fluids

MDPI

Article

Numerical Simulation of Multiphase Multicomponent Flow in Porous Media: Efficiency Analysis of Newton-Based Method

Timur Imankulov [1], Danil Lebedev [2], Bazargul Matkerim [1], Beimbet Daribayev [1] and Nurislam Kassymbek [1,*]

[1] Computer Science Al-Farabi Kazakh National University, Almaty 050040, Kazakhstan; imankulov_ts@mail.ru (T.I.); bazargulmm@gmail.com (B.M.); beimbet.daribayev@gmail.com (B.D.)
[2] Astana IT University, Nur-Sultan 010000, Kazakhstan; danil.lebedev.0881@gmail.com
* Correspondence: nuryslam.qassymbek@gmail.com

Abstract: Newton's method has been widely used in simulation multiphase, multicomponent flow in porous media. In addition, to solve systems of linear equations in such problems, the generalized minimal residual method (GMRES) is often used. This paper analyzed the one-dimensional problem of multicomponent fluid flow in a porous medium and solved the system of the algebraic equation with the Newton-GMRES method. We calculated the linear equations with the GMRES, the GMRES with restarts after every m steps—GMRES (m) and preconditioned with Incomplete Lower-Upper factorization, where the factors L and U have the same sparsity pattern as the original matrix—the ILU(0)-GMRES algorithms, respectively, and compared the computation time and convergence. In the course of the research, the influence of the preconditioner and restarts of the GMRES (m) algorithm on the computation time was revealed; in particular, they were able to speed up the program.

Keywords: Newton's method; Krylov subspace method; GMRES; GMRES (m); ILU-GMRES; multicomponent flow

check for updates

Citation: Imankulov, T.; Lebedev, D.; Matkerim, B.; Daribayev, B.; Kassymbek, N. Numerical Simulation of Multiphase Multicomponent Flow in Porous Media: Efficiency Analysis of Newton-Based Method. *Fluids* **2021**, *6*, 355. https://doi.org/10.3390/fluids6100355

Academic Editors: Goodarz Ahmadi and Pouyan Talebizadeh Sardari

Received: 2 September 2021
Accepted: 24 September 2021
Published: 8 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The modeling of hydrodynamic processes that take place in oil reservoirs is one of the difficult problems of fluid mechanics. This is due to the fact that the processes occurring in reservoirs can be very complex. It is necessary to take into account phase transitions, chemical transformations, temperature effects, degassing processes when pressure drops at the bottom and in the reservoir, etc. When simulating multiphase, multicomponent flow in porous media, the phase properties can vary depending on the composition of phases, temperatures (in oil reservoirs, they can reach up to 200 °C) and pressure (in oil reservoirs, it can reach up to 150 MPa). For adequate modeling of complex flow processes that occur in oil reservoirs, it is necessary to take into account the component composition of the phases [1,2]. Accounting for convection, diffusion, phase transitions and chemical reactions makes it difficult to simulate such problems [3–6].

In [7,8], the authors used various methods and schemes (IMPES, SS, etc.) in order to simulate multiphase, multicomponent flow in porous media. Other methods and schemes can be found in [9–12]. We could find Newton's method often in current research works in this area. Newton's method underlies many algorithms used to solve modeling problems. The compositional modeling of oil and gas reservoirs requires the construction and solution of large, rare, non-linear systems. The solution of sparse linear systems is the most time-consuming part of many sciences and engineering applications; we usually spend over 70% of the time on solving linear systems obtained from Newton's methods [8].

Linear solvers and preconditioners have been studied for decades, and various methods have been developed [13–15]. Saad et al. developed the linear solver of the generalized minimal residuals method (GMRES), which is a general-purpose solver for asymmetric linear systems [16,17]. Vinsome developed the ORTHOMIN solver (a variant of the generalized coupled residuals method with incomplete orthogonalization), which was developed initially for reservoir modeling [18]. The BICGSTAB solver (stabilized bis-conjugate gradient method) is also widely used for oil modeling. We could apply these linear solvers of the Krylov subspace methods to any kinds of linear systems. We can understand the main idea of the Krylov subspace method as the process of recursively constructing the residual vector for each iteration step. Let the linear algebraic equations of the solution be:

$$Ax = b \tag{1}$$

Take $K_m = K_m(A, r_0) = span\{r_0, Ar_0, A^2r_0, \ldots, A^{m-1}r_0\}$ where $r_0 = b - Ax_0$, and x_0 is the initial solution, looking for an approximate solution x_m from $x_0 + K_m$, so that the residual vector is orthogonal to another subspace L_m, that is, $r_m = b - Ax_m \perp L_m$, and then K_m is the Krylov subspace, and the above method is called the Krylov subspace method.

