

Parallel Computation in Efficient Non-Linear Finite Element Analysis with Applications to Soft-Ground Tunneling Project

by

Yo-Ming Hsieh

M.S., Department of Civil Engineering
National Taiwan University, 1997

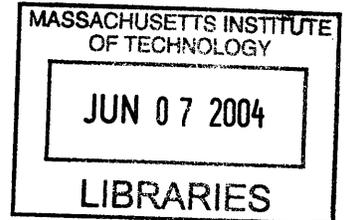
B.S., Department of Civil Engineering
National Taiwan University, 1995

Submitted to the Department of Civil and Environmental Engineering
in Partial Fulfillment of the Requirements for the Degree of

**Doctor of Philosophy in
Geotechnical and Geoenvironmental Engineering at the
Massachusetts Institute of Technology**

June 2004

© 2004 Massachusetts Institute of Technology. All rights reserved.



Signature of Author:

Department of Civil and Environmental Engineering
May 21, 2004

Certified by:

Andrew J. Whittle
Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by:

Heidi Nepf
Chairman, Committee for Graduate Students

BARKER

Parallel Computation in Efficient Non-Linear Finite Element Analysis with Applications to Soft-Ground Tunneling Project

by

Yo-Ming Hsieh

Submitted to the Department of Civil and Environmental Engineering on May 21, 2004
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in
Geotechnical and Geoenvironmental Engineering

ABSTRACT

Reliable prediction and control of ground movements represent an essential component of underground construction projects in congested urban environments, to mitigate against possible damage to adjacent structures and utilities. This research was motivated by the construction of a large underground cavern for the Río Piedras station in San Juan, Puerto Rico. This project involved the construction of a large, horseshoe-shaped cavern (17m wide and 16m high) in weathered alluvial soils. The crown of the cavern is located less than 5.5m below existing buildings in a busy commercial district. Structural support for the cavern was provided by a series of 15 stacked drifts. These 3m square-section galleries were excavated mainly by hand and in-filled with concrete, while a compensation grouting system was designed to mitigate effects of excavation-induced ground movements on the overlying structures. Unexpectedly large settlements occurred during drift construction and overwhelmed the grouting system that was intended to compensate for tunnel-induced movements. Although two dimensional, non-linear finite element analyses of the stacked-drift construction suggest that movements exceeding 100mm can be expected, the 2-D representation of excavation and ground support is overly simplistic and represents a major source of uncertainty in these analyses. Massive computational efforts make more comprehensive 3-D models of the construction sequence completely impractical using existing finite element software with direct or iterative solver methods.

This thesis develops, implements, and applies an efficient parallel computation scheme for solving such large-scale, non-linear finite element analyses. The analyses couple a non-overlapping Domain Decomposition technique known as the FETI algorithm (Farhat & Roux, 1991) with a Newton-Raphson iteration scheme for non-linear material behavior. This method uses direct factorization of the equilibrium equations for sub-domains, while solving a separate interface problem iteratively with a mechanically consistent, Dirichlet preconditioner. The implementation allows independence of the number of sub-domains from the number of processors. This provides flexibility on mesh decomposition, control between iterative interface solutions and direct sub-domain solutions, and load balance in shared heterogeneous clusters. The analyses are performed with the developed code, FETI-FEM (programmed in C++ and MPI) using syntax consistent with pre-existing ABAQUS

software. Benchmark testing on a Beowulf cluster of 16 interconnected commodity PC computers found excellent parallel efficiency, while the computation time scales with the number of finite elements, N_E , according to a power law with exponent, $p = 1.217$.

Parallel 3-D FE analyses have been applied in modeling the drift excavation, primary lining and infilling for the stacked-drift construction assuming a simplified soil profile. The resulting FE model comprised approximately 30,000 20-noded quadratic displacement-based elements, representing almost 400,000 degrees of freedom (at least one order of magnitude larger than any prior model reported in the geotechnical literature) and was sub-divided into 168 sub-domains. The analyses for 60 construction stages required a total of 3 days of computation time using 14 processors. The maximum predicted ground movements at the end of construction range between 80 – 90mm, and are consistent with averaged field measurements. Tunnel-induced volume losses at the ground surface were equivalent to 1.0 – 1.8% of the excavated drift volume and were well correlated with concrete infill events.

The proposed analyses represent a significant advance in computational capabilities for simulating ground response associated with complex construction projects. Further research is needed to extend the capabilities of the current FETI-FEM program to handle coupled flow and deformation problems. Other advances in the modeling of grout injection processes and constitutive behavior of weathered alluvial soil offer further challenges in modeling the behavior of the stacked-drift cavern in Río Piedras.

Thesis Supervisor: Andrew J. Whittle

Title: Professor of Civil and Environmental Engineering

ACKNOWLEDGEMENTS

I would like to thank my mentor Prof. Andrew J. Whittle for his insights and advices throughout the past few years. In addition, I appreciate the freedom given to me to explore, the opportunities to shape my skills, and the effort and time he put into my thesis. I would also like to express my gratitude to my thesis committee members: Prof. Herbert Einstein, Prof. Ruaidhri O'Connor at MIT, and Prof. Youssef Hashash at University of Illinois, Urbana Champaign for their feedback, encouragement, review, and comment on my thesis. Additionally, I am very grateful for the unrestricted access Prof. O'Connor has given me to codex-hammer for my explorations and experiments.

The financial support from the Department of Civil and Environmental Engineering at MIT is highly appreciated, and this assistantship greatly expands my horizon in various aspects, and makes the "MIT experience" a more enjoyable one.

Many thanks to the friendly supporting staff in the department headquarter: Patricia Dixon, Cynthia Stewart, Anthee Travers, Donna Hudson, and many others for their help throughout these years. I am especially thankful for the continuous care and warmth from Patricia Dixon in all these years. I also appreciate the warm and kindly support from Alice Kalemkarian and Sheila Fay in 1-353.

I thankfully have my friends in building one and building five of MIT in my several hundred days at MIT, work with me, grow with me, share with me, and feel the change around MIT with me.

Finally, I thank my family for their unconditional love and support for me.

TABLE OF CONTENTS

ABSTRACT	3
ACKNOWLEDGEMENTS	5
TABLE OF CONTENTS	7
LIST OF TABLES	11
LIST OF FIGURES	13
CHAPTER 1 INTRODUCTION	19
1.1 BACKGROUND ON TREN URBANO PROJECT	19
1.2 THESIS ORGANIZATION.....	20
REFERENCE	25
CHAPTER 2 EVALUATION OF GROUND DEFORMATIONS DUE TO SOFT GROUND TUNNELING	27
2.1 INTRODUCTION	27
2.2 SOFT GROUND TUNNELING.....	27
2.3 EMPIRICAL METHODS	31
2.4 ANALYTICAL METHODS FOR TUNNELING DEFORMATION EVALUATION.....	32
2.5 TWO-DIMENSIONAL FINITE ELEMENT METHODS FOR TUNNELING	33
2.6 THREE-DIMENSIONAL FINITE ELEMENT ANALYSES.....	39
2.7 SUMMARY.....	42
REFERENCES	56
CHAPTER 3 COMPUTATIONAL STRATEGIES TOWARD THREE-DIMENSIONAL FINITE ELEMENT ANALYSES	61
3.1 INTRODUCTION	61
3.2 OVERVIEW OF FINITE ELEMENT METHOD.....	61
3.3 NONLINEAR SOLUTION TECHNIQUES	62
3.4 DIRECT SOLUTION TECHNIQUES	66
3.4.1 <i>Dense Gauss Elimination</i>	66
3.4.2 <i>Performance Considerations</i>	68
3.4.3 <i>Sparse Gauss Elimination</i>	69
3.5 ITERATIVE SOLUTION TECHNIQUES	71
3.5.1 <i>Subspace Method and Residual Minimization</i>	71
3.5.2 <i>Krylov Subspace Method</i>	73
3.6 DIRECT VERSUS ITERATIVE METHODS	75
3.7 PARALLEL COMPUTATION	77

3.7.1	<i>Parallel Computing Hardware</i>	78
3.7.2	<i>Programming for Parallel Computers</i>	79
3.7.3	<i>Parallel Efficiency and Communication Overhead</i>	80
3.7.4	<i>Parallelization of Direct and Iterative solvers</i>	82
3.8	DOMAIN DECOMPOSITION METHOD.....	84
3.8.1	<i>Overlapping Domain Decomposition Method</i>	84
3.8.2	<i>Non-overlapping Domain Decomposition Method</i>	86
3.8.2.1	Formulation of Displacement Based Interface Problem	86
3.8.2.2	Formulation of a Force-Based Interface Problem - FETI Algorithm.....	88
3.8.2.3	Preconditioners for the FETI Algorithm.....	91
3.9	SUMMARY	93
3.10	RECOMMENDATIONS.....	96
	REFERENCE	116
CHAPTER 4 EFFICIENT FINITE ELEMENT SOLUTIONS FOR TUNNELING		119
4.1	INTRODUCTION	119
4.2	BASE-CASE ANALYSIS OF 3-D TUNNEL EXCAVATION.....	119
4.2.1	<i>Computation Requirement</i>	120
4.2.2	<i>Results of Base-Case Example</i>	121
4.2.3	<i>Ground Deformation</i>	123
4.2.4	<i>Tunneling Induced Shearing</i>	124
4.2.5	<i>Tunnel Wall Convergence</i>	125
4.2.6	<i>Summary</i>	126
4.3	EFFICIENT TUNNELING SOLUTION VIA REMESHING	127
4.3.1	<i>Introduction</i>	127
4.3.2	<i>A Simple Remeshing Algorithm for Time-Efficient 3-D Tunnel Analyses</i>	129
4.3.3	<i>Preliminary Calculations</i>	130
4.3.4	<i>Discussion</i>	131
4.4	IMPLEMENTATION OF SCHWARZ ALTERNATING SCHEME WITH ABAQUS	133
4.4.1	<i>Introduction</i>	133
4.4.2	<i>Performing Analyses using the Schwarz Alternating Scheme</i>	134
4.4.3	<i>Preliminary Calculations and Convergence Criterion</i>	135
4.4.4	<i>Discussion</i>	138
4.5	IMPLEMENTATION OF THE FETI METHOD UTILIZING ABAQUS	138
4.5.1	<i>Introduction</i>	138
4.5.2	<i>Interfacing with ABAQUS</i>	139
4.5.3	<i>Rigid Body Mode Extraction</i>	140
4.5.4	<i>Implemented Preconditioners</i>	141

4.5.5	<i>Load Balancing</i>	142
4.5.6	<i>Implementation of FETI Solver</i>	143
4.5.7	<i>Preliminary Performance Evaluation</i>	147
4.6	FULL FINITE ELEMENT PROGRAM IMPLEMENTATION WITH FETI METHOD	150
4.6.1	<i>Introduction</i>	150
4.6.2	<i>Input Files</i>	150
4.6.3	<i>Finite Elements and Material Models</i>	152
4.6.4	<i>Validations</i>	153
4.6.5	<i>Extrusion Program</i>	154
4.7	PERFORMANCE EVALUATION OF FETI_FEM	155
4.7.1	<i>Computing Hardware</i>	155
4.7.2	<i>Performance Evaluation on a Structured Mesh</i>	157
4.7.3	<i>Performance Evaluation for Circular Tunnel Analyses</i>	158
4.7.4	<i>Effect of Communication Fabric on the Performance of FETI_FEM</i>	160
4.7.5	<i>Effect of Preconditioners</i>	161
4.7.6	<i>Suggestions for Future Enhancements on FETI_FEM</i>	162
4.8	SUMMARY	163
	REFERENCES	213

CHAPTER 5 FINITE ELEMENT ANALYSES FOR A STACKED-DRIFT TUNNEL

	CONSTRUCTION	215
5.1	THE RÍO PIEDRAS STATION OF THE TREN URBANO PROJECT	215
5.1.1	<i>Ground Conditions</i>	216
5.1.2	<i>Underground Construction for the Río Piedras Station</i>	217
5.2	MEASURED SURFACE SETTLEMENT DURING CONSTRUCTION	219
5.3	2D FINITE ELEMENT ANALYSES FOR THE RÍO PIEDRAS STATION	224
5.3.1	<i>Introduction</i>	224
5.3.2	<i>Description of 2-D Model</i>	225
5.3.3	<i>Simulation Procedure for Stacked-Drift Construction</i>	227
5.3.4	<i>Results for 2-D Models</i>	229
5.3.5	<i>The Effect of Construction Sequences and Concreting</i>	232
5.3.6	<i>Summary of 2-D FE Models</i>	235
5.4	THREE-DIMENSIONAL FINITE ELEMENT ANALYSES WITH FETI_FEM	236
5.4.1	<i>Finite Element Mesh and Domain Decomposition Strategy</i>	236
5.4.2	<i>Simulation Sequences and Assumptions</i>	237
5.4.3	<i>Computation Time and Solution History</i>	238
5.4.4	<i>Computation Results</i>	239
5.5	COMPARISONS OF COMPUTED AND MEASURED SURFACE SETTLEMENT	241

5.6	ASSESSMENT OF BUILDING DAMAGES BY STACKED-DRIFT TUNNELING	243
5.7	SUMMARY.....	247
	REFERENCES	297
CHAPTER 6 SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS.....		299
6.1	SUMMARY.....	299
6.2	CONCLUSIONS.....	303
6.3	RECOMMENDATIONS.....	306
	REFERENCES	310
APPENDIX.....		319

LIST OF TABLES

Table 2-1 Empirical Relationships for Estimating Transversal Surface Settlements.....	44
Table 2-2 Analytical Solutions for Evaluating Ground Deformations in Elastic Half Plane.	45
Table 2-3 Parameters for 2D Plane-Strain Analyses (Oettl et al. 1998)	45
Table 2-4 Length of Transition Zone versus Plasticity (Lee and Rowe 1990a)	45
Table 2-5 Solution Time for a 3D Finite Element Tunneling Analysis.....	46
Table 3-1 Comparisons of Newton-Raphson with Automatic Sub-stepping and Explicit Integration with Auto-Stepping.....	98
Table 3-2 Comparisons of Direct and Iterative Solvers	98
Table 3-3 The PCPG Algorithm for Solving the FETI Interface Problem.....	99
Table 4-1 Material Properties and K_0 for Base-case Problem	167
Table 4-2 Computation Resources Required for Base-case Problem.....	167
Table 4-3 Solution Time for Full and Decomposed Finite Element Solution.....	167
Table 4-4 Ported Keywords from ABAQUS for finite element model definition	168
Table 4-5 Effect of Different Preconditioners with Fast Ethernet and Myrinet	168
Table 5-1 Parameters for P-1	251
Table 5-2 Determination of Lining Parameters for the Stacked-drift construction	251
Table 5-3 Simulation Procedure for Stacked-Drift Tunnel in 2-D FE Models	252
Table 5-4 Summary of Performed Parametric Studies.....	252
Table 5-5 Parameters for P-2 & P-13	253
Table 5-6 Parameters for P-3.....	253
Table 5-7 Parameters for P-4.....	253
Table 5-8 Parameters for P-5.....	253
Table 5-9 Parameters for P-6.....	253
Table 5-10 Parameters for P-8.....	254
Table 5-11 Parameters for P-9.....	254
Table 5-12 Matching Parameters between Mohr-Coulomb and Drucker-Prager Yield Criteria	254
Table 5-13 Parameters for P-10.....	255
Table 5-14 Parameters for P-11.....	255
Table 5-15 Parameters for P-12.....	255
Table 5-16 Simulation Steps for Three-Dimensional Finite Element Model.....	256

Table 5-17 Summary of Contributions of Surface Settlement from 3D Analyses..... 257

:

LIST OF FIGURES

Figure 1-1 Tren Urbano Rail Transit System	22
Figure 1-2 Section 7 of Tren Urbano Project	23
Figure 1-3 Illustration of the New Río Piedras Station.....	24
Figure 1-4 Approximate Construction History of Stacked-drifts.....	24
Figure 2-1 Staged Excavation Sequences for Heathrow Express Trial Tunnel	47
Figure 2-2 Application of Jet Grouting for Tunnel Construction.....	47
Figure 2-3 Two Types of Closed Face Shield Machines.....	48
Figure 2-4 Settlement Trough as Normal Probability Function.....	49
Figure 2-5 Trough Width Parameter vs. Soil Types	49
Figure 2-6 Determination of parameter m	50
Figure 2-7 Correlations between Ground Loss and Load Factor	50
Figure 2-8 Basic Modes of Tunnel Wall Displacements.....	51
Figure 2-9 Effect of Plasticity Models on Surface Settlement Trough.....	51
Figure 2-10 2D vs. 3D Analyses for 3-D Trapdoor Problems	52
Figure 2-11 Stress Rotation due to Tunneling	53
Figure 2-12 Stress Paths for 2D and 3D Finite Element Analyses	54
Figure 2-13 Three-Dimensional Finite Element Mesh	54
Figure 2-14 MOMIS Database for FE Models in Underground Work	55
Figure 3-1 Summary of the Finite Element Procedure and its Complexity.....	100
Figure 3-2 Newton-Raphson Iteration Scheme	100
Figure 3-3 Flowchart for Newton Raphson Iteration.....	101
Figure 3-4 Convergence Rate for Iterative Solutions	101
Figure 3-5 Illustration of Dense Gauss Elimination	102
Figure 3-6 Pseudo Code for the Dense Gauss Elimination.....	103
Figure 3-7 Effect of Memory Hierarchy	104
Figure 3-8 Variants of Dense Gauss Elimination.....	105
Figure 3-9 Effect of Optimization Using BLAS on Dense Gauss Elimination.....	106
Figure 3-10 Flow Chart for General Subspace Method.....	107
Figure 3-11 Arnoldi-Modified Gram-Schmidt Algorithm.....	107
Figure 3-12 Conjugate Gradient (CG) Algorithm	107
Figure 3-13 Generalized Conjugate Residual (GCR) Algorithm	108

Figure 3-14 Left Preconditioned Conjugate Gradient Algorithm.....	108
Figure 3-15 Comparisons of Solvers in NE/Nastran.....	109
Figure 3-16 Solution Time Scaling with Problem Size N.....	109
Figure 3-17 Shared Memory Architecture	110
Figure 3-18 Distributed Memory Architecture	110
Figure 3-19 System architecture for Earth Simulator.....	110
Figure 3-20 1D and 2D Block-cyclic Layout for 4 Processors.....	111
Figure 3-21 Parallel Efficiency of Direct Solvers in Commercial Finite Element Codes.....	112
Figure 3-22 Parallel Efficiency Degradation on Cray T3E-900.....	112
Figure 3-23 Schwarz Alternating Scheme	113
Figure 3-24 Flowchart for Schwarz Alternating Scheme.....	113
Figure 3-25 Schwarz Alternating Scheme for Multiple Sub-domains.....	113
Figure 3-26 Flowchart for Multiple Sub-domain Schwarz Alternating Scheme	114
Figure 3-27 Non-overlapping Domain Decomposition Method.....	114
Figure 3-28 Offshore Oil Platform.....	115
Figure 3-29 Two Sub-domain Cantilever Problem.....	115
Figure 4-1 Finite Element Mesh for Base-case Study	169
Figure 4-2 Observed Boundary Effects for Elastic Analyses	169
Figure 4-3 Boundary Effects for Elasto-plastic Analyses.....	170
Figure 4-4 Magnitude of Displacement Sectional Contours	171
Figure 4-5 Centerline Longitudinal Surface Settlement Profile,.....	171
Figure 4-6 Transverse Surface Settlement Troughs at Central Section	172
Figure 4-7 Development of Surface Movement with Excavation	173
Figure 4-8 Surface Deformation Evolution with Tunnel Face Advancement.....	174
Figure 4-9 Contour of the Vertical Movement at Initial Excavation Step.....	175
Figure 4-10 Upward Movement for Elastic Analyses	176
Figure 4-11 Upward Movement for Elasto-Plastic Analyses	176
Figure 4-12 Three-Dimensional von-Mises Stress Distribution.....	178
Figure 4-13 Three-Dimensional Shear Strain Distribution	178
Figure 4-14 Tunnel Wall Convergence for Elasto-Plastic Analyses.....	179
Figure 4-15 Numerical Inaccuracies at Tunnel Walls.....	179
Figure 4-16 Adaptive Remeshing Procedure.....	180

Figure 4-17 Ideal Meshing Scheme for Efficient 3-D Tunneling Analyses	181
Figure 4-18 Problems with Solution Mapping during Remeshing Process	182
Figure 4-19 Tunnel Wall Convergence with Remeshing Procedure	183
Figure 4-20 Flowchart for the <i>DD</i> Program	184
Figure 4-21 Input File Specification for the <i>DD</i> Program	185
Figure 4-22 Overlapping Domain Decomposition for Extruded 3-D Tunnel Mesh	186
Figure 4-23 Calculation Steps of Implemented Overlapping Domain Decomposition	187
Figure 4-24 Convergence of Overlapping Domain Decomposition	188
Figure 4-25 Alternative Error Measures for Overlapping Domain Decomposition.....	188
Figure 4-26 Decomposed Finite Element Mesh.....	189
Figure 4-27 Convergence History of Schwarz Alternation for the 8-sub-domain problem	189
Figure 4-28 Comparisons between decomposed problem and full finite element solution.	190
Figure 4-29 Extracting Information from ABAQUS for FETI Algorithm.....	191
Figure 4-30 Rigid Body Mode Extraction Algorithm	191
Figure 4-31 Flowchart for <i>FETI-DD</i>	192
Figure 4-32 Flowchart of Preparations on Worker Processes	193
Figure 4-33 Flowchart of FETI Iteration Implementation on the Master Process.....	194
Figure 4-34 Speed Differences between Scatter and Broadcast Operations	195
Figure 4-35 Speed Differences between Gather and Reduction Operations	196
Figure 4-36 Mesh Decomposition for Preliminary Performance Evaluation.....	196
Figure 4-37 Parallel Efficiency for 8-Sub-domain Calculation	197
Figure 4-38 Parallel Efficiency for 20-Sub-domain Calculation	198
Figure 4-39 Input Files for FETI_FEM	199
Figure 4-40 Example Finite Element Definition File for Sub-domains.....	200
Figure 4-41 Simplified Class Diagram for Element Classes.....	200
Figure 4-42 Simplified Class Diagram for Material Classes.....	201
Figure 4-43 Initial Validation Problems	202
Figure 4-44 Convergence History of Two Test Problems with von Mises yield criterion ...	203
Figure 4-45 Convergence History of a Test Problem with Porous Elasticity	203
Figure 4-46 Input file EXTRUDE.INI for EXTRUDE Performing Uniform Extrusion ..	204
Figure 4-47 A Uniformly Extruded Mesh.....	204
Figure 4-48 Non-Uniform Extrusion Input File	205

Figure 4-49 A Non-uniformly Extruded Mesh	205
Figure 4-50 Flowchart from 2-D to 3-D Domain Decomposed Analyses	206
Figure 4-51 Hardware Setup of Codex-hammer	206
Figure 4-52 NetPIPE Performance on Fast Ethernet and on Myrinet in Codex-hammer	207
Figure 4-53 System Diagram of Two Interconnected Computers	208
Figure 4-54 Mesh Decomposition for Performance Evaluation on a Structured Mesh	208
Figure 4-55 Observed Solution Time Scaling with Problem Size for the Structured Mesh.	209
Figure 4-56 Observed Speedup and Parallel Efficiency for the Structured Mesh	209
Figure 4-57 Mesh Decomposition for Performance Evaluation for Tunneling	210
Figure 4-58 Measured Parallel Efficiency and Speedup for an Unstructured Mesh	210
Figure 4-59 Components of Total Execution Time versus Number of Processors	211
Figure 4-60 Effect of Fast Ethernet and Myrinet on Parallel Performance	211
Figure 4-61 Speedups Using Myrinet on Various Components in FETI_FEM	212
Figure 5-1 Stacked-Drift Construction for the Río Piedras Station	258
Figure 5-2 Soil Profile for the Río Piedras Station Site	259
Figure 5-3 Stacked-Drift Construction and Surface Benchmarks of Río Piedras Station	260
Figure 5-4 Excavation and Construction Sequence of the Stacked-Drift Construction	261
Figure 5-5 Grouting Activities during Stacked-Drift Construction	262
Figure 5-6 Groups of Surface Benchmarks	263
Figure 5-7 Settlement Records from Surface Benchmark Clusters	264
Figure 5-8 Number of Compensation Grouting Performed between CD80 – CD100	268
Figure 5-9 Total Number of Compensation Grouting Performed	268
Figure 5-10 Measured Surface Settlement at the End of Excavation	269
Figure 5-11 Consolidation Grouting, Excavation, and Settlement above the Grouting Gallery	270
Figure 5-12 2-D Finite Element Model for Stacked-Drift Tunnel Analyses	271
Figure 5-13 Stress Paths of Lab Experiments Conducted at MIT for UC and MZ	272
Figure 5-14 Measured Drained Shear Modulus versus Axial Strain for Middle Zone Soil ..	273
Figure 5-15 Calculated Surface Movement, Reference Configuration	274
Figure 5-16 Effect of K_0 on Settlement Troughs	275
Figure 5-17 Effect of Soil Stiffness on Surface Settlement Troughs	276
Figure 5-18 Effect of Lining Stiffness on Surface Settlement Troughs	277

Figure 5-19 Effect of Friction Angle of Middle Zone Soil on Surface Settlement Troughs	278
Figure 5-20 P-7: Effect of Concreting on the Surface Settlement Troughs	279
Figure 5-21 Effect of Different Soil Yield Criteria for the Middle-Zone Soil.....	280
Figure 5-22 Plane-Strain Finite Element Calculations with Alternative Sequences.....	281
Figure 5-23 Volume Losses of Plane-Strain Finite Element Calculations	281
Figure 5-24 Upward Displacement Contours from 2D FEA with Top-Down Sequence ...	282
Figure 5-25 Upward Displacement Contours from 2D FEA with Bottom-Up Sequence...	282
Figure 5-26 Two-Dimensional Decomposition of the Finite Element Model.....	283
Figure 5-27 Special Consideration of Decomposition with Lining Elements	283
Figure 5-28 Three-Dimensional Mesh Decomposition used in FETI_FEM for SDT Model	284
Figure 5-29 Solution History for Three-Dimensional Analysis of SDT using FETI_FEM	284
Figure 5-30 Comparison of the SDT Model Size with MOMIS Database.....	285
Figure 5-31 Incremental Surface Movement during Stacked-Drift Construction	286
Figure 5-32 Evolution of Surface Settlement above Grouting Gallery	288
Figure 5-33 Comparison of 2D and 3D Settlement Trough.....	288
Figure 5-34 Observed Ground Loss from Three-Dimensional Finite Element Analyses ...	289
Figure 5-35 Comparisons between Calculation Results and In-situ Measurements at Selected Offset Locations	290
Figure 5-36 Building Damage Evaluation Chart.....	292
Figure 5-37 Sagging and Hogging of Building Deformation due to Tunneling.....	292
Figure 5-38 Damage Category Chart for $L/H = 1$, Hogging Mode.....	293
Figure 5-39 Settlement Trough under East-Side Buildings for a Particular Section	294
Figure 5-40 Damage Assessment of Stacked-Drift Tunneling with 3-D FEA in the End of Construction.....	295
Figure 5-41 Damage Assessment of Building Damages with 2D Finite Element Analyses throughout the Construction.....	296

Chapter 1

Introduction

1.1 Background on Tren Urbano Project

Figure 1-1 shows the route alignment for phase I of Tren Urbano, a 17.2km, 15-station rail transit system in San Juan, Puerto Rico. The Tren Urbano system aims to reduce traffic congestion within greater San Juan, and future phased extensions of the network are planned to connect old San Juan, the airport, and the eastern town of Carolina. The total cost of the first phase of Tren Urbano is \$1.676 billion¹. The entire system of Tren Urbano comprises 16 stations, 10 of which are elevated, 4 are at grade or in open cut, and 2 are constructed underground. Section 7 of the alignment (shown in Figure 1-2) links the two underground stations, Río Piedras and University of Puerto Rico (UPR), and was constructed entirely underground beneath the densely populated area and the business district of Río Piedras for a length of approximately 1500m. The choice of pursuing for underground construction for this section is based on the decision of local agencies and citizens to preserve the historic Río Piedras section of the city and minimize the disturbance to the community.

Figure 1-2 shows a more detailed view of Section 7 involving four distinct methods of construction through weathered alluvial deposits² in this segment: 1) twin tunnels bored with an Earth Pressure Balance (EPB) tunnel-boring machine, 2) four short sections of open face construction with shotcrete using the NATM (New Austrian Tunneling Method), 3) stacked-drift construction for the Río Piedras station, and 4) cut and cover excavations for the University of Puerto Rico station. The new Río Piedras Station is constructed using the stacked-drift tunneling method and houses two levels of platforms. The cavern is approximately 17m wide, 16m high, and spans 150m between two 30m deep access shafts.

¹ Transportation Infrastructure Finance, <http://tifa.fhwa.dot.gov/fs4.htm>

² The Hato Rey formation or 'old alluvium' deposited in the late Pliocene or early Pleistocene. Extensive information regarding the subsurface conditions can be found in Zhang (2002).

The crown of the tunnel is less than 5.5m below existing commercial buildings along the busy Avenida Ponce de Leon, Figure 1-3.

The stacked-drift tunneling method is unusual. Prior to the excavation of the main cavern, 15 small square-section drifts (each $3m \times 3m$) adjacent to the core are excavated and in-filled with concrete to form a supporting structural arch. Figure 1-4 illustrates the approximate construction sequence of the stacked-drifts for the new Río Piedras Station. The construction started in May 1998 and completed in early 2000. During the construction, several buildings experienced damage due to a larger than expected surface settlement (up to 15cm) despite diligent efforts to control movements through compensation grouting and consolidation grouting programs.

This research work 1) reviews the current modeling capability of tunneling processes. 2) Investigates the ground deformation induced by the stacked-drift tunneling process through compilation of in-situ measures. 3) Improves future stacked-drift constructions by identifying the key performance factors through extensive numerical modeling. 4) Develops an original non-linear general-purpose program using parallel/cluster computing technique to practically solve large-scale finite element models.

1.2 Thesis Organization

Chapter 2 reviews the current practice for estimating ground deformations due to tunneling, including limitations of two-dimensional finite element analyses. The current strategy of 3-D finite element models in tunneling is also reviewed based on publications in the literature.

Chapter 3 identifies the bottleneck in achieving computational efficiency for large-scale finite element models and seeks alternative methods for improving the capabilities of large-scale finite element models. The Chapter introduces the FETI algorithm as a specific Domain Decomposition technique that can be scaled for parallel computation on multiple processors.

Chapter 4 presents computational aspects of finite element solutions for tunneling in soft ground through analyses of a base-case circular tunnel using a commercially available finite element software package. Techniques that utilize existing finite element software are proposed via remeshing and overlapping domain decomposition methods. Finally, the FETI

algorithm is implemented, and its performance is demonstrated for modeling the advancing tunnel heading.

Chapter 5 presents the case study for the stacked-drift tunneling in Section 7 of the Tren Urbano project. The in-situ measured data and major construction activities are compiled and presented. Comparisons with both 2-D and 3-D finite element analyses are performed for this construction. Parametric calculations are done in 2-D finite element analyses to identify important factors of the stacked-drift construction. Comparisons between computed and measured ground movements are made, and damage assessment is made based on the computed ground movements.

Finally, chapter 6 summarizes and concludes findings, and recommends future directions of numerical modeling of tunneling projects.

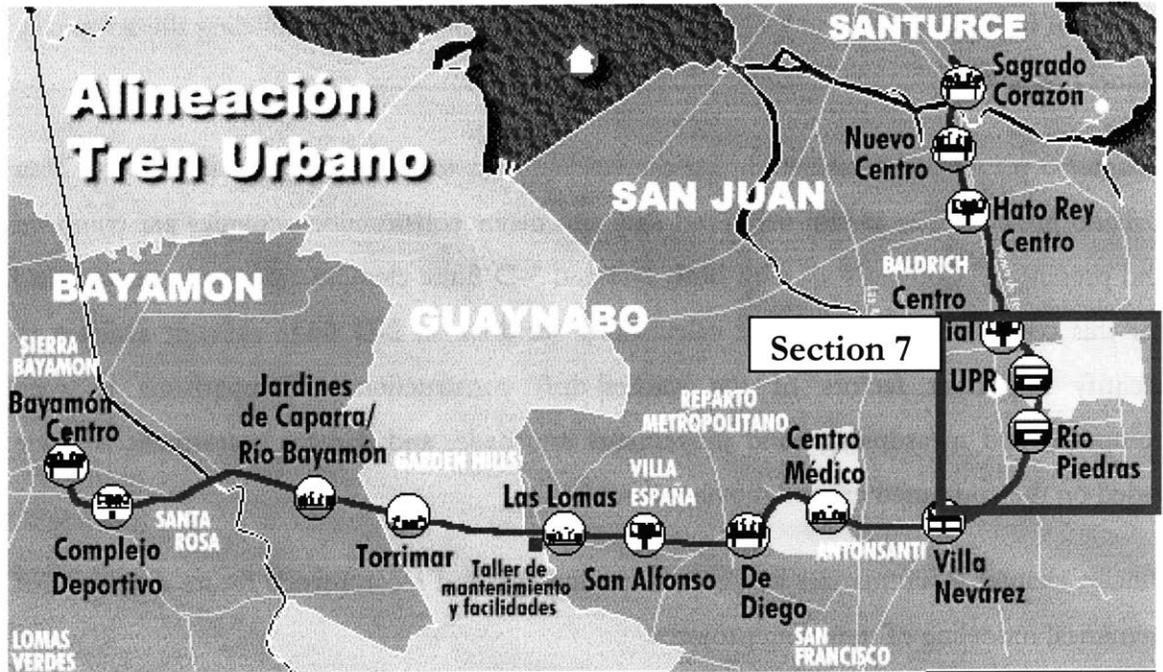


Figure 1-1 Tren Urbano Rail Transit System

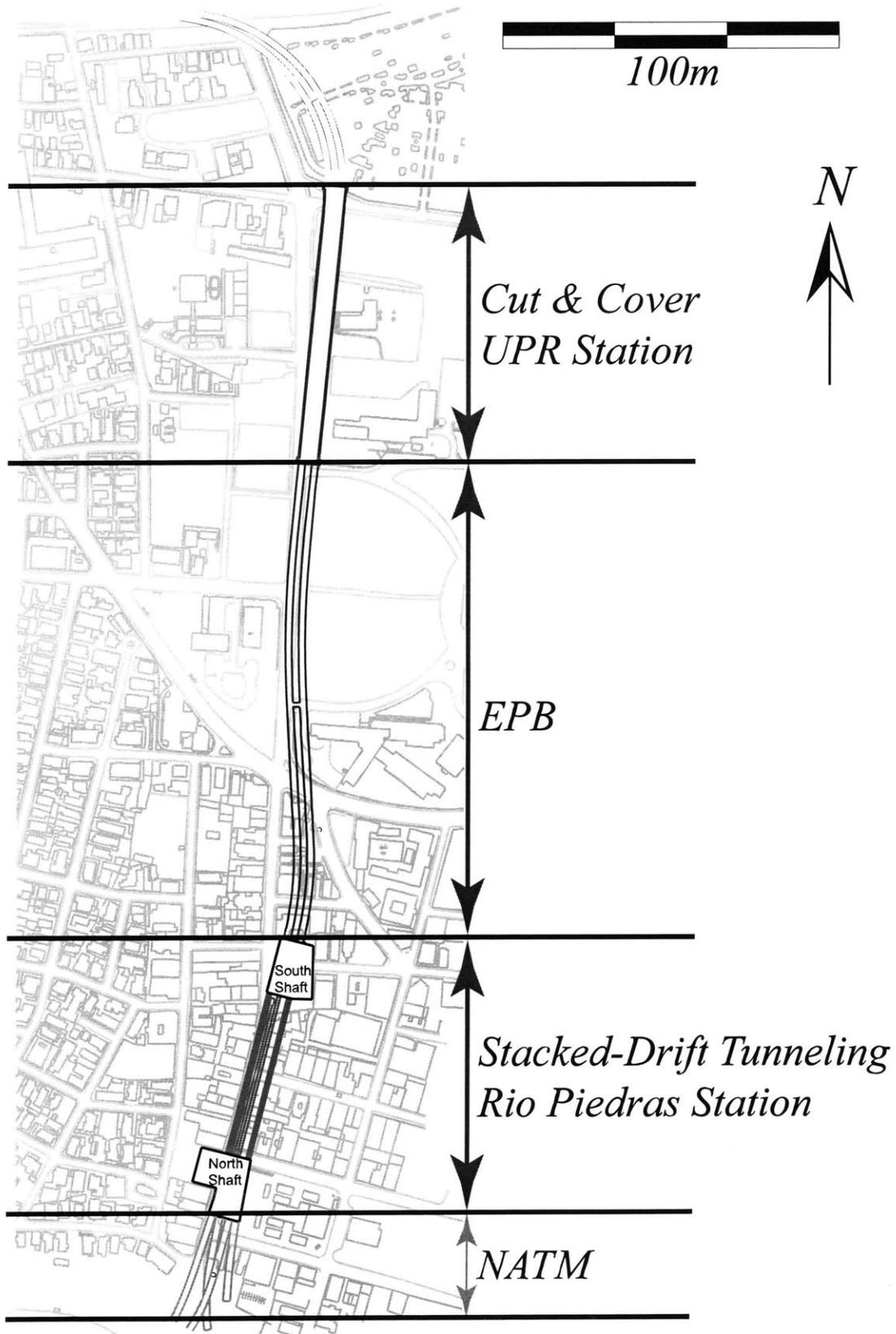


Figure 1-2 Section 7 of Tren Urbano Project

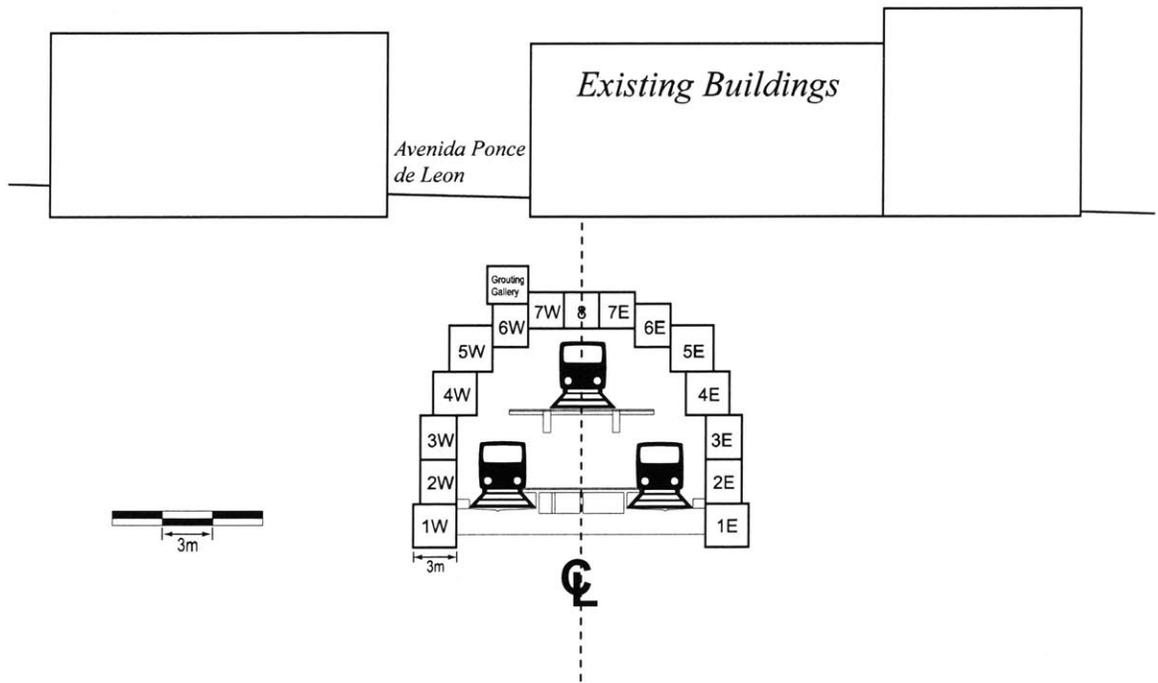


Figure 1-3 Illustration of the New Río Piedras Station

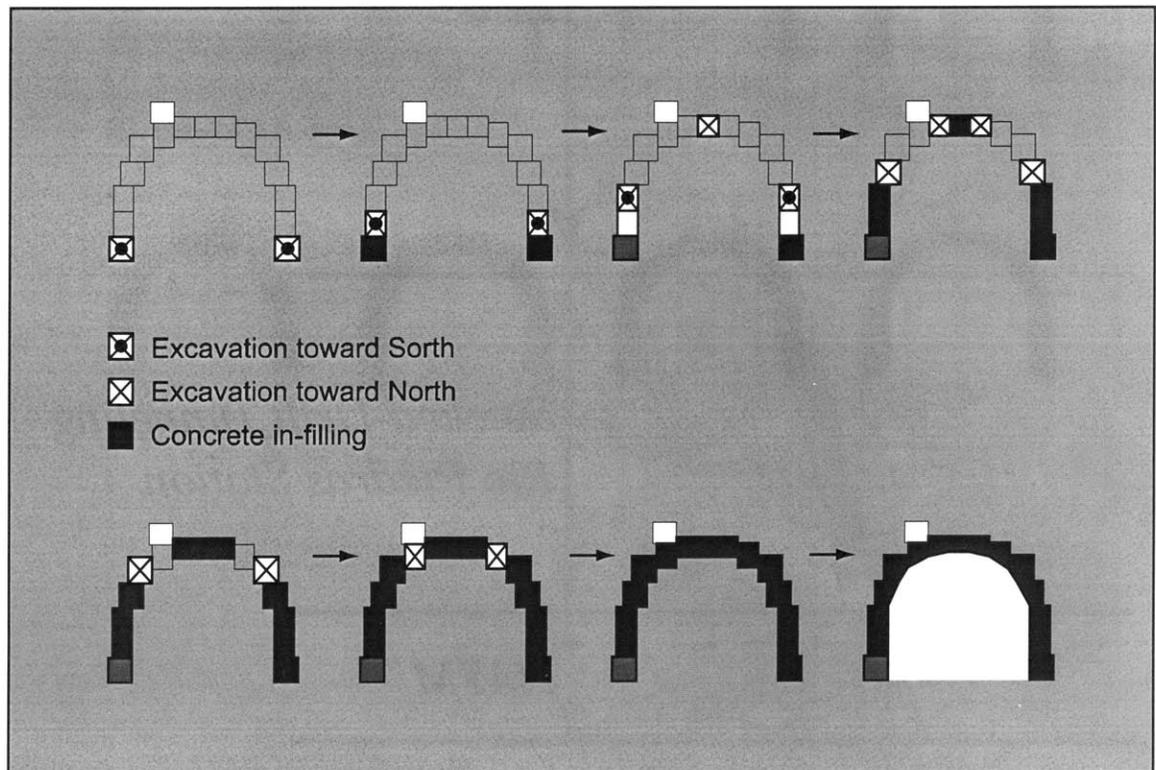


Figure 1-4 Approximate Construction History of Stacked-drifts

Reference

Zhang, G. (2002). "Laboratory Characterization of a Highly Weathered Old Alluvium in San Juan, Puerto Rico," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge.

Chapter 2

Evaluation of Ground Deformations Due to Soft Ground Tunneling

2.1 Introduction

Efficient use of underground space is increasingly important for buildings and transportation corridors in congested urban environments. Underground construction of these projects involves deep excavations or tunneling, and the design must ensure stability at all stages of construction while mitigating impacts on adjacent buildings, utilities, or infrastructures. Current design practice relies on limit equilibrium for stability assessment, and empirical relationships for evaluating construction-induced ground deformations.

This chapter provides a critical review of the diverse methods used to evaluate ground deformations caused by soft ground tunneling ranging from empirical data from prior projects to simplified analytical methods and numerical computations, particularly using non-linear finite element methods. The discussion of numerical methods focuses on approximations used in conventional 2-D models and the current prediction capabilities of 3-D analyses.

2.2 Soft Ground Tunneling

The processes of tunnel construction in soft ground can be sub-divided according to the competence of the ground itself: 1) competent ground, 2) semi-competent ground, and 3) incompetent ground.

Competent ground remains stable or self-standing during the process of excavation, such that open face tunneling methods can be used in construction. Open face shield tunneling has been used in competent soft ground since the 19th century, while other open face constructions use minimal temporary support during excavations (e.g. sprayed concrete/shotcrete linings) for excavation of tunnels with irregular geometries.

The simplest open face tunnel boring machines comprise a shield with an attached rotary cutting head, and are preferred for long, circular tunnel bores. Although ground conditions

may be stable during excavations, long term loading conditions (surcharge effects, ground water flow, and soil consolidations, etc.) are generally resisted by a permanent lining system, which may be installed either concurrently with the heading advance or constructed as a secondary layer of support (i.e. post-excavation).

The primary lining system suppresses ground movements at the tunnel walls through physical contact between support system and the ground. Installation of linings is usually facilitated by over-excavating the opening. Ground movements can thus be reduced by backfilling (e.g. grouting) the space between the cavity and the lining system. Shotcrete or sprayed concrete applications do not need over-excavation, but suffer a delay time for the concrete strength to build up. Support delay is a measure of the delay between completion of excavation and full effectiveness of the lining system, and can be measured in either time or in the unsupported length between the excavating face and the fully supported tunnel segments. The cause of delayed support are 1) physical constraints, 2) overcut gap between the linings and the ground, 3) time for concrete to harden in-situ, and 4) workmanship.

Besides primary lining system, the stability of the tunnel cavity and tunnel-induced deformations can be improved or reduced by staging open face excavations. Figure 2-1 show examples of the staged excavation for Heathrow express trial tunnel (Karakus and Fowell 2003). Three different excavation sequences were compared for this trial tunnel, and it is found the maximum surface settlement corresponding to these three types of excavation sequences are 27.9mm, 26.8mm, and 40.3mm, respectively (Bowers 1997), suggesting the type-2 (Figure 2-1) excavation sequence produces the least surface settlement.

Semi-competent ground with marginal face stability can be effectively constructed using pressurized tunneling methods by maintaining high air pressure using compressed air inside the tunnel cavity. Additional stabilization schemes such as grouting (Figure 2-2), soil nailing, and ground freezing or closed-face tunneling methods are usually necessary for **incompetent ground** with unstable face conditions, or prone to raveling. These incompetent ground conditions often comprise water bearing sands and gravels, or soft clays.

Closed face tunnel boring machines for tunneling in incompetent ground include a) slurry shield and b) Earth Pressure Balance (EPB) machines (Figure 2-3) that are widely used in

clays. Steering control and advance of these boring machines involves a complex feedback between the face pressure and jacking forces. The shield includes a sharp cutting edge for excavation, and the shield can be jacked ahead of the excavated face to enhance stability using the reaction force from the permanently installed lining. Excavated materials are transported from the cutting face through a screw conveyor (Figure 2-3(b)). Mair and Taylor (1997) report recent advances in EPB machines by injecting slurries or foams to improve the flow of excavated soil through the screw conveyor. All TBM machines erect prefabricated segmental concrete linings inside the tail of the advancing shield to facilitate the lining installation.

Besides closed face machine excavation, further enhancement to the face stability may be necessary to overcome difficult ground conditions. Face stabilization can be accomplished by 1) mechanical insertions of structural materials, and 2) ground modification or improvement to improve face stability and to reduce the face intrusion:

1. Mechanical insertions: structural members (e.g. reinforcement bars) are inserted into the ground (usually from within the tunnel cavity) to form bonding between soils and structural insertions to improve stability or reduce deformation. They can be installed in the tunnel heading to stabilize the tunnel face and reduce face intrusion (e.g. soil nailing, Ng and Lee, 2002; Taguchi et al. 1993), around the tunnel heading to improve face stability and reduce the support delay (e.g. fore-poling, Aydan et al. 1988). They can also be used around the perimeter of opening to restrain tunnel wall convergence (e.g. bolting, usually used in rock tunneling). The ground deformations toward opening are resisted by skin friction or bonding between the ground and inserted structural members and their stiffness, and load is transferred to the intact or less disturbed ground through the installed structural member.
2. Ground modification or improvement: possible solutions range from permeation grouting (to strengthen the soil ahead of the advancing tunnel heading) to ground freezing (used very successfully in a massive tunnel jacking project in Boston, Deming et al, 2000). Ground freezing is a temporary measure for improving engineering properties of the ground by freezing the pore fluid, thus improves

stiffness, strength, and reduces permeability of the soil. Additionally, obstructions in the ground can be “*held in place for removal by the mining equipment without process interruption*” (Deming et al. 2000). It should be noted that freezing ground water causes heave, and thawing causes settlement, thus designs must take these freezing-thawing cycle into considerations.

Ground deformations are inevitably induced by all tunnel construction procedures. In the case of open face tunneling, there is full release/redistribution of stress (generating strains and deformations within the soil mass) around the tunnel heading. As the heading advances, some fraction of the in-situ stress is transferred to the primary lining system. Ground deformations around a closed-face TBM occurs due to the over-cut and/or pitching of the rotary cutting head, differential pressures acting at the face, the tail void between the shield and segmented lining, and shearing at soil-shield interface. There are also movements caused by grout injection into the tail void, and displacements of the segmental lining system.

In general, the generation of ground deformations due to tunneling is controlled by a complex set of interactions involving: 1) soil excavation, 2) the primary lining system for stabilizing the opening, 3) the properties of the soil, particularly changes in profile or mixed face conditions, and 4) mitigation strategies ranging from ground improvement (through grouting, soil freezing, etc.) to compensation actions (compensation grouting).

If undesirable surface settlements did occur due to tunnel construction, active measures such as compensation grouting can be performed to control and even reverse the surface settlement. Compensation grouting injects grouts into the ground to compensate the ground loss caused by underground construction and stress relief, and has been successfully applied to several underground construction projects such as Jubilee Line extension project (Harris et al. 1996).

