An Efficient Reservoir Simulation by The Successive Explicitization Process

Hiroyuki TOSAKA* and Yukihiro MATSUMOTO**

(Received March 15, Accepted May 18, 1987)

Abstract: This paper presents a mathematical method named "Successive Explicitization Process (SEP)" that realizes considerable reduction in computing time required for the numerical reservoir simulation. The SEP attains the aimed reduction by successive and complete explicitization of the solution on a grid block basis during the multi-unknown Newton-Raphson iterative process. It economizes the reservoir simulation by not only the fully implicit technique but IMPES or SEQ substantially. The SEP could be used as a strong alternative of Adaptive Implicit Method proposed by G.W. THOMAS and D.H. THURNAU.

Described here are the mathematical treatment of SEP and some example results which show its validity.

1. Introduction

Backed by the advancement in computer technology, the up-to-date reservoir simulators tend to employ the fully implicit technique that offers robust stability to users who are engaged in actual reservoir simulation. The superiority of the fully implicit technique, however, is restrained by serious increase of the requirements of computing work and storage on main memory of CPU compared with IMPES or SEQ type formulation, especially when the objective reservoir problem becomes large. In general field-wide reservoir problems, fully implicit simulation cannot cope with IMPES or SEQ in terms of computing time or cost without making its inherent stability the most of it by taking sufficiently larger time interval than the other two.

The purpose of the study here is to augment the attraction and applicability of the fully implicit technique by resolving the part of the demerits incurred on it as mentioned above.

Historically, many researches were undertaken to make the performance of the simulation more economical. One direction has been to improve the technique of matrix solution in which the significant portion of the algebraic work required

for simulation is consumed. As represented by the works on fast matrix solvers, Incomplete Factorization, Nested Factorization and other iterative solvers showed the superiority over the direct ones in most of the problems. Hence, utilizing such fast solver is the one way for economization. On the other hand, G. W. THOMAS and D. H. THURNAU proposed a technique called Adaptive Implicit Method (AIM). AIM is said to be able to reduce both computing time and required storage, in which the degree of implicitness of the grid block is adaptively shifted in space and time. Here, the term "implicitness" means how many variables for a grid block are solved simultaneously.

SEP proposed here could be lined in the latter but differs from AIM. Comparing the both methods, it can be said that AIM reduces the implicitness of the grid block, hence, the problem, at most to first-degree adaptively, while SEP reduces the size of the problem by forcing the implicitness of certain grid blocks to zero-degree, i.e., complete exclusion from the calculation process. More detailed comparison between the SEP and AIM will be done in the later section.

The SEP offers considerable reduction of CPU time with essentially a minor modification of matrix solver and Jacobian make-up process. Although SEP was originally aimed to apply to

* The University of Tokkyo

** Japan Oil Engineering Co., Ltd.

fully implicit simulator, it can work on IMPES or SEQ type simulation, in which first-degree implicitness is used, for improving the performance where AIM actually cannot.

2. Mathematical Treatment

2.1 Conceptual Derivation of SEP

For convenience, the flow equation which describes the petroleum reservoir problem is written in residual form after finite differencing as follows,

$$r_i^m = Qf_i^m - Qw_i^m - (\Delta Qac)_i^m / \Delta t = 0 \quad (1)$$

where,

Qf : flux between two adjacent blocks

Qw : source/sink term

Δt : discretised time slice

ΔQac : increment of accumulation during Δt

i : discretized indicator of the grid block

m : m -th flow equation, $m=1, \dots, n$.

The flux Qf_i^m actually consists of the terms shown below.

$$Qf_i^m = qf_{i,x-}^m + qf_{i,x+}^m + qf_{i,y-}^m + qf_{i,y+}^m + qf_{i,z-}^m + qf_{i,z+}^m \quad (2)$$

where, qf is the flux through each interface of the block i with the neighboring grid blocks, and six subscripts such as x^- , x^+ , etc., denote that fluxes are summed in six directions of the discretized three dimensional space (7-point finite differencing for a three dimensional problem is assumed).

For the system with n unknowns per grid block, eq. (1) can be expressed in vector form as,

$$\bar{R}_i = (r_i^1, r_i^2, \dots, r_i^n)^t \quad (3)$$

$$\bar{X}_i = (x_i^1, x_i^2, \dots, x_i^n)^t \quad (4)$$

where, r_i^m and x_i^m are the m -th residual and unknown of i -th grid block respectively, and the symbol crowned by bar and superscript t denote a vector and vector transposition (i.e., column vector), respectively.