The GMRES algorithm [17] is one of the popular Krylov subspace methods, which is used to solve the solution of asymmetric linear equations. GMRES has very small residuals, but as the number of iterations increases, the amount of calculation and storage increases linearly, which is not efficient. So, the commonly used restarted GMRES, denoted by GMRES(m) [19], performs m iterations of GMRES, and then the resulting approximate solution is used as the initial guess to start another set of m iterations. This process repeats until the residual norm is small enough. The group of m iterations between successive restarts is referred to as a cycle, and m is referred to as the restart parameter. We indicate the restart cycle number with a subscript: x_i is the approximate solution after i cycles, or $m \times i$ total iterations, and r_i is the corresponding residual. The shortcoming of this method is that it reduces the robustness of the original GMRES algorithm and cannot guarantee the convergence of the algorithm. In addition, due to the fact that the restarted algorithm convergence may become very slow, this may affect the good convergence of the GMRES algorithm on specific issues due to the selection problem of m.

Besides the restarted GMRES(m) algorithm, another way to reduce the number of iterations of the GMRES method is that we use preconditioning as an explicit or implicit modification to the system of linear equations. Therefore, similar to other iterative methods, we usually combine the GMRES algorithm with a preconditioner for accelerating convergence. We could effectively and easily realize the incomplete LU factorization for sparse matrices with no fill or ILU(0) to obtain the preconditioning matrices [16,17,20,21]. Incomplete LU factorization assumes that sparse matrix A is represented in the form A = LU + R, where the matrices L and U are lower and upper triangular matrices, respectively, and the matrix R is the residual of the decomposition. After the ILU factorization is performed, the matrix M = LU can be used as a preconditioner. ILU factorization is also the basis of many preconditions, such as domain decomposition methods.

This article considers the mathematical model describing the process of multiphase, multicomponent fluid flow linearized using Newton's method in a simulation, and we solve the system of the algebraic equation with one of the Krylov subspace methods—the Newton-GMRES method. We especially consider a one-dimensional problem of multicomponent filtration of a liquid in a porous medium with the Newton-GMRES method. We solve and analyze the problem with the pure GMRES method, restarted GMRES(m) and preconditioned ILU(0)-GMRES method.

2. Mathematical Modeling

The considered model of multiphase, multicomponent flow in porous media makes it possible to simulate such modern methods of hydrocarbon production as the injection of coolant into an oil reservoir, combustion, injection of chemical reagents (polymer, surfactant) and others.

Let us consider a mathematical model of the multicomponent flow of three-phase (w—water, o—oil, g—gas) fluid in a porous medium. We write the mass conservation law for each component [8]:

$$\frac{\partial(\phi\, \xi_w S_w)}{\partial t} + \nabla \cdot (\xi_w \mathbf{u}_w) = q_w, \tag{2}$$

$$\frac{\partial\left(\phi\left(x_{mo}\xi_o S_o + x_{mg}\xi_g S_g\right)\right)}{\partial t} + \nabla\cdot\left(x_{mo}\xi_o\mathbf{u}_o + x_{mg}\xi_g\mathbf{u}_g\right) = q_m \tag{3}$$

$m = 1.2, \ldots N_c$, where N_c—the number of components; ϕ—the porosity of porous media; $x_{mo},\ x_{mg}$—the mole fraction of component m in the oil and gas phases; $\xi_w,\ \xi_o,\ \xi_g$—the molar density of the water, oil and gas phases; $S_w, S_o,\ S_g$—the saturation of the water, oil and gas phases; $q_w,\ q_m$—the molar velocities of the water flow and m-th component, respectively. In Equations (2) and (3), the flow rates $\mathbf{u}_\alpha$ are given by the Darcy law:

$$\mathbf{u}_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha}\mathbf{k}(\nabla p_\alpha - \rho_\alpha\wp\nabla z),\ \alpha = w, o, g \tag{4}$$

where $\wp$—the gravitational acceleration; ρ_α—the pressure of phase α.

In addition to the differential Equations (2)–(4), the following algebraic expressions hold:

$$\sum_{m=1}^{N_c} x_{mo} = 1,\ \sum_{m=1}^{N_c} x_{mg} = 1 \tag{5}$$

$$S_w + S_o + S_g = 1 \tag{6}$$

$$p_{cow} = p_o - p_w,\ p_{cgo} = p_g - p_o \tag{7}$$

where p_{cow}—the capillary pressure between the oil and water phases; p_{cg}—the capillary pressure between the gas and oil phases.

