Министерство образования и науки Республики Казахстан

Карагандинский университет имени академика Е.А. Букетова

Факультет математики и информационных технологий

Кафедра алгебры, мат. логики и геометрии им. профессора Т.Г. Мустафина

**Есенбаева Гульсим Ахмадиевна**

**Курс лекций**

**по дисциплине «Компьютерная механика (на английском)»**

Образовательная программа: «7М05402 Механика»

Караганда 2022

The Ministry of Education and Science of the Republic of Kazakhstan

Buketov Karaganda University

The Faculty of Mathematics and Information Technologies

The Chair of Algebra, Mathematical Logic and Geometry named after Professor Mustafin T.G.

**Yessenbayeva Gulsim Akhmadievna**

**The course of lectures**

**by the discipline «Computer mechanics (in English)»**

Educational program: «7M05402 Mechanics»

Karaganda 2022

**Lecture 1**

**Lecture topic: Code development.** **Running HFRANC2D for Hydraulic Fracture Simulations**

**The plan**

**1. Code development**

1.1. FRANC2D Hierarchy

1.2. Fluid Flow Implementation and Embedding with FRANC2D

**2. User’s Guide**

2.1. Introduction

2.2. Running HFRANC2D for Hydraulic Fracture Simulations

2.2.1. Preprocess of Simulation

2.2.2. Mechanical Analysis

2.2.3. Propagating the fracture

2.2.4. Results and Analysis

**1. Code development**

Computer Methods in Applied Mechanics and Engineering was founded over three decades ago, providing a platform for the solution in this important field of science and engineering. The range of appropriate contributions is very wide. It covers any type of computational method for the simulation of complex physical problems leading to the analysis and design of engineering products and systems. This includes theoretical development and rational applications of mathematical models, variational formulations, and numerical algorithms related to finite element, boundary element, finite difference, finite volume, and meshless discretization methods in the following fields of computational science and engineering:

• solid and structural mechanics

• fluid mechanics

• mechanics of materials

• heat transfer

• dynamics

• geomechanics

• acoustics

• biomechanics

• nanomechanics

• molecular dynamics

• quantum mechanics

• electromagnetics

and also includes virtual design, multiscale phenomena, from nanoscale to macroscale, multiphysics problems, parallel computing, optimization, probabilistic and stochastic approaches.

CMAME publishes original papers at the forefront of modern research describing significant developments of computational methods in solving problems of applied mechanics and engineering

FRANC2D is an interactive, platform-unrelated program that was designed to perform discrete crack modeling in two dimensions. Stresses and deformations are determined by means of the finite element method. Integral remeshing is performed to allow the finite element mesh to be modified automatically or semi-automatically to represent new crack configurations [Wawrzynek, P. and Ingraffea, 1996]. FRANC2D also provides graphical and tabular post-processing functions that allow the user to examine the results.

Subroutines were developed to determine the fluid pressure that should be applied to the walls of the fracture to satisfy fluid continuity and stress equilibrium. This was accomplished by initially writing code to analyze the inflation and propagation of a fracture using analytical expressions for displacement and stress intensity. The analytical expressions were used during initial development because they execute quickly. The code was combined with FRAC2D by substituting the analytical expressions for displacements and stress intensity for expressions calculated numerically by FRANC2D.

Additional development was required because the finite element grid used by FRANC2D differed from the finite difference grid used by the flow calculations. The finite element grid is revised by FRANC2D after every step of propagation, indeed, this remeshing feature is perhaps the most important attribute of FRANC2D. However, the constantly changing finite element grid requires the code to translate between the finitedifference and finite element grids. This is accomplished by linearly interpolating valuesat two points on one mesh to determine the value on the other mesh.

EXC\_AUTO\_DRIVER is the subroutine that FRANC2D uses for automatic propagation. This subroutine was modified to accommodate the changes required to analyze hydraulic fractures, and the modified subroutine is called EXC\_AUTO\_DRIVER\_FLOW. The modifications include calls to subroutines to initialize variables used to analyze fluid flow, as well as a call to the subroutine, fluidflowcalc, that calculates the fluid pressure either during inflation or propagation.

**1.1. *FRANC2D Hierarchy***

The FRANC2D program was initially developed in the VAX/VMS environment and implemented with VAX-11 FORTRAN language [Srinivasan, 1988]. It has been developed to a multi-platform, FORTRAN 77 standard program, which can be compiled and installed, then used in various computer systems including: DEC Alpha Digital Unix, GNU/Linux, HP UNIX, IBM AIX, SGI IRIX, SUN Solaris, ULTRIX Unix, and Microsoft Windows XP/2000/NT[Wawrzynek and Ingraffea, 1996].

FRANC2D is arranged in a subdirectory system. Most of the subdirectories contain several logically related subroutines. Each subdirectory is given a unique name, most of which are three-letter codes derived from abbreviations of the contents of the subdirectory[Srinivasan, 1988]. All the subroutine names start with this three-letter code except for some of the fluid flow subroutines and analysis subroutines. The subdirectory named “inc” includes all the COMMON block files that contain global variables used by all the FORTRAN source files. Subdirectory “Main” has the main program “franc.ff” (franc.f for Microsoft Windows system) in it. In SUN Solaris system, the final executable file: Franc, will be created in subdirectory: Main[Wawrzynek and Ingraffea, 1996]. In Microsoft Windows system, the final executable file will be named: f2d, and will be created in subdirectory: f2d. Subdirectory “examples” contains some example model that new user ca start with. Subdirectory “tools” contains some programs working as pre-processors to convert “.ff” source files to “.f” files, write “makefile” for each subdirectory under SUN Solaris system [personal communications with Carter]. Microsoft Windows system will not use this subdirectory.

Most of the changes made to couple fluid flow with mechanical analysis are in the EXC subdirectory. The FORTRAN source code file extension is .ff for SUN Solaris system, and .f for Microsoft Windows system. The following lists new FORTRAN files that include one or more subroutines each have been added into EXC subdirectory and a short explanation.

excaperflow.ff– fracture aperture calculation

exclengthflow.ff – fracture length calculation

excvoluflow.ff– fracture volume calculation

fluidflowini.ff – initialization of fluid flow calculation

fluidflowcalc.ff – fluid flow calculation

The source codes are in a CD-ROM attached in this thesis

**1.2. *Fluid Flow Implementation and Embedding with FRANC2D***

The automatic propagation subroutine (EXC\_AUTO\_DRIVER\_FLOW) is rewritten based on the propagation criterion: strength intensity factor equals fracture toughness will be kept when propagation occurs due to the fluid flow and other possible loadings. There are two subroutines called by EXC\_AUTO\_DRIVER\_FLOW: FLUID\_FLOW\_INIT and FLUID\_FLOW\_CALC, which will implement fluid flow initialization and calculation, respectively. Both of them contain several subroutines in their file body. They will be described in “Code Structure” part later in this thesis.

**2. User’s Guide**

This part of the work is to help those who want to use HFRANC2D as an analysis tool, but have never used FRANC2 before. Through reading this part, together with a little practice, the user can focus on the simulation problem and analysis, instead of the software itself.

*Some words about computational mechanics and the MDSolids.* There are two primary ways in which the computer can be of valuable assistance to you as you study the topics in this Mechanics of Materials text. First, you can use math application software or a spreadsheet program or a programming language of your own choice to write a computer program to solve a mechanics of materials problem. By writing your own computer programs for several of these exercises, you will not only gain experience in programming, but you should also gain valuable experience in organizing efﬁcient, systematic solutions of mechanics of materials problems. Use of one of the two computer programs is another way in which the computer can be of valuable assistance to you.

The more powerful of these two computer programs is the award-winning MDSolids educational software. Since MDSolids is written in Visual Basic, it is only available for use on computers running a Windows operating system (95, 98, NT). For Macintosh users, the MechSOLID software.

The MDSolids software package and ninety special MDSolids-based example problems that are provided with Mechanics of Materials will enable you to use the computer for solving problems in axial deformation, torsion, bending, combined-loading, and buckling. MDSolids provides both systematic problem-solving procedures and a user-friendly graphical interface, and through its use you can gain valuable insight .

1. The MDSolids educational software package is copyrighted by its author, Dr. Timothy A. Philpot. It was a winner of the 1998 Premier Award for Excellence in Engineering Education Courseware. The MDSolids website is www.mdsolids.com .

2 Windows is a registered trademark of the Microsoft Corporation.

3 Macintosh is a registered trademark of Apple Computer, Inc. A-43 A-44 Appendix G into the behavior of structural members and systems under various loading conditions.

***MDSolids Modules.*** MDSolids consists of modules, which are similar to book chapters in that each module focuses on speciﬁc mechanics of materials concepts and problem-solving methods. There are currently twelve MDSolids modules: Basic Stress and Strain Problems Beam and Strut Axial Structures Truss, Analysis and Stresses, Statically Indeterminate Axial Structures, Torsion Members. Determination Beams Flexure Section Properties, Column Buckling Mohr’s Circle Analysis, General Analysis of Axial, Torsion, and Beam Structures Pressure Vessels The modules can be accessed in any sequence. MDSolids is powerful enough so that many different structural conﬁgurations and loadings can be analyzed with each separate module, but the modules are also coordinated so that results from one module are available for use in related modules.

***MDSolids Features.*** MDSolids is an educational software package developed speciﬁcally for the introductory mechanics of materials course. The version of MDSolids consists of two closely integrated parts: (1) the basic MDSolids educational software, and (2) a special supplement of ninety example problems.

Features of the MDSolids Software: Some of the key features of the basic MDSolids software are:

• Versatility: As indicated above, MDSolids has computational modules pertaining to all of the topics taught in a typical mechanics of materials course.

The scope of MDSolids offers routines to help students at all levels of understanding.

• Ease of Input: Graphic cues are provided to guide users in entering data, so that the student is able to deﬁne a problem intuitively and directly without the need for a user’s manual.

• Visual Communication: Each MDSolids routine features a picture, sketch, or plot that graphically depicts important aspects of the problem. For a number of topics, including stresses in beams, deﬂection of beams, Mohr’s Circle for stress and strain, and others, plots that show the results are generated.

• Correct Solution and Intermediate Results: MDSolids is an ‘‘electronic solutions manual,’’ giving not only the correct solution for each problem but also providing intermediate steps that can be used to conﬁrm the problemsolving approach.

• Text-based Explanations: Many of the MDSolids modules provide extra explanations to describe in words how the calculations are performed.

• Help Files: The MDSolids Help Files contain instructions for using the software, but, more importantly, they contain theoretical background and practical suggestions for solving various types of problems.

**2.1. *Introduction***

HFRANC2D is a two-dimensional, menu-driven interactive simulation code. Users will use a mouse and a keyboard to control the behavior of fracture simulation.

HFRANC2D has two basic ways to simulate propagation of fracture, automatic propagation and manually step-by-step propagation, as described in the user manual [Wawrzynek and Ingraffea, 1996].

These procedures will still need to be followed when doing hydraulic fracturing simulation. However, parameters of fluid flow will also need to be input and adjusted.

The general way to simulate hydraulic fractures by HFRANC2D, is consisted in the following steps (figure 3.2-1):

1. Construct initial mesh Build model geometry, create finite element grids using CASCA, which is a separate mesh generator developed by Cornell Fracture Group [Wawrzynek and Ingraffea, 1996].
2. Prepare simulation Open the model file in HFRANC2D, set up boundary conditions (BCs), material properties, loading conditions, initiate fracture.
3. Analysis Perform numerical simulation by propagating the fracture.
4. Evaluate Review graphical display and data showing simulation results.

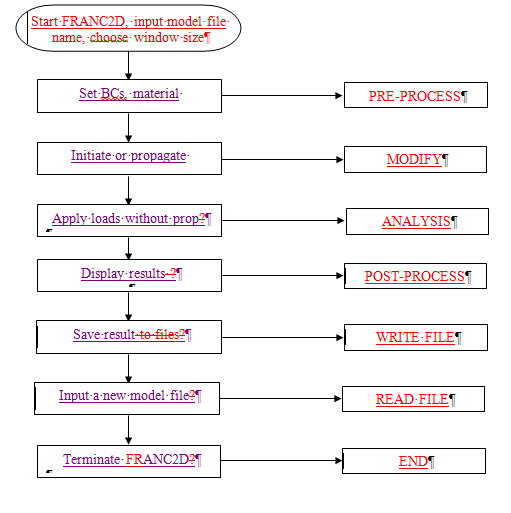


Figure 3.2-1. Flow chart of performing simulation in FRANC2D

**2.2. *Running HFRANC2D for Hydraulic Fracture Simulations***

Running a simulation of hydraulic fracturing using HFRANC2D is similar to other fracture simulators. Nevertheless, some specific settings need to be done for fluid flow analysis before running the hydraulic fracturing simulation procedure. The HFRANC2D menu system has been altered for fluid flow, although most of them are still the same as described in the manual [Wawrzynek and Ingraffea, 1996]. Here I assume that a new model with file extension name “inp” has been established by the mesh generator: CASCA.

**2.2.1. *Preprocess of Simulation***

In a console window (on a PC, this means a DOS Command window), change directory to the working directory where the HFRANC2D executable file exists. This file is named “f2d.exe” on Microsoft Windows system, where it is named “Franc” in SUN Solaris workstations. Run it by typing the executable file name “f2d” or “Franc” after command prompt, then hit “Enter”. It will ask for a model file name. Input a model file name without extension, then hit “Enter”. Then type “1” if this is first time to perform simulation with FRACN2D or “0” if HFRANC2D has already simulated with this model and saved it. After that choose HFRANC2D main window size. Use small size by type “s” and hit “Enter”, This can prevent some button being displaced out side the screen. In the HFRANC2D main window, select “PRE-PROCESS” button, this will bring up the lower level pre-process menu, in which the user may set up material mechanical properties, fluid flow parameters, boundary conditions, and loads.

To set up the mechanical properties, such as Young’s Modulus, Poisson’s ratio, etc, select “MATERIAL” button, which is number 4 from the top of the menu. Then the material setup menu will show up. Up to 6 different kinds of material can be set in one model.

To set up the fluid flow conditions, such as volumetric fluid flow rate into the fracture, click the “FLUID FLOW” button in the preprocess menu, which is at the bottom of the pre-process menu. This is a newly added feature of HFRANC2D in order to include controls to the fluid flow being used.

To set up the boundary conditions, select the “FIXITY” button, which is at the top of the pre-process menu.In this menu, the user can choose to fix the boundary of the model in one direction (x or y), or both directions. Normally, to simulate hydraulic fracturing in soil, at least one side should be fixed, and at least one fixity of x direction and one fixity of y direction should be chosen, so that after the model have been applied loads, the model will not have rigid body movement, which will cause the simulation to fail.

To setup loads of the model, select the “LOADS” button, which is at the top of the pre-process menu. There are various loading types can be chosen. However, for one model, only 6 different loading cases can be chosen. Uniform crack pressure may be set before the crack has been initiated. A file format for residual load is listed in Appendix B.

*Important:*

After the pre-process has been performed, one should return to the main menu and choose “WRITE FILE” button the save the model to a new file by giving it a different name. Material properties can only be changed before body load is selected, or the body load will concentrate at the part of the model. Even remove the body load before change the material properties, this misinterpreting still exists. The reason is unclear. It would be better to start a new model for different material properties.

The necessary steps needed in “PRE-PROCESS” are “FIXITY”, “LOADS”, “MATERIALS”, and “PROBLEM TYPE”. After all the settings, save the model with “WRITE FILE”.

**2.2.2. *Mechanical Analysis***

After pre-process is finished, a mechanical analysis should be performed to ensure the boundary conditions, loads, and other conditions are applied correctly. In the Main HFRANC2D menu, select “ANALYSIS”, the analysis menu will show up. This menu offers selections for linear and non-linear analyses. I used “Linear analysis” from this menu for all of the development. Two selections will appear under linear analysis: “DIRECT STIFF” and “DYN RELAX”. Choose “DIRECT STIFF”. Linear elastic fracture mechanics and direct stiffness method are assumed to be consistent with simulation of hydraulic fracturing in soil. The stress analysis will be conducted and details related to the execution of the analysis are displayed in a window.

After this initial analysis is complete (the analyses I conducted executed in a few seconds or less), then check the deformation of the model by selecting the “POST-PROCESS” button, and then the “DEFORMD MESH”. Elastic displacements are typically too small to be critically reviewed on the screen, but the “DEFORMD MESH” capability allows the deformation to be exaggerated. Change the exaggeration factor to adjust the deformation view of the model in a reasonable manner. Also, check the distribution of stresses. Both the deformation and the stress distribution should be consistent with expectations.

It is possible that the model has been set up incorrectly and the initial conditions differ from expectations. This occurred several times while learning to use the program. As a result, it is a good idea to check the stresses and displacements prior to propagating a fracture.

**2.2.3. *Propagating the fracture***

After the model is setup and has been checked, the fracture may be propagated. This procedure is performed by selecting the “MODIFY” button in the main menu, then select the “MOVE CRACK” button in the modify menu, and the“AUTOMATIC” button in the move crack menu. In the automatic propagation mode, the user can choose the stress intensity (SIF) calculation methods, propagation direction decision methods, propagating steps, crack increment length, etc. The default selection of SIF calculation method and propagation direction decision methods should be kept for hydraulic fracture simulation. The crack increment length is set to a certain number, which can only be changed by altering source code at present. However, the extension length of fracture can be decided by choosing a certain amount of propagation steps. After setting up necessary parameters, the fracture will propagate in a continuous manner by selecting the “PROPAGATE” in the automatic menu.

**2.2.4. *Results and Analysis***

The fluid pressure, length, and aperture of the fracture is recorded with regarding to the injecting time. These data are written to a text file named “PressurevsTime.txt”. This file will be created only after the propagation is performed successfully. Deformation of the model can be evaluated by viewing the deformed grid, as outlined above. HFRANC2D can create screen shot by selecting the “SNAP”, which appears in the lower right corner.

**The basic literature**

1. Alehossein, H. and Korinets, A. *Numerical elastic analysis of a surface crack near Boussinesque load*, International Journal of Fracture 86: L39-L44, 1997.

2. Carter, B., Ingraffea, A., *Effects of casing and interface behavior on hydraulic fracture*, Computer Methods and Advances in Geomechanics, H. Siriwardane and M. Zaman, Eds., A. A. Balkema, Rotterdam, Vol. 2, 1561-1566, 1994.

3. Castell, M., Ingraffea, A. R., Irwin, L. *Fatigue crack growth in pavements*, 126, 4, 283-290, ASCE J. Transportation Engrg., 2000.

4. Geertsma, J., *Two-dimensional fracture-propagation models*, in Recent Advances in Hydraulic Fracturing. SPE Monograph, edited by Gidley, J.L., et al, 1989.

5. Geertsma, J., and Klerk, F., *A rapid method of predicting width and extent of hydraulically induced fractures*, Journal of Petroleum Technology, Vol. 21, No. 12, Dec. 1969, pp. 1571-1581.

6. Charsley, A.D., Martin, C.D., and McCreath, D.R. *Sleeve-fracturing limitations for measuring in situ stress in an anisotropic stress environment*, International Journal of Rock Mechanics & Mining Sciences 40 (2003) 127-136.

7. Fairbanks, C. D. and Andrus, R. Initial Geotechnical Investigation at Simpson Station Experimental Site, Pendleton, South Carolina.Department of Civil Engineering, Clemson University. Unpublished Report, August 2002.

8. Gidley, J.L., S.A. Holditch, and D.E. Nierode, and R.W. Veatch. 1989 Recent Advances in Hydraulic Fracturing. SPE Monograph Volume 12.452 p.

9. Grise, W. R. and Zargari, A. 1997. *Delamination and cracking failures in high-voltage stator winding coatings*, Proceedings of the 24th Electrical/Electronics Insulation Conference, 24th, 1997, 835-839.

10. Ingraffea, A.R., Numerical modeling of fracture propagation, Rock fracture mechanics, Edited by H.P. Rossmanith, published by Springer-Verlay, 1983, pp. 151-208.

11. Ingraffea, A.R., Theory of crack initiation and propagation in rock, Fracture mechanics of rock, 1987, by Academic Press Inc. (London) Ltd., pp. 71-110.

12. Ingraffea, A.R., *Case studies of simulation of fracture in concrete dams*, Fracture Mechanics Vol. 35, No. 1/2/3, pp. 553-564, 1990.

13. James, M., and Swenson, D., *FRANC2D/L: a crack propagation simulator for plane layered structures*, available from <http://www.mne.ksu.edu/~franc2d/>.

14. Lamkin, S. J., Wawrzynek, P. A. and Ingraffea, A.R., *Two - dimensional numerical simulation of interacting fractures in rock*, Fracture of Concrete and Rock: Recent Developments, S. P., Shah, S.E. Swartz, B. Barr, editors, Elsevier Science Publishing, New York, N.Y., 1989, pp. 121 - 131.

15. Lawn, B., Fracture of brittle solids, Second edition, 1993, Cambridge University Press, 1993, pp. 23-26.

16. Linsbauer, H. N., Ingraffea, A. R., Rossmanith, H. P., Wawrzynek, P. A., *Simulation of cracking in a large arch dam: part i*, Journal of Structural Engineering, Vol. 115, No. 7, July 1989, pp. 1599 - 1615.

17. Linsbauer, H. N., Ingraffea, A. R., Rossmanith, H. P., Wawrzynek, P. A., *Simulation of cracking in a large arch dam: part ii*, Journal of Structural Engineering, Vol. 115, No. 7, July, 1989, pp. 1616 - 1630.

18. Murdoch, L. C., 1991. *TheHydraulic fracturing of soil*, Ph.D. dissertation, University of Cincinnati.

19. Murdoch, L. C., 1993. *Hydraulic fracturing of soil during laboratory experiments*, Parts a-c. Geotechnique, v. 43, no. 2: 255-287.

20. Murdoch, L. C., 1994. *Induced Fractures*. United States Environmental Protection Agency. p. 39-55.

21. Murdoch, L. C., 1995. *Hydraulic and implulse fracturing for low permeability soils; in petroleum contaminated low permeability soil: hydrocarbon distribution processes, exposure pathways, and in situ remediation technologies*. Edited by Walden, T. American Petroleum Institute. p. C-1 - C-32, E-1 - E-34.

22. Murdoch, L.C., 1995. *Forms of hydraulic fractures created during a field test in overconsolidated glacial drift*. Quarterly Journal of Engineering Geology, 28, 23-35.

23. Murdoch, L.C. and Chen, J-L, (1997). *Effects of conductive fractures during in situ electroosmosis*, Journal of Hazardous Materials 55 (1997) 239-262.

24. Murdoch, L.C. and Slack, W. 2002. *Forms of hydraulic fractures in shallow fine-grained formations*. Journal of Geotechnical and Geoenvironmental Engineering, Vol. 128, No. 6, June 1, 2002, pp. 479-487.

25. Murdoch, L.C. 2002. *Analysis of an idealized shallow hydraulic fracture*. Journal of Geotechnical and Geoenvironmental Engineering, Vol. 128, No. 6, June 1, 2002, pp. 488-495.

26. Nilson, R.H., *An integral method for predicting hydraulic fracture propagation driven by gases or liquids,* International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 10, 191-211 (1986).

27. Nilson, R.H., and Griffiths, S.K., *Numerical analysis of hydraulically-driven fractures*, Computer Methods in Applied Mechanics and Engineering 36 (1983) 359-370.

28. Nilson, R.H., *Similarity solutions for wedge-shaped hydraulic fractures driven into a permeable medium by a constant inlet pressure*, International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 12, 477-495 (1988).

29. Nykanen, T. 1999. *Geometric dependency of fatigue strength in a transverse load-carrying cruciform joint with partially-penetrating K-welds*, Welding in the world, Vol. 43, No. 5, sept-oct 1999, 47-53.

30. Perkins, T. K., and Kern, L. R., *Width of hydraulic fracture*, Journal of Petroleum Technology, Vol. 13, Sept. 1961, pp. 937-949.

31. Pollard, D.D., and Holzhausen, G., On the mechanical interaction between a fluid-filled fracture and the earth’s surface, Techonophysics, 53 (1979) 27-57.

32. Richardson, J., 2003, *Forms of hydraulic fractures at shallow depth in piedmont soil*, Master’s thesis, Clemson University.

33. Rogers, C.A. and Hancock, G.J. (2001). *Fracture toughness of G550 sheet steels subjected to tension*, Journal of Constructional Steel Research 57 (2001) 71-89.

34. Sowers, G.F., and Richardson, T.L. 1982. Residual Soils of the Piedmont and Blue Ridge. Transportation Research Record No. 919, National Academy Press, Washington D.C., 10-20.

35. Srinivasan, M., *Fracture Analysis Code, A computer-aided teaching tool*, Master’s Thesis, Cornell University.

36. Tada, H., Paris, P.C. & Irwin, G.R. (1985). *The stress analysis of cracks handbook*, 2nd edn. St. Louis: Paris Productions.

37. Trego, A. and Cope, D. (2001). *Evaluation of damage tolerance analysis tools for lap joints*, AIAA Journal Vol.39, No. 12, December 2001.

38. Wang, H. F., and Anderson, M. P., 1982, *Introduction to groundwater modeling*, Academic Press, Inc.

39. Wawrzynek, P. and Ingraffea, A. (1996). FRANC2D: A two dimensional crack propagation simulator, Version 2.7 user’s guide. NASA Contractor Report 4572 (1994).

**The supplementary literature**

1. Ralph H. Abraham and Jerrold E. Marsden. Foundations of Mechanics. -Wesley, 1978.

2. V. I. Arnold. Mathematical Methods of Classical Mechanics. - Springer Verlag, 1980.

3. Lothar Nordheim. The Principles of Mechanics.Vol. 2. - Springer, 1972.

4. Jean-Marie Souriau. Structure des Systèmes Dynamiques. - Dunod Université, Paris, 1970. (English translation: Birkhäuser Boston, 1998.)

5. E. C. G. Sudarshan and N. Mukunda.Classical Dynamics: A Modern Perspective. - John Wiley & Sons, 1974.

6. V. I. Arnold, V. V. Kozlov and A. I. Neishtadt. Mathematical Aspects of Classical and Celestial Mechanics.Dynamical Systems III. - Springer Verlag, 1988.

7. Giovanni Gallavotti. The Elements of Mechanics. - Springer Verlag, 1983.

8. F. R. Gantmakher. Lectures on analytical mechanics. - Fizmatgiz, 1960. (English translation by G. Yankovsky. - Mir Publishing, 1970.)

9. Herbert Goldstein. Classical Mechanics.2nd edition. - Wesley, 1980.

10. Jorge V. José and Eugene J. Saletan.Classical Dynamics: A Contemporary Approach. - Cambridge University Press, 1998.

11. Cornelius Lanczos. The Variational Principles of Mechanics.4th edition.- University of Toronto Press, 1970; Dover Publications, 1982.

12. Пронина Р.Ф. Пособие по переводу английской научно-технической литературы. - М.: Высшая школа, 1973.

13. Циммерман М., Веденеева К. Русско-английский научно-технический словарь переводчика. - М.: Наука, 1991.

14. Арушанян О.Б. Русско-английский словарь по прикладной математике и механике. (Russian-English dictionary of applied mathematics and mechanics). – М.: МГУ, 2012.

15. Пумпянский А.Л. Чтение и перевод английской научно-технической литературы: Лексика, грамматика, фонетика, упражнения /Худ.обл. М. В. Драко.— М.: ООО «Попурри», 1997.— 608 с.

16. L. D. Landau and E. M. Lifshitz. Mechanics.3rd edition.Course of Theoretical Physics.Vol. 1. - Pergamon Press, 1976.

17. Jerrold E. Marsden and Tudor S. Ratiu.Introduction to Mechanics and Symmetry. - Springer Verlag, 1994.

18. Florian Scheck. Mechanics, From Newton's Laws to Deterministic Chaos.2nd edition. - Springer-Verlag, 1994.

19. E. T. Whittaker. A Treatise on Analytical Dynamics.- Cambridge University Press, 1937.

20. Walter Thirring. A Course in Mathematical Physics 1: Classical Dynamical Systems, translated by Evans M. Harell, - Springer-Verlag, 1978.

