Fanxiang Xu

A Local Time-Stepping Method for Multiphase Flow in Porous Media.

TU Delft
Delft University of Technology

A Local Time-Stepping Method for Multiphase Flow in Porous Media

By

Fanxiang Xu

in partial fulfilment of the requirements for the degree of

Master of Science
in Applied Earth Sciences

at the Delft University of Technology,
to be defended publicly on Friday July 27, 2018 at 10:00 AM.

Principle Advisor:	Dr. H. Hajibeygi	TU Delft
Co-Supervisor:	Ir. M. Cusini	TU Delft
Thesis committee:	Prof. dr. W. Rossen	TU Delft
	Dr. A. Dieudonné	TU Delft

TU Delft

Abstract

Simulation of multiphase flow in natural subsurface formations include selection of time-step size, i.e., the discrete snapshots (steps) over which the time-dependent nonlinear process is investigated. The simulation results would naturally depend on the size of the discrete time step, according to the order of accuracy of the implemented numerical scheme. For accurate analyses the required time steps might be needed to be very small, which then leads to very costly simulations, which are often times impractical in real-world simulation applications. Taking coarse-scale time-step sizes can reduce the computational costs, however, it results in lower simulation accuracy. Note that a common simulation practice is to take constant time-step everywhere in the computational domain. To resolve this challenge, this MSc thesis report presents a novel local time-stepping (LTS) method for multiphase flow in porous media. In this method, different time-steps can be applied in one computational domain; allowing for employing small time steps in sub-domains with higher sensitivity to the time step size and bigger time steps elsewhere. The LTS method is developed for all types of sequentially-coupled and fully-coupled simulation strategies. For all of these developments, the interfacial flux continuity is the key requirement to connect the sub-domains of different time zones. The fluxes chosen for the inner stages of the coarse time-step are consistent with the chosen simulation strategy. More precisely, in the case of the implicit-pressure-explicit-saturation (IMPES) method these fluxes are lagged in time (explicit), while for the sequential implicit and fully implicit (FIM) methods they are calculated at the current (implicit) time-step. For implicit transport method, the refined zone partition matrices are developed to algebraically form the refined zone Jacobian matrix and refined zone residual vector. The numerical results show that only a fraction of the domain can be simulated with small time steps; while the rest can be simulated at a bigger one. Local time-stepping method provides accurate simulation results compared with the fine-scale time-step simulation results for all three simulation strategies. Specially, the local time-stepping for FIM simulations, which is the mostly common commercial-grade simulation approach, can preserve the accuracy of the results and also the computational efficiency. For the studied cases, the computational efficiency gain of LTS for sequential implicit method is not as significant as that of the FIM. To exploit full adaptivity in simulation, as presented in Appendix B, one can directly apply the developed method of this thesis to adaptive implicit method (AIM); where sequential and fully implicit coupling approaches are applied in different sub-domains of the reservoir. Overall, LTS casts a promising approach to optimize accuracy-efficiency tradeoff when it comes to time step selection.

Acknowledgements

During this nine months project I gained the deeper understanding of the reservoir simulation for multiphase flow in porous media. Also I have learned a lot about the multi-rates simulation and adaptive simulation and the good strategies like the CPR method to help convergence of simulator.

Firstly I would express my grateful thanks to Professor Hadi Hajibeygi. As my main supervisor he guided me with many useful tips and provided me with many paper resource. My PhD supervisor Matteo Cusini gave me so many useful tips for the problems I faced in the simulation even when he was very busy at his own PhD work at that time. I would like to express my special thanks to Ludovica Delpopolo from Politecnico di Milano in Italy. She helped me a lot in solving the well conditions problems and convergence problems. Finally, I want to thank my parents Yadong Xu and Chongde Zhou for their unconditional support at any time.

Contents

1. Introduction 1

2. Discretized Model and Simulation Strategies 3

2.1 Governing equation and discretization using finite-volume method 3

2.2 IMPES method 5

2.3 Sequential Implicit-Method 6

2.4 Fully Implicit Method 8

3. Local Time-stepping Strategy 10

3.1 Flux connection between different time zones 10

3.2 Local Time-Stepping for IMPES 12

3.2.1 Governing Equation 12

3.2.2 Refined Time Zones Selection 12

3.3.3 Dynamic determination of refined zone 13

3.3 Local Time-stepping for Sequential Implicit 15

3.3.1 Governing Equation 15

3.3.2 Predictor-Corrector Strategy 15

3.3.3 Refined zone partition matrices 16

3.4 Local Time-stepping for Fully Implicit 17

3.4.1 Governing Equation 17

4. Numerical Results 19

4.1 Numerical results for IMPES 20

4.1.1 1D test case 20

4.1.2 2D test case 22

4.2 Numerical results for Sequential Implicit Method 24

4.2.1 1D test case 24

4.2.2 2D test case 27

4.3 Numerical Results for FIM 29

4.3.1 1D test case 29

4.3.2 2D test case 32

5. Conclusions and Future Work Recommendation 34

References 35

Appendix A. Algorithm 37

A.1. Local time-stepping in IMPES 37

A.2. Local time-stepping in Sequential Implicit 38

A.3. Local time-stepping in FIM 39

Appendix B. Adaptive Implicit Method 40

B.1. Flux between explicit and implicit zone 40

B.2. Selection of implicit zone 40

B.3. Algorithm 41

B.4. AIM numerical results 42

Appendix C. LTS in Sequential Implicit on SPE-10 Top Layer 45

1. Introduction

Reservoir simulation, an often-employed strategy in oil field development projects, depends on the discretization of the governing equation describing the multiphase flow in porous model. The spatial and time scales of the discretized model are two important factors that affect the accuracy and computational efficiency of the simulation. Multi-scale reservoir simulation is the strategy that can simulate the reservoir behavior at a coarser scale while keeping track of the results at the fine scale [1]. This strategy provides higher computational efficiency of simulation and keeps the accuracy of the final results. Currently, multi-scale reservoir simulation refers to the multi-scale in space (or grid size), e.g., the multi-level dynamic FIM multiscale for multiphase flow in fractured reservoirs [2,3]. In fact, similar multi-scale strategies could also be applied to the time discretization since the timestep size also directly influences the CPU time required for the reservoir simulation. Larger time-step means less total number of time-steps, and consequently more efficient simulation CPU time.

The local time-stepping (LTS) strategy enables different time-step sizes in the same reservoir model domain [4]. The method splits the time discretization in coarse and refined time zones. In the coarse time zone, most of the grid cells can advance in time using larger time-steps. In the refined time zone, smaller time-steps are taken to obtain the required accurate information. The main issue involved in this strategy is how to deal with the fluxes between coarse and refined zones and how to synchronize both zones.

In [4], flux continuity strategy to solve the time discretization in a multi-scale fashion was presented. The flux continuity enforces mass balance between refined and coarse time zones for a given coarse time-step. To interpolate the fluxes at the inner stage of one coarse time-step the authors use the interpolated variable values to compute the fluxes at the inner stage. They suggest using the piecewise linear interpolation strategy to find the inner stages variable values. They successfully apply the method to solve a linear partial differential equation and analyze its stability criterions and convergence properties. In [5], authors successfully combined the flux continuity requirement with the finite volume discretization strategy, which, by design, is mass conservative. In their paper they apply a predictor-corrector strategy to determine the zones that need to be refined. It is shown that this local time-stepping strategy can significantly reduce the CPU time.

Different multi-stage computation methods applied to multiple time-step simulation can be found in the literature as multi-rates methods. In [6] authors proposed the multi-rate method for hyperbolic problems using the Runge-Kutta method. In this work the cells are separated into fast and slow solution cells and then the second order Runge-Kutta method is used on both group of cells. This method can decouple the cells that require different time-stepping strategies, however guaranteeing synchronization. In [7] the author applied the TR-BDF2 method to perform the multi-rate computation and still use a linear interpolation to find the variable values at the inner stages of one coarse time-step. TR-BDF2, or trapezoidal rule backward-differentiation-formula method, is one-step, second order algorithm with L-stability property [8]. In [7] the automatic time-step refinement is done by evaluating an error estimator, which is an efficient way to detect the zones that need refinement. More recently, in [9] the flux refinement is suggested as a new strategy to determine the local time-stepping and again an error estimator is used to decide where the fluxes should be recomputed under the smaller time-steps. In this work they employ a linear interpolation to compute the inter-zone fluxes.

The adaptive implicit (AIM) method is another alternative to multi-scale-in-time reservoir simulation, which was firstly proposed in [10]. While FIM can guarantee stability, it requires the solution of a large linear system, which can be computationally expensive. In AIM, on the other hand, only the variables that change most rapidly in the space/time are treated implicitly. In [10], the Jacobian matrix is constructed with all the explicit and implicit variables. The drawback of this strategy is that the explicit cells are still approached by the implicit way since they were also put into the Jacobian matrix, which may lead to an expensive solution of the linear system.

In [11], it is proposed that the Courant-Friedrich-Levy (CFL) condition should be used to determine whether an unknown is treated explicitly or implicitly. The sparsity of the system matrices that arises from (FIM), (AIM), Inexact Adaptive Newton (IAN) and the Thomas and Thurnau (TT) methods is discussed. The Jacobian in AIM is simpler than that in FIM since there is no entry for the explicit cells in the AIM Jacobian matrix. Also, the author states that the TT method is not the real adaptive implicit method as the explicit cells were also implicitly simulated. IAN and TT methods may have slower convergence rates when CFL approaches 1, while the AIM's convergence rate is not affected by CFL condition. From the numerical results of [11], AIM can provide the more accurate results than the FIM while using less computation time. In [11] the sequential implicit method is used.

In the present work a new local time-stepping strategy is applied to three reservoir simulation methods: IMPES, sequential implicit and FIM. The remainder of this report is organized as follows. In chapter two, the governing equations and their discretization are discussed, as well as an overview of the simulation methods employed in this work. In chapter three, it is discussed how the local time-stepping strategy is applied to the different reservoir simulation methods described in chapter two. In chapter four, the numerical results obtained from the local time-stepping strategy applied to the IMPES, sequential implicit and FIM methods are discussed. Finally, conclusions drawn from this work and the recommendation of the future work are reported in chapter five.

2. Discretized Model and Simulation Strategies

2.1 Governing equation and discretization using finite-volume method

The hyperbolic mass conservation law for incompressible fluid and rock, after neglecting capillary pressure and gravity effects, can be expressed as

$$\frac{\partial}{\partial t}(\emptyset S_\alpha) - \nabla \cdot (\lambda_\alpha \cdot \nabla P) = q_\alpha, \tag{1}$$

where, S_α is the saturation of phase α for example water, and $\emptyset$ is the rock porosity. P is the pressure of the flow in the reservoir, q_α is the source term (e.g., injection and production wells) of phase α. Furthermore, $\lambda_\alpha$is the mobility of phase α given by

$$\lambda_\alpha = \frac{K \cdot k_{r\alpha}}{\mu_\alpha}, \tag{2}$$

where K is the absolute permeability of the reservoir, and $k_{r\alpha}$ is the relative permeability of phase α and μ_α is the viscosity of the phase α. Here, the Brook-Corey model [12] is used to compute the relative permeability. The water and oil phases relative permeability are respectively given by

$$k_{rw} = k_{rwe} \cdot \left[\frac{(S_w - S_{wc})}{(1 - S_{or} - S_{wc})}\right]^{n_w}, \tag{3}$$

$$k_{ro} = k_{roe} \cdot \left[\frac{(S_o - S_{or})}{(1 - S_{or} - S_{wc})}\right]^{n_o}, \tag{4}$$

where k_{rwe} is the water relative permeability end point, k_{roe} is the oil relative permeability end point, S_{wc} is the connate water saturation, S_{or} is the residual oil saturation, n_w is the water Corey exponent and n_o is the oil Corey exponent.

For the oil and water two-phase flow ($\alpha = w, o$), the flow (pressure) equation can be obtained by adding both mass balance phases' equations (Eq. (1)), leading to

$$-\nabla \cdot (\lambda_t \cdot \nabla P) = q_t, \tag{5}$$

where $\lambda_t = \lambda_o + \lambda_w$ is the total mobility and the $q_t = q_o + q_w$ is the total source term.

From Darcy's law, the total fluid velocity U can be computed as

$$U = -\lambda_t \cdot \nabla P, \tag{6}$$

from which the water saturation can be computed from Eq. (1) as

$$\frac{\partial}{\partial t}(\emptyset S_w) + \frac{\partial}{\partial x}(f_w U) = q_w. \tag{7}$$

In Eq. (7), f_w represents the fractional flow of the water phase, defined as

$$f_w = \frac{\lambda_w}{\lambda_t}. \tag{8}$$

For incompressible 1D reservoir simulation, the discrete flow equation based on Eq. (5) using finite volume scheme [13] reads

$$A \cdot \left[-T_{i+\frac{1}{2}}(P_{i+1} - P_i) + T_{i-\frac{1}{2}}(P_i - P_{i-1})\right] = q'_{t,i} = A\,\Delta x\, q_{t,i}, \tag{9}$$

in which the transmissibility T, for equidistant mesh size of Δx, is defined as

$$T_{i+1/2} = \frac{1}{\Delta x} \cdot \frac{2\lambda_i \lambda_{i+1}}{\lambda_i + \lambda_{i+1}} \tag{10}$$

$$T_{i-1/2} = \frac{1}{\Delta x} \cdot \frac{2\lambda_i \lambda_{i-1}}{\lambda_i + \lambda_{i-1}} \tag{11}$$

Here, $T_{i+1/2}$ refers to the transmissibility at the right boundary of the control volume i, and $T_{i-1/2}$means the transmissibility at the left boundary. The source term $q'_{t,i} = \Delta V \, q_{t,i}$ is in unit of m^3/s, and A is the cross section area that is perpendicular to the flux of the gird block and $A \times \Delta x = \Delta V$.

Eq. (7) can be discretized with equidistant cell of grid size Δx as

$$A \cdot [\emptyset \frac{\Delta x \cdot (S_{w,i}^{n+1} - S_{w,i}^{n})}{\Delta t} + (f_w U)_i{}^R - (f_w U)_i{}^L] = q'_{w,i}. \tag{12}$$

Here, $(f_w U)$ shows the fluxes at the boundaries of the control volumes and n is the time-step index. Using the upwind method, the fluxes can be calculated. The upwind weighting [14] is employed to determine the fluid properties at the control volume boundaries L and R from the property values at the immediate neighboring cells. For the 1D reservoir model setup shown in Fig. 2.1, the A is chosen as 1 m^2. Because the flow direction is known to happen from the injection well to the production well, after applying the upwind strategy, Eq. (12) can be rewritten as

$$A \cdot [\emptyset \frac{\Delta x \cdot (S_{w,i}^{n+1} - S_{w,i}^{n})}{\Delta t_i} + f_{w,i} U_{i+1} - f_{w,i-1} U_i] = q'_{w,i}. \tag{13}$$

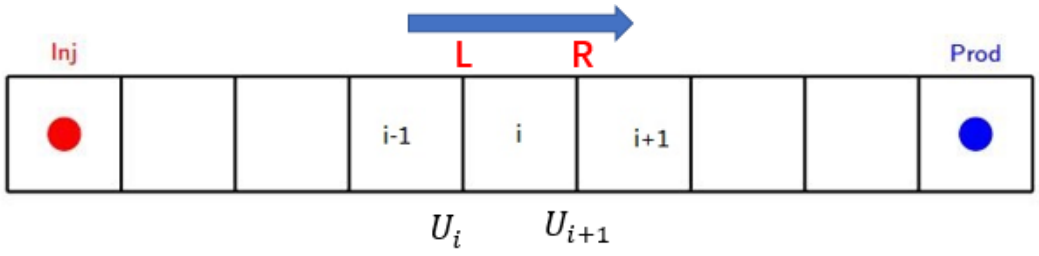


Figure 2.1. 1D reservoir equidistant cells model

The 2D discretization of Eq. (7) for equidistant grid cells with size Δx in the *x* direction and Δy in the *y* direction ($\Delta z = 1$) leads to

$$\emptyset \frac{\Delta x \cdot \Delta y \cdot (S_{w,i}^{n+1} - S_{w,i}^{n})}{\Delta t} + \Delta y \cdot [(f_w U)_i{}^E - (f_w U)_i{}^W] + \Delta x \cdot [(f_w U)_i{}^N - (f_w U)_i{}^S] = q'_{w,i}, \tag{14}$$

Similarly, the fluxes at the different boundaries are calculated using the upwind strategy. For the homogeneous 2D reservoir setup shown Fig. 2.2, because again the flux direction is known, Eq. (14) can be rewritten as

$$\emptyset \frac{\Delta x \cdot \Delta y \cdot (S_{w,i}^{n+1} - S_{w,i}^{n})}{\Delta t} + \Delta y \cdot (f_{i,j} U_{i+1,j} - f_{i-1,j} U_i) + \Delta x \cdot (f_{i,j} U_{i,j+1} - f_{i,j-1} U_{i,j}) = q'_{w,i} \tag{15}$$

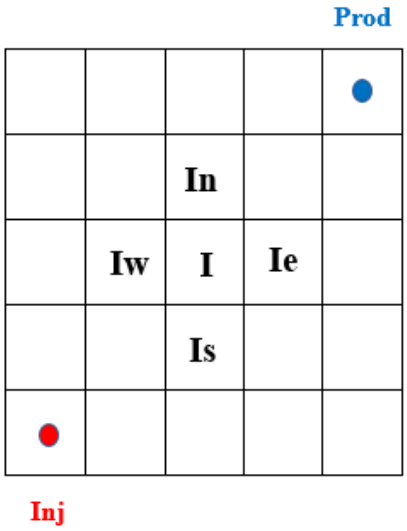


Figure 2.2. 2D reservoir model.

Note that, so far, there is no assumption in which time-step the fluid properties are evaluated. This is going to be addressed next, when the different coupling strategies are discussed.

2.2 IMPES method

The IMPES method has been widely used to solve the flow and transport equations when the coupling between them is not very strong (e.g., in the absence of capillarity and compositional effects). For the reservoir shown in Fig. 2.1, Eq. (13) can be rewritten as

$$A \cdot [\emptyset \frac{\Delta x \cdot (S_{w,i}^{n+1} - S_{w,i}^{n})}{\Delta t} + \left(f_{w,i} U_{i+1}\right)^{n} - \left(f_{w,i-1} U_{i}\right)^{n}] = {q'}_{w,i}{}^{n}. \quad (16)$$

Note that the water saturation-dependent fluid properties at the boundaries and source terms are evaluated lagged in time (explicit). In order to avoid instability, a stable time-step Δt can be computed from the CFL condition as

$$\Delta t_{safe} = CFL \cdot \left(\frac{\Delta x}{\frac{U}{\emptyset} \frac{\partial f}{\partial S}_{max}} \right), \quad (17)$$

where $\frac{\partial f}{\partial S}_{max}$ is the maximum value of the fractional function derivative with respect to the saturation. In addition, 0 < CFL < 1 is an input parameter that can be used to control the time-step size within the stable range. Fig. 2.3 illustrates $\frac{\partial f}{\partial S}_{max}$ and $\frac{\partial f}{\partial S}_{shock}$. Note that the $\frac{\partial f}{\partial S}_{max} > \frac{\partial f}{\partial S}_{shock}$ from Fig. 2.3.

