Scaling up of Effective Absolute Permeability

Fatemeh Razavi Z. and Clayton V. Deutsch

A common problem in Geostatistical reservoir modelling is the calculation of effective permeability to represent a high resolution regular grid. The flowsim program has been used since the mid 1980s for this purpose. This research note reviews the implementation and recommends some changes including applying another iterative solver (SIP - the strongly implicit procedure) and convergence criteria that permit reliable results without excessive computational effort. The results are checked with a direct solver. Four examples are presented.

Introduction

System of Equations is sometimes solved by direct methods for instance Gauss elimination or LU decomposition. The pressure Equation that must be solved for effective permeability calculation is quite large and sparse.The CPU and storage cost of direct methods is fairly high. Iterative solution methods are commonly employed. In an iterative method, one should guess a solution (which is the starting point) and use the Equation to systematically improve it (Ferziger, 2002). If the number of iterations is small and each iteration is cheap, an iterative solver may cost less than a direct method and this is usually the case in CFD related problems including the pressure Equation in upscaling.

The flowsim program, that is commonly used in CCG, is a 3D-one phase flow simulator used to calculate effective permeability. The solution of large linear systems of Equations takes most of the computational time. Several efforts have been done to develop linear solvers and the pre-conditioners that will improve the performance of flow simulators by using a fast solver. In 1981, Watts applied the preconditioned conjugate gradient method to solve the pressure Equation and compared with SIP. He showed that SIP is faster than preconditioned conjugate gradient for 3D problems but slower for 2D one (Watts and James, 1981).

In 1988, Eisenstat described a collection of block preconditioners for use in solving large, sparse, linear systems of Equations by iterative methods, and compared their performance with several point preconditioners in solving some systems arising in numerical reservoir simulation (Eisenstat et all, 1988). Brand presented the successful application of an incomplete LU (ILU) factorization technique coupled with generalized conjugate-gradient (as an acceleration) to solve the set of Equations (Brand et all, 1990). In the field of Mathematics in 2001 at University of Texas, Eaton completed his PhD research related to multigrid preconditioner. He discussed that how a multigrid preconditioner has been successfully applied for a 2D flow problem (Eaton, 2001). Furthermore, several solver packages have been used in Integrated Parallel Accurate Reservoir Simulators (IPARS). Generalized minimum residual (GMRES) with various preconditioners like Linear Successive Over Relaxation, Incomplete Lower Upper and Algebric Multi Grid are the most popular methods in IPARS (most of them are for parallel simulations) (Klie and Wheeler, 2005). The parallel solvers could be applied for different structures of the algebraic systems when the domain of study is decomposed (Mary. F. Wheeler, 1999).

Algebric Multigrid method is another efficient solver and preconditioner for flow Equation system that found lots of consideration recently. A package named SAMG (Algebraic Multigrid Methods for Systems) has been developed in this area by Fraunhofer (Klaus Stüben and Tanja Clees, 2007). Mishev discussed comprehensively how AMG could be one of the most efficient solvers that is handling linear systems resulted by the discretization of the pressure Equation in IMPES and Sequential Implicit formulations. Also he believes to the high applicability of AMG in parallel algorithms (Ilya Mishev, 2011). HYPRE is an available library of high performance preconditioners to solve large sparse linear systems. It includes several solvers and preconditioners that are highly applicable in parallel multiphysics and multiscale simulation. (Robert, 2002). Our focus is on the flowsim program and the performance of linear solvers that could be used in flowsim program. Linear Successive Over Relaxation (LSOR) is the iterative solver that has been used for a long time in the program.

The Strongly Implicit Procedure (SIP) is another iterative solver that is recommended to be used (Weinstein and Stone, 1969). Moreover, GBAND which is a direct solver has been added to the flowsim program to get the exact solution of the pressure Equations. By knowing the exact solution, we can easily compare the convergence of the iterative solvers to the exact solution. Comparative studies have been done on some cases considering different permeability fields to study the effectiveness of the mentioned iterative algorithms.

Problem Formulation: Building Pressure Matrix

The input to the flowsim program is a fine scale 3-D Cartesian grid of permeability (Kx, Ky and Kz) and it will be scaled to a coarser 3-D Cartesian grid of effective permeability (Keffx, Keffy and Keffz). The arithmetic, geometric and harmonic averages are also reported for checking. The effective permeability in each direction is calculated by solving the steady-state single-phase flow Equations with no flow boundary conditions (Deutsch, 1989).

The effective permeability in the X direction is calculated by Equation 1. Where nx, ny and nz are model discretization numbers in x, y and z directions and q_{ave} is the average of cumulative input and output flow rates. p_{in} and p_{out} are pressures of input and output boundary grids which are set to 0 and 100 respectively.

$$rkeff_x = 2q_{ave} \frac{nx}{n_z n_y \, (p_{in} - p_{out})} \tag{1}$$

$$q_{ave} = \frac{(q_{in} + q_{out})}{2} \tag{2}$$

q_{in} and q_{out} are cumulative input and output flow rates that are calculated by Equations 3 and 4.

$$q_{in} = \sum_{inlet\ blocks} (p_{in} - p_i) \ k_i \tag{3}$$

$$q_{out} = \sum_{outlet\ blocks} (p_i - p_{out}) \ k_i \tag{4}$$

To obtain q_{in} and q_{out}, we need to know pressure distribution of the model and here is the place of solver. The solver will solve a large system of pressure equations and we will have the pressure field necessary to calculate effective permeability. The resulted pressure field will have values between p_{in} and p_{out}. q_{in} and q_{out} should be the same; any difference is an indication of numerical instability.