21. Англо-русский словарь математических терминов: около 20000 терминов / под ред. П. С. Александрова. - Изд.3-е, стер. - М.: Мир, 2001.

22. Голденков М. Современный активный English. ЧеРо. – М: Физматлит, 2002.-448с.

**The list of periodicals**

1. Английский язык: Прил. к газете "1 сентября".

2. Иностранные языки в высшей школе.

**The source list on electronic media**

1. КутателадзеС.С. Russian-English in writing. Изд. института математики. – Новосибирск, 2000.

2. Игнатьев-Каллэхэм Л. Русско-английский политехнический словарь. - М.: Уайли, 1995.

3. A.J. Lohwater's Russian-English dictionary of the mathematical sciences / R.P. Boas (ed.). AmericanMathematicalSociety, 1990.

4. Сосинский А. Б. Как написать математическую статью по-английски. — М: Изд-во «Факториал Пресс», 2000. — 112 с.

**The Internet sources**

1. http://ru.wikipedia.org

2. <http://lingualeo.com/>

**Lecture 2**

**Lecture topic: Developing HFRANC2D. Subroutines EXC\_FLOW**

**The plan**

**1. Developing HFRANC2D**

1.1. Developing HFRANC2D

1.1.1. Develop HFRANC2D in SUN Solaris System

1.1.2. Develop HFRANC2D in Microsoft Windows System

1.2. Output Files

**2. Code Structure**

2.1. Subroutine EXC\_AUTO\_DRIVER\_FLOW

2.1.1. Narrative of EXC\_AUTO\_DRIVER\_FLOW

2.1.2. Variables for subroutine EXC\_AUTO\_DRIVER\_FLOW

2.2. Subroutine EXC\_APER\_FLOW

2.2.1. Narrative of EXC\_APER\_FLOW

2.2.2.Variables for subroutine EXC\_APER\_FLOW

2.3. Subroutine EXC\_LENGTH\_FLOW

2.3.1. Narrative of EXC\_LENGTH\_FLOW

2.3.2. Variables for subroutine EXC\_LENGTH\_FLOW

**1. Developing HFRANC2D**

**1.1. *Developing HFRANC2D***

To simulate hydraulic fracturing in shallow formations, and to couple fluid flow into this fracture mechanical analysis code, some development was done in the SUN Solaris and Microsoft Windows environments. Different compiler and linkers are used in these different systems. However, the FORTRAN source code of HFRANC2D remains unchanged to fit the different systems. The new code was written in FORTRAN 90/95, which is supported by both systems.

**1.1.1. *Develop HFRANC2D in SUN Solaris System***

The source code in SUN Solaris systems is in the files with an extension name “.ff”. In each subdirectory of HFRANC2D, there is a text file named “DIRFiles”, which includes all the FORTRAN source files in the current subdirectory. If source files were added or deleted from a subdirectory, the name of those source files should also be added or deleted from that “DIRFiles” text file.

To build HFRANC2D’s executable file “Franc” under SUN Solaris, simply go to the root directory of HFRANC2D. There is a text file named “compf2d”, which contains several commands to build a new executable file. Under current directory, type this file name: “compf2d” behind command prompt, then hit “enter”. A new executable file “Franc” will be built and created in the subdirectory “Main”.

Normally, it takes 5-10 minutes to run command file “compf2d” to create all the libraries for HFRANC2D then link them together. This will compile all the FORTRAN source coed file despite it is changed recently or not. So, it is necessary to only compile the recently changed source files, then link to executable. To do this, the developer should make sure no file is added or deleted from the directory. Change directory to “Main”. Run command “make”. This will only compile those source files with changes in their code. Then run “make link” twice, so that updated libraries can be linked to created an executable file “Franc”. This file can be moved to any directories and run independently.

There are lots of powerful tools in SUN Solaris system to debug the HFRANC2D. One often used in this study is the SUN Workshop. This is an interactive debug tool with graphical interfaces.

**1.1.2. *Develop HFRANC2D in Microsoft Windows System***

HFRANC2D can be developed by Compaq Visual FORTRAN (CVF) version 5.0. Other compilers and development environments may work, but they have not been tested during this work. To open the HFRANC2D source files in CVF, go to the subdirectory “f2d”, there is a file named “f2d.dsw”, which is the workspace file recognized by CVF. Double click it, all the projects and source files will be opened in CVF.

Before compile HFRANC2D, some setting maybe made in CVF. Go to CVF’s menu. Choose “Project”, then “Settings”, then “Link”. In the “Object/Library modules”, add in three object files: gra.lib, utl.lib, utltim.obj. Then save the settings. These three are C library and system libraries, which can’t be created and found by CVF. So they need to be added manually.

**1.2. *Output Files***

After analyzing hydraulic fracturing problems, the user may want to output the result to a readable format, and also to preserve the model that has been analyzed. Some data of the forms of the hydraulic fractures will be written to several text files within the same directory of the HFRANC2D executable file “Franc” (or “f2d” in Microsoft Windows system). By clicking the button “WRITE FILE” in the main HFRANC2D menu, current settings of the model will be saved into a file. The model can be reused and analyzed by inputting the same name and choosing “0” for “Restart” while message is prompted in the DOS console window.

One output file that records the pressure, aperture, length, volume, pumping time, etc is named “PressurevsTime.txt” at present. This text file is created when a fracture is simulated under automatic propagation mode. It is created when subroutine EXC\_AUTO\_DRIVER\_FLOW is executed. Ten variables are recorded into PressurevsTime.txt after each step of fracture automatic propagation. The names and short explanations of these variables are listed in the order as they appear in the PressurevsTime.txt file as follows:

Step\_Count counter of fracture propagation steps

tiphead hydraulic head of the fracture tip

fp\_print(1) fluid pressure at the fracture tip

numcells number of finite grids used in fluid flow simulation

Crack\_Length length of the fracture (from mouth to tip)

Opening the aperture at the fracture mouth opening

SIF(1,1) mode I stress intensity factor

Volume volume of fracture

Fracvol volume of fracture calculated by fluid flow subroutines

Pumping\_Time time needed to fill fluid into fracture of Volume with a constant flow rate

The user may also want to know the fluid pressure and aperture distributions along the fracture face, from mouth opening to the fracture tip. “PProfile.txt” will record the fluid pressure and aperture profile for this purpose. The fluid pressure values will be written to this file after the fracture propagates and before fluid flows into the fracture. So it will record the fluid pressure that brings the stress intensity factor to the fracture toughness.

**2.** **Code Structure**

The hydraulic fracturing propagation procedure is implemented by a subroutine named EXC\_AUTO\_DRIVER\_FLOW. This subroutine is originated from a subroutine named EXC\_AUTO\_DRIVER, with fluid flow calculation subroutine being called in it. Another important change is that the propagation criterion is set in this subroutine so that propagation only happens when mode I stress intensity equals to the fracture toughness. Then the data of hydraulic fracture forms and fluid flow are output to text files.

The propagation criterion is established by a subroutine named “adjusttipheadtoprop”. Stress intensity factor is compared to the fracture toughness. Adjustment is then made according to the comparison results.

The geometric form of the hydraulic fracture is described by pressure, aperture, and length. Three subroutines, EXC\_APER\_FLOW, EXC\_LENGTH\_FLOW, EXC\_VOLU\_FLOW were developed to describe the changes of those three variables with time. The time will be calculated by dividing volume by pumping rate.

Fluid flow calculation is implemented in two subroutines: “fluidflowinit” and “fluidflowcalc”. Initialization of parameters and the finite difference grid is implemented in “fluidflowinit”. Other fluid flow calculations are completed in “fluidflowcalc”. Both of them are called by EXC\_AUTO\_DRIVER\_FLOW so that fluid flow is coupled with mechanical analysis.

**2.1. *Subroutine EXC\_AUTO\_DRIVER\_FLOW***

Subroutine EXC\_AUTO\_DRIVER\_FLOW controls automatic propagation when the user clicks “PROPAGATE” button in the automatic propagation menu. This subroutine will propagate the fracture when the propagation criterion is satisfied.

**2.1.1. *Narrative of EXC\_AUTO\_DRIVER\_FLOW***

The EXC\_AUTO\_DRIVER\_FLOW implements the hydraulic fracturing propagation in an automatic mode. Once the user has clicked the button “PROPAGATE” in HFRANC2D, this routine will be triggered and run. It functions as follows (figure 3.3-1):

1. Initialize propagation completion flag, step count, and other variables.
2. Initialize the fluid pressure with proper values.
3. Call ANL\_LINEAR to calculate the aperture of crack with the initial fluid pressure.
4. Calculate the stress intensity factor, and write it to winged data base file (WDB), which records fracture model information.
5. Assign a value to fracture toughness. This toughness value will be used as propagation criteria. This method is used if the fracture is assumed to be propagating using the applied initial pressure.
6. Check the crack tip position, and make sure it is far enough from a boundary so that the fracture will stay in the model geometry after propagation. Stop propagation and go to 14 if it is too close to a boundary.
7. Call FLUID\_FLOW\_CALC to determine fluid pressures that satisfy stress equilibrium.
8. Calculate the geometry of the fracture.
9. Calculate the time needed to pump fluid into the crack.
10. Save data into a text file.
11. Calculate the increment to be added to the crack. Revise the crack length.
12. Extend fracture and remesh the vicinity of fracture tip automatically.
13. Check the step count. If it has reached the intended steps, then stop propagation, and return. Else, go to step 6.

**2.1.2. *List of variables for subroutine EXC\_AUTO\_DRIVER\_FLOW***

Local variables of subroutine EXC\_ AUTO\_DRIVER \_FLOW have been defined in the front part of the code, which is labeled “local variables and symbols”. Local constants used by subroutine EXC\_ AUTO\_DRIVER \_FLOW have been defined in the front part of the code, which is labeled “local constants”. Global variables are defined in “include” files, which are quoted in the section “global variables and symbols”.

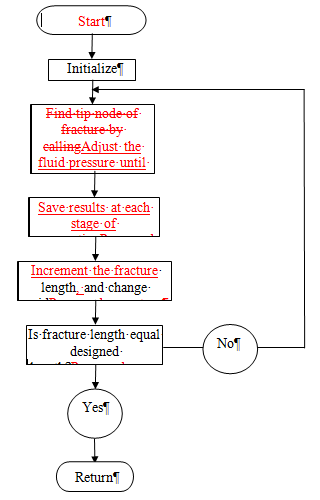


Figure 3.3-1. Flow chart of subroutine EXC\_AUTO\_DRIVER\_FLOW

**2.2. *Subroutine EXC\_APER\_FLOW***

The subroutine EXC\_APER\_FLOW will return an array that includes aperture profile of the fracture from fracture mouth opening to fracture tip. The following flow chart describes the procedure of this subroutine (figure 3.3-2).

**2.2.1. *Narrative of EXC\_APER\_FLOW***

Aperture of the crack is calculated by subroutine EXC\_APER\_FLOW. It will return the crack apertures of each segment in the crack between crack mouth opening, and crack tip. The number of aperture values is decided by the finite element grids around the fracture (figure 3.3-3). In the example fracture, there are five node pairs from the crack mouth opening to the crack tip. So, there will have five aperture values between each crack node pair.

**2.2.2.*Variables for subroutine EXC\_APER\_FLOW***

Local variables of subroutine EXC\_APER\_FLOW has been defined in the front part of the code, which is labeled “local variables and symbols”.

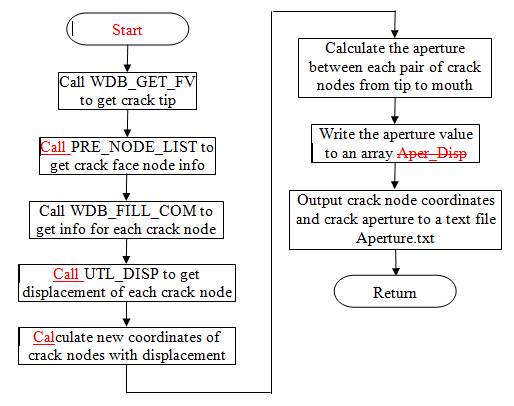


Figure 3.3-2. Flow chart of subroutine EXC\_APER\_FLOW

Local constants used by subroutine EXC\_APER\_FLOW have been defined in the front part of the code, which is labeled “local constants.” Global variables are defined in “include” files, which are quoted in the section “global variables and symbols.” This subroutine takes two input variables: “Data\_Storage” and “A”. It has one output variable: “Aper\_Disp”, which is an array with all the fracture aperture data recorded.

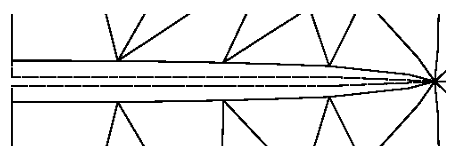


Figure 3.3-3.Finite element grids around a fracture. The fracture shown here has an opening at left, and has a tip at right. The dash line is the initial fracture cross section, solid line represent the fracture cross section after deformation

**2.3. *Subroutine EXC\_LENGTH\_FLOW***

The subroutine EXC\_LENGTH\_FLOW will calculate the length of the fracture, from crack mouth open to crack tip; it will also calculate the distance between crack mouth opening to each node pair. So two variables are returned, one is an single REAL type variable: Crack\_Length, which will record the length of fracture; another is an REAL type array, which will contain a serial values of distance from crack mouth opening to crack nodes.

**2.3.1. *Narrative of EXC\_LENGTH\_FLOW***

The procedure of subroutine EXC\_LENGTH\_FLOW is similar to its brother subroutine: EXC\_APER\_FLOW. Basically, they first find a crack tip node. Then crack face nodes information is got for both left and right crack face sides. The crack face node information include crack face node index, crack face node numbers, and crack face node coordinates. Then displacement of each crack face node is computed by a function, and is added to the crack face node coordinates. Then it will loop through all the crack face node pairs to calculate the distances between crack mouth opening and them. Here the distance calculated is the distance from the mid-point of the crack mouth opening to the mid-point of each pair of crack face node (figure 3.3-4).

Once the subroutine is called it will be executed in the following order:

1. Call subroutine WDB\_GET\_FV to get the first crack tip node of the fracture.

2. Reserve a piece of memory space by calling subroutine UTLGET.

3. Call subroutine PRE\_NODE\_LIST to get crack face node index, node coordinates, and node numbers of one side. This subroutine will be called twice to retrieve data for both left and right sides.

4. Loop through crack tip node till crack mouth opening node, to call function UTL\_DISP to return displacement of crack face nodes. UTL\_DISP is called twice for both x and y coordinates of the crack face nodes.

5. Displacement of each crack face node will be added to their original coordinate, this will be its new coordinate.

6. Calculate the mid-point position of each crack face node pairs.

7. Calculate the distance from the mid-point of crack mouth node pair to other crack face node pairs. Fracture length is the distance from crack mouth mid-point to crack tip.

8. Output crack length, distances from crack mouth to crack face node pairs, crack face node coordinates, mid-point of crack face node pairs, and crack face node indexes to a text file named “length.txt”.

9. Return.

The subroutine EXC\_LENGTH\_FLOW could be described by a flow chart (figure 3.3-5).

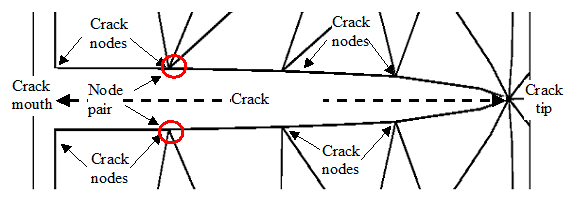


Figure 3.3-4. Terms used in fracture geometry calculation

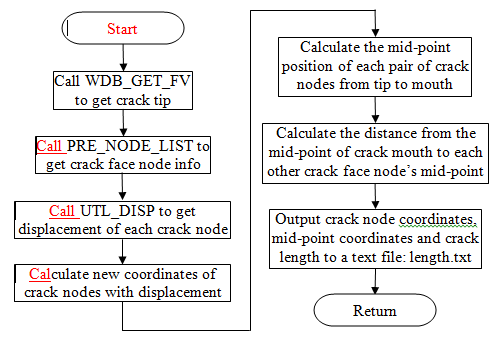


Figure 3.3-5. Flow chart of subroutine EXC\_LENGTH\_FLOW

**2.3.2. *Variables for subroutine EXC\_LENGTH\_FLOW***

Local variables of subroutine EXC\_LENGTH\_FLOW has been defined in the front part of the code, which is labeled “local variables and symbols”. Local constants used by subroutine EXC\_LENGTH\_FLOW have been defined in the front part of the code, which is labeled “local constants”. Global variables are defined in “include” files, which are quoted in the section “global variables and symbols”. This subroutine takes one input variable: “Data\_Storage”. It has two output variables: “Crack\_Length”, and “Open\_to\_Node\_Length”. The detailed explanations of these variables are listed in appendix (figure 3.3-6.).

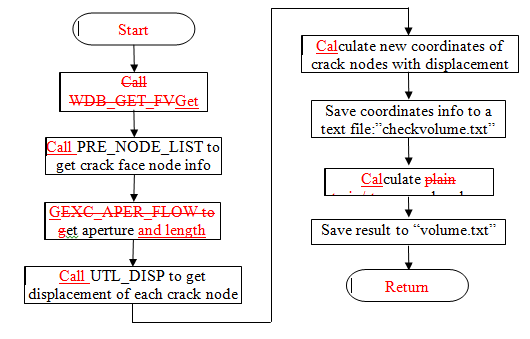


Figure 3.3-6. Flow chart of subroutine EXC\_VOLU\_FLOW

**The basic literature**

1. Alehossein, H. and Korinets, A. (1997). *Numerical elastic analysis of a surface crack near Boussinesque load*, International Journal of Fracture 86: L39-L44, 1997.

2. Carter, B., Ingraffea, A., *Effects of casing and interface behavior on hydraulic fracture*, Computer Methods and Advances in Geomechanics, H. Siriwardane and M. Zaman, Eds., A. A. Balkema, Rotterdam, Vol. 2, 1561-1566, 1994.

3. Castell, M., Ingraffea, A. R., Irwin, L. *Fatigue crack growth in pavements*, 126, 4, 283-290, ASCE J. Transportation Engrg., 2000.

4. Geertsma, J., *Two-dimensional fracture-propagation models*, in Recent Advances in Hydraulic Fracturing. SPE Monograph, edited by Gidley, J.L., et al, 1989.

5. Geertsma, J., and Klerk, F., *A rapid method of predicting width and extent of hydraulically induced fractures*, Journal of Petroleum Technology, Vol. 21, No. 12, Dec. 1969, pp. 1571-1581.

6. Charsley, A.D., Martin, C.D., and McCreath, D.R. *Sleeve-fracturing limitations for measuring in situ stress in an anisotropic stress environment*, International Journal of Rock Mechanics & Mining Sciences 40 (2003) 127-136.

7. Fairbanks, C. D. and Andrus, R. Initial Geotechnical Investigation at Simpson Station Experimental Site, Pendleton, South Carolina.Department of Civil Engineering, Clemson University. Unpublished Report, August 2002.

8. Gidley, J.L., S.A. Holditch, and D.E. Nierode, and R.W. Veatch. 1989 Recent Advances in Hydraulic Fracturing. SPE Monograph Volume 12.452 p.

9. Grise, W. R. and Zargari, A. 1997. *Delamination and cracking failures in high-voltage stator winding coatings*, Proceedings of the 24th Electrical/Electronics Insulation Conference, 24th, 1997, 835-839.

10. Ingraffea, A.R., Numerical modeling of fracture propagation, Rock fracture mechanics, Edited by H.P. Rossmanith, published by Springer-Verlay, 1983, pp. 151-208.

11. Ingraffea, A.R., Theory of crack initiation and propagation in rock, Fracture mechanics of rock, 1987, by Academic Press Inc. (London) Ltd., pp. 71-110.

12. Ingraffea, A.R., *Case studies of simulation of fracture in concrete dams*, Fracture Mechanics Vol. 35, No. 1/2/3, pp. 553-564, 1990.

13. James, M., and Swenson, D., *FRANC2D/L: a crack propagation simulator for plane layered structures*, available from <http://www.mne.ksu.edu/~franc2d/>.

14. Lamkin, S. J., Wawrzynek, P. A. and Ingraffea, A.R., *Two - dimensional numerical simulation of interacting fractures in rock*, Fracture of Concrete and Rock: Recent Developments, S. P., Shah, S.E. Swartz, B. Barr, editors, Elsevier Science Publishing, New York, N.Y., 1989, pp. 121 - 131.

15. Lawn, B., Fracture of brittle solids, Second edition, 1993, Cambridge University Press, 1993, pp. 23-26.

16. Linsbauer, H. N., Ingraffea, A. R., Rossmanith, H. P., Wawrzynek, P. A., *Simulation of cracking in a large arch dam: part i*, Journal of Structural Engineering, Vol. 115, No. 7, July 1989, pp. 1599 - 1615.

17. Linsbauer, H. N., Ingraffea, A. R., Rossmanith, H. P., Wawrzynek, P. A., *Simulation of cracking in a large arch dam: part ii*, Journal of Structural Engineering, Vol. 115, No. 7, July, 1989, pp. 1616 - 1630.

18. Murdoch, L. C., 1991. *TheHydraulic fracturing of soil*, Ph.D. dissertation, University of Cincinnati.

19. Murdoch, L. C., 1993. *Hydraulic fracturing of soil during laboratory experiments*, Parts a-c. Geotechnique, v. 43, no. 2: 255-287.

20. Murdoch, L. C., 1994. *Induced Fractures*. United States Environmental Protection Agency. p. 39-55.

21. Murdoch, L. C., 1995. *Hydraulic and implulse fracturing for low permeability soils; in petroleum contaminated low permeability soil: hydrocarbon distribution processes, exposure pathways, and in situ remediation technologies*. Edited by Walden, T. American Petroleum Institute. p. C-1 - C-32, E-1 - E-34.

22. Murdoch, L.C., 1995. *Forms of hydraulic fractures created during a field test in overconsolidated glacial drift*. Quarterly Journal of Engineering Geology, 28, 23-35.

23. Murdoch, L.C. and Chen, J-L, (1997). *Effects of conductive fractures during in situ electroosmosis*, Journal of Hazardous Materials 55 (1997) 239-262.

24. Murdoch, L.C. and Slack, W. 2002. *Forms of hydraulic fractures in shallow fine-grained formations*. Journal of Geotechnical and Geoenvironmental Engineering, Vol. 128, No. 6, June 1, 2002, pp. 479-487.

25. Murdoch, L.C. 2002. *Analysis of an idealized shallow hydraulic fracture*. Journal of Geotechnical and Geoenvironmental Engineering, Vol. 128, No. 6, June 1, 2002, pp. 488-495.

26. Nilson, R.H., *An integral method for predicting hydraulic fracture propagation driven by gases or liquids,* International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 10, 191-211 (1986).

27. Nilson, R.H., and Griffiths, S.K., *Numerical analysis of hydraulically-driven fractures*, Computer Methods in Applied Mechanics and Engineering 36 (1983) 359-370.

28. Nilson, R.H., *Similarity solutions for wedge-shaped hydraulic fractures driven into a permeable medium by a constant inlet pressure*, International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 12, 477-495 (1988).

29. Nykanen, T. 1999. *Geometric dependency of fatigue strength in a transverse load-carrying cruciform joint with partially-penetrating K-welds*, Welding in the world, Vol. 43, No. 5, sept-oct 1999, 47-53.

30. Perkins, T. K., and Kern, L. R., *Width of hydraulic fracture*, Journal of Petroleum Technology, Vol. 13, Sept. 1961, pp. 937-949.

31. Pollard, D.D., and Holzhausen, G., On the mechanical interaction between a fluid-filled fracture and the earth’s surface, Techonophysics, 53 (1979) 27-57.

32. Richardson, J., 2003, *Forms of hydraulic fractures at shallow depth in piedmont soil*, Master’s thesis, Clemson University.

33. Rogers, C.A. and Hancock, G.J. (2001). *Fracture toughness of G550 sheet steels subjected to tension*, Journal of Constructional Steel Research 57 (2001) 71-89.

34. Sowers, G.F., and Richardson, T.L. 1982. Residual Soils of the Piedmont and Blue Ridge. Transportation Research Record No. 919, National Academy Press, Washington D.C., 10-20.

35. Srinivasan, M., *Fracture Analysis Code, A computer-aided teaching tool*, Master’s Thesis, Cornell University.

36. Tada, H., Paris, P.C. & Irwin, G.R. (1985). *The stress analysis of cracks handbook*, 2nd edn. St. Louis: Paris Productions.

37. Trego, A. and Cope, D. (2001). *Evaluation of damage tolerance analysis tools for lap joints*, AIAA Journal Vol.39, No. 12, December 2001.

38. Wang, H. F., and Anderson, M. P., 1982, *Introduction to groundwater modeling*, Academic Press, Inc.

39. Wawrzynek, P. and Ingraffea, A. (1996). FRANC2D: A two dimensional crack propagation simulator, Version 2.7 user’s guide. NASA Contractor Report 4572 (1994).

**The supplementary literature**

1. Ralph H. Abraham and Jerrold E. Marsden. Foundations of Mechanics. -Wesley, 1978.

2. V. I. Arnold. Mathematical Methods of Classical Mechanics. - Springer Verlag, 1980.

3. Lothar Nordheim. The Principles of Mechanics.Vol. 2. - Springer, 1972.

4. Jean-Marie Souriau. Structure des Systèmes Dynamiques. - Dunod Université, Paris, 1970. (English translation: Birkhäuser Boston, 1998.)

5. E. C. G. Sudarshan and N. Mukunda.Classical Dynamics: A Modern Perspective. - John Wiley & Sons, 1974.

6. V. I. Arnold, V. V. Kozlov and A. I. Neishtadt. Mathematical Aspects of Classical and Celestial Mechanics.Dynamical Systems III. - Springer Verlag, 1988.

7. Giovanni Gallavotti. The Elements of Mechanics. - Springer Verlag, 1983.

8. F. R. Gantmakher. Lectures on analytical mechanics. - Fizmatgiz, 1960. (English translation by G. Yankovsky. - Mir Publishing, 1970.)

9. Herbert Goldstein. Classical Mechanics.2nd edition. - Wesley, 1980.

10. Jorge V. José and Eugene J. Saletan.Classical Dynamics: A Contemporary Approach. - Cambridge University Press, 1998.

11. Cornelius Lanczos. The Variational Principles of Mechanics.4th edition.- University of Toronto Press, 1970; Dover Publications, 1982.

12. Пронина Р.Ф. Пособие по переводу английской научно-технической литературы. - М.: Высшая школа, 1973.

13. Циммерман М., Веденеева К. Русско-английский научно-технический словарь переводчика. - М.: Наука, 1991.

14. Арушанян О.Б. Русско-английский словарь по прикладной математике и механике. (Russian-English dictionary of applied mathematics and mechanics). – М.: МГУ, 2012.

15. Пумпянский А.Л. Чтение и перевод английской научно-технической литературы: Лексика, грамматика, фонетика, упражнения /Худ.обл. М. В. Драко.— М.: ООО «Попурри», 1997.— 608 с.

16. L. D. Landau and E. M. Lifshitz. Mechanics.3rd edition.Course of Theoretical Physics.Vol. 1. - Pergamon Press, 1976.

17. Jerrold E. Marsden and Tudor S. Ratiu.Introduction to Mechanics and Symmetry. - Springer Verlag, 1994.

18. Florian Scheck. Mechanics, From Newton's Laws to Deterministic Chaos.2nd edition. - Springer-Verlag, 1994.

19. E. T. Whittaker. A Treatise on Analytical Dynamics.- Cambridge University Press, 1937.

20. Walter Thirring. A Course in Mathematical Physics 1: Classical Dynamical Systems, translated by Evans M. Harell, - Springer-Verlag, 1978.

21. Англо-русский словарь математических терминов: около 20000 терминов / под ред. П. С. Александрова. - Изд.3-е, стер. - М.: Мир, 2001.

22. Голденков М. Современный активный English. ЧеРо. – М: Физматлит, 2002.-448с.

**The list of periodicals**

1. Английский язык: Прил. к газете "1 сентября".

2. Иностранные языки в высшей школе.