If the objective reservoir is discretised into N blocks, the overall residual vector and solution vector become as follows.

$$\bar{\Omega} = (\bar{R}_1, \bar{R}_2, \dots, \bar{R}_N)^t \quad (5)$$

$$\bar{X} = (\bar{X}_1, \bar{X}_2, \dots, \bar{X}_N)^t \quad (6)$$

What is sought in the simulation is to obtain $\bar{X}$ that satisfies,

$$\bar{\Omega} = 0 \quad (7)$$

for the specified time slice.

As $\bar{\Omega}$ is a non-linear function of $\bar{X}$, some iterative technique is needed. Commonly used is the Newton-Raphson iterative scheme, in which non-

linearity is overcome by the following procedure,

$$J^k \cdot \delta \bar{X}^{k+1} = -\bar{\Omega}^k \quad (8)$$

where, J is the Jacobian matrix which consists of the partial derivatives of $\bar{\Omega}$ with respect to each solution variable. For example, the element of the k -th equation of i -th block is,

$$\Delta r_i^k / \Delta x_j^m \quad (m=1, \dots, n, \quad j=1, \dots, N) \quad (9)$$

where, Δ denotes numerical differentiation.

In Newton-Raphson method, nonlinear solution is found stepwisely, and most often the absolute value of the solution is the largest in the first iteration and becomes smaller quadratically in subsequent iterations.

Starting with known solution at L -th time step, if after K iterations termination criteria are satisfied, the numerical solution for the $(L+1)$ st time step is obtained as,

$$\bar{X}^{L+1} = \bar{X}^L + \sum_{k=1}^K \delta \bar{X}^k \quad (10)$$

When applied to a multi-unknown system like reservoir problem, it is expected that the solution of respective block converges at different iteration level, that is, the convergence rate is never uniform for all the solutions due to the different degree of non-linear effect.

In reservoir simulation, severe non-linearity is caused by source/sink terms in well blocks and their neighbours, while the majority of the blocks are under modest or linear change. This fact characterizes the performance of the Newton-Raphson iterative scheme, in which large number of unknowns with different non-linearity are solved simultaneously. This means that the solution of the grid blocks located distant from well blocks can be found after one or two iterations because of small non-linearity, and the fluctuation of the solution there will become substantially smaller in subsequent iterations. On the other hand, obtaining the solution of the well block and its neighbours needs more iterations because of serious non-linearity.

The facts mentioned above suggest us that, for practical purposes, simultaneous solution for all the unknowns on every Newton-Raphson iteration is not required, in other words, an excessive burden could be saved.

2.2 Mathematical Practices of SEP

Practically in SEP, the absolute value of the solution of each grid block is checked on every Newton-Raphson iteration. If it is less than the

given tolerance, the solution is locked (fixed) at the latest iteration level and the block is marked as "locked". In subsequent Newton-Raphson iterations, the flow equations for the "locked" blocks are completely excluded from the calculation. As a result, both the process of preparing Jacobian and residual vector, and the process of matrix solution are all reduced in size and work.

For brevity, the solution of the block $(i-1)$ at K -th iteration is assumed to become less than the given tolerance for locking as,

$$|\delta x_{i-1}^m| \leq \varepsilon^m \quad (m=1, \dots, n) \quad (11)$$

where, ε^m is the tolerance of the solution for m -th unknown. Then, the solution for the $(L+1)$ st step can be fixed as,

$$\bar{X}_{i-1}^{L+1} = \bar{X}_{i-1}^L + \sum_{k=1}^K \delta \bar{X}_{i-1}^k \quad (12)$$

Because by this iteration stage, the solution of the surrounding blocks of $(i-1)$ are also expected to be close to but still slightly greater than the tolerance, the fluctuation of the flux between $(i-1)$ and any of adjacent blocks in subsequent iteration can be neglected. Therefore, for example, for the flux between $(i-1)$ and (i) , it is fixed as shown in eqs. (13) and (14).

$$q_{i,x-}^{m,(L+1)} = q_{i,x-}^{m,(K)} \quad (13)$$

$$q_{i-1,x+}^{m,(L+1)} = q_{i-1,x+}^{m,(K)} \quad (14)$$

This approximation is very close to what is expected in ordinary fully implicit solution and has no material balance errors inherently because the in-and-out flow relevancy is exactly maintained by eqs. (13) and (14). After the above procedure is made, the block $(i-1)$ is excluded from any process concerned with iterative solution.

