Flow Simulation on Unstructured Grids

S. Fatemeh Razavi, Juliana Leung, Jeff Boisvert

This paper describes how the finite volume form of mass conservation law is applied to model single phase flow in a 3D discrete fracture model meshed by unstructured tetrahedral grids. Unstructured grids aid to handle complex fracture media more precisely in comparison with structured grids. By applying a simplified model, transmissibility is calculated for all connections in the fractured model including matrix-matrix, fracture-fracture and matrix-fracture connections. Therefore flow is measured along and between fractures. The flow simulator will be predominantly based on connectivity lists. Creating the connectivity lists requires significant preprocessing efforts especially for a 3D model. Making the connectivity lists and data lists are described in details in this paper.

1. Introduction

Flow modeling in fractured media is necessary to predict and manage reservoir performance. Fractured model should be able to determine the flow through fractures and the interaction between the fractures and the matrix. In fractured media, numerical simulation will be more challenging because of large difference in matrix and fracture permeabilties and as a result, the permeability distribution is not homogenous. This will cause irregular transport and could be captured precisely on a discretized model using convection diffusion equation.

Numerical representation of fractured media is done by using two simplified model used for flow simulation (Bajaj, 2009).

1) Dual porosity, dual permeability model

In this model, two continuums are considered to represent fracture network and matrix independently. Flow occurs mainly in the fracture network and intersection of matrix and fracture is simulated by a transfer function. Evaluating the transfer function is not simple and will cause in inaccurate flow predictions.

2) Discrete fracture model

In this model, effect of each fracture on fluid flow is accounted by clear representing of each fracture individually and high and low permeability fractures and matrix-fracture connectivity are modeled realistically. Fractures will be represented by lines and planes in 2-D and 3-D respectively. Therefore, flow simulations through fractured media could be done more accurately by using discrete fracture model.

Finite volume and finite element techniques are two methods mostly applied for discritization purpose in the case of unstructured grids. Finite volume method is computationally more efficient and ensures mass conservation while it is as easy as finite difference scheme with higher accuracy. That's why; finite volume is the preferable method for flow simulation (Bajaj, 2009). By applying finite volume scheme fractured porous media is divided to control volumes and the location of variables (grid points) are usually defined as the barycenter of each grid cell (control volume). Boundary nodes are added in a separate step.

In this work, goal is to describe implementing a 3D discrete fracture model which is based on finite volume method for single phase flow simulation.

2. Mathematical background and Numerical Modeling

Goal is finding the solution of partial differential equation named pressure equation which is describing flow in subsurface. Pressure equation is obtained based on the fundamental law of mass conservation and Darcy's law (Bajaj, 2009). The focus of this paper is on single phase flow

2.1. Continuity equation

"Conservation laws are resulting by considering a known quantity of control mass (CM) and its extensive properties such as mass, momentum and energy. We have to deal with the flow within a certain spatial region called control volume (CV) rather than a control mass. So, the conservation laws should be transformed into a CV form and with the intensive variables rather than extensive properties which this transformation is done by Reynolds' Transport

Theorem (RTT). RTT states that the rate of changing in the amount of extensive property in the CM, is the rate of the property change within the CV in addition to the net flux of it trough the CV boundary due to fluid motion relative to the CV boundary" (Ferziger, 2002). Considering Ω as CV with Φ as porosity. ∂Ω is the surface of CV and n is the normal vector at any point of the surface. Based on mass conservation law on Ω, we will have:

(Rate of inflow - rate of outflow) + Source = accumulation

$$\int_{\partial\Omega} (-\rho u)\,.\,ndS + \int_{\Omega} \tilde{m}\, d\Omega = \int_{\Omega} \frac{\delta}{\delta t}(\phi\rho) d\Omega \tag{1}$$

Where in equation 1, ρ is fluid density (lb/ft^3), u is Darcy velocity (ft/day) and $\tilde{m}$ is source/sink term (unit mass/ (unit volume × unit time)).