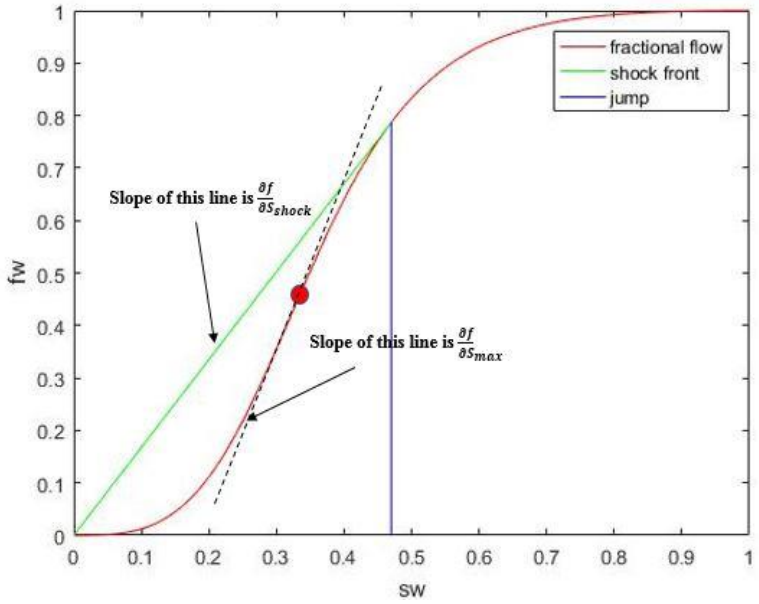


Figure 2.3. Fractional flow curve (red curve). Also shown are $\frac{\partial f}{\partial S_{shock}}$ and $\frac{\partial f}{\partial S_{max}}$.

2.3 Sequential Implicit-Method

In sequential implicit method both flow and transport equations are solved implicitly, in a sequentially iterative procedure (i.e., first pressure then saturation), with fluid properties being evaluated at the current (implicit) time-step. For the 1D reservoir model shown in Fig. 2.1, the discretized transport Eq. (13) reads

$$A \cdot [\emptyset \frac{\Delta x \cdot (S_{w,i}^{n+1} - S_{w,i}^{n})}{\Delta t} + \left(f_{w,i} U_{i+1}\right)^{n+1} - \left(f_{w,i-1} U_{i}\right)^{n+1}] = {q'}_{w,i}^{\,n+1}. \tag{18}$$

Flux at time n+1 is still unknown so a non-linear iterative solution strategy (e.g. Newton–Raphson) method is needed. This is done by iteratively solving Eq. (18) starting from an initial guess. The basic idea of Newton's method is that the a converged solution at time n+1 is achieved when the residual

$$R_w^{n+1} = A \cdot [\emptyset \frac{\Delta x \cdot \left(S_{w,i}^{n+1} - S_{w,i}^{n}\right)}{\Delta t} + \left(f_{w,i} U_{i+1}\right)^{n+1} - \left(f_{w,i-1} U_{i}\right)^{n+1}] - {q'}_{w,i}^{\,n+1} \tag{19}$$

approaches 0. Newton's method requires the linearization of Eq. (18) that can be written as

$$R_w^{n+1} \approx R_w^{v} + \frac{\partial\, {R_w}^{v}}{\partial\, S_w} \cdot \left(S_w^{v+1} - S_w^{v}\right) = R_w^{v} + \frac{\partial\, {R_w}^{v}}{\partial\, S_w} \cdot \delta S_w^{v+1}, \tag{20}$$

where v and $v + 1$ represent, respectively, the previous and the current iteration in the non-linear loop. Thus, the linearized system can be written as

$$J^{v} \cdot \delta S_w^{v+1} = -R_w^{v} \tag{21}$$

where J is the so-called Jacobian matrix and the R_w is the residual vector. Eq. (21) is solved, then the Jacobian matrix and residual vector are updated and the procedure is repeated until convergence to a user-defined tolerance is reached. At convergence, the nonlinear solution to Eq. (18) is found.

Enhanced stability (as a result of implicit time integration) is the main advantage of implicit strategies however the iterative process can be very time consuming. Sometimes the Newton method may face to convergence difficulties (i.e., iterations do not reduce the residual to the desired tolerance norms). In these cases, improved-

iterative strategies such as the Appleyard chopping [15] can be applied. The modified Appleyard chopping from [15] is used in the present work, and is shown in Fig. 2.4.

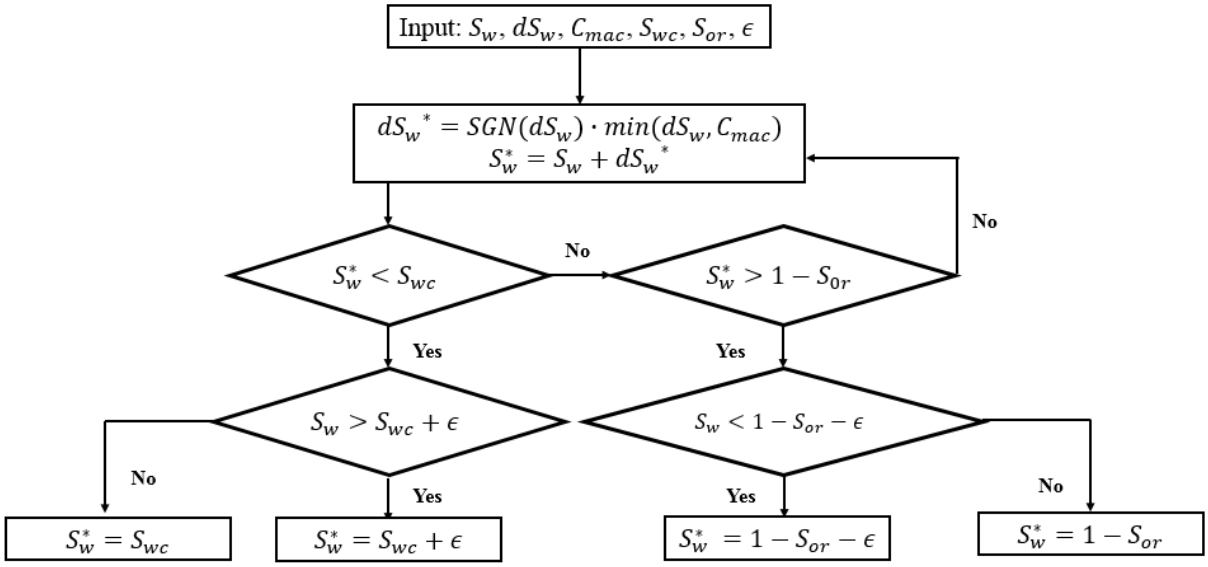


Figure 2.4. Modified Appleyard chopping algorithm.

Another convergence challenge can arise due to the S-shaped fractional flow curve (i.e., inflection point). It may happen that the updated saturation values 'jump' repeatedly from one side of the inflection point to the other. As shown in Fig. 2.5, the updated saturation values jump repeatedly from B to A and back to B. In [16] the authors proposed a solution for this convergence issue. The main idea of the method is to check if the second-order derivatives on the fractional flow curve evaluated at the solutions v and $v + 1$ have opposite signs. This indicates whether the Newton updates cross the inflection point. If so, to over damp the updates, one has to choose the average of both solutions as the update at $v + 1$. The overall idea is of the method is sketched in Fig. 2.6.

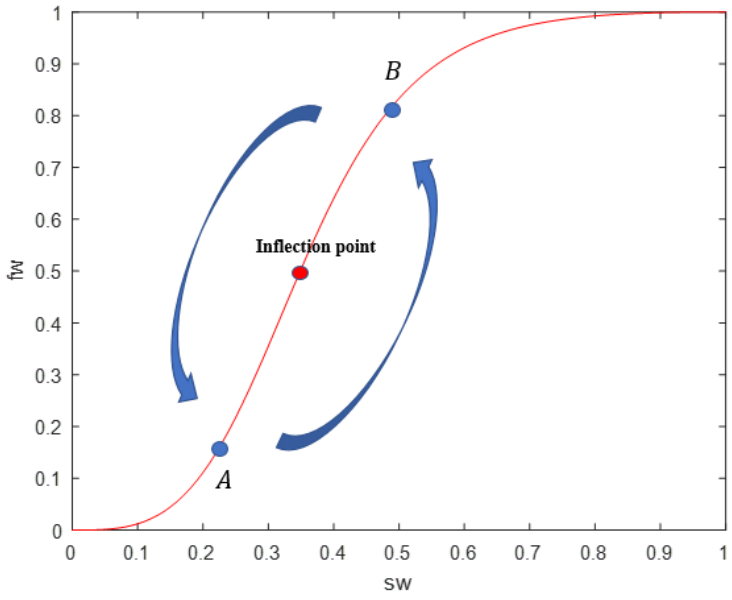


Figure 2.5. Update of saturation values 'jump' around the inflection point in S-shape fractional flow

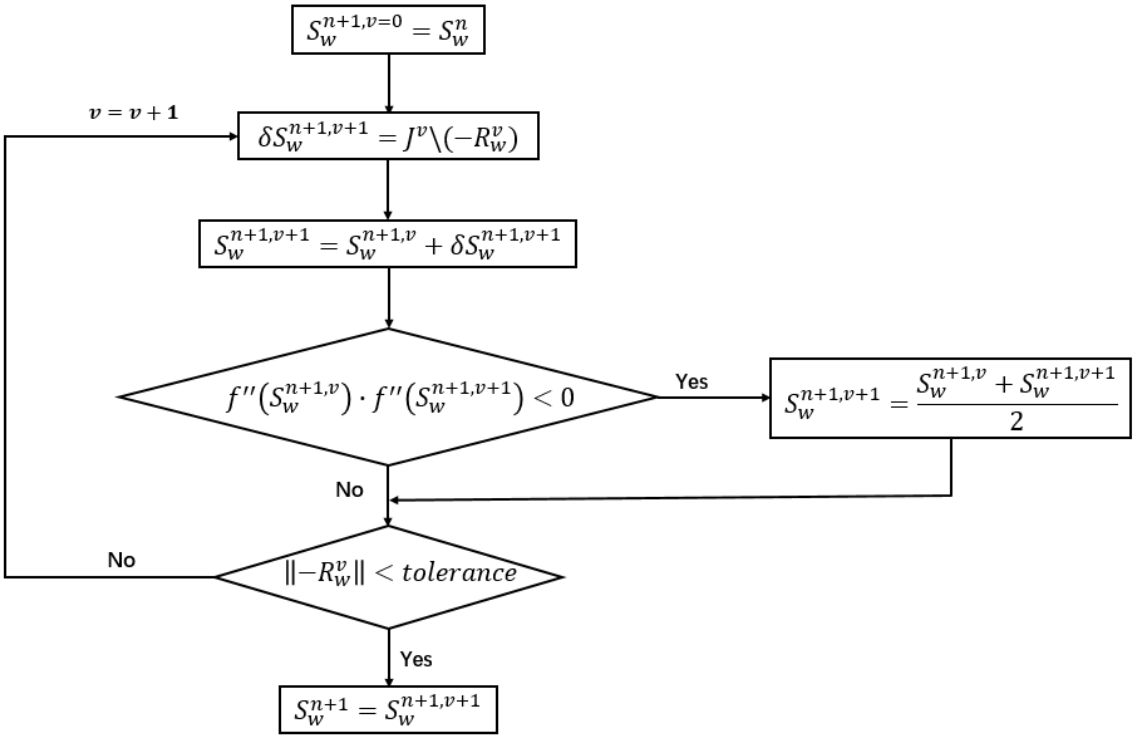


Figure 2.6. Update of saturation values 'jump' around the inflection point in S-shape fractional flow

2.4 Fully Implicit Method

The fully implicit (FIM) method is the industry-standard method of choice, due to its stability, for reservoir simulation when the coupling between pressure and saturation is strong, e.g. in the presence of capillary pressure and compositional effects. Still, for the 1D reservoir model shown in Fig. 2.1 the discretization of the fully implicit method for water phase and for equidistant cell with grid size Δx reads

$$A \cdot [\emptyset \frac{\Delta x \cdot (S_{\alpha,i}^{n+1} - S_{\alpha,i}^{n})}{\Delta t} - \left[(\lambda_\alpha \cdot \nabla P)_i{}^R - (\lambda_\alpha \cdot \nabla P)_i{}^L\right]^{n+1}] = q'_{\alpha,i}{}^{n+1}. \quad (22)$$

Eq. (22) can be rewritten for cell i as

$$\begin{aligned} &A \cdot [\phi \frac{\Delta x (S_{\alpha,i}^{n+1} - S_{\alpha,i}^{n})}{\Delta t} - \\ &\left[TK_{i+1/2} \cdot k_{\mu,\alpha}(i) \cdot (P_{i+1} - P_i) - TK_{i-1/2} \cdot k_{\mu,\alpha}(i-1) \cdot (P_i - P_{i-1})\right]^{n+1}] \\ &= q'_{\alpha,i}{}^{n+1} \end{aligned} \quad (23)$$

where TK is the transmissibility defined by the harmonic average of the absolute permeability between two neighboring cells as

$$TK_{i+1/2} = \frac{2 \cdot K(i) \cdot K(i+1)}{K(i) + K(i+1)} \cdot \frac{1}{\Delta x} \quad (24)$$

$$TK_{i-1/2} = \frac{2 \cdot K(i) \cdot K(i-1)}{K(i) + K(i-1)} \cdot \frac{1}{\Delta x} \quad (25)$$

$$k_{\mu,\alpha} = \frac{k_{r\alpha}}{\mu_\alpha} \tag{26}$$

Eq. (23) can be re-written in residual form as

$$R_\alpha^{n+1}(i) = A\cdot[\phi\frac{\Delta x(S_{\alpha,i}^{n+1} - S_{\alpha,i}^{n})}{\Delta t} - \left[TK_{i+1/2}\cdot k_{\mu,\alpha}(i)\cdot(P_{i+1}-P_i) - TK_{i-1/2}\cdot k_{\mu,\alpha}(i-1)\cdot(P_i - P_{i-1})\right]^{n+1}] - q'^{\,n+1}_{\alpha,i} \tag{27}$$

Similarly to what was done in the sequential implicit method, the linearized form of Eq. (27) reads

$$R_\alpha^{n+1} \approx R_\alpha^{v+1} = R_\alpha^{v} + \frac{\partial R_\alpha}{\partial S_\alpha}^{v}\cdot(S_\alpha^{v+1} - S_\alpha^{v}) + \frac{\partial R}{\partial P}^{v}\cdot(P^{v+1} - P^{v}) \tag{28}$$

The linear system that needs to be solved for a give iteration v in FIM method can be written as

$$J^v \cdot \begin{bmatrix}\delta P^{v+1}\\ \delta S_w^{v+1}\end{bmatrix} = -\begin{bmatrix}R_w^v\\ R_o^v\end{bmatrix}, \tag{29}$$

where the Jacobian matrix for the FIM method can be written as

$$J^v = \begin{bmatrix}J_{w,P}^v & J_{w,S_w}^v\\ J_{o,P}^v & J_{o,S_w}^v\end{bmatrix}. \tag{30}$$

Therefore, in the FIM method pressure and saturation are updated at the same time. However, FIM can be time-consuming due to the large number of iterations for the larger linear system (compared to sequential implicit) that needs to be solve at every iteration.

Along with the Appleyard chopping, the Constrained-Pressure-Residual (CPR) method can be used in the FIM method so that good initial guess for the non-linear solution is obtained from the flow equation solution [17]. By rewriting Eq. (29) as

$$\begin{bmatrix}J_{w,p} & J_{w,S}\\ J_{o,p} & J_{o,S}\end{bmatrix}\cdot\begin{bmatrix}\delta P\\ \delta S_w\end{bmatrix} = \begin{bmatrix}A & B\\ C & D\end{bmatrix}\cdot\begin{bmatrix}\delta P\\ \delta S_w\end{bmatrix} = -\begin{bmatrix}R_w\\ R_o\end{bmatrix}$$

The true IMPES style of the CPR method provides an initial guess to the non-linear solution by recovering the flow equation as

$$(A - B\cdot D^{-1}\cdot C)\cdot\delta P^{1/2} = -(R_w - B\cdot D^{-1}\cdot R_o). \tag{31}$$

Eq. (31) is used as the initial guess to provide an early (prediction) update of the pressure (global) component, followed by the an iterative process on the full system, i.e.,

$$J\cdot\begin{bmatrix}\delta P\\ \delta S_w\end{bmatrix}^1 = -\begin{bmatrix}R_w(\delta P^{1/2})\\ R_o(\delta P^{1/2})\end{bmatrix}.$$

Note that the residual vector is updated based on the prediction of Eq. (31). Significantly This can help the overall convergence properties of the FIM method.

3. Local Time-stepping Strategy

When the different simulation methods were described in chapter 2, constant time-step size (everywhere in the domain) was considered. Usually, to reduce the truncation error resulting from the first order discretization in time and to keep stability of e.g. the IMPES method, small time-steps are used instead of large time-steps. However, this leads to a more computationally expensive simulations. By taking large time-steps on most parts of the model while only taking small time-steps, one can significantly reduce the computational cost, while, very importantly, maintaining the accuracy of the simulation results. Special attention must be paid to sub-regions which are most sensitive (in terms of the quality of the solution) to the time-step size. Typically, this will be regions with sharp fronts.

3.1 Flux connection between different time zones

The local time-stepping (LTS) strategy employs variable time-steps in different parts of the model, namely coarse and refined zones. However, the computation of the fluxes at the interface between the coarse and the refined time zones poses a challenge. The inter-zone fluxes are illustrated in Figure 3.1. Fluxes $F_{\partial\Omega}$ from the coarse time cell i to the refined time cell $i+1$ at time t^n and t^{n+1} are known. $\partial\Omega$ is the boundary between coarse and refined time zones. As showed in Fig. 3.1, there are five stages from $t^{n,0}$ to $t^{n,4}$ that the cell $i+1$ would go through from t^n to t^{n+1} and reach synchronization with the coarse time zone cell. The fluxes at the inner stages $t^{n,1}$, $t^{n,2}$ and $t^{n,3}$ from coarse time-step i are required. However, the fluxes from the coarse time cell at these three inner stages are unknown. So, it is necessary to find a way to interpolate the fluxes at the inner stages between $F_{\partial\Omega}^n$ and $F_{\partial\Omega}^{n+1}$.

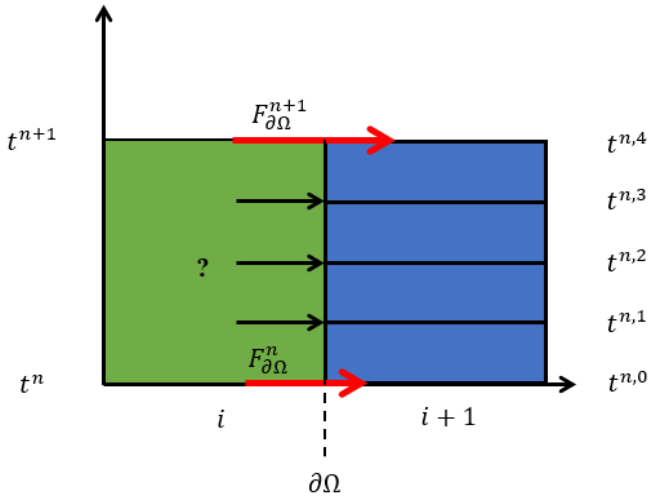


Figure 3.1. Unknown fluxes from coarse time-step cell at inner stages.