**The source list on electronic media**

1. КутателадзеС.С. Russian-English in writing. Изд. института математики. – Новосибирск, 2000.

2. Игнатьев-Каллэхэм Л. Русско-английский политехнический словарь. - М.: Уайли, 1995.

3. A.J. Lohwater's Russian-English dictionary of the mathematical sciences / R.P. Boas (ed.). AmericanMathematicalSociety, 1990.

4. Сосинский А. Б. Как написать математическую статью по-английски. — М: Изд-во «Факториал Пресс», 2000. — 112 с.

**The Internet sources**

1. http://ru.wikipedia.org

2. <http://lingualeo.com/>

**Lecture 3**

**Lecture topic: Subroutines. Changes Made in HFRANC2D**

**The plan**

**1. Subroutines**

1.1. Subroutine EXC\_VOLU\_FLOW

1.1.1. Narrative of EXC\_VOLU\_FLOW

1.1.2. Variables for subroutine EXC\_VOLU\_FLOW

1.2. Subroutine FLUID\_FLOW\_INIT

1.3. Subroutine Initialize

1.4. Subroutine MakeFlowGrid

1.5. Subroutine TrialApt

1.6. Subroutine InitializeTrial

1.7. Subroutine FLUID\_FLOW\_CALC

1.8. Subroutine FindHeadSolution

1.9. Subroutine CheckKI

1.10. Subroutine AdjustTipHeadToProp

**2. Changes Made in HFRANC2D**

2.1. Changes made in subroutine UTL\_CRACK\_FACE\_FORCE\_FLOW

2.2. Other changes

**1. Subroutines**

**1.1. *Subroutine EXC\_VOLU\_FLOW***

The subroutine EXC\_VOLU\_FLOW will be called to return a value of fracture volume. The flow chart of EXC\_VOLU\_FLOW are shown in previous page (figure 3.3-6).

**1.1.1. *Narrative of EXC\_VOLU\_FLOW***

The volume of the hydraulic fracture is calculated by the subroutine EXC\_VOLU\_FLOW, which exists in a file named excvoluflow.f. This file is in the subdirectory of \EXC.

The EXC\_VOLU\_FLOW will return the volume of a crack. Both AxisymmetricAxial symmetric and rectangular plane stress/strain conditions are considered. The subroutine will be implemented under following order:

1. Search for all crack tip nodes, at present only one crack tip will be used in modeling.

2. If a crack tip is found, then start to process; else return to the caller subroutine.

3. Call PRE\_NODE\_LIST to get the crack node information, which includes: index of nodes, coordinates of nodes, and node numbers of one side of the crack.

4. Call EXC\_APER\_FLOW to calculate the aperture of the crack.

5. Call EXC\_LENGTH\_FLOW to calculate the length of the crack.

6. Loop through all the crack nodes starting from crack tip nodes to retrieve the displacement value by calling function UTL\_DISP.

7. Add the displacement value to the original coordinates of crack nodes.

8. After finding the new coordinates for all the crack nodes, write them to a text file.

9. If the model is set under axial symmetric condition, then calculate the volume of crack under this condition. First, volume of every segment of the crack will be calculated. Then all those values will be added together. That will give the total volume of crack.

10. If the model is set under rectangular plane stress/strain condition, calculate the area of the crack. Then multiply the area with the thickness of the material. This will give the volume of the crack.

11. Save the volume to a text file.

12. Return.

**1.1.2. *Variables for subroutine EXC\_VOLU\_FLOW***

Local variables of subroutine EXC\_VOLU\_FLOW has been defined in the front part of the code, which is labeled “local variables and symbols”. Local constants used by subroutine EXC\_VOLU\_FLOW have been defined in the front part of the code, which is labeled “local constants”. Global variables are defined in “include” files, which are quoted in the section “global variables and symbols”.

This subroutine takes two vectors as input variables: “Data\_Storage” and “A”. It has one output variable: “Volume”, which has the value of the fracture volume.

**1.2. *Subroutine FLUID\_FLOW\_INIT***

This subroutine will initialize fluid flow by calling subroutine “Initialize”, and make the first guess of aperture and hydraulic head. This subroutine and the subroutine “FLUID\_FLOW\_CALC” were first written in one subroutine by Murdoch [2002]. The original subroutine were then separated into two subroutines for the coupling with HFRANC2D. Each subroutine will implement a specific purpose in fluid flow simulation. This subroutine can be found in the “Appendix A” and a CD-ROM at the end of this thesis.

The FLUID\_FLOW\_INIT is called by EXC\_AUTO\_DRIVER\_FLOW. The execution of this subroutine is in the following order:

1. Call EXC\_LENGTH\_FLOW to calculate fracture length at present.
2. Assign this length value as initial fracture length.
3. Start to count computer CPU time.
4. Setup a ratio (Crackmult) for fracture extension.
5. Setup a value for the length of fluid flow finite difference grid (delr).
6. Calculate number of fluid flow grids needed.
7. Call subroutine Initialize.
8. Call subroutine MakeFlowGrid.
9. Call subroutine TrialApt.
10. Assign the mode I stress intensity factor as fracture toughness, also as propagation criterion.
11. Call subroutine InitializeTrial.
12. Call subroutine InitializeLeakoff.
13. Call subroutine WriteResults.
14. Call subroutine CheckUnsatFuncs.
15. Assign the trial hydraulic head at fracture tip to head at tip.
16. Return.

**1.3. *Subroutine Initialize***

The subroutine Initialize is called by subroutine FLUID\_FLOW\_INIT to initialize variables used in fluid flow simulation. Some trial values of fracture geometry, such as fracture length, fracture aperture at the opening, volume, will also be initialized. This subroutine is contained in file “fluidflowinit.ff” (“fluidflowinit.f” for MS Windows system).

When subroutine Initialize is called it will execute in the following order:

1. Assign a value to the following values: aperture tolerance for iteration convergence, confining pressure of the hydraulic fracture, current half length of the fracture, time step and previous one, elastic modulus, flow constant, height of fracture, head tolerance for iteration convergence, initial aperture of the fracture,

Initial crack volume, initial crack volume, critical stress intensity factor, last time step, Carter coefficient of leak off, mass balance tolerance, output file name for fluid flow simulation results, Poisson’s ration, volumetric flow rate into the fracture, effective aperture after closure, initial time, and injection time.

2. Assign a logical value to the following Boolean variables: crack closing first step during closure, and propagating.

3. Initialize aperture, fracture slope, leak off flux, elevation, hydraulic head, fluid flow flux in the fracture.

4. Call DetermineFPFem to calculate the fluid pressure inside fracture.

5. Initialize the data log to 0. This data log contains pressure, length, aperture, and time. Then return.

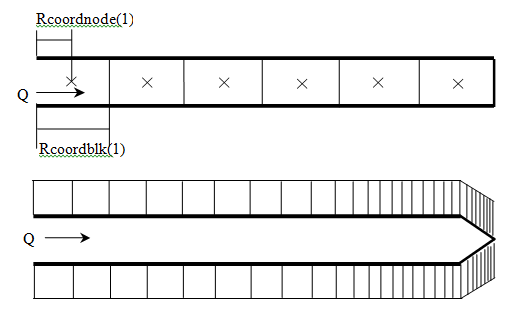


Figure 3.3-7. FDM for flow and FEM for deformation simulations

**1.4. *Subroutine MakeFlowGrid***

The subroutine MakeFlowGrid will define finite difference grids used for fluid flow simulation. It is contained in the file “fluidflowinit.ff” ( “fluidflowinit.f” for MS Windows system). This subroutine will make evenly spaced grid for fluid flow in fracture. The finite difference grids start at the fracture mouth-opening, end at the fracture tip (figure 3.3-7). The size of the grid is manually selected by a variable “delr”.

Subroutine MakeFlowGrid will define the finite difference grid for fluid flow simulation in the following orders:

1. Calculate the distance from the fracture mouth to each end of the finite difference grid block. This distance is referred to as an array “rcoordblk” (figure 3.3-7).

2. Calculate the distance from the fracture mouth to the center of each of the finite difference grid. This distance is referred to as an array “rcoordnode” (figure 3.3-7).

3. Return

**1.5. *Subroutine TrialApt***

The subroutine TrialApt will determine fracture aperture created by a given fluid pressure. This subroutine is contained in the file “fluidflowcalc.ff” (“fluidflowcalc.f” in MS Windows system). It is called by subroutine FLUID\_FLOW\_INIT to compute the trial aperture after the trial hydraulic head has been guessed. This subroutine is also called by other subroutines so that each time hydraulic head has been changed, the according aperture will be computed.

The subroutine TrialApt will be executed in the following order:

1. Call subroutine DetermineFPFem to compute the fluid pressure applied to the finite element used for mechanical analysis. Fluid pressure induced by hydraulic head is first computed for finite difference grid, which is different from finite element grid used for mechanical analysis. Then the fluid pressure will be interpolated and transformed to finite element grid.

2. Compute the fracture aperture induced by fluid pressure together with other pressure and loads. This is done by calling subroutines ANL\_LINEAR, PST\_SIF\_ADD\_INFO, PST\_AUTO\_SIF, and EXC\_APER\_FLOW. Thus, apertures for every finite element grid, induced fluid pressure and other loads are got.

3. Call subroutine TransferFemtoFd to convert apertures at finite element grid location to apertures at finite difference grid location used by fluid flow simulation.

4. Add initial aperture to the aperture has just been calculated. This initial aperture value was initially created when the fracture is initially created by cutting a slot in the material.

5. Return.

**1.6. *Subroutine InitializeTrial***

The subroutine InitializeTrial will compute the trial aperture after a trial head value is given. This subroutine is called by FLUID\_FLOW\_INIT. It is contained in the file “fluidflowinit.ff” (“fluidflowinit.f” in MS Windows system).

The subroutine InitializeTrial is executed in the following order:

1. Set logical variable “FoundSolution” to false.

2. Call function “FracVolRect” to compute the fracture volume and store it to a variable “fracvol”.

3. Assign current trial hydraulic head values with doubled hydraulic head values. Save the old trial hydraulic head values.

4. Call subroutine DetermineFPFem to determine fluid pressure induced by this trial hydraulic head.

5. Call subroutine TrialApt to calculate the trial aperture induced by the new trial hydraulic head. Then call this subroutine again to calculate the trial aperture induced by the old trial hydraulic head.

6. Return.

**1.7. *Subroutine FLUID\_FLOW\_CALC***

Subroutine FLUID\_FLOW\_CALC will implement most of the fluid flow simulation in the fracture. Fluid losses into the formation will also be simulated in this subroutine.

Subroutine FLUID\_FLOW\_CALC will be called by subroutine EXC\_AUTO\_DRIVER\_FLOW, at each time automatic propagation is executed. The subroutine will execute the following actions above call:

1. If time step is smaller than stop step, then enter a DO loop. Else go to 7.

2. Call subroutine Find Head Solution to calculate heads in fracture. This routine will set hokfornow to true if it converges properly.

3. If injecting but not propagating, then call subroutine CheckKI to check the stress intensity. Propagating is set to true by this routine when KI = KIc

4. If propagating, call subroutine AdjustTipHeadToProp, which will adjust hydraulic head at fracture tip according to propagation criterion. This routine will set foundsolution to true when the propagation criterion is met.

5. If foundsolution is true or hokfornow is true, exit this do loop by going to 7. Otherwise return to 1, check the count of do loop, if it is greater than 15, then print out an error message.

6. Call CheckMB to check mass balance.

7. Call SaveResults to save results to output text files.

8. Assign number of new fluid grid to be added to 1.

9. Return.

**1.8. *Subroutine FindHeadSolution***

Subroutine FindHeadSolution is called by FLUID\_FLOW\_CALC to determine hydraulic heads in the fracture. This hydraulic head is affected by changes in the aperture of the fracture. The aperture of the fracture is determined by the mechanical analysis HFRANC2D has. By doing this fluid, flow analysis is coupled with the mechanical analysis in HFRANC2D. This subroutine exists in the file “fluidflowcalc.ff” (“fluidflowcalc.f” in MS Windows system).

The subroutine FindHeadSolution is executed in the following order:

1. Start a loop (loop counter variable is “jj”) with the max loop count set to double the max iteration number, which is defined and assigned early in this subroutine.

2. If the trial aperture in the fluid flow grid next to last grid including fracture tip is zero, then check trial hydraulic head in every fluid flow grid to see if they are zero; if so, make them equal to hydraulic head value at fracture tip. This is only required if more than one grid block is added between iterations.

3. Call TrialApt to compute a trial aperture values for trial hydraulic heads.

4. Call subroutine DetermineHead to compute hydraulic heads while using the trial aperture values from step 3.

5. Call subroutine DetermineFPFem to compute fluid pressures along the finite element grid.

6. Call subroutine TrialApt to compute fracture aperture with the current fluid pressures

7. Call subroutine CheckForChange to get the changes of both fracture apertures and hydraulic heads. A new guess for aperture is determined as the weighted average of the previous two iterations. “Hokfornow” is set to true when the heads and/or apertures have changed by an amount that is less than a tolerance value.

8. If the logical variable “hokfornow” is true and “jj” is greater than 2, then go to 11 exit the loop.

9. If “jj” is half of the max iteration number, assign “true” to a logical variable changeweight. This allows the iteration to continue with a revised weighting factor. Sometimes this helps convergence. This is the end of “jj” loop.

10. If “jj” is greater than twice the max iteration number, then print out an error message in the console window on the screen.

11. Return.

**1.9. *Subroutine CheckKI***

The subroutine CheckKI is called in the subroutine FLUID\_FLOW\_CALC to compare the mode I stress intensity factor with a critical value of stress intensity. Depending on the comparison result, adjustment will be made to the hydraulic head at tip. This subroutine is contained in the file “fluidflowcalc.ff” (“fluidflowcalc.f” in MS Windows system).

The subroutine CheckKI will be executed in the following order:

1. Assign mode I stress intensity factor, which is represented by a global variable “SIF(1,1)”, to a variable “CurrentKI”.

2. If CurrentKI is greater than the critical value of stress intensity, KIC, and the fracture is not propagating, then set the “propagating” to be true.

3. If the fracture is not propagating, increase the hydraulic head at the tip by 10%, and let a logical variable “foundsolution” be true.

4. Return.

**1.10. *Subroutine AdjustTipHeadToProp***

The subroutine AdjustTipHeadToProp is called by subroutines EXC\_AUTO\_DRIVER\_FLOW, and FLUID\_FLOW\_CALC. When it is called, the hydraulic head at the fracture will be adjusted according to a tolerance value of the comparison of current mode I stress intensity factor and a critical value of it. This subroutine is existed in a file named “adjusttipheadtoprop.ff” (“adjusttipheadtoprop.f” for MS Windows system).

When subroutine adjusttipheadtoprop is called, it will execute in the following orders:

1. Call subroutine DetermineFPFem to compute fluid pressure with given hydraulic heads.

2. Call subroutines ANL\_LINEAR, PST\_SIF\_ADD\_INFO, PST\_AUTO\_SIF to calculate stress intensity with fluid pressure.

3. Assign mode I stress intensity factor, which is represented by a global variable “SIF(1,1)”, to a variable “CurrentKI”.

4. Set a tolerance value of stress intensity factor to 1 percent.

5. Assign maximum and minimum stress intensity factors. They will be 1 percent more or less than the critical stress intensity factor.

6. If CurrentKI is between the maximum and minimum stress intensity factors, and a loop counter “jj” is greater than 1, then assign logical variable “foundsolution” to true.

7. IF “foundsolution” is not true, and “jj” is 1, decrease the hydraulic head at fracture by 10 percent.

8. If “foundsolution” is not true, and “jj” is greater than 1, then adjust the hydraulic head at the fracture tip using linear interpolation of the previous values of head and stress intensity.

9. If “foundsolution” is not true, then store current values of hydraulic head at fracture tip, current stress intensity factor, and new hydraulic head at fracture tip.

10. Return.

**2. Changes Made in HFRANC2D**

Several changes are made in HFRANC2D’s original source code. These changes are made so that fluid flow simulation can be performed. Some other changes are made to suit the changes of menus and buttons in the interface.

**2.1. *Changes made in subroutine UTL\_CRACK\_FACE\_FORCE\_FLOW***

The subroutine UTL\_CRACK\_FACE\_FORCE\_FLOW will apply pressure to the crack face. Its original version (UTL\_CRACK\_FACE\_FORCE) can only apply uniform load to the crack face. In this version, an array variable “fp\_F2D(200)” was introduced into this subroutine in place of a single floating variable “Crack\_Pressure” used before, so that pressure distribution can be applied to the crack face. This will enable the use of fluid pressure created by coupled fluid flow.

**2.2. *Other changes***

Some menu buttons are added into the existing menu system so that controls of fluid flow can be handled interactively. A button named “FLUID FLOW” is added under menu pre-process. When it is clicked, a fluid flow menu will be displayed. Now it has only one button “VOL FLOW RATE”, which will be used to input a value for the volumetric fluid flow rate injected into fracture. Several buttons are added into post-process menu. In this menu, when button “MORE OPTIONS” is clicked, four buttons will be displayed for the more options menu. The four buttons are “APERTURE”, “VOLUME”, “LENGTH”, and “PRESSURE vs. TIME”, which are listed from right top corner downwards, respectively. The first three buttons will be called the appropriate subroutines, namely EXC\_APER\_FLOW, EXC\_VOLU\_FLOW, and EXC\_LENGTH\_FLOW, to execute. When “PRESSURE vs. TIME” button is clicked, a figure showing the fluid pressure varying as a function of time will be displayed in a separate window. More tests need to be performed on the functionality of this button.

**The basic literature**

1. Alehossein, H. and Korinets, A. (1997). *Numerical elastic analysis of a surface crack near Boussinesque load*, International Journal of Fracture 86: L39-L44, 1997.

2. Carter, B., Ingraffea, A., *Effects of casing and interface behavior on hydraulic fracture*, Computer Methods and Advances in Geomechanics, H. Siriwardane and M. Zaman, Eds., A. A. Balkema, Rotterdam, Vol. 2, 1561-1566, 1994.

3. Castell, M., Ingraffea, A. R., Irwin, L. *Fatigue crack growth in pavements*, 126, 4, 283-290, ASCE J. Transportation Engrg., 2000.

4. Geertsma, J., *Two-dimensional fracture-propagation models*, in Recent Advances in Hydraulic Fracturing. SPE Monograph, edited by Gidley, J.L., et al, 1989.

5. Geertsma, J., and Klerk, F., *A rapid method of predicting width and extent of hydraulically induced fractures*, Journal of Petroleum Technology, Vol. 21, No. 12, Dec. 1969, pp. 1571-1581.

6. Charsley, A.D., Martin, C.D., and McCreath, D.R. *Sleeve-fracturing limitations for measuring in situ stress in an anisotropic stress environment*, International Journal of Rock Mechanics & Mining Sciences 40 (2003) 127-136.

7. Fairbanks, C. D. and Andrus, R. Initial Geotechnical Investigation at Simpson Station Experimental Site, Pendleton, South Carolina.Department of Civil Engineering, Clemson University. Unpublished Report, August 2002.

8. Gidley, J.L., S.A. Holditch, and D.E. Nierode, and R.W. Veatch. 1989 Recent Advances in Hydraulic Fracturing. SPE Monograph Volume 12.452 p.

9. Grise, W. R. and Zargari, A. 1997. *Delamination and cracking failures in high-voltage stator winding coatings*, Proceedings of the 24th Electrical/Electronics Insulation Conference, 24th, 1997, 835-839.

10. Ingraffea, A.R., Numerical modeling of fracture propagation, Rock fracture mechanics, Edited by H.P. Rossmanith, published by Springer-Verlay, 1983, pp. 151-208.

11. Ingraffea, A.R., Theory of crack initiation and propagation in rock, Fracture mechanics of rock, 1987, by Academic Press Inc. (London) Ltd., pp. 71-110.

12. Ingraffea, A.R., *Case studies of simulation of fracture in concrete dams*, Fracture Mechanics Vol. 35, No. 1/2/3, pp. 553-564, 1990.

13. James, M., and Swenson, D., *FRANC2D/L: a crack propagation simulator for plane layered structures*, available from <http://www.mne.ksu.edu/~franc2d/>.

14. Lamkin, S. J., Wawrzynek, P. A. and Ingraffea, A.R., *Two - dimensional numerical simulation of interacting fractures in rock*, Fracture of Concrete and Rock: Recent Developments, S. P., Shah, S.E. Swartz, B. Barr, editors, Elsevier Science Publishing, New York, N.Y., 1989, pp. 121 - 131.

15. Lawn, B., Fracture of brittle solids, Second edition, 1993, Cambridge University Press, 1993, pp. 23-26.

16. Linsbauer, H. N., Ingraffea, A. R., Rossmanith, H. P., Wawrzynek, P. A., *Simulation of cracking in a large arch dam: part i*, Journal of Structural Engineering, Vol. 115, No. 7, July 1989, pp. 1599 - 1615.

17. Linsbauer, H. N., Ingraffea, A. R., Rossmanith, H. P., Wawrzynek, P. A., *Simulation of cracking in a large arch dam: part ii*, Journal of Structural Engineering, Vol. 115, No. 7, July, 1989, pp. 1616 - 1630.

18. Murdoch, L. C., 1991. *TheHydraulic fracturing of soil*, Ph.D. dissertation, University of Cincinnati.

19. Murdoch, L. C., 1993. *Hydraulic fracturing of soil during laboratory experiments*, Parts a-c. Geotechnique, v. 43, no. 2: 255-287.

20. Murdoch, L. C., 1994. *Induced Fractures*. United States Environmental Protection Agency. p. 39-55.

21. Murdoch, L. C., 1995. *Hydraulic and implulse fracturing for low permeability soils; in petroleum contaminated low permeability soil: hydrocarbon distribution processes, exposure pathways, and in situ remediation technologies*. Edited by Walden, T. American Petroleum Institute. p. C-1 - C-32, E-1 - E-34.

22. Murdoch, L.C., 1995. *Forms of hydraulic fractures created during a field test in overconsolidated glacial drift*. Quarterly Journal of Engineering Geology, 28, 23-35.

23. Murdoch, L.C. and Chen, J-L, (1997). *Effects of conductive fractures during in situ electroosmosis*, Journal of Hazardous Materials 55 (1997) 239-262.

24. Murdoch, L.C. and Slack, W. 2002. *Forms of hydraulic fractures in shallow fine-grained formations*. Journal of Geotechnical and Geoenvironmental Engineering, Vol. 128, No. 6, June 1, 2002, pp. 479-487.

25. Murdoch, L.C. 2002. *Analysis of an idealized shallow hydraulic fracture*. Journal of Geotechnical and Geoenvironmental Engineering, Vol. 128, No. 6, June 1, 2002, pp. 488-495.

26. Nilson, R.H., *An integral method for predicting hydraulic fracture propagation driven by gases or liquids,* International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 10, 191-211 (1986).

27. Nilson, R.H., and Griffiths, S.K., *Numerical analysis of hydraulically-driven fractures*, Computer Methods in Applied Mechanics and Engineering 36 (1983) 359-370.

28. Nilson, R.H., *Similarity solutions for wedge-shaped hydraulic fractures driven into a permeable medium by a constant inlet pressure*, International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 12, 477-495 (1988).

29. Nykanen, T. 1999. *Geometric dependency of fatigue strength in a transverse load-carrying cruciform joint with partially-penetrating K-welds*, Welding in the world, Vol. 43, No. 5, sept-oct 1999, 47-53.

30. Perkins, T. K., and Kern, L. R., *Width of hydraulic fracture*, Journal of Petroleum Technology, Vol. 13, Sept. 1961, pp. 937-949.

31. Pollard, D.D., and Holzhausen, G., On the mechanical interaction between a fluid-filled fracture and the earth’s surface, Techonophysics, 53 (1979) 27-57.

32. Richardson, J., 2003, *Forms of hydraulic fractures at shallow depth in piedmont soil*, Master’s thesis, Clemson University.

33. Rogers, C.A. and Hancock, G.J. (2001). *Fracture toughness of G550 sheet steels subjected to tension*, Journal of Constructional Steel Research 57 (2001) 71-89.

34. Sowers, G.F., and Richardson, T.L. 1982. Residual Soils of the Piedmont and Blue Ridge. Transportation Research Record No. 919, National Academy Press, Washington D.C., 10-20.

35. Srinivasan, M., *Fracture Analysis Code, A computer-aided teaching tool*, Master’s Thesis, Cornell University.

36. Tada, H., Paris, P.C. & Irwin, G.R. (1985). *The stress analysis of cracks handbook*, 2nd edn. St. Louis: Paris Productions.

37. Trego, A. and Cope, D. (2001). *Evaluation of damage tolerance analysis tools for lap joints*, AIAA Journal Vol.39, No. 12, December 2001.

38. Wang, H. F., and Anderson, M. P., 1982, *Introduction to groundwater modeling*, Academic Press, Inc.

39. Wawrzynek, P. and Ingraffea, A. (1996). FRANC2D: A two dimensional crack propagation simulator, Version 2.7 user’s guide. NASA Contractor Report 4572 (1994).

**The supplementary literature**

1. Ralph H. Abraham and Jerrold E. Marsden. Foundations of Mechanics. -Wesley, 1978.

2. V. I. Arnold. Mathematical Methods of Classical Mechanics. - Springer Verlag, 1980.

3. Lothar Nordheim. The Principles of Mechanics.Vol. 2. - Springer, 1972.

4. Jean-Marie Souriau. Structure des Systèmes Dynamiques. - Dunod Université, Paris, 1970. (English translation: Birkhäuser Boston, 1998.)

5. E. C. G. Sudarshan and N. Mukunda.Classical Dynamics: A Modern Perspective. - John Wiley & Sons, 1974.

6. V. I. Arnold, V. V. Kozlov and A. I. Neishtadt. Mathematical Aspects of Classical and Celestial Mechanics.Dynamical Systems III. - Springer Verlag, 1988.

7. Giovanni Gallavotti. The Elements of Mechanics. - Springer Verlag, 1983.

8. F. R. Gantmakher. Lectures on analytical mechanics. - Fizmatgiz, 1960. (English translation by G. Yankovsky. - Mir Publishing, 1970.)

9. Herbert Goldstein. Classical Mechanics.2nd edition. - Wesley, 1980.

10. Jorge V. José and Eugene J. Saletan.Classical Dynamics: A Contemporary Approach. - Cambridge University Press, 1998.

11. Cornelius Lanczos. The Variational Principles of Mechanics.4th edition.- University of Toronto Press, 1970; Dover Publications, 1982.

12. Пронина Р.Ф. Пособие по переводу английской научно-технической литературы. - М.: Высшая школа, 1973.

13. Циммерман М., Веденеева К. Русско-английский научно-технический словарь переводчика. - М.: Наука, 1991.

14. Арушанян О.Б. Русско-английский словарь по прикладной математике и механике. (Russian-English dictionary of applied mathematics and mechanics). – М.: МГУ, 2012.

15. Пумпянский А.Л. Чтение и перевод английской научно-технической литературы: Лексика, грамматика, фонетика, упражнения /Худ.обл. М. В. Драко.— М.: ООО «Попурри», 1997.— 608 с.

16. L. D. Landau and E. M. Lifshitz. Mechanics.3rd edition.Course of Theoretical Physics.Vol. 1. - Pergamon Press, 1976.

17. Jerrold E. Marsden and Tudor S. Ratiu.Introduction to Mechanics and Symmetry. - Springer Verlag, 1994.

18. Florian Scheck. Mechanics, From Newton's Laws to Deterministic Chaos.2nd edition. - Springer-Verlag, 1994.

19. E. T. Whittaker. A Treatise on Analytical Dynamics.- Cambridge University Press, 1937.

20. Walter Thirring. A Course in Mathematical Physics 1: Classical Dynamical Systems, translated by Evans M. Harell, - Springer-Verlag, 1978.

21. Англо-русский словарь математических терминов: около 20000 терминов / под ред. П. С. Александрова. - Изд.3-е, стер. - М.: Мир, 2001.

22. Голденков М. Современный активный English. ЧеРо. – М: Физматлит, 2002.-448с.

**The list of periodicals**

1. Английский язык: Прил. к газете "1 сентября".

2. Иностранные языки в высшей школе.

