AN EFFICIENT SOLUTION PROCEDURE FOR MULTIPLE INTERACTING CONTINUA FLOW

Mohan S. Seth* and Mineyuki Hanano**

* Technical Software & Engineering, Inc.
2506 Springwood Lane, Richardson, Texas 75082, U.S.A.
e-mail: mseth@metronet.com

** JMC Geothermal Engineering Co., Ltd.
72-2 Samasori, Ukai, Takizawa-mura, Iwate 020-01, Japan
e-mail: hanano@geothermal.co.jp

Key Words: linear equations, matrix decomposition, multiple interacting continua, reservoir simulation

ABSTRACT

Invocation of the Multiple Interacting Continua (MINC) concept in simulation of multi-phase fluid and heat flow in reservoirs entails solution of a very large set of sparse linear equations. The task is computationally burdensome and frequently impractical. We present here a simple and efficient procedure which results in a several-fold reduction in the computing time and the memory requirements. The procedure is based on decomposing the solution matrix into (a) a set of tridiagonal matrices arising from individual MINC blocks, and (b) a matrix arising from flow in the fractures. The procedure is ideally suitable for parallel processing computers such that the flow and accumulation coefficients, and solution for MINC blocks can be performed concurrently. Theoretical and actual computational performance parameters are presented.

1. INTRODUCTION:

A typical habitat of geothermal reservoir fluids is porous/fractured rock media located at a depth ranging from the surface to several thousand meters. In order to efficiently exploit this resource and to undertake development plans, a good understanding of the flow of fluid and energy in the porous/fracture rock is of paramount importance. Unfortunately, adequate characterization of individual fractures in terms of their spatial distribution, orientation, width, etc. is not feasible. Furthermore, mathematical simulation involving discrete fracture representation itself is computationally impractical except for a few idealized hypothetical systems. Such systems are primarily used as a research tool for better understanding of the processes involved.

In order to circumvent the above drawbacks, a simplified model for flow in fractures and rock matrix was first proposed by Soviet scientists (Barenblatt et al., 1960), which was later introduced to the U.S. petroleum industry by Warren and Root (1963). The simplification invoked use of average properties of fractures and matrix on a macroscopic scale rather than the precise characteristics of individual fractures. Such models, known as the dual-porosity models, have also been in wide use for studying geothermal reservoirs. The essence of the model lies in the observations that the fractures generally have extremely large permeability and, therefore, essentially act as conduits to fluid flow, while the matrix with its much greater storage capacity and very low permeability (typically, millidarcy to less than microdarcy) feeds the fractures associated with it. In such a dual-porosity model, all the fractures are grouped into one continuum while the matrix into another. The flow between the matrix and fracture is considered to be in the semi-steady state, and controlled by interporosity flow parameters.

Pruess (1983) and Pruess and Narasimhan (1985) extended the concept of above described two continua model to one with multiple interacting continua (MINC). They pointed out that the transients in low permeability matrix may persist for several months to years, and hence the transient flow behavior within the matrix should be accurately represented. Their conceptual MINC blocks were represented by a set of nested blocks with fine nested discretization at the fracture-matrix interface and gradually increasing spacing away from the fracture as shown in Figure 2. They had also suggested that the MINC formulation *can* be invoked into any existing simulator based on integral finite-difference formulation by proper input data directives, and that no programming changes were necessary in the simulator.

Zimmerman and Bodvarsson (1992) and Zimmerman et al. (1993a) further confirmed that two continua model of Barenblatt et al. (1960) and Warren and Root (1963) is grossly inadequate for studying transient phenomena, and that the MINC method is accurate provided the matrix is subdivided into about ten or more interacting continua. They pointed out that many practical problems with MINC formulation result in a large number of discretized elements which renders the method computationally burdensome and unattractive. In an effort to overcome this deficiency, they proposed a semi-analytical procedure applicable to single phase flow systems, where the interflow between matrix and fracture was calculated using Vermeulen's (1953) equation and represented it as a source/sink term into the finite-differenced fracture flow equations. The method was later extended to two-phases problems (Zimmerman et al., 1993b) where one of the phases is not mobile. In addition to uncertainty as to the accuracy of Vermeulen equation for more general boundary conditions, its extension to highly non-linear two-phase flow problems is not apparent to us.

As discussed above, the MINC method accurately describes the transient phenomena, and applicable to a general class of multi-phase multi-component nonisothermal flow problems; our efforts were, therefore, primarily focussed on overcoming the computational burden that the method imposes. In the following, we present a simple and efficient procedure based on the coefficient-matrix decomposition for implementing the MINC method. The procedure has proven to be substantially faster, requires much less memory, is easy to implement into any existing geothermal simulator, and computationally robust. Furthermore, the method can be readily implemented on a parallel processing computing system with a high degree of parallelism.

