Petroleum and Coal

Article Open Access

MULTI-SCALE METHOD FOR MODELING AND SIMULATION OF TWO PHASE FLOW IN RESERVOIR USING MRST

M. Aslam Abdullah[1], Abhishek Panda[1], Shivam Gupta[1], Saurabh Joshi[1], Aruna Singh[1], Nagamalleswara Rao[1]*

[1] *Department of Chemical Engineering, VIT University, Vellore, India*
[2] *Centre for Disaster Management and Mitigation, VIT University, Vellore, India*

Received February 12, 2019; Accepted May 13, 2019

Abstract

The multi-scale method which is being used is generally used for solving coupled equations, and this method is widely used for reservoir simulation. The coupling factors which needs to be solved are the pressure equation and the transport equation .Both the equations are going to be solved as a decoupled system, while the pressure equation will be solved in a coarser grid and the transport equation will be solved in a finer grid. There are several multi-scale methods which can be used for reservoir simulation, but the multi-scale mixed finite element (MsMFE) method is the one which grabs all the sub grid topographical diverseness into the coarse scale using mathematical basis functions. A global formulation can be used to couple the important multi-scale information which will be grabbed by the basis functions thereby providing good resemblance to the solution for the subsurface flow.According to the literature the most commonly used formulation which is being used for the multi-scale mixed finite element method (MsMFE) for incompressible two phase flow mainly deals with common flow physics. In this paper, the formulation which is being used takes into considerations the gravity, compressibility, spatially dependent capillary and relative permeability effects. Our main aim is to find out the efficiency of this formulation and the Multi-scale mixed finite element method (MsMFE) by comparing our results with the results obtained from two different reservoirs in India. We have used MRST (MATLAB Reservoir Simulation Toolbox) for the simulation of the reservoir. Our results will include pressure distribution, Flux distribution and the saturation fields throughout the reservoir.

***Keywords**: MRST; MSMFE; Modeling; Simulation; Reservior.*

1. Introduction

During reservoir rock formation, there are many physical processes which occur on multiple time and length scales. These processes affect the movement of hydrocarbons in the subsurface rock formation. There is a lot of information that can be extracted from these different scales, i.e. time and length scales varying from micrometer scale to kilometer scale to be integrated to build multi-million cells incorporated high-resolution models. These models describing the non-homogeneous reservoir properties in proper structure can be built using contemporary reservoir characterization and geostatistical modeling techniques. It is preferred to use a coarser model with the reduced data to derive discrete flow equations rather than using highly detailed geo-cellular models to solve multi-phase flow equations as the later involves a very high computational cost. This upscaling process affects the simulation because prominent fine scale properties are lost as well as it takes a lot of time to get finished.

To simulate reservoirs having multiple scales, using the upscaling method is very cumbersome, so the need for upscaling had to be reduced. Efendiv and Hou proposed multiscale method [12] in 2009 which were designed to get definite and adequate results. This method offers an effective scheme such that the global flow equations are assimilated with effects from irresolute scale and underlying differential operators are also consistent. With this method, there was no need of computing effective coarse-scale properties, and it also promised the advantage of capturing fine scale petrophysical and geological details directly into the coarse scale simulation model.

There are several multi-scale methods applicable to reservoir simulation, and the literature contains some of it namely Dual-grid methods [6-8,14], (adaptive) local- global methods [10-11], finite-element methods [17], mixed finite-element methods [1-5,9] and finite volume multi-scale methods [15-18,30].All these methods include the basic idea of integrating fine scale into coarse scale equations. Local flow problems are solved to estimate local flow effects of petro-physical properties in fine scale and served as a fine scale sub-resolution multiscale basis function. Local or global information or a combination of both could be used as boundary conditions to localize each basis function. A reduced set of degrees-of-freedom which is affiliated with a coarser grid is computed using the global flow problems represented in terms of multi scale basis function. Flow equations can be upscaled very accurately using this technique. Alternately, pressure and mass conservative flux field can be constructed using the basis functions which is then solved using a transport solver formulated on some intermediate grid.

Upscaling method and multiscale method are very different from each other despite their few similarities. Coarse scale solutions are generated using the upscaling method whereas the main intent of the multiscale method is to obtain efficient and proper approximations on the fine or intermediate scale [13]. Many upscaling techniques use the non-physical coarse scale properties which are avoided by the multiscale method during the natural coupling between local and global scales.

Our main aim is to find out the efficiency of the formulation used in the method and the Multi-scale mixed finite element method (MsMFE) by comparing our results with the results obtained from two different reservoirs in India. We are going to use MRST (MATLAB Reservoir Simulation Toolbox) for the simulation of the reservoir. Our results will include pressure distribution, Flux distribution and the saturation fields throughout the reservoir.

2. Mathematical modeling

The coupled system of the equation that is the elliptical pressure equation and the parabolic transport equation is used for the modeling of two-phase flow in the reservoir. The transport equation which is to be used for modeling is derived from two basic equations which describe fluid flow in the reservoir [1-2,30-31]. The two basic equations are the continuity equation and Darcy’s law. The continuity equation is a mathematical representation of the law of conservation of mass whereas the Darcy’s law is being used to derive a relation between the pressure and the velocity for flow in a reservoir which is a porous material.

$$\nabla . v = q \qquad \vec{v} = -K\lambda[\nabla p_o - \tilde{g}(S_w)\nabla z + h(S_w)\nabla p_c] \qquad (1)$$

$$\phi \partial S_w/\partial t + \nabla . f_w(S_w)[v + K\lambda_o(S_w)\left((\rho_w - \rho_o)g\nabla z + \nabla pc\left((S_w)\right)\right] = \frac{q_w}{\rho_w} \qquad (2)$$

The parameters which are being used in the above-mentioned equations are as follows: S_w represents the saturation of water in the porous reservoir; S_o represents the saturation of oil in the porous reservoir, $S_w+S_o=1$. Hence, we can say that $S_w,S_o \leq 1$. f_w represents fractional flow function of water and f_o represents fractional flow function of oil where $f_{wfff}=\lambda_w/\lambda$. λ_w represents phase mobility of water and λ_o represents phase mobility of oil and λ represents the total mobility. $\lambda = \lambda_w + \lambda_o$. u_w represents the velocity of water whereas u_o represents the velocity of oil and $u = u_w + u_o$ where u is the total velocity. ϕ represents the porosityand K is the permeability tensor. ρ_w represents the density of water and ρ_o represents the density of oil. g is the acceleration due to gravity. P_w is the pressure of water and p_o is the pressure exerted by oil. p_c is the capillary pressure, $p_c=p_o-p_w$ .q is the source/sink term in the above equation $\tilde{g}(S_w) = (f_w(S_w)\rho_w + f_o(S_w)\rho_o)g$ and $h(S_w) = f_w(S_w)$.