**The source list on electronic media**

1. КутателадзеС.С. Russian-English in writing. Изд. института математики. – Новосибирск, 2000.

2. Игнатьев-Каллэхэм Л. Русско-английский политехнический словарь. - М.: Уайли, 1995.

3. A.J. Lohwater's Russian-English dictionary of the mathematical sciences / R.P. Boas (ed.). AmericanMathematicalSociety, 1990.

4. Сосинский А. Б. Как написать математическую статью по-английски. — М: Изд-во «Факториал Пресс», 2000. — 112 с.

**The Internet sources**

1. http://ru.wikipedia.org

2. <http://lingualeo.com/>

**Lecture 4**

**Lecture topic: Applications. Comparison of Numerical and Analytical Solution**

**The plan**

**1. Applications**

1.1. Verification

1.1.1. Static Solution

1.1.1.1. Model Setting:

1.1.2. Simulation Results

1.2. Comparison of Numerical and Analytical Solution

1.2.1. Pressure

1.2.2. Aperture

1.2.3. Length

**2. Shallow Soil Problems**

2.1. General Case

2.1.1. Model setting

2.1.2. Analysis

2.1.3. Results

2.1.3.1. Pressure

2.1.3.2. Form

2.1.3.3. Aperture

**1. Applications**

The modified code is applied to simulate hydraulic fractures under different site conditions. One model is setup to numerically simulate the condition of Tada analytical solution that was discussed previously in chapter 1. The results from numerical simulation and Tada’s analytical solution is compared together to verify the validity of this modified code.

Other models are simulated to explore the simulation results of the hydraulic fracturing in soil. While numerical analysis is performed in the office, fieldwork of producing induced hydraulic fracturing has also been performed [Richardson, 2003]. So, another approach to examine the numerical results will be to compare these numerical results got from HFRANC2D to some field measurements of hydraulic fractures that have been created in the field.

HFRANC2D is used to simulate a simple model with uniform fluid pressure in an infinite elastic, homogeneous media. The results of driving pressure, length, and aperture are then compared to analytical results.

HFRANC2D is also used to simulate the hydraulic fractures in shallow soil under different site conditions. Results from these simulations are compared to the field experiments conducted by Jim Richardson at Clemson University [Richardson, 2003].

**1.1. *Verification***

HFRANC2D is verified by simulating known one-dimensional fracture problems after it is coupled with fluid flow. The simulation results are then compared to the analytical solutions.

**1.1.1. *Static Solution***

A circular horizontal hydraulic fracture cut through elastic media is simulated for the details through the propagation. Uniform fluid driving pressure is applied to the fracture face. The fluid driving pressure, fracture length, and aperture at the center of the fracture is recorded during the propagation.

**1.1.1.1. *Model Setting***

A hydraulic fracture model is built to simulate circular crack under driving pressure. The model is formulated in axisymmetric coordinates(r, z). The original coordinates are set at the center of the left boundary of the domain (figure 4.1-1). The z increases upwards and decreases downwards from the original point. R increases from the left boundary of the domain to the right.

The model has 20 meters by 20 meters rectangular domain (figure 4.1-1). The model is fixed on the top and bottoms boundaries, so that no movement will happen on the top and bottom boundaries in r direction. Moreover, the boundaries on the two sides of the r directions have also been fixed, so that no movement will happen along y direction in these two boundaries.

The model is assumed to be an elastic, homogeneous material with a Young’s modulus of typical soil. Previous lab experiments have shown that the plastic area is limited in a small region around the crack tip, so that linear elastic fracture mechanics can still be used [Murdoch, 1993b,c]. In this model, the material in the domain is assumed to be linear-elastic.

A crack is initiated from the original point with a length of 1 meter. It is parallel to the top and bottom boundaries of the model domain. With the crack opening at the original point, the crack tip is 1 meter away from it along the r direction.

The crack is loaded by a 10 kPa uniformly distributed driving pressure all over the crack face, both in upper face and down face. No other loadings are applied in this model. The constant pressure in the crack can be used to as a rough approximation of the pressure caused by fluid flow [Carter et al, 1994].

The maximum length of the crack created during this simulation is less than approximately 0.1 of the distance to the nearest boundary. As a result, the effects of the boundaries are expected to be negligible, so the model is expected to behave as if the crack was embedded in an infinite medium.

A mechanical analysis is first performed, and the deformation of the model and fracture vicinity can be checked. This is done to check if the loading condition is applied correctly.

For example, if the fluid pressure has not been applied to the all the length of the fracture, in most case the tip and its vicinity, then the deformation of the fracture will be irregular. Then select the distance that how long the fracture needs to be propagated. Decide fracture length by selecting propagation steps. Fracture propagation is modeled in a series of steady-state propagation steps. Each step will propagate the fracture in a designed length that is ten percent of the current fracture length. Select the designed steps to propagate the fracture to a wanted length. Fifteen steps are selected for this study.HFRANC2DHFRANC2D

The numerical solution is then compared to the Tada analytical solution, which is discussed in section 1.3.4. The Tada analytical solution has been derived to give driving pressure, length, and aperture as a function of time. These relations are described in equations (1-23), (1-25), and (1-26).

The same values are used in the Tada solution (except the Tada solution assumes the extent of the domain is infinite) to get the pressure, length, and aperture as a function of injecting time. These results will be plotted together with the numerical simulation results.

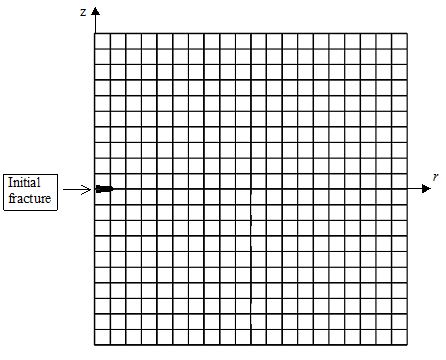
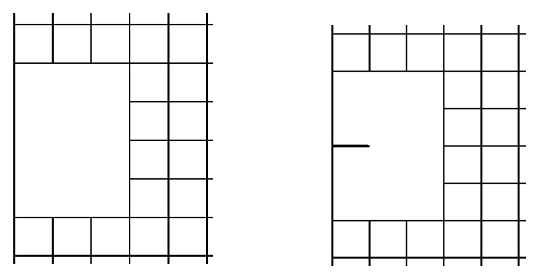


Figure 4.1-1. Model geometry for verification. Initial fracture with a length of 1 m is created at the left edge



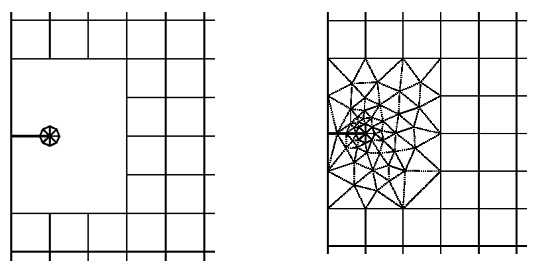
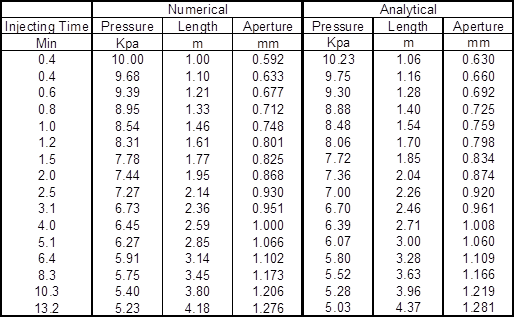


Figure 4.1-2. Fracture initiation and remesh around fracture. These works are completed by HFRANC2D automatically, after fracture location is known. The fracture will be created in the order from top to bottom, left to right

**1.1.2. *Simulation Results***

A two-dimensional hydraulic fracture propagation has been performed to examine the fluid pressure, crack length, and crack mouth opening as a function of pumping time (Table 2). The results from analytical solution is calculated by using equations (1-23), (1-25) and (1-26). The length is counted from center of the fracture to the tip. Aperture is measured at the center of the fracture.

**Table 2**. Numerical simulation results of a static problem compared to analytical solution [Tada et al, 1985]



**1.2. *Comparison of Numerical and Analytical Solution***

**1.2.1. *Pressure***

The driving pressure of the fracture is plotted in an Cartesian coordinate system, with the  axis representing injection time, and  axis representing driving pressure in the fracture (figure 4.1-3). The numerical and analytical solutions are similar. The relative error between the numerical and analytical results is slightly negative at the onset of propagation, but it increases and oscillates between 0.01 and 0.04 (figure 4.1-4). The coefficient of determination is 0.9897 (table3).

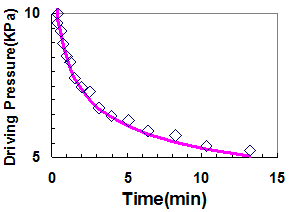


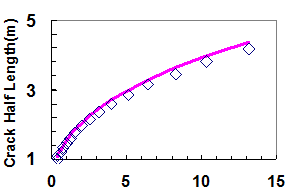
**1.2.2. *Aperture***

The aperture of the fracture at the center of it is plotted in the same way that the driving pressure has been plotted. However, the  axis represents the aperture at this time (figure 4.1-3). The aperture values simulated from numerical model and analytical solutions are almost identical. The relative error for aperture is between –0.06 and 0.01 (figure 4.1-4). At the beginning stage of the propagation, the error is relative high, around –0.06 (-0.0594). With the fracture propagating, it diminishes, and oscillates around +0.01 and –0.01. The coefficient of determination is 0.9950 (table 3).

**1.2.3. *Length***

The fracture length is plotted the same way that driving pressure and aperture have been plotted as a function of time. The fracture length increases with the fracture extends (figure 4.1-3). However, the increasing rate diminishes with the fracture extends. The relative error of the length is between –0.06 and –0.04 (figure 4.1-4). The coefficient of determination for length is 0.9853 (table 3).





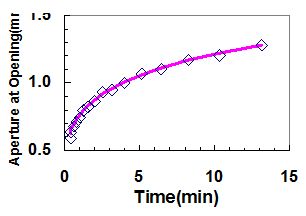


Figure 4.1-3.Comparison of numerical simulation and analytical solution results. The solid lines in the figure are the numerical simulation results, the box series are the analytical simulation results [Tada et al, 1985]

**Table 3.** SSE, SSM, and coefficient of determination for pressure, length, and aperture



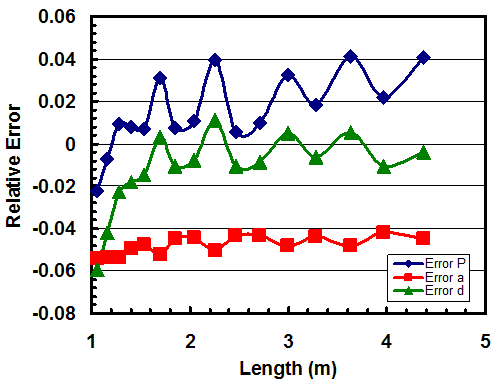


Figure 4.1-4. Relative errors between numerical and analytical simulation results

**2. Shallow Soil Problems**

Hydraulic fracture propagation in shallow Piedmont soil is simulated with HFRANC2D. The driving pressure, fracture length, and fracture aperture at the center are logged during the propagation simulation. Fracture trajectory starting from well bore where the fracture is initiated to the leading edge, is retrieved from finite element grid node coordinates information, and the trajectory is used to compare to the forms of fractures observed in the subsurface.

Forms of fractures created in the field are analyzed by first simulating relatively simple conditions where the *in situ* stress is a result of only the body forces from the unit weight of the soil. Moreover the soil is assumed to be uniform. Other, more complicated conditions, will be analyzed in the following pages by building on the results of the idealized case.

Two models were established to study the effects of site conditions in the Piedmont soils. One tried to simulate hydraulic fracture propagation created in shallow depth, where high lateral stress exists. Another model simulates the hydraulic fracture propagation with a second kind of soil material below the soil layer where the fracture is created. The underlying soil has a softer stiffness than the soil above it.

The results show that the fracture initiated in a soil layer with high lateral stress will tend to extend horizontally. The fracture will extend horizontally away from the borehole, then it will start to curve slightly toward the ground. The fracture created in the soil with an under lying soft soil layer will tend to curve downwards toward this soft soil layer, after it is initiated horizontally in the well bore. Before it intercepts the soft layer, the fracture will curve back upwards toward the ground surface. One explanation for the behavior of the fractures under this condition is that the fracture tends to curve toward a softer material near it. When the fracture is short in the beginning stage of propagation, the fracture prefers to curve toward the softer layer that is closer to it than the air above ground surface. After the fracture gets longer, the fracture can start to feel the affects of the air above ground surface, and tends to curve towards it [from personal talks with Murdoch, 2003].

Many hydraulic fractures have been created in shallow silty clay glacial drift at a depth of 2 meters [Murdoch, 1995]. These fractures have displayed flat-lying to gently-dipping linear trajectory in the excavation trenches. However, hydraulic fractures created in the Piedmont soil have displayed different features [Richardson, 2003]. Generally, these fractures curve downwards instead of horizontally, after they are initialized. The fractures then propagate toward ground surface. Explanations of these field observations are attempted by using numerical simulations.

**2.1. *General Case***

A model is built to simulate hydraulic fracture propagation under soil self-weight and fluid pressure. Parameters in the simulation were decided by lab and field experiments. The simulation results were then compared to field measurements.

**2.1.1. *Model setting***

A hydraulic fracture created at shallow depth is assumed to start as a circular cavity with an elliptical cross-section. This cavity is axisymmetric. In every cross-section of the fracture, the two sides of the area are identical, thus, only one side of the fracture will need to be considered.

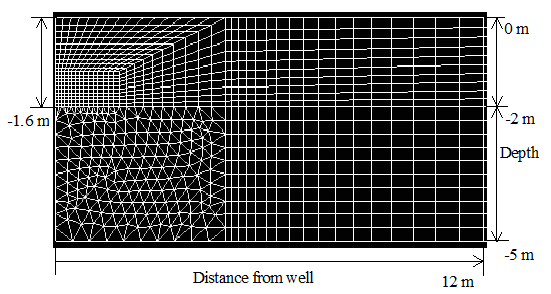


Figure 4.2-1.Model geometry of shallow soil problems. The fracture is initiated at the depth of 1.6 m, left edge of the domain

A rectangular model is created, and represents the soil mass. It has a size of 12 m in radial direction, and 5 m in depth (figure 4.2-1). The domain is axisymmetric, so it starts from the origin of radial axis, and increase. The top of the domain represents ground surface.

The finite element grid mesh is denser mesh near the fracture tip, and coarser away from the fracture. The computation speed and precision can be well balanced by introducing a fine mesh around fracture, while using coarse mesh in the rest of the model domain.

The hydraulic fracture is initiated as a thin edge crack in the depth of 1.6 m, in the center of the domain. The initial length of the fracture is 0.15 m, which is normal in the field [Richardson, 2003].

The material in the model domain is assumed to be homogeneous and isotropic. The Young’s modulus in this model is 34.5 MPa, which is about the average of the value of lab soil tests [Fairbaanks and Andrus, 2002]. The Poission’s ratio is using 0.4. The fracture toughness of the soil is set to be 0.3 MPa , which is close to lab experiments [Murdoch, 1993b].

Set the soil self-weight by choosing “BODY FORCE” in the HFRANC2D’s loading menu. A gravity accelaration of –9.8 is used in the model, since the positive direction of vertical axis is upward. Uniform fluid pressure is applied in the fracture. This fluid pressure will create the consequent stress intensity *KI* at the propagation of fracture.

**2.1.2. *Analysis***

This model was run by assuming that the fluid pressure is uniformly distributed over the length of the fracture. The fluid pressure changes with time to satisfy the coupling of flow and deformation, and propagation criteria, but fluid pressure is uniform within the fracture. As a result, the flow of fluid within the fracture is not considered in this analysis.

The analysis was run in automatic propagation mode where the fracture is increased by 0.1 of the total length during each time step.

**2.1.3. *Results***

**2.1.3.1. *Pressure***

The pressure data from the output file “PressurevsTime.txt” is plotted as a function of time. The pressure decreases with the time in both the simulation and the field observations (figure 4.2-2). The predicted values are similar to the observed pressures and the general trends are the same. The observed pressures increase and decrease more than the simulated results. Samples were taken in the field by opening a valve, which causes the pressure to drop momentarily. Fluid properties also change as a function of time and this causes fluctuations in the injection pressure. Deviations between the predicted and observed values are probably a result of these variations, which are not included in the simulation.

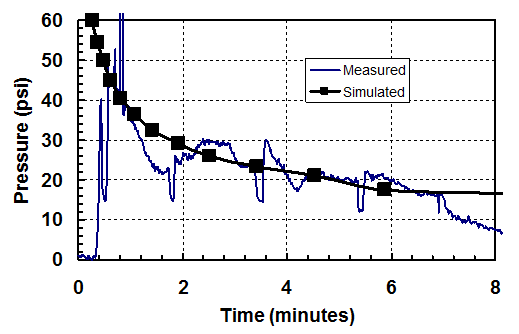


Figure 4.2-2. Pressure data from simulation and field measurements

**2.1.3.2. *Form***

The form of the fracture in the field was determined by making radial cross-sections from the structural contour map produced by Richardson [2003]. The cross-sections cut through the center of the fracture. The center differs from the location of the borehole, but the center was used as the origin of the cross-sections to be consistent with the axial symmetry used in the model.

The fracture path in the simulation shows a flat-lying feature near where the initial slot was created. Then it curves upward, to form a gently dipping feature. Note that the fracture departs somewhat from the axial symmetry assumed by the model because the traces along the four cross-sections are somewhat different. The fracture has extended to a length of nearly 3 m when the propagation stops.

The four observed sections, and it occurs within the envelop created by the four sections. 4Three of the observed sections (1, 2, and 4) dip more steeply than the simulated section, but one observed section dips more gently (figure 4.2-3).

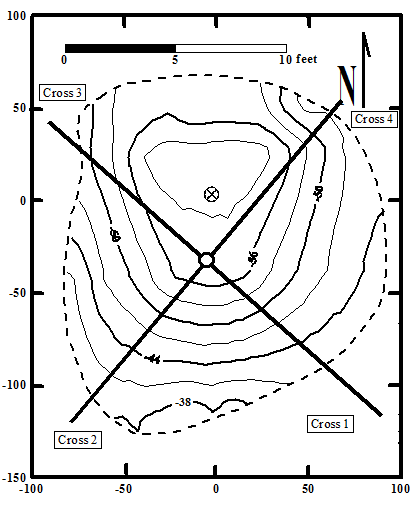


Figure 4.2-3.Structural contour map of Fracture H. The contour interval is 3 inches [Richardson, 2003, page 115]. The cross 1-4 are cross sections used for plot fracture path curves

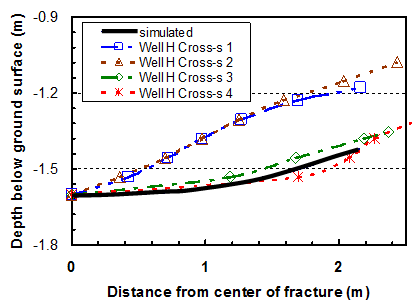


Figure 4.2-4. Fracture paths from simulation and field measurements

**2.1.3.3. *Aperture***

The fracture aperture predicted by the model is plotted as a function of length, and these results are compared to profiles of the uplift of the ground surface overlying the fracture (Figure 4.2-6). The profiles of uplift were obtained along 4 sections that extend through the point of maximum uplift in Richardson (2003) (figure 4.2-5). Field measurements of uplift were used as a surrogate for aperture because aperture could not be measured in the field when the fracture was inflated with fluid pressure.

The model predicts that the fracture aperture is a maximum of approximately 0.028 m, and similar values of uplift were observed in the field. The aperture diminishes with distance from the well and tapers to zero approximately at a radial distance of approximately 2.2 m. The uplift profiles also diminish with distance, but the dome predicted by uplift is broader than the aperture predicted by the model.

Previous work by Murdoch and Slack (2002) concluded that uplift values were similar to the aperture of underlying fractures in some cases, but this fracture may be an exception. Sand is distributed over a region that is considerably smaller than the region where uplift was observed, based on in Richardson (2003) (figure 4.2-5). As a result, it seems feasible that the extent of the fracture was actually smaller than the uplifted region. The region of the fracture filled with sand forms a crudely equant region whose radial extent ranges from 2.0 m to 2.6 m from an axis taken at the point of maximum uplift. The range of the radial extent of sand is shown as the hatched box in (Figure 4.2-6). The model predicts that the radial extent of the fracture should be 2.2 m, which is approximately the size of the region containing sand. As a result, the radial extent of the fracture predicted by the model is consistent with the radial extent of the sand observed in the field experiments.

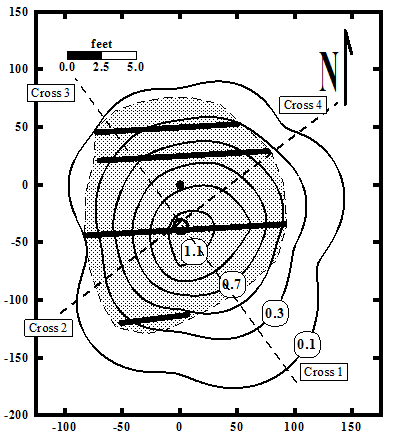


Figure 4.2-5. Map of uplift contours of Fracture H. The region containing sand (shaded), trace of trench faces (heavy lines), and injection casing (filled cicle) are also indicated [Richardson, 2003, Page 121]. Cross 1-4 are used to plot fracture aperture cross sections and compare to numerical simulation results

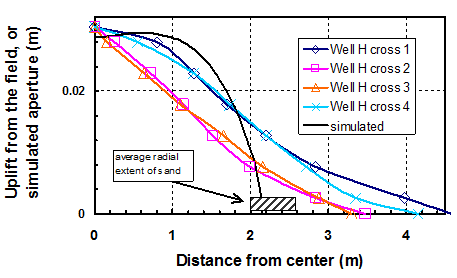


Figure 4.2-6. Aperture profile and uplift of ground surface. Aperture is calculated form numerical simulation. Uplift of ground surface is from field measurement. The rectangular box in the bottom center of this figure indicates how far the sand can reach in the field experiments

**The basic literature**

1. Alehossein, H. and Korinets, A. (1997). *Numerical elastic analysis of a surface crack near Boussinesque load*, International Journal of Fracture 86: L39-L44, 1997.

2. Carter, B., Ingraffea, A., *Effects of casing and interface behavior on hydraulic fracture*, Computer Methods and Advances in Geomechanics, H. Siriwardane and M. Zaman, Eds., A. A. Balkema, Rotterdam, Vol. 2, 1561-1566, 1994.

3. Castell, M., Ingraffea, A. R., Irwin, L. *Fatigue crack growth in pavements*, 126, 4, 283-290, ASCE J. Transportation Engrg., 2000.

4. Geertsma, J., *Two-dimensional fracture-propagation models*, in Recent Advances in Hydraulic Fracturing. SPE Monograph, edited by Gidley, J.L., et al, 1989.

5. Geertsma, J., and Klerk, F., *A rapid method of predicting width and extent of hydraulically induced fractures*, Journal of Petroleum Technology, Vol. 21, No. 12, Dec. 1969, pp. 1571-1581.

6. Charsley, A.D., Martin, C.D., and McCreath, D.R. *Sleeve-fracturing limitations for measuring in situ stress in an anisotropic stress environment*, International Journal of Rock Mechanics & Mining Sciences 40 (2003) 127-136.

7. Fairbanks, C. D. and Andrus, R. Initial Geotechnical Investigation at Simpson Station Experimental Site, Pendleton, South Carolina.Department of Civil Engineering, Clemson University. Unpublished Report, August 2002.

8. Gidley, J.L., S.A. Holditch, and D.E. Nierode, and R.W. Veatch. 1989 Recent Advances in Hydraulic Fracturing. SPE Monograph Volume 12.452 p.

9. Grise, W. R. and Zargari, A. 1997. *Delamination and cracking failures in high-voltage stator winding coatings*, Proceedings of the 24th Electrical/Electronics Insulation Conference, 24th, 1997, 835-839.

10. Ingraffea, A.R., Numerical modeling of fracture propagation, Rock fracture mechanics, Edited by H.P. Rossmanith, published by Springer-Verlay, 1983, pp. 151-208.

11. Ingraffea, A.R., Theory of crack initiation and propagation in rock, Fracture mechanics of rock, 1987, by Academic Press Inc. (London) Ltd., pp. 71-110.

12. Ingraffea, A.R., *Case studies of simulation of fracture in concrete dams*, Fracture Mechanics Vol. 35, No. 1/2/3, pp. 553-564, 1990.

13. James, M., and Swenson, D., *FRANC2D/L: a crack propagation simulator for plane layered structures*, available from <http://www.mne.ksu.edu/~franc2d/>.

14. Lamkin, S. J., Wawrzynek, P. A. and Ingraffea, A.R., *Two - dimensional numerical simulation of interacting fractures in rock*, Fracture of Concrete and Rock: Recent Developments, S. P., Shah, S.E. Swartz, B. Barr, editors, Elsevier Science Publishing, New York, N.Y., 1989, pp. 121 - 131.

15. Lawn, B., Fracture of brittle solids, Second edition, 1993, Cambridge University Press, 1993, pp. 23-26.

16. Linsbauer, H. N., Ingraffea, A. R., Rossmanith, H. P., Wawrzynek, P. A., *Simulation of cracking in a large arch dam: part i*, Journal of Structural Engineering, Vol. 115, No. 7, July 1989, pp. 1599 - 1615.

17. Linsbauer, H. N., Ingraffea, A. R., Rossmanith, H. P., Wawrzynek, P. A., *Simulation of cracking in a large arch dam: part ii*, Journal of Structural Engineering, Vol. 115, No. 7, July, 1989, pp. 1616 - 1630.

18. Murdoch, L. C., 1991. *TheHydraulic fracturing of soil*, Ph.D. dissertation, University of Cincinnati.

19. Murdoch, L. C., 1993. *Hydraulic fracturing of soil during laboratory experiments*, Parts a-c. Geotechnique, v. 43, no. 2: 255-287.

20. Murdoch, L. C., 1994. *Induced Fractures*. United States Environmental Protection Agency. p. 39-55.

21. Murdoch, L. C., 1995. *Hydraulic and implulse fracturing for low permeability soils; in petroleum contaminated low permeability soil: hydrocarbon distribution processes, exposure pathways, and in situ remediation technologies*. Edited by Walden, T. American Petroleum Institute. p. C-1 - C-32, E-1 - E-34.

22. Murdoch, L.C., 1995. *Forms of hydraulic fractures created during a field test in overconsolidated glacial drift*. Quarterly Journal of Engineering Geology, 28, 23-35.

23. Murdoch, L.C. and Chen, J-L, (1997). *Effects of conductive fractures during in situ electroosmosis*, Journal of Hazardous Materials 55 (1997) 239-262.

24. Murdoch, L.C. and Slack, W. 2002. *Forms of hydraulic fractures in shallow fine-grained formations*. Journal of Geotechnical and Geoenvironmental Engineering, Vol. 128, No. 6, June 1, 2002, pp. 479-487.

25. Murdoch, L.C. 2002. *Analysis of an idealized shallow hydraulic fracture*. Journal of Geotechnical and Geoenvironmental Engineering, Vol. 128, No. 6, June 1, 2002, pp. 488-495.

26. Nilson, R.H., *An integral method for predicting hydraulic fracture propagation driven by gases or liquids,* International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 10, 191-211 (1986).

27. Nilson, R.H., and Griffiths, S.K., *Numerical analysis of hydraulically-driven fractures*, Computer Methods in Applied Mechanics and Engineering 36 (1983) 359-370.

28. Nilson, R.H., *Similarity solutions for wedge-shaped hydraulic fractures driven into a permeable medium by a constant inlet pressure*, International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 12, 477-495 (1988).

29. Nykanen, T. 1999. *Geometric dependency of fatigue strength in a transverse load-carrying cruciform joint with partially-penetrating K-welds*, Welding in the world, Vol. 43, No. 5, sept-oct 1999, 47-53.