Ground conditions for the tunneling operation in Río Piedras (Chapter 1) were considered competent, enabling open face NATM construction to be used for 4 tunnel segments at the southern edge of the section 7. Similarly open face construction was used for the stacked-drift construction of the Río Piedras Station Cavern (Chapter 5). Twin bore tunnels between

the RP to UPR stations were built using an EPB machine (closed face operation) with precast segmented concrete lining.

2.3 Empirical Methods

Most empirical methods published to date are based on the error function or the normal/Gaussian curve (Peck 1969) to describe transverse surface settlement troughs above a cylindrical tunnel as shown in Figure 2-4 with Equation (2.1):

$$\frac{u_y}{u_y^0} = \exp\left[-\left(\frac{x}{2x_i}\right)^2\right] \quad (2.1)$$

where u_y is the surface settlement, u_y^0 is the settlement that occurs directly above the centerline of the tunnel, x is the horizontal distance from the centerline on the tunnel, and x_i is the inflection point on the distribution curve, frequently referred to as the trough width parameter. According to this distribution curve, the volume of the surface settlement trough ΔV_s is controlled by the parameters u_y^0 and x_i :

$$\Delta V_s \cong \sqrt{2\pi} \cdot x_i \cdot u_y^0 \doteq 2.5 \cdot x_i \cdot u_y^0 \quad (2.2)$$

Assuming there is no volume change within the soil mass (i.e. undrained condition, $\Delta V_g = 0$ in Figure 2-4), then the surface volume ΔV_s can be equated with volume lost at the tunnel cavity such that the tunnel volume loss, V_L can be written as:

$$V_L = \frac{\Delta V_L}{\pi R^2} = \frac{\Delta V_s}{\pi R^2} \cong 0.8 \frac{u_y^0 x_i}{R R} \quad (2.3)$$

The empirical relationship in Equation (2.1) has been further extended to estimate subsurface settlements by further data from field measurements (Moh et al. 1996; O'Reilly and New 1982) and centrifuge model tests (Grant and Taylor 2000; Mair et al. 1993). Table 2-1 summarizes empirical correlations proposed in the past for estimating the trough width parameters, x_i for both surface and subsurface settlements.

Figure 2-5 summarizes measurements of the trough width parameters in clays, and sands and gravels based on recent work by Mair and Taylor (1997). These data suggest that the ratio x_i / H is approximately constant for a given soil type, while wider settlement troughs occur in clays ($x_i / H = 0.5$) compared to sands ($x_i / H = 0.35$).

Besides the settlement trough width x_i , the volume loss V_L is frequently reported from case studies in the literature (interpreted from the best fit Gaussian curve). Figure 2-7 presents a correlation (Macklin 1999) of the volume loss, V_L in stiff clays with the load factor:

$$LF = \frac{N}{N_C} \quad (2.4)$$

where N is the stability number for the tunnel, and N_c is the limiting stability number from analyses (e.g. Kimura and Mair, 1981).

The surface settlement trough in the longitudinal direction is given by Attewell and Woodman (1982) by integrating the normal distribution curve given in Equation (2.1) along the tunnel driving direction, z :

$$\frac{u_y}{u_y^0} = \left[G\left(\frac{z - z_i}{x_i}\right) - G\left(\frac{z - z_f}{x_i}\right) \right] \cdot \exp\left[\frac{-x}{2x_i^2}\right] \quad (2.5)$$

where $G(\xi)$ is defined by:

$$G(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\xi} \exp\left[\frac{-\beta^2}{2}\right] d\beta \quad (2.6)$$

Current design practice relies heavily on these two empirical curves (Equation (2.1) & (2.5)) to account for the three-dimensional surface deformation characteristics of tunneling-induced surface settlement. These empirical methods provide quick estimates of tunnel-induced transverse and longitudinal deformations with a trough width parameter, and 30-years of experience have been accumulated working with these parameters.

2.4 Analytical Methods for Tunneling Deformation Evaluation

Analytical solutions provide complete and consistent solutions to describe both ground deformations and stress distributions caused by tunneling (Sagaseta et al. 1998). Most analytical solutions are obtained by imposing displacement boundary conditions at circular tunnel cavities in elastic half plane. For circular tunnels, elastic solutions in a half-plane have been obtained for three basic tunnel deformation modes shown in Figure 2-8: 1) uniform convergence (UC), 2) pure distortion (ovalization, with no change in volume of the cavity), and 3) vertical translation, and these modes can be superimposed to produce various shapes of deformed tunnel cavities. Table 2-2 summarizes these analytical solutions derived in

elastic half plane, and they are based on either complex variable formulations or superposition of singularities.

From these analytical solutions, it is generally found that the settlement trough caused by the uniform convergence mode (of tunnel deformations) results in wider surface settlement trough than field observations (Sagaseta 1987). Although the pure distortion mode produces narrower settlement trough than the uniform convergence mode (Verruijt and Booker 1996), the analytical trough width is still wider than in-situ measures (Loganathan and Poulos 1998).

All the presented analytical solutions in Table 2-2 are derived based on the linear-elastic solution in a planar half-space, and ignore the presence of linings or other structures. Ground-lining interactions can only be considered approximately from solutions in full-plane (i.e. deep tunnels) (Bobet 2001; Einstein and Schwartz 1979), or by adjusting artificial parameters (Loganathan and Poulos 1998).

Plasticity has been incorporated in analytical solutions for tunneling using cavity contraction analyses (Yu and Rowe 1999) with Mohr-Coulomb yield criterion in full plane (implying deep tunnels). Their results show soil plasticity affects predictions of ground movements up to one radius beyond the tunnel cavity, thus it was suggested by Whittle and Sagaseta (2001) that analytical solutions based on elastic theory are adequate to predict far-field ground movements due to tunneling. The near-field ground movements due to plasticity, suggested by Pinto (2000), can be approximated using elastic solutions with reduction factors based on specified shear stiffness and strength parameters for the soil mass.

Current analytical solutions are limited in the following aspects for soft ground tunneling: 1) linear elasticity, 2) homogeneous ground condition, and 3) lack of lining. Soil non-linearity and heterogeneity (e.g. soil layers) are difficult to incorporate into pure analytical solutions, and these are important factors for predicting reliably the tunneling-induced ground deformations.

2.5 Two-Dimensional Finite Element Methods for Tunneling

Empirical and analytical solutions provide quick estimates of tunneling-induced ground deformations. However, the empirical methods are based on a limited set of data and should be used with caution when extrapolating to new ground conditions or construction

methods, etc. Similarly, the simplified soil behavior used in analytical solutions may be unrealistic and explain pervasive discrepancies noted in comparisons with empirical settlement troughs.

Non-linear finite element methods offer a general numerical procedure that can account for complex material behavior, geometries, and in-situ conditions (e.g. ground water, anisotropic in-situ stresses, etc.) using appropriate constitutive models for the soil and modeling procedures to represent the excavation and construction sequence.

Two-dimensional plane-strain finite element analyses, which consider the initial and final configuration of tunneled ground, have been extensively used to estimate the transverse settlement troughs caused by tunneling. There are three subtly different techniques used to represent the excavation and lining support in 2-D FE tunneling: 1) load reduction technique, 2) stiffness reduction technique, and 3) displaced boundary technique.

The load reduction method is also known as convergence-confinement method (Panet and Guenot 1982). The method first removes the finite elements representing the excavated ground and applies a set of equilibrating forces to the nodes on the tunnel cavity (such that no displacements occur). These equilibrating forces are then reduced with a prescribed reduction factor, β . Lining elements are then activated, and the remaining equilibrating force is reduced to zero (generating stresses in the lining system). The prescribed load reduction factor can be estimated from prior experience of similar projects (Bernat et al. 1997; Canetta et al. 1996), from back analyses, or from three-dimensional finite element analyses by matching calculated and measured displacements. The volume loss method (Addenbrooke et al. 1997) is a special load reduction technique which involves unloading to prescribed volume loss (defined by the ratio between the volume of settlement trough and the removal volume of the tunnel cavity).

The stiffness reduction technique (Swoboda 1979) reduces the stiffness of the ground being excavated in the finite element mesh with a certain prescribed reduction factor, α . After reaching the targeted stiffness, the elements representing the excavation are removed and lining elements are activated. In order to determine the reduction factor for the stiffness before linings are activated, empirical charts (Vassilev and Hristov 1988) were prepared for

practical tunnel designs based on a three-dimensional finite element analyses. The stiffness reduction technique has also been used to prepare critical deformation design charts (Wagner and Schuler 1988). This technique has been used for modeling Heathrow trial tunnel with staged excavation by Atzl and Mayr (1994) with 20% of stiffness reduction prior the lining activation.

Schweiger et al. (1997) have compared the load reduction (β) and stiffness reduction (α) techniques, and established an equivalence between these two methods for full-face excavation in a linear elastic material (where the gravity-induced stress gradient is negligible).

$$\alpha = \frac{1}{2.5} \cdot \frac{\beta}{1 - \beta} \quad (2.7)$$

The results for multi-staged excavations and for non-linear material behaviors, however, are quite different (Oetl et al. 1998). The main difference between load reduction and stiffness reduction is the contribution of the mass in the excavated core. For load reduction technique, the mass is completely removed, while the mass of the core is retained with the stiffness reduction technique.

The displaced boundary technique prescribes a displacement boundary condition at the tunnel cavity (similar to analytical solutions). After imposing the prescribed displacement and achieving equilibrium, lining elements are activated, and the soil-structure interaction starts. Based on gap parameters (Rowe et al. 1983) and refined by three-dimensional finite element analyses (Lee et al. 1992), a procedure can be established (Rowe and Lee 1992) to prescribe displacement boundary conditions. Commercial programs such as Plaxis also provide the capability to specify a uniform contraction parameter that prescribes a uniform radial displacement to simulate the ground loss.

All these three techniques for plane-strain tunneling analyses have been employed for practical engineering analyses successfully with carefully chosen “reduction factors” or “gap parameters”. These factors, however, require prior knowledge or back analyses, that can substantially affect the calculated ground deformation.

The use of any one of these three approximation techniques depends on the available capabilities of the FE program. The load reduction technique is most conveniently applied

in finite element programs that provide manual time-stepping controls for each individual step. This capability is usually provided to control Newton-Raphson solution scheme for non-linear analyses due to the finite convergence radius of the scheme. The time-scale used in such scenario usually does not have physical meaning, but is instead a pseudo quantity that controls the amount of out-of-balance force in the system to be solved. By using manual time stepping control, a prescribed terminating time corresponds to the load reduction factor can be specified to terminate an excavation step, and then proceed to the lining activation step.

The stiffness reduction technique can be employed by changing material properties in the excavated area, or by overlapping two layers of elements with initial and final material properties. The change of material properties can then be simulated by activating and removing the layer of elements. As a result, this technique can be readily used by most finite element packages that provide capabilities for addition or removal of elements.

The displaced boundary technique can be used universally by all displacement based finite element programs. However, prescribing a reasonable deformed tunnel cavity is more difficult than either load reduction or stiffness reduction method.

Soils characteristically exhibit non-linear stress-strain behavior even at relatively small shear strains (Burland 1989). Their stiffness can depend on state of stress, strain level, direction of shear, stress history, etc. Hence, realistic modeling of soil behavior remains a challenging task. The effect of specific aspects of material behavior (linear elasticity, perfect plasticity, anisotropic stiffness, small strain non-linearity, and strength) has all been studied in the context of tunnel induced ground deformation using 2-D FE analyses. It is generally agreed that deformations calculated with homogeneous linear-elasticity produce surface settlement troughs that are shallower and wider than measured field behavior (as presented by empirical normal distribution troughs, Equation (2.1)).

Constitutive models for modeling the soil behavior have pronounced effect on the calculated ground movements using finite element method. Linearly-elastic perfectly-plastic material models, are frequently employed in finite element models. Oettl et al. (1998) have compared predictions of the transverse settlement trough, Figure 2-9, for a subway tunnel

(roughly 6.75m in diameter with 20m overburden) in Vienna, Austria using a variety of simple elasto-plastic models with parameters listed in Table 2-3 and initial $K_0 = 0.6$. The excavation was done in three stages, and elastic linings are activated at the end of each stage with full stress release. The results in this particular tunneling configuration suggest: 1) different yield functions in shear (MC vs. DP) without hardening have negligible effect on the shape of the settlement trough. 2) The effect of yielding in shear narrows the settlement trough slightly. 3) The addition of cap yielding has a pronounced effect on both magnitude and shape of the surface settlement trough. However, cap-yielding may in fact producing an unrealistically wide settlement trough, since it is wider than the trough computed using linear elasticity that is already known to be shallower and wider than the measured field behaviors.

Soil shear stiffness properties generally play a more important role in the computed surface settlement trough (yielding can be quite localized around the opening, while the stiffness properties describe the entire mass behavior). It has been confirmed numerically (Lee and Rowe 1989; Simpson et al. 1996) and analytically (Chatzigiannelis and Whittle 2003) that incorporating anisotropic stiffness reduces the width of the surface settlement trough. Based on a numerical study for Jubilee Line Extension in London, Addenbrooke et al. (1997) shows that small-strain non-linear stiffness is also important to produce realistic transverse settlement troughs when tunnel excavation is represented by the volume loss method (and is necessary to match the measured volume loss). They also reported that stiffness anisotropy has negligible effect once the non-linear stiffness is modeled.

The volumetric behavior of soil also has an influence on the calculated surface settlements. Based on model tests and three-dimensional finite element calculations for lowering-basement problems (three-dimensional trapdoor) (Nakai and Xu 1995), it was concluded that soil volumetric behavior dictates that differences on the shape of settlement troughs in clay (contraction) and in sand (dilation) as shown in Figure 2-10(a).¹

¹ Here we have discussed several attempts to understand the effect of constitutive modeling of soils on improving the prediction of surface settlements due to tunneling. Appendix A presents a different approach using one of the most advanced soil model: MIT-S1 and compares with analytical elastic solutions to understand the effect of constitutive models on the predicted surface settlement due to tunneling.

Tunneling is intrinsically a three-dimensional problem with three-dimensional ground deformations, stress redistribution, and flow around the tunnel heading. Approximation techniques for 2-D plane strain analyses are highly empirical, do not necessarily correctly account for the three-dimensionality, and require major judgments to determine appropriate approximation factors that incorporates three-dimensional effect, workmanship, physical gaps, etc. as discussed by Lee et al. (1992). These approximations cannot describe correctly the transition from the undisturbed state to the final configuration. For example, Eberhardt (2001) shows a computed model of the stress patterns and principal stress rotation for a finite element in an elastic medium due to advancing tunnel heading (Figure 2-11). As a result, incorporating elaborate soil models into 2-D plane strain analyses does not necessarily generate more realistic predictions.

Predictions of strains around the tunnel heading will clearly be affected by inelasticity, path dependent material modeling. For example, as the material yields, the yield surface evolves with the change of strains or stresses, and the evolution of yield faces can be very different between three-dimensional and plane-strain analyses. Nakai and Xu (1995) conducted both two-dimensional (plane-strain) and full three-dimensional finite element analyses with constitutive models t_{ij} -sand and t_{ij} -clay (Nakai et al. 1995), both include hardening behavior of soils. Their results show significant differences between plane-strain models and three-dimensional models as shown in Figure 2-10(b), while the normalized settlement trough shown in Figure 2-10(a) are quite similar except for the case of sandy material with 20cm of basement-lowering.

Ohtsu et al. (1999) made comparisons between plane-strain and three-dimensional finite element analyses with coupling of deformations and flow. For elastic material, they found that stress paths in the ground are different between 2-D and 3-D models, Figure 2-12 (The Drucker-Prager failure envelope in Figure 2-12 is not used in the analyses, but rather a check of yield condition for potential problems). It can be seen from Figure 2-12 that if elasto-plastic analyses were performed, the stress paths from 2-D FE analyses would have satisfied the yield condition immediately after excavations and plastic deformation would have occurred, while 3-D FE analyses remain in elastic regime due to drainage from the tunnel face. Therefore, the difference between 2-D and 3-D analyses is dependent on the permeability of soils and advance rate of tunneling operations (Ohtsu et al. 1999).

Three-dimensionality in elasto-plastic analyses is also important for constitutive laws without hardening in deformation predictions. In undrained conditions, Lee and Rowe (1990b) found the transition from tunnel heading to steady-state plane-strain condition for unlined tunnels depends on the amount of plasticity, as shown in Table 2-4. The more yielding that occurs (higher value of face stability number, N) the longer length of tunnel is needed to reach steady state conditions.

In conclusion, plane-strain models and three-dimensional models can yield different solutions for the following conditions:

1. Stress-flow coupled analyses: the differences are caused by the different drainage conditions. The difference is further increased by the presence of plasticity due to different effective stress paths between plane-strain and three-dimensional analyses.
2. Elasto-plastic analyses without hardening: the differences of deformations between plane-strain calculation and three-dimensional steady state deformation are unknown. However, the transitional zone, which cannot be accurately described by plane-strain analyses, between the tunnel heading and the steady state zone are lengthened by the occurrence of yielding in the soil.
3. Elasto-plastic analyses with hardening behavior: the differences are caused by different yield surface evolution between plane-strain and three-dimensional analyses.
4. It can be further inferred that simulations with material models incorporating strain or stress level dependent stiffness will yield different solutions between plane-strain and three-dimensional analyses.

2.6 Three-Dimensional Finite Element Analyses

Three-Dimensional finite element analyses offer a general technique for modeling tunnel excavation with full considerations of 1) realistic material behavior, 2) complex geometries, 3) soil-structure interactions, 4) flow conditions, and 5) construction methods.

Three-dimensional finite element analyses have been used to understand the complex ground response to tunneling. It has been observed that: 1) the length of transition from

undisturbed ground to plane strain conditions for shallow unlined tunnels is lengthened by the plasticity (Lee and Rowe 1990a; Lee and Rowe 1990b) in undrained conditions. (i.e. tunneling in low permeability clays) 2) Principal stress rotates as tunnel heading advances, and the principal stress directions affect plastic strains in anisotropic soils (Whittle 1987) or fracture initiation in brittle soils (Eberhardt 2001). 3) The stress changes of the ground around tunnel headings are further complicated by the presence of flows (Ohtsu et al. 1999).

The effects of modeling linings in three-dimensional analyses have also been investigated. Traditionally, linings are modeled using shell or plate elements, which give direct output of bending moments (used to design the lining). Shell elements, however, may produce unrealistic reductions in settlements behind the tunneling face, and continuum elements are preferred (Augarde et al. 1999). The performance of continuum elements is validated (Augarde and Burd 2001) by comparing to analytical solutions (Einstein and Schwartz 1979).

Three-dimensional finite element analyses have also been applied to model face reinforcement using soil nailing (Ng and Lee 2002; Yoo and Shin 2003), with soil nails represented by structural elements. It was recognized that soil nails reduce the excess pore pressure generation and yielding of soils at the tunnel headings. By improving the face stability, they also reduce settlement and volume losses. The amount of reduction increase with the axial rigidity of soils nails, nailing density, and nail length, but the efficiency diminishes after rigidity reaches a critical value.

Modern tunneling techniques such as shield tunneling usually incorporate active face supports in order to minimize ground loss caused by ground deformation associated with tunnel face intrusion. However, these construction technologies are also difficult to incorporate realistically in numerical models. Akagi and Komiya (1996) proposed the concept of “excavating elements” to represent the disturbed materials in front of the shield machine using remeshing procedures to rearrange finite elements in front of the shield machine in order to avoid numerical inaccuracies associated with highly distorted elements. Although this procedure is physically reasonable, the selection of material properties for the excavating elements can only be achieved through back-fitting to measured data. As a result, the procedure cannot be used in predicting ground movements. The procedure was later

extended by Komiya et al. (1999) with details such as of jacking forces, and self-weight and stiffness of shield machines.

To evaluate the influence of tunnel excavations on existing buildings, full finite element models including buildings and tunnels without linings (Burd et al. 2000) and with linings (Mroueh and Shahrour 2003) have been used, and significant interactions between tunnels and buildings have been observed. Even though the magnitude of settlement trough is not very different from green-field condition (ignoring the existence of buildings) and the far field movement is similar, the surface distortion is increased significantly at the vicinity of the building due to the constraint imposed by the building. Burd et al. (2000), based on fully three-dimensional finite element models, suggested the soil-structure interaction between the ground and the building is less important for buildings subjected to hogging than sagging deformation. Furthermore, decoupled analyses using green-field ground deformations give reasonable prediction for buildings in the hogging mode, while giving overly conservative damage assessment for buildings in the sagging mode.

The effects of tunneling on existing pile foundations have also been investigated using three-dimensional finite element analyses (Mroueh and Shahrour 2002). The interactions between a tunnel and a single pile, a tunnel and a 4-pile group without cap, and a tunnel and 4-pile group with cap were considered, and the pile deflections were evaluated. They found that the tunnel has the largest influence when the pile tip is just below the tunnel invert. Free-field calculations (absence of the piles) give conservative estimates for the deflection and bending moments induced.

Other applications of three-dimensional finite element analyses include complex geometries with two tunnels intersecting at 45° (Tsuchiyama et al. 1988); complex ground-support interactions including fore-pole, shotcrete, steel ribs, and rockbolt (Aydan et al. 1988).

Three-dimensional finite element models are inevitably very large due to a) the physical dimensions must extend far from the tunnel to handle non-linear soil properties, b) the tunnel diameter is generally smaller than its length, while more refined discretization is needed around the tunnel cavity to model accurately the changes of stresses. As a result,

three-dimensional finite element meshes lead to a large linear system of unknowns that is both memory and computationally intensive.

Figure 2-13 shows a three-dimensional finite element mesh used for an analysis of deformations around a cylindrical tunnel. Symmetry is used to halve the model size, which consists of 8010, 20-noded brick elements, (36,030 nodes, and 108,090 degrees of freedom). Table 2-5 summarizes both the run time and storage requirement for this particular finite element model with 31 steps of unsupported excavation in elastic or elasto-plastic, cohesive soil. Elastic analyses for this problem requires 4.67-5.20 hours of CPU time (depending on the numerical integration scheme), while comparable elasto-plastic calculations (with von Mises yield criterion) require 24.5 – 27.5 hours (i.e. 5.25 times as long as elastic calculations). Due to the long computation time, most 3-D finite element analyses for tunnel simulations published in the literature consider highly simplified excavation sequences (e.g. Burd et al. 2000), with relative crude meshes for 3-D analyses.

Mestat et al. (2004) compiled a database MOMIS, listing size of finite element models used for simulating underground constructions between 1975 – 2002, Figure 2-14. They suggested the average number of nodes used for 2-D FE models is 2500, and 10000 for 3-D FE models. These numbers suggest the relative coarseness of 3-D FE models compared to the 2-D ones. Assuming the nodes are evenly distributed in each dimension, then on average 2-D models have 50 nodes for each dimension ($50^2 = 2500$), while 3-D models have less than 22 nodes per dimension, less than half of the 2-D models.

In conclusion, currently computation time for three-dimensional tunneling analyses is still too lengthy to be practical, and prohibits the adoption of advanced soils models and more accurate modeling.

2.7 Summary

- Reliable predictions of tunneling-induced ground deformations are very important in the urban environment in order to avoid or mitigate potential damage to existing buildings and utilities.

- Current design practice relies on empirical methods to estimate ground deformations caused by tunneling. However, critical parameters such as the trough width parameter x_i (Figure 2-5) are difficult to estimate reliably as they are controlled, in large part, by details of the soil profile and construction methods that are used.
- Analytical methods provide quick preliminary estimates to tunneling induced ground deformations based on simplified models of soil behavior. These solutions provide a more comprehensive model for describing the distribution of ground movements, but also lack predictive power as ground loss parameters must be calibrated to field data.
- The non-linear finite element method is currently the most general solution technique that can cope with complex ground condition, soil behavior, soil-structure interactions, complex geometries, etc.
- Plane-strain (2-D) finite elements models are frequently used for estimating tunneling-induced ground deformations. However, these models require gross simplifications (load reduction, stiffness reduction, or prescribed displacements) to represent stress changes around the tunnel heading, support condition, etc. The accuracy of predicted movements is then largely a function of these ad hoc approximation factors, which again must be calibrated to field measurement.
- Three-dimensional finite element analyses require the least number of assumptions for analyzing tunneling deformations. However, the required computation resources prohibit such calculation used in standard practice. Few researchers have adopted three-dimensional models, but usually involve simplified soil behaviors and excavation procedures to make these 3-D analyses affordable.

Table 2-1 Empirical Relationships for Estimating Transversal Surface Settlements

Surface Settlement		
Atkinson and Potts (1977)	$i_0 = 0.25(z_0 + 0.5D)$	Loose sand with surcharge
Atkinson and Potts (1977)	$i_0 = 0.25(1.5z_0 + 0.25D)$	Dense sands and OC clays without surcharge
Clough and Schmidt (1981)	$i_0 = \left(\frac{D}{2}\right) \cdot \left(\frac{z_0}{D}\right)^{0.8}$	Tunneling in soft ground
O'Reilly and New (1982)	$i_0 = K \cdot z_0$	K 0.4 – 0.7: stiff – soft, silty clays 0.2 – 0.3: granular materials
Loganathan and Poulos (1998)	$\frac{i_0}{D} = 0.575 \left(\frac{z_0}{D}\right)^{0.9}$	saturated clays
Subsurface Settlement		
O'Reilly and New (1982)	$i_z = K \cdot (z - z_0)$	K 0.4 – 0.7: stiff – soft, silty clays 0.2 – 0.3: granular materials
Mair et al.(1993)	$i_z = \frac{0.175 + 0.325 \left(1 - \frac{z}{z_0}\right)}{1 - \frac{z}{z_0}} \cdot z_0$	clays
Moh et al. (1996)	$i_z = bD \left(\frac{z_0 - z}{D}\right)^m$	$m = 0.4$ for sands $m = 0.8$ for clays Figure 2-6
<p>i_0 : the trough width parameter for surface settlement trough estimation z_0 : the depth to the center of tunnel D : the diameter of the tunnel i_z : the trough width parameter for subsurface settlements. z : the distance from ground surface for the subsurface settlements of interest</p> <p>Settlement trough: $\frac{\delta}{\delta_{\max}} = \exp\left(-\frac{x^2}{2i^2}\right)$</p>		

Table 2-2 Analytical Solutions for Evaluating Ground Deformations in Elastic Half Plane

	ν	K_0	UC ⁽¹⁾	PD ⁽²⁾	VT ⁽³⁾	Phy. Tun.	3D	Sol. ⁽⁴⁾	Remark
Sagaseta (1987)	0.5	1.0	v					Ex.	
Verruijt and Booker (1996)	Any	any	v	v				Ap.	
Verruijt (1997)	Any	any	v			v		Ex.	
Loganathan and Poulos (1998)	Any	any	v	v				Ap.	⁽⁵⁾
Pinto (2000)	Any	any	v	v	v	v	v	Ap.	
Pinto (2000)	Any	any	v	v	v			Ex.	
Chatzigiannelis and Whittle (2001)	Any	any	v	v	v	v		Ap.	⁽⁶⁾
Strack and Verruijt (2002)	Any	any	v			v		Ex.	

(¹) Uniform convergence mode
 (²) Pure distortion mode, tunnel deforms without change the volume of cavity
 (³) Vertical translation mode
 (⁴) Solution: Exact or Approximate
 (⁵) Semi-analytical, due to the artificial parameter introduced
 (⁶) Cross anisotropic elasticity (5 parameters)

Table 2-3 Parameters for 2D Plane-Strain Analyses (Oetl et al. 1998)

	Notations	Units	Soil	Shotcrete
Specific weight	γ	kN/m ³	19	25
Young's modulus	E	kPa	59,000	2.8·10 ⁷
Poisson's ratio	ν	–	0.38	0.30
Cohesion	c	kPa	73	–
Friction angle	ϕ	°	20.4	–
Shape parameter	K	–	1.0	–
Shape parameter	α	–	0.001	–
Shape parameter	R	–	0.64	–
Initial cap position	p_b^0 ε_{hol}^{pl}	kPa	80	–
		–	–0.11	–

Note: K , α , R listed are parameters for the yield function: Drucker-Prager-Cap model in ABAQUS; p_b^0 and ε_{hol}^{pl} defines the position of yield surfaces.

Table 2-4 Length of Transition Zone versus Plasticity (Lee and Rowe 1990a)

Stability Number $N = \gamma H / S_u$	Length of transition zone	Maximum surface settlement
0 (elastic)	2.0D	δ_z
2.5	3.5D	1.94 δ_z
3.0	6.0D	3.17 δ_z

Note: calculations based on a 3-D FE model tunnel with overburden depth, $H = 4m$ and tunnel diameter, $D = 2m$,

Table 2-5 Solution Time for a 3D Finite Element Tunneling Analysis

Analyses ⁽¹⁾	CPU time ⁽²⁾ (seconds)	Storage ⁽³⁾ (GB)
Elastic - Reduced (8 Gauss points)	16833 (4 hours, 41 minutes)	1.2
Elastic - Full (27 Gauss points)	18609 (5 hours, 11 minutes)	2.6
Plastic – Reduced (8 Gauss points)	87991 (24 hours, 27 minutes)	1.2
Plastic – Hybrid (27 Gauss points + DOF for pressure)	99187 (27 hours, 33 minutes)	3.2

(1) A total of 108090 unknowns, and 31 steps. Plastic analyses require 5-6 iterations per step.

(2) Digital Personal Workstation 600au, Alpha 21164 600MHz CPU, 1088MB RAM

(3) Exclude temporary storage

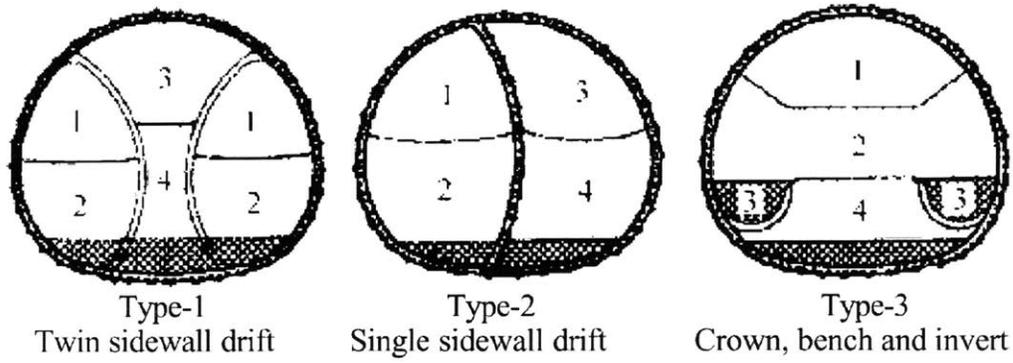


Figure 2-1 Staged Excavation Sequences for Heathrow Express Trial Tunnel
(Karakus and Fowell 2003)

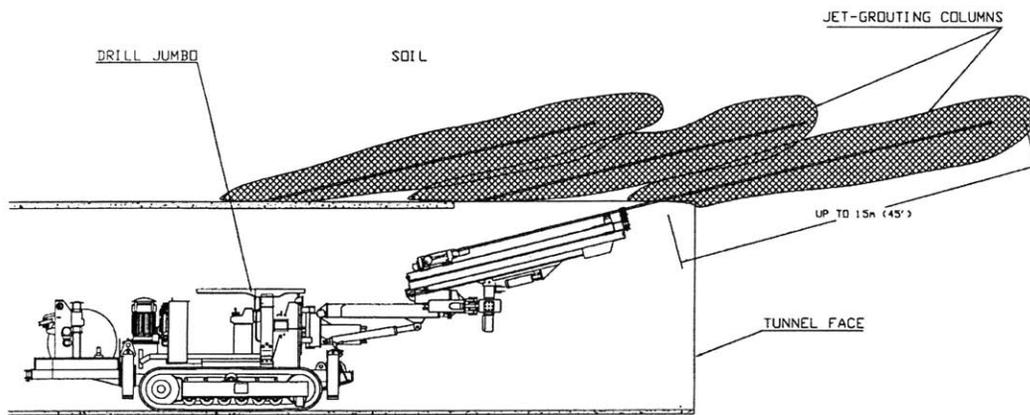


Figure 2-2 Application of Jet Grouting for Tunnel Construction
(Henn 1996)

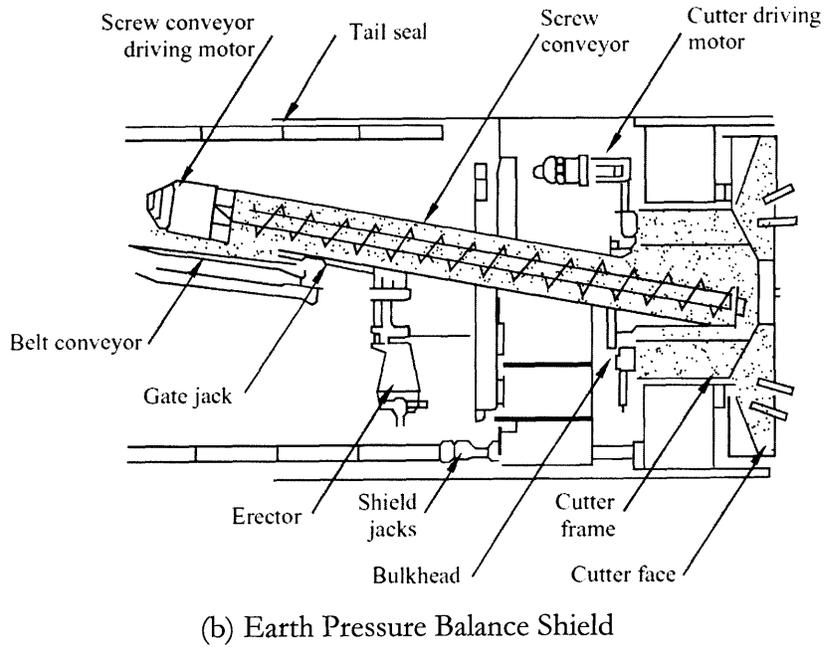
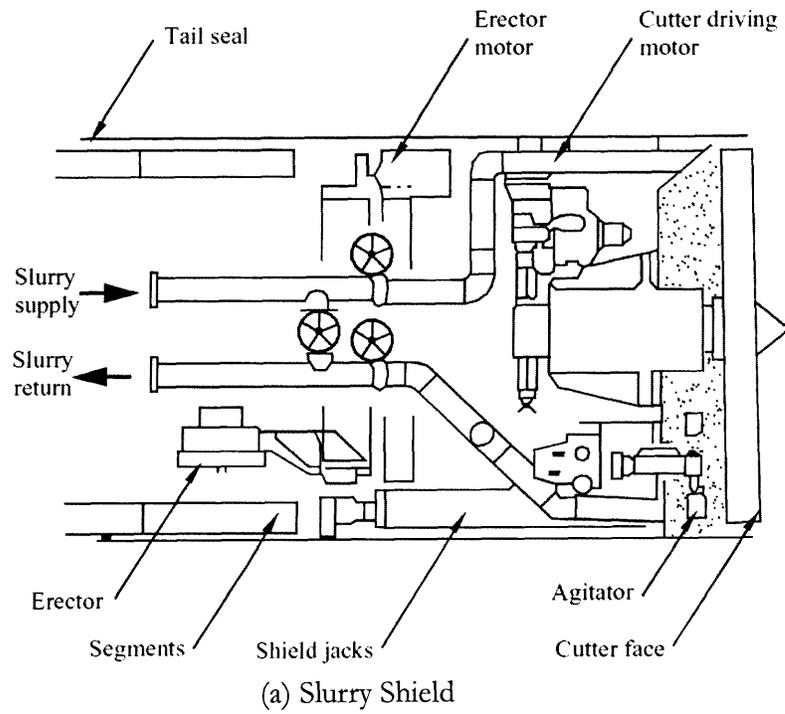


Figure 2-3 Two Types of Closed Face Shield Machines
(After Fujita, 1989)

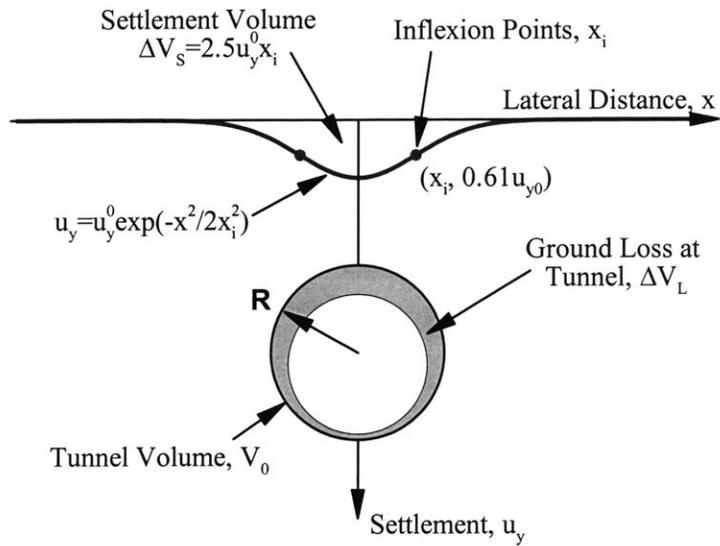


Figure 2-4 Settlement Trough as Normal Probability Function

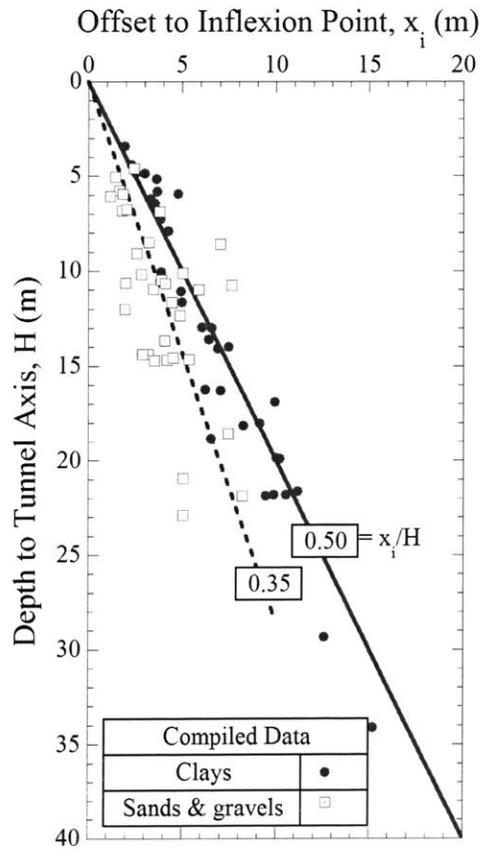


Figure 2-5 Trough Width Parameter vs. Soil Types (Mair and Taylor 1997)

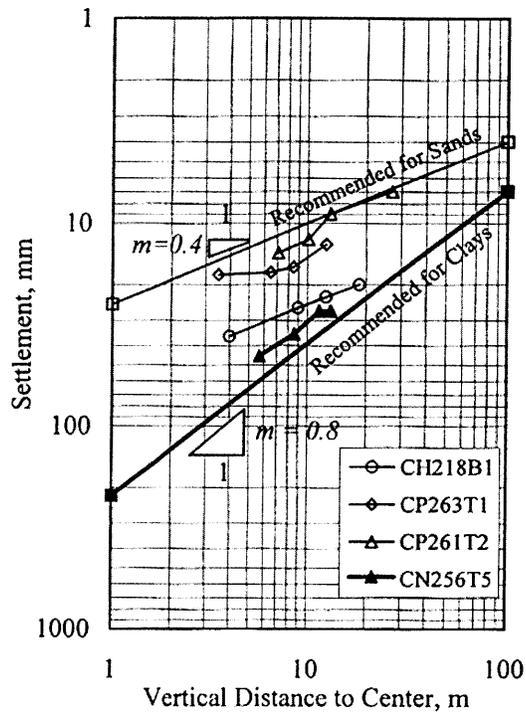


Figure 2-6 Determination of parameter m (Moh et al. 1996)

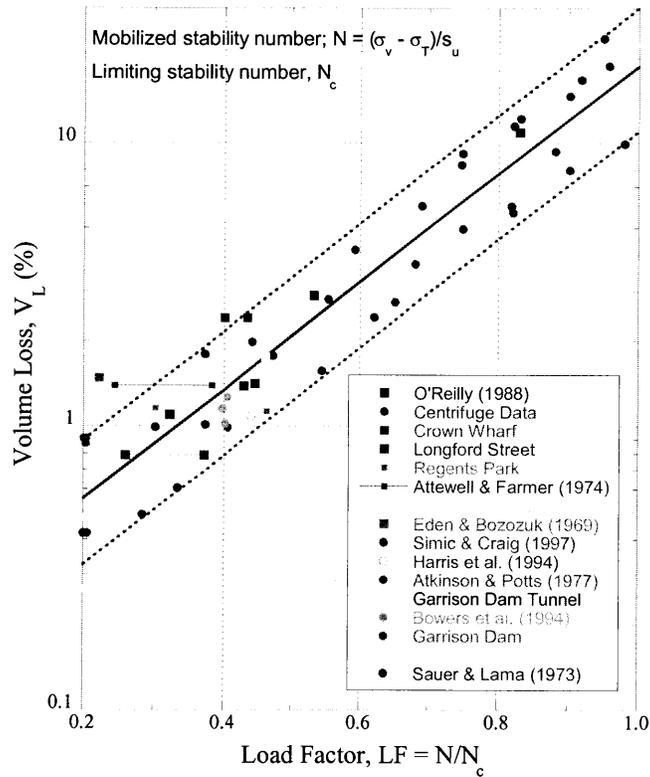


Figure 2-7 Correlations between Ground Loss and Load Factor (Macklin 1999)

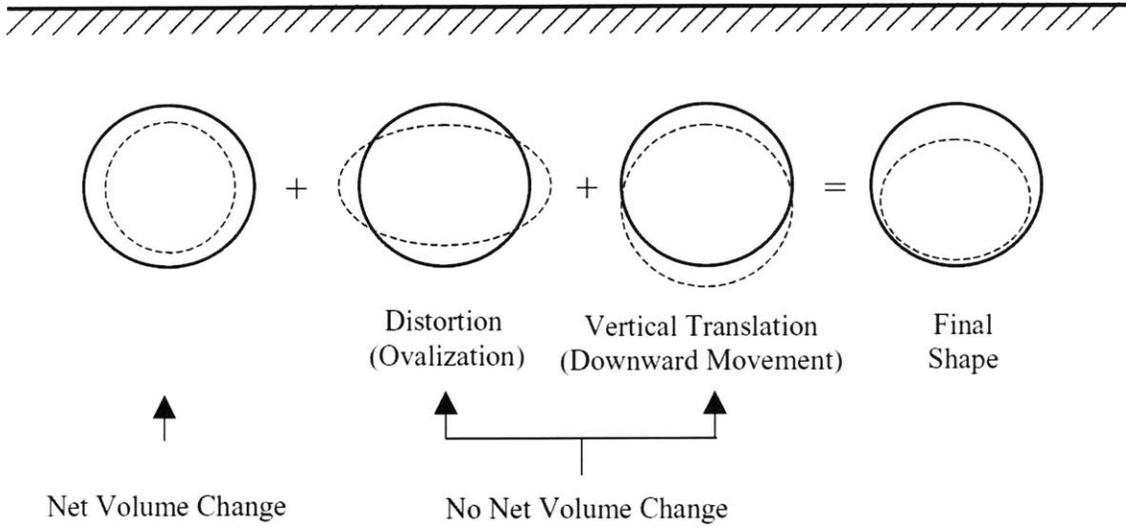


Figure 2-8 Basic Modes of Tunnel Wall Displacements
(Pinto 2000)

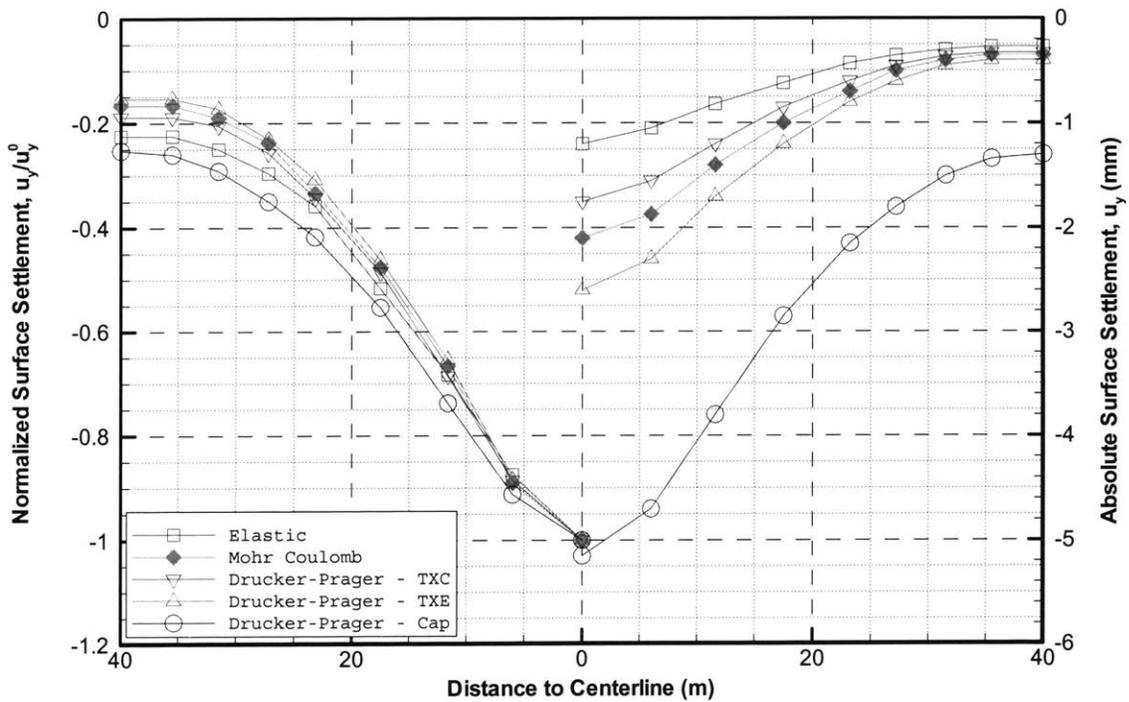
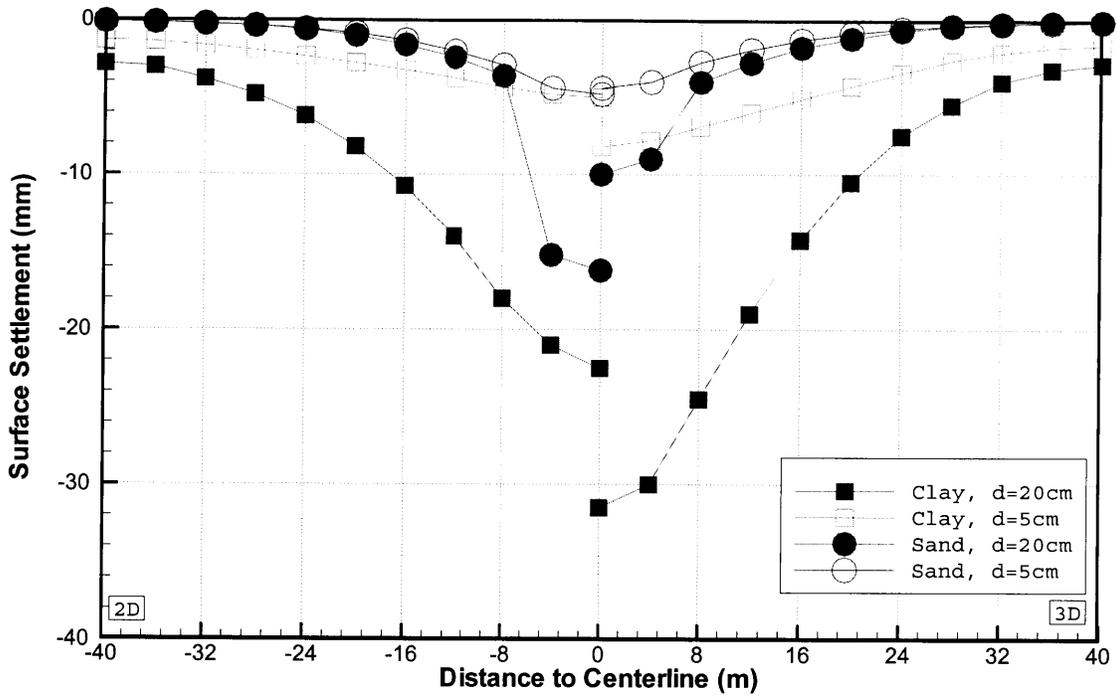
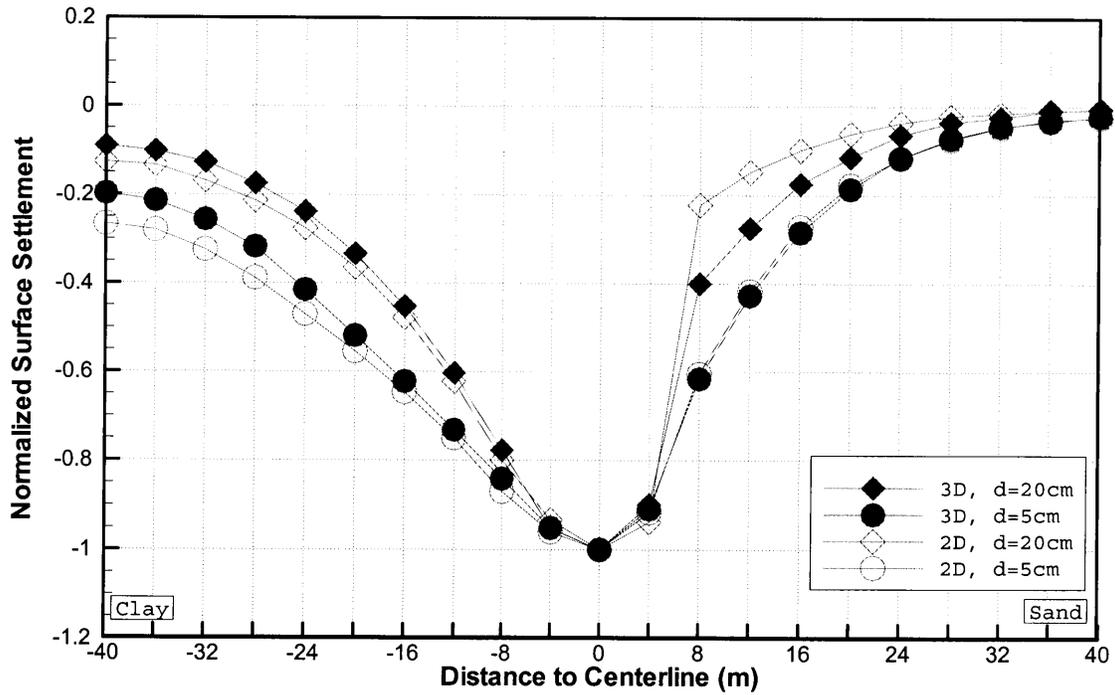


