HIGH PERFORMANCE COMPOSITIONAL 3D RESERVOIR SIMULATION IN CONJUNCTION WITH UNSTRUCTURED GRIDS

Ivens da Costa Menezes Lima

Anthônio Nunes Moreira Netto

João Henrique Bessa Gomes

Edilson Pimentel Drumond Filho

Joaquim Bento Cavalcante Neto

Francisco Marcondes

ivenscml@yahoo.com.br

anthonio@fisica.ufc.br

joao.hgomes@hotmail.com

epdrumond@gmail.com

joaquimb@lia.ufc.br

marcondes@ufc.br

Federal University of Ceará (UFC)

Campus do Pici, 60440-900, Fortaleza, Ceará, Brazil

Kamy Sepehrnoori

kamys@mail.utexas.edu

The University of Texas at Austin

1 University Station, 78712, Austin, Texas, USA

Abstract. *Large-scale simulation projects involving hundreds of thousands of gridblocks require application of parallelization techniques in order to achieve practical computational times. There are essentially three ways of parallelizing a simulator by using distributed memory, shared memory, and a combination of the two mentioned approaches. This work is*

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

based on the first approach, where the domain is divided among the processors involved and each process is responsible only for its portion of the computations. This approach has two main advantages; it reduces the required memory per process and allows the simulations to be carried out using clusters with large number of processors. In this work, we used open source libraries to partition the computational domain, manage the grid information between the processors, and solve the linear system of equations generated from the discretization of partial differential equations modeling the fluid flow in the reservoir. ParMetis (Parallel Graph Partitioning and Fill-reducing Matrix Ordering) is used to partition the computational domains, FMDB (Flexible Distributed Mesh Database) is responsible to manage the grid information between the processors, and PETSc (Portable, Extensible Toolkit for Scientific Computation) solves the linear systems of equations. The numerical approach is based on the Element based Finite Volume Method (EbFVM) in conjunction with unstructured grids. In fact, the main challenge was to manage the grid, fluids, and reservoir data set in such a way that the communications between the processors were reduced. We used an in-house compositional, multicomponent/multiphase simulator called UTCOMP, which was developed at The University of Texas at Austin, in order to perform this implementation. We show that the EbFVM adapts well for modeling reservoirs with complex geometries and we also present its performance in a parallel mode. The results are presented in terms of oil and gas production curves, speedup curves and CPU times for various case studies.

Keywords: *Parallelization, EbFVM, 3D unstructured grids, UTCOMP simulator*

1. INTRODUCTION

One of the most valuable petroleum engineer tool is the computational simulation. Based on data collected in field and through simulation, it is possible for engineers to evaluate the best strategy in order to achieve the maximum production rates. Simulators allow engineers to analyze the optimal configuration of injector and producer wells, study what of type fluid will be injected and its composition, perform a history matching analysis, investigate the petroleum thermodynamic behavior, among other things. These are complex tasks to perform and they are also time consuming. Simulators must provide reliable results in the shortest possible time.

There are many ways to improve the speed of a simulator. This work is focused on parallelization, i.e., give the computer program the ability to be executed by more than one processor at the same time. Parallelization, basically can be categorized in three types: distributed memory, shared memory, and a hybrid of both (Grama et al., 2003). This work makes use of the first type of parallelization in conjunction with 3D unstructured meshes, and shows how the CPU time and memory required by the simulation can be significantly reduced when using this technique.

The simulator used in the development of this work is called UTCOMP. UTCOMP is a multiphase/multicomponent, compositional simulator developed at The University of Texas at Austin (Chang, 1990) to handle several hydrocarbon recovery processes. Even though UTCOMP has many numerical formulations implemented (IMPEC, Semi-implicit, and fully implicit), this work is based on an IMPEC (Implicit Pressure Explicit Composition) formulation in conjunction with the Element based Finite Volume Method (EbFVM) (Araújo et al., 2016; Marcondes and Sepehrnoori, 2010).

1.1 Parallelism

Although the supercomputing trough vectorization arose in the 1970 decade, it was only used in reservoir simulation in the late 1980's (Killough, 1993). Since then, many works have been published in this area. Wheeler and Smith (1990) investigated parallelization of 3D implicit formulations for two phase (water-oil) flows. Killough and Bhogeswara (1990, 1991) investigated a parallelization of a compositional simulator using Cartesian grids in hypercube computers. Killough (1993) studied some aspects of the parallelization as the load balance and the solution of linear systems. Ghori et al. (1995) have investigated UTCOMP simulator in parallel using shared memory and distributed memory environments using Cartesian grids.

Parallelization is such an important feature for modern simulators that petroleum companies have been investing and applying this technique to their simulators in the last fifteen years. This can be observed in the works of Shiralkar et al. (2005), Usadi et al. (2007), Dogru et al. (2008, 2009), and Beckner et al. (2015). The main goal is to simulate petroleum reservoirs with the maximum accuracy, and also enabling the elaboration of specific strategies for each field in order to maximize the oil recovery (Dogru et al., 2009).

1.2 Unstructured meshes

The use unstructured meshes in reservoir simulation date from the 1980 decade. Rozon (1989) used the CVFEM (Control Volume Finite Element Method), developed originally by Baliga and Patankar (1980) to computational fluid dynamics, to simulate a monophase flow

using quadrilateral elements. Maliska (2004) explained that the CVFEM designation gives a wrong idea that we are employing a finite element method based on control volumes, and he suggested the substitution of the CVFEM nomination by the EbFVM (Element based Finite Volume Method), since we still have a finite volume approach that only borrows from finite element method the idea of elements and shape functions.

Cordazzo et al. (2005) demonstrated the geometric flexibility and the grid orientation effect provided by the EbFVM for two-dimensional, two-phase (water-oil) flows. Marcondes and Sepehrnoori (2010) investigated the viability of the method to solve multiphase/multicomponent, heterogeneous and anisotropic problems and showed that much less grid elements are necessary to obtain the same accuracy results when compared to Cartesian grids. Marcondes et al. (2013) expanded EbFVM for 3D grids in conjunction with a fully implicit compositional simulator. Araújo et al. (2016) applied the EbFVM approach for 3D compositional simulation using an IMPEC (Implicit Pressure Explicit Composition) formulation. Fernandes et al. (2015) employed TVD (Total Variation Diminishing) interpolation function for 3D compositional reservoirs using the EbFVM and an IMPEC approach. All aforementioned works were implemented in serial mode. Lima et al. (2015) presented the parallelization of the UTCOMP simulator for the IMPEC approach using 2D unstructured grids. This paper is devoted to the parallelization of the IMPEC approach of the UTCOMP simulator using 3D unstructured grids and the EbFVM approach.