3. Discretization (finite scale)

Discretization of the computational domain Ω leads to a set of {C_i} of N non overlapping and matching polyhedral cells. There will be a sequential solving of flow and transport equation. First flow equation will be solved to get explicit fluxes at the cell interfaces which is used to evolve the saturations in time step Δt using the transport equation. Following unknowns will be used in the paper: vector of outward fluxes from the cell C_i is denoted by u_i, the pressure at the cell center is denoted by $p_{i,}$ and the pressure at the cell faces is denoted by π_i. To relate the three quantities, we will use the Darcy's law through a matrix T_i of one-sided transmissibilities.

$$ui = T_i[e_i p_i - \pi_i - \tilde{g}(S_i)\Delta z_i + h(S_i)(e_i p_c(\overrightarrow{x_v}, S_i) - p_{ci})],\ e_i = (1, \dots\dots.1)^T \tag{3}$$

The vector of differences in the z coordinate of the cell center $\overrightarrow{xi}$ and and the face centroids is denoted by Δz_i and the capillary pressure p_{ci} at the cell faces is defined as the linear interpolation of the capillary pressure in the neighboring cells. On the other hand, one can utilize multipoint techniques to characterize transmissibilities on organized and unstructured lattices utilizing multi-point fluxes, see e.g., [26-27,29,31].

The following discrete linear system for global flow problem can be derived by augmenting eq. 3 with pressure and flux continuity across cell faces.

$$\begin{bmatrix} \boldsymbol{B} & C & D \\ C^T & 0 & 0 \\ D^T & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ -p \\ \pi \end{bmatrix} = \begin{bmatrix} -G(S)\Delta z + H(S)\Delta p_c \\ q \\ 0 \end{bmatrix} \tag{4}$$

Here, Cell wise ordered outward face fluxes is denoted by u, the pressure at the cell center and pressure at the cell faces is denoted by p and π respectively.

The first row within the block matrix equation corresponds to eq. 3 for all grid cells and the right-hand terms $G(S)\Delta z$ and $H(S)\Delta p_c$ are the one-sided face contributions similar to gravity and capillary effects. The matrix B represents block diagonal with single block T_i^{-1} defined per cell. The matrix C is additionally block diagonal and has one block entry e_i per cell that duplicates the cell pressures to one value for every cell face. Hence, within the second row of eq. 4, the converse matrix C^T sums the face fluxes to outline one mass-conservation equation per cell. Finally, every column of D corresponds to a singular face and has one (for boundary faces) or two (for interior faces) unit entries resembling the index of the face within the cell wise ordering. In the following, the transport equation, eq. 2 is solved on the fine-scale employing a normal transport problem solver with upstream- weighted mobilities and two-point discretization of the second-order capillary term. The temporal discretization could also be explicit or implicit. If needed, improved numerical accuracy can be obtained by exploiting higher-order upwind schemes, just like the wave-oriented multi-dimensional schemes Lamine and Edwards [20-21].

4. Multi-scale mixed finite elements

The main concept behind using multiscale mixed finite-element method is to build a special approximation space consisting of a set of coarse-scale basic functions that meet the local flow equation. Therefore, the MsMFE method is formulated on the basis of two hierarchically nested grids. Rock and rock fluid properties are displayed on the geo-cellular fine-scale grid, while the basic functions and corresponding degrees of freedom are associated with a coarse simulation grid used to solve the problem of global flow. The coarse grid blocks are defined as a connected set of fine grid cells and can have arbitrary shapes in principle. However, if the blocks are somewhat regular, the best numerical resolution is obtained, following the layered structures of stratigraphic grids [2] or adapting to high-contrast characteristics [24].

4.1. Approximations for multiscale method

We begin by writing the solution to Eq 4 to formally define the MsMFE method as the sum of the basic functions and a residual on a fine scale.

$$u = \psi u_c + \tilde{u} \quad , \qquad p = \phi p_c + \tilde{p}, \qquad \pi = \Pi \pi_c + \tilde{\pi} \tag{5}$$

Here, the vector of outward fluxes over the coarse-block interfaces is denoted by u_c; p_c denotes the coarse-block pressure vector, and π denotes the coarse-block face pressure vector. Similarly $\tilde{u}$, $\tilde{p}$ and $\tilde{\pi}$ represent the reminder which have variations on the fine grid. The fine scale reconstruction operators for $\vec{v}$, p and π is denoted by the matrices Ψ, Φ, and π. Each column in Ψ corresponds to a multi-scale base function for the flux associated with a unique coarse-grid face and is represented as a fine-scale flux vector of n_f x 1.

For incompressible flow, the pressure is rarely explicitly used, with the exception of determining well rates by using appropriate good models. Therefore, we define the pressure within each coarse block to be constant and replace Φ with a simple extension operator I, which maps the block cells with a constant value from each coarse block. Similarly, an extension operator J replaces ∏, which maps a constant value from each coarse face to the individual cell faces of the coarse face. A reconstruction operator R= diag(ψ , I, J) is thus defined which makes it possible to map the degrees of freedom x_c=[u_c,-p_c, π_c] on the coarse scale to the corresponding fine scale quantities x= [u,-p, π].

In case of coarse system In order to create a global system on the coarse grid, we need a compression operator to bring the Eq 4 which is a fine scale system to the space covered by our multi-scale functions. R^T is a natural choice here since the transposed operators I and J correspond to the sum of all fine cells of a coarse block and all fine cell faces, which are part of the faces of the coarse blocks. Multiply eq. 4 from the left with R^T, substitute x=Rx_c and rearrange the terms to get

$$\begin{bmatrix} \psi^T B\psi & \psi^T CI & \psi^T DJ \\ I^T C^T \psi & 0 & 0 \\ J^T D^T \psi & 0 & 0 \end{bmatrix} \begin{bmatrix} u_c \\ -p_c \\ \pi_c \end{bmatrix} = \begin{bmatrix} \psi^T (H(S)\Delta p_c - G(S)\Delta z) - \psi^T (B\tilde{u} - C\tilde{p} + D\tilde{\pi}) \\ I^T q - I^T C^T \tilde{u} \\ -J^T D^T \tilde{\pi} \end{bmatrix} \tag{6}$$

The terms of the fine- scale reminder can be eliminated as follows: If we interpret the coarse- scale pressure as the w- weighted average of the real pressure, $p_c^i = \int_{B_i} wpd\vec{x}$(where w is the source term used to define basis function) then $\tilde{p}$ disappears. The following coarse-scale system is obtained after neglecting the terms $\tilde{u}$ and $\tilde{\pi}$.