Figure 2-9 Effect of Plasticity Models on Surface Settlement Trough
(Redraw after Oetl et al. 1998)



(a) Absolute Settlement Trough



(b) Normalized Settlement Trough

Figure 2-10 2D vs. 3D Analyses for 3-D Trapdoor Problems
(Redraw from Nakai and Xu, 1995)

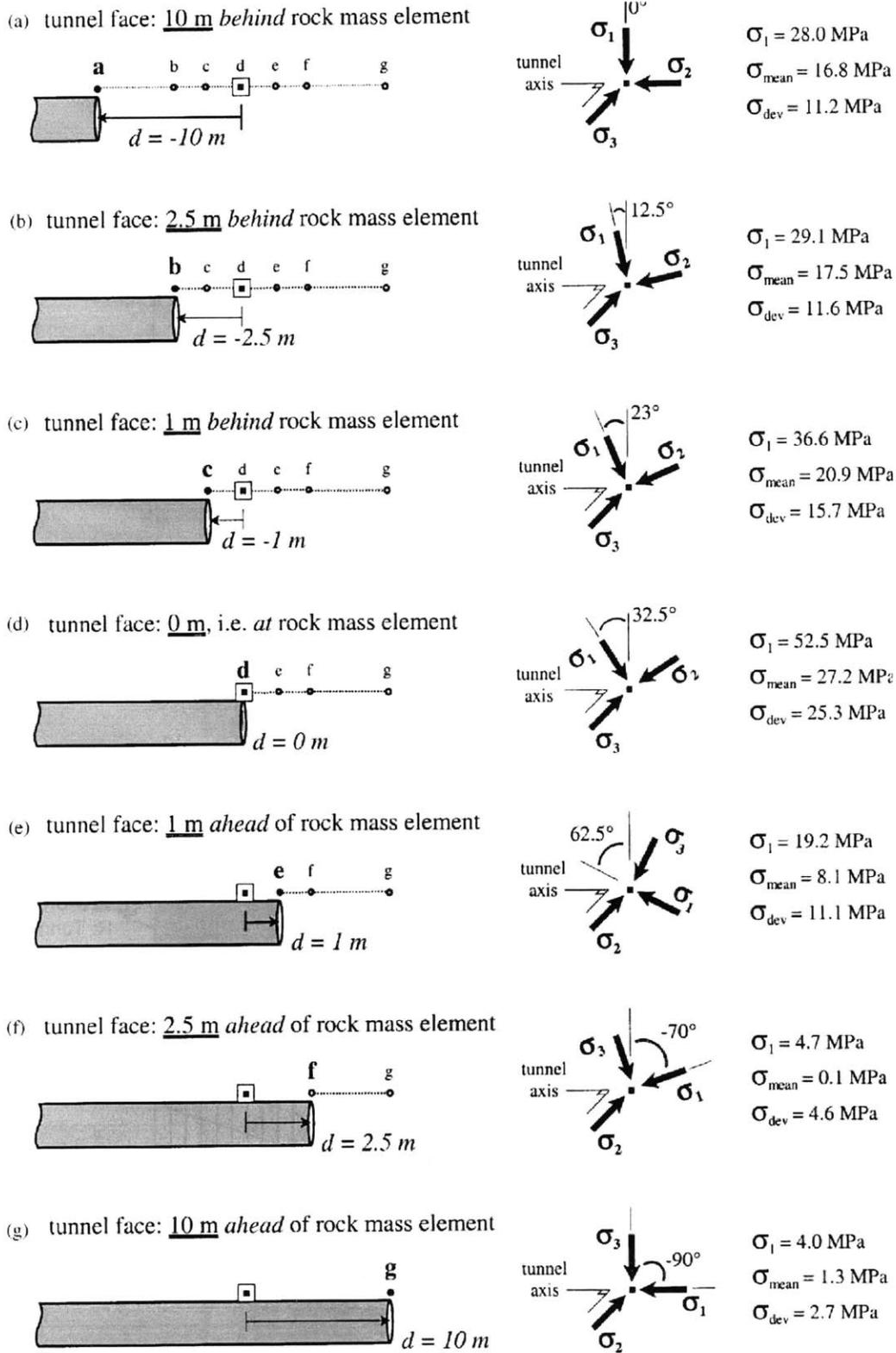


Figure 2-11 Stress Rotation due to Tunneling
(Eberhardt 2001)

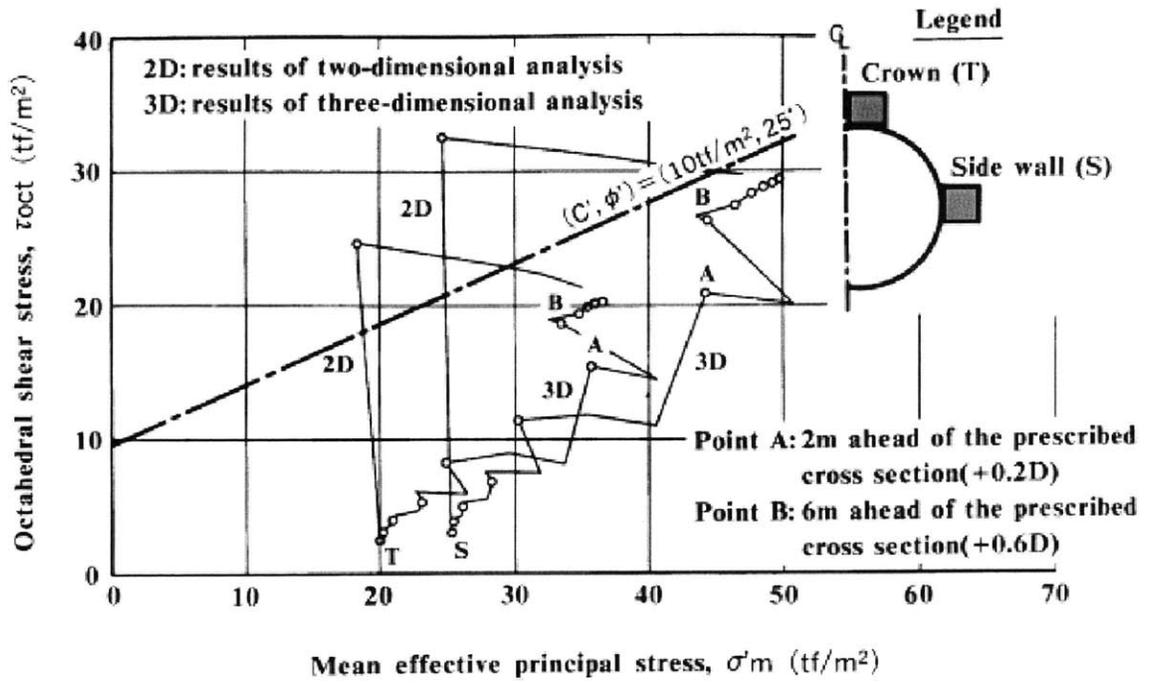


Figure 2-12 Stress Paths for 2D and 3D Finite Element Analyses (Ohtsu et al. 1999)

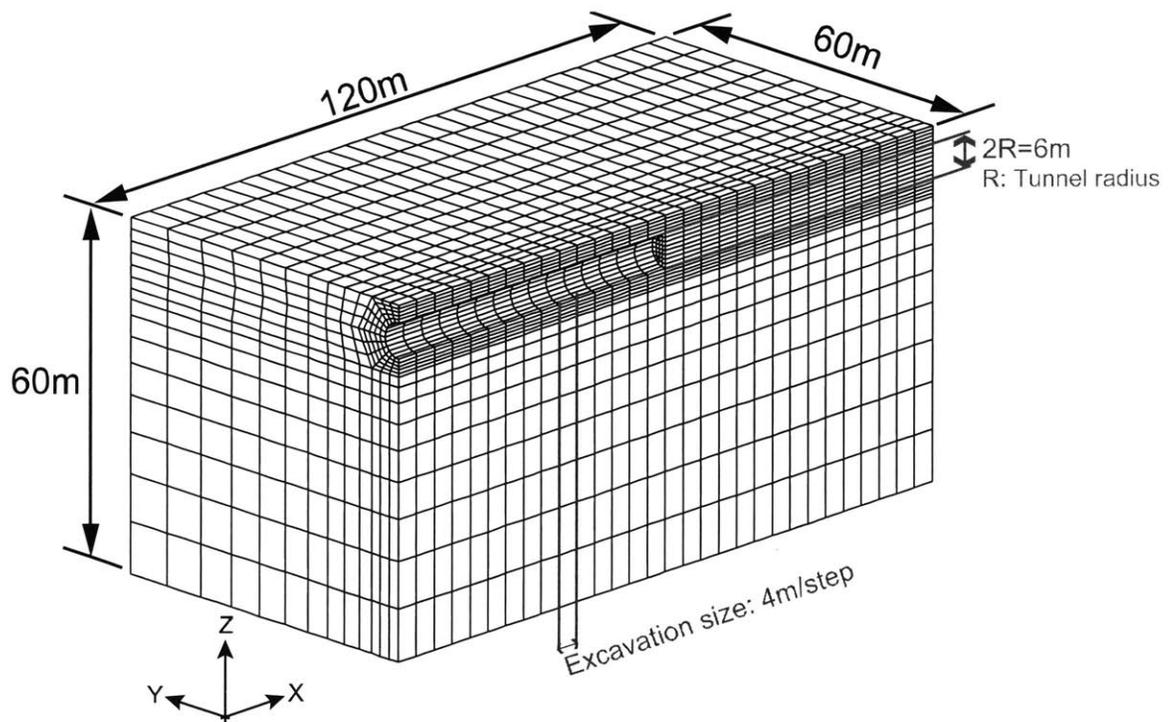


Figure 2-13 Three-Dimensional Finite Element Mesh

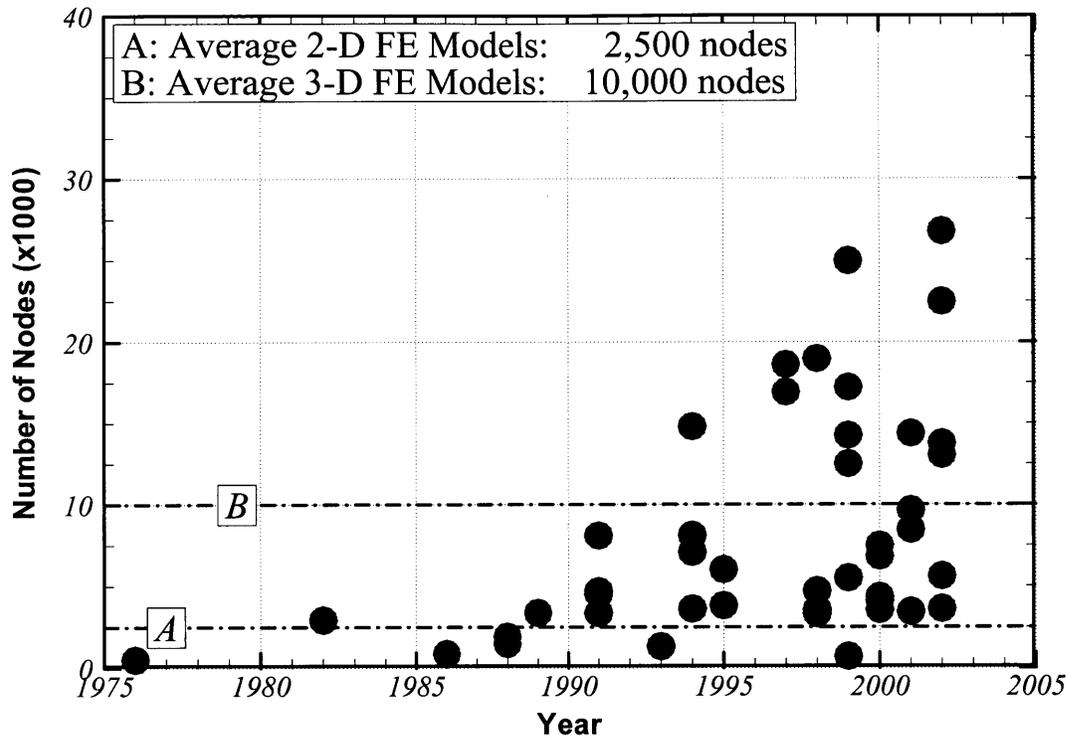


Figure 2-14 MOMIS Database for FE Models in Underground Work
(Mestat et al. 2004)

References

- Addenbrooke, T. I., Potts, D. M., and Puzrin, A. M. (1997). "Influence of pre-failure soil stiffness on the numerical analysis of tunnel construction." *Geotechnique*, 47(3), 693-712.
- Akagi, H., and Komiya, K. (1996). "Finite Element Simulation of Shield Tunnelling Processes in Soft Ground." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 447-452.
- Atkinson, J. H., and Potts, D. M. (1977). "Subsidence above shallow tunnels in soft ground." *Journal of the Geotechnical Engineering Division - American Society of Civil Engineers*, 103(4), 307-325.
- Attewell, A. P., and Woodman, J. P. (1982). "Predicting the dynamics of ground settlement and its derivatives caused by tunnelling in soil." *Ground Engineering*, 15(8), 13-36.
- Atzl, G. V., and Mayr, J. K. "FEM-analysis of Heathrow NATM trial tunnel." *Numerical Methods in Geotechnical Engineering*, Rotterdam, 195-201.
- Augarde, C. E., and Burd, H. J. (2001). "Three-dimensional finite element analysis of lined tunnels." *International Journal for Numerical and Analytical Methods in Geomechanics*, 25, 243-262.
- Augarde, C. E., Wisser, C., and Burd, H. J. (1999). "Numerical Modelling of Tunnel Installation Procedures." *Numerical Methods in Geomechanics*, Pande, Pietruszczak, and Schweiger, eds., 329-334.
- Aydan, O., Kyoya, T., Ichikawa, Y., Kawamoto, T., Ito, T., and Shimizu, Y. (1988). "Three-dimensional Simulation of an Advancing Tunnel Supported with Forepoles, Shotcrete, Steel Ribs and Rockbolts." *Numerical Methods in Geomechanics*, G. Swoboda, ed., 1481-1486.
- Bernat, S., Cambou, B., and Santosa, P. (1997). "Modeling of Soil-Structure Interaction During Tunnelling in Soft Soil." *Computer Methods and Advances in Geomechanics*, 1377-1382.
- Bobet, A. (2001). "Analytical solutions for shallow tunnels in saturated ground." *Journal of Engineering Mechanics*, 127(12), 1258-1266.
- Bowers, K. H. (1997). "An appraisal of the New Austrian Tunnelling Method in soil and weak rock," PhD Thesis, The University of Leeds.
- Burd, H. J., Houlsby, G. T., Augarde, C. E., and Liu, G. (2000). "Modelling tunnelling-induced settlement of masonry buildings." *Proceedings of the Institution of Civil Engineers Geotechnical Engineering*, 143(1), 17-29.
- Burland, J. B. (1989). "Small is Beautiful - the stiffness of soils at small strains." *Canadian Geotechnical Journal*, 26(4), 499-516.
- Canetta, G., Cavagna, B., and Nova, R. (1996). "Experimental and Numerical Tests on the Excavation of a Railway Tunnel in Grouted Soil in Milan." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 479-484.
- Chatzigiannellis, Y., and Whittle, A. J. (2003). "Deformation caused by shallow tunneling in cross-anisotropic soil." *submitted for publication*.
- Clough, G. W., and Schmidt, B. (1981). "Design and performance of excavations and tunnels in soft clay." *Soft Clay Engineering*, Elsevier Scientific Publishing Company, Amsterdam, The Netherlands, 569-634.
- Deming, P. W., Lacy, H. S., and Chang, D. K. "Ground freezing for tunnel face stabilization." *North American Tunneling '00*.

- Eberhardt, E. (2001). "Numerical modelling of three-dimension stress rotation ahead of an advancing tunnel face." *International Journal of Rock Mechanics and Mining Sciences*, 38(4), 499-518.
- Einstein, H. H., and Schwartz, C. W. (1979). "Simplified analysis for tunnel supports." *Journal of the Geotechnical Engineering Division-ASCE*, 105(4), 499-518.
- Fujita, K. "Underground construction, tunnel, underground transportation." *Proceedings 12th International Conference on Soil Mechanics and Foundation Engineering*, Rio de Janeiro, 2159-2176.
- Grant, R. J., and Taylor, R. N. (2000). "Tunnelling-induced ground movements in clay." *Proceedings of the Institution of Civil Engineers Geotechnical Engineering*, 143(1), 43-55.
- Harris, D. I., Pooley, A. J., Menkiti, C. O., and Stephenson, J. A. "Construction of low level tunnels below Waterloo Station with compensation grouting for Jubilee line extension." *Geotechnical aspects of underground construction in soft ground*, 361-366.
- Henn, R. W. (1996). *Practical Guide to Grouting of Underground Structures*, American Society of Civil Engineers.
- Karakus, M., and Fowell, R. J. (2003). "Effects of different tunnel face advance excavation on the settlement by FEM." *Tunnelling and Underground Space Technology*, 18(5), 513-523.
- Kimura, T., and Mair, R. J. "Centrifugal testing of model tunnels in soft clay." *Proceedings of 10th International Conference on Soil Mechanics and Foundation Engineering*, Stockholm, 319-322.
- Komiya, K., Soga, K., Akagi, H., Hagiwara, T., and Bolton, M. D. (1999). "Finite Element Modelling of Excavation and Advancement Processes of a Shield Tunnelling Machine." *Soils and Foundations*, 39(3), 37-52.
- Lee, K. M., and Rowe, R. K. (1989). "Deformations caused by surface loading and tunnelling: the role of elastic anisotropy." *Geotechnique*, 39(1), 125-140.
- Lee, K. M., and Rowe, R. K. (1990a). "Finite element modelling of the three-dimensional ground deformations due to tunnelling in soft cohesive soils. Part 2 - results." *Computers & Geotechnics*, 10(2), 111-138.
- Lee, K. M., and Rowe, R. K. (1990b). "Finite element modelling of the three-dimensional ground deformations due to tunnelling in soft cohesive soils. Part I - method of analysis." *Computers & Geotechnics*, 10(2), 87-109.
- Lee, K. M., Rowe, R. K., and Lo, K. Y. (1992). "Subsidence Owing to Tunnelling. I. Estimating the Gap Parameter." *Can. Geotech. J.*, 29, 929-940.
- Loganathan, N., and Poulos, H. G. (1998). "Analytical prediction for tunneling-induced ground movements in clays." *Journal of Geotechnical and Geoenvironmental Engineering*, 124(9), 846-856.
- Macklin, S. R. (1999). "Prediction of volume loss due to tunnelling in overconsolidated clay based on heading geometry and stability number." *Ground Engineering*, 32(4), 30-33.
- Mair, R. J., and Taylor, R. N. "Bored tunnelling in the urban environment." *14th International Conference on Soil Mechanics and Foundation Engineering*, Hamburg.
- Mair, R. J., Taylor, R. N., and Bracegirdle, A. (1993). "Subsurface Settlement Profiles above Tunnels in Clays." *Geotechnique*, 43(2), 315-320.
- Mestat, P., Bourgeois, E., and Riou, Y. (2004). "Numerical Modelling of embankments and underground works." *Computers and Geotechnics*, 31, 227-236.
- Moh, Z.-C., Hwang, R. N., and Ju, D. H. (1996). "Ground Movements Around Tunnels in Soft Ground." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 725-730.

- Mroueh, H., and Shahrour, I. (2002). "Three-dimensional finite element analysis of the interaction between tunneling and pile foundations." *International Journal for Numerical and Analytical Methods in Geomechanics*, 26(3), 217-230.
- Mroueh, H., and Shahrour, I. (2003). "A full 3-D finite element analysis of tunneling-adjacent structures interaction." *Computers and Geotechnics*, 30(3), 245-253.
- Nakai, T., and Xu, L. (1995). "3D and dilatancy effects on settlements and earth pressures over lowering basement." *Soils & Foundations*, 35(4), 73-84.
- Nakai, T., Xu, L. M., and Yamazaki, H. (1995). "Prediction of Surface Settlement Profiles due to Tunnel Excavation: Model Tests and Numerical Analyses." *Numerical Methods in Geomechanics*, Pande and Pietruszczak, eds., 495-500.
- Ng, C. W. W., and Lee, G. T. K. (2002). "A three-dimensional parametric study of the use of soil nails for stabilising tunnel faces." *Computers and Geotechnics*, 29(8), 673-697.
- Oettl, G., Stark, R. F., and Hofstetter, G. (1998). "Comparison of elastic-plastic soil models for 2D FE analyses of tunnelling." *Computers & Geotechnics*, 23(1-2), 19-38.
- Ohtsu, H., Ohnishi, Y., Taki, H., and Kamemura, K. (1999). "A Study on Problems Associated with Finite Element Excavation Analysis by the Stress-flow Coupled Method." *Int. J. Numer. Anal. Meth. Geomech.*, 23, 1473-1492.
- O'Reilly, M. P., and New, M. B. "Settlements above tunnels in the United Kingdom - their magnitudes and prediction." *Tunnelling*, 173-181.
- Panet, M., and Guenot, A. "Analysis of convergence behind the face of a tunnel." *Tunnelling '82, Papers Presented at the 3rd International Symposium.*, Brighton, Engl, 197-204.
- Peck, R. B. (1969). "Deep excavations and tunneling in soft ground." University of Illinois, Urbana, 225-290.
- Pinto, F. (2000). *Analytical methods to interpret ground deformations due to soft ground tunneling.*
- Rowe, R. K., and Lee, K. M. (1992). "Subsidence Owing to Tunnelling. II. Evaluation of a Prediction Technique." *Can. Geotech. J.*, 29, 941-954.
- Rowe, R. K., Lo, K. Y., and Kack, G. J. (1983). "Method of estimating surface settlement above tunnels constructed in soft ground." *Canadian Geotechnical Journal*, 20(1), 11-22.
- Sagaseta, C. (1987). "Analysis of undrained soil deformation due to ground loss." *Geotechnique*, 37(3), 301-320.
- Sagaseta, C., Verruijt, A., and Booker, J. R. (1998). "Surface settlements due to deformation of a tunnel in an elastic half plane." *Geotechnique*, 48(5), 709-713.
- Schweiger, H. F., Schuller, H., and Pottler, R. (1997). "Some Remarks on 2-D-models for Numerical Simulation of Underground Constructions with Complex Cross Sections." *Computer Methods and Advances in Geomechanics*, 1303-1308.
- Simpson, B., Atkinson, J. H., and Jovicic, V. (1996). "The Influence of Anisotropy on Calculations of Ground Settlements Above Tunnels." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 591-594.
- Strack, O. E., and Verruijt, A. (2002). "A complex variable solution for a deforming buoyant tunnel in a heavy elastic half-plane." *International Journal for Numerical and Analytical Methods in Geomechanics*, 26(12), 1235-1252.
- Swoboda, G. "Finite element analysis of New Austrian Tunneling Method (NATM)." *Proc. 3rd Int. Conf. Numer. Methods Geomech.*, Aachen, 581-586.
- Taguchi, Y., Hada, M., Kagawa, K., and Hara, T. "Soil nailing technique in tunnel support." *Proceedings of the 6th Engineering Foundation Conference on Shotcrete for Underground Support, May 2-6 1993, Niagara-on-the-Lake, Ont, Can*, 158-165.

- Tsuchiyama, S., Hayakawa, M., Shinokawa, T., and Konno, H. (1988). "Deformation Behaviour of the Tunnel Under the Excavation of Crossing Tunnel." *Numerical Methods in Geomechanics*, G. Swoboda, ed., 1591-1596.
- Vassilev, V. H., and Hristov, T. N. (1988). "Influence of the heading face and a two-dimensional calculation model of tunnel linings." *Numerical Methods in Geomechanics*, G. Swoboda, ed., 1551-1555.
- Verruijt, A. (1997). "Complex variable solution for a deforming circular tunnel in an elastic half-plane." *International Journal for Numerical and Analytical Methods in Geomechanics*, 21(2), 77-89.
- Verruijt, A., and Booker, J. R. (1996). "Surface settlements due to deformation of a tunnel in an elastic half plane." *Geotechnique*, 46(4), 753-756.
- Wagner, H., and Schulter, A. (1988). "Geonumerical Computations for the Determination of Critical Deformations in Shallow Tunnelling." *Numerical Methods in Geomechanics*, G. Swoboda, ed., 1531-1536.
- Whittle, A. J. (1987). "A constitutive model for overconsolidated clays with application to the cyclic loading of friction piles," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge.
- Yoo, C., and Shin, H.-K. (2003). "Deformation behaviour of tunnel face reinforced with longitudinal pipes--laboratory and numerical investigation." *Tunnelling and Underground Space Technology*, 18(4), 303-319.
- Yu, H. S., and Rowe, R. K. (1999). "Plasticity solutions for soil behaviour around contracting cavities and tunnels." *International Journal for Numerical & Analytical Methods in Geomechanics*, 23(12), 1245-1279.

Chapter 3

Computational Strategies toward Three-Dimensional Finite Element Analyses

3.1 Introduction

Finite element analysis is an essential tool for engineering design. However, large-scale non-linear finite element analyses are computationally expensive. This cost is expressed in terms of the computational resources and solution time associated with both the large numbers of unknowns and the iterative solutions of material and geometric non-linearities. This chapter identifies key factors affecting the efficiency of large-scale finite element analyses, and describes methods available for practical three-dimensional computations.

3.2 Overview of Finite Element Method

The finite element method is a versatile numerical method for solving partial differential equations. The first step of the method is to discretize or subdivide the problem domain into finite number of simple geometric regions known as elements. Each element is uniquely defined by several points known as nodes. By discretization, the finite element method can reduce a problem with complex geometry into several simple regions. Mathematically, finite element method transforms continuous problems with infinite number of unknowns into discrete problems with a finite number of unknowns.

Once the problem is discretized, the original governing equation may need to be re-written by integration (weak formulation), such that the original point-wise governing equations of the continuum become element-wise governing equations interrelating element-connected nodes. This is usually done by Gauss integration, a form of numerical integration requiring weighted function evaluations at locations, known as Gauss points, on the interior of the elements. With this transformation, some solution variables (e.g. the displacements) are defined at the nodes, while others (e.g. stresses, strains) are defined at integration points within the elements.

The final stage of the finite element formulation is appropriately assembling all elemental governing equations into a system of equations. For static/quasistatic problems, this assemblage is described by the matrix equation:

$$\mathbf{KU} = \mathbf{F} \quad (3.1)$$

In a displacement-based formulation for mechanical systems, \mathbf{K} is referred to as the stiffness matrix, \mathbf{U} is the vector of unknown nodal displacements to be solved to achieve force equilibrium with \mathbf{F} , the residual or unbalanced force vector. Equation (3.1) may need to be solved several times if \mathbf{K} or \mathbf{F} are dependent on the unknown \mathbf{U} .

Figure 3-1 summarizes the steps, associated variables, and their complexities for a typical static/quasi-static non-linear finite element analysis. The shadowed steps may need to be evaluated iteratively for solving non-linear problems. Most steps in Figure 3-1 scale linearly with number of nodes or elements (i.e. the solution time scales up 10 times if the problem size grows 10 times larger), except solving the unknown displacements via $\mathbf{KU} = \mathbf{F}$. The solution time for solving $\mathbf{KU} = \mathbf{F}$ depends largely on the solver employed and the characteristics of the problem, and the scaling varies between linear scaling (tri-diagonal matrices) to cubical scaling (dense Gauss elimination.)

Based on the overview of the finite element procedure, the solution time for the linear system $\mathbf{KU} = \mathbf{F}$ dominates the overall solution time for large-scale finite element models, and it is essential to employ efficient algorithms for solving $\mathbf{KU} = \mathbf{F}$ to obtain efficient finite element solutions. It is equally important to use good solution schemes in order to minimize the number of iterations for non-linear solutions, as each iteration involves solving a linear system of equations.

3.3 Nonlinear Solution Techniques

Finite element analyses for geotechnical engineering applications frequently involve non-linear material behavior. In order to obtain solutions to non-linear global equilibrium

equations, Newton-Raphson (NR) iteration schemes combined with automatic sub-stepping are commonly used in commercial finite element products such as ABAQUS¹ and ANSYS².

The NR iteration scheme with notations for solving mechanical systems is illustrated in Figure 3-2 and the flowchart is presented in Figure 3-3. The mechanical system can be characterized by the internal force vector \mathbf{F}_{int} due to internal stresses, and the external force vector \mathbf{F}_{ext} due to body forces and externally applied loads, where the stiffness matrix \mathbf{K} expresses the rigidity of the system at a given configuration. The goal is to find the displacement that satisfies force equilibrium condition $\mathbf{R} = \mathbf{F}_{int} - \mathbf{F}_{ext} = \mathbf{0}$, where \mathbf{R} is the force residual following the unknown non-linear system response curve. The superscript in Figure 3-2 and Figure 3-3 denotes the iteration number. It is important to store the system configuration (e.g. stresses, strains, state variables) at the beginning of the iteration, as this is the only known valid configuration during the iteration process. The intermediate configurations need to be based on this valid configuration plus the total displacement vector \mathbf{U} , rather than use the previous intermediate configuration plus the incremental displacement correction vector $\Delta\mathbf{u}^i$. This is because the iteration process is essentially a trial-and-error process, and the trial displacements can dramatically deviate from the true response before the equilibrium condition is reached. These trial displacements before convergence can impose incorrect or undesirable strain paths on materials and affect solutions for strain-path dependent materials. Therefore, it is important to restore the previous stored valid configurations before new trial displacements are imposed.

One of the most attractive characteristics of NR iterative procedure is its quadratic convergence rate. The convergence rate b is defined using the following relationship:

$$\mathbf{R}^{i+1} = a(\mathbf{R}^i)^b \quad (3.2)$$

1 A general-purpose commercial finite element package, product of ABAQUS, Inc. <http://www.abaqus.com>

2 A general-purpose commercial finite element package, product of ANSYS, Inc. <http://www.ansys.com>

Figure 3-4 compares the convergence characteristics between linear convergence ($b = 1$) and quadratic convergence ($b = 2$) with $a = 0.5$. With quadratic convergence, the relative residual ($\mathbf{R}^i / \mathbf{R}^0$) reduces to 10^{-10} in only five iterations, while 32 iterations are needed if the convergence rate is linear ($b = 1$). However, quadratic convergence is only achievable when the initial trial is sufficiently close to the true solution, otherwise NR iterations may exhibit slow convergence or even divergence. Therefore, NR iterations are usually combined with heuristics to monitor the progress of NR iterations, identify the undesirable convergence characteristics, and change step sizes accordingly to obtain robust and efficient non-linear solutions.

The quality of the tangent operator (the stiffness matrix \mathbf{K} in mechanical systems) used in NR iterations is also important to its convergence rate, and quadratic convergence is only possible with exact tangent operators. Variants of NR iterations such as BFGS (Broydon-Fletcher-Goldfarb-Shanno, Matthies and Strang 1979) and MNR (Modified Newton Raphson) use approximate tangent operators to save computation costs on forming the exact tangent operator. However, approximate tangent operators can deteriorate the convergence rate ($b < 2$) and hence require more iterations to converge than the full NR iterations.

Simo and Taylor (1985) found that when applying NR in finite element analysis for solving the global equilibrium equation, the calculation of tangent stiffness needs to be consistent with the integration scheme used for stress-update occurring at integration points. At integration points, updating stresses for given strain increments is often accomplished with a backward Euler integration scheme. This scheme is stable and accurate with large step sizes, and always yields a solution regardless the strain step size (Jeremic and Sture 1997). Because backward Euler uses the stiffness evaluated at integration points of the final unknown configuration, inconsistency occurs if the conventional stiffness (evaluated at the initial configuration) for solving global equilibrium equations. The consistent tangent stiffness needs derivatives of the conventional stiffness to approximate/anticipate the stiffness at the final configuration (via Taylor expansion). The derivatives of stiffness, however, are not always easy to evaluate for complex constitutive models.

Apart from NR iterative procedures with automatic sub-stepping, the non-linear global equilibrium equations can also be solved through explicit integration by treating the non-linear problem as a system of ordinary differential equations (Abbo and Sloan 1996). The explicit integration approach, unlike the NR method with automatic sub-stepping, does not correct unbalanced forces at the end of each sub-step, and the unbalanced forces are carried forward to the next sub-step. Therefore, an explicit integration approach generally requires small load increments to control error accumulations. Errors associated with explicit integration can be controlled by automatically adjusting the step size based on local error measures (Abbo and Sloan 1996), and further error reduction can be achieved by one or two NR iterations. The auto-stepping scheme has been successfully applied to elasto-plastic deformation analyses (Abbo and Sloan 1996) and deformation-flow coupled analyses (Sheng and Sloan 2003; Sloan and Abbo 1999).

Table 3-1 compares the two non-linear solution strategies: Newton-Raphson iteration (NR) versus Explicit Integration with Automatic Sub-stepping (EIAS), and the table entries are explained as follows:

1. Purpose of sub-stepping: for NR, sub-stepping is used to make sure the initial estimate of a solution is sufficiently close to the true solution so that the method will converge. EIAS sub-stepping are used to control the error introduced within each substep to avoid unacceptable accumulative error at the end of integration process.
2. Number of sub-steps: NR sub-steps only when necessary (e.g. to get convergent behavior); EIAS needs to make small sub-steps to avoid accumulative error. As a result, NR requires fewer sub-steps than EIAS.
3. Iterations per sub-step: NR uses an iterative procedure within each sub-step to control integration error, and requires at least one iteration step to make sure the error in the sub-step is within a tolerance. It is necessary to limit the number of iterations permitted within a sub-step for NR to make sure the analysis progresses. EIAS controls integration error by sub-stepping, thus does not need any iterative procedure within sub-steps. However, EIAS may include optional iterative steps to reduce the error introduced in a sub-step further.

4. Tangent stiffness: Simo and Taylor (1985) showed that a consistent tangent stiffness matrix is required for NR to have optimal convergence characteristics (e.g. quadratic convergence). EIAS, on the other hand, does not rely on the consistent tangent stiffness. Consistent tangent stiffness requires second-order derivatives of yield functions of material constitutive relationships, and these can be difficult to obtain for complex yield functions, flow rules, etc.
5. Sub-step size: Because the NR method controls error by iterative procedure within each sub-step, while EIAS controls error by controlling sub-step sizes, the NR method can use larger sub-step sizes than EIAS method.
6. Sub-step size adjustment: After the current sub-step is completed, both NR and EIAS methods may adjust the sub-step size. NR adjusts the sub-step size based on the observed number of iterations needed for convergence. EIAS adjusts the sub-step size based on the estimated local error.

NR and EIAS methods are two non-linear solution methods with different error-control strategies. NR relies on iterative procedure to obtain acceptable error within each sub-step, and reduces sub-step sizes when iterative procedure fails. EIAS relies on using small sub-step sizes to avoid error accumulation associated with explicit integration.

One clear advantage of EIAS over the NR method is EIAS does not require consistent tangent stiffness matrix, thus is easier to use with complex/advanced constitutive models that may have yield functions that are difficult to obtain second-order derivatives. In terms of performance, there is no clear advantage or evidence suggesting that one is superior to the other.

3.4 Direct Solution Techniques

3.4.1 Dense Gauss Elimination

One of the most robust and widely used methods for solving a linear system of equations of the form $\mathbf{KU} = \mathbf{F}$ is the direct method, also known as Gauss elimination, which consists of two phases: forward elimination and backward substitution. Figure 3-5 demonstrates solving $\mathbf{KU} = \mathbf{F}$ with Gauss elimination. Gauss elimination utilizes the property that the linear

combinations of equations within the system does not alternate the solution to the system of equations, thus the row operations shown in Figure 3-5 do not change the solution to the system of equations.

It is seen from Figure 3-5 that the calculation of Gauss elimination starts by a forward elimination process, which zeros out all entries below the diagonal of the matrix \mathbf{K} , and the coefficients used to eliminate rows are recorded in a lower triangular matrix (b. in Figure 3-5). Since the entries below the diagonal are all zero after the elimination is complete, the storage space is used to store the lower triangular matrix of coefficient instead of using separate memory space. Figure 3-6(a) is the algorithm for performing forward elimination. The process is also known as LU factorization.

Once the forward elimination process is completed, the solution to $\mathbf{KU} = \mathbf{F}$ can be obtained by first performing a forward elimination on \mathbf{F} with coefficients stored in the lower triangle matrix (c. in Figure 3-5), and then followed by the backward substitution process (d. in Figure 3-5) on \mathbf{F} with coefficients stored in the upper triangle matrix. The backward substitution process effectively eliminates entries above the diagonal of the \mathbf{K} matrix and turns it into an identity matrix, as shown in e. of Figure 3-5. Figure 3-6(b) is the pseudo code for the solution process.

From Figure 3-6(a), it can be concluded that the total number of floating-point operations, $OPS_{factorization}$, need for a square matrix of size N is (1 operation from line 3, and 2 operations from line 5 of Figure 3-6(a)):

$$OPS_{factorization} = \frac{2}{3}N^3 - \frac{1}{2}N^2 - \frac{1}{6}N \cong \frac{2}{3}N^3 \quad (3.3)$$

Similarly, for solving unknown \mathbf{U} with factorized \mathbf{K} with Figure 3-6(a), the total floating-point operations, OPS_{solve} , required based on Figure 3-6(b) (2 operations from line 3, 1 operation from line 7, and 2 operations from line 9) is:

$$OPS_{solve} = 2N^2 - N \quad (3.4)$$

3.4.2 Performance Considerations

The memory hierarchy of modern computer hardware plays an important role in the efficiency of computations, especially the ones with heavy memory references such as Gauss elimination. Modern processors are very fast at doing computations, but often stall when accessing data for the computation or instructions that reside in the main memory¹. Thus, modern computers are equipped with cache memories, which are small and high-speed memories that reside between the processor and the main memory. The cache memory stores data that is frequently used or anticipated by the processor to provide fast access to the needed data; this improves the performance of the computing system. Even though the existence of cache should be transparent to programmers, great benefit can be achieved by explicitly exploiting the memory hierarchy.

Figure 3-7 shows the pronounced effect of the memory hierarchy on the performance of Gauss elimination using Linpack² benchmark on various processors. The horizontal axis of Figure 3-7 is the size of memory needed to store the matrix at double precision, and vertical axis is million floating-point operations per second, thus a measurement of the processor's performance. It is seen from Figure 3-7 that most processors show increasing performance with growing matrix size until the matrix saturates the L2 cache, except PPC740/750 processor that shows decreasing performance after L1 cache is saturated. Therefore, it is necessary to consider the memory hierarchy when the application performance is important.

BLAS (Basic Linear Algebra Subprograms, Lawson et al., 1979) is a library for numerical calculations that involve vector and matrix operations. BLAS defines three levels of essential application programming interfaces (API) that perform 1) vector-vector operations; 2) matrix-vector operations; and 3) matrix-matrix operations that have both generic implementations³ and vendor-supplied implementations. The vendor supplied BLAS

¹ The main memory today refers to the DRAM installed on the motherboard.

² Developed by Jack Dongarra, Jim Bunch, Cleve Moler and Pete Stewart. 1 Feb 84 Available at <http://www.netlib.org/linpack/>

³ Available at <http://www.netlib.org/blas>

implementations are usually highly optimized using assembly language for their processors, and therefore should be used whenever possible. Alternatively, a freely available self-tuned BLAS known as ATLAS¹ can be employed.

Figure 3-8 shows three different variants of Gauss elimination (Dongarra 1997) algorithms without pivoting using BLAS-1, BLAS-2, and BLAS-3. These variants perform exactly the same mathematical operations, except the order is alternated. Dramatic differences in the performance of these different implementations can be seen in Figure 3-9, which was obtained by running the four algorithms shown in Figure 3-6 and Figure 3-8 on a Pentium III-1GHz computer with Intel Math Kernel Library². It is seen that as higher-level BLAS subroutines are used, the higher performance the code delivers. This is because high-level BLAS subroutines have more data to compute, and therefore have more freedom to rearrange the data access pattern to performance calculations, and therefore, better use of the cache memory.

In conclusion, an efficient high-performance programs that process large amounts of data on modern computers require considerations of how to exploit the memory hierarchies via data locality.

3.4.3 Sparse Gauss Elimination

Dense Gauss elimination is a prohibitively expensive algorithm for solving large problems of size N due to its $O(N^3)$ computational cost. However, the finite element method often produces very sparse matrices for large problems. For example, a three-dimensional finite element mode with 100,000 nodes contains up to 300,000 unknowns, and results in a square matrix equation of dimension up to 300000×300000 . To store the entire dense matrix in double precision, 670GB of memory is needed, and is impractical to solve even with today's

¹ ATLAS self-tunes several memory-hierarchy related parameters in its algorithm during the compilation of its library functions to automatically achieve good performance on different architectures. Available at <http://math-atlas.sourceforge.net/>

² Available at <http://developer.intel.com/software/products/mkl/>

supercomputing capacity. However, each row of the matrix contains well under 100 non-zeros per row (Poole et al. 2003). The nonzeros of the matrix in each row represent the connectivity relationship (e.g. between adjacent finite elements), and thus the common degree of freedom. By taking advantage of the sparsity property, the required memory to store the matrix and the required computation can be dramatically reduced.

Various techniques exploiting the sparsity of finite element matrices are used to trim down memory requirements (and indeed speed up the total computation). These techniques differ in their storage schemes (e.g. banded, skyline, compressed row, and compressed column) and the method of calculating a solution. Generally, the sparse Gauss elimination process involves 1) analysis of the sparsity structure to determine a pivot ordering; 2) symbolic factorization to generate a structure for non-zeros; and 3) numerical factorization (Duff 1996).

In the first phase of sparse Gauss elimination, sparsity structures are analyzed to minimize fill-ins (nonzero entries caused by row eliminations), or to minimize profile or bandwidth (for banded storage schemes). The more fill-ins that are generated, the more memory is needed to store these additional nonzero entries, causing more arithmetic operations and longer solution time. The actual numerical values in \mathbf{K} are not important in this phase, and the analysis of sparsity structure only concerns the layout of nonzeros in \mathbf{K} . As a result, this phase does not perform any floating-point calculations, and the integer calculation performance of the processor dictates the efficiency. This phase could take nontrivial time to compute, but can be reused as long as the matrix nonzero structure remains constant (e.g. in a step of non-linear elasto-plastic calculations where only the value stiffness are updated and connectivity remains identical). Reverse Cuthill-McKee (Cuthill and McKee 1969), nested dissection (George 1973), minimum degree with multiple eliminations (Liu 1985), and approximate minimum degree (Amestoy et al. 1996) are all algorithms for ordering sparse matrices to reduce fill-ins.

The second phase of sparse Gauss elimination examines how the nonzero entries interact with each other, and tries to arrange operations for the numerical factorization phase in order to utilize high-level BLAS subroutines in order to achieve high performance. Again, this phase does not concern the actual numerical values in the matrix, but rather their

locations. The frontal method (Duff and Reid 1983) and the supernodal approach (Ashcraft et al. 1987) are both commonly used for this purpose.

The final phase for sparse Gauss elimination is numerical factorization. This phase is driven by the ordering determined from preceding phases and does the actual floating-point computation. Row permutation for partial pivoting is also done in this stage to maintain numerical stabilities and accuracies.

Sparse Gauss elimination is still an active research area for applied mathematicians and computer scientists, and is fundamental to numerical solutions in almost all disciplines. A list of freely available software, compiled by Dongarra for sparse linear system solvers can be found at <http://www.netlib.org/utk/people/JackDongarra/la-sw.html>.

3.5 Iterative Solution Techniques

There are a number of iterative techniques for solving a linear system of equations $\mathbf{KU} = \mathbf{F}$, and can be categorized into two families: 1) Gauss-Jacobi (Gauss-Jacobi iteration, Gauss-Seidel iteration, and successive over-relaxation); and 2) Krylov subspace (e.g. conjugate gradient, generalized minimum residual, bi-conjugate gradient, among many others.) Saad and Vorst (2000) provide an excellent overview and evolution of these methods.

Successive over-relaxation methods were very popular before 1980 for solving large-scale problems due to their modest memory requirement comparing to direct methods, but were later overshadowed by the more powerful techniques (Saad and Vorst 2000) such as Krylov subspace methods. Gauss-Jacobi family methods today are used in conjunction with Krylov subspace methods as preconditioners, or in multigrid methods as smoothers.

3.5.1 Subspace Method and Residual Minimization

To solve a linear system of equations, the subspace method (Saad 1996) approximates the solution to $\mathbf{KU} = \mathbf{F}$ (with \mathbf{U} as the exact solution) by a weighted combination of m search directions (vectors) $\boldsymbol{\omega}_i$ as follows:

$$\mathbf{x}_m = \sum_{i=0}^{m-1} \alpha_i \boldsymbol{\omega}_i \tag{3.5}$$

where α_i are weighting factors, and \mathbf{x}_m is the approximate solution constituted by m search directions. The linear combination of search directions ω_i forms a vector space Ω . If the search directions ω_i and weightings α_i are known in Equation (3.5), then the error or the residual R defined in the L2-norm¹ of the approximation solution can be written as:

$$R = \|\mathbf{r}\|_2 = \|\mathbf{F} - \mathbf{K}\mathbf{x}_m\|_2 = \left\| \mathbf{F} - \sum_{i=0}^{m-1} \alpha_i \mathbf{K}\omega_i \right\|_2 \quad (3.6)$$

where \mathbf{r} is the residual vector. Equation (3.6) can be interpreted as approximating the right-hand-side vector \mathbf{F} by a weighted combination of transformed vectors (ω_i transformed by \mathbf{K}). The solution procedure of the subspace method to solve $\mathbf{K}\mathbf{U} = \mathbf{F}$ is summarized in Figure 3-10. It is seen that each iteration of the subspace method generates a pair of (ω_i, α_i) , and at most add one additional rank to the solution vector space Ω .

In Figure 3-10, the general subspace method for solving $\mathbf{K}\mathbf{U} = \mathbf{F}$ requires two components: 1) a method to generate search directions ω_i , and 2) a method to determine weightings α_i to minimize the residual. These missing details differentiate different subspace based algorithms.

Assuming the search directions ω_i are known, the projection method provides a framework to determine the corresponding weightings α_i to achieve a unique approximate solution. Ideally, if the exact solution is found, the residual vector \mathbf{r} in Equation (3.6) is zero, thus the residual vector is orthogonal to all vectors. However, \mathbf{r} is generally not a zero vector because \mathbf{x}_m is an approximate solution constructed from a limited number (m) of search directions. The unknown weightings α_i can then be determined by orthogonalizing the residual vector \mathbf{r} to m linearly independent vectors, which ensure the uniqueness of the weightings. The m linearly independent vectors form another subspace \mathbf{L} , which can be treated as a reference of the quality of approximation, and projection methods make best effort on the subspace \mathbf{L} (to be orthogonalized to all vectors in this subspace). The projection method can be summarized mathematically in the following statement:

¹ For a vector \mathbf{r} , the L2-norm is defined by: $\|\mathbf{r}\|_2 = \sqrt{\mathbf{r} \cdot \mathbf{r}}$

$$\text{Find } \mathbf{x}_m \in \Omega, \text{ such that } \mathbf{r} = (\mathbf{F} - \mathbf{K}\mathbf{x}_m) \perp \mathbf{L} \quad (3.7)$$

In summary, the subspace method with projection technique determines the approximate solution to a linear system of equations using two vector spaces: 1) Ω , which forms the basis of the approximate solution; and 2) \mathbf{L} , which determines the weightings of each basis vector in Ω by orthogonalizing each basis vector to the vector space \mathbf{L} . Note that Ω and \mathbf{L} can be chosen arbitrarily, and the subspace method (with projection technique) provides the framework to solve the linear system using an iterative approach.

Although \mathbf{L} is a vector space of any choice, the quality of the approximate solution in each iteration depends on the choice of \mathbf{L} . Two common choices for \mathbf{L} are $\mathbf{L} = \Omega$, known as orthogonal projection, and $\mathbf{L} = \mathbf{K}\Omega$, known as oblique projection. It can be proven (Saad 1996) that $\mathbf{L} = \Omega$ minimizes the residual measured in \mathbf{K} -norm¹ for symmetric positive definite \mathbf{K} ; while $\mathbf{L} = \mathbf{K}\Omega$ minimizes the residual $\|\mathbf{F} - \mathbf{K}\mathbf{U}\|_2$ for arbitrary square matrices. Most subspace-based iterative methods are based on either one of these two optimality properties.