For the closure of the system (1)–(7), we set the conditions for the equilibrium relation below, which is described by Gibbs law for the composition system:

$$f_{mo}(p_o, x_{1o}, x_{2o,}, \ldots, x_{N_co}) = f_{mg}\left(p_g, x_{1g}, x_{2g}, \ldots, x_{N_cg}\right), \tag{8}$$

$$f_{m\alpha} = p_\alpha x_{m\alpha}\varphi_{m\alpha},\ m = 1.2, \ldots N_c, =, o, g \tag{9}$$

where $\varphi_{m\alpha}$—the fugacity coefficient of component m in phase α which is oil or gas, and f_m and f_{mg} are fugacity functions of m-th component in the gas and liquid phases, respectively.

We described the distribution of chemical components in the hydrocarbon phase by the K-value method (instantaneous evaporation coefficient) and determined each component by the following expression:

$$K_m = x_{mg}/x_{mo}\ ,\ m = 1, 2, \ldots, N_c$$

We described the thermodynamic behavior of fluids under reservoir conditions using Peng–Robinson equations of state [22], in which the relationship between pressure, temperature and phase volume is determined.

Initial and boundary conditions of system (2)–(9) in a one-dimensional field are as follows [0, 1]:

$$S_w(x,0) = S_w^0(x)\ S_o(x,0) = S_o^0(x)\ p_o(x,0) = p_o^0(x)$$

$$p_o(0,t) = p^{inj}\ p_o(1,t) = p^{prod}$$

$$S_w(0,t) = {S_w}^{inj}\ \frac{\partial S_w}{\partial x}(1,t) = 0$$

$$S_o(0,t) = 0\ \frac{\partial S_w}{\partial x}(1,t) = 0$$

where *inj*—injection; *prod*—production. $S_g,\ p_w, p_g$ are calculated using (5) and (6).

3. Numerical Method

In the considered system of differential equations, several components in the hydrocarbon system are provided. From the computational point of view, the computation time and the number of operations for solving this system increase with increasing the number of components. Therefore, it is necessary to group several components with the same chemical compositions into pseudo-components. We wrote the system of equations for three phases and two components as below (neglecting gravitational forces):

$$\begin{gathered}
\phi\frac{\partial(\xi_w S_w)}{\partial t} + div(\xi_w \mathbf{u}_{\mathrm{w}}) = q_w \\
\phi\frac{\partial\left(x_{1o}\xi_o S_o + x_{1g}\xi_g S_g\right)}{\partial t} + div\left(x_{1o}\xi_o \mathbf{u}_{\mathrm{o}} + x_{1g}\xi_g \mathbf{u}_{\mathrm{g}}\right) = q_1 \\
\phi\frac{\partial\left(x_{1o}\xi_o S_o + x_{2g}\xi_g S_g\right)}{\partial t} + div\left(x_{2o}\xi_o \mathbf{u}_{\mathrm{o}} + x_{2g}\xi_g \mathbf{u}_{\mathrm{g}}\right) = q_2 \\
\mathbf{u}_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha}\mathbf{k}\nabla p_\alpha, \qquad \alpha = o,\ g,\ w \\
x_{1o} + x_{2o} = 1,\ x_{1g} + x_{2g} = 1,\ S_o + S_w + S_g = 1 \\
p_{cow} = p_0 - p_w,\ p_{cgo} = p_g - p_0 \\
f_{1o}(p_o,\ x_{1o},\ x_{2o}) = f_{1g}\left(p_g,\ x_{1g},\ x_{2g}\right),\ f_{2o}(p_o,\ x_{1o},\ x_{2o}) = f_{2g}\left(p_g,\ x_{1g},\ x_{2g}\right)
\end{gathered} \tag{10}$$

System of Equation (10) has dependent variables (x_1o, x_2o, x_1g, x_2g, u_α, p_α, S_α). By changing variables and using several arithmetic operations (described in detail in the article [23]), you can obtain a nonlinear system of 6 equations with 6 unknowns (x_1o, L, z, F, S, p), where F—the total mass variable; L and V—the mass fractions, z_m—the total mole fraction:

$$F = \xi_o S_o + \xi_g S_g$$

$$L = \frac{\xi_o S_o}{F},\ V = \frac{\xi_g S_g}{F}$$

$$z_m = Lx_{mo} + (1-L)x_{mg},\ m = 1,2,\ldots,N_c$$

This system is linearized by Newton's method. For function v, we set:

$$v^{n+1,l+1} = v^{n+1,l} + \Delta v$$

where l refers to the number of Newton's iteration, and Δv represents the increment in this iteration step; n—the time approximation level.