These calculations are done in the X, Y and Z directions separately to get effective permeability in each direction. In a structured 3D model, there are six neighbors for each grid block except at the boundaries of the model. The pressure at the block center is related to the pressures of the adjacent blocks through the pressure equation (see Figure 1). There is a separate pressure equation for each grid block in the model which results in a 7 diagonal pressure matrix for each set of boundary conditions. Grid indexing is an important point that should be considered carefully not only in making the pressure equation but also when applying the linear solver. Different indexing will affect the values on the diagonals. If (i , j , k) is the index of the grid at the center, the diagonals are listed in Table 1.

Table 1

Diagonal	Index	Related Array
Bottom diagonal	(i , j , k - 1)	AB
South diagonal	(i , j - 1 , k)	AS
West diagonal	(i - 1 , j , k)	AW
Main diagonal	(i , j , k)	AP
East diagonal	(i + 1 , j , k)	AE
North diagonal	(I , j + 1 , k)	AN
Top diagonal	(I , j , k + 1)	AT

In the flowsim program, a 1-D index for each grid center with 3-D index (i , j , k) is calculated by:

$$ip = (k-1) \times nx \times ny + (j-1) \times nx + i \tag{5}$$

The above indexing will result in the following pressure Equation for each grid block with the ip index:

$$P(ip - nx \times ny) \times AB(ip) + P(ip - nx) \times AS(ip) + P(ip - 1) \times AW(ip) + P(ip) \times AP(ip) + P(ip + 1) \times AE(ip) + P(ip + nx) \times AN(ip) + P(ip + nx \times ny) \times AT(ip) = 0 \tag{6}$$

If the 1-D index is calculated as in Equation 7, then the pressure equation would change to that is presented by Equation 8.

$$ip2 = (k - 1) \times nx \times ny + (i - 1) \times ny + j \tag{7}$$

$$P(ip2 - nx \times ny) \times AB(ip2) + P(ip2 - 1) \times AS(ip2) + P(ip2 - ny) \times AW(ip2) + P(ip2) \times AP(ip2) + P(ip2 + ny) \times AE(ip2) + P(ip2 + 1) \times AN(ip2) + P(ip2 + nx \times ny) \times AT(ip2) = 0 \tag{8}$$

By writing the pressure equation for all grids, a pressure matrix is generated that is a sparse matrix. In the next section, the linear solvers to solve the linear system of pressure equations are described. In the flowsim program, the diagonals are as follows:

Table 2

dxp	AE
dxm	AW
dyp	AN
dym	AS
dzp	AT
dzm	AB
d0	AP

Solving the Pressure Equations

The performance of two iterative solvers is investigated: Linear Successive Over Relaxation (LSOR) and Strongly Implicit Procedure (SIP). Comparative studies are done on some cases to investigate the effectiveness of the mentioned algorithms considering different permeability fields. The performance is investigated when converging to the exact solution. The exact solution is obtained by applying the GBAND algorithm that is a direct solver applicable for banded matrices.

Direct Solver

GBAND is a direct solver for the solution of banded matrices without pivoting. The input of the algorithm is a one dimensional array containing the band of the diagonal matrix sorted by rows. The required dimension of the array is:

$$Band\ dimesnion = O \times (2 \times M + 1) - M \times M - M \tag{9}$$

where *M* is the number of diagonals above the main diagonal and *O* is number of equations. The number of diagonals above and below the main diagonal are the same. More details can be found in the Aziz and Sattari book (Aziz and Sattari, 1979) is suggested to read.

Iterative (indirect) Solvers: LSOR, SIP

a) Linear Successive Over Relaxation

Successive over relaxation (SOR) is a popular iterative method that is an accelerated version of the Gauss Seidel algorithm. Consider a 2D model with 5 points, if each iteration is started in the lower left (southwest) of the domain, successive over relaxation method can be
written as follows:

$$P_i^{n+1} = w\frac{P_i - A_{j-1}P_{j-1}^{n+1} - A_{i-1}P_{i-1}^{n+1} - A_{j+1}P_{j+1}^{n} - A_{i+1}P_{i+1}^{n}}{A_i} + (1 - w)P_i^n \tag{10}$$

n is the iteration counter and *w* is the over-relaxation factor for acceleration, it must be greater than 1. It is hard to find the optimum value for over-relaxation factor in complex problems. In general, the value is larger for larger grids. The number of iterations will be proportional to the number of grid points in one direction, when the optimum value of overrelaxation factor is used. When *w* has the value 1, SOR reduces to the Gauss Seidel method. It is specifically designed for algebraic equations and usually converges in a small number of iterations (Ferziger, 2002).

b) Strongly Implicit Procedure

Strongly Implicit Procedure (SIP) is an incomplete lower-upper decomposition method which has found use in CFD problems. It was proposed by Stone in 1968 (Stone, 1968). Stone improved the convergence of Incomplete Lower Upper decomposition SIP. The Strongly Implicit Procedure usually converges in a small number of iterations as well. One of the positive points of the SIP method is its application not only as a good iterative technique but also as a preconditioner for conjugate gradient methods and a smoother for multigrid method (Ferziger, 2002). A 3D vectorized version of SIP has been given by Leister and Peric in 1994 (Leister and Peric, 1994). The rate of convergence in Stone`s method can be improved by changing Stone`s parameter (alpha), which is a problem dependent parameter, from iteration to iteration. Investigating the dependence of the convergence behavior on the parameter alpha between 0.92-0.94 were found to give results close to the optimum ones for a wide range of problems. These values are suggested for general use (Leister and Peric, 1994). In the flowsim program, the value of alpha has been determined based on descriptions in Weinstein paper (Weinstein and Stone, 1969).