To maintain mass conservation, the flux continuity requirement introduced in [2] must be satisfied at $\partial\Omega$, which is expressed as

$$F_{\partial\Omega}^{n\ to\ n+1} \cdot t_c = \sum_1^m (F_{\partial\Omega}^{n,r} \cdot t_r) \quad at\ \partial\Omega, \tag{32}$$

where r is the number of stage and m is the time zone ratio given by

$$m = \frac{t_c}{t_r}, \tag{33}$$

where t_c is the coarse time-step and t_r is the refined time-step or smaller time-step For the IMPES method, Eq. (32) reads

$$F_{\partial\Omega}^{n} \cdot t_c = \sum_1^m (F_{\partial\Omega}^{n,r} \cdot t_r) \quad at\ \partial\Omega. \tag{34}$$

For implicit method, Eq. (32) reads

$$F_{\partial\Omega}^{n+1} \cdot t_c = \sum_1^m (F_{\partial\Omega}^{n,r} \cdot t_r) \quad at\ \partial\Omega. \tag{35}$$

In the present work, upwind weighting is used to determine the fluid properties in the flux computation. Therefore, according to Fig. 3.1 the fluxes $F_{\partial\Omega}^{n,r}$ are determined by the coarse time-step cell i. Thus, for IMPES, at any stage r,

$$F_{\partial\Omega}^{n,r} = F_{\partial\Omega}^{n} \tag{36}$$

then Eq. (34) changes to

$$F_{\partial\Omega}^{n} \cdot t_c = \sum_1^m (F_{\partial\Omega}^{n,r} \cdot t_r) \quad = F_{\partial\Omega}^{n} \cdot t_r \cdot m. \tag{37}$$

Similarly, for implicit transport method if

$$F_{\partial\Omega}^{n,r} = F_{\partial\Omega}^{n+1} \tag{38}$$

then Eq. (35) is rewritten as

$$F_{\partial\Omega}^{n+1} \cdot t_c = \sum_1^m (F_{\partial\Omega}^{n,r} \cdot t_r) \quad = F_{\partial\Omega}^{n+1} \cdot t_r \cdot m. \tag{39}$$

This means that to acquire inner stage fluxes from the coarse time zone to refined time zone, the IMPES method requires the fluxes computed at time n and for the implicit transport solver the fluxes at time $n+1$ should be used (see Figs. 3.2 and 3.3).

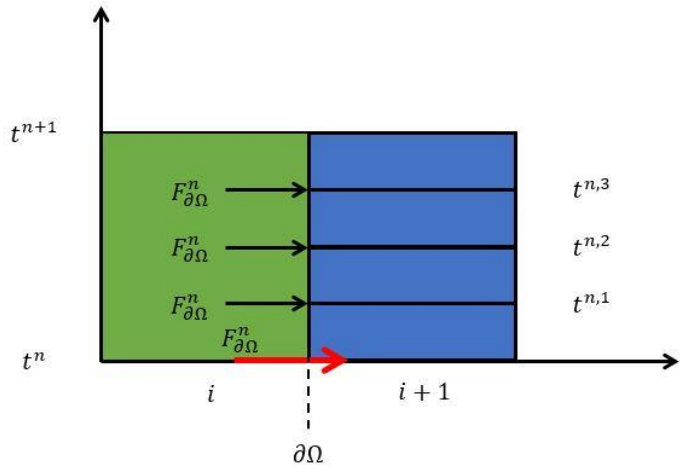


Figure 3.2. Inner stage fluxes for LTS in explicit transport

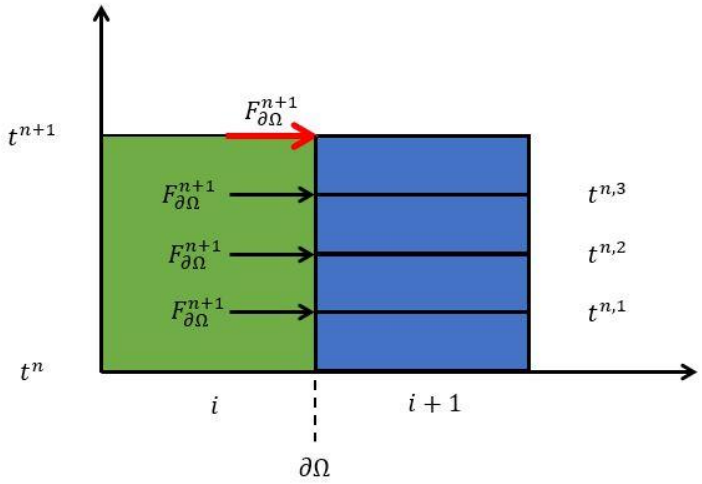


Figure 3.3. Inner stage fluxes for LTS in implicit transport

No special treatment is required for fluxes from the refined to coarse time zones, as the necessary fluxes are readily available. More precisely, as in the LTS strategy the coarse time zone is updated first, before updating the refined zone cells, all the coarse time zone cells are already at t^{n+1}. There is no need to consider this fluxes and update the coarse time zone again.

From now on, for the sake of a convenient presentation, let the cell i in Fig. 3.1 be I_{cr} and the cell $i+1$ be I_{rc}.

3.2 Local Time-Stepping for IMPES

3.2.1 Governing Equation

For IMPES, the flow equation is firstly solved and then the transport equation is solved after the total velocity U is computed and, hence, kept constant. The Local Time-Stepping (LTS) method for IMPES allows for taking different time steps at different location of the domain. This enhances the simulation speedup by avoiding the most restricting cell (based on the CFL stability condition) to limit the globally-selected time step. For IMPES we only need to interpolate the fractional flow to find the inner stage fluxes. Therefore, the transport equation for cell I_{rc} can be written as

$$\emptyset \cdot A \cdot [\frac{\Delta x}{\Delta t} \cdot \left(S_{w,i}^{n,r} - S_{w,i}^{n,r-1}\right) + \left(f_{w,i} U_{i+1}\right)^{n,r-1} - \left(f_{w,i-1} \cdot U_i\right)^{n}] = {q'}_{w,i}^{\,n} \qquad (40)$$
$$\forall r \in \{1, \dots, m\}.$$

After all the inner stages calculation the refined cells will be at the stage t^{n+1} and the synchronization is achieved. Note that, for IMPES, m is usually not an integer because the two time-steps are calculated instead of user input. This would be explained in section 3.2.2. m can be divided into the integer part M and the decimal part M_1. Thus, in the LTS for IMPES, simulation from stage $m-1$ to m takes the time-step size of

$$dt = M_1 \cdot dt_1. \qquad (41)$$

The stage r should change from 1 to $M+1$ as the m here is not the integer number so the stage here should be $M+1$ instead of m.

The same LTS strategy is applied in 2D simulations. The only difference is that it is necessary to check whether the four immediate neighboring cells are coarse time-step cell or refined time-step cell.

3.2.2 Refined Time Zones Selection

The refined time zones selection will be based on the stable time-step limit, i.e., the CFL condition. The safe time-step dt_{safe} as mentioned in Eq. (17) would be used as refined time-step or dt_1. To determine the coarse time-step size, or dt_2, the percentage of how many cells should be involved in the refined time zones should be firstly determined. For the sake of computation efficiency, there should be only a limited percentage of cells inside the refined time zone. This percentage can be a user input parameter.

Stable time limits for each cell would be computed then. Because of the upwind strategy applied in here the stable time limits were computed at the boundaries instead of the cells center. This method can prevent the potential unstable result to the shock front. Note that the stable time needs to be calculated based on the upwind flux values at the interfaces. As shown in Fig. 3.4, dt_{stable}^{L} is the stable time limit computed at the left boundary

$$dt_{stable}^{L} = CFL \cdot \frac{dx}{U_i} \cdot \frac{\partial f}{\partial S_{i-1}} \qquad (42)$$

and dt_{stable}^{R} is the stable time–step size computed at the right boundary as

$$dt_{stable}^{R} = CFL \cdot \frac{dx}{U_{i+1}} \cdot \frac{\partial f}{\partial S_i} \qquad (43)$$

The stable time–step size of cell i is then the minimum of these two values

$$dt_{stable} = \min(dt_{stable}^{L}, dt_{stable}^{R}) \qquad (44)$$

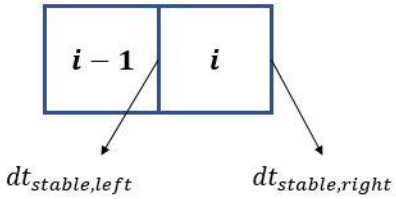


Figure 3.4. Boundary based stable time limits calculation

The cell immediate in front of the shock front would have the stable time limit value equal to the upwind cell's stable time limit to avoid the problem mentioned above. The idea of selecting dt_2 is shown in Fig. 3.5. All non-zero stable time-step sizes can be sorted in ascending order so that dt_2 can be determined based on the user-defined fixed percentage cutoff point.

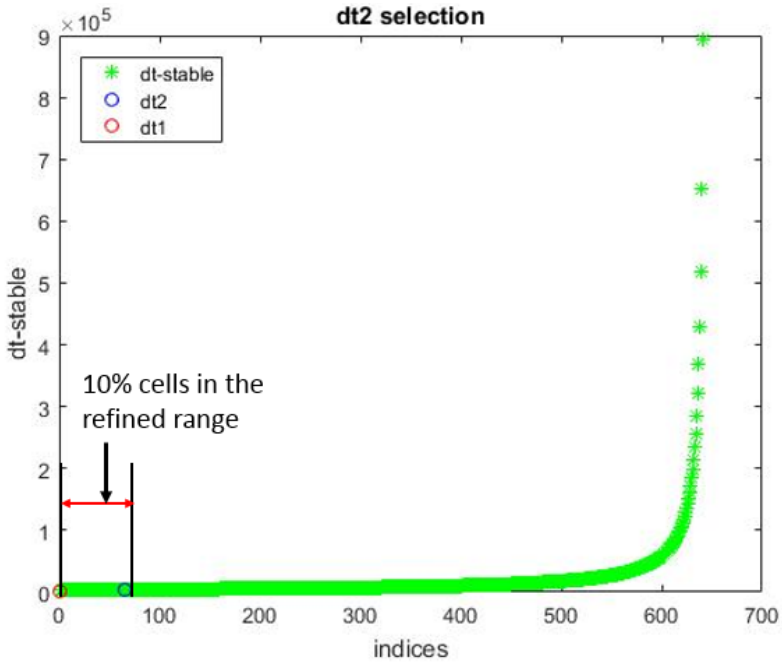


Figure 3.5. Selection of coarse time-step by setting the fixed percentage of refined zone cells

After computing dt_1 and dt_2, the time zones can be defined as follows. If the stable time-step limit is higher than dt_2 then this cell should be in coarse time zone; otherwise, to preserve stability, the cell should be in the refined time zone.

For 2D IMPES simulation, the computation of two time-steps is similar to the 1D method. The stable time limit value is chosen as the minimum one between the four values calculated at the four boundaries. To preserve the stability of the 2D method the percentage of the refined zone cells should be larger, e.g. 15%.

3.3.3 Dynamic determination of refined zone

As stated above, before the transport equation solution, the refined time zone is already set up. It is assumed that the shock will not leave the refined zone. But this cannot be always guaranteed. Also, for some of the cells inside

the refined zone, after a few inner stages calculation, the stable time-step limit for this cell may become higher than dt_2 and there is no need to keep them inside the refined zone anymore.

In order to track the moving shock front, more cells should be added into the refined zone. Otherwise, the shock front may leave the old refined zone. This is illustrated in Fig. 3.6. The tracking can be done by adding one more cells after each inner step calculation. For IMPES, only one more cell at each small time-step is enough since CFL < 1 . From the CFL definition, $CFL = \frac{\Delta t}{\Delta x} \cdot \Lambda$, where Λ is the characteristic velocity. The CFL represents the number of the cells the wave crosses at each Δt. Therefore, for IMPES, CFL < 1 ensures in each time-step the front cannot cross more than one cell. So, in each small time-step, adding one cell into the refined zone is enough to make sure that the refined zone dynamically tracks the moving shock front.

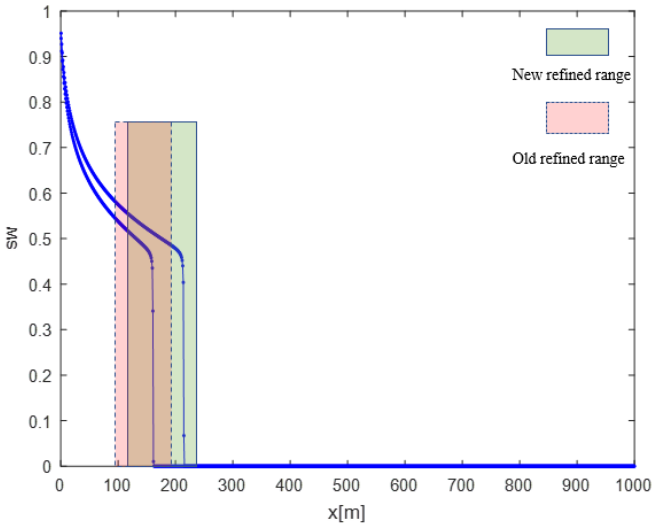


Figure 3.6. The dynamic in refined zone

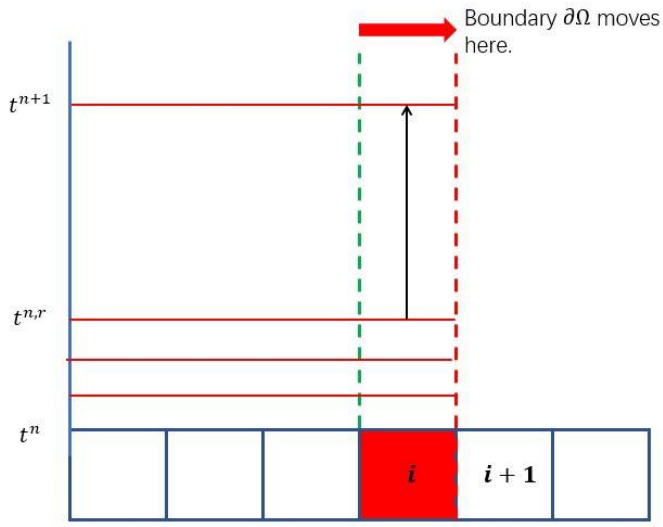


Figure 3.7. Remove cells from refined zone

If cell i was in stage r and its stable time-step size was already higher than dt_2, then the last inner stage calculation of cell i should use the time-step computed as

$$dt = dt_2 - r * dt_1. \quad (45)$$

The idea to remove the cell from refined zone is to let them jump to t^{n+1} directly as shown in Fig. 3.7. After removing the cells from the refined zone, the boundary between the coarse time zone and refined time zone $\partial\Omega$ will move.

In order to facilitate the dynamic refined zone implementation , a time zone partition vector is used. Refined cells are set to 1 at the corresponding partition vector positions, while the remainder positions are set to 0. For example, in a 1D reservoir model of four cells, if cell number two and three are in the refined zone then the partition vector P1 is [0,1,1,0]. If any cell is removed, the corresponding index in the vector changes to 0 and the new added cell position set to 1.

All the aforementioned multi-time-step simulation strategies were implemented on a MATLAB reservoir simulator from scratch (developed by the author). The LTS algorithm for IMPES is can be found in Appendix A.

3.3 Local Time-stepping for Sequential Implicit

3.3.1 Governing Equation

Similar to IMPES, the flow equation is also firstly solved and then the transport equation is solved after computation of total velocity. Thus, likewise, the flux interpolation is simply the fractional flow interpolation. The governing equation of the refined zone cell I_{rc} can be written as

$$A \cdot [\emptyset \frac{\Delta x \cdot \left(S_{w,i}^{n,r} - S_{w,i}^{n,r-1}\right)}{\Delta t} + \left(f_{w,i} U_{i+1}\right)^{n,r} - \left(f_{w,i-1} U_i\right)^{n+1}] = {q'}_{w,i}^{n,r}, \quad (r = 1 \dots m) \tag{46}$$

which can be written in residual form as

$$R_w^{n,r+1} = A \cdot [\emptyset \frac{\Delta x \cdot \left(S_{w,i}^{n,r} - S_{w,i}^{n,r-1}\right)}{\Delta t} + \left(f_{w,i} U_{i+1}\right)^{n,r} - \left(f_{w,i-1} U_i\right)^{n+1}] - {q'}_{w,i}^{n,r} \tag{47}$$

3.3.2 Predictor-Corrector Strategy

The advantage of implicit method is the unconditional stability for any time-step size. So, in the LTS for the sequential implicit method, both coarse and refined time-step sizes can be determined by the user. The refined zone is determined by a predictor-corrector strategy. In the strategy, firstly the simulator is run for a large time-step size from t^n to t^{n+1} for all the reservoir cells. This is called the predictor stage. Then the refined zone is determined based on the simulation results at t^{n+1} and the refined zone would be run again from t^n to t^{n+1} for correction, which is called corrector stage.

The main advantage of this predictor-corrector strategy is that there is no need to determine the refined zone dynamically. In this work, since we are interested in accurately tracking the shock front, the refined zone will be determined by $\frac{\partial f}{\partial S}$. At t^{n+1}, cells where $\frac{\partial f}{\partial S} > \frac{\partial f}{\partial S_{shock}}$ will be considered in the refined zone for correction with smaller time-steps. There may be situations where some cells contain saturation values lower than the shock front saturation value. But in this case $\frac{\partial f}{\partial S}$ is also less than $\frac{\partial f}{\partial S_{shock}}$. So, the detection of the refined zone cells can be made as

$$\frac{\partial f}{\partial S} \geq \frac{\partial f}{\partial S_{shock}} \quad or \; S_w \leq S_{w,shock} \tag{48}$$

Fig. 3.8 illustrates the situation.

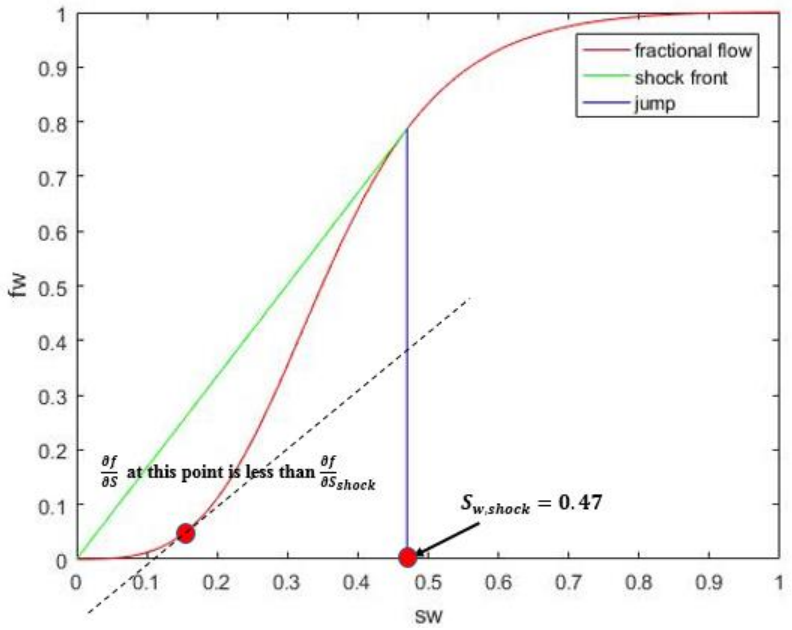


Figure 3.8. Cells with saturation value less than shock front saturation value but with $\frac{\partial f}{\partial S} < \frac{\partial f}{\partial S_{shock}}$

3.3.3 Refined zone partition matrices

In the corrector stage, the linear system of the refined zone cells need to be rebuilt. Saturation values for the refined zone cells are set back to time t^n and only the residuals for the refined zone cells are calculated. There is no need to compute derivatives for the coarse time zones cells inside the refined zone Jacobian since they were already updated to the next coarse time-step. So, one can build a partition matrix that selects only the terms related to the refined zone cells from the Jacobian.