2. GOVERNING EQUATIONS

The IMPEC approach of the UTCOMP simulator is based on Acs et al. (1985). In this approach, the primary variables is the oil pressure and number of moles that are evaluated explicitly after the oil pressure that is solved implicitly. The secondary variables are evaluated next.

2.1 Mass balance

The mass balance equation, ignoring the dispersion term, for each component is given by

$$\frac{1}{V_b}\frac{\partial N_i}{\partial t} - \vec{\nabla}\cdot\left(\sum_{j=1}^{n_p}\xi_j x_{ij}\vec{\vec{k}}\lambda_{rj}\left(\nabla P_j - \gamma_j \nabla D\right)\right) + \frac{q_i}{V_b} = 0, \tag{1}$$

where N is the number of moles, V_b is the bulk volume, t is the time, n_p is the number of phases, ξ_j is the molar fraction of phase j, x_{ij} is the molar fraction of component i in phase j, $\vec{\vec{k}}$ is the absolute permeability tensor, λ_{rj} is the relative mobility of phase j, P is oil pressure, γ_j is the specific gravity of phase j, D is the depth, q_i is the sink/source well term for component i. Velocity of each phase is evaluated accordingly to Darcy's law for multiphase flow in porous media. The well term is considered positive if it is an injector well, and negative otherwise. Depth is positive in downward direction.

2.2 Pressure equation

The pressure equation for the Acs et al. (1985) is obtained from the assumption that all pore volume is occupied by the fluid available into the porous media. Using this hypothesis and neglecting the dispersion term, the pressure equation is given by

$$\left(\phi^{\circ} C_f - \frac{1}{V_b}\frac{\partial V_T}{\partial P}\right)\frac{dP}{dt} = \sum_{i=1}^{n_c+1}\left\{\bar{V}_{Ti}\vec{\nabla}\cdot\left[\sum_{j=1}^{n_p}\xi_j x_{ij}\vec{\vec{k}}\lambda_{rj}\left(\nabla P - \gamma_j \nabla D\right)\right]\right\} - \sum_{i=1}^{n_c+1}\bar{V}_{Ti}\frac{q_i}{V_b} \tag{2}$$

where ϕ° is the porosity in a reference pressure, C_f is the rock compressibility factor, V_T is the total volume of fluid, n_c is the number of hydrocarbon components, n_c *+1* denotes the water component, and $\bar{V}_{Ti}$ is the derivative of fluid volume related to number of moles of component *i*. Further information about the demonstration can be found in Chang (1990).

3. ELEMENT BASED FINITE VOLUME METHOD

In EbFVM approach, the domain is divided into elements, and these elements are further divided into sub-elements according to the number of nodes of each element. The conservation equations (material balance equations and pressure equation) are then integrated for each sub-element. Due to that, the sub-elements are called sub-control volumes (SCV). The control volume (CV) around each node of the grid is formed by gathering all the sub-control volumes that share the same node. Figure 1 shows an example of a 2D unstructured mesh, highlighting the SCV for elements 1, 2, 6, and 7 (marked in blue) and CV around node 5.

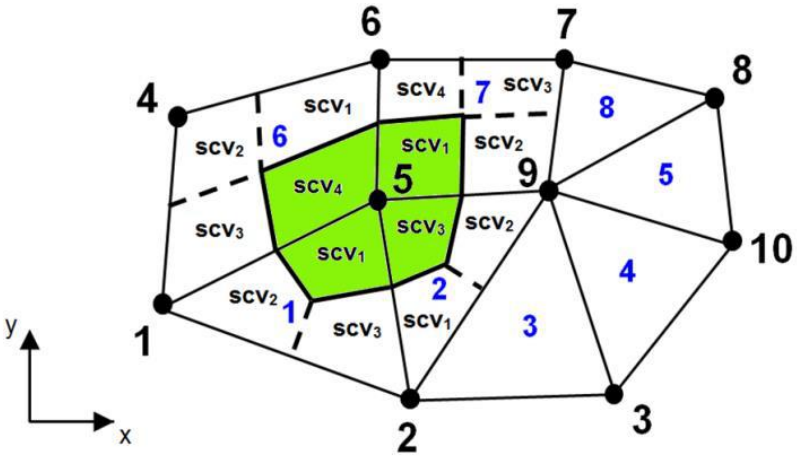


Figure 1. 2D unstructured grid

3.1 Shape functions

The EbFVM utilizes the shape functions to evaluate the mass flow rates along the areas of each sub-control volume surrounding each node. Figure 2 show the elements in the physical plane. In that figure, the interface marked with circle are the areas, where the fluxes need to be evaluated. In order to generalize, the shape functions are commonly written in the computational plane, where the element shape is all regular no matter the element is distorted in the physical plane.