Stopping Criterion

A stopping (convergence) criterion is needed to determine when to stop the iteration process. Ideally the distance of the last iteration to the true solution could be measured. The difference between a computed iterative solution and the true solution of a linear system is a measure of error. In practice, we would not have the true solution, but we can solve for it in test cases to establish the convergence properties of the different iterative methods. It would be reasonable to choose the iterative algorithm with less CPU time and memory storage.

Applying the Algorithms

All the calculations have been done in the flowsim program that includes all three solvers: LSOR, SIP and GBAND. The exact solution is determined by applying the direct solver (GBAND). The goal is to investigate the convergence behavior of the SIP and LSOR algorithms when dealing with various K fields with different levels of complexity. Rapid convergence of an iterative method is the key factor for its effectiveness. Convergence is defined as the reduction of the iteration error below some tolerance (Ferziger, 2002).

Stopping (Convergence) Criteria for the Current Problem

The stopping criterion for the algorithms is the maximum change made to the pressure field in a given iteration. If the change is low enough (less than the input residual), then we assume the pressure field is close enough to the exact solution. When we know the exact solution by applying a direct solver, then we can easily check that if the iterative algorithms have reached the exact solution or not. We considered the permeability error as stopping criteria. When it is below 0.1 % the algorithms stop. Permeability iteration errors are calculated by equations 11 and 12. Permeability and pressure convergence are presented through plotting the errors versus iterations when converging to the exact solution.

$$Kerror(i) = \frac{\left|Keff_i^{calc} - Keff^{\,true}\right|}{Keff^{\,true}} \,.100 \tag{11}$$

$$Perror(i) = \frac{1}{N}\sum_{n=1}^{N} \frac{|P_n^i - P_n^{true}|}{P_n^{true}} \,.100 \tag{12}$$

Case Studies: Some Results

Four cases have been investigated. All are 3D problems. The first, third and forth cases are generated in Matlab and the second case is a 10 by 10 by 11 model generated by the sgsim program. The first and forth cases are models with 10 by 10 by 10 grids and the third one is a 10 by 10 by 11 model. The difference between 4th and 2nd cases is high permeability grid blocks in shale layers in the 4th model. Plots are generated in Matlab by loading the output files of flowsim. GBAND is used to get the exact solution. To apply GBAND algorithm, the 7 diagonals dimensions based on the chosen indexing are as follows:

Table 4

Indexing	$ip = (k-1) \times nx \times ny + (j-1) \times nx + i$	$ip = (k-1) \times nx \times ny + (i-1) \times ny + j$
Dimension	dimension(AB) = (nx * ny * nz) - (nx * ny) dimension(AS) = (nx * ny * nz) - (nx) dimension(AW) = (nx * ny * nz) - (1) dimension(AP) = (nx * ny * nz) dimension(AE) = (nx * ny * nz) - (1) dimension(AN) = (nx * ny * nz) - (nx) dimension(AT) = (nx * ny * nz) - (nx * ny)	dimension(AB) = (nx * ny * nz) - (nx * ny) dimension(AS) = (nx * ny * nz) - (1) dimension(AW) = (nx * ny * nz) - (ny) dimension(AP) = (nx * ny * nz) dimension(AE) = (nx * ny * nz) - (ny) dimension(AN) = (nx * ny * nz) - (1) dimension(AT) = (nx * ny * nz) - (nx * ny)

Knowing the exact solution, the permeability error and pressure error (when comparing to the exact solution) are calculated in each iteration and in x, y and z direction separately. CPU time is measured in x, y and z directions when applying GBAND, LSOR and SIP. The resulting graphs and tables are presented for each case separately. To get the plots, the following steps are taken on each case:

1) Run flowsim by applying GBAND to get the exact pressure profiles and exact upscaled permeability values in X, Y and Z directions.

2) Input the exact pressure solution and effective K(s) solution to flowsim in x, y and z directions to measure error when applying LSOR and SIP.

3) Run LSOR with 10000 iterations to get K and p convergence error report in all directions.

4) Again run LSOR by considering the stopping criteria and measuring the CPU time in all directions.

5) Run SIP with 10000 iterations to get k and p convergence error report in all directions.

6) Again run SIP by considering the stopping criteria and measuring the CPU time in all directions.

The calculated effective permeability by SIP and LSOR in all three directions are the same as the exact effective permeability calculated by the direct solver, see table 5. The measured CPU time when running the algorithms have been tabulated in Tables 6 to 8. A summary of the error plots for different cases are presented in the next section as well.

Table 5

Exact Solution Resulted by GBAND							
	Kx	Ky	Kz	Khori.	Karith.	Kgeom.	Kharm.
Case 1	500.5	500.5	1.718	500.5	500.5	31.623	1.998
Case 2	545.909	545.909	1.890	545.909	545.909	43.288	2.197
Case 3	436.977	440.731	357.521	438.854	538.079	185.906	0.014
Case 4	501.945	501.945	7.891	501.945	540.460	41.687	2.171

Table 6

Case 1: S.C. < = 0.1%						
	$CPU\ time\ X$	$CPU\ time\ Y$	$CPU\ time\ Z$	niterX / S.C.	niterY / S.C.	niterZ
SIP	0.0468003	0.0468003	27.8954602	12 / 0.00055	12 / 0.00055	> 10e5/13.45
LSOR	0.0624004	0.0624004	31.6370028	48 / 0.00072	48 / 0.00072	> 10e5 /13.99
GBAND	0.2340015	0.2184014	0.2340015	Direct Solver		

Table 7

Case 2: S.C. < = 0.1%						
	$CPU\ time\ X$	$CPU\ time\ Y$	$CPU\ time\ Z$	niterX	niterY	niterZ
SIP	0.0312002	0.0312002	0.2496016	11	11	499
LSOR	0.0312002	0.0312002	0.5616036	48	48	1511
GBAND	0.2675	0.2309	0.2675	Direct Solver		

