is the thermal expansivity [1/K].

The effect of the variable S_H on the pore compressibility is described by the equation listed in Moridis (2014) as

$$\alpha_P = \exp\left\{\ln\left(\alpha_{PL}\right) + \left(\ln\left(\alpha_{PU}\right) - \ln\left(\alpha_{PL}\right)\right)\left[1 - B_x\left(2.25,\ 2.25,\ S_S^*\right)\right]\right\} \tag{5.46}$$

$$S_H^* = \frac{S_H - S_{Hmin} + \delta}{S_{Hmax} - S_{Hmin} + 2\delta} \tag{5.47}$$

where S_H is the solid hydrate saturation [-], α_{PL} is the lower limit of the medium compressibility [1/Pa] computed at S_H=S_{Hmax}, α_{PU} is the upper limit of the pore compressibility [1/Pa] computed at $S_H = S_{Hmin}$, δ is a smoothing factor [-], and B_x is the incomplete beta function. The relationship between $\mathrm{S_H}$ and α_P is shown in **Fig. 5.2**.

The effective permeability is evaluated using the "Original Porous Medium" (OPM) model (Moridis 2014). This model assumes that (a) porosity is not affected by the emergence of hydrate and/or ice, (b) the intrinsic permeability of the porous media is not affected by the evolution of the solid phases, although it can change with variations in the pressure, and (c) the fluid flow as a relative permeability issue is controlled by the saturations of the various phases in the pores. Thus, the intrinsic permeability is computed as

$$k = k_0 F_{r\phi} \tag{5.48}$$

where k_0 is the original intrinsic permeability [m^2],

$$k_{r\phi} = \begin{cases} 1 & \text{the effect of } \phi \text{ changes on k is neglected} \\ \exp\left[\gamma\left(F_{PT} - 1\right)\right] & \text{the effect of } \phi \text{ chages on k is accounted for} \end{cases} \tag{5.49}$$

and γ is the empirical parameter (Rutqvist and Tsang, 2002).

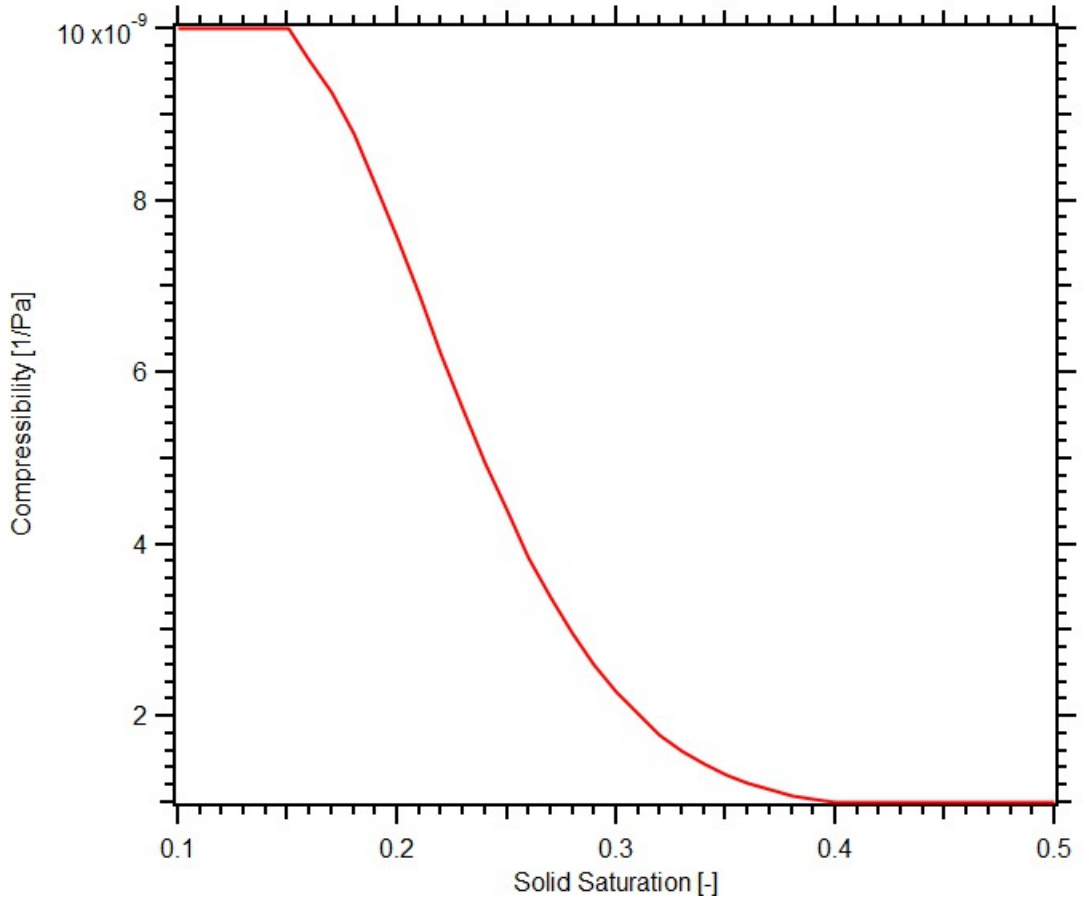

Figure 5.2: Compressibility of an unconsolidated porous medium impregnated with cementing solid phase. In this example, S_{Hmin} = 0.15, S_{Hmax} = 0.4, $\alpha_{PU} = 10^{-8} Pa^{-1}$, $\alpha_{PL} = 10^{-9} Pa^{-1}$ and δ = 0.015 (Moridis *et al.,* 2008; 2009; 2011).

In the OPM model used in JFTS+H, the effective permeability $k_{eff} = kk_{r\beta}(\beta = A, G)$ that is used in the estimation of the phase fluxes in Eqs. 5.25 and 5.28 involves an assumption that the relative permeabilities of the mobile phases are functions of their saturations, i.e., the presence of the solid phases does not affect the intrinsic permeability of the hydrate-free porous medium. In JFTS+H, the relative permeabilities of gas and aqueous phases can be obtained by one of several available options, including the models of Corey (1964), Stone (1974), vanGenucthen (1980) and Parker *et al.* (1987).

5.5 Primary Variable and Phase Change

The number of primary variables does not change during simulation, but the thermodynamics quantities used as primary variables can change in the process of simulation to allow for the seamless consideration of emergence or disappearing phases (Moridis, 2014). The primary variables in each phase can be shown in **Table 5.2** (Equilibrium model) and **Table 5.3** (Kinetic model). Possible phase transitions are shown in **Figure 5.3**.

5.6 Numerical Discretization

The continuum Eq. 5.1 is discretized in space using the integral finite difference method (IFDM) of Edwards (1972) and Narasimhan and Witherpoon (1976), which assumes that all the properties and conditions are uniform within each subdivion (element) of the discretized domain and are assigned to its centroid. Thus, the left-hand side of Eq. 5.1 becomes

$$\int_{V_n} M dV = V_n M_n^\kappa \tag{5.50}$$

where M_n^κ is the accumulation of any mass component κ of over the volume V_n of element n. The principles of IFDM hold that the line connecting the centroids of two elements m and n be perpendicular to their common interface. This leads to the approximation of the surface integrals by a discrete algebraic sum of fluxes over each surface segment A_{nm} of the boundary (outer surface) of element n, i.e.,

$$\int_{\Gamma_n} \boldsymbol{F}^\kappa \cdot \boldsymbol{n} d\Gamma = \sum_m A_{nm} F_{nm} \tag{5.51}$$

where F_{nm} is the average value of the normal component of $\boldsymbol{F}$ over the surface segment A_{nm} between volume elements V_n and V_m. The discretized flux is expressed in terms of appropriate averages of all the parameters for elements V_n and V_m.

$$F_{\beta,nm} = -k_{nm} \left[\frac{k_{r\beta}\rho_\beta}{\mu_\beta} \right]_{nm} \left[\frac{P_{\beta,n} - P_{\beta,m}}{D_{nm}} - \rho_{\beta,nm} g_{nm} \right] \tag{5.52}$$

Table 5.2: Primary variables in the JFTS+H equilibrium model.

Phase	**State Identifier**	**PV1**	**PV2**	**PV3**
1-Phase: G	**Gas**	P_gas	Y_m_G	T
1-Phase: A	**Aqu**	P	X_m_A	T
2-Phase: A+G	**AqG**	P_gas	S_aqu	T
2-Phase: I+G	**IcG**	P_gas	S_ice	T
2-Phase: H+G	**GsH**	P_gas	S_gas	T
2-Phase: A+H	**AqH**	P	S_aqu	T
2-Phase: A+I	**AqI**	P	S_aqu	X_m_A
2-Phase: I+H	**IcH**	P	S_ice	T
3-Phase: A+H+G	**AGH**	P_gas	S_aqu	T
3-Phase: A+I+G	**AIG**	P_gas	S_aqu	S_gas
3-Phase: A+I+H	**AIH**	P	S_aqu	S_ice
3-Phase: I+H+G	**IHG**	S_gas	S_ice	T
Quadruple Point:	**QuP**	S_gas	S_aqu	S_ice

The possible primary variables are: P, pressure [Pa]; P_gas, gas pressure [Pa]; T, temperature [C]; X_m_A, mass fraction of CH_4 dissolved in the aqueous phase [-]; Y_m_G, mass fraction of CH_4 in the gas phase [-]; S_aqu, aqueous phase saturation [-]; S_gas, gas saturation, [-]; S_hyd, hydrate saturation [-]; S_ice, ice saturation [-]; X_i_A, mass fraction of an inhibitor in the aqueous phase [-]. If an inhibitor is present, X_i_A becomes the 3^{rd} primary variable, and the 3^{rd} primary variable becomes the 4^{th} primary variable.

Table 5.3: Primary variables in the JFTS+H kinetic model.

Phase	**State Identifier**	**PV1**	**PV2**	**PV3**	**PV4**
1-Phase: G	**Gas**	P_gas	Y_m_G	S_hyd	T
1-Phase: A	**Aqu**	P	X_m_A	S_hyd	T
2-Phase: A+G	**AqG**	P_gas	S_aqu	S_hyd	T
2-Phase: I+G	**IcG**	P_gas	S_ice	S_hyd	T
2-Phase: H+G	**GsH**	P_gas	S_gas	S_hyd	T
2-Phase: A+H	**AqH**	P	S_aqu	X_m_A	T
2-Phase: A+I	**AqI**	P	S_aqu	S_gas	X_m_A
2-Phase: I+H	**IcH**	P	S_ice	S_gas	T
3-Phase: A+H+G	**AGH**	P_gas	S_aqu	S_gas	T
3-Phase: A+I+G	**AIG**	P_gas	S_aqu	S_hyd	S_gas
3-Phase: A+I+H	**AIH**	P	S_aqu	S_ice	T
3-Phase: I+H+G	**IHG**	P_gas	S_gas	S_ice	T
Quadruple Point:	**QuP**	P_gas	S_aqu	S_gas	S_ice

The possible primary variables are: P, pressure [Pa]; P_gas, gas pressure [Pa]; T, temperature [C]; X_m_A, mass fraction of CH_4 dissolved in the aqueous phase [-]; Y_m_G, mass fraction of CH_4 in the gas phase [-]; S_aqu, aqueous phase saturation [-]; S_gas, gas saturation, [-]; S_hyd, hydrate saturation [-]; S_ice, ice saturation [-]; X_i_A, mass fraction of an inhibitor in the aqueous phase [-]. If an inhibitor is present, X_i_A becomes the 4^{rd} primary variable, and the 4^{rd} primary variable becomes the 5^{th} primary variable.

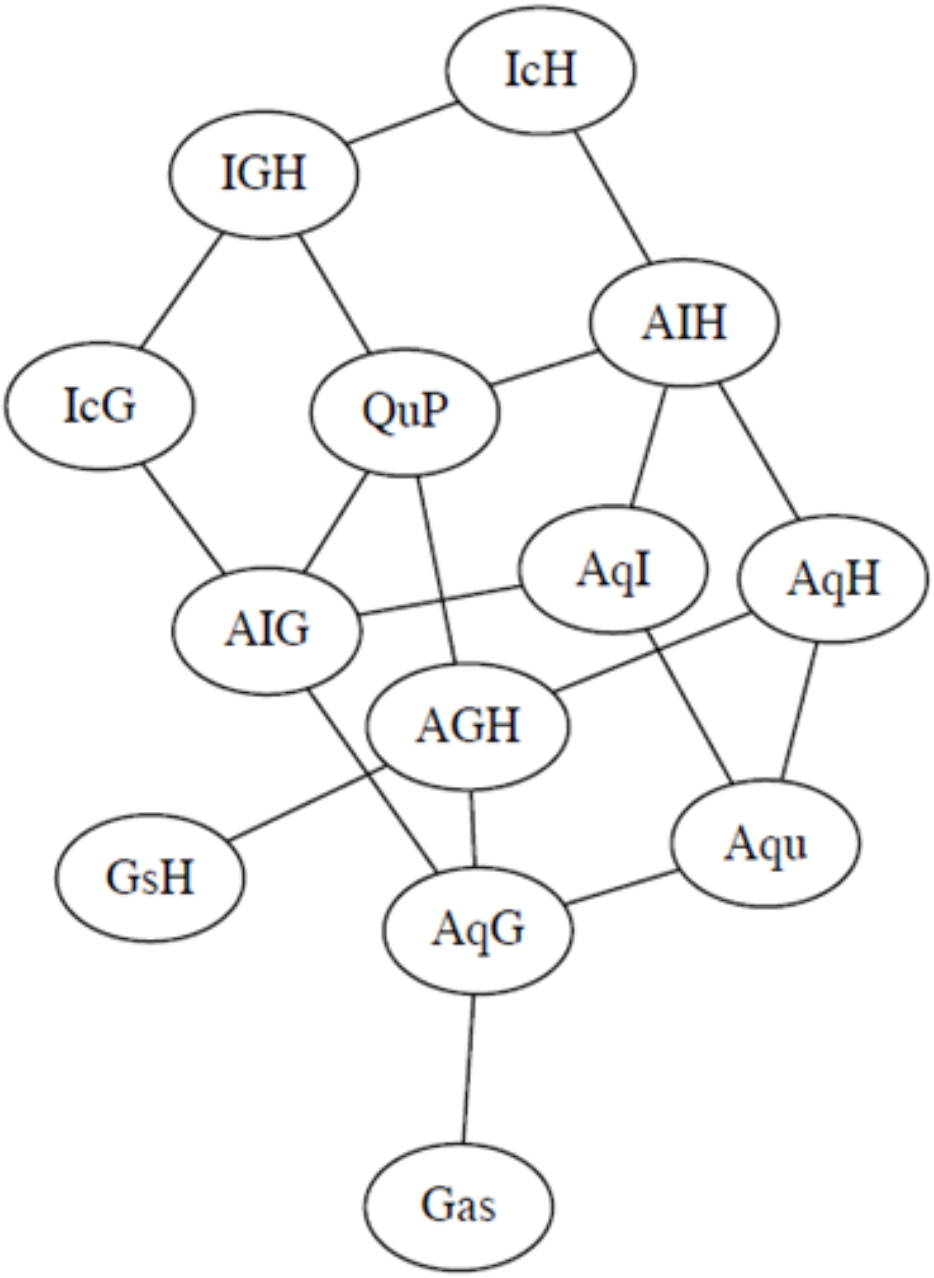

Figure 5.3: Phase transition among all possible phases.

where the subscripts (nm) denote a suitable average of the quantity in question at the interface between grid blocks (elements) n and m.

Substituting Eqs. 5.50 and 5.52 into the governing Eq. 5.1 yields the discretized form of the fundamental equation of compositional mass balance and flow through porous media as

$$\frac{dM_n^\eta}{dt} = \frac{1}{V_n}\sum_m A_{nm}F_{nm}^\eta + q_n^\eta; \qquad \eta = \kappa,\ \theta \tag{5.53}$$

The time derivatives in Eq. 5.53 are approximated using a standard backward Euler method. The component and heat accumulations, fluxes and sink/source terms are all evaluated at the new time level, $t^{k+1} = t^k + \Delta$ t, and the resulting fully implicit method provides maximum numerical stability. Discretizing time, expanding the various terms and collecting leads to the residual formulation of the following set of coupled non-linear, implicit algebraic equations

$$R_n^{\kappa,\ k+1} = M_n^{\kappa,\ k+1} - M_n^{\kappa,k} - \frac{\Delta t}{V_n}\left(\sum_m A_{nm}F_{nm}^{\kappa,\ k+1} + V_n q_n^{\kappa,\ k+1}\right) = 0 \tag{5.54}$$

For each volume V_n, there are N_η = N_κ+1 equations in JFTS+H, where N_κ is the number of mass components and the last equation is that of the heat balance. Note that simulation of hydrate flow and behavior invariably involves significant heat exchanges, so the hydrate formation or dissociation can never be treated as an isothermal process. Thus, the maximum possible value $max\{N_\eta\} = 3$ (when an inhibitor such as salt is not involved) or 4 (when an inhibitor is included) in an equilibrium model, and $max\{N_\eta\} = 4$ or 5, respectively, in kinetic model, leading to a total of $N_\eta \times N_E$ coupled nonlinear equations in a system discretized into N_E grid blocks.

Eq. 5.54 is solved by the Newton-Raphson (NR) iteration, which is implemented as follows. Using the Taylor series approach, the implicit Eq. 5.54 is expanded as

$$R_n^{\kappa,\ k+1}\left(x_{i,\ p+1}\right) = R_n^{\kappa,\ k+1}\left(x_{i,p}\right) + \sum_i \left.\frac{\partial R_n^{\kappa,\ k+1}}{\partial x_i}\right|_p \left(x_{i,p+1} - x_{i,p}\right) = 0 \tag{5.55}$$

The index p denotes the (NR) iteration step. In JFTS, the terms $\frac{\partial R_n}{\partial x_i}$ in the Jacobian matrix can be

evaluated either by numerical differentiation or by the more efficient process of automatic differentiation that has significantly lower memory requirements. The NR iterations continue until the residuals are reduced below a preset convergence tolerance

$$\left| \frac{R^{\kappa,k+1}_{n,\ p+1}}{M^{\kappa,k+1}_{n,\ p+1}} \right| \leq \epsilon_1 \tag{5.56}$$

The default relative convergence criterion is 10^{-5}.

The solutions of the matrix equations in the study were obtained using the bi-conjugate gradient stabilized (Van der Vorst, 1992) solver (BiCG-stab), a single-step incomplete LU decomposition (Saad, 1994) as the preconditioner, the Coordinate (COO) matrix storage format for serial computations, and the Compressed Sparse Row (CSR) matrix storage format for parallel computations (Stoer and Bulirsch, 2002).

5.7 Flowchart

5.7.1 Serial Computing

The flowchart of the serial computations in JFTS is shown in **Fig. 5.4**, in which the term 'NR' denotes the Newton-Raphson iteration. The operations in the chart component identified as 'Phase State Machine' check unrealistic conditions and phase transition.

5.7.2 Parallel Computing

The flowchart of the parallel computations in JFTS is shown in **Figs. 5.5** and **5.6**. The parts in red denote the MPI-based operations.

5.8 MPI Operations

5.8.1 MPI Initialization

This part initializes the MPI execution environment: the rank of each processor and the number of processors. This must be called before any other MPI functions.

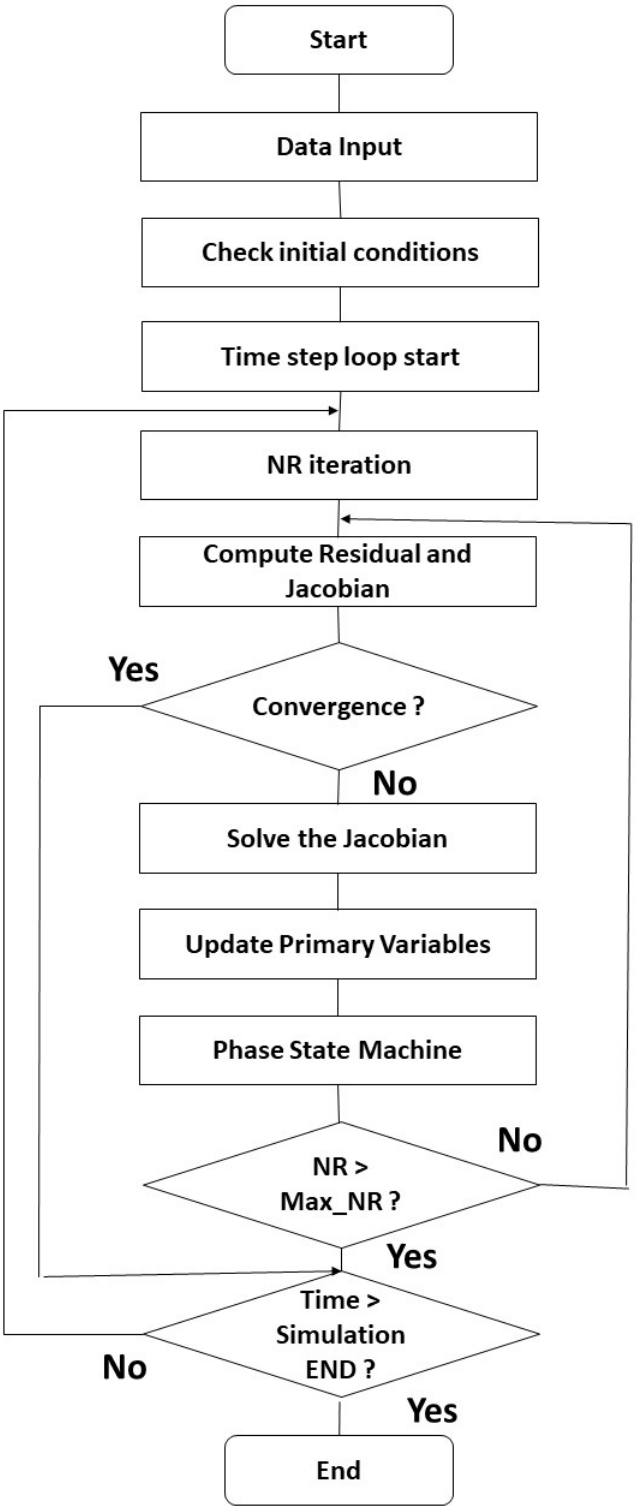

Figure 5.4: Flowchart in serial computing.

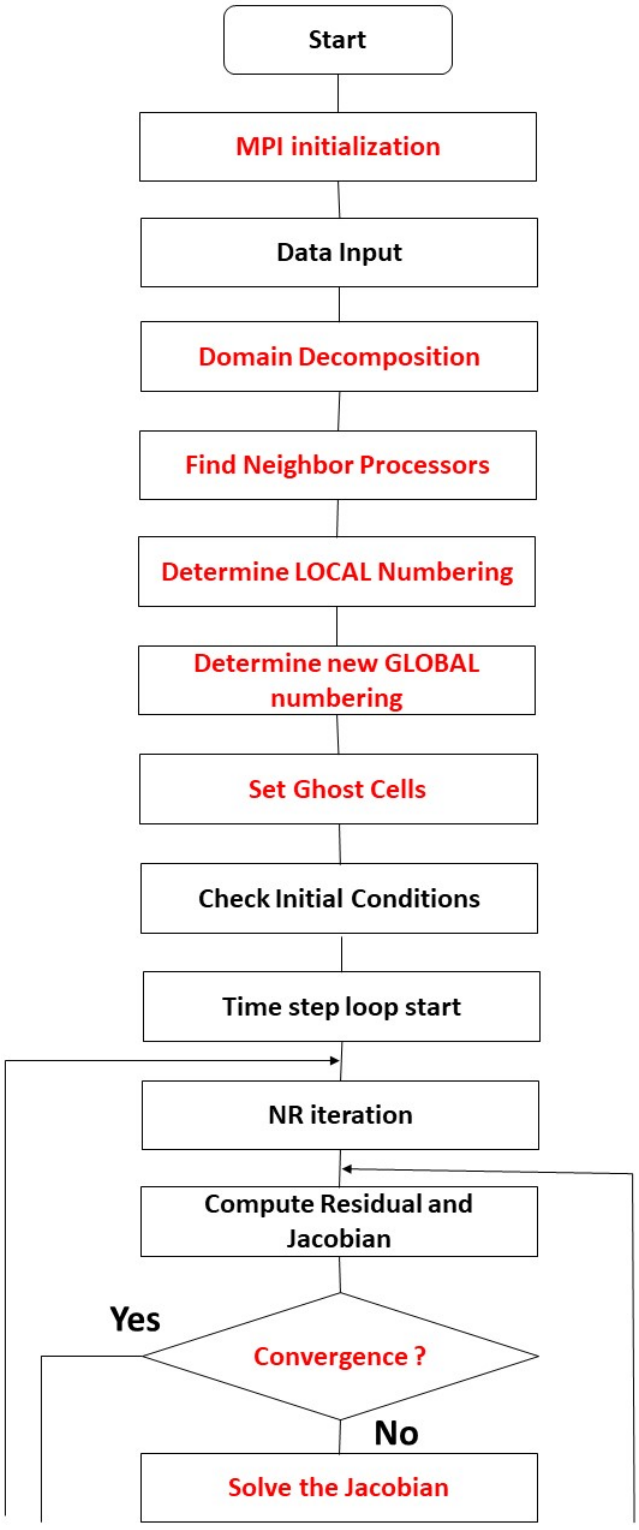

Figure 5.5: Flowchart in parallel computing(1).

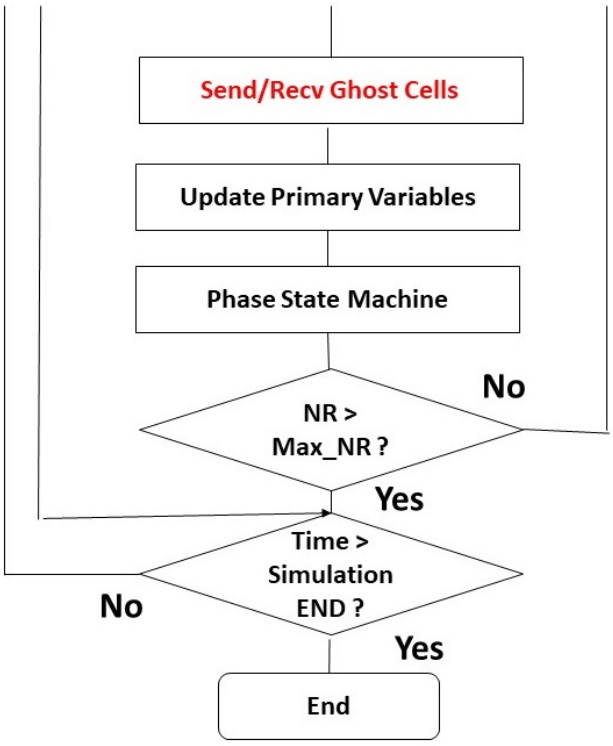

Figure 5.6: Flowchart in parallel computing(2).

5.8.2 Domain Decomposition

In a parallel system involving distributed memory, each processor has its own memory and cannot access the memory of any other processor. Accessing data stored in the memory of other processors requires processor communication, which is a time-consuming operation that, unless carefully managed, can adversely effect the parallel performance. An effective domain decomposition scheme – in which the domain is subdivided into particular subdomains (involving groups of appropriately-selected elements) that are assigned to different processors – can be important in minimizing processor communication and increasing the effectiveness of parallelization. This task is accomplished in JFTS through the use of the METIS library, which is based on the multilevel-recursive bisection and multilevel k-way algorithms (Gropp et, al., 1989). An example is shown in the 6×6 grid shown in **Fig. 5.7**, in which the entire domain is decomposed into 4 subdomains assigned to different processors.

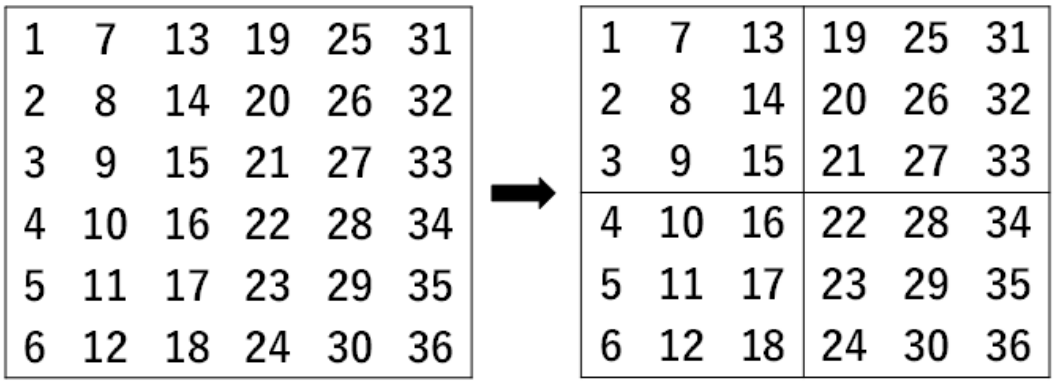

Figure 5.7: Global ordering and domain decomposition for a 2-dimensional array with 4 processors.

5.8.3 Find Neighbor Processors

This is a function used for MPI operations. This function assigns a specific subdomain to each processor and determines the subdomain neighbors. **Fig. 5.8** shows the element numbering order in the entire domain after assigning the subdomains to the four processors PE0, PE1, PE2 and PE3.

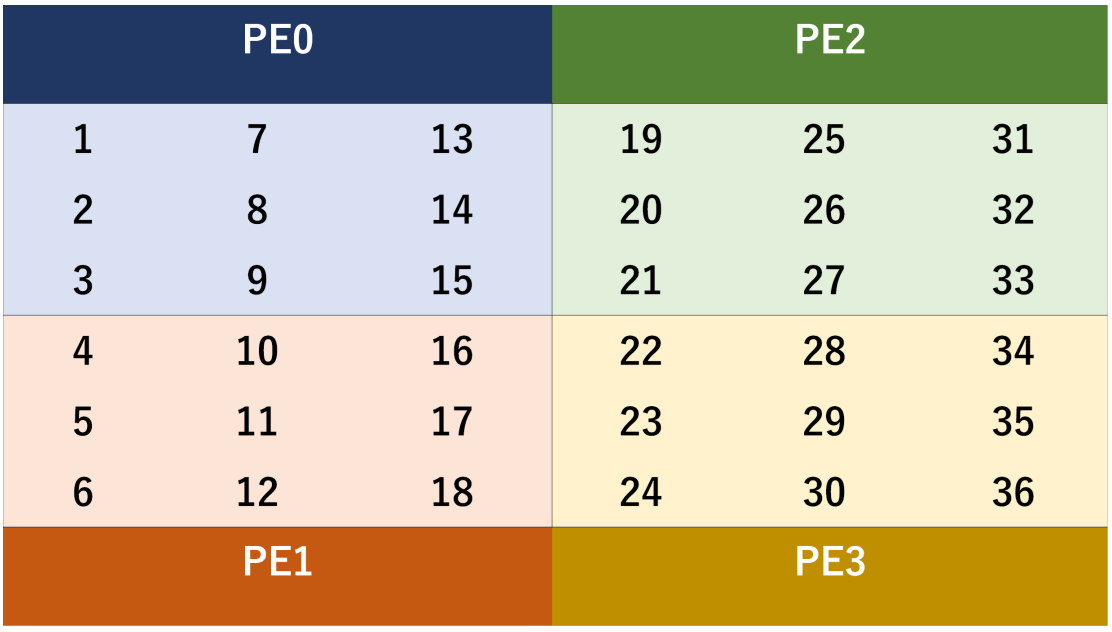

Figure 5.8: Assigning processors to subdomains and determining their neighbors.

5.8.4 Determine LOCAL Ordering

This is also a function used for MPI operations. This function determines the *local* ordering, which begins with the number 1 in each processors-assigned subdomain. **Fig. 5.9** shows the local ordering in the subdomain of each processor.

5.8.5 Determine New Global Ordering

This is also a function used for MPI operations. This function determines the new *global* ordering, which is continuous in the processor-assigned subdomains of the entire domain. The new global ordering is shown in **Fig. 5.10**, and is the one used in the computations of the Jacobian matrix.

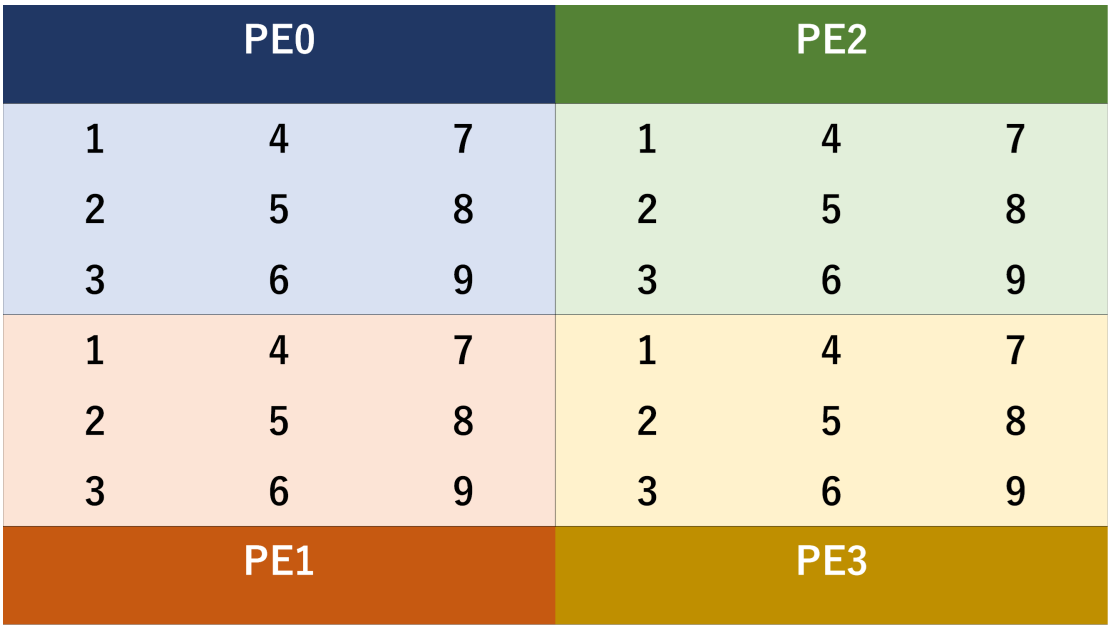

Figure 5.9: Local ordering in the subdomains of the 2D system assigned to the 4 processors of Fig. 5.7.

5.8.6 Set Ghost Cells

This is also a function used for MPI operations. This function sets ghost cells, i.e., cells (elements, gridblocks) that are not assigned to any given processors but to its neighbors, and which

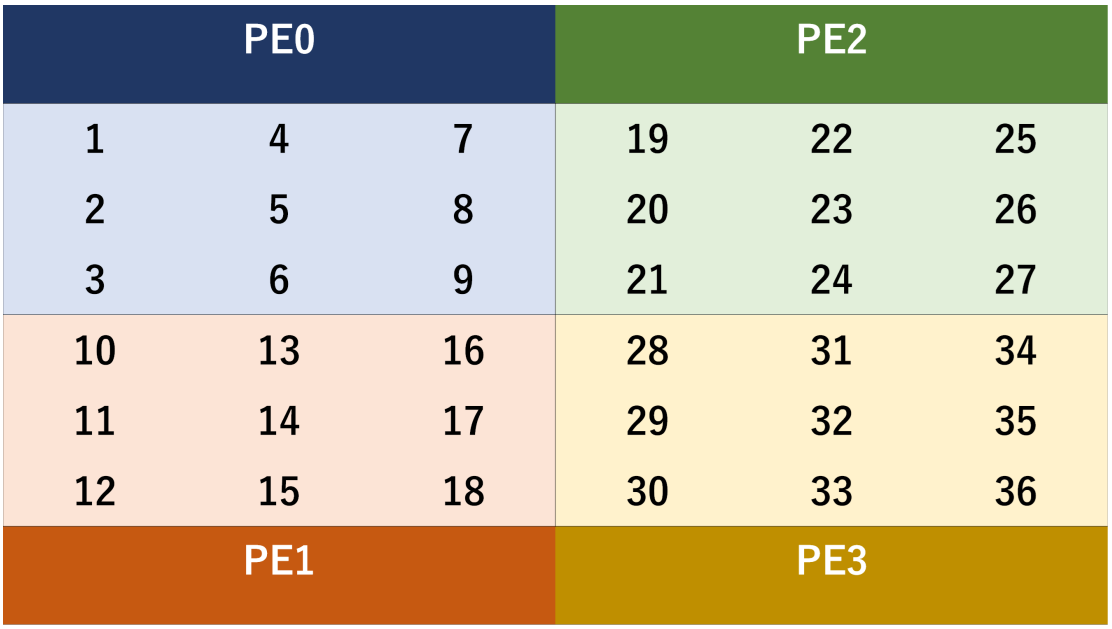

Figure 5.10: New global ordering of the elements of the 2D grid distributed among 4 processors.

are needed to compute the flow terms. **Fig. 5.11** shows the ghost cells associated with each one of the 4 processors. JFTS needs to store both the new global and the local cell ordering. In the local ordering, ghost cells are counted after the subdomains cells.

5.8.7 Checking Convergence

To check the convergence of the solutions of the Jacobian obtained from the matrix solver, the maximum residual in the entire domain is compared to the maximum allowable tolerance (an input to the code). In the JFTS parallel computing, all maximum residuals are gathered and compared using the "*Allreduce*" function to determine their maximum values.

5.8.8 Solution of the Jacobian Matrix

The Jacobians generated in the JFTS simulations are solved using the LIS matrix solver package (Nishida, 2010). LIS includes both serial and parallel matrix solver (multi-threading and multi-processor) options. The type of the desired matrix solvers and the format of the entry of the Jacobian matrix are provided by the user as input data. Each processor computes the sub-parts of the Jacobian matrix and the residual matrix. An image of the parallelized "Ax=b" Jacobian is

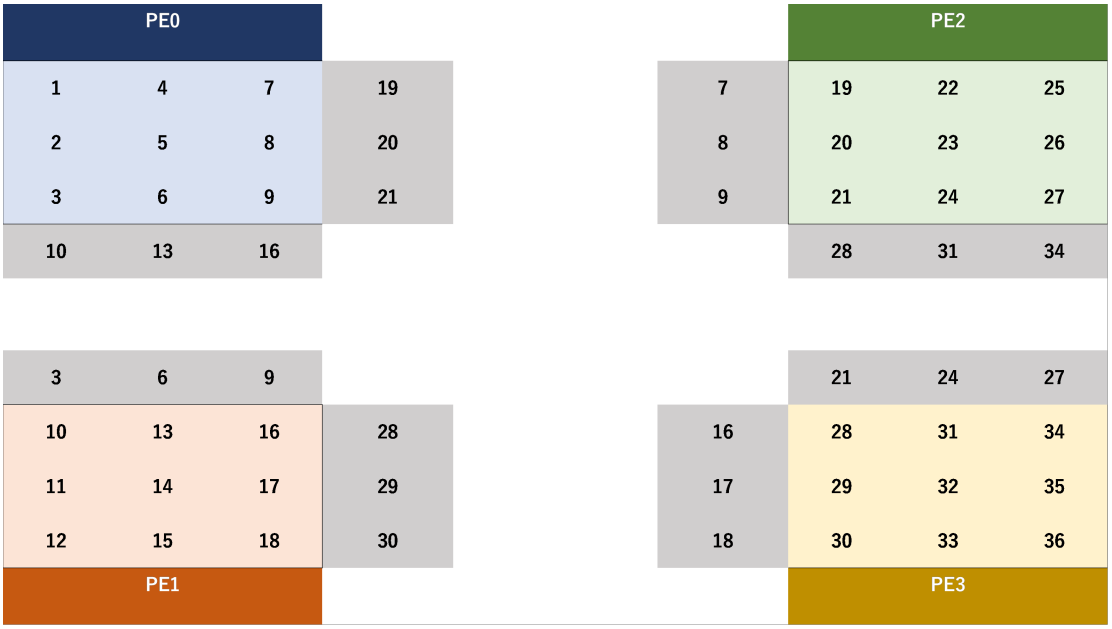

Figure 5.11: Ghost cells in the 2D system distributed among the 4 processors.

shown in **Fig. 5.12**.

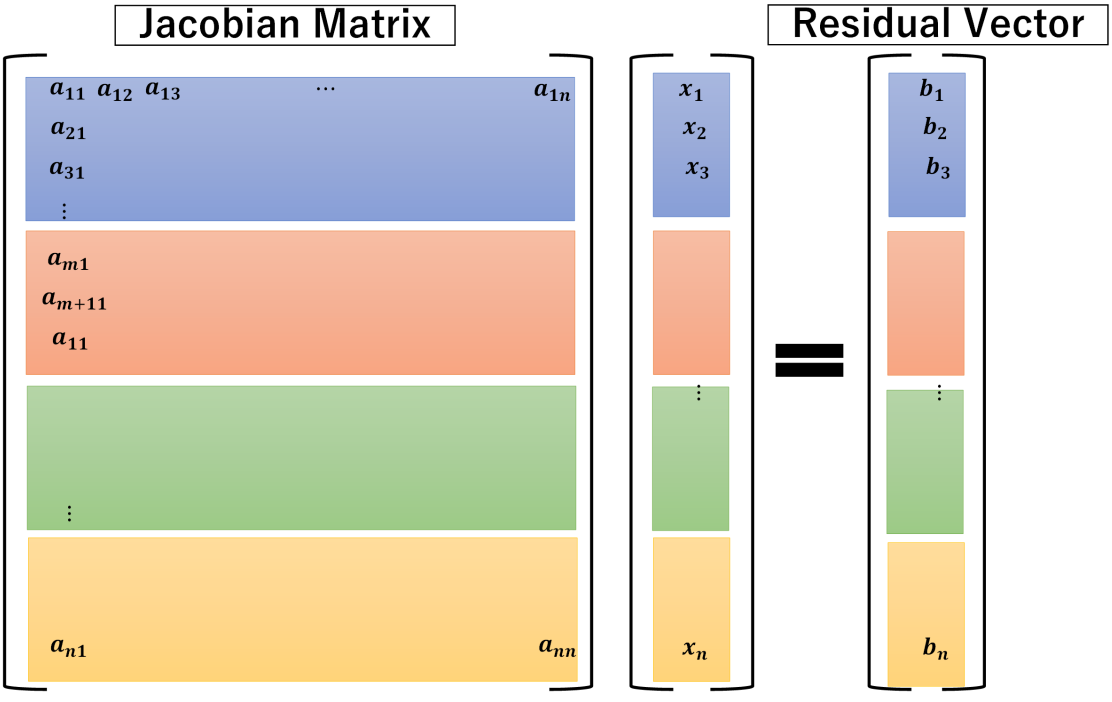

Figure 5.12: Jacobian matrix and residual vector for parallel matrix solver.

5.8.9 Send/Recv Ghost Cells

Each processor has access to only the data within its assigned subdomain (in this example, access to 9 elements only). To update the data in the ghost cells, (the 6 grayed cells in **Fig. 5.11**), the primary variables and state indices are communicated among processors using P-to-P communications.

5.9 Code Validations

The JFTS+H code is validated by comparing its predictions to the results of the TOUGH+ HYDRATE v.1.5 simulator (hereafter referred to as the T+H simulator), which is one of the most widely-used numerical simulator for the description of the overall system behavior during (a) gas production from natural hydrate accumulations in geologic media and (b) in laboratory studies of CH_4-hydrate behavior. In the validation process, the JFTS+H performance is tested against all the example problems listed in the T+H User's Manual (Moridis, 2014). These include problems of both equilibrium and kinetic hydrate formation and dissociation and involve pressure, thermal and inhibitor effects.

5.9.1 Test_1T: Thermal Stimulation, Equilibrium Dissociation, No Inhibitor (Moridis, 2014)

This 1-D problem simulates a thermal stimulation process under equilibrium dissociation conditions in a 1 m long domain. The initial conditions involve water and hydrate coexisting at pressure $p_0 = 4.0 \times 10^6$ Pa and a temperature $T_0 = 1.2$ °C. At a time $t = 0$, the boundary at $x = 0$ becomes permeable and its temperature is set at 45 °C, i.e., at a temperature sufficiently high to induce dissociation at this p). As the temperature in the domain increases, hydrate dissociation is expected to occur and for gas to evolve. The porous medium has a porosity $\phi = 0.3$, a permeability $\mathrm{k} = 2.96 \times 10^{-13}$ m^2, a critical mobile porosity $\phi_\mathrm{c} = 0.05$, a porosity reduction exponent of 3, and a pore compressibility $\alpha_p = 1.0^{-8}\ 1/Pa$. The 'wet' thermal conductivity of the porous medium is $k_{w\theta} = 3.1$ W/m/K and is typical of fully-saturated sandy media.

Figs. 5.13 to **5.17** provide comparisons of the results from the JFTS+H and the T+H simulators in this relatively simple study of thermally-induced equilibrium hydrate dissociation without any

inhibitor effects. The results of the two simulators either practically coincide or exhibit very small deviations, thus providing evidence of the validity of the JFTS+H simulator. The small deviations are attributed to differences in the $P-T$ equilibrium curves between the two simulators, especially in the vicinity of the quadruple point. The study provides early insights into the hydrate behavior in response to thermal stimulation. The dissociation front advances relatively slowly and its position is marked by the highest p in the domain (higher than the initial p_0), which is caused by the gas release. The location of the advancing front that is associated with the higher p in the domain is confirmed by S_H distribution in **Fig. 5.17**. Note the higher (than the initial, and increasing with time) S_H ahead of the dissociation front, which is caused by gas flow in both directions about the front and hydrate reformation ahead of it because of the elevated p at this location. The highest S_G (**Fig. 5.16**) is observed immediately behind the dissociation front, flowing toward the permeable boundary at $x - 0$; the lowest S_A occurs immediately ahead of the front (where the S_H is the highest and increasing over time.

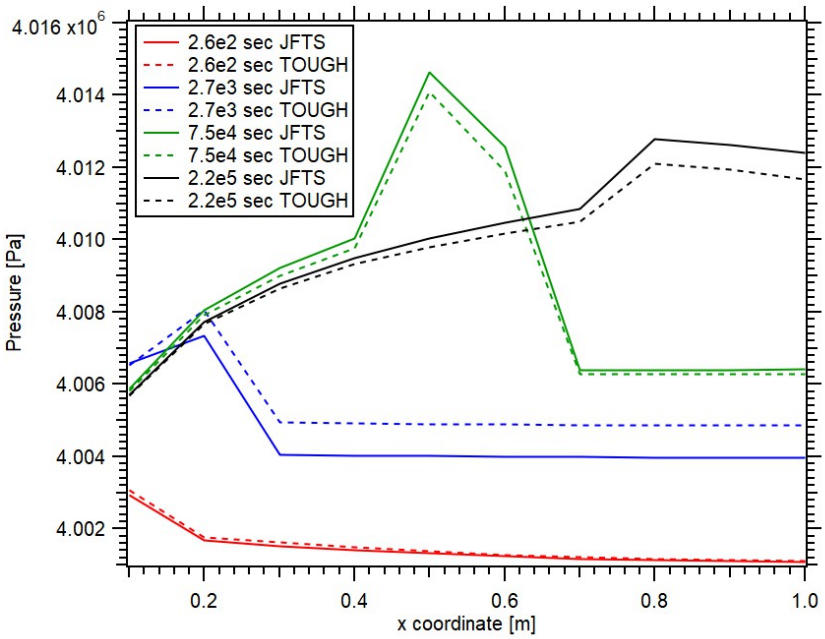

Figure 5.13: Comparison of the JFTS+H and T+H results: evolution of pressure distributions over time in the 1D problem of Test_1T of Moridis (2014).