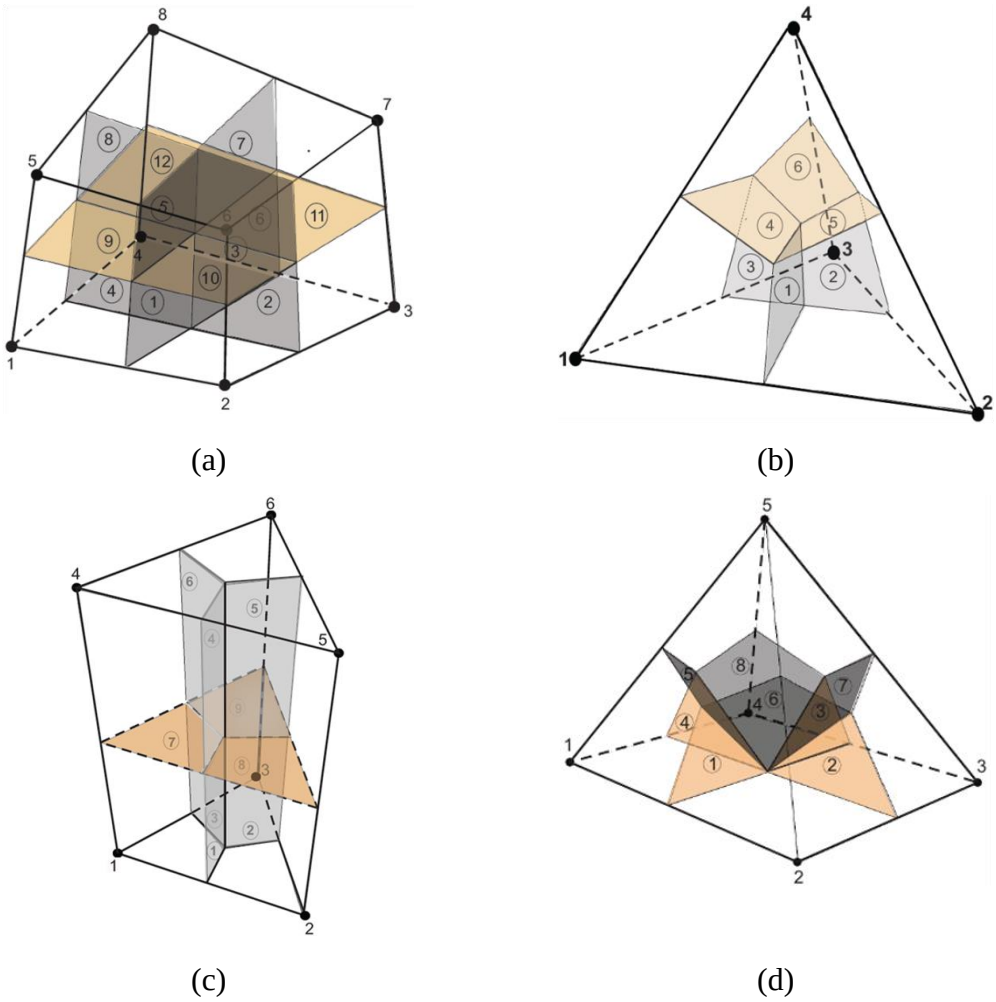


Figure 2.3D elements in the physical plane: (a) hexahedron, (b) tetrahedron, (c) prism and (d) pyramid

The functions in the computational plane (ξ,η,γ) for tetrahedrons, prisms, pyramids and hexahedrons are given by Eqs. (3) through (6), respectively:

$$
\begin{aligned}
N_1(\xi,\eta,\gamma) &= 1-\xi-\eta-\gamma \\
N_2(\xi,\eta,\gamma) &= \xi \\
N_3(\xi,\eta,\gamma) &= \eta \\
N_4(\xi,\eta,\gamma) &= \gamma
\end{aligned}
\tag{3}
$$

$$
\begin{aligned}
N_1(\xi,\eta,\gamma) &= (1-\xi-\eta)(1-\gamma); \\
N_2(\xi,\eta,\gamma) &= \xi(1-\gamma); \\
N_3(\xi,\eta,\gamma) &= \eta(1-\gamma); \\
N_4(\xi,\eta,\gamma) &= \gamma(1-\xi-\eta); \\
N_5(\xi,\eta,\gamma) &= \xi\gamma; \\
N_6(\xi,\eta,\gamma) &= \eta\gamma.
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
N_1(\xi,\eta,\gamma) &= \frac{1}{4}\left[(1-\xi)(1-\eta)-\gamma+\frac{\xi\eta\gamma}{1-\gamma}\right]; \\
N_2(\xi,\eta,\gamma) &= \frac{1}{4}\left[(1+\xi)(1-\eta)-\gamma-\frac{\xi\eta\gamma}{1-\gamma}\right]; \\
N_3(\xi,\eta,\gamma) &= \frac{1}{4}\left[(1+\xi)(1+\eta)-\gamma-\frac{\xi\eta\gamma}{1-\gamma}\right]; \\
N_4(\xi,\eta,\gamma) &= \frac{1}{4}\left[(1-\xi)(1+\eta)-\gamma-\frac{\xi\eta\gamma}{1-\gamma}\right]; \\
N_5(\xi,\eta,\gamma) &= \gamma.
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
N_1(\xi,\eta,\gamma) &= \frac{1}{8}(1+\xi)(1-\eta)(1+\gamma);\\
N_2(\xi,\eta,\gamma) &= \frac{1}{8}(1+\xi)(1-\eta)(1-\gamma);\\
N_3(\xi,\eta,\gamma) &= \frac{1}{8}(1-\xi)(1-\eta)(1-\gamma);\\
N_4(\xi,\eta,\gamma) &= \frac{1}{8}(1-\xi)(1-\eta)(1+\gamma);\\
N_5(\xi,\eta,\gamma) &= \frac{1}{8}(1+\xi)(1+\eta)(1+\gamma);\\
N_6(\xi,\eta,\gamma) &= \frac{1}{8}(1+\xi)(1+\eta)(1-\gamma);\\
N_7(\xi,\eta,\gamma) &= \frac{1}{8}(1-\xi)(1+\eta)(1-\gamma);\\
N_8(\xi,\eta,\gamma) &= \frac{1}{8}(1-\xi)(1+\eta)(1+\gamma),
\end{aligned}
\tag{6}
$$

4. PARALLELIZATION

Parallelization is a technique that consists in rewriting the code in order to make it able to be executed by two or more processors. Its main objective is to reduce computational time and amount of memory required per processor. There are basically three types of parallelization: distributed memory, shared memory, and a combination of both (Grama et al., 2003). This work focus on the first type, where the domain is divided among a number of processors defined by the user, and each processor is responsible for solving only its portion of the reservoir.

A parallelized simulator makes it possible to run large simulations in clusters of computers; simulations that otherwise could not be executed in serial mode. Reducing the execution time by the use of parallelization, engineers can perform more runs, and therefore optimize the operational parameters of the production.

4.1 Auxiliary libraries

The parallelization of a code involves many steps: load balancing and domain partition, exchange of information between processes, solving the linear system in parallel, among them. To develop the code to perform these tasks in an optimized manner is a very hard work to do. Hence, in this work we have used some auxiliary open source libraries to help us to integrate the aforementioned tasks (Silva, 2008).

Flexible distributed Mesh Database. FMDB (Seol, 2005) provides all necessary geometric information about the grid: number of elements and vertices, vertices' index and coordinates, and element connectivities. One processor can simply call FMDB functions to obtain all these information. It is important to mention that FMDB does not divide the grid. It simply calls Zoltan/ParMetis to perform the division, and manages the data provided by them.

Zoltan/ParMetis. Zoltan (Devine at al., 2002) and ParMetis (Parallel Graph Partitioning and Fill-reducing Matrix Ordering) (Lasalle and Karypis, 2013; Schloegel at al., 2002) are the open source libraries responsible for the grid division. They use the k-way partitioning method to divide the grid in an optimized way. Using this method the ghost layers are created with as few elements as possible and makes the elements within a processor spatially grouped together. This type of division diminishes the amount of communication performed between processors, and improves the efficiency of the simulation. Figure 3 shows a grid divided in an efficient manner.