4.2. Methods to find Multiscale basis function

$$\begin{bmatrix} \psi^T B\psi & \psi^T CI & \psi^T DJ \\ I^T C^T \psi & 0 & 0 \\ J^T D^T \psi & 0 & 0 \end{bmatrix} \begin{bmatrix} u_c \\ -p_c \\ \pi_c \end{bmatrix} = \begin{bmatrix} \psi^T (H(S)\Delta p_c - G(S)\Delta z) \\ I^T q \\ 0 \end{bmatrix} \tag{7}$$

There are two different ways of calculating basic functions: In the single-block method, fine-scale fluxes must be specified over the coarse interface associated with the basis function. The method is not very accurate unless some kind of global flow information is included in the interface flux. Therefore, we will use a two-block method which does not impose any condition on the interface between two coarse blocks. The resulting method is not convergent, but usually gives reasonable precision on finite grids. In order to define the method, we consider two blocks of B_i and B_j, which share the common coarse face $\Gamma_{ij}=\partial B_i \cap \partial B_j$ and let B_{ij} be a sub-set of Ω containing B_i and B_j. The two- block multiscale base function is defined after neglecting the influence of gravity and capillary forces as:

$$\psi_{ij} = -K\nabla\phi_{ij}, \qquad \nabla.\psi_{ij} = w_{ij}(\overrightarrow{x}) = \begin{cases} w_i(\overrightarrow{x}), & if\ \vec{x} \in B_i, \\ -w_j(\overrightarrow{x}), & if\ \vec{x}\ \in B_j, \\ 0, & otherwise, \end{cases} \tag{8}$$

If we consider capillary forces, then the following basis function must be added to the already formed set,

$$\psi^c{}_{ij} = -K\left(\nabla\phi^c{}_{ij} - h(S)\nabla p_c(S)\right), \qquad \nabla.\psi^c{}_{ij} = 0 \tag{9}$$

5. Codes and results

Creating a grid. Here we have created a 200-200-50 rectangular reservoir in 10-10-10 Cartesian grid (Figure 1).

```
[nx, ny, nz] = deal( 10,  10, 10);
[Dx, Dy, Dz] = deal(200, 200, 50);
G = cartGrid([nx, ny, nz], [Dx, Dy, Dz]);
```

```
G = computeGeometry(G);

plotGrid(G), view(3), axis tight
```

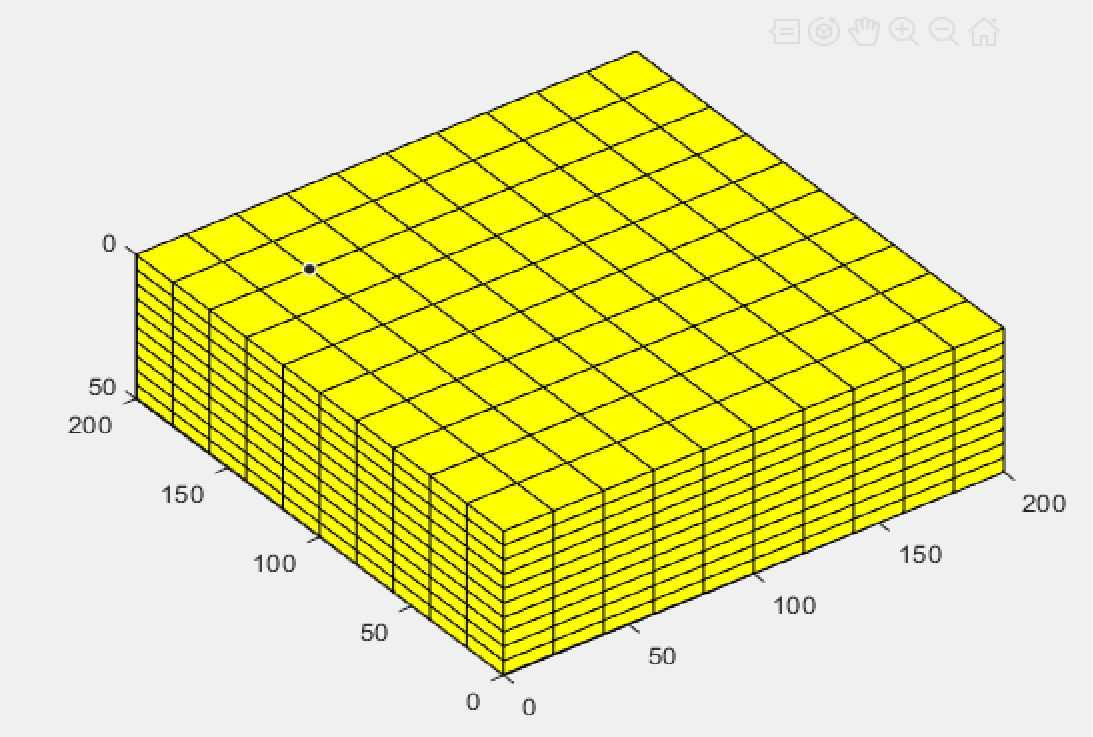

Figure 1. 200-200-50 rectangular reservoir in 10-10-10 Cartesian grid

We assume a ‘Horizontal’ well which is drilled into 6 grids of reservoir. The theoretical representation of the reservoir is mentioned. (Figure 2)

```
nperf = 6;
I = repmat(2, [nperf, 1]);
J = (1 : nperf).' + 1;
K = repmat(5, [nperf, 1]);

cellInx = sub2ind(G.cartDims, I, J, K);
W = addWell([], G, rock, cellInx, 'Name', 'P1', 'Dir', 'y' );
gravity reset on, g = norm(gravity);
[z_0, z_max] = deal(0, max(G.cells.centroids(:,3)));
equil  = ode23(@(z,p) g .* rho(p), [z_0, z_max], p_r);
p_init = reshape(deval(equil, G.cells.centroids(:,3)), [], 1);  clear equil
clf
show = true([G.cells.num, 1]);
cellInx = sub2ind(G.cartDims, ...
    [I-1; I-1; I; I;   I(1:2) - 1], ...
    [J  ; J;   J; J;   nperf  + [2 ; 2]], ...
    [K-1; K;   K; K-1; K(1:2) - [0 ; 1]]);

show(cellInx) = false;

plotCellData(G, convertTo(p_init, barsa), show, 'EdgeColor', 'k')
plotWell(G, W, 'height', 10)
view(-125, 20), camproj perspective
```