30. Perkins, T. K., and Kern, L. R., *Width of hydraulic fracture*, Journal of Petroleum Technology, Vol. 13, Sept. 1961, pp. 937-949.

31. Pollard, D.D., and Holzhausen, G., On the mechanical interaction between a fluid-filled fracture and the earth’s surface, Techonophysics, 53 (1979) 27-57.

32. Richardson, J., 2003, *Forms of hydraulic fractures at shallow depth in piedmont soil*, Master’s thesis, Clemson University.

33. Rogers, C.A. and Hancock, G.J. (2001). *Fracture toughness of G550 sheet steels subjected to tension*, Journal of Constructional Steel Research 57 (2001) 71-89.

34. Sowers, G.F., and Richardson, T.L. 1982. Residual Soils of the Piedmont and Blue Ridge. Transportation Research Record No. 919, National Academy Press, Washington D.C., 10-20.

35. Srinivasan, M., *Fracture Analysis Code, A computer-aided teaching tool*, Master’s Thesis, Cornell University.

36. Tada, H., Paris, P.C. & Irwin, G.R. (1985). *The stress analysis of cracks handbook*, 2nd edn. St. Louis: Paris Productions.

37. Trego, A. and Cope, D. (2001). *Evaluation of damage tolerance analysis tools for lap joints*, AIAA Journal Vol.39, No. 12, December 2001.

38. Wang, H. F., and Anderson, M. P., 1982, *Introduction to groundwater modeling*, Academic Press, Inc.

39. Wawrzynek, P. and Ingraffea, A. (1996). FRANC2D: A two dimensional crack propagation simulator, Version 2.7 user’s guide. NASA Contractor Report 4572 (1994).

**The supplementary literature**

1. Ralph H. Abraham and Jerrold E. Marsden. Foundations of Mechanics. -Wesley, 1978.

2. V. I. Arnold. Mathematical Methods of Classical Mechanics. - Springer Verlag, 1980.

3. Lothar Nordheim. The Principles of Mechanics.Vol. 2. - Springer, 1972.

4. Jean-Marie Souriau. Structure des Systèmes Dynamiques. - Dunod Université, Paris, 1970. (English translation: Birkhäuser Boston, 1998.)

5. E. C. G. Sudarshan and N. Mukunda.Classical Dynamics: A Modern Perspective. - John Wiley & Sons, 1974.

6. V. I. Arnold, V. V. Kozlov and A. I. Neishtadt. Mathematical Aspects of Classical and Celestial Mechanics.Dynamical Systems III. - Springer Verlag, 1988.

7. Giovanni Gallavotti. The Elements of Mechanics. - Springer Verlag, 1983.

8. F. R. Gantmakher. Lectures on analytical mechanics. - Fizmatgiz, 1960. (English translation by G. Yankovsky. - Mir Publishing, 1970.)

9. Herbert Goldstein. Classical Mechanics.2nd edition. - Wesley, 1980.

10. Jorge V. José and Eugene J. Saletan.Classical Dynamics: A Contemporary Approach. - Cambridge University Press, 1998.

11. Cornelius Lanczos. The Variational Principles of Mechanics.4th edition.- University of Toronto Press, 1970; Dover Publications, 1982.

12. Пронина Р.Ф. Пособие по переводу английской научно-технической литературы. - М.: Высшая школа, 1973.

13. Циммерман М., Веденеева К. Русско-английский научно-технический словарь переводчика. - М.: Наука, 1991.

14. Арушанян О.Б. Русско-английский словарь по прикладной математике и механике. (Russian-English dictionary of applied mathematics and mechanics). – М.: МГУ, 2012.

15. Пумпянский А.Л. Чтение и перевод английской научно-технической литературы: Лексика, грамматика, фонетика, упражнения /Худ.обл. М. В. Драко.— М.: ООО «Попурри», 1997.— 608 с.

16. L. D. Landau and E. M. Lifshitz. Mechanics.3rd edition.Course of Theoretical Physics.Vol. 1. - Pergamon Press, 1976.

17. Jerrold E. Marsden and Tudor S. Ratiu.Introduction to Mechanics and Symmetry. - Springer Verlag, 1994.

18. Florian Scheck. Mechanics, From Newton's Laws to Deterministic Chaos.2nd edition. - Springer-Verlag, 1994.

19. E. T. Whittaker. A Treatise on Analytical Dynamics.- Cambridge University Press, 1937.

20. Walter Thirring. A Course in Mathematical Physics 1: Classical Dynamical Systems, translated by Evans M. Harell, - Springer-Verlag, 1978.

21. Англо-русский словарь математических терминов: около 20000 терминов / под ред. П. С. Александрова. - Изд.3-е, стер. - М.: Мир, 2001.

22. Голденков М. Современный активный English. ЧеРо. – М: Физматлит, 2002.-448с.

**The list of periodicals**

1. Английский язык: Прил. к газете "1 сентября".

2. Иностранные языки в высшей школе.

**The source list on electronic media**

1. КутателадзеС.С. Russian-English in writing. Изд. института математики. – Новосибирск, 2000.

2. Игнатьев-Каллэхэм Л. Русско-английский политехнический словарь. - М.: Уайли, 1995.

3. A.J. Lohwater's Russian-English dictionary of the mathematical sciences / R.P. Boas (ed.). AmericanMathematicalSociety, 1990.

4. Сосинский А. Б. Как написать математическую статью по-английски. — М: Изд-во «Факториал Пресс», 2000. — 112 с.

**The Internet sources**

1. http://ru.wikipedia.org

2. <http://lingualeo.com/>

**Lecture 5**

**Lecture topic: Simulation of Hydraulic Fracture with Underlying Soft Material and within High Residual Load Region**

**The plan**

**1. Simulation of Hydraulic Fracture with Underlying Soft Material**

1.1. Conceptual Model

1.2. Young’s Modulus

1.3. Results

1.4. Discussion

**2. Simulation of Hydraulic Fracture within High Residual Load Region**

2.1. Conceptual Model

2.2. Analysis

2.3. Results

2.4. Discussion

2.5. Conclusions

2.6. Future work

Appendix A. Source Code

Appendix B. Residual Load Input File Format

**1. Simulation of Hydraulic Fracture with Underlying Soft Material**

A hydraulic fracture model is established to simulate details of forms of hydraulic fracture propagation when softer material is underlying the layer where fracture is created. The trajectory of the fracture, the uplift of the ground surface and the aperture of the fracture are plotted as functions of the distance from the borehole.

**1.1. *Conceptual Model***

The model is similar to the model in general case, except the domain is divided into two layers of different elastic modulus (figure 4.2-1). The upper layer coincides with the B horizon of the soil and is assumed to be 2 m thick, based on field observation. The elastic modulus of the upper layer  is assumed to be equal to or greater than the modulus of the lower layer .

**1.2. *Young’s Modulus***

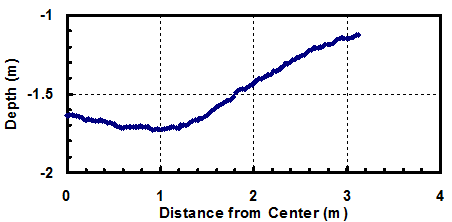
Young’s modulus of elasticity for soil material at the site was determined as the ratio of applied normal stress to normal strain during triaxial tests [Fairbanks and Andrus, 2002]. Lab experiments showed that the Young’s modulus of the Piedmont soil increases from 3500 psi to 7000 psi, when confining stress increases [Fairbanks and Andrus, 2002]. The modulus of the top layer  is set to 34.5 MPa (5000 psi), because this the approximate value for confining stress typical of the field experiments did. The values of  are changing slightly for different site conditions, so several values of  were used. 13.8 MPa (2000 psi), 20.7 MPa (3000 psi), and 27.6 MPa (4000 psi) are used as the Young’s modulus for the underlying layer. A model with uniform elastic modulus of 34.5 MPa is used as a benchmark. Thus, models with 13.8 MPa, 20.7 MPa, 27.6 MPa, and 34.5 MPa under-lying material are named model 2, model 3, model 4, and model 4, respectively.

**1.3. *Results***

The fracture length, fluid pressure, and aperture of the fracture are recorded during the propagation of the hydraulic fracture as a function of time. The fracture propagation trajectory, and aperture profile are recorded after the propagation is completed.

Output coordinates of the fracture face nodes from the HFRANC2D simulation code demonstrate the path of the hydraulic fracture through the soil material in its cross-section. The hydraulic fracture propagation trajectory is plotted with the radial extension of the fracture, fracture length (figure 4.2-7). Different trajectories caused by different Young’s modulus of underlying material is plotted together to get a comparison.

Forms of the fractures are nearly identical during the first meter of growth, but then the fractures underlain by softer material tend to curve downward, whereas the one in uniform material tends to curve upward (figure 4.2-7). This amount that the fracture curves downward increases with the contrast in moduli between the layers. I expect that the downward propagation would also increase as the distance between the initial crack and the underlying soft layer was diminished, although simulation. The dip angle of all of the trajectories is about 20°, but this dip occurs farther from the borehole with fractures that have interacted with an underlying layer than it does with fractures in a uniform material.



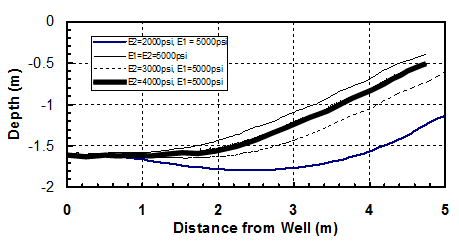


Figure 4.2-7. Fracture path from field observation (top) and

simulation (bottom)

**1.4. *Discussion***

In a uniformly elastic soil material, a hydraulic fracture tends to propagate away from the initial horizontal slot. It first shows a flat path, then will slightly curve upwards toward the ground surface (figure 4.2-7).

Numerical simulation has revealed the details of hydraulic fracture in a material with under-lying soft materials. The fracture forms observed in the field displayed that some fractures created in the Piedmont soil tend to curve downward, after they are initiated horizontally [Richardson, 2003]. They will propagate away from the borehole, and propagate toward ground surface with a dip angle of about 20 degrees [Richardson, 2003]. These features are consistent with the numerical simulation results showed here.

The pressure log recorded during the fracturing of one well in Pendleton, SC, has showed that the pressure is at its maximum value when the fracture starts to propagate [Richardson, 2003]. Then the pressure decreases when the fracturing continues. The numerical simulation of the fluid pressure log behaves in the same style, although it is a smooth curve compared to field observations.

**2. Simulation of Hydraulic Fracture within High Residual Load Region**

A model is built to examine the effect of residual load in shallow soil on the forms of fractures. The fracture is considered to be two-dimensional and axisymmetric. The center of the fracture is also the well bore where the fracture is initiated. Residual compression exists in the shallow soil layer where the fracture is initiated. Trajectory of the fracture maybe obtained from screen when the model is running in HFRANC2D.

**2.1. *Conceptual Model***

Residual horizontal compression is applied in this model all over its domain. The amount of the residual stress applied in the model is varied from one order of magnitude less than the fluid pressure applied to the fracture, to one order of magnitude more than the fluid pressure. This creates linear increment of vertical and horizontal stress in the soil.

For the convenience of comparison, the general case model, and the results from this analysis are listed together with the model with residual loadings. Both the model with high residual loading and model without residual loading are applied with body force induced by gravity. Young’s modulus is 34.5 KPa, as it was in the model with different materials discussed before. In this model, only one uniform elastic material is considered and used in simulation. The fracture toughness is 0.05 MPa ,

**2.2. *Analysis***

The soil in the Piedmont region where some hydraulic fractures are made is residuum evolved from *in situ* chemical weathering. Researchin this region showed that the high lateral stress may exist in residual soil that was inherited from the bedrock [Sowers and Richardson, 1982]. Wetting and drying of the soil exposed to the air is one mechanism that develops high lateral compression stresses. To examine the ratio of the horizontal stress to the vertical stress, also known as , *in situ* stress tests were performed in a test facility where hydraulic fractures was made, in Pendleton, SC. These *in situ* stress tests were conducted by Shaun Malin for his Master’s thesis research. His preliminary tests show that  ranges from 3 to 10.In addition, a commercial company, Soil Materials Engineers Inc, has conducted flat blade dilatometer tests at the site [Richardson, 2003]. These tests showed that  ranges from 3 to 3.5 in a depth range of 3 to 5 ft below ground surface, where the hydraulic fractures are created. Both Malin’s data and the commercial tests suggest that horizontal compression stress is several times more than the vertical stress in the shallow depth in the Piedmont soil.

The ambient state of stress in HFRANC2D is determined as the sum of gravitational and residual components. The models used to simulate field conditions consider an elastic half-space where the horizontal strain is zero, so the relation between vertical and horizontal stresses

 (4-1)

where the residual stress is applied in the horizontal direction, and the vertical stress is

 (4-2)

and is Poisson’s ratio,  is soil density,  is gravity acceleration,  is the depth from ground surface to the position where vertical and horizontal stress is decided,  is the lateral residual compression stress.

The coefficient of earth pressure at rest is

 (4-3)

FRANC2D allows the residual stress to vary arbitrarily, and in many natural cases I expect the residual horizontal stress in a soil may be large at shallow depths but decrease at greater depths. Nevertheless, over a relatively small range of depths where a hydraulic fracture may occur, it seems reasonable to assume that the residual stress is uniform for the purpose of initial evaluations. This gives

 (4-4)

for the profile of the lateral earth pressure as a function of depth in HFRANC2D. This shows that the ratio of horizontal to vertical stresses is uniform in the absence of a residual stress. Where residual stresses occur, this function predicts that *K*o is large at shallow depths and it decreases as an inverse function of depth and approaches  as the vertical stress becomes significantly larger than the residual stress. This function gives extremely large values of *K*o at very shallow depths, but it gives a reasonable approximation to the distribution of *K*o in some soil settings if the very shallow depths are ignored.

**2.3*. Results***

Numerical simulation of this model is performed for different values of . The trajectory of the hydraulic fracture, the fluid pressure, and the length of the fracture were recorded in the simulation using the same methods that are used for modeling hydraulic fracture with different underlying materials.

The form of the fractures changes markedly with changes in . Under fluid pressure and gravity body force loading, the fracture will curve upwards vertically upward (figure 4.2-8). Where residual load is applied to the same model in addition to the fluid pressure and gravity, the fracture will first propagate as a near flat-lying pattern, then a gently-dipping path. Under this circumstance, the fracture propagates away from well bore further than the fracture under no residual lateral stress.

The field measurements of fracture path are similar to the simulation results when there is residual horizontal loading in the model domain.

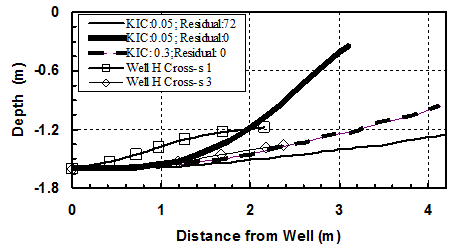


Figure 4.2-8. Fracture path with different *in situ* stress and fracture toughness. The path of cross section 1 and 3 for well H are measured form Richardson’s field data [2003]. KIC has a unit of MPa, residual load has a unit of KPa

**2.4. *Discussion***

The field experiments of hydraulic fracturing have been conducted in the Piedmont soil, where high lateral stress exists. The field excavation of hydraulic fractures by Richardson [2003], has revealed that the trajectory of the hydraulic fractures in the field where high lateral stress existing, shows a similar pattern when compared to the numerical solutions.

The fluid pressure is decreasing after fracture starts to propagate. This is verified by field logging of injection pressure [Richardson, 2003]. The length of the fracture extends longer in high lateral residual stress model. This has also been confirmed by field observation. The fracture created in a low lateral stress field will tend to curve upward immediately after it is initiated horizontally.

Comparison of Figures 4.2-8 and 4.2-4 shows that fracture form, and the rate at which the fracture curves upward in particular, appear to depend on both the in situ stress and the fracture toughness. The fracture toughness used in the simple case shown in Figure 4.2-4 is greater than it is in Figure 4.2-8. The dip of the fracture in 4.2-8 where the residual stress is zero is significantly steeper than it is for 4.2-8. This suggests that increasing either fracture toughness or residual stress will cause the dip of a fracture to flatten. An effect of fracture toughness on the dip of shallow hydraulic fractures has not been recognized in the past, and an investigation of the sensitivity of this effect appears to be warranted during future investigations.

**2.5. *Conclusions***

The finite element fracture simulator FRANC2D is coupled with fluid flow analysis and a propagation criteria to create a code called HFRANC2D that can simulate the propagation of curving hydraulic fractures. The simulator is verified by analyzing simple, idealized problems for which analytical models generally available. An axisymmetric edge fracture model was established to simulate hydraulic fracture propagation. In the simulation, the driving pressure, fracture length, aperture at the fracture center were recorded with time. Verification showed that the error between numerical and analytical solutions is less than 5 percent. This verification exercise used a model that assumes the fluid pressure is uniformly distributed in the fracture, so it verifies the use of the stress intensity propagation criteria, but it does not verify the fluid flow part of the code.

Preliminary analyses of hydraulic fractures created at shallow depths in Piedmont soil were simulated using HFRANC2D. A simple model assuming a homogeneous material loaded only by gravity (no residual stress) was able to predict the fluid pressure, distribution of aperture and the basic form of a hydraulic fracture created in the field and described by Richardson. Many fractures created in the field resemble the one used in this example, suggesting that the simulator is capable of predicting the general characteristics of shallow fractures.

Some fractures have characteristics that differ from the ideal case outlined above. Four models were established to simulate hydraulic fractures initiated within a formation that is underlain by a layer of softer material. The Young’s modulus of these materials range underlying from 2 kPsi to 5 kPsi, whereas the fracture was embedded in material with *E* = 5 kpsi. Numerical simulation showed that hydraulic fractures will tend to curve downward toward an underlying soft layer, but they curve upward toward the ground surface with continued growth. This causes the fracture form to have a low-lying region a meter or so from the borehole, according to the analyses. Hydraulic fractures were recently observed in the field with a similar form (Richardson, 2003). These fractures were created in a relatively stiff clayey silt underlain by a softer saprolite. The results of HFRANC2D simulations explain the peculiar, downward propagation of these fractures as interaction with the underlying saprolite.

Numerical analysis was also conducted to simulate the fracture propagation in a region where high lateral compressiveresidual stress exists. The simulations show that a high residual stress will tend to reduce the dip of the fracture, which is consistent with expectations (e.g. Murdoch, 1995). Numerical experiments conducted using a high residual stress assumed a fracture toughness that is less than the analysis used to evaluate ideal conditions (which used a higher toughness but also assumed no residual stress). Comparison of these results suggests that increasing fracture toughness can affect fracture form by reducing the dip of the fracture.

**2.6. *Future work***

The simulator HFRANC2D has several aspects need to be advanced. More complicated fluid flow problems, such as power law flow need to be included in this code. Menu selections need to be developed for this change to be made. The input and output format need to be changed to a more universal file style among many simulators. It will be more convenient to combine several existing output files to one output file, so that the data processing will be more promptly.

The Young’s modulus and fracture toughness seems have important affects on the curve of hydraulic fracture simulations. It is recommended in the future to conduct geotechnical test on measuring these two parameters precisely. The field data of fracture geometry and soil properties also need to be calibrated to advance the capability of this simulator.

The affect of residual load is just been preliminary investigated. It will be valuable to perform more simulations on how the residual load will change the forms of hydraulic fracture curves. The field experiments to measure the *in situ* stress in Piedmont soil need to be accomplished in a satisfied degree. So that field data can be used as an input for the computer simulation.

**Appendix A. Source Code**

C

C SUBROUTINE: EXC\_AUTO\_DRIVER\_FLOW

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C

C PURPOSE: To propagate crack automatically

C

C CALL: EXC\_AUTO\_DRIVER\_FLOW (Analysis\_available,Data\_storage,A)

C

C INPUT: Analysis\_available

C

C Data\_storage(\*) - Data storage vector

C

C A(\*) - Dynamic Allocation vector

C

C HISTORY: Modified 08-26-2002 Qingfeng Tan

C Pressure is kept constant here, KI increases with time

C Calling adjusttipheadtoprop will adjust pressure and

C keep the KI = KIc

C Modified 02-13-2003 Qingfeng Tan

C loadcommon.inc is added

C Modified 04-13-2003 Qingfeng Tan

C MainFrxLinear is added

C Modified 04-23-2003 Qingfeng Tan

C fluidflowinit and fluidflowcalc is added instead of

C MainFrxLinear

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SUBROUTINE EXC\_AUTO\_DRIVER\_FLOW (Analysis\_available,

1 Data\_storage,A)

USE flowdefs

USE leakoffdefs

IMPLICIT NONE

C Arguments:

LOGICAL Analysis\_available ! Flag for the first config.

INTEGER\*4 Data\_Storage(\*) ! Data base vector

INTEGER\*4 A(\*) ! Address vector

C Global variables:

include '../inc/maincommon.inc' ! main common block

include '../inc/wdbcommon.inc' ! wdb common block

include '../inc/wdbdef.inc' ! wdb definitions

include '../inc/autopropcom.inc' ! automatic prop. common

include '../inc/anlcommon.inc' ! analysis common block

include '../inc/loadcommon.inc' ! loading values, added by Qingfneg Tan

include '../inc/fluidflowdefs.inc' ! fluid flow common block

C Local constants:

REAL\*4 Flow\_rate ! Flow rate to the fracture

PARAMETER (Flow\_rate = 0.000001) ! 1cm^3/sec

C Local Variables and symbols:

REAL\*4 Aper\_Disp(200) ! Aperture array from tip to mouth

REAL\*4 Crack\_length ! Crack length, from tip to mouth

REAL\*4 fp\_F2D(200) ! Driving pressure in crack

REAL\*4 incr ! Increment length of propagation

REAL\*4 Max\_ki ! Maximum KI value

REAL\*4 Open\_to\_Node\_Length(200) ! Distance from crack mouth to points which

! are of node pairs along the crack face

REAL\*4 tipH ! Adjusted crack pressure (head of tip)

REAL\*4 Pumping\_Time ! Injecting time of certain volume crack

REAL\*4 Volume ! Volume of crack inflated

REAL\*4 Xnew,Ynew ! Coordinates of new crack tip

REAL\*4 Crackmult

REAL\*4 headattip

REAL\*4 cputimeend

REAL\*4 TOL1

LOGICAL Gap\_flag

LOGICAL Initialized

LOGICAL Joint\_flag

INTEGER\*4 i ! step counter

INTEGER\*4 jj ! step counter to control the adjustment of crack pressure

INTEGER\*4 l\_int\_cnt

INTEGER\*4 List(100)

INTEGER\*4 Max\_node

INTEGER\*4 MNG\_Set

INTEGER\*4 Num\_tips ! Number of tips

INTEGER\*4 status

INTEGER\*4 step\_count ! Step counter

INTEGER\*4 Vertex

INTEGER\*4 stopstep

INTEGER\*4 numcells

INTEGER\*4 numflowcells

INTEGER\*4 numcellstoadd

INTEGER\*4 timestep

INTEGER\*4 mm

CHARACTER\*40 Text4,Text5,Text6

DATA Text4 / ' Crack achieved a stable configuration'/

DATA Text5 / ' Crack tip is close to a discontinuity'/

DATA Text6 / ' Go Step by Step '/

C Initialize the control variables that control the looping

Initialized = .FALSE.

Propag\_done = .FALSE.

Proximity = .FALSE.

Stable = .FALSE.

step\_count = 0

C Check to see if we have any gap elements

OPEN(UNIT=14, FILE = 'BeforeTIPHEAD.txt')

OPEN(UNIT=9,FILE = 'Aperprofile.txt')

OPEN(UNIT=16, FILE = 'PressureVSTime.txt')

OPEN(UNIT=17, FILE = 'PProfile.txt')

timestep=1

CALL ANL\_Count\_Int (l\_int\_cnt,Gap\_flag,Joint\_flag,

1 Data\_storage)

CALL fluid\_flow\_init(Data\_Storage,A,timestep,

1 headattip,numcells)

C Propagation starts here

10 IF ((.NOT. Propag\_done).AND.

1 (timestep.LE.maxtimesteps)) THEN

C Check for proximity to material interfaces or structural boundaries

IF (Proximity) THEN

CALL DSP\_MISC\_TEXT(Text5,Text6)

Propag\_done = .TRUE.

WRITE(\*,\*)

1 ' Crack is close to a boundary: go step by step'

GO TO 10

END IF

C Fluid flow calculation

CALL FLUID\_FLOW\_CALC(Data\_storage,A,

1 timestep,headattip,numcells)

WRITE(\*,\*) 'Stress Intensity Factor: ',SIF(1,1)

CALL WriteResults(numcells) ! write the output file

timestep = timestep + 1

C Now propagate the crack.

IF (propagating) THEN

step\_count = step\_count + 1

IF ( step\_count .GT. Num\_steps ) THEN

Propag\_done = .TRUE.

GOTO 10

END IF

WRITE(\*,\*) 'Step: ',step\_count

WRITE(\*,\*) 'Total steps: ',num\_steps

incr=delr

CALL PST\_AUTO\_INTER (incr,Xnew,Ynew,Data\_storage,A)

CALL EXC\_AUTO\_REMESH(Xnew,Ynew,Data\_Storage,A)

CALL EXC\_LENGTH\_FLOW(Data\_storage,Crack\_Length,

1 Open\_to\_Node\_Length,numflowcells)

currenthalflen=Crack\_Length

numcellsold=numcells

numcells=numcells+1

END IF

C Determine the geometry of the fracture

CALL EXC\_VOLU\_FLOW(Data\_storage,A,Volume)

WRITE(\*,\*) "Volume",Volume

Pumping\_Time = Volume/Flow\_rate

CALL EXC\_LENGTH\_FLOW(Data\_storage,Crack\_Length,

1 Open\_to\_Node\_Length,numflowcells)

CALL ReviseElev(numflowcells)

WRITE(\*,\*) 'Crack\_length', Crack\_length

CALL EXC\_APER\_FLOW(Data\_storage,A,Aper\_Disp)

WRITE(16,\*) Step\_Count,tiphead,fp\_print(1),

1 numcells,Crack\_Length,Opening,

2 SIF(1,1),Volume,fracvol,Pumping\_Time

DO i=1,Crack\_face\_Nodes

WRITE(17,\*) fp\_print(i)

END DO

Status = MNG\_Set('RSP')

IF (Step\_Count.EQ.11) THEN

mm=mm

END IF

GOTO 10

END IF ! end IF started line 10

CALL cpu\_time(cputimeend)

runtime=cputimeend-cputimestart

TOL1=1 ! place to stop during debugging

CLOSE(9)

CLOSE(14)

CLOSE(16)

CLOSE(17)

WRITE (\*,\*) ' Automatic Propagation done'

RETURN

END

C Header: @(#) excaperflow.ff 2.3 5

C

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C SUBROUTINE: EXC\_APER\_FLOW (Return the crack aperture)

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C

C PURPOSE: Given a crack tip/edge, this routine returns

C the aperture vertices on the crack face and the

C nodal coordinates.