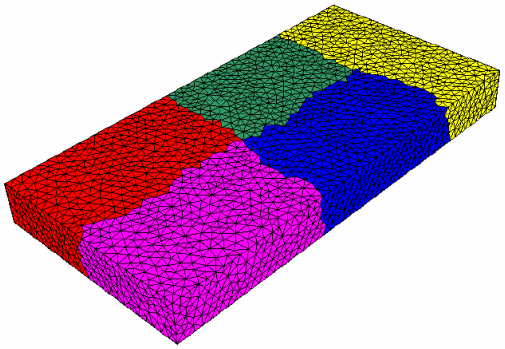


Figure 3. Unstructured 3D mesh divided in five processes

A fundamental concept in the development of a parallel reservoir simulator are the ghost layers. They are the portion of the reservoir where each processor has to exchange information with the surrounding processors. Ghost layers make it possible for one processor to assemble correctly the terms of the equation of control volumes located in the frontier of that processor. Although this paper is focused on 3D grids, Fig. 4 shows the ghost layers for a 2D just to simplify the visualization.

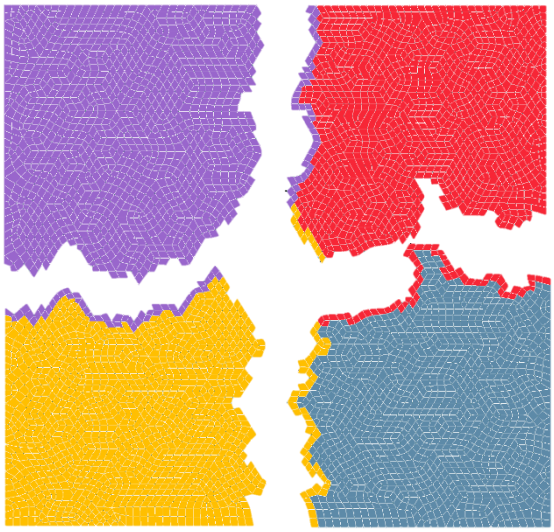


Figure 4. Unstructured divided 2D mesh with highlighted ghost layers

The ghost layers are not created automatically by any library used in this work. For this reason, it was necessary to create a subroutine to perform this task. It uses the divided grid information provided by Zoltan/ParMetis and managed by FMDB.

Portable Extensible Toolkit for Scientific Computation. PETSc (Balay et al., 2015) is a library with many functions to deal with vectors and matrices in parallel. It also has some built-in solvers and pre-conditioners optimized for parallel execution. We used the standard options for the parallel solver: Block-Jacobi for domain decomposition and GMRES for the solution of the local linear system. The good performance of this library presented in literature (Han et al., 2007; Silva, 2008) made PETSc the natural choice for the UTCOMP simulator.

OpenMPI. All parallel operations, i.e. everything involving communication between processors, are performed by functions from the OpenMPI (Gabriel et al., 2004) library. It is one of the many implementations of MPI (Message Passing Interface). Libraries based on MPI are the most used in the development of distributed memory parallel programs. In UTCOMP, all communication is carried out by either collective or asynchronous peer-to-peer functions, therefore decreasing the execution time.

5. RESULTS

Herein it is shown the results obtained with the implementation of parallelization in UTCOMP. The results are shown in terms of oil and gas production curves, CPU times and speedup curves.

5.1 Case study 1

This case study consists of a gas flooding process in a reservoir with an impermeable region that due to use of unstructured grids is not necessary to have grid and hence inactive cells. The injected and resident fluids are characterized by six hydrocarbon components and three phases

exist in the reservoir (water, oil and gas); although water is immobile during the whole simulation. There are seven wells, being four injectors and three producers, and all of the wells are completed in all 16 layers of the reservoir. Tables 1 through 3 describe the reservoir properties, reservoir and injection fluid compositions, and well constraints.

Table 1. Reservoir properties – Case 1

Properties	**Value**
Pressure	10.34 MPa
Temperature	344.26 K
Porosity	0.35
Permeability (X, Y, Z)	10^{-14}, 10^{-14} and 10^{-14} m²
Initial water saturation	0.17

Table 2. Reservoir and injection fluid compositions – Case 1

Component	**Initial composition in reservoir**	**Composition for the injection fluid**
C_1	0.50	0.770
C_3	0.03	0.200
C_6	0.07	0.010
C_{10}	0.20	0.010
C_{15}	0.15	0.005
C_{20}	0.05	0.005

Table 3. Well operation constraints – Case 1

Well	**Operation constraint**	**Value**
Injector	Constant flow rate	28317 m³/day
Producer	Constant BHP	8.96 MPa

Figure 5 shows the grid used for this case study that is composed of 36,975 hexahedrons and has 41,392 vertices. Producer wells are identified by the red head, and injectors by the blue head. When working with unstructured grids, one can simply the reservoir grid and create the grid only for the regions, where the fluid flow occurs; therefore, eliminating the use of inactive cells, which is common when working with structured grids.