Table 8

Case 3: S.C. < = 0.1%						
	$CPU\ time\ X$	$CPU\ time\ Y$	$CPU\ time\ Z$	niterX	niterY	niterZ
SIP	0.0312	0.0468	0.0468	21	14	24
LSOR	0.0312	0.0468	0.0312	44	49	51
GBAND	0.2496	0.2496	0.2496	Direct Solver		

Discussion and Conclusion

A comparative study has been done in this paper to show the performance of linear successive over relaxation and strongly implicit algorithms on flowsim Program. SIP and LSOR are iterative solvers which are highly used to solve very large and sparse systems of linear equations. We know the exact solution of the system of pressure equations by applying a direct solver (GBAND) to flowsim program. Then the pressure convergence and permeability convergence to the exact solution have been investigated. CPU time has been measured when applying all algorithms as well. Based on presented case studies, it is clear that direct solver takes much longer than iterative solvers (about 10 times). For the permeability distributions shown in Figure 5, the comparative studies have been performed. Permeability convergence and pressure convergence to the exact solution for all cases have been presented in X, Y and Z directions for both iterative algorithms but some of them are presented in this paper as representative ones and other cases plots are tabulated in Tables 9 and 10. SIP and LSOR convergence graphs have been shown in red (dashed line) and blue (solid line) respectively.

Looking at the resulted figures, the following points are worthy to mention:

a) Calculations on the z direction of the first case (Figure 5a) is not an easy problem since the algorithms need more iterations to converge; see Figure 7. Both algorithms could converge to the exact solution

b) The difference between the pressure convergence behavior of the algorithms for first, second and 4th cases in Z direction is worthy to consider. By looking at Figure 7 for the first case pressure convergence, Figure 8 for the second case, one could figure it out that when there is low permeable layers in the model, the algorithms needs more efforts to satisfy the stopping criterion and also the algorithms have unpredictable and relatively unstable behavior when converging to the exact solution. When in each layer there are just some low permeable grids, see Figure 5d, algorithms are converging more easily comparing to the first case that the whole layer is low permeable (see Figures 7 and 11). The difference between the first and second cases is an additional low permeable layer at the bottom of the first case that results in a different behavior of algorithms, see Figures 7 and 8.

c) Both algorithms are converging to the exact pressure solution for all cases and in different directions.

d) Horizontal effective permeability is easily calculated by mathematical averaging of effective permeability values in X and Y directions.

e) The memory cost of the algorithms is relatively minor since only the banded part of the matrix is kept.

f) Generally, the level of residuals for SIP is reduced higher orders in less Iterations and SIP converges faster than LSOR for the examined cases.

g) SIP required less computational effort (less iterations) than LSOR while the CPU time for each inner iteration of SIP is more expensive than LSOR.

h) Looking at K convergence graphs, SIP is more stable. Less CPU time and stability convergence could be mentioned as considerable advantages of SIP.

A summary of error plots are presented in the Tables 9 and 10. For instance, effective permeability error plots in X, Y and Z directions are completely similar to the Figure 6a as listed in Table 9.

Table 9

	Kx_error vs. iteration	Ky_error vs. iteration	Kz_error vs. iteration
Case 1	Figure 6a	Figure 6a	Figure 6a
Case 2	Figure 6a	Figure 6a	Figure 6a
Case 3	Figure 6a	Figure 6a	Figure 6a
Case 4	Figure 6a	Figure 6a	Figure 11

Table 10

	Px_error vs. iteration	Py_error vs. iteration	Pz_error vs. iteration
Case 1	Figure 6b	Figure 6b	Figure 7
Case 2	Figure 6b	Figure 6b	Figure 8
Case 3	Figure 9	Figure 9	Figure 9
Case 4	Figure 6b	Figure 6b	Figure 11

Future Work

Influence of boundary conditions, grid size on algorithms convergence and also investigating the permeability tensor properties could be among the future plan of the research. The convergence of the algorithms to the exact solution (known by GBAND) could be studied as well when dealing with more permeability fields to have a representative plot of number of algorithms iterations versus complexity of the models.

Acknowledgment

I would like to acknowledge the contributions of Dr. John Manchuk for recoding Strongly Implicit Algorithm (SIP) and also for his valuable comments.

References

Aziz Kh., Settari A., 1979, "Petroleum Reservoir Simulation", Applied Science Publishers.

Brand, Clemens W., Heinemann, Zoltan E., A New Iterative Solution Technique for Reservoir Simulation Equations on Locally Refined Grids Mining U. Leoben, SPE 18410-PA.

Deutsch, C. V., 1999, Reservoir Modeling with Publicity Available Software, Computers and Geosciences.

Eaton, Frank Joseph, 2001, A multigrid preconditioner for two-phase flow in porous media thesis (Ph. D.) University of Texas at Austin.

Ferziger J. H., 2002, Computational Methods for Fluid Dynamics, 97-105, Springer.

Klaus Stüben and Tanja Clees, 2007, Algebraic Multigrid Methods (AMG) for the Efficient Solution of Fully Implicit Formulations in Reservoir Simulation, SPE 105832-MS.

Klie H., Wheeler M.F., 2005, Krylov-Secant Methods for Accelerating the Solution of Fully Implicit Formulations, Center for Subsurface Modeling, U. of Texas at Austin, SPE 105832.

Leister, H. J., Peric, M., 1994, Vectorized Strongly Implicit Solving Procedure for a Seven-Diagonal Coefficient Matrix.

Robert D. Falgout , Ulrike Meier Yang, 2002, hypre: a Library of High Performance Preconditioners.