3.5.2 Krylov Subspace Method

The Krylov subspace method completes the formulation in 3.5.1 with residual minimization by providing search directions ω_i . The Krylov subspace is a subspace spanned by an arbitrary starting vector, \mathbf{v} , and a polynomial of \mathbf{K} as follows:

$$\text{m-th Krylov subspace } \overline{\mathbf{K}}_m \equiv \text{span}\{\mathbf{v}, \mathbf{K}\mathbf{v}, \mathbf{K}^2\mathbf{v}, \mathbf{K}^3\mathbf{v}, \dots, \mathbf{K}^i\mathbf{v}, \dots, \mathbf{K}^{m-1}\mathbf{v}\} \quad (3.8)$$

where m is a prescribed dimension of the Krylov subspace. For simplicity, another set of linearly independent vectors \mathbf{p}_i that spans the same vector space is defined:

$$\overline{\mathbf{K}}_m = \text{span}\{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{m-1}\} = \text{span}\{\mathbf{v}, \mathbf{K}\mathbf{v}, \mathbf{K}^2\mathbf{v}, \mathbf{K}^3\mathbf{v}, \dots, \mathbf{K}^i\mathbf{v}, \dots, \mathbf{K}^{m-1}\mathbf{v}\} \quad (3.9)$$

The initial residual vector \mathbf{r}_0 corresponding to an arbitrarily selected initial estimate, \mathbf{x}_0 , is usually used as the starting vector \mathbf{v} , and the base vectors \mathbf{p}_i are commonly constructed from the Arnoldi's method or Modified Gram-Schmidt (MGS) procedure shown in Figure

¹ $\{[\mathbf{K}(\mathbf{U} - \mathbf{x}_m)] \cdot [(\mathbf{U} - \mathbf{x}_m)]\}^{1/2}$

3-11. Arnoldi's method (Arnoldi 1951) is a procedure to convert a given set of vectors into a set of linearly independent vectors spanning the same vector space, while the MGS procedure is a mathematically equivalent variant, but is numerically stable when round-off is presented (Saad 1996). Thus, the MGS procedure provides a method to satisfy both Equation (3.9) and the orthogonality condition $\mathbf{p}_i \cdot \mathbf{p}_j = 0, \forall i \neq j$. Based on the initial estimate and residual, the approximate solution constructed from the Krylov subspace and corresponding residual can be written as:

$$\begin{cases} \mathbf{x}_m = \mathbf{x}_0 + \sum_{i=0}^{m-1} \alpha_i \mathbf{p}_i \\ \mathbf{r}_m = \mathbf{r}_0 - \sum_{i=0}^{m-1} \alpha_i \mathbf{K} \mathbf{p}_i \end{cases}, \begin{cases} \mathbf{x}_{m+1} = \mathbf{x}_m + \alpha_m \mathbf{p}_m \\ \mathbf{r}_{m+1} = \mathbf{r}_m - \alpha_m \mathbf{K} \mathbf{p}_m \end{cases} \quad (3.10)$$

In Equation (3.10), α_m is the coefficient to be determined by the projection method, thus depends on the choice of vector space \mathbf{L} . Figure 3-12 and Figure 3-13 summarize the Conjugate Gradient (CG) and Generalized Conjugate Residual (GCR) algorithms, based on orthogonal projection ($\mathbf{L} = \Omega$) and oblique projection ($\mathbf{L} = \mathbf{K}\Omega$). Respectively the difference in algorithms are evident on line 3 and line 6 – 7. Due to these different projections, the CG algorithm applies to symmetric positive definite matrices, while the GCR algorithm applies to general square matrices. There are many variants of the Krylov subspace method based on the same principle, with variations on projection space or different generation methods for orthogonal bases \mathbf{p}_i to reduce memory storage requirements or computation.

The convergent rate of the Krylov subspace method depends on the spectrum of eigenvalues of the system to be solved, and is proportional to the ratio between the maximum and the minimum eigenvalues. Therefore, Krylov subspace methods are inefficient for ill-conditioned systems with a wide spread spectrum of eigenvalues.

Preconditioning is a technique that attempts to cluster eigenvalues or to change eigenvalue distributions in order to accelerate the convergence of Krylov subspace method, and is “*simply a means of transforming the original linear system into one which has the same solution, but which is likely to be easier to solve with an iterative solver*” (Saad 1996). There are three ways to precondition a system:

$$\left\{ \begin{array}{l} \text{Left Preconditioning: } \mathbf{PKU} = \mathbf{PF} \\ \text{Right Preconditioning: } \mathbf{KP}\bar{\mathbf{U}} = \mathbf{F} \Rightarrow \mathbf{U} = \mathbf{P}\bar{\mathbf{U}} \\ \text{Split Preconditioning: } \mathbf{P}_L\mathbf{K}\mathbf{P}_R\bar{\mathbf{U}} = \mathbf{P}_L\mathbf{F} \Rightarrow \mathbf{U} = \mathbf{P}_R\bar{\mathbf{U}} \end{array} \right. \quad (3.11)$$

where \mathbf{P} is the preconditioner. There is no obvious advantage between the different preconditioning methods in Equation (3.11). However, for a symmetric system (e.g. symmetric \mathbf{K}), the split preconditioning technique can be used to preserve the symmetry of the preconditioned system, while the other two techniques may result in an unsymmetrical system (even though \mathbf{K} is symmetric). Preconditioners should be efficient to compute in order not to offset their advantages. Figure 3-14 shows a left preconditioned conjugate gradient algorithm.

It has been noted by Saad (1996):

Lack of robustness is a widely recognized weakness of iterative solvers, relative to direct solvers. This drawback hampers the acceptance of iterative methods in industrial applications despite their intrinsic appeal for very large linear systems.

In general, the reliability of iterative techniques, when dealing with various applications, depends much more on the quality of the preconditioner than on the particular Krylov subspace accelerators used.

Unfortunately, there is no “general-purpose” preconditioner that guarantees convergence, nor is there a guideline on selecting or constructing preconditioners. As a result, all commercially available finite element packages are built based on direct solvers, and very few of these packages offer iterative solvers (even as an option).

3.6 Direct versus Iterative Methods

Table 3-2 compares the different characteristics of direct and iterative solvers (based on Krylov subspace methods) in the following aspects:

1. Computation: In each step of the direct method, a sub-matrix of \mathbf{K} is updated, and therefore the matrix \mathbf{K} needs to be explicitly formed and stored. For iterative methods, each step of the iteration involves evaluations of a matrix-vector product, residual evaluation, and search direction generation. Therefore, for iterative methods,

the matrix \mathbf{K} can be defined implicitly, as long as the matrix-vector product can be computed.

2. Memory requirement: For sparse linear systems, direct methods require more memory than iterative methods because the matrix \mathbf{K} , and updates of \mathbf{K} , need to be formed and stored explicitly. As a result, the memory requirement grows as the factorization progresses because additional non-zeros (fill-ins.) are generated during the factorization process. In contrast, iterative methods only need to store one additional search vector in each iteration, and therefore are more memory efficient. Figure 3-15 shows a comparison of solvers used in the commercial finite element code NE/Nastran. It is obvious from Figure 3-15 that the iterative solver (VIS) uses far less memory than the sparse direct solver (VSS) and block profile solver (LDU).
3. Solution time dependency: For solving a dense system with direct methods, the solution time grows cubically with the number of unknowns, while for iterative methods the convergence time depends on the spectrum of eigenvalues rather than the number of unknowns. For direct solutions of sparse systems, arithmetic operations can be avoided on the zero entries, thus the solution time for solving sparse systems is governed by both the problem size and the effectiveness of sparsity exploitation (this is illustrated by the comparison of VSS versus LCD in Figure 3-15).
4. Preconditioning: Both direct and iterative methods may have a preconditioning step for the \mathbf{K} matrix. For direct methods, the preconditioning step permutes rows and columns without changing solutions to the system in order to minimize the fill-ins generated during the factorization process, and therefore reduce both the memory requirement and arithmetic operations. Iterative methods, on the other hand, transform the system by multiplying the system \mathbf{K} by the preconditioner, changing the solution in order to reduce the spectrum of eigenvalues (i.e. clustering eigenvalues).
5. Problem size scaling: The scaling of solution to the problem size can be quantified by considering the relations between solution time and the number of unknowns using a power law relation $t = \alpha N^p$. For direct solution techniques, the exponent

ranges from $p=3$ for a dense system; to $p=1.2 - 1.7$ for a sparse system of equations solved by the multi-frontal method (Fish and Shek 1999). Hsieh and Whittle (2003) reported $p=1.51$ for simple three-dimensional tunneling analyses with direct multi-frontal solver in ABAQUS; while $p=1.33$ is reported for aeroacoustics applications with NASA's GPS solver (Watson and Storaasli 2000). The preconditioned conjugate gradient (PCG) method shows $p=1.17 - 1.33$ (Fish and Shek 1999). Figure 3-16 illustrates the scaling with various p between 1.0 – 1.7.

6. Robustness: Even though Krylov subspace based iterative methods hold several advantages such as memory requirement and problem size scaling, direct solvers still dominate most of the finite element packages available today because of their robustness. Direct solution techniques will always find a solution to nonsingular systems, while iterative methods may not converge to a satisfactory answer unless the system is well conditioned. Therefore, direct solvers can be considered as a true black-box solution technique (with minimal user intervention), while iterative methods require user interventions (to specify termination tolerance, preconditioning, monitoring, etc.).

Based on the comparison, it can be concluded that direct solution techniques are well suited for small to medium size problems or for ill-conditioned systems; while iterative solvers should be employed for large projects that cannot be solved efficiently with direct solvers. Direct methods, however, have the advantage for cases with multiple load cases (thus a constant \mathbf{K} and multiple \mathbf{F}), as the time-consuming factorization needs to be done only once, while iterative methods need to restart the iteration process from the beginning for each right hand side.

3.7 Parallel Computation

Iterative methods with appropriate preconditioners provide possibilities for solving large-scale problems on modest size computers, and a limited number of commercial finite element codes with iterative solvers are now available (e.g. ABAQUS 6.4-1, 2003; ANSYS 5.1, 1995, and Plaxis 3D Tunnel, 2003) However, problem size grows dramatically from two-dimensions to three-dimensions. A simple mesh extrusion from a 2-D mesh with 50 layers generates at least 50 times more unknowns (degrees of freedom), and 3-D connectivity

gives rise to more non-zero entries in the global stiffness matrix (i.e. nodes have additional connectivity to nodes on other planes of extrusion). Based on Figure 3-16, the solution time for a single linear system solution is expected to be 110 times longer than two-dimensional simulations for $p = 1.2$, and 350 times longer for $p = 1.5$. Parallel computation techniques offer the ability to aggregate the computing power from multiple interconnected processing units to compensate the increased model size from two-dimensions to three-dimensions.

3.7.1 Parallel Computing Hardware

There are two basic types of parallel computing hardware architecture: shared memory and distributed memory. Pure shared memory architecture, as illustrated in Figure 3-17, physically connects several processors to a single pool of memory through a high-speed system bus. All connected processors have equal access to all available memory resources. The advantage of shared memory architecture is easy programming, as the use of memory resources is no different from single processor systems. However, there are several shortcomings associated with shared memory architecture:

1. In the pure shared memory architecture as illustrated in Figure 3-17, the access to memory resources passes through a shared system bus. The latency¹ can increase greatly when multiple processors actively access memories.
2. Synchronization mechanisms are needed to ensure consistency of a particular memory address when being referenced and modified by multiple processors. The synchronization increases the data access latency.
3. When local cache memory is used to speed up the reference to shared memory resources, it is difficult to ensure the coherence between the cache and the memory, and additional mechanisms are needed to notify all caches that reference a modified memory address. Such hardware specialization and support can add dramatically to the economic cost.

¹ The time to access memory.

4. Due to aforementioned shortcomings, scalability of pure shared memory architecture is quite limited (usually less than 32 processors).

For distributed memory architecture, as illustrated in Figure 3-18, each processor has its own private memory and address space. The data exchange between processors requires explicit transfer through an interconnecting network fabric. An advantage of distributed memory architecture over shared memory architecture is scalability. Memory synchronization and cache coherence are less critical. Therefore, distributed memory architecture computers can scale to more than 1000 processors.

Modern supercomputers are usually hybrids of the two architectures. Each node in Figure 3-18 can comprise of 2 – 8 processors sharing a pool of memory. Global shared memory can be logically achieved on distributed memory hardware via software techniques. Figure 3-19 shows the architecture of the Earth Simulator, the fastest supercomputer currently in operation. This machine provides further hardware acceleration support for vectorized floating-point arithmetic on each processor.

Beowulf cluster computers are low cost high performance distributed memory architecture. Beowulf are constructed from COTS (Components Off The Shelf) and interconnected using high-bandwidth network switches. They lack the efficiency of custom-made machines, but serve as excellent prototyping and general purpose parallel computers. Typical Beowulf clusters today consist of several single or dual processors x86 personal computers with 32-bit Intel or AMD processors, fast or gigabit Ethernet interconnections, and GNU/Linux or FreeBSD operating systems¹.

3.7.2 Programming for Parallel Computers

Programming or porting code for parallel computers is a non-trivial task. Different tools and considerations are needed for programming on a distributed memory architecture compared to on a shared memory architecture.

¹ Beowulf mailing list FAQ, version 2 by Kragen Sitaker (1999). <http://www.canonical.org/~kragen/beowulf-faq.txt>

The standard today for utilizing shared memory machines is OpenMP, which is “a specification for a set of compiler directives, library routines, and environment variables that can be used to specify shared memory parallelism in Fortran and C/C++ programs.”¹ Parallel programs on shared memory machines make each parallel task a thread, and variables within each thread can be made either private, or shared with other threads, thus the complexity of data flow can be completely hidden from programmers. However, large shared memory parallel computers today have system architectures similar to the one in Figure 3-19, and access to remote memory addresses incurs additional latency. Therefore, the efficiency of parallel code can be improved by explicitly considering data locality.

The MPI (Message Passing Interface) is a ‘de facto’ standard specification for programming distributed memory systems. Unlike OpenMP, there is no shared variable in MPI due to the nature of the distributed environment, thus information sharing requires explicit message passing between processors. The advantage is programmers are forced to treat each processor autonomously, only sharing information when necessary. Perhaps the largest obstacle to migrating to a distributed memory machine is the lack of automatic translator to migrate a sequential code to a parallel code using MPI.

In summary, writing code for parallel computers is not a trivial task, and requires an understanding of the underlying hardware architecture (including the communication link) in order to utilize these resources efficiently. There are two standard APIs today for parallel programming: MPI, targeting distributed memory architecture for message passing; and OpenMP, for shared memory architectures (often) utilizing threads. Due to the hybrid-nature (distributed and shared) of modern parallel computers, it may be necessary to use both APIs concurrently.

3.7.3 Parallel Efficiency and Communication Overhead

Parallel efficiency is most commonly described using the following performance measures:

¹ OpenMP Architecture Review Board, <http://www.openmp.org/index.cgi?faq#Q1>

$$Speed\ Up = \frac{T_{base}}{T_{np}} \quad (3.12)$$

$$Efficiency = \frac{T_{base}}{T_{np}} \times \frac{N_{base}}{N_{np}} = Speed\ Up \times \frac{N_{base}}{N_{np}} \quad (3.13)$$

where T_{base} is a reference computation time using N_{base} processors as the reference configuration, T_{np} is the computation time with N_{np} processors. In reality, parallel efficiency of 100% is achievable only when the communication between processors is negligible.

There are two distinct components when considering the performance of parallel computation: arithmetic operations and communications. The time for arithmetic operations can be reduced by using multiple processors, but the time for communication is either constant, or most often increases with the number of processors involved in the computation. Therefore, as more and more processors are used, the cost for communication becomes significant.

Assume a computation task needs T_{one} to complete with a single processor, and perfect distribution of computation load on N_{np} available processors, the computation time alone is T_{one} / N_{np} . However, the total solution time needs to include the communication time, T_{comm} , to distribute and collect computations results from the N_{np} processors. Thus,

$$T_{total} = \frac{T_{one}}{N_{np}} + T_{comm} \quad (3.14)$$

And the parallel efficiency based on (3.13):

$$Efficiency = \frac{T_{one}}{T_{one} + N_{np}T_{comm}} \quad (3.15)$$

From (3.15), it is observed that as $T_{comm} \rightarrow 0$, the parallel efficiency approaches unity, and efficiency decreases with increasing N_{np} while holding both T_{one} and T_{comm} constant. In reality, however, T_{comm} usually increases with increasing N_{np} due to one of more of the following reasons:

1. increased amount of aggregate communication for each processor

2. increased overhead for routing network traffic
3. network congestion to a particular processor resulting in a load imbalance

Therefore, there are two possibilities to gain parallel efficiency based on (3.15) by 1) reducing time for communication T_{comm} by employing high-bandwidth and low-latency communication links; 2) increasing computation load T_{one} ; and/or 3) design efficient communication patterns or algorithms to mitigate the scaling dependency of T_{comm} .

3.7.4 Parallelization of Direct and Iterative solvers

Both direct and iterative solution methods have been adapted to parallel computing platforms¹. However, direct methods require more effort than iterative methods to parallelize. As indicated in Table 3-2, each step of the direct method operates on a sub-matrix of the original system \mathbf{K} , thus a large amount of communication is needed during the elimination process when \mathbf{K} is distributed among multiple processors. In addition, to perform forward elimination, the partial pivoting required to maintain numerical stability incurs additional communication needs in order to permute rows during factorization. Therefore, direct methods require more communication between processors than iterative methods.

In order to alleviate the communication demand of parallel direct methods, careful data layout of \mathbf{K} onto available processors is essential to achieve high-performance factorization. Figure 3-20 shows 1-D and 2-D block-cyclic layouts, two commonly used partition schemes for dense matrices. The cyclic mapping provides better load-balancing than simple partitioning (Blackford et al. 1997). Two-dimensional block-cyclic is also used in SuperLU-DIST (Li and Demmel 2003) for parallel sparse matrix factorization. The data layout enables the use of BLAS-3 matrix updates, achieves good load-balancing, and less communication than other data layouts.

¹ Example: TAUCS, a library of sparse linear solvers by Sivan Toledo, with Doron Chen and Vladimir Rotkin. Available at <http://www.tau.ac.il/~stoledo/taucs/>

Due to the high communication demand of direct methods, parallel efficiency degrades rapidly with the increasing number of processors for a given problem. Figure 3-21¹ shows the parallel efficiency of direct solvers used in two commercial finite element codes ABAQUS v6.2 – 6.3 and ANSYS v5.7, running on multi-processor workstations. It is seen that the parallel efficiency degrades rapidly as the number of processors increases for the two commercial codes. However, better scalability (e.g. less degradation of parallel efficiency with increasing number of processors) is seen in Figure 3-22 using two publicly available codes SuperLU (Li and Demmel 1999) and MUMPS (Amestoy et al. 2000). The difference on performance can be attributed to: 1) Cray-T3E has efficient interconnections (e.g. higher-bandwidth, lower-latency), and 2) SuperLU and MUMPS use message-passing explicitly, while ABAQUS and ANSYS use shared-memory.

Iterative methods, on the other hand, require only matrix-vector products at each step of iteration, thus parallelization is much easier than for direct methods, and the communication requirement is a round trip of a single vector. However, iterative methods usually need preconditioning in order to achieve good convergence rates, and preconditioners may incur additional communication and computation overhead. Unfortunately, there are minimal data published on the parallel performance of iterative methods.

In conclusion, direct solution methods show undesirable scaling characteristics with the problem size, and do not parallelize well even when utilizing tightly coupled supercomputers with high-speed interlinks (such SGI Origin 2000 and Cray T3E) as shown in Figure 3-21 and Figure 3-22. Therefore, applying direct solution methods with loosely coupled computers (e.g. Beowulf cluster) does not provide an efficient solution. Iterative methods are less sensitive to problem size than direct methods, easy to parallelize, but less robust. Preconditioning is usually needed to improve the performance of iterative solvers, but finding good preconditioners is also difficult. Thus, other methods must be considered in order to solve large-scale geotechnical engineering problems.

¹ Ansys Data taken from Poole et al. (2003), ABAQUS data taken from ABAQUS' website:
<http://www.abaqus.com/products/performance63.html>

3.8 Domain Decomposition Method

Domain decomposition is a methodology for solving large-scale problems by decomposing the problem into several smaller independent sub-problems (sub-domains), and then solving the smaller sub-problems in serial or in parallel using multiple processors. The solution to the solution to the original large-scale problem is recovered by combining the solution of sub-problems.

There are several advantages for employing domain decomposition methods:

1. The solution time for solving a linear system of equations scales with the problem size super-linearly. As a result, by subdividing the original problem into smaller ones, the summation of the solution times for the sub-problems is less than solving the original problem.
2. The sub-problems can be solved independently from each other, thus can easily take advantage of the availability of multi-processors by assigning sub-problems to different processors.
3. Because large-scale problems can demand excessive amounts of in-core memory to solve using direct methods, the use of secondary storage (e.g. hard disk) is often necessary (i.e. out-of-core solvers). This dramatically slows down the solution process. By employing domain decomposition methods, the out-of-core memory requirement can be avoided as sub-problems can be solved independently.

There are two categories of domain decomposition: overlapping and non-overlapping, and will be described in the subsequent sections.

3.8.1 Overlapping Domain Decomposition Method

Overlapping domain decomposition methods subdivide the original domain with overlapping regions between sub-domains at the boundaries. The overlaps between sub-domains are then used to exchange information between sub-domains until the governing equations are satisfied everywhere.

Schwarz (1869) first proposed an alternating scheme for solving partial differential equations on irregular geometries from known solutions on regular geometries. This scheme is illustrated in Figure 3-23, with an accompanying flowchart in Figure 3-24. As can be seen, the solution process alternates between the solution for domain 1 and domain 2. The alternating scheme can easily be extended to handle the case with multiple sub-domains as in Figure 3-25 with calculation procedures summarized in Figure 3-26. It is seen in Figure 3-26 that a queue that compiles the sub-domains to be solved is added in order to keep track of calculations to be performed, and sub-domains in the queue can be solved in parallel. The initial selection of sub-domains to be solved should consider which sub-domain is “information rich”. For example, the sub-domain that has unbalanced forces due to excavation or external forces should be calculated first.

This one-level domain decomposition method can suffer from slow convergence characteristics, as the information from one sub-domain can only propagate to its neighbor sub-domains within a single iteration. Figure 3-25 demonstrates this problem with a conceptual model. Information in sub-domain 13 will require at least 3 iterations to reach sub-domain 4 (13 – 10 – 7 – 4) via corner overlaps; and 6 iterations (13 – 14 – 10 – 11 – 7 – 8 – 4) via edge overlaps. Therefore, it can be expected that the iteration count to achieve convergence grows with the number of sub-domains (as information needs to travel further from one end to the other) and with the amount of overlapping between sub-domains. The convergence of overlapping domain decomposition methods can be improved by introducing multi-level domain decomposition, which adds coarse grids (e.g. lower resolution meshes) to help information propagation, and therefore improves the convergence characteristics. However, generating coarse meshes can be difficult for unstructured grids. It is also difficult to decompose automatically general unstructured meshes with appropriate overlaps, as 1) sub-domains need to have good geometrical shape; 2) identify circumference elements surrounding a sub-domain of any shape is difficult for unstructured meshes; 3) determine appropriate amount of overlaps automatically is also problematic. No general-purpose finite element calculation package based on the overlapping domain decomposition method is mentioned in the literature.

Once \mathbf{U}_B is solved in Equation (3.24), the displacement field for each sub-domain can be calculated independently from each other by solving equation (3.25):

$$\mathbf{K}_{II}^{(i)}\mathbf{U}_I^{(i)} + \mathbf{K}_{IB}^{(i)}\mathbf{U}_B = \mathbf{F}_I^{(i)} \quad (3.25)$$

3.8.2.2 Formulation of a Force-Based Interface Problem - FETI Algorithm

The FETI (Finite Element Tearing and Interconnecting, Farhat and Roux 1991) method is the first non-overlapping domain decomposition method that utilizes Lagrangian multipliers to interconnect sub-domains. The Lagrangian multiplier has the physical meaning of interconnecting forces between sub-domains. By considering the two-sub-domain cantilever beam problem illustrated in Figure 3-29, the equilibrium condition for each sub-domain can be written as follows:

$$\begin{cases} \mathbf{K}_1\mathbf{U}_1 = \mathbf{F}_1 - \mathbf{B}_1\lambda \\ \mathbf{K}_2\mathbf{U}_2 = \mathbf{F}_2 - \mathbf{B}_2\lambda \end{cases} \quad (3.26)$$

$$\lambda^{4 \times 1} = \left[\text{int } f_2^x \quad \text{int } f_2^y \quad \text{int } f_5^x \quad \text{int } f_5^y \right]^T$$

$$\mathbf{B}_1^{8 \times 4} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T, \quad \mathbf{B}_2^{8 \times 4} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}^T$$

where \mathbf{K}_1 , \mathbf{K}_2 are stiffness matrix for sub-domains; \mathbf{U}_1 , \mathbf{U}_2 are unknown displacement fields; \mathbf{f}_1 , \mathbf{f}_2 are forcing terms as in standard finite element formulation without considering decomposition; λ is the Lagrangian multiplier that contains all the interaction forces between sub-domains; and \mathbf{B}_1 , \mathbf{B}_2 are selection operators that ‘pick up’ local interaction forces from λ . To recover the solution to the original problem, displacement compatibility is required at interfaces, and can be written as equation (3.27):

$$\mathbf{B}_1^T\mathbf{U}_1 + \mathbf{B}_2^T\mathbf{U}_2 = \mathbf{0} \quad (3.27)$$

By writing equation (3.26) and equation (3.27) in matrix form:

$$\begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{K}_2 & \mathbf{B}_2 \\ \mathbf{B}_1^T & \mathbf{B}_2^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{0} \end{Bmatrix} \quad (3.28)$$

The unknown displacement fields in equation (3.26) can be represented by:

$$\begin{cases} \mathbf{U}_1 = \mathbf{K}_1^{-1}[\mathbf{f}_1 - \mathbf{B}_1\lambda] \\ \mathbf{U}_2 = \hat{\mathbf{K}}_2^{-1}[\mathbf{f}_2 - \mathbf{B}_2\lambda] + \mathbf{R}_2\alpha_2 \end{cases} \quad (3.29)$$

$$\mathbf{K}_2\hat{\mathbf{K}}_2^{-1}\mathbf{K}_2 = \mathbf{K}_2 \quad (3.30)$$

where $\hat{\mathbf{K}}_2^{-1}$ is generalized inverse defined in equation (3.30), $\mathbf{R}_2^{8 \times 3}$ contains all the rigid body modes of sub-domain 2 in its column vectors, and $\alpha_2^{3 \times 1}$ contains the linear combination of rigid body modes. By substituting equation (3.29) into the displacement compatibility condition in equation (3.27):

$$[\mathbf{B}_1^T\mathbf{K}_1^{-1}\mathbf{B}_1 + \mathbf{B}_2^T\hat{\mathbf{K}}_2^{-1}\mathbf{B}_2]\lambda - \mathbf{B}_2^T\mathbf{R}_2\alpha_2 = \mathbf{B}_1^T\mathbf{K}_1^{-1}\mathbf{f}_1 + \mathbf{B}_2^T\hat{\mathbf{K}}_2^{-1}\mathbf{f}_2 \quad (3.31)$$

Equation (3.31) contains two unknowns (λ, α_2) in a single equation. Therefore, one additional equation is needed in order to solve the unknowns. Consider the rigid body condition in the sub-domain 2:

$$\mathbf{K}_2\mathbf{R}_2 = \mathbf{0} \Rightarrow \mathbf{R}_2^T\mathbf{K}_2^T = \mathbf{R}_2^T\mathbf{K}_2 = \mathbf{0} \Rightarrow \mathbf{R}_2^T\mathbf{K}_2\mathbf{U}_2 = \mathbf{R}_2^T(\mathbf{f}_2 - \mathbf{B}_2\lambda) = \mathbf{0} \quad (3.32)$$

Note that the symmetry of \mathbf{K}_2 is introduced in equation (3.32). (3.31) and (3.32) can be combined in the matrix form as:

$$\begin{bmatrix} \mathbf{B}_1^T\mathbf{K}_1^{-1}\mathbf{B}_1 + \mathbf{B}_2^T\hat{\mathbf{K}}_2^{-1}\mathbf{B}_2 & -\mathbf{B}_2^T\mathbf{R}_2 \\ -\mathbf{R}_2^T\mathbf{B}_2 & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \lambda \\ \alpha_2 \end{Bmatrix} = \begin{bmatrix} \mathbf{B}_1^T\mathbf{K}_1^{-1}\mathbf{f}_1 + \mathbf{B}_2^T\hat{\mathbf{K}}_2^{-1}\mathbf{f}_2 \\ -\mathbf{R}_2^T\mathbf{f}_2 \end{bmatrix} \quad (3.33)$$

The algorithm can easily be extended to N_s sub-domains case with N_f floating sub-domains as follows:

$$\begin{aligned} & \begin{bmatrix} \mathbf{F}_1 & -\mathbf{G}_1 \\ -\mathbf{G}_1^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \lambda \\ \alpha \end{Bmatrix} = \begin{bmatrix} \mathbf{d} \\ -\mathbf{e} \end{bmatrix} \\ & \mathbf{F}_1 = \sum_{i=1}^{N_s} \mathbf{B}_i^T \hat{\mathbf{K}}_i^{-1} \mathbf{B}_i \\ & \mathbf{d} = \sum_{i=1}^{N_s} \mathbf{B}_i^T \hat{\mathbf{K}}_i^{-1} \mathbf{f}_i \\ & \mathbf{G}_1 = \begin{bmatrix} {}^1(\mathbf{B}^T \mathbf{R}) & {}^2(\mathbf{B}^T \mathbf{R}) & \dots & {}^{N_f}(\mathbf{B}^T \mathbf{R}) \end{bmatrix} \\ & \mathbf{e} = \begin{bmatrix} {}^1(\mathbf{R}^T \mathbf{f})^T & {}^2(\mathbf{R}^T \mathbf{f})^T & \dots & {}^{N_f}(\mathbf{R}^T \mathbf{f})^T \end{bmatrix}^T \end{aligned} \quad (3.34)$$

Equation (3.34) can be solved as a constrained minimization problem:

$$\min_{\lambda} \left\{ \phi(\lambda) = \frac{1}{2} \lambda^T \mathbf{F}_1 \lambda - \lambda^T (\mathbf{G}_1 \alpha + \mathbf{d}) \right\} \text{ subject to } \mathbf{G}_1^T \lambda = \mathbf{e} \quad (3.35)$$

where

$$\frac{d\phi(\lambda)}{d\lambda} = \mathbf{F}_I \lambda - (\mathbf{G}_I \alpha + \mathbf{d}) \quad (3.36)$$

The preconditioned conjugate projection gradient method (Farhat and Roux 1991) is used to solve equation (3.35) iteratively. The procedure starts by first finding an initial estimate, λ_0 , that satisfies the constraint $\mathbf{G}_I^T \lambda_0 = \mathbf{e}$, and then further corrections $d\lambda$ is constrained with:

$$\mathbf{G}_I^T d\lambda = \mathbf{0} \quad (3.37)$$

The objective function $\phi(\lambda)$ in equation (3.35) can be re-written:

$$\phi(\lambda) = \frac{1}{2} \lambda^T \mathbf{F}_I \lambda - \lambda^T \mathbf{d} - (\lambda_0^T \mathbf{G}_I \alpha + \mathbf{d} \lambda^T \mathbf{G}_I \alpha) \quad (3.38)$$

The term $\lambda_0^T \mathbf{G}_I \alpha$ can be dropped because it is a constant, and therefore does not affect the result of minimization; and $\mathbf{d} \lambda_0^T \mathbf{G}_I \alpha$ can also be dropped as a consequence of equation (3.37). Therefore, the original problem in equation (3.35) becomes:

$$\begin{aligned} \min_{\lambda} \phi(\lambda) &= \frac{1}{2} \lambda^T \mathbf{F}_I \lambda - \lambda^T \mathbf{d} \\ \lambda &= \lambda_0 + d\lambda \\ \mathbf{G}_I^T \lambda_0 &= \mathbf{e} \\ \mathbf{G}_I^T d\lambda &= \mathbf{0} \end{aligned} \quad (3.39)$$

Equation (3.39) suggests the solution λ is independent from the α , the weighting of each rigid body motions. This is because the interconnecting force λ should not be a function of rigid body motions that produces zero force change within substructures. One of the initial estimate λ_0 that satisfies $\mathbf{G}_I^T \lambda_0 = \mathbf{e}$ can be obtained:

$$\lambda_0 = \mathbf{G}_I (\mathbf{G}_I^T \mathbf{G}_I)^{-1} \mathbf{e} \quad (3.40)$$

A suitable correction $d\lambda$ satisfying $\mathbf{G}_I^T d\lambda = \mathbf{0}$ can also be calculated by a projection operator:

$$\mathbf{P} = \mathbf{I} - \mathbf{G}_I (\mathbf{G}_I^T \mathbf{G}_I)^{-1} \mathbf{G}_I^T \quad (3.41)$$

Finally, the Preconditioned Conjugate Gradient (PCG) method with preconditioner \mathbf{M} combined with the projection operator in equation (3.41) is used to solve equation (3.39), and can be summarized in Table 3-3.

The termination condition (convergence tolerance) for the FETI method has been discussed in detail by Farhat and Roux (1994), and it is suggested that the global residual $\|\mathbf{F} - \mathbf{KU}\|$ can be approximated using the preconditioned projected residual \mathbf{z}_i in the step 5 above (Farhat and Roux 1994; Rixen and Farhat 1999).

The advantage of the FETI algorithm is that it forms a natural coarse problem related to the rigid body modes of all sub-domains (due to solution of the auxiliary equation (3.37)). The natural coarse space of the rigid body modes is small (at most 3 for 2D problems, and 6 for 3D problems) and is generated automatically. Furthermore, the coarse problem helps to propagate information globally, thus ensuring the scalability of the algorithm. The FETI method has been demonstrated to scale up to 1000 processors (Bhardwaj et al. 2000).

Various extensions have been made to the original FETI algorithm (referred to as the one-level FETI method) previously described to solve problems with shells and plates (Farhat et al. 1998a; Farhat et al. 2000; Farhat and Mandel 1998), multi-point constraints (Farhat et al. 1998b), and contact problems (Dureisseix and Farhat 2001). Various performance enhancements have also been proposed to help convergence characteristics by improving the initial estimate (Gosselet et al. 2003) and on problems involves multiple loads (Farhat et al. 1994; Rey and Lene 1998), heterogeneous material properties (Rixen and Farhat 1999), and incompressible conditions (Vereecke et al. 2003).

3.8.2.3 Preconditioners for the FETI Algorithm

Two traditional preconditioners have been proposed for the FETI method: 1) the Dirichlet preconditioner and 2) the lumped preconditioner. Both preconditioners are sub-domain based and can be constructed from the sub-domains stiffness matrix, \mathbf{K} , and the selection operator, \mathbf{B} :

$$\text{Dirichlet: } \mathbf{M} = \sum_{s=1}^{s=N} \mathbf{B}^{(s)} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{bb}^{(s)} \end{bmatrix} \mathbf{B}^{(s)T}, \quad \mathbf{S}_{bb}^{(s)} = \mathbf{K}_{bb}^{(s)} - \mathbf{K}_{ib}^{(s)} \mathbf{K}_{ii}^{(s)-1} \mathbf{K}_{ib}^{(s)} \quad (3.42)$$

$$\text{Lumped: } \mathbf{M} = \sum_{s=1}^N \mathbf{B}^{(s)} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{bb}^{(s)} \end{bmatrix} \mathbf{B}^{(s)T} \quad (3.43)$$

It can be shown that the Dirichlet preconditioner is numerically scalable (iteration count is independent of the number and size of sub-domains) while the more economical lumped preconditioner is not. The lumped preconditioner is computationally inexpensive, and has been recommended for second order elasticity problems, while the Dirichlet preconditioner is preferred for shell and plate problems (Farhat et al. 1998a; Farhat and Mandel 1998).

Mechanically consistent preconditioners (Rixen and Farhat 1999) scale the traditional preconditioners (Equation (3.44), (3.45)) by a diagonal matrix $\mathbf{A}^{(s)}$ determined from the multiplicity (number of sub-domains attached to a node) and/or the stiffness of the sub-domains. Mechanically consistent preconditioners greatly improve the convergence rate for heterogeneous problems with minimal cost.

$$\mathbf{M} = \sum_{s=1}^{s=N} \mathbf{A}^{(s)} \mathbf{B}^{(s)} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{bb}^{(s)} \end{bmatrix} \mathbf{B}^{(s)T} \mathbf{A}^{(s)} \quad (3.44)$$

$$\mathbf{M} = \sum_{s=1}^N \mathbf{A}^{(s)} \mathbf{B}^{(s)} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{bb}^{(s)} \end{bmatrix} \mathbf{B}^{(s)T} \mathbf{A}^{(s)} \quad (3.45)$$

The Dirichlet preconditioner provides the best numerical scalability, but it requires extensive computation and more memory. Incomplete Cholesky factorization can be used to provide an alternative to Dirichlet preconditioner with similar convergence rate and requires less memory and computation (Charmpis and Papadrakakis 2002).

The use of the lumped preconditioner was recommended by Farhat and Roux (1994) because it usually yields more efficient computation time than the Dirichlet preconditioner. However, this recommendation was based on the performance observed on supercomputers in the early 1990's, containing relatively slow processors compared to the supporting network interconnection.

Based on our numerical experiments, however, the use of mechanically consistent Dirichlet preconditioners for PC clusters with Ethernet interconnection is recommended. Although the preconditioner is computationally more expensive and uses more memory, it reduces the number of FETI iterations needed to achieve a convergent solution, and thus reduce

communications between nodes. This is especially ideal for PC clusters interconnected with Ethernet as modern processors for personal computers are now relatively powerful, compared to an Ethernet interconnection where the high latency is a more pronounced limiting factor.

For PC clusters equipped with high-speed low-latency interconnection such as Myrinet¹, the choice of preconditioner is problem dependent. For finite element models that are solvable with a few FETI iterations, $O(100)$, lumped preconditioners are recommended. Otherwise, the Dirichlet preconditioner should be used if there are no other constraints such as memory requirements.

3.9 Summary

- Large-scale finite element models require lengthy computation times to solve the linear system of equilibrium equations $\mathbf{KU} = \mathbf{F}$. When considering non-linear problems, this complexity increases significantly with the additional cost of iterative operations for intermediate solution steps.
- Direct solution techniques are mature and robust, but require explicit exploitation of sparsity and utilization of high-level BLAS subroutines to solve sparse systems of equations efficiently. In addition, the solution time for direct methods scales unfavorably with the problem size. In general the solution time follows a power-law scaling, $t = \alpha N^p$, describing solution time versus problem size. We have found $p = 1.51$ for a three-dimensional tunneling simulation, and generally $p = 1.2 - 3.0$ for general direct solution techniques. For these reasons, direct solution techniques are considered impractical for large-scale problems.
- Krylov subspace based iterative methods require less memory, are less sensitive to the problem size, and take advantage of sparsity of the equation system. As a result, iterative methods have become the solver of choice for large-scale problems.

¹ Myrinet is a high-bandwidth, low-latency interconnection fabric using fiber-optics. A product of Myricom, Inc. <http://www.myricom.com>

However, iterative methods may fail to converge or converge very slowly for ill-conditioned systems.

- Preconditioning is frequently applied to iterative methods in order to improve the rate of convergence or help to solve ill-conditioned systems. However, there are no universal preconditioners, and the preconditioning can be computationally intensive and memory inefficient.
- The domain decomposition method is a divide-and-conquer strategy that decomposes one large problem into several smaller ones, solves each sub-problem, and recovers the solution to the original problem by combining results from sub-problems.
- Domain decomposition methods are suitable for application on parallel computers. They can also be used on single-processor computers to reduce memory requirements.
- Overlapping domain decomposition iteratively exchanges the solution within sub-domains until the governing equations are satisfied globally. However, information exchange between distant sub-domains propagates relatively slowly through all intermediate sub-domains (one sub-domain per iteration). As a result, the iteration count increases as the number of sub-domains increases. Multi-level overlapping domain decomposition methods construct one or more coarse grids to help propagate information between sub-domains and thus, accelerate convergence. However, constructing coarse grids is a non-trivial process.
- Nonoverlapping domain decomposition methods introduce a reduced interface problem after decomposition. The reduced interface problem can be solved by either direct substructuring or iterative substructuring.
- The interface problem for nonoverlapping domain decomposition can use either displacement as unknowns, or use interaction forces as unknowns (FETI algorithm.) The former sums Schur complements from all sub-domains to form an interface stiffness matrix that is smaller than the original composite global stiffness matrix.

However, the interface stiffness matrix is a dense matrix, and can be expensive to solve if the interface problem is big.

- The FETI algorithm forms the reduced interface problem with Lagrangian multipliers, which have the physical meaning of interaction forces between sub-domains. The Lagrangian multipliers are then solved using a preconditioned conjugate projection gradient (PCPG) algorithm.
- The FETI algorithm uses iterative methods to solve the interface problem, thus large interface problems can be solved efficiently. Furthermore, the projection step in the PCPG iteration helps to propagate information globally on the coarse problem formed by the rigid body modes of sub-domains, and therefore the convergence rate does not deteriorate as the number of sub-domains increases.
- Various enhancements have been made to FETI algorithm to solve plate and shell problems, contacts, and multi-point constraints; and various performance-enhancing techniques are available for heterogeneous materials, multiple load conditioners, and incompressible conditions.

3.10 Recommendations

- Sparse direct solvers are recommended for small to medium size problems that in-core memory is sufficient to hold the factorized matrices. Furthermore, direct methods are mature, robust, and are true black-box solvers that do not require user interventions.
- Iterative solvers are more memory efficient, less sensitive to the problem size, and consequently enable larger problems to be solved on the same machine compared to direct solvers. However, iterative solvers may fail to converge, and should always be used with preconditioners to improve the rate and/or attainment of convergence.
- Iterative solvers are recommended when the problem is large and cannot be managed with direct methods due to either solution time or lack of memory resources. Again, preconditioners suited to the problem must be incorporated.
- With the availability of multi-processor computers, domain decomposition methods (DDM) can be used on large-scale problems. This overcomes situations where problems cannot be solved with direct or iterative methods in a timely manner. Nonoverlapping DDM are preferred to overlapping DDM for finite element problems as the latter require a more elaborate procedure to design and resolve domain overlaps.
- The FETI algorithm has a proven parallel efficiency and scalability, and has been successfully applied with computer clusters and achieves good parallel efficiency.
- The mechanically consistent Dirichlet preconditioner is recommended for FETI algorithm implementations on PC clusters interconnected with Ethernet, if there are no other constraints such as limited memory.
- For a PC cluster with high-speed, low-latency, interconnections, it is encouraged to try both mechanically consistent Dirichlet preconditioners and mechanically

consistent lumped preconditioners. In this case, a performance comparison is advantageous.

Table 3-1 Comparisons of Newton-Raphson with Automatic Sub-stepping and Explicit Integration with Auto-Stepping

	NR with automatic sub-stepping ⁽¹⁾	Explicit auto-stepping ⁽²⁾
Purpose for sub-stepping	Convergence behavior	Error control
Number of sub-steps	Less	More
Iterations per sub-step	≥ 1	0 – 2
Tangent stiffness	Consistent	Conventional
Sub-step size	Larger	Smaller
Sub-step size adjustment	Iterations to convergence	Magnitude of local error

(1) Based on ABAQUS non-linear solver

(2) Based on Algorithm by Abbo and Sloan (1996)

Table 3-2 Comparisons of Direct and Iterative Solvers

	Direct Solvers ⁽¹⁾	Iterative Solvers ⁽²⁾
Operations	Sub-matrix update	Matrix-vector product
Memory requirement	Large	Small
Solution time dependency	Problem size	Eigenvalue distribution
Precondition	Minimize fill-ins ⁽³⁾	Cluster eigenvalues ⁽⁴⁾
Problem size scaling, $t \propto N^p$	$p = 1.2 - 3$ ⁽⁵⁾	$p = 1.17 - 1.33$ ⁽⁶⁾
Robustness	Very good	Depend on Preconditioner

(1) Gauss elimination, banded solver, sparse solver, frontal solver, etc.

(2) CG, PCG, BICG, BICGSTAB, GMRES, GCR, etc.

(3) Profile reduction, MMD, AMD, Sloan algorithm, etc.

(4) Incomplete LU, Incomplete Cholesky, diagonal scaling, etc.