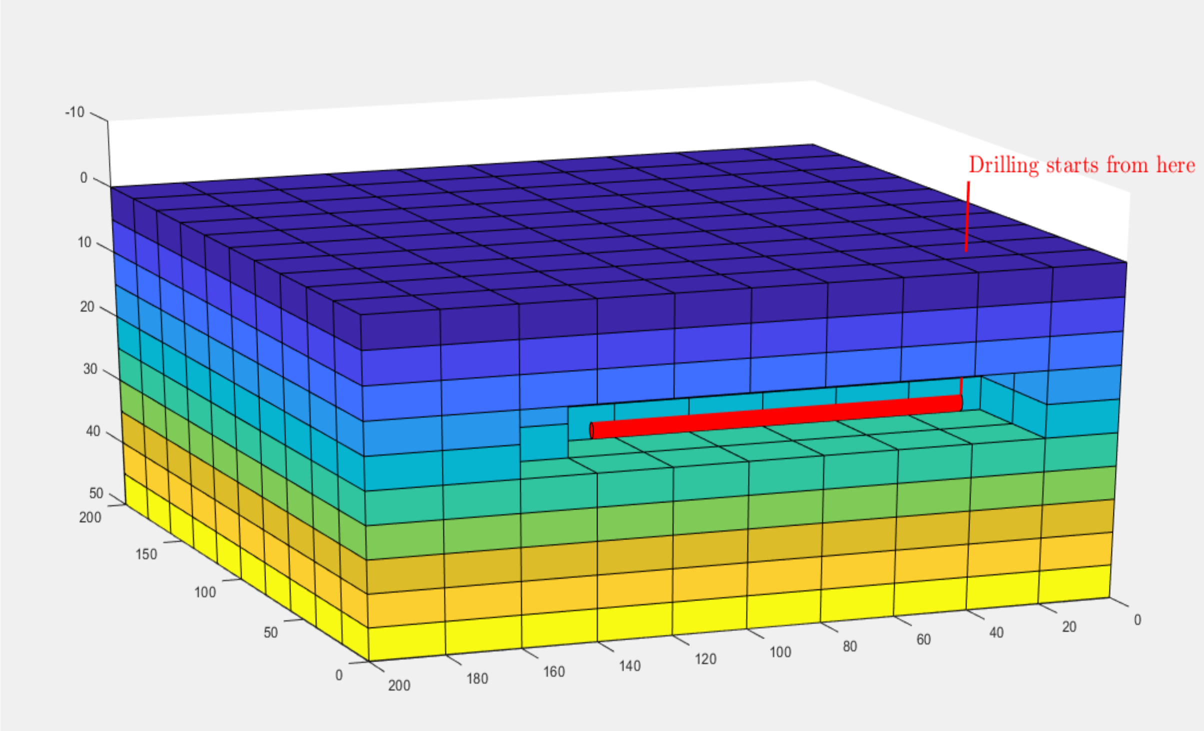

Figure 2. Theoretical representation of model of Reservoir having 'Horizontal Well'

Constructing in a Cartesian grid with assuming 200 mD permeability and 1 viscosity and 1000 density (Figure 3).

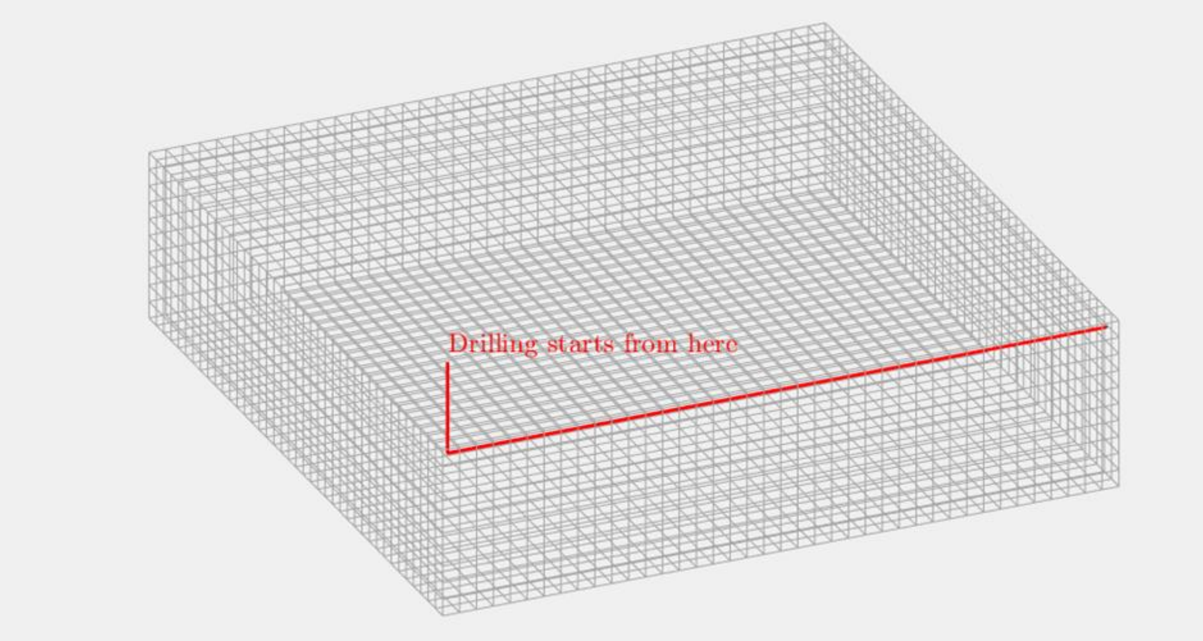

Figure 3. Cartesian grid with assuming 200 mD permeability and 1 viscosity and 1000 density

Plotting the pressure distribution at our reservoir (Figure 4). Assuming 200 bar or 2900 psi at top of reservoir.

```
[nx, ny, nz] = deal( 10,  10, 10);
[Dx, Dy, Dz] = deal(200, 200, 50);
```

```
gravity reset on
G = cartGrid([nx, ny, nz], [Dx, Dy, Dz]);
G          = computeGeometry(G);
rock.perm  = repmat(0.2*darcy(), [G.cells.num, 1]);
fluid = initSimpleFluid('mu' , [    1,   10]*centi*poise     , ...
                            'rho', [1014, 859]*kilogram/meter^3, ...
                            'n'   , [    2,    2]);bc           = pside([], G,
'TOP', 100.*barsa);
S    = computeMimeticIP(G, rock);
sol = incompMimetic(initResSol(G , 0.0), G, S, fluid, 'bc', bc);
newplot;
subplot(8, 1, [1 3])
   plotCellData(G, convertTo(sol.pressure(1:G.cells.num), barsa), ...
               'EdgeColor', 'k');
   set(gca, 'ZDir', 'reverse'), title('Pressure Distribution')
   view(45,5), cx = caxis; colorbar
```

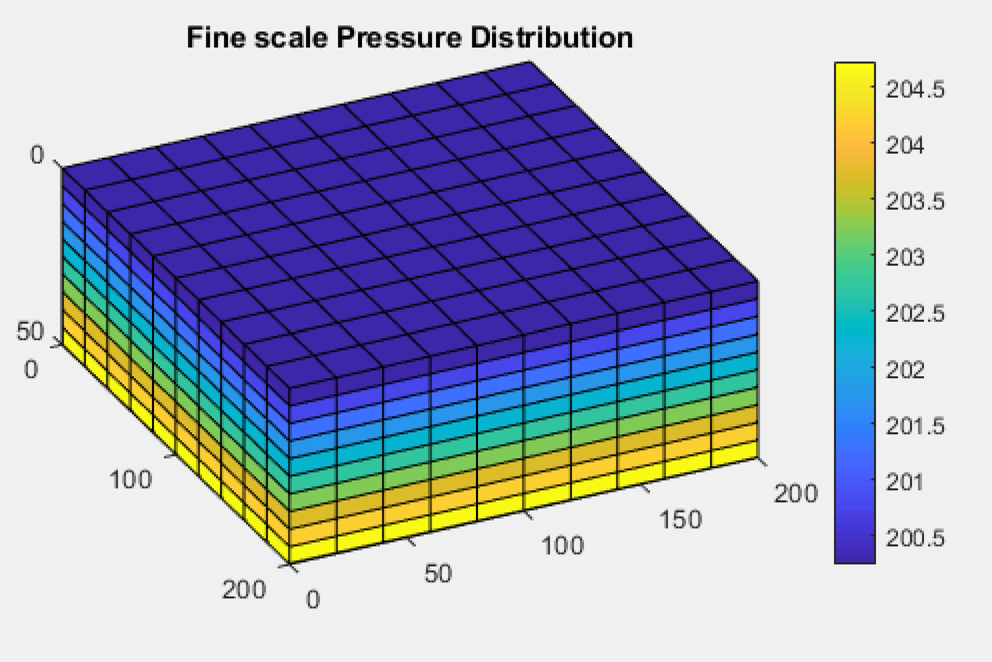

Figure 4. Fine scale pressure distribution

For the multiscale system, plotting the coarse scale solution (Figure 5).