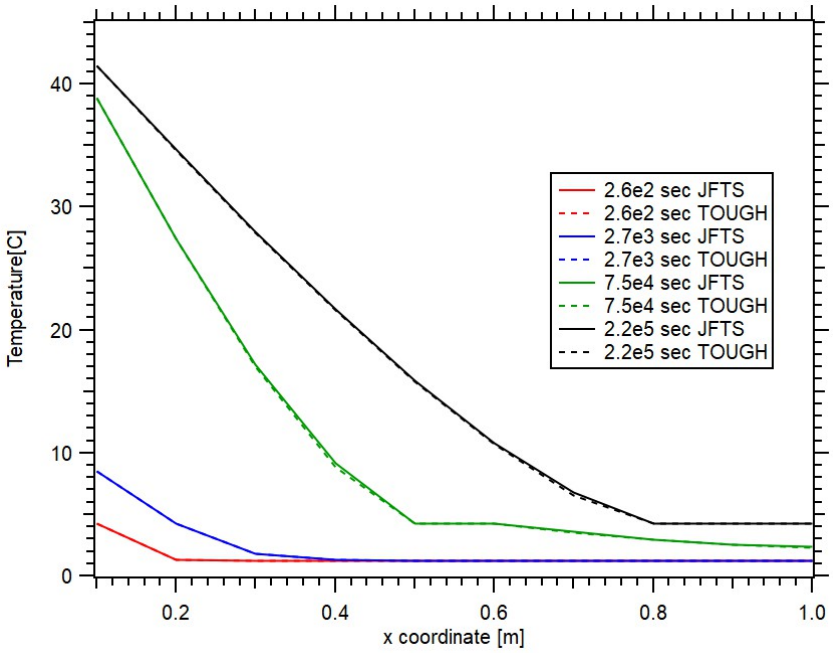

Figure 5.14: Comparison of the JFTS+H and T+H results: evolution of temperature distributions over time in the 1D problem of Test_1T of Moridis (2014).

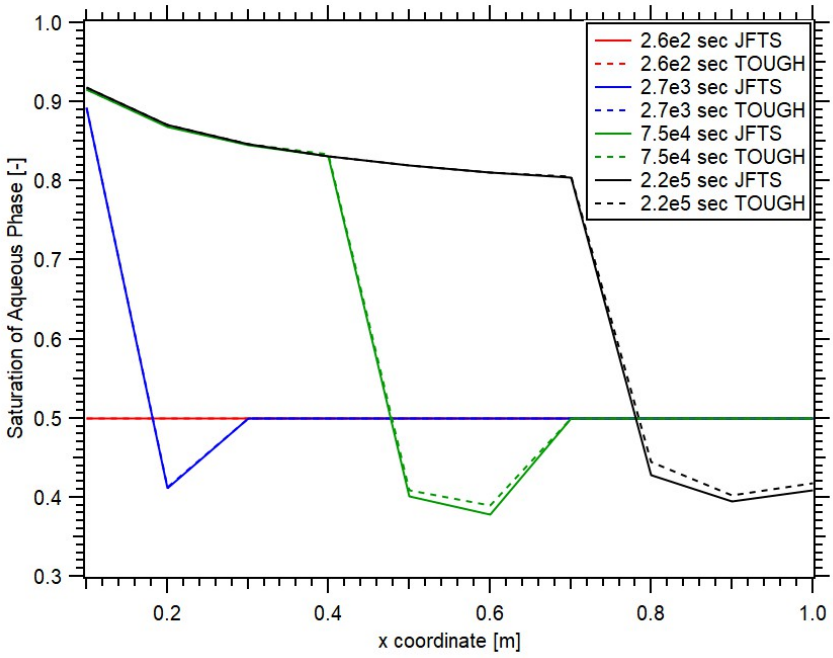

Figure 5.15: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the aqueous phase saturation over time in the 1D problem of Test_1T of Moridis (2014).

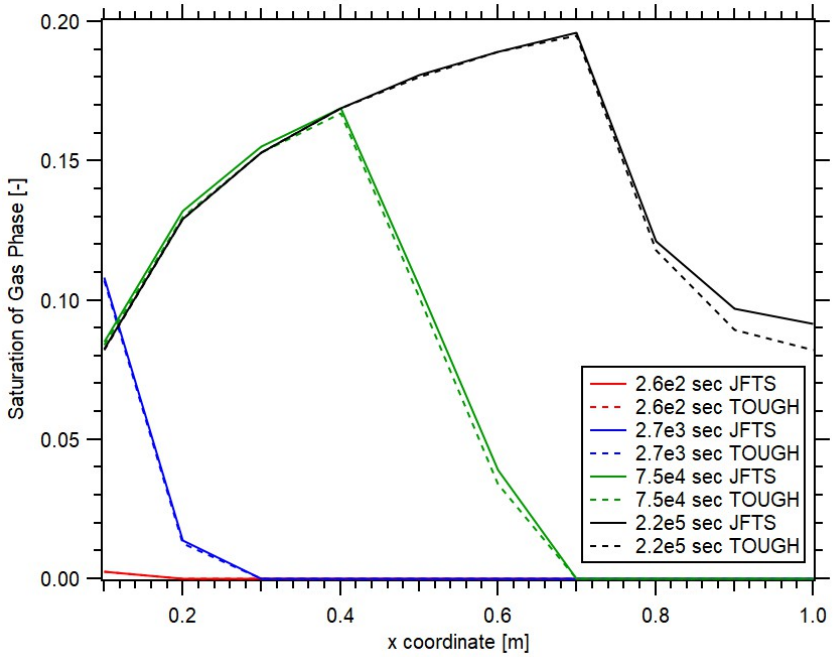

Figure 5.16: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the gas phase saturation over time in the 1D problem of Test_1T of Moridis (2014).

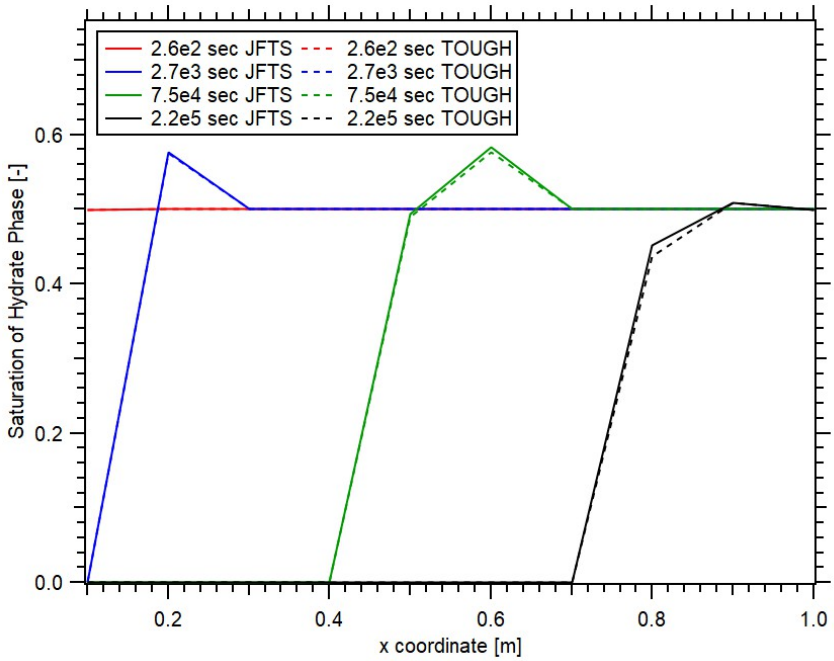

Figure 5.17: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the hydrate phase saturation over time in the 1D problem of Test_1T of Moridis (2014).

5.9.2 Test_1Tk: Thermal Stimulation, Kinetic Dissociation, No Inhibitor (Moridis, 2014)

This 1-D problem has the same initial conditions and boundary conditions as Test_1T (Moridis, 2014). The only difference is that hydrate dissociation is treated as a kinetic reaction. Additional parameters needed for the simulation of the kinetic process are the activation energy ($\Delta E_a = 8.1 \times 10^4\ J/mol$), the intrinsic hydration reaction constant ($K_0 = 3.6 \times 10^4\ kg/(m^2\ Pa\ s)$), and the area adjustment factor ($F_A = 1.0$ [dimensionless]). **Figures 5.18** to **5.22** provide a comparison of the corresponding JFTS+H and T+H results and show very similar predictions, but with a delay in the kinetic study that is consistent with the retardation expected in kinetic studies.

Figs. 5.23 and **5.24** show respectively comparisons of the equilibrium and kinetic estimates of (a) the rates of CH_4 release, Q_R and (b) of the cumulative volume of released CH_4, M_R. A significant (but expected) observation is that the kinetic rate is initially lower than the equilibrium rate because of the kinetic retardation, but the two coincides later. The comparison of the corresponding M_R in **Fig. 5.24** shows very little difference between the two estimates, indicating that the kinetic reaction is very fast.

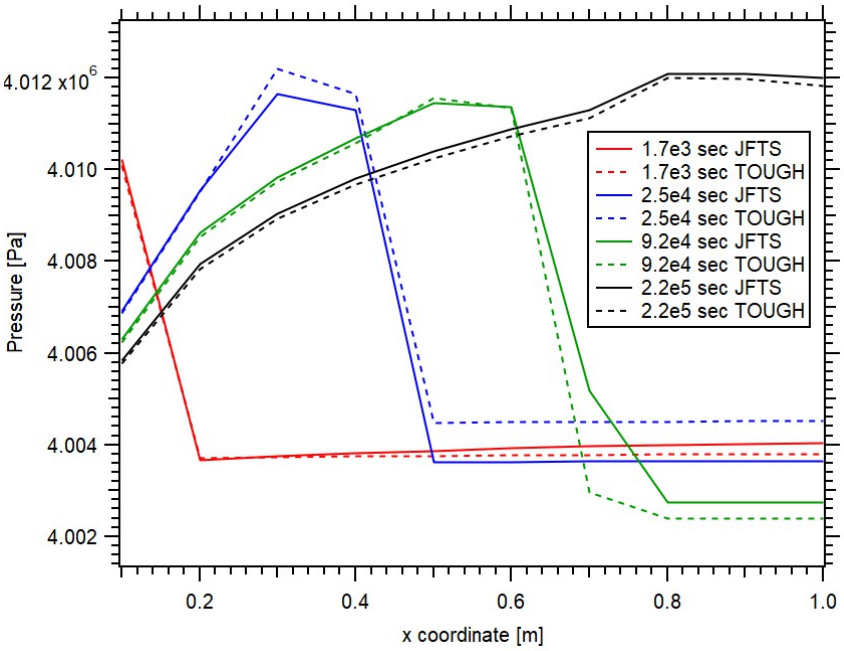

Figure 5.18: Comparison of the JFTS+H and T+H results: evolution of pressure distributions over time in the 1D problem of Test_1Tk of Moridis (2014).

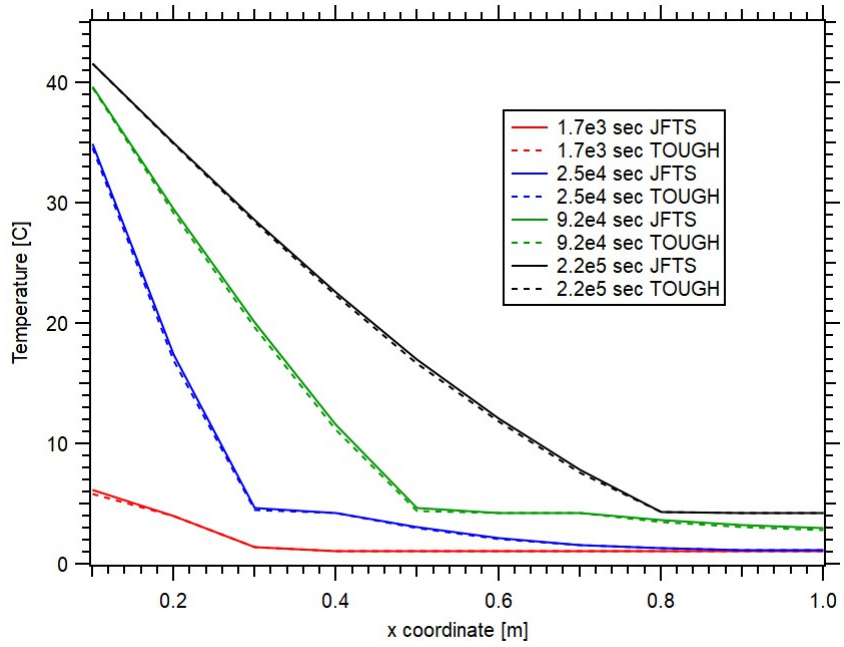

Figure 5.19: Comparison of the JFTS+H and T+H results: evolution of temperature distributions over time in the 1D problem of Test_1Tk of Moridis (2014).

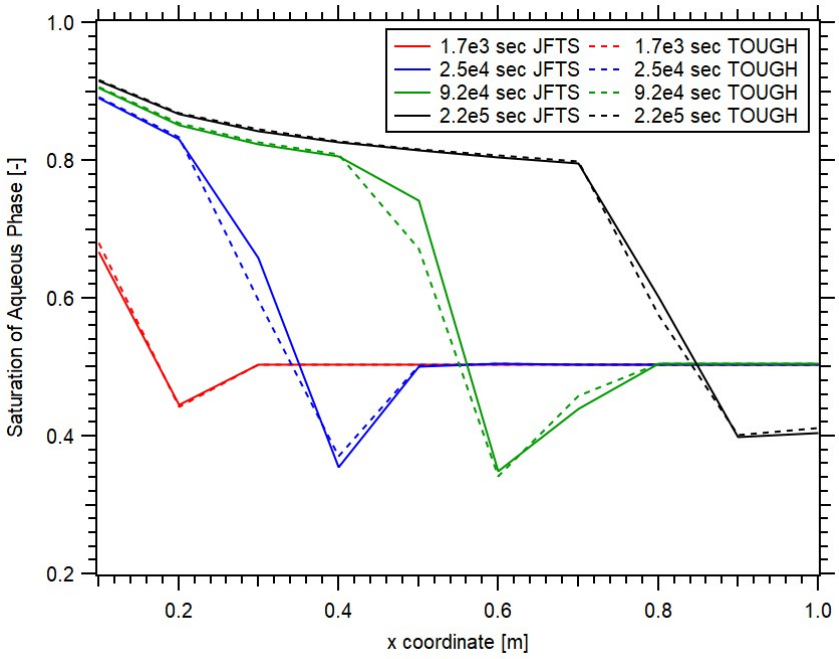

Figure 5.20: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the aqueous phase saturation over time in the 1D problem of Test_1Tk of Moridis (2014).

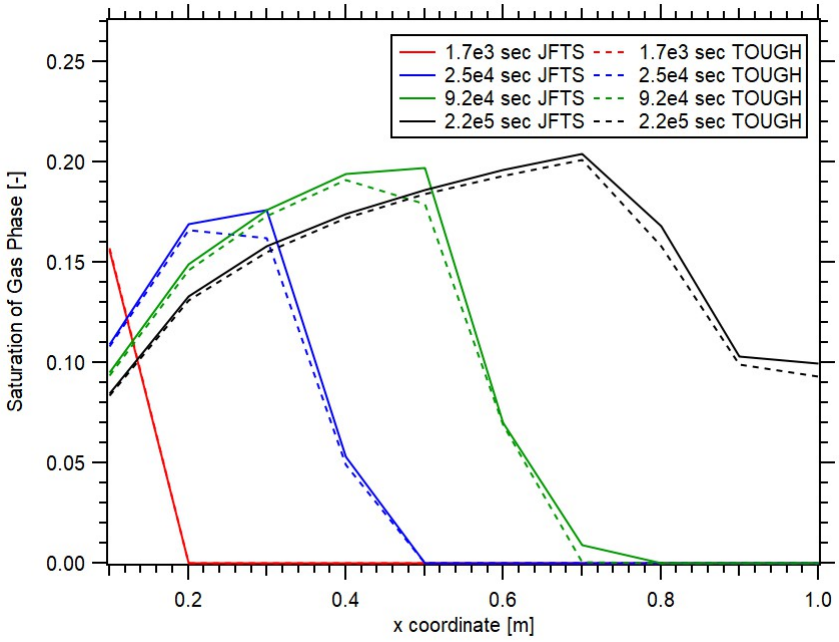

Figure 5.21: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the gas phase saturation over time in the 1D problem of Test_1Tk of Moridis (2014).

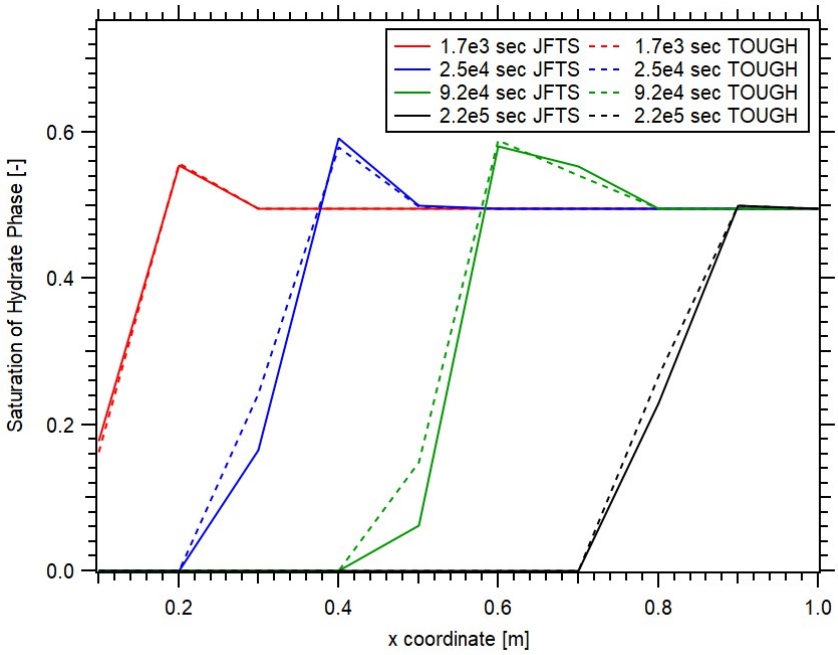

Figure 5.22: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the hydrate phase saturation over time in the 1D problem of Test_1Tk of Moridis (2014).

5.9.3 Test_1P: Depressurization, Equilibrium Dissociation, No Inhibitor (Moridis, 2014)

This problem involves the same domain as that in the previous cases, and differs in (a) the initial temperature $T_0 = 6$ ^{o}C, and (b) in the conditions at the $x = 0$ boundary, which is now kept at a constant $T_b = 1.0$ °C pressure of $p_b = 2.7 \times 10^6$ Pa, i.e., at a p that is lower than the initial pressure in the domain ($\mathrm{p_0} = 6 \times 10^6$ Pa) but lower that the equilibrium pressure p_e at the initial temperature of , but higher than the quadruple point pressure p_Q of the CH_4-hydrate. The pressure difference between the domain and boundary and the non-zero permeability allow depressurization, which induces hydrate dissociation but no ice formation. **Figs. 5.25** to **5.29** provide comparisons of the JFTS+H and the T+H results for this problem. The excellent agreement of the results from the two simulators provide additional evidence in support of the validity of the JFTS+H predictions. As before, small deviations (especially at later times) are attributed to differences in the equations of the P-T relationship in the two codes.

The rapid depressurization in **Fig. 5.25** was expected, given the small size of the domain and

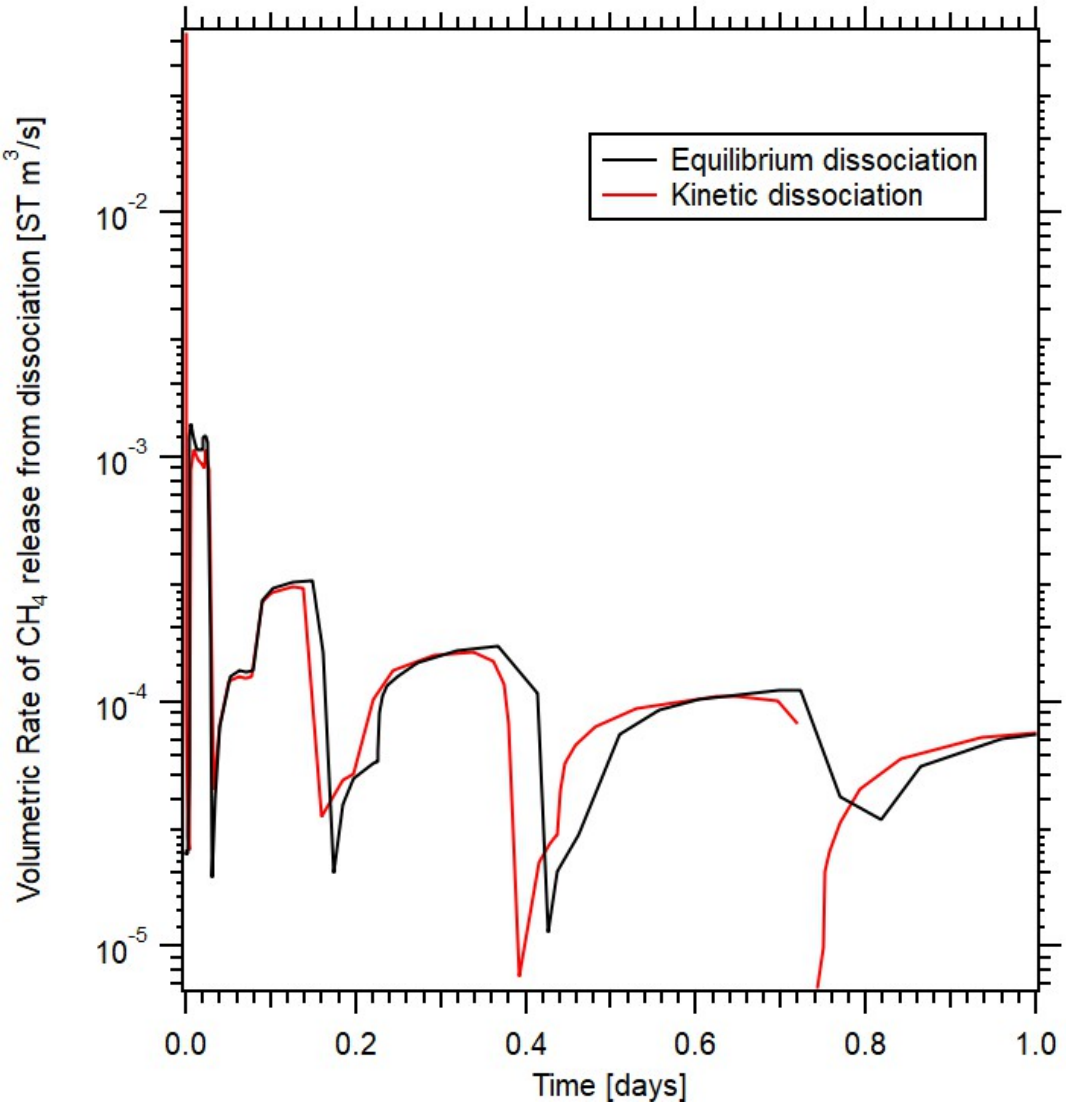

Figure 5.23: JFTS+H simulation results: comparison of the volumetric rates of $\mathrm{CH_4}$ release from hydrate dissociation in the 1D problem of Test_1T of Moridis (2014) and Test_1Tk of Moridis (2014).

the intrinsic permeability of the system. The temperature evolution in **Fig. 5.26** declines rapidly as dissociation proceeds because of the strongly endothermic reaction of hydrate dissociation without an external heat source. Compared to the thermal stimulation cases, the depressurization process is shown to cause a much faster rate of hydrate dissociation. S_G reaches a maximum rather early, and then declines rapidly because of continuous removal through the $x-0$ boundary and the exhaustion of the hydrate.

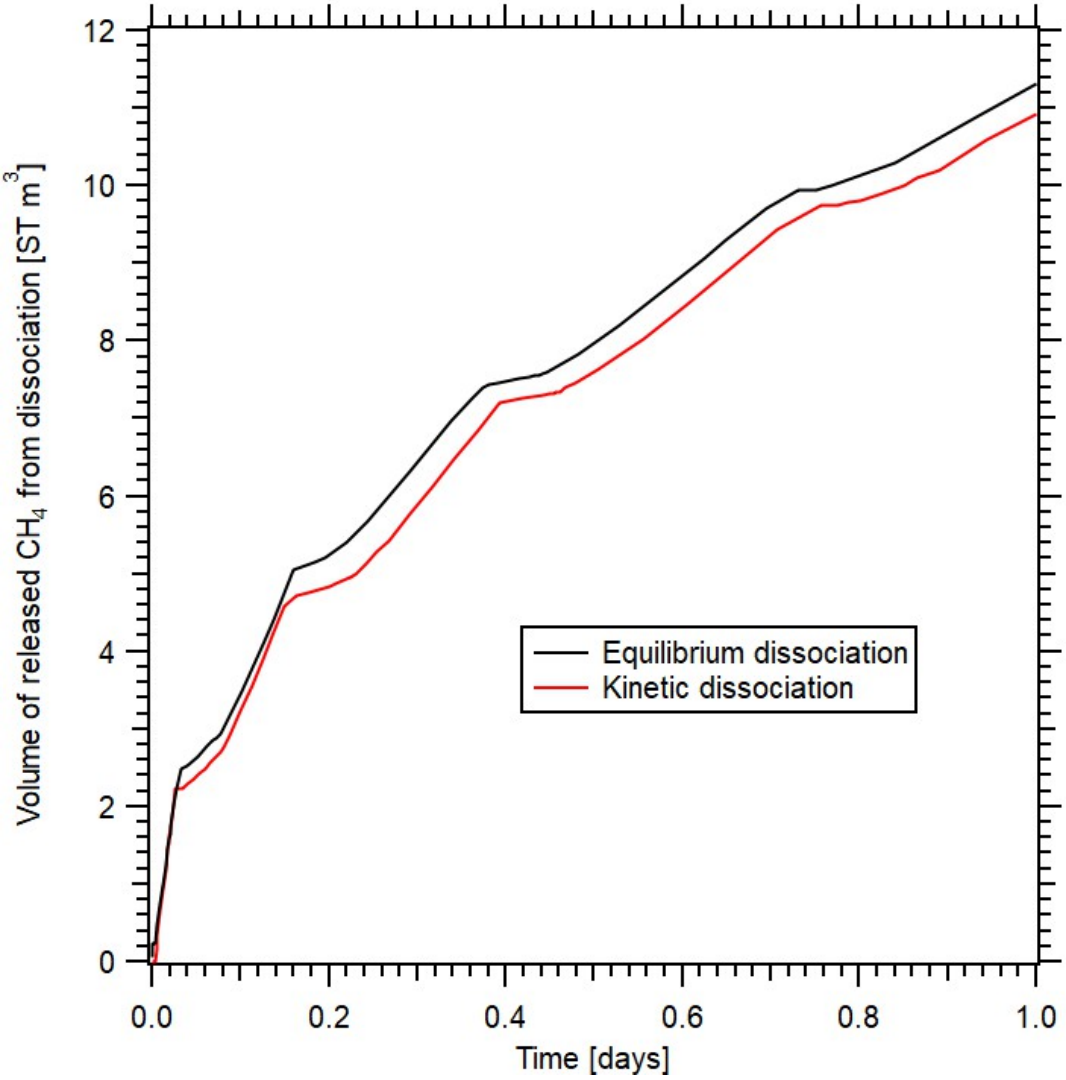

Figure 5.24: JFTS+H simulation results: comparison of the cumulative volumes of CH_4 release from hydrate dissociation in the 1D problem of Test_1T of Moridis (2014) and Test_1Tk of Moridis (2014).

5.9.4 Test_1Pk: Depressurization, Kinetic Dissociation, No Inhibitor (Moridis, 2014)

The single difference from the Test_1P case discussed in the previous section is in the treatment of hydrate dissociation, which is now as a kinetic reaction. **Figs. 5.30** to **5.34** provide comparisons of the JFTS+H and the T+H results for this depressurization case. Inspection of the results shows (a) very similar predictions of the pressure, temperature, and phase saturation distributions obtained from both the kinetic and the equilibrium simulations, indicating that hydrate dissociation is

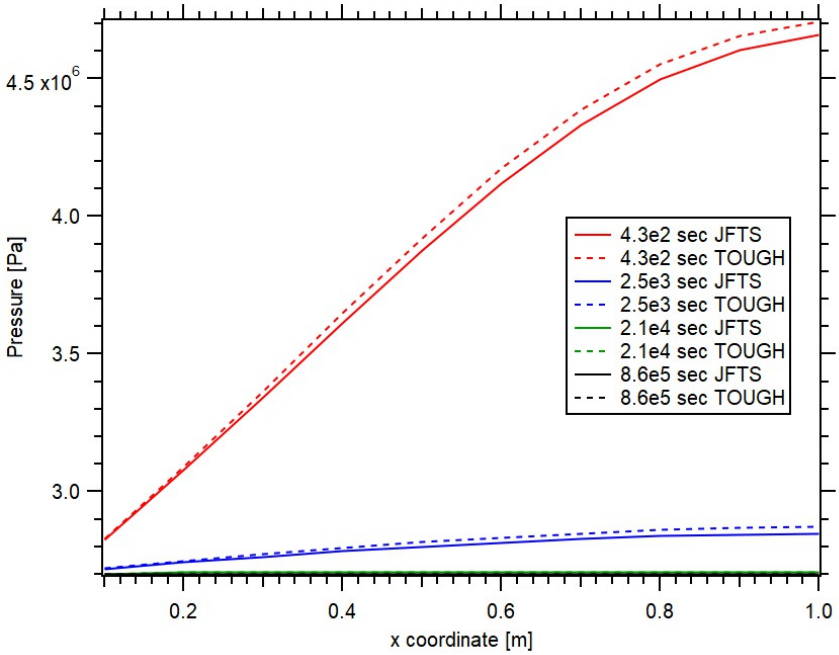

Figure 5.25: Comparison of the JFTS+H and T+H results: evolution of pressure distributions over time in the 1D problem of Test_1P of Moridis (2014).

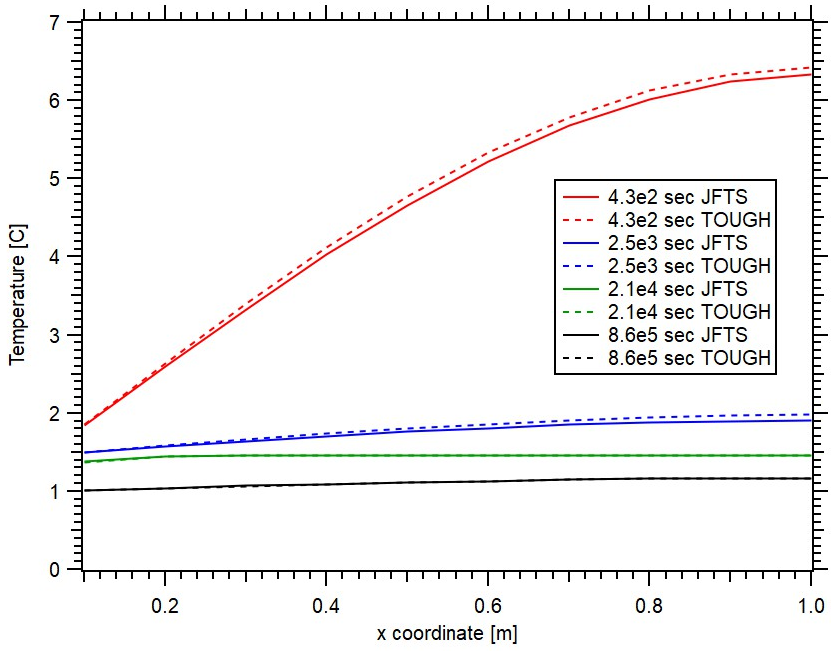

Figure 5.26: Comparison of the JFTS+H and T+H results: evolution of temperature distribution over time in the 1D problem of Test_1P of Moridis (2014).

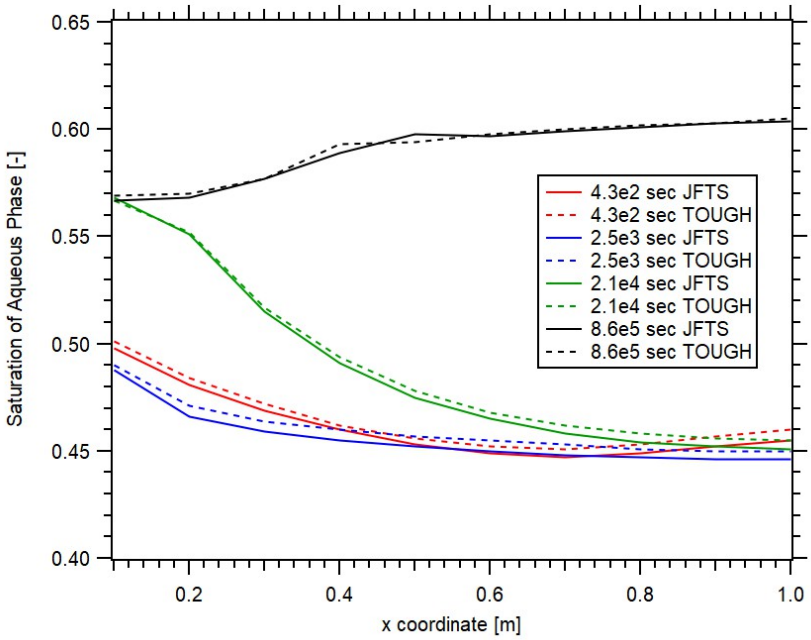

Figure 5.27: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the aqueous phase saturation over time in the 1D problem of Test_1P of Moridis (2014).

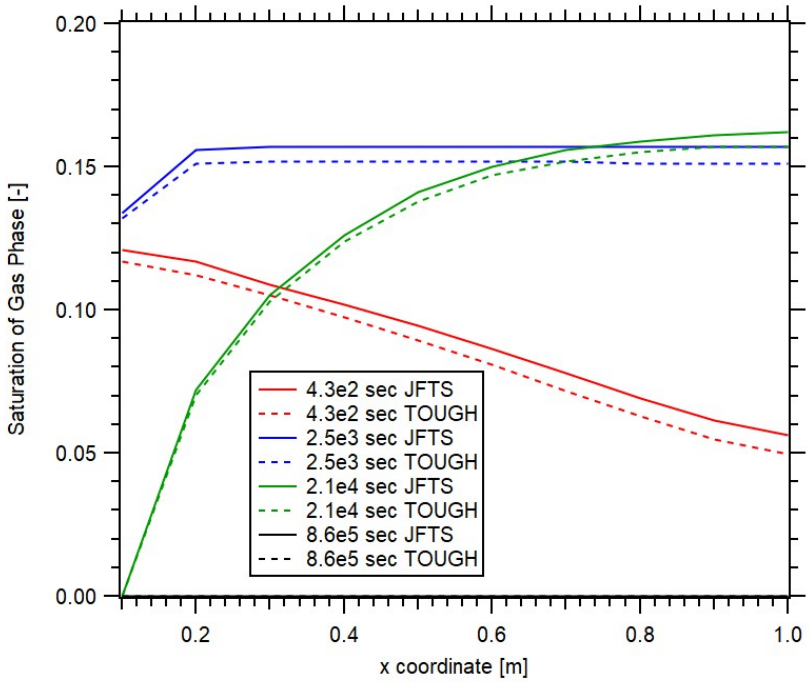

Figure 5.28: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the gas phase saturation over time in the 1D problem of Test_1P of Moridis (2014).

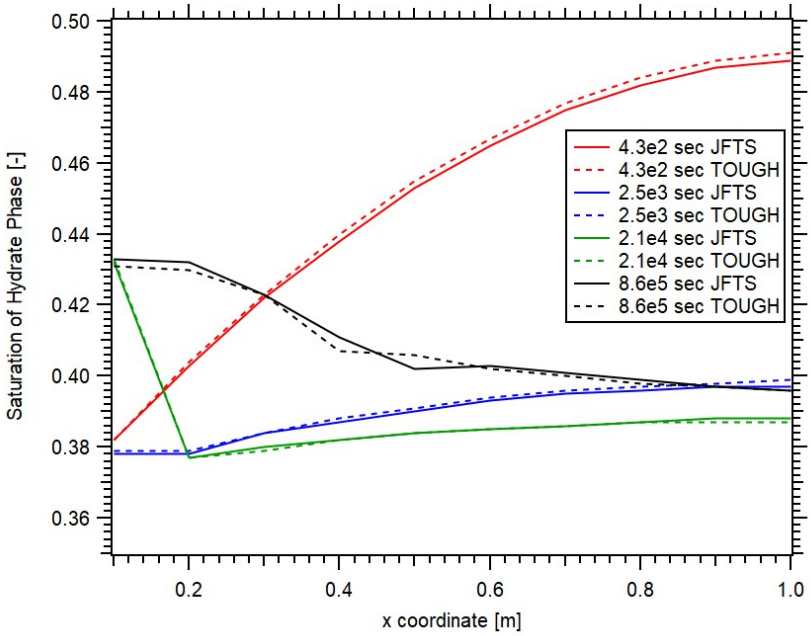

Figure 5.29: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the hydrate phase saturation over time in the 1D problem of Test_1P of Moridis (2014).

a very fast process and (b) the excellent agreement of the JFTS+H results with the T+H solution, an observation that further strengthens the case for the validity of the JFTS+H code.

5.9.5 Test_1P_ice: Depressurization, Equilibrium Dissociation, No Inhibitor, Ice Evolution (Moridis, 2014)

The only difference between this problem and Test_1P (Moridis, 2014) is in the pressure of boundary condition at $x = 0$, which is now kept constant at $\mathrm{p_b} = 5.0 \times 10^5$ Pa, i.e., at pressure lower than the quadruple point pressure $p_Q = 2.65 \times 10^6$ of CH_4-hydrates. The significant $p_0 - p_b$ difference is expected to lead to faster CH_4 dissociation at higher rates, and the emergence of ice. The proximity in the results of the JFTS+H and the T+H simulations in **Figs. 5.35** to **5.40** provide is further evidence of the validity of the JFTS+H code. Early in the simulation $p > p_e$, and the aqueous and hydrate phases coexistent. The low boundary pressure causes rapid depressurization, leading (a) initially to gas evolution and three-phase (aqueous, gas and hydrate) when (and where) $p > p_e$ and (b) finally to ice evolution and four-phase evolution at the quadruple point when p

reaches p_Q. This pressure is maintained until the hydrate disappears, after which the pressure in the domain falls below p_Q. The excellent agreement of the JFTS+H and T+H results add to the evidence of the validity of the JFTS+H code.

Fig. 5.41 shows Q_R in Cases Test_1Pk, Test_1Pk and Test_1P_ice. The equilibrium and kinetic dissociation show very different rate patterns. With the exception of very early (and practically inconsequential) times, the rate for kinetic dissociation is lower than that for equilibrium dissociation (Cases Test_1Pk, Test_1Pk, respectively), but the two converge and practically coincide at later times. This is also reflected in **Fig. 5.42** of M_R, in which the estimate from the equilibrium dissociation is higher than that from the kinetic reaction, until the two coincide because both results correspond to the maximum possible released volume corresponding to $p_b = 2.7 \times 10^6$ Pa. Consistent with expectations, Q_R in Test_1P_ice (Moridis, 2014) is higher than in the other two cases because of the lower boundary pressure, and remains so until the hydrate is exhausted as early as 10^{-2} days because of the very low $p_b = 5.0 \times 10^5$ Pa.

5.9.6 Test_1TbS: Hydrate Formation, Equilibrium Reaction, Inhibitor Effects (Moridis, 2014)

This problem is similar to that in Test_1T, from which it differs in the initial and boundary conditions. The initial conditions involve a two-phase (water and gas) system, $p_0 4.0 \times 10^6$ Pa and $T_0 = 8.5$ °C, and an inhibitor concentration (mass fraction in the aqueous phase) $X = 1.0^{-3}$. The $x = 0$ boundary is impermeable, thermally conductive, and at a constant $T_b = 1.2$ °C. The temperature in the domain is expected to decrease below the hydrate equilibrium temperature T_e, thus causing hydrate formation. Additionally, the inhibitor effect is described by a reference temperature depression $\Delta T_{D,r} = 5.0$ $[K]$ at the reference mole fraction $Y^i_{A,r} = 0.01$ in the aqueous phase (Eq. 5.19).

The comparisons of the JFTS+H and the T+H results solutions in **Figs. 5.43** to **5.47** lead to repetition of earlier observations on the correctness and validity of the JFTS+H predictions. The colder boundary causes a continues decline in the domain temperature (**Fig**. **5.44**), and the formation of hydrate (with a density lower than that of water) causes the slight increase in pressure

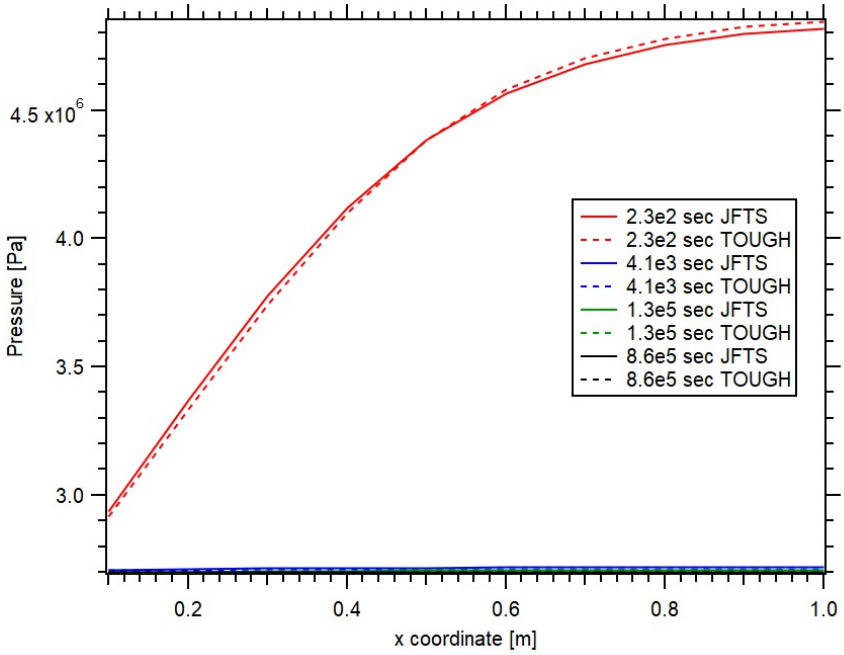

Figure 5.30: Comparison of the JFTS+H and T+H results: evolution of pressure distributions over time in the 1D problem of Test_1Pk of Moridis (2014).

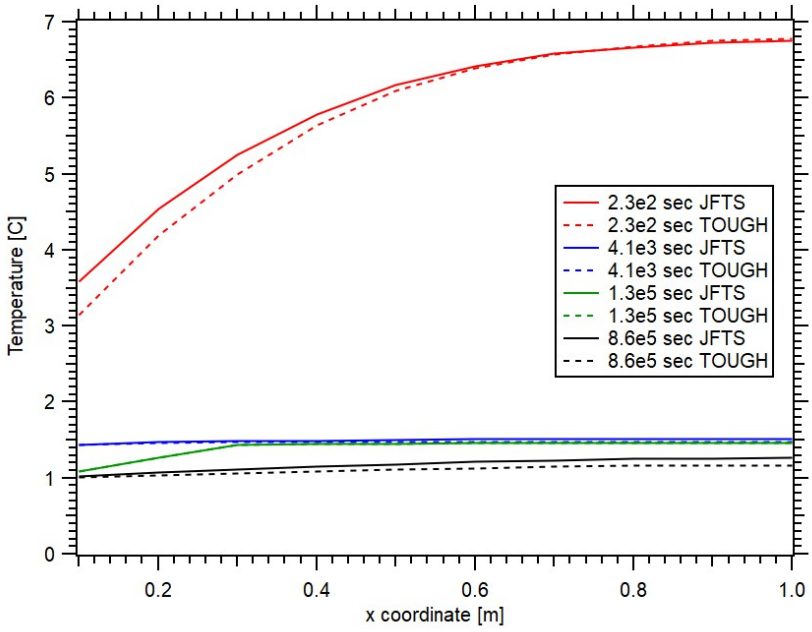

Figure 5.31: Comparison of the JFTS+H and T+H results: evolution of temperature distributions over time in the 1D problem of Test_1Pk of Moridis (2014).

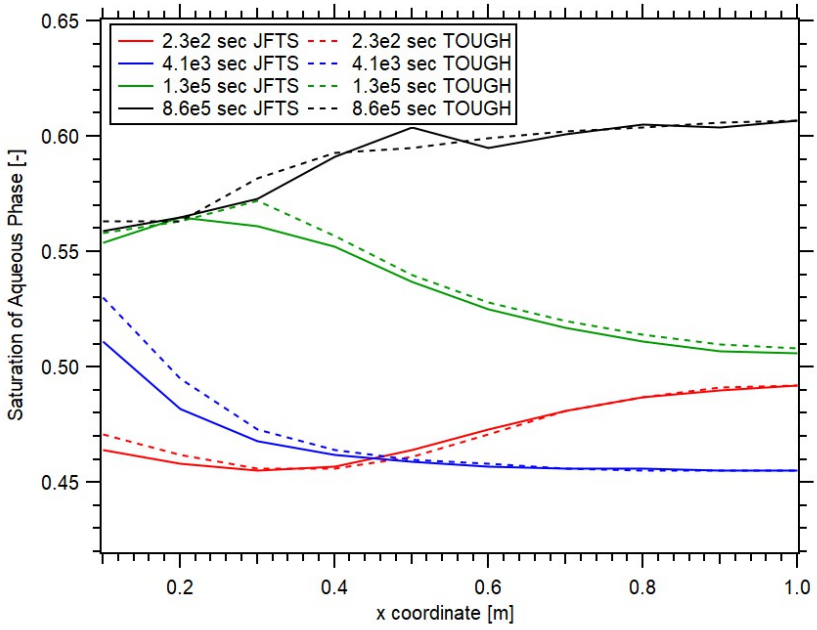

Figure 5.32: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the aqueous phase saturation over time in the 1D problem of Test_1Pk of Moridis (2014).

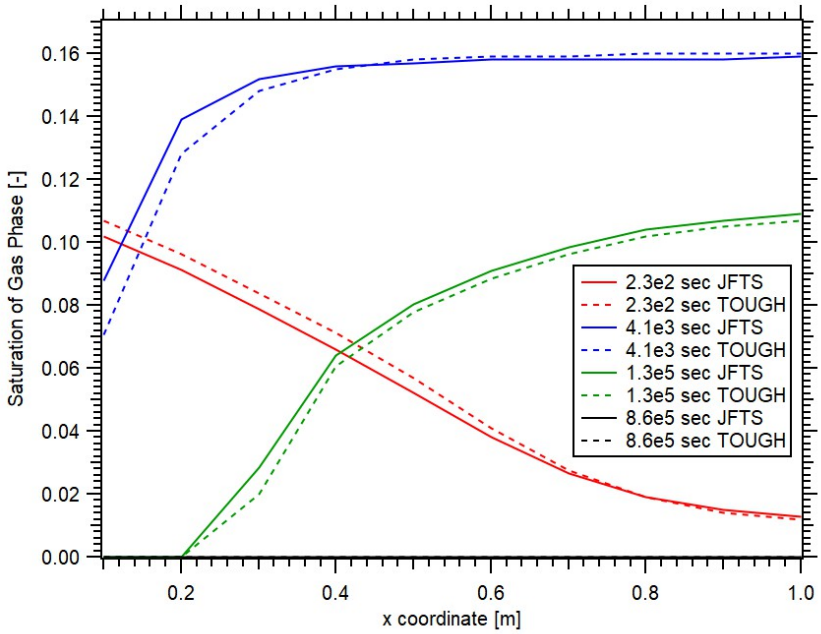

Figure 5.33: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the gas phase saturation over time in the 1D problem of Test_1Pk of Moridis (2014).