(5) 1.2 – 1.7 for multifrontal methods (Fish and Shek 1999)

(6) Conjugate gradient (Fish and Shek 1999)

Table 3-3 The PCPG Algorithm for Solving the FETI Interface Problem

1. Calculate λ_0 with Equation (3.40)	$\lambda_0 = \mathbf{G}_I (\mathbf{G}_I^T \mathbf{G}_I)^{-1} \mathbf{e}$
2. Evaluate initial residual	$\mathbf{r}_0 = \mathbf{d} - \mathbf{F}_I \lambda_0$
3. Start iteration process	$i = 0$
4. Project the residual into null space of \mathbf{G}_I	$\boldsymbol{\omega}_i = \mathbf{P} \cdot \mathbf{r}_i$
5. Precondition	$\mathbf{z}_i = \mathbf{M} \boldsymbol{\omega}_i$
6. Re-project the preconditioned search direction	$\mathbf{p}_i = \mathbf{P} \cdot \mathbf{z}_i$
7. Back-orthogonalize with all previous search directions	$\mathbf{p}_i = \mathbf{p}_i - \sum_{j=0}^{i-1} \frac{\mathbf{p}_i \cdot (\mathbf{F}_I \mathbf{p}_j)}{\mathbf{p}_j \cdot (\mathbf{F}_I \mathbf{p}_j)} \mathbf{p}_j$
8. Determine the weighting of the new search direction	$\zeta = \frac{\mathbf{p}_i \cdot \mathbf{r}_i}{\mathbf{p}_i \cdot (\mathbf{F}_I \mathbf{p}_i)}$
9. Update solution λ	$\lambda = \lambda + \zeta \mathbf{p}_i$
10. Update residual \mathbf{r}_i	$\mathbf{r}_{i+1} = \mathbf{r}_i - \zeta \mathbf{F}_I \mathbf{p}_i$
11. Go back to step 4 if not converged	$i = i + 1$

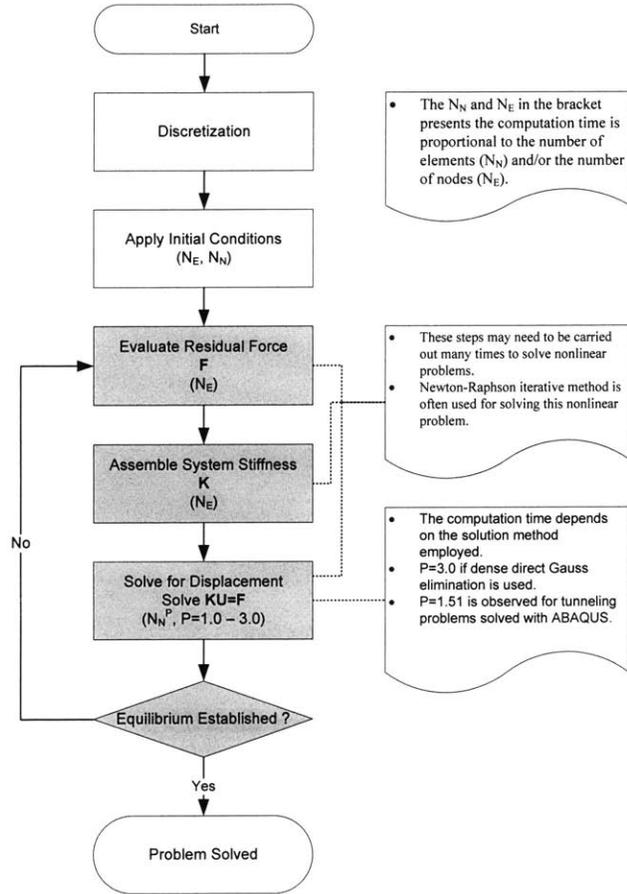


Figure 3-1 Summary of the Finite Element Procedure and its Complexity

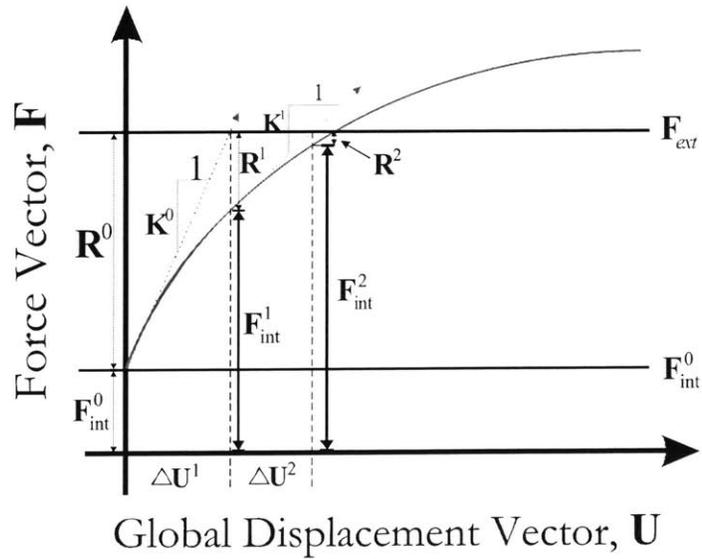


Figure 3-2 Newton-Raphson Iteration Scheme

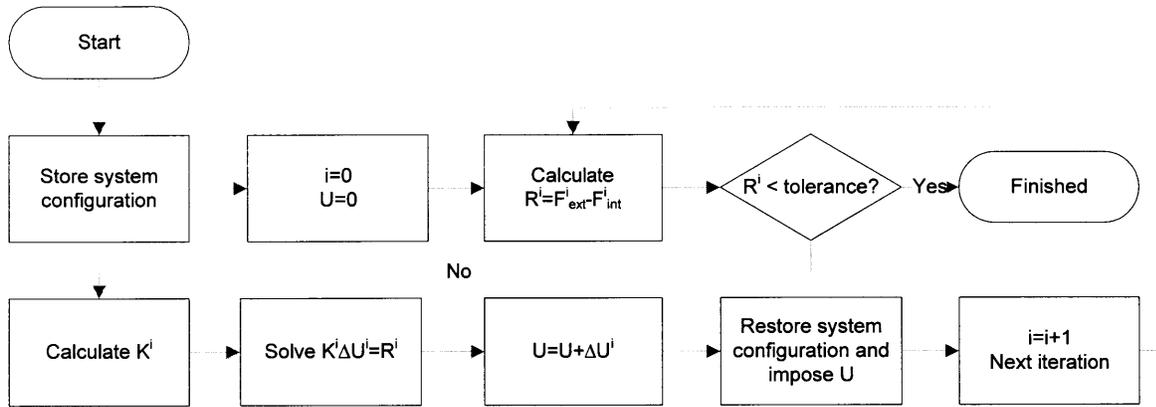


Figure 3-3 Flowchart for Newton Raphson Iteration

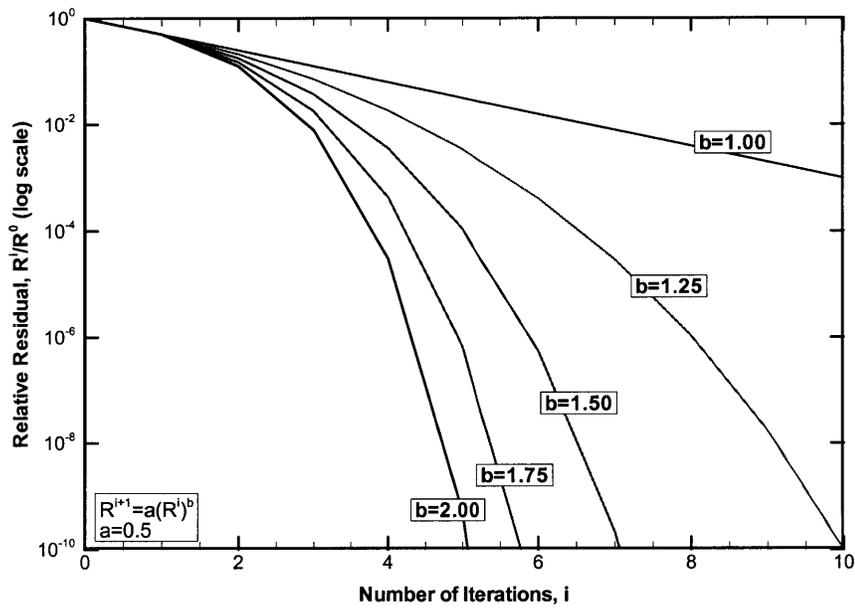


Figure 3-4 Convergence Rate for Iterative Solutions

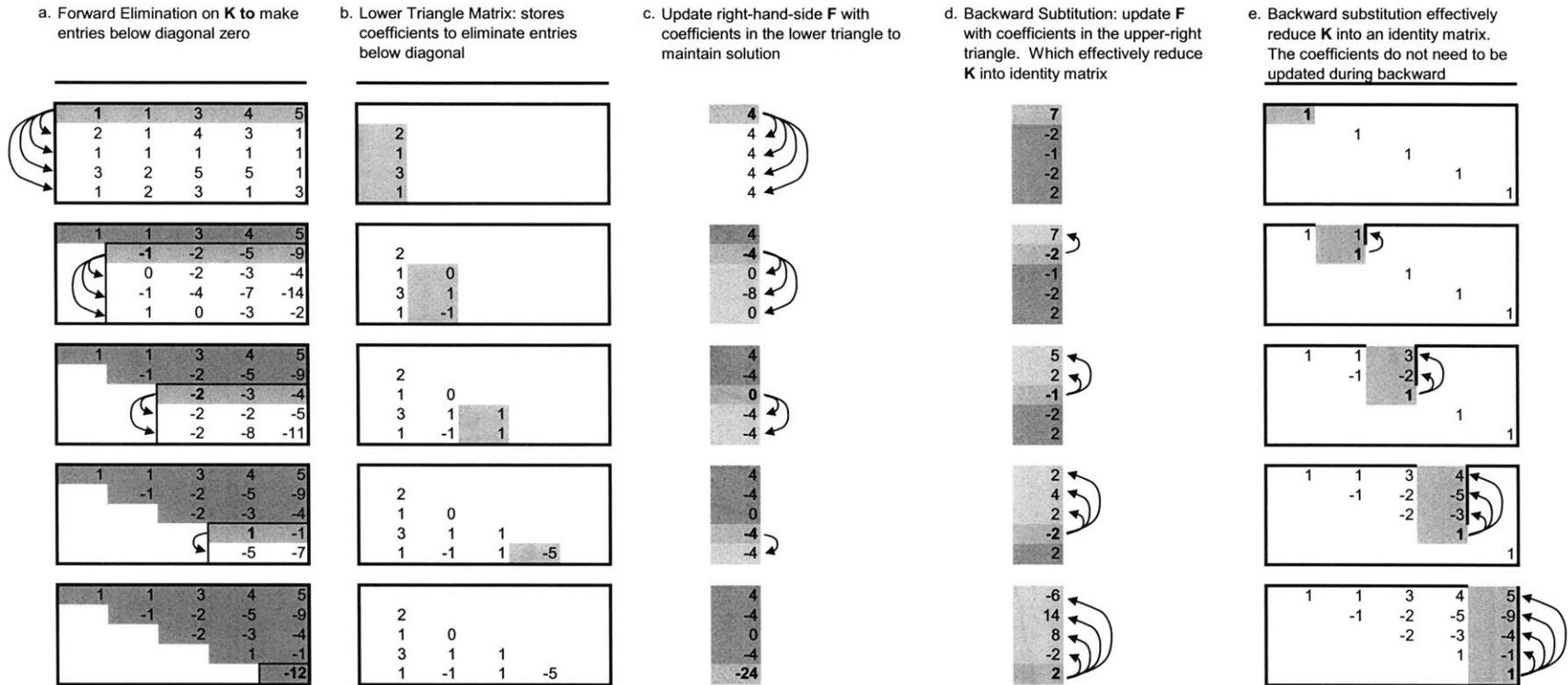


Figure 3-5 Illustration of Dense Gauss Elimination

1. *For* $i := 1, \dots, (N - 1)$ *Do* :
2. *For* $j := (i + 1), \dots, N$ *Do* :
3. $\mathbf{K}(j, i) = \mathbf{K}(j, i) / \mathbf{K}(i, i)$
4. *For* $k := (i + 1), \dots, N$ *Do* :
5. $\mathbf{K}(j, k) = \mathbf{K}(j, k) - \mathbf{K}(j, i)\mathbf{K}(i, k)$
6. *EndDo*
7. *EndDo*
8. *EndDo*

(a) Forward Elimination (LU factorization)

1. *For* $i := 1, 2, \dots, N - 1$ *Do* :
2. *For* $j := (i + 1), (i + 2), \dots, N$ *Do* :
3. $\mathbf{F}(j) = \mathbf{F}(j) - \mathbf{F}(i)\mathbf{K}(j, i)$
4. *EndDo*
5. *EndDo*

6. *For* $i := N, N - 1, \dots, 1$ *Do* :
7. $\mathbf{F}(i) = \mathbf{F}(i) / \mathbf{K}(i, i)$
8. *For* $j := (i - 1), (i - 2), \dots, 1$ *Do* :
9. $\mathbf{F}(j) = \mathbf{F}(j) - \mathbf{F}(i)\mathbf{K}(j, i)$
10. *EndDo*
11. *EndDo*

(b) Forward elimination and backward substitution on \mathbf{F} for solving \mathbf{U}

Figure 3-6 Pseudo Code for the Dense Gauss Elimination

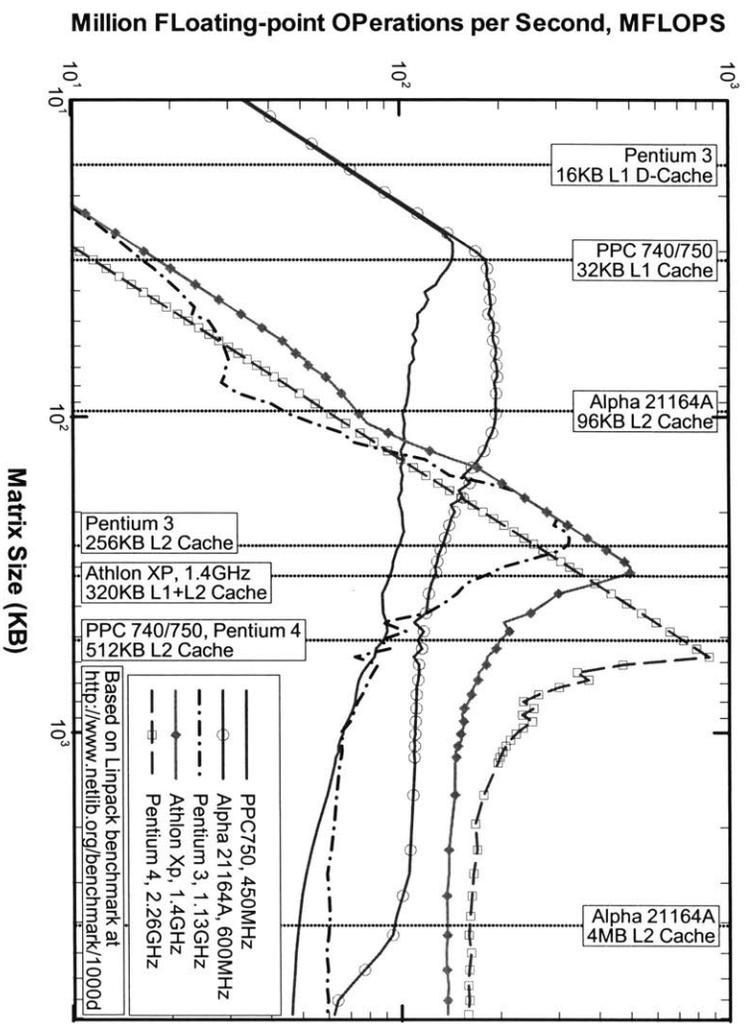
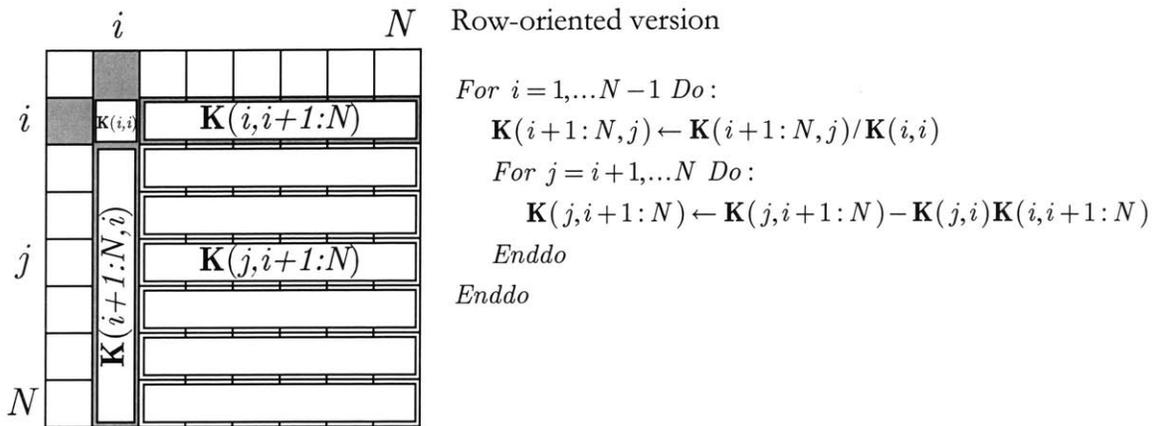


Figure 3-7 Effect of Memory Hierarchy

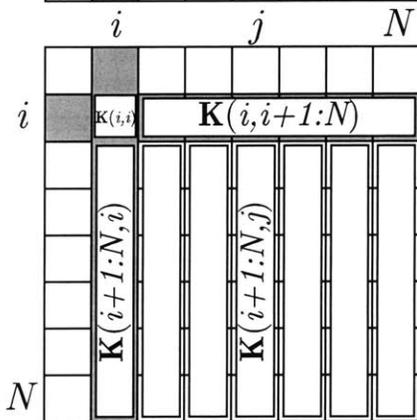


Row-oriented version

```

For i = 1, ... N - 1 Do :
    K(i+1:N, j) ← K(i+1:N, j) / K(i, i)
    For j = i+1, ... N Do :
        K(j, i+1:N) ← K(j, i+1:N) - K(j, i) K(i, i+1:N)
    Enddo
Enddo

```



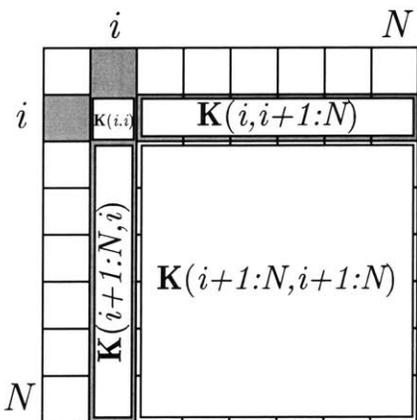
Column-oriented version

```

For i = 1, ... N - 1 Do :
    K(i+1:N, j) ← K(i+1:N, j) / K(i, i)
    For j = i+1, ... N Do :
        K(j, i+1:N) ← K(j, i+1:N) - K(j, i) K(i, i+1:N)
    Enddo
Enddo

```

(a) Dense Gauss Elimination Using BLAS-1 Update



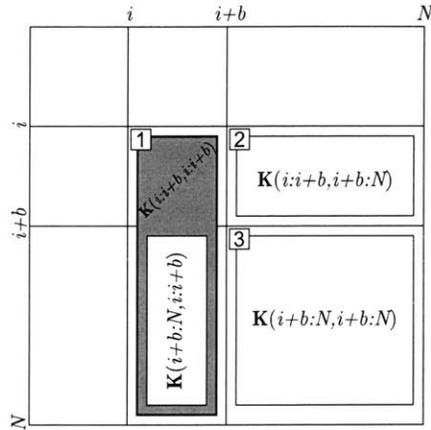
```

For i = 1, ... N Do :
    K(i+1:N, i) ← K(i+1:N, i) / K(i, i)
    K(i+1:N, i+1:N) ←
        K(i+1:N, i+1:N) - K(i+1:N, i) K(i, i+1:N)
Enddo

```

(b) Dense Gauss Elimination Using BLAS-2 Update

Figure 3-8 Variants of Dense Gauss Elimination



For $i := 1, \dots, N$ step b Do :

If $(i+b) \geq N$ then

$K(i:N, i:i+b) \leftarrow$ BLAS-2 Gauss Elimination on $K(i:N, i:N)$

Else

1 $K(i:N, i:i+b) \leftarrow$ BLAS-2 Gauss Elimination on $K(i:N, i:i+b)$

2 $K(i:i+b, i+b:N) \leftarrow$ Triangular solve on $K(i:i+b, i+b:N)$
with lower-left of $K(i:i+b, i:i+b)$

3 $K(i+b:N, i+b:N) \leftarrow K(i+b:N, i+b:N)$
 $- K(i:i+b, i+b:N)K(i+b:N, i:i+b)$

End if

EndDo

(c) Dense Gauss Elimination Using BLAS-3 Update

Figure 3-8 (Cont'd) Variants of Dense Gauss Elimination

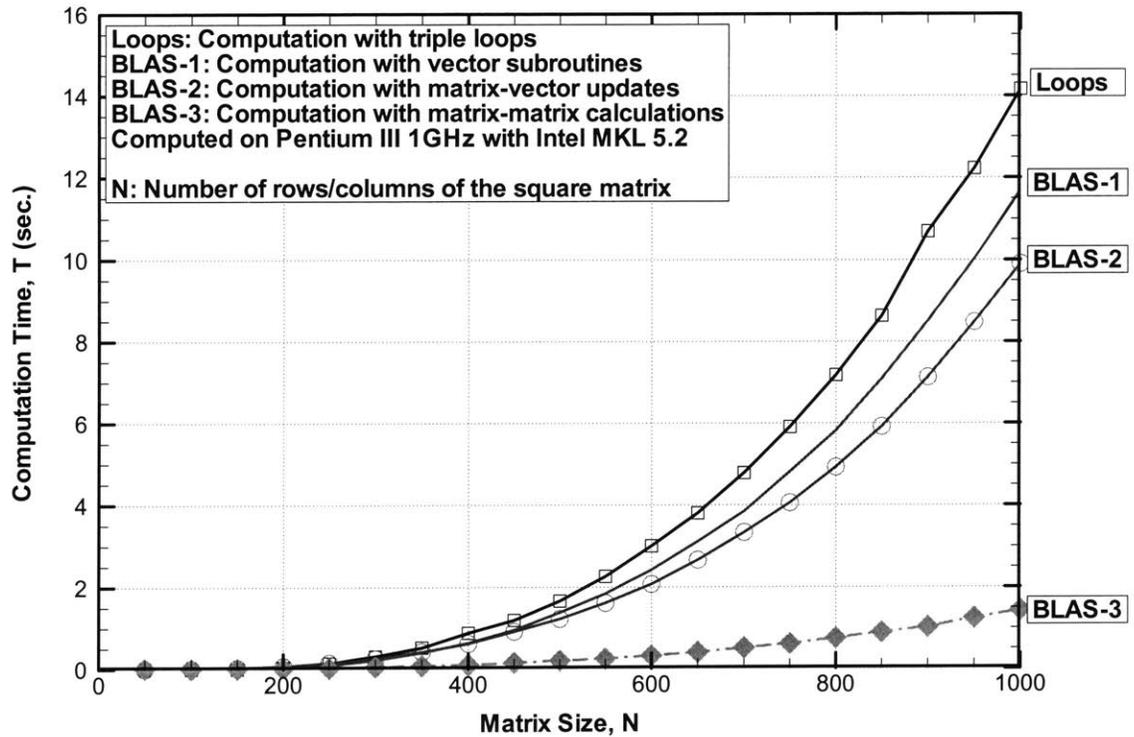


Figure 3-9 Effect of Optimization Using BLAS on Dense Gauss Elimination

1. For $i = 1, 2, \dots, \text{maxIT}$ Do :
2. Determine a vector ω_i
3. Determine a corresponding weighting α_i to minimize residual $R = \|\mathbf{r}_i\|_2$
 where $\mathbf{r}_i = \mathbf{F} - \mathbf{K}\mathbf{x}_i$, $\mathbf{x}_i = \sum_{j=1}^i \alpha_j \mathbf{w}_j$
4. If $R < \text{tolerance}$ goto 6
5. EndDo
6. End

Figure 3-10 Flow Chart for General Subspace Method

1. Choose a vector \mathbf{v}_1 of norm 1
2. For $j = 1, 2, \dots, m$ Do :
3. Compute $\mathbf{w}_j := \mathbf{A}\mathbf{v}_j$
4. For $i = 1, \dots, j$ Do :
5. $\mathbf{h}_{ij} = (\mathbf{w}_j, \mathbf{v}_i)$
6. $\mathbf{w}_j := \mathbf{w}_j - \mathbf{h}_{ij}\mathbf{v}_i$
7. EndDo
8. $\mathbf{h}_{j+1,j} = \|\mathbf{w}_j\|_2$. If $\mathbf{h}_{j+1,j} = 0$ Stop
9. $\mathbf{v}_{j+1} = \mathbf{w}_j / \mathbf{h}_{j+1,j}$
10. EndDo

Figure 3-11 Arnoldi-Modified Gram-Schmidt Algorithm
(Saad 1996)

1. Compute $r_0 := b - Ax_0, p_0 := r_0$
2. For $j = 0, 1, 2, \dots$, until convergence Do :
3. $\alpha_j := (r_j \cdot r_j) / (Ap_j \cdot p_j)$
4. $x_{j+1} := x_j + \alpha_j p_j$
5. $r_{j+1} := r_j - \alpha_j Ap_j$
6. $\beta_j := (r_{j+1} \cdot r_{j+1}) / (r_j \cdot r_j)$
7. $p_{j+1} := r_{j+1} + \beta_j p_j$
8. EndDo

Figure 3-12 Conjugate Gradient (CG) Algorithm
(Saad 1996)

1. Compute $r_0 := b - Ax_0, p_0 := r_0$
2. For $j = 0, 1, 2, \dots$, until convergence Do :
3. $\alpha_j := (r_j \cdot Ap_j) / (Ap_j \cdot Ap_j)$
4. $x_{j+1} := x_j + \alpha_j p_j$
5. $r_{j+1} := r_j - \alpha_j Ap_j$
6. Compute $\beta_{ij} := (Ar_{j+1} \cdot Ap_i) / (Ap_i \cdot Ap_i)$, for $i = 0, 1, \dots, j$
7. $p_{j+1} := r_{j+1} + \sum_{i=0}^j \beta_{ij} p_i$
8. EndDo

Figure 3-13 Generalized Conjugate Residual (GCR) Algorithm
(Saad 1996)

1. Compute $r_0 := b - Ax_0, z_0 := \mathbf{P}r_0, p_0 := z_0$
2. For $j = 0, 1, 2, \dots$, until convergence Do :
3. $\alpha_j := (r_j \cdot z_j) / (Ap_j \cdot p_j)$
4. $x_{j+1} := x_j + \alpha_j p_j$
5. $r_{j+1} := r_j - \alpha_j Ap_j$
6. $z_{j+1} := \mathbf{P}r_{j+1}$
7. $\beta_j := (r_{j+1} \cdot z_{j+1}) / (r_j \cdot z_j)$
8. $p_{j+1} := z_{j+1} + \beta_j p_j$
9. EndDo

Figure 3-14 Left Preconditioned Conjugate Gradient Algorithm
(Saad 1996)

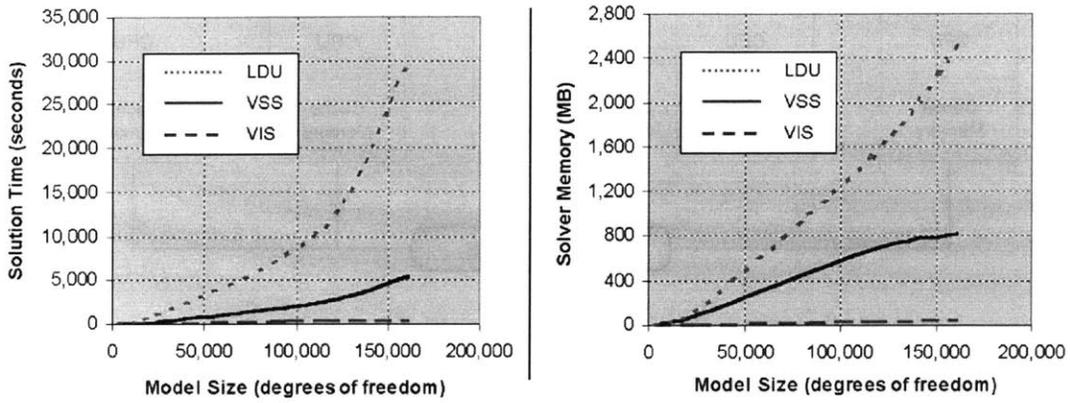


Figure 3-15 Comparisons of Solvers in NE/Nastran
(Weinberg 2000)

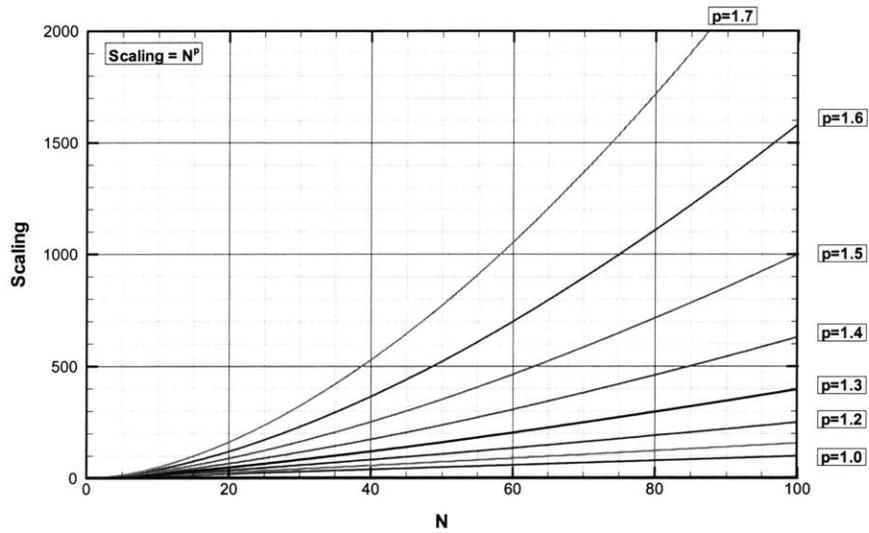


Figure 3-16 Solution Time Scaling with Problem Size N

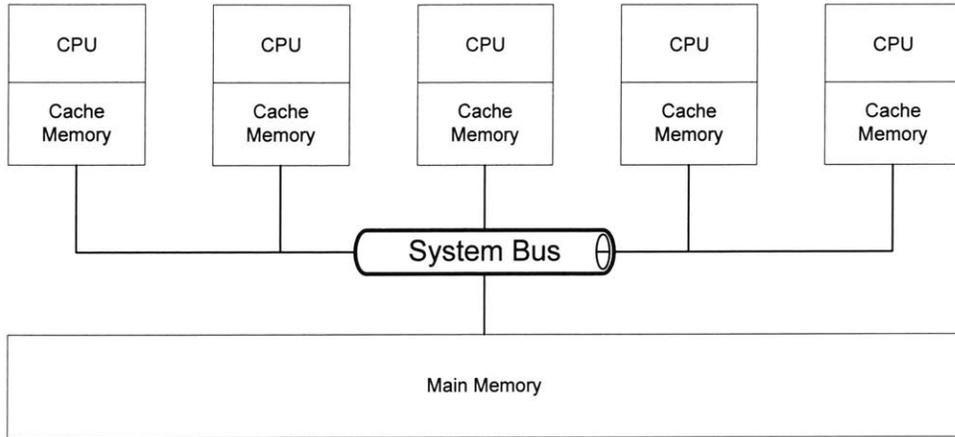


Figure 3-17 Shared Memory Architecture

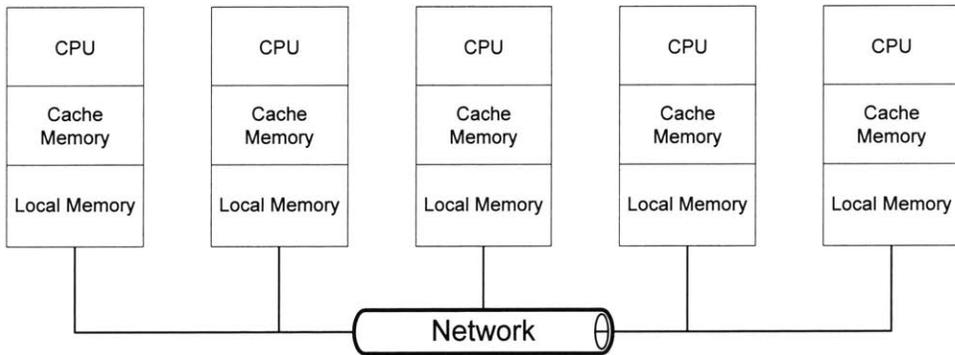


Figure 3-18 Distributed Memory Architecture

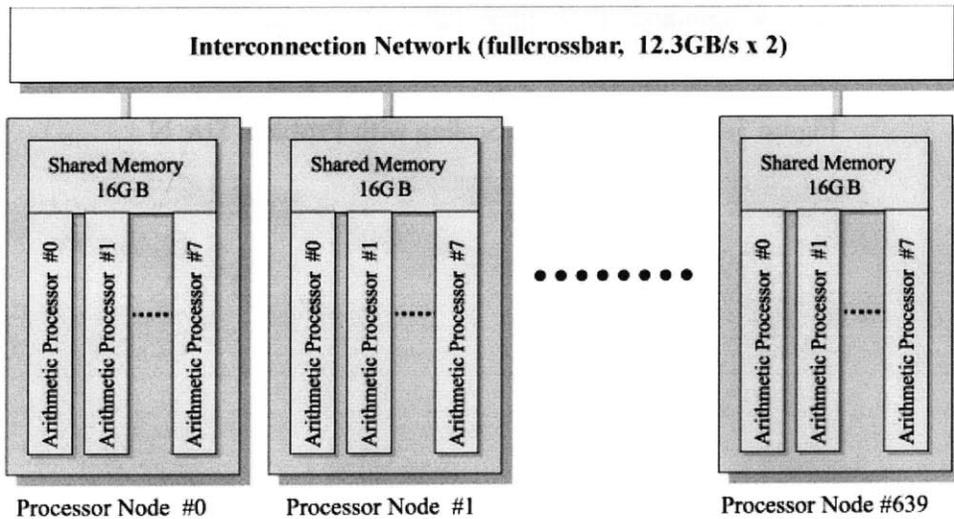
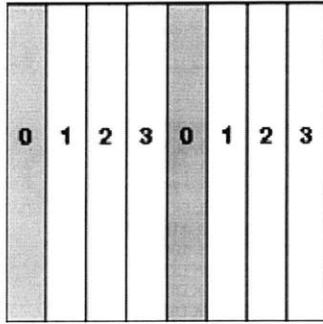
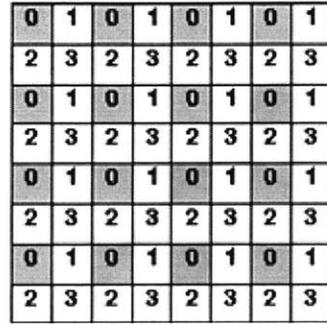


Figure 3-19 System architecture for Earth Simulator
<http://www.es.jamstec.go.jp/esc/eng/Hardware/system.html>

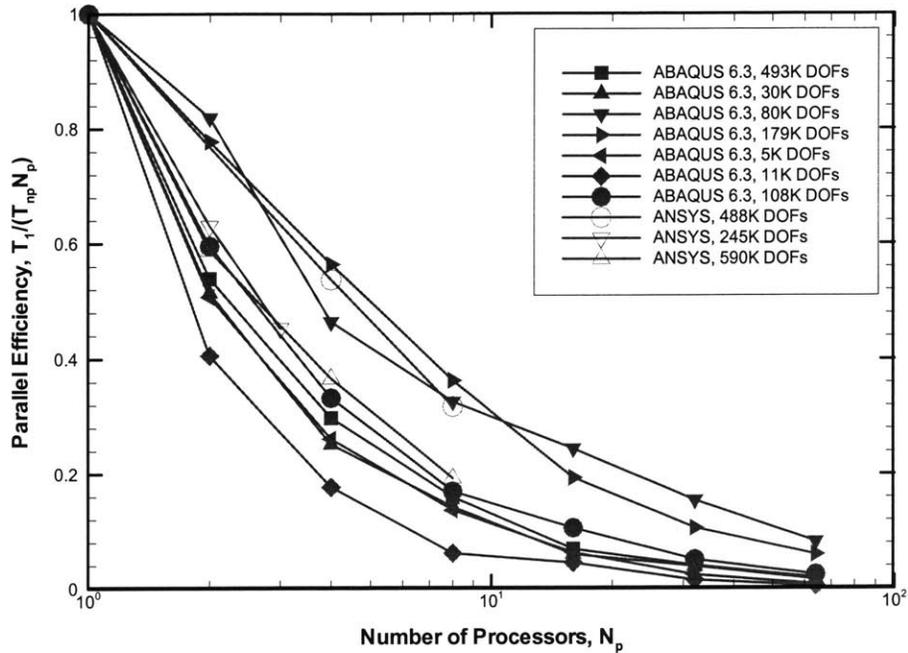


(a) 1D Block-cyclic layout



(b) 2D Block-cyclic layout

Figure 3-20 1D and 2D Block-cyclic Layout for 4 Processors
(Blackford et al. 1997)



*ABAQUS data from <http://www.abaqus.com/products/performance63.html>
 ** ANSYS data from Poole et al. (2003)
 *** Running on SGI O2000 (ANSYS) or HP Superdome-64(SD64K) (ABAQUS)

Figure 3-21 Parallel Efficiency of Direct Solvers in Commercial Finite Element Codes

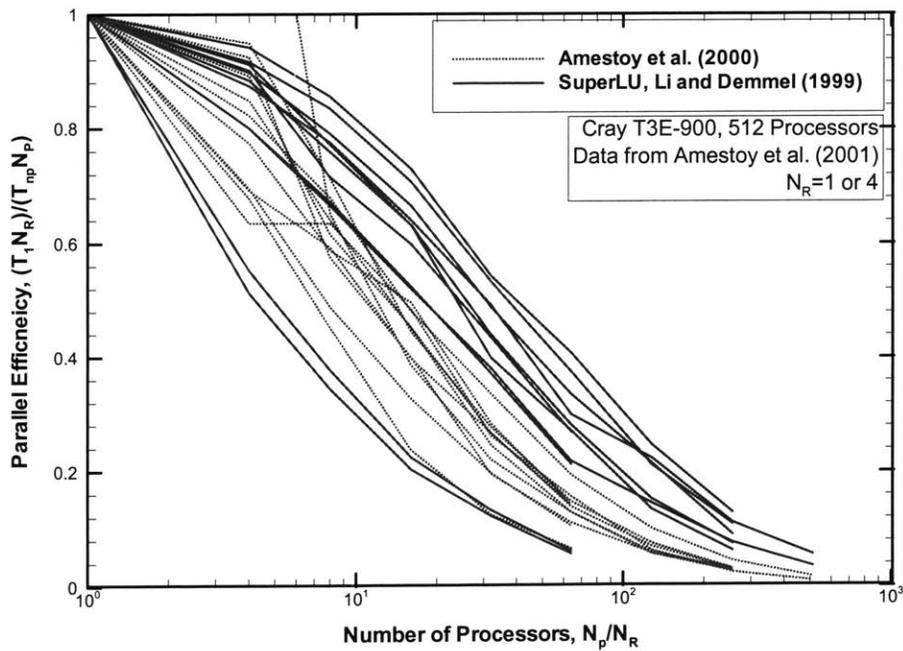


Figure 3-22 Parallel Efficiency Degradation on Cray T3E-900

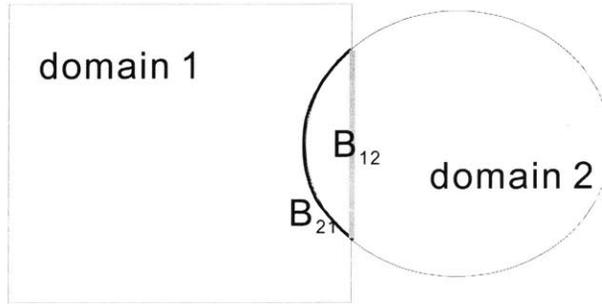


Figure 3-23 Schwarz Alternating Scheme

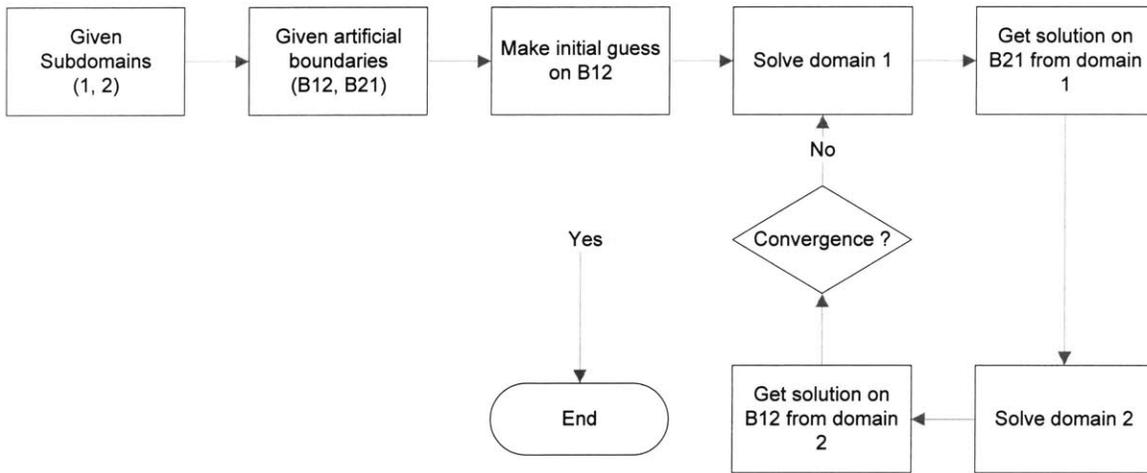


Figure 3-24 Flowchart for Schwarz Alternating Scheme

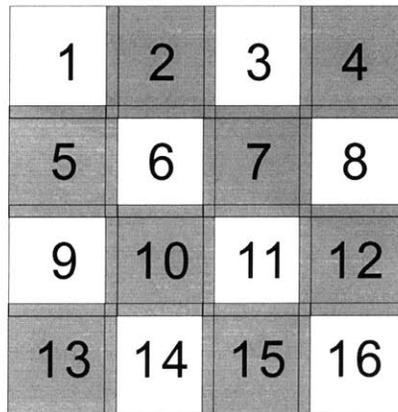


Figure 3-25 Schwarz Alternating Scheme for Multiple Sub-domains

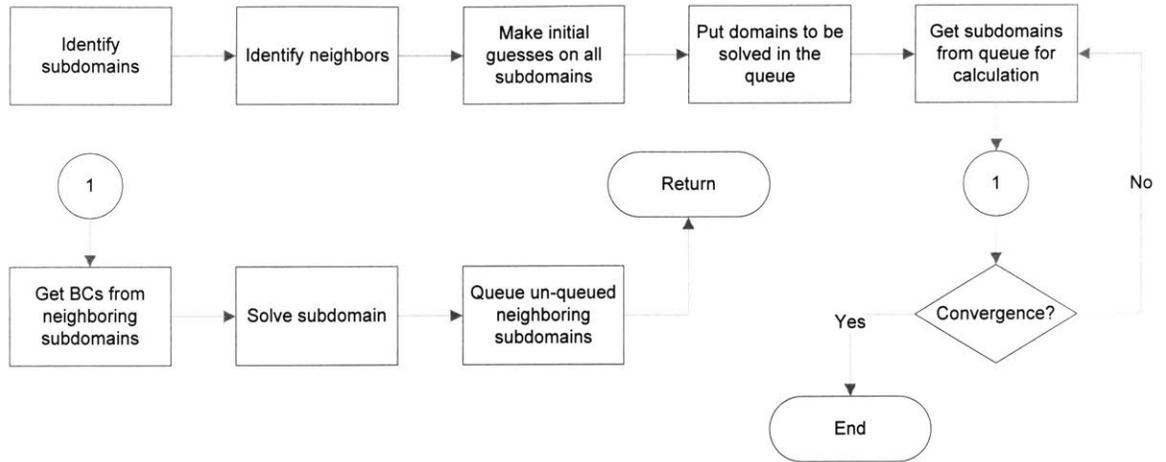


Figure 3-26 Flowchart for Multiple Sub-domain Schwarz Alternating Scheme

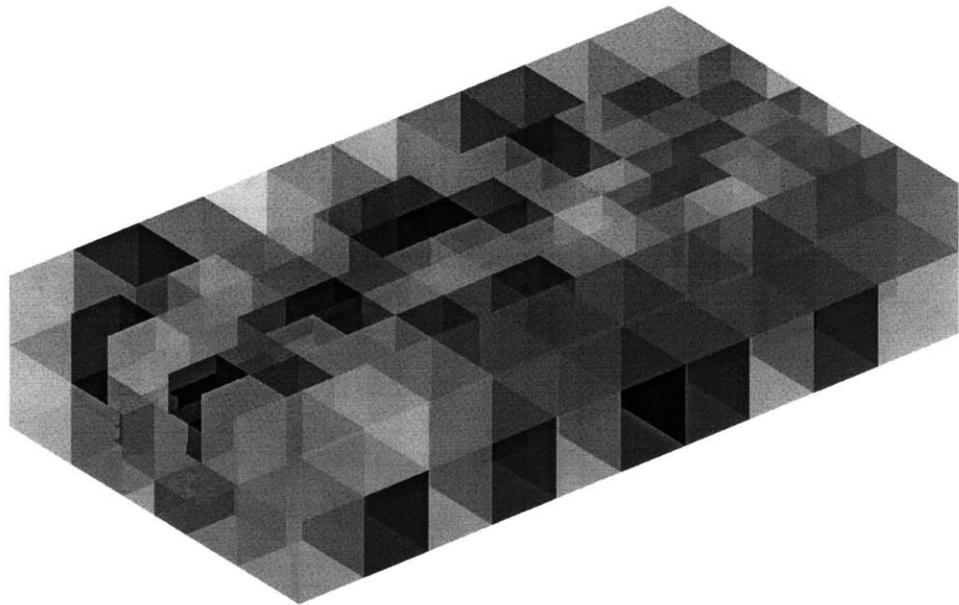


Figure 3-27 Non-overlapping Domain Decomposition Method

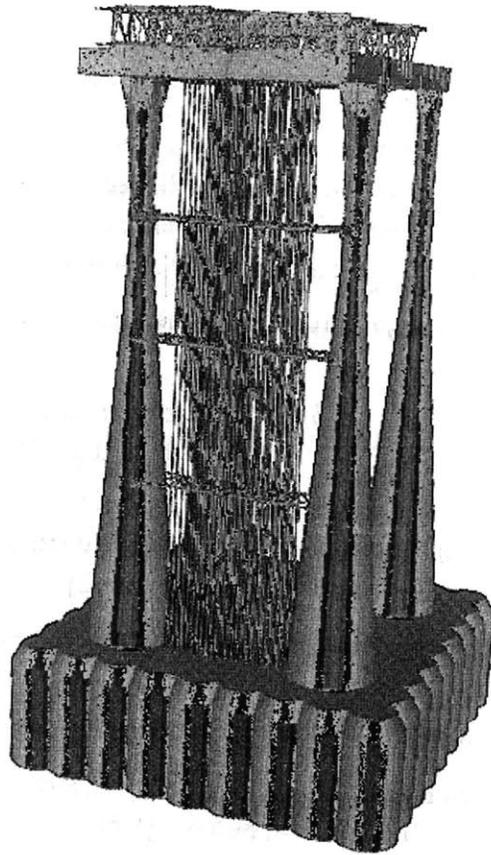


Figure 3-28 Offshore Oil Platform
(Smith et al. 1996)

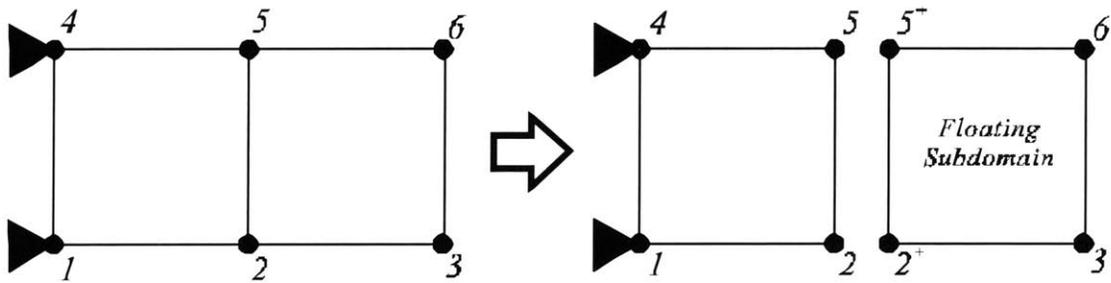


Figure 3-29 Two Sub-domain Cantilever Problem

Reference

- Abbo, A. J., and Sloan, S. W. (1996). "Automatic load stepping algorithm with error control." *International Journal for Numerical Methods in Engineering*, 39(10), 1737-1759.
- Amestoy, P., Davis, T. A., and Duff, I. S. (1996). "An approximate minimum degree ordering algorithm." *SIAM Journal on Matrix Analysis and Applications*, 17, 886-905.
- Amestoy, P. R., Duff, I. S., and L'Excellent, J.-Y. (2000). "Multifrontal parallel distributed symmetric and unsymmetric solvers." *Comput. Methods Appl. Mech. Eng.*, 501-520.
- Arnoldi, W. E. (1951). "The principle of minimized iteration in the solution of the matrix eigenvalue problem." *Quart. Appl. Math.*, 9, 17-29.
- Ashcraft, C., Grimes, R., Lweis, J., Peyton, B., and Simon, H. (1987). "Progress in sparse matrix methods for large sparse linear systems on vector supercomputers." *International Journal of Supercomputer Applications*, 1, 10-30.
- Bhardwaj, M., Day, D., Farhat, C., Lesoinne, M., Pierson, K., and Rixen, D. (2000). "Application of the FETI method to ASCI problems - scalability results on 1000 processors and discussion of highly heterogeneous problems." *International Journal for Numerical Methods in Engineering*, 47(1-3), 513-535.
- Blackford, L. S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongara, J., Hammarling, S., Henry, G., Petitet, A., Stanley, K., Walker, D., and Whaley, R. C. (1997). *Scalpack Users' Guide*, Society for Industrial & Applied Mathematics.
- Charmpis, D. C., and Papadrakakis, M. (2002). "Enhancing the performance of the FETI method with preconditioning techniques implemented on clusters of networked computers." *Computational Mechanics*, 30(1), 12-28.
- Cuthill, E., and McKee, J. "Reducing the bandwidth of sparse symmetric matrices." *24th ACM National Conference*, 157-172.
- Dongarra, J. (1997). "Online Lecture Notes, <http://www.cs.utk.edu/~dongarra/WEB-PAGES/SPRING-2000/lect08.pdf>."
- Duff, I., and Reid, J. (1983). "The multifrontal solution of indefinite sparse symmetric linear equations." *ACM Trans. Mathematical Software*, 9, 302-325.
- Duff, I. S. (1996). "Sparse numerical linear algebra: direct methods and preconditioning." *Technical Report TR/PA/96/22*.
- Dureisseix, D., and Farhat, C. (2001). "A numerically scalable domain decomposition method for the solution of frictionless contact problems." *International Journal for Numerical Methods in Engineering*, 50(12), 2643-2666.
- Farhat, C., Chen, P. S., Mandel, J., and Roux, F. X. (1998a). "The two-level FETI method Part II: Extension to shell problems, parallel implementation and performance results." *Computer Methods in Applied Mechanics and Engineering*, 155(1-2), 153-179.
- Farhat, C., Crivelli, L., and Roux, F. X. (1994). "Extending Substructure Based Iterative Solvers to Multiple Load and Repeated Analyses." *Computer Methods in Applied Mechanics and Engineering*, 117(1-2), 195-209.
- Farhat, C., Lacour, C., and Rixen, D. (1998b). "Incorporation of linear multipoint constraints in substructure based iterative solvers. Part 1: A numerically scalable algorithm." *International Journal for Numerical Methods in Engineering*, 43(6), 997-1016.
- Farhat, C., Lesoinne, M., and Pierson, K. (2000). "A scalable dual-primal domain decomposition method." *Numerical Linear Algebra with Applications*, 7(7-8), 687-714.

- Farhat, C., and Mandel, J. (1998). "The two-level FETI method for static and dynamic plate problems Part I: An optimal iterative solver for biharmonic systems." *Computer Methods in Applied Mechanics and Engineering*, 155(1-2), 129-151.
- Farhat, C., and Roux, F. X. (1991). "A Method of Finite-Element Tearing and Interconnecting and Its Parallel Solution Algorithm." *International Journal for Numerical Methods in Engineering*, 32(6), 1205-1227.
- Farhat, C., and Roux, F.-X. (1994). "Implicit Parallel Processing in Structural Mechanics." *Computational Mechanics Advances* 2, 1-124.
- Fish, J., and Shek, K. L. (1999). "Computational Aspects of Incrementally Objective Algorithms for Large Deformation Plasticity." *International Journal for Numerical Methods in Engineering*, 44, 839-851.
- George, A. (1973). "Nested dissection of a regular finite element mesh." *SIAM Journal Numer. Anal.*, 10, 345-363.
- Gosset, P., Rey, C., and Rixen, D. J. (2003). "On the initial estimate of interface forces in FETI methods." *Computer Methods in Applied Mechanics and Engineering*, 192(25), 2749-2764.
- Hsieh, Y.-M., and Whittle, A. J. "A computational strategy for solving three-dimensional tunnel excavation problems." *Second M.I.T. Conference on Computational Fluid and Solid Mechanics*, Cambridge, Massachusetts, USA.
- Jeremic, B., and Sture, S. (1997). "Implicit integrations in elastoplastic geotechnics." *Mechanics of Cohesive-Frictional Materials*, 2(2), 165-183.
- Lawson, C. L., Hanson, R. J., Kincaid, D., and Krogh, F. T. (1979). "Basic linear algebra subprograms for FORTRAN usage." *ACM Transactions on Mathematical Software*, 5(3), 308-323.
- Li, X. S., and Demmel, J. W. "A scalable sparse direct solver using static pivoting." *Ninth SIAM Conference on Parallel Processing for Scientific Computing*, San Antonio, Texas, U.S.A.
- Li, X. S., and Demmel, J. W. (2003). "SuperLU DIST: A Scalable Distributed-Memory Sparse Direct Solver for Unsymmetric Linear Systems." *ACM Transactions on Mathematical Software*, 29(2), 110-140.
- Liu, J. W.-H. (1985). "Modification of the minimum degree algorithm by multiple elimination." *ACM Trans. Mathematical Software*, 11, 141-153.
- Mathies, H., and Strang, G. (1979). "Solution of non-linear finite element equations." *International Journal for Numerical Methods in Engineering*, 14(11), 1613-1626.
- Poole, G., Liu, Y.-C., and Mandel, J. (2003). "Advancing analysis capabilities in Ansys through solver technology." *Electronic Transactions on Numerical Analysis*, 15, 106-121.
- Przemieniecki, J. S. (1963). "Matrix structural analysis of substructures." *Am. Inst. Aero. Astro. J.*, 1, 138-147.
- Rey, C., and Lene, F. "Reuse of Krylov Spaces in the Solution of Large-scale Nonlinear Elasticity Problems." *Ninth International Conference on Domain Decomposition Methods*, 466-470.
- Rixen, D. J., and Farhat, C. (1999). "A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems." *International Journal for Numerical Methods in Engineering*, 44(4), 489-516.
- Saad, Y. (1996). *Iterative Methods for Sparse Linear Systems*, PWS Publishing Co.
- Saad, Y., and Vorst, H. A. v. d. (2000). "Iterative solution of linear systems in the 20-th century." *Journal of Computational and Applied Mathematics*, 123(1-2), 1-33.
- Schwarz, H. A. (1869). "Über einige Abbildungsaufgaben." *Ges. Math. Abb.*, 11, 65-83.