For the $(K+1)$ st iteration, the residual and Jacobian element of the block (i) which is not "locked" yet becomes as,

$$\begin{aligned} r_i^{j,(K+1)} = & q f_{i,x-}^{j,(L+1)} + q f_{i,x+}^{j,(K+1)} + q f_{i,y-}^{j,(K+1)} \\ & + q f_{i,y+}^{j,(K+1)} + q f_{i,z-}^{j,(K+1)} + q f_{i,z+}^{j,(K+1)} \\ & - Qw_i^{j,(K+1)} - \Delta Qac_i^{j,(K+1)} / \Delta t \\ & j=1, 2, \dots, n \end{aligned} \quad (15)$$

$$\begin{aligned} \Delta r_i^{j,(K+1)} / \Delta x_{i-1}^l = & \Delta q f_{i,x-}^{j,(K+1)} / \Delta x_{i-1}^l = 0 \\ & l=1, 2, \dots, n \end{aligned} \quad (16)$$

$$\Delta q f_{i-1,x+}^{j,(K+1)} / \Delta x_i^l = 0 \quad l=1, 2, \dots, n \quad (17)$$

Eq. (16) means that the Jacobian elements of the i -th block related to the unknowns of the $(i-1)$ th block are set to zero because of the locking executed on the $(i-1)$ th block as Eq. (12).

Figs. 1-(a) through (d) are presented to show illustratively how SEP reduces the matrix size on

Fig. 1 Illustration of successive explicitization process (SEP) for 5×5 homogeneous problem.

a simple and homogeneous problem.

As for locking tolerance, it is prospected that the smaller the value, the closer the solution by SEP becomes to that of ordinary fully implicit solution.

2.3 Matrix Preparation and Solution in SEP

The point whether SEP works well or not depends on the manipulation of matrix, i.e., preparation and solution.

Ordinary fully implicit simulator solves solution for all the active blocks on every Newton-Raphson iteration. But in SEP, the following calculations are excluded from matrix preparation and solution process.

- Calculation of PVT properties (FVF, viscosity, solution GOR) for "locked" blocks.
- Calculation of saturation functions (relative permeability, and capillary pressure)

Fig. 2 A work flow of SEP.

- for "locked" blocks.
- iii. Numerical differentiation with respect to solution variables of "locked" blocks.
- iv. Calculation of Jacobian elements relating to the "locked" blocks.
- v. Calculation of residuals for the "locked" blocks.
- vi. Algebraic operations on matrix elements relating to the "locked" blocks.

As the CPU time is mainly consumed on these processes, the performance of the simulation is much improved if the number of "locked" blocks becomes large. In Fig. 2, a flow diagram is presented to illustrate the whole process of the SEP.

For the solution of matrix equation, Nested Factorization method is used in this study, in which the columns and rows that cross on the "locked" blocks are completely neglected from preconditioning, solution and acceleration processes. The SEP can also be included in both direct solvers (Gaussian routine, D4 or band matrix routine) and iterative solvers (LSOR, SIP, ILU, ADI, etc.) without problem.

Table 1 Basic data listing for three-phase coning problem (after Reference 3)

<u>Geometry</u>	
Radial extent, ft	2,050
Wellbore radius, ft	0.25
Radial position of first block center, ft	0.84
Number of radial blocks	10
Radial block boundary, ft	0.25, 2.00, 4.32, 9.33, 20.17, 43.56, 94.11, 203.32, 439.24, 948.92, 2050.00
Number of vertical layers	15
Dip angle, degrees	0
Depth to top of formation, ft	9000
<u>Rock and Fluid Data</u>	
Pore compressibility, psi ⁻¹	4×10 ⁻⁶
Water compressibility, psi ⁻¹	3×10 ⁻⁶
Oil compressibility for undersaturated oil, psi ⁻¹	1×10 ⁻⁵
Oil viscosity compressibility for undersaturated oil, psi ⁻¹	0
Stock-tank oil density, lbm/cu ft	45.0
Stock-tank water density, lbm/cu ft	63.02
Standard condition gas density, lbm/cu ft	0.0702
<u>Initial Conditions</u>	
Depth of gas/oil contact, ft	9,035
Oil pressure at gas/oil contact, ft	3,600
Capillary pressure at gas/oil contact, psi	0
Depth of water/oil contact, ft	9,209
Capillary pressure at water/oil contact, psi	0
<u>Well Data</u>	
Completed blocks*	(1,7)(1,8)
Permeability/thickness, m-ft	6,200 480
Skin	0 0
Minimum BHFP, psi	3,000
Pump depth, ft	9,110
<u>Production Schedule</u>	
Time Period(days)	Oil Production rate(stab/d)**
1 to 10	1000
10 to 50	100
50 to 720	1000
720 to 900	100

* The well is located at the inner boundary of each block.