In the (l + 1)-th layer of the Newton iterative process, the value of the unknowns is updated by the following law:

$$\begin{gathered}
x_{mo}^{n+1,l+1} = x_{mo}^{n+1,l} + \Delta x_{mo}^{n+1,l+1},\ m = 1,2,\ \ldots, N_c - 1, \\
z_{mo}^{n+1,l+1} = z_{mo}^{n+1,l} + \Delta z_{mo}^{n+1,l+1},\ m = 1,2,\ \ldots, N_c - 1, \\
L^{n+1,l+1} = L^{n+1,l} + \Delta L^{n+1,l+1}, \\
F^{n+1,l+1} = F^{n+1,l} + \Delta F^{n+1,l+1}, \\
S^{n+1,l+1} = S^{n+1,l} + \Delta S^{n+1,l+1}, \\
p^{n+1,l+1} = p^{n+1,l} + \Delta p^{n+1,l+1}
\end{gathered} \tag{11}$$

The calculation algorithm is as follows [23]:

1. Equation (11) is substituted into the finite difference form of a system of nonlinear equations of 6 variables.
2. We obtain a system of linear equations with unknown Δp_i and solve it using the GMRES method.
3. Further, using the known Δp_i, we find the unknown Δx_{1oi} (Δx_{2oi} is found by relation (5)), ΔL_i, Δz_i, ΔF_i, ΔS_i.
4. Next, Δx_{1gi} and Δx_{2gi} are calculated according to (5).

5. Next, the unknowns x_{1oi}, L_i, z_i, F_i, S_i, are found using (10).

As mentioned earlier, the resulting system of linear equations was also solved by the ILU(0)-GMRES method. There is a limitation in the incomplete LU decomposition; namely, problems may arise when the diagonal element of the matrix is equal to zero. This problem can be solved by selecting the reference element or, in other words, by pivoting [24]. In our problem, the diagonal elements of the matrix had a nonzero value, and finding the preconditioner did not require additional operations.

In the Figure 1, you can see a diagram of the computational algorithm.