- Sheng, D., and Sloan, S. W. (2003). "Time stepping schemes for coupled displacement and pore pressure analysis." *Computational Mechanics*, 31(1-2 SPEC), 122-134.
- Simo, J. C., and Taylor, R. L. (1985). "Consistent tangent operators for rate-independent elastoplasticity." *Computer Methods in Applied Mechanics and Engineering*, 48(1), 101-118.
- Sloan, S. W., and Abbo, A. J. (1999). "Biot consolidation analysis with automatic time stepping and error control. Part 2: Applications." *International Journal for Numerical and Analytical Methods in Geomechanics*, 23(6), 493-529.
- Smith, B., Bjorstad, P., and Gropp, W. (1996). *Domain Decomposition : Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press; (June 13, 1996).
- Vereecke, B., Bavestrello, H., and Dureisseix, D. (2003). "An extension of the FETI domain decomposition method for incompressible and nearly incompressible problems." *Computer Methods in Applied Mechanics and Engineering*, 192(31-32), 3409-3429.
- Watson, W. R., and Storaasli, O. O. (2000). "Application of NASA general-purpose solver to large-scale computations in aeroacoustics." *Advances in Engineering Software*, 31(8), 555-561.
- Weinberg, D. J. (2000). "Performance assessment of NE/Nastran's new sparse direct and iterative solvers." *Advances in Engineering Software*, 31(8), 547-554.

Chapter 4

Efficient Finite Element Solutions for Tunneling

4.1 Introduction

This chapter explores techniques to achieve efficient large-scale three-dimensional finite element solutions for tunneling problems. The chapter starts with an analysis of an example problem consisting of an unsupported shallow circular tunnel excavated in a cohesive soil using a conventional direction solution technique (with ABAQUSTM). The effectiveness of using an automated adaptive remeshing scheme is then evaluated. Alternative approaches using parallel computation is then explored using overlapping domain decomposition (Schwarz alternating scheme) and non-overlapping domain decomposition (FETI method, Finite Element Tearing and Interconnecting, Farhat and Roux (1991)). Finally, a full finite element program is developed using the one-level FETI method, and the performance of the analysis is evaluated.

4.2 Base-case Analysis of 3-D Tunnel Excavation

In order to understand the computation time needed for a three-dimensional tunneling analysis with ABAQUS¹, a base-case example was studied in detail. The base-case example models the excavation of an unlined circular tunnel with a radius $R = 3m$, with a centerline embedment depth, $H = 6m$, below the ground surface. The surrounding soil is a low-permeability cohesive soil undergoing undrained shearing². Calculations are performed assuming elastic and elastic-perfectly-plastic material behavior, and Table 4-1 lists reference material properties and K_0 conditions.

The 3-D finite element mesh shown in Figure 4-1 is extruded from a 60m by 60m two-dimensional mesh with an extrusion length of 120m, generating 30 layers of finite elements

¹ A commercial general-purpose finite element product of ABAQUS, Inc.

² In this case, the clay is effectively incompressible. The current analysis approximates this condition with Poisson's ratio $\nu = 0.49$

each with a layer thickness of 4m. The resultant finite element mesh consists of 8010 elements and 36030 nodes, resulting in a total of over 100,000 degrees of freedom. The tunnel excavation through the soil medium is modeled, in 30 steps, with an excavation length $L_{exca} = 4m$ per step.

4.2.1 Computation Requirement

The analysis of the finite element model shown in Figure 4-1 was performed using ABAQUS 5.8-18. The code uses a sparse direct solver (Gauss elimination) with a multi-frontal technique to exploit the sparsity of the stiffness matrix.

Table 4-2 shows the computation time and storage requirements for the base-case analyses performed on a Digital personal workstation 600au with a 600MHz 21164 Alpha processor, 4MB level-3 cache, 1088MB RAM, with SCSI interfaced hard drives. To isolate the computation time from other activities occurring on the computer, only the CPU time for the analysis is reported in Table 4-2.

Reduced integration (Zienkiewicz et al. 1971) is used frequently in finite element analyses to avoid the locking phenomenon (Malkus and Hughes 1978) caused by overly stiff response of finite elements in bending, for nearly incompressible conditions (e.g. $\nu \rightarrow 0.5$). It is also used to reduce memory consumption and computation time by simplifying the evaluation of numerical integrals. However, reduced integration should be avoided for places where stress concentrations are expected.

The results summarized in Table 4-2 show a CPU time of 4 hours, 41 minutes for the elastic analysis with reduced integration, compared to 5 hours, 11 minutes for the same configuration with full integration. The computation time difference between full integration and reduced integration lies in the evaluation of equilibrium conditions performed on all finite elements within the model. Although the latter requires more than three times the total number of numerical integrations (27 Gauss points vs. 8 Gauss points), the total computation time only increases by 11%.

Further calculations using the elasto-plastic soil model require more than five times the computation (CPU) time. This increase is caused by the Newton-Raphson iterations required to solve the non-linear material response to the tunnel excavation using the default

convergence criterion. An average of five iterations per excavation step is observed to converge. It should be noted that the base-case example is mechanically stable (i.e., no lining is necessary to support the tunnel cavity). More iteration steps may be required to achieve equilibrium in weaker soils (that may need lining or other ground support) where face stability is marginal and will result in longer computation times.

The results in Table 4-2 show that the run-time is roughly proportional to the number of iterations and less dependent on the integration scheme for the example problem. This suggests that the most time consuming part of the calculation lies in the equation solving process, namely the Gauss elimination procedure employed by ABAQUS¹.

4.2.2 Results of Base-Case Example

Boundary effects play an important role in deformation analyses, especially in geotechnical problems, which require the modeling of large domains (usually limited by the underlying rock layers). There are three standard techniques to model infinite domains: 1) conventional extended finite element meshes with displacements defined for the far field boundaries, 2) use of infinite finite elements (Zienkiewicz et al. 1983), or 3) coupling the finite element method with the boundary element method (Zienkiewicz et al. 1977). However, both boundary elements and infinite elements require special formulations, and are only available for linear elastic analyses or for elasto-plastic calculations with boundaries beyond the zone of plasticity around the tunnel. These latter two options may not be available in standard finite element packages, and therefore applying roller-type fixed boundary conditions in the far field is conventionally applied for geotechnical engineering analyses.

Figure 4-2 shows the boundary effects observed in the base-case elastic analysis. The three curves shown in Figure 4-2 represent the incremental ground surface vertical displacements at the centerline between excavation steps ($\Delta X / R \cong 1.3$) observed at three points (A, B, and C):

¹ This is no particular surprise. As Figure 3-1 suggests equation solving scales super-linearly with the number of degrees of freedom, and ABAQUS/Standard user's manual states "*Linear equation solution in ABAQUS is often the most time consuming part of the analysis (especially for large models), and the storage of the equations occupies the largest part of the disk space during the calculations*".

- A. A fixed observation point above the tunnel entry
- B. A moving observation point above the tunnel face
- C. A fixed observation point above the tunnel exit

From Figure 4-2, it is seen that the fixed observation points A and C on the boundaries continue to observe excavation effects when the tunnel face is up to $X/2R = 8$ away from these boundaries; while the movements above the heading (B) is affected by the boundaries at a distance $X/2R = 4$. It is also seen from Figure 4-2 the steady state has a constant incremental settlement of $\Delta Z/2R = 0.005\%$. Figure 4-3 reports similar computed surface movements for the undrained, elasto-plastic analyses at the same three observation points. In this case, point A above the tunnel entry observes incremental movements of $\Delta Z/2R = 1\%$ when the tunnel face is $X/2R = 2$ away from the tunnel entry boundary. The steady state is not reached until the tunnel face reaches $X/2R = 10$.

The last incremental ground movement in Figure 4-3 should correspond to the convergence from a 3-D solution to a 2-D solution, as the final geometrical configurations are identical. It is seen a relatively large incremental settlement occurs at the exit boundary, suggesting that 3-D settlements are smaller than for 2-D solutions. This is because the unexcavated ground in front of the tunnel heading acts as “support” for the tunnel heading. Lee and Rowe (1990a) reported the stress condition in front of the tunnel face (corresponding to the “supporting ground”) to a limited extent is close to a triaxial compression mode based on observations made in their three-dimensional finite element analyses.

Based on the observations, it can be concluded that boundary effects extend at least 10-tunnel diameters before a steady state deformation pattern can be developed. The Boundary-Effect-Free (BEF) distance does, however, depend on several factors, including the constitutive soil behavior, tunnel shape, depth, soil profile (stiffness and strength), and in-situ stress conditions.

Figure 4-4 shows the sectional displacement contours after the tunnel is fully excavated in the elasto-plastic medium, and the boundary effects can be on the first few sections, as there are visible differences from the rest sections on the top of the tunnel crown.

4.2.3 Ground Deformation

Figure 4-5 shows the surface settlement profile for elasto-plastic analyses, with full integration, above the centerline of the tunnel axis relative to the tunnel heading. It is seen that the surface settlement profile converges to a single curve, except when the tunnel face is close to entry and exit boundaries. A small amount of uplift (less than 0.2% of the tunnel diameter, $2R$) is observed ahead of the tunnel face, while the steady state ground settlement of $\Delta Z/2R \cong 3.8\%$ occurs approximately $\Delta X/2R \cong 6 - 8$ behind the face (small additional settlements continue to develop beyond this zone).

Figure 4-6 shows transverse surface settlement profiles at the central section of the model varying with the tunnel heading location. Little surface movement occurs before the tunnel face passes this cross-section ($\Delta Z/2R \cong 0.17\%$ when the tunnel face is directly below the section), but settlements quickly build up behind the tunnel face, reaching a steady state approximately 6 diameters behind the tunnel face. The surface settlement trough extends $Y/2R \cong 3$ from tunnel centerline.

Some far-field movements at $Y/2R = 10$ are observed in Figure 4-6, while no movement is expected in the far-field boundary. From Figure 4-6(b), it is seen 1) the far-field movement increases linearly regardless of the location of the tunnel heading and reaches $\Delta Z/2R \cong 0.28\%$ at the end of the tunnel excavation, and 2) the slope for $Y/2R = 10$ matches the initial slope for the curve with $Y/2R = 0$. Based on numerical experiments, this far-field movement is the result of a near incompressible condition ($\nu = 0.49$), and can be eliminated by allowing volume changes of materials (thus reducing Poisson's ratio, ν). These results suggest excavation in incompressible materials cause disturbance to a larger extent than in compressible materials.

The evolution of surface settlements ΔZ is shown in Figure 4-7 and Figure 4-8 throughout the entire excavation. The contours are drawn from both a two-dimensional top-view and three-dimensional isometric view, to visualize both the transverse and longitudinal settlement distributions. The sequence of these contours is from left to right, from top to bottom, with a 4m difference on the location of tunnel heading. In Figure 4-7, a tiny upward movement of the ground surface in front of the tunnel heading is initially observed.

This upward movement in the order of 1mm seems to be caused by the buoyancy effect, as suggested in Figure 4-9, which shows upward vertical displacements in the range of 0 – 10mm throughout the FE model, confirming this effect. Note that the heave from these elasto-plastic calculations is much smaller than the computed heave in the elastic material (9mm vs. 60mm heave at the tunnel invert). The upward surface movement is no longer observed after the tunnel heading passes the middle section of the finite element model (Figure 4-7).

Figure 4-10 and Figure 4-11 show the vertical movement for elastic and elasto-plastic analyses, specifically as the tunnel face reaches the central section of the FE model. The upward movement contours are quite distinct between elastic and elasto-plastic analyses. There is a region of uplift surrounding where the tunnel is excavated in an elastic medium (Figure 4-10), and downward movements only occur close to the centerline behind the heading; while in elasto-plastic analysis, the settlement above the tunnel outweighs the buoyant uplift and shows overall downward movement on the ground surface.

4.2.4 Tunneling Induced Shearing

Figure 4-12 shows sectional contour plots at intervals of $\Delta X/2R = 1$ of Mises shear stresses¹ while the tunnel face is $\Delta X/2R = 10$ away from the tunnel entry (i.e. heading in the middle of the finite element model). Behind the tunnel face (on the excavated side of the tunnel face), it is seen from Figure 4-12 that all sections that are $\Delta X/2R = 6$ away are similar in all aspects (the size of plastic zone, distribution of shear stress, etc.) The plastic zone around the tunnel cavity is fully developed $\Delta X/2R = 1$ away behind the tunnel face, while additional plasticity develops at then ground up to $\Delta X/2R = 6$ behind the tunnel heading.

In front of the tunnel heading, there is no plastic zone observed at $\Delta X/2R = 1$ away from the tunnel face, and tunneling induced very little shearing at the $\Delta X/2R = 1$ section.

¹ $q = \sqrt{\frac{3}{2} s_{ij} s_{ij}}$, $s_{ij} = \sigma_{ij} - \frac{1}{3} \sigma_{kk} \delta_{ij}$, δ_{ij} : Kronecker delta

Therefore, for this particular case with stable face conditions, the plasticity does not significantly effect predictions of ground movements ahead of the advancing tunnel heading.

Figure 4-13 shows the sectional contours of total shear strain using the FEMAP¹ definition. It is seen that the shear strain at the tunnel face is between 0% – 5%, and the lower half of the tunnel face shows more shear strain than the upper half due to the stress gradient. Behind the tunnel face, shear strain accumulates and reaches a steady state after $\Delta X / 2R \cong 3$, and significant changes from tunnel face to a distance of $\Delta X / 2R \cong 2$, suggesting severe shearing occurs in this region.

4.2.5 Tunnel Wall Convergence

Figure 4-14 compares the tunnel wall convergence at three locations around the tunnel cavity (crown, springline, and invert) from analyses with advancing rate (excavation length per step) of $\Delta X / 2R = 3.33$ and $\Delta X / 2R = 0.67$ at six locations (X at intervals of 20m between 20m – 120m) of tunnel face throughout the finite element model. The tunnel wall moves inward into the excavation from all three locations monitored. This is as expected, with most movement seen from the crown, and similar magnitudes of movement from springline and invert. The convergence curves for each location (crown, springline, and invert) show similar patterns irrespective of the tunnel face locations and converge to a single backbone curve. Furthermore, the computed convergence curve behind the tunnel face ($\Delta X / 2R < -5$) are in good agreement with the 2-D plane-strain solutions.

All of the convergence curves from 3-D analyses show characteristic saw-tooth patterns whose period is controlled by the excavation increment. The magnitude of the saw-tooth profile ranges from 4cm at the tunnel invert to 1cm at the tunnel crown. These oscillations are associated with the finite element discretization. For each simulation, the period of the saw-tooth pattern along the tunnel advance direction corresponds to the mid-point of the second order finite element used. Similar patterns are also observed and reported as “irregular shrinkage” (Augarde et al. 1999), or are reflected on the lining forces as “zig-

¹ Finite Element Modeling And Post-processing, a commercial Windows-based finite element pre/post processing program by Electronic Data System Corp. <http://www.ugs.com/products/nx/simulation/advanced/femap/>

zagging normal forces” (Vermeer 2001). The magnitude of the saw-tooth can be reduced by increasing the meshing density in the tunnel axis direction.

Figure 4-15 shows a detailed view of the convergence curve for the tunnel invert (I) and crown (C), with tunnel heading at $X/2R = 16.7$. Simulation with $\Delta X/2R = 3.33$ excavation steps produces a saw-tooth oscillation with a magnitude of $\Delta U/2R = 4\%$ at the tunnel invert, magnitude of $\Delta U/2R = 2\%$ at tunnel crown, with a period of $X/2R = 3.33$. In contrast, when the excavation step reduces to $\Delta X/2R = 0.67$, a reduction of the oscillation magnitude by 50% is seen at the tunnel crown, but little or no reduction is observed at the tunnel invert.

A more detailed examination of Figure 4-14 shows that there is a very close agreement between 3-D and 2-D (plane-strain) convergence predictions at locations $X/2R < -6$ behind the face (excluding boundary effects close to the tunnel entry). Agreement is particularly good for the invert and springline (S) locations, while the 3-D analyses tend to produce larger settlements at the tunnel crown (Figure 4-14 shows $\Delta U/2R \cong 4.95\%$ at the crown for 2-D analyses, and $\Delta U/2R \cong 5.1\% \pm 0.1\%$ for 3-D analyses).

4.2.6 Summary

The results of the base-case 3-D tunnel analyses can be summarized as follows:

- 3-D finite element analyses of tunneling processes in elasto-plastic soils are computationally demanding. Problems are especially severe if the face stability is low (e.g. large zone of plasticity developing around the tunnel heading).
- The deformation behind the tunnel face reaches a steady state approximately six diameters behind the advancing tunnel face.
- Boundaries have minimal effect on predicted ground deformations only when the tunnel heading is 10 tunnel diameters from the boundary.
- Tunneling causes little shearing beyond one-tunnel diameter in front of the tunnel face, and the plastic zone surrounding the excavation is fully developed at one tunnel-diameter distance after the tunnel face passes.

- Based on the tunnel-wall convergence curve, the ground deforms when the tunnel face is still one tunnel diameter away, and reaches a steady state after tunnel heading has passed at a distance of five tunnel-diameters.
- The current finite element model has inadequate discretization along the tunnel axis direction, generating spurious saw-tooth pattern in the computed tunnel-wall convergence curve. However, due to the lengthy solution time, as well as the memory requirement, further mesh refinement is impractical.

4.3 Efficient Tunneling Solution via Remeshing

4.3.1 Introduction

Adaptive remeshing (h refinement) is a technique used for finite element analyses on large-deformation problems such as metal forming (Cheng and Kikuchi 1986), and geotechnical applications including shallow foundation (Hu and Randolph 1998a; Hu and Randolph 1998b; Hu et al. 1999), tunnel headings (Komiya et al. 1999), and penetration analyses (Susila and Hryciw 2003) to control solution accuracies related to mesh distortion. The remeshing technique has also been applied in tunneling analysis to excavate accurate volume of materials during tunneling process (Abu-Farsakh and Voyiadjis 1999). The adaptive remeshing procedure, as illustrated in Figure 4-16, uses of standard finite element calculations with three additional steps: 1) error estimation, 2) new mesh generation, and 3) interpolation for solution mapping between meshes.

The error estimation step determines the finite element solution error associated with the discretization. The estimated error is then used as a reference to refine the mesh. According to Zienkiewicz et al. (1999), all error estimators are derived from two types of procedures: Residual based (Babuska and Rheinboldt 1978) or Recovery based (Zienkiewicz and Zhu 1987). A summary and relative numerical performance of various error estimators is given by John (2000).

In the mesh refinement step, a new mesh is automatically generated based on both the old mesh and the discretization error determined from the error estimation step. Finally, once the new mesh is generated, the solution from the old mesh is mapped onto the new mesh via

interpolation or extrapolation. Three different interpolation methods were evaluated by Hu and Randolph (1998), namely 1) An Inverse Distance Algorithm (IDA), 2) Arbitrary Linear Interpolation (ALI), and 3) The Unique Element Method (UEM). The UEM performs the solution mapping by finding the element in the old mesh containing the stress point of the new mesh and interpolates using closest three Gauss points within a single element of the old mesh to interpolate the variables on the new Gauss point. They reported that the unique element method (UEM) provided the best solution among these three schemes.

When designing a finite element mesh, it is a conventional practice to assign higher mesh densities to areas with high stress concentration or strain gradients so as to maximize the solution accuracy with minimum computational cost. For tunnel analyses, the most highly stressed/strained region is around the excavated heading (i.e. at the cavity front). The tunnel heading moves inside the finite element mesh as excavation progresses. As a result, high mesh densities are needed throughout the entire finite element mesh in the direction of the tunnel axis if a step-by-step excavation is to be modeled with a single fixed mesh.

For simulations of tunneling processes with progressive excavation, finite element meshes are commonly generated by extruding a 2-D mesh. With this approach, it is easy to visualize the finite element mesh and to control the quality of finite elements. Some special-purpose finite element software has extrusion as its only option for conducting 3-D analyses (e.g. PLAXIS 3D-Tunnel¹). However, a simple extruded mesh for 3-D tunnel analyses is often inefficient since:

1. The mesh density in the extrusion direction is controlled by the excavation step size, while high-density mesh is only needed around the tunnel face.
2. Only a small fraction of the degrees of freedom close to the active tunnel heading is usefully employed at a given stage of the tunnel excavation, and coarse discretization is sufficient for regions distant from the active tunnel face.

Therefore, adaptive remeshing seems well suited for 3-D tunnel excavation.

¹ PLAXIS 3D-Tunnel, a specialized finite element package by PLAXIS BV in geotechnical applications. <http://www.plaxis.nl>

4.3.2 A Simple Remeshing Algorithm for Time-Efficient 3-D Tunnel Analyses

A new adaptive remeshing algorithm specifically designed for efficient 3-D tunnel analyses with an extruded mesh is proposed. The new algorithm improves the numerical accuracy of tunneling analysis without introducing additional unknowns, and therefore retains the time-efficiency of the original mesh.

Figure 4-17 shows an ideal meshing scheme for tunneling analyses. Due to the transition from undisturbed ground to a fully excavated condition, the model should have high mesh density in the vicinity of the advancing tunnel face (e.g. $-5 \leq \Delta X / 2R \leq 1$), and moves with the tunnel face as excavation advances. The new remeshing algorithm has two steps: 1) increase the mesh density in front of them High Mesh Density (HMD) zone by halving one layer of existing mesh; and 2) Recombine these two layers when the excavation front has passed through this zone. This is illustrated in Figure 4-17. Therefore, the total number of degrees of freedom is kept constant throughout the entire analyses, and only three layers (one layer in the front, and two layers at the tail) of finite elements are involved during the remeshing process.

The new algorithm differs from the traditional remeshing algorithms in the following aspects:

1. The zone of remeshing is predefined, thus there is no need to evaluate the discretization error. The purpose of the new remeshing algorithm is to dynamically generate economic meshes rather than control discretization errors for large-deformation analyses.
2. The mesh refinement does not depend on the discretization error, but rather depends on the location of tunnel headings, while traditional adaptive remeshing is guided by discretization error.
3. The computational cost of remeshing is constant, and insignificant compared to traditional adaptive refinement. Since only three layers of elements are modified with identical amount of calculations between different steps, the proposed algorithm does not need error estimation and does not need to search for regions to refine.

4. Traditionally, remeshing is done at regions exhibiting high discretization error and high stress/strain gradients. For this reason, the solution mapping/interpolation process is unreliable due to potentially high variability of variables in a short distance. Figure 4-18 shows load-displacement curves of a footing problem using three different interpolation algorithms (Hu and Randolph 1998b). Fluctuations are seen in all three curves, and the error was attributed to unbalanced forces after the solution mapping. The new algorithm propose remeshes at regions where gradients of stress/strain should be insignificant, so the error introduced by solution mapping should be minimal.

By employing the proposed remeshing technique, improved numerical accuracy is achievable without introducing any additional degrees of freedom or increased solution time. We note that the proposed remeshing procedure is suitable for a single tunnel excavation, or when multiple tunnel headings progressing concurrently at the same rate, whereby the HMD zone can be shared by all tunnel headings¹.

4.3.3 Preliminary Calculations

Figure 4-19 compares the tunnel wall convergences at the invert, springline, and crown from a uniform mesh calculation (dashed lines with open symbols, from Figure 4-14) and a non-uniform mesh calculation with remeshing (solid lines with filled symbols). The model is the base-case tunnel excavation in elasto-plastic soil, as previously examined (Table 4-1).

Figure 4-19(a) shows the convergence curves for the tunnel face advanced to 20m from start, for which remeshing is done only once where finite elements are coarsened at 0m – 4m and are refined at 24m – 28m. Therefore, the differences between solutions are mainly caused by differing mesh densities along the tunnel excavation direction. By inspecting results at 4m – 20m, it is seen that the magnitude of the saw-tooth oscillations are effectively halved by doubling the mesh density. This result reinforces our intuition that these oscillations are controlled by the spatial discretization in the finite element model.

¹ An implementation of the remeshing algorithm used for calculations in Section 4.3.3 is detailed in Appendix B.

Figure 4-19(b) shows the convergence curves for tunnel face at 60m. As seen in the figure, the convergence curves calculated with the remeshing procedure are much smoother than those for the uniform mesh. This is because 1) higher mesh density results in smaller discretization errors, and 2) the remeshing procedure effectively drops the mid-side nodes when coarsening the mesh. It is also seen from the figure that the convergence curves calculated from the uniform mesh exceeds the 2-D solution; while the solution with remeshing shows smaller movement and asymptotically approaches the 2-D solution.

4.3.4 Discussion

The proposed remeshing procedure is specifically designed for 3-D tunneling analyses with finite element methods, and is currently limited to a single tunnel heading application.

The advantage of remeshing away from the tunnel heading is the reduced interpolation error due to low stress/strain gradients¹ than remeshing around the tunnel heading (that will be done with traditional remeshing algorithms targeted to reduce discretization errors.) Therefore, simple interpolation schemes may be adequate. Further enhancements are possible by incorporating more sophisticated interpolation algorithms to enable more accurate solution mapping between meshes.

Until the recent release of version 6, ABAQUS did not have the capability to transfer/map solutions between different meshes comprised of 2nd order elements. For this reason, a simple 1-D interpolation (along the extrusion direction) was used in the prototype implementation of the mesh refinement algorithm. In the future, the new ABAQUS capability for solution mapping (*MAP SOLUTION) can be employed. Better numerical accuracy is expected, as the interpolation scheme used in ABAQUS is more sophisticated than the simple 1-D linear interpolation used in the prototype implementation presented here. The solution-mapping algorithm in ABAQUS version 6 consists of the following three steps:

¹ Low stress/strain gradient meaning less variation of these variables for a given distance than high stress/strain gradients. Therefore, interpolation of variables such as stress/strain is more accurate in regions with low stress/strain gradients than regions with high stress/strain gradient.

1. In the old mesh, all variables on stress points are first extrapolated to the nodes of the same element.
2. For each node in the old mesh, the variables from different elements sharing the node are averaged.
3. Identify the element in the old mesh that contains stress points in the new mesh. Then use standard finite element interpolation to interpolate the variable on the node to each new stress point.

For each new stress point, the solution mapping in ABAQUS and UEM require searching the finite element in the old mesh containing the new stress point (the process is similar to contact detection in particulate system). If no special arrangement is used for this process, the computation time (to search for containing elements,) approximately scales in the order of $N_E^{old} \times N_E^{new}$ ¹, where N_E^{old} is the number of elements in the old mesh, and N_E^{new} is the number of elements in the new mesh. As $N_E^{new} > N_E^{old}$ (usually), the scaling is at least $(N_E^{old})^2$. This process is computationally expensive for large-scale models. In fact, based on numerical experiments with the base-case tunnel model, the solution mapping process takes longer than solving the equilibrium equations, which is usually the bottleneck for large-scale 3-D finite element solutions.

The proposed remeshing scheme is far more efficient than most remeshing algorithms based on discretization errors for the following reasons:

1. The elements to be rearranged are pre-determined by the tunnel heading location, thus evaluation of discretization error is unnecessary.
2. The solution mapping in traditional remeshing algorithm requires a search step/procedure for more accurate interpolations such as UEM or solution mapping in ABAQUS. This is because the refined region is unknown beforehand. The cost

¹ Linear search algorithm, as this is the only applicable search algorithm for general arrangement of finite elements.

of search increases quadratically with the number of elements in the finite element model, and becomes unpractical for 3-D finite element models.

3. The proposed remeshing algorithm scales linearly with the number of elements per layer, can thus be applied to large-scale 3-D finite elements models with insignificant computational cost for performing remeshing.
4. Even if a more advanced interpolation scheme is applied with the proposed remeshing scheme, at most three layers of elements need to be searched to perform an interpolation similar to ABAQUS version 6 instead of the entire finite element model. Thus, significant performance differences are expected between the proposed remeshing scheme and traditional ones.

The remeshing algorithm developed here provides a considerable advantage over traditional codes and enables large 3-D tunnel analysis at a scale and resolution that previously intractable on standard computational platforms.

4.4 Implementation of Schwarz Alternating Scheme with ABAQUS

4.4.1 Introduction

The proposed 1-D remeshing (along the extrusion direction) procedure for tunneling can save computation time for large-scale extruded finite element meshes. However, it is not suffice to tackle completely general tunneling problems (e.g. multiple headings, tunnel intersections, etc.), which requires considerable additional computation time for elasto-plastic analyses. This section considers possible the additional benefit of parallel computation through interconnected computers.

Overlapping domain decomposition (Smith et al. 1996), ODD, subdivides the finite element domain to be solved into several sub-domains, and each sub-domain can be solved independently. The recovery of the solution to the original problem is done through an iterative process by passing solution variables (e.g. displacement) in the overlaps between neighboring sub-domains, and no other information (e.g. stiffness matrix) is needed. Consequently, ODD is considerable convenient to implement with an existing finite element

program (such as ABAQUS) compared to other domain decomposition methods (Section 4.5).

4.4.2 Performing Analyses using the Schwarz Alternating Scheme

Schwarz alternating scheme (described in section 3.8.1) is one of the earliest overlapping domain decomposition method, and is implemented into the program *DD*, a message passing parallel program written in C++ using MPI. Figure 4-20 shows the flowchart for this program. It should be noted that the program *DD* only help exchange information between sub-domains, and all finite element calculations are done by using ABAQUS.

The program *DD* takes a user-defined input file listed in Figure 4-21. The input files contain the following information: 1) number of sub-domains; 2) domain definition file names; 3) action definition file names – the action definition file contains the ABAQUS *include file* for performing analyses; and 4) number of analysis steps and step definitions.

The domain definition files *Dxx.def* listed in Figure 4-21 contain 1) sub-domain identification; 2) the name of a restart file for ABAQUS prepared by the user containing initial equilibrium condition; 3) the number of neighboring sub-domains; and a definition for each neighboring sub-domain; 4) element set name for the overlapping elements; 5) element numbers in the overlapping region; 6) interface nodes between neighboring sub-domain; and 7) initial estimate for the initial boundary condition.

Once the input files are prepared, along with restart files generated from ABAQUS, the analysis may begin. The program contains two major portions (Figure 4-20): the master process and the worker processes. With N participating processors, there will be one master process and $N - 1$ worker processes.

The master process starts by reading the input files and creates a task queue to store the sub-domains that need to be solved. Initially, these sub-domains are those with critical activities, such as material removal or load applications. After initialization of the task queue, the master process starts to listen to requests from worker processes, and distribute sub-domains as well as their associated data that is needed for worker processes such as the initial condition files (**.res*), the interface solutions, etc. Upon completion of sub-domain solves by worker processes, the interface solutions for adjacent sub-domains are extracted on the

worker processes and submitted back to the master process, and these adjacent sub-domains are added to the task queue as their boundary conditions have been changed. Therefore, more and more sub-domains are added to the task queue. When interface solutions are received from worker processes upon the completion of a sub-domain, the master process will check if the convergence is reached. If it is converged, the master process will notify all worker processes and the analysis for the current step will terminate, and then proceed to the next step.

The worker processes start by querying the master process for new work. Once a task is obtained from the master process, along with the boundary conditions, an input file for ABAQUS can be prepared. The input file is a restart analysis for ABAQUS, with actions (if any) and imposed displacement boundary conditions. Once the sub-domain solution is completed, ABAQUS outputs **.fil* data files. The displacements on the artificial interface nodes that are inside the sub-domain solved but are on the boundaries of neighboring sub-domains, are then extracted from the **.fil* file, and passed to the master process. Then worker processes then check with the master processes to check if the convergence has reached. Once the convergence is satisfied, the worker processes submit the domain solution (**.res*) file to the master process, and these restart files become the initial condition for the next step. The worker processes then request new work until the entire analysis is completed.

As consequence of having the master node manage the transfer of nodal displacements between sub-domains, the worker processes do not communicate with one another. Since the computation of each sub-domain is decoupled from the others, the parallel efficiency is expected to be high.

4.4.3 Preliminary Calculations and Convergence Criterion

Figure 4-22 illustrates the mesh decomposition for the base-case tunneling problem discussed in 4.2. For simplicity, the mesh is decomposed along the extrusion direction into 10 sub-domains, and each sub-domain contains four slices of elements (2 overlaps with adjacent sub-domains), except the last sub-domain that has only three layers of elements. Therefore, it is equivalent to a finite element model with a total of 30 layers of elements.

The process of calculation is demonstrated in Figure 4-23. Assuming the tunnel excavation starts at sub-domain #1, the calculation starts from sub-domain #1. When the artificial boundary conditions on a sub-domain are not yet fully defined, a roller-type boundary condition is assumed (i.e. no constraint on deformations in the vertical plane). Once sub-domain #1 is solved, its neighboring sub-domain #2 is then solved with displacement boundary condition applied on the artificial boundaries defined from interiors of the sub-domain #1. Once sub-domain #2 is solved, its neighbors #1 and #3 are calculated, and the calculation continues. It is seen in Figure 4-23 that after step 8 the calculation is alternating between odd-numbered sub-domains and even-numbered sub-domains, and the proposed calculation procedure becomes the well-known red-black ordered¹ calculation. However, the ordering emerges naturally from the model, rather than a pre-determined sequence. It is not effective to apply red-black ordering for such problems initially, because the residual (unbalanced force) is located at the sub-domains where construction is occurring. Therefore, it is more effective to use the proposed calculation procedure starting the calculation with the most residuals, and then propagate the residual through the neighboring sub-domains.

Figure 4-24 shows the convergence characteristics of the proposed calculation procedure for both elastic analysis and elasto-plastic analysis, with an error measure defined by the ratio of maximum displacement corrections to the maximum displacement in the same sub-domain. It is seen that both elastic and elasto-plastic analyses exhibit similar convergence behavior. Therefore, the proposed overlapping domain decomposition procedure is capable of solving the base-case tunneling problem.

In order to terminate the iterative calculation procedure, a convergence criteria must be specified. The variable used for the convergence criterion needs to be easily obtainable from ABAQUS. As a result, dual convergence criteria are used for the subsequent analyses:

$$\frac{\delta u^{int}}{u_{max}^{int}} \leq 1\% \quad (4.1)$$

¹ Red-black ordering is a well described communication pattern for parallel computation. Adjacent sub-domains are colored differently, and sub-domains of the same color are calculated in parallel

$$\frac{\delta p}{p} \leq 0.5\% \quad (4.2)$$

where δu^{int} is the correction of displacements on artificial interfaces, u_{max}^{int} is the maximum displacement increment amongst all artificial interface nodes, $\delta p/p$ is the correction of confining pressures normalized by the confining pressure at the same interface node. As shown in Figure 4-25, confining pressure based error measure is consistently larger than the one based on von Mises stress¹, q .

From Figure 4-24 and Figure 4-25, it is apparent that least 150 sub-domain-solves or alternations between sub-domains are necessary to obtain a convergent solution. In order to accelerate convergence, three-dimensional decomposition is preferred. Decomposition in three-dimensions allows more neighbors attached to each sub-domain and thus, propagation of information such as residual forces is faster than one-dimensional decomposition, and provides better utilization of multiple computers.

In order to evaluate the performance of the overlapping domain decomposition method, the model in Figure 4-1 has been decomposed into eight sub-domains as shown in Figure 4-26, and one-layer of elements are overlapped between sub-domains. An elastic calculation with three steps of excavation and the parameters in Table 4-1 is performed. In this example, Poisson's ratio ν is reduced to 0.3 to avoid poor convergence behavior associated with incompressibility.

Figure 4-27 shows the convergence history for the elastic calculation, and Figure 4-28 compares the iterative solution of ODD with the full finite element solution (direct solution method). It is seen that reasonable agreement is obtained between the two solutions, except for the vertical displacement. The results suggest the solution is not yet converged, and more calculations are needed. However, the solution time with two computers has already exceeded the solution time on a single processor, as shown in Table 4-3. Therefore, the calculation procedure is not efficient.

¹ $q = \sqrt{\frac{3}{2} s_{ij} s_{ij}}$

4.4.4 Discussion

The proposed overlapping domain decomposition method appears to be capable to obtain convergent solutions. The calculation procedure is fully asynchronous and requires small amounts of communication, thus it should be scalable in terms of using increasing numbers of computers and solving large-scale problems. The procedure is known as a one-level domain decomposition method, which lacks a coarser mesh to help propagate information between unconnected sub-domains. As a result, the convergence rate deteriorates as the number of sub-domain increases because each iteration step can only move information from one sub-domain to its neighbors, and it therefore takes several iterations from one end of the entire domain to the other end. There is also a second practical issue for the current implementation¹.

Besides the limitation, it is difficult to decompose a single domain into several overlapping sub-domains automatically. Manual decomposition is manageable only for a very small number of sub-domains. Thus, it is difficult to apply the proposed overlapping scheme toward general large finite element simulations. As a result, other methods for parallel computation are investigated.

4.5 Implementation of the FETI Method Utilizing ABAQUS

4.5.1 Introduction

The FETI algorithm (detailed in Section 3.8.2.2) is a non-overlapping domain decomposition method for solving large-scale finite element models. There are several advantages of the FETI method over other domain decomposition methods:

1. Producing non-overlapping sub-domains for a given mesh is easier than producing overlapping ones.
2. The FETI method formulates a reduced interface problem that is described by sparse matrices. On the other hand, the reduced interface problem formulated using

¹ Each participating processor consumes one additional ABAQUS license, thus it could be costly to run on multiple computers due to the licensing cost.

substructuring or Schur complement approach (a non-overlapping domain decomposition method) results in a dense system that can be too big to be solved efficiently.

3. The FETI method uses interconnecting forces between sub-domains as unknowns. As a result, sub-domains may undergo rigid body motions. By solving the Rigid Body Modes (RBM) of sub-domains, a coarse auxiliary problem (less or equal to 3 RBMs for 2-D sub-domains, and less or equal to 6 RBMs for 3-D sub-domains) is formulated. This auxiliary problem helps to propagate information globally, accelerates the convergence of FETI iterations, and results in good parallel efficiency and high scalability.

Therefore, the FETI method is implemented in the program *FETI_DD*, to solve large-scale 3-D tunnel problems. The *FETI_DD* program uses ABAQUS to compute required matrices and performs FETI iterations to solve the equilibrium conditions with parallel computation technique. This section starts by explaining how to extract the required matrices for the FETI method from ABAQUS, a technique to obtain rigid body modes, preconditioners for the FETI method, and load-balancing for the available processors. A detailed description of the implementation is given in 4.5.6, and the section ends with the performance evaluation.

4.5.2 Interfacing with ABAQUS

Non-overlapping domain decomposition methods require sub-domain stiffness matrices and load vectors in order to carry out the analysis. Programs such as ABAQUSTM do not provide any access to the global stiffness matrix for the finite element model, but can report the element stiffness matrix upon request (using the keyword **ELEMENT MATRIX OUTPUT*.) Therefore, it is possible to form the sub-domain stiffness matrix from the element stiffness matrices extracted from ABAQUS. The load vector can be extracted from its components in ABAQUS: the concentrated load (**CLOAD*), distributed load (**DLOAD*), and nodal forces due to internal stresses. This information can be output from ABAQUS through the keywords **NODE FILE*, **EL FILE*, and **ELEMENT MATRIX OUTPUT*. Figure 4-29 shows the short code for ABAQUS to output needed information for carrying out calculations using FETI algorithm.

The output from ABAQUS is given at the node or element level, and standard stiffness assemblage procedure in the finite element method is required to form the global equation $\mathbf{K}\mathbf{u} = \mathbf{F}$. Thus, a global ordering of nodes needs to be determined. The implemented ordering is based on the attributes of nodes, and interface nodes are ordered last, while the interior nodes are ordered first. So the resultant system can be written as:

$$\begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ib} \\ \mathbf{K}_{bi} & \mathbf{K}_{bb} \end{bmatrix} \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_b \end{bmatrix} = \begin{bmatrix} \mathbf{F}_i \\ \mathbf{F}_b \end{bmatrix} \quad (4.3)$$

The subscripts in (4.3) denote interiors (i) or boundaries (b). Even though such ordering is not required for the FETI algorithm, it is convenient for calculating preconditioners and provides the possibility to apply the Schur complement approach to solve the same problem.

Once the global stiffness matrix and the residual vector are determined, the FETI algorithm can be applied to solve the displacement field. Force residuals are then obtained by imposing the calculated displacement field of each sub-domain into ABAQUS as displacement boundary conditions (using the keyword *BOUNDARY).

4.5.3 Rigid Body Mode Extraction

The rigid body mode extraction of sub-domains is vitally important for the FETI algorithm. Inaccurately calculation of the rigid body modes may result in dramatic reduction in convergence rate, divergence of solution, or convergence to a spurious displacement fields (Farhat and Geradin 1998).

Rigid body modes can be computed either as a by-product of factorization (Farhat and Roux 1994), or through analytical procedures to obtain all rigid body modes and then eliminate constrained rigid body modes through some geometric procedure (Park et al. 1997) or singular value decomposition (Farhat and Geradin 1998).

Since the FETI algorithm requires not only the rigid body modes, but also the generalized inverse matrix \mathbf{K}^+ , it is cost-effective to obtain the rigid body modes through factorization, and rigid body modes are related to the zero pivoting condition during factorization. In

order to factorize efficiently the stiffness matrix \mathbf{K} , the sequential version of SuperLU¹ (Demmel et al. 1999; Li 1996) is used with modifications. The unmodified SuperLU does not calculate rigid body modes, and the factorization subroutine will immediately return an error when zero-pivoting conditions are encountered. In order to utilize SuperLU to capture rigid body modes, the subroutine *dpivotL* (located in *dpivotL.c*) is modified. First, the detection of zero pivots (the zero-pivoting condition) is changed from exactly zero to a small tolerance of 1×10^{-8} . This value is based on several numerical experiments which indicated that the value gives a reasonable criterion to capture the correct number of rigid body modes. When the pivot value is less than the tolerance, the zero-pivoting condition is triggered, and the entries below the zero pivots will be changed to zero. The relaxation of the zero-pivot condition is necessary to account for inaccuracies associated with the round-off error².

Once the sub-domain stiffness matrix \mathbf{K} is factorized by the modified SuperLU to $\mathbf{L}_{\mathbf{K}^+}$ and $\mathbf{U}_{\mathbf{K}^+}$ (lower and upper triangles), the procedure (Farhat and Roux 1991) expressed in Matlab³ syntax in Figure 4-30 is applied to obtain rigid body vectors and to prepare the resultant triangular matrices $\mathbf{L}_{\mathbf{K}^+}$ and $\mathbf{U}_{\mathbf{K}^+}$ to calculate matrix-vector product between the generalized inverse \mathbf{K}^+ and any given vector.

4.5.4 Implemented Preconditioners

There are many preconditioners available for the FETI method: lumped preconditioner and Dirichlet preconditioner (Farhat and Roux 1994), lumped and Dirichlet preconditioner enhanced with mechanical consistency (Rixen and Farhat 1999), incomplete Cholesky and successive over relaxation (SSOR) and their variants (Charnpis and Papadrakakis 2002).

¹ SuperLU is an open-sourced numerical package to factorize a general asymmetric sparse matrix and to solve matrix equations with explicit exploitation of memory hierarchies. BLAS-2 and BLAS-3 subroutines are used to deliver high-performance factorization.

² Details on the sparse matrix storage that was adopted in the programming is detailed in Appendix D.

³ Product of the MathWorks, Inc. <http://www.mathworks.com>

The current *FETI-DD* implements the aforementioned four preconditioners, and the mechanical consistency condition is met by using the multiplicity of interface nodes (the number of sub-domains attached to a node) rather than using the actual values of stiffness on diagonals of the sub-domain stiffness matrix. It has been shown that using interface nodal multiplicity is more efficient to compute than using stiffness of sub-domains (Rixen and Farhat 1999).

4.5.5 Load Balancing

Load balancing refers to the method of task distribution among the available processors. A perfect load balancing method ensures the computation time for each processor is the same, thus the overall computation time is minimized.

Figure 4-31 shows the overall flowchart of the *FETI-DD* program that has been implemented by interfacing with *ABAQUS*. The implementation of *FETI-DD* allows independence between the number of sub-domains and the number of processors. This provides greater flexibility in domain decomposition, as well as load balancing.

Load balancing is performed when distributing the sub-domains to worker. When worker processors ask for sub-domains from the master process, they receive information about sub-domains, and immediately perform factorization on the sub-domain to calculate the generalized inverse, \mathbf{K}^+ , and the rigid body vectors, \mathbf{R} . Once the factorization is complete and the rigid body vectors are extracted from the sub-domain stiffness matrix, it will request another sub-domain from the master process. The process continues until all sub-domains have been assigned to worker processes. This load-balancing scheme uses the apparent factorization efficiency processors to determine the workload distribution, thus the heterogeneity amongst processors due to different speed or different workload is implicitly considered, and the optimal computation time should be achieved.

The load balance method used is based on the following assumptions:

1. The workload or the workload distribution on all worker processors remains constant during the course of computation. The assumption can easily be invalidated in a multi-user environment, as other computation tasks can be started by other users after the task assignment stage. However, it is difficult to do load

balancing during computation, as sub-domains would need to be migrated between worker processes.

2. The fast worker processors are equipped with more memory than slow ones. This is because the fast processors will factorize more sub-domains than slower processors, and more memory is required to store these factorized matrices.
3. The factorization of stiffness matrix is the most time consuming process involved in the sub-domain calculations, thus it can be used as a performance metric for load balancing.
4. The distribution of workload assigned by *FETI_DD* remains constant throughout the entire course of analysis. This may be true if the analysis does not involve changes of geometries, but untrue for applications where materials are taken in and out of the system such as tunneling and excavation tasks which change the number of unknowns in the sub-domains.

4.5.6 Implementation of FETI Solver

There are two possibilities for implementing a flexible FETI solver where the number of sub-domains is independent of the number of the processors: 1) create a process for each sub-domain, so each processor may host many sub-domains via multiple processes; or 2) create an intermediate manager to manage multiple sub-domains on each available processor. The multi-process approach is inefficient because if there is no coordination between processes, multiple calculations on different processes may be carried out at the same time, and the processor needs to work on different processes at the same time. This can generate large overhead for context switching¹. Furthermore, multiple processes on a single processor may also compete for networking resources, such that the same message from the master process may need to be transferred multiple times to serve different processes on a single processor. Thus, the intermediate manager approach is adopted. A similar conclusion was drawn by Charmpis and Papadrakakis (2002).

¹ Context switching refers to processors switching between different processes running in the system, and usually involves the save and restore values in registers (memory resides in the processor).

As illustrated in Figure 4-31, once all sub-domains are distributed to the available worker processes, the master process waits until the worker processes report residual forces in the sub-domains. Once all worker processes have reported the residual forces, the master process also needs to evaluate residuals on artificial interfaces between sub-domains. If global equilibrium is achieved (i.e. below the specified tolerance), the analysis proceeds to the next step or the next increment for non-linear analyses; otherwise, the program starts the FETI calculation.

Before the FETI iteration process starts, each worker process carries out the calculations summarized in Figure 4-32. Concurrently, the master process proceeds to the steps documented in Figure 4-33 showing the flowchart for the one-level FETI method including the steps, the mathematical equations associated with each step, the realization of each step, and the communication costs associated with each step on the master process.

At step two (the preparation step) of the master process in Figure 4-33, the matrix $\mathbf{G}_I^T \mathbf{G}_I$ (Equation 3.41) is computed in parallel using all worker processes (the B step in Figure 4-32). Each worker process computes a fraction of $\mathbf{G}_I^T \mathbf{G}_I$, which is then collected and factorized on the manager process. It is possible to distribute the computed $\mathbf{G}_I^T \mathbf{G}_I$ onto several worker processes to perform a parallel factorization. However, $\mathbf{G}_I^T \mathbf{G}_I$ is a symmetric matrix with dimensions equal to the total number of rigid body modes in the entire system, thus $\mathbf{G}_I^T \mathbf{G}_I$ is small enough to be factorized efficiently on a single processor for the current targeted problem size, N . No further communication is needed for calculating the inverse, $(\mathbf{G}_I^T \mathbf{G}_I)^{-1}$.

There are four different kinds of communications used in the FETI solver: *broadcast*, *reduction*, *scatter*, and *gather*.

1. *Broadcast* sends a vector to all processes, and all processes receive identical vectors.
2. *Reduction* collects vectors of the same size from all processes, and applies a reduction operation such as summation or finding a maximum to reduce all vectors into a single vector of the same size on the root process.

3. *Scatter* distributes a vector across the processes, and each process receives only a portion of the scattered vector;
4. *Gather* composes a vector on the root process, and each process contributes a fraction of the composed vector.

From Figure 4-33, it is seen that there are two pairs of scatter/gather operations within the main iteration loop of FETI iteration, additional two pairs of scatter/gather for null-space projections, and one pair of broadcast/reduce for storing searched directions \mathbf{p}_i and for back-orthogonalization. Each operation incurs a data flow of a vector of size N_{compat} (the number of compatibility equations between sub-domains), or size N_{rigid} (the number of rigid body modes of all sub-domains in the entire system). For each FETI iteration step with 64-bit double precision storage, the implementation uses $32(N_{rigid} + 4N_{compat})$ bytes for scatter and gathering, $16N_{compat}$ bytes for broadcast and reduction, and four bytes (integer) used to notify all processes of convergence. Therefore, it is possible to estimate the maximum number of FETI iterations that can be achieved for a given communication link after the decomposition is determined (i.e. N_{rigid} and N_{compat} are known). For example, assuming the underlying communication link has a theoretical bandwidth of T_b Mbps¹, thus the maximum numbers of FETI iterations that can be performed on the communication link based on the implementation is:

$$\frac{T_b}{8} \frac{1}{(32N_{rigid} + 144N_{compat} + 4)} \quad (4.4)$$

Equation (4.4) provides an upper bound to estimate the number of FETI iterations that can be performed for a particular communication link. The actual performance (number of FETI iterations per second) will be less than the estimate in Equation (4.4) due to network latencies (the overhead to start up a communication).

The amount of communication for each FETI iteration step is dictated by how and where the calculations are done. For example, it is possible to store \mathbf{G}_I on the master process

¹ 10^6 bits per seconds

(Charmpis and Papadrakakis 2002), and calculations involving \mathbf{G}_I such as null-space projection can be done locally on the master process and hence, save communication on the null-space projection step. It is also possible to store search directions \mathbf{p}_i on the master process and perform back-orthogonalization on the master process alone.