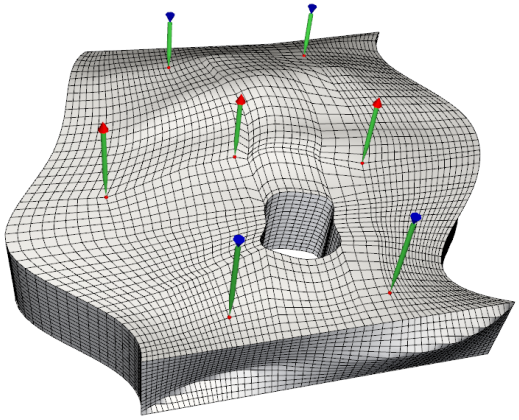
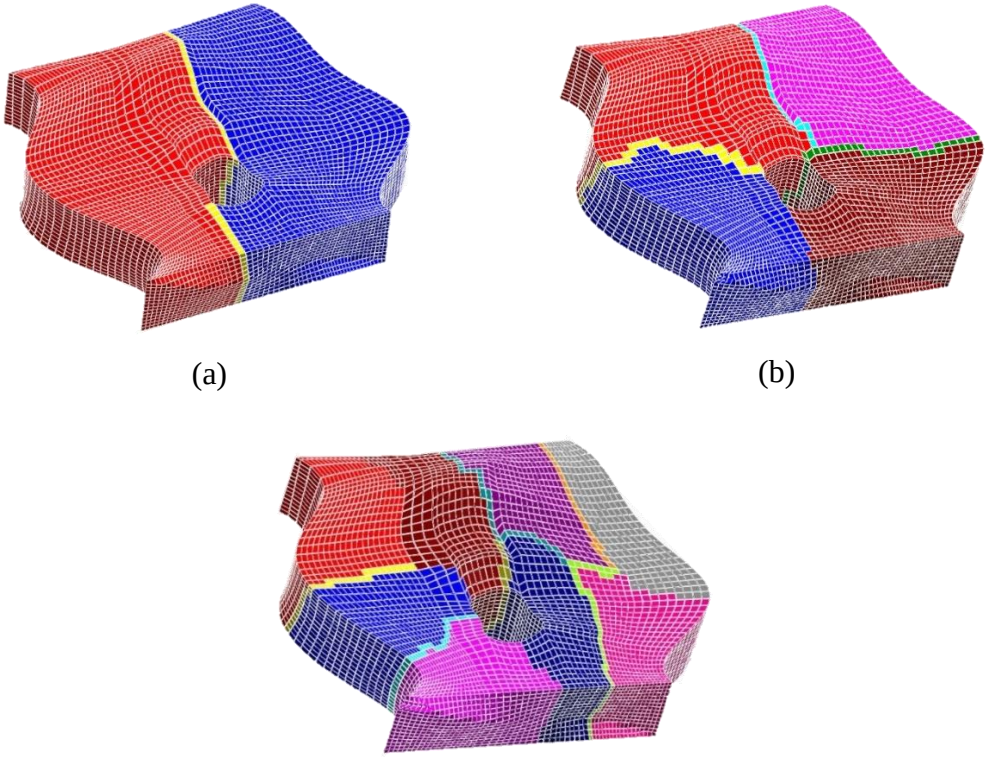


Figure 5. Grid with "hole" – case study 1

Figure 6 shows the grid used for this case study divided in 2, 4, and 8 processes.

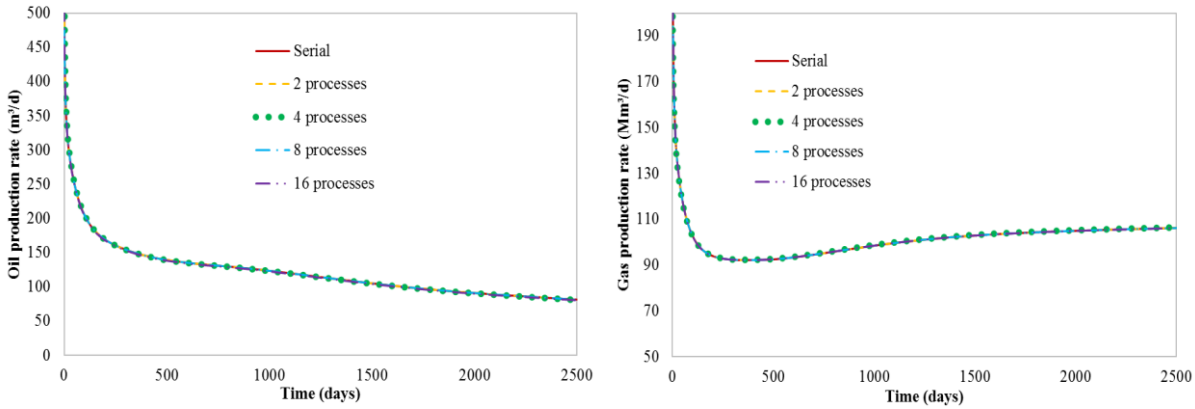


(a) (b)

(c)

Figure 6. Mesh divided in (a) 2, (b) 4 and (c) 8 processes for Case1

The oil and gas production curves for up to 2,500 days of production are shown in Fig. 7.



(a) (b)

Figure 7. Oil and gas production curves for Case 1

From Fig. 7, it is possible to observe that the curves are independent of the number of processors used, hence demonstrating the parallelization was correctly implemented.

Figure 8 shows the CPU times, and Fig. 9 shows the speedup curve for this case.

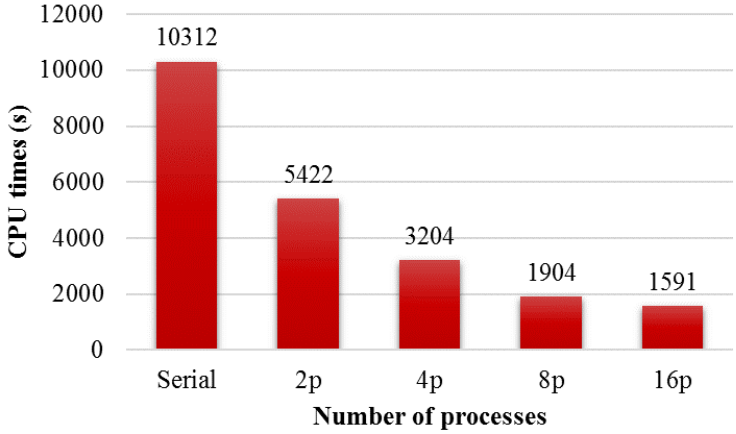


Figure 8. CPU times for Case 1

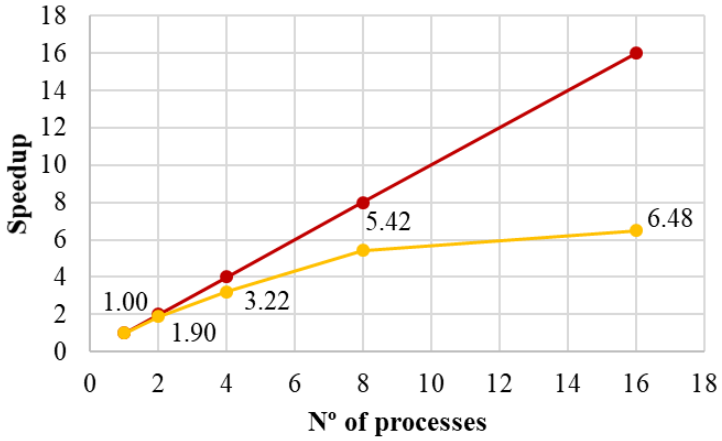


Figure 9. Speedup for Case 1

It can observed that as the number of processes increase, the speedup also increases. This shows that the parallelization has the expected effect, and the simulation runs faster in parallel when compared to serial mode. Although efficiency is below 50% for 16 processes, it is still executed more than six times faster than the serial simulation.