** The production schedule is to be maintained until the BHFP is equal to the constraint value.

3. Test Results

To examine the validity of SEP, test runs were made by SEP version of VOL33S-V3, a fully implicit 3-D, 3-P black-oil reservoir simulator, developed by and in use in Japan Oil Engineering Co., Ltd.

The problem used here was presented by H. G.

Fig. 3 Examples of typical gridwise performance of SEP in three-phase coning problem.

WEINSTEIN, J. E. CHAPPELEAR, and J. S. NOLEN for SPE's second comparative solution project. The basic input data of the problem is briefly presented in Table 1. Due to the main purpose of the simulation here mentioned above, the detail performance curves, requested in his paper to report for the participants of the project, are omitted.

Comparison was made between fully implicit solutions with and without SEP under following items.

- checking the possible existence and degree of difference in solution due to the locking process in SEP.
- checking gridwise performance of SEP.
- measuring the reduction rate of CPU time by SEP.

Criteria used in the test runs are as follows.

- Pressure convergence tolerance: 1.0psi
- Saturation convergence tolerance: 0.001

In Fig. 3, typical performances of the SEP are presented illustratively for specified time of the simulation. They show that the "locked" blocks firstly appear in remote periphery of the well and rapidly multiply in number toward the adjacent area of well blocks as the Newton-Raphson

Fig. 4 Decrease of no. of implicit blocks with respect to Newton-Raphson iteration by SEP.

iteration proceeds. Finally when the solutions of all the blocks satisfy the convergence criteria, they are all "locked" and the calculation for the time step ceases. Fig. 4 shows the decrease of the number of implicit or "unlocked" blocks against the number of Newton-Raphson iteration, at several

Table 2 Comparison of the solutions with and without SEP for three-phase coning problem

Time (days)	Maximum absolute difference of the solutions		
	Pressure (psi)	Water saturation (frac.)	Gas Saturation (frac.)
50	0.033	1.41×10^{-6}	2.09×10^{-4}
720	2.23	2.11×10^{-5}	1.26×10^{-3}
900	2.42	4.05×10^{-5}	1.51×10^{-3}

Table 3 Comparison of the well performance with and without SEP for three-phase coning problem

Time (days)	Water cut (frac.)		Gas oil ratio (scf/stb)		Recovery (%)	
	ORD.	SEP	ORD.	SEP	ORD.	SEP
50	0.12	0.12	1394	1394	—	—
720	0.42	0.42	2938	2937	—	—
900	0.39	0.39	1342	1342	0.1834	0.1837

* ORD.—ordinary fully implicit solution

Table 4 Comparison of the total performance between ordinary fully implicit solution and SEP for three-phase coning problem

	Total No. of Newton-Raphson	Average No. of implicit blocks	Total CPU time
Ordinary fully implicit	1	1	1
SEP	0.87	0.44	0.55

* The values for SEP are normalized by the values for ordinary fully implicit solution.

time steps during 50 to 720 days. The feature of the decreasing curve of the number of the implicit blocks is quite steep, therefore, CPU time for a iteration becomes considerably less in later iterations.

Table 2 compares the possible difference of the solutions between ordinary fully implicit solution and the solution with the SEP. The maximum difference between the two is insignificant for every solution variable, whereas certain cumulative tendency is observed. At the end of simulation, i.e., at 900 days, the maximum difference observed is still negligible for practical purposes. Obtained well performance was almost identical as shown in Table 3.

The comparison of total performance between ordinary fully implicit solution and the SEP is

summarized in Table 4. The SEP reduced the total number of Newton-Raphson iterations slightly and average implicit blocks per iteration considerably. It resulted in 45% reduction of CPU time required for ordinary solution.

4. Comparison between SEP and AIM

The authors want to make some comments on our method in comparison with AIM, in order to clarify the difference between the two methods.

1. SEP uses "blockwise" explicitization technique, while AIM employs "unknownwise" explicitization. This very difference characterizes the structure and algorithm of the two methods. The SEP basically uses ordinary fully implicit code and block matrix solver (iterative or direct) with a minor modification. On the other hand, AIM needs to modify or reorder the Jacobian elements and special direct-like solver to solve it. Considering the ultimate efficiency of CPU time and storage, AIM may be superior to SEP. However, from the point of simplicity the SEP code is includable in most of the existing fully implicit simulator without problem.
2. SEP inherently contains no material balance errors because "blockwise" explicitization can be done consistently at the interface of "implicit" and "explicit" blocks. And this process seems to improve the stability and rate of convergence because insignificant numerical perturbation observed in implicit calculation process is eliminated by the explicitization. From our understanding, AIM may suffer from some degree of material balance errors because interfaces of different implicitness coexist in the system.
3. Using AIM, user may optimize the size of required main memory after several trial runs, while the SEP always needs the memory size required for fully implicit solution. According to reference 1, the maximum memory reduction of 35% from ordinary fully implicit code is realized in AIM when implicitness is equal to one. This relatively small value of reduction, compared with the CPU time reduction, is because saving is limited for the variable used for the Jacobian elements and related works. In

the SEP, such memory saving is basically difficult without reviewing the total code of the simulator. However, from the point of the current and future advancement in computer, we foresee that memory size itself is not a serious bottle-neck even for fully implicit code in a large problem.