By applying divergence theorem to the first term of LHS of equation 1 which is convective term, the surface integral will be transformed to the volume integral and differential coordinate-free form of the continuity equation will be obtained as follows, equation 2 (Bajaj, 2009).

$$-\nabla\,.\,(\rho u) + \tilde{m} = \frac{\delta}{\delta t}(\phi\rho) \tag{2}$$

2.2. Darcy Law

Modeling of filtration in porous media is performed with Darcy's law for low flow velocities (Aarnes, 2005). Flow apparent velocity, u, is related to gravity forces and pressure though the following formula:

$$u = -\frac{K}{\mu}(\nabla p + \rho g \nabla \mathrm{z}) \tag{3}$$

K is permeability, μ is viscosity, g is gravitational constant and z is the spatial coordinate.

There are some assumptions behind Darcy law:

1) Flow is laminar. For turbulent flow, which occurs at high velocities, the pressure gradient increases at a greater rate compared to the flow rate,
2) Flow is steady sate and the pressure of the reservoir does not change with time,
3) Isothermal condition: Temperature change results in a change in viscosities,
4) The fluid is incompressible: compressible fluids have a different pressure gradient as compared to incompressible fluids,
5) Formation is considered to be homogeneous with Incompressible rock (constant porosity) & Isotropic permeabilities.

2.3. Pressure equation

By neglecting the gravity forces (horizontal flow) in equation (3) and combining it with continuty equation, equation(2), pressure equation is obtained.

$$\nabla\,.\,(\rho\frac{K}{\mu}\nabla p) + \tilde{m} = \frac{\delta}{\delta t}(\phi\rho) \tag{4}$$

Considering incompressible fluid and rock, equation 4 will be as follows:

$$-\nabla\,.\left(\frac{K}{\mu}\nabla p\right) = \frac{\tilde{m}}{\rho} = \tilde{q} \tag{5}$$

$\tilde{q}$ is the volumetric source term (Bajaj, 2009).

2.4. Pressure Equation Discretization

In this part, we will present a cell-centered finite volume method named Two Point Flux Approximation (TPFA). This is one of the simplest discretization techniques to solve pressure equation. As it is understandable from the name of the method, it uses two points to approximate the flux. TPFA is exact for orthogonal or K-orthogonal

systems with anisotropic permeability. The neighboring cell averages pressures are used to estimate the flux through the interface between the adjoining cells.
Consider the integral form of the equation (5):

$$\int_{\Omega_i} (\tilde{q} - \nabla . u) dV = 0 \quad (6)$$

By applying divergence theorem, the volume integral will be transformed to surface integral and equation (4) is concluded.

$$\int_{\partial\Omega_i} (-\lambda \nabla p) . n dS = \int_{\Omega_i} \tilde{q} dV \quad (7)$$

λ is the ratio of permeability to viscosity named fluid mobility. $\partial\Omega_i$ is the total area of ith CV which is the summation of areas of the CVs' interfaces, $\partial\Omega_{ij}$ s.
f_{ij} is flux through the interface $\partial\Omega_{ij}$ (the common face between cell I and cell j) that should be estimated across the interface from a set of neighboring cell pressures.

$$\partial\Omega_{ij} = \Omega_i \cap \Omega_j \quad (8)$$

$$f_{ij} = -\int_{\partial\Omega_{ij}} (\lambda \nabla p . n) dS = T_{ij} (P_i - P_j) \quad (9)$$

T_{ij} is defined as the transmissibility of the surface $\partial\Omega_{ij}$. To get the total flux through the cell, the summation of the equation (9) should be calculated over all the interfaces of the CV. By replacing equation (9) in equation (7), the TPFA scheme for the pressure equation is obtained. j is the number of faces of each CV. The procedure is identical for any CV shape (structured or unstructured) and any dimension.