In order to avoid ambiguity we use the term "coefficient-matrix" to mean the matrix arising from a set of linearized flow equations, while the term "matrix" without the prefix "coefficient" is used to reference the rock matrix, i.e., the low-permeability continuum.

We also present a comparison of the computing time, and the memory requirements on several different problems which were run on TOUGH2 (Pruess, 1991) and SIM.FIGS, a brief description of the latter is given in Hanano and Seth (1995).

2. MATRIX DECOMPOSITION PROCEDURE

For brevity and without loss in generality, consider a two dimensional fracture-matrix domain discretized into nx and ny blocks in x and y directions, respectively. Let NMINC be the number of nested continua in which each grid block is further subdivided as shown in Figure 1. The outermost continuum represents the fracture and the

rest (NMINC−1) represent the matrix. We are illustrating the procedure for an ordered set of equations such as arise in finite−difference formulation; however, the method is applicable to arbitrary ordered blocks, connectivity and dimensionality. The nested elements are numbered increasing outward for convenience.

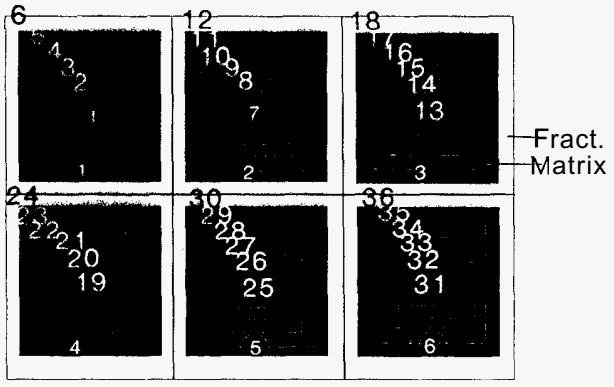

Figure 1 Conceptual MINC structure (after Pruess and Narasimhan, **1985).** Each matrix blocks discretized into 5 elements with fractured blocks shown in circled numbers.

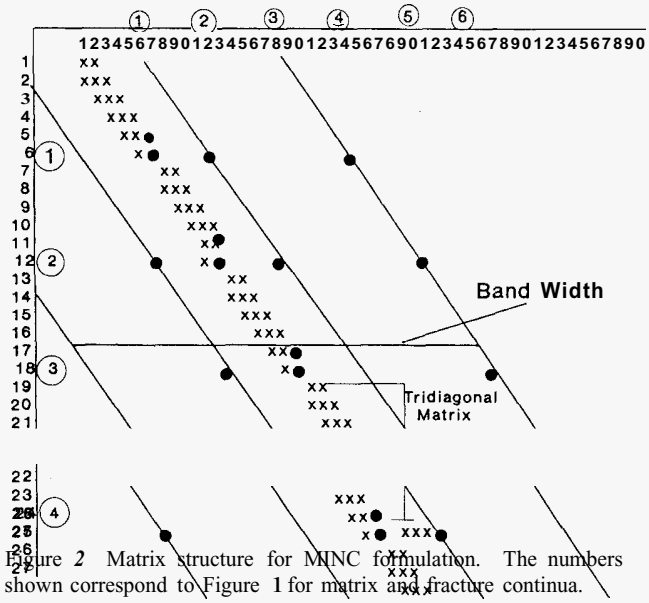

Figure *2* Matrix structure for MINC formulation. The numbers shown correspond to Figure 1 for matrix and fracture continua.

Writing the flow equations in difference form at each of the elements results in a coefficient−matrix as shown in Figure *2.* The solid dots represents the coefficients for the fracture continuum, while the x's represent the coefficients for the matrix blocks. The standard ordering of the blocks results in a banded structure of the coefficient−matrix with band width:

$$\text{Ibw} = (2\times nx\times \text{NMINC}+1)\times np \qquad (1)$$

with total number of equations

$$= nb\times \text{NMINC}\times np \qquad (2)$$

where,

nb = nxxny

np = **number** of unknowns/block.

An examination of the coefficient−matrix structure reveals that:

1. coefficients arising from the matrix continua have a well known structure of tridiagonal matrix, typical of 1−D problems of length (NMINC−1), and

2. the last row in each of the tridiagonal matrices, has an additional coefficient resulting from the coupling between the fracture and the matrix.