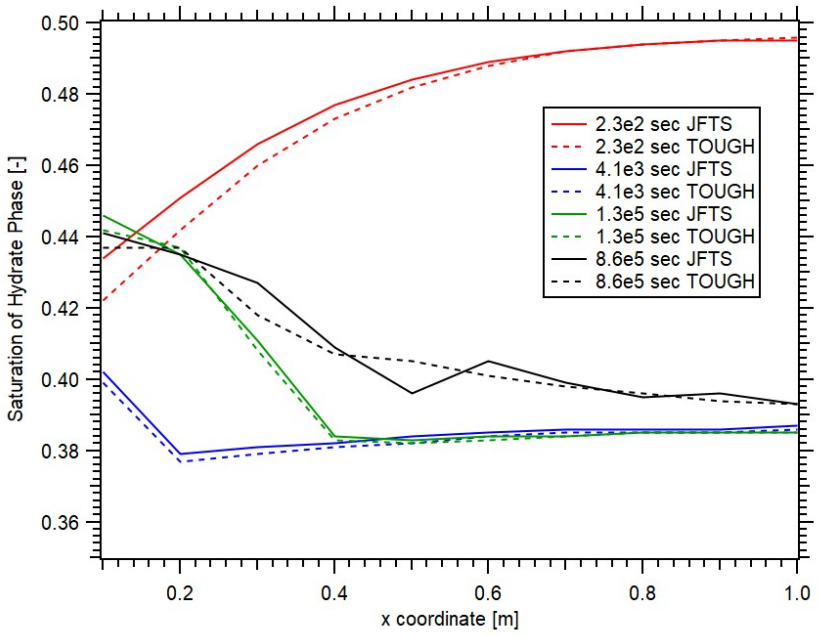

Figure 5.34: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the hydrate phase saturation over time in the 1D problem of Test_1Pk of Moridis (2014).

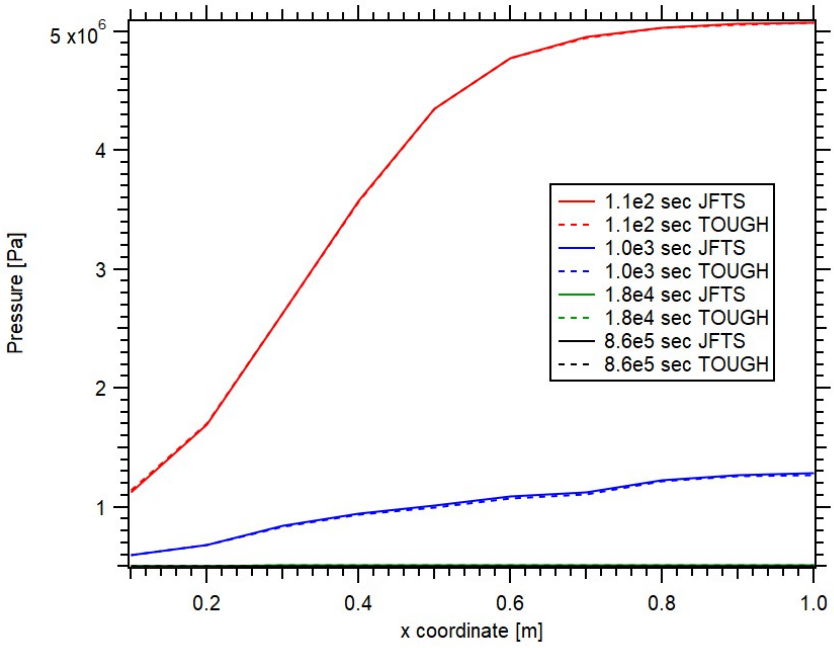

Figure 5.35: Comparison of the JFTS+H and T+H results: evolution of pressure distributions over time in the 1D problem of Test_1P_ice of Moridis (2014).

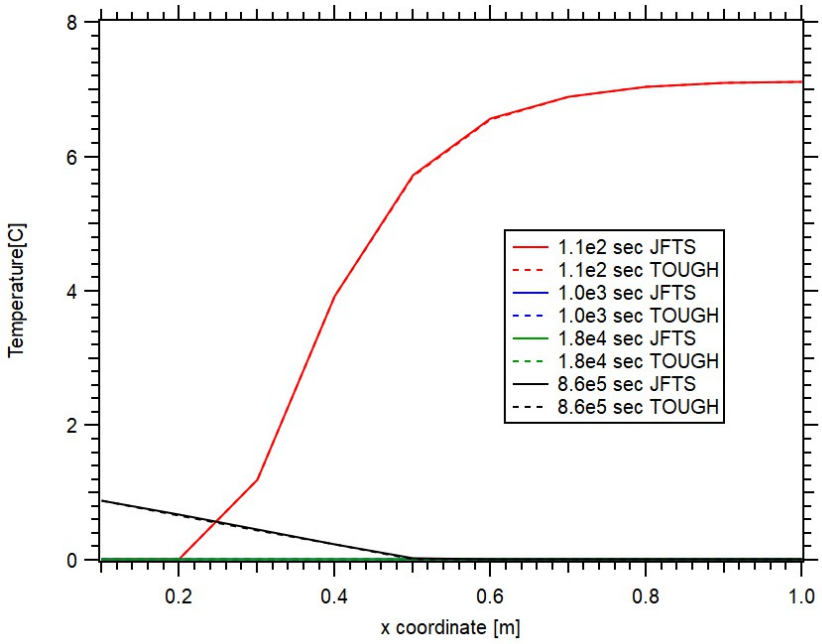

Figure 5.36: Comparison of the JFTS+H and T+H results: evolution of temperature distributions over time in the 1D problem of Test_1P_ice of Moridis (2014).

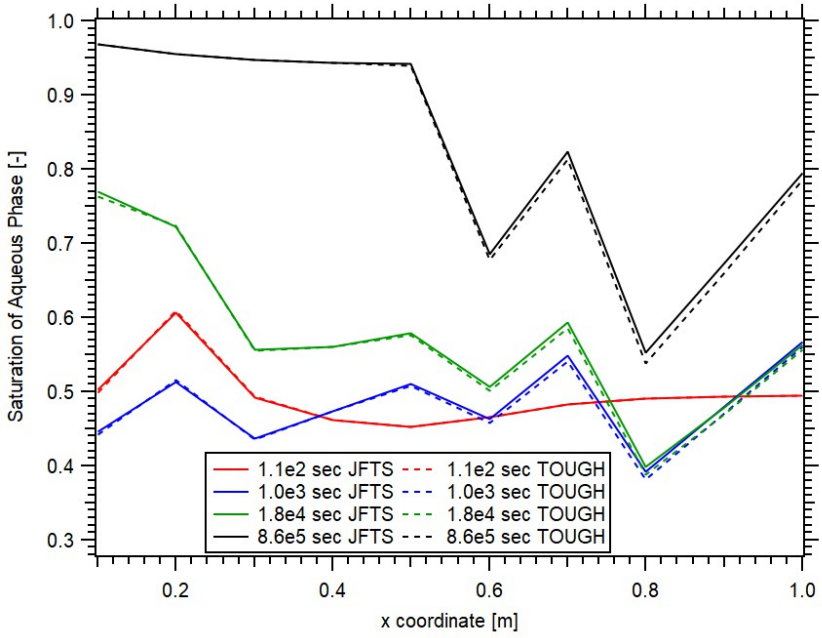

Figure 5.37: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the aqueous phase saturation over time in the 1D problem of Test_1P_ice of Moridis (2014).

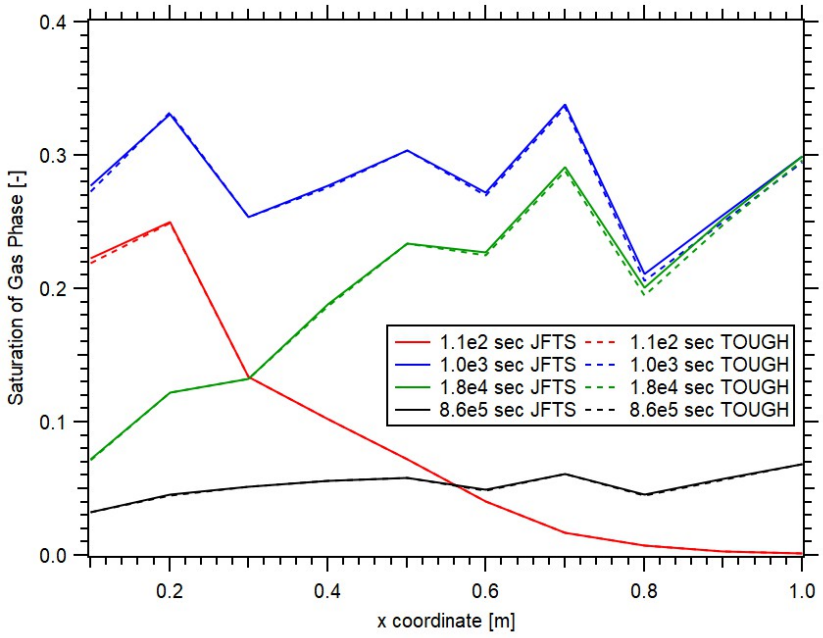

Figure 5.38: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the gas phase saturation over time in the 1D problem of Test_1P_ice of Moridis (2014).

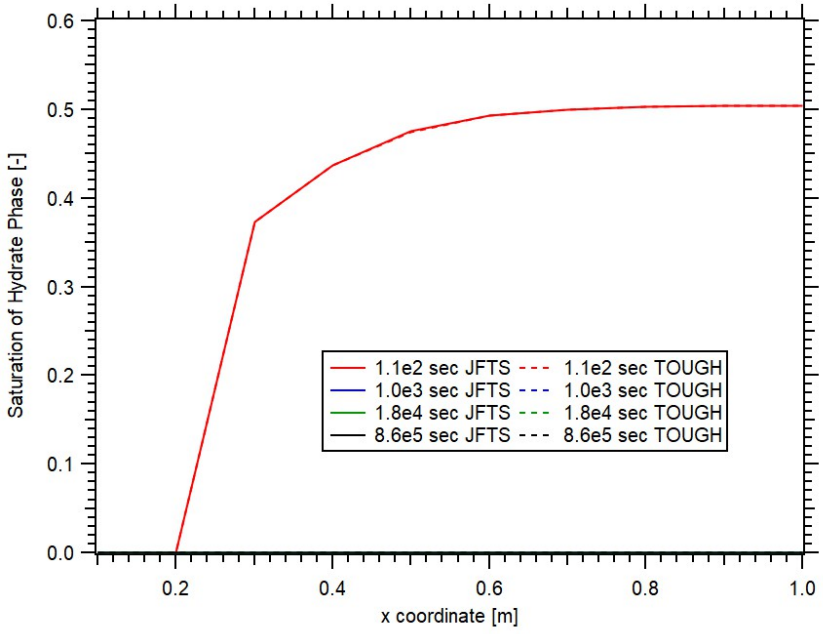

Figure 5.39: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the hydrate phase saturation over time in the 1D problem of Test_1P_ice of Moridis (2014).

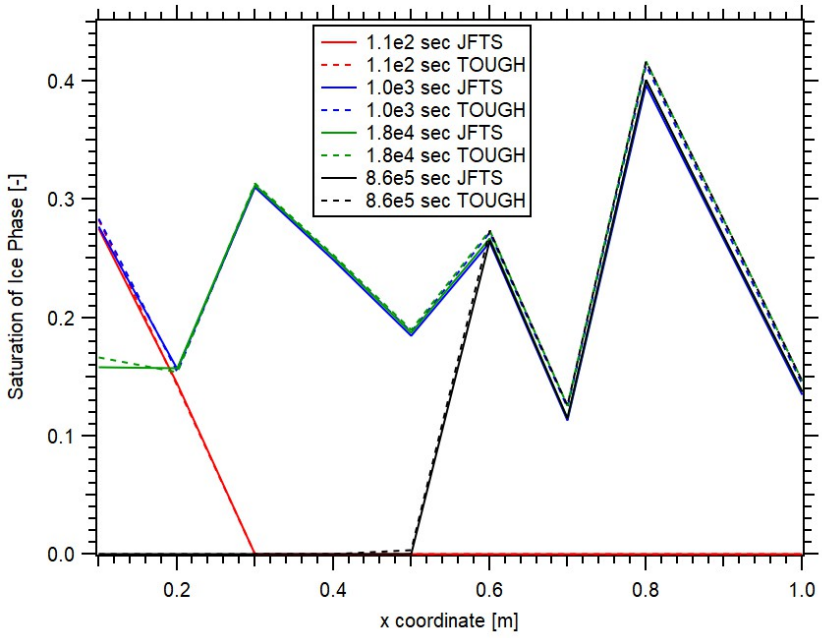

Figure 5.40: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the ice phase saturation over time in the 1D problem of Test_1P_ice of Moridis (2014).

shown in **Fig**. **5.43**. The continuous decrease in S_A and S_G (**Figs**. **5.45** and **5.46**) is consistent expectations because they are both consumed to create hydrate, the saturation of which continues to increase. Salinity keeps increasing because the formation of hydrate (which does not allow inhibitor in its crystals) leads to an increase of the inhibitor concentration in the shrinking aqueous phase

5.9.7 Test_1TbSk: Hydrate Formation, Kinetic Reaction, Inhibitor Effects (Moridis, 2014)

The only difference between this problem and Case Test_1TbS that was discussed in the previous section was the treatment of the hydration reaction, which is described here by a kinetic model. The single difference is hydrate dissociation is treated as kinetic reaction. The proximity of the JFTS+H solutions to the T+H solutions in **Figs. 5.49** to **5.53** further reinforces confidence in the validity of the JFTS+H code. The evolution of pressure, temperature, and phase saturations over time are very analogous to those in the equilibrium Test_1TbS but somewhat delayed in their response, as expected because of the kinetic retardation (which, however, is not excessive because

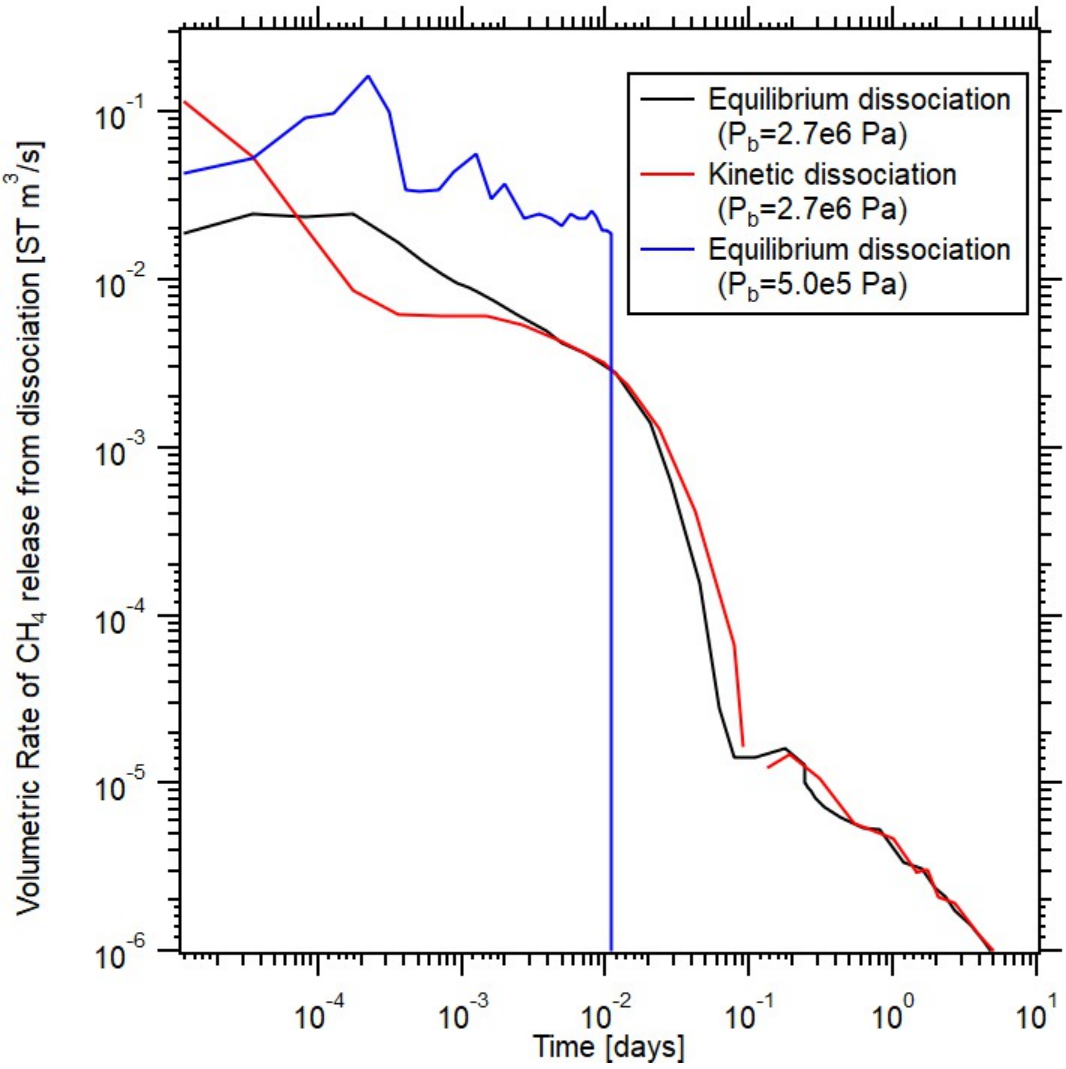

Figure 5.41: JFTS+H simulation results: comparison of the volumetric rates of $\mathrm{CH_4}$ release from depressurization-induced hydrate dissociation in the 1D problems of Test_1P of Moridis (2014), Test_1Pk of Moridis (2014) and Test_1P_ice of Moridis (2014).

of the relatively high speed of the kinetic reaction).

Figure 5.54 shows Q_R during hydrate formation. The rates associated with the equilibrium and kinetic hydrate reaction for $T_0 = 8.5$ °C are very close, with the kinetic results exhibiting a slight delay that appears to disappear after about $t = 2$ days, when Q_R practically coincide. As expected, when the initial temperature is reduced to T = 4.5 °C, Q_R (which is a measure of hydrate formation) is significantly larger because the lower temperature accelerates the formation of the hydrate. Note that, in all cases, the initial rapid increase in Q_R is followed by a gradual decrease

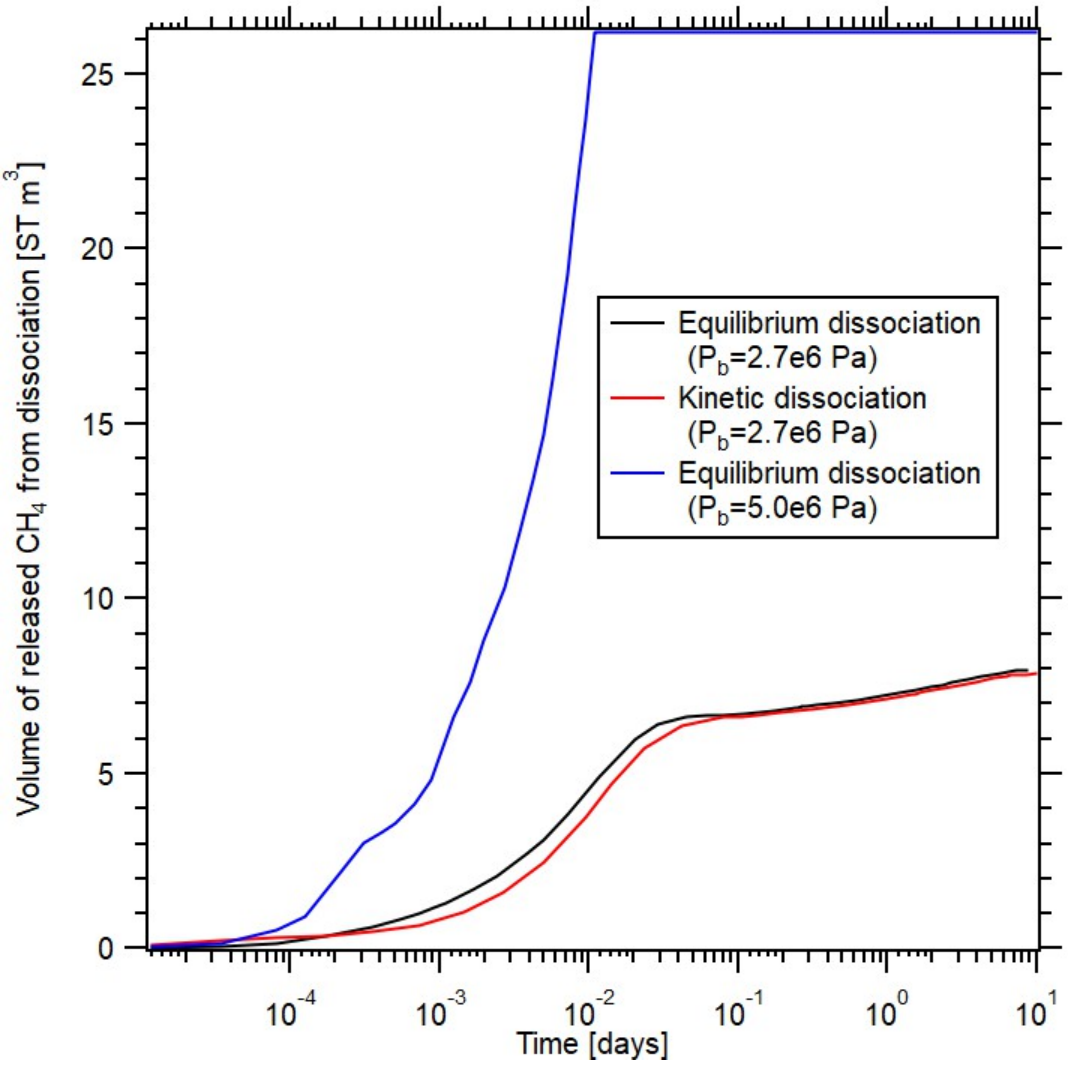

Figure 5.42: JFTS+H simulation results: comparison of the cumulative volumes of $\mathrm{CH_4}$ release from depressurization-induced hydrate dissociation in the 1D problems of Test_1P of Moridis (2014), Test_1Pk of Moridis (2014) and Test_1P_ice of Moridis (2014).

as the maximum amount of hydrate that is possible for a given T_0 and X_0 is reached. The same behavior is evident in the evolution of M_R in **Fig. 5.55**: the volumes associated with the kinetic and equilibrium formation are very similar (with the equilibrium-associated ones being slightly larger) for $T_0 = 8.5$ °C, and the volume for $T_0 = 5.5$ °C is significantly larger because of higher consumption rate and more hydrate formation.

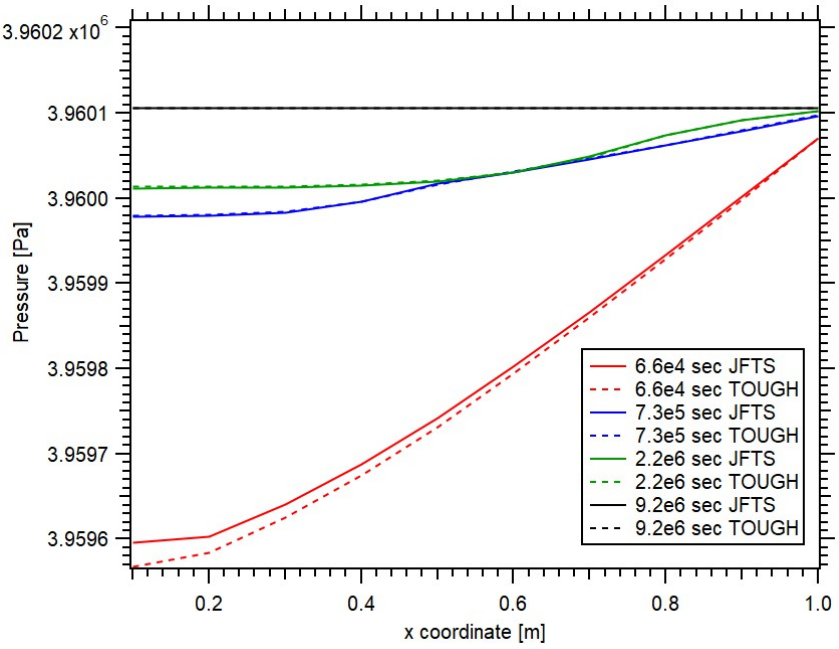

Figure 5.43: Comparison of the JFTS+H and T+H results: evolution of pressure distributions over time in the 1D problem of Test_1TbS of Moridis (2014).

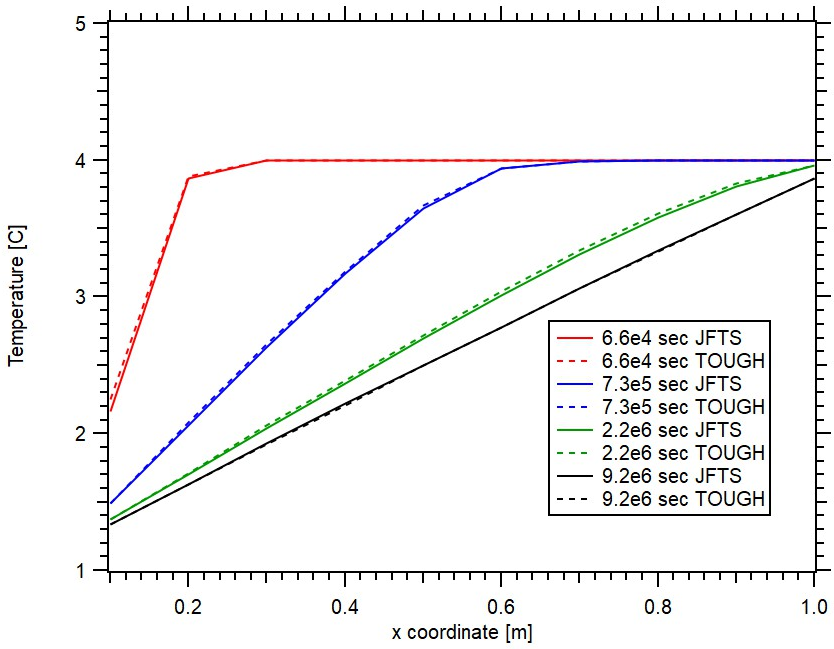

Figure 5.44: Comparison of the JFTS+H and T+H results: evolution of temperature distributions over time in the 1D problem of Test_1TbS of Moridis (2014).

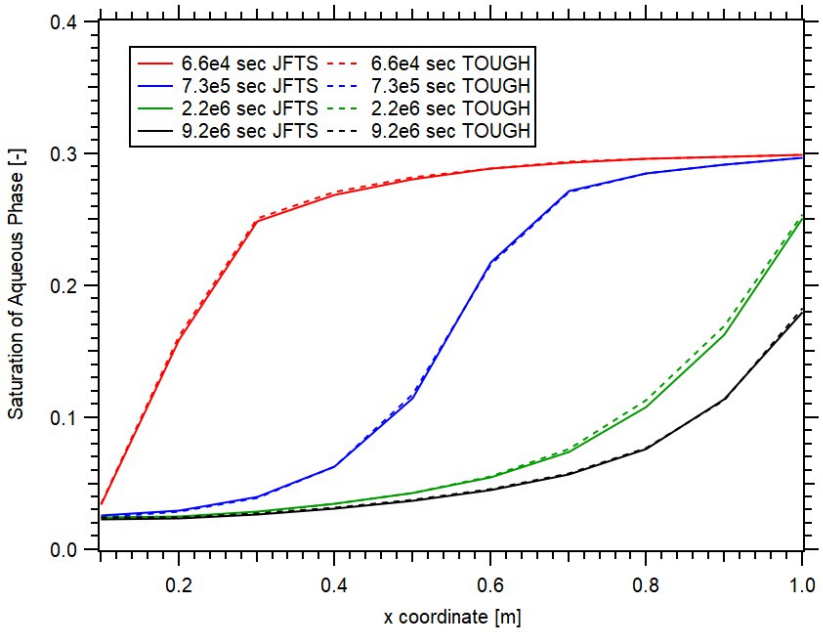

Figure 5.45: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the aqueous phase saturation over time in the 1D problem of Test_1TbS of Moridis (2014).

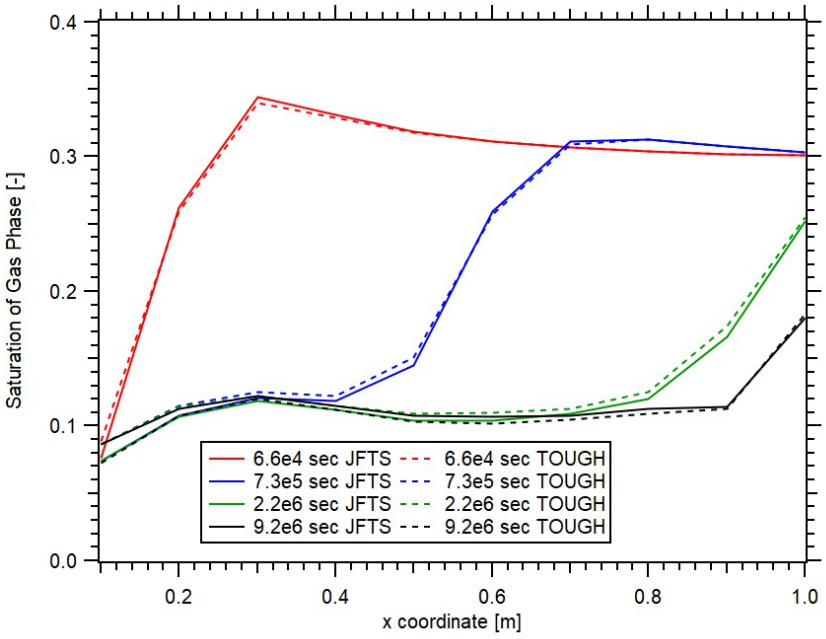

Figure 5.46: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the gas phase saturation over time in the 1D problem of Test_1TbS of Moridis (2014).

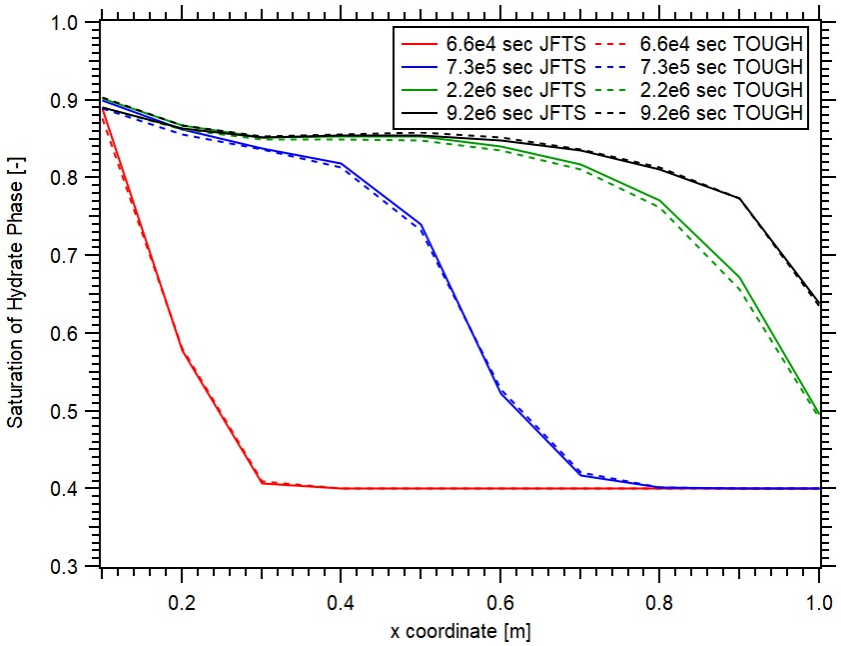

Figure 5.47: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the hydrate phase saturation over time in the 1D problem of Test_1TbS of Moridis (2014).

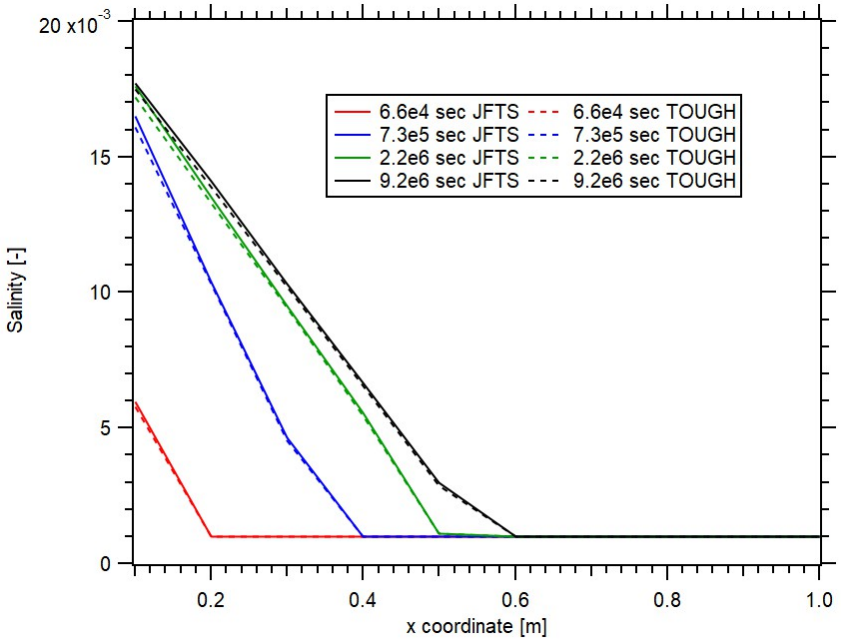

Figure 5.48: Comparison of the JFTS+H and T+H results: evolution of the salinity over time in the 1D problem of Test_1TbS of Moridis (2014).

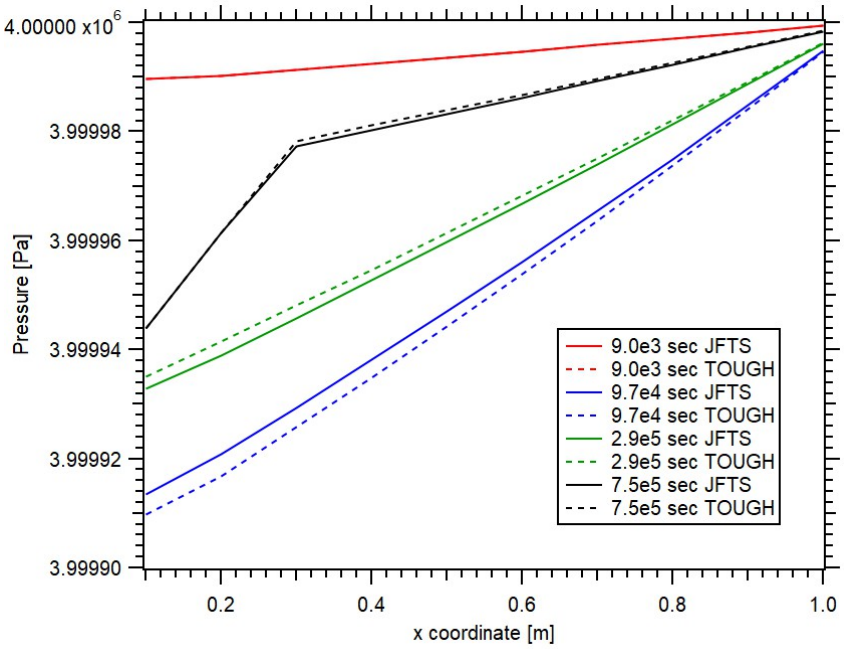

Figure 5.49: Comparison of the JFTS+H and T+H results: evolution of pressure distributions over time in the 1D problem of Test_1TbSk of Moridis (2014).

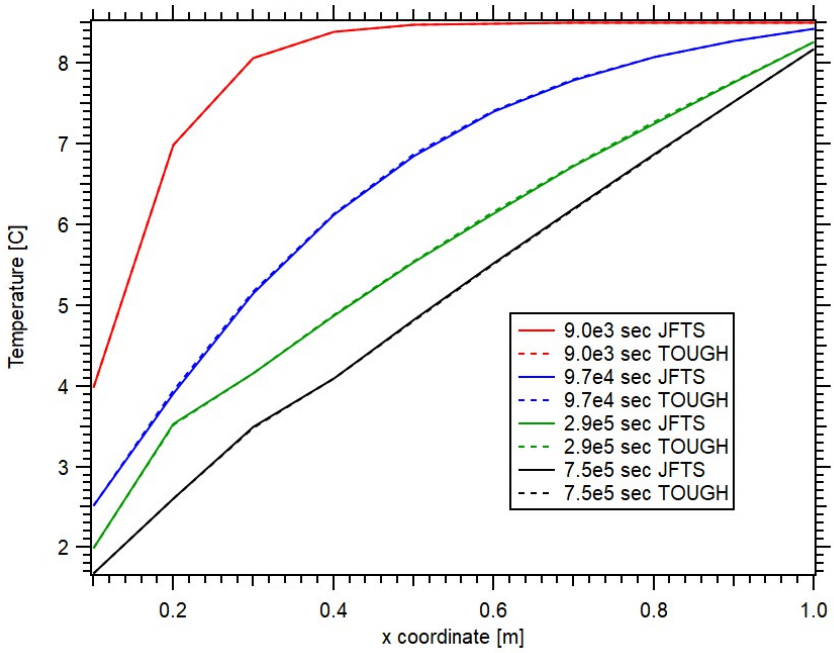

Figure 5.50: Comparison of the JFTS+H and T+H results: evolution of temperature distributions over time in the 1D problem of Test_1TbSk of Moridis (2014).

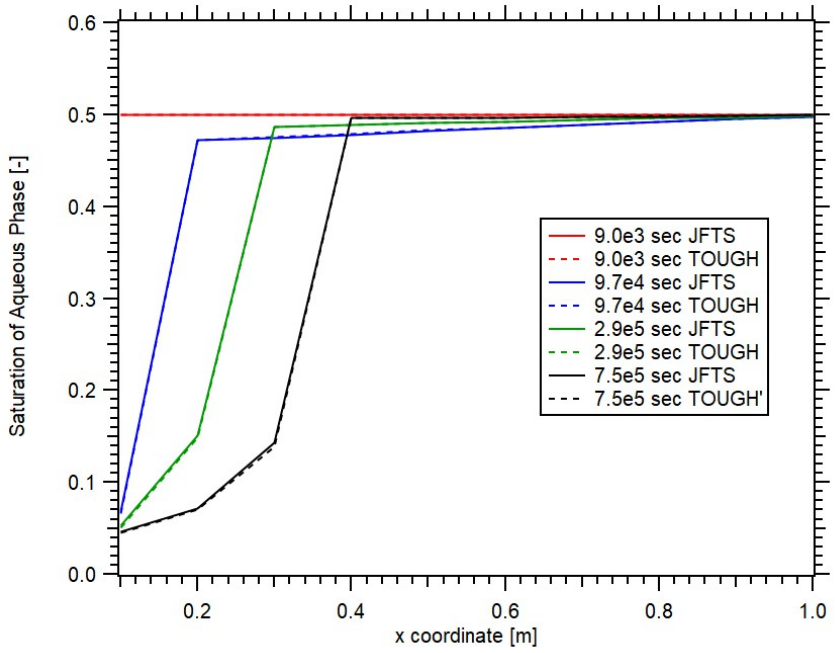

Figure 5.51: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the aqueous saturation over time in the 1D problem of Test_1TbSk of Moridis (2014).

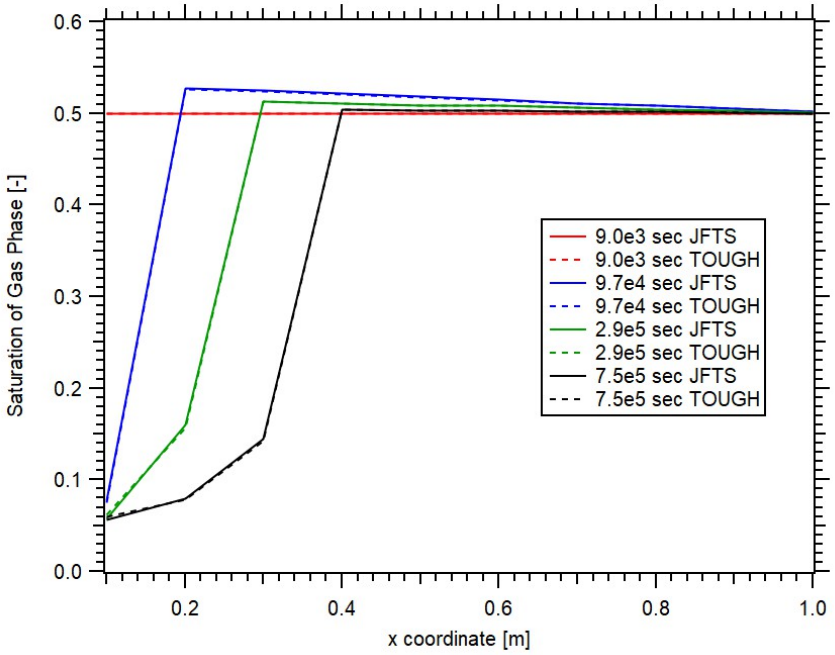

Figure 5.52: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the gas saturation over time in the 1D problem of Test_1TbSk of Moridis (2014).

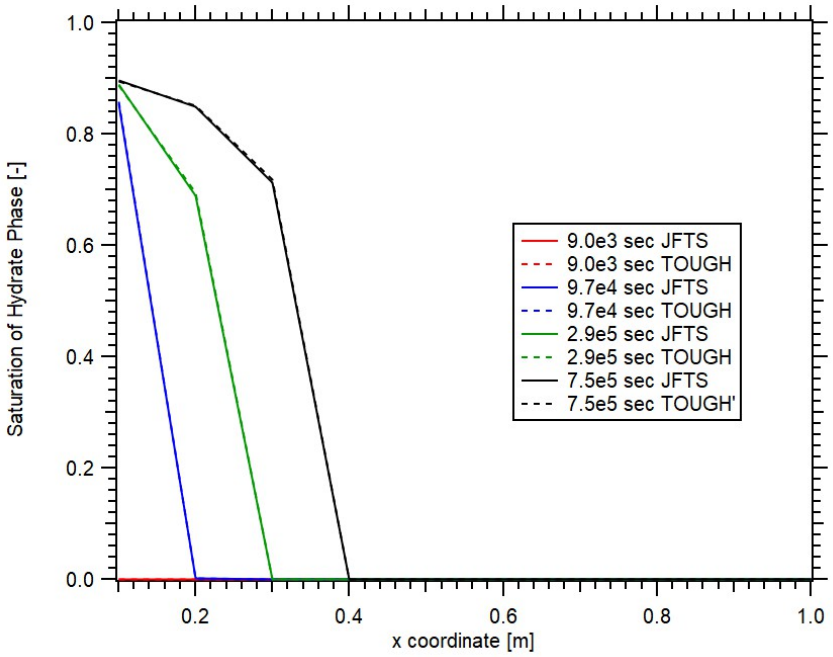

Figure 5.53: Comparison of the JFTS+H and T+H results: evolution of the spatial distribution of the hydrate saturation over time in the 1D problem of Test_1TbSk of Moridis (2014).

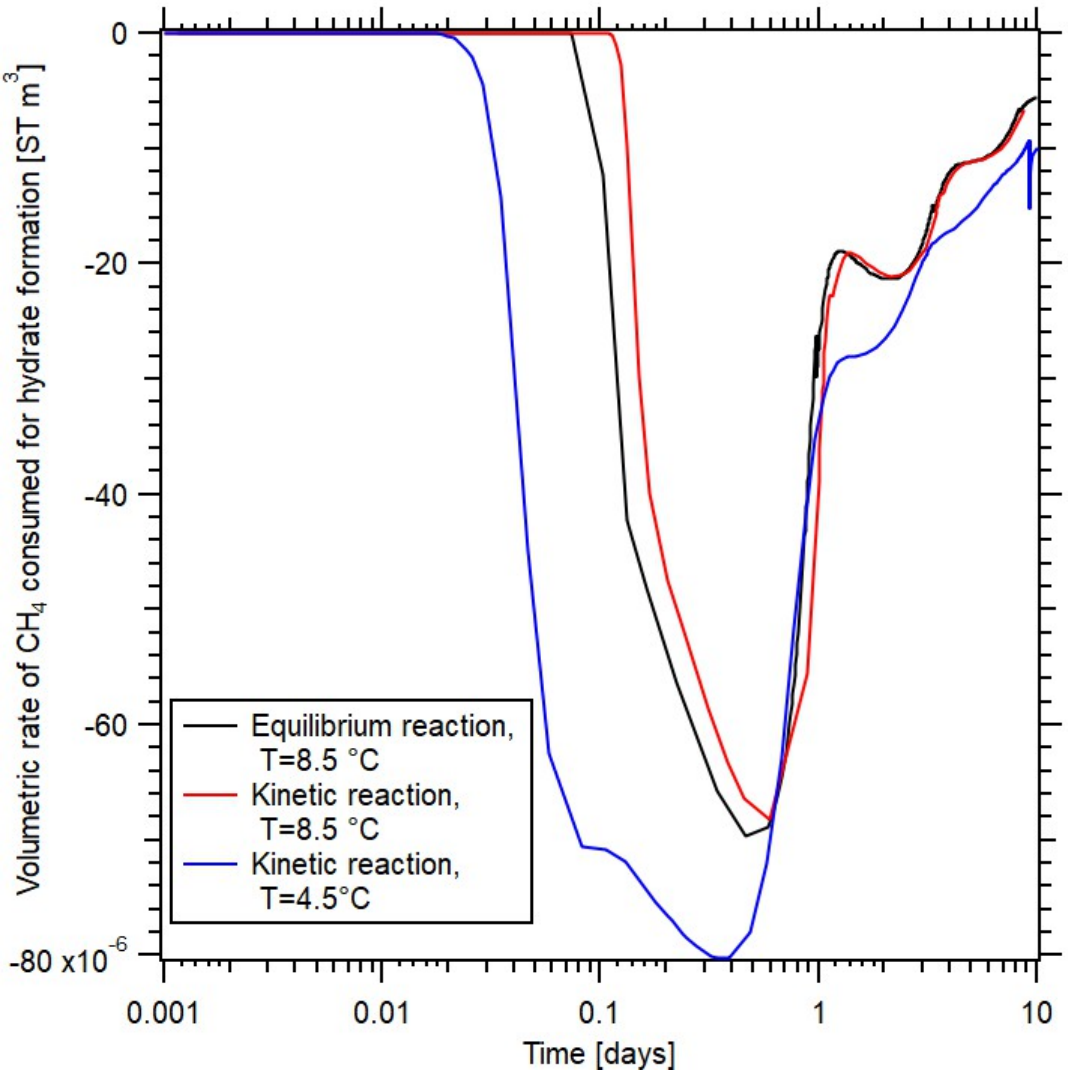

Figure 5.54: JFTS+H simulation results: comparison of the volumetric rates of CH_4 consumed during hydrate formation in the 1D problems of Test_1TbS of Moridis (2014) and Test_1TbSk of Moridis (2014).