$$\sum_j T_{ij} (P_i - P_j) = \int_{\Omega_i} \tilde{q} dV = Q \qquad \forall \Omega_i \cap \Omega_j \quad (10)$$

2.5. Transmissibility Estimation

For our case which is a 3D discrete fracture model (DFM) and meshed by unstructured grids, we use the simplified DFM presented by Karimifard et al. (Karimi-Fard, 2003). The simplifications help to calculate transmissibility for fracture-fracture, matrix-fracture and matrix-matrix connections with TPFA technique. In the model, the place of the unknowns is at the barycenter of the grids. Our 3D simulator will be based on the connectivity lists. The connections between all the CVs should be specified.
For any CV, flow at the interface $\partial\Omega_{ij}$, is calculated as follows:

$$T_{ij} (P_i - P_j) = Q_{ij} \quad (11)$$

Where T_{ij} is the transmissibility at the interface $\partial\Omega_{ij}$ and P_i is the pressure at cell i. T_{ij} is defined as follows: (Karimi-Fard, 2003)

$$T_{ij} = \frac{\alpha_i \alpha_j}{\alpha_i + \alpha_j} \quad , \alpha_i = \frac{AK_i}{D_i} n_i . d_i \quad (12)$$

T_{ij} is calculated for all interfaces. α is evaluated for each grid cell. A is the area of the interface between the adjacent cells. n_i is unit normal to the interface inside CV_i. d_i is Unit vector along the line joining the center of triangle to the center of interface. D_i is the distance between the centroid of the interface and the centroid of the cell i. K_i is the permeability of CV_i.

3. Connectivity Lists

In our case, unstructured grids are Delaunay tetrahedral grids generated by TetGen for a 3D DFM.

Connectivity information between the unstructured elements in 2D/3D have to be found to solve the pressure equation for single phase flow. In a 3D Delaunay mesh, each tetrahedron is linked with 4 adjacent tetrahedrons and in a 2D Delaunay mesh, each triangle is linked with 3 adjacent triangles with equal flux through

the common interface. Note that fractures are represented by lines in 2D and planes in 3D. The following connectivity lists have to be created for 2D and 3D mesh.

Table 1: Necessary Connectivity Lists

Connectivity lists: 2D/3D
A set of neighboring CVs for each CV
A set of CVs sharing common edge (2D) / face (3D)
Boundary CVs
Boundary Edges/Faces
Fracture Edges/Faces
CVs connected to the fractures named fracture CVs
Fracture-Fracture Connectivity List
Boundary Fracture CVs

For a 2D DFM, mesh by Delaunay triangular grids, reference (Bajaj, 2009) is a suitable source to study. Investigating the connections for transmissibility approximation is mainly divided to 3 groups: Matrix-Matrix connection, Matrix-Fracture connection, Fracture-Fracture connections (Fracture Intersections). As mentioned before, to think about the connections, the simplified discrete fracture model proposed by Karimi-Fard (Karimi-Fard, 2003) is applied for our 3D DFM. Grid domain and computational domain are separated in their model. Nodes are 0D objects. Segments defined by 2 nodes are 1D objects. Convex polygons defined by segments are 2D objects. Convex polyhedral defined by convex polygons are 3D objects.

a) Matrix-Fracture Connection

In grid domain, the dimension of fracture is the dimension of matrix - 1. While in computational domain, the dimension of fracture is the same as dimension of matrix and we account for the thickness of the fracture (see Figure 1).

Volume correction is necessary for large DFMs. The equivalent pore volume from matrix CVs that connect with the fractures should be removed. The amount of pore volume removed depends on the number and size of fractures to which the matrix CV is connected. There is no need for volume correction for small DFN s.

In 3D computational domain, fracture grids are CVs including 5 faces (2 triangular and 3 rectangular faces). In fact, by adding fracture CVs and calculating related transmissibilities, flow is considered along and between the fractures.