C

C CALL: CALL exc\_aper\_flow(Crack\_tip,Data\_storage,A,Aperture)

C

C

C INPUT: Data\_Storage -Block of memory space

C

C OUTPUT:

C Aper\_Disp -Returned for aperture values

C starting from tip to mouth(another tip)

C Opening -crack aperture at mouth

C HISTORY: Created 09-01-02 Qingfeng Tan

C Modified 12-20-02 Qingfeng Tan

C Displacement info added into it

C Modified 03-13-03 Qingfeng Tan

C Add Opening into argument list

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SUBROUTINE EXC\_APER\_FLOW(Data\_storage,A,Aper\_Disp)

IMPLICIT NONE

C Arguments:

INTEGER\*4 A(\*) ! Address vector for displacement

INTEGER\*4 Crack\_tip ! Crack tip node

INTEGER\*4 Data\_Storage ! Data storage vector

C Global Variables and symbols:

include '../inc/maincommon.inc' ! Ptr\_Disp defined in it

include '../inc/dspparams.inc' ! dsp parameters

include '../inc/dspcommon.inc' ! dsp common

include '../inc/wdbcommon.inc' ! wdb common block

include '../inc/wdbdef.inc' ! wdb definitions

include '../inc/workcommon.inc' ! working common block

include '../inc/loadcommon.inc' ! load common block

include '../inc/fluidflowdefs.inc' ! fluid flow variables

C Local constants:

INTEGER\*4 Max\_nodes ! max node numbers in one side of crack

PARAMETER (Max\_nodes = 200)

C Local Varaiables and symbols:

REAL\*4 Aper\_init(Max\_nodes) ! aperture of crack

REAL\*4 Aper\_Disp(Max\_nodes) ! aperture with displacement by loading

REAL\*4 Disp\_x ! node's displacement in x direction

REAL\*4 Disp\_y ! node's displacement in y direction

REAL\*4 Left\_Coord\_List(2,Max\_nodes) ! Coordinates for left crack nodes

! 1:x coordiante, 2:y coordinate

REAL\*4 Left\_Coord\_List\_Disp(2,Max\_nodes)! new coords of nodes on left crack

! face after displacement,

! 1:x coordinate, 2:y coordinate

REAL\*4 Right\_Coord\_List(2,Max\_nodes) ! Coordinates for right crack nodes

! 1:x coordiante, 2:y coordinate

REAL\*4 Right\_Coord\_List\_Disp(2,Max\_nodes)! new coords of nodes on right crack

! face after displacement,

! 1:x, 2:y

REAL\*8 UTL\_DISP ! Displacement function

REAL\*4 x1(Max\_nodes) ! left node's x subtract right node's x

REAL\*4 x1\_Disp(Max\_nodes) ! left node's x subtract right node's x

! including displacement

REAL\*4 y1(Max\_nodes) ! left node's y subtract right node's y

REAL\*4 y1\_Disp(Max\_nodes) ! left node's y subtract right node's y

! including displacement

INTEGER\*4 Address ! pointer address

INTEGER\*4 I ! loop counter

INTEGER\*4 IS ! wdb signal

INTEGER\*4 Left\_Crack\_nodes ! total number of left crack nodes

INTEGER\*4 Left\_Node\_List(Max\_nodes) ! index of left crack nodes

INTEGER\*4 Load\_case ! Load case number (1-6)

INTEGER\*4 Right\_Crack\_nodes ! total number of right crack nodes

INTEGER\*4 Right\_Node\_List(Max\_nodes) ! index of left crack nodes

INTEGER\*4 Status ! result of a wdb function

INTEGER\*4 utl\_and ! a logical AND operation

INTEGER\*4 Vertices(3) ! three vertices along one side of a grid

C Processing begins here:

C Search for crack tip nodes

10 Status = WDB\_GET\_FV (Crack\_tip,Data\_storage)

20 IF (Status .EQ. WDB\_\_NORMAL) THEN

IF (WDB\_CT\_Node) THEN

C Reserve a segment of memory location

CALL UTLGET(A,Max\_nodes\*3,Address)

C First call PRE\_NODE\_LIST get node list of the crack

Call PRE\_NODE\_LIST (Crack\_tip,.true.,Vertices,

1 Left\_Node\_List,Left\_Coord\_List,

2 Max\_nodes,Left\_Crack\_nodes,Data\_storage)

Call PRE\_NODE\_LIST (Crack\_tip,.false.,Vertices,

1 Right\_Node\_List,Right\_Coord\_List,

2 Max\_nodes,Right\_Crack\_nodes,

3 Data\_storage)

C Store the node number and coordinates

C Calculate apertures between two edges and store the values

C Assign aperture of tip to zero

C Then start from the node next to tip till mouth of the crack

OPEN(unit = 5, file = 'aperture.txt')

DO I=1,Left\_Crack\_nodes

C First get displacement of each crack face node

C Calculate the displacement of the left node of the crack

Status = WDB\_FILL\_COM (Left\_Node\_List(I),

1 Data\_storage)

IF (Status .NE. WDB\_\_Normal) THEN

WRITE (\*,97) Status

97 FORMAT (' DSP\_DEF\_MESH: WDB\_Fill\_com ',

1 'returned --> ',Z8)

RETURN

END IF

IF (WDB\_X\_equation .NE. 0) THEN

Disp\_x = UTL\_DISP(WDB\_X\_Equation,0,A(Ptr\_Disp))

Left\_Coord\_List\_Disp(1,I)=Left\_Coord\_List(1,I)

1 +Disp\_x

ELSE IF (WDB\_Load\_case(1) .EQ. 15) THEN

IS = WDB\_FILL\_LOAD(Left\_Node\_List(I),Data\_storage)

Disp\_x = WDB\_X\_load(1) \* Load\_factors(1)

Left\_Coord\_List\_Disp(1,I)=Left\_Coord\_List(1,I)

1 +Disp\_x

END IF

x1(I)=Left\_Coord\_List(1,I)-Right\_Coord\_List(1,I)

IF (WDB\_Y\_equation .NE. 0) THEN

C Get displacement for each node

Disp\_y = UTL\_DISP(WDB\_Y\_Equation,0,A(Ptr\_Disp))

C Add displacement to the original x coordinates

Left\_Coord\_List\_Disp(2,I)=Left\_Coord\_List(2,I)

1 +Disp\_y

ELSE IF (WDB\_Load\_case(1) .EQ. 15) THEN

IS = WDB\_FILL\_LOAD(Left\_Node\_List(I),Data\_storage)

Disp\_y = WDB\_Y\_load(1) \* Load\_factors(1)

Left\_Coord\_List\_Disp(2,I)=Left\_Coord\_List(2,I)

1 +Disp\_y

END IF

C Now do the same displacement calculation to the right node opposite to the

C left node calculated above in the crack face

Status = WDB\_FILL\_COM (Right\_Node\_List(I),

1 Data\_storage)

IF (Status .NE. WDB\_\_Normal) THEN

WRITE (\*,97) Status

RETURN

END IF

IF (WDB\_X\_equation .NE. 0) THEN

Disp\_x = UTL\_DISP(WDB\_X\_Equation,0,A(Ptr\_Disp))

Right\_Coord\_List\_Disp(1,I)=Right\_Coord\_List(1,I)

1 +Disp\_x

ELSE IF (WDB\_Load\_case(1) .EQ. 15) THEN

IS = WDB\_FILL\_LOAD(Right\_Node\_List(I),

1 Data\_storage)

Disp\_x = WDB\_X\_load(1) \* Load\_factors(1)

Right\_Coord\_List\_Disp(1,I)=Right\_Coord\_List(1,I)

1 +Disp\_x

END IF

IF (WDB\_Y\_equation .NE. 0) THEN

C Get displacement at y direction for each node

Disp\_y = UTL\_DISP(WDB\_Y\_Equation,0,A(Ptr\_Disp))

C Add displacement to the original x coordinates

Right\_Coord\_List\_Disp(2,I)=Right\_Coord\_List(2,I)

1 +Disp\_y

ELSE IF (WDB\_Load\_case(1) .EQ. 15) THEN

IS = WDB\_FILL\_LOAD(Right\_Node\_List(I),

1 Data\_storage)

Disp\_y = WDB\_Y\_load(1) \* Load\_factors(1)

Right\_Coord\_List\_Disp(2,I)=Right\_Coord\_List(2,I)

1 +Disp\_y

END IF

y1(I)=Left\_Coord\_List(2,I)-Right\_Coord\_List(2,I)

Aper\_init(I)=sqrt(x1(I)\*\*2+y1(I)\*\*2)

x1\_Disp(I)=Left\_Coord\_List\_Disp(1,I)

1 -Right\_Coord\_List\_Disp(1,I)

y1\_Disp(I)=Left\_Coord\_List\_Disp(2,I)

1 -Right\_Coord\_List\_Disp(2,I)

Aper\_Disp(I)=sqrt(x1\_disp(I)\*\*2+y1\_disp(I)\*\*2) -

1 Aper\_init(I) ! total aperture subtract slot aperture

WRITE(5,\*) Aper\_init(I),Left\_Coord\_List\_Disp(1,I),

1 Left\_Coord\_List\_Disp(2,I),

2 Right\_Coord\_List\_Disp(1,I),

3 Right\_Coord\_List\_DISP(2,I),Disp\_x,Disp\_y,

4 x1\_disp(I),y1\_disp(I),Aper\_Disp(I),

5 Left\_Node\_List(I),Right\_Node\_List(I)

END DO

Opening = Aper\_Disp(Left\_Crack\_nodes)

CLOSE (UNIT = 5)

C Display aperture of crack

ENDIF

Status = WDB\_GET\_NV (Crack\_tip,Data\_storage)

GOTO 20

END IF

C Check to make shur that the reason that we exited the loop was that

C we ran out of nodes

IF (utl\_and(Status,WDB\_Err\_mask)

1 .NE. WDB\_\_Nomore) THEN

WRITE (\*,\*) 'exc\_aper\_flow:WDB error --> ',Status

RETURN

END IF

RETURN

END

C

C Header: @(#) exclengthflow.ff 2.3 09/15/02 15:10:00 Last Update:

C

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C SUBROUTINE: EXC\_LENGTH\_FLOW (Return the crack elevation)

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C

C PURPOSE: Given a crack tip/edge, this routine returns

C the length from mouth to tip or to nodes along

C one side of the crack face

C

C CALL: CALL exc\_length\_flow(Data\_storage,Crack\_Length,

C Open\_to\_Node\_Length)

C

C

C INPUT: Data\_storage -pos. or neg. side of crack

C which is to be listed

C -passed in for crack loading

C

C OUTPUT: Crack\_length -Returned for length values

C starting from tip to mouth

C (or another tip)

C Open\_to\_Node\_Length -Returned for length values

C starting from mouth opening

C to crack face nodes

C

C HISTORY: Created 09-15-02 Qingfeng Tan

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SUBROUTINE EXC\_LENGTH\_FLOW(Data\_storage,Crack\_Length,

1 Open\_to\_Node\_Length)

USE flowdefs

IMPLICIT NONE

C Arguments:

INTEGER\*4 Data\_Storage(\*) ! Data storage vector

C Global Variables and symbols:

include '../inc/maincommon.inc'

include '../inc/wdbcommon.inc'

include '../inc/wdbdef.inc' ! wdb definitions

include '../inc/workcommon.inc'

include '../inc/loadcommon.inc'

include '../inc/fluidflowdefs.inc' ! fluid flow variables

C Local constants:

INTEGER\*4 Max\_nodes ! maximum node numbers

PARAMETER (Max\_nodes = 200)

C Local Varaiables and symbols:

REAL\*4 Crack\_length ! crack length, from from to tip

REAL\*4 Left\_Coord\_List(2,Max\_nodes) ! Left crack face node coords

! (i,j), i:1-> x coordinates, 2-> y

! coordinates; j -> node index, starts

! from 1-tip node, ends at mouth node

REAL\*4 Right\_Coord\_List(2,Max\_nodes)! right crack face node coords

REAL\*4 Open\_to\_Node\_Length(Max\_nodes) ! crack from mouth opening to

! each mid point of crack face node

! pairs, parameter=1, then it's from

! mouth to mouth; last nonzero value

! is from mouth to tip

REAL\*4 x(Max\_nodes) ! x coord of center point of crack

! face node pairs

REAL\*4 y(Max\_nodes) ! y coords of center point of

! crack face node pairs

INTEGER\*4 Crack\_tip ! Crack tip node index

INTEGER\*4 I ! Loop counter

INTEGER\*4 j ! Loop counter

INTEGER\*4 Left\_Crack\_nodes ! node number in left crack face

INTEGER\*4 Right\_Crack\_nodes ! node number in left crack face

INTEGER\*4 Left\_Node\_List(Max\_nodes) ! node index of left crack face

INTEGER\*4 Right\_Node\_List(Max\_nodes) ! node index of right crack face

INTEGER\*4 Status ! result of wdb function call

INTEGER\*4 utl\_and ! function operating logical AND

INTEGER\*4 Vertices(3) ! three vertices along one side of a grid

INTEGER\*4 numflowcells ! numbers of cells used in flow calc

C Processing begins here:

C Search for crack tip nodes

10 Status = WDB\_GET\_FV (Crack\_tip,Data\_storage)

20 IF (Status .EQ. WDB\_\_NORMAL) THEN

IF (WDB\_CT\_Node) THEN

Call PRE\_NODE\_LIST (Crack\_tip,.true.,Vertices,

1 Left\_Node\_List,Left\_Coord\_List,

2 Max\_nodes,Left\_Crack\_nodes,Data\_storage)

Call PRE\_NODE\_LIST (Crack\_tip,.false.,Vertices,

1 Right\_Node\_List,Right\_Coord\_List,

2 Max\_nodes,Right\_Crack\_nodes,

3 Data\_storage)

Crack\_Face\_Nodes=Left\_Crack\_nodes

C Store the node number and coordinates

C Calculate Length and store the values

C Then start from the node next to tip till mouth of the crack

OPEN(UNIT = 5, FILE = 'length.txt')

DO I = 1, Left\_Crack\_nodes

x(I)=ABS(Left\_Coord\_List(1,I)-Right\_Coord\_List(1,I))/2

x(I)=x(I)+MIN(Left\_Coord\_List(1,I),

1 Right\_Coord\_List(1,I))

y(I)=ABS(Left\_Coord\_List(2,I)-Right\_Coord\_List(2,I))/2

y(I)=y(I)+MIN(Left\_Coord\_List(2,I),

1 Right\_Coord\_List(2,I))

END DO

C Assign first length to zero, which is from mouth to mouth

Open\_to\_Node\_Length(1)=0

DO I = 2,Left\_Crack\_nodes

Open\_to\_Node\_Length(Left\_Crack\_nodes-I+2) =

1 SQRT((y(I-1))\*\*2+(x(I-1))\*\*2)

END DO

DO I=1,Left\_Crack\_Nodes

Rcoord\_Fem(I)=Open\_to\_Node\_Length(I)

Elev\_Fem(i)=(Right\_Coord\_List(2,i)+

1 Left\_Coord\_List(2,i))/2

2 -(Right\_Coord\_List(2,Left\_Crack\_Nodes)+

3 Left\_Coord\_List(2,Left\_Crack\_Nodes))/2

END DO

Crack\_length = Open\_to\_Node\_Length(Left\_Crack\_nodes)

C Write crack length to global variable

Frac\_Length=Crack\_length

C Write all results to text file

DO I =1, Left\_Crack\_nodes-1

WRITE(5,\*) I,Open\_to\_Node\_Length(I),

1 Left\_Coord\_List(1,I),Right\_Coord\_List(1,I),

2 Left\_Coord\_List(2,I),Right\_Coord\_List(2,I),x(I),

3 y(I),Left\_Node\_List(I),Right\_Node\_List(I)

END DO

CLOSE (UNIT = 5)

C Wirte length from mouth to every node as radial coords to global

numflowcells=Left\_Crack\_nodes-1

END IF

C Check if another node is availabile

Status = WDB\_GET\_NV (Crack\_tip,Data\_storage)

GOTO 20

END IF

C Check to make sure that the reason that we exited the loop was that

C we ran out of nodes

IF (utl\_and(Status,WDB\_Err\_mask)

1 .NE. WDB\_\_Nomore) THEN

WRITE (\*,\*) 'WDB error --> ',Status

RETURN

END IF

RETURN

END

C

C Header: @(#) excvoluflow.ff 2.3 10/03/02 15:10:00 Last Update:

C

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C SUBROUTINE: EXC\_VOLU\_FLOW (Return the crack volume)

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C

C PURPOSE: Given a crack tip/edge, this routine returns

C the cross-section area then the volume of

C the crack.

C

C CALL: CALL exc\_volu\_flow(Data\_storage,A,Volume)

C

C

C INPUT: Data\_storage -pos. or neg. side of crack

C which is to be listed

C -passed in for crack loading

C

C A -Max. number of nodes allocated

C

C OUTPUT: Volume -Returned for volume values

C of a circular disk shaped crack

C starting from tip to mouth

C (or another tip)

C

C HISTORY: Created 10-03-02 Qingfeng Tan

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SUBROUTINE EXC\_VOLU\_FLOW(Data\_storage,A,Volume)

C Arguments:

INTEGER\*4 Crack\_tip ! Crack tip node

INTEGER\*4 Data\_Storage ! Data storage vector

INTEGER\*4 A(\*) ! Address vector

C Global Variables and symbols:

include '../inc/dspparams.inc' ! dsp parameters

include '../inc/dspcommon.inc' ! dsp common

include '../inc/exccommon.inc' ! exc common block

include '../inc/maincommon.inc' ! the fefap common

include '../inc/matparams.inc' ! material parameters

include '../inc/matcommon.inc' ! material common block

include '../inc/wdbcommon.inc' ! wdb common block

include '../inc/wdbdef.inc' ! wdb definitions

include '../inc/workcommon.inc' ! work common block

include '../inc/loadcommon.inc' ! lock common block

include '../inc/fluidflowdefs.inc' ! fluid flow variables

C Local constants:

REAL\*4 pi ! circle rate

PARAMETER (pi = 3.1415926535)

INTEGER\*4 Max\_nodes ! max node number

PARAMETER (Max\_nodes = 200) ! some array can have

C Local Varaiables and symbols:

REAL\*4 Aperture(Max\_nodes) ! aperture of each segment of crack with initial aperture in it

REAL\*4 Aper\_NoIniAper(Max\_nodes)! aperture of each segment of crack, without initial

! aperture in it

REAL\*4 Area(Max\_nodes) ! area of the cross-section of crack

REAL\*4 Crack\_Length ! total length of the crack

REAL\*4 Disp\_x ! displacement in x/y directions

REAL\*4 Disp\_y ! displacement in x/y directions

REAL\*4 Left\_Coord\_List(2,Max\_nodes) ! Left crack face node coords

! (i,j), i:1-> x coordinates, 2-> y coordinates;

! j -> node index, starts from 1-tip node,

! ends at mouth node

REAL\*4 Left\_Coord\_List\_Disp(2,Max\_nodes) ! Left crack face coords with displacement

REAL\*4 L(Max\_nodes) ! average length of one crack segment

REAL\*4 mid1\_x ! x coords of mid point of first pair of crack nodes

! in one segment of crack

REAL\*4 mid1\_y ! y coords of mid point of first pair of crack nodes

! in one segment of crack

REAL\*4 mid2\_x ! x coords of mid point of second pair of crack nodes

! in one segment of crack

REAL\*4 mid2\_y ! y coords of mid point of second pair of crack nodes

! in one segment of crack

REAL\*4 Open\_to\_Node\_Length(200) ! crack length, from mouth to crack nodes pair center

REAL\*4 r1(Max\_nodes) ! distance from center point to y axis, see drawing above

REAL\*4 r2(Max\_nodes) ! distance to y axis

REAL\*4 Right\_Coord\_List(2,Max\_nodes) ! Right crack face node coords

REAL\*4 Right\_Coord\_List\_Disp(2,Max\_nodes) ! Right crack face coords with displacement

REAL\*4 Total\_Area ! total cross-section area of crack

REAL\*4 Total\_Vol\_IniAper ! Total volume of initial slot (0.02 thick)

REAL\*8 UTL\_DISP ! return displacement by this function

REAL\*4 Volume ! volume of crack

REAL\*4 Vol(Max\_nodes) ! volume of each segment crack

REAL\*4 Vol\_IniAper(Max\_nodes) ! volume of initial slot (0.02 thick)

INTEGER\*4 Left\_Crack\_nodes ! total number of nodes in left face of crack

INTEGER\*4 Right\_Crack\_nodes ! total number of nodes in right face of crack

INTEGER\*4 I ! loop counter

INTEGER\*4 Left\_Node\_List(Max\_nodes)! node index array for left face nodes of crack

INTEGER\*4 Right\_Node\_List(Max\_nodes)! node index array for right face nodes of crack

INTEGER\*4 Load\_case ! loading case number (1-6)

INTEGER\*4 Status ! result of wdb function call

INTEGER\*4 utl\_and ! a logical AND function

INTEGER\*4 Vertices(3) ! three vertices along one side of a grid

C Processing begins here:

C Search for crack tip nodes

10 Status = WDB\_GET\_FV (Crack\_tip,Data\_storage)

20 IF (Status .EQ. WDB\_\_NORMAL) THEN

IF (WDB\_CT\_Node) THEN

C First call PRE\_NODE\_LIST get node list of the crack

C Reserve a segment of memory location

CALL UTLGET(A,Max\_nodes\*3,Address)

CALL PRE\_NODE\_LIST (Crack\_tip,.true.,Vertices,

1 Left\_Node\_List,Left\_Coord\_List,

2 Max\_nodes,Left\_Crack\_nodes,Data\_storage)

CALL PRE\_NODE\_LIST (Crack\_tip,.false.,Vertices,

1 Right\_Node\_List,Right\_Coord\_List,

2 Max\_nodes,Right\_Crack\_nodes,Data\_storage)

C Then start from the node next to tip till mouth of the crack

C Call EXC\_APER\_FLOW to get aperture information of the crack

CALL EXC\_APER\_FLOW(Data\_storage,A,Aperture)

C Call EXC\_LENGTH\_FLOW to get length of crack

CALL EXC\_LENGTH\_FLOW(Data\_storage,Crack\_Length,

1 Open\_to\_Node\_Length,numcellsold)

C Check the new Crack nodes coordinates with displacement in it

OPEN(UNIT = 14, FILE = 'checkvolume.txt')

C Loop through all the nodes to calculate new coords after displacement

DO I=1,Left\_Crack\_nodes

C First get displacement of each crack face node

C Calculate the displacement of the left node of the crack

Status = WDB\_FILL\_COM (Left\_Node\_List(I),

1 Data\_storage)

IF (Status .NE. WDB\_\_Normal) THEN

WRITE (\*,97) Status

97 FORMAT (' DSP\_DEF\_MESH: WDB\_Fill\_com ',

1 'returned --> ',Z8)

RETURN

END IF

IF (WDB\_X\_equation .NE. 0) THEN

Disp\_x = UTL\_DISP(WDB\_X\_Equation,0,A(Ptr\_Disp))

Left\_Coord\_List\_Disp(1,I)=Left\_Coord\_List(1,I)

1 +Disp\_x

ELSE IF (WDB\_Load\_case(1) .EQ. 15) THEN

IS = WDB\_FILL\_LOAD(Left\_Node\_List(I),Data\_storage)

Disp\_x = WDB\_X\_load(1) \* Load\_factors(1)

Left\_Coord\_List\_Disp(1,I)=Left\_Coord\_List(1,I)

1 +Disp\_x

END IF

IF (WDB\_Y\_equation .NE. 0) THEN

Disp\_y = UTL\_DISP(WDB\_Y\_Equation,0,A(Ptr\_Disp))

Left\_Coord\_List\_Disp(2,I)=Left\_Coord\_List(2,I)

1 +Disp\_y

ELSE IF (WDB\_Load\_case(1) .EQ. 15) THEN

IS = WDB\_FILL\_LOAD(Left\_Node\_List(I),Data\_storage)

disp\_y = WDB\_Y\_load(1) \* Load\_factors(1)

Left\_Coord\_List\_Disp(2,I)=Left\_Coord\_List(2,I)

1 +Disp\_y

END IF

C Now do the same displacement calculation to the right node opposite to the

C left node calculated above in the crack face

Status = WDB\_FILL\_COM (Right\_Node\_List(I),

1 Data\_storage)

IF (Status .NE. WDB\_\_Normal) THEN

WRITE (\*,97) Status

RETURN

END IF

IF (WDB\_X\_equation .NE. 0) THEN

Disp\_x = UTL\_DISP(WDB\_X\_Equation,0,A(Ptr\_Disp))

Right\_Coord\_List\_Disp(1,I)=Right\_Coord\_List(1,I)

1 +Disp\_x

ELSE IF (WDB\_Load\_case(1) .EQ. 15) THEN

IS = WDB\_FILL\_LOAD(Right\_Node\_List(I),

1 Data\_storage)

Disp\_x = WDB\_X\_load(1) \* Load\_factors(1)

Right\_Coord\_List\_Disp(1,I)=Right\_Coord\_List(1,I)

1 +Disp\_x

END IF

IF (WDB\_Y\_equation .NE. 0) THEN

Disp\_y = UTL\_DISP(WDB\_Y\_Equation,0,A(Ptr\_Disp))

Right\_Coord\_List\_Disp(2,I)=Right\_Coord\_List(2,I)

1 +Disp\_y

ELSE IF (WDB\_Load\_case(1) .EQ. 15) THEN

IS = WDB\_FILL\_LOAD(Right\_Node\_List(I),

1 Data\_storage)

Disp\_y = WDB\_Y\_load(1) \* Load\_factors(1)

Right\_Coord\_List\_Disp(2,I)=Right\_Coord\_List(2,I)

1 +Disp\_y

END IF

WRITE(14,\*) Disp\_x,Disp\_y,

1 Left\_Coord\_List\_Disp(1,I),

2 Left\_Coord\_List\_Disp(2,I),

3 Right\_Coord\_List\_Disp(1,I),

4 Right\_Coord\_List\_Disp(2,I),

5 Left\_Node\_List(I),

6 Right\_Node\_list(I)

END DO

C Calculate the aperture without initial aperture

Aper\_NoIniAper(1) = Aperture(1)

Aper\_NoIniAper(2) = Aperture(2)

DO I = 3, Left\_Crack\_nodes

Aper\_NoIniAper(I) = Aperture(I).02

END DO

C Calculate volume for axial symmetric problem

IF (Prob\_type .EQ. 0) THEN ! Axial symmetric

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C This is how the volume of crack is defined under axial symmetric condition

C

C ^

C Y /|\ \_\_\_\_\_---------> Crack tip\_

C | r2 /\ /

C |-------/--\* /

C | / / \----------> Aperture

C | / / /

C |\_\_ / /--/---------> L

C | \ / /

C Crack opening |------\* /

C | r1 \/

C |\_\_\_\_\_\_/

C

C V = pi \* (R1 + R2) \*L \* Aperture, when this is the tip segment, than average

C this by 2.