Similar to the IMPES method, a partition vector P1 can be constructed. The generation of the partition matrix will be illustrated via an example. Let's consider the four-cells grid shown in Fig. 3.9.

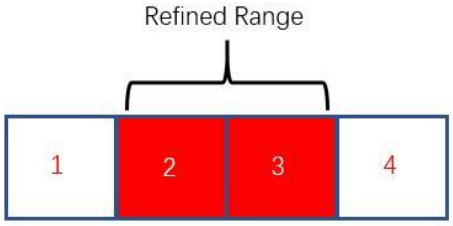


Figure 3.9. Four cells reservoir model with refined zone

The Jacobian matrix is constructed as

$$J = \begin{bmatrix} a11 & 0 & 0 & 0 \\ a21 & a22 & 0 & 0 \\ 0 & a32 & a33 & 0 \\ 0 & 0 & a43 & a44 \end{bmatrix}$$

The partition vector is then

$$\mathrm{P1} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

which can be restated as a diagonal matrix as

$$\mathrm{diag(P1)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Partition matrices are then constructed as follows. A matrix P2 is constructed from the rows of P1 that contain 1's and a matrix P3 is constructed from the columns of P1that contain 1's. So, P2 and P3 reads

$$\mathrm{P2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \mathrm{P3} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Thus the refined Jacobian matrix J_r can be built as

$$J_r = P2 \cdot J \cdot P3 = \begin{bmatrix} a22 & 0 \\ a32 & a33 \end{bmatrix} \tag{49}$$

and the refined residual as

$$R_{w,r} = \mathrm{P2} \cdot R_w \tag{50}$$

This computation is performed iteratively in the refined zone, up until all the refined zone cells reach synchronization with the coarse time zone cells.

The LTS algorithm for the sequential implicit method can be found in Appendix A.

3.4 Local Time-stepping for Fully Implicit

3.4.1 Governing Equation

In the fully implicit method, since pressure and saturation are coupled, fluxes depend on both pressure and saturation at *n*+1. Let's consider the model shown in Fig. 3.10. The cell interface inflow flux acts as the Neumann boundary condition. At the other boundary, where the fluxes cross from a refined time zone to coarse time zone, there is a Dirichlet boundary for pressure.

Figure 3.10. Refined zone mathematics model including Neumann boundary condition and Dirichlet boundary condition

Thus, the governing equation at the cell neighboring the coarse time zones can be written as

$$A \cdot [\emptyset \frac{\Delta x \cdot (S_{\alpha,i}^{n,r} - S_{\alpha,i}^{n,r-1})}{\Delta t} - \left[TK_{i+1/2} \cdot \left(k_{\mu,\alpha}(i) \cdot \left(\frac{P_{i+1} - P_i}{\Delta x}\right)\right)^{n,r} - F_{\partial\Omega}^{n,r}\right]] = (q'_{\alpha,i})^{n,r},$$
$$F_{\partial\Omega}^{n,r} = TK_{i-1/2} \cdot \left[k_{\mu,\alpha}(i-1) \cdot \left(\frac{P_i - P_{i-1}}{\Delta x}\right)\right]^{n+1} \quad (r = 1 \ldots . m) \tag{51}$$

which can be rewritten in residual form as

$$R_\alpha^{n+1} = A \cdot [\emptyset \frac{\Delta x \cdot (S_{\alpha,i}^{n,r} - S_{\alpha,i}^{n,r-1})}{\Delta t} - \left[TK_{i+1/2} \cdot \left[k_{\mu,\alpha}(i) \cdot \left(\frac{P_{i+1} - P_i}{\Delta x}\right)\right]^{n,r} - F_{\partial\Omega}^{n,r}\right]] - (q'_{\alpha,i})^{n,} \tag{52}$$

The same predictor-corrector strategy used in the sequential implicit method is used in the FIM LTS strategy. Also, the refined Jacobian matrix residual vector can be built from the P2 and P3 partition matrices, respectively, as

$$J_r = \begin{bmatrix} P2 \cdot J_{w,p} \cdot P3 & P2 \cdot J_{w,S} \cdot P3 \\ P2 \cdot J_{o,p} \cdot P3 & P2 \cdot J_{o,S} \cdot P3 \end{bmatrix} \tag{53}$$

$$R_r = \begin{bmatrix} P2 \cdot R_w \\ P2 \cdot R_o \end{bmatrix} \tag{54}$$

The algorithm of the LTS in FIM is shown in Appendix A.

4. Numerical Results

In this chapter simulation results of the local time-stepping methods in IMPES, sequential implicit and FIM will be presented. Fine-scale in time simulation results will be used to compare with the local time-stepping results to check their quality. CPU time of local time-stepping and fine scale in time simulation will be compared. The mass conservation of the LTS method is also checked. Note that the CPU times are all based on the developed MATLAB code and presented only as an estimate of the computational efficiency. Only a proper implementation with a compliable programming language (e.g. C++) can demonstrate the realistic speedup of the method.

The general fluids properties are listed in Table 4.1. These properties will be used in the all the test cases.

Table 4.1. List of the fluids properties

μ_w (cp)	μ_o (cp)	k_{roe}	k_{rwe}
1	10	1	1
n_w	n_o	S_{wc}	S_{or}
3	2	0	0

For all the following 1D test cases assuming the reservoir model is homogeneous with $K = 10^{-13}\ m^2$ (or 100 mD). Porosity is 0.2. There is one injection well at the left boundary and one production well at the right boundary. The length of the reservoir is 1000m and the number of grids is 1000.

For the 2D test cases we use a section of the SPE-10 top layer as the test model [18]. The original whole SPE-10 top layer contains 220 by 60 cells data and this section we use is the data included from row number 121 to 160 and column number from 21 to 60. The distribution of the absolute permeability is shown in Fig. 4.1. One should note that there is a high permeability channel in the model. The porosity of the 2D reservoir model is still 0.2. There is one injection well at the north east corner and one production well at the south west corner.

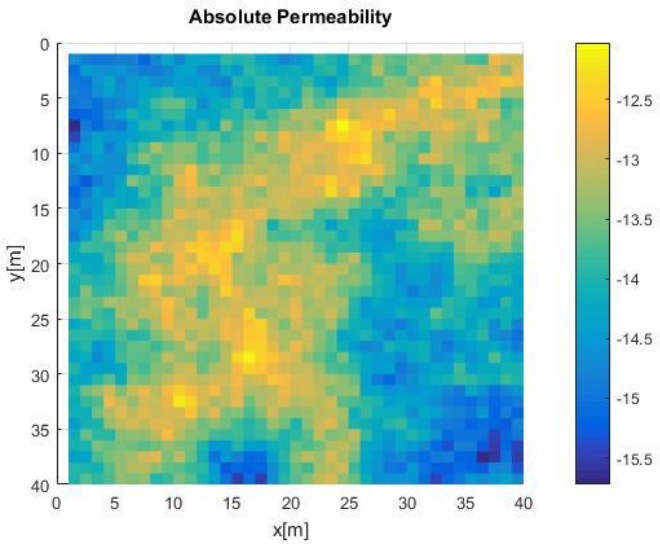


Figure 4.1. Absolute permeability filed of a sector of the SPE-10 top layer. Permeabilities are shown in common logarithm.

4.1 Numerical results for IMPES

4.1.1 1D test case

In this test case the injection well is constant injection rate controlled and the production well is bottom hole pressure controlled. Reservoir properties and well properties are listed in Table 4.2.

Table 4.2. 1D IMPES test case reservoir and well properties

PI($\frac{m^3}{s \cdot Pa}$)	injection rate ($\frac{m^3}{s}$)	initial water saturation	production well pressure (Pa)
1000	3×10^{-5}	0	0

The endpoint that the simulator should stop is when the total amount of water injected becomes larger than 40% of total pore volume. The total amount of the water injected in the LTS method and the fine-scale in time method are kept as same. 10% of the cells are involved into the refined zone. CFL number is 0.95. Coarse-scale time-step is 4106 s/step and fine-scale time-step is 2119 s/step. CPU time for the fine-scale simulation is 26s and for the local time-stepping method is 18s. So the LTS in 1D IMPES is computation efficient.

Simulation results of LTS method and fine-scale time-step method are shown in Fig. 4.2 (a). 'LTS coarse' in the legend refers to the coarse time zone and 'LTS refined' means the refined zone in the local time stepping in the local time stepping simulation. Expectedly, the difference between saturation shock front positions is very small, as seen in Fig. 4.2 (b). The pressure distribution difference between the fine-scale (with smallest stable time step everywhere) and LTS (local time-stepping) is shown in Fig. 4.2 (c) and Fig. 4.2 (d). The pressure difference is calculated as

$$dP = \left(\frac{P_{LTS} - P_{fine-scale}}{P_{fine-scale}}\right) \times 100\%. \quad (55)$$

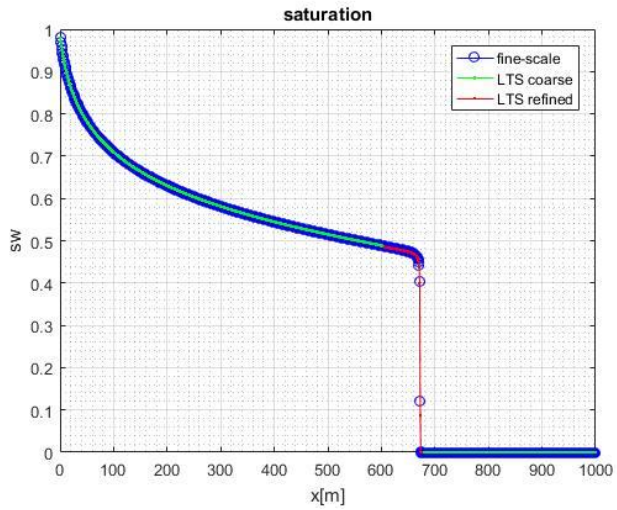
Since the pressure solver is still saturation dependent, from equation (2) to (5), when the saturation simulation results are precise enough, the pressure simulation results are also in very high quality. So the LTS in 1D IMPES results are in very high quality. With less than 10% cells in the refined zone while the rest cells are simulated with coarse-scale time-step the results are still in very high quality compared to the constant fine-scale time-step simulation, which means the local time-stepping in 1D IMPES can generate high quality results at the lower computational costs.

Fig. 4.2 (e) shows the percentage of cells involved in the refined zone in each step. The number of cells in refined zone is growing because the number of refined zone cells is 10% of all the cells with non-zero $\frac{\partial f}{\partial S}$ values. With the shock front moving forward the number of non-zero $\frac{\partial f}{\partial S}$ cells is growing so there would be more cells in the refined zone. Note that the refined zone percentage is less than 10% because the 10% is required on all the non-zero $\frac{\partial f}{\partial S}$ values cells instead of the total number of the cells in model.

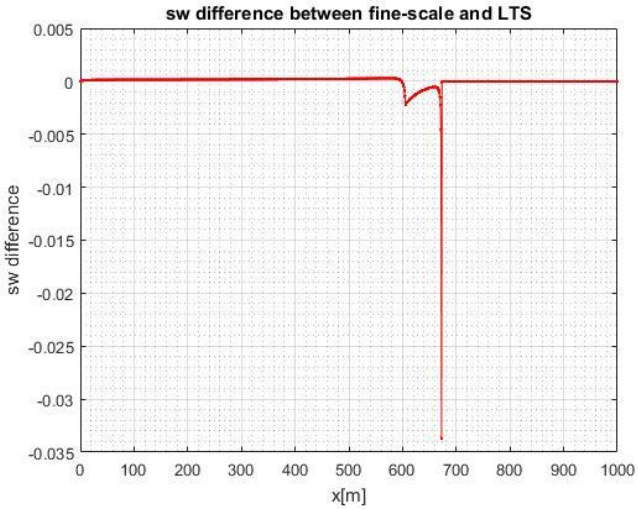
The mass conservation check is plotted as in Fig. 4.2 (f). The mass difference between each time-step is computed as following:

$$dmass\ \% = \left(\frac{mass^{n+1} - mass^{n} - Q \cdot dt}{Q \cdot dt}\right) \times 100\% \quad (56)$$

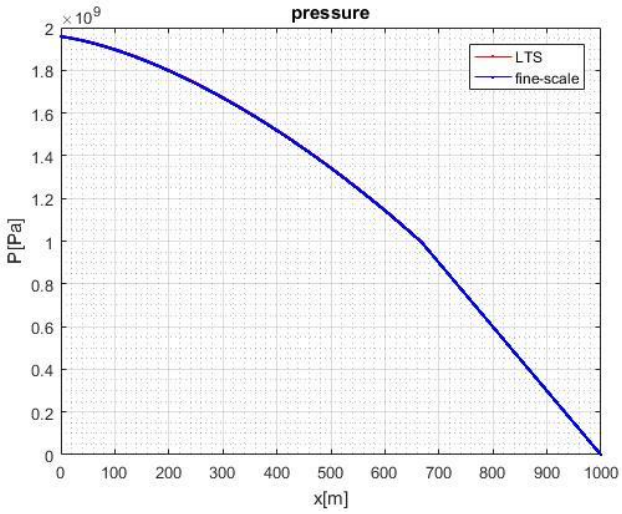
The fine-scale constant time-step simulation is strictly mass conservative. From Fig. 4.2 (f), the mass difference is so small. Thus the local time-stepping in the explicit solver is the strictly mass-conservative and the fluxes continuity at the interface can guarantee the mass-conservative for the 1D explicit solver.



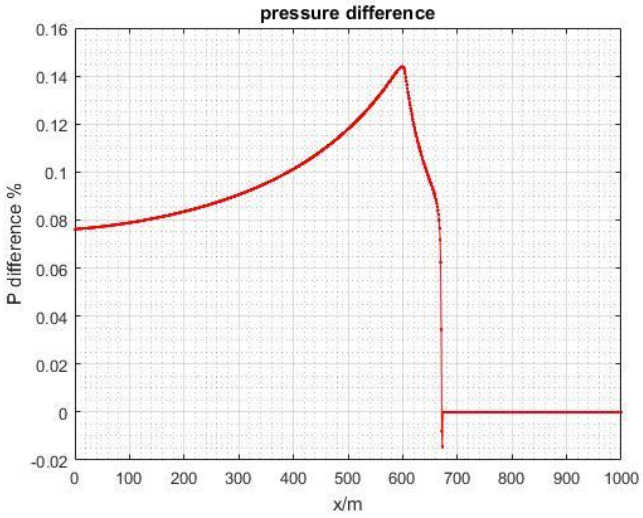
(a) Saturation distribution



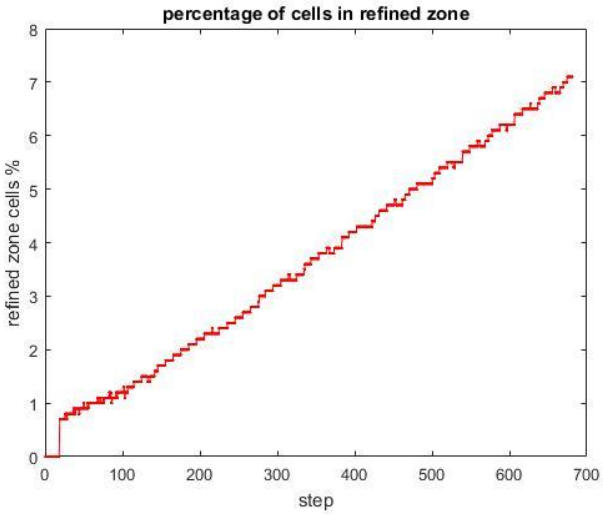
(b) Saturation difference



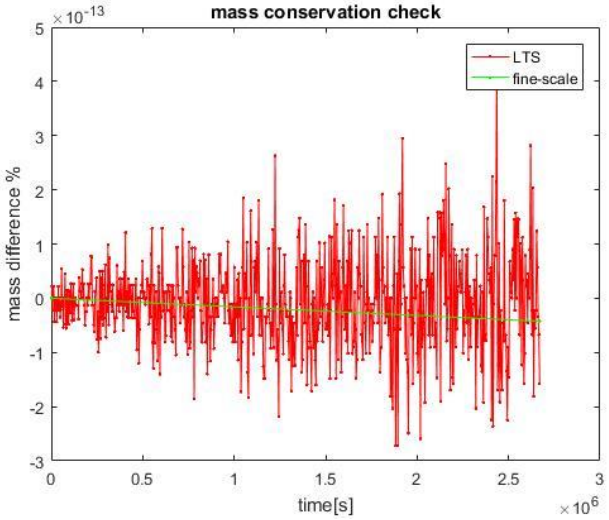
(c) Pressure distribution



(d) Pressure difference



(e) percentage of the cells in refined zone



(f) mass conservation check

Figure 4.2 .1D IMPES numerical results with LTS method and fine-scale in time method. CFL number is 0.95. Coarse time step is 4106 s and the fine-scale time-step is 2119 s. CPU Time for LTS is 18s and for constant fine-scale time-step is 26s.

4.1.2 2D test case

In this test case the injection well is constant injection rate controlled and production well is bottom hole pressure controlled. The well properties and the reservoir properties are shown in the Table 4.3.

