Appendix A of the PhD thesis entitled

THERMO-HYDRO-MECHANICAL (THM) COUPLING IN FRACTURED/POROUS GEOMATERIALS

by

CUI Xin

at

The University of Hong Kong

User manual of DDFS^{3D}

A.1 Overview

DDFS^{3D} is a set of open-source codes which utilizes DDM, FSM and the hybrid of DD-FS to efficiently and accurately simulate fractures embedded in large 3D domains. DDFS^{3D} is developed in Fortran language with a number of modules to hold subroutines and variables. CPU acceleration with OpenMP library is in place. Boundary conditions, covering a range of stress, displacement and the hybrid of the two, are supported by DDFS^{3D}. Far-field (*in-situ*) stress can also be applied. The joint element is available to account for the elastic behavior of fracture gouge/asperity in normal and tangential directions. Note that the current version of DDFS^{3D} does not provide any pre- or post-processing functions. Users have to prepare the mesh beforehand and rely on other programs/codes to visualize the results. In addition, the OpenMP library has to be compiled together with the source codes of DDFS^{3D} to enable CPU acceleration. Otherwise, DDFS^{3D} will be run in a serial manner.

A.2 Components of DDFS3D

Figure A.1 shows the components of DDFS^{3D}. Two commonly used element types, namely constant triangular element and constant quadrilateral element, are provided in DDFS^{3D}. For each element type, three branches including DDM, FSM and the hybrid of DD-FS are developed independently. In principle, both DDM and FSM simulations can be carried out by the DD-FS program, but the codes of DD-FS are a bit more complicated than the other two. The main purpose

of publishing DDFS^{3D} is to encourage secondary development based on DDFS^{3D}, which can benefit the DD-FS research community and enhance the application of DD-FS. It is unnecessary for DDM and FSM to always go hand in hand, thus we here provide three independent branches to give users sufficient discretion. Three companion programs (Parts 1, 2 & 3), which have different purposes, are incorporated under each branch. Part 1 is used to solve for the primary unknowns. Part 2 evaluates the stresses and displacements at the center of gravity of each element in its local coordinate system. Part 3 computes the stresses and displacements at arbitrary points other than those in Part 2 in the predefined coordinate system(s). The source codes and sample input and output files (numerical examples in section 4.3 of the companion PhD thesis) are included in the respective folders of each program. Users can compile the source code files and copy the input file(s) to the same folder to run the program.

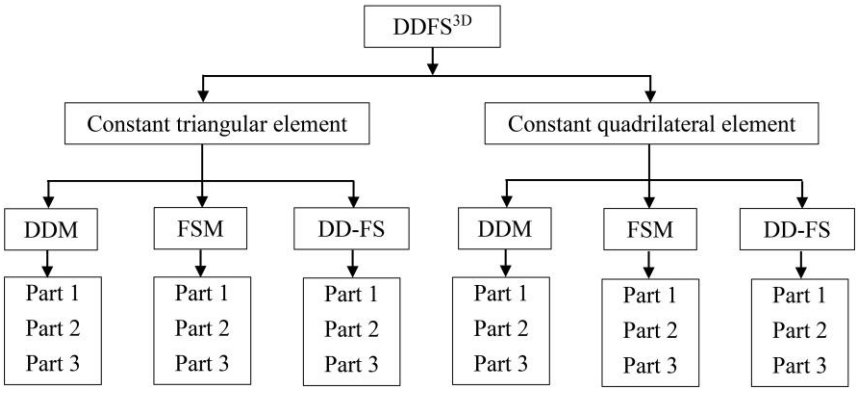

Figure A.1 Component tree of DDFS^{3D}.

A.3 Input files

A.3.1 Part 1

Part 1 is used to solve for the primary unknowns, namely displacement discontinuities (DD) and/or fictitious stresses (FS). Its input file is structured as follows (Figure A.2):

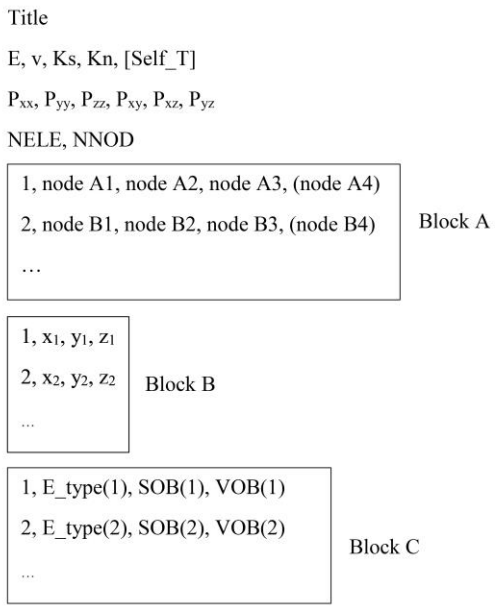

Figure A.2 Structure of input of Part 1 (file name: input_part1.txt).

Symbols

Title: title of the project

E: elastic modulus

v: Poisson's ratio

K_s: tangential stiffness coefficient for joint element

K_n: normal stiffness coefficient for joint element

Self_T: type of subroutine to evaluate self-influence coefficients, applicable to triangular elements only.

0: analytical integration method

1: numerical integration method

P_{xx}-P_{yz}: components of far-field stress

NELE: number of elements

NNOD: number of nodes

Block A: nodes of each element arranged in a counterclockwise sequence. Three nodes for one triangular element and four nodes for one quadrilateral element.

Block B: coordinates of nodes in the global coordinate system

Block C: element type and boundary conditions

E_type: element type

0: FSM element

1: DDM element

2: joint element

SOB: sort of boundary condition. Three entries are required including one in the normal direction and two in the tangential directions.