Multiscale only captures coarse-scale gravity effects. To get fine scale result, we are adding finescale equation which we did earlier to solution of multiscale effect, which gives us following result –

```
p  = partitionUI(G, [Nx, Ny, Nz]);
p  = processPartition  (G, p);
CG = generateCoarseGrid(G, p);

CS = generateCoarseSystem(G, rock, S, CG, ones([G.cells.num, 1]),'bc', bc);
xrMs = solveIncompFlowMS (initResSol(G, 0.0), G, CG, p, ...
                          S, CS, fluid, 'bc', bc);
subplot(8, 1, [1 3])
   plotCellData(G, convertTo(xrMs.pressure(1:G.cells.num), barsa), ...
               'EdgeColor', 'k');
   set(gca, 'ZDir', 'reverse'); title('Coarse scale Pressure Distribution')
```

```
    view(45,5); caxis(cx); colorbar

subplot(3, 2, [5 6]);
    plot(1:nz, convertTo(sol .pressure(1:nx*ny:nx*ny*nz), barsa()), '-o',...
           1:nz, convertTo(xrMs.pressure(1:nx*ny:nx*ny*nz), barsa()), '-*');
    legend('fine','coarse','Location','NorthWest');
```

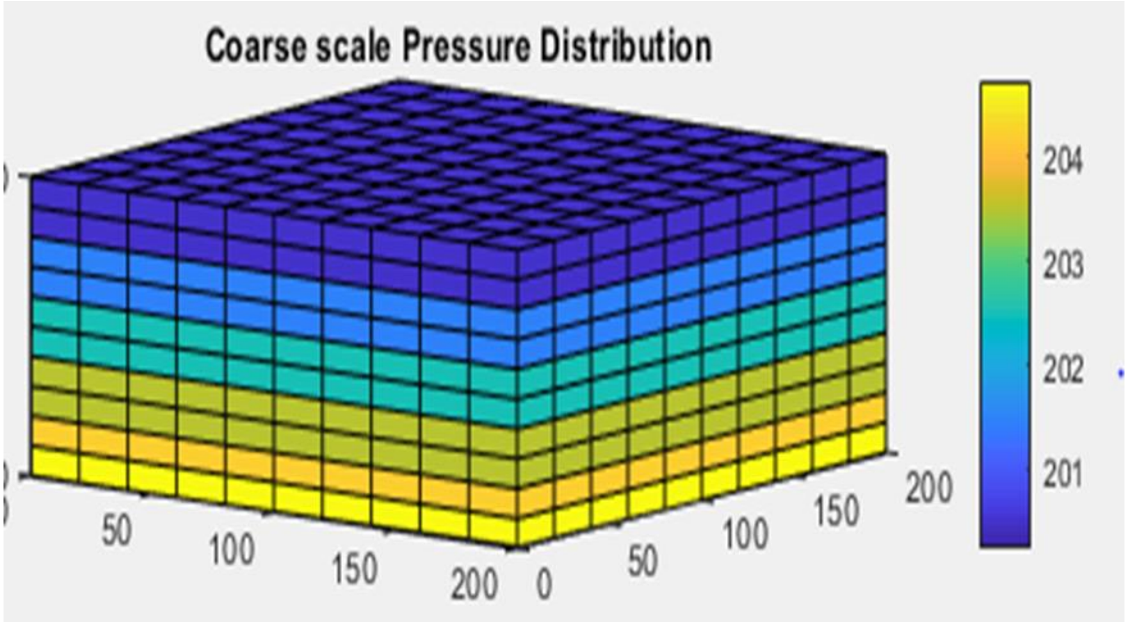

Figure 5. Coarse scale pressure distribution

Now if we compare both, we get the following result. We get the following pressure distribution per grid (Figure 6)

```
subplot(8, 1, [1 3]);
   plot(1:nz, convertTo(sol .pressure(1:nx*ny:nx*ny*nz), barsa()), '-o',...
        1:nz, convertTo(xrMs.pressure(1:nx*ny:nx*ny*nz), barsa()), '-*');
   legend('fine','coarse','Location','NorthWest');
```

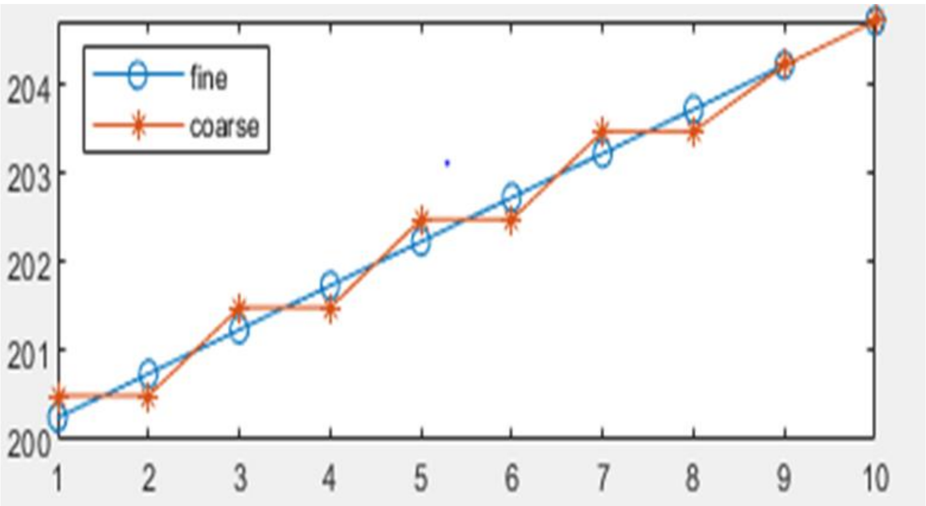

Figure 6. Pressure distribution grid

Now we are checking Permeability distribution over reservoir (Figure 7). We use boundary condition of flux of 1 on the LHS and Dirichlet boundary conditions p = 0 on RHS of grid.

```
verbose = true;
[nx, ny, nz] = deal( 10,  10, 10);
[Dx, Dy, Dz] = deal(200, 200, 50);
G = cartGrid([nx, ny, nz], [Dx, Dy, Dz]);
G = computeGeometry(G);
K          = logNormLayers([nx, ny, nz], 1); K = 10 * K / mean(K(:));
rock.perm = bsxfun(@times, [10, 1, 0.1], convertFrom(K, milli*darcy()));
fluid = initSimpleFluid('mu' , [    1,  10]*centi*poise     , ...
                         'rho', [1014, 859]*kilogram/meter^3, ...
                         'n'  , [    2,    2]);
gravity off
bc = fluxside([], G, 'LEFT',  100*meter()^3/day());
bc = pside    (bc, G, 'RIGHT', 0);
newplot
plotCellData(G,log10(convertTo(rock.perm(:,1),milli*darcy))); shading  fac-
eted;
title('Permeability Distribution');
view(3), camproj perspective, axis tight off
cs = [50 100:100:1000];
h=colorbar;
```