The calculation of \mathbf{G}_I times a vector \mathbf{x} in the current implementation is done by utilizing the definition of $\mathbf{G}_I = \mathbf{B}^T \mathbf{R}$. First the vector \mathbf{x} is transferred to worker processes by a *scatter* operation based on the information in \mathbf{B}^T . The product $\mathbf{R}\mathbf{x}$ is then evaluated on all worker processes, *gathered* onto the master process, and the master process performs the calculation $\mathbf{B}^T \mathbf{R}\mathbf{x}$. The evaluation of $\mathbf{B}^T \mathbf{R}\mathbf{x}$ is done by recognizing that \mathbf{B}^T is an operator that maps contributions of each sub-domain onto the global compatibility equations, and each row of \mathbf{B}^T has exactly two entries: +1 and -1. Consequently, the *gathering* from worker processes to the master process can be done in six steps: 1) Reordering the global compatibility equation so that the positive contribution of each worker process is continuous, 2) Each worker process prepares the positive contribution (+1), 3) Gather the positive contributions into a vector \mathbf{a} on the master process, 4) Each worker process prepares the negative contribution. 5) *Gather* the negative contributions into a vector \mathbf{b} on the master process, 6) Finally, $\mathbf{y} = \mathbf{a} - \mathbf{m}\mathbf{b} = \mathbf{G}_I \mathbf{x}$, and \mathbf{m} is a row reordering operator that maps \mathbf{b} from the continuous negative contribution space to the positive contribution space.

If the calculation of \mathbf{G}_I times a vector \mathbf{x} is to be done on worker processes, then each calculation will require a *broadcast* rather than a *scatter* to distribute the vector \mathbf{x} , and a *reduction* operation (instead of the *gathering* operation) is needed to report the product back to the master process. *Scatter* is a faster operation than *broadcast* because less data needs to be transmitted on the communication fabric, and the difference can be dramatic with short length vectors. Figure 4-34 shows the speed difference between *scatter* and *broadcast* on both fast Ethernet and on Myrinet connections. It is seen that the scatter operation is at least 10 times faster for a 64-bit double precision vector with a length between 5000 and 20000, and the differences drop to 4 times speed difference with Fast Ethernet, and 1.5 times for

Myrinet. The drop observed in Figure 4-34 corresponds to the switching of algorithms for collective communication for short and long messages in the MPI implementation¹.

Even greater effects can be seen on the reduction/gather operation. The current implementation permits the calculation of $\mathbf{G}_I \mathbf{v}$ be assembled on the master with two gather operations and one single vector addition by calculating \mathbf{B} on the master node. If \mathbf{B} is considered on the worker nodes, a reduction operation must be performed to get results back onto the master node. The reduction operation will cause a vector of identical size being transferred on the communication fabric several times and perform vector additions in order to get the vector summed onto the master process. Figure 4-35 shows the dramatic speed differences between gather and reduction operations.

To conclude, it is possible to implement the same one-level FETI algorithm on parallel computers with many variants by carrying out calculations by different ways. There is no particular arrangement that is the most efficient, as it depends on the problem size to be solved, the communication fabric, the process speed, the number of processors, and the amount of available memory, etc. The current implementation conducts most calculations in parallel, except the factorization of $\mathbf{G}_I^T \mathbf{G}$ and the product between $(\mathbf{G}_I^T \mathbf{G}_I)^{-1}$ and a given vector. As a result, the performance for solving large problems should be reasonable on parallel computers with high-speed interconnections. Performance of the implemented FETI solver will be evaluated later in this section.

4.5.7 Preliminary Performance Evaluation

To evaluate the applicability and performance of the FETI method using the preliminary implementation, *FETI_DD*, a simple plane-strain test problem is set up to simulate a single step of excavation. The preliminary implementation shown in Figure 4-31 allows a single step analysis with nonlinearity, and full Newton-Raphson iteration is used to solve non-linear problems.

¹ More details on the collective communication algorithms for short and long messages in MPICH, one implementation of MPI, can be found at <http://www-unix.mcs.anl.gov/~thakur/papers/mpi-coll.pdf>

Figure 4-36 shows the plane strain finite element model decomposed into eight sub-domains, and each sub-domain has 25 by 25 square 8-noded elements, thus with aspect ratio of one. A single excavation is performed in an elastic medium, with considerations of non-linear geometry effect (NLGEOM directive in ABAQUS.) Thus, three iterations are needed to achieve a convergent solution. Figure 4-37 shows the observed parallel efficiency on four main portions of FETI-DD using 1 – 4 computers ranging between Pentium III 866MHz and Pentium III 1GHz equipped with 256MB to 512MB of memory, and loosely interconnected through 100base-T non-dedicated Ethernet connections (MITnet) physically spanning across three floors of building one at MIT. This incurs the additional cost of communication through multiple network switches. The setup is particularly challenging because different CPU clock speeds, available physical memory, and high variability of interconnection conditions.

Reasonable parallel efficiency is suggested in Figure 4-37 for this setup. The four components: initial setup (calling ABAQUS, extract stiffness matrix, rigid body mode extraction,) FETI iteration, displacement solution, and residual evaluation (imposing displacement field into ABAQUS input file, invoking ABAQUS, extract forces) are measured from the master process, thus the time for each component is determined by the slowest computation time amongst all worker processes. From Figure 4-37, it is seen the residual evaluation and initial setup achieved high parallel efficiency, as there are no communications during residual evaluation, and only few communications when doing initial setup. Furthermore, these two components take more than 73% of the total computation time. As a result, high parallel efficiency is expected. On the other hand, the displacement solution and FETI iteration do not show good parallel efficiency. The total computation time for displacement solution is negligible, and the FETI iteration takes 27% of the computation time.

The parallel efficiency for FETI iteration is observed to be 90% on a two-processor setup, and then the efficiency drops rapidly to 60% with three processors, and further drops to 50% with four processors. There are two explanations for the observed efficiency degradation: 1) at two-processor setup, the clock speeds for CPUs are 1.0GHz and 933MHz, and next added processors have clock speed of 866MHz. As a result, load imbalance is possible, as all sub-domains are equally sized; 2) the intermediate network fabric has higher

communication latency with more than two processors. As the first two processors are physical located in the same room, they may be connected to the same network switch. Other processors are located two floors away and in a different wing of the building, thus it is very likely that extra hops (i.e. networking through more switches or routers) are needed to reach these processors, generating higher communication latency.

Further evaluation has been done with a 20-sub-domain setup with an identical global finite element mesh shown in Figure 4-36, and the decomposition yields sub-domains with 25 elements by 10 elements (i.e. aspect ratio, $W/H = 2.5$). Figure 4-38 summarizes the parallel efficiency and computation time for the 20-sub-domain simulation. It is seen that the computation time for residual evaluation (computation of unbalanced forces) and initial setup drops to 57%, and the computation time for FETI iteration increased to 43%. A smoother degradation of parallel efficiency is observed for the FETI iteration, but is less efficient than the 8-sub-domain simulation. It can be observed that the absolute computation time for residual evaluation and initial setup is similar between 8-sub-domain simulation and 20-sub-domain simulation with the same number of processors, but the computation time for the FETI iteration doubled.

The reasons for the increased computation time for FETI iteration are: 1) larger number of interface unknowns need to be solved, and more unknowns on the interface generate more communications. 2) The 8-sub-domain decomposition yields the perfect domain aspect ratio ($W/H = 1.0$), and the total number of FETI iterations is 206, while the 20-sub-domain decomposition requires 479 iterations. 3) The economic lumped preconditioner was used, and therefore the iteration count will be more sensitive to the number and the aspect ratio of sub-domains. As a result, it takes longer to complete the 20-sub-domain simulation than the 8-sub-domain decomposition, even though calculations of the interiors should be the same when the times are aggregated.

One major drawback for the proposed FETI-DD implementation is interfacing with ABAQUS, as can be seen in Figure 4-37 and Figure 4-38 that major computation time is spent on invoking ABAQUS to evaluate residuals and initial setup. Even though it is advantageous to interface with ABAQUS (to utilize built-in constitutive laws and element types, etc.), computation cycles are wasted in ABAQUS because this program always solves

the resultant finite element equations with direct methods. Apart from factorization, interfacing with ABAQUS also requires the generation of input files (for ABAQUS), and must parse the output files from ABAQUS, incurring additional overhead.

However, the preliminary results do show potentially scalable performance by incorporating interconnected computers, even for such a small-scale problem (5000 8-noded elements, 30602 unknowns) that be solved on 1GHz PIII in 41 seconds. Therefore, FETI-DD delivers the potential to solve large problems using loosely interconnected PC computers.

4.6 Full Finite Element Program Implementation with FETI Method

4.6.1 Introduction

The previous section shows that FETI-DD (one-level FETI method) can be used effectively for geotechnical engineering analyses, but faces several limitations: 1) multiple ABAQUS licenses are required for concurrent calculations; 2) ABAQUS was restricted to Windows operating systems on PC computers (at the time FETI-DD was programmed); 3) the overhead (including I/O time and unnecessary factorization as previously discussed) invoking ABAQUS overshadows the potential performance gain with multiple interconnected computers.

To remove these limitations, FETI-DD was further developed into a self-contained parallel finite element program, FETI_FEM. The program was developed on the 16-node Beowulf parallel cluster known as **codex-hammer** running the GNU/Linux operating system. The machine is housed in the Department of Civil and Environmental Engineering at MIT. This development eliminates the dependency of ABAQUS, and takes advantage of the dedicated cluster, **codex-hammer** for large-scale computation¹.

4.6.2 Input Files

FETI_FEM program performs calculations based on human-readable text input files. These input files define the finite element problem and controls several run-time parameters. The

¹ The experience and recommendation of the author on using Beowulf clusters is documented in Appendix C.

input files can be categorized as follows: 1) problem definition, 2) sub-domain definition, and 3) finite element definition. Examples of problem definition and sub-domain definition are given in Figure 4-39.

1. The problem definition file, *PROBLEM.DEF*¹, defines various parameters for controlling the finite element solution tolerances for non-linear solutions, amount of information to be recorded, tuning parameters for FETI iterations and SuperLU, etc. Besides various parameter settings, the file also defines the total number of sub-domains, and the files defining sub-domains.
2. The sub-domain definition files specify: 1) the sub-domain number, 2) the filename for the finite element model definition of the sub-domain, 3) the boundary condition definition filename, 4) the file that lists all interfacial node numbers, 5) the node set name containing a list of all nodes, and 6) the element set name that contains all elements. The format is shown in Figure 4-39.
3. The finite element definition files, as listed in Figure 4-40, define nodes, elements, material models, node sets, element sets, and actions (e.g. concentrated nodal load or material removal) to perform on the finite element model. The syntax of the finite element definition file replicates the syntax of ABAQUS input files, and a list of supported keywords is listed in Table 4-4. There are several advantages for replicating the syntax:
 - 1) The results of calculations can be easily compared with ABAQUS, in order to validate the implementation.
 - 2) Pre-processing programs that generate input files for ABAQUS such as FEMAP and ABAQUS/CAE can be used to prepare meshes for *FETI_FEM*.
 - 3) The input files can be easily checked and visualized with ABAQUS.

¹ Currently the filename is hard-coded in the program.

- 4) The implementation provides a smooth transition from *FETI_DD* to *FETI_FEM*.
- 5) Large-scale ABAQUS calculations can be easily adapted to *FETI_FEM* for parallel calculation.

4.6.3 Finite Elements and Material Models

Four element types are currently implemented in the *FETI_FEM*: a 4-noded plane strain elements with full integration (CPE4¹), 4-noded plane strain elements with reduced integration (CPE4R), 8-noded plane strain elements (CPE8), and 20-noded brick elements (C3D20)².

As shown in Figure 4-41, an abstract class element is defined, along with common attributes and methods to all finite elements. All finite element classes (C3D20, CPE8, CPE4, etc.) are the subclass of the abstract class element. The abstract class creates a framework to program new elements types, and no modification to the main program is needed when new elements are defined. The correct function based on the finite element type will be called via the dynamic binding mechanism of C++, thus the program can be easily extended to accommodate more finite element types.

Figure 4-42 shows a simplified class diagram for the material models implemented. They can be separated into elastic and elasto-perfect-plastic models. The two elastic models are implemented: linear elasticity (*ELASTIC) and porous-elasticity (*POROUS ELASTIC), and three different yield functions have been programmed: von Mises (*PLASTIC), Drucker Prager (*DRUCKER PRAGER), and Matsuoka-Nakai (*MOHR COULOMB). Therefore, six possible elasto-plastic model combinations are possible.

¹ The CPE4 finite element uses selectively reduced-integration on the volumetric strain, thus the volumetric strain on all Gauss points is replaced with the average volumetric strain. Selectively reduced-integration is used to improve volumetric behavior on nearly incompressible conditions in low-order elements.

² Details on the formulation of the finite elements implemented can be found in Appendix E and F.

The integration of constitutive laws are all done with implicit backward Euler scheme and follow the general framework of Jeremic and Sture (1997). The advantages of backward Euler integration are: 1) stability, 2) larger step sizes are allowed, 3) the consistency condition at yield is automatically satisfied, 4) finding intersection with a yield surface is not required, and 5) quadratic convergence of Newton iteration at global level is recovered with the use of consistent linearization.

4.6.4 Validations

In order to ensure a correct implementation of *FETI_FEM*, elastic calculations on the test problems shown in Figure 4-43 were conducted. The calculation results are compared with *FETI_DD* and ABAQUS, and all results are matched perfectly based on the following criterion:

1. All nodal displacements are matched.
2. For models with less than 10 elements, each element stiffness matrix, nodal forces due to internal stresses (NFORC), and body force load (DLOAD) are matched between ABAQUS and *FETI_FEM*.

The elastic calculations performed validates: 1) the implementation of the FETI solver; 2) element-level calculations such as DLOAD, NFORC, and the element stiffness matrix; 3) assemblage of the global finite element equations; and 4) geometry changes.

In standard finite element procedures, geometry changes remove the contribution of the deactivated elements. However, element deactivation is more complicated in the FETI method, as the removal of finite elements may change the interconnectivity (\mathbf{B}) between sub-domains. Therefore, it is important to validate the test case 7 in Figure 4-43. The test case 7 is composed by three cubical sub-domains, and the excavation takes place right beside the artificial interface between two sub-domains.

Elasto-plastic calculations are also validated through the simple cases shown in Figure 4-43. An additional validation criterion for non-linear analysis is checked against comparable models in ABAQUS, namely the non-linear convergence history. The added criterion

ensures the correct implementation of non-linear solution schemes, as well as the constitutive models.

Figure 4-44 shows convergence history for elasto-plastic analysis using von Mises yield criterion. It is seen that the convergence history matches perfectly with the solution obtained from ABAQUS, except the solution using inconsistent linearization with forward Euler stepping integration scheme, which exhibits linear convergence. Similar validation is also done for porous elasticity, and the convergence history is shown in Figure 4-45. A perfect match between ABAQUS and *FETI_FEM* is achieved.

4.6.5 Extrusion Program

In order to facilitate the preparation of three-dimensional finite element models to be analyzed with *FETI_FEM*, the extrusion program *EXTRUDE* was created. The program shares the finite element class created for *FETI_FEM*, and extrudes plane strain meshes into three-dimensional meshes.

EXTRUDE looks for the file *EXTRUDE.INI*, as in Figure 4-46, in the current working directory. The file provides *EXTRUDE* with the location of the decomposed 2-D meshes (*D01.INP* to *D12.INP* in Figure 4-46), how many extruded layers of elements are there in one sub-domain to be created, the total length of extrusion, total number of layers in the extruded direction, and the global boundary conditions. It is worth noting that the number of layers extruded for each 2-D sub-domain can be different, this provides a better opportunity to create “well-shaped” three-dimensional sub-domains. Figure 4-47 shows the extruded mesh based on the input file given in Figure 4-46, and all sub-domains are extruded uniformly. As can be seen in Figure 4-47, sub-domains on the ends of y-axis have aspect ratio that deviate from the optimal due to extrusion with only three layers per sub-domain. By using non-uniform extrusion with the input file in Figure 4-48, better aspect ratios are obtained for all sub-domains as shown in Figure 4-49.

Based on the information provided in *EXTRUDE.INI*, *EXTRUDE* will generate sub-domains, and other information for *FETI_FEM* including interfacial nodes for each sub-domain, boundary conditions for each sub-domain, problem definition file *PROBLEM.DEF*, and two template files *COMMON.INC* and *MATERIAL.INC* that are included and shared

by all sub-domain finite element input files. The *EXTRUDE* routine also extrudes node sets and element sets in addition to nodes and elements. In addition to the files generated for *FETI_FEM*, the current implementation generates an input file that contains the complete finite element model that can be used by ABAQUS to check if the model is generated correctly and run an analysis to validate the output from *FETI_FEM*. Figure 4-50 summarizes the steps from a two-dimensional single mesh to three-dimensional finite element calculation using *FETI_FEM*.¹

4.7 Performance Evaluation of *FETI_FEM*

4.7.1 Computing Hardware

Figure 4-51 shows the hardware setup of **codex-hammer.mit.edu**, a cluster computer in the MIT Department of Civil and Environment Engineering. Each computation node is equipped with 1GB of physical memory and 80GB of hard drive, and computation nodes are connected by both fast Ethernet and Myrinet². The fast Ethernet connection provides connections to the compile server **cs01.cdx.net** and the front-end gateway computer, **codex-hammer.mit.edu**. The fast Ethernet is also responsible for network file system (NFS) and remote shell connections. Thus, the Myrinet is entirely dedicated for parallel computation.

The Myrinet connection is a high-speed, low latency network interconnection consisting of switches, fiber optical cables, and PCI network cards. The fiber optics used for Myrinet interconnection is capable of full-duplex operation at 2Gbps. The Myrinet network card features on-board processors responsible for tasks such as error-checking and direct memory access (DMA) to host computers. The Myrinet network card is connected to the host computer through its PCI bus, and it is often the bottleneck for high-bandwidth devices such as Myrinet on commodity computers.

¹ A post-processing program FEMPost is written by the author to post-process the output from *FETI_FEM* and is detailed in Appendix G.

² Registered trademarks of Myricom, Inc.

The computation nodes of **codex-hammer** use commodity motherboards with a 32-bit 33MHz PCI bus that has theoretical peak bandwidth of 132MB/sec, and the actual performance depends largely on the implementation of PCI bus of the motherboard. Based on the diagnostics program (`gm_debug`) for Myrinet, the actual performance of the PCI bus through DMA is 84MB/sec for reading content of memory to the PCI bus, and 129MB/sec for writing. The performance is achieved only after tuning the registers of PCI bus controller. The PCI adjustments are now included in the official release of the Myrinet software. As a result, the maximum possible network transfer rate using Myrinet in this setup is 84MB/sec for sending data, and 129MB/sec for receiving data.

Bandwidth and latency determine the efficiency of parallel computers, as bandwidth measures the amount of data can be moved for a given time, and latency determines the minimum cost for each communication operation. NetPIPE¹ is a tool for evaluating networking performance between two nodes by bouncing messages of various sizes back and forth to evaluate the bandwidth and latency of the interconnection.

Figure 4-52 shows the performance of connectivity between two of the computer nodes: fen01 and fen02. These nodes are representative of all computation nodes in **codex-hammer**. The measured inter-processor data transfer bandwidth are 89.74Mbps and 626.88Mbps for fast Ethernet and Myrinet, respectively. TCP is used for Fast Ethernet and GM, the native APIs for Myrinet, is used in these tests for respective connections. The performance of MPI, the de facto API for message passing programming, is also evaluated using two implementations: MPICH-1.2.5², and MPICH-GM³. Slight overhead and lower bandwidth is seen by using MPICH-1.2.5 and MPICH-GM that are implementations of MPI on top of TCP and on GM APIs. The measured latencies, the time to send a small data packet (64 bytes or less) is conventionally referred to as the latency of a networking fabric,

¹ NetPIPE: Network Protocol Independent Performance Evaluator, <http://www.scl.ameslab.gov/netpipe/>

² MPICH: a implementation of MPI available at <http://www-unix.mcs.anl.gov/mpi/mpich/>

³ MPICH-GM: a customized port of MPICH by Myricom, Inc. tailored for the Myrinet networking hardware using their GM API.

are 50 microseconds for fast Ethernet with TCP protocol, 74 microseconds for fast Ethernet with MPICH-1.2.5, 8.7 microseconds for Myrinet with GM, and 8.9 microseconds for Myrinet with MPICH-GM. The low latency of Myrinet can be attributed to an OS bypassing mechanism in the GM library which directly controls data flow to the networking hardware. Fast Ethernet, on the other hand, is driven by the operating system and information to be transmitted must pass through at least four layers of abstractions in the networking stack.

The CPUs on the computation nodes are AMD AthlonXP 1600+ running at 1.4GHz. These CPUs are capable of carrying out two floating-point calculations per clock-cycle, providing a theoretical peak of 2.8Gflops. The processor is hosted on the front-side bus running at 266MHz (133MHz double pumped), with a maximum bandwidth of 2.1GB/sec to the north bridge and to the memory subsystem. Figure 4-53 illustrates the two interconnected systems with the hardware components directly involved in parallel calculations. The bandwidth for each link between components is also shown to illustrate locations that bound the overall throughput of computation and communication.

4.7.2 Performance Evaluation on a Structured Mesh

There are two important characteristics for a parallel program solving large-scale problems: 1) the parallel efficiency, which measures how well multiple processors are utilized; and 2) the scaling of the solution time with the problem size. In order to evaluate these two characteristics for *FETI_FEM*, a perfectly structured mesh shown in Figure 4-54 was created, with cubical finite elements and ideal decomposition of sub-domains with aspect ratio of one. The program *FETI_FEM* is then applied to compute a single step of excavation analysis as a baseline timing measurement. The structured mesh allows for easy scaling of the problem size by changing the depth of extrusion and maintains perfect aspect ratio of sub-domains simultaneously.

Figure 4-55 compares the observed scaling of the solution time versus the problem size of both *FETI_FEM* and ABAQUS 5.8-18. It is seen from Figure 4-55 that the scaling curve of the solution time versus the problem size can be well fitted by the power law:

$$T = \alpha N_E^p \tag{4.5}$$

where T is the total solution time, N_E is the number of elements (second-order brick elements) in the model, α and p are characteristic coefficients of the solvers. The parameter N_E is used as a measure of problem size in (4.5) because it is independent of the mesh decomposition. The coefficient p is an important characteristic that dictates the sensitivity of solution time to the problem size. It is seen from Figure 4-55 that for this particular problem, $p = 1.5136$ for ABAQUS, while $p = 1.217$ for *FETI_FEM*. Therefore, *FETI_FEM* is superior to ABAQUS in terms of the sensitivity of solution time to problem sizes, and is more suitable for solving large-scale problems.

Figure 4-56 shows the observed speedup and parallel efficiency of *FETI_FEM* with Myrinet solving the structured mesh with a single step of excavation. The reference configuration uses four processors because the *FETI_FEM* program cannot fit all needed information into physical memory without less than four processors. Figure 4-56 shows the *FETI_FEM* exhibits super-linear speedup initially with parallel efficiency greater than one, and the efficiency gradually drops to unity as the number processors increases. The super-linear speedup is most probably be caused by load imbalance at the reference configuration. Overall, these results show that *FETI_FEM* is capable of utilizing multiple processors efficiently.

4.7.3 Performance Evaluation for Circular Tunnel Analyses

The structured mesh in Figure 4-54 is well suited for the FETI method as it has ideal aspect ratio for all sub-domains, but finite element meshes for real engineering applications are rarely structured. Therefore, the elastic analysis of the base-case tunnel problem discussed in Section 4.2 has been re-analyzed with *FETI_FEM* using the mesh decomposition shown in Figure 4-57.

The analysis is performed with a reduced Poisson's ratio of 0.3 because the implemented FETI method is not suitable for incompressible and nearly incompressible materials, and the convergence rate is unacceptable for these materials. This limitation is seen on most iterative solvers, and special treatments such as fixed formulation (Bathe 1996) of displacements and pressures or special preconditioners are needed. An extension to the

FETI method by Vereecke et al. (2003) has also been proposed to overcome the incompressibility condition¹.

Figure 4-58 shows the parallel efficiency and speedup of overall computation time and three distinct stages in the calculation with Fast Ethernet:

1. Initial setup: in this stage, the master process distributes sub-domains to all worker processes. The worker processes factorize the sub-domain stiffness matrix and extract the rigid body modes from the sub-domain stiffness matrix. In this stage, no communication is needed between worker processes, and sub-domains are presented by input files with ABAQUS syntax, requiring very little bandwidth. As a result, super-linear parallel efficiency is observed in Figure 4-58 for this stage. However, the super-linear speedup is an artifact of the master-worker programming model. With two processors, only one processor is doing computation while the other processor is idle most of the time waiting for the worker process to fetch a computation job. When one additional processor is added, the computation ideally will be cut in half as two processors are doing computation, while one processor is still idle most of the time. Thus, the apparent parallel efficiency in this case is 1.33
2. FETI iteration: this stage performs the FETI iteration to solve the formulated interface problem, and requires intensive collective communications between the master process and all worker processes as demonstrated in Figure 4-33. As a result, the parallel efficiency degrades as the number of processors increases.
3. Residual evaluation: residual evaluation stage imposes the calculated displacement field from the FETI method to sub-domains to evaluate the equilibrium condition. No communication between workers is needed, and residual forces are summed on the master process. Similar speedup and parallel efficiency as the initial setup stage are seen due to the artifacts of master-worker programming model.

¹ Direct solvers, however, are not affected by the incompressible condition, and the solution will remain identical when solving the linear system of equations.

Figure 4-59 shows the fractions of the three components varying with the number of processors used for calculation. It is seen that the fraction of FETI iteration over the total execution time increases with number of processors, measuring 61% with 3 processors and reaching 83% with 16 processors. The large percentage suggests the FETI iteration loop should be targeted for further performance enhancement.

The measured computation time for 31 excavation steps in Figure 4-58 cannot reflect the true solution time. This is because the system are shared by other computationally intensive tasks from other users, thus do not represent the actual solution time when the FETI calculation monopolizes the system. However, the parallel efficiency should not be affected by the other process in the system when the CPU usage of the other process is constant on all participating processors.

From Figure 4-58, *FETI_FEM* uses 9994 seconds with 3 processors and 3972 seconds with 16 processors to complete the analysis, while ABAQUS uses 18609 seconds (Table 4-2) on geohazard.mit.edu. *FETI_FEM* cannot complete the analysis with less than three processors because the memory resource becomes exhausted. Therefore, the performance of *FETI_FEM* does not seem to be efficient, even though computers in the cluster have both better theoretical peak performance (2.8Gflops versus 1.2Gflops) and memory bandwidth (2.1GB/sec versus 1.2GB/sec). However, the computers are shared by one other computational task. Additionally, and the calculation used fast Ethernet rather than Myrinet, thus the performance is not optimal for calculations shown in Figure 4-58. Nevertheless, the observed solution time reduces with increasing number of processors used, and exhibits about 50% parallel efficiency with 16 processors using Fast Ethernet.

4.7.4 Effect of Communication Fabric on the Performance of FETI_FEM

To understand how important the communication fabric is to FETI_FEM, two series of tests are conducted using Fast Ethernet with MPICH-1.2.5 and Myrinet with MPICH-GM-1.2.5. The underlying algorithms used in these libraries are assumed identical because the same version numbers (1.2.5), thus the observed performance differences should owe to the communication hardware entirely. The tests are based on the circular tunnel analyses shown in Figure 4-57, but with two steps of excavations instead of 31 steps.

Figure 4-60 shows the effect of different networking fabric on the parallel performance. It is seen that better speedup and parallel efficiency are achieved using Myrinet compared to Fast Ethernet. For the 14-processor case, *FETI_FEM* achieves 84.3% parallel efficiency using Myrinet, compared to 58% efficiency using Fast Ethernet.

The effect of different networking fabric on different components of *FETI_FEM* is illustrated in Figure 4-61 by the ratio of measured time (vertical axis) from different communication fabric spent on initial setup, residual evaluation, and FETI iteration. It is seen the time spent on initial setup and on residual evaluation are similar between fast Ethernet and Myrinet (the ratio is close to unity), and the ratio is insensitive to the number of processors as there are little communications in these stages. On the contrary, the time spent on the FETI iteration stage shows dramatic differences between these two communication links. Myrinet has lower latency and higher bandwidth, and therefore communications can be performed at much higher rate than Fast Ethernet. As a result, using Myrinet achieves 240% improvement of time spent on FETI iterations compared to Fast Ethernet as shown in Figure 4-61, and the gap between Myrinet and fast Ethernet increases as the number of processors increases. Therefore, it is essential to use high-speed networking fabrics for large clusters when solving finite element models using *FETI_FEM*.

4.7.5 Effect of Preconditioners

Table 4-5 summarizes the calculation times using 14 processors for the stacked-drift tunneling model¹ to be introduced in Chapter 5 with both Fast Ethernet and Myrinet using different preconditioners. It can be seen from these results that preconditioners have pronounced effects on the number of iterations for the convergence of the FETI method. Among the implemented preconditioners, the Mechanically Consistent (MC) Dirichlet preconditioner yields the lowest iteration count, while the lumped preconditioner yields the highest. The last column of Table 4-5 shows the time needed to perform a single FETI iteration, and it is seen that two to three times more iterations can be performed using Myrinet than using Fast Ethernet. This is because Myrinet imposes lower communication cost than Fast Ethernet. It should be noted that for Myrinet, even though the number of

¹ The model contains 379,842 degrees of freedom, 126,494 nodes, and 30,053 20-noded elements with 168 sub-domains.

iterations for the MC lumped preconditioner is 25% more than the MC Dirichlet preconditioner, the overall solution time is nearly identical because the MC lumped preconditioner is cheaper to compute than the MC Dirichlet preconditioner, thus makes up the differences on the iteration count. Similarly, the solution time using the lumped preconditioner is less than using the Dirichlet preconditioner with Myrinet. It has been suggested by Farhat and Roux (1994) that the lumped preconditioner is preferred as it is cheaper to compute, requires less memory, and provides faster solution times than the Dirichlet preconditioner. However, this is not true for Fast Ethernet. As seen in Table 4-5, a larger number of iterations increase the solution time for Fast Ethernet, as the cost per iteration is much higher with Fast Ethernet than with Myrinet.

In conclusion, the MC version of lumped and Dirichlet preconditioners should always be used, as their computation cost is nearly identical to the original (lumped and Dirichlet) preconditioners, and MC version always produces lower iteration counts. The choice between MC lumped and MC Dirichlet preconditioners depends on the available amount of memory, the communication fabric used, and how much reduction of iteration counts can be achieved by using MC Dirichlet preconditioner. For Fast Ethernet connections, the MC Dirichlet is preferred if the amount of memory is sufficient, while for Myrinet, the MC lumped preconditioner can be used as there appears to be less than 15% difference in iteration counts obtained with MC Dirichlet.

4.7.6 Suggestions for Future Enhancements on FETI_FEM

Many performance enhancements are possible for *FETI_FEM*. The current implementation parallelizes almost every calculation within the FETI iteration loop and distributes computations onto all worker processes except the calculations involve $(\mathbf{G}_I^T \mathbf{G}_I)^{-1}$, which is calculated solely on the master process¹. It is possible to parallel factorize $\mathbf{G}_I^T \mathbf{G}_I$, but should be done with very few processors instead of all processors.

It is also possible to collect \mathbf{G}_I from all worker processes and store them on the master process (Charmpis and Papadrakakis 2002), and no communications are then needed for the

¹ This matrix is relatively small and densely populated, thus can be effectively factorized on a single modern processor.

null-space projection step (B in Figure 4-33) and therefore fewer communication is needed during the FETI iteration. However, this arrangement increases the workload on the master process and reduces the workload on all worker processes, and may be slower for large problems than the current implementation.

The current implementation evenly distributes search directions \mathbf{p}_i onto all processes, and utilizes all processes for back-orthogonalization. Alternative back-orthogonalization is also implemented by using the master process alone, thus removes one pair of broadcast/reduce operations from the FETI iteration. However, the alternative is less efficient based on numerical experiments, and gradual degradation of CPU utilization on all worker processors are observed because the master process expends more time for back-orthogonalization. However, the alternative may be advantageous if the master processor is significantly faster than worker processes.

Currently the load distribution is determined entirely based on the factorization performance of participating processors and ignores the available memory resources. Currently the master process use relatively less memory than worker processes because worker processes need to store sub-domain information, while the master process does not have any sub-domain. As a result, the memory resources on the master processor are not fully utilized (compared to the worker processes).

At the worker process level, all sub-domain information is stored in the memory, and not all of this information is needed during the FETI calculation. Therefore, it is feasible to store the information temporarily into hard disks, and free the memory for FETI calculations, enabling larger problems to be solved than the current implementation.

4.8 Summary

- In order to solve three-dimensional finite element models for tunneling, several efficient finite element solutions (remeshing, Schwarz alternating scheme, and the FETI method) have been attempted. It is found parallel computation is necessary to have a general methodology for solving finite element models for tunneling with desired accuracy.

- It is found that the remeshing technique described in Section 4.3 can be an efficient finite element solution for modeling tunneling process. However, it is difficult to generalize this approach for applications with multiple tunnel headings advancing in different directions.
- Schwarz alternating scheme is an overlapping domain decomposition method that solves physical problems by exchanging information between adjacent sub-domains. This method can be easily applied on interconnected parallel computers with existing finite element software packages.
- Based on numerical experiments, the Schwarz alternating scheme exhibits slow convergence and hence, offers no advantage over traditional finite element solutions using a single processor.
- One-level FETI algorithm (*FETI_DD*) is investigated by using ABAQUS to calculate stiffness matrices and force vectors. Therefore, the development can be concentrated on the solution technique, and most capabilities of ABAQUS can be directly employed.
- *FETI_DD* demonstrated good parallel efficiency even with a non-dedicated communication network. The calculation of stiffness matrices and force vectors were performance bottlenecks observed. Therefore, *FETI_DD* was further extended into a new code, *FETI_FEM*, an independent finite element package that can be easily deployed on various computing platforms.
- *FETI_FEM* is programmed to utilize input files prepared for ABAQUS. This simplifies the validation process by permitting direct comparison with calculations using ABAQUS.
- The program *EXTRUDE* was created to extrude manually decomposed plane-strain meshes into three-dimensional meshes with decomposition in the extrusion direction. The decomposition in the extrusion direction is determined by prescribing the number of layers per sub-domain.

- The mesh decomposition is done manually at the beginning of the analysis. Meshes need to be decomposed carefully so that one sub-domain does not become two separate entities during the material removal operation.
- It is possible to automatically decompose a finite element mesh through existing packages such as METIS (Karypis and Kumar 1998). However, there is little control on how the mesh is decomposed. As a result, automatic mesh decomposition needs to be done at every single analysis step, and the number of sub-domains will no longer be constant.
- The calculation results of *FETI_FEM* need to be processed through *FEMPost*, a code developed for this work. This post-processing step is necessary to generate data files for visualization using Tecplot. Additionally, second order brick elements need to be processed into first order brick elements in order to be visualized.
- *FETI_FEM* shows excellent parallel efficiency and insensitivity to the problem size. The solution time versus problem size relationship can be fitted well with a power law using an exponent of $p = 1.217$. The direct solver in ABAQUS exhibits greater sensitivity with $p = 1.5136$. Thus, *FETI_FEM* provides a dramatic improvement in the ability to solve large-scale FE analysis of tunneling problems.
- The choice of preconditioner depends on the communication fabric used. For Fast Ethernet, the MC Dirichlet preconditioner should be used if the available memory is sufficient. Otherwise, the MC lumped preconditioner should be used. As the communication cost is high for Fast Ethernet, each saving on the FETI iteration count improves the solution efficiency.
- For Myrinet, the choice between MC Dirichlet and MC lumped depends on how much reduction of iteration counts can be achieved using the MC Dirichlet preconditioner.
- *FETI_FEM* demonstrates good computational and parallel efficiency, and is used to solve the stacked-drift tunneling construction in Chapter 5.

Table 4-1 Material Properties and K_0 for Base-case Problem

Material	Young's modulus E [MPa]	Poisson's ratio ν ⁽¹⁾	Density [kN / m ³]	K_0	Yield stress ⁽³⁾ [KPa]
Linear Elastic	10	0.49	16	0.96 ⁽²⁾	---
Elasto-Plastic				1.0	67

⁽¹⁾ Undrained condition \equiv incompressible condition

⁽²⁾ Based on elasticity theory ($K_0 = \nu / (1 - \nu)$)

⁽³⁾ Von-Mises yield stress: $\sigma_y = \sqrt{3J_2}$, $J_2 = \frac{1}{2} \sigma_{ij} \sigma_{ij}$

Table 4-2 Computation Resources Required for Base-case Problem

Analyses	CPU Time ⁽¹⁾ (seconds)		Storage ⁽²⁾ [GB]
	[seconds]	[hours]	
Elastic, Reduced Integration ⁽³⁾	16833	4.7	1.2
Elastic, Full Integration ⁽²⁾	18609	5.2	2.6
Plastic, Reduced Integration ⁽³⁾	87991	24.4	1.2
Plastic, Hybrid Formulation ⁽⁵⁾	99187	27.6	3.2

⁽¹⁾ Computer: 600MHz Alpha 21164 CPU, 1088MB RAM, SCSI hard drive

⁽²⁾ Excluding the temporary storage needs.

⁽³⁾ 8 Gauss points

⁽⁴⁾ 27 Gauss points

⁽⁵⁾ 27 Gauss points + 1 additional unknown for pressure

Table 4-3 Solution Time for Full and Decomposed Finite Element Solution

	Decomposed FEA [seconds] ⁽¹⁾	Full FEA [seconds] ⁽¹⁾
Sub-domain #1, #5	78	---
Sub-domain #2, #6	23	---
Sub-domain #3, #7	21	---
Sub-domain #4, #8	36	---
Total	316	787
Number of sub-domain solves	107	1
Total solution time	3134 ⁽²⁾	787 ⁽³⁾

⁽¹⁾ Wall clock time

⁽²⁾ Pentium III-1.0GHz and Pentium III-866MHz, 512MB RAM

⁽³⁾ Pentium III-1.0GHz

Table 4-4 Ported Keywords from ABAQUS for finite element model definition

	Keyword		Keyword
Node Generation	*NODE	Step Definition	*STEP
	*NCOPY		*MODEL CHANGE
	*NGEN		*STATIC
	*NFILL		*GEOSTATIC
	*NSET		*CLOAD
Element Generation	*ELEMENT		*DLOAD
	*ELSET		*END STEP
	ELSET		
	GENERATE		
	*ELGEN		Silently Ignored
Initial Condition	*INITIAL CONDITIONS, TYPE=STRESS, GEOSTATIC	*NODE PRINT	
Material Definition	*SOLID SECTION	*NODE FILE	
	*MATERIAL	*EL PRINT	
	*ELASTIC		
	*POROUS ELASTIC		
	*PLASTIC	*EL FILE	
	*DRUCKER PRAGER	*ELEMENT MATRIX OUTPUT	
	*MOHR COULOMB	*PREPRINT	
	*DENSITY	*RESTART	

Table 4-5 Effect of Different Preconditioners with Fast Ethernet and Myrinet

	Preconditioner	Total Time (sec.)	FETI Iterations ⁽¹⁾	FETI time (sec.)	Time per FETI iteration (sec.)
Myrinet	Lumped	203.5	309	130.1	0.42
	Dirichlet	211.9	278	136.9	0.49
	MC Lumped	171.6	237	97.2	0.41
	MC Dirichlet	170.5	190	94.9	0.50
Fast Ethernet	Lumped	446.9 (2.2)	309	362.0	1.17 (2.8)
	Dirichlet	427.0 (2.0)	278	339.3	1.22 (2.5)
	MC Lumped	367.8 (2.1)	237	282.3	1.19 (2.9)
	MC Dirichlet	323.3 (1.9)	190	236.6	1.25 (2.5)

⁽¹⁾ The last two steps excavation steps

⁽²⁾ The numbers in the bracket denotes the ratio between fast Ethernet and Myrinet, thus the speed difference

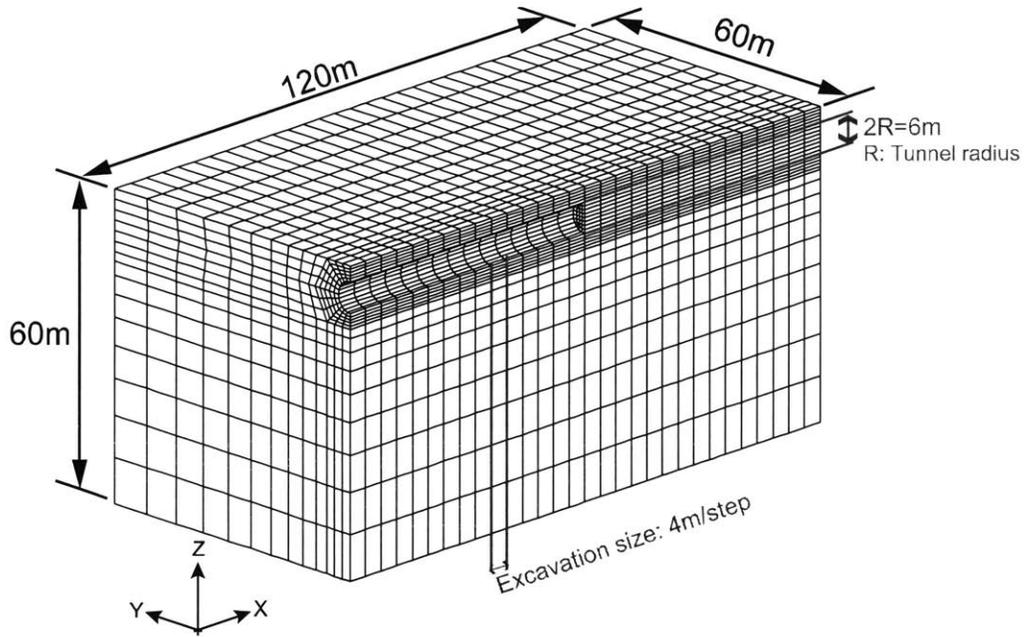


Figure 4-1 Finite Element Mesh for Base-case Study

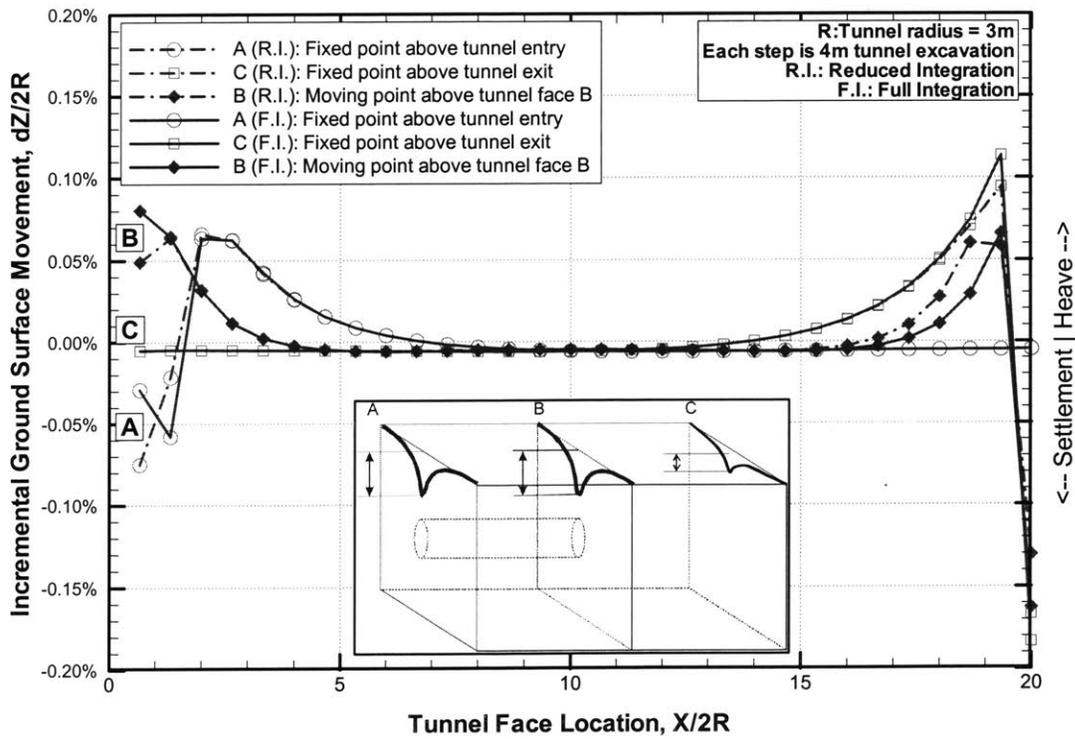
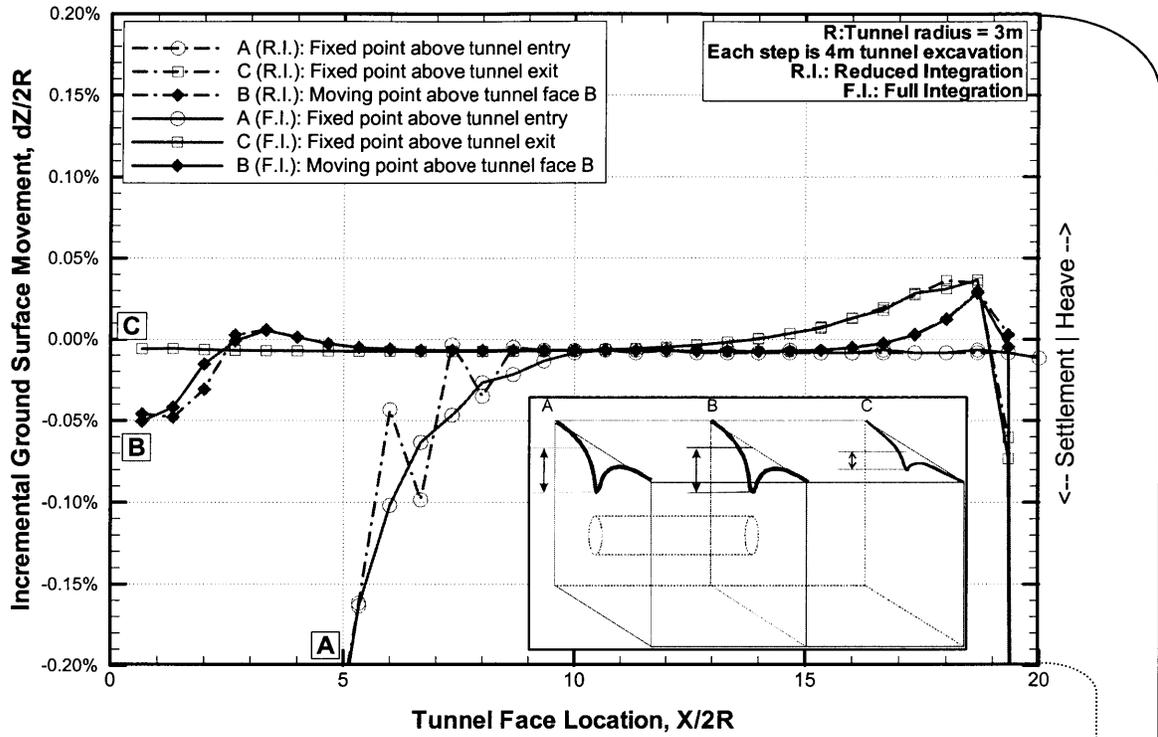
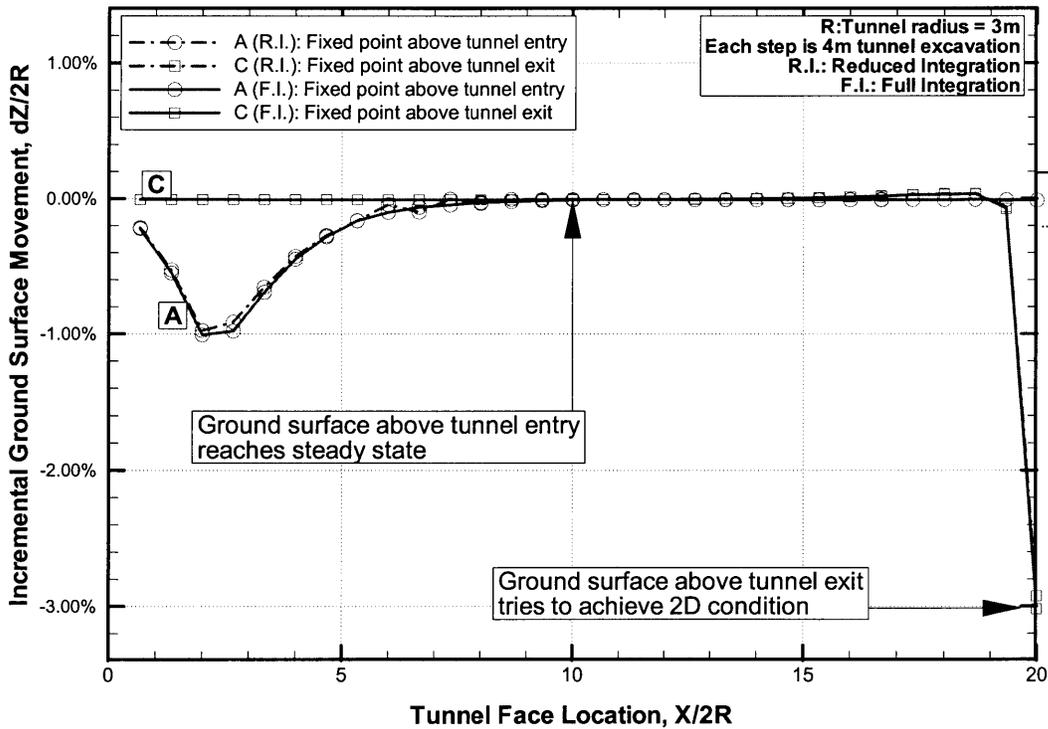