C

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C Calculate the volume first segment between tip and the nodes next to it

r1(2) = (Right\_Coord\_List\_Disp(1,2)+

1 Left\_Coord\_List\_Disp(1,2))/2.0

mid1\_x = r1(2)

mid1\_y = (Right\_Coord\_List\_Disp(2,2)+

1 Left\_Coord\_List\_Disp(2,2))/2.0

r2(2) = Right\_Coord\_List\_Disp(1,1)

mid2\_x = Right\_Coord\_List\_Disp(1,1)

mid2\_y = Right\_Coord\_List\_Disp(2,1)

L(1) = SQRT((mid1\_x-mid2\_x)\*\*2 + (mid1\_y-mid2\_y)\*\*2)

Vol(2) = pi \* ( r1(2)+r2(2) ) \* L(1) \* Aperture(2)/2.0

Vol\_IniAper(2) = pi \* (r1(2)+r2(2)) \* L(1) \* 0.005/2.0

C Calculate the rest part of the crack volume

OPEN(UNIT = 18, FILE = 'voldata.txt')

Do I = 3, Left\_Crack\_nodes

r1(I) = (Right\_Coord\_List\_Disp(1,I)+

1 Left\_Coord\_List\_Disp(1,I))/2.0

mid1\_x = r1(I)

mid1\_y = (Right\_Coord\_List\_Disp(2,I)+

1 Left\_Coord\_List\_Disp(2,I))/2.0

r2(I) = (Right\_Coord\_List\_Disp(1,I-1)+

1 Left\_Coord\_List\_Disp(1,I-1))/2.0

mid2\_x = r2(I)

mid2\_y = (Right\_Coord\_List\_Disp(2,I-1)+

1 Left\_Coord\_List\_Disp(2,I-1))/2.0

L(I) = SQRT((mid1\_x-mid2\_x)\*\*2 + (mid1\_y-mid2\_y)\*\*2)

Vol(I) = pi \* ( r1(I)+r2(I) ) \* L(I) \* Aperture(I)

Vol\_IniAper(I) = pi \* (r1(I)+r2(I)) \* L(I) \* 0.02

WRITE(18,\*) Left\_Coord\_List\_Disp(1,I),

1 r1(I),r2(I),L(I),Aperture(I),Vol(I)

END DO

CLOSE(18)

C Add all the segment volume together

Volume = 0.0

DO I = 2, Left\_Crack\_nodes

Volume = Volume + Vol(I)

END DO

Total\_Vol\_IniAper = 0.0

DO I = 2, Left\_Crack\_nodes

Total\_Vol\_IniAper = Total\_Vol\_IniAper

1 + Vol\_IniAper(I)

END DO

ELSE ! Plane stress and strain

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C This is how the volume of crack is defined under plain stress/strain condition

C

C ^

C Y /|\ D(x4,y4)

C | /\----/-------> Crack tip

C | / \*--/-------> Aperture(I-1)

C | / / \/C(x3,y3)

C | / / /

C |A(x1, / /--/---------> Open\_to\_node\_Length(I)-

C |----- \ / / Open\_to\_node\_Length(I-1)

C | y1) \*--/---------> Aperture(I)

C Crack Opening | \/B(x2,y2)

C |-------/

C

C V =

C

C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C Calculate the area of crack

Area(1) = Aperture(1)\*Open\_to\_Node\_Length(Left\_Crack\_Nodes-1)/2

DO I=2, Left\_Crack\_nodes

Area(I) = (Aperture(I)+Aperture(I-1))/2.0\*

1 Open\_to\_node\_Length(Left\_crack\_Nodes-(I-2))

END DO

C Calculate the total area of the crack

Total\_Area = 0.0

DO I = 1, Left\_Crack\_Nodes

Total\_Area = Total\_Area + Area(I)

END DO

C Calculate the total volume of the crack

Volume = Total\_Area \* Mat\_le\_iso\_thk \* 2.0

END IF

ENDIF

Status = WDB\_GET\_NV (Crack\_tip,Data\_storage)

GOTO 20

END IF

IF (utl\_and(Status,WDB\_Err\_mask)

1 .NE. WDB\_\_Nomore) THEN

WRITE (\*,\*) 'WDB error --> ',Status

RETURN

END IF

C Record the data to a file

OPEN(UNIT = 15, FILE = 'volume.txt')

WRITE(15,\*) 'VOLUME', Volume

CLOSE (unit = 15)

RETURN

END

**Appendix B. Residual Load Input File Format**

The input file for residual load has an extension name of “.rsi”. After HFRANC2D read the “.rsi” file, it will create a “.rsc” file including interpreted residual loads over the whole model domain. Assume a model has a rectangular geometry. Residual load needs to be applied all over its domain. The four corner nodes’ coordinates are (starting from the lower left corner, counted counter-clockwise): (, ), (, ), (, ), and (, ). Uniform residual normal stresses , and , shear stress  are applied over all the domain. Then the residual load file should look like:

B 



0. 0. 0. 0.

B 



0. 0. 0. 0.

B 



0. 0. 0. 0.

B 



0. 0. 0. 0.

, , ,  are residual loads that will applied under plane stress or plain strain condition. Lines with zeros are strain values. They are ignored in current version of HFRANC2D. The file format shown here can initiate a uniform residual load all over the domain. Also distributive residual load can be applied by insert positions where the residual load has changed.

In section 4.2.3, residual load is applied to simulate hydraulic fractures in shallow depth. The model domain used in section 4.2.3 is a rectangular domain that has 12 m long and 5 m in depth. The origin of the coordinates is at the lower left corner. A lateral uniform compression residual load with a value of 72300 Pa want to be applied all over the model domain. Then, the residual load input file used in this model is written as:

B 0.0 –5.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 12.0 -5.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 12.0 -4.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 12.0 -3.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 12.0 -2.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 12.0 -1.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 12.0 0.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 0.0 0.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 0.0 -1.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 0.0 -2.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 0.0 -3.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

B 0.0 -4.0

-72300.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0

Distributive residual load maybe used in the future with the input file written in this format. IF the residual load is uniform, then only the node at the corner needs to be defined.

**The basic literature**

1. Alehossein, H. and Korinets, A. (1997). *Numerical elastic analysis of a surface crack near Boussinesque load*, International Journal of Fracture 86: L39-L44, 1997.

2. Carter, B., Ingraffea, A., *Effects of casing and interface behavior on hydraulic fracture*, Computer Methods and Advances in Geomechanics, H. Siriwardane and M. Zaman, Eds., A. A. Balkema, Rotterdam, Vol. 2, 1561-1566, 1994.

3. Castell, M., Ingraffea, A. R., Irwin, L. *Fatigue crack growth in pavements*, 126, 4, 283-290, ASCE J. Transportation Engrg., 2000.

4. Geertsma, J., *Two-dimensional fracture-propagation models*, in Recent Advances in Hydraulic Fracturing. SPE Monograph, edited by Gidley, J.L., et al, 1989.

5. Geertsma, J., and Klerk, F., *A rapid method of predicting width and extent of hydraulically induced fractures*, Journal of Petroleum Technology, Vol. 21, No. 12, Dec. 1969, pp. 1571-1581.

6. Charsley, A.D., Martin, C.D., and McCreath, D.R. *Sleeve-fracturing limitations for measuring in situ stress in an anisotropic stress environment*, International Journal of Rock Mechanics & Mining Sciences 40 (2003) 127-136.

7. Fairbanks, C. D. and Andrus, R. Initial Geotechnical Investigation at Simpson Station Experimental Site, Pendleton, South Carolina.Department of Civil Engineering, Clemson University. Unpublished Report, August 2002.

8. Gidley, J.L., S.A. Holditch, and D.E. Nierode, and R.W. Veatch. 1989 Recent Advances in Hydraulic Fracturing. SPE Monograph Volume 12.452 p.

9. Grise, W. R. and Zargari, A. 1997. *Delamination and cracking failures in high-voltage stator winding coatings*, Proceedings of the 24th Electrical/Electronics Insulation Conference, 24th, 1997, 835-839.

10. Ingraffea, A.R., Numerical modeling of fracture propagation, Rock fracture mechanics, Edited by H.P. Rossmanith, published by Springer-Verlay, 1983, pp. 151-208.

11. Ingraffea, A.R., Theory of crack initiation and propagation in rock, Fracture mechanics of rock, 1987, by Academic Press Inc. (London) Ltd., pp. 71-110.

12. Ingraffea, A.R., *Case studies of simulation of fracture in concrete dams*, Fracture Mechanics Vol. 35, No. 1/2/3, pp. 553-564, 1990.

13. James, M., and Swenson, D., *FRANC2D/L: a crack propagation simulator for plane layered structures*, available from <http://www.mne.ksu.edu/~franc2d/>.

14. Lamkin, S. J., Wawrzynek, P. A. and Ingraffea, A.R., *Two - dimensional numerical simulation of interacting fractures in rock*, Fracture of Concrete and Rock: Recent Developments, S. P., Shah, S.E. Swartz, B. Barr, editors, Elsevier Science Publishing, New York, N.Y., 1989, pp. 121 - 131.

15. Lawn, B., Fracture of brittle solids, Second edition, 1993, Cambridge University Press, 1993, pp. 23-26.

16. Linsbauer, H. N., Ingraffea, A. R., Rossmanith, H. P., Wawrzynek, P. A., *Simulation of cracking in a large arch dam: part i*, Journal of Structural Engineering, Vol. 115, No. 7, July 1989, pp. 1599 - 1615.

17. Linsbauer, H. N., Ingraffea, A. R., Rossmanith, H. P., Wawrzynek, P. A., *Simulation of cracking in a large arch dam: part ii*, Journal of Structural Engineering, Vol. 115, No. 7, July, 1989, pp. 1616 - 1630.

18. Murdoch, L. C., 1991. *TheHydraulic fracturing of soil*, Ph.D. dissertation, University of Cincinnati.

19. Murdoch, L. C., 1993. *Hydraulic fracturing of soil during laboratory experiments*, Parts a-c. Geotechnique, v. 43, no. 2: 255-287.

20. Murdoch, L. C., 1994. *Induced Fractures*. United States Environmental Protection Agency. p. 39-55.

21. Murdoch, L. C., 1995. *Hydraulic and implulse fracturing for low permeability soils; in petroleum contaminated low permeability soil: hydrocarbon distribution processes, exposure pathways, and in situ remediation technologies*. Edited by Walden, T. American Petroleum Institute. p. C-1 - C-32, E-1 - E-34.

22. Murdoch, L.C., 1995. *Forms of hydraulic fractures created during a field test in overconsolidated glacial drift*. Quarterly Journal of Engineering Geology, 28, 23-35.

23. Murdoch, L.C. and Chen, J-L, (1997). *Effects of conductive fractures during in situ electroosmosis*, Journal of Hazardous Materials 55 (1997) 239-262.

24. Murdoch, L.C. and Slack, W. 2002. *Forms of hydraulic fractures in shallow fine-grained formations*. Journal of Geotechnical and Geoenvironmental Engineering, Vol. 128, No. 6, June 1, 2002, pp. 479-487.

25. Murdoch, L.C. 2002. *Analysis of an idealized shallow hydraulic fracture*. Journal of Geotechnical and Geoenvironmental Engineering, Vol. 128, No. 6, June 1, 2002, pp. 488-495.

26. Nilson, R.H., *An integral method for predicting hydraulic fracture propagation driven by gases or liquids,* International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 10, 191-211 (1986).

27. Nilson, R.H., and Griffiths, S.K., *Numerical analysis of hydraulically-driven fractures*, Computer Methods in Applied Mechanics and Engineering 36 (1983) 359-370.

28. Nilson, R.H., *Similarity solutions for wedge-shaped hydraulic fractures driven into a permeable medium by a constant inlet pressure*, International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 12, 477-495 (1988).

29. Nykanen, T. 1999. *Geometric dependency of fatigue strength in a transverse load-carrying cruciform joint with partially-penetrating K-welds*, Welding in the world, Vol. 43, No. 5, sept-oct 1999, 47-53.

30. Perkins, T. K., and Kern, L. R., *Width of hydraulic fracture*, Journal of Petroleum Technology, Vol. 13, Sept. 1961, pp. 937-949.

31. Pollard, D.D., and Holzhausen, G., On the mechanical interaction between a fluid-filled fracture and the earth’s surface, Techonophysics, 53 (1979) 27-57.

32. Richardson, J., 2003, *Forms of hydraulic fractures at shallow depth in piedmont soil*, Master’s thesis, Clemson University.

33. Rogers, C.A. and Hancock, G.J. (2001). *Fracture toughness of G550 sheet steels subjected to tension*, Journal of Constructional Steel Research 57 (2001) 71-89.

34. Sowers, G.F., and Richardson, T.L. 1982. Residual Soils of the Piedmont and Blue Ridge. Transportation Research Record No. 919, National Academy Press, Washington D.C., 10-20.

35. Srinivasan, M., *Fracture Analysis Code, A computer-aided teaching tool*, Master’s Thesis, Cornell University.

36. Tada, H., Paris, P.C. & Irwin, G.R. (1985). *The stress analysis of cracks handbook*, 2nd edn. St. Louis: Paris Productions.

37. Trego, A. and Cope, D. (2001). *Evaluation of damage tolerance analysis tools for lap joints*, AIAA Journal Vol.39, No. 12, December 2001.

38. Wang, H. F., and Anderson, M. P., 1982, *Introduction to groundwater modeling*, Academic Press, Inc.

39. Wawrzynek, P. and Ingraffea, A. (1996). FRANC2D: A two dimensional crack propagation simulator, Version 2.7 user’s guide. NASA Contractor Report 4572 (1994).

**The supplementary literature**

1. Ralph H. Abraham and Jerrold E. Marsden. Foundations of Mechanics. -Wesley, 1978.

2. V. I. Arnold. Mathematical Methods of Classical Mechanics. - Springer Verlag, 1980.

3. Lothar Nordheim. The Principles of Mechanics.Vol. 2. - Springer, 1972.

4. Jean-Marie Souriau. Structure des Systèmes Dynamiques. - Dunod Université, Paris, 1970. (English translation: Birkhäuser Boston, 1998.)

5. E. C. G. Sudarshan and N. Mukunda.Classical Dynamics: A Modern Perspective. - John Wiley & Sons, 1974.

6. V. I. Arnold, V. V. Kozlov and A. I. Neishtadt. Mathematical Aspects of Classical and Celestial Mechanics.Dynamical Systems III. - Springer Verlag, 1988.

7. Giovanni Gallavotti. The Elements of Mechanics. - Springer Verlag, 1983.

8. F. R. Gantmakher. Lectures on analytical mechanics. - Fizmatgiz, 1960. (English translation by G. Yankovsky. - Mir Publishing, 1970.)

9. Herbert Goldstein. Classical Mechanics.2nd edition. - Wesley, 1980.

10. Jorge V. José and Eugene J. Saletan.Classical Dynamics: A Contemporary Approach. - Cambridge University Press, 1998.

11. Cornelius Lanczos. The Variational Principles of Mechanics.4th edition.- University of Toronto Press, 1970; Dover Publications, 1982.

12. Пронина Р.Ф. Пособие по переводу английской научно-технической литературы. - М.: Высшая школа, 1973.

13. Циммерман М., Веденеева К. Русско-английский научно-технический словарь переводчика. - М.: Наука, 1991.

14. Арушанян О.Б. Русско-английский словарь по прикладной математике и механике. (Russian-English dictionary of applied mathematics and mechanics). – М.: МГУ, 2012.

15. Пумпянский А.Л. Чтение и перевод английской научно-технической литературы: Лексика, грамматика, фонетика, упражнения /Худ.обл. М. В. Драко.— М.: ООО «Попурри», 1997.— 608 с.

16. L. D. Landau and E. M. Lifshitz. Mechanics.3rd edition.Course of Theoretical Physics.Vol. 1. - Pergamon Press, 1976.

17. Jerrold E. Marsden and Tudor S. Ratiu.Introduction to Mechanics and Symmetry. - Springer Verlag, 1994.

18. Florian Scheck. Mechanics, From Newton's Laws to Deterministic Chaos.2nd edition. - Springer-Verlag, 1994.

19. E. T. Whittaker. A Treatise on Analytical Dynamics.- Cambridge University Press, 1937.

20. Walter Thirring. A Course in Mathematical Physics 1: Classical Dynamical Systems, translated by Evans M. Harell, - Springer-Verlag, 1978.

21. Англо-русский словарь математических терминов: около 20000 терминов / под ред. П. С. Александрова. - Изд.3-е, стер. - М.: Мир, 2001.

22. Голденков М. Современный активный English. ЧеРо. – М: Физматлит, 2002.-448с.

**The list of periodicals**

1. Английский язык: Прил. к газете "1 сентября".

2. Иностранные языки в высшей школе.

**The source list on electronic media**

1. КутателадзеС.С. Russian-English in writing. Изд. института математики. – Новосибирск, 2000.

2. Игнатьев-Каллэхэм Л. Русско-английский политехнический словарь. - М.: Уайли, 1995.

3. A.J. Lohwater's Russian-English dictionary of the mathematical sciences / R.P. Boas (ed.). AmericanMathematicalSociety, 1990.

4. Сосинский А. Б. Как написать математическую статью по-английски. — М: Изд-во «Факториал Пресс», 2000. — 112 с.

**The Internet sources**

1. http://ru.wikipedia.org

2. <http://lingualeo.com/>

**Lecture 6**

**Lecture topic: Response of beam due to impact load.** **A large plate with a small through the thickness crack**

**The plan**

**1. Response of beam due to impact load**

1.1. Introduction

1.2. Analytic Method

1.3. Numeral results and discussion

1.4. Conclusions

1.5. Barrier reflection for E >V0

1.6. Examples

**2. A large plate with a small through the thickness crack with an initial length**

**1. Response of beam due to impact load**

**1.1. *Introduction***

Since structure may subject to crushing of comparatively rigid heavy object as the falling rock in mountain area or falling heavy loads in factories and as the terror threat level elevated in the world recently, the demand of impact resistance structure incited many researches and designers to study the impact responses of the structure. While the main support member of the structures, beam plays a main role to affect the structures impact responses performance. Hence, the purpose of this study is to analyze the impact responses of beam which is the fundamental concept for the other structure members.

Impact load is not only provided high loading intensity over a short period of time, but it also causes the fracture and damage occurs. Hence, the ability to simulate the impact event by analytical or numerical analysis greatly reduces the costs by eliminating the requirement of full scale prototypes. Similarly, the saving of time required and safety factor are improved through the modeling using analytical and numerical method.

According to Selcuk and Frank (2009) [1] , although there are wide ranges of methods available in present practice, it is hard to find a single general accepted method for the structure design to resist impact loads. Some methods with relatively practical formulations and recommendations tend to put several limitations on geometry and modeled structures types to simplify the impact problem formulation. This simplification may ignore the important effect and lead to inaccurate as well as empirical information about impact response of analyzed structure. Thus, it causes the unsafe and inefficient design of structure.

Therefore, it will be great contributions if the numerical modeling method which capable to reflect the complexity of the model geometry is developed for study the beam responses due to the impact load. Since the powerful finite element software – ABAQUS not only can define the complexity of geometry, but also can modeling the transient load with different step analysis, this modeling method have high accuracy with shorter analysis time. However, to verify the accuracy of this numerical modeling method, the conventional analytical method may be utilized as well.

In the past, a number of attempts have been carried out to study the dynamic responses of beam analytically. Hoppmann (1948) [2] has predicted the responses of a uniform beam resting on Winkler model of elastic foundation with energy method. However, only simply supported beam on elastic foundation has been analyses with Energy Method.

Prasad and Kameswara (1979) [3] have utilized model analysis method and energy method for predicting the responses of the beam-foundation system subjected to impact loads. The beam-foundation system consists of free-free Euler-Bernoulli beam resting on Winkler-Zimmerman model of elastic foundation represented by linear and independent elastic springs.

Nevertheless, the comparison of results obtained from modal analysis method, energy method and finite element modeling Abaqus software to verify the responses result from the ABAQUS modeling methods have never been conduct and will be study in this paper.

**1.2. *Analytic Method***

Modal analysis and energy method (using Lagrange’s Equation) are extensively used by practicing engineers for design complicated structural and mechanical systems subjected to various types of external dynamic loads, such as blast overpressures, earthquake acceleration and gusty wind forced (Demeter, 1995)[4]. This is because of their convenient way to represent the externally applied time dependent loads and computation of dynamic stresses and displacements. The fundamental concept utilization of these methods to modeling the impact responses of beam will be presented in the full paper.

Since beam can be supported by a number of boundary condition, three of the boundary condition which are the simply support beam, fixed-fixed beam and cantilever beam will be covering in this paper.

**1.3. *Numeral results and discussion***

In order to have a clear clarification, the dynamic displacement responses of beam due to the impact hammer load at the mid-span is study by modal analysis,energy method and the finite element package ABAQUS. The beam boundary condition consider in this paper are the simply supported beam, fixed-fixed beam and cantilever beam. The dimensions of the study beam model are showed in Fig. 1.

The beam modes natural frequencies are essential values which will affect the vibration resonance. For analytical method, mode natural frequencies can be calculated from modal analysis and energy method equation.

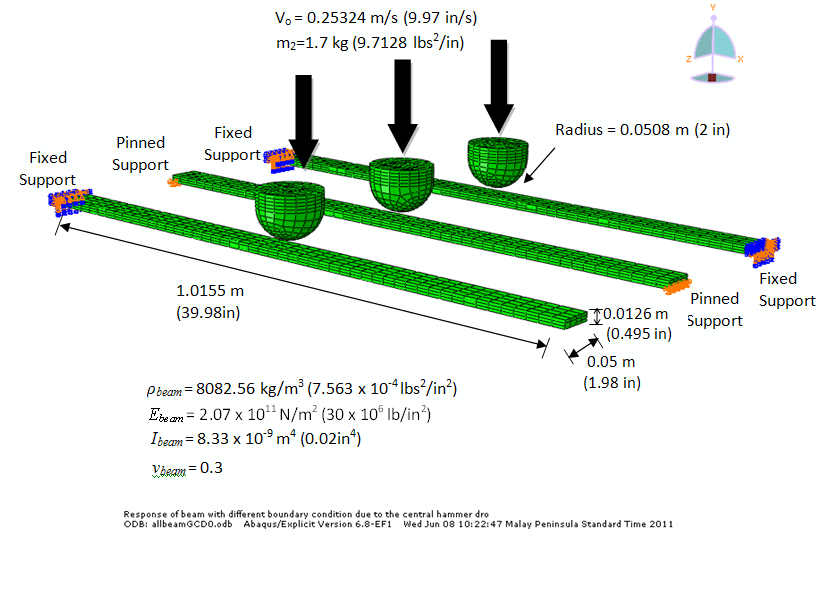


Fig. 1. Geometery of the studied beam model.

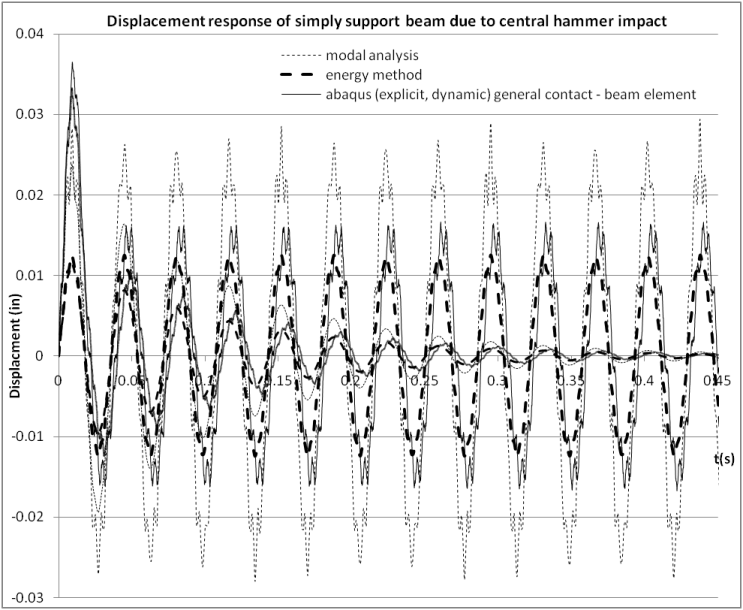


Fig. 2.The central point displacment of simply supported beam due to central hammer with *ξ* =0 and *ξ* =0.05.



Fig. 3. The central point displacment of fixed-fixed beam due to central hammer with *ξ* =0 and *ξ* =0.05.

While Abaqus software is capable to extract the natural frequency of the model through the perturbation method, the natural frequencies of the simply supported beam will be first studied and its values are indicated in a table which will presented in the full paper.

As the displacement responses of beam due to the impact force is the essential consideration if desire to design the impact resistance structure. Hence, the displacement of different support beam which acquired from these three methods will be indicated in Fig.2-3.

Due to the limitation of pages number, the results obtained for the cantilever beam are not able to show in this short paper. However, the results of cantilever beam will be presented and discussed in the full paper.

From Fig.2-3, it can be clearly observed that Abaqus modeling displacement results are within the range of the results from modal analysis and energy method. When the viscous damping coefficient of 0.05 is modeling, the results from the methods mentioned in this study are also showed in Fig.2-3. Although the modal analysis still calculated a higher values results, but the results from energy method and abaqus modeling method are still in the good agreement.

Therefore, it can conclude from all these results that Abaqus modeling method have a good performance as compare to the conservational analytic method like modal analysis and energy method.

**1.4. *Conclusions***

The main goal of this paper is to model the elastic impact responses of beam when subject to the center hammer drop case. By considered the boundary condition, initial displacement and initial velocity of the beam, its dynamic displacement can be analytical modeling by using model analysis and energy method. Furthermore, by using the beam element, explicit, dynamic of Abaqus modeling method, the displacement responses of beam can be study accurately. From the results observation, it can conclude that Abaqus software has great performance on modeling elastic dynamic displacement of beam. Since analytical method are somehow troublesome in handle huge amount of calculation, Abaqus modeling method has great brilliant potential to use in study the impact responses of beam more deeply.

**1.5. *Barrier reflection for E >V0*** *(a numerical treatment)*

Let a flux of particle be incident from the left upon a potential barrier V0 , at the origin (x=0) ,such that the energy of the particles E > V0. ( Use units h(bar)=1, m=1)

The wave function on the left is

ψI (x) = A ei k1\*x + B e –ik1\*x, (x≤ 0) (1)

where

k1=( 2\*E)1/2 .

The wave function on the right is

ψII (x) = C ei k2\*x, (x≥ 0) (2)

where k2=( 2\*(E-V0))1/2 .

Applying the boundary conditions

ψI (0) = ψII (0) (3)

and

(d ψI /dx ) x=0 = (dψII /dx )x=0, (4)

gives

A + B = C, (5)

i( A – B) = i C. (6)

Solving them in terms of A, one gets

B= (k1-k2)\*A/(k1+k2), (7)

C= (2k1\*A)/(k1+k2). (8)

In the present numerical treatment we give trial values to the reflection coefficient B, which is real, and integrate a complex wave function.

What we see by examing the plots of the real and imaginary parts of is that ψII is not of the correct form , ( cos(k2 x) + i sin(k2 x) ) if B is not the correct value.

Three examples are presented , first

B > (k1-k2)\*A/(k1+k2),

Then

B = (k1-k2)\*A/(k1+k2)

and finally

B < (k1-k2)\*A/(k1+k2).

*The data is E= 1, V0 =1.*

The theoretical reflection coefficient is

btheor.= (k1-k2)\*A/(k1+k2)= 0.351000428

**1.6. *Examples***

***Example 1.***

bi= 1.5\* bteor= 0.526500642

An unacceptable solution develops on the region x>0.



The real and imaginary parts ( for x>0) do not have the same amplitude and do not represent the analytical solution

ψII (x) = C (cos(k2\*x) + i sin(k2\*x) ).

***Example 2.***

b= bteor= 0.351000428

An accepatble solution develops in the region x>0.



The real and imaginary parts (for x>0) have the same amplitude and represent the analytical solution

ψII (x) = C (cos(k2\*x) + i sin(k2\*x) )

***Example 3.***

Choosing a reflection coefficient bi< b (theoretical)

bi,bteor= 0.175500214 0.351000428.

An unacceptable solution develops in the region x>0.



Again, the real and imaginary parts ( for x>0) do not have the same amplitude and do not represent the analytical solution

ψII (x) = C (cos(k2\*x) + i sin(k2\*x)).

c barrier reflection E>V0

c C=(2.\*k1\*A)/(k1+k2)

c B= (k1-k2)\*A/(k1+k2)

real k1,k2

complex psi(0:6000),rooti ,g

data E,v0 / 1.3 , 1./

data xi,x0,xf,nstep1/ -3.,0., 5., 1000/

g(i)= -2.\*(e-vx)\*psi(i)

rooti=(0.,1.)

a=1.

k1=sqrt(2.\*e)

k2=sqrt(2.\*(e-v0))

bi= ((k1-k2)\*A)/(k1+k2)

bteor=((k1-k2)\*A)/(k1+k2)

bf=1.2

b=1.5\*bteor

niter=1

db=(bf-bi)/float(niter)

dx=abs(x0-xi)/float(nstep1)

nstep2=int((xf-xi)/dx)

print\*,'bi,bteor=',b,bteor

print\*,' '

psi(0)=a\*exp(+rooti\*k1\*xi)+b\*exp(-rooti\*k1\*xi)

psi(1)=a\*exp(+rooti\*k1\*(xi+dx))+b\*exp(-rooti\*k1\*(xi+dx))

kp=int(float(nstep2)/70.)

kount=kp

do 20 j=1,niter

do 10 i=2,nstep2

x=xi+ dx\*float(i)

vx=v(x-dx,v0)

psi(i)=2.\*psi(i-1)-psi(i-2) + dx\*\*2\*g(i-1)

if(i.eq.kount)then

print 110, x, real(psi(i)),aimag(psi(i))

kount=kount+kp

endif

10 continue

b=b+db

20 continue

110 format(1x,'x,real(psi),imag(psi)=', 3(4x,e10.3))

c=real(psi(nstep1))

print\*,' '

print\*,'cteor, c=',(2.\*k1\*A)/(k1+k2) ,c

print\*,' '

do 30 i=nstep1,nstep2,kp

x=dx\*float(i-nstep1)

print 120 ,x,cos(k2\*x), real(psi(i))/c,sin(k2\*x),aimag(psi(i))/c

30 continue

120 format(1x,'x,cos,real/c,sin,im/c=',5(2x,e9.2))

stop

end

function v(x,v0)

if(x.lt.0.)v=0.

if(x.ge.0.)v=v0

return

end

**2. A large plate with a small through the thickness crack with an initial length**

***Problem:***

A large plate contains a small through the thickness crack with an initial length of *2a0*. The crack is normal to the remote applied tension **, which pulsates between the levels *max* and *min*. Using the propagation equation:



find the expression for the total number of cycles to fatigue fracture.