Table 4.3. 2D IMPES test case reservoir and well properties

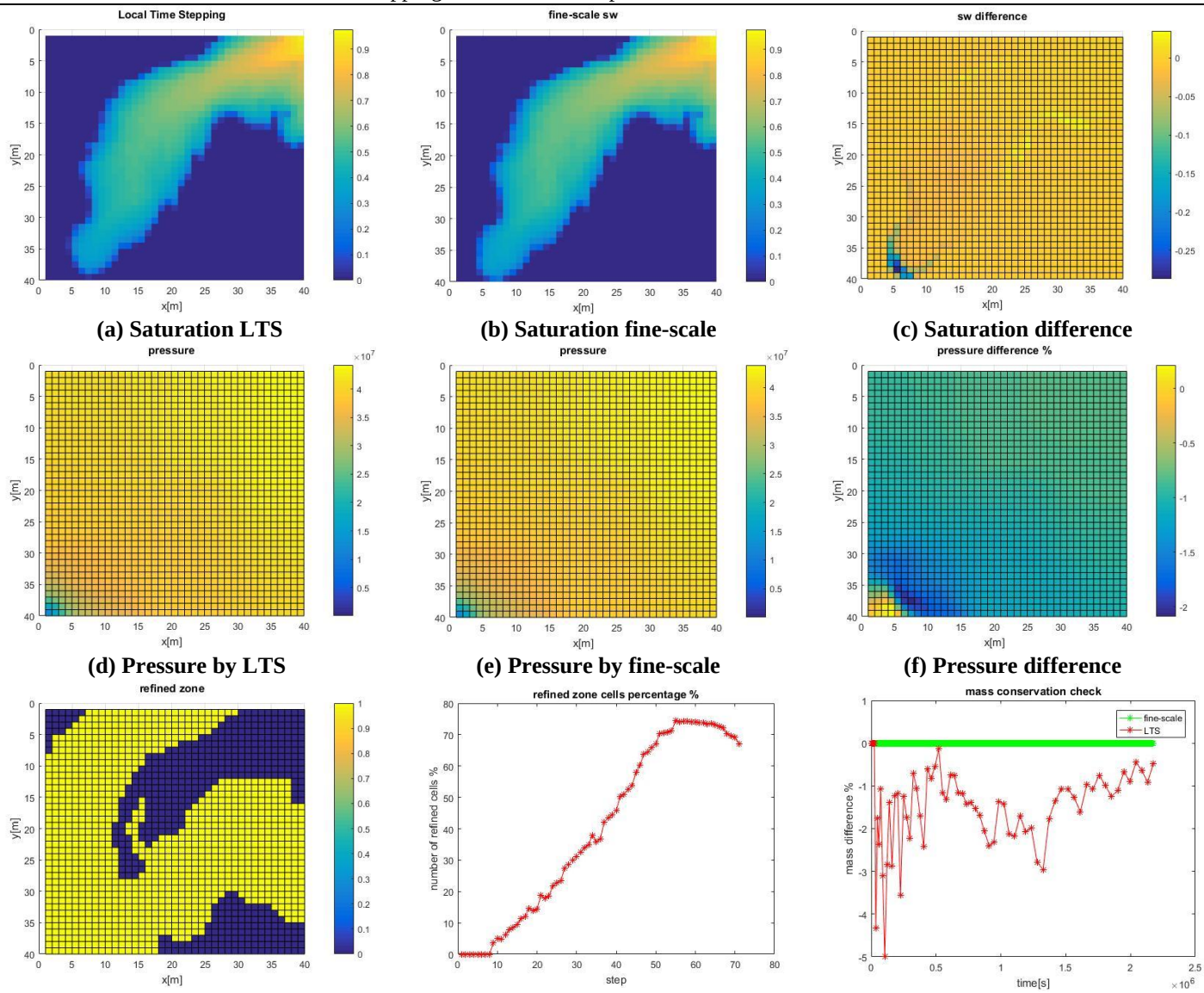
PI($\frac{m^3}{s \cdot Pa}$)	injection rate ($\frac{m^3}{s}$)	initial water saturation	production well pressure (Pa)
1000	3×10^{-5}	0	0

The endpoint that the simulator should stop is when the total amount of water injected becomes larger than 20% of the pore volume and still the amount of water injected for LTS method and fine-scale in time method are kept as same. 15% of the cells are involved into the refined zone. CFL number is 0.9. Coarse-scale time-step is 43600 s/step and fine-scale time-step is 3115 s/step. CPU time for the fine-scale simulation is 47s and for the local time-stepping method is 8s. So LTS in 2D IMPES is computation efficient.

The saturation simulation results are shown in Fig. 4.3 (a) and Fig. 4.3 (b). There is difference between the results in LTS and fine-scale in time simulation at the shock front according to Fig. 4.3 (c). The shock front position of LTS result is behind that of fine-scale in time simulation result, which is consistent with the 1D simulation result. The difference is small and located in the very small area. The pressure simulation results are shown in Fig. 4.3 (d) and Fig. 4.3 (e) and the pressure difference is shown in Fig. 4.3 (f). The maximum pressure difference can be 2% at the shock front position but for the rest of the reservoir the pressure difference is around 1%. So the simulation results of LTS in 2D IMPES are still in good quality.

From Fig. 4.3 (g) and Fig. 4.3 (h), it is clear that large percentage of the cells are involved in refined zone. This is because that the time zone ratio m is really large in the 2D test case. In this test case the m is 13.93. This means that in total there will be 14 more cells added into the refined zone in each coarse time-step simulation. So in fact in the real simulation of LTS in 2D IMPES the percentage of the refines zone is far beyond 15%. At maximum of more than 70% cells are involved in the refined zone. The large amount of cells in the refined zone is not affecting the computation efficiency and give the good simulation result. Note that after step 50 the percentage of the cells involved in the refined zone is decreasing. This is because that the refined zone approaches the boundary of the reservoir and there is no space for refined zone growing.

From Fig. 4.3 (i) the mass difference in each time-step is really large at the initial time, which can be 5% at maximum. At the later time the mass difference in each coarse time step is in average of 1 %This mass conservation check result is not the mass conservative scheme. Considering the time zone ratio difference between the 1D IMPES case ($m = 1.9$) and the 2D IMPES case ($m = 13.9$) it is possibly to conclude that the larger the difference between the two time-steps the mass conservation would be affected more. This may require later research.



(a) Saturation LTS **(b) Saturation fine-scale** **(c) Saturation difference**

(d) Pressure by LTS **(e) Pressure by fine-scale** **(f) Pressure difference**

(g) Refined zone(yellow part) **(h) refined zone cells percentage** **(i) Mass conservation check**

Figure 4.3. 2D IMPES numerical results with LTS method and fine-scale in time method. CFL number is 0.9. Coarse time step is 43600 s and the fine-scale time-step is 3115 s. CPU time for the LTS method is 8s and for the fine-scale in time method is 47s.

4.2 Numerical results for Sequential Implicit Method

4.2.1 1D test case

In this test case the injection well is constant injection rate controlled and production well is bottom hole pressure controlled. Reservoir properties and well properties are listed in Table 4.4.

Table 4.4. 1D sequential implicit method test case reservoir and well properties

PI($\frac{m^3}{s \cdot Pa}$)	injection rate ($\frac{m^3}{s}$)	initial water saturation	production well pressure (Pa)
1000	3×10^{-5}	0	0

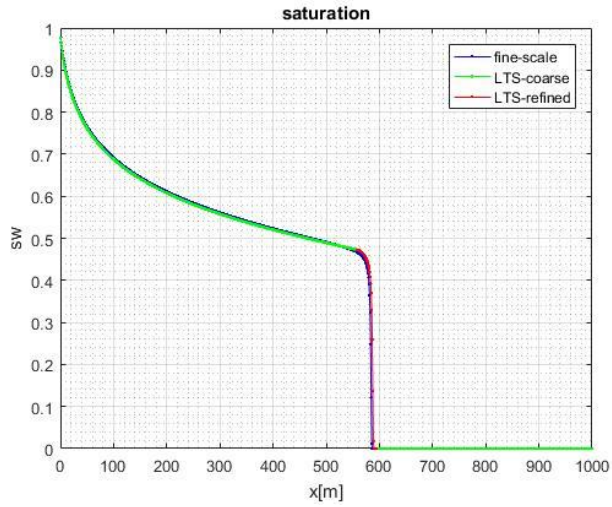
The endpoint of the simulator is when total time for the simulation is 640 hrs. The coarse-scale time-step is set as 8hrs/step and the fine-scale time-step is set as 1hr/step. CPU time for the fine-scale simulation is 43s and for the local time-stepping method is 38s. CFL number at coarse time zone is 13 and at fine-scale zone is 1.6. So the LTS in 1D sequential implicit is computation efficient.

From Fig. 4.4 (a) and Fig. 4.4 (b) the difference between the saturation values is very minor. The shock front position difference is about 5m in length. From Fig. 4.4 (d) the pressure difference is less than 1%. So the simulation result is in very high quality.

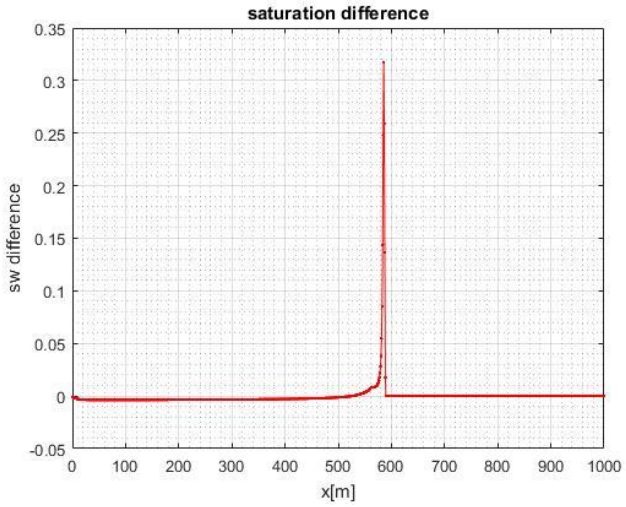
From Fig. 4.4 (e) there are only around 3.5% cells involved in the refined zone at the last time step. The number of refined zone cells is also growing. With the increasing time the shock front of the coarse-scale time-step simulation would smear out and since the refined zone detection is based on the coarse-scale time-step simulation by the predictor-corrector strategy then the more smear out of the shock front the more cells will be included in the refined zone by Eq. (48).

Based on Fig. 4.4 (f), the average of the mass difference is so small thus the LTS in 1D sequential implicit method is mass-conservative. This means fluxes continuity at the different time zone boundaries can make sure the mass-conservative for the 1D implicit transport method.

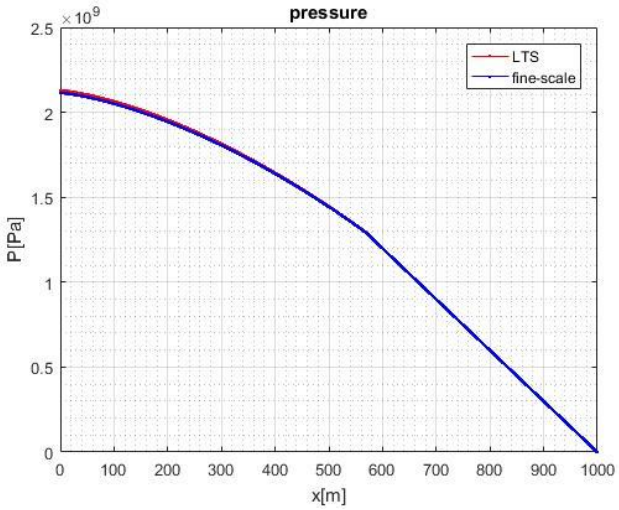
It is also necessary to check if the result of the LTS method is in between the fine-scale time-step simulation result and coarse-scale in time simulation results. These three time scales results are shown in Fig. 4.4 (g). It is clear that the LTS shock front is in between the fine-scale time-step result and the coarse-scale time-step result so the simulation result from the LTS is reasonable.



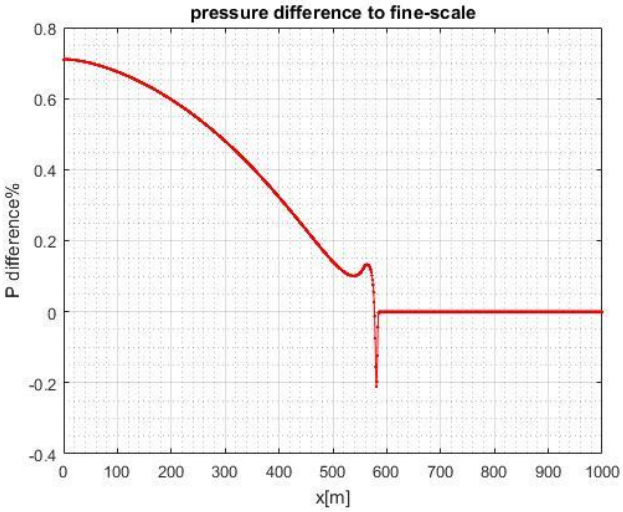
(a) Saturation distribution



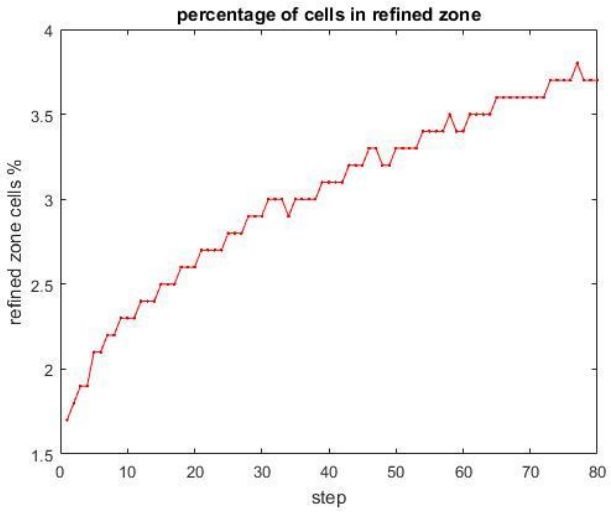
(b) Comparison of saturation to fine-scale results



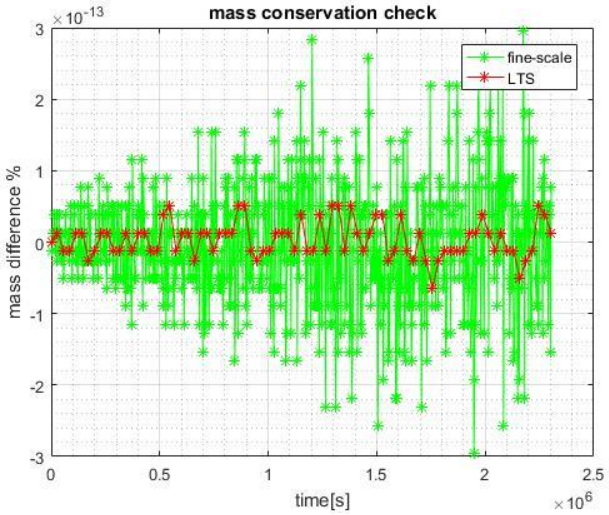
(c) Pressure distribution



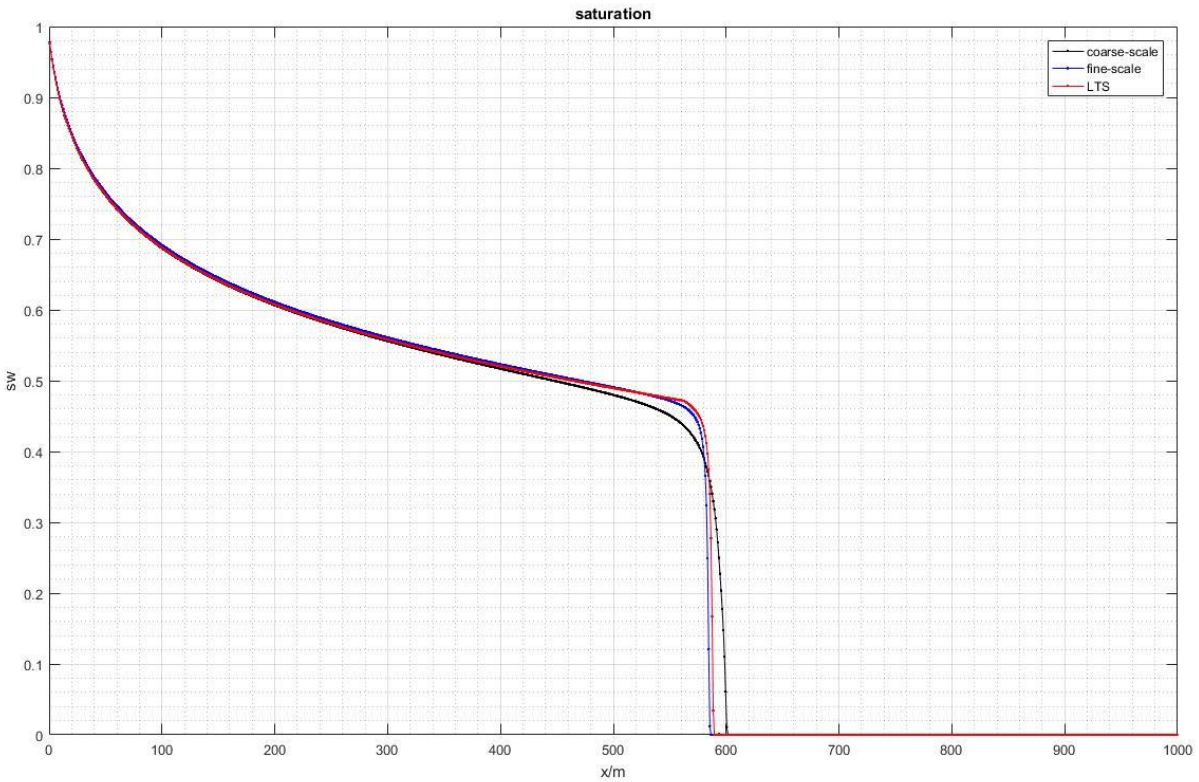
(d) Pressure difference



(e) Refined zone cells number each step



(f) Mass conservation check



(g) Comparison of results between fine-scale in time and coarse-scale in time and LTS

Figure 4.4. 2D sequential implicit numerical results with LTS method and fine-scale in time method. Coarse time step is 8hrs and the fine-scale time-step is 1 hr. Total simulation time is 640 hrs. CPU time for the LTS is 38s and for the fine-scale in time simulation is 43s.

4.2.2 2D test case

In this test case injection well is injection rate controlled and production well is bottom hole pressure controlled. The well properties and the reservoir properties are shown in the Table 4.5.

Table 4.5. 2D sequential implicit test case reservoir and well properties

PI($\frac{m^3}{s \cdot Pa}$)	injection rate ($\frac{m^3}{s}$)	initial water saturation	production well pressure (Pa)
1000	3×10^{-5}	0	0

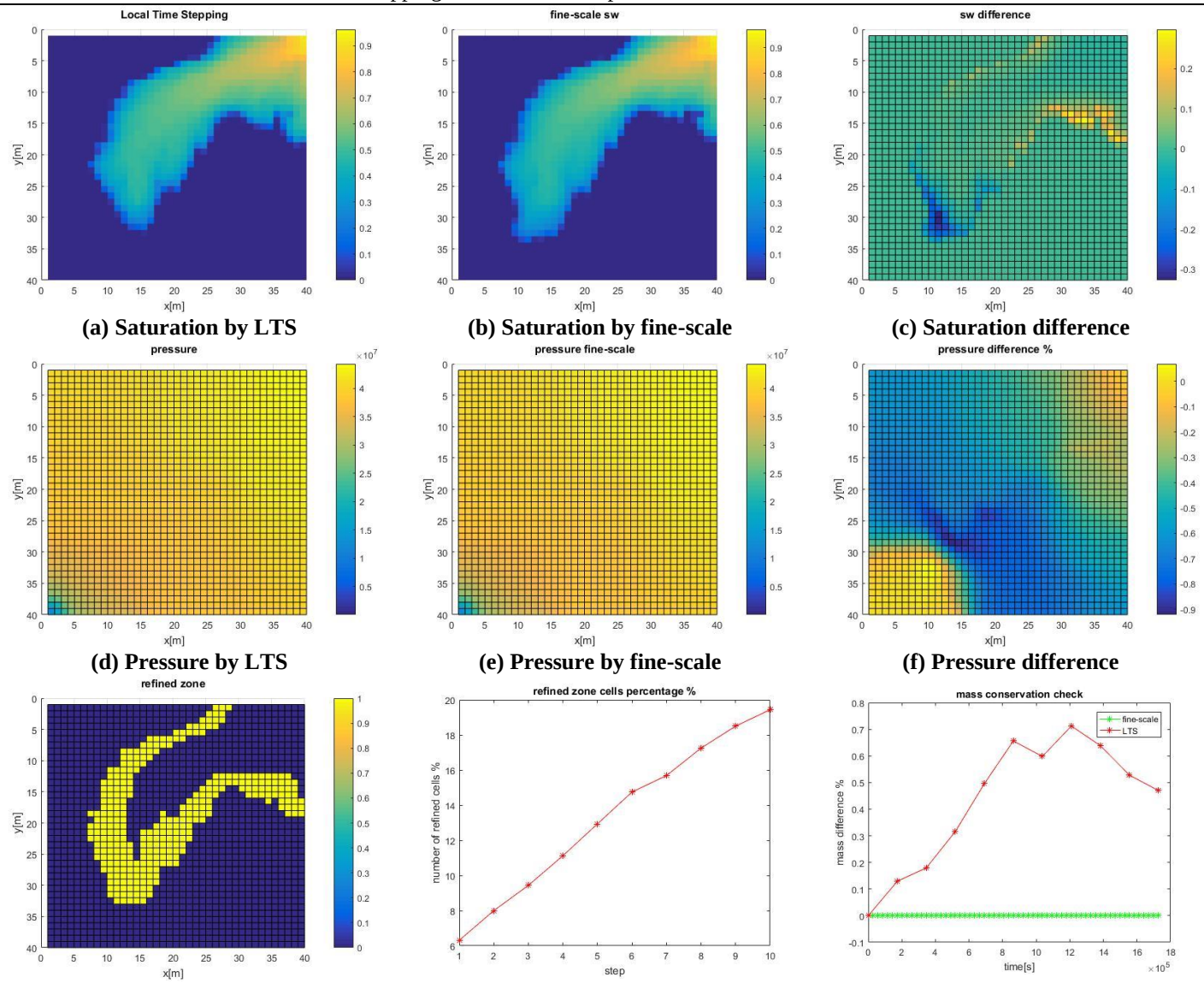
The endpoint of the simulator is when total time for the simulator is 20 days. The coarse time-step is set as 2 days/step and the fine-scale time-step is set as 8 hrs/step. CFL number for the coarse time zone is 9.4 and for the refined time zone is 1.5. The CPU time for local time-stepping method is 15.7s and for fine-scale method is 9.6s. This means that in the 2D reservoir model there is no computation efficiency for LTS in sequential implicit method.