Table 2: Matrix-Fracture Connection, Simplified Model

Connection	**Grid Domain**	**Computational Domain**
2D	Matrix: 2D, Fracture: 1D	Matrix: 2D, Fracture: 2D
3D	Matrix: 3D, Fracture: 2D	Matrix: 3D, Fracture: 3D

b) Matrix-Matrix Connection

For matrix-matrix connection, grid domain and computational domain are identical (see figure 2).

Table 3: Matrix-Matrix Connection, Simplified Model

Connection	**Grid Domain**	**Computational Domain**
2D	Matrix: 2D, Matrix: 2D	Matrix: 2D, Matrix: 2D
3D	Matrix: 3D, Matrix: 3D	Matrix: 3D, Matrix: 3D

c) Fracture Intersections/Star Delta Transformation

To handle the intersections, an intermediate control volume (figure 3a) is considered for numerical connections between two or more fractures. It helps to consider flow redirection and thickness variation. Because fracture intersection CVs (figure 3b) are small and cause instability in the calculations, they are eliminated by applying star delta transformation (Karimi-Fard, 2003).

To approximate the equivalent transmissibility for intersecting fracture network, a network resistor is considered (figure 4).

Based on equation 12, to estimate transmissibility between fracture CV_1 and intersecting CV_0 in figure 4, D_1 value is considerably larger than D_0 and the permeability of CV_0 is relatively the same as CV_1 permeability. Hence, the value of α_1 is negligible in comparison with α_0 (equation 13).

$$T_{10} = \frac{\alpha_1 \alpha_0}{\alpha_1 + \alpha_0}, \quad \alpha_1 \ll \alpha_0 \;\; then \;\; T_{10} \simeq \alpha_1 \tag{13}$$

Similarly, $T_{20} \simeq \alpha_2$ and $T_{30} \simeq \alpha_3$. Considering the mentioned assumptions, T_{12}, T_{13} and T_{23} values are approximated as follows (see figure 4):

$$T_{12} = \frac{T_{10} T_{20}}{T_{10} + T_{20} + T_{30}} \simeq \frac{\alpha_1 \alpha_0}{\alpha_1 + \alpha_2 + \alpha_3}$$

$$T_{23} \simeq \frac{\alpha_2 \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3} \tag{14}$$

$$T_{13} \simeq \frac{\alpha_1 \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3}$$

In fact, CV_0 has been removed by mentioned simplifications. This is named start delta transformation. By applying star delta transformation and making the equivalent resistor network for intersecting fracture *CVs*, CV_0 is eliminated and the equivalent transmissibility for connecting fractures is simply calculated based on equation 15. Star delta transformation, is exact for single phase flow and is applicable with good approximation for multi phase flow.

$$T_{ij} = \frac{\alpha_i \alpha_j}{\sum_{k=1}^{n} \alpha_k} \tag{15}$$

4. Solving Pressure Equation

4.1. Matrix Structure

Pressure equation, equation 10, is applied for all cells. This will result in a system of equations and a matrix of coefficient including transmissibility values. For structured grids, coefficient matrix is clearly symmetric but in the case of unstructured grids, symmetry is not conserved and matrix is not well defined (Bajaj, 2009).

The dimension of this matrix is N ×N where N is the sum of the number of tetrahedrons in an unstructured 3D mesh and number of fractures in the domain. It consists of matrix-matrix transmissibility, matrix-fracture transmissibility and fracture - fracture transmissibility matrix which all are assembled together in one matrix (see Figure 5).