In the above equation, ** is a constant, *ΔKth* is the threshold stress intensity difference, and *E* is the Young Modulus of the material.

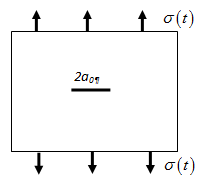
***Solution:***

From equation 7.8 we have,



We calculate first the critical length *ac* :

For the configuration under consideration (Figure below) the SIF is given by 



By definition, we have

at*a*= *ac*

(*KIC* is the fracture toughness, see section 4.6.1)and

.

Note that



Reporting this last expression in the integral, we obtain







Simplifying by *E* and introducing the expression of *ac*



Finally the total number of cycles *Nc* for failure is given by,

.

**The basic literature**

1. Selcuk Saatci and Frank K. Vecchio, “Nonlinear Finite Element Modling of Reinforced Concrete Structures under Impact Loads”, ACI Structural Journal, September – October, 2009.

2. Hoppmann, W. H. Jr., “Impact of a Mass on a Damped Elastically Supported Beam”, Journal of Applied Mechanics, (ASME), 15, 2, pp. 125-136.

3. Bharat Bhushan Prasad and N. S. V. Kameswara Rao, “Beam-Foundation Interaction due to Impact Load”, Indian Geotechnical Journal, Vol. 9, pp. 257-278, July-September 1979.

4. Demeter G. Fertis,“Mechanical and structural vibration”, A Wiley Interscience publication, pp. 448, 1995

5. Chee Siang Chong, Kameswara Rao N.S.V. Response of beam due to impact load /World Journal of Engineeriing, Sabah, Malaysia.

6. Quantum Mechanics by Eugen Merzbacher

7. Quantum Mechanics by Albert Messiah

8. Quantum Mechanics, Vol. 2 by Claude Cohen-Tannoudji, Bernard Diu, and Frank Laloe

9. Schaum's Outline of Quantum Mechanics (Schaum's) by Eliahu Zaarur and Phinik Reuven

**The supplementary literature**

1. Ralph H. Abraham and Jerrold E. Marsden. Foundations of Mechanics. -Wesley, 1978.

2. V. I. Arnold. Mathematical Methods of Classical Mechanics. - Springer Verlag, 1980.

3. Lothar Nordheim. The Principles of Mechanics.Vol. 2. - Springer, 1972.

4. Jean-Marie Souriau. Structure des Systèmes Dynamiques. - Dunod Université, Paris, 1970. (English translation: Birkhäuser Boston, 1998.)

5. E. C. G. Sudarshan and N. Mukunda.Classical Dynamics: A Modern Perspective. - John Wiley & Sons, 1974.

6. V. I. Arnold, V. V. Kozlov and A. I. Neishtadt. Mathematical Aspects of Classical and Celestial Mechanics.Dynamical Systems III. - Springer Verlag, 1988.

7. Giovanni Gallavotti. The Elements of Mechanics. - Springer Verlag, 1983.

8. F. R. Gantmakher. Lectures on analytical mechanics. - Fizmatgiz, 1960. (English translation by G. Yankovsky. - Mir Publishing, 1970.)

9. Herbert Goldstein. Classical Mechanics.2nd edition. - Wesley, 1980.

10. Jorge V. José and Eugene J. Saletan.Classical Dynamics: A Contemporary Approach. - Cambridge University Press, 1998.

11. Cornelius Lanczos. The Variational Principles of Mechanics.4th edition.- University of Toronto Press, 1970; Dover Publications, 1982.

12. Пронина Р.Ф. Пособие по переводу английской научно-технической литературы. - М.: Высшая школа, 1973.

13. Циммерман М., Веденеева К. Русско-английский научно-технический словарь переводчика. - М.: Наука, 1991.

14. Арушанян О.Б. Русско-английский словарь по прикладной математике и механике. (Russian-English dictionary of applied mathematics and mechanics). – М.: МГУ, 2012.

15. Пумпянский А.Л. Чтение и перевод английской научно-технической литературы: Лексика, грамматика, фонетика, упражнения /Худ.обл. М. В. Драко.— М.: ООО «Попурри», 1997.— 608 с.

16. L. D. Landau and E. M. Lifshitz. Mechanics.3rd edition.Course of Theoretical Physics.Vol. 1. - Pergamon Press, 1976.

17. Jerrold E. Marsden and Tudor S. Ratiu.Introduction to Mechanics and Symmetry. - Springer Verlag, 1994.

18. Florian Scheck. Mechanics, From Newton's Laws to Deterministic Chaos.2nd edition. - Springer-Verlag, 1994.

19. E. T. Whittaker. A Treatise on Analytical Dynamics.- Cambridge University Press, 1937.

20. Walter Thirring. A Course in Mathematical Physics 1: Classical Dynamical Systems, translated by Evans M. Harell, - Springer-Verlag, 1978.

21. Англо-русский словарь математических терминов: около 20000 терминов / под ред. П. С. Александрова. - Изд.3-е, стер. - М.: Мир, 2001.

22. Голденков М. Современный активный English. ЧеРо. – М: Физматлит, 2002.-448с.

**The list of periodicals**

1. Английский язык: Прил. к газете "1 сентября".

2. Иностранные языки в высшей школе.

**The source list on electronic media**

1. КутателадзеС.С. Russian-English in writing. Изд. института математики. – Новосибирск, 2000.

2. Игнатьев-Каллэхэм Л. Русско-английский политехнический словарь. - М.: Уайли, 1995.

3. A.J. Lohwater's Russian-English dictionary of the mathematical sciences / R.P. Boas (ed.). AmericanMathematicalSociety, 1990.

4. Сосинский А. Б. Как написать математическую статью по-английски. — М: Изд-во «Факториал Пресс», 2000. — 112 с.

**The Internet sources**

1. http://ru.wikipedia.org

2. <http://lingualeo.com/>

3. [www.geocities.com/serienumerica](http://www.geocities.com/serienumerica)

4. [www.geocities.com/serienumerica2](http://www.geocities.com/serienumerica2)

5. [reibaretti2004@yahoo.com](mailto:reibaretti2004@yahoo.com)

**Lecture 7**

**Lecture topic: Application of Perturbation theory in Classical Mechanics**

**The plan**

**1. Applications in Classical Mechanics**

1.1. Introduction

1.2. Perturbation Hamiltonian

1.3. Perturbation expansion

1.4. Projectile Motion

1.5. Damped Harmonic Oscillator

1.6. Three body problem

1.6.1. Secular Perturbation theory applied to 3-Body problem

1.6.2. Secular Modes in Two- and Three-Planet Systems

**2. Simulation Results**

2.1. Spring –Mass system with no damping

2.2. Spring-mass system with damping factor

2.3. Conclusions

**1. Applications in Classical Mechanics**

**1.1. *Introduction***

A basic theoretical and mathematical overview of the utility of perturbation theory in simple mechanical systems had been described in this lecture.

Perturbation theory is a very broad subject with applications in many areas of the physical sciences. The basic principle is to find a solution to a problem that is similar to the one of interest and then to cast the solution to the target problem in terms of parameters related to the known solution. Usually these parameters are similar to those of the problem with the known solution and differ from them by a small amount. The small amount is known as a perturbation and hence the name perturbation theory. The word “perturbation” implies a small change. Thus, one usually makes a small change in some parameter of a known problem and allows it to propagate through to the answer. One makes use of all the mathematical properties of the problem to obtain equations that are solvable usually as result of the relative smallness of the perturbation.

It is the theory which is the study of the effects of small disturbances .If the effects are small, the disturbances are said to be regular, otherwise they are said to be singular. The basic idea in this theory is to obtain an approximate solution of a mathematical problem by exploiting the presence of a small dimensionless parameter-the smaller the parameter, the more accurate the approximate solution.

Basically the perturbation theory can be divided into two approaches: time dependent and time independent perturbations. There are many point of analogy between the classical perturbation techniques and their quantum counterparts. Generally, classical perturbation theory is considerably more complicated than the corresponding quantum mechanical version.

All of the problems in classical mechanics from elementary principles, central force problems, rigid body motion, oscillations, and theory of relativity had almost exact solutions but in chaos and advanced topics the great majority of problems in classical mechanics cannot be solved exactly and here the perturbation theory comes into play to solve the respective solution in an approximate fashion.

**1.2. *Perturbation Hamiltonian***

In a physical problem that cannot be solved directly the Hamiltonian differs only slightly from the Hamiltonian for a problem that can be solved rigorously. The more complicated problem is then said to be a perturbation of the soluble problem and the difference between the two is called the perturbation Hamiltonian .Therefore the perturbation theory consist of techniques for obtaining approximate solutions based on the smallness of the perturbation Hamiltonian and on the assumed smallness of the changes in the solutions. In general, even when the change in the Hamiltonian is small, the eventual effect of the perturbation on the motion can be large. This suggests that any perturbation solution must be carefully analyzed to be sure that it is physically correct.

The differential equations that describe the dynamics of a system of particles are definitely nonlinear and so one must be somewhat more clever in applying the concept of perturbation theory.

**1.3. *Perturbation expansion***

A regular perturbation case is an equation of the form: D(x;φ)=0 containing a parameter such that the full solution xsol approaches the solution x0 of the simplified equation D(x;φ;0)=0 which tends to 0.

The basic regular perturbation methodology is simple:

1. Write the solution as a power series in



2. Insert the power series into the equation and rearrange to a new power series in

D(xsol;”)= D(

=P0(,0)+P1(;)+P2 (;;)+……..

3. Set each coefficient in the power series equal to 0 and solve the resulting systems

P0(x0;0)=D(x0;0)=0,

P1(x0;x1)=0,

P2(x0;x1;x2)=0.

This determines x0;x1;x2 .The idea applies in many different contexts :

a).approximate solutions to algebraic and transcendental equations;

b).approximate expressions to definite integrals;

c). ordinary and partial differential equations.

The perturbation analysis is often complementary to numerical techniquies.The perturbation analysis sometimes gives asymptotic relationships that are more useful than a small number of numerical experiments. However, in other cases there are really no small parameters and we have to rely on numerical solutions.

**1.4. *Projectile Motion***

The projectile motion in 2-D without considering air resistance with initial velocity of the projectile as v0  and the angle of elevation as θ ,then the force F=mg, the force component becomes :

x-direction;

y-direction .

Neglecting the height, assume x=y=0 at t=0,then

, ,

.

The speed and total displacement as functions of time are found to be

,



to find the range by determining the value of x when the projectile falls back to ground that is when y=0



one value of y=0 occurs for t=o and the other one for t=T



the Range R is found from

.

Next ,if we add the effect of air resistance to the motion of the projectile then there will be decreases in range under the assumption that the force caused by air resistance is directly proportional to the projectile’s motion

The initial conditions are the same as above initial case

.

Then the equations of motion, become



The solution is



and

.

The range R’ which is the range including the air resistance ,can be found as previously by calculating the time T required for the entire trajectory and then substituting this value into above equation for x. The time T is found as previously by finding t=T when y=0.therefore from above equation we find

.

This is a transcendental equation and we cannot obtain an analytic expression for “T”. Therefore perturbation method is used to find an approximate solution .To use this method,we find an expansion parameter which is normally small and in the present case this parameter is the retarding force constant k assuming it to be small. Expand the exponential term of the transcendental equation in a power series with the intention of keeping only the lowest terms of kn  ,where k is the expansion parameter.



If only the terms in the expansion through k3 ,this equation can be rearranged to yield

.

The expansion parameter k in the denominator of the first term on the right hand side of this equation and expand this term in a power series



If we insert this expansion into the first term in the right hand side of the equation and keep only the terms in k to first order then we have

,

neglect,the terms of order  and higher.The limit 0 ie., no air resistance then the above equation gives us the same result as

.

Therefore if k is small but non-vanishing the flight time will be approximately equal to .If by using this approximate value for =,we obtain



this is the desired approximate expression for the flight time .

Next, the equation for x in expanded form:



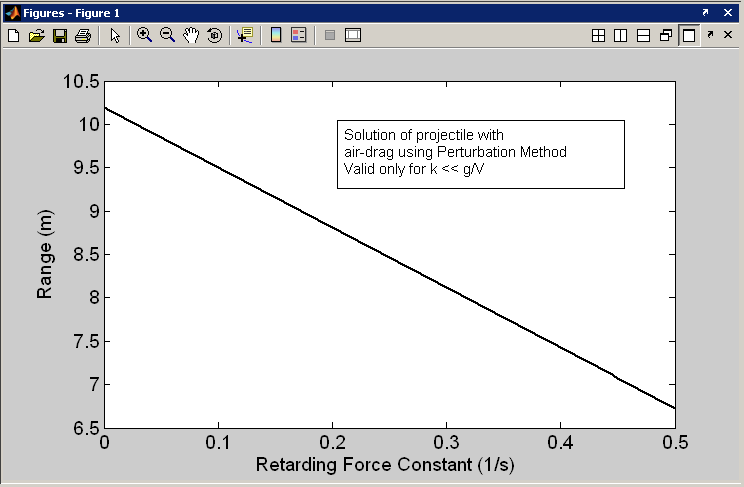
Since, then the approximate range  will be

.

The quantity 2UV/g can be written as

.

The range values calculated approximately: perturbation method is plotted as a function of the retarding force constant k:



**Note:** As the retarding force constant is increased the range R’ gets decreased linearly

The simulation is carried out in Matlab – Simulink Tool.

Ref: Fig2-9; pg: 69; Marion & Thornton, Classical Dynamics of particles and systems, 4th Edition

**1.5. *Damped Harmonic Oscillator***

The oscillation in a simple harmonic oscillator is a free oscillation, once it is set into oscillation, the motion would never cease. To analyze the motion in this case a term representing the damping force is incorporated into the differential equation. It is assumed that the damping force is a linear function of the velocity. Thus if a particle of mass m moves under the combined influence of a linear restoring force –kx and a restoring force ,the differential equation describing the motion is :

,

which can be written as

.

Here β=b/2m is the damping parameter and

ω0 =

is the characteristic angular frequency in the absence of damping. The roots of the auxiliary equation are



The general solution of the equation is

]

Next considering the differential equation for perturbation method:

,

for definiteness, the initial conditions are y(0)=0,y’(0)=1.

Now



for the second order the expression reduces to:

,

.

The initial conditions don’t depend on ε ,so they break into y j (0) =0 and

yo’(0)=1, yo’(0)=1 ,y2 ‘(0)=0

set the coefficient of each power of ε equal to 0 and applying the corresponding boundary condition .

The solution for .

Substitute this into the equation for y1:

.

From the method of undetermined coefficients for an inhomogeneous linear equation with the forcing term “on resonance “



homogeneous solution

The solution satisfying the null initial conditions is

.

This is called a secular term , because it grows with time t .

For the second –order term we get the equation

.

The solution will involve t2  times a trig function and so on to a higher orders. The secular terms get worse .so the constructed equation will be

.

To judge this approximation,comparing it with the exact solution. It is

.

Therefore from expanding this in a Taylor series in ε ,with t fixed ,we get agreement with the perturbative solution . For any given t, our approximation is good if ε is sufficiently small.

**1.6. *Three body problem***

The three-body problem is one of the most celebrated problems in celestial mechanics.. In particular it focuses on the seminal contribution of the French mathematician Henri Poincaré whose attempt to find a solution led him to the discovery of mathematical chaos. The general mathematics of the problem is discussed in many classic texts on both analytical dynamics and celestial mechanics. The three-body problem can be simply stated: three particles move in space under their mutual gravitational attraction; given their initial conditions, determine their subsequent motion. It can therefore be described by a set of nine second-order differential equations. The problem naturally extends to any number of particles, and in the case of n particles it is known as the n-body problem.

Over the years attempts to find a solution to the three-body problem has spawned a wealth of research. Between 1750 and the beginning of the twentieth century more than 800 papers relating to the problem were published invoking a roll call of distinguished mathematicians and astronomers, and hence, as is often the case with such problems, its importance is now perceived as much in the mathematical advances generated by attempts at its solution, as in the actual problem itself. These advances have come in many different fields, including, in recent times, the theory of dynamical systems.

A special case of the three-body problem which has featured prominently in research as a result of its simplified form and its practical applications is what Poincaré called the 'restricted' three-body problem. In this formulation two of the bodies, known as the primaries, revolve around their centre of mass in circular orbits under the influence of their mutual gravitational attraction and hence form a two body system in which their motion is known. A third body, generally known as the planetoid, assumed massless with respect to the other two, moves in the plane defined by the two revolving bodies and, while being gravitationally influenced by them, exerts no influence of its own. The problem is then to ascertain the motion of the third body.

This particular case of the three-body problem is the simplest one of importance and in the context of Poincaré’s work is especially significant since most of his results pertain to this formulation. Apart from its simplifying characteristics, it also provides a good approximation for real physical situations, as, for example, in the problem of determining the motion of the moon around the earth, given the presence of the sun. In this instance, the problem is almost circular (the eccentricity of the earth’s orbit is approximately 0.017), almost planar (both the earth’s orbit and the moon’s orbit are nearly in the plane of the ecliptic), and the values of the mass ratios and the mean distances between the bodies satisfy the conditions. The formulation also provides a reasonable approximation to the system consisting of the sun, Jupiter and a small planet.

Apart from its intrinsic appeal as a simple to state problem, the three-body problem has a further attribute which has been responsible for the abundance of potential solvers: its intimate link with the fundamental question of the stability of the solar system. That is, the question of whether the planetary system will always keep the same form as it has now, or whether eventually one of the planets will escape from the system or, perhaps worse, experience a collision. It is a question which has concerned astronomers for centuries, ever since it was first observed that the motions of the earth and of the other planets were not precisely regular and periodic.

Since bodies in the solar system are approximately spherical and their dimensions extremely small when compared with the distances between them, they can be considered as point masses. Under Newton's law and to a first approximation, the planets move in elliptical orbits around the sun, the sun being at one of the foci of the ellipse. This description is a first approximation because it only allows for the interaction between the sun and the particular planet whose motion is being described and does not take into account the forces between the individual planets. These other forces cause perturbations to the original elliptical orbit so that it very slowly changes and it is conceivable that these very slow changes could, after a very long period of time, alter the present orbits in such a way that a planet could be thrown out of the system or a collision could occur. Although such a scenario does not agree with observations made over the last 1,000 years, it is quite a different thing to prove mathematically that it could not happen, and it is the search for such a mathematical proof that provides the connection with the three-body problem. Ignoring all other forces such as solar winds or relativistic effects and taking only gravitational forces into account, the solar system can be modelled as a ten-body problem having one large mass (the sun) and nine small ones, and investigated accordingly.

**1.6.1. *Secular Perturbation theory applied to 3-Body problem***

It addresses long-period oscillations in planetary orbits, with a history of more than 200 years. Many of the fundamental questions in celestial mechanics have been answered, some interesting ones remain just beyond the scope of basic secular theory. The recent burst of extrasolar planetary system detections has triggered renewed interest in this subject, as simple extensions of the theory may have the potential to explain many of the orbital properties of these systems.

The solution to a two-body system, consisting of a planet and a star, can be described in terms of five fixed orbital elements that define an elliptical Keplerian orbit, and a time-depended one that gives the position of the planet alone the orbit. In systems with more than two planets, Keplerian orbits are no longer exact solutions due to gravitational interactions between the planets. However, since the gravitational forces are still dominated by the central body, each of the planets follows a nearly Keplerian orbit. A similar set of orbital elements, known as osculating elements can be defined at instant in time; these elements vary slowly due to perturbations from other planets. The long-term oscillations of the osculating orbital elements in time is the subject of secular perturbation theory.

The classical secular theory, developed by Laplace and Lagrange , begins with Lagrange's planetary equations which are a set of ordinary differential equations guiding the time evolution of the osculating elements in terms of the perturbing potential Thesecular parts of the perturbing potential are obtained by averaging over all relevant orbital periods so that all short-period terms related to planetary positions along their orbits vanish. The result of this averaging procedure is that the semimajor axes of the planets remain constant, the pericenters and nodes precess, and the eccentricities and inclinations vary quasi-periodically. A linear approach has been applied to the Solar System for more than one and a half centuries details of the theory, which is second order insmall orbital inclinations and eccentricities, and first order in masses Although the application of this low order theory is limited, it gives useful conclusions for the Solar System.

**1.6.2. *Secular Modes in Two- and Three-Planet Systems***

The discovery of multi-planet extra solar systems of the simple two-planet system has drawn renewed attention. The aligned and anti-aligned states of orbits are actually the two eigenmodes and confirmed that with eccentricity damping due to any external forces, one of the two eigenmodes decays quickly leading to either apsidal alignment or anti-alignment, depending on the masses and semimajor axes of the two planets. In addition to apsidal lock, the eccentricities ratio of the two planets remains constant, if eccentricity damping is the only external effect. This result can be used to determine the eccentricity of one planet in a two-planet system, if apsidal lock is observed and the eccentricity of the other planet can be measured.

With the rapid pace of observational discoveries, many additional extra solar systems with two or more planets will emerge. It is important to understand the secular modes and their interaction with other perturbations in these systems. To work in a three planet system, the first problem face is the identification of the eigenmodes. In a two-planet system, there are two natural apsidal co-precession states (aligned and anti-aligned and they compose the two eigenmodes. However, in a three-planet system, there are four natural states while the system can have only three eigenmodes. Dependingon mass ratios and orbital spacings, different sets of three eigenmodes are selected from the four natural states.

The secular potential application has the potential to explain the observed significant non-zero eccentricities of the planets in systems. A further step will be the studyof systems with four planets with applications to the major planets in our Solar System. The possibility of damping of inclinations and eccentricities of the giant planets byinteractions can also be considered .

The solutions and applications for planetary systems with three orfour planets have interesting results from this theory . This approach allows to learn what determines about orbital spacings, eccentricities and inclinations in planetary systems.

**2. Simulation Results**

**2.1. *Spring –Mass system with no damping***

The spring mass system with no damping had been model and simulated in Simulink with an impulse signal an as input to the system.

|  |
| --- |
|  |

Simulink Model for spring-mass system without damping

The equation of motion for spring-mass system having no damping is given as,

.

ODE45 (Dormand-Prince) algorithm with relative tolerance of 10\*e(-6) is used for simulations. Impulse force is applied at t=5s and total time for simulation is 30s.

|  |
| --- |
|  |

Impulse Input Signal

|  |
| --- |
|  |

Output Response: Oscillations

**2.2. *Spring-mass system with damping factor***

The equation of motion for spring-mass system with damping is given as,

.

A term additional to k which constraints the position of body is also included.

|  |
| --- |
|  |
|  |

Output Response for Damped System

System is run with the same parameters as previous. As evident from the output response, because of damping, the oscillations die and body stabilizes after 20s. The settling time, natural frequency and maximum overshoot parameters are function of spring constant and damping factor of the system are given solution of second order equation (in laplace domain) as,

.

**2.3. *Conclusions***

A simple approach of the use of Perturbation theory in mechanical systems has been discussed and analyzed. The mathematics involved in this theory gets complicated for complex systems. This theory is applicable to many areas of science like Quantum Mechanics, Semiconductor Physics, High-Energy particle Physics, etc.

Formal perturbation theory provides a nice adjunct to the formal theory of celestial mechanics as it shows the potential power of various techniques of classical mechanics in dealing with problems of orbital motion. Due to the nonlinearity of the Newtonian equations of motion, the solution to even the simplest problem can become very involved. Nevertheless, the majority of dynamical problems involving a few objects can be solved one way or another. Therefore because of this nonlinearity that so many different areas of mathematics and physics must be brought together in order to solve these problems.

**The basic literature**

1. Classical Dynamics of particles and systems, Marion &Thornton 4th Edition

2. Classical Mechanics, Golstein, Poole & Safko, Third Edition

3. A First look at Perturbation theory, James G.Simmonds & James E.Mann,Jr

4. Perturbation theory in Classical Mechanics, F M Fernandez, Eur.J.Phys.18, 1997.

5. Introduction to Perturbation Techniques ,Nayfeh. A.H.

**The supplementary literature**

1. Ralph H. Abraham and Jerrold E. Marsden. Foundations of Mechanics. -Wesley, 1978.

2. V. I. Arnold. Mathematical Methods of Classical Mechanics. - Springer Verlag, 1980.

3. Lothar Nordheim. The Principles of Mechanics.Vol. 2. - Springer, 1972.

4. Jean-Marie Souriau. Structure des Systèmes Dynamiques. - Dunod Université, Paris, 1970. (English translation: Birkhäuser Boston, 1998.)

5. E. C. G. Sudarshan and N. Mukunda.Classical Dynamics: A Modern Perspective. - John Wiley & Sons, 1974.

6. V. I. Arnold, V. V. Kozlov and A. I. Neishtadt. Mathematical Aspects of Classical and Celestial Mechanics.Dynamical Systems III. - Springer Verlag, 1988.

7. Giovanni Gallavotti. The Elements of Mechanics. - Springer Verlag, 1983.

8. F. R. Gantmakher. Lectures on analytical mechanics. - Fizmatgiz, 1960. (English translation by G. Yankovsky. - Mir Publishing, 1970.)

9. Herbert Goldstein. Classical Mechanics.2nd edition. - Wesley, 1980.

10. Jorge V. José and Eugene J. Saletan.Classical Dynamics: A Contemporary Approach. - Cambridge University Press, 1998.

11. Cornelius Lanczos. The Variational Principles of Mechanics.4th edition.- University of Toronto Press, 1970; Dover Publications, 1982.

12. Пронина Р.Ф. Пособие по переводу английской научно-технической литературы. - М.: Высшая школа, 1973.

13. Циммерман М., Веденеева К. Русско-английский научно-технический словарь переводчика. - М.: Наука, 1991.

14. Арушанян О.Б. Русско-английский словарь по прикладной математике и механике. (Russian-English dictionary of applied mathematics and mechanics). – М.: МГУ, 2012.

15. Пумпянский А.Л. Чтение и перевод английской научно-технической литературы: Лексика, грамматика, фонетика, упражнения /Худ.обл. М. В. Драко.— М.: ООО «Попурри», 1997.— 608 с.

16. L. D. Landau and E. M. Lifshitz. Mechanics.3rd edition.Course of Theoretical Physics.Vol. 1. - Pergamon Press, 1976.

17. Jerrold E. Marsden and Tudor S. Ratiu.Introduction to Mechanics and Symmetry. - Springer Verlag, 1994.

18. Florian Scheck. Mechanics, From Newton's Laws to Deterministic Chaos.2nd edition. - Springer-Verlag, 1994.

19. E. T. Whittaker. A Treatise on Analytical Dynamics.- Cambridge University Press, 1937.

20. Walter Thirring. A Course in Mathematical Physics 1: Classical Dynamical Systems, translated by Evans M. Harell, - Springer-Verlag, 1978.

21. Англо-русский словарь математических терминов: около 20000 терминов / под ред. П. С. Александрова. - Изд.3-е, стер. - М.: Мир, 2001.

22. Голденков М. Современный активный English. ЧеРо. – М: Физматлит, 2002.-448с.

**The list of periodicals**

1. Английский язык: Прил. к газете "1 сентября".

2. Иностранные языки в высшей школе.

**The source list on electronic media**

1. КутателадзеС.С. Russian-English in writing. Изд. института математики. – Новосибирск, 2000.

2. Игнатьев-Каллэхэм Л. Русско-английский политехнический словарь. - М.: Уайли, 1995.

3. A.J. Lohwater's Russian-English dictionary of the mathematical sciences / R.P. Boas (ed.). AmericanMathematicalSociety, 1990.

4. Сосинский А. Б. Как написать математическую статью по-английски. — М: Изд-во «Факториал Пресс», 2000. — 112 с.

**The Internet sources**

1. http://ru.wikipedia.org

2. <http://lingualeo.com/>