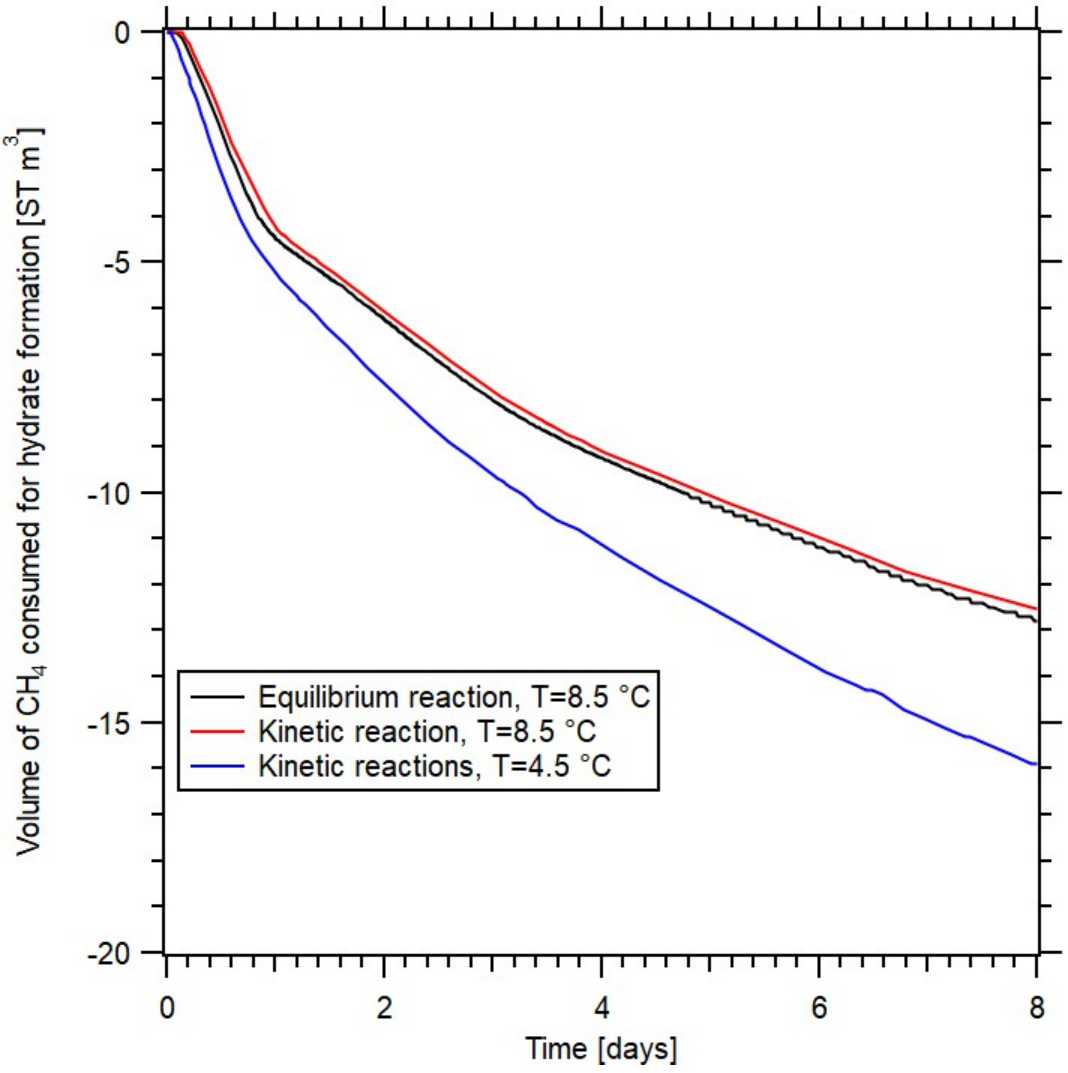

Figure 5.55: JFTS+H simulation results: comparison of the cumulative volumes of CH_4 consumed during hydrate formation in the 1D problems of Test_1TbS of Moridis (2014) and Test_1TbSk of Moridis (2014).

6. RESULTS, ANALYSIS, AND DISCUSSION

6.1 Validation and Field Case Application 1: The Krishna-Godawari Basin

6.1.1 Description of the Geologic System

The geologic model of the reservoir is composed of three hydrate-bearing layers and two hydrate-free interlayers, all of which define the production target. This reservoir system is overlain and underlain by very-low-permeability muds that form the overburden and underburden, respectively. Thus, this system represents a Class 3 hydrate deposit according to the nomenclature proposed by Moridis *et al.* (2009). The well is completed over the entire thickness of the production target (i.e., the hydrate-bearing layers and the hydrate-free interlayers), and gas production is based on depressurization-induced dissociation of the CH_4-hydrates for the reasons explained by Moridis and Reagan (2007).

The geologic model of the hydrate accumulation and the well configuration are shown in **Fig. 6.1**. This represents a real-life oceanic hydrate accumulation offshore India and is based on an early interpretation of geophysical surveys at the NGHP-02-09 site in the Krishna-Godawari basin (Boswell *et al.,* 2019). A recent study of fluid production from this deposit – based on the latest interpretation of the geophysical surveys and the consequent update of the geologic model – can be found in Moridis *et al.* (2019b), where the system dimensions and other specifics are discussed in detail.

The initial conditions and the flow and thermal properties of the system used in the simulations are listed in **Table 6.1**, which also describes the relative permeability and capillary pressure of all subdomains of the system, including the vertical well. The initial P and T distributions in the reservoir profile were determined by following the initialization process described by Moridis and Reagan (2007a; Moridis *et al.,* 2019c). The uppermost and lowermost grid block layers were assigned time-invariable constant conditions and properties. The temperature at the top (ocean floor) and bottom boundaries (several 100 m below the bottom of the H3 hydrate-bearing layer)

are 0.482 °C and 51.2 °C, respectively. The initial pressure distribution is assumed to follow the hydrostatic distribution, which is a valid assumption based on evidence from practically all natural hydrate systems. The initial pressure and thermal gradients are computed by the JFTS+H, and the initialization simulation was run until thermal, hydraulic, and chemical equilibrium steady-state conditions were achieved throughout the domain. The initial pressure and temperature distributions along z-direction are shown in **Figs. 6.2** and **6.3**.

6.1.2 Domain Discretization

The system is described into a 2-dimensional cylindrical grid that includes 170x149 = 25,330 gridlocks in (r,z), resulting in a Jacobian matrix equation comprising a total of 170x149x4= 101,320 simultaneous linear equations. The JFTS+H parallel simulation decomposed the entire domain using the METIS application (Karypis, 2013) and divided it into a number of subdomains equal to the number of processors. The approach followed by the METIS library in the subdivision of the domain is as follows: (a) equitable distribution of the number of gridblocks among all the available processors (to the extent possible), but with no guarantee of contiguity of all gridblocks assigned to any given processor, and (b) minimization of the number of communications among processors. **Fig. 6.4** shows an example of the METIS domain decomposition that shows the subdomains into which METIS partitions the domain when 16 processors are used for the simulation.

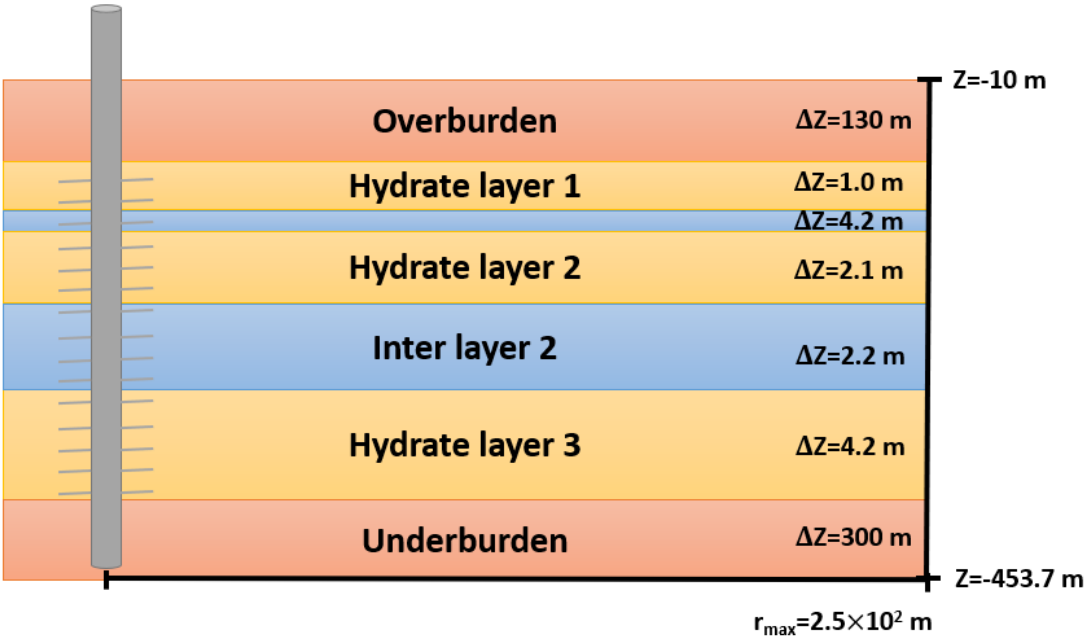

Figure 6.1: The layered geological model of the reservoir used in the 2D study.

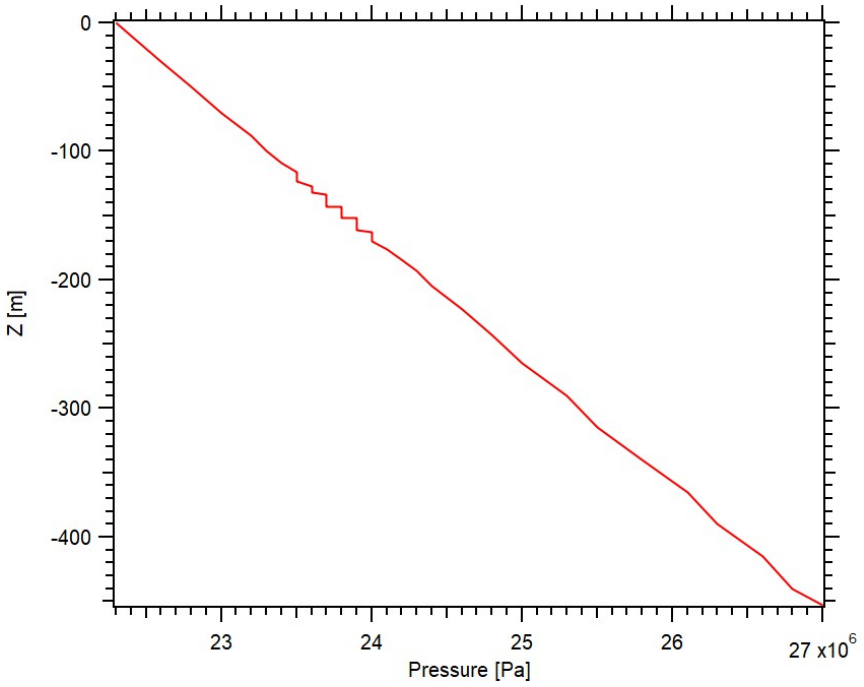

Figure 6.2: Initial pressure distribution along z-direction in the 2D problem of **Fig. 6.1**.

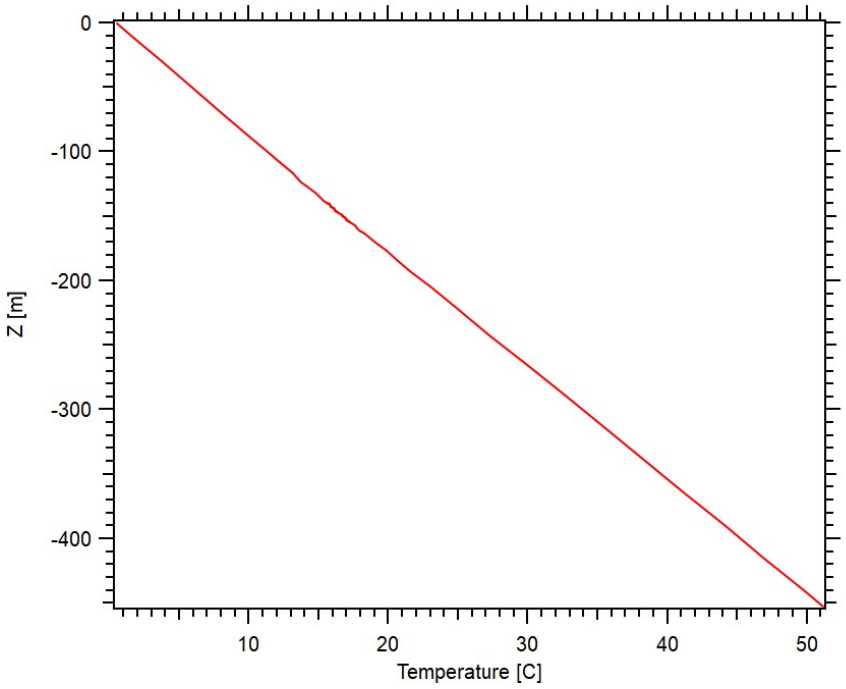

Figure 6.3: Initial temperature distribution along z-direction in the 2D problem of **Fig. 6.1**.

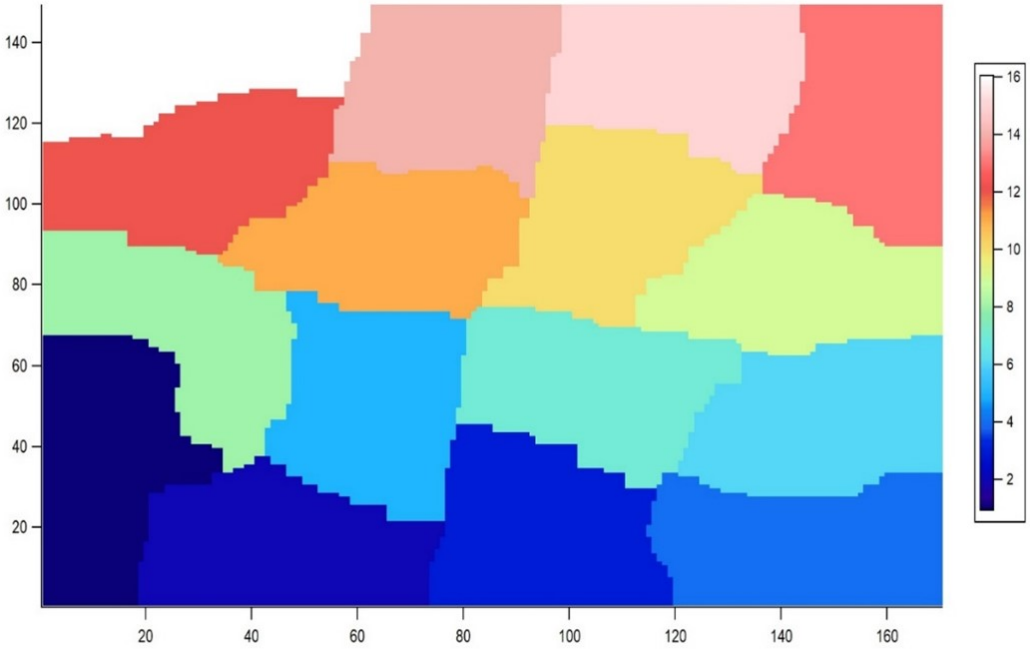

Figure 6.4: The domain decomposition of the 2D problem that results from the application of METIS (Karypis, 2013) in a JFTS+H simulations that involves 16 processors.

Table 6.1: Properties of the hydrate deposit in **Fig. 6.1**.

Initial pressure at ocean floor (P_T)	2.24×10^7 Pa
Pressure distribution	Hydrostatic
Initial temperature at ocean floor (T_T)	0.482 °C
Initial temperature at bottom boundary (T_B)	51.2 °C
Gas composition	100% CH_4
Initial saturations in the H1, H2, H3	$S_H = 0.64$
Intrinsic permeability k_r of H1, H2, H3	1.78×10^{-13} m^2
Porosity ϕ of H1, H2, H3	0.45
Intrinsic permeability k_r of M1, M2	2.04×10^{-16} m^2
Porosity ϕ of M1, M2	0.67
Intrinsic permeability of OB, UB	1.78×10^{-16} m^2
Porosity ϕ of OB, UB	0.5
Wet and dry thermal conductivity	$k_{\theta D} = 1.0$ W/m/K; $k_{\theta W} = 1.45\ W/m/K$
Specific heat C_R	800 J/kg/K
Grain density ρ_R	2650 kg/m^3
Relative permeability model	$k_{rA} = (S_A^*)^n$; $k_{rG} = (S_G^*)^m$ S_A^*=$(S_A - S_{irA})$/$(1 - S_{irA})$ S_G^*=$(S_G - S_{irG})$/$(1 - S_{irA})$
Relative permeability exponents m, n	$m = 5.0; n = 3.0$
Capillary pressure model	$P_{cap} = -P_0 \left[(S_A^*)^{(-1/\lambda)} - 1 \right]^{\lambda}$
P_0 and λ in H1, H2, H3	10^4 Pa; 0.45
S_{irA}, S_{irG} in H1, H2, H3	$0.25, 0.01$
P_0 and λ in M1, M2	10^5 Pa; 0.15
S_{irA}, S_{irG} in M1, M2	$0.5, 0.05$
P_0 and λ in OB, UB	10^5 Pa; 0.15
S_{irA}, S_{irG} in OB, UB	$0.5, 0.05$

6.1.3 Simulation Results

The JFTS+H simulator provides estimates of all relevant parameters, including Q_R, M_R and volume of the released methane gas, V_R and produced reservoir fluids, Q_P and Q_A, the amount of free gas, V_F accumulating in the reservoir, the salinity, the mass rates and cumulative amounts of produced fluids, M_P and M_A. Additionally, the code has capabilities that allow monitoring of (a) conditions in any subdomain of arbitrary shape and volume and (b) flows of fluids and heat (rates and cumulative amounts) across any desirable interface of arbitrary size and area. The JFTS+H and the T+H predictions shown in **Figs. 6.5** to **6.11** are in excellent agreement, thus providing further

evidence of the validity of the JFTS+H code. Minor differences between the two sets of solutions are attributed to differences in the processors, the languages and the solver specifications, as well as the differences in the $P - T$ equilibrium equations and the thermophysical property estimates, as JFTS+H includes the most recent scientific advances and relationships for their computation.

Note that although the JFTS+H simulations were conducted on both the Ada cluster and the macOS desktop system, only a single set of the JFTS+H results are shown in **Figs. 6.5** to **6.11** because of the virtual coincidence of the parallel simulations regardless of (a) the computational platform and (b) the number of the processors.

Fig. 6.5 shows Q_R. At the beginning of the simulation Q_R shows a very sharp peak. after reaching some point, Q_R increases monotonically. **Fig. 6.5** shows M_R. **Fig. 6.7** and **6.8** show Q_P and Q_A from the single vertical well, respectively. Similar to Q_R, both Q_P and Q_A show very sharp peaks and increase until the end of the simulation. **Fig. 6.9** shows the water/gas ratio, R_{wg}. At the very beginning of the simulation, R_{wg} shows extremely high values (i.e. more than 300) and declines rapidly. After 1 day, R_{wg} persists at high levels during simulation. **Fig. 6.10** shows the salinity, X_s, and X_s show the highest values at the beginning of the simulation. **Fig. 6.11** shows V_F and V_F increase monotonically.

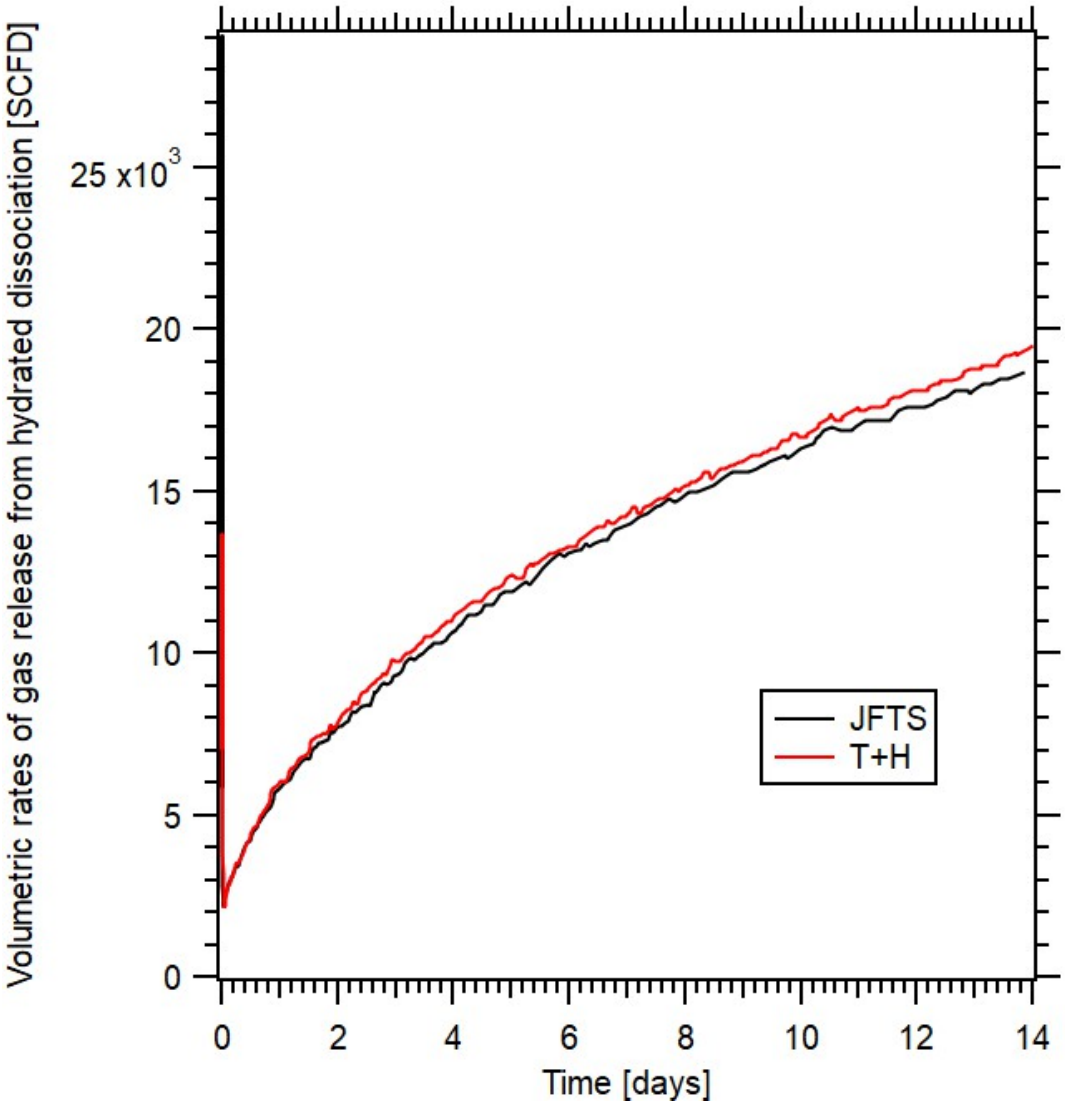

Figure 6.5: Comparison of the JFTS+H and T+H results: volumetric rates of the CH_4 gas release from hydrate dissociation in the 2D problem of **Fig. 6.1**. Note: the Ada and MacPro results coincide.

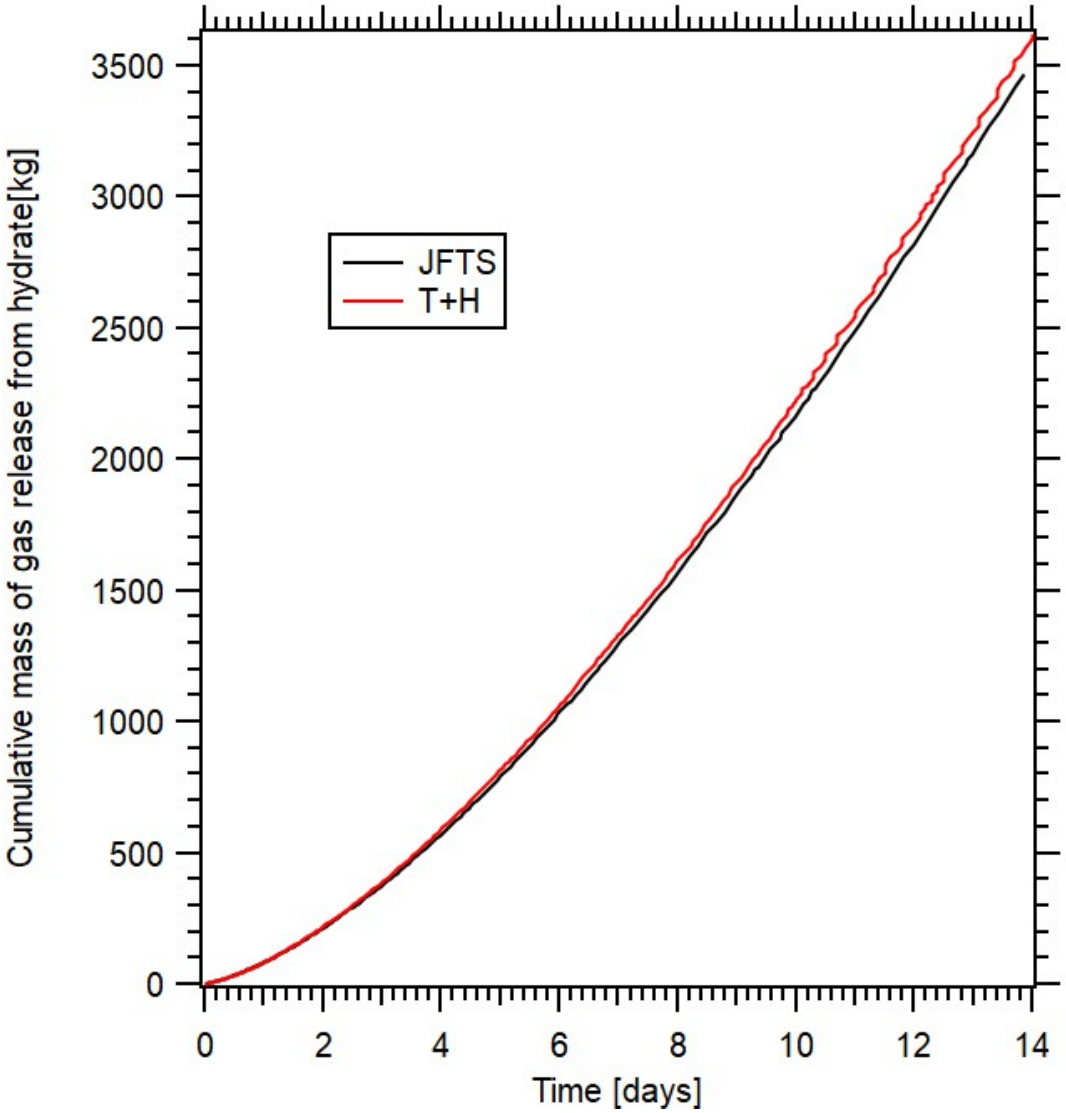

Figure 6.6: Comparison of the JFTS+H and T+H results: cumulative mass of the CH_4 gas released from hydrate dissociation in the 2D problem of **Fig. 6.1**.

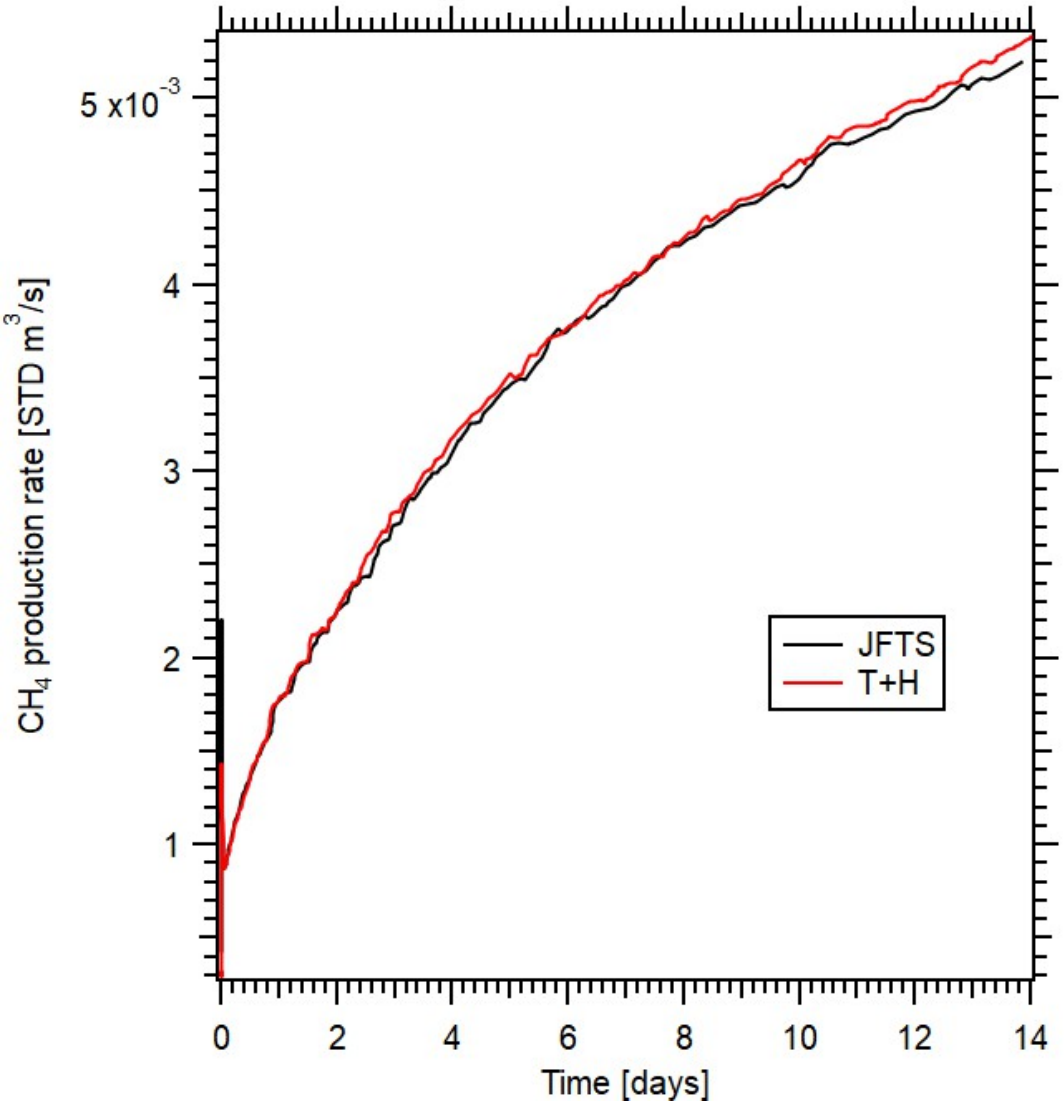

Figure 6.7: Comparison of the JFTS+H and T+H results: volumetric rates of the CH_4 production in the 2D problem of **Fig. 6.1**.

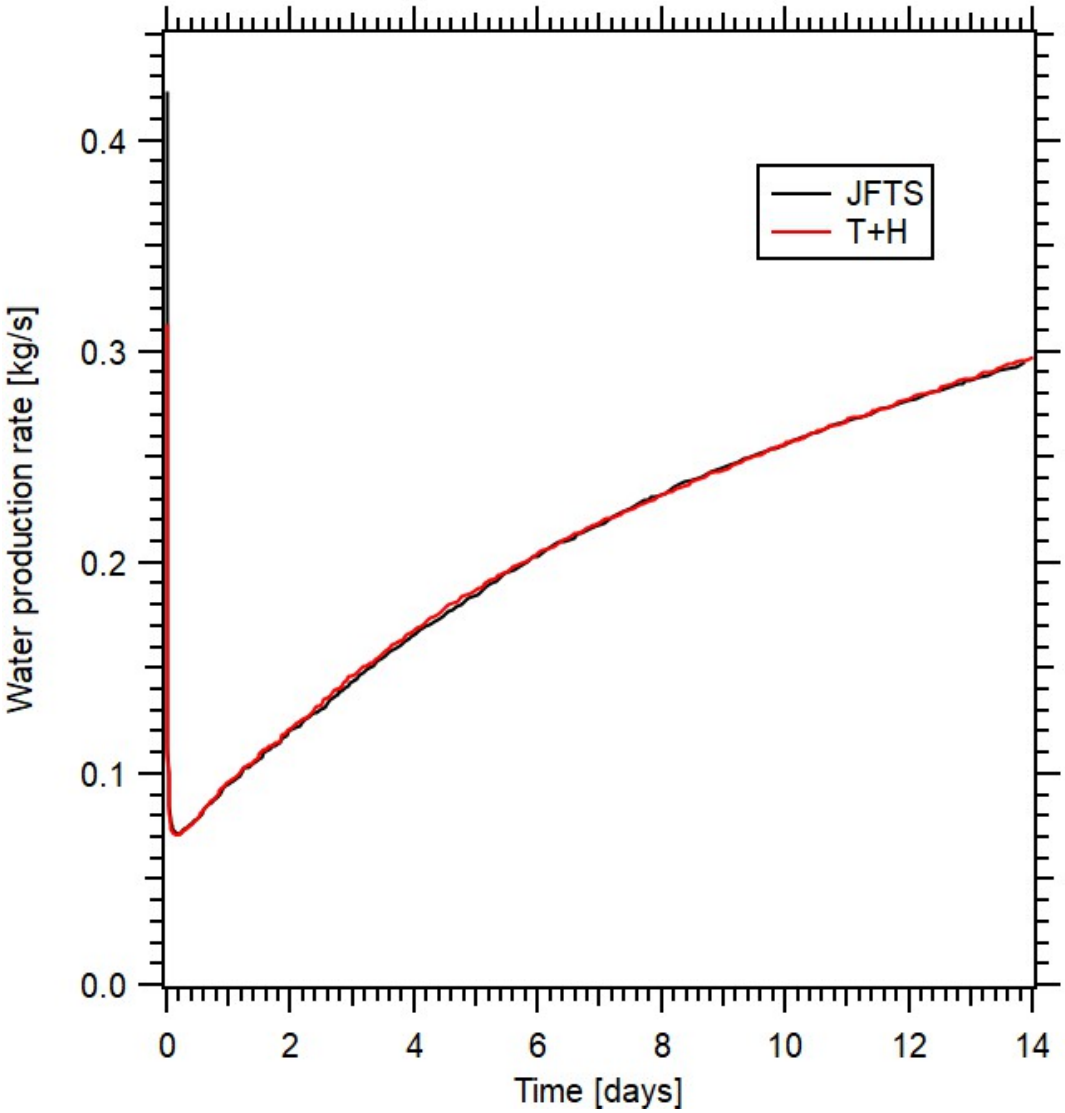

Figure 6.8: Comparison of the JFTS+H and T+H results: mass rates of the water production in the problem of **Fig. 6.1**.

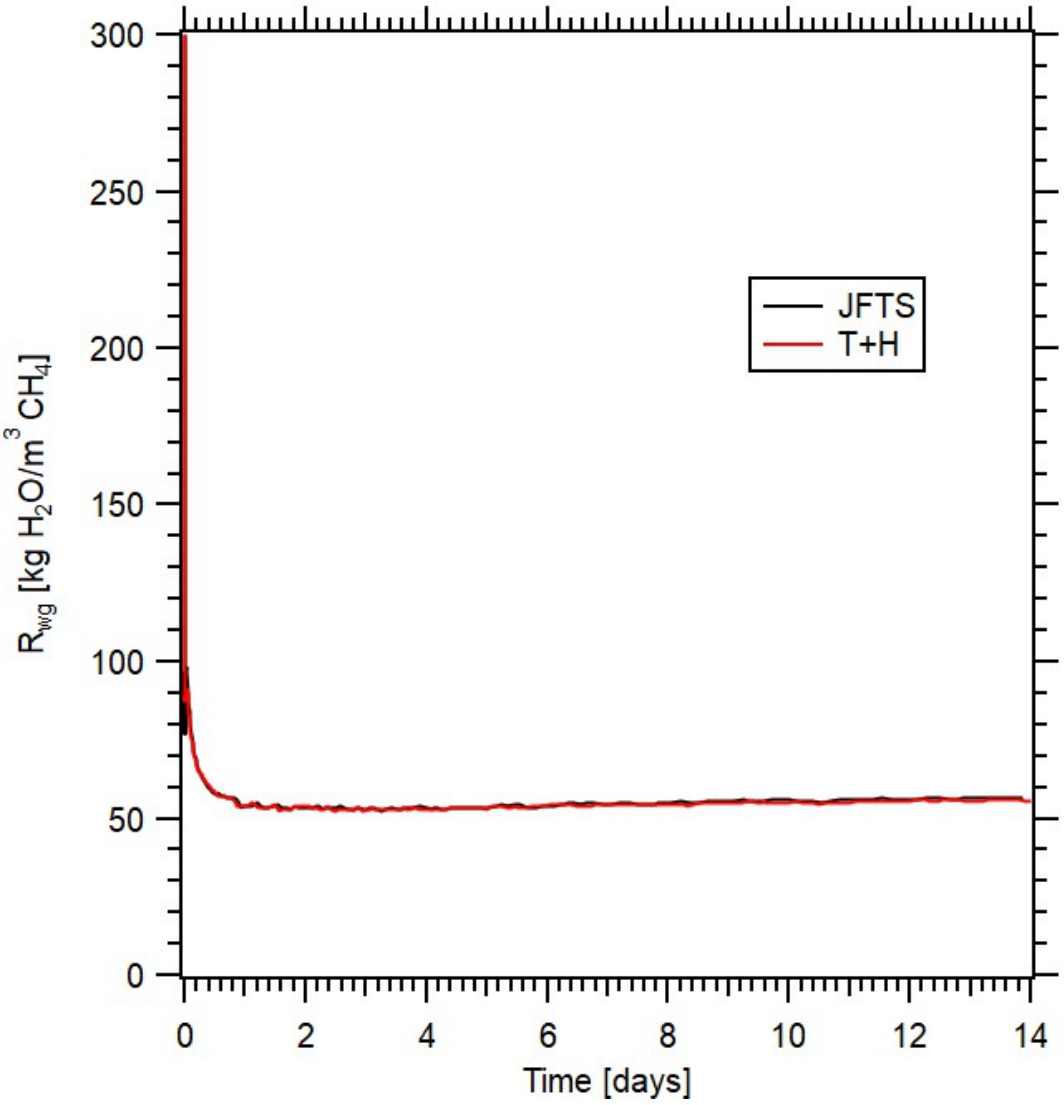

Figure 6.9: Comparison of the JFTS+H and T+H results: the water/gas ratio in the 2D problem of **Fig. 6.1**.

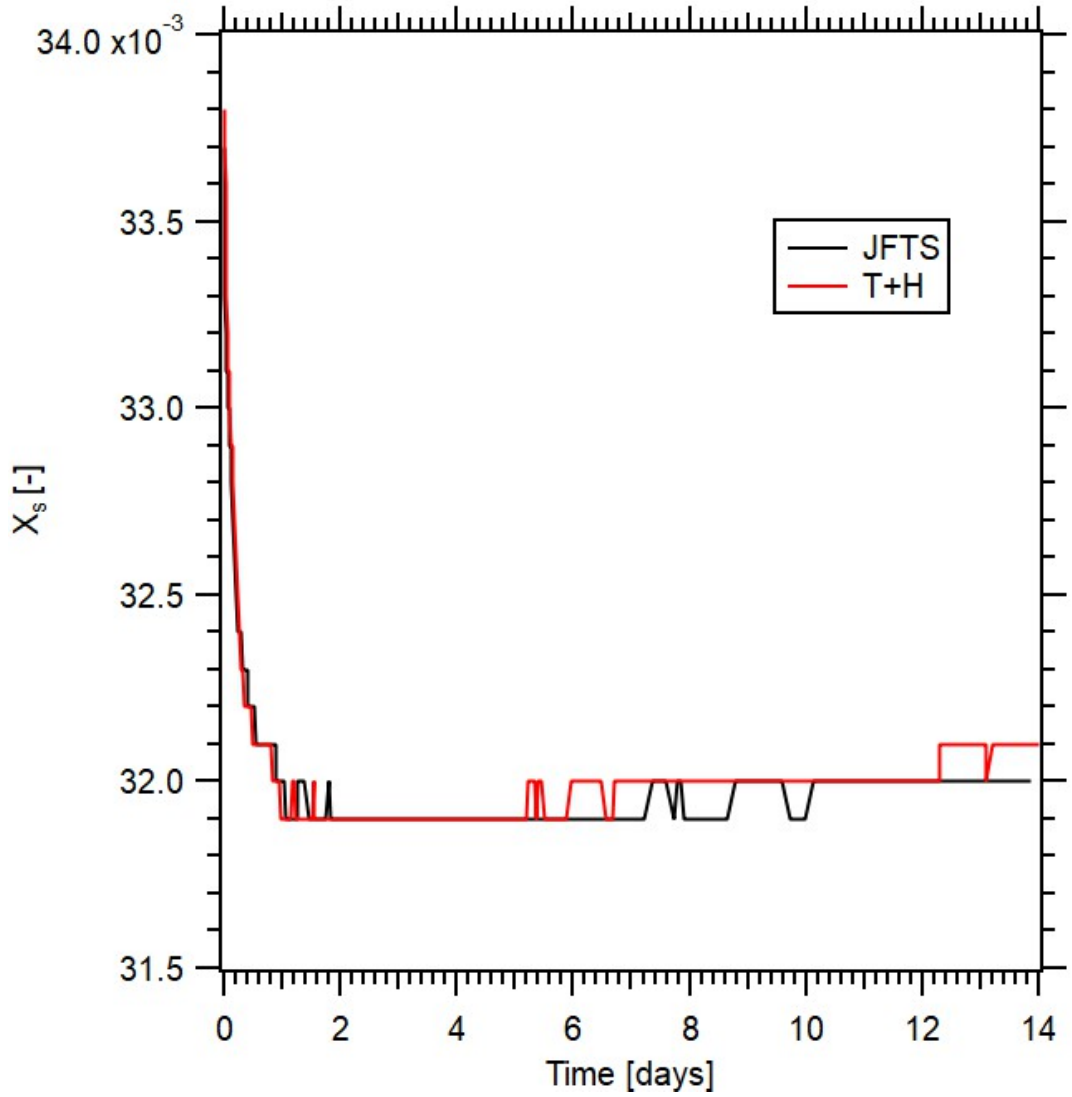

Figure 6.10: Comparison of the JFTS+H and T+H results: the salinity in the 2D problem of **Fig. 6.1**.

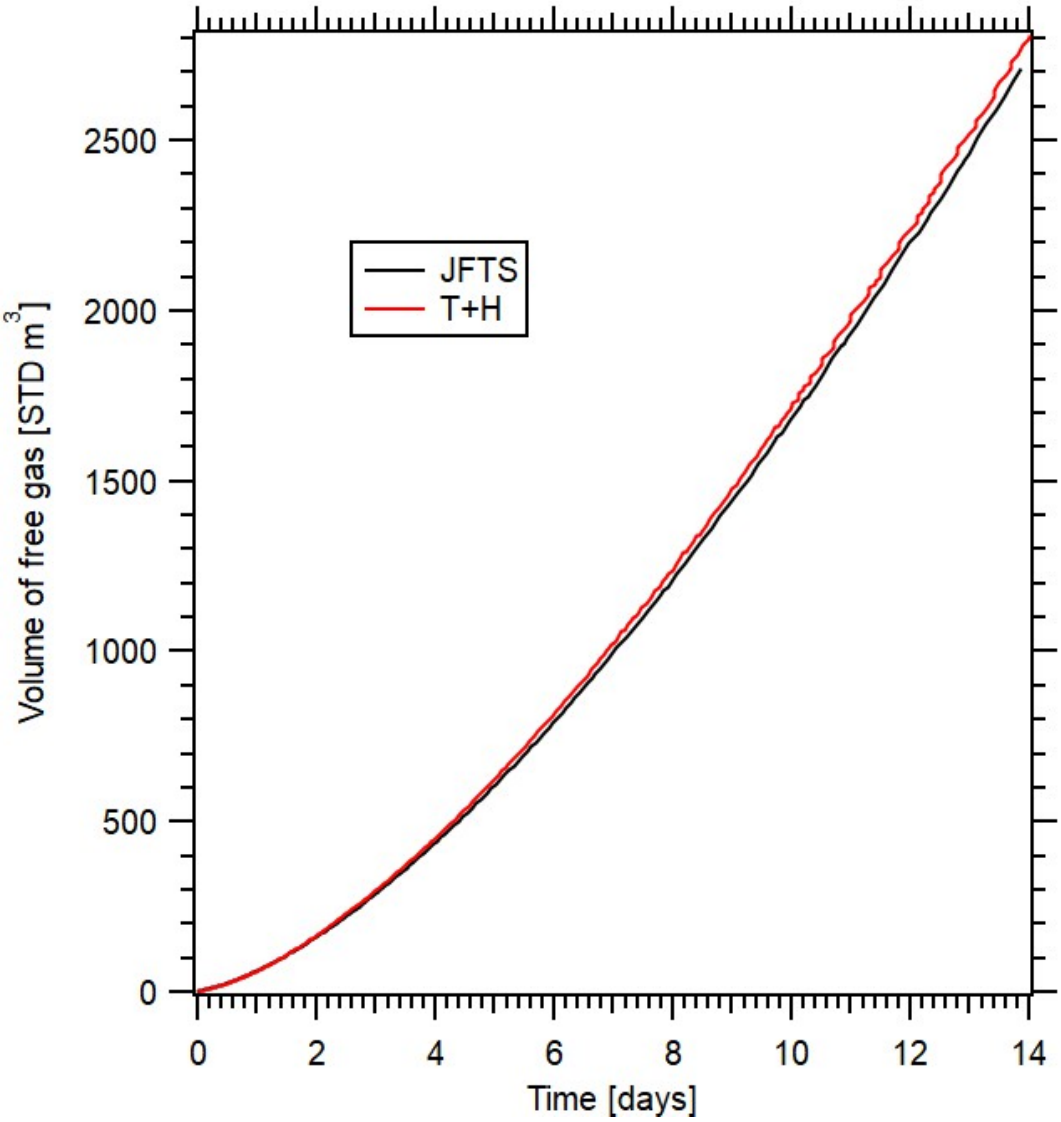

Figure 6.11: Comparison of the JFTS+H and T+H results: volume of the free gas in the reservoir in the 2D problem of **Fig. 6.1**.

The JFTS+H code also predicts the evolution over time of the spatial distributions of key properties, conditions and parameters, providing reliable estimates of the the processes evolving during dissociation during the course of production. **Figs. 6.12** to **6.16** show the spatial distributions of pressure, temperature, and saturations in the vicinity of the well at $t = 7$ days and at the end of the simulation ($t = 14$ days).

JFTS+H tracked the mass rates of water flows across the key interfaces denoted in **Fig. 6.17**.

Fig. 6.18 shows the evolution of water inflows across these key interfaces, as well as Q_A. The results in this figure indicate that (a) significant water inflows occur only through the top and bottom of the hydrate-beating system, (b) the water inflow from the bottom of the hydrate layer 3 exceeds that from the top of hydrate layer 1, (c) the boundaries at the ocean floor and at the base of the domain do not contribute to water inflows, indicating that the pressure disturbance during production does not reach the, (as expected, given the short production period), and (d) the water production at the well significantly exceeds the water inflows from the boundaries, which is attributed to the high permeability and considerable thickness of Interlayer 2.