4.2. BCGSTAB

To solve the resulted nonsymmetrical coefficient matrix in each time step, Biconjugate gradients stabilized method could be a suitable choice. It's an iterative method to solve nonsymmetric linear systems numerically. The

Biconjugate gradients stabilized algorithm is available in Matlab and applicable by the following command: X = bicgstab(A , b). It attempts to solve the system of linear equations A × X = b for X which is pressure values including matrix and fracture grids' pressures. The N by N coefficient matrix A is a matrix described in part 4.1. square, large and sparse. The column vector b has length N and is RHS values in the system of equations including source/sink and boundary condition terms. By calculating the pressure values in each time step, b is updated and new values for pressures are obtainable. Note that coefficient matrix doesn't change and remains the same. RHS values will be zero for fracture grids and as mentioned before based on the source/sink and boundary condition terms, are calculated for the matrix grids only.
For an isolated flow system, the following condition should be satisfied on the reservoir boundary: $u.n = 0$ (Aarnes J., 2005). Where u is Darcy velocity and n is normal vector of the boundary surface.

4.3. Convergence Analysis

Accuracy of the solution could be tested by a solution obtained from a very fine mesh named reference solution. Different degrees of grid refinement are to be considered in this stage. By increasing the mesh refinement, solution will converge to the reference solution. In this case, single phase flow, equations are solved to find the pressure solution only. For convergence analysis, mean pressure errors can be computed as (Bajaj, 2009):

$$E(p) = \frac{||P_{ref} - P||^2}{||P_{ref}||^2} \qquad (16)$$

P_{ref} and P are pressure values on reference and non-reference mesh respectively.

5. Some Preprocessing Calculations

The simulator is based on the data structure that consists of two parts, connectivity lists and data lists.

5.1. Connectivity lists

In 2D, finding the connectivity lists are much easier and less complicated. But in 3D, it needs lots of preprocessing efforts. It is really important to note that for tetrahedrons with common face on fracture planes, the connectivity list is completely changing because of adding fracture grids. Fracture matrix assembly is calculated based on the information of this part. Using TetGen output, which is a list of nodes, tetrahedrons and faces, some codes developed to extract necessary information about:

1) Matrix-matrix connectivity list

Two adjacent tetrahedrons, has a common triangular face. The size of tetrahedron-tetrahedron connectivity list matrix is n × 4 (n is number of tetrahedrons). Each row shows the tetrahedron number, and each column in a row presents the connecting tetrahedron number. In each row, the presence of negative number means that the tetrahedron has a boundary face, see Figure 6.

2) Tetrahedron to face connectivity list

Common faces between each two tetrahedrons, boundary faces and boundary tetrahedrons are extracted in this part. Table (4) presents tetrahedron to face connections. Matrix size is N_{face} × 5 where N_{face} is the total number of tetrahedron faces in the domain. Face number is the same as row number. The first three columns are node numbers creating a face in the domain and the two columns are the tetrahedron numbers sharing that face. Negative number in each row presents a particular boundary face.

Table 4

node 1	node 2	node 3	Tetrahedron number 1	Tetrahedron number 2
s	w	r	o	p
f	g	h	-1	u

From table (4), lots of useful information can be extracted by codes including boundary faces and boundary tetrahedrons. Boundary faces are the faces which are not common between two tetrahedrons and just belong to one tetrahedron which that tetrahedron will be a boundary tetrahedron and the face will be a boundary face. For example, the second face in table 4 is a boundary face which only belongs to "u" th tetrahedron and therefore u is a boundary tetrahedron.

3) Fracture-tetrahedron connectivity list

In this section, fracture faces (the faces which form the fractures), fracture tetrahedrons (the tetrahedrons connected to the fracture planes), boundary fracture tetrahedrons and adjacent fracture faces are distinguishable. The faces on each fracture plane are recognizable by double testing:

a) Relation between the fracture plane normal vector (np) and a face normal vector (nf) on that plane should be: $| np / nf | = r$ where "r" is a natural number and,
b) One point of the face should satisfy the fracture plane equation as well.

By finding the fracture faces, fracture tetrahedrons including fracture faces and boundary fracture tetrahedrons are extracted by codes. Boundary fracture tetrahedrons are fracture tetrahedrons that have boundary faces based on table 4.

A connectivity list of adjacent fracture faces with common edge should be extracted as well. It will be an array with size: $(3 \times 2) \times N_{fface}$. N_{fface} is the number of fracture faces. This connectivity list is useful to extract information about fracture intersections.