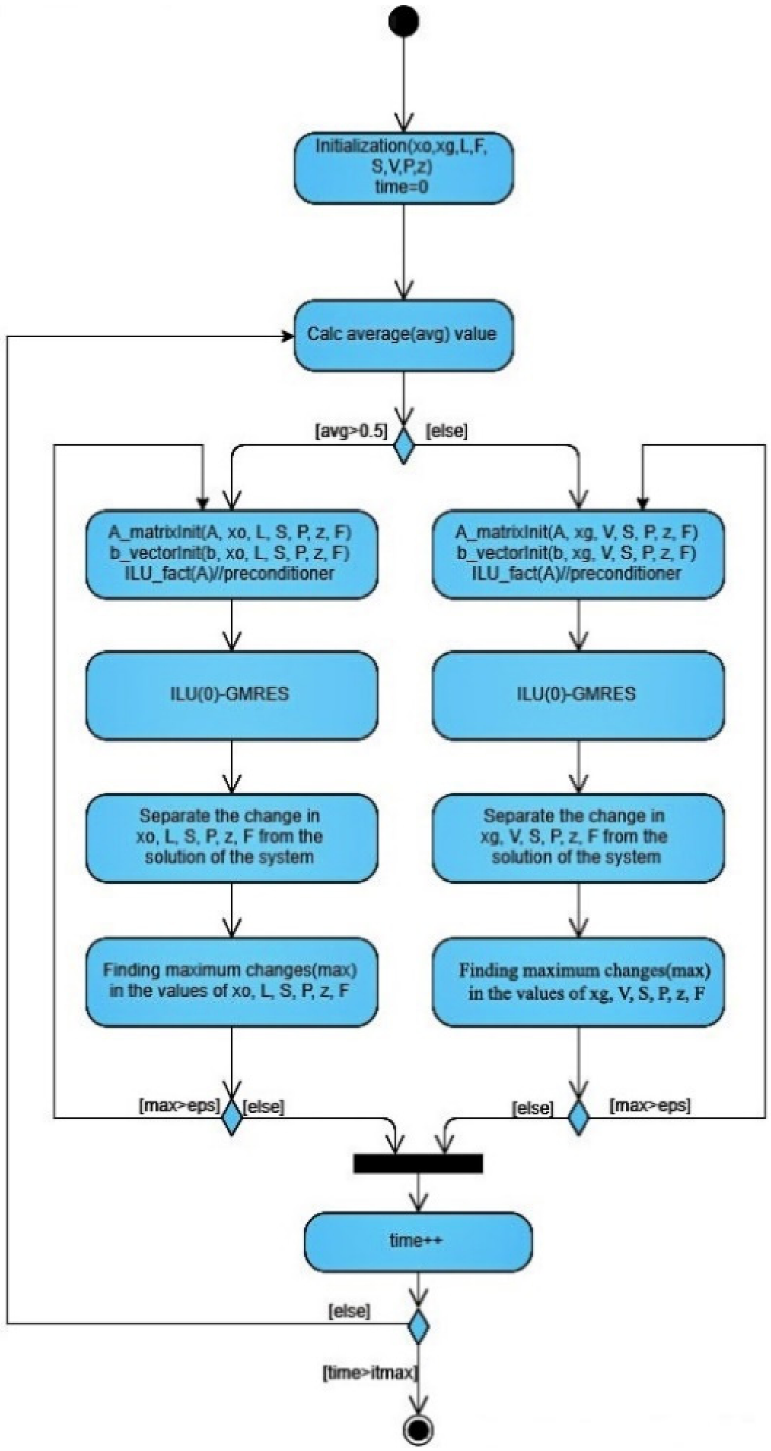
Initialization(xo,xg,L,F, S,V,P,z) time=0
Calc average(avg) value
[avg>0.5]
[else]
A_matrixInit(A, xo, L, S, P, z, F) b_vectorInit(b, xo, L, S, P, z, F) ILU_fact(A)//preconditioner
A_matrixInit(A, xg, V, S, P, z, F) b_vectorInit(b, xg, V, S, P, z, F) ILU_fact(A)//preconditioner
ILU(0)-GMRES
ILU(0)-GMRES
Separate the change in xo, L, S, P, z, F from the solution of the system
Separate the change in xg, V, S, P, z, F from the solution of the system
Finding maximum changes(max) in the values of xo, L, S, P, z, F
Finding maximum changes(max) in the values of xg, V, S, P, z, F
[max>eps]
[else]
[else]
[max>eps]
time++
[else]
[time>itmax]

Figure 1. Activity diagram of calculation process.

4. Results and Discussion

To find Δp_i, the algorithms GMRES, restarted GMRES(m) and preconditioned ILU(0)-GMRES were used. The matrix of coefficients of this system of linear equations is shown in Table 1 for various mesh sizes. The characteristics of the computer are the processor with 4-core Intel Core i7 3770 and RAM with 8 GB. In order to perform the calculation, we created six different dimensional matrices as shown in Table 1, which shows the number of points in each matrix and the number of nonzero elements. We have obtained a comparative graph of the time for calculating one Newton-GMRES iterative calculation (Figure 2) and the number of the GMRES inner iteration (Figure 3). In Table 2, we can see the acceleration obtained by preconditioning.

Table 1. Matrices information for numerical experiments.

Matrices Number	#1	#2	#3	#4	#5	#6
Number of nodes	3,240,000	1,2960,000	29,160,000	81,000,000	129,960,000	190,440,000
Nonzero elements	8338	16,738	25,138	41,938	53,138	64,338