This field-inspired case describes a modestly large 2D system that involves strongly non-linear processes and over 100K equations. Application of the parallel version of JFTS+H in both super-computer and desktop platforms is shown to achieve significant scalable speed-ups without loss of computational accuracy in the prediction of the results. In the analysis of the parallel computational performance, **Figs. 6.19** to **6.21** show, respectively, the relationships between (a) the total elapsed time, (b) the speed-up, and (c) the efficiency and the number of processors on the Ada cluster. **Figs. 6.22** to **6.24** show the analogous evaluation of the parallel simulations on the MacPro platform. The results in **Fig. 6.19** indicate that the total elapsed time decreases significantly and monotonically as the number of processors increases and this trend persists for up to 512 processors, which is believed to be the optimum number of processors: a lower speed-up is observed when 1024 processors are involved. Given this result, the continuous and consistent improvement in the parallel performance of the JFTS+H solution observed on the MacPro with an increasing number of processors was expected, as their maximum number of processors (12) is over 40 times lower than the number associated with optimal performance on the Ada cluster. These results indicate the effectiveness and reliability of the parallel performance of JFTS+H, which exhibits significant speedups and scalability. Efficiency on both Ada cluster and MacPro up to 8 processors show more than or nearly equal to 0.9, which are cost-effective. When the computational environment is limited, 8-processor parallel computing can achieve practical speed-up.

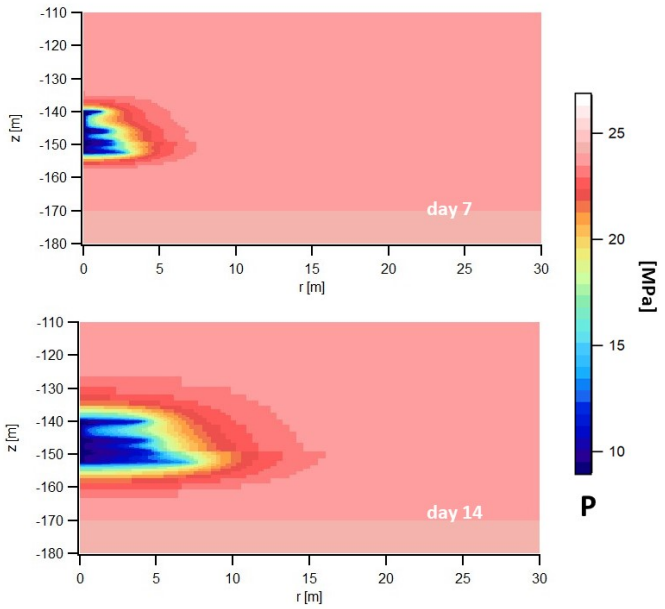

Figure 6.12: JFTS+H prediction of evolution of the spatial distribution of pressure in the well vicinity in the 2D problem of **Fig. 6.1**.

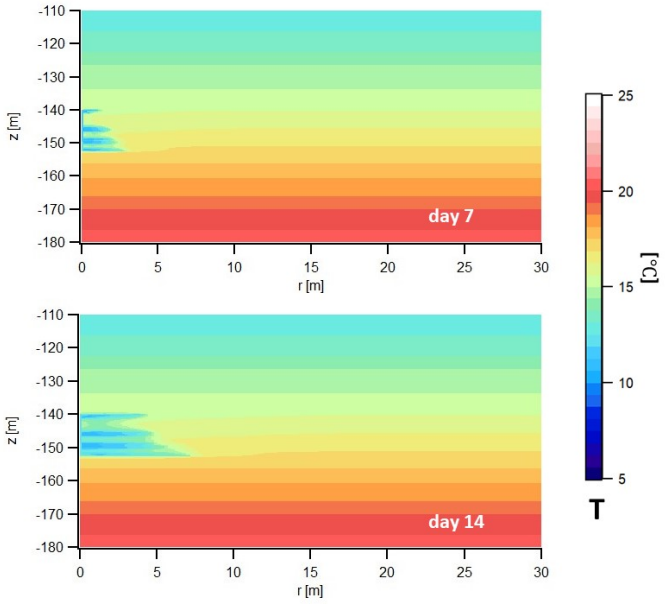

Figure 6.13: JFTS+H prediction of evolution of the spatial distribution of temperature in the well vicinity in the 2D problem of **Fig. 6.1**.

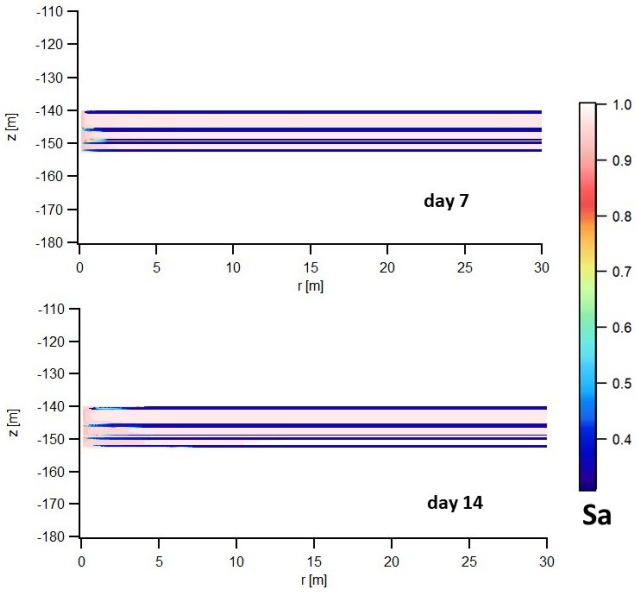

Figure 6.14: JFTS+H prediction of evolution of the spatial distribution of aqueous saturation in the well vicinity in the 2D problem of **Fig. 6.1**.

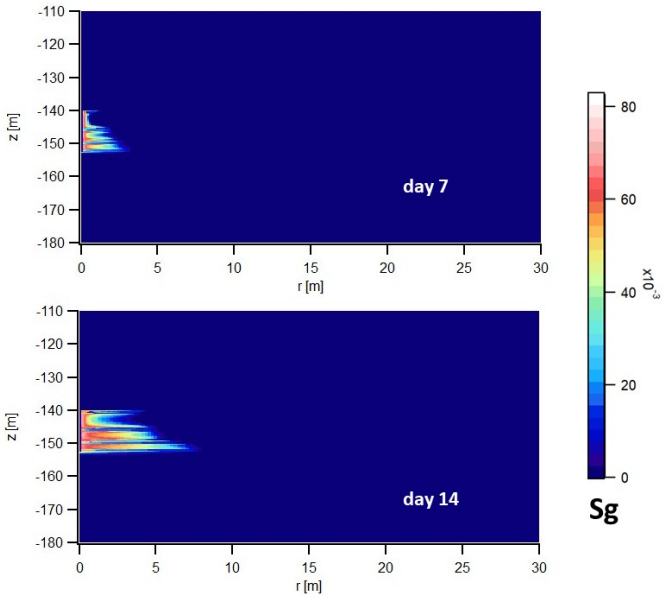

Figure 6.15: JFTS+H prediction of evolution of the spatial distribution of gas saturation in the well vicinity in the 2D problem of **Fig. 6.1**.

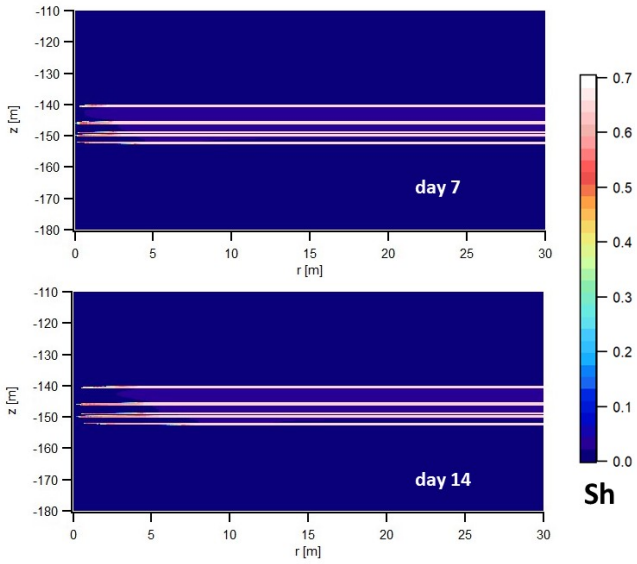

Figure 6.16: JFTS+H prediction of evolution of the spatial distribution of hydrate saturation in the well vicinity in the 2D problem of **Fig. 6.1**.

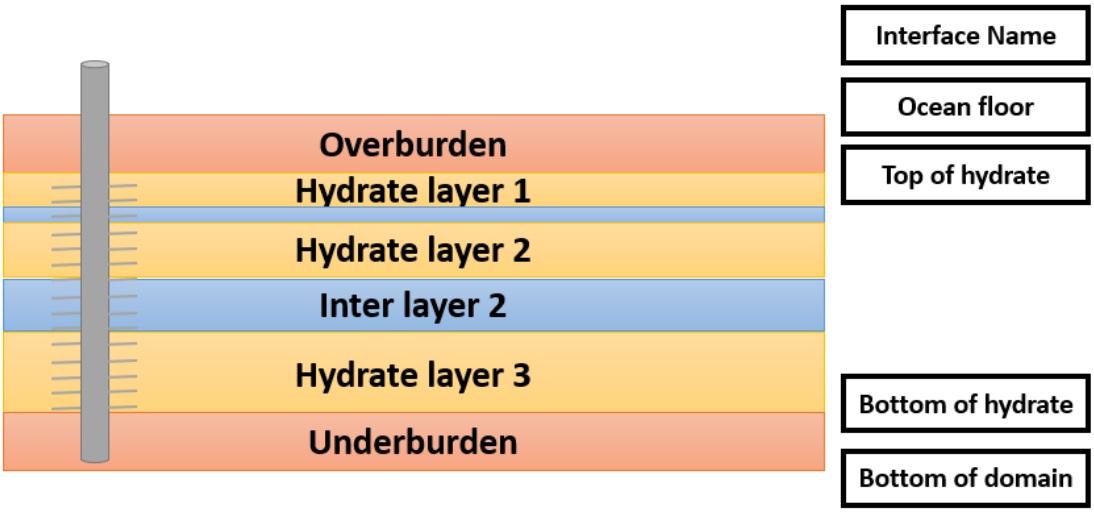

Figure 6.17: Locations of key interfaces in the 2D problem of **Fig. 6.1**.

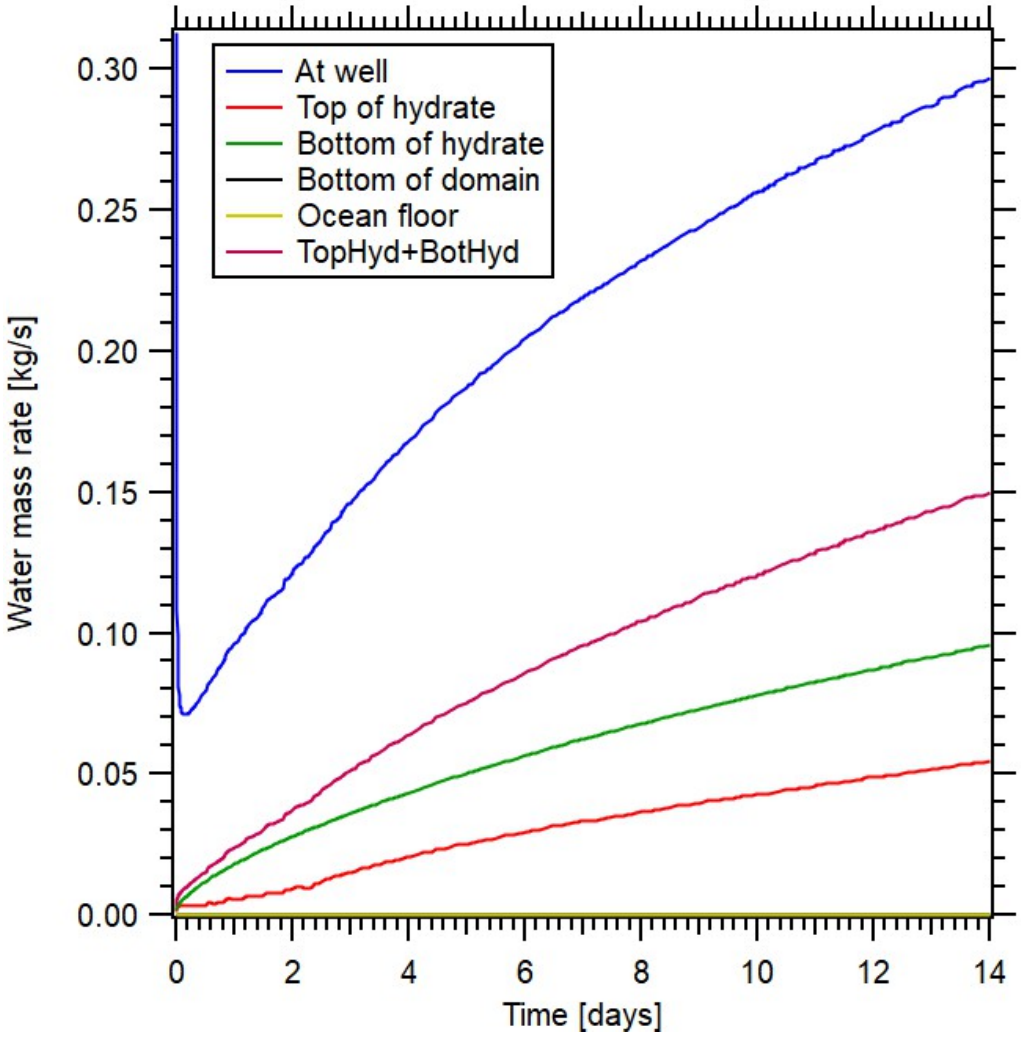

Figure 6.18: JFTS+H prediction of evolution of water inflows across key interfaces and water production at the well in the 2D problem of **Fig. 6.1**.

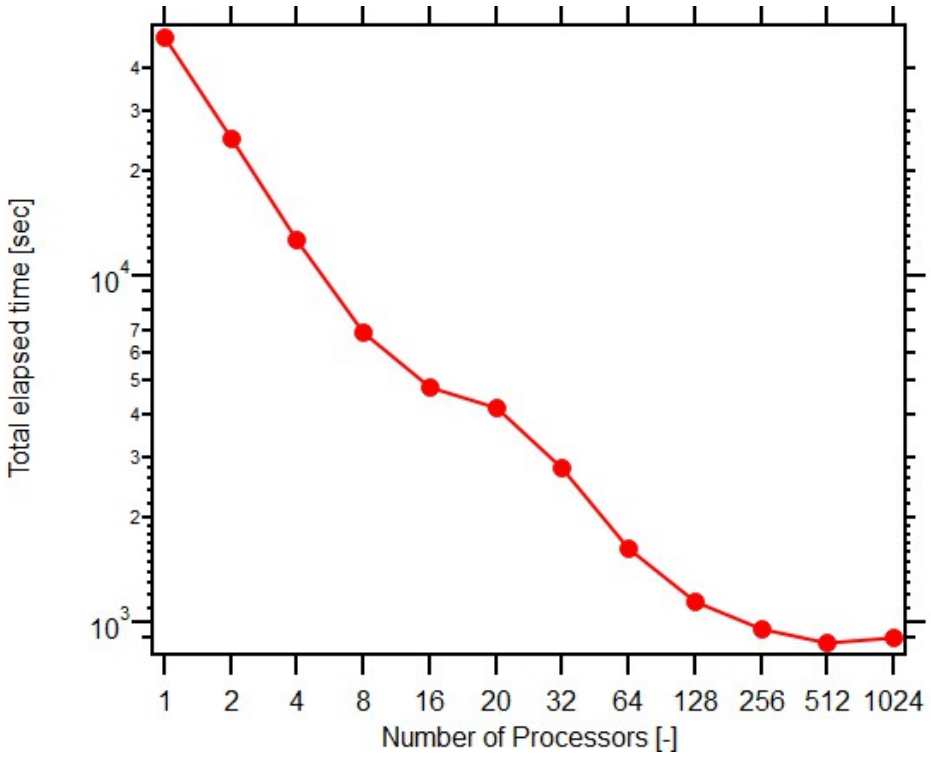

Figure 6.19: JFTS+H parallel performance: the relationship between total elapsed time and number of processors in the study of 2D the problem of **Fig. 6.1** on the Ada cluster.

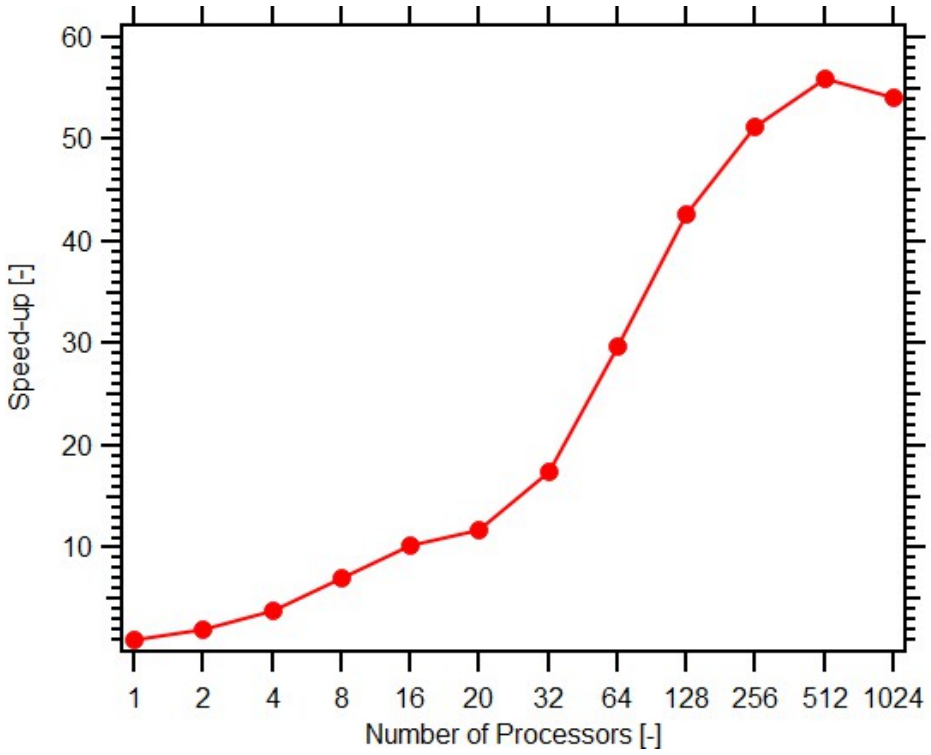

Figure 6.20: JFTS+H parallel performance: the relationship between speed up and number of processors in the study of the 2D problem of **Fig. 6.1** on the Ada cluster.

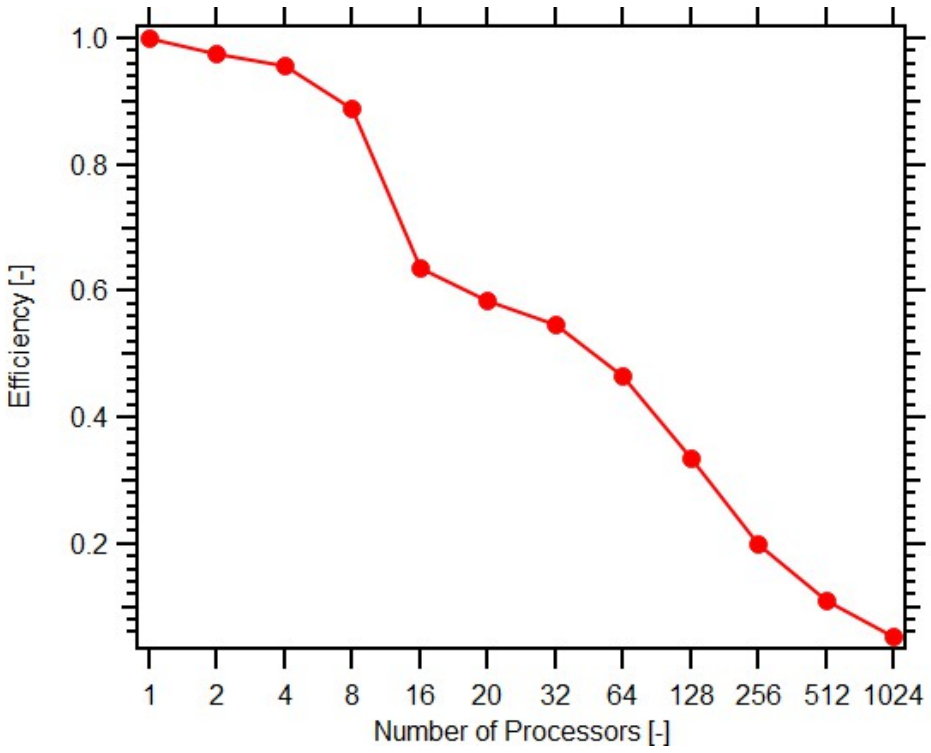

Figure 6.21: JFTS+H parallel performance: the relationship between efficiency and number of processors in the study of the 2D problem of **Fig. 6.1** on the Ada cluster.

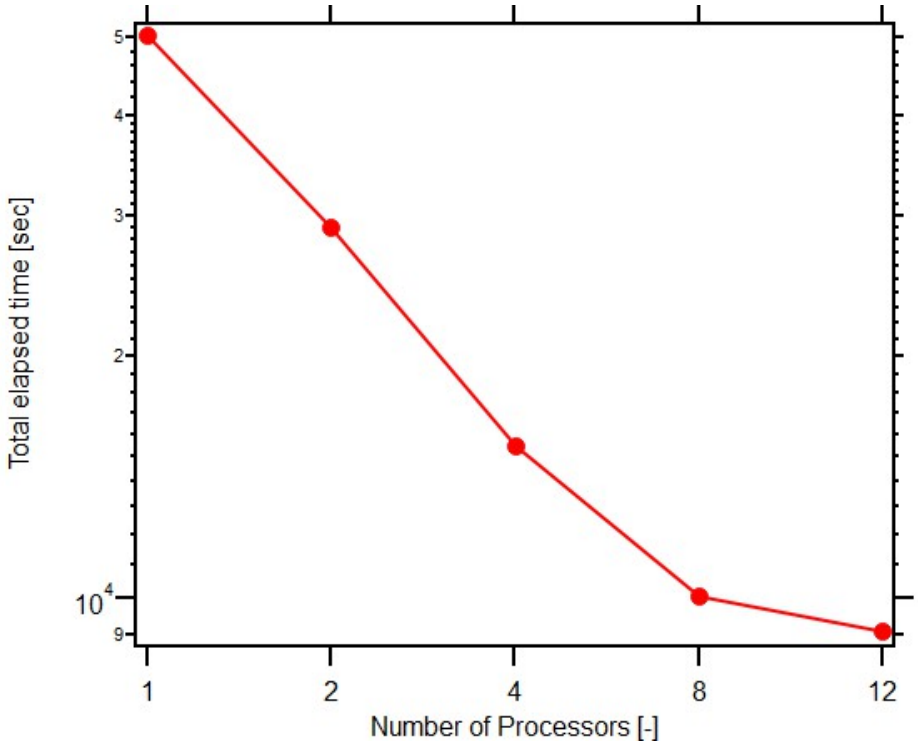

Figure 6.22: JFTS+H parallel performance: the relationship between total elapsed time and number of processors in the study of the 2D problem of **Fig. 6.1** on the MacPro.

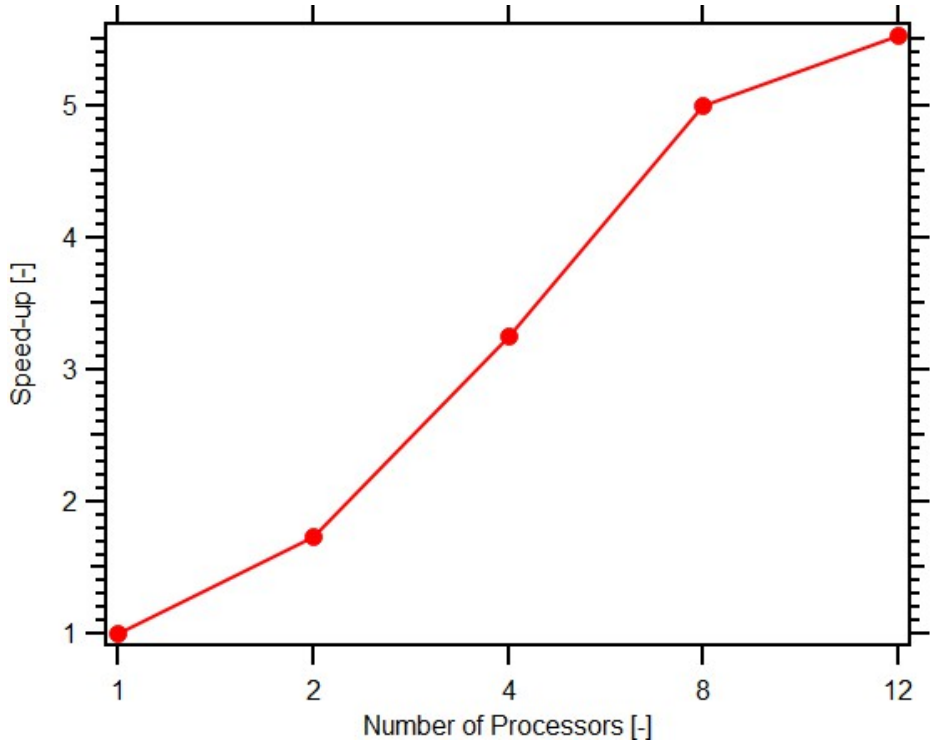

Figure 6.23: JFTS+H parallel performance: the relationship between speed up and number of processors in the study of the 2D problem of **Fig. 6.1** on the MacPro.

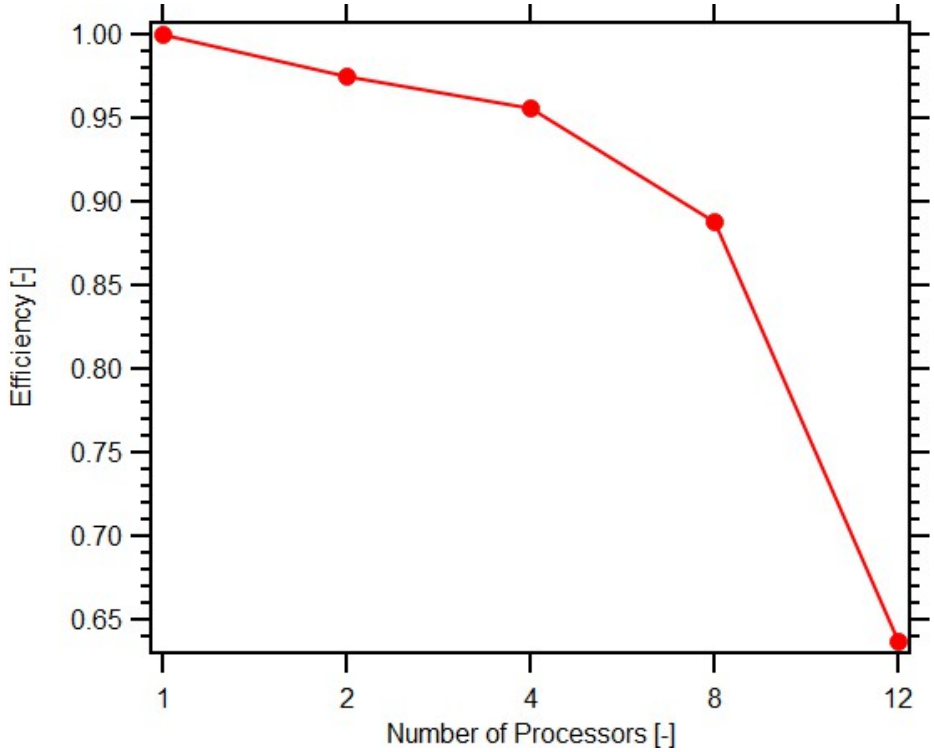

Figure 6.24: JFTS+H parallel performance: the relationship between efficiency and number of processors in the study of the 2D problem of **Fig. 6.1** on the MacPro.

6.2 Field Case Application 2: A Realistic Arctic, Permafrost-Associated Deposit

6.2.1 Description of the Geologic System

The reservoir model is composed of three hydrate-bearing layers and three hydrate-free interlayers. The reservoir regions are overlain and underlain by low- (but not zero-) permeability media that form the overburden and underburden. A simple sketch of the system geometry and configuration is shown in **Fig. 6.25**. Gas production is based on depressurization-induced dissociation of the CH_4-hydrates in Unit B. In an effort to minimize the stresses in the vicinity of the well and possible well stability issues, the bottomhole pressure is time-variant, decreasing in stages. The simulation covered a production period of 120 days under the following variable bottomhole pressure P_w regime: $t = 0$ to $t = 30$ days, $P_w = 6.58$ Pa; from $t = 30$ days to $t = 60$ days, $P_w = 4.58$ Pa; and $P_w = 2.80$ Pa for $t > 60$ days. I investigate three scenarios: the reference case (Case Ref), which involved a production well completed in the top 10 m of Unit B; a case of lower effective permeability system associated with the maximum possible value of the exponent of the relative permeability model (Case nmax); and a case involving a 10m-long well, the top of which was at a depth of 3 m from the top of Unit B (Case LoW). The variation of the exponents n with the depth of unit B in Cases Ref and nmax is shown in **Fig. 6.26**; the variation of the initial S_H with the depth of unit B in all cases is shown in **Fig. 6.27**.

The initial pressure and temperature distributions in the domain are computed from JFTS+H simulations that continue until thermal, hydraulic, and chemical equilibrium, i.e., steady-state conditions, are achieved. The initial pressure and temperature distribution with the vertical elevation z are shown in **Figs. 6.28** and **6.29**, respectively. The irreducible water saturation, porosity, and logarithmic natural of permeability are shown in **Figs 6.30**, **6.31**, and **6.32**, respectively. These data, along with those from **Figs. 6.26** and **Figs. 6.27**, indicate the significant vertical heterogeneity in the properties of this system.

6.2.2 Domain Discretization

The system is discretized into a 2D cylindrical grid that includes 627x343 = 215,061 gridlocks in (r, z), resulting in a Jacobian matrix equation comprising a total of 627x343x4= 860,244 simultaneous linear equations of mass and heat balance. As in Field Case 1, the JFTS+H parallel simulator decomposes the entire domain using the METIS application (Karypis, 2013), which divides it into a number of subdomains equal to the number of processors. **Fig. 6.33** shows an example of the METIS domain decomposition that shows the subdomains into which METIS partitions the domain when 64 processors are used for the simulation.

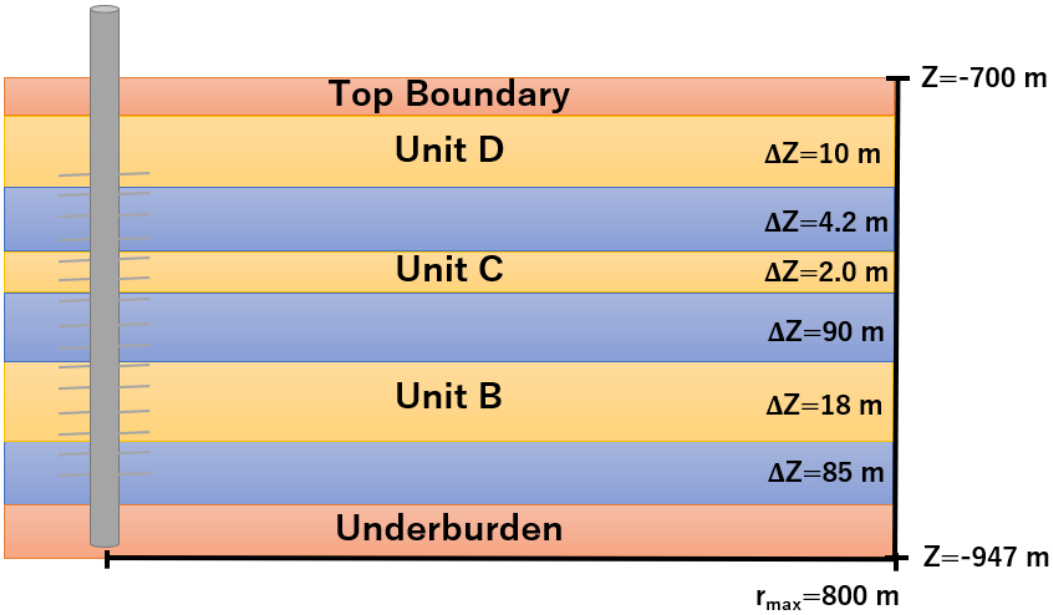

Figure 6.25: The layered geological model of the reservoir used in the 2D arctic field case.

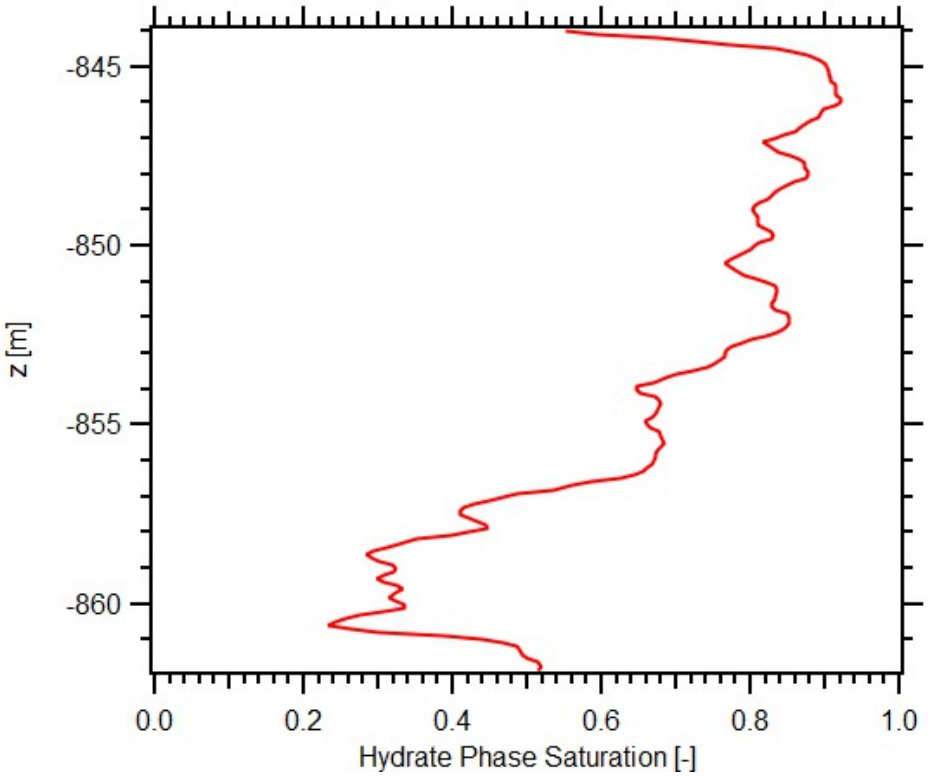

Figure 6.26: Exponents n distribution along z-direction in the 2D problem of **Fig. 6.25**.

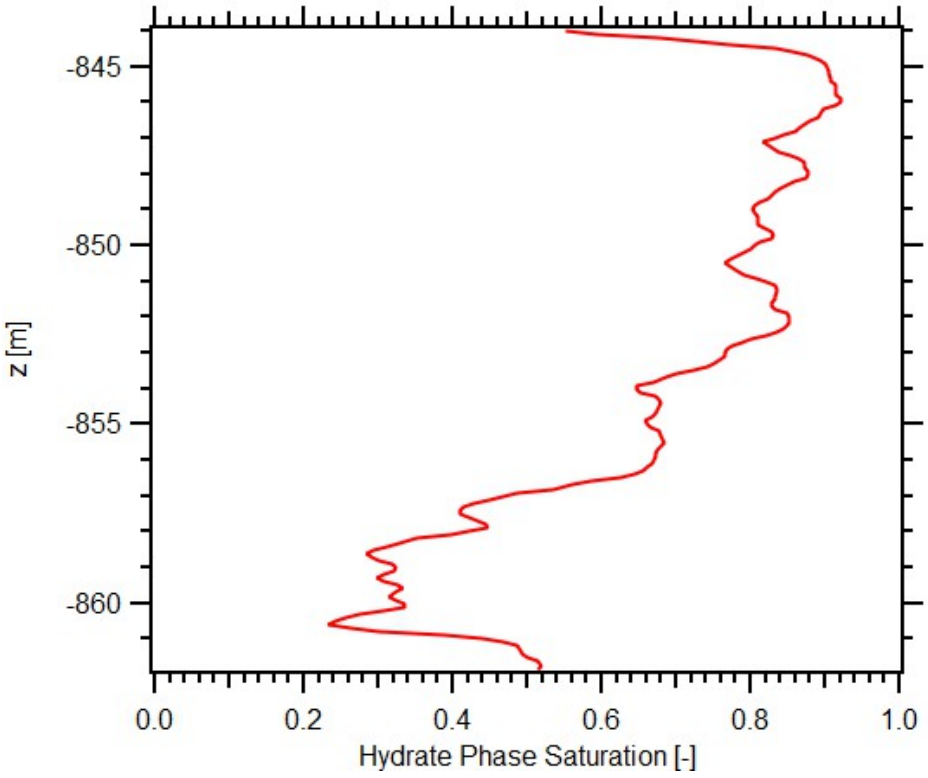

Figure 6.27: Initial hydrate saturation distribution along z-direction in the 2D problem of **Fig. 6.25**.

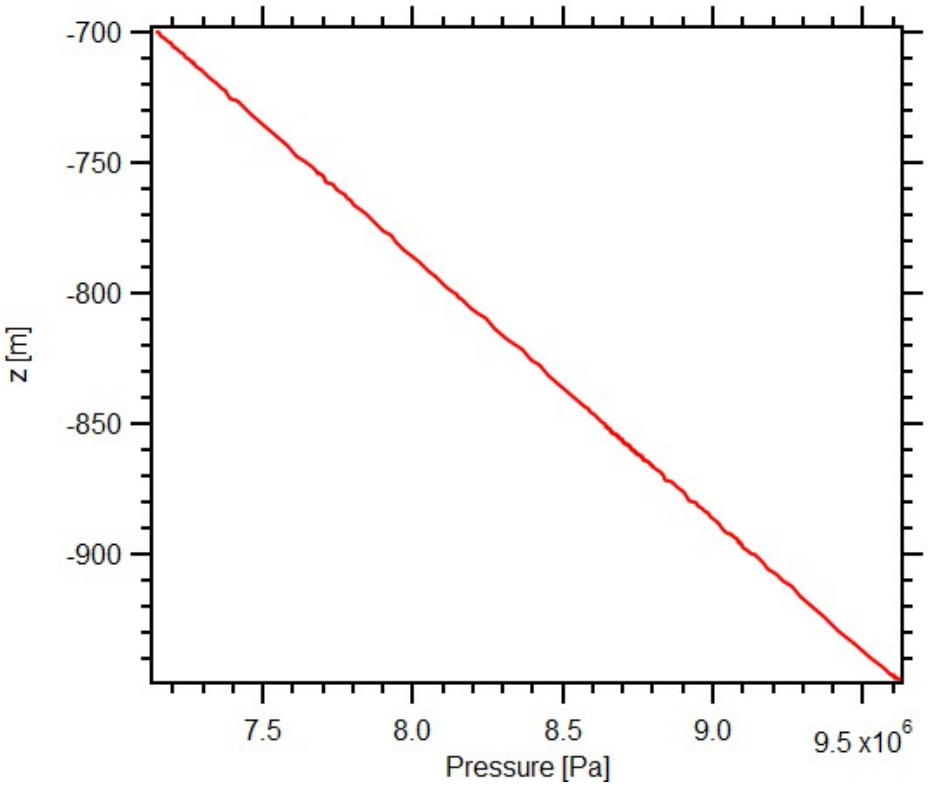

Figure 6.28: Initial pressure distribution along z-direction in the 2D problem of **Fig. 6.25**.

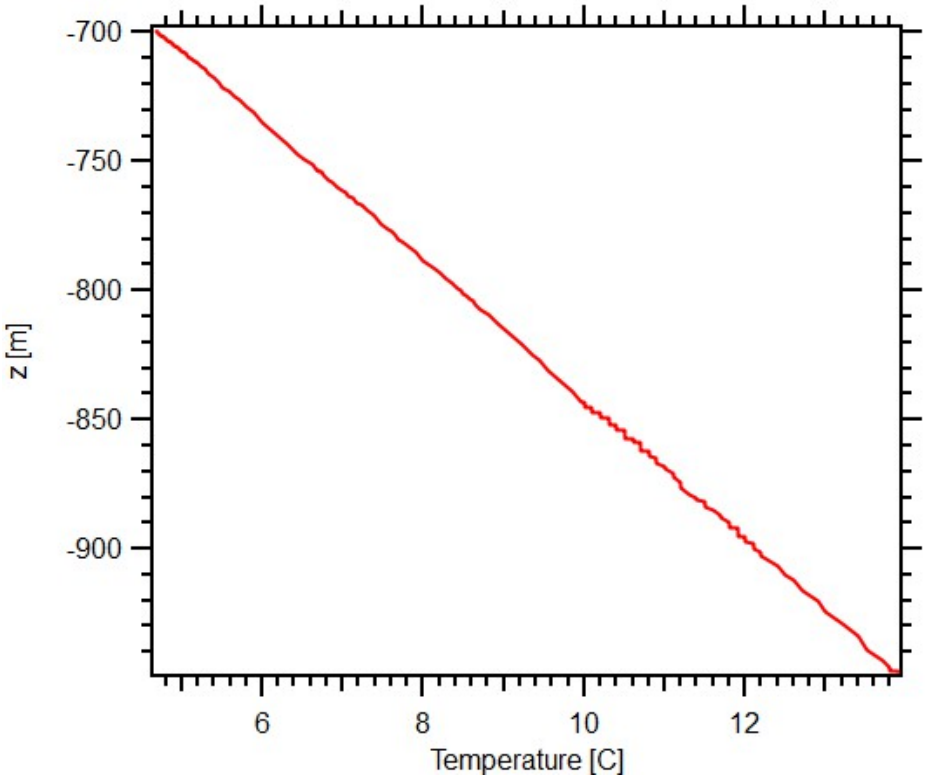

Figure 6.29: Initial temperature distribution along z-direction in the 2D problem of **Fig. 6.25**.

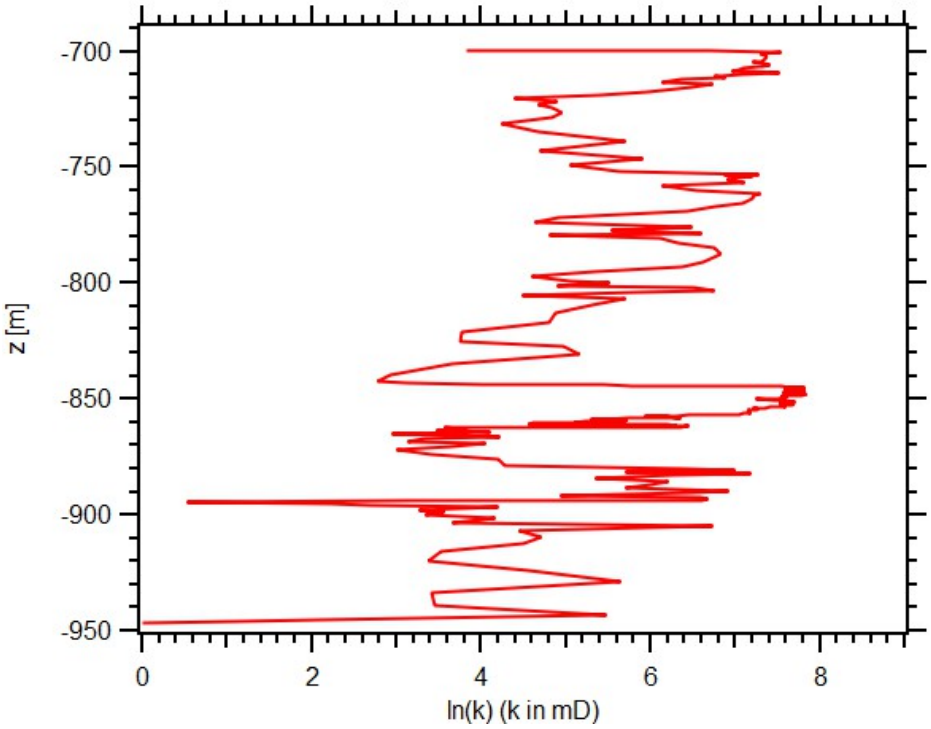

Figure 6.30: Natural logarithm of permeability distribution along z-direction in the 2D problem of **Fig. 6.25**.

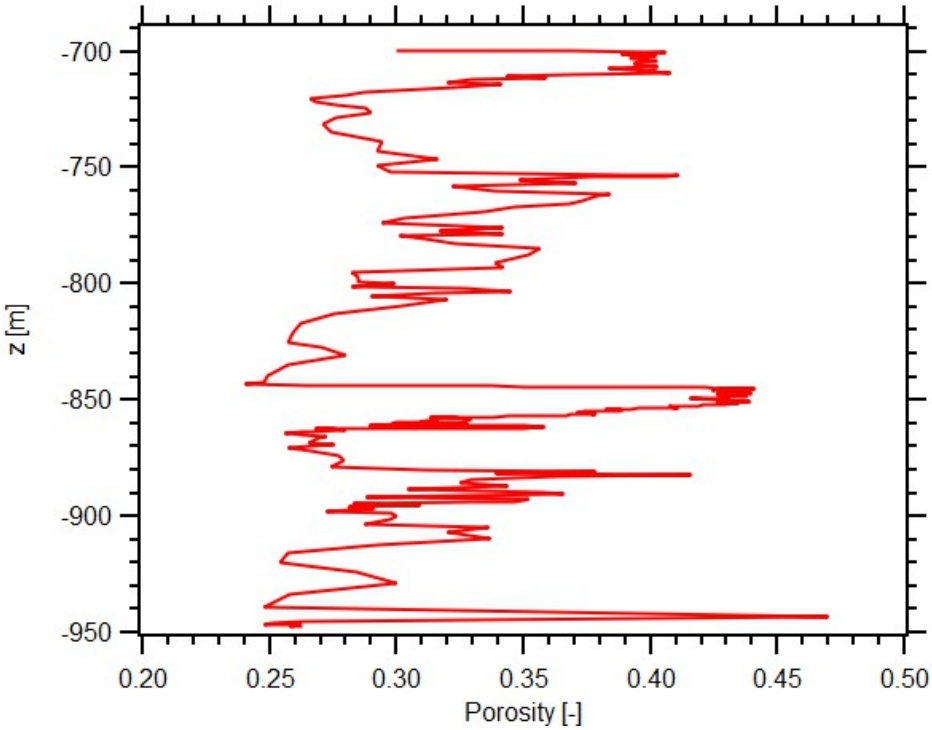

Figure 6.31: Porosity distribution along z-direction in the 2D problem of **Fig. 6.25**.