Stanley C. Eisenstat, Howard C. Elman, Martin H. Schlutz, Yale U., 1988, Block-Preconditioned Conjugate-Gradient-Like Methods for Numerical Reservoir Simulation, SPE 13534-PA.

Stone, H. L., 1968, Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations.

Watts III, James W., 1981, Conjugate Gradient-Truncated Direct Method for the Iterative Solution of the Reservoir Simulation Pressure Equation, SPE 8252-PA.

Weinstein H. G., Stone H. L., Kwan, T. V., 1969, Iterative Procedure for Solution of Systems of Parabolic and Elliptic Equations in Three Dimensions, Esso Co., Houston, Texas.

Appendix: Documenting Parameters

$\boldsymbol{rkeff_x}$	Effective Permeability in X direction	w	over relaxation factor
$\boldsymbol{q_{ave}}$	average flow rate	dym	AS in flowsim program
$\boldsymbol{n_x}$	number of discretization in x direction	dzp	AT in flowsim program
$\boldsymbol{n_y}$	number of discretization in y direction	dzm	AB in flowsim program
$\boldsymbol{n_z}$	number of discretization in z direction	d0	AP in flowsim program
$\boldsymbol{P_{in}}$	inlet grids pressure	O	Number of Equations
$\boldsymbol{P_{out}}$	outlet grids pressure	M	Number of diagonals in pressure matrix
$\boldsymbol{q_{in}}$	cumulative input flow rates	$K_{Arithmatic}$	Arithmetic Permeability Average
$\boldsymbol{q_{out}}$	cumulative output flow rates	$K_{Harmonic}$	Harmonic Permeability Average
$\boldsymbol{k_i}$	permeability of i^{th} grid	S.C.	Stopping Criteria
$\boldsymbol{p_i}$	pressure of i^{th} grid	Kerror(i)	permeability error in i^{th} iteration
i	grid index in X direction	Perror(i)	pressure error in i^{th} iteration
j	grid index in Y direction	$\boldsymbol{Keff_i^{calc}}$	calculated effective permeability in i^{th} iteration
k	grid index in Z direction	$\boldsymbol{Keff^{true}}$	exact effective permeability obtained by GBAND
ip, ip2	cumulative index	N	Number of grids
AB	Bottom diagonal of pressure matrix	$\boldsymbol{P_n^{true}}$	exact pressure of n^{th} grid obtained by GBAND
AS	South diagonal of pressure matrix	$\boldsymbol{P_n^i}$	pressure of n^{th} grid in i^{th} iteration
AW	West diagonal of pressure matrix	Kx	effective permeability in X direction
AP	Main diagonal of pressure matrix	Ky	effective permeability in Y direction
AE	East diagonal of pressure matrix	Kz	effective permeability in Z direction
AN	North diagonal of pressure matrix	Khori.	Horizontal Permeability
AT	Top diagonal of pressure matrix	Karith.	Arithmetic Permeability
dxp	AE in flowsim program	Kgeom.	Geometric Permeability
dxm	AW in flowsim program	Kharm.	Harmonic Permeability
dyp	AN in flowsim program	niterX	number of iterations in X direction

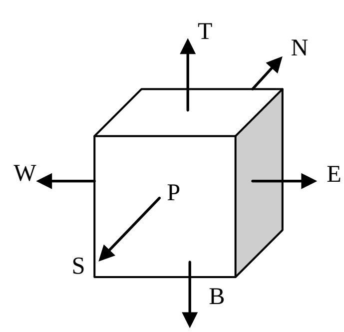
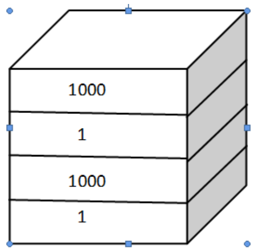
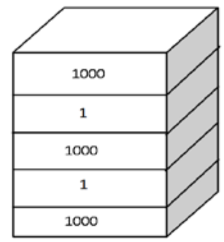
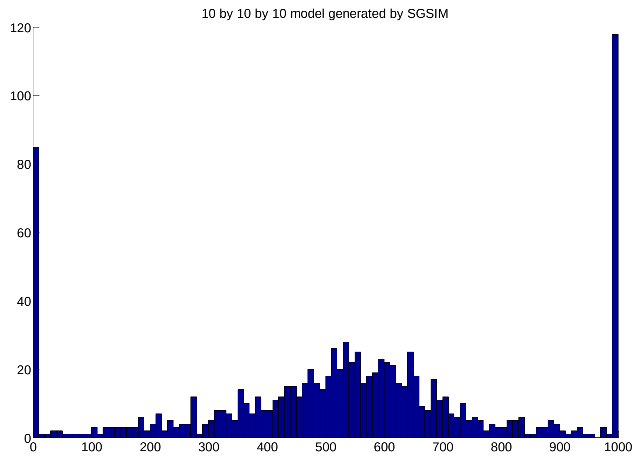