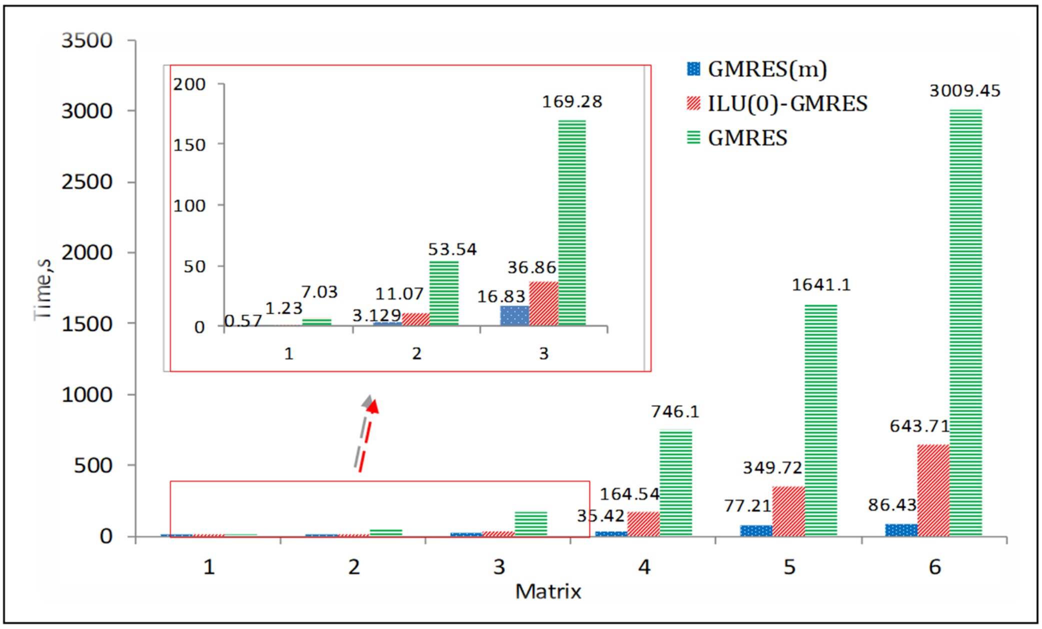

Figure 2. CPU time comparison of GMRES, GMRES(m) and ILU (0)-GMRES for performing one complete Newton-GMRES iteration.

Table 2. Acceleration obtained by preconditioning.

Matrices Number	GMRES Runtime(s)	ILU(0)-GMRES Runtime(s)	Acceleration
#1	7.03	1.23	5.715447154
#2	53.54	11.07	4.836495032
#3	169.28	36.86	4.592512208
#4	746.1	164.54	4.534459706
#5	1641.1	349.72	4.692611232
#6	3009.45	643.71	4.675164282

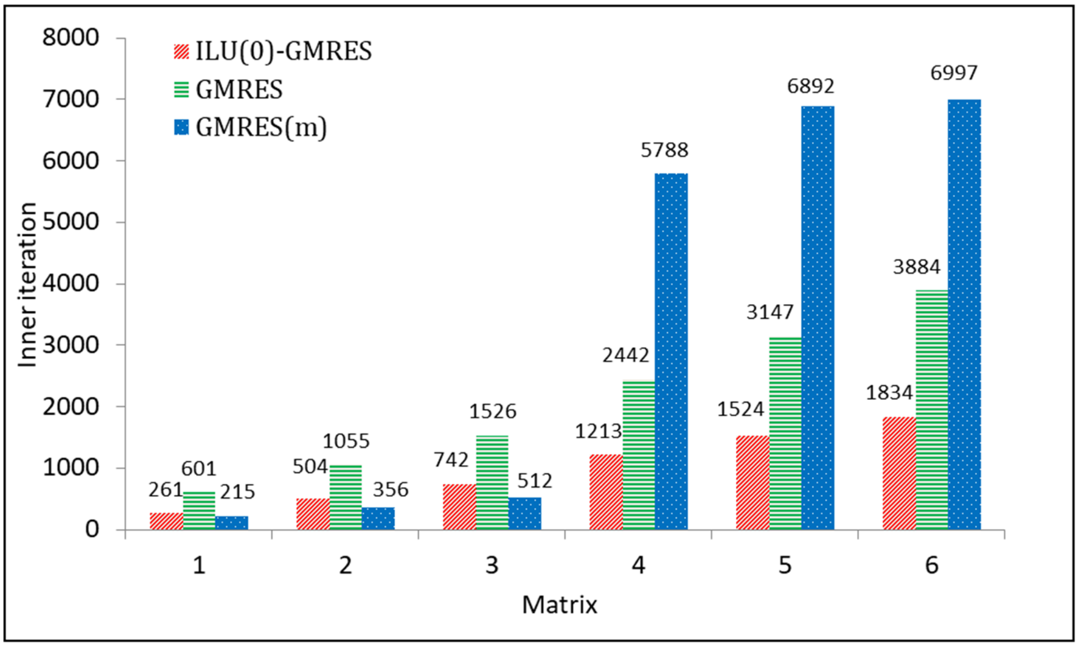

Figure 3. The number of inner iterations for the convergence of the GMRES, GMRES(m) and ILU(0)-GMRES methods.