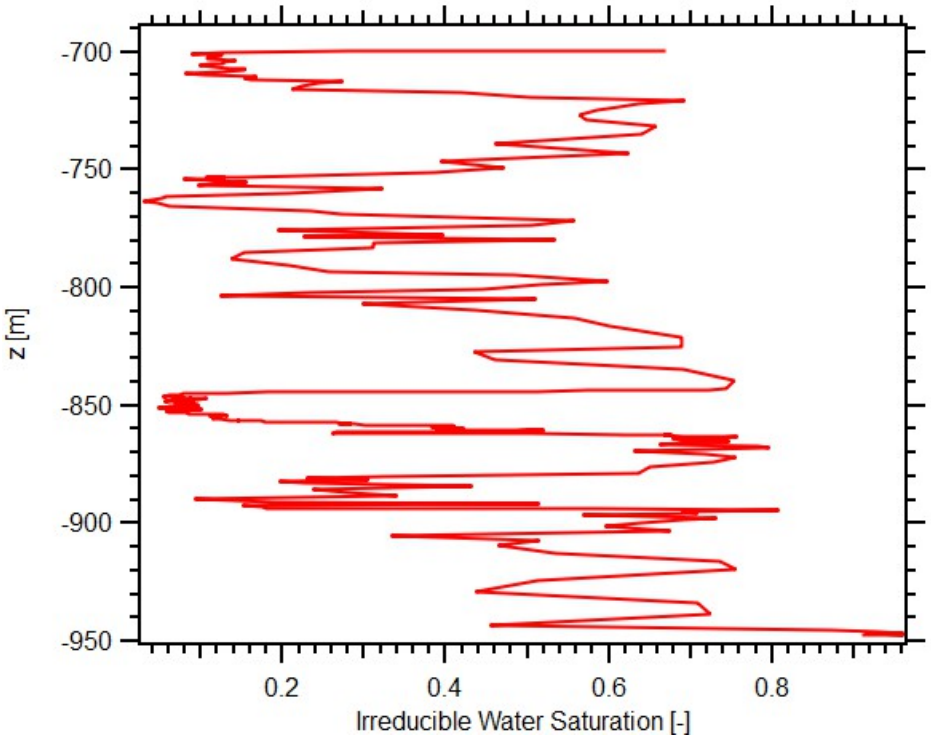

Figure 6.32: Irreducible water saturation distribution along z-direction in the 2D problem of **Fig. 6.25**.

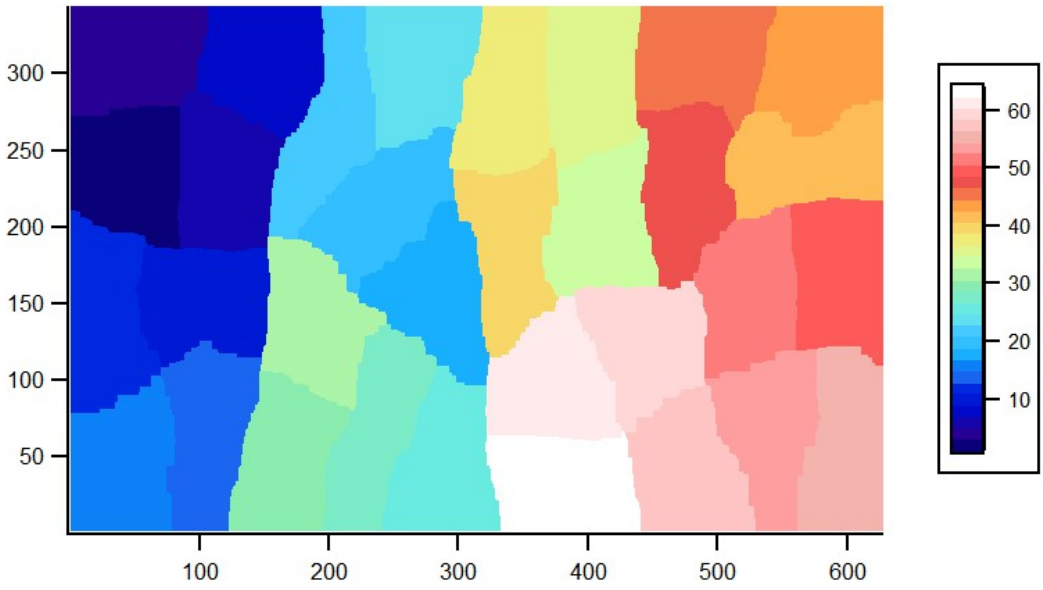

Figure 6.33: The domain decomposition of the 2D problem that results from the application of METIS (Karypis, 2013) in a JFTS+H simulations that involves 64 processors.

6.2.3 Simulation Results

The size of the problem practically precluded the application of the serial T+H simulator for the solution of this problem, which was solved exclusively on the Ada cluster and on the MacPro desktop computer. Q_R, M_R, and Q_P in all three cases are shown in **Figs. 6.34**, **6.35**, and **6.36** respectively. Consistent with expectations, Q_R exhibit a significant jump in magnitude at the times of the $\mathrm{P_w}$ reduction at the well, i.e., at $t = 0$, 30 and 60 days after the inception of production. M_R in Scenario 2 (Case nmax) and Scenario 3 (Case LoW) is 42.7 % lower and 69.0 % higher than that in the reference case. The obvious conclusions drawn from these observations are that (a) the location of the well and (b) the effective permeability strongly influence the hydrate dissociation process and the related gas production. Q_A, R_{wg}, and V_F for all three scenarios are shown in **Figs. 6.37**, **6.38** and **6.39**, respectively, and they show the same pattern of relative magnitudes in the three cases: Case LoW is associated with the highest quantities, rates and volumes, and Case nmax with the lowest.

The JFTS+H code also predicts the evolution over time of the spatial distributions of pressure, temperature, and phase saturations distributions in all three scenarios — accurately describing the evolving dissociation processes during the course of production. **Figs. 6.40** to **6.54** show the pressure, temperature, and saturations behavior in the vicinity of the well for the reference Case Ref at day 30, day 60, day 90, and at the end of the simulation (t = 120 days).

The locations of the key boundaries (where water inflows are monitored) are shown in **Fig. 6.55**. **Figs. 6.56** to **6.58** show the evolution of water inflows across the top and bottom boundaries of Unit B, as well as Q_A. In Cases Ref, nmax and LoW respectively. Note that water inflows represent a large fraction of the water production at the well in all cases, a situation that is not desirable because of the consequent difficulty to rapidly depressurize the system and, thus, maximize hydrate dissociation and gas production. **Figs. 6.59** to **6.61** show the water inflows across other key interfaces in the three cases, and these are consistently much lower than those across the top and bottom boundaries of Unit B.

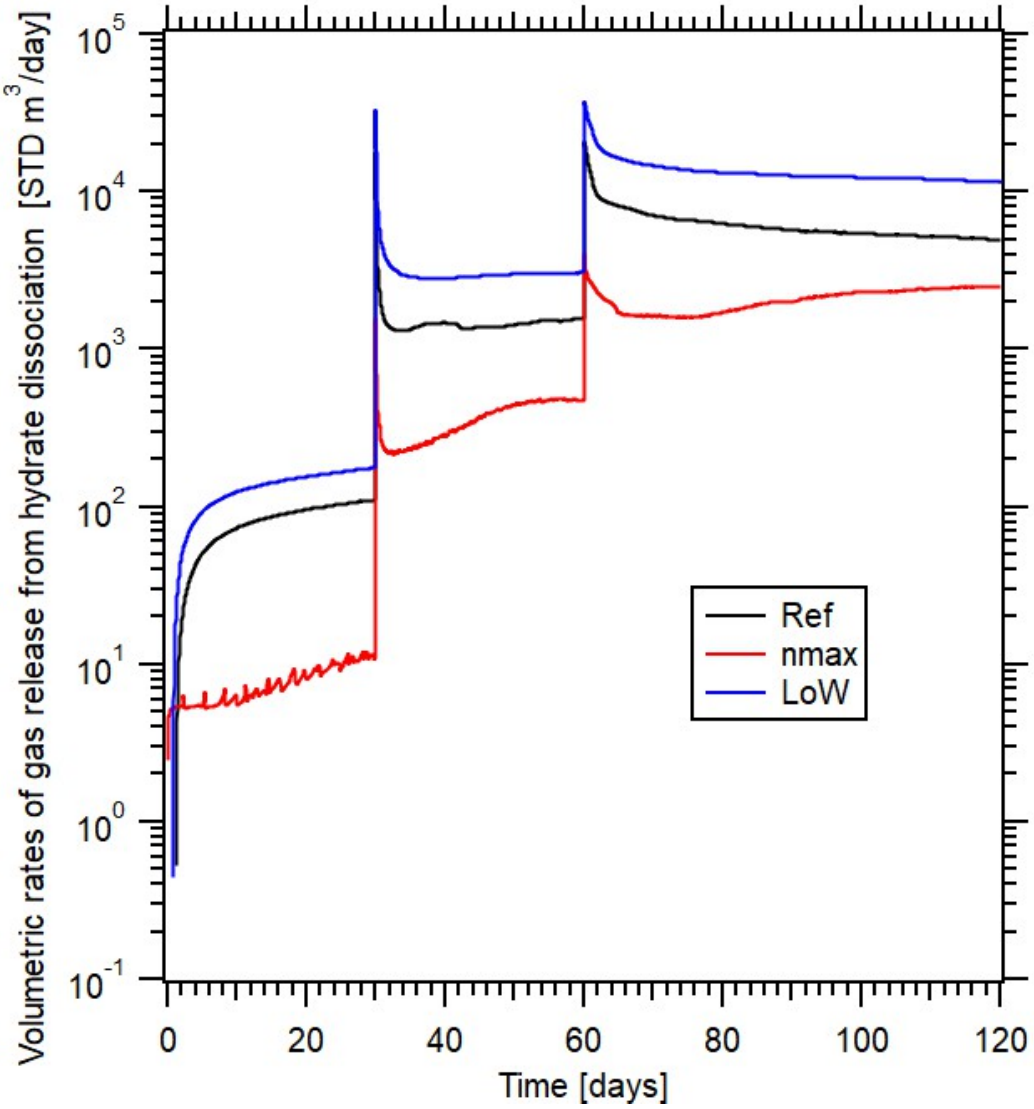

Figure 6.34: JFTS+H simulation results: volumetric rates of the CH_4 gas release from hydrate dissociation in the problem of **Fig. 6.25**. The Ada and the MacPro results coincide.

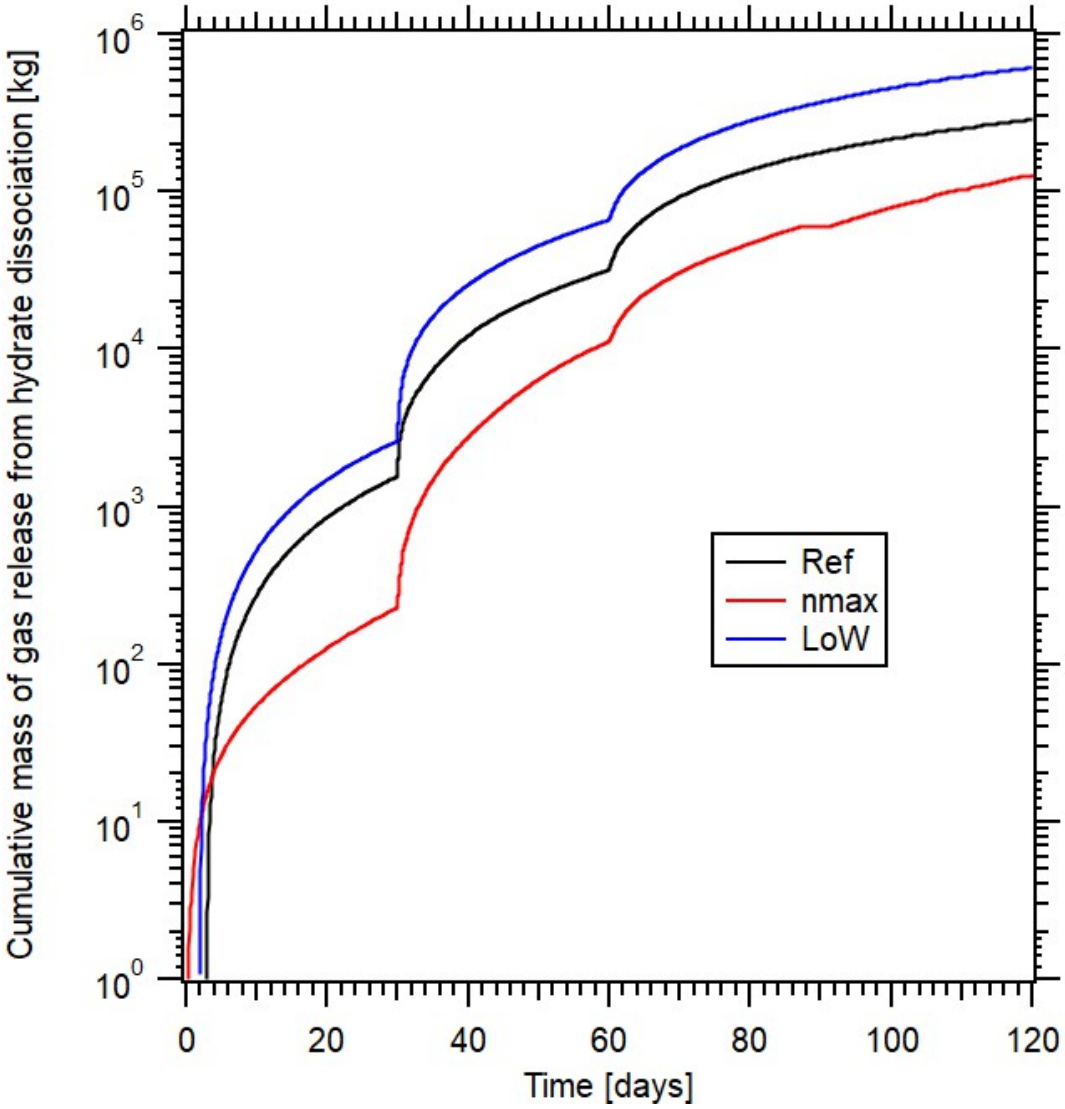

Figure 6.35: JFTS+H simulation results: cumulative mass of the CH_4 gas released from hydrate dissociation in the 2D problem of **Fig. 6.25**. The Ada and the MacPro results coincide.

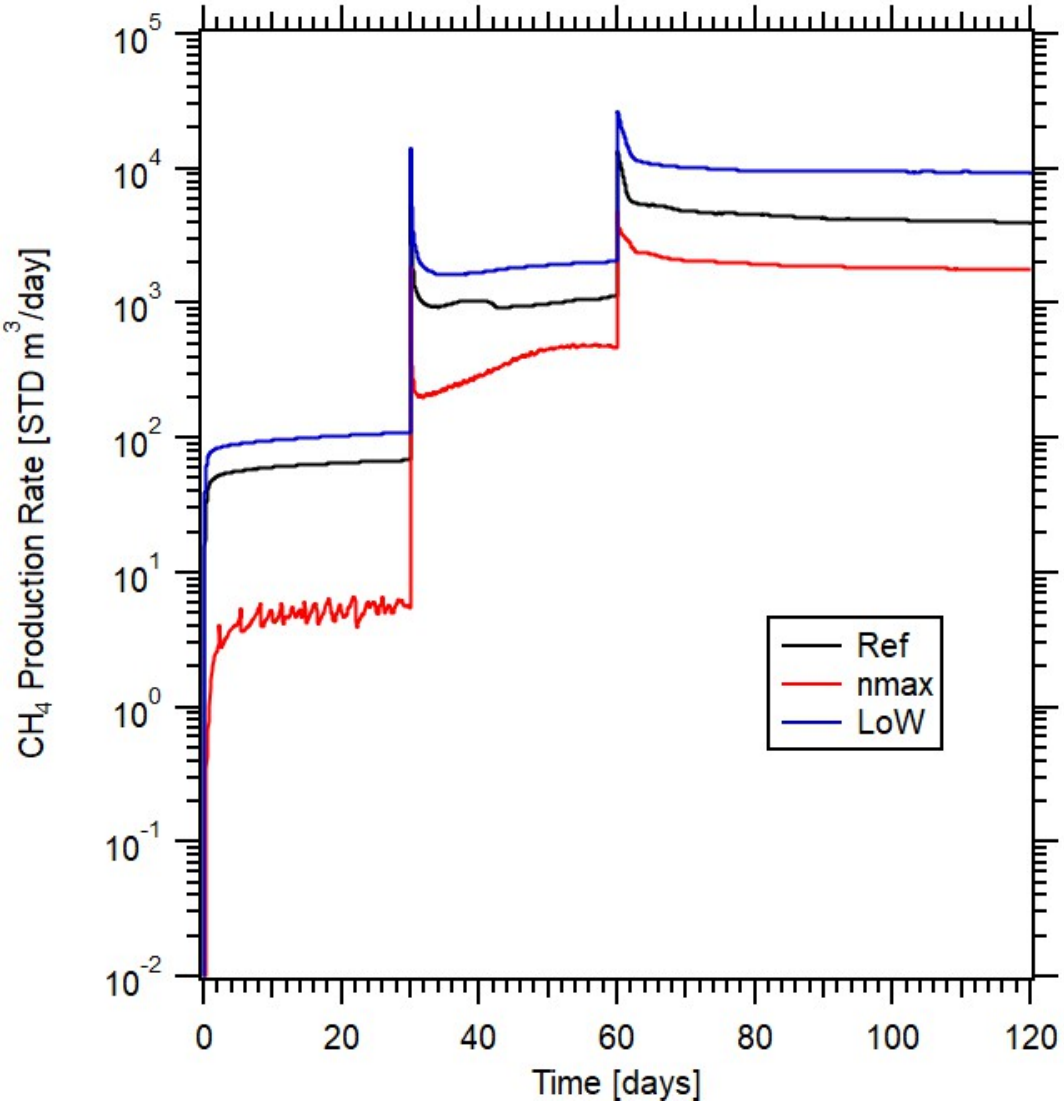

Figure 6.36: JFTS+H simulation results: mass rates of the CH_4 production in the 2D problem of **Fig. 6.25**.

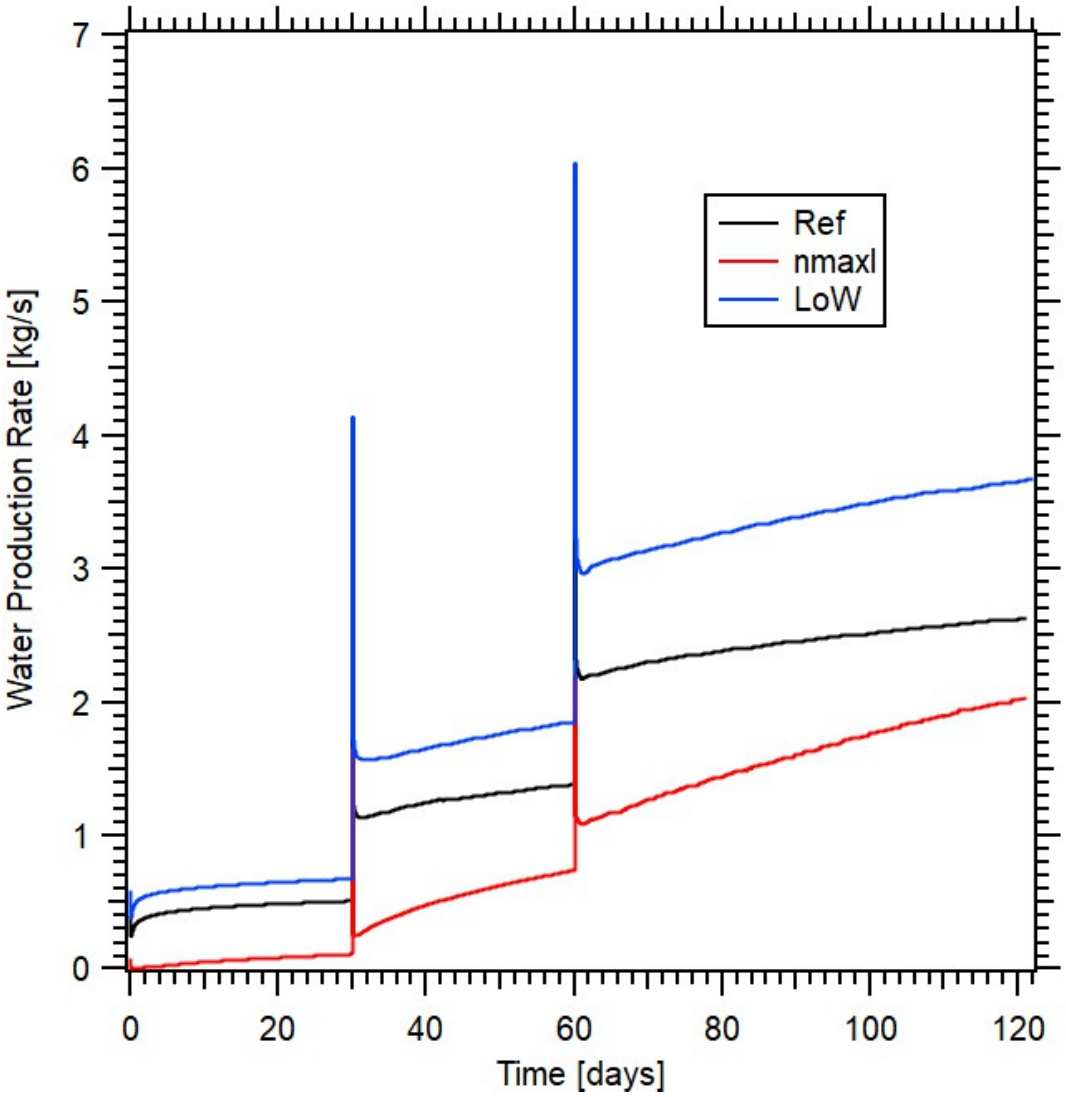

Figure 6.37: JFTS+H simulation results: mass rates of the water production in the 2D problem of **Fig. 6.25**.

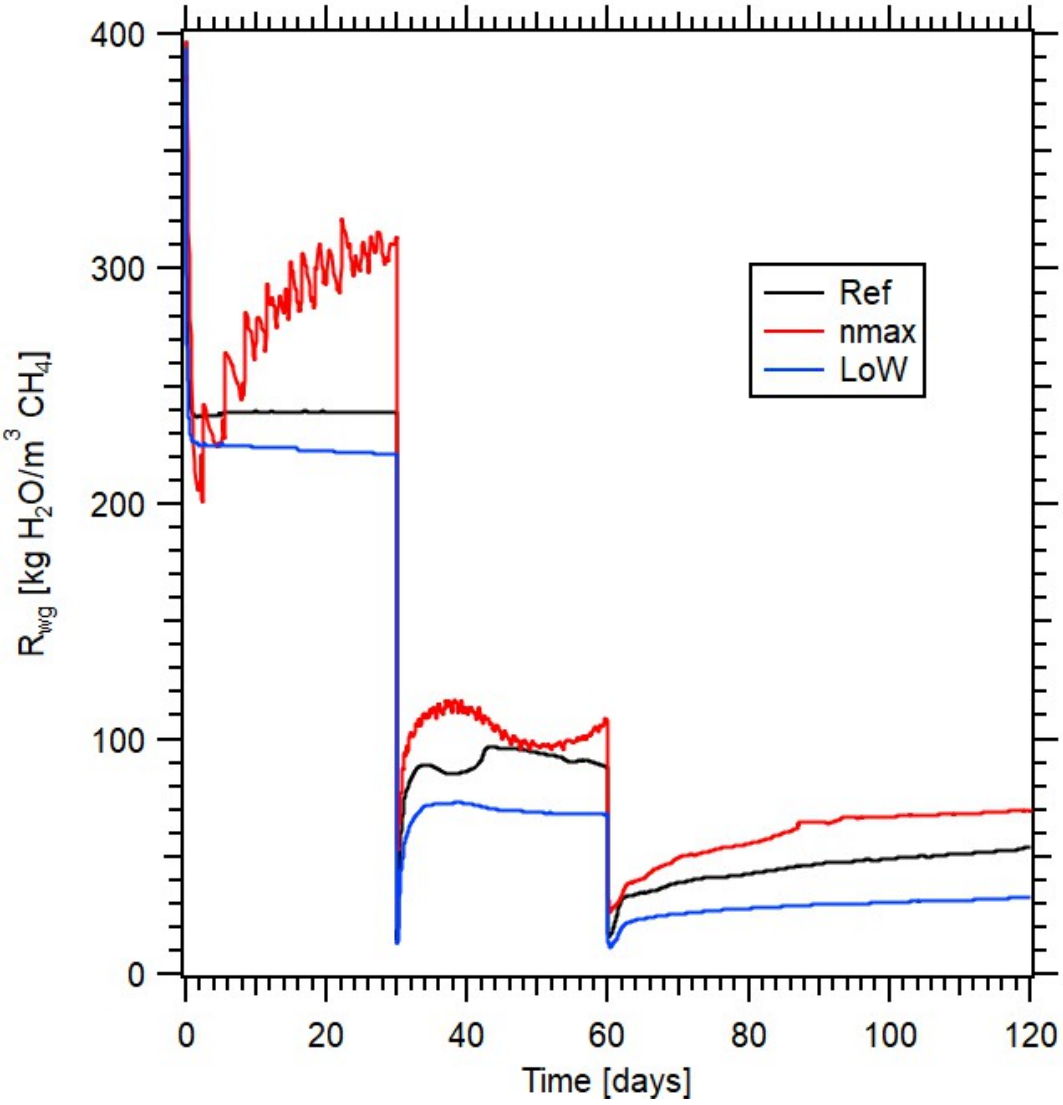

Figure 6.38: JFTS+H simulation results: the water/gas ratio in the 2D problem of **Fig. 6.25**.

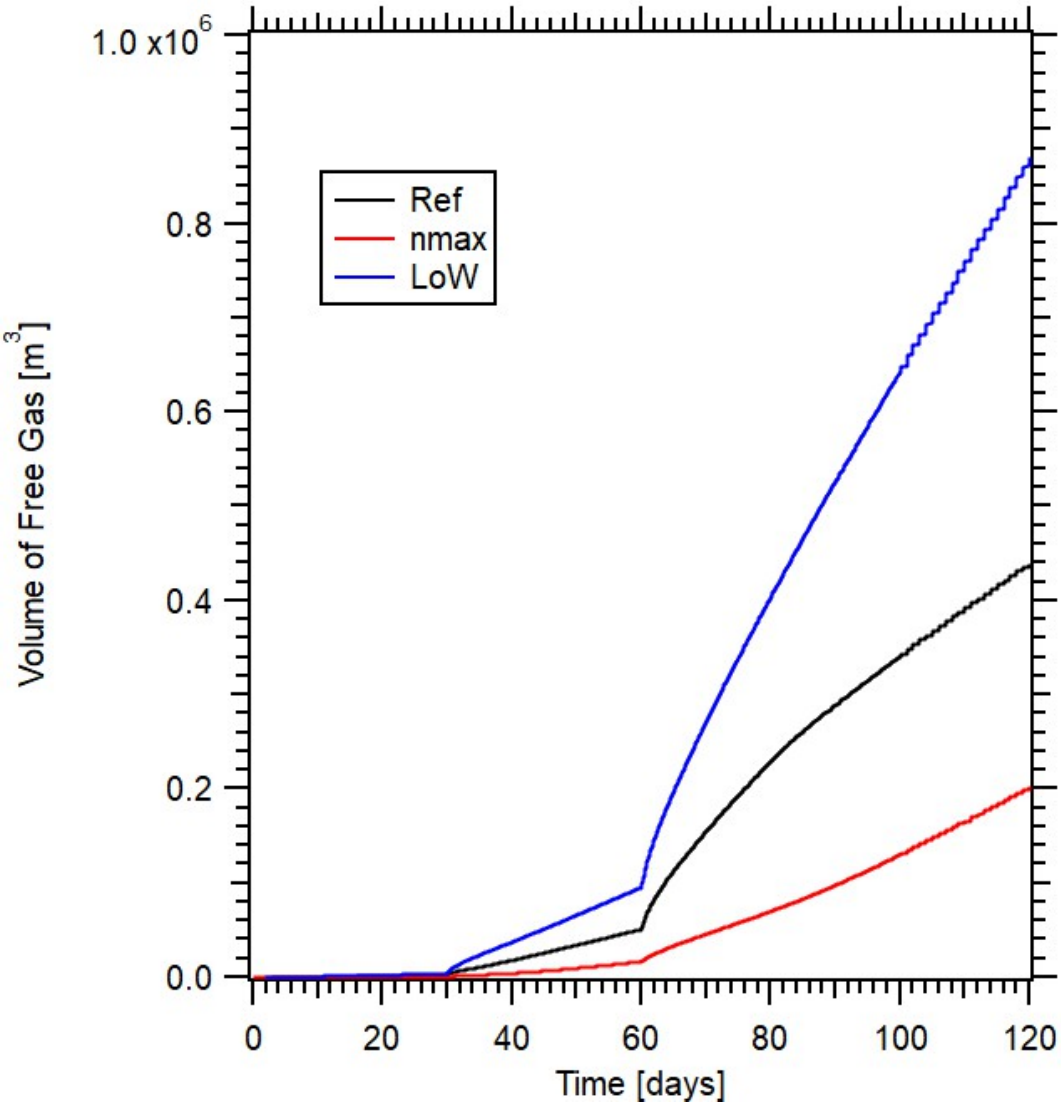

Figure 6.39: JFTS+H simulation results: volume of the free gas in the 2D problem of **Fig. 6.25**.

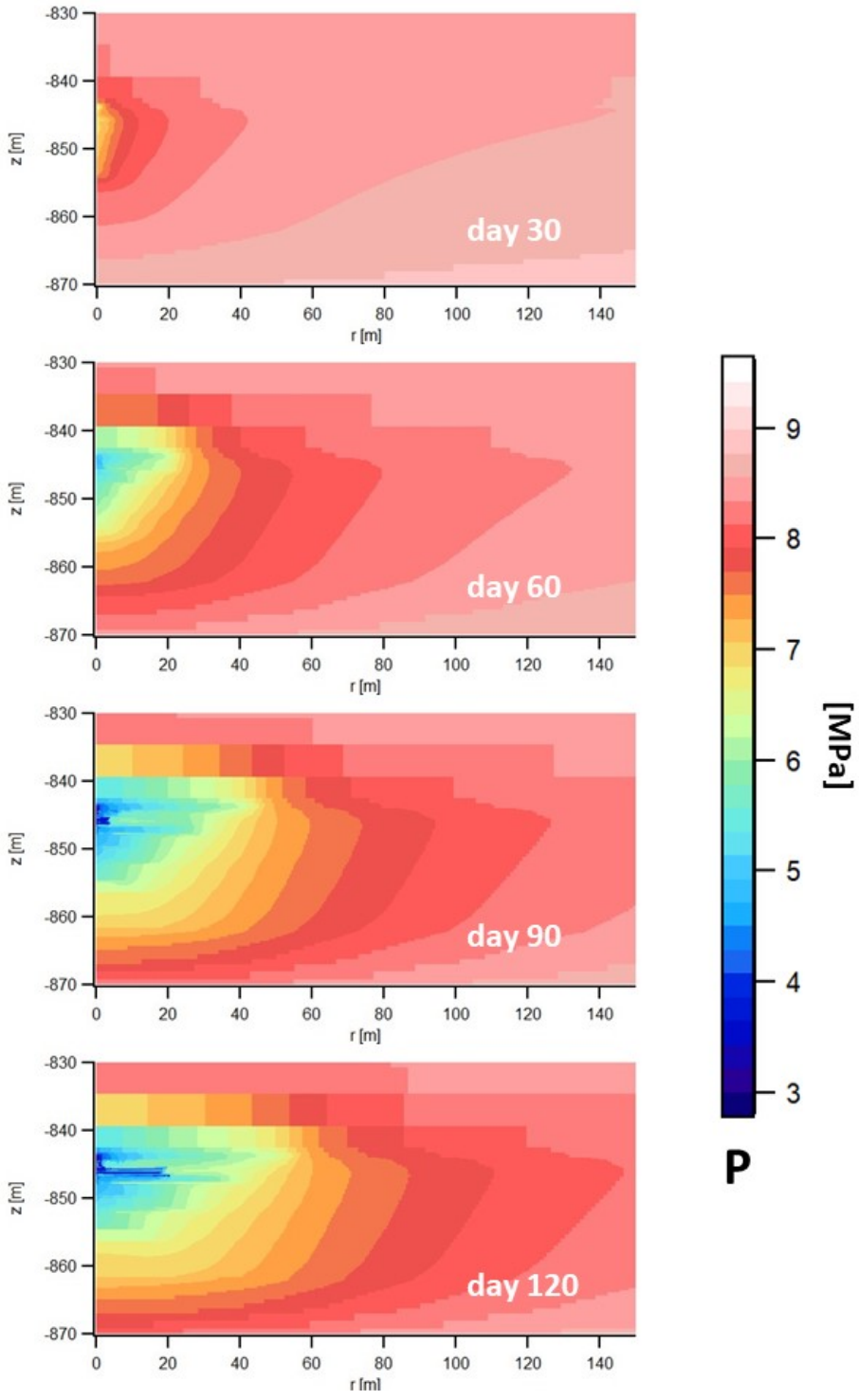

Figure 6.40: Case Ref: JFTS+H prediction of evolution of the spatial distribution of pressure in the well vicinity in the 2D problem of **Fig. 6.25** (Ref).

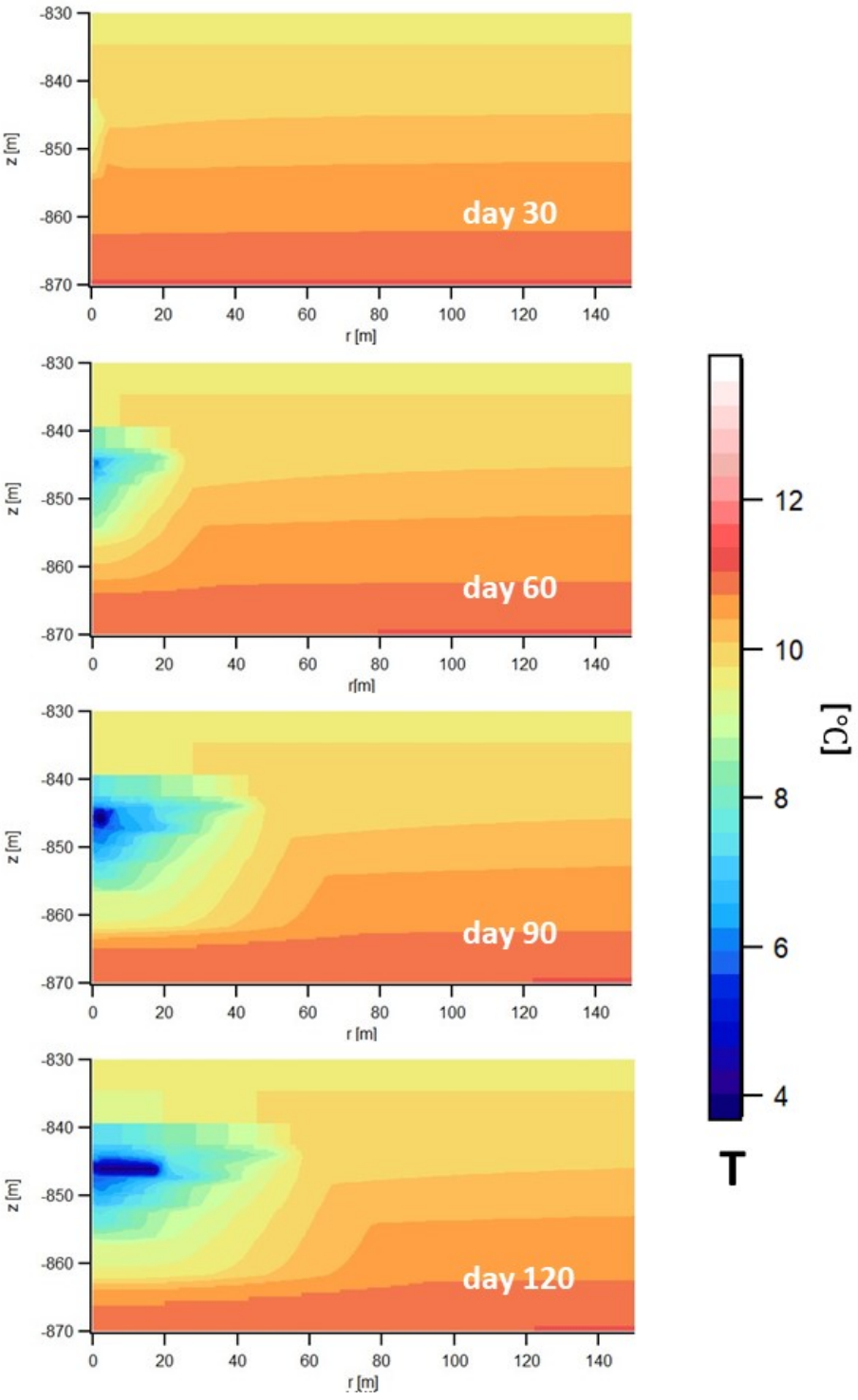

Figure 6.41: Case Ref: JFTS+H prediction of evolution of the spatial distribution of temperature in the well vicinity in the 2D problem of **Fig. 6.25** (Ref).

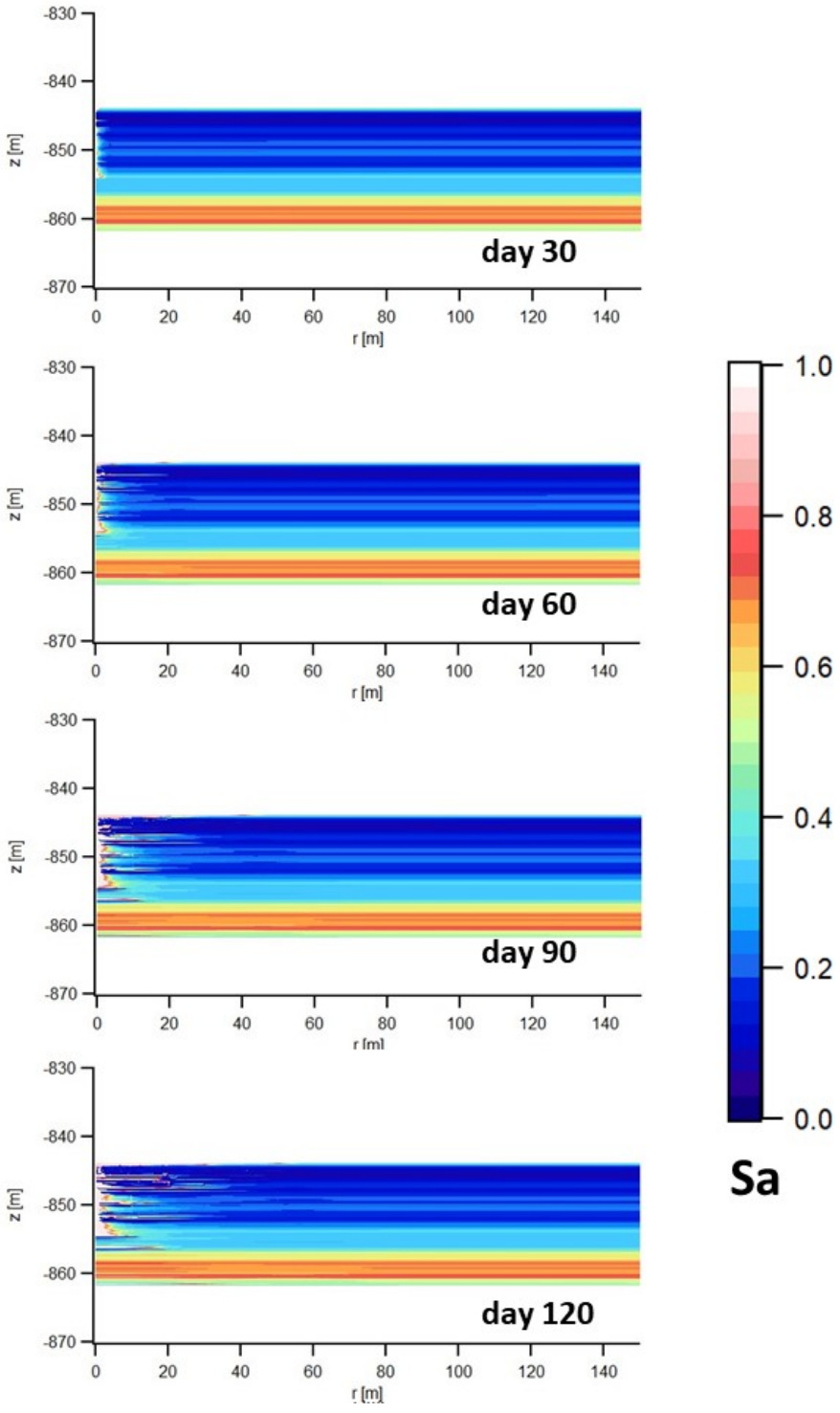

Figure 6.42: Case Ref: JFTS+H prediction of evolution of the spatial distribution of aqueous phase saturation in the well vicinity in the 2D problem of **Fig. 6.25** (Ref).

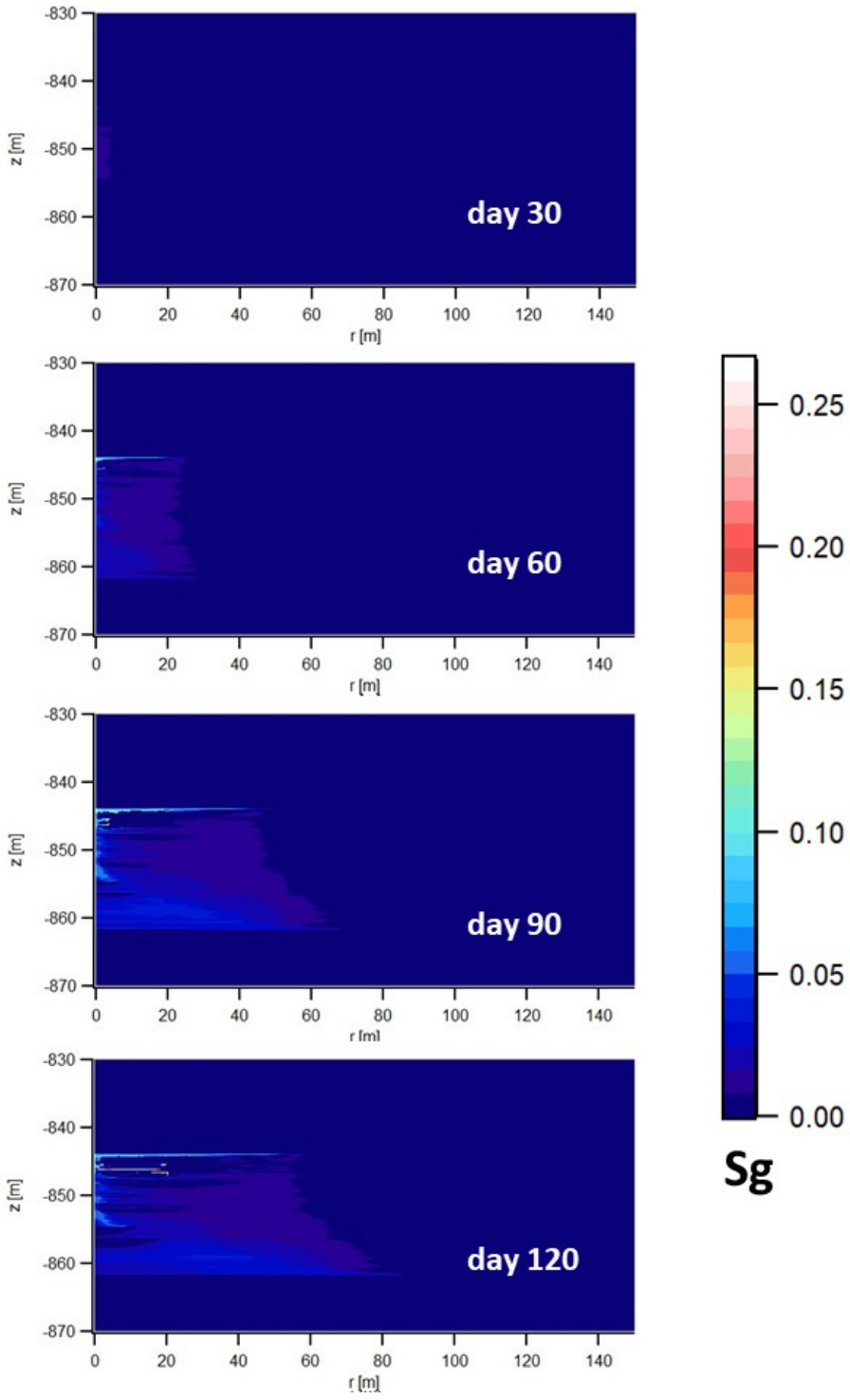

Figure 6.43: Case Ref: JFTS+H prediction of evolution of the spatial distribution of gas phase saturation in the well vicinity in the 2D problem of **Fig. 6.25** (Ref).

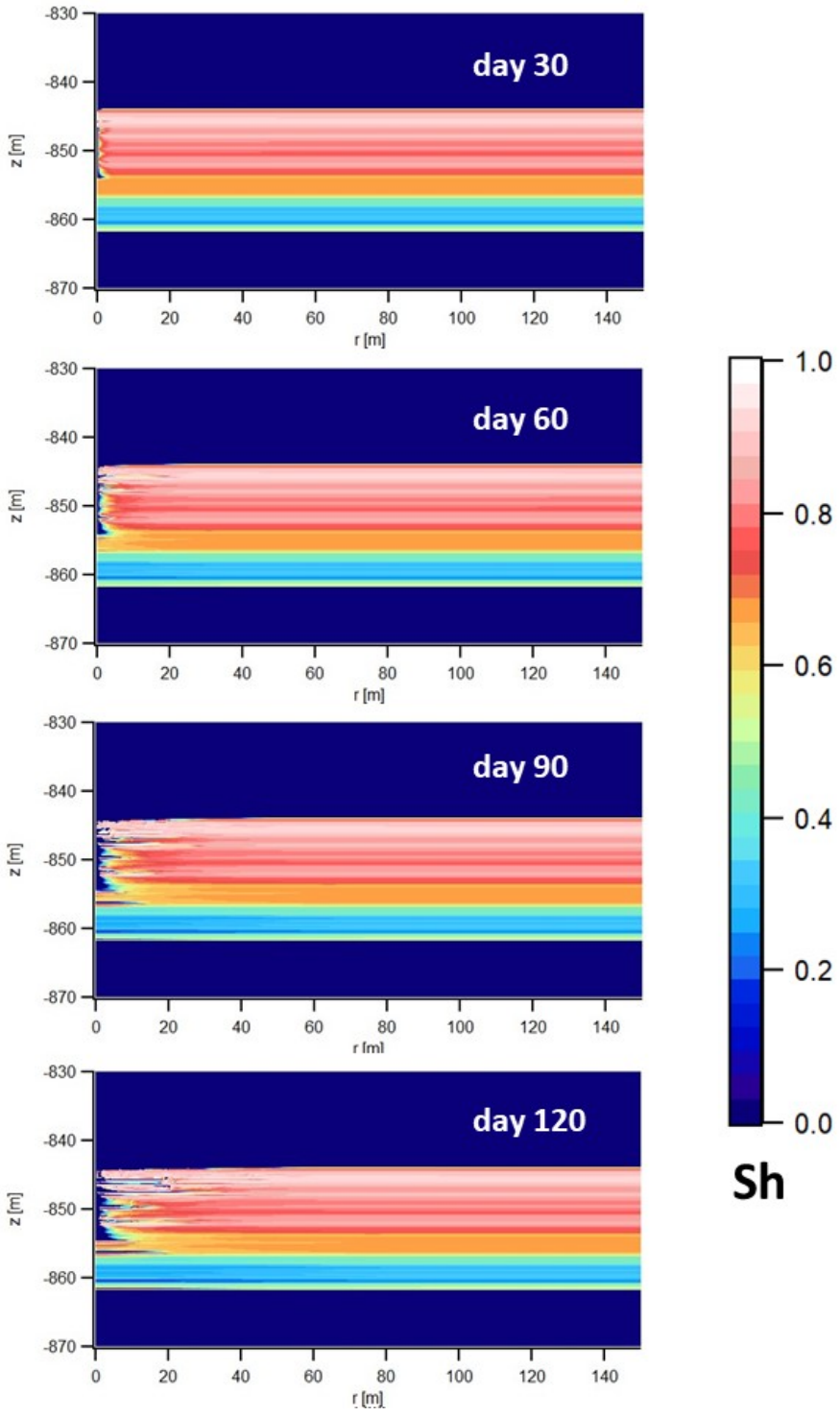

Figure 6.44: Case Ref: JFTS+H prediction of evolution of the spatial distribution of hydrate phase saturation in the well vicinity in the 2D problem of **Fig. 6.25** (Ref).