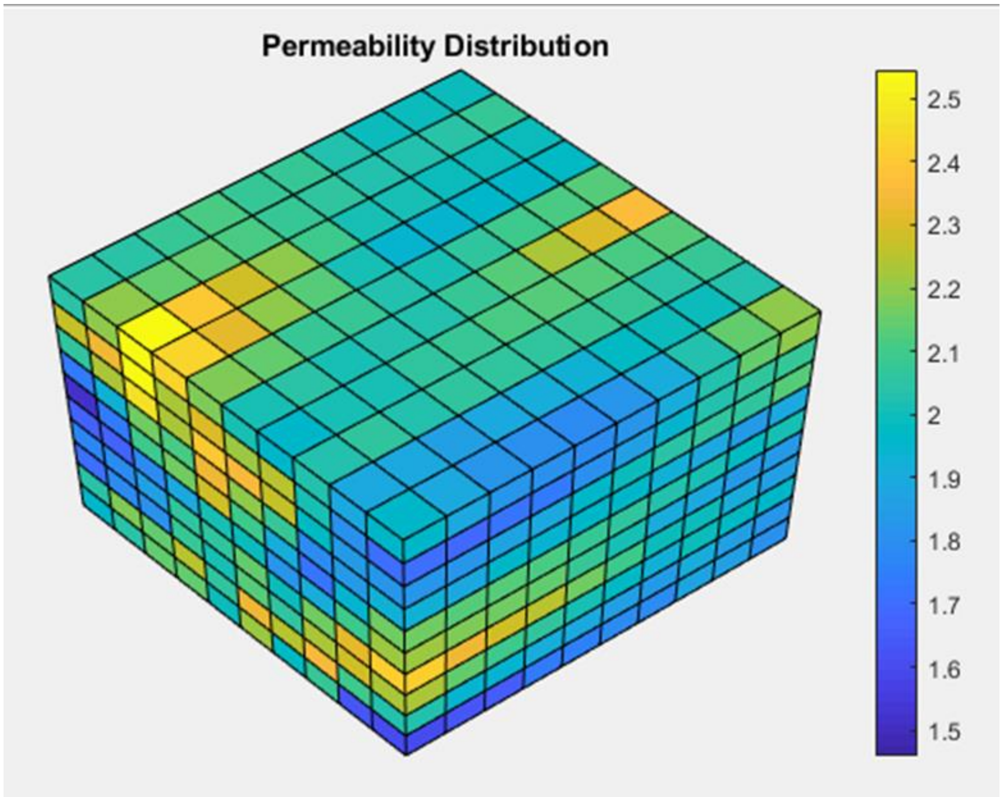

Figure 7. Permeability distribution

Then we find flux intensity distribution (Figure 8)

```
p  = partitionUI(G, [Nx, Ny, Nz]);
p  = processPartition(G, p);
```

```
CG    =    generateCoarseGrid(G,    p,    'Verbose',    verbose);display(CG);
disp(CG.cells); disp(CG.faces);
S  = computeMimeticIP(G, rock, 'Verbose', verbose);
display(S);
CS = generateCoarseSystem(G, rock, S, CG, ones([G.cells.num, 1]), ...
                         'Verbose', verbose, 'bc', bc);


newplot;
subplot(1,2,1),
cellNo  = rldecode(1:G.cells.num, diff(G.cells.facePos), 2) .';
C       = sparse(1:numel(cellNo), cellNo, 1);
D       = sparse(1:numel(cellNo), double(G.cells.faces(:,1)), 1, ...
                 numel(cellNo), G.faces.num);

   spy([S.BI                  , C         , D        ; ...
        C', zeros(size(C,2), size(C,2) + size(D,2)); ...
        D', zeros(size(D,2), size(C,2) + size(D,2))]);

   xRef = incompMimetic    (initResSol(G, 0.0),  G, S, fluid, ...
                        'bc', bc, 'MatrixOutput',true);
xMs  = solveIncompFlowMS(initResSol(G, 0.0), G, CG, p, S, CS, fluid, ...
                        'bc', bc, 'MatrixOutput', true, 'Solver', 'hybrid');
 clf
plot_var  = @(x) plotCellData(G, x);
plot_pres = @(x) plot_var(convertTo(x.pressure(1:G.cells.num), barsa()));
plot_flux = @(x) plot_var(accumarray(cellNo, ...
   abs(convertTo(faceFlux2cellFlux(G, x.flux), meter^3/day))));
subplot('Position',[0.02 0.02 0.46 0.42]),
   plot_flux(xRef); title('Flux intensity Distribution')
   view(3), camproj perspective, axis tight equal, camlight headlight
   cax2 = caxis; colorbar
```