5. Conclusions

Concluding the work presented, the following features of the SEP can be summarized.

1. SEP is a method that is devised to attain rapid solution through successive explicitization of the implicit unknowns by the use of differential convergence phenomenon observed in multi-unknown Newton-Raphson iteration.
2. SEP runs on fully implicit simulator with identical or more improved stability and material balance consistency the fully implicit solution provides.
3. SEP can be applicable to IMPES or SEQ simulator for improving their performances.
4. For the test problems, fully implicit solution with SEP reduced the cpu time to almost 55% of the ordinary solution.

6. Acknowledgements

The authors wish to appreciate Japan Oil Engineering Co., Ltd. for permission to publish this paper, and the colleague who were cooperative to make SEP into practical use by applying to large sophisticated problem.

Additionally, thanks have to be placed on the engineers of Kuwait Oil Company who gave the necessity and incentives to the authors for this work.

References

- 1) G. W. THOMAS and D. H. THURNAU 1983: "Reservoir Simulation Using an Adaptive Implicit Method", *SPEJ*, Oct., pp. 759-768.
- 2) A. S. ODEH, 1981: "Comparison of Solutions to a Three-Dimensional Black-Oil Reservoir Simulation Problem", *JPT*, Jan., pp. 13-25.
- 3) H. G. WEINSTEIN, J. E. CHAPPELEAR, and J. S. NOLEN, 1986: "Second Comparative Solution Project: A Three-Phase Coning Study", *JPT*, Mar., pp. 345-353.
- 4) J. R. APPELYARD, I. M. CHESHIRE, and R. K. POLLARD: "Special Techniques for Fully Implicit Simulators", *Numerical Method*, pp. 395-408.
- 5) VINSOME, P. K. W., 1976: "Orthomin, an Iterative Method for Solving Sparse Banded Sets of Simultaneous Linear Equations", *SPE* 5729, Feb.
- 6) W. I. BERTIGER and F. J. KELSEY, 1985: "Inexact Adaptive Newton Methods", *SPE* 13501.

要 旨

逐次陽化過程 (Successive Explicitization Process) による完全陰解法型油層シミュレーションの高速化手法
登坂 博行*・松本 行弘**

近年の油層シミュレーターの多くは無条件安定性を誇る完全陰解法を採用し、従来の IMPES や SEQ 型のシミュレーターの弱点であった、時間スライスの大きさに敏感な数値解の発散、を克服している。しかしこの方法は、コンピュータ上の計算時間と記憶容量を必然的に増大させるもので、ハードウェアの制約等がある場合、IMPES や SEQ の方がはるかに実用的で、対象問題が難しい場合に限りそのメリットが十分に発揮される。

ここでは完全陰解法のメリットを生かしながらそれに要する計算時間を大幅に短縮する数学的方法に関して述べる。

筆者らが“逐次陽化過程 (Successive Explicitization Process)”と名づけたこの方法では、複数の未知数を含む非線形系に対する Newton-Raphson 反復過程において観測される、各未知数の収束速度差に注目し、収束条件を満たす未知数から逐次整合的に固定 (陽化) し、引き続き反復過程から完全に掃き出してしまふ。これにより、反復過程の前進に従いそれに関与する未知数の個数は急激に減少し、各段階でマトリックス演算に必要とされる計算量も比例的に減少するため、計算時間の大幅な短縮が実現される。この方法は、G.W. THOMAS と D. H. THURNAU が考察した AIM (Adaptive Implicit Method) とは異なり、完全陰解法型シミュレーターに適用された時は完全陰解法と同一以上の安定性を保証し、しかもより一般的に IMPES や SEQ タイプのシミュレーターに適用した場合にも、それら固有の安定性を維持しつつ加速することが可能である。

本法の実用性のチェックは、H. G. WEINSTEIN, J. E. CHAPPELEAR, J. S. NOLEN が提出している三相コーニング問題を使って行われ、通常型完全陰解法に比し、約 2 倍の高速性を示した。

* 東京大学工学部資源開発工学科

** 日本オイルエンジニアリング㈱