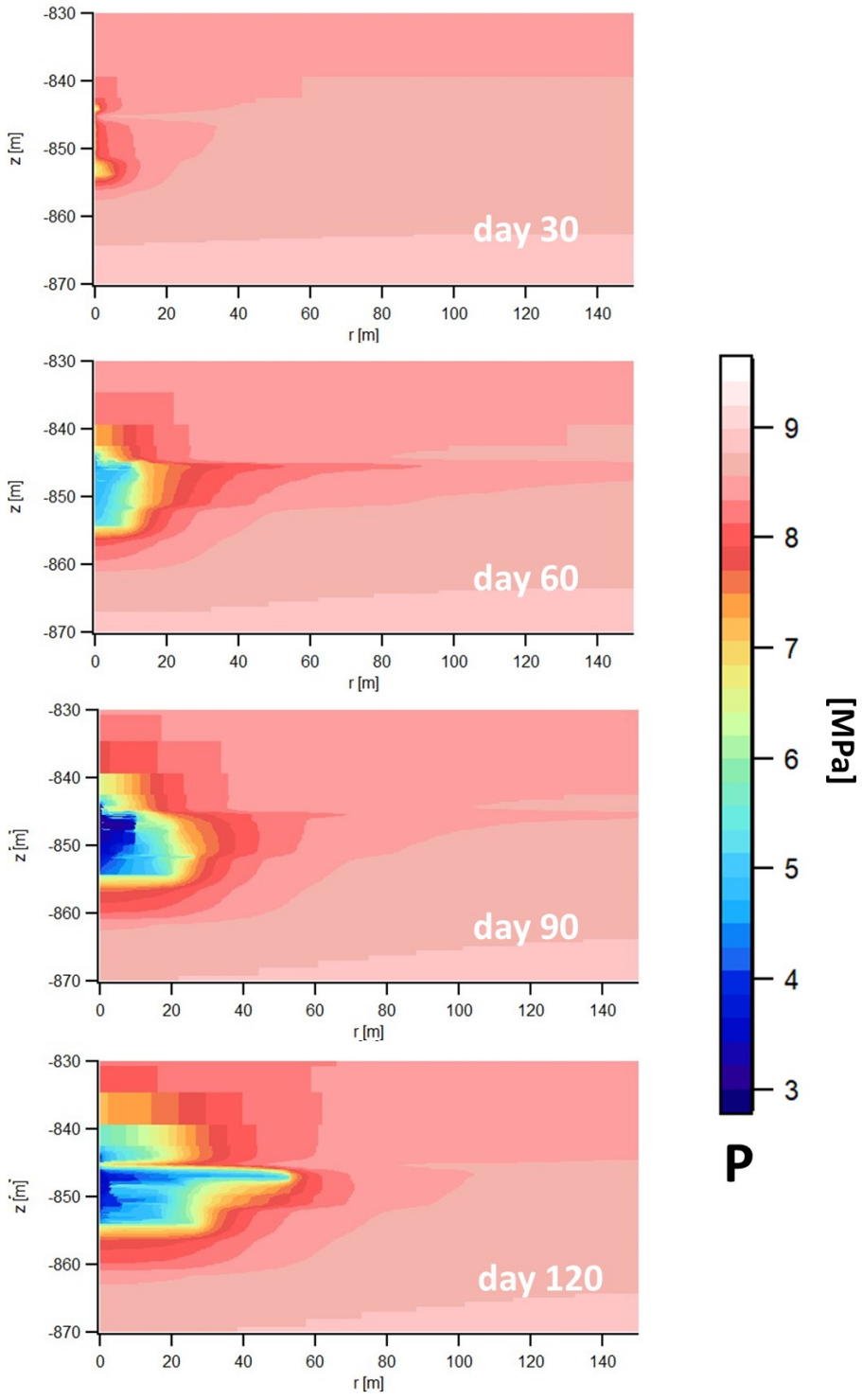

Figure 6.45: Case Ref: JFTS+H prediction of evolution of the spatial distribution of pressure in the well vicinity in the 2D problem of **Fig. 6.25** (nmax).

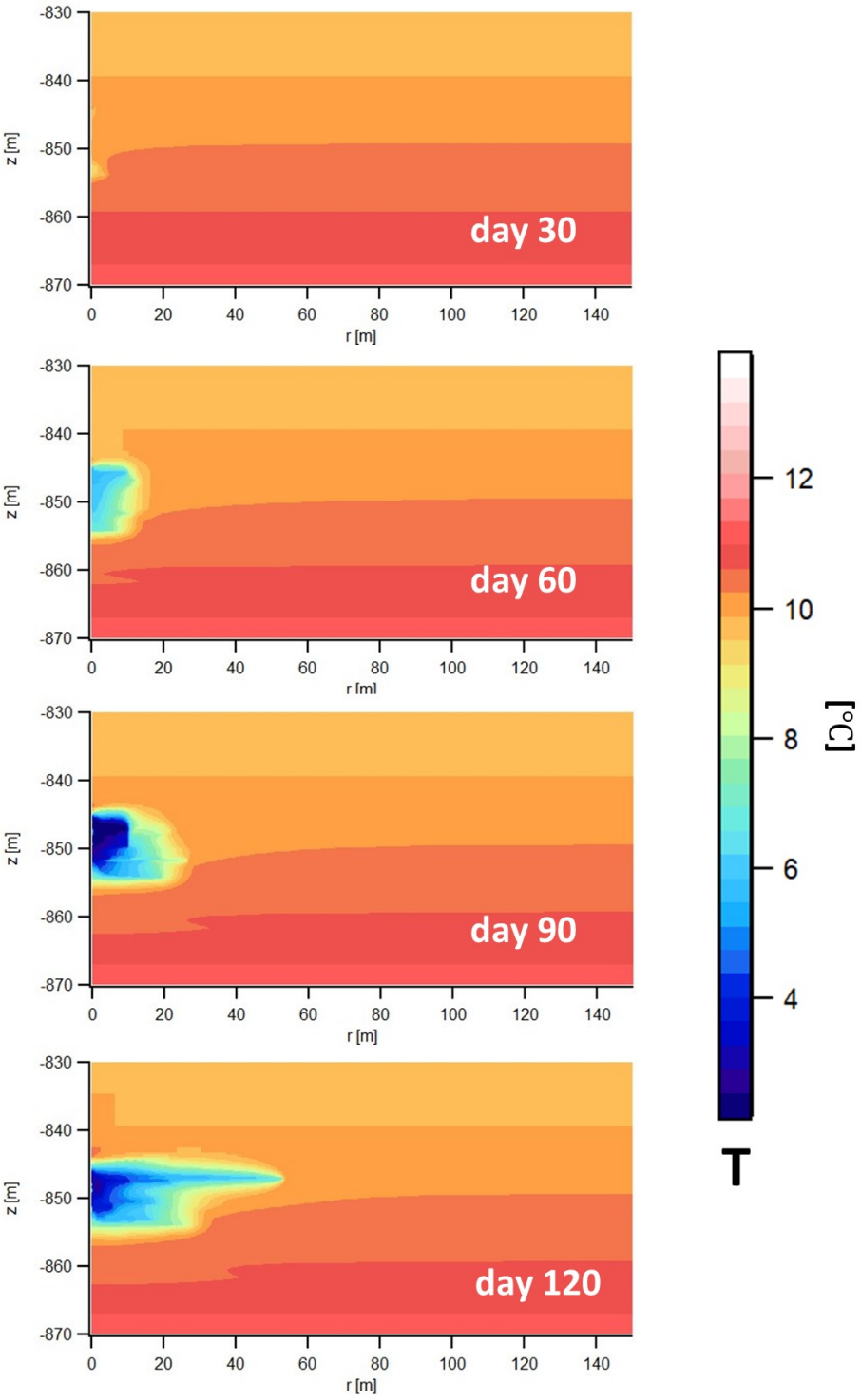

Figure 6.46: Case Ref: JFTS+H prediction of evolution of the spatial distribution of temperature in the well vicinity in the 2D problem of **Fig. 6.25** (nmax).

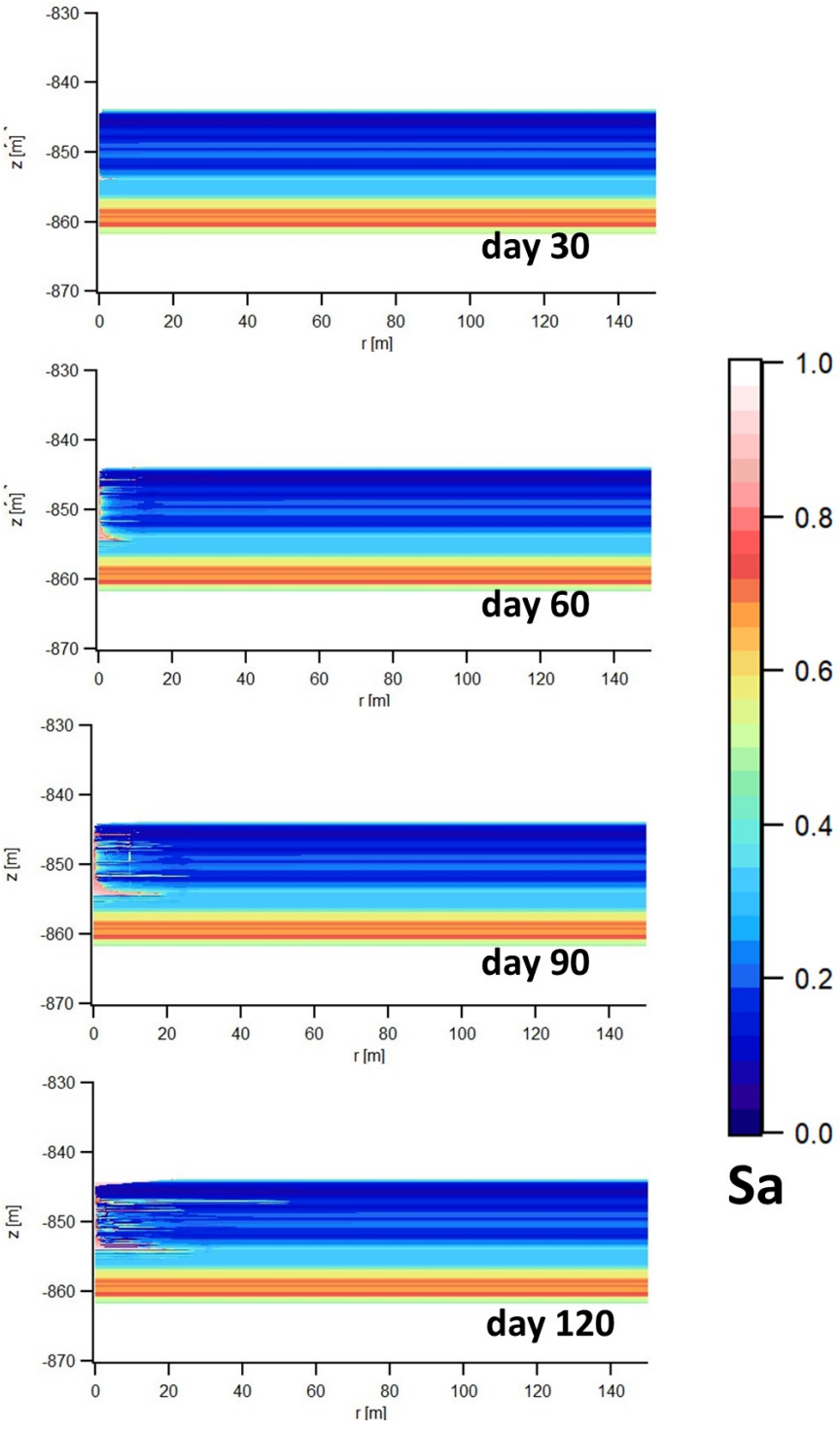

Figure 6.47: Case Ref: JFTS+H prediction of evolution of the spatial distribution of aqueous phase saturation in the well vicinity in the 2D problem of **Fig. 6.25** (nmax).

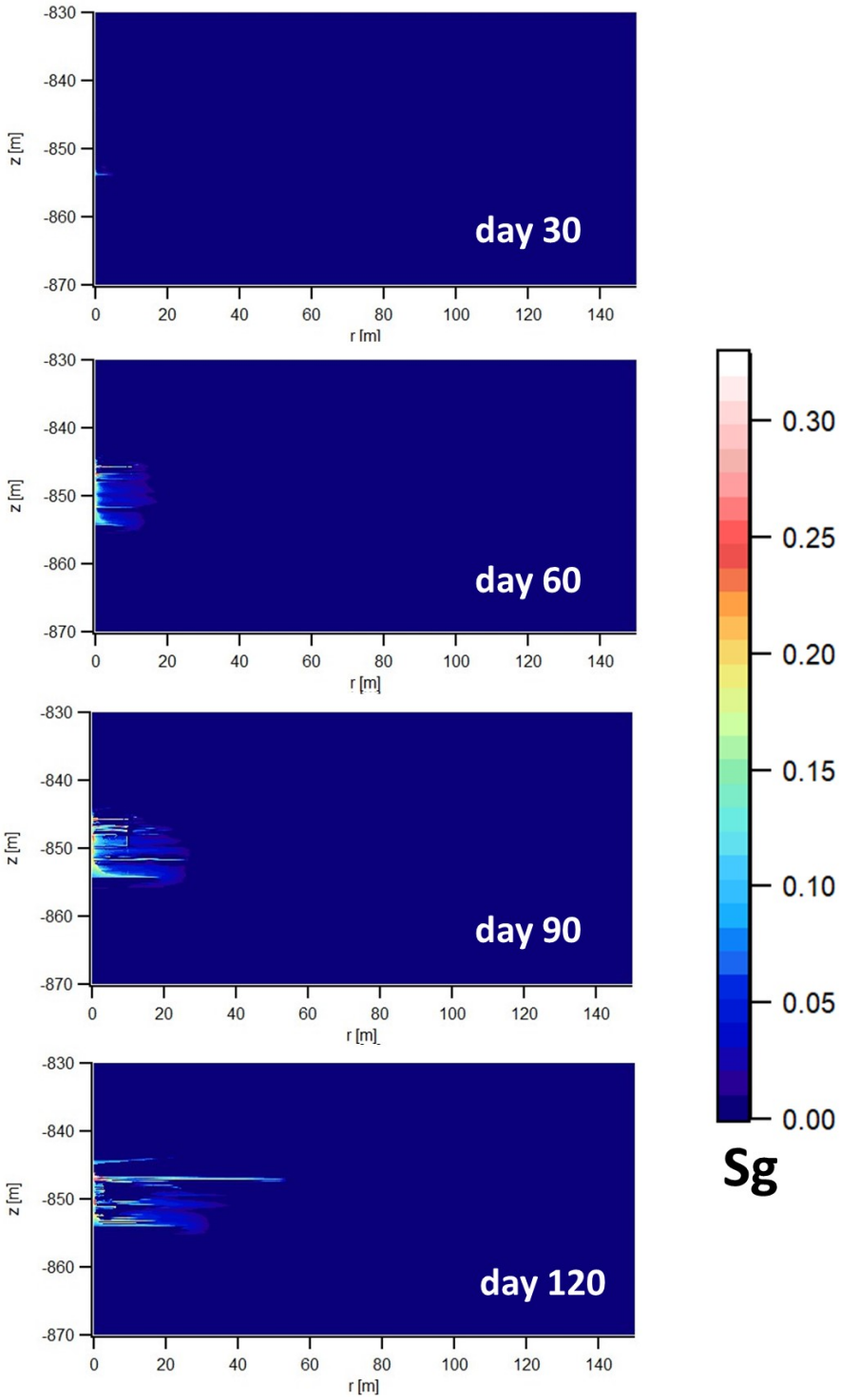

Figure 6.48: Case Ref: JFTS+H prediction of evolution of the spatial distribution of gas phase saturation in the well vicinity in the 2D problem of **Fig. 6.25** (nmax).

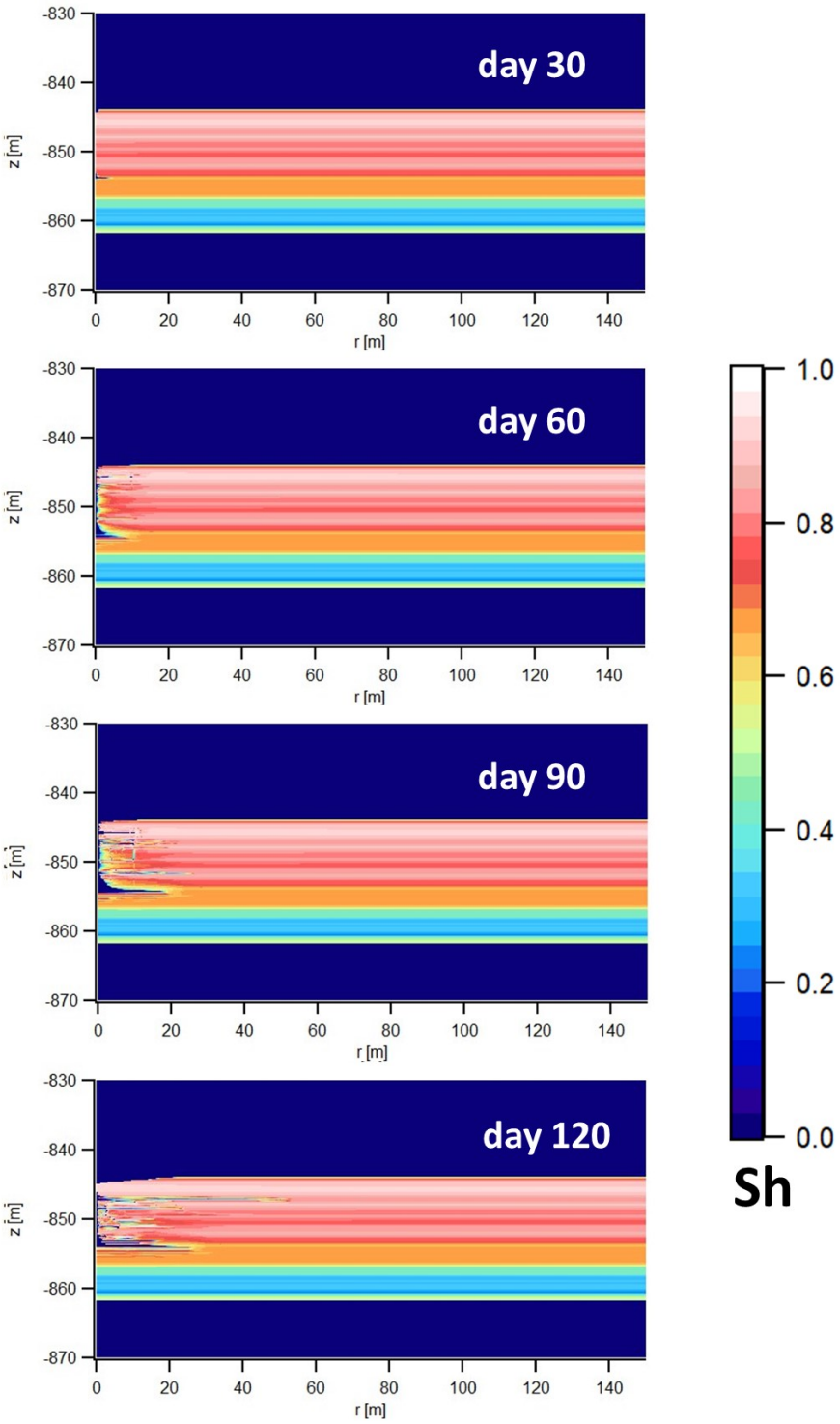

Figure 6.49: Case Ref: JFTS+H prediction of evolution of the spatial distribution of hydrate phase saturation in the well vicinity in the 2D problem of **Fig. 6.25** (nmax).

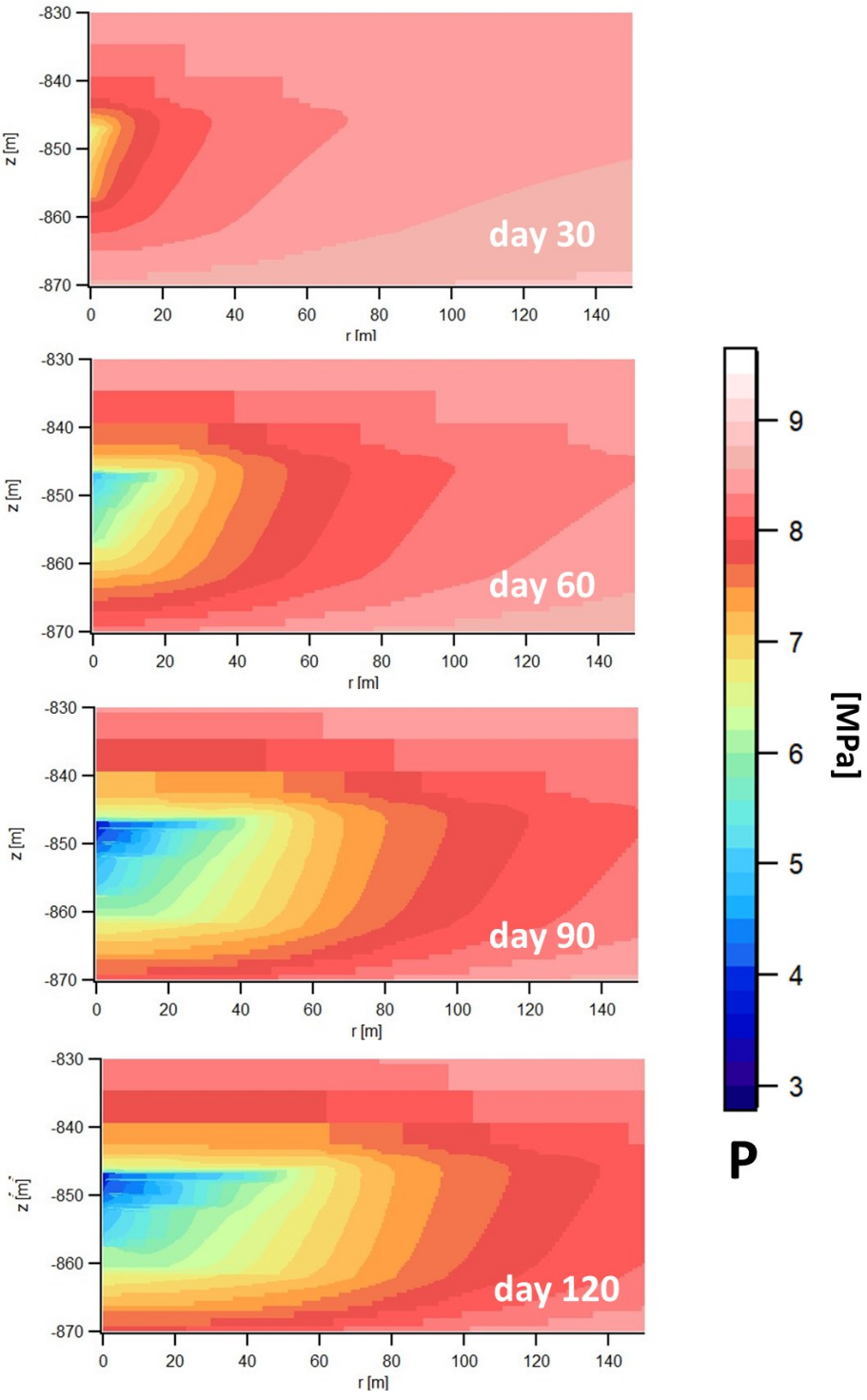

Figure 6.50: Case Ref: JFTS+H prediction of evolution of the spatial distribution of pressure in the well vicinity in the 2D problem of **Fig. 6.25** (LoW).

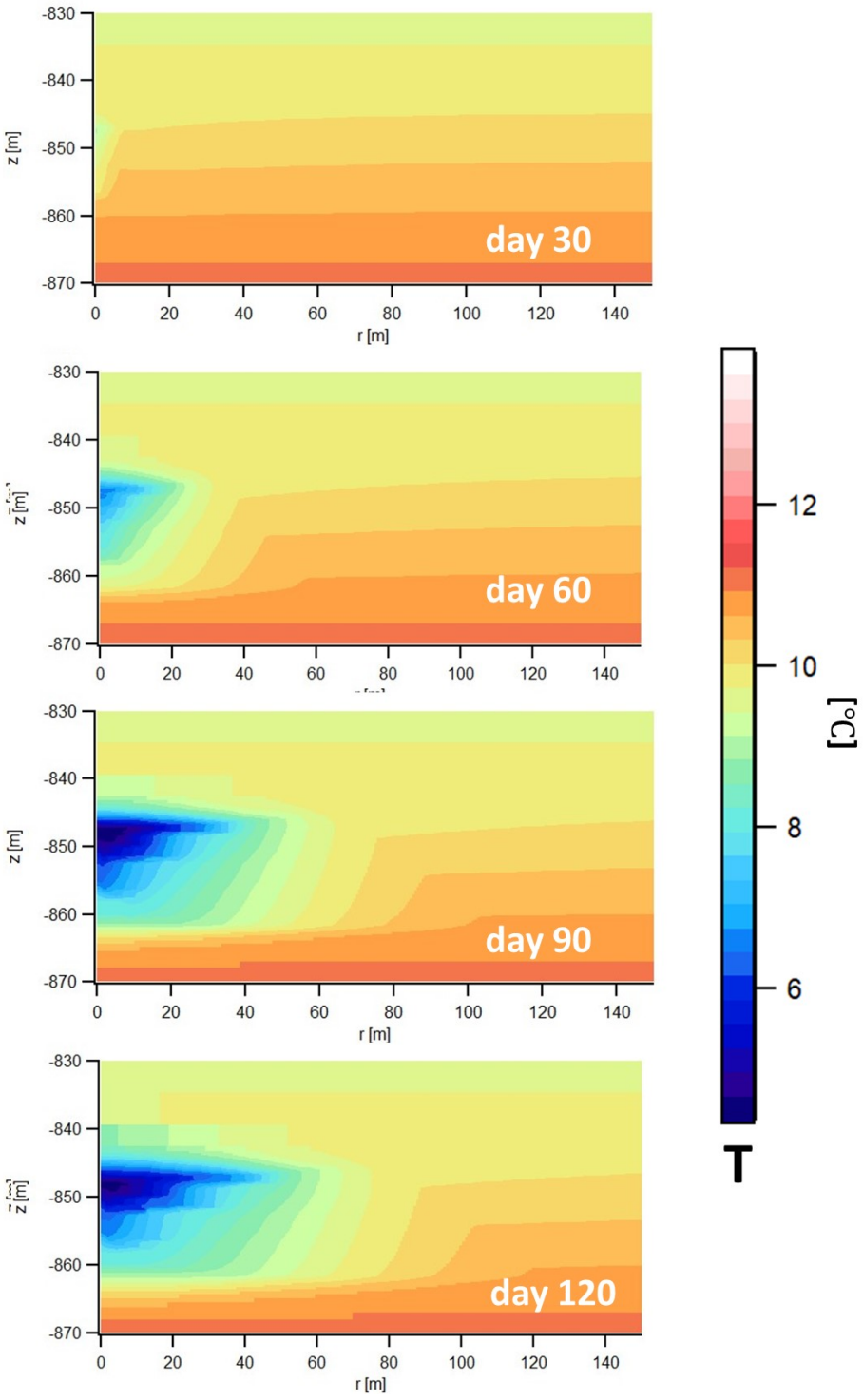

Figure 6.51: Case Ref: JFTS+H prediction of evolution of the spatial distribution of temperature in the well vicinity in the 2D problem of **Fig. 6.25** (LoW).

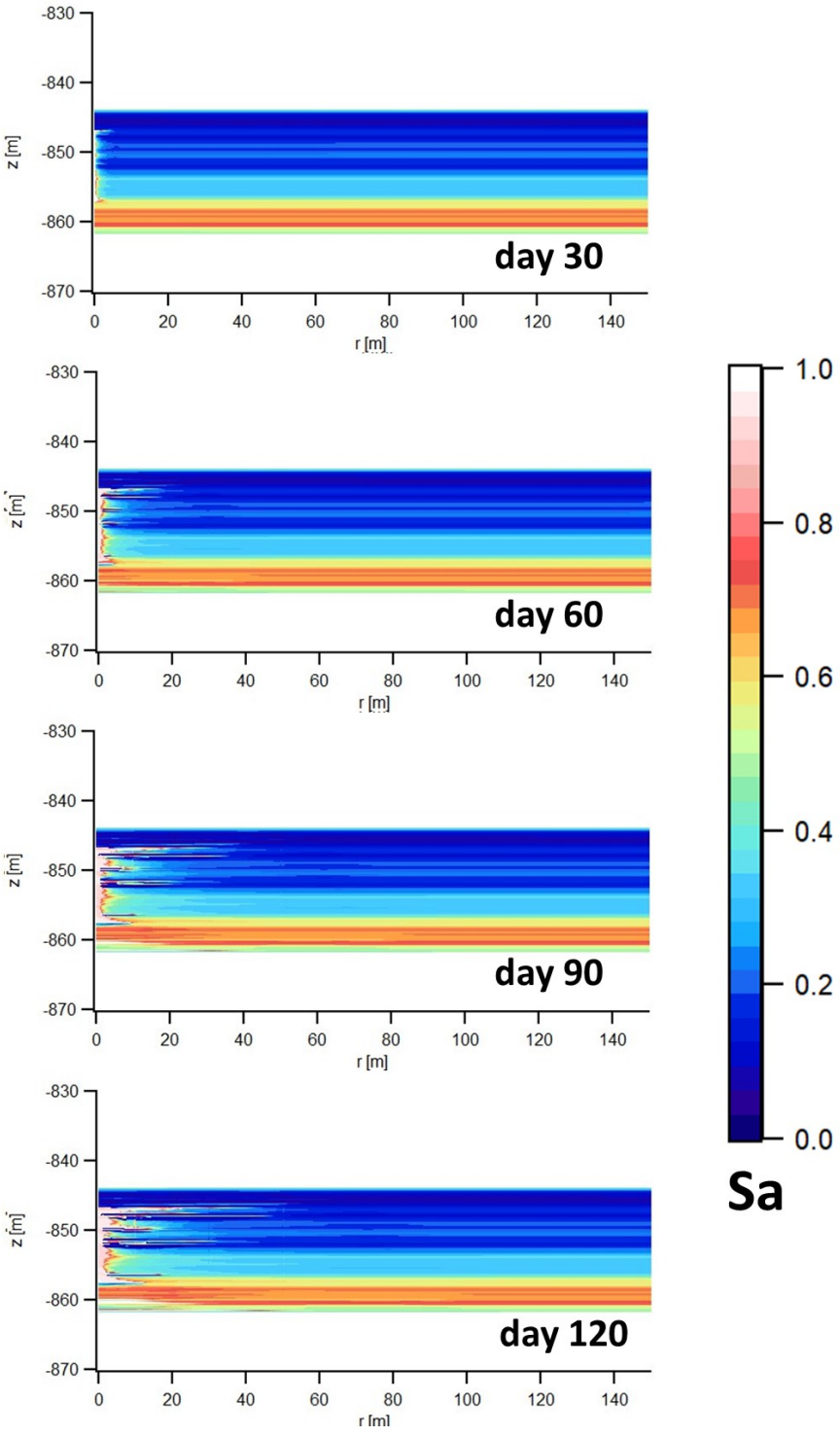

Figure 6.52: JCase Ref: FTS+H prediction of evolution of the spatial distribution of aqueous phase saturation in the well vicinity in the 2D problem of **Fig. 6.25** (LoW).

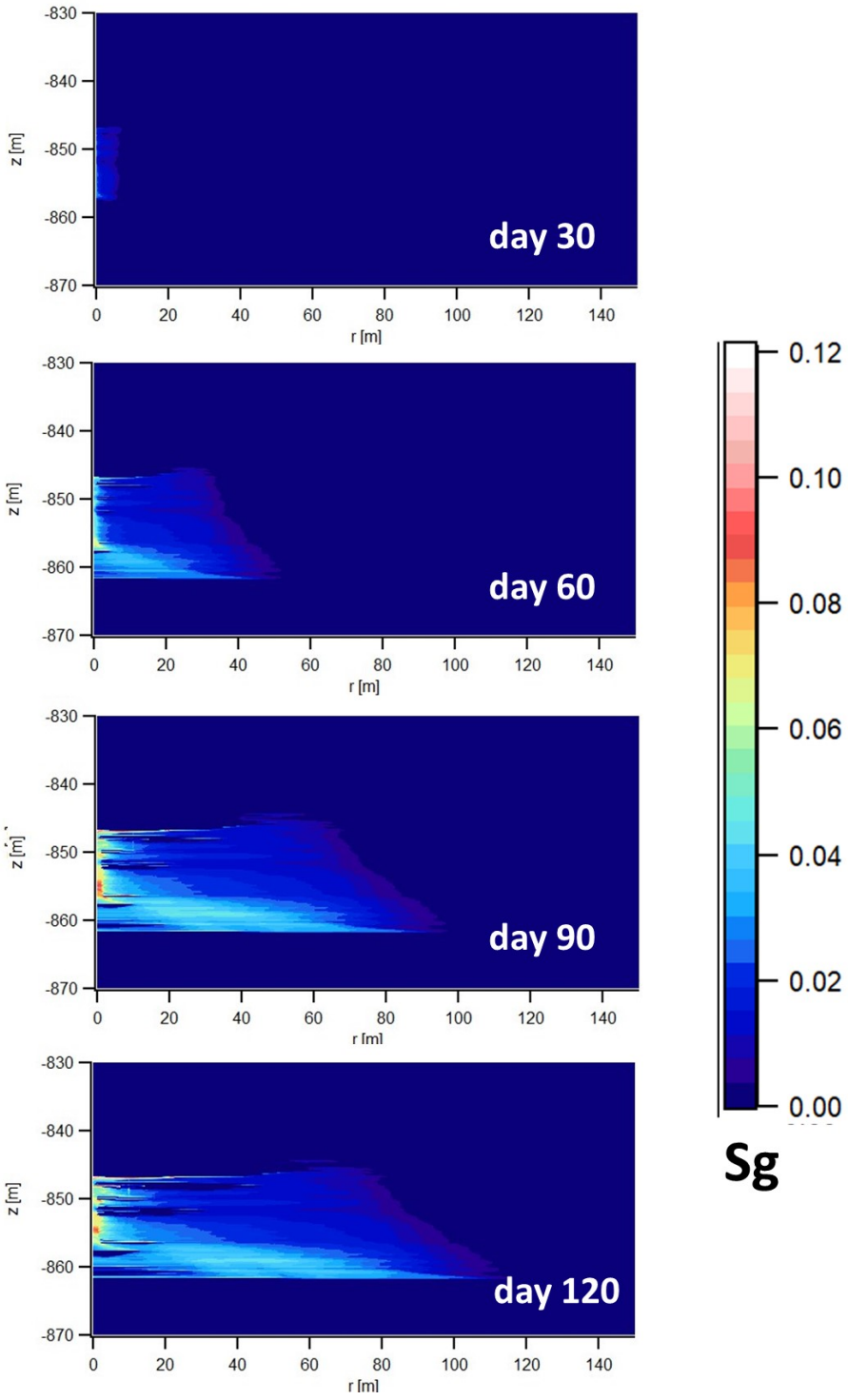

Figure 6.53: Case Ref: JFTS+H prediction of evolution of the spatial distribution of gas phase saturation in the well vicinity in the 2D problem of **Fig. 6.25** (LoW).

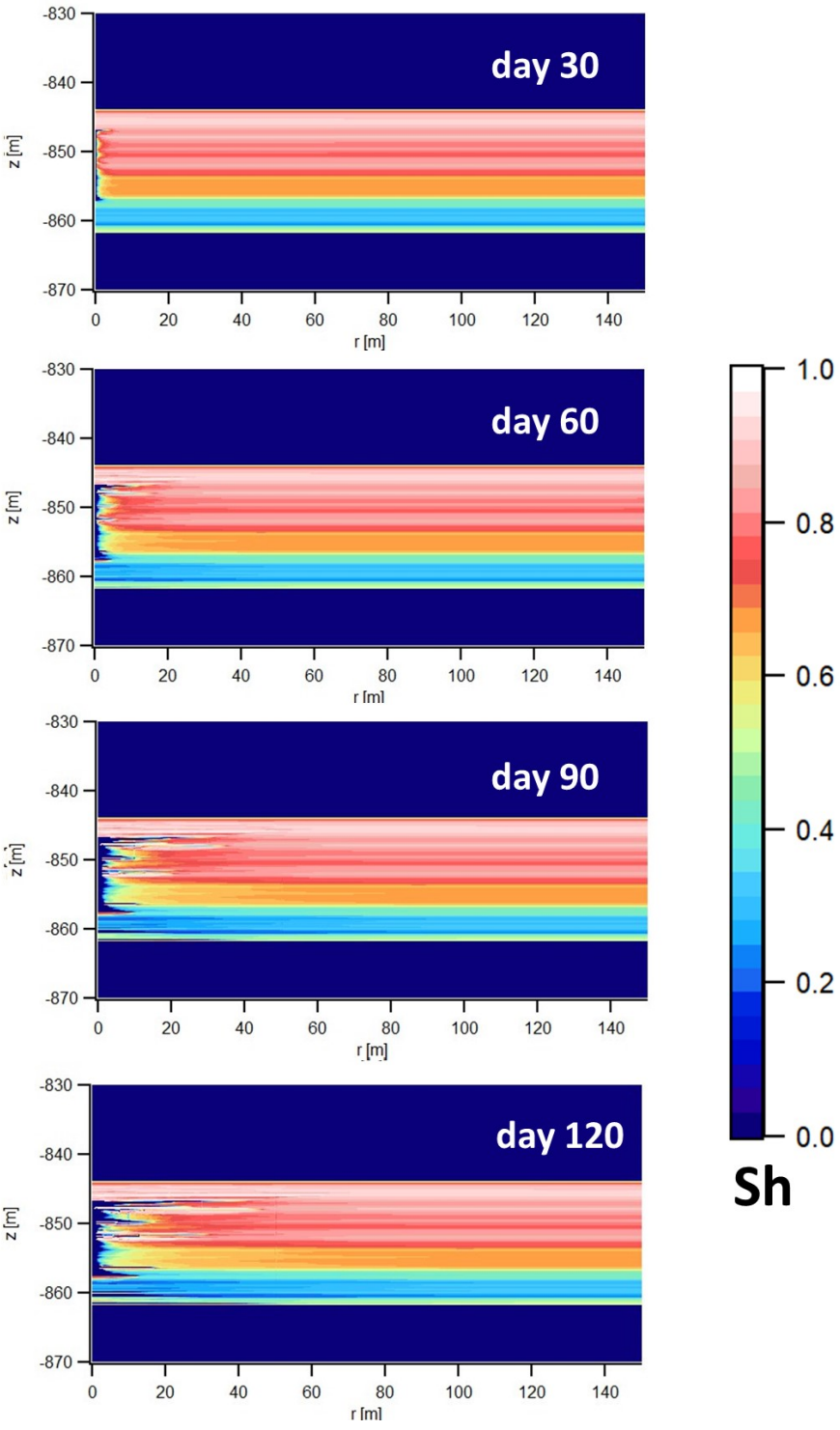

Figure 6.54: Case Ref: JFTS+H prediction of evolution of the spatial distribution of hydrate phase saturation in the well vicinity in the 2D problem of **Fig. 6.25** (LoW).

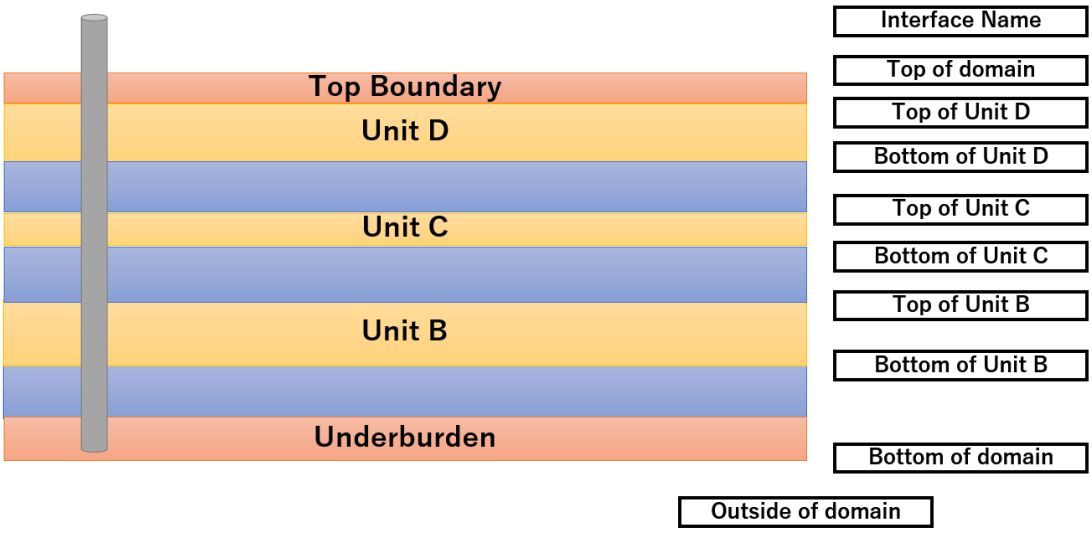

Figure 6.55: Locations of key interfaces in the 2D problem of **Fig. 6.25**.

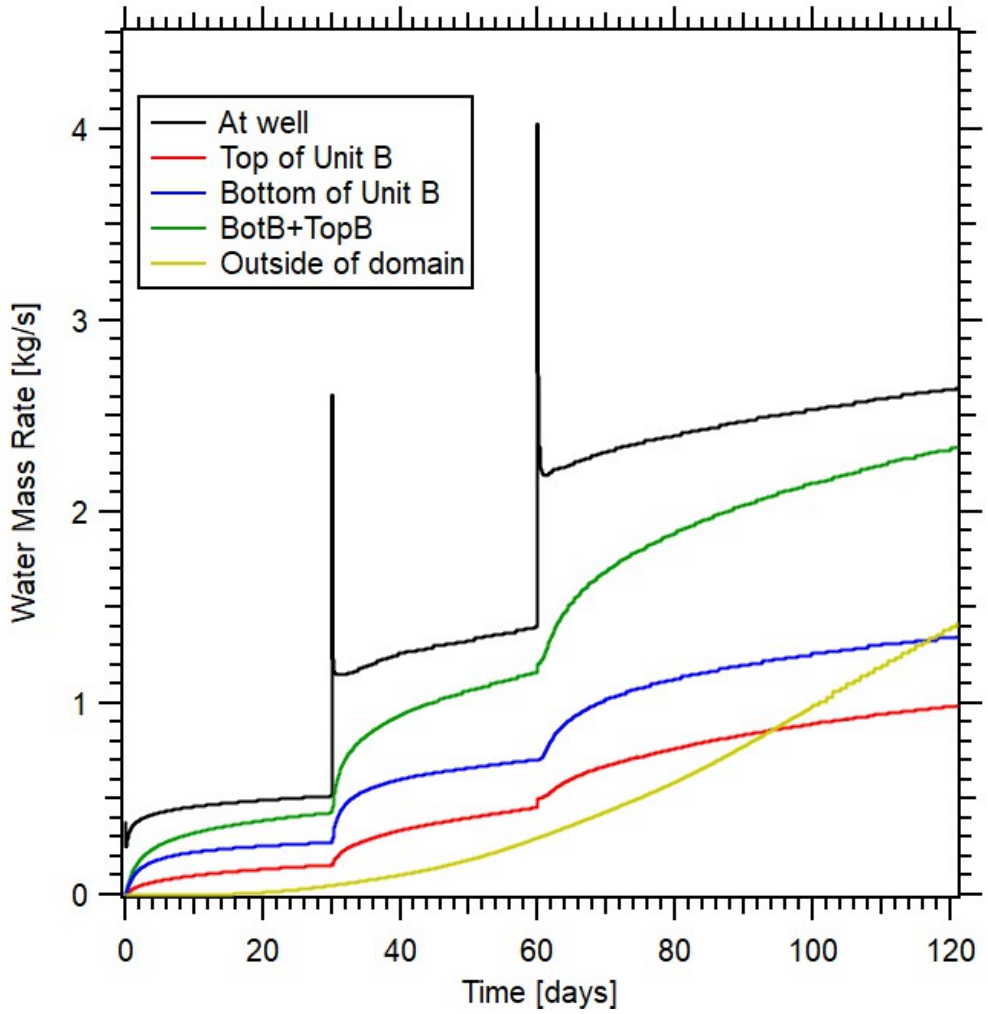

Figure 6.56: JFTS+H prediction of evolution of water inflows across key interfaces and water production at the well in the 2D problem of **Fig. 6.25** (Ref).

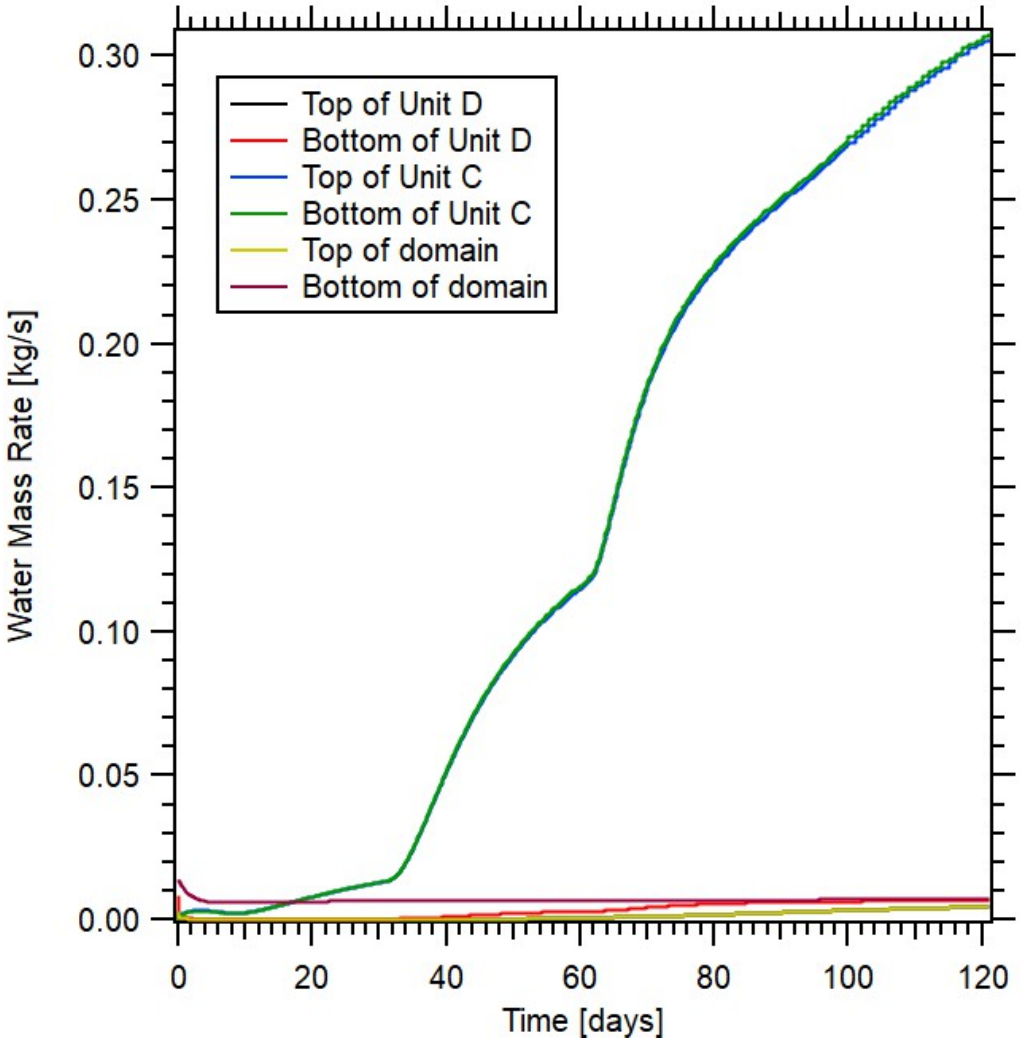

Figure 6.57: JFTS+H prediction of evolution of water inflows across key interfaces and water production at the well in the 2D problem of **Fig. 6.25** (nmax).