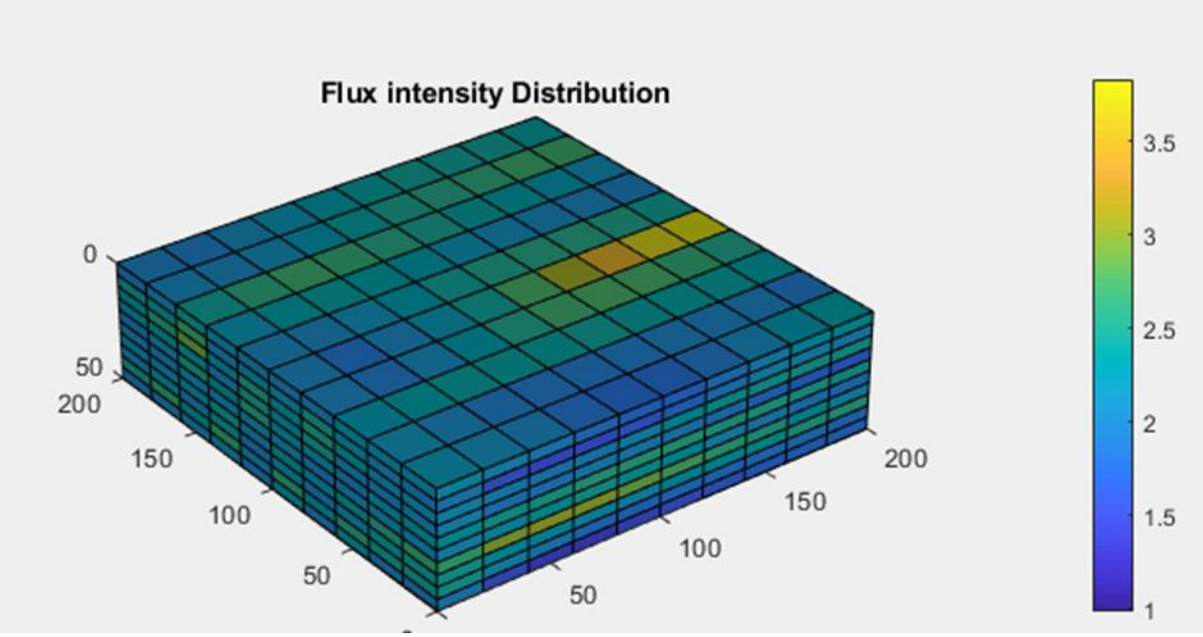

Figure 8. Flux intensity distribution

Now after trying our Simulation on 'Cartesian grid', we simulated our reservoir on 'Corner-point grid' aswell, for experimental purposes (Figure 9).

We have used same values for all properties like we did on Cartesian grid. For boundary condition: we used Dirchlet condition for 1 bar pressure on LHS and 0 bar at RHS.

```
nx = 10; ny = 10; nz = 10;
Nx = 200; Ny = 200; Nz = 50;
verbose = true;
grdecl  = simpleGrdecl([nx, ny, nz], 0.15);
G       = processGRDECL(grdecl); clear grdecl;
G       = computeGeometry(G);
[rock.perm, L] = logNormLayers([nx, ny, nz], [100, 400, 50, 500]);
fluid = initSimpleFluid('mu' , [   1,  10]*centi*poise     , ...
                            'rho', [1014, 859]*kilogram/meter^3, ...
                            'n'  , [   2,   2]);
rock.perm      = convertFrom(rock.perm, milli*darcy);
westFaces = find(G.faces.centroids(:,1) == 0);
bc        = addBC([], westFaces, 'pressure',          ...
                     repmat(1*barsa(), [numel(westFaces), 1]));

xMax      = max(G.faces.centroids(:,1));
eastFaces = find(G.faces.centroids(:,1) == xMax);
bc        = addBC(bc, eastFaces, 'pressure',          ...
                  repmat(0, [numel(eastFaces), 1]));
clf
   plotCellData(G, log10(rock.perm(:))); shading faceted
   title('Reservoir using Corner-point grid')
   view(3), camproj perspective, axis tight off, camlight headlight

   set(h, 'YTick', c, 'YTickLabel', num2str(10.^c'));
```

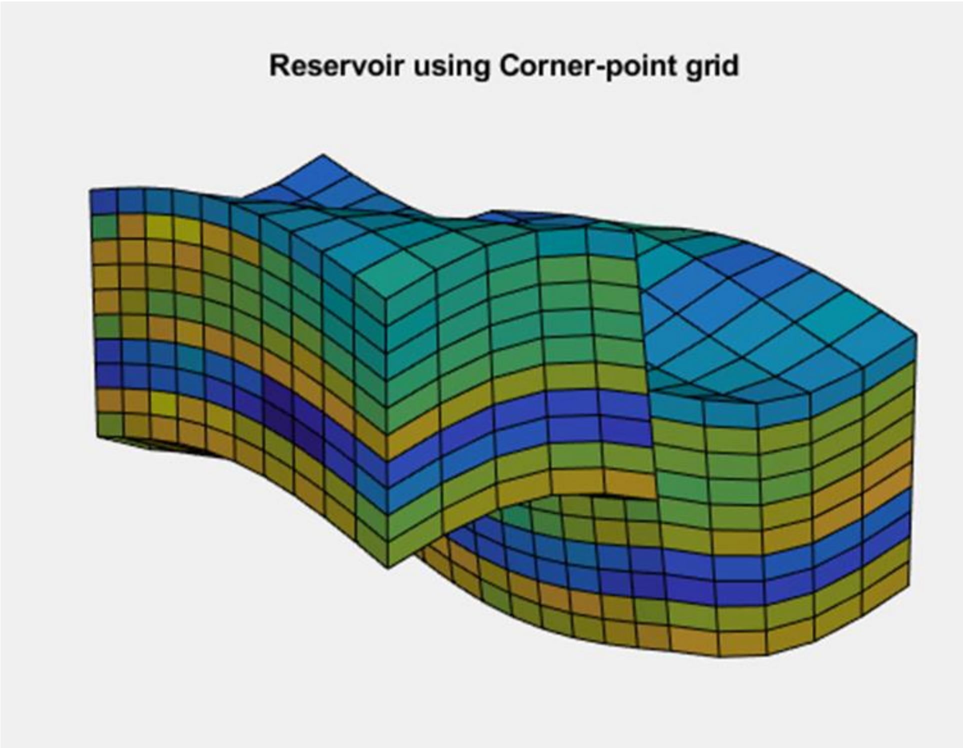

Figure 9. Reservoir using Corner-point grid

After this, we have also plotted our reservoir using Multiscale Finite Volume method with experimental data (Figure 10).

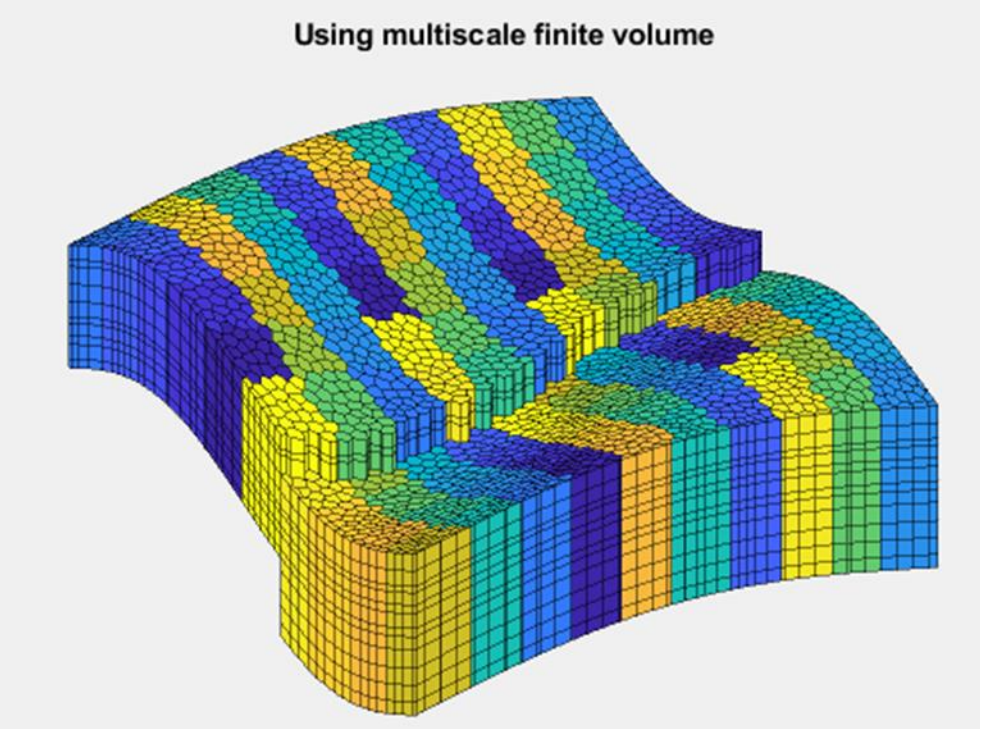

Figure 10. Reservoir using Multiscale Finite Volume

6. Conclusion

In this paper, we have modified multi scale mixed finite element method (MsFEM) for incompressible flow including two phases. We have also implemented the same for simulation of two-phase flow in a porous reservoir and we have compared our results with the results obtained from highly reputed companies from India.