4) Fracture-fracture intersection

The investigation is limited to fracture planes and the fracture faces. We need to have a list of common edges on two or more fracture planes. When an edge is common between two or more fracture faces and fracture faces are belonging to different fracture planes, then the fracture faces should be marked as they will make the fracture-fracture intersecting grid cells.

5.2. Data lists

1) Normal vectors

a) To calculate Matrix-Matrix assembly (MMA)

Array Size: $(4 \times 5) \times n_{mgrids}$ where n_{mgrids} is the number of tetrahedral cells.

In table 5, second column is tetrahedrons' number adjacent to the tetrahedron in the first column.

In each row, the normal vector of the common face between tetrahedrons in the first and 2nd columns is shown in the last 3 columns. "m" is changing from 1 to n_{mgrids} for matrix cells and from 1 to n_{fgrids} for fracture cells. a, b, c and d are tetrahedrons adjacent to m^{th} tetrahedron for a matrix cell and for a fracture cell, adjacent cells will include two matrix cells and three fracture grid cells.

b) To calculate Matrix-Fracture assembly (MFA)

Array Size: $(5 \times 5) \times n_{fgrids}$ n_{fgrids}: number of fracture cells

Second column is grids' number adjacent to fracture cell in the first column. It is including two tetrahedrons and 3 fracture grids with three faces (Figure: should be added). Fracture grids have 5 faces including 2 triangular and three rectangular faces.

Table 5: Sample data structure for normal vector

a) MMA						b) MFA				
m	a	N1-x	N1-y	N1-z		m	a	N1-x	N1-y	N1-z
m	b	N2-x	N2-y	N2-z		m	b	N2-x	N2-y	N2-z
m	c	N3-x	N3-y	N3-z		m	c	N3-x	N3-y	N3-z
m	d	N4-x	N4-y	N4-z		m	d	N4-x	N4-y	N4-z
						m	e	N5-x	N5-y	N5-z

2) Permeability

a) MMA

Matrix Permeability

Matrix size: $n_{mgrids} \times 4$, n_{mgrids}: number of tetrahedral cells

Three first columns are tetrahedrons' barycenter's coordinates and 4th column is including K values simulated by geostatistical modeling using "psgsim" (Manchuk, 2010),

b) MFA

Fracture Permeability

Matrix size: $n_{fgrids} \times 4$, n_{fgrids}: number of fracture grid cells

Three first columns are fracture grid barycenters and 4th column is K values.

Fracture permeability is calculated based on the following formula: $e^2/12$ where "e" is fracture thickness. The key point is that the barycenter of the fracture grids are considered the same as the barycenter of the fracture face and to be exact, by adding the fracture grids and considering the thickness for fracture planes, the barycenter of the tetrahedral matrix grids will be changed but the changes are negligible as the fractures' thickness is a small number and if we are facing with a large DFM, volume correction should be performed which will result in changing the barcycenters of matrix grids.

3) Area of the CVs' faces

a) MMA : Array size: $(4 \times 3) \times n_{mgrids}$

The area of the common face between the tetrahedrons in first and second columns is presented in the third column. The second column is replaced based on the calculated connectivity lists in the previous section.

b) MFA: Array size: $(5 \times 3) \times n_{fgrids}$

4) Centroid (barycenters) of the faces of all tetrahedrons

a) MMA: array size $(4 \times 5) \times n_{mgrids}$

In each row, the barycenter of the common face between tetrahedrons in the first and 2nd columns is shown in the last 3 columns.

b) MFA: array size $(5 \times 5) \times n_{fgrids}$

To find the barycenter of the two triangular faces of a fracture cell, the barycenter of the fracture grid should be projected along the direction of normal vector with the length equal to the half of thickness.