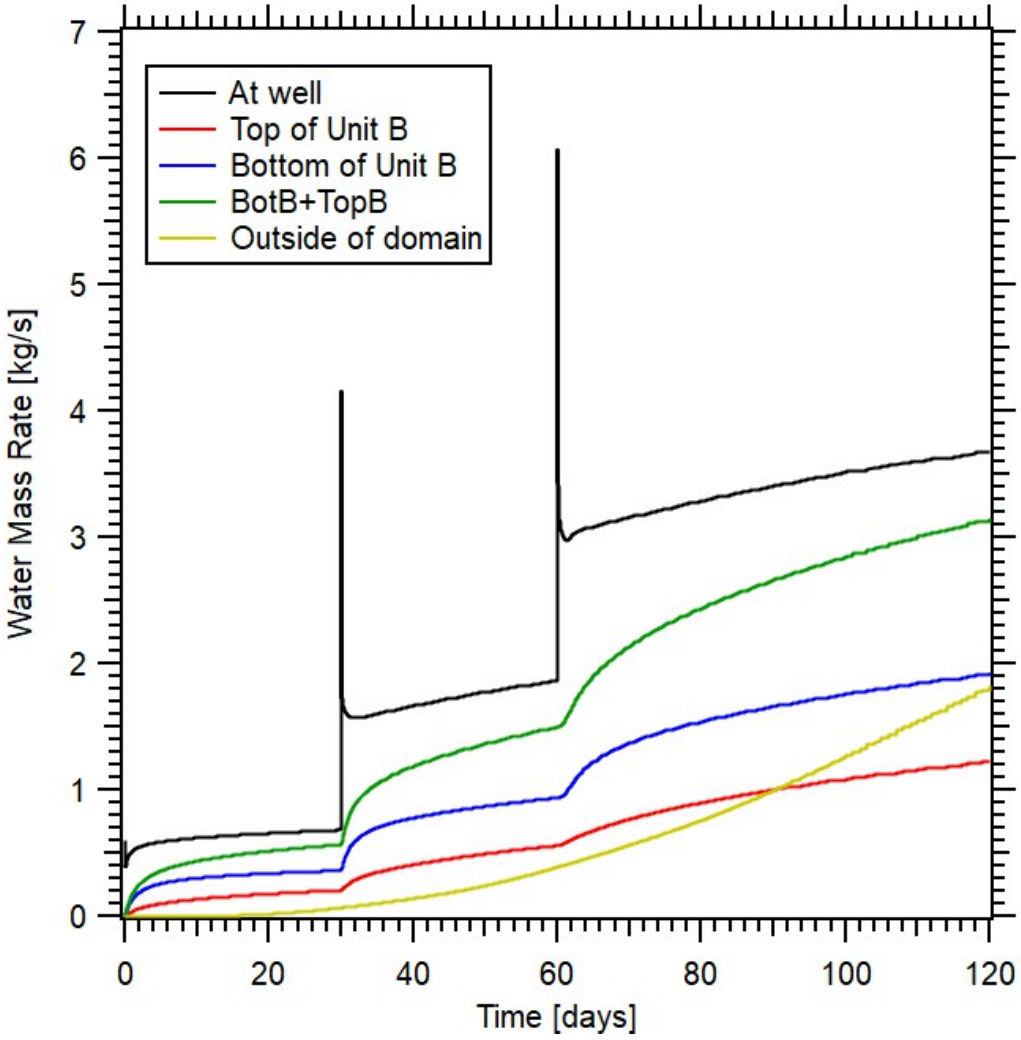

Figure 6.58: JFTS+H prediction of evolution of water inflows across key interfaces and water production at the well in the 2D problem of **Fig. 6.25** (LoW).

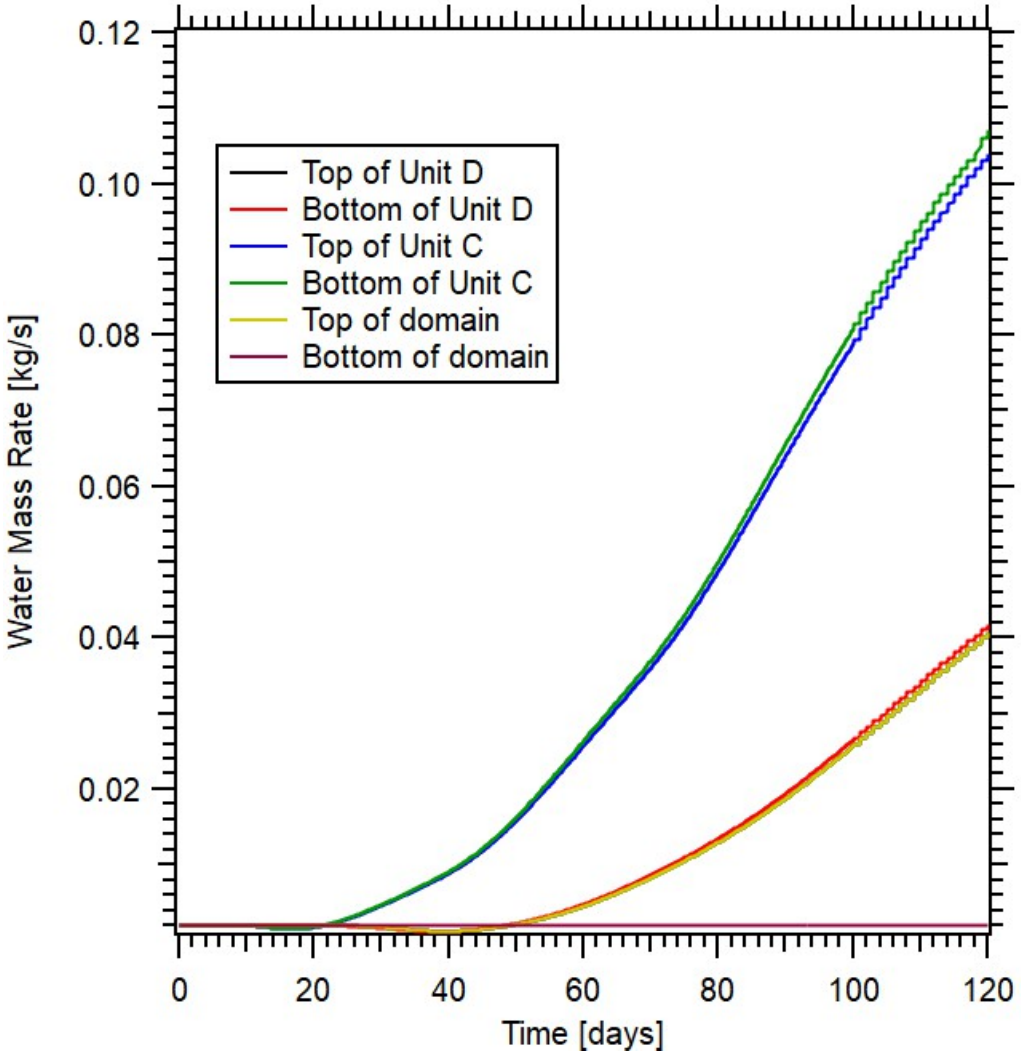

Figure 6.59: JFTS+H prediction of evolution of water inflows across key interfaces in the 2D problem of **Fig. 6.25** (Ref).

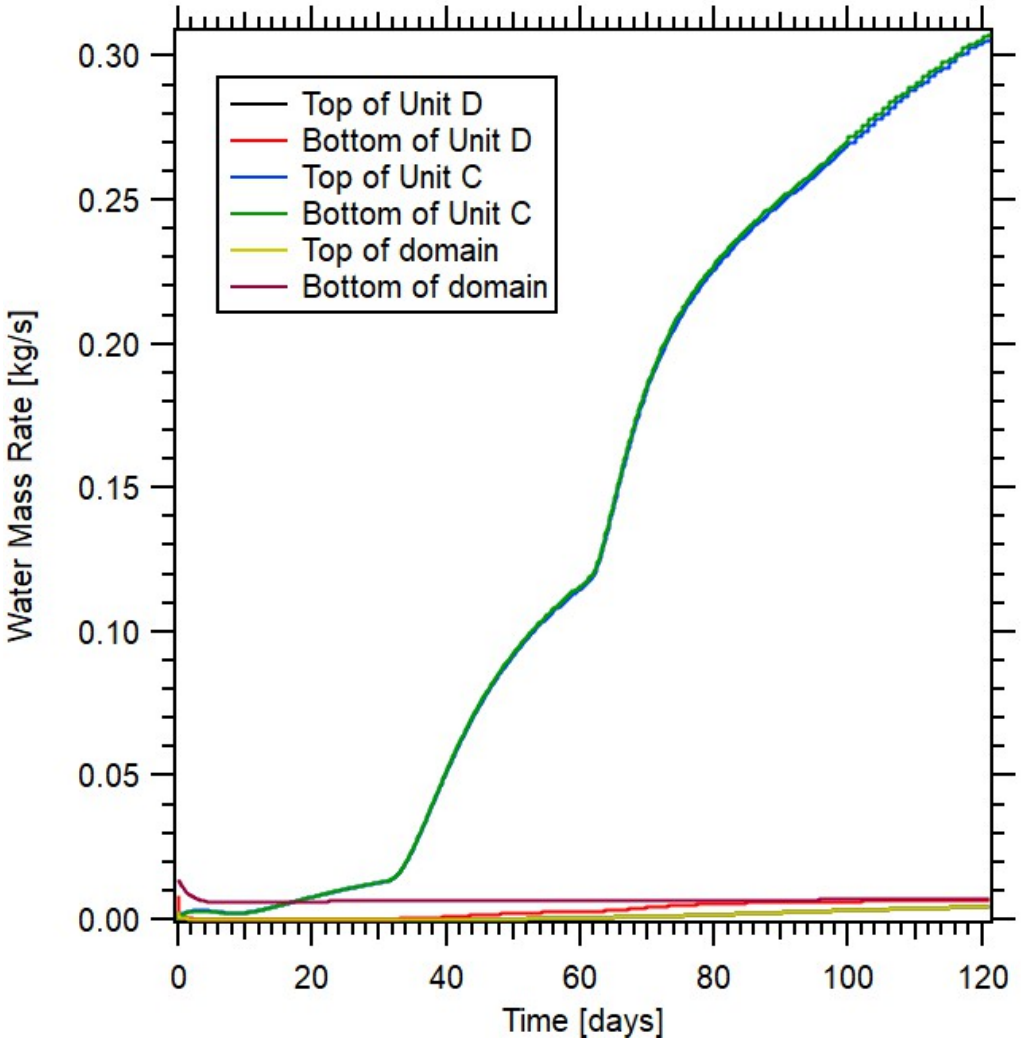

Figure 6.60: JFTS+H prediction of evolution of water inflows across key interfaces in the 2D problem of **Fig. 6.25** (nmax).

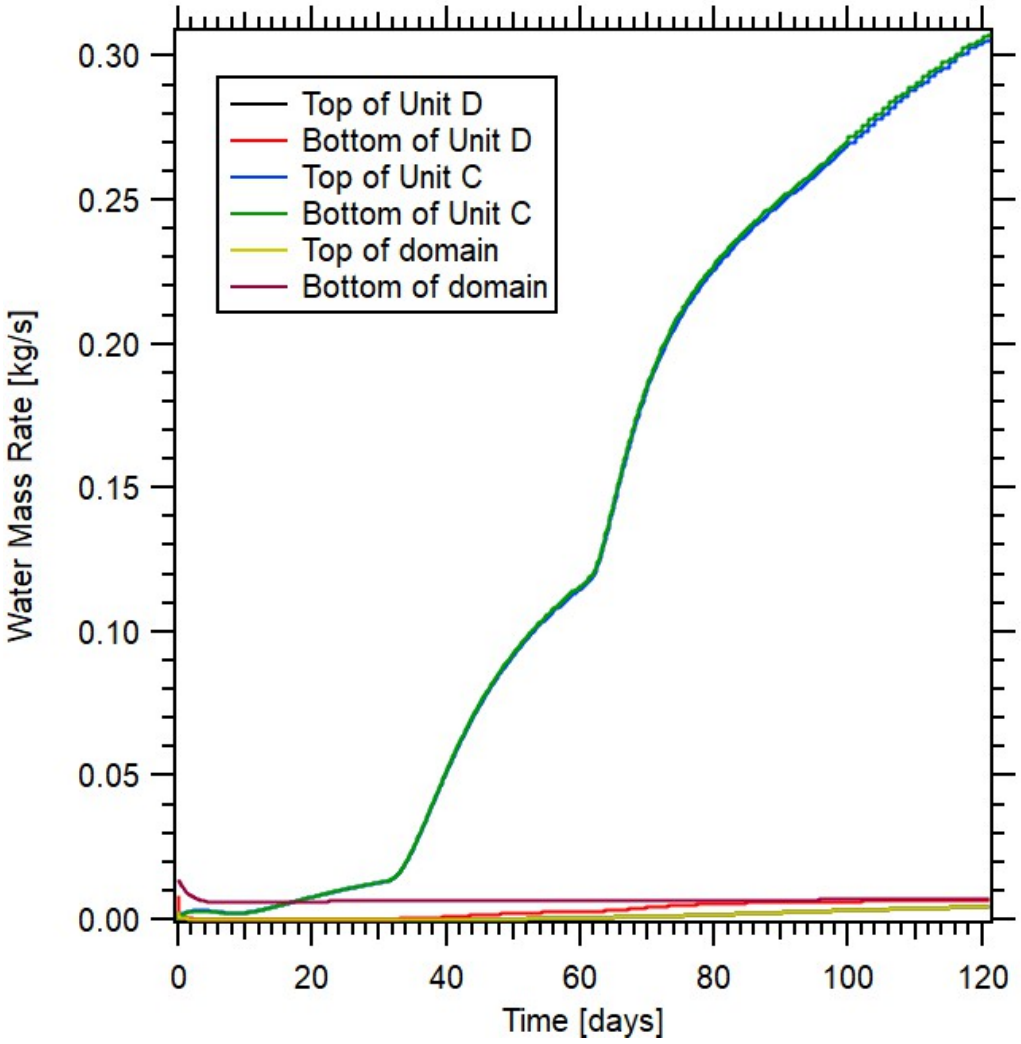

Figure 6.61: JFTS+H prediction of evolution of water inflows across key interfaces in the 2D problem of **Fig. 6.25** (LoW).

6.2.4 Parallel Computation Results and Performance

As in all previous simulations, JFTS+H is shown to work well and to exhibit a very promising parallel performance in the solution of this problem on both the Ada supercomputer cluster and the MacPro desktop computer. **Figs 6.62** and **6.63** show the effect of the number of processors on the total elapsed time for the two computational platforms and show significant improvements in performance in both environments. **Figs 6.64** and **6.65** show the relationship between the speed-up and the number of processors. This problem is too huge to compute in serial code. I estimated

the serial time from the result of the minimum number of processors (i.e. 16 for the Ada cluster and 8 for the MacPro). Efficiency calculated from the estimated speed-up are shown in **Figs 6.66** and **6.67**. The results on the Ada cluster indicate further improvements can be expected with the parallel simulation with larger number of processors. In this study, the available execution time for large number of processors is limited.

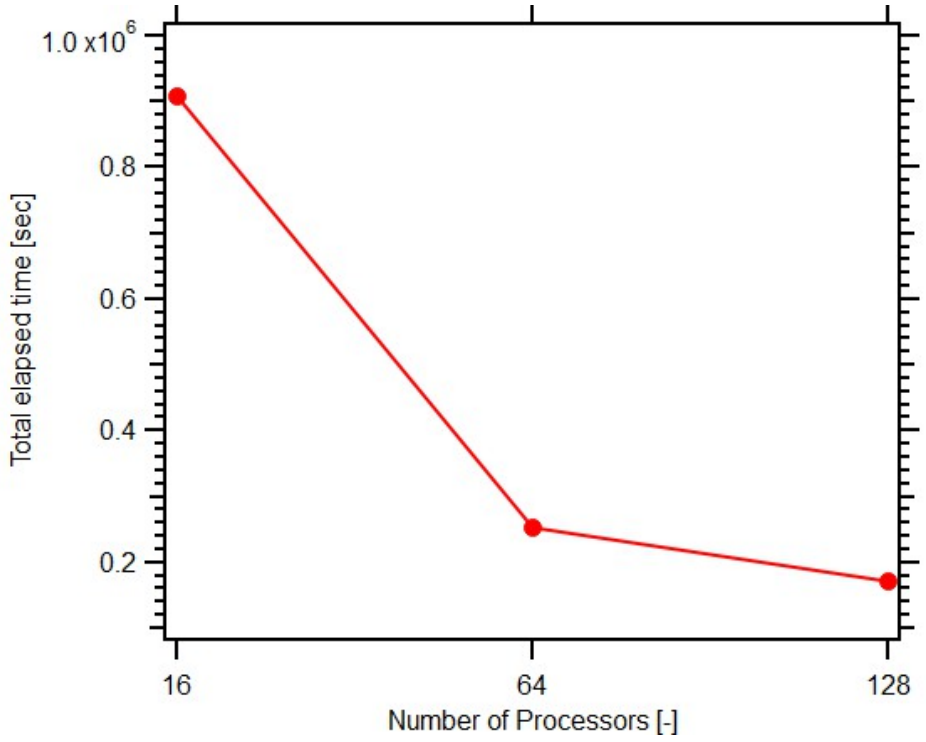

Figure 6.62: JFTS+H parallel performance: the relationship between total elapsed time and number of processors in the study of **Fig. 6.25** on the Ada cluster.

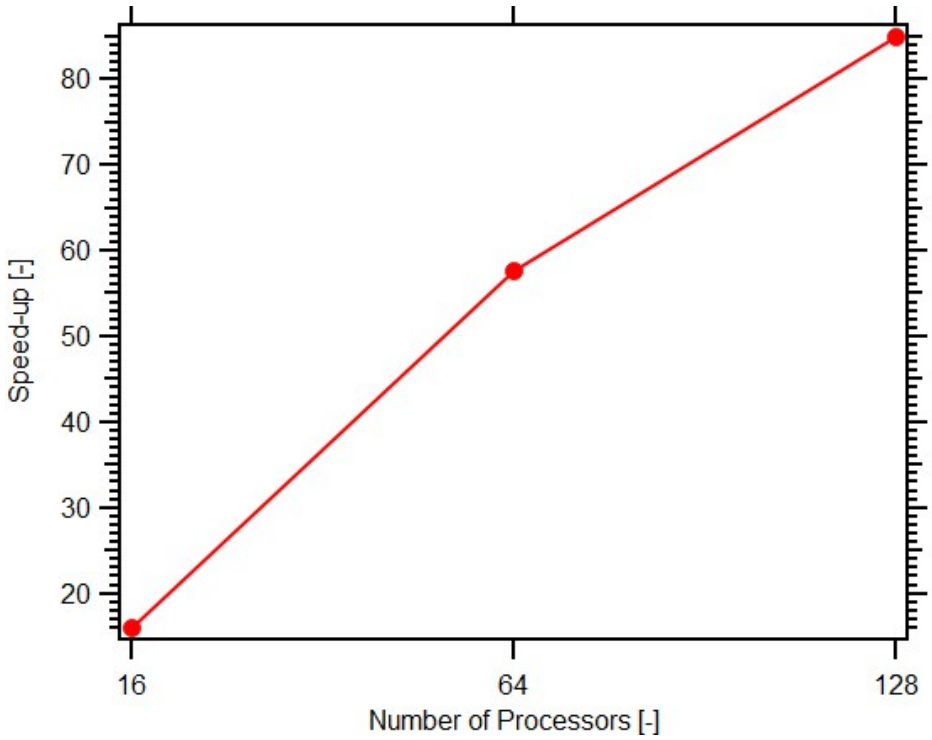

Figure 6.63: JFTS+H parallel performance: the relationship between speed-up and number of processors in the study of **Fig. 6.25** on the Ada cluster.

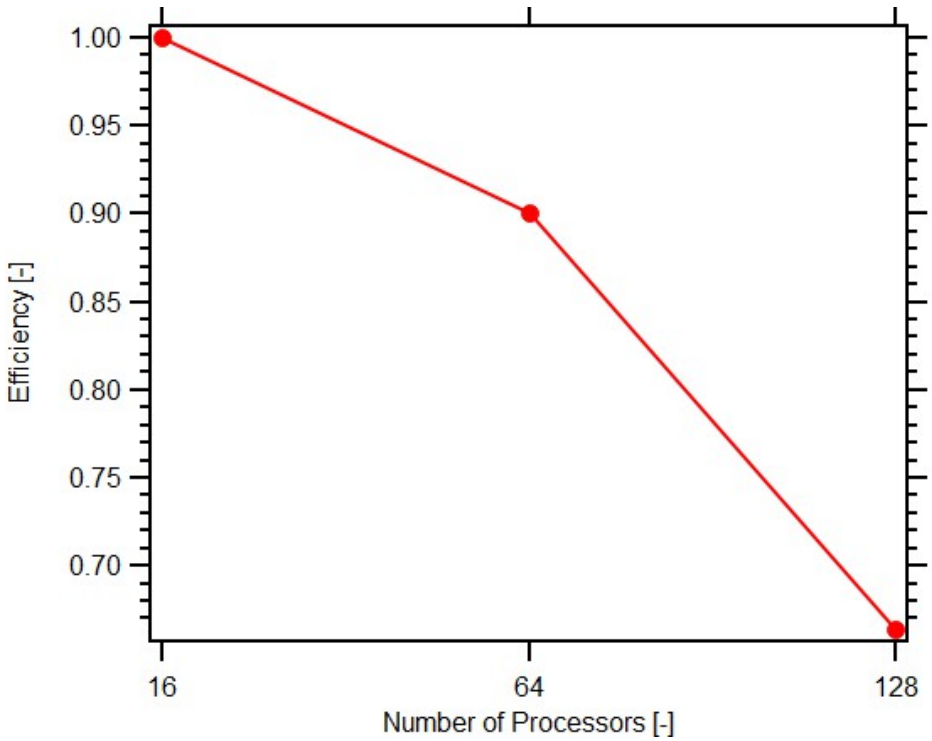

Figure 6.64: JFTS+H parallel performance: the relationship between efficiency and number of processors in the study of **Fig. 6.25** on the Ada cluster.

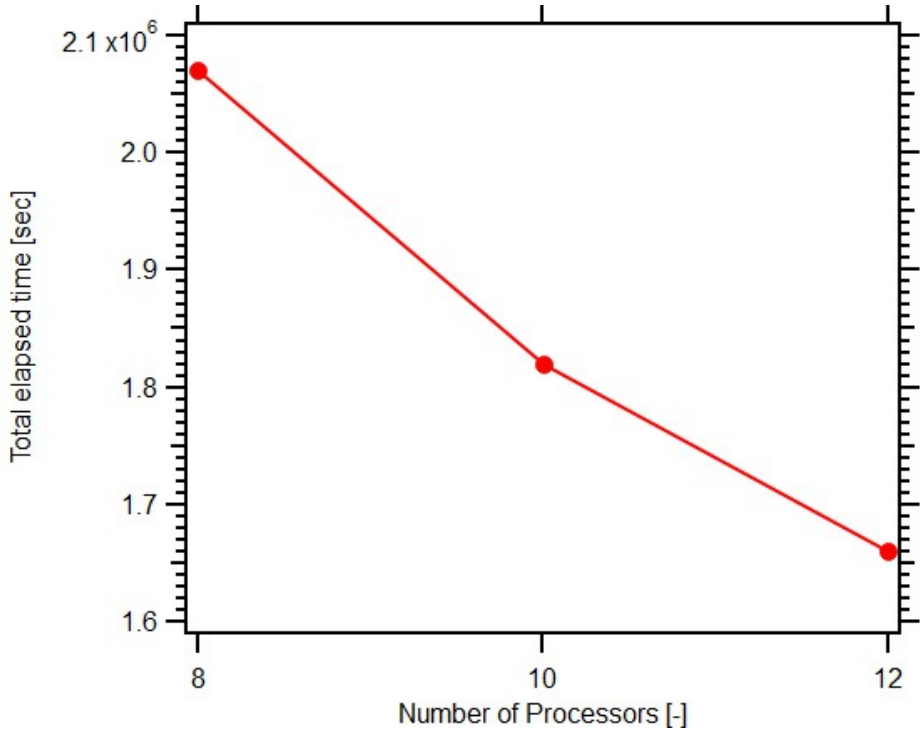

Figure 6.65: JFTS+H parallel performance: the relationship between total elapsed time and number of processors in the study of **Fig. 6.25** on the MacPro.

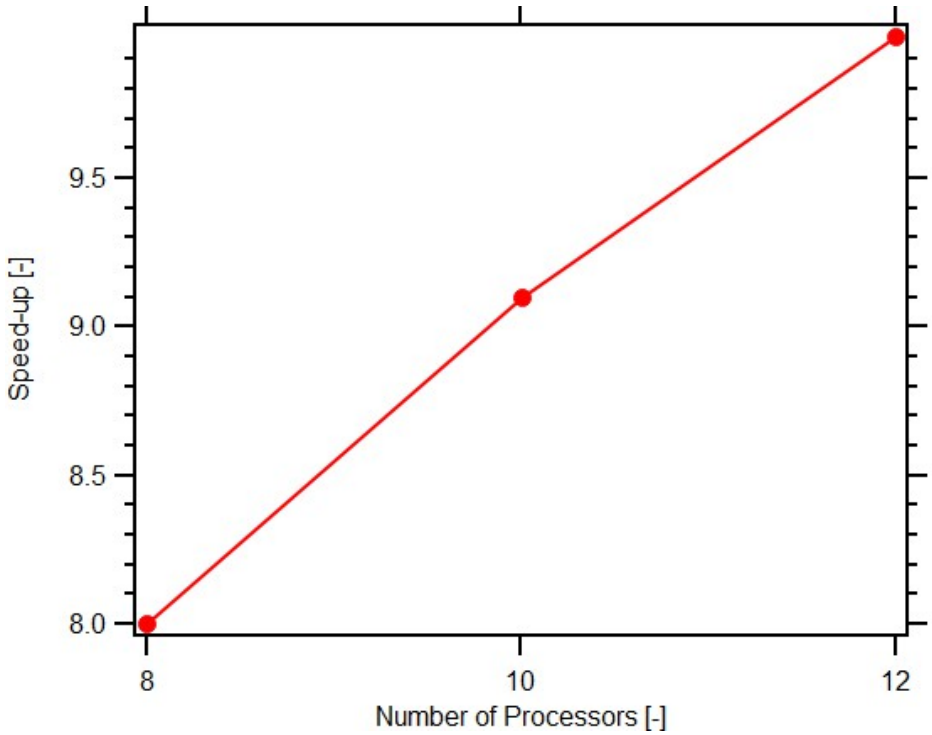

Figure 6.66: JFTS+H parallel performance: the relationship between speed-up and number of processors in the study of **Fig. 6.25** on the MacPro.

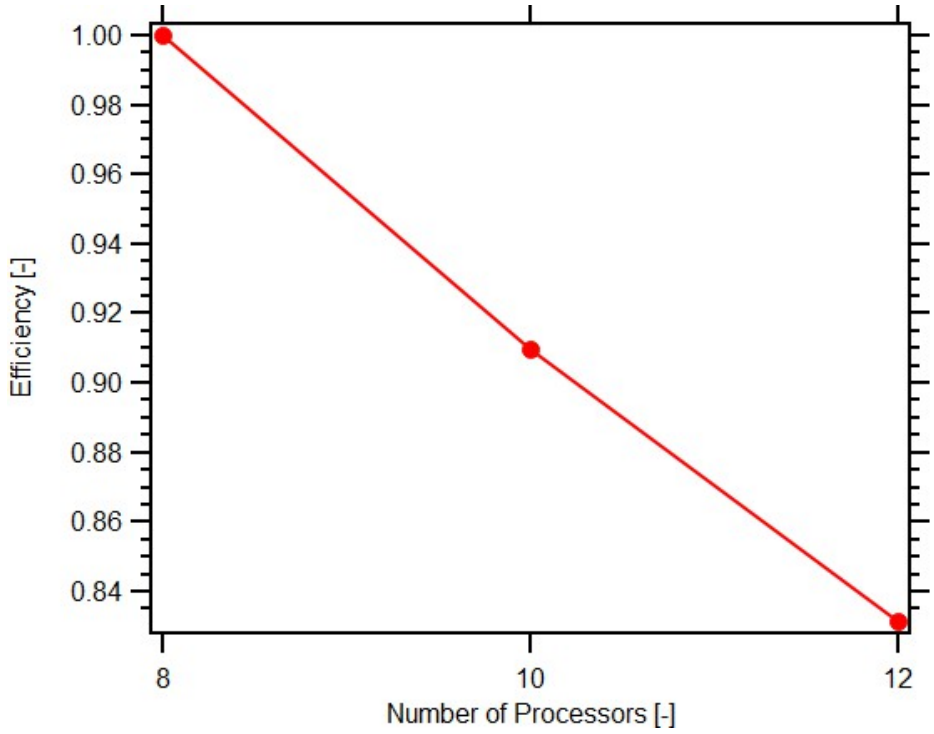

Figure 6.67: JFTS+H parallel performance: the relationship between efficiency and number of processors in the study of **Fig. 6.25** on the MacPro.

7. SUMMARY AND CONCLUSIONS

7.1 Deliverables

- JFTS, a fully-implicit, Jacobian-based 3D simulator as the general framework for the solution of problems of coupled flow and transport of fluids and heat in reactive, non-isothermal systems. JFTS is written in Julia and offers both a serial version and a parallel version that is based on the MPI-approach, uses the METIS domain decomposition method and solves the large Jacobian matrices using the LIS package of parallel Conjugate Gradient (CG) solvers.

- A JFTS module (also written in Julia and based on the MPI approach) for the solution of the problem of hydrate behavior in complex geologic media that describes the phase diagram of the H_2O+CH_4 system, and recommendations of appropriate preconditioned CG options in the LIS package for efficient solution of the resulting Jacobian.

- Extensive predictive capabilities of the resulting JFTS+H code that include detailed description of the evolution over time of (a) the hydrate dissociation/formation, (b) fluid production, and (c) spatial distributions of all important phase saturations and thermophysical properties, all in formats conducive to easy visualization.

- Validation of the JFTS+H code against the T+H, a widely used hydrate simulator.

- Application of JFTS+H to the solution of 2 large reservoir-scale problems of increasing size, and comparison to results from the T+H simulator.

- Evaluation of the parallel performance of JFTS+H against the JFTS+H serial version.

7.2 Summary and Conclusions

I developed the JFTS+H simulator to significantly accelerate the simulation and analysis of the complex problems of coupled flow, thermal/thermo-dynamic and chemical processes associated with the formation and dissociation of hydrates in geologic media. Because of their complexity

and the persistent steep gradients that do not become progressively more diffusive with time (an important feature of hydrate problems), solution of these problems require a very large number of short timesteps and excessively (and often unacceptably) long execution times when using the widely-used serial T+H code. Parallelization of the hydrate simulations appears to be the obvious solution to the problem, and JFTS+H was designed and developed to address this issue by combining the unique computational features and speed of the Julia language with its built-in parallelization capabilities. I validated the JFTS+H code against the T+H simulator for a wide range of simpler synthetic problems and a large field problem, covering all possible methods of hydrate dissociation (individually and in combination) and kinetic or equilibrium reactions of hydrate dissociation and formation.

The JFTS+H simulator can model both laboratory experiments and field-scale problems of gas production from hydrate deposits and provides estimates of all relevant parameters and variables that describe the evolution over time of (a) hydrate dissociation, (b) gas production and (c) the spatial distributions of all relevant phase saturations and thermophysical properties in the reservoir, all in formats suitable for easy visualization . Analysis and evaluation of the computational performance of the JFTS+H simulator shows conclusively that the MPI-based parallel version of the code has an impressive parallel performance, delivering significant and scalable speed-ups on diverse computational platforms without any code modification, reducing the execution times by orders of magnitude over those needed by the serial T+H code for the solution of the same problems, and making possible the solution of previously intractable problems within reasonable time frames.

REFERENCES

Atkinson, A. B, A Model for the PTX Properties of $H_2O - NaCl$, M.S. Thesis, Virginia Polytechnic Institute and State University, 2002.

Ballard, A.L., A Non-Ideal Hydrate Solid Solution Model for a Multi-Phase Equilibria Program, Ph.D. Thesis, Colorado School of Mines, 2002.

Balay, S., PETSc User's Manual Revision 3.12, Mathematics and Computer Science Division, Argonne National Laboratory, 2019.

Barney, B., Introduction to Parallel Computing Tutorial, Livermore Computing, 2021.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V.B., 2017. Julia: A Fresh Approach to Numerical Computing, Society for Industrial and Applied Mathematics, Vol. 59, No. 1, pp. 65-98.

Boswell, R., E. Myshakin, G.J. Moridis, Y. Konno, T.S. Collett, M.T. Reagan, T. Ajayi and Y. Seol, India National Gas Hydrate Program Expedition 02 Summary of Scientific Results: Numerical Simulation of Reservoir Response to Depressurization, Journal of Marine and Petroleum Geology, In press, 2019 (doi: 10.1016/j.marpetgeo.2018.09.026).

Computer Modeling Group (CMG) Ltd, 2007. CMG STARS. Calgary, Alberta, Canada.

Corey, A.T., The Interrelation Between Gas and Oil Relative Permeabilities, Producers Monthly, 38-41, November 1954.

Cuyt, A., Computational Implementation of the Multivariate Halley Method for Solving Nonlinear Systems of Equation, ACM Transactions on Mathematical Software, 1985.

Demirbas, A., Methane Gas Hydrate, Green Energy and Technology 161-171, 2010.

Driesner, T and Heinrich, C. A, The system $H_2O - NaCl$. Part I; Correlation formulae for phase relations in temperature-pressure-composition space from 0 to 1000 °C 0 to 5000 bar, and 0 to 1 X_{NaCl}, Isotope Geochemistry and Mineral Resources, ETH Zentrum NO, CH-8092 Zurich, Switzerland, 2004.

Driesner, T, The system $H_2O - NaCl$. Part II: Correlations for molar volume, enthalpy, and isobaric heat capacity from 0 to 1000 °C 1 to 5000 bar, and 0 to 1 X_{NaCl}, Isotope Geochemistry and Mineral Resources, ETH Zentrum NO, CH-8092 Zurich, Switzerland, 2006.

Fukusako, S., Thermophysical Properties of Ice, Snow, and Sea Ice, International Journal of Thermophysics, Vol. 11, No. 2, 1990.

Gropp, W., Lusk, E., and Skjellum, A., using MPI, Third Edition, Scientific and Engineering Computation Series, 2014.

Gupta, A., Methane Hydrate Dissociation Measurements and Modeling: The Role of Heat Transfer and Reaction Kinetics, Ph.D. Thesis, Colorado School of Mines, 2007.

Hass, J. L, Physical Properties of the Coexisting Phases and Thermochemical Properties of the H_2O Component in Boiling NACl Solutions, Bulletin-1421A, 1976.

Innes, M., Edelman, A., Fischer, K., Rackauckas, C., Saba, E., Shah, V. E., Tebbutt, W., A Differentiable Programming System to Bridge Machine Learning and Scientific Computing, Cornell University, 2019.

International Association for the Properties of water and Steam (IAPWS). Revised Release on the IAWPS Industrial Formulation 1997 for the Thermodynamic Properties and Steam, Lucerne, Switzerland, 2007.

International Association for the Properties of water and Steam (IAPWS). Release on the IAWPS Industrial Formulation 2008 for the Thermodynamic Properties of Seawater, Berlin, Germany, 2008.

International Association for the Properties of water and Steam (IAPWS). Revised Release on the Equation of State 2006 for H_2O O Ice lh, Doorwerth, The Netherlands, 2009.

International Association for the Properties of water and Steam (IAPWS). Revised Release on the Pressure along the Melting and Sublimation Curves of Ordinary Water Substance, Plzen, Czech Republic, 2011.

Kamath, V.A., Study of heat transfer characteristic during dissociation of gas hydrates in porous media, Ph.D. dissertation, Univ. of Pittsburg, Pittsburg, PA, 1984.

Kamath, V.A. and S.P. Godbole, Inhibitor-Induced Hydrate Dissociation, J. Pet. Tech. (1987), 39, 1379.

Kanwar, V. and Tomar, S.K., Exponentially fitted variants of Newton's method with quadratic and cubic convergence, International Journal of Computer Mathematics Vol. 86, No. 9, 1603-1611, 2009.

Karypis, G., METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices version 5.1.0, Department of Computer Science & Engineering, University of Minnesota, 2013.

Kim, H.C., P.R. Bishnoi, R.A. Heideman and S.S.H. Rizvi, Kinetics of methane hydrate decomposition, Chem. Eng. Sci., 42, 1645-1653, 1897.

Katz, D., Handbook of Natural Gas Engineering. McGraw-Hill, New York, 2002.

Kowalsky, M.B., and G.J. Moridis, Comparison of Kinetic and Equilibrium Reaction Models in Simulating the Behavior of Gas Hydrates in Porous Media, Journal of Energy Conversion and Management, 48(6), 1850-1863, 2007 (doi: 10.1016/j.enconman.2007.01.017)

Kurihara, M., Funatsu, K., Ouchi, H., Masuda Y., and Narita, H., 2005. Investigation On Applicability Of Methane Hydrate Production Methods To Reservoirs With Diverse Characteristics, Paper 3003 presented at the 5th International Conference on Gas Hydrates, Trondheim, Norway, 13–16 June, Proceedings, Vol. 3, 714-725.

Kurihara, M., Sato, A., Ouchi, H., Natira, H., Masuda, Y., Saeki, T., and Fujii, T., Prediction of Gas Productivity from Eastern Nankai Trough Methane-Hydrate Resrvoirs, paper OTC-19382, 2008 Offshore Technology Conference, Houston, Texas, 5-8 may 2008.

Kvenvolden, K.A., Methane hydrate-a major reservoir of carbon in the shallow geosphere. Chem. Geol. 71, 41-51, 1988.

Makogon, Y.F., Hydrates of Hydrocarbons. Penn Well Publishing Co. Tulsa, OK, 1997.

Goddipati, M., Anderson, B.J., 3D Reservoir Modeling of Depressurization-Induced Gas Production from Gas Hydrate Reservoirs at the Walker Ridge Site, Northern Gulf of Mexico, OTC-23582, 2012 Offshore Technology Conference, Houston, Texas, 30 April-3 May 2012.

Masuda, Y., Fujinaga, Y., Naganawa, S., Fujita, K., Sato, K., and Hayashi, Y., 1999. Modelling and experimental studies on dissociation of methane gas hydrates in Berea sandstone cores; presented at 3rd International Conference on Gas Hydrates, July 18-22, Salt Lake City, Utah. Unpublished manuscript. The University of Tokyo, Department of Geosystem Engineering, Tokyo, Japan

Masuda, Y., Kurihara, M., Ohuchi, H., and Sato T., 2002. A field-scale simulation study on gas productivity of formations containing gas hydrates; Proceedings, 4th International Conference on Gas Hydrates, May 19–23, Yokohama, Japan. Keio University, Yokohama, Japan. P. 40-46.

Masuda, Y., Naganawa, S., Ando, S., and Sato, K., 1997. Numerical calculation of gas-production performance from reservoirs containing natural gas hydrates; paper SPE 38291, Proceedings, Western Regional Meeting, June 25-27, Long Beach, California. Society of Petroleum Engineers Inc., Richardson TX.

Mao, S. and Duan, Z., The Viscosity of Aqueous Alkali-Chloride Solutions up to 623 K, 1000 bar, and High Ionic Strength, Int J Thermophys, 2009.

Moridis, G.J., Numerical studies of gas production from methane hydrate, SPEJ, 32(8) 359-370, 2003.

Moridis, G.J., and K. Pruess, User's Manual of the TOUGH+ v1.5 Core Code: A General-Purpose Simulator of Non-Isothermal Flow and Transport Through Porous and Fractured Media, Lawrence Berkeley National Laboratory Report LBNL-6869E, August 2014.

Moridis, G.J, User's Manual of the Hydrate v1.5 option of TOUGH+ 1.5: A Code for the Simulation of System Behavior in Hydrate-Bearing Geologic Media, Lawrence Berkeley National Laboratory Report LBNL-6869E, August 2014.

Moridis, G.J., and M.T. Reagan, Strategies for Gas Production from Oceanic Class 3 Hydrate Accumulations, paper OTC-18865, 2007 Offshore Technology Conference, Houston, Texas, 30 April – 3 May 2007; doi 10.4043/25384-MS

Moridis, G.J., T.S. Collett, R. Boswell, M. Kurihara, M.T. Reagan, C. Koh and E.D. Sloan, Toward Production from Gas Hydrates: Current Status, Assessment of Resources, and Simulation-Based Evaluation of Technology and Potential, SPE Reservoir Evaluation & Engineering, 12(5): 745-771, 2009; doi: 10.2118/114163-PA).

Moridis, G.J, Queiruga, A. F., and Reagan, M. T, Simulation of Gas Production from Multilayered Hydrate-Bearing with Fully Coupled Flow, Thermal, Chemical and Geomechanical Processes Using TOUGH+ Millstone. Part 1: Numerical Modeling of Hydrates, Transport in Porous Media, 128(2), 405-430, June 2019a (doi: 10.1007/s11242-019-01254-6).

Moridis, G.J., M.T. Reagan, A.F. Queiruga and R. Boswell, Evaluation of the Performance of the Oceanic Hydrate Accumulation at Site NGHP-02-09 in the Krishna-Godavari Basin During a Production Test and During Single and Multi-Well Production Scenarios, Journal of Marine and Petroleum Geology, 108, 660-696, 2019b (doi: 10.1016/j.marpetgeo.2018.12.001).

Moridis G., M.T. Reagan and A.F. Queiruga, Gas Hydrate Production Testing: Design Process and Modeling Results, OTC-29432-MS, 2019 Offshore Technology Conference, 6–9 May, 2019c, Houston, Texas (doi:10.4043/29432-MS)

Nash, J. 2019. Profiling tool wins and woes. The Julia Programming Language.

Nishida, A., Experience in developing an open-source scalable software infrastructure in Japan. In: Computational Science and Its Applications-ICCSA 2010. Lecture Notes in Computer Science, Vol. 6017, 2010.

Parker J.C., Lenhard R.J. and T. Kuppusamy, A Parametric Model for Constitutive Properties Governing Multiphase Flow in Porous Media, Water Resour. Res., 23(4), 618- 624, 1987.

Peng, D.Y., and D.B. Robinson. A New Two-Constant Equation of State, Indust. And Engr. Chemistry: Fundamentals 15, 59-64, 1976.

Phale, H.A., Zhu, T., White, M.D., McGrail, B.P., Simulation Study on Injection of CO2-Microemulsion for Methane Recovery From Gas-Hydrate Reservoirs, paper SPE-100541, 2006 SPE Gas Technology Symposium, Calgary, Alberta, 15-17 May, 2006.

Perkel, J.M. 2019. Julia: come for the syntax, stay for the speed, Nature 572, 141-142 (2019). https://doi.org/10.1038/d41586-019-02310-3.

Reagan, M. T, Moridis, G. J., Freeman, C.M., Boyle, K.L, Keen, N.D., Massively Parallel Simulation of Production from Oceanic Gas Hydrate Deposits, International Petroleum Technology Conference, 26-28 March, Beijing, China, 2013; doi: 10.2523/IPTC-17026-MS

Reagan, M. T, Queiruga, A. T, and Moridis, G.J, Simulation of Gas Production from Multilayered Hydrate-Bearing Media with Fully Coupled Flow, Thermal, Chemical and Geomechanics Processes Using TOUGH+Millstone. Part 3: Production Results, Transport in Porous Media, 2019.

Revels, J., Lubin, M., and Papamarkou, T., Forward-Mode Automatic Differentiation in Julia, Cornell University, 2016.

Peng, D.Y., and D.B. Robinson. A New Two-Constant Equation of State, Indust. And Engr. Chemistry: Fundamentals 15, 59-64, 1976.

Schnetter, E. and Wilcox, L.C. 2021. JuliaParallel/MPI.jl, https://github.com/JuliaParallel/MPI.jl

Sloan, E.D., Clathrate Hydrates of Natural Gases, Marcel Dekker, Inc.: New York, NY, 1998.

Saad, Y., ILUT: A Dual Threshold Incomplete LU Factorization, University of Minnesota Computer Science Department, 1994.

Sakamoto, Y., Kaneko, F., Nakno, Y., Nakamura, K., and Komai, T., Numerical Study on Enhanced Gas Recovery from Methane Hydrate Reservoir During In-situ Heating Process by Acid Injection, International Journal of Offshore and Polar Engineers Vol. 29, No. 3, September 2019, pp. 347-358; https://doi.org/10.17736/ijope.2019.jc744

Stoer, J. and R. Bulirsch, Introduction to Numerical Analysis (3rd ed.), 2002. Berlin, New York, Springer-Verlag. ISBN 978-0-387-95452-3.

Stone, H.L., Probability Model for Estimating Three-Phase Relative Permeability, Trans. SPE of AIME, 249, 214-218, 1970.

Sun, X. and Mohanty, K.K., Simulation of Methane Hydrate Reservoirs, Society of Petroleum Engineering Inc., 2005.

Van der Vorst, H.A., Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, SIAM J. Sci. Stat. Comput., 13(2): 631–644, 1992 (doi:10.1137/0913035. hdl:10338.dmlcz/104566

Van Genuchten, M., A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci. Soc. 44, 892-898, 1980.

White, M.D., STOMP-HYDE-KE A Numerical Simulator for the Production of Natural Gas Hydrate using Guest Molecule Exchange with CO_2 and N_2 Revision 1.0, PNNL-22130, Pacific Northwest National Laboratory, 2012.

White, M.D., Lee, W.L., Guest Molecule Exchange Kinetic for the 2012 lgnik Sikumi Gas Hydrate Field Trial, paper OTC-25374-MS, 2014 Offshore Technology Conference, Houston, Texas, 5-8 May 2014.

Yamamoto, K., Wang, X.-X., Tamaki, M., Suzuki, K., The second offshore production of methane hydrate in the Nankai Trough and gas production behavior from a heterogeneous methane hydrate reservoir, Royal Society of Chemistry, RSC Adv., 2019, 9, 25987-26013, 2019.

Yuan, Y., Xu, T., Xia, Y., Xin, X., Comparison of Simplistic and Geologically Descriptive Production Modeling for Natural-Gas Hydrate by Depressurization, SPE Journal, 563-578, 2019.