Figure 1: naming convention for adjacent grid blocks

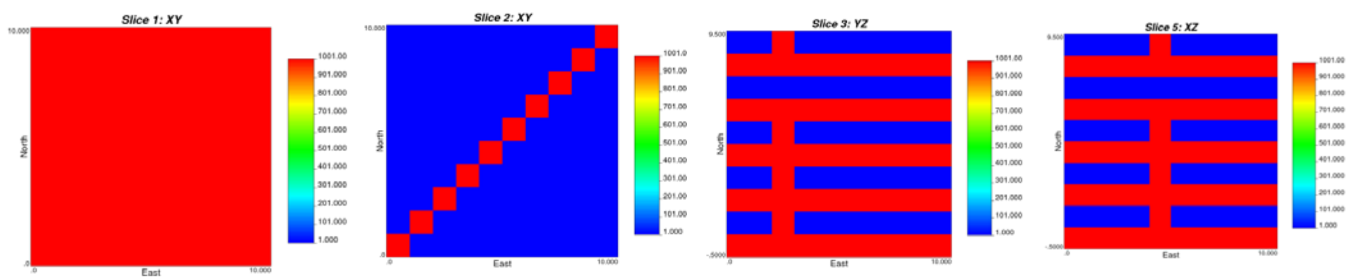




a) First Model b) Second Model

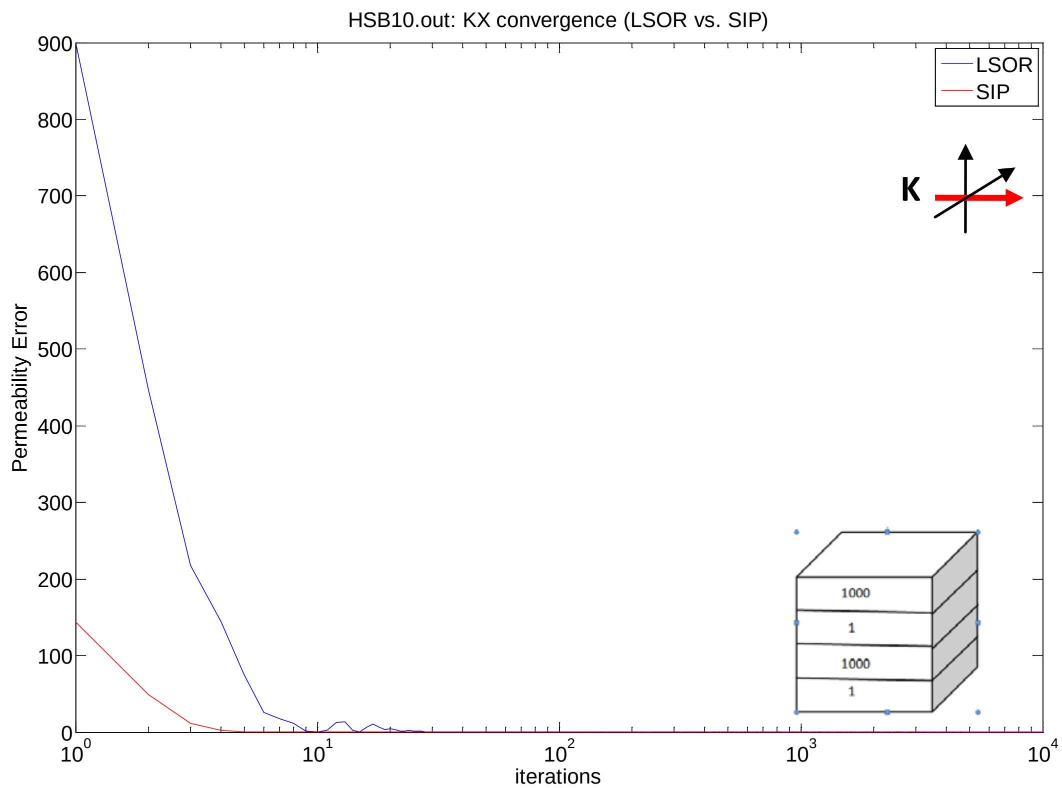