References

[1] Aarnes JE. On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation. Multiscale Modeling & Simulation, 2004; 2(3):421-39.

[2] Aarnes JE, Efendiev Y. A multiscale method for modeling transport in porous media on unstructured corner-point grids. Journal of Algorithms & Computational Technology, 2008 Jun; 2(2):299-318.

[3] Aarnes JE, Krogstad S, Lie KA. Multiscale mixed/mimetic methods on corner-point grids. Computational Geosciences, 2008 Sep 1; 12(3):297-315.

[4] AGMG 2012. Iterative solution with AGgregation-based algebraic MultiGrid. http://homepages.ulb.ac.be/~ynotay/AGMG/.

[5] Alpak FO, Pal M, Lie KA. A multiscale adaptive local-global method for modeling flow in stratigraphically complex reservoirs. SPE Journal, 2012; 17(04):1-056.

[6] Arbogast T. Implementation of a locally conservative numerical subgrid upscaling scheme for two-phase Darcy flow. Computational Geosciences, 2002;6(3-4):453-81

[7] Arbogast T, Bryant SL. A Two-Scale Numerical Subgrid Technique for Waterflood Simulations. SPEJ, 7(4): 446-457. SPE-81909-PA; 2002.

[8] Audigane P, Blunt MJ. Dual mesh method for upscaling in waterflood simulation. Transport in Porous Media, 2004; 55(1):71-89.

[9] Chen Z, Hou T. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. Mathematics of Computation, 2003; 72(242):541-76.

[10] Chen Y, Durlofsky LJ. Adaptive local–global upscaling for general flow scenarios in heterogeneous formations. Transport in porous Media, 2006; 62(2):157-85.

[11] Chen Y, Durlofsky LJ, Gerritsen M, Wen XH. A coupled local–global upscaling approach for simulating flow in highly heterogeneous formations. Advances in Water Resources, 2003; 26(10):1041-60.

[12] Efendiev Y, Hou TY. Multiscale finite element methods: theory and applications. Springer Science & Business Media; 2009 Jan 10.

[13] Farmer CL. Upscaling: a review. International journal for numerical methods in fluids, 2002; 40(1-2):63-78.

[14] Guérillot DR, Verdiere S. Different pressure grids for reservoir simulation in heterogeneous reservoirs. InSPE Reservoir Simulation Symposium 1995 Jan 1. Society of Petroleum Engineers.

[15] Hajibeygi H, Jenny P. Multiscale finite-volume method for parabolic problems arising from compressible multiphase flow in porous media. Journal of Computational Physics, 2009; 228(14):5129-47.

[16] Hajibeygi H, Jenny P. Adaptive iterative multiscale finite volume method. Journal of Computational Physics, 2011; 230(3):628-43.

[17] Hou TY, Wu XH. A multiscale finite element method for elliptic problems in composite materials and porous media. Journal of computational physics, 1997; 134(1):169-89.

[18] Jenny P, Lee SH, Tchelepi HA. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. Journal of Computational Physics, 2003; 187(1):47-67.

[19] Kippe V, Aarnes JE, Lie KA. A comparison of multiscale methods for elliptic problems in porous media flow. Computational Geosciences, 2008; 12(3):377-98.

[20] Lamine MS, Edwards MG. Higher order multidimensional wave oriented upwind schemes for flow in porous media on unstructured grids. InSPE Reservoir Simulation Symposium 2009 Jan 1. Society of Petroleum Engineers.

[21] Lamine S, Edwards MG. Higher order multidimensional upwind convection schemes for flow in porous media on structured and unstructured quadrilateral grids. SIAM Journal on Scientific Computing. 2010 Apr 14;32(3):1119-39.

[22] Lee SH, Wolfsteiner C, Tchelepi HA. Multiscale finite-volume formulation for multiphase flow in porous media: black oil formulation of compressible, three-phase flow with gravity. Computational Geosciences, 2008; 12(3):351-66.

[23] Lie KA, Krogstad S, Ligaarden IS, Natvig JR, Nilsen HM, Skaflestad B. Open-source MATLAB implementation of consistent discretisations on complex grids. Computational Geosciences, 2012; 16(2):297-322.

[24] Natvig JR, Lie KA, Krogstad S, Yang Y, Wu XH. Grid adaption for upscaling and multiscale methods. In Proceedings of ECMOR XIII–13th European Conference on the Mathematics of Oil Recovery 2012 Sep 13.

[25] Notay Y. An aggregation-based algebraic multigrid method. Electronic transactions on numerical analysis, 2010; 37(6):123-46.

[26] Pal M, Edwards MG. Quasimonotonic continuous Darcy-flux approximation for general 3d grids of any element type. InSPE Reservoir Simulation Symposium 2007 Jan 1. Society of Petroleum Engineers.

[27] Pal M, Edwards MG, Lamb AR. Convergence study of a family of flux-continuous, finite-volume schemes for the general tensor pressure equation. International journal for numerical methods in fluids, 2006; 51(9-10):1177-203.

[28] Pal M, Lamine S, Lie KA, Krogstad S. Multiscale method for two and three-phase flow simulation in subsurface petroleum reservoirs. In ECMOR XIII-13th European Conference on the Mathematics of Oil Recovery 2012 Sep 10.

[29] Parramore ET, Edwards MG, Pal M, Lamine S. CVD-MPFA based multiscale formulation on structured and unstructured grids. In ECMOR XIII-13th European Conference on the Mathematics of Oil Recovery 2012 Sep 10.

[30] SINTEF 2011. Matlab Reservoir Simulation Toolbox (MRST), version 2011a. http://www.sintef.no/MRST.

[31] Wheeler MF, Xue G, Yotov I. A multiscale mortar multipoint flux mixed finite element method. ESAIM: Mathematical Modeling and Numerical Analysis, 2012; 46(4):759-796.

To whom correspondence should be addressed: Dr. M. Aslam Abdullah, Department of Chemical Engineering, VIT University, Vellore, India