From 1D to 2D reservoir the shape and the number of the cells involved in the refined zone or shock front becomes larger. In the final step there are about 20% cells involved into the refined zone simulation from Fig. 4.5 (h). The refined zone Jacobian matrix is then 20% of the size compared to full reservoir Jacobian matrix. For the 1D test case shown in Fig. 4.4 (e) the refined zone Jacobian matrix is only around 3% in size of the full reservoir Jacobian matrix. For sequential implicit method the Jacobian matrix only contains the saturation derivatives so the size of the Jacobian matrix affects the CPU time on iterative computation directly. So for 1D test case when the reservoir model is very large there is still computation efficiency kept but for 2D reservoir simulation this efficiency advantage disappears.

When the 2D reservoir model is increasing there is still no computation efficiency for LTS in sequential implicit method. In Appendix C the test case run on the whole SPE-10 top layer is shown.

Simulation results for saturation are shown in Fig. 4.5 (a) and Fig. 4.5 (b) and the results for pressure are shown in Fig. 4.5 (d) and 4.5 (e). From Fig. 4.5 (c) the difference between the LTS results and the fine-scale results are mostly located around the shock front region but the difference area is very large and the distance between the results are around 4m. This is consistent with the 1D test case result. Pressure difference is less than 1% from Fig. 4.5 (f). So the simulation results of LTS in 2D sequential implicit method are not in good quality. Refined zone is mostly located along the shock front region from Fig. 4.5 (g)

Fig. 4.5 (i) represents the mass conservation error in each time step and clearly it is not mass conservative.



(a) Saturation by LTS **(b) Saturation by fine-scale** **(c) Saturation difference**

(d) Pressure by LTS **(e) Pressure by fine-scale** **(f) Pressure difference**

(g) Refined zone(yellow part) **(h) Refined zone cells percentage** **(i) Mass conservation check**

Figure 4.5 .2D sequential implicit numerical results with LTS method and fine-scale in time method. Coarse time step is 2 days and the fine-scale time-step is 8 hrs. Total simulation time is 20 days. CPU time for the LTS method is 15.7s and for the fine-scale in time is 9.6s.

4.3 Numerical Results for FIM

4.3.1 1D test case

In this test case injection well is constant injection rate controlled and production well is bottom pressure controlled. Reservoir properties and well properties are listed in Table 4.6.

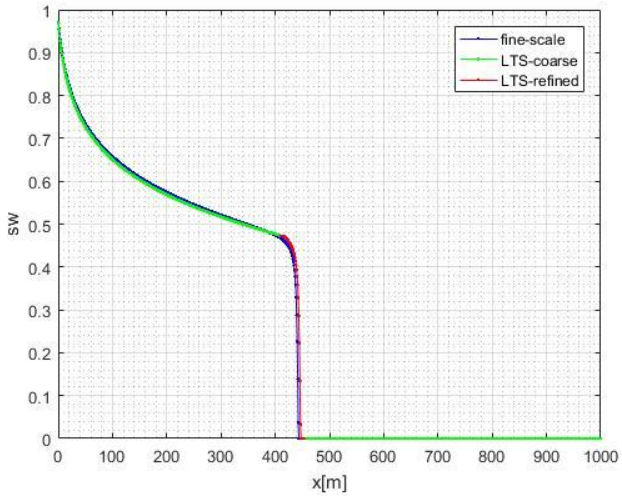
Table 4.6. 1D FIM test case reservoir and well properties

PI($\frac{m^3}{s \cdot Pa}$)	injection rate ($\frac{m^3}{s}$)	initial condition	production well pressure (Pa)
1000	3×10^{-5}	initial $S_w = 0$ initial $P = 10^4 Pa$	0

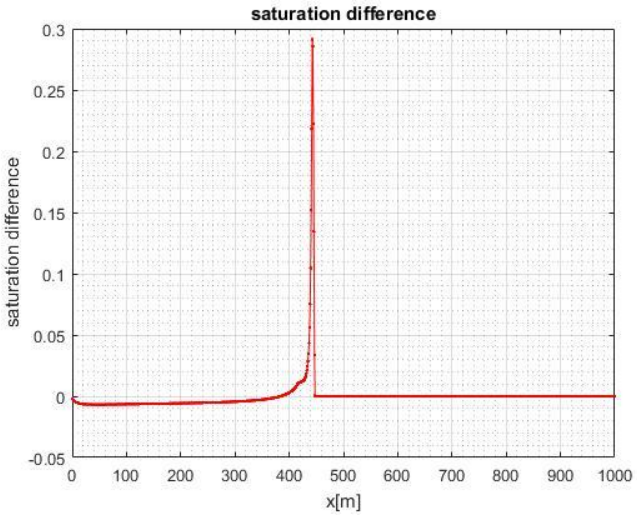
The total simulation time is 20 days. The coarse-scale time-step is set as 12 hrs /step and the fine-scale time-step is set as 2 hrs/step. The CFL number for the coarse time zone is then 19 and for the refined time zone is 3. CPU time for the fine-scale in time simulation is 313s and for the LTS method is 232s, which means that LTS in 1D FIM is computation efficient. The Jacobian matrix in FIM contains both pressure and saturation derivatives while the Jacobian matrix in the sequential implicit method only contains the saturation derivatives. So solving the fully implicit Jacobian matrix consumes more CPU time and the advantage of solving the reduced size refined zone Jacobian matrix in fully implicit method is more significant in reducing the CPU time.

The simulation results of saturation is shown in Fig. 4.6 (a) and pressure is shown in Fig. 4.6 (c). From Fig. 4.6 (b) the simulation result of LTS in 1D FIM method is really close to the simulation result of fine-scale in time FIM and the difference is small. According to Fig. 4.6 (d), in the refined zone the pressure difference is less than 0.5%, which means the pressure reconstruction in the refined zone is successful. So the simulation results of the LTS method in 1D FIM is good.

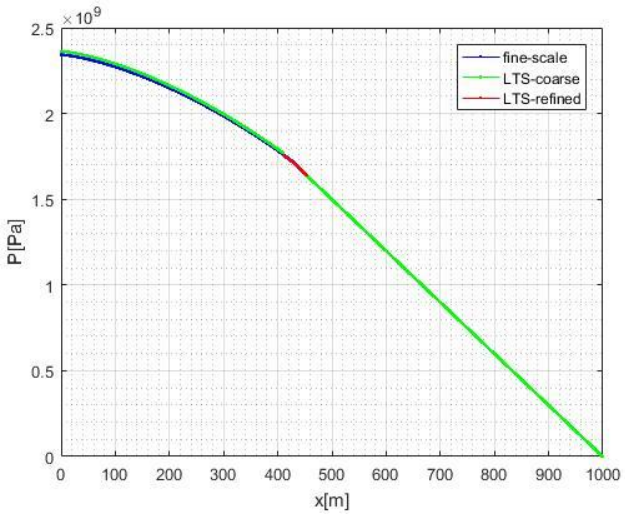
From Fig. 4.6 (e) there is only 4% of cells involved in the refined zone at the last step. Also from Fig. 4.6 (f) the mass difference is very small so the LTS in FIM 1D is mass conservative. Similar to the Fig. 4.4 (g), Fig. 4.6 (g) shows that the shock front position of the LTS is in between the coarse time result and fine-scale in time result. Thus the result of the local time-stepping in fully implicit method is acceptable.



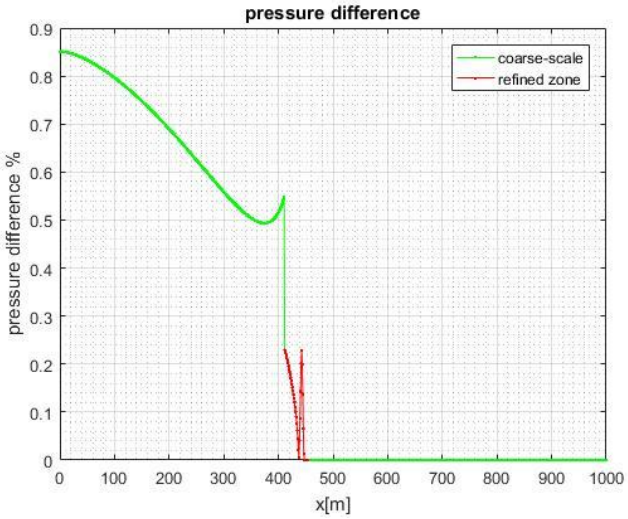
(a) Saturation distribution



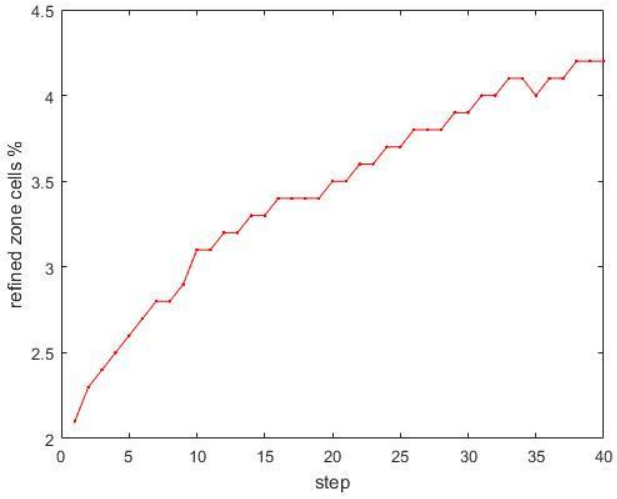
(b) Comparison of saturation to fine-scale results



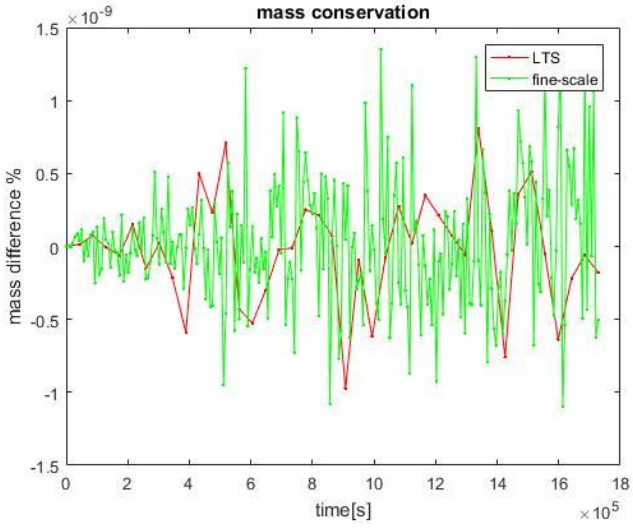
(c) Pressure distribution



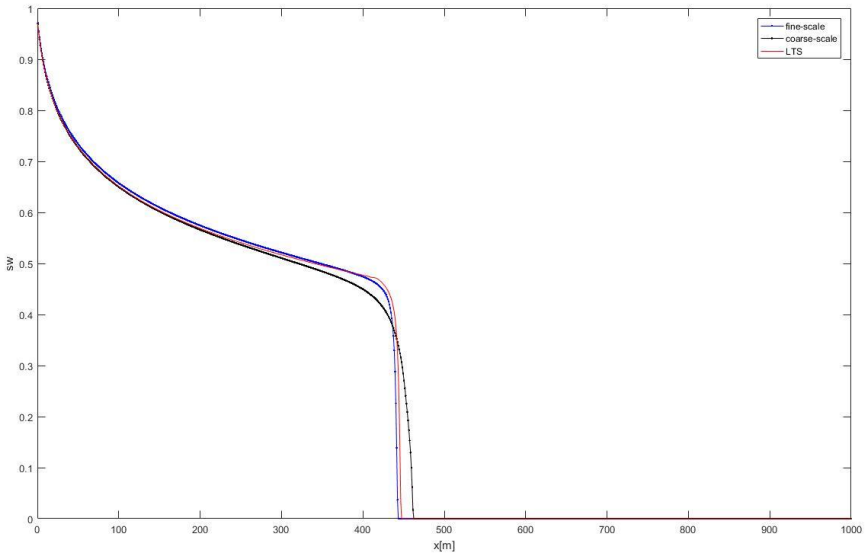
(d) Comparison of saturation to fine-scale results



(e) Refined zone cells number each step



(f) Mass conservation check



(g) Shock front position comparison between three time-scales

Figure 4.6. 1D FIM numerical results with LTS method and fine-scale in time method. Coarse time step is 12 hrs and the fine-scale time-step is 2 hrs. Total simulation time is 20 days. CPU time for the LTS is 232s and for the fine-scale in time is 313s.

4.3.2 2D test case

In this test case injection well is rate controlled and the production well is bottom hole pressure controlled. The well properties and the reservoir properties are shown in the Table 4.7.

Table 4.7. 2D FIM test case reservoir and well properties

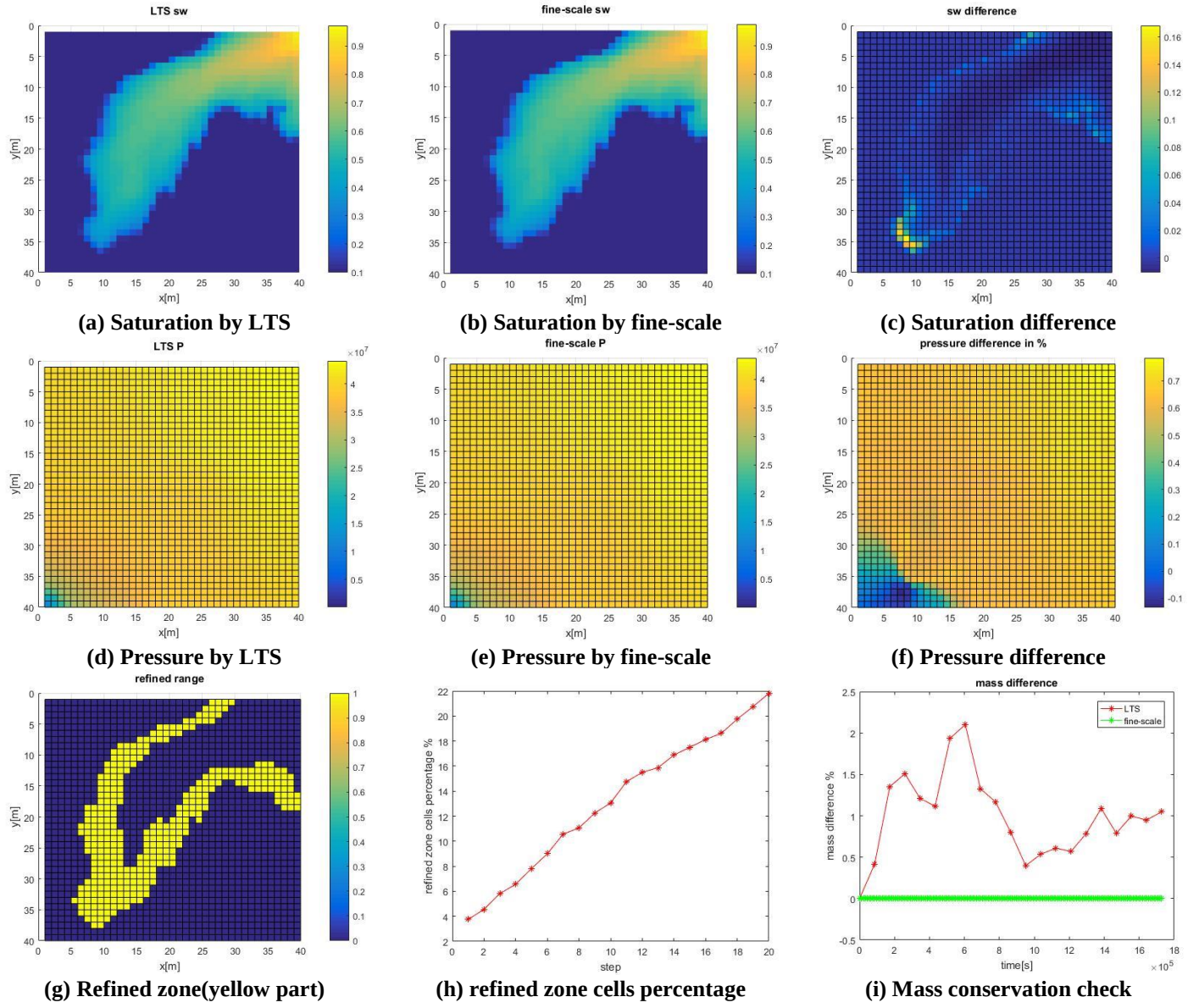
PI($\frac{m^3}{s \cdot Pa}$)	injection rate ($\frac{m^3}{s}$)	initial condition	production well pressure (Pa)
1000	3×10^{-5}	initial $S_w = 0$ initial $P = 10^4 Pa$	0

The total simulation time is 20 days. The coarse time-step is set as 24 hrs/step and the fine-scale time-step is set as 3 hrs/step. CFL number for the coarse time zone is 5.8 and for the refined time zone is 0.8. The CPU time for local time-stepping method is 322s and for fine-scale method is 1144s. So LTS for 2D FIM is computation efficient.