c) Third model



d) Fourth model

Figure 5: Differenet perambility fields used in the report



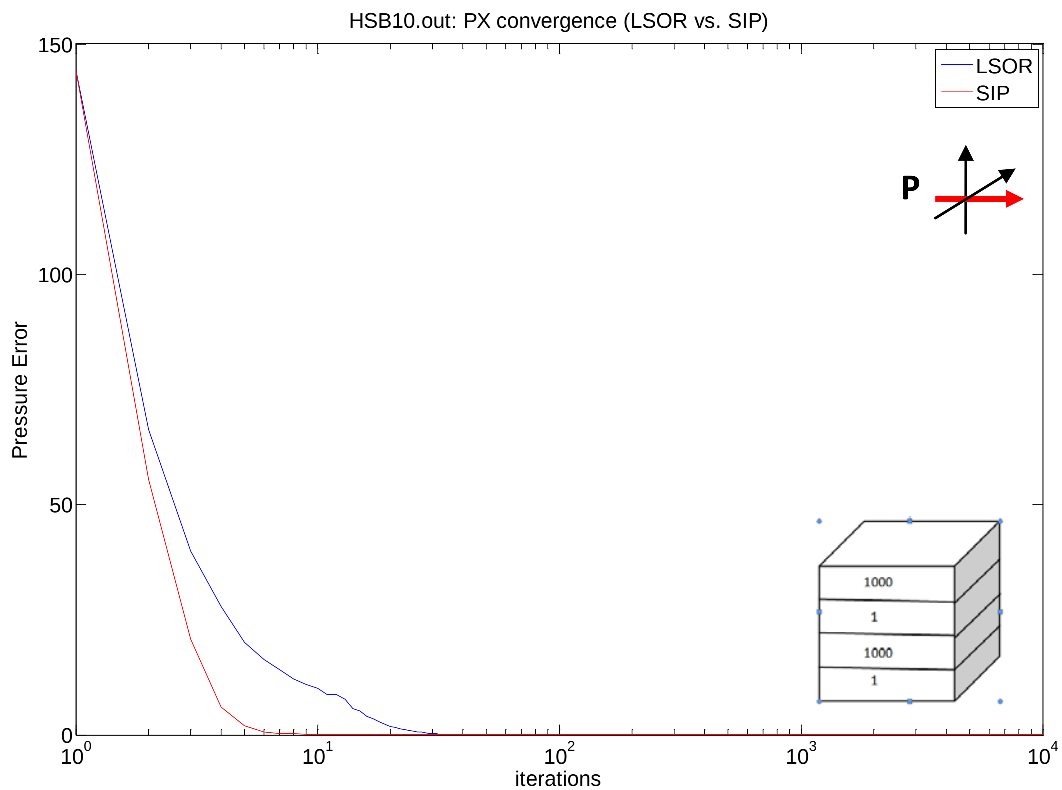


Figure 6a and 6b

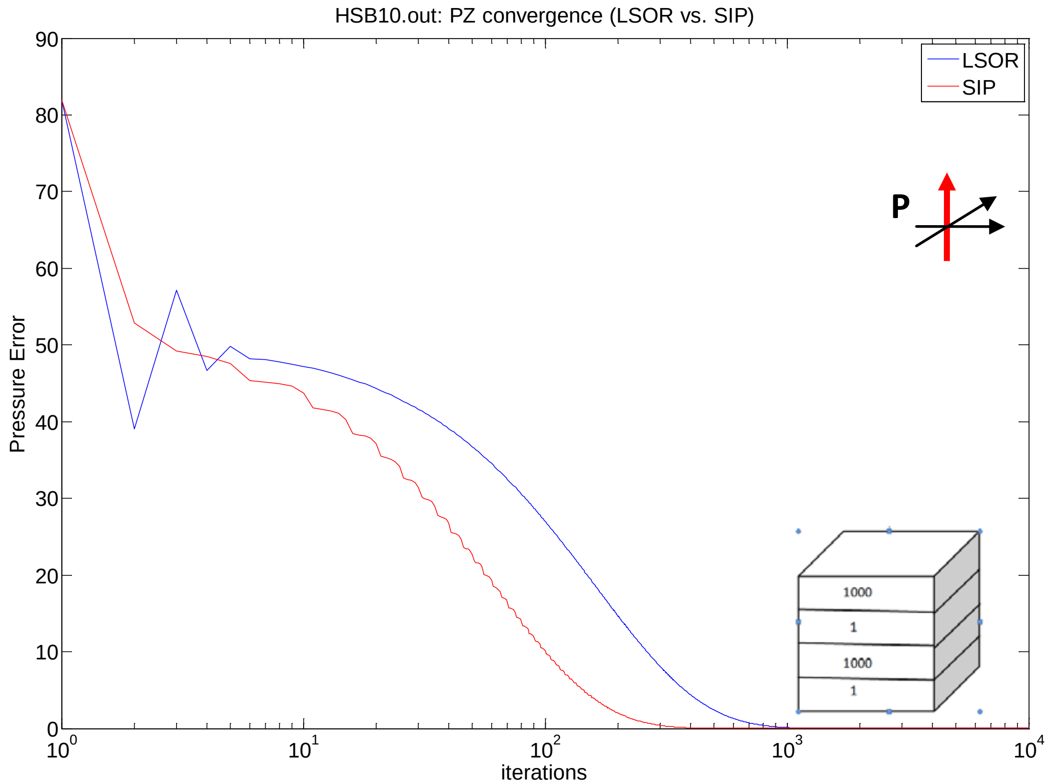