The above characteristics permit **us** to solve all the 1−D problems arising from the matrix continua independent of each other in terms of the solution vector for the associated fracture, employing the well known Thomas algorithm for 1−D problems. This permits decoupling of the matrix and fracture flow equations. More specifically, we proceed as follows:

1. during the forward sweep of the Thomas algorithm, the last equation takes the form:

$$C_m\times\delta P_m = R_m - C_f\times\delta P_f \qquad (3)$$

or

$$\delta P_m = (R_m - C_f\times\delta P_f)/C_m \qquad (4)$$

where C, R and SP are respectively, the diagonal coefficients, residual, and the change in the solution vector over an iteration, subscript "m" refers to the last equation of 1−D problems, and "f" refers to the associated fractures. Note that C is a sub−matrix of order np, and SP and **R** are column vectors of length np.

2. Next, we substitute Eq.(4) in the fracture equations and thereby eliminate all the couplings between the fracture and matrix equations.

3. The resulting fracture equations are then solved by any desired iterative or direct method.

4. The fracture solution vector is then substituted in Eq.(3) to calculate δP_m.

5. The backward sweep of the Thomas algorithm is now completed to get solution vector for the matrix elements.

Steps 1 through 5 are repeated at each Newtonian iteration.

3. COMPUTATIONS AND MEMORY:

The coefficient matrix arising from the decoupled equations for fractures will have the same structure as the one for single porosity systems. Since these equations may be solved efficiently by several different direct and iterative methods, we give here an estimate of incremental computational work required and memory needed for implementing the MINC formulation.

The work required in terms of number of multiplications and divisions for solution of 1−D problem of dimension n, using the Thomas algorithm is simply $(5\times n-4)\times np\times np$. In addition, we require $12n$ $(= 3\times np\times np\times n)$ more operations for two unknowns per grid block. Thus, the total work estimate with NMLNC continua is:

$$W = nx\,xny\,x16\times(2\times\text{NMINC}-3) \qquad (5)$$

$$\text{NMINC} > 1$$

If NMINC = *2* corresponding to Warren and Root model, the above procedure requires 16 more operations per grid−block for a two−phase problem. The work estimate varies linearly with the number of MINC elements, unlike the direct elimination methods.

If the entire coefficient matrix made up of fracture and **matrix is** solved by the band algorithm, the work ratio can be easily estimated to be:

$$WR = ny\times \text{NMINC}^3/(ny\times ny+5\text{NMINC}-9) \qquad (6)$$

If nyxny >> 5×NMINC−4, the work ratio, WR approaches NMINC^3. Thus, for a typical problem with NMMC = 10, the new algorithm

will be about 3 orders of magnitude faster than the band algorithm. The conclusion is valid for 3-D problems where ny is simply replaced by nyxnz, nz being the number of blocks in the z-direction.

The memory required for the proposed procedure is np×np×nx×nz×(NMINC-1). Again comparing with the band algorithm, the memory ratio may be estimated as:

$$MR=NMINC\times(2\times NMINC\times ny+1)/2\times(ny\times NMINC+1) \quad (7)$$

The above ratio will approach $NMINC^2$ for problems with ny >> NMINC. As before, for 3-D problems, ny may be replaced by nyxnz in the Eq.(7).

A comparison of work and memory with sparse matrix techniques and iterative methods is not straightforward, as they vary with the matrix structure, ordering of the blocks, and the number of iterations in case of iterative methods. Pruess (1991) gives an estimate of memory space for MA28 sparse matrix solver of Duff (1977):

$$M = ICN + IRN + 50\times neq + 8\times NCON + CO + 20\times neq \quad (8)$$

where,

neq = number of equations,
NCON = number of interfaces.

The lengths of ICN, IRN, CO are computed internally by simulating the elimination process and are generally greater than the total number of non-zero matrix coefficients. For comparison, we present the calculated values of M in the later section.

We may point out that MA28 solver, in our experience, is a quite efficient solver in both computational speed and memory management. However, as with any direct method, it cannot compete with the iterative methods beyond a certain size problem. This is important in the context of the proposed procedure in that by decomposing the matrix, we can reduce the dimensionality of the problem, and can avail the **use** of such solvers for much larger problems.

4. RESULTS AND DISCUSSIONS:

We have applied the above-described procedure to a variety of different problems and compared the computing time and the memory requirements with the conventional method (Pruess and Narasimhan, 1985; **Pruess,** 1987) where the entire coefficient matrix is solved without decomposition. Direct elimination type methods including the well known band algorithm, sparse matrix technique using MA28 linear equation solver (Duff, 1977), and D4-ordered (Price and Coats, 1973) algorithm were used for solving the linear equations. Nested Factorization with Orthomin accelerator (NF) iterative solver of Appleyard and Cheshire (1983) was also applied to the decomposed coefficient matrix for fractures. All computations were performed in double precision on a PC-486/66 clone. The source codes were compiled using Lahey's F77-EM/32 Fortran compiler without invoking any special optimization parameters. Table-1 below summarizes the results.