Saturation results are shown in Fig. 4.7 (a) and Fig. 4.7 (b). Pressure results are shown in Fig. 4.7 (d) and Fig. 4.7 (e). From Fig. 4.7 (c) the difference between the fine-scale time-step results and the LTS results are only located in the small area around the shock front and the difference is minor. The pressure difference in the refined zone is less than 1% from Fig. 4.7 (f). So the simulation results are in good quality.

Fig .4.7 (g) presents the refined zone. Around 22% of cells involved in the refined zone simulation at the last step from Fig. 4.7 (h), which makes solving the refined Jacobian matrix efficient. The mass conservation check in Fig. 4.7 (i) shows that the mass difference for the LTS is around 1% and the maximum is 2%, which is a little bit higher than the estimation.

It is important to note that when the CFL number in the refined zone is close to 1 or higher then there is a problem of convergence in the refined zone iterative computation. After debugging the simulator, the updated pressure after each loop varies a lot at the boundary between the coarse time zone and refined time zone. This may be the main problem that leads to failure of convergence. Reason for this is still the Neumann boundary condition at the boundary between the two time zones. The modification of the boundary condition is needed.



(a) Saturation by LTS **(b) Saturation by fine-scale** **(c) Saturation difference**

(d) Pressure by LTS **(e) Pressure by fine-scale** **(f) Pressure difference**

(g) Refined zone(yellow part) **(h) refined zone cells percentage** **(i) Mass conservation check**

Figure 4.7. 2D FIM numerical results with LTS method and fine-scale in time method. Coarse time step is 24 hrs and the fine-scale time-step is 3 hrs. Total simulation time is 20 days. CPU time for the LTS is 322s and for the fine-scale in time is 1144s.

5. Conclusions and Future Work Recommendation

Local time-stepping method is presented as an alternative for multi-scale in time reservoir simulation. It allows for different time-steps to be used in the same domain. To maintain the mass conservation, the flux continuity is used. At each inner stage of a given large time-step, the fluxes at time n are used in the IMPES method, while the fluxes at time $n + 1$ in the implicit methods.

In IMPES the local time-stepping simulation results are in good quality as there is minor difference compared to the fine-scale in time results and the CPU time for simulation is reduced. For LTS in 2D IMPES large parts of the reservoir domain are involved in the refined zone simulation but this doesn't affect the computation efficiency. The developed local time-stepping for 1D IMPES simulation strategy was formulated strictly mass-conservative due to the flux continuity requirement. For 2D test cases this mass-conservative scheme is not promised.

For the implicit transport method as sequential implicit and FIM, the results of simulation are both good compared to the fine-scale in time simulation results. For sequential implicit method there is not much computation gain with LTS method for small domains where the front takes a big part of the reservoir. However, even with the studied small domain sizes, the LTS computation efficiency is significant for FIM. The limited percentage of the cells in the refined zone reduced the size of the Jacobian matrix and it makes the iterative computation much faster. The coupling of the pressure and saturation derivatives in Jacobian matrix makes iterative computation of FIM linearized system very time consuming. So by using the refined matrixes to reduce the size of the Jacobian matrix and residual vector the CPU time on iterative computation can be reduced.

Mass-conservation for all the 1D test cases are maintained, which means the fluxes continuity can guarantee the mass conservation in the local time stepping. For all the 2D test cases the mass conservation is not maintained. This is the point that requires future researching.

Table 5.1 concludes the aspects of local time-stepping methods in three kinds of simulation methods.

Table 5.1. Conclusions of local time-stepping in different simulation methods

	IMPES	Sequential Implicit	FIM
Computation Efficiency	Yes	No	Yes
Results Quality	Yes	Yes	Yes

For the later stage of researching of this topic, the application of this method to the 3D reservoir model simulation is needed. In 3D it is obvious that in each coarse time-step there will be large number cells involved in the shock front or refined zone. Also this method can be combined with the multi-scale in space strategy as the more computation efficient strategy.

For LTS in FIM the problem mentioned in the section 4.3.2 needs to be solved. Modification of the boundary condition for the refined zone is important to make sure the simulator can converge when the CFL number in the refined zone is close to or higher than 1.

As for the concern of the mass conservation, the flow from refined zone to coarse time zone needs to be investigated. This flux is the potential problem that causes the difference of the shock front positions and the non-conservative mass scheme. In 2D this kind of fluxes increase much more than 1D test case which may causes the mass difference.

References

[1]. Hajibeygi, H., Bonfigli, G., Hesse, M. A., & Jenny, P. (2008). Iterative multiscale finite-volume method. *Journal of Computational Physics*, *227*(19), 8604-8621.

[2]. Cusini, M., van Kruijsdijk, C., & Hajibeygi, H. (2016). Algebraic dynamic multilevel (ADM) method for fully implicit simulations of multiphase flow in porous media. *Journal of Computational Physics*, 314, 60-79.

[3]. Praditia, T., Helmig, R., & Hajibeygi, H. (2018). Multiscale formulation for coupled flow-heat equations arising from single-phase flow in fractured geothermal reservoirs. *Computational Geosciences*, 1-18.

[4]. Ewing, R. E., Lazarov, R. D., & Vassilevski, P. S. (1990). Finite difference schemes on grids with local refinement in time and space for parabolic problems I. Derivation, stability, and error analysis. *Computing*, *45*(3), 193-215.

[5]. Faille, I., Nataf, F., Willien, F., & Wolf, S. (2008). Local time-steps for a finite volume scheme. *arXiv preprint arXiv:0805.0550*.

[6]. Constantinescu, E. M., & Sandu, A. (2007). Multirate timestepping methods for hyperbolic conservation laws. *Journal of Scientific Computing*, *33*(3), 239-278.

[7]. Delpopolo, L.C. (2016). *Application of the multirate TR-BDF2 method to the time discretization of nonlinear conservation laws*. Politecnico di Milano.

[8]. Bank, R. E., Coughran, W. M., Fichtner, W., Grosse, E. H., Rose, D. J., & Smith, R. K. (1985). Transient simulation of silicon devices and circuits. *IEEE Transactions on electron devices*, 32(10), 1992-2007.

[9]. Delpopolo, L.C., Bonaventura, L., Scotti, A., & Formaggia, L. (2018). A conservative implicit multirate method for hyperbolic problems. *arXiv preprint arXiv:1802.04650.*

[10]. Thomas, G. W., & Thurnau, D. H. (1983). Reservoir simulation using an adaptive implicit method. *Society of Petroleum Engineers Journal*, *23*(05), 759-768.

[11]. Russell, T. F. (1989). Stability analysis and switching criteria for adaptive implicit methods based on the CFL condition. In *SPE Symposium on Reservoir Simulation*. Society of Petroleum Engineers.

[12]. Brooks, R., & Corey, T. (1964). Hydraulic properties of porous media. Hydrology Papers, *Colorado State University*, 24, 37.

[13]. Eymard, R., Gallouët, T., & Herbin, R. (2000). Finite volume methods. *Handbook of numerical analysis*, 7, 713-1018.

[14]. Courant, R., Isaacson, E., & Rees, M. (1952). On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, 5(3), 243-255.

[15].Younis, R. M. (2011). *Modern advances in software and solution algorithms for reservoir simulation*. Stanford University.

[16]. Jenny, P., Tchelepi, H. A., & Lee, S. H. (2009). Unconditionally convergent nonlinear solver for hyperbolic conservation laws with S-shaped flux functions. *Journal of Computational Physics*, *228*(20), 7497-7512.

[17]. Cusini, M., Lukyanov, A. A., Natvig, J., & Hajibeygi, H. (2015). Constrained pressure residual multiscale (CPR-MS) method for fully implicit simulation of multiphase flow in porous media. *Journal of Computational Physics*, 299, 472-486.

[18]. SPE Comparative Solution Project. Retrieved from website:

https://www.spe.org/web/csp/datasets/set02.htm.

Appendix A. Algorithm

A.1. Local time-stepping in IMPES

Table A.1.Algorithm of local time-stepping in IMPES

Local time-stepping in IMPES Algorithm

Input: fluids properties, reservoir model, well conditions.
Set the endpoint Q_T and the injection rate Q. Set up per.
Compute shock front $\frac{\partial f}{\partial S_{shock}}$.
While $Q \cdot dt_2 < Q_T$
    Compute $f_w^n(S_w^n)$ and $\lambda_t^n(S_w^n)$;
    $P^{n+1} = F(\lambda_t^n)$ and obtain total velocity U^{n+1};
    Compute fine-scale time-step dt_1 , dt_{stable} and dt_2 ;
    $P1 = zeros(N, 1)$;
    if $dt_{stable}(i) < dt_1$
        $P1(i) = 1$; %in the refined zone
    else
        $P1(i) = 0$;
    end
    at initial time use fine scale simulation :$S_{w,\Omega_c+\Omega_r}^{n+1} = T\left(S_{w,\Omega_c+\Omega_r}^{n}, dt_1, U_t^{n+1}\right)$;
    for r=1:m+1
        $S_{w,\Omega_c}^{n+1} = T_c\left(S_{w,\Omega_c}^{n}, dt_2, P1\right)$;
        $S_{w,\Omega_r}^{n+1} = T_r\left(S_{w,\Omega_r}^{n}, dt_1, P1\right)$;
        adjustment of $P1$for dynamic refined zone;
        update properties;
    end
    update the properties;
    $t = t + dt_2$;
end

Sketch of the algorithm is shown in Table A.1. Q_T is the user input total amount of water injection end point. For example it can be 50% of the total pore volume. per is the percentage of the cells that should be involved in the refined zone for example 10%. $P^{n+1} = F(\lambda_t^n)$ represents the pressure solver to compute the pressure at time n+1. Ω_r represents the refined time zone and Ω_c represents coarse time zone. T represents the explicit solver on the whole reservoir model and T_c represents the explicit solver on coarse time zones and T_r represents the explicit solver on refined time zones. m is the ratio between the time zones as in Eq. (33).

Note that some of the symbols used in here will be used in the later algorithm and their meaning don't change.

A.2. Local time-stepping in Sequential Implicit

Table A.2.Algorithm of local time-stepping in Sequential Implicit

Local time-stepping in sequential implicit Algorithm

Input: fluids properties, reservoir model properties, well conditions.
Set the total simulation time T. Set up coarse time-step dt_2 and fine-scale time-step dt_1.
Compute shock front $\frac{\partial f}{\partial S_{shock}}$.
While $t<T$
Compute $\lambda_t^n(S_w^n)$;
$P^{n+1} = F(\lambda_t^n)$ and obtain total velocity U_t^{n+1};
While converged==0
Construct $R_w(dt_2)$ and $J(dt_2)$;
$\delta S_w = J/-R_w$;
$S_w^{v+1} = S_w^v + \delta S_w$;
if $\|R_w\|_2 \leq \varepsilon$
converged=1;
end
update properties
end
$P1 = zeros(N, 1)$;
if $\frac{\partial f}{\partial S}(i) \geq \frac{\partial f}{\partial S_{shock}}$ || $S_w^{n+1}(i) \leq S_w^{shock}$
$P1(i) = 1$; %in the refined zone
else
$P1(i) = 0$;
end
Set $S_{w,\Omega_r} = S_{w,\Omega_r}^n$
for r=1:m
While converged==0
Construct $R_w(dt_1)$ and $J(dt_1)$;
Construct the refined matrix $P2, P3$;
Construct reduced Jacobian J_r , $R_{w,r}$
$\delta S_{w,\Omega_r} = J_r/-R_{w,r}$;
$S_{w,\Omega_r}^{v+1} = S_{w,\Omega_r}^v + \delta S_{w,\Omega_r}$;
if $\left\|R_{w,r}\right\|_2 \leq \varepsilon_r$
converged=1;
end
update properties;
end
end
update properties;
$t = t + dt_2$;
end

Sketch of the algorithm is shown in Table A.2. $\|R_w\|_2$ means that the Euclidean norm of the residual vector is used. ε is the tolerance of the residual for convergence in here. ε_r is the tolerance for convergence in the refined zone. Note that they can be equal but it is not allowed that the $\varepsilon_r > \varepsilon$. This may cause the convergence failure in the predictor stage.

A.3. Local time-stepping in FIM

Table A.3.Algorithm of local time-stepping in FIM

Local time-stepping in FIM Algorithm

Input: fluids properties, reservoir model properties, well conditions.
Set the endpoint of time-steps T. Set up coarse time-step dt_2 and fine-scale time-step dt_1.
Compute shock front $\frac{\partial f}{\partial S_{shock}}$.
While $t<T$
  While converged==0
    Construct $R(dt_2)$ and $J(dt_2)$;
    $\begin{bmatrix}\delta P \\ \delta S_w\end{bmatrix} = J/-R$;
    $S_w^{v+1} = S_w^v + \delta S_w$;
    $P^{v+1} = P^v + \delta P$;
    if $\|R\|_2 \leq \varepsilon$
      converged=1;
    end
    update properties
  end
  $P1 = zeros(N, 1)$;
  if $\frac{\partial f}{\partial S}(i) \geq \frac{\partial f}{\partial S_{shock}}$ $||S_w^{n+1}(i) \leq S_w^{shock}$
    $P1(i) = 1$; %in the refined zone
  else
    $P1(i) = 0$;
  end
  Set $S_{w,\Omega_r} = S_{w,\Omega_r}^n$ and $P_{\Omega_r} = P_{\Omega_r}^n$;;
  for r=1:m
    While converged==0
      Construct $R(dt_1)$ and $J(dt_1)$;
      Construct the refined matrix $P2, P3$;
      Construct reduced Jacobian J_r , R_r
      $\begin{bmatrix}\delta P_{\Omega_r} \\ \delta S_{w,\Omega_r}\end{bmatrix} = J_r/-R_r$;
      $P_{\Omega_r}^{v+1} = \mathrm{P}_{\Omega_r}^v + \delta P_{\Omega_r}$;
      $S_{w,\Omega_r}^{v+1} = S_{w,\Omega_r}^v + \delta S_{w,\Omega_r}$;
      if $\|R_r\|_2 \leq \varepsilon_r$
        converged=1;
      end
      update properties
    end
  end
  update properties
  $t = t + dt_2$;
end

Appendix B. Adaptive Implicit Method

The adaptive implicit method is different from the local time-stepping method as the time-step used in the adaptive implicit method is constant everywhere. The main idea for adaptive implicit method is only do implicit simulation on small part of the cells in the reservoir model instead of doing implicit simulation on the whole reservoir model to save the CPU time.

B.1. Flux between explicit and implicit zone

At the boundary between the explicit cell and implicit cell, the flux draws the attention. The fluxes from the explicit zone is still at time n but the fluxes required by the implicit zone should be at time $n+1$, as shown in Fig. B.1. One strategy is to update the explicit zone to time $n+1$ firstly then do simulation on the implicit zone. Since the fully implicit method is used in the implicit zone the pressure needs to be reconstructed. Here the pressure value at the two explicit zone cells immediate neighboring the implicit zone can be used as the Dirichlet boundary of the implicit zone problem, which is similar to the strategy used in local time-stepping in fully implicit method.

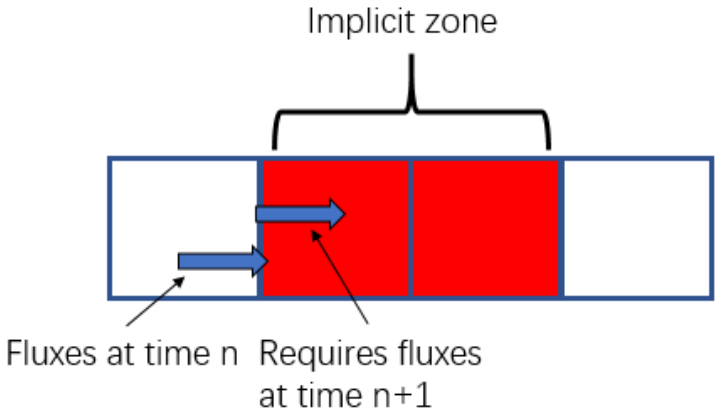


Figure. B.1. Fluxes at the boundary between the explicit cell and implicit cell

B.2. Selection of implicit zone

In this research the selection of the implicit zone would be based on the CFL numbers following Russell's paper. For cell i if its CFL number is higher than 1 then this cell should be treated as implicit. Usually for the two phases flow in the reservoir the implicit zone will be the cells around the shock front.

A problem happened to this selection method is that it may not catch up with the moving shock front. This is similar to the problem stated in section 3.3.3. Shock front may leave the implicit zone selected at time n as shown in Fig. B.2. A few cells should be added to make sure implicit zone can track with the shock front dynamically. Thus after determining the implicit zone more cells should be added even their CFL numbers are 0. Here is the simple example to illustrate the addition of cells. There are 8 equidistant cells for the simple 1D reservoir model similar to Fig. 2.1 with flow from left to right,. If still the partition vector $P1$ is used in here and assume the cell 2,3,4 is selected as implicit cells at time n then :

$$P1 = [0; 1; 1; 1; 0; 0; 0; 0]$$

If the maximum CFL number is 2, then add the two cells immediate in front of the last implicit cell. Partition vector $P1$ becomes:

$$P1 = [0; 1; 1; 1; 1; 1; 0; 0]$$

This partition vector will also be used in implicit zone computation. Similar to the refined Jacobian in section 3.3.3 the refined matrix $P2$ and $P3$ are used in here to make the implicit zone Jacobian matrix and residual vector

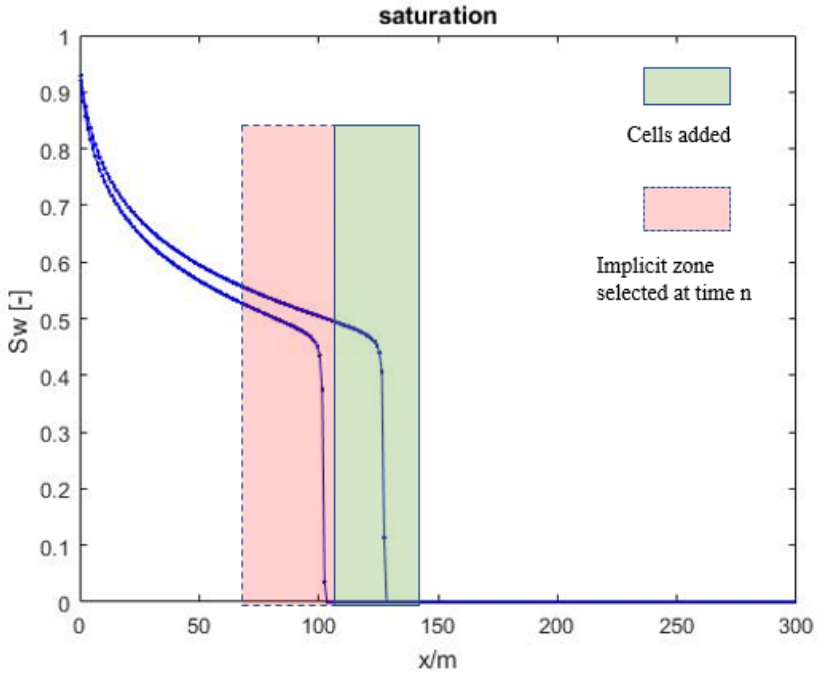


Figure. B.2. Cells added to the implicit zone

B.3. Algorithm

Table B.1.Algorithm of local time-stepping in FIM

AIM Algorithm
Input: fluids properties, reservoir model properties, well conditions. Set the total simulation time T Set up time-step dt. **While** $t<T$ Compute $f_w^n(S_w^n)$ and $\lambda_t^n(S_w^n)$; **if** t==0 call for fully implicit solver; **else** Compute CFL numbers and determine implicit zone with $P1$; $P^{n+1} = F(\lambda_t^n)$ and obtain total velocity U^{n+1}; $S_{w,\Omega_E}^{n+1} = T_E\left(S_{w,\Omega_E}^n, dt, P1\right)$; Set $P_{\Omega_I} = P_{\Omega_I}^n$; **While** converged==0 Construct R and J; Construct the matrix $P2, P3$; Construct reduced Jacobian J_r , R_r

$\begin{bmatrix} \delta P_{\Omega_I} \\ \delta S_{w,\Omega_I} \end{bmatrix} = J_r / -R_r$;
$P_{\Omega_I}^{v+1} = \mathrm{P}_{\Omega_I}^{v} + \delta P_{\Omega_I}$;
$S_{w,\Omega_I}^{v+1} = S_{w,\Omega_I}^{v} + \delta S_{w,\Omega_I}$;
if $\|R_r\|_2 \leq \varepsilon$
converged=1;
end
update properties
end
end
update properties
$t = t + dt$;
end

Sketch of the algorithm is shown in Table B.1. Ω_I represents the implicit zone and Ω_E represents explicit zone. m is the ratio between the time zones.