As one can see, the GMRES algorithm with ILU(0) preconditioning works 4.5–5 times faster than the algorithm without preconditioning for all matrices. This result allows us to conclude that this speedup will persist as the size of the task increases

As one can see from Figure 2, the GMRES(m) algorithm takes less time than the other two computing algorithms. However, it was selected experimentally by repeatedly giving values ranging from 100 to 1500 to obtain the appropriate value of six different matrices. For example, experimental calculations for matrix #3 are shown in Table 3; as we have seen from this table, when m = 500, algorithms obtain the minimum computational time and number of iterations, and for matrix #4, as can be seen from Table 4, the algorithm is executed faster when m = 100.

Table 3. Numerical experiments for matrix 3 with different m values.

Matrix #3	CPU time(s)	Residual	Inner iteration
GMRES(100)	23.374	0.00982457	5286
GMRES(200)	56.605	0.00997898	5502
GMRES(300)	49.395	0.0099962	2942
GMRES(400)	50.373	0.00999233	2184
GMRES(500)	**16.83**	**0.00994755**	**512**
GMRES(600)	24.017	0.00939816	604

Table 4. Numerical experiments for matrix 4 with different m values.

Matrix #4	CPU time(s)	Residual	Inner iteration
GMRES(100)	**35.4249**	**0.00993622**	**5788**
GMRES(200)	184.447	0.00993313	10513
GMRES(300)	281.269	0.00999677	10751
GMRES(400)	200.089	0.00999497	5471
GMRES(500)	218.66	0.00999747	4473
GMRES(600)	260.9249	0.00999968	4435

As can be seen from Table 4, in some cases, the total number of iterations is fewer than when m = 100, but the algorithm is slower. This is due to the fact that at m = 100, at each restart, the program builds a smaller orthogonalization basis.

The appropriate value of m for the matrix of Table 2 is given in Table 5. As we have noted, in different matrix sizes, the value of m for the GMRES(m) algorithm is different. However, the optimal m value of matrix #4, #5 and #6 is 100, but with substantial inner iterations (Figure 3). Although the ILU(0)-GMRES algorithm is slower than the GMRES(m) algorithm, there is no need to carry out repeated experiments to obtain the optimal value of m. Moreover, with the ILU(0)-GMRES algorithm, we had better convergence then GMRES(100) for matrix #4, #5 and #6. Moreover, the GMRES algorithm is the slowest and most assured the number of iterations.

Table 5. Appropriate m value for different size of matrices.

Matrices Number	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**
Appropriate m value	200	300	500	100	100	100

5. Conclusions

This article discusses the Newton-ILU0-GMRES method for solving the problem of multiphase, multicomponent flow in a porous medium. The equations describing the motion of a multicomponent, multiphase flow in a porous medium were linearized using Newton's method. The resulting system of linear equations was solved using the GMRES algorithm. In addition, a preconditioner and a modification of the method with restarts (GMRES(m)) were used to speed up the method. The analysis and comparison (calculation time, convergence) of the Newton-GMRES, GMRES(m) and ILU0-GMRES methods were carried out. The numerical results demonstrate that for the one-dimensional problem of multicomponent flow in a porous medium, the GMRES algorithm is the most time-consuming. The GMRES algorithm with ILU(0) preconditioning works 4.5–5 times faster than the algorithm without preconditioning. For the problem size equal or bigger than matrix #4 for the GMRES(m) algorithm, the suitable m value is 100, but convergence is slow. The ILU(0)-GMRES algorithm calculated slower than the GMRES(m) at the larger size of the matrix but with better convergence in solving the given problem.

Similar results can be obtained for two-dimensional and three-dimensional equations by applying to them an analogous method of solving, but with the condition that the matrices of the resulting systems of linear equations satisfy the constraints of Incomplete LU factorization.

Author Contributions: Conceptualization, methodology, T.I. and B.M.; software, validation, D.L. and N.K.; formal analysis, T.I., B.M., D.L. and B.D.; investigation, resources, data curation, T.I., B.M. and B.D.; writing—original draft preparation, T.I., B.M. and N.K.; writing—review and editing, D.L. and B.D.; visualization, B.M. and N.K.; supervision, project administration, funding acquisition, T.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Education and Science of the Republic of Kazakhstan, grant number AP09563516.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pederson, K.S.; Christensen, P.L. *Phase Behavior of Petroleum Reservoir Fluids*; CRC Press: Boca Raton, FL, USA, 2006; p. 422.
2. Ahmed, T. *Reservoir Engineering Handbook*, 3rd ed.; Gulf Professional Publishing: Oxford, UK, 2006; p. 1376.
3. Maurand, N.; Mayoura, S.; Parra, T.; Araujo, C.; Furtado, C. Coupling Compositional Flow, Thermal Effects and Geochemistry Reactions when Injecting CO_2 in a Carbonated Oil Field. *Energy Procedia Elsevier* **2014**, *51*, 316–325. [CrossRef]
4. Farshidi, S.; Fan, Y.; Durlofsky, L.; Tchelepi, H. Chemical Reaction Modeling in a Compositional Reservoir-Simulation Framework. *Soc. Pet. Eng.* **2013**. [CrossRef]