5) Barycenters of all the CVs

a) MMA: terahedrons' barycenters are saved in a $(n_{mgrid} \times 3)$ matrix.

b) MFA: Barycenters of the fracture grids which is a $(n_{fgrids} \times 3)$ matrix.

Each row shows the centroid of one tetrahedron with number the same as row's number.

6) a) MMA: In each tetrahedron, the distance between the centroid of each face and the tetrahedron's barycenter is calculated. Information is saved in a $(4 \times 3) \times n_{mgrids}$ array. m: 1, …, n_{mgrids}

c) MFA: In each fracture grid cell, the distance between the centroid of each face and fracture grid's barycenter is calculated and saved in a $(5 \times 3) \times n_{fgrids}$ array. m: 1, …, n_{fgrids}

7) For each CV, calculating the unit vectors joining the CV barycenter to the centroid of the CV's faces is done.

a) MMA: For all tetrahedrons, the information of four unit vectors joining the tetrahedron barycenter to the centroid of tetrahedron's faces is saved in a $(4 \times 5) \times n_{mgrids}$ array. m: 1, …, n_{mgrids}

b) MFA:

In each fracture grid cell, the information of five unit vectors joining the fracture grid barycenter to the centroid of grid's faces is saved in a $(5 \times 5) \times n_{fgrids}$ array. m: 1, …, n_{fgrids}

By applying the information presented in data lists and connectivity lists, the assembled transmissibility matrix including matrix-matrix assembly, fracture-matrix assembly and fracture-fracture assembly is calculated.

6. Conclusions

The paper mainly describes how finite volume form of mass conservation law is applied to model single phase flow in a 3D DFM which is gridded by unstructured tetrahedral mesh using TetGen software. Unstructured grids help us to handle complex fracture media more precisely in comparison with structured grids.

Applying the simplified model presented by Karimi-Fard (Karimi-Fard, 2003), transmissibility is calculated for matrix-matrix, fracture-fracture and matrix-fracture connections. Hence, flow is considered along and between fractures as well. To handle flow at the fracture intersections, control volumes at the fracture intersections are eliminated using star-delta transformation rule which will result in computational efficiency and numerical stability (Karimi-Fard, 2003).

The finite volume simulator is mainly based on connectivity lists. Carefully creating the connectivity lists is the main concern which needs lots of preprocessing efforts especially on 3D unstructured grids. By listing the connections, transmissibility is calculated in three separate parts, for matrix-matrix, matrix-fracture and fracture-fracture connections and flow simulation follows the transmissibility calculation. The success of applying finite volume on a sample 2D DFM has been demonstrated by Bajaj (Bajaj, 2009). Their research shows that finite volume can provide precise solution for flow problem in a 2D DFM. This could be investigated on our 3D model as well. Some extensions will be done on the current research including considering multiphase flow and use Multi Point Flux Approximation (MPFA) technique instead of TPFA. Matlab Reservoir Simulation Toolbox (MRST) could be a good reference to develop codes for future endeavors.

References

Aarnes J., 2005, "*An Introduction to the Numerics of Flow in Porous Media Using Matlab*", SINTEF ICT, Dept. of Applied Mathematics, Oslo,

Bajaj, R., 2009, "*An Unstructured Finite Volume Simulator for Multiphase Flow Through Fractured-Porous Media*", MSc Thesis, MIT University,

Ferziger, J, Peric, M., 2002, "*Computational Methods for Fluid Dynamics*", Book, Springer,

Karimi-Fard, M., Durlofsky, L., Aziz, K., 2003, "*An Efficient Discrete Fracture Model Applicable for General Purpose Reservoir Simulator*", SPE 79699,

Manchuk, J., 2010, "*Geostatistical Modeling of Unstructured Grids for Flow Simulation*", PhD thesis, University of Alberta,

Razavi, S.F., Boisvert, J., Leung, J., 2011, "*Unstructured Grid Generation in 2-D and 3-D*", Paper 213, CCG Annual Report 13, University of Alberta.