Figure 7

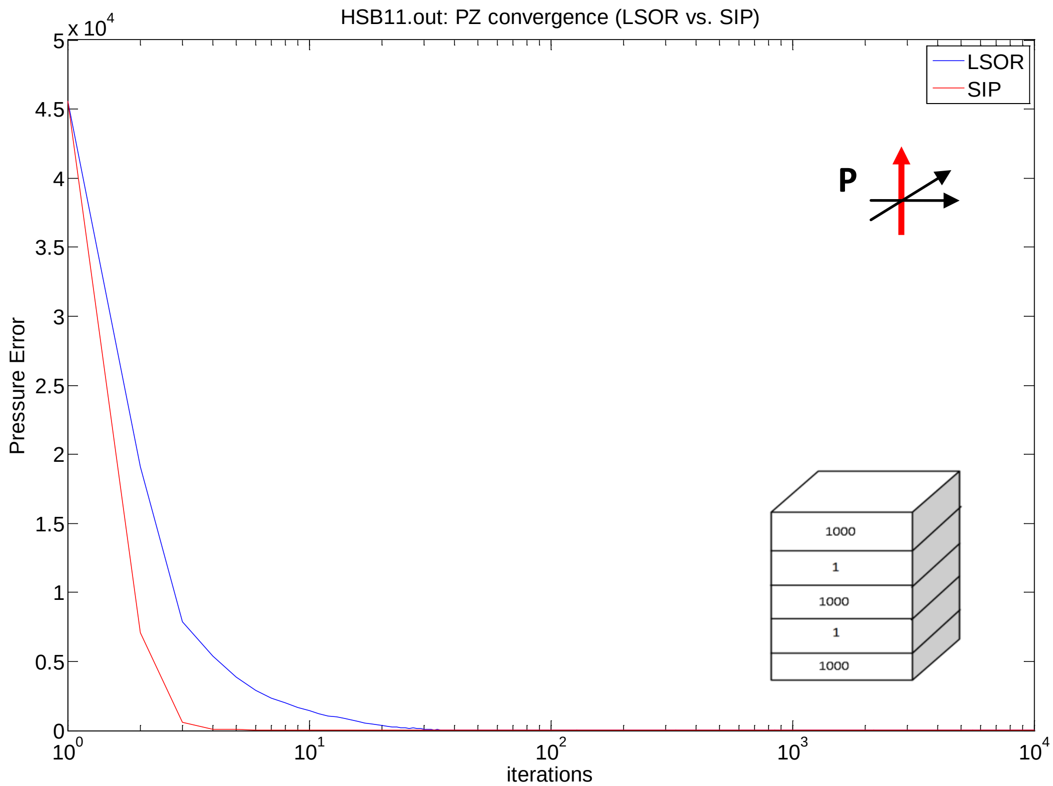


Figure 8

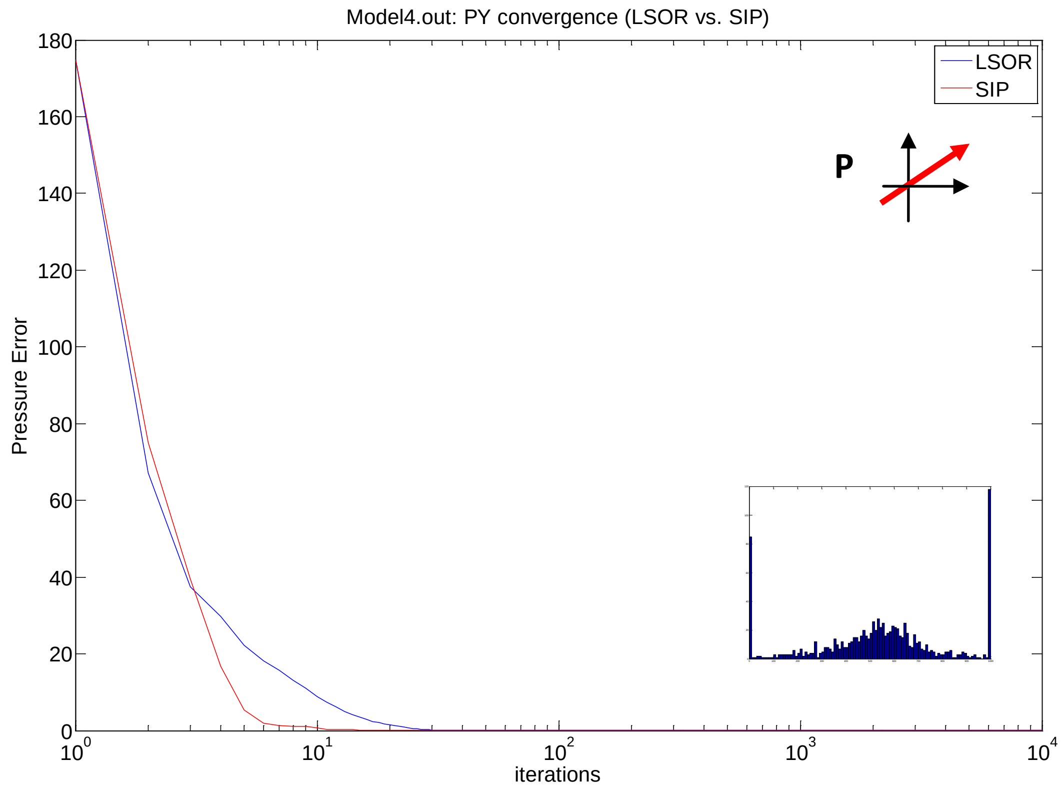


Figure 9

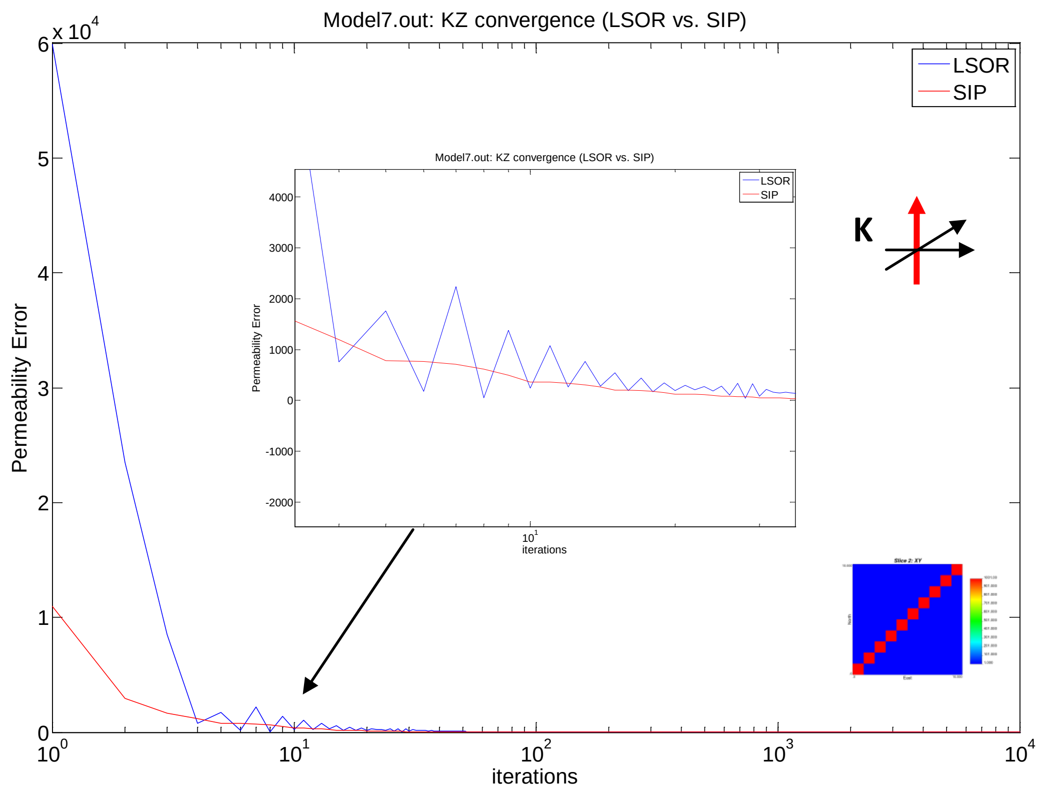


Figure 10