5. Delshad, M.; Thomas, S.G.; Wheeler, M.F. Parallel numerical reservoir simulations of non-isothermal compositional flow and chemistry. *Soc. Pet. Eng.* **2011**, *16*, 239–248. [CrossRef]
6. VBorisov, E.; Kritskiy, B.V.; Marchenko, N.A.; Mitrushkin, D.A.; Savenkov, E.B. *Nonisothermal Compositional Flow Model with Chemical Reactions and Active Solid Phase for Reservoir Simulation*; Keldysh Institute of Applied Mathematics: Moscow, Russia, 2013; Volume 91.
7. Chen, Z. *Reservoir Simulation: Mathematical Techniques in Oil Recovery*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2006.
8. Chen, Z.; Huan, G.; Ma, Y. *Computational Methods for Multiphase Flows in Porous Media*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2006. [CrossRef]
9. Aceto, L.; Pandolfi, R.; Trigiante, D. One parameter family of linear difference equations and the stability problem for the numerical solution of ODEs. *Adv. Differ. Equ.* **2006**, *2006*, 019276. [CrossRef]
10. Rees, D.A.S. Nonlinear Convection in a Partitioned Porous Layer. *Fluids* **2016**, *1*, 24. [CrossRef]
11. Pawar, S.; San, O. CFD Julia: A Learning Module Structuring an Introductory Course on Computational Fluid Dynamics. *Fluids* **2019**, *4*, 159. [CrossRef]
12. El Ouafa, M.; Vincent, S.; Le Chenadec, V. Monolithic Solvers for Incompressible Two-Phase Flows at Large Density and Viscosity Ratios. *Fluids* **2021**, *6*, 23. [CrossRef]
13. Lacroix, S.; Vassilevski, Y.; Wheeler, J.; Wheeler, M. Iterative Solution Methods for Modeling Multiphase Flow in Porous Media Fully Implicitly. *SIAM J. Sci. Comput.* **2003**, *25*, 905–926. [CrossRef]
14. Vabishchevich, P.N.; Vasilyeva, M.V. Iterative methods for solving the pressure problem at multiphase filtration. *arXiv* **2011**, arXiv:1107.5479.
15. Wang, B.; Wu, S.; Li, Q. Applications of BILU0-GMRES in reservoir numerical simulation. *Acta Pet. Sin.* **2013**, *34*, 954–958.
16. Barrett, R.; Berry, M.; Chan, T.F.; Demmel, J.; Donato, J. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd ed.; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1994.
17. Saad, Y. *Iterative Methods for Sparse Linear Systems*, 2nd ed.; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2003.
18. Vinsome, P.K.W. Orthomin, an iterative method for solving sparse sets of simultaneous linear equations. *Soc. Pet. Eng.* **1976**. [CrossRef]
19. Saad, Y.; Schults, M. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.* **1986**, *7*, 856–869. [CrossRef]
20. Klie, H.; Sudan, H.; Li, R.; Saad, Y. Exploiting capabilities of many core platforms in reservoir simulation. *SPE RSS Reserv. Simul. Symp.* **2011**, 21–23. [CrossRef]
21. Mittal, R.C.; Al-Kurdi, A.H. An efficient method for constructing an ILU preconditioner for solving large sparse nonsymmetric linear systems by the GMRES method. *Comput. Math. Appl.* **2003**, *185*, 391–403. [CrossRef]
22. Peng, D.Y.; Robinson, D.B. A new two-constant equation of state. *Ind. Eng. Chem. Fundam.* **1976**, *15*, 59–64. [CrossRef]
23. Chen, Z.; Huan, G.; Wang, H. Computer Simulation of Compositional Flow Using Unstructured Control Volume Finite Element Methods. *Computing* **2006**, *78*, 31–53. [CrossRef]
24. Chow, E.; Saad, Y. Experimental study of ILU preconditioners for indefinite matrices. *J. Comput. Appl. Math.* **1997**, *86*, 387–414. [CrossRef]