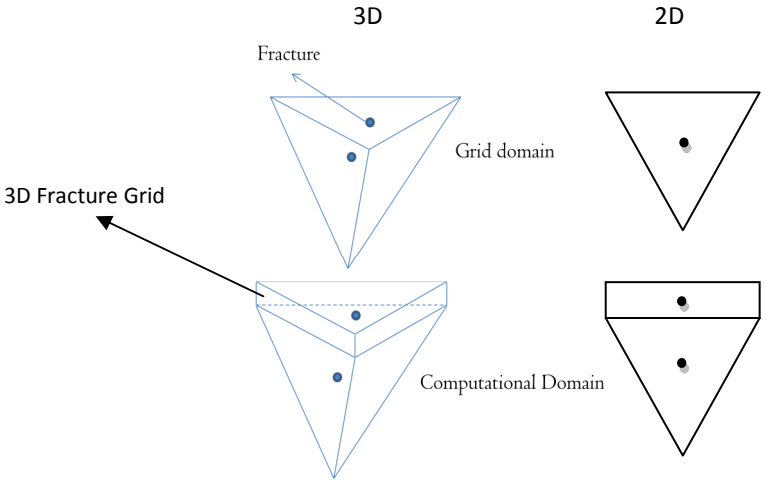

Figure 1: Matrix-Fracture Connection

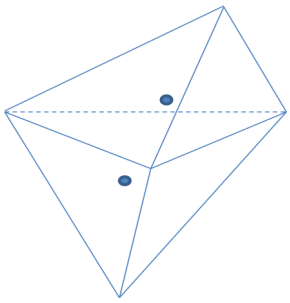

Figure 2: Matrix-Matrix Connection

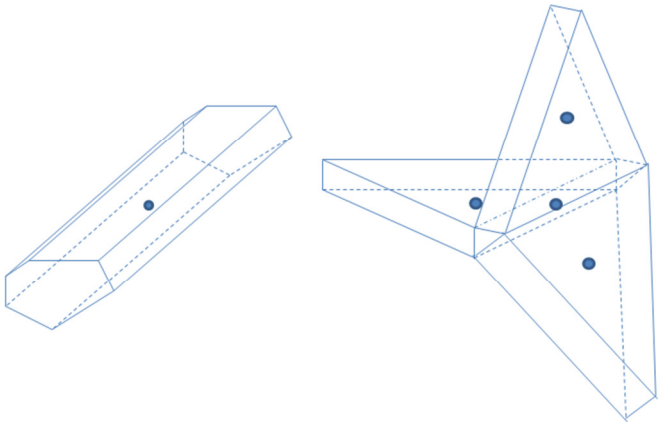

Figure 3: a) Intermediate CV which is the intersection of 6 fractures with different thicknesses b) Three intersecting Fracture CVs

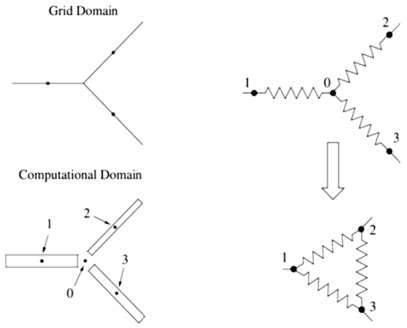

Figure 4: Star Delta Transformation (Karimi-Fard, 2003)

Matrix - Matrix Assembly	Matrix -Fracture Assembly
Transient of Matrix - Fracture Assembly	Fracture - Fracture Assembly

Figure 5: Transmissibility Matrix for a DFM Meshed by Unstructured Grids

0	15783	814	10422	9240
1	3731	8177	12847	1004
2	5744	4753	7069	-1
3	6275	28949	20470	6372
4	21113	8235	14111	17330
5	16893	7497	16937	11503
6	29892	18455	27776	23335
7	6057	5848	3436	-1
8	4314	2010	4307	974

Figure 6: Matrix-Matrix Connectivity List