B.4. AIM numerical results

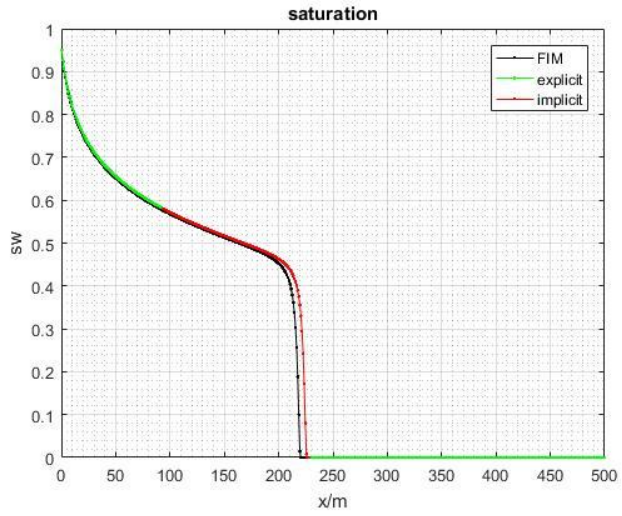
In this test case there is one injection well at the left boundary cell of the reservoir and one production well at the right boundary cell. Injection well is constant injection rate controlled and production well is bottom pressure controlled. Reservoir properties and well properties are listed in table B.2. Fluids properties are the same as Table 4.1.

Table B.2. 1D AIM test case reservoir and well properties

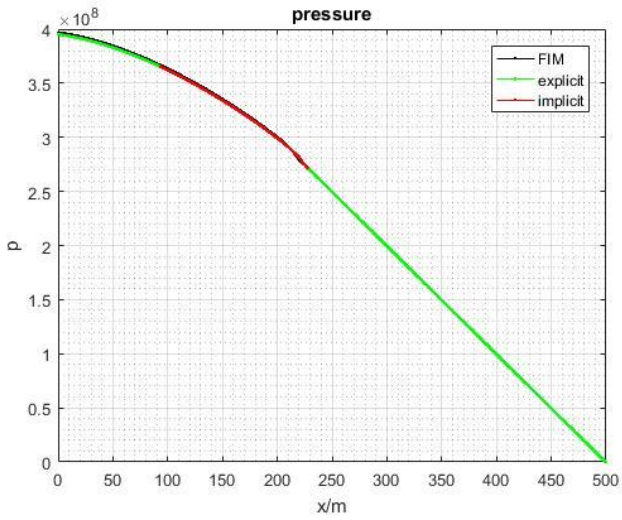
PI($\frac{m^3}{s \cdot Pa}$)	injection rate ($\frac{m^3}{s}$)	initial condition	porosity
1000	10^{-5}	initial $S_w = 0$ initial $P = 10^4 Pa$	0.2
production well pressure (Pa)	reservoir length(m)	number of grid cells	grid size (m)
10^4	500	500	1

The absolute permeability K is $10^{-13}\ m^2$ or 100 mD everywhere in this homogeneous reservoir. The endpoint of the simulator is when total time for the simulator is 700 hrs. The time-step is set as 7 hrs/step and the CFL number higher than 0.95 will be set as the implicit cell.

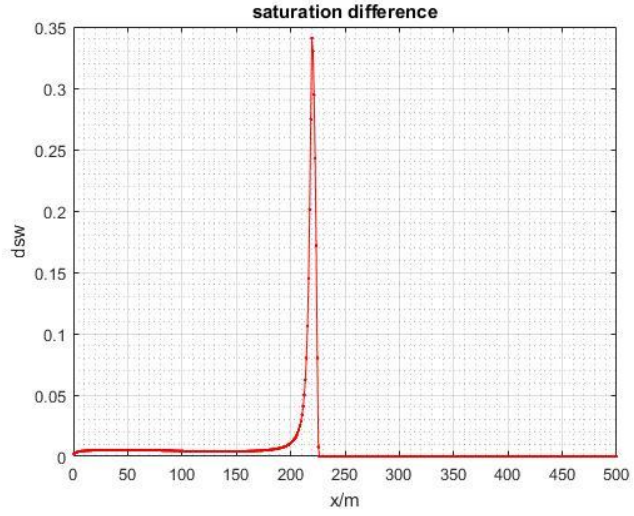
CPU time for the FIM method is 30s and the for the AIM is 9s. Saturation result is shown in Fig. B.3 (a) and pressure result is shown in Fig. B.3 (b).



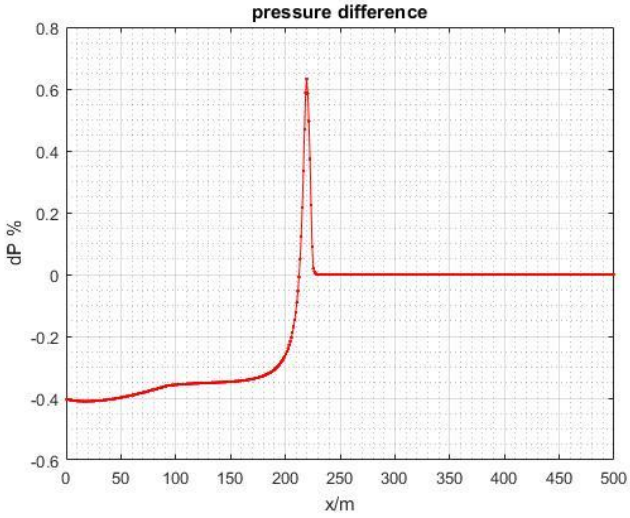
(a) Saturation comparison



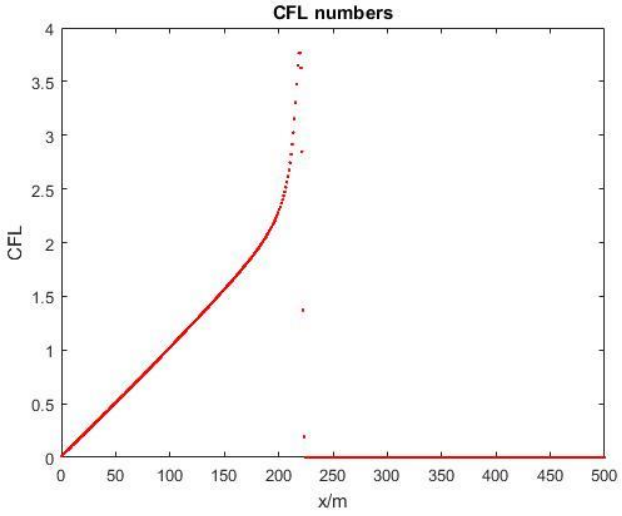
(b) Pressure comparison



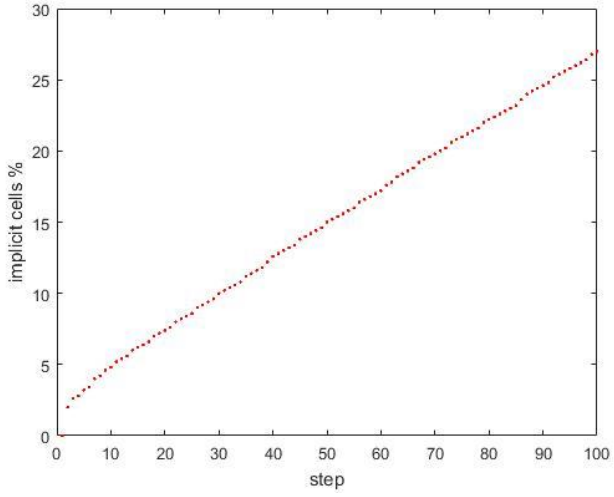
(c) Saturation difference



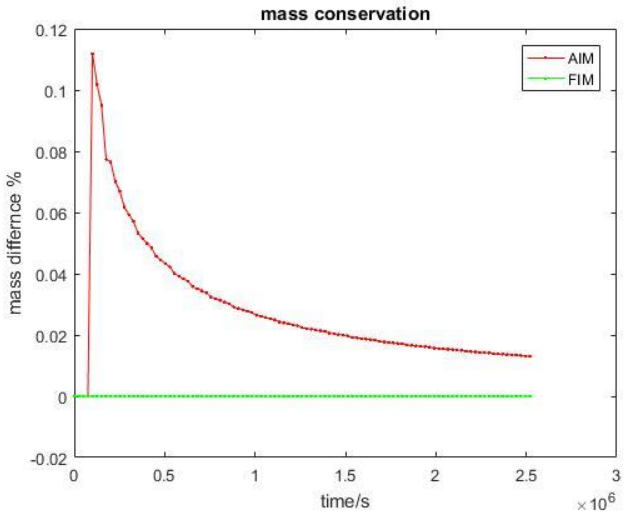
(d) Pressure difference



(e) CFL numbers



(f) Implicit cells number



(g) Mass conservation check

Figure B.3. 1D AIM numerical results with AIM and FIM. Time step is 7 hrs. Total simulation time is 700 hrs. CPU time for FIM is 30s and for AIM is 9s.

From Fig. B.3 (c), it is obvious that the quality of the adaptive implicit simulation result is a little bit different compared to the fully implicit method results. However, the pressure result is very good from Fig. B.3 (d) that the difference is only around 0.4%. Thus the simulation result of adaptive implicit method is acceptable. There are around 20% cells being put into implicit zone at the last step from figure B.3 (f). and the reduced Jacobian becomes efficient for computer to compute. From figure B.3 (g) the adaptive implicit method is not strictly mass conservative.

Thus we can state that the AIM combined with the fully implicit method and IMPES is very computation efficient but the simulation results are only acceptable. It is not strict mass-conservative method. How to determine the implicit zone for the next time-step would be the main problem for 2D AIM simulation. More strategies need to be considered in how to make the implicit zone correctly.

Appendix C. LTS in Sequential Implicit on SPE-10 Top Layer

In this test case there is one injection well at the north-west corner and one production well at the south-east corner. Injection well is injection rate controlled and production well is bottom hole pressure controlled. The well properties and the reservoir properties are shown in the table C.1. The permeability field is shown in Fig. C.1.

Table C.1. 2D sequential implicit test case reservoir and well properties

PI($\frac{m^3}{s \cdot Pa}$)	injection rate ($\frac{m^3}{s}$)	initial condition	porosity
1000	3×10^{-5}	initial $S_w = 0$	0.2
production well pressure (Pa)	reservoir length(m)	number of grid cells	grid size (m)
0	220×60	220×60	1×1

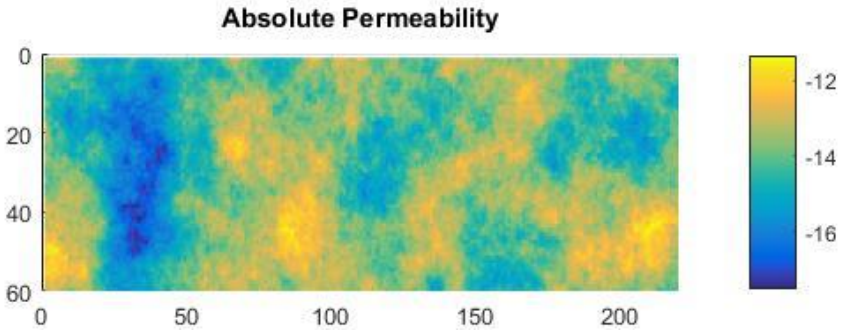
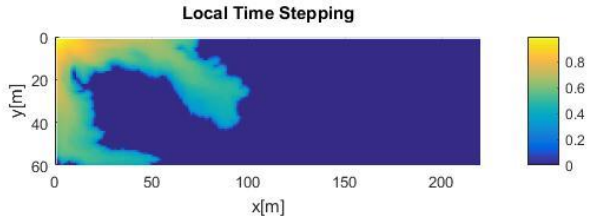


Figure C.1. Permeability field of SPE-10 top layer. Permeability is in common logarithm scale

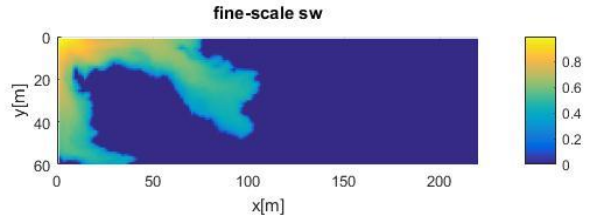
The endpoint of the simulator is when total time for the simulator is 120 days. The coarse time-step is set as 4 days/step and the fine-scale time-step is set as 16 hrs/step. CFL number for the coarse time zone is 7.3 and for the refined time zone is 1.2. The CPU time for local time-stepping method is 2228s and for fine-scale method is 1833s. So in the 2D large size reservoir model the LTS in sequential implicit is still not computation efficient.

Simulation results for saturation are shown in Fig. C.2 (a) and (b) and the results for pressure are shown in Fig. C.2 (e) and (f). From Fig. C.2 (c) the difference between the LTS results and the fine-scale results are mostly located around the shock front region. This difference is still very significant. Pressure difference is around 1% from Fig. C.2 (g) while at one part of the shock front the pressure difference is about 4%. So this simulation result is acceptable and in good quality. From Fig. C.2 (i) represents the mass difference.

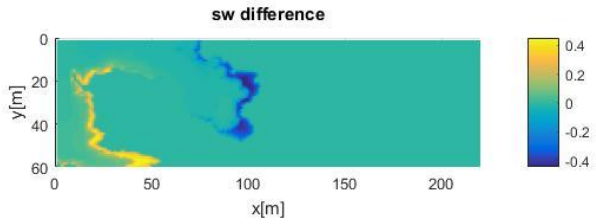
In the larger test case like SPE-10 top layer, there is still no computation efficiency for LTS in 2D sequential implicit method. Thus this strategy may need deeper investigation as it may not contain computation efficient and not generate precise simulation results.



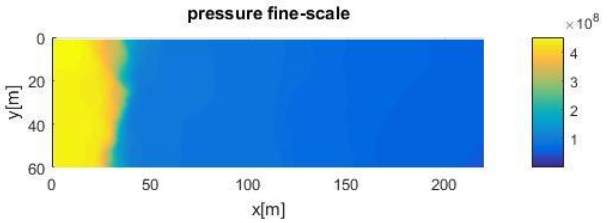
(a) Saturation by LTS



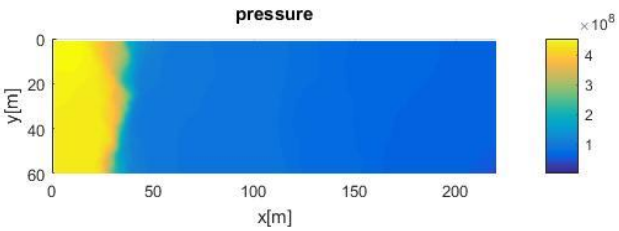
(b) Saturation by fine-scale in time



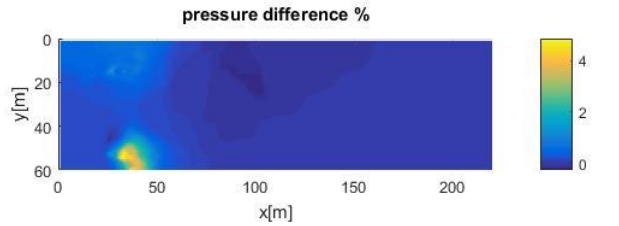
(c) Saturation difference



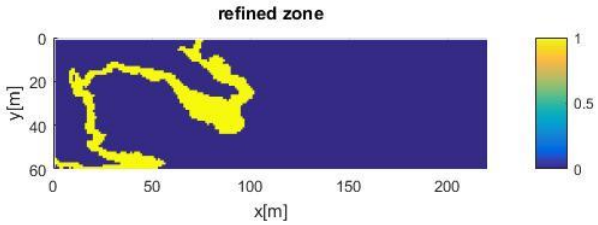
(e) Pressure by LTS



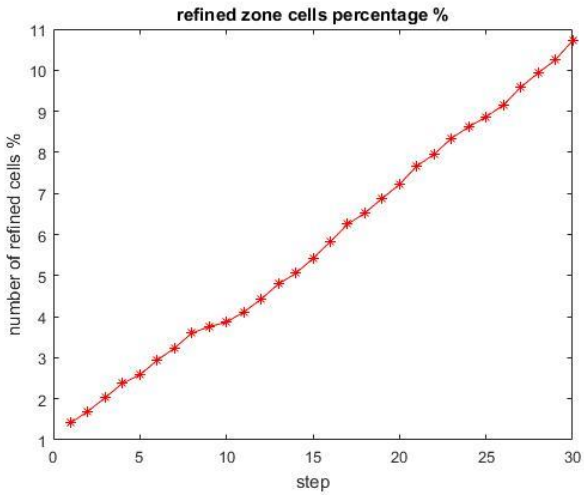
(f) Pressure by fine-scale in time



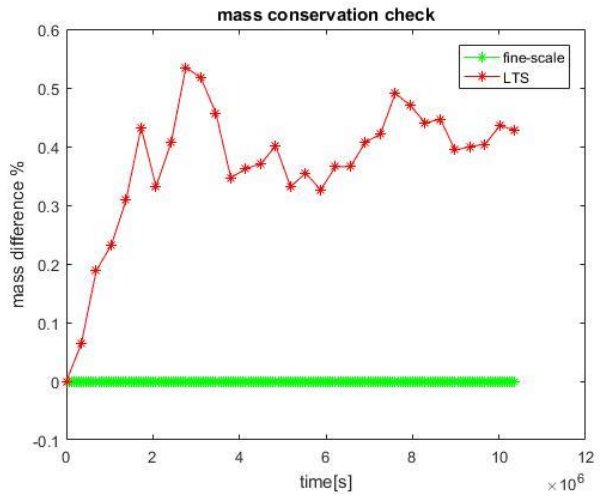
(g) Pressure difference



(d) Refined zone (yellow part)



(h) Refined zone cells percentage



(i) Mass conservation check

Figure C.2 .2D sequential implicit numerical results with LTS method and fine-scale in time method on the whole SPE-10 top layer. Coarse time step is 4 days and the fine-scale time-step is 16 hrs. Total simulation time is 120 days. CPU time for the LTS method is 1833s and for the fine-scale in time is 2228s.