5.2 Case study 2

This case consists of a gas flooding process characterized by three hydrocarbon components. Initially, only two phases are present in the reservoir (oil and water). As the recovery process proceeds, a gas phase appears in the reservoir. For the case study 2, a quarter-of-five-spot configuration is employed and the domain is discretized through a grid composed only of hexahedrons with 203,456 nodes. The reservoir data, injection fluid compositions, and well operational constraints are given in Tables 4 through 6, respectively.

Table 4. Reservoir properties – Case 2

Properties	**Value**
Width, height, and depth	397.0, 397.0 and 91.5 m
Pressure	20.68 MPa
Temperature	300 K
Porosity	0.30
Permeability (X, Y, Z)	10^{-13}, 10^{-13} and 10^{-14} m²
Initial water saturation	0.25

Figure 10 presents the grid used for case study 2 that has 203,456 nodes and 182,250 hexahedrons. Again, the injector well is the one with the blue top, and the producer is the one with the red top; both wells are completed in all layers of the reservoir. Figure 11 shows the grid divided in 2, 4, and 8 processes.

Table 5. Reservoir and injection fluid compositions – Case 2

Component	**Initial composition in reservoir**	**Composition in injection fluid**
CO_2	0.01	0.95
C_1	0.19	0.05
nC_{16}	0.80	-

Table 6. Well operation constraints – Case 2

Well	**Operation constraint**	**Value**
Injector	Constant flow rate	566340 m^3/day
Producer	Constant BHP	20.68 MPa

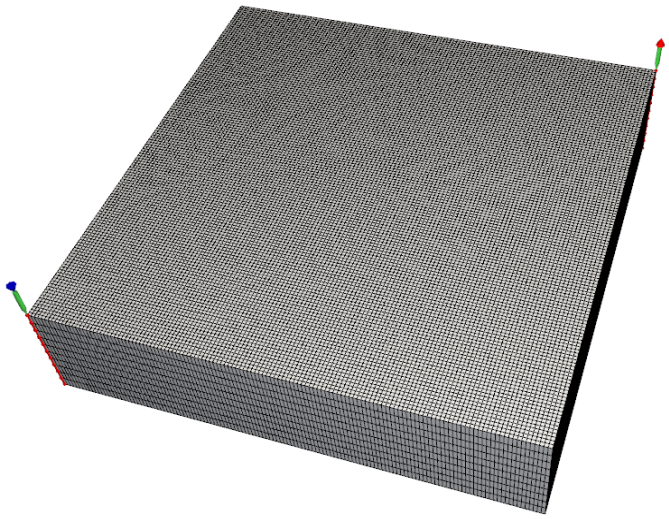


Figure 10. Hexahedron grid

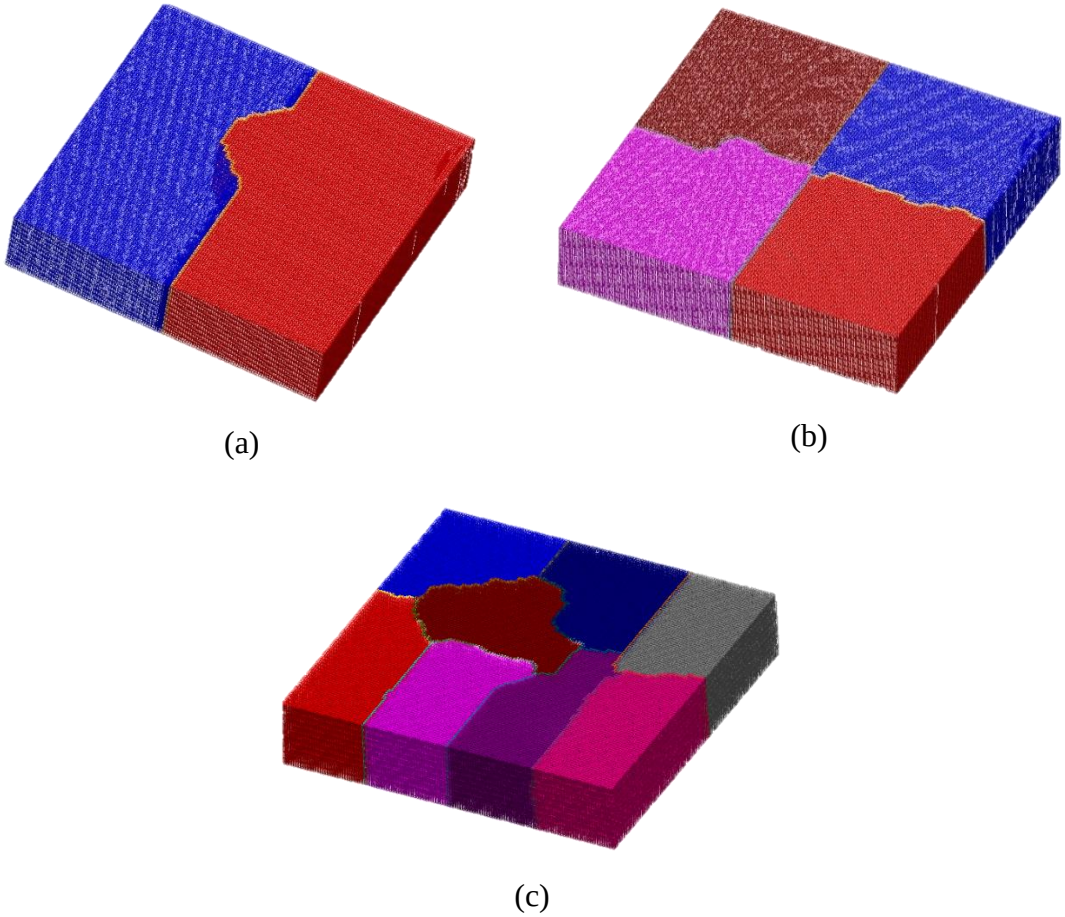


Figure 11. Hexahedron grid divided in (a) 2 processes, (b) 4 processes, and (c) 8 processes

The oil and gas production rates obtained with 1 through 16 processes are presented in Fig. 12. From this figure, we can observe a good match for the oil and gas rates obtained with all the processors investigated, demonstrating once again that the parallelization was correctly implemented.