0: stress boundary condition

1: displacement boundary condition

For example, 1, 1, 1 indicates displacement boundary condition in all normal and tangential directions; 0, 1, 1 indicates stress boundary condition in the normal direction, while displacement boundary condition in the tangential directions; 0, 0, 0 indicates stress boundary condition in all normal and tangential directions.

VOB: value of boundary condition, which accompanies SOB. Three entries are required including one in the normal direction and two in the tangential directions.

As a companion of SOB explanation above: 0.0, 0.0, 0.0 indicates zero displacements (fixed) in all normal and tangential directions; -1.e6, -0.05, -0.05 indicates 1.e6 compressive stress in the normal direction while -0.05 displacements in both tangential directions; 0.0, 0.0, 0.0 indicates all normal and tangential directions are free of stress. DDFS^{3D} does not have any built-in units and users have to determine a system of units a priori.

Here we explain a bit more about how to arrange the nodes of each element and how DDFS^{3D} sets up the local coordinate system for a candidate element. We take the quadrilateral element as an example. The four nodes of each quadrilateral element are arranged in a counterclockwise manner ($A \to B \to C \to D$ shown in Figure A.3). The vector $\mathbf{AB}$ is selected as the ξ axis in the local coordinate system, and the cross product $\mathbf{AB} \times \mathbf{AC}$ is defined as the γ axis; η axis is obtained by $\gamma \times \xi$. Note that γ is in line with the outward normal vector $\boldsymbol{n}$ to the small

patch of the boundary. For a triangular element, the nodes and local coordinate system are organized in the same way.

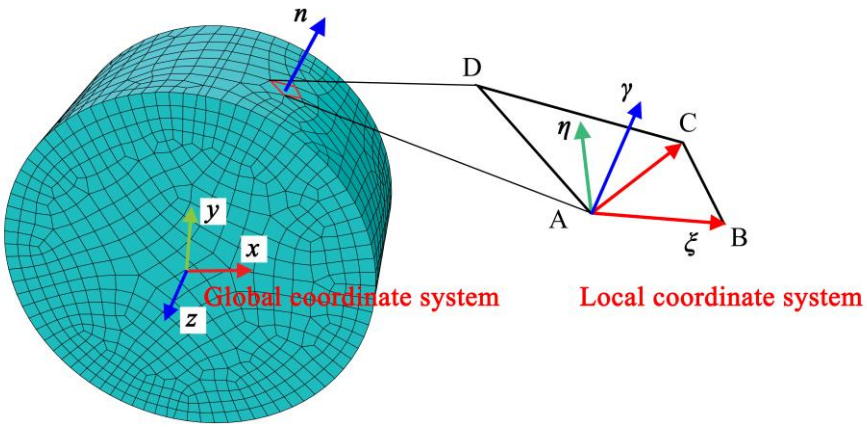

Figure A.3 Global and local coordinate systems defined in DDFS[3D].

A.3.2 Part 2

In addition to the input of Part 1, Part 2 also requires the DD/FS solved by Part 1 to evaluate the stresses and displacements at the center of gravity of each element. The data structure of DD and/or FS is shown in Figure A.4.

```
1, DD(1)/FS(1)
2, DD(2)/FS(2)
…
```

Figure A.4 The DD/FS solved by Part 1 are utilized as the input for Part 2 to evaluate the stresses and displacements at the center of gravity of each element in the local coordinate system (file name: D.txt for DDM, P.txt for FSM and DP.txt for DD-FS).

Symbols

DD: displacement discontinuity. Three components for each element in the local normal and tangential directions.

FS: fictitious stress. Three components for each element in the local normal and tangential directions.

A.3.3 Part 3

In addition to the input of Part 1 and the DD/FS file mentioned in section A.3.2, Part 3 also requires another input file indicating the coordinates of arbitrary field points (other than those in Part 2) and the associated coordinate system where stresses and displacements are to be evaluated.

```
NOP
1, x1, y1, z1, LCS(1)
2, x2, y2, z2, LCS(2)
…
```

Figure A.5 Provide coordinates of field points and the associated coordinate system(s) for Part 3 to evaluate the stresses and displacements therein (file name: input_part3.txt).

Symbols

NOP: number of points.

x_i, y_i, z_i: coordinates of *i*th element in the global coordinate system.

LCS(i): matrix to define the local coordinate system ($\xi\eta\gamma$) in which the stresses and displacements are to be evaluated at the candidate point. LCS(i) has nine components, namely cos<ξ, x>, cos< ξ, y>, cos<ξ, z>, cos< η, x>, cos< η, y>, cos< η, z>, cos< γ, x>, cos< γ, y>, cos< γ, z>, which are the cosines to perform coordinate transformation following Eq. (A.1).

$$\begin{pmatrix} \xi \\ \eta \\ \gamma \end{pmatrix} = \begin{pmatrix} cos<\xi, x> & cos<\xi, y> & cos<\xi, z> \\ cos<\eta, x> & cos<\eta, y> & cos<\eta, z> \\ cos<\gamma, x> & cos<\gamma, y> & cos<\gamma, z> \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \text{(A.1)}$$

A.4 Solver for asymmetric linear system equations

The coefficient matrix of the boundary element method is full and asymmetric. Considering the large scale of 3D simulation, the direct solver (Gauss elimination) is not efficient as compared with iteration solvers. To select an appropriate iteration method, the property of the coefficient matrix should be well understood. The diagonal entries of the coefficient matrix of DDM and/or FSM are significantly larger than those off-diagonal ones, which leads to decent convergency of

Gauss-Seidel (GS) iteration. Therefore, GS iteration is selected as the solver for DDFS^{3D}. The GS solver shows excellent robustness according to the author's experience.