As shown in column one of the table, we chose problems ranging from small 2-D with dimensions of **7** by 7 to medium sized 3-D problems of dimensions 15×15×10 fracture blocks. The values given in the parenthesis reflects the number equations to be solved. Since we have two unknown per discretized element, the total number of elements are half of this value. The largest sample problem give rise to 45,000 equations with 22,500 elements. The number of NMINC elements considered are either 8 or 10. The impact on computational and memory for other values of NMINC can be easily deduced from these two values by linear extrapolation for our method. Also, note that NMINC includes the fracture continuum also, ie there are NMINC-1 elements in the matrix. The acronyms MD and NF respectively stands for the present matrix decomposition method, and the Nested Factorization method. MD+NF implies that the solution of the decomposed coefficient matrix for fractures is solved by NF method. In other cases, MD was used in conjunction with D4 ordered direct method of Price and Coats (1973). Comparison runs with MA28 and D4 methods without MINC formulation suggests that the two methods are approximately comparable in speed for the chosen problems. All runs with *MA28* were carried out using the TOUGH2 simulator.

Table 1 Comparison of Solution Time and Memory Requirements

Problem Sire	Minc Elements	Method	cpu time per Newtonian iteration (sec)	Memory+ (K-words)	cpu ratio*	Memory Ratio
7x7x1 (882)	10	Band	193.40	440	.0077	.0909
		HA28	1.50	40	1.	1.
		MD	.035 (.03)	4	42.8	10.
20x20x1 (5,600)	8	MA28	2.62	250	1.	1.
		MD	.65 (.30)	27	4.0	9.3
20x20x1 (7.2001	10	MA28	7.52	304	1.	1.
		MD	.68 (.34)	34	11.1	8.9
8x8x8 (7.1681	8	HA28	44.3	281	1.	1.
		MD	2.8	35	15.8	8.0
		MD+NF	0.49 (0.30)	-	90.4	
8x8x8 (9,216)	10	MA28	55.7	498	1.	1.
		MD	2.9 (0.35)	43	19.2	11.6
15x15x10 (31,500)	8	MA28	Failed	-		
		MD+NF	2.69 (1.68)	153		
15x15x10 (45,000)	10	MD+NF	3.03 (2.02)	189		

+ Memory is tabulated as incremental over a non-minc system.
* Ratios are calculated with MA28 as the referenced method.
Numbers in the parenthesis are tine required per Newtonian iteration for solving 1-D MINC blocks.

The total computing time in the solvers only was captured to 1/100th of a second, and presented in the table as the time per Newtonian iteration. The numbers in the parenthesis in this column correspond to the time required per iteration for solving the 1-D matrix equations. Comparison of band algorithm is included only for the smallest problem, as the performance of it degraded dramatically with larger size problems. MA28 failed to produce acceptable solutions for the last two large problems possibly due to excessive round-off error. Note that the cpu ratio ranged from roughly 4 to over 19. **The** ratio increases to over 90 when NF was used for solving the fracture equation. **Also,** the ratio increases with larger ***size*** problems. The time for solving 1-d matrix equations varies slightly but linearly with the number of MINC elements, and primarily depends on the number of discretized fracture elements.

The memory presented reflects only the incremental memory required for solution with MINC implementation and without it. The words are double precision 8 byte words. **All** integer arrays (4 bytes) were converted to 8 byte words. The memory required for the test problems were generally an order of magnitude **less** with higher memory ratio for 3-D problems.

It is important to note that in the decomposition method, the amount of work required for the matrix equations is fixed and modest per Newtonian iteration.

For large problems where iterative methods are used, the computational time obviously goes up with the number of **inner** iterations. If the entire matrix is solved by such iterative methods, the work component required for the matrix equations will far exceed the 1-D solver time for these equations. For example, with one line **SOR** method, a single iteration will equal the time required for solving the matrix equation with the proposed method, while **SOR** may require in excess of 10 iterations.

Other significant benefit of decomposition is that the coefficient-

matrix structure remains the same as the single porosity system. This permits application of a number of efficient and specialized solvers (eg D4, NF) which are quite effective for structured matrices.

A fully implicit formulation for multiphase flow may lead to non-diagonally dominant coefficient matrix. In this case, many iterative methods fail to converge, and others converge rather slowly. By decomposing the matrix, the overall dimensionality of the problem is substantially reduced, in which case direct methods such as MA28 may be advantageously, used as discussed above.