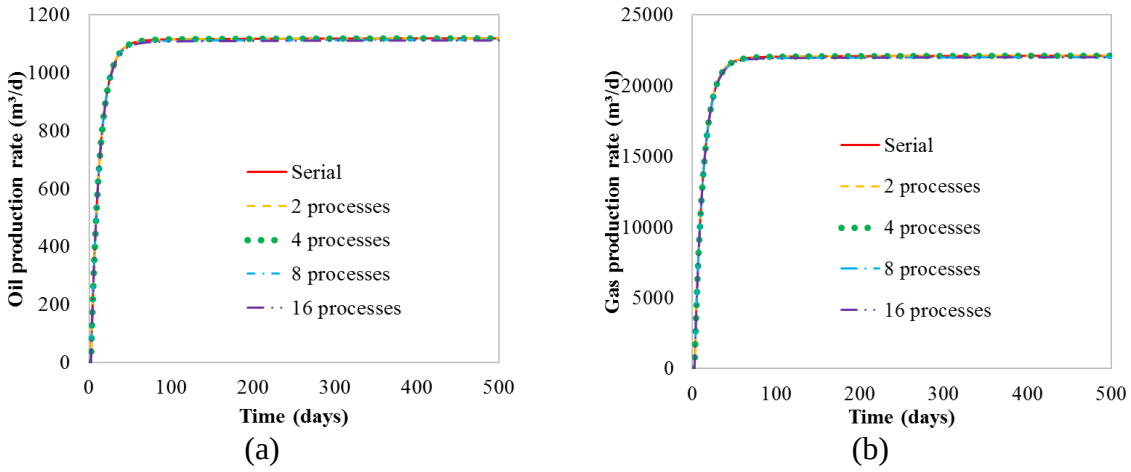


Figure 12. (a) Oil and (b) gas production curves

Figure 13 shows the CPU times obtained with the simulation for case study 2, and Fig. 14 presents the speedups for 2, 4, 8, and 16 processes.

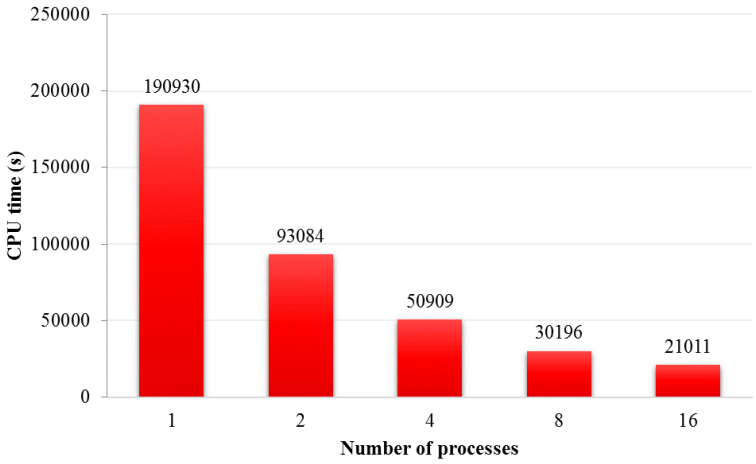


Figure 13. CPU times for Case 2

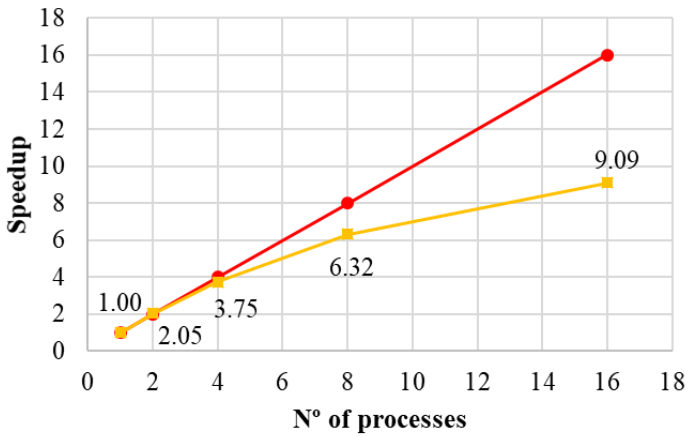


Figure 14. Speedup for Case 2

It can be observed that the CPU time was greatly decreased with parallelization. The result for 16 processes ran 9 times faster than the serial execution, as shown in Fig. 14. Of course this behavior cannot be extended for all simulations, since each one has its own CPU time and speedup curve. Geometric parameters, as number of elements and grid division, reservoir parameters as rock porosity and permeability, and thermodynamic parameters as pressure, temperature and number of components, all these affect the CPU time, and therefore, make each case unique. For this reason, one cannot expect same speedup curves for different cases. Nevertheless, it is expected that any case has its CPU time reduced when running in parallel.

As the number of processes increases, the efficiency decreases and the real speedup distances itself from the ideal one. This behavior is expected and occurs as a consequence of the augmentation of communication between processes. The more processes, more exchange of information is needed to keep each process with the updated data.

6. CONCLUSIONS

Parallelization was successfully implemented in the UTCOMP simulator using unstructured grids. It resulted in great reduction in CPU simulation times, as expected. The use of open source libraries helped to reduce the costs of implementation. It also aided in tasks that, otherwise, would be very difficult to accomplish, such as optimized grid division and parallel solving of the linear system.

The authors believe that EbFVM coupled with parallel unstructured grids provide an excellent tool for petroleum engineers. It allies the geometric flexibility and the conservative property of the EbFVM approach with the computational speed of parallelization, allowing the possibility of running simulations with complex grids and large numbers of elements in computer clusters with several processors.

7. ACKOWLEDGEMENTS

The authors would like to thank The University of Texas at Austin, for providing the code where this work was developed, and Petrobras, for the financial support. Also, we thank ESSS for providing Kraken® for preprocessing and postprocessing of the results.

8. REFERENCES

Acs, G., Doleschall, S., Farkas, E., 1985. General Purpose Compositional Model. *Society of Petroleum Engineers Journal*, vol. 25, n. 4, p. 543–553.

Araújo, A. L. S., Fernandes, B. R. B., Drumond Filho, E. P., Araujo, R. M., Lima, I. C. M., Gonçalves, A. D. R., Marcondes, F., Sepehrnoori, K., 2016. 3d Compositional Reservoir Simulation in Conjunction With Unstructured Grids. Brazilian Journal of Chemical Engineering, vol. 33, n. 2, p. 347–360.

Balay, S., Abhyankar, S., Adams, M., Brown, J., Brune, P., Buschelman, K., Gropp, W., 2015. PETSc User's Manual Work. Condado de DuPage, Argonne National Laboratory.

Baliga, B. R., Patankar, S. V., 1980. A New Finite-Element Formulation For Convection-Diffusion Problems. *Numerical Heat Transfer*, vol. 3, n. 4, p. 393–409.

Beckner, B. L., Haugen, K. B., Maliassov, S., Dyadechko, V., Wiegand, K. D., 2015. General Parallel Reservoir Simulation. *SPE Abu Dhabi International Petroleum Exhibition and Conference.* Abu Dhabi: Society of Petroleum Engineers.