6. PARALLELIZATION:

As discussed, each matrix block is independent of each other, and the decomposition and solution of 1-D matrix problem is carried out sequentially. This renders the method ideally suited for parallel processing computing systems or farm computing. We have initiated efforts to implement such processing using PVM (Parallel Virtual Machine) software designed by Oak Ridge National Laboratory. PVM software permits a group of heterogeneous computer systems including work stations, mainframes and vector processing machines, to run as a single virtual machine. PVM implements the message-passing paradigm at a primitive level and uses the distributed method of parallel processing.

For this application, the master-slave model can be effectively used (Frietas, 1994, private communication). In this mode, the master program spawns and directs some number of slave programs which perform the primary parallel computations. In our case, only one subroutine will be required for parallelization. This program will be transformed into a separate slave-program, which will then be controlled by the master program as mentioned above.

Depending on the number of computers running in parallel, the overall cpu time may approach close to the time required for just the solution of the fracture equations.

7. CONCLUSIONS

1. An efficient matrix decomposition method is developed for solving flow equations arising from Multiple Interacting Continua formulation of fracture-matrix rock complex.

2. The method requires approximately three orders of magnitude less computing time and two orders of magnitude less storage space, as compared to the band algorithm. It is also one to two orders of magnitude faster than MA28, and requires substantially less memory.

3. The method is general and can be applied to any multicomponent, multiphase, multidimensional nonisothermal problem.

4. The method is applicable to any co-ordinate geometry.

6. The method is simple to implement into any existing geothermal simulator.

7. The method is ideally suited for parallel computations.

Acknowledgement

This work was supported by JMC Geothermal Engineering Co., Ltd. which is gratefully acknowledged.

REFERENCES:

Appleyard, J.R., and Cheshire, I.M. (1983) Nested factorization. *Presented at the Seventh SPE Symposium on Reservoir Simulation, San Francisco, Calif:*

Barenblatt, G.I., Zheltov, Y.P., and Kochina, I.N. (1960) Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks. *J. Appl. Math. Mech. Engl. Transl.*, Vol.24, pp.1286-1303.

Duff, I.S. (1977) *A set of Fortran subroutines for sparse unsymmetric linear equations,* AERE Harwell Report R 8730.

Hanano, M. and Seth, M.S. (1995) Numerical modeling of hydrothermal convection systems including super-critical fluids. *World Geothermal Congress, Florence, Italy. May 1995.*

Price, H.S. and Coats, K.H. (1973) Direct methods in reservoir simulation. *SPE # 4278 presented at the Third SPE Symposium of Numerical Simulation of the Soc. Pet. Eng. of AIME, Houston, Teras.*

Pruess, K. (1983) *GMINC- A mesh generator for flow simulations in fractured reservoirs.* Rep. LBL-15227, Lawrence Berkeley Lab., Berkeley, Calif.

Pruess, K., and Narasimhan, T.N. (1985) A practical method for modeling heat and fluid flow in fractured porous media. *Soc. Pet. Eng. J.*, Vol.25, pp.14-26.

Pruess, K. (1987) *TOUCH user's guide.* Rep. LBL-20700, Lawrence Berkeley Lab., Berkeley, Calif.

Pruess, K. (1991). *A general purpose numerical simulator for multiphase fluid and heat flow.* Rep. LBL-29400, Lawrence Berkeley Lab., Berkeley, Calif.

Vermeulen, T. (1953) Theory of irreversible and constant-pattern solid diffusion. *Ind. Eng. Chem.,* Vol.45, pp.1664-1670.

Warren, J.E., and Root, P.J. (1963) The behavior of naturally fractured reservoirs. *Soc. of Pet. Eng. J.,* Vol.3, pp.245-255.

Zimmerman, R.W., and Bodvarsson, G.S. (1992) Semi-analytical treatment of fracture/matrix flow in a dual porosity simulator for unsaturated fractured rock mass. *Presented at the Third International High-Level Radioactive Waste Management Conference, Las Vegas, Nevada.*

Zimmerman, R.W., Chen, G., Hadgu, T., and Bodvarsson, G.S. (1993a) A numerical dual-porosity model with semianalytical treatment of fracture/matrix flow. *Water Resources Research,* Vol.29, pp.2127-2137.

Zimmerman, R.W., Hadgu, T., and Bodvarsson, G.S. (1993b) Development of dual-porosity model for vapor-dominated fractured geothermal reservoirs using a semi-analytical fracture/matrix interaction term. *Presented at the Eighteenth Workshop on Geothermal Reservoir Engineering, Stanford University, Calif.*