Chang, Y.-B., 1990. Development and Application of an Equation of State Compositional Simulator. PhD Thesis, The University of Texas at Austin.

Cordazzo, J., Maliska, C. R., Silva, A. F. C., Hurtado, F. S. V., 2005. An Element Based Conservative Scheme using Unstructured Grids for Reservoir Simulation. *SPE Youth Forum.*

Devine, K. Boman, E., Heaphy, R., Hendrickson, B., Vaughan, C., 2002. Zoltan data management services for parallel dynamic applications. Computing in Science & Engineering, vol. 4, n. 2, p. 90–96.

Dogru, A. H., Fung, L. S. K., Middya, U., Al-Shaalan, T., Pita, J. A., 2009. A Next-Generation Parallel Reservoir Simulator for Giant Reservoirs. *SPE Reservoir Simulation Symposium.* The Woodlands: Society of Petroleum Engineers. Available in: <http://www.earthdoc.org/publication/publicationdetails/?publication=42008>

Fernandes, B. R. B., Gonçalves, A. D. R., Drumond Filho, E. P., Lima, I. C. M., Marcondes, F., Sepehrnoori, K., 2015. A 3D Total Variation Diminishing Scheme for Compositional Reservoir Simulation Using the Element-Based Finite-Volume Method. *Numerical Heat Transfer, Part A: Applications*, vol. 67, n. 8, p. 839–856.

Gabriel, E., Fagg, G. E., Bosilca, G., Angskun, T., Dongarra, J. J., Squyres, J. M., Sahay, V., Kambadur, P., Barrett, B., Lumsdaine, A., Castain, R. H., Daniel, D. J., Graham, R. L., Woodall, T. S., 2004. Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. *11th European PVM/MPI Users' Group Meeting*. p. 97–104.

Ghori, S. G., Wang, C. H., Lim, M. T., Pope, G. A., Sepehrnoori, K., Wheeler, M. F., 1995. Compositional Reservoir Simulation on CM-5 and KSR-1 Parallel Machines. *13th SPE Symposium on Reservoir Simulation.* San Antonio: Society of Petroleum Engineers. Available in: <http://www.onepetro.org/doi/10.2118/29140-MS>

Grama, A., Gupta, A., Karypis, G., Kumar, V., 2003. *Introduction to Parallel Computing.* 2a ed. Harlow: Pearson Education Limited.

Han, C., Delshad, M., Sepehrnoori, K., Pope, G. A., 2007. A Fully Implicit, Parallel, Compositional Chemical Flooding Simulator. *SPE Annual Technical Conference and Exhibition.* Dallas: Society of Petroleum Engineers. Available in: <http://www.onepetro.org/doi/10.2118/97217-PA>

Killough, J. E., 1993. Will Parallel Computing Ever Be Practical? *Middle East Oil Show.* Manama: Society of Petroleum Engineers. Available in: <http://www.onepetro.org/doi/10.2118/25556-MS>

Killough, J. E., Bhogeswara, R., 1990. Simulation of Compositional Reservoir Phenomena on a Hypercube. *Proceedings of the First IMA/SPE European Conference on the Mathematics of Oil Recovery.* Oxford University Press.

Killough, J. E., Bhogeswara, R., 1991. Simulation of Compositional Reservoir Phenomena on a Distributed-Memory Parallel Computer. *Journal of Petroleum Technology*, vol. 43, n. 11, p. 1368–1374.

Lasalle, D., Karypis, G., 2013. Multi-threaded graph partitioning. *Proceedings of the IEEE 27th International Parallel and Distributed Processing Symposium*, IPDPS 2013.

Lima, I. C. M., Avedo, V. N., Oliveira, J. C. B., Bessa, J. H. G., Neto, J. B. C, Marcondes, F., Sepehrnoori, K, 2015. 2D High Performing Compositional Petroleum Reservoir Simulation in Conjunction with Unstructured Meshes. *23rd ABCM International Congress of Mechanical Engineering.* Rio de Janeiro, Brazil.

Maliska, C. R., 2004. Heat Transfer and Computational Fluid Dynamics. 2a ed. ed. *LTC*. Rio de Janeiro. (In Portuguese).

Marcondes, F.; Sepehrnoori, K., 2010. An element-based finite-volume method approach for heterogeneous and anisotropic compositional reservoir simulation. *Journal of Petroleum Science and Engineering*, vol. 73, n. 1–2, p. 99–106.

Marcondes, F., Santos, L. O. S., Varavei, A., Spepehrnoori, K., 2013. A 3D hybrid element-based finite-volume method for heterogeneous and anisotropic compositional reservoir simulation. *Journal of Petroleum Science and Engineering*, vol. 108, p. 342–351. Available in: <https://doi.org/10.1016/j.petrol.2013.04.024>

Rozon, B. J. A., 1989 Generalized Finite Volume Discretization Method for Reservoir Simulation. *Reservoir Simulation Symposium.* Houston.

Schloegel, K.; Karypis, G.; Kumar, V., 2002. Parallel static and dynamic multi-constraint graph partitioning. *Concurrency and Computation: Practice and Experience*, vol. 14, n. 3, p. 219–240.

Seol, E. S., 2005. FMDB: Flexible Distributed Mesh Database For Parallel Automated Adaptive Analysis. *Rensselaer Polytechnic Institute.*

Shiralkar, G. S., Fleming, G. C., Watts, J. W., Wong, T. W., Coats, B. K., Mossbarger, R., Robbana, E., Batten, A. H., 2005. Development and Field Application of a High Performance, Unstructured Simulator with Parallel Capability. *SPE Reservoir Simulation Symposium.* Houston: Society of Petroleum Engineers. Available in: <http://www.spe.org/elibrary/servlet/spepreview?id=00093080>

Silva, R. S., 2008. Simulação de Escoamento Bifásico Oléo-Água em Reservatórios de Petróleo Usando Computadores Paralelos de Memória Distribuída. Phd Thesis. Universidade Federal de Pernambuco.

Usadi, A. K., Mishev, I. D., Shaw, J. S., Wiegand, K. D., 2007. Parallelization on Unstructured Grids. *SPE Reservoir Simulation Symposium.* Houston: Society of Petroleum Engineers. Available in: <http://www.onepetro.org/doi/10.2118/106063-MS>

Wheeler, J. A.; Smith, R. A., 1990. Reservoir Simulation on a Hypercube. *SPE Reservoir Engineering*, vol. 5, n. 4, p. 544–548.