Figure 4-2 Observed Boundary Effects for Elastic Analyses

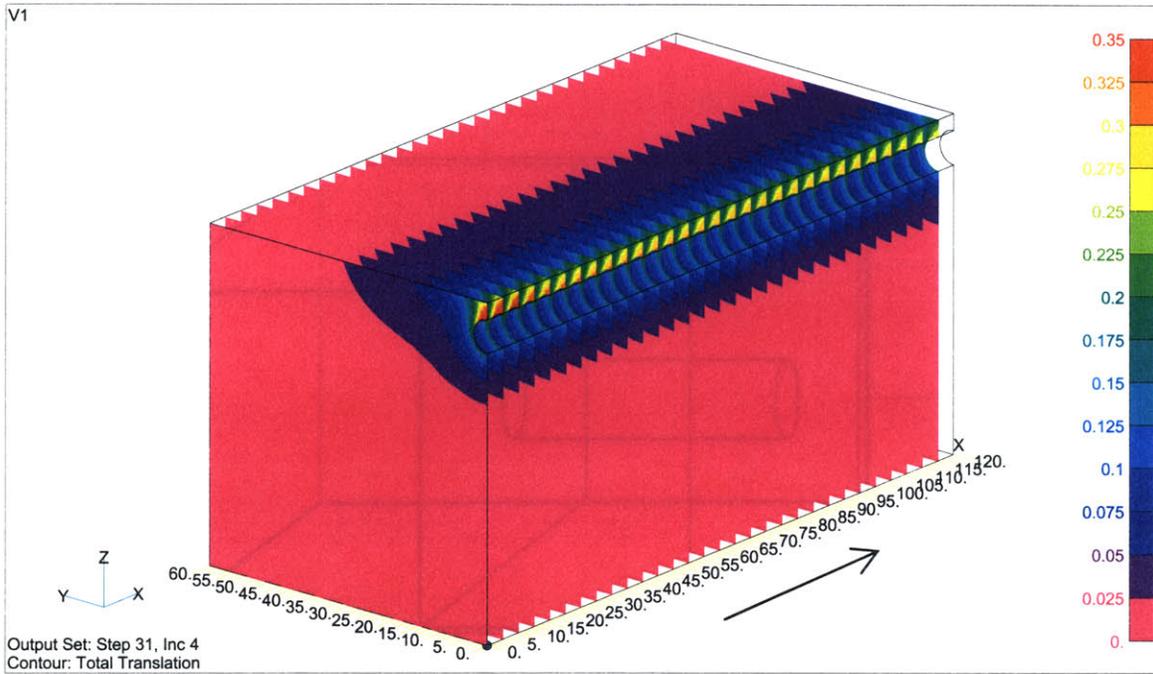


(a) Incremental Surface Movement, between -0.2% – 0.2%



(b) Incremental Surface Movement, between -3.4% – 1.4%

Figure 4-3 Boundary Effects for Elasto-plastic Analyses



(Sections are four-meter apart, unit: m)

Figure 4-4 Magnitude of Displacement Sectional Contours

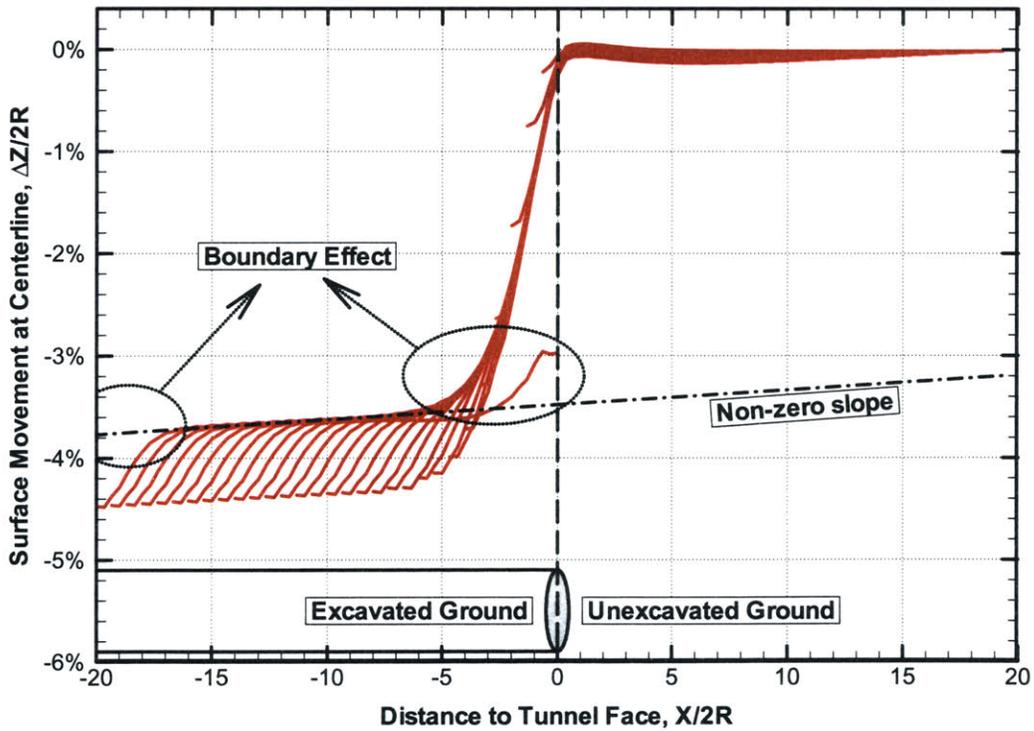


Figure 4-5 Centerline Longitudinal Surface Settlement Profile, Base-case Elasto-plastic analysis

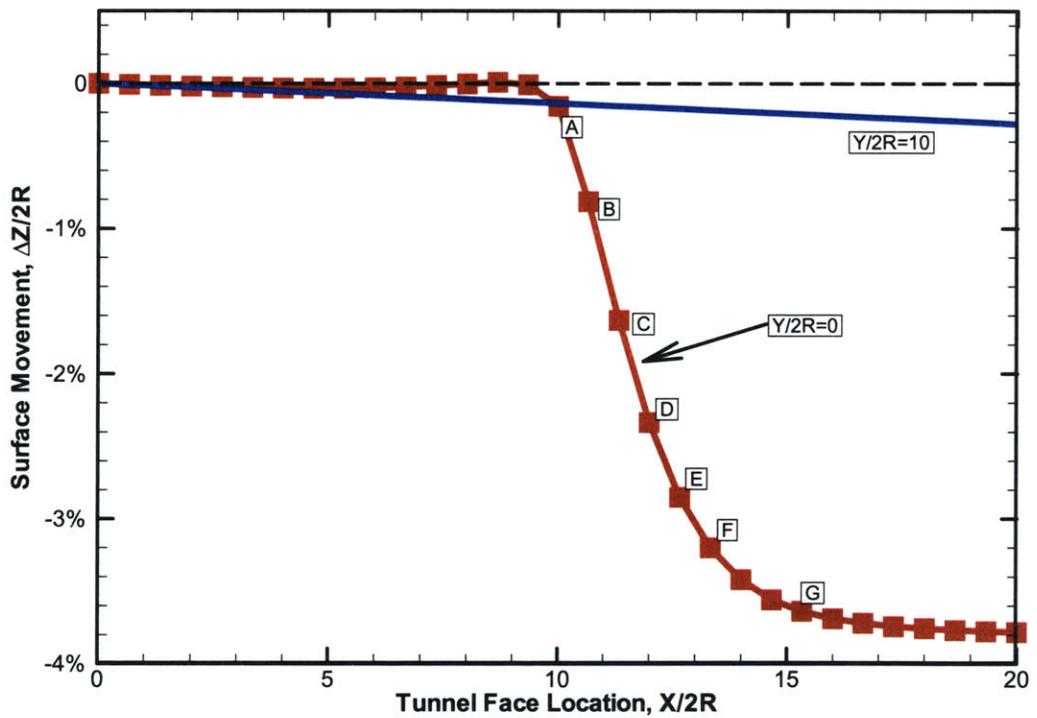
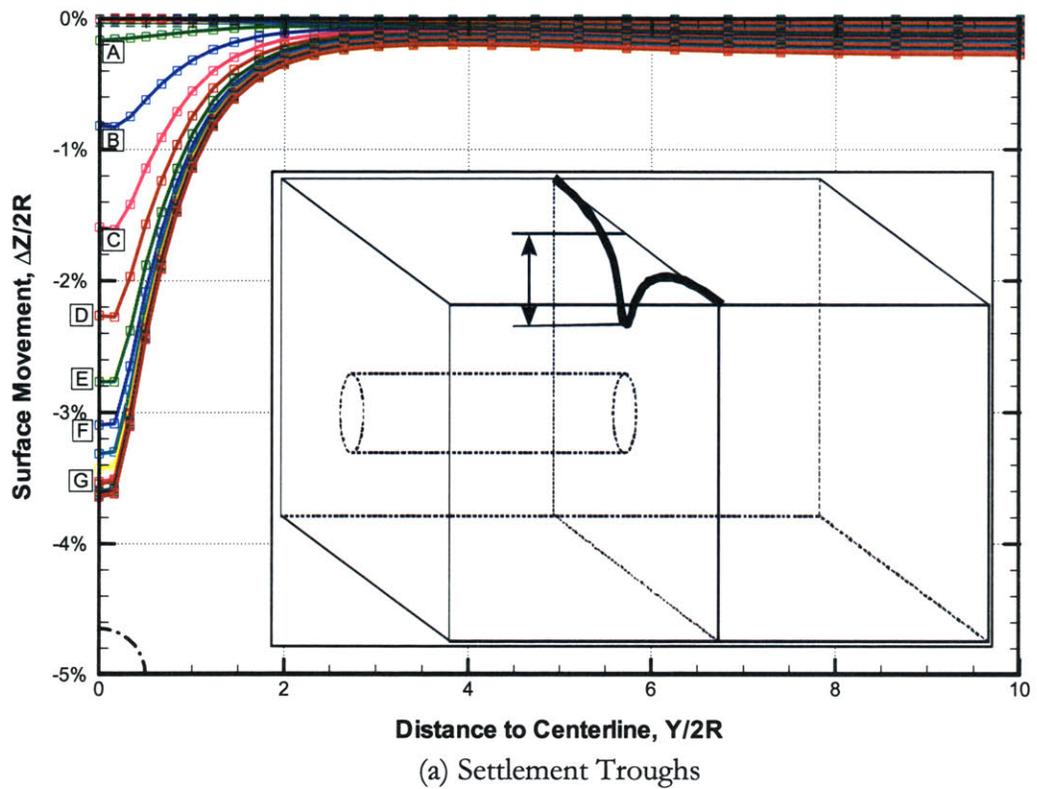


Figure 4-6 Transverse Surface Settlement Troughs at Central Section
Base-case Elasto-plastic Analysis

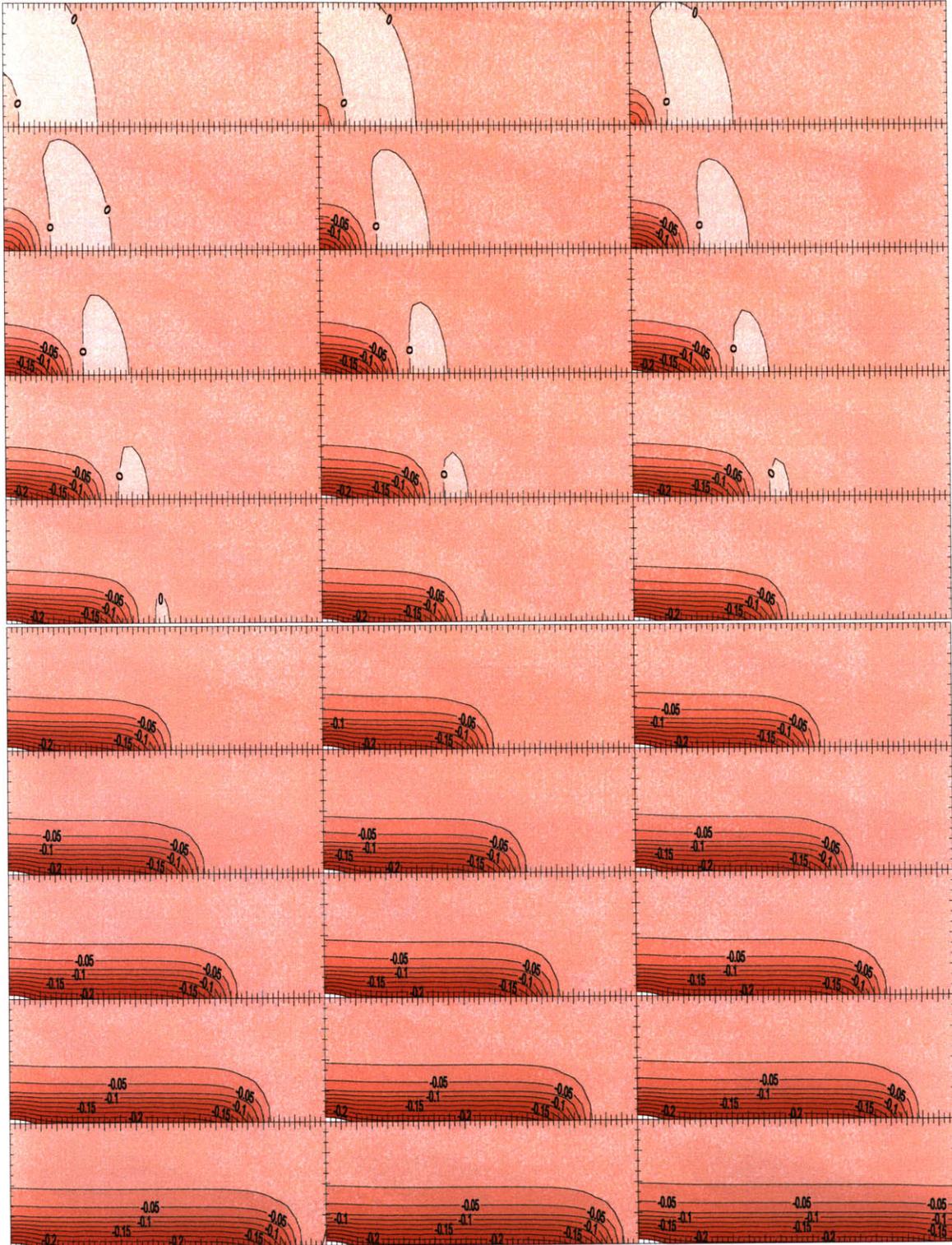


Figure 4-7 Development of Surface Movement with Excavation

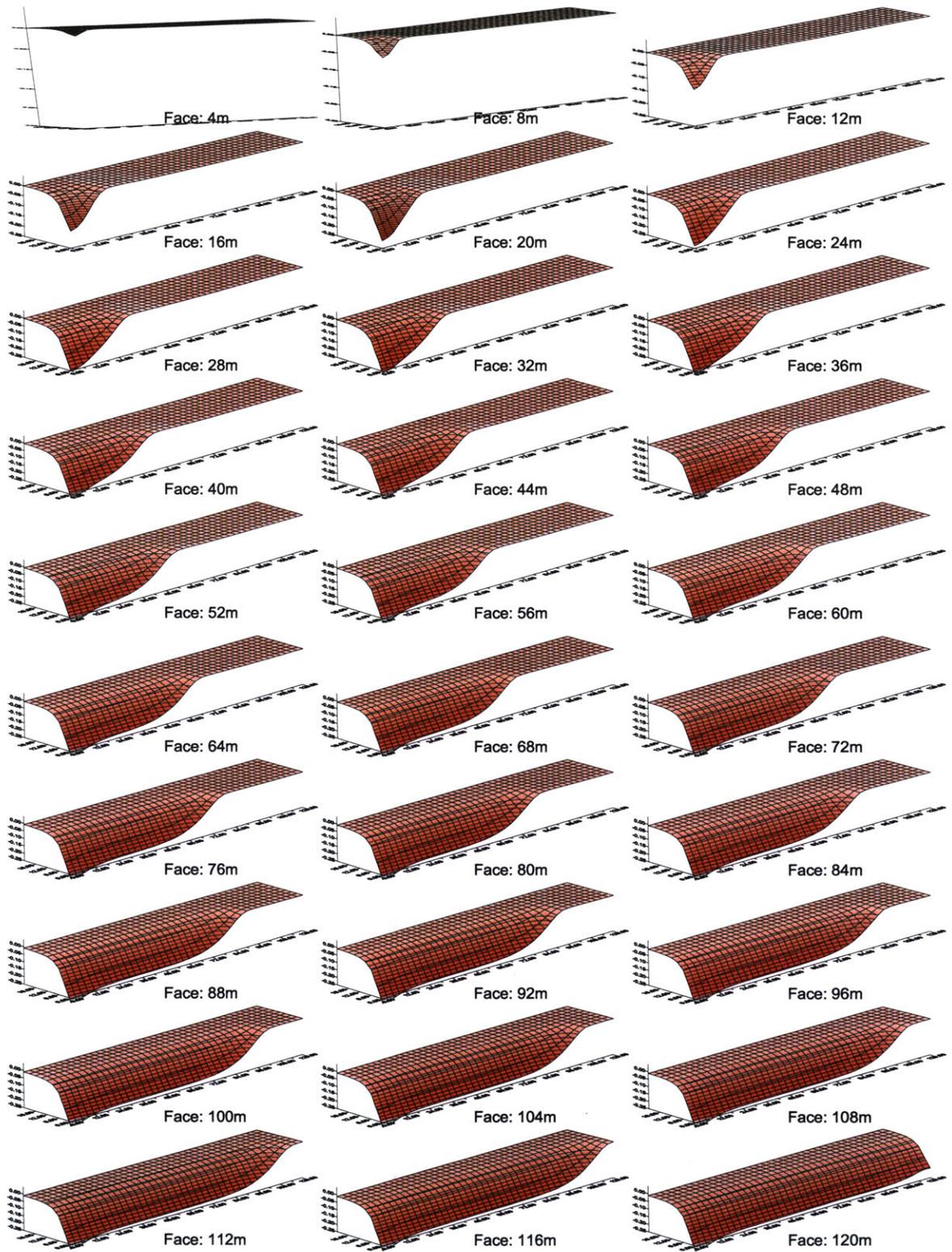
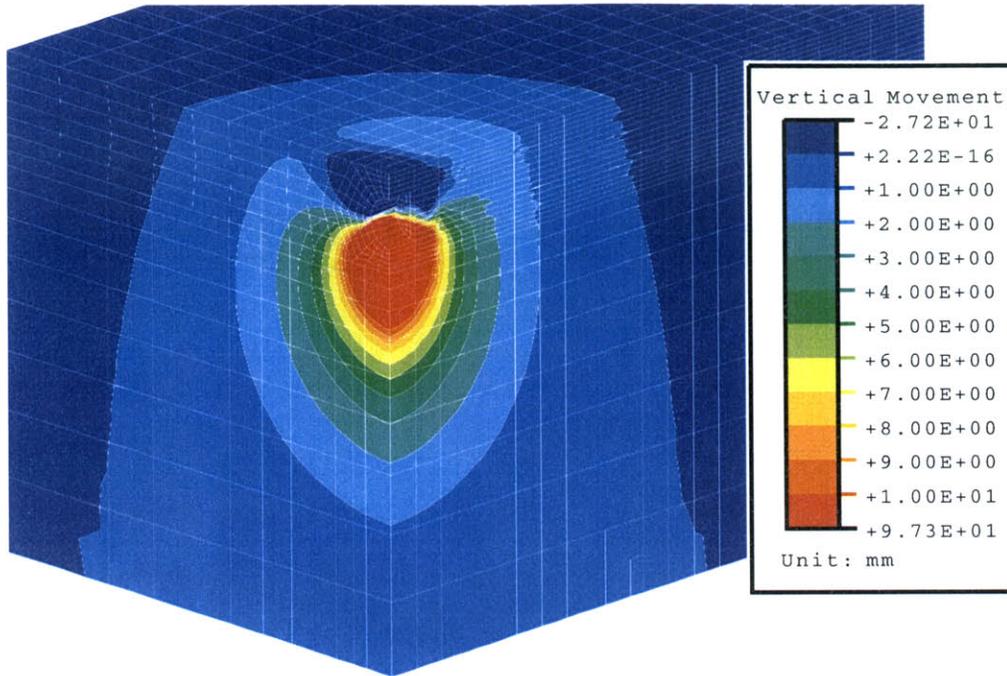


Figure 4-8 Surface Deformation Evolution with Tunnel Face Advancement



$$\Delta X/2R = \frac{4}{6} \cong 0.67$$

Figure 4-9 Contour of the Vertical Movement at Initial Excavation Step

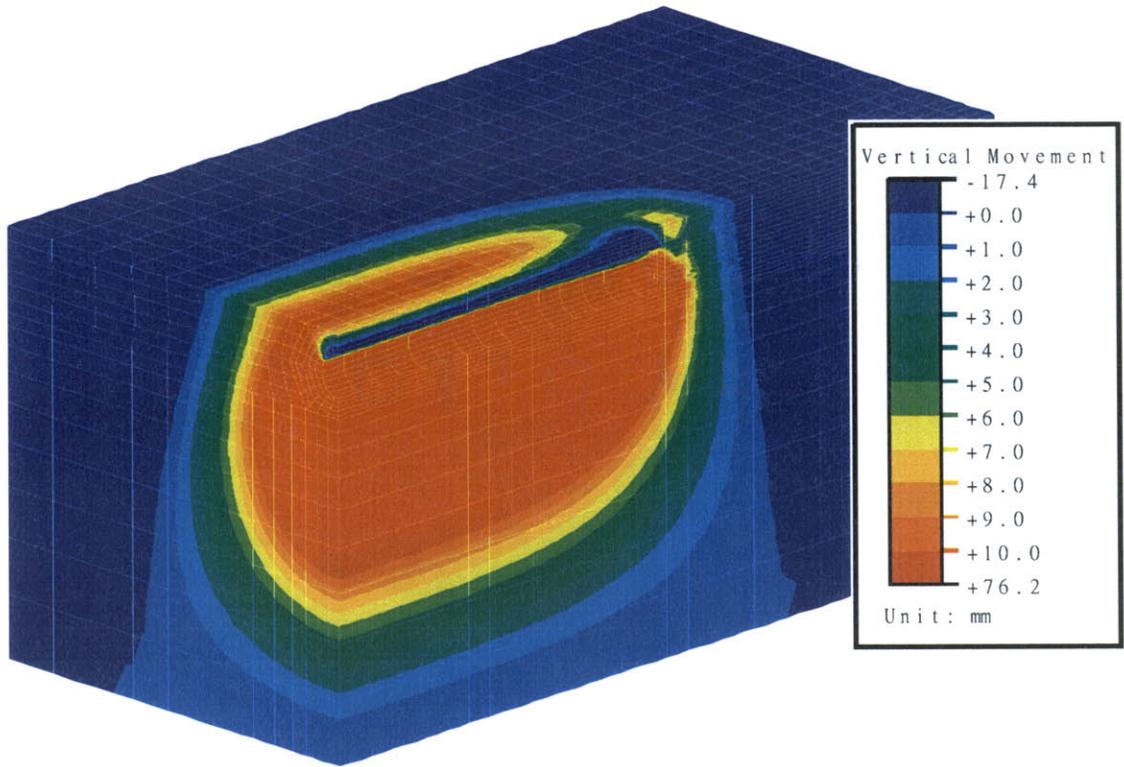


Figure 4-10 Upward Movement for Elastic Analyses

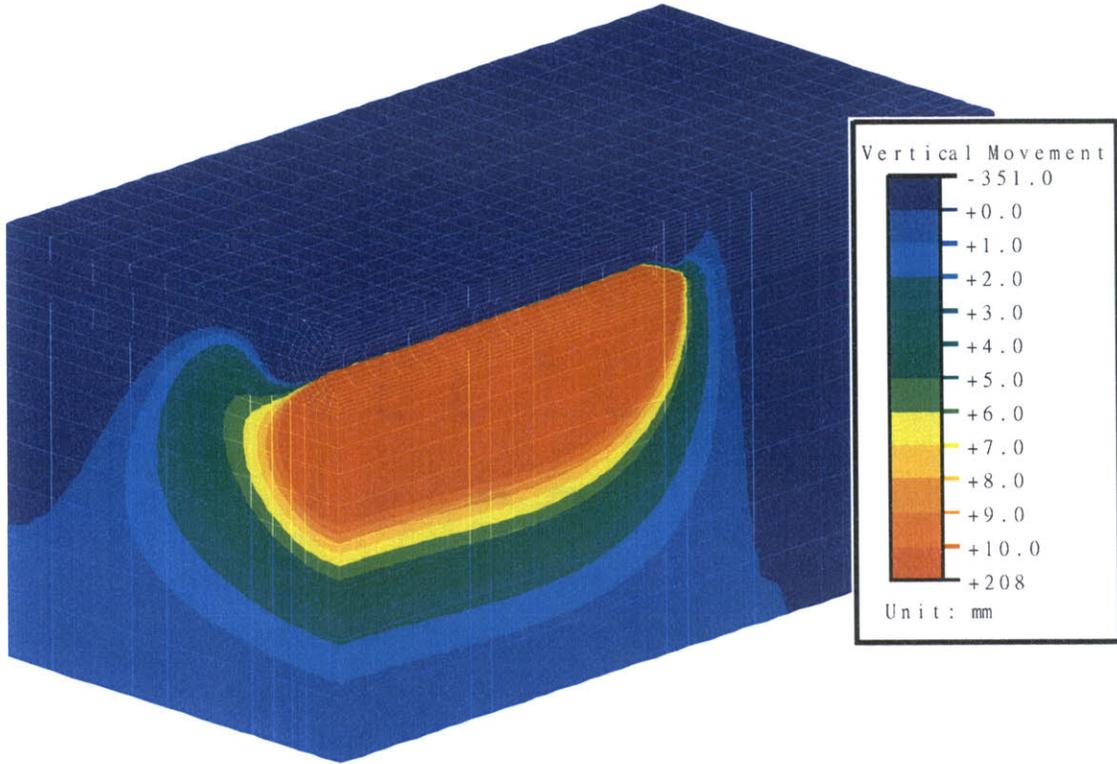
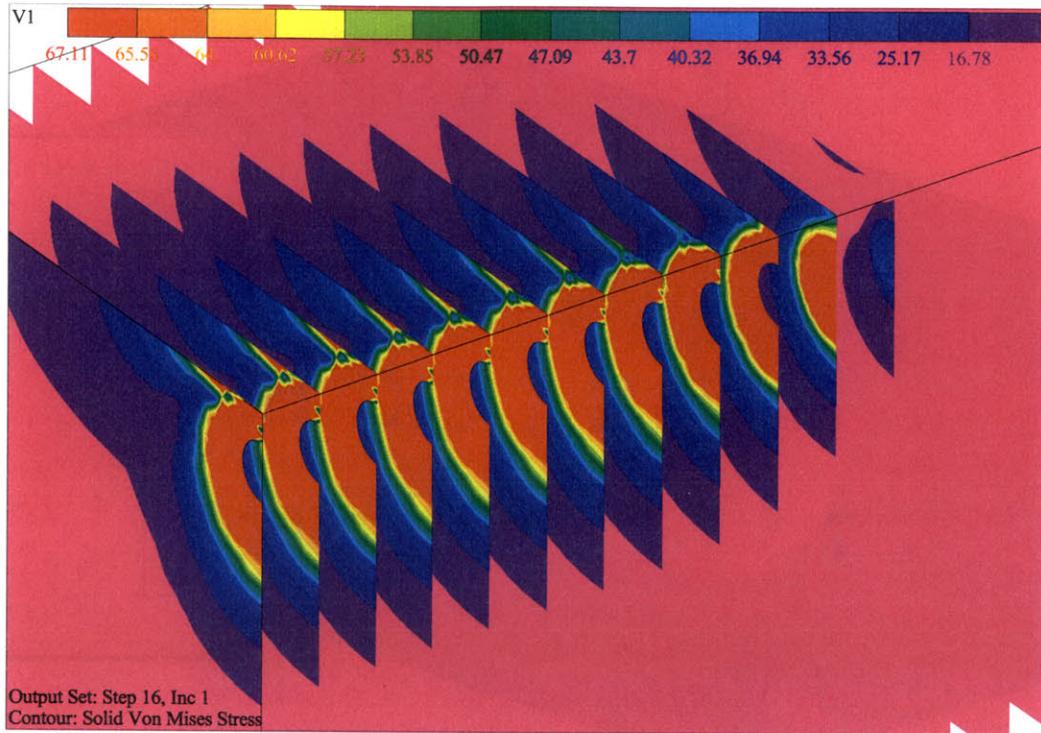
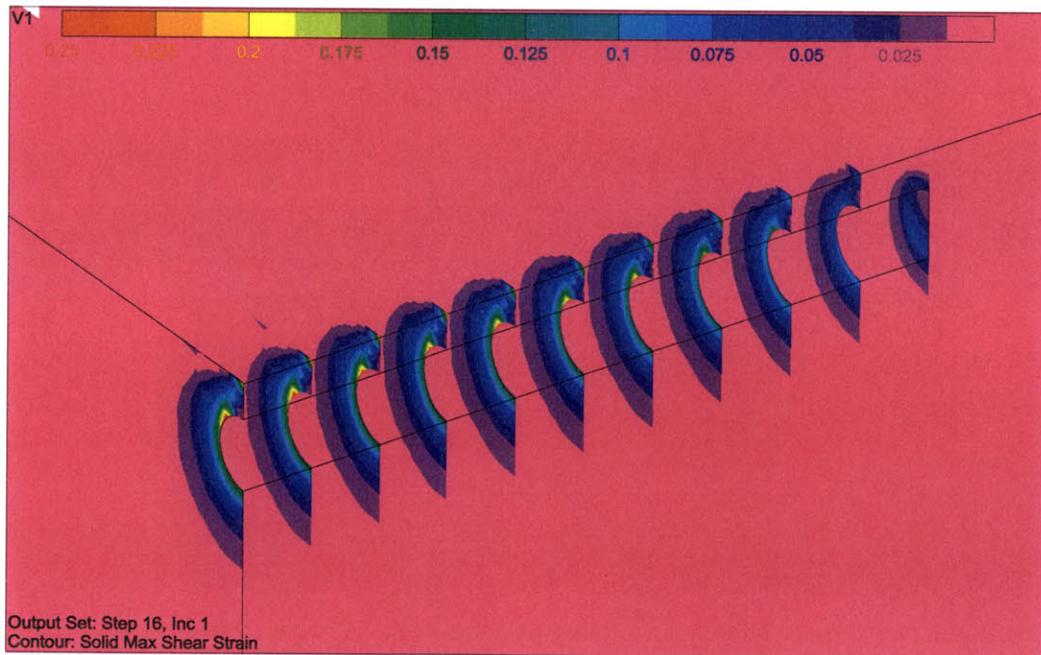


Figure 4-11 Upward Movement for Elasto-Plastic Analyses



(Sections are one tunnel diameter apart)

Figure 4-12 Three-Dimensional von-Mises Stress Distribution



(sections are one tunnel diameter apart)

Figure 4-13 Three-Dimensional Shear Strain Distribution

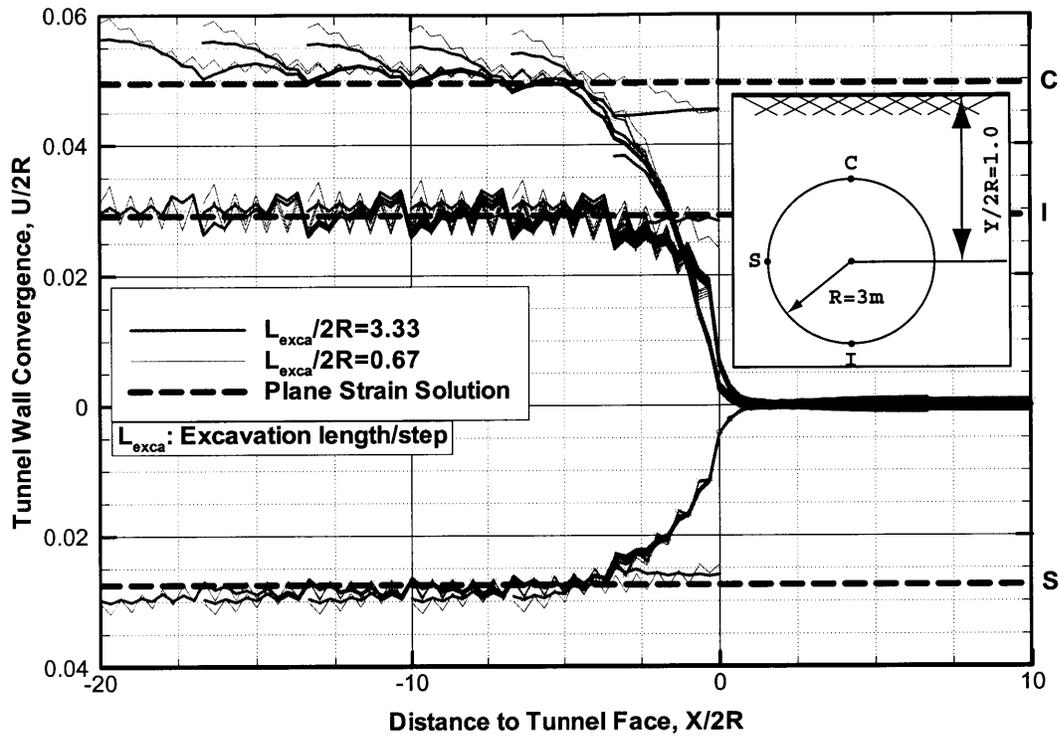


Figure 4-14 Tunnel Wall Convergence for Elasto-Plastic Analyses

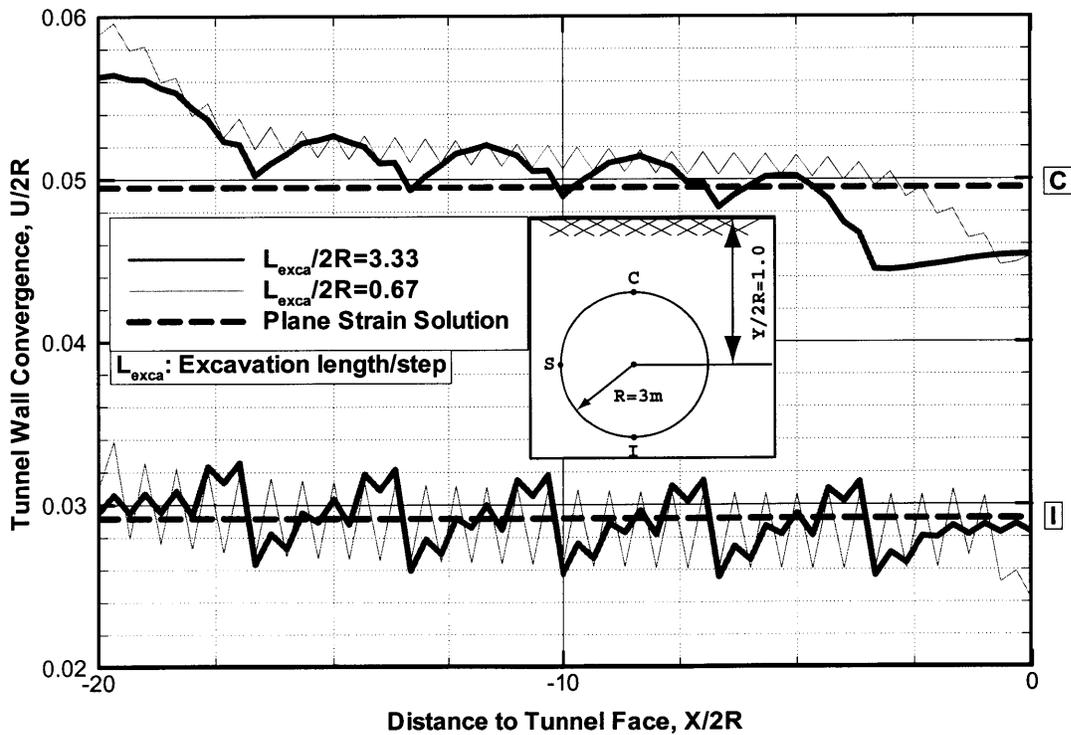


Figure 4-15 Numerical Inaccuracies at Tunnel Walls

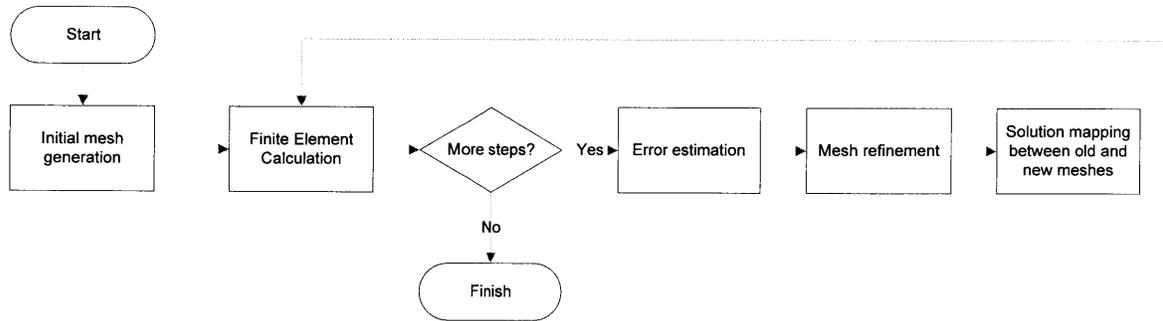


Figure 4-16 Adaptive Remeshing Procedure

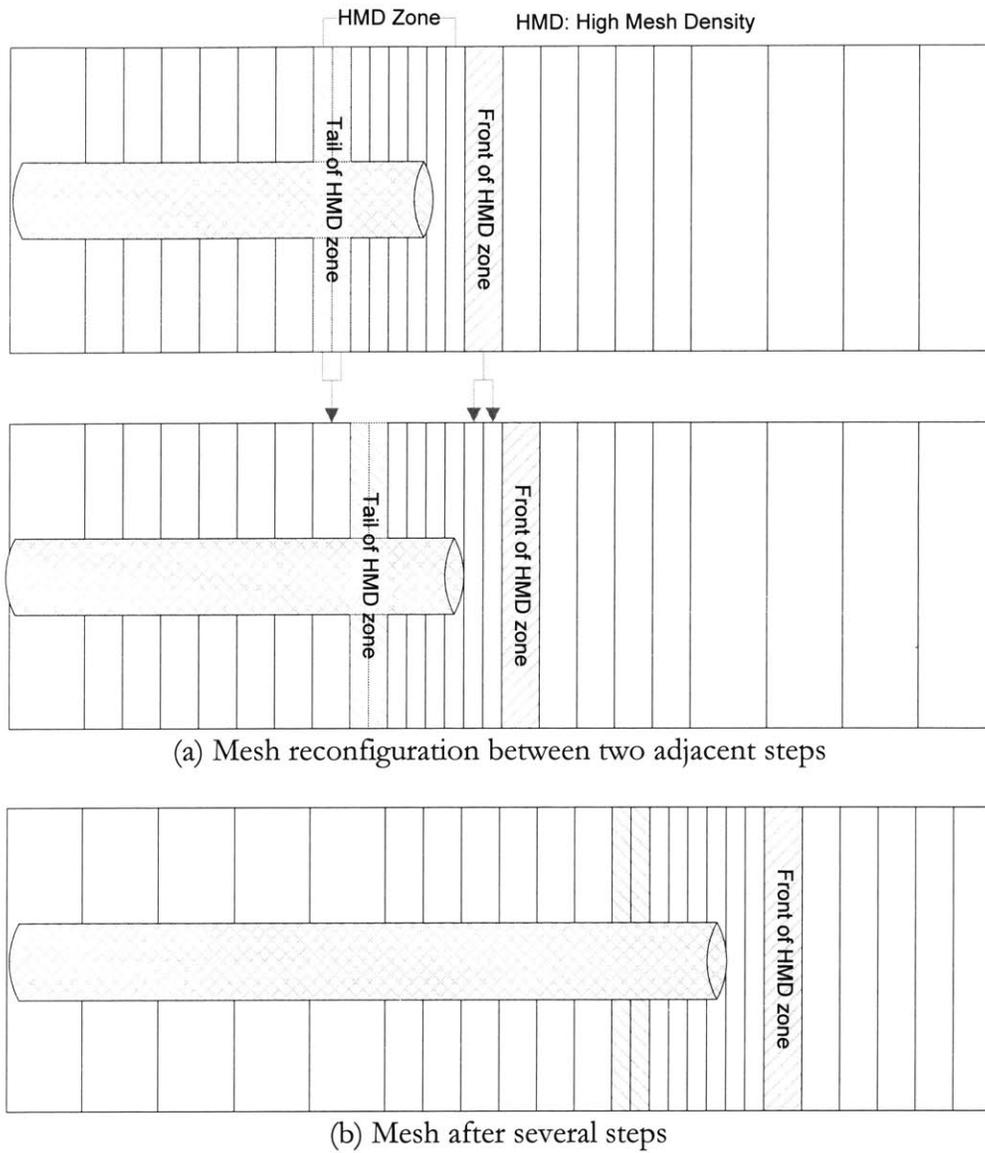
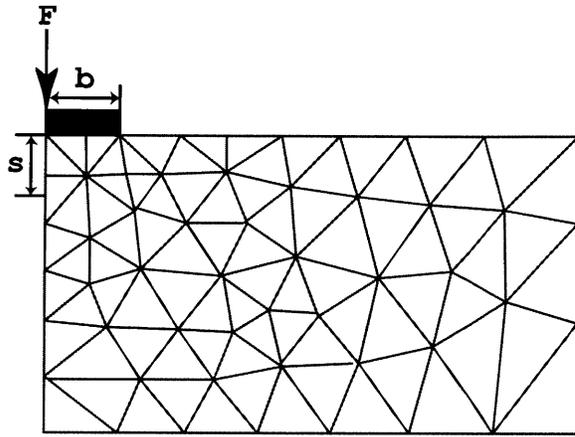
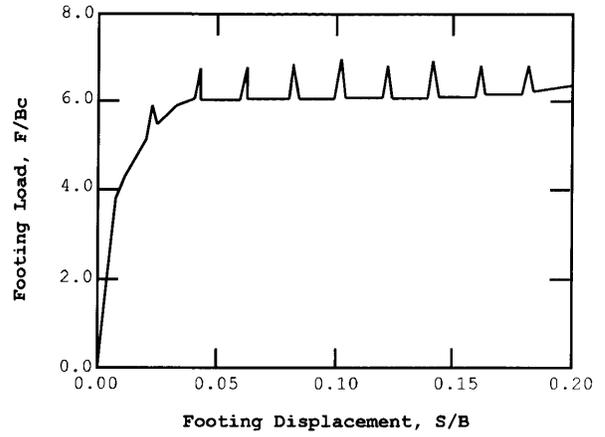


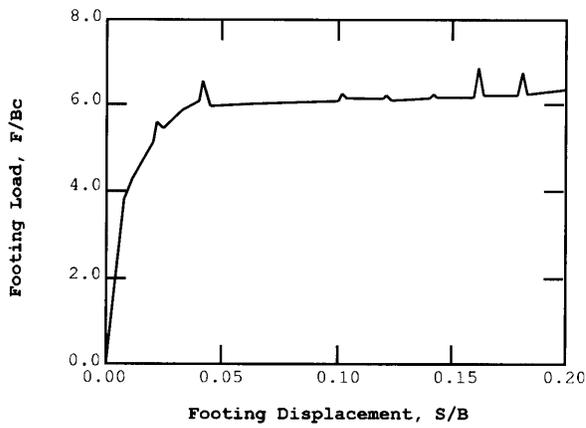
Figure 4-17 Ideal Meshing Scheme for Efficient 3-D Tunneling Analyses



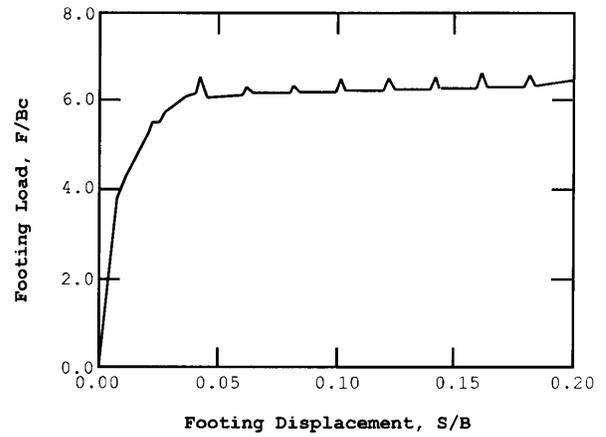
(a) Initial finite element mesh



(b) Load-displacement, ALI interpolation

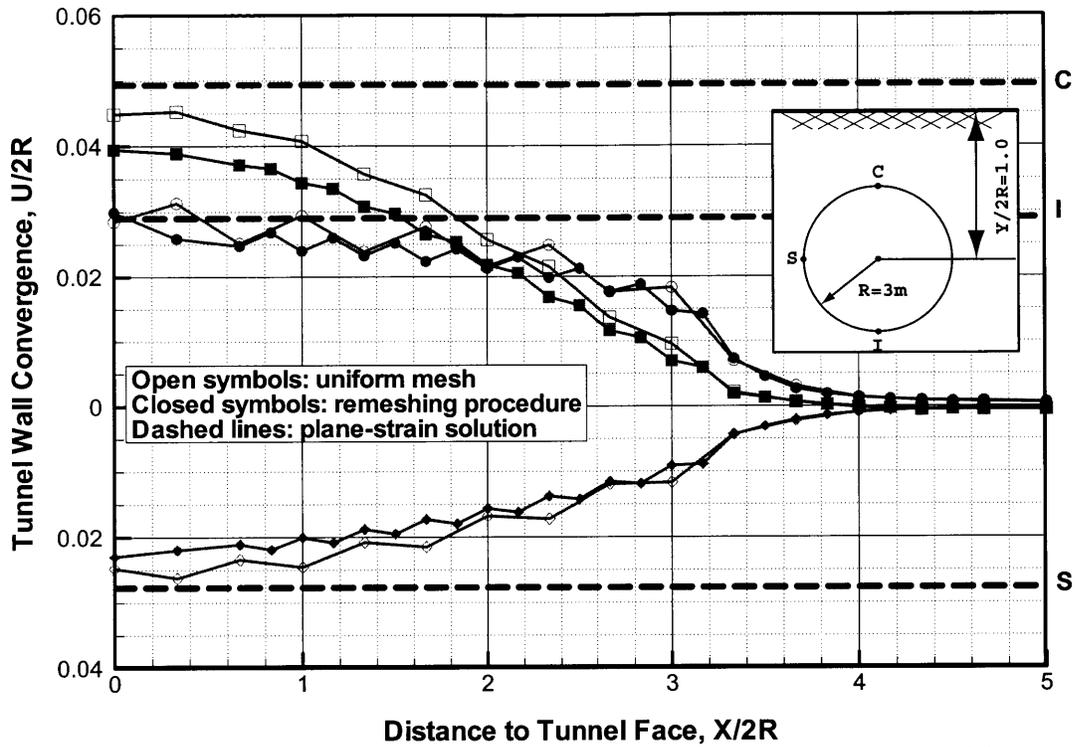


(c) Load-displacement, IDA interpolation

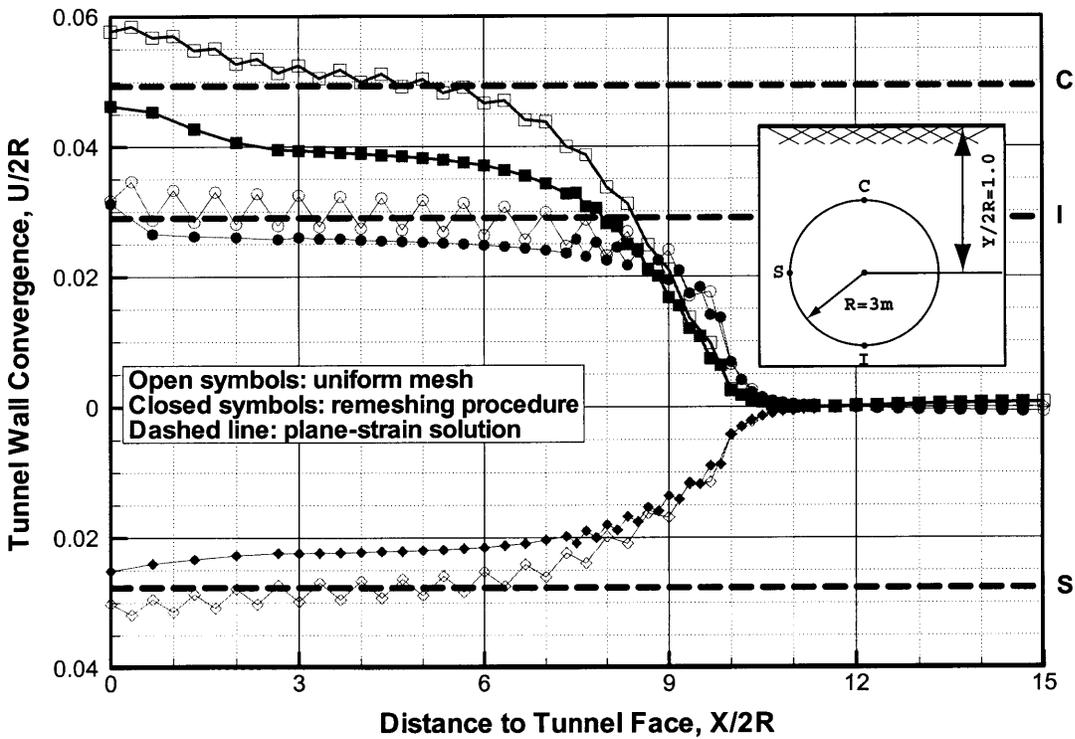


(d) Load-displacement, UEM interpolation

Figure 4-18 Problems with Solution Mapping during Remeshing Process
(Hu and Randolph 1998b),



(a) Tunnel Wall Convergence, Effect of Mesh Density, Tunnel Face at $X/2R = 3.33$



(b) Tunnel Wall Convergence, Tunnel Face at $X/2R = 10$

Figure 4-19 Tunnel Wall Convergence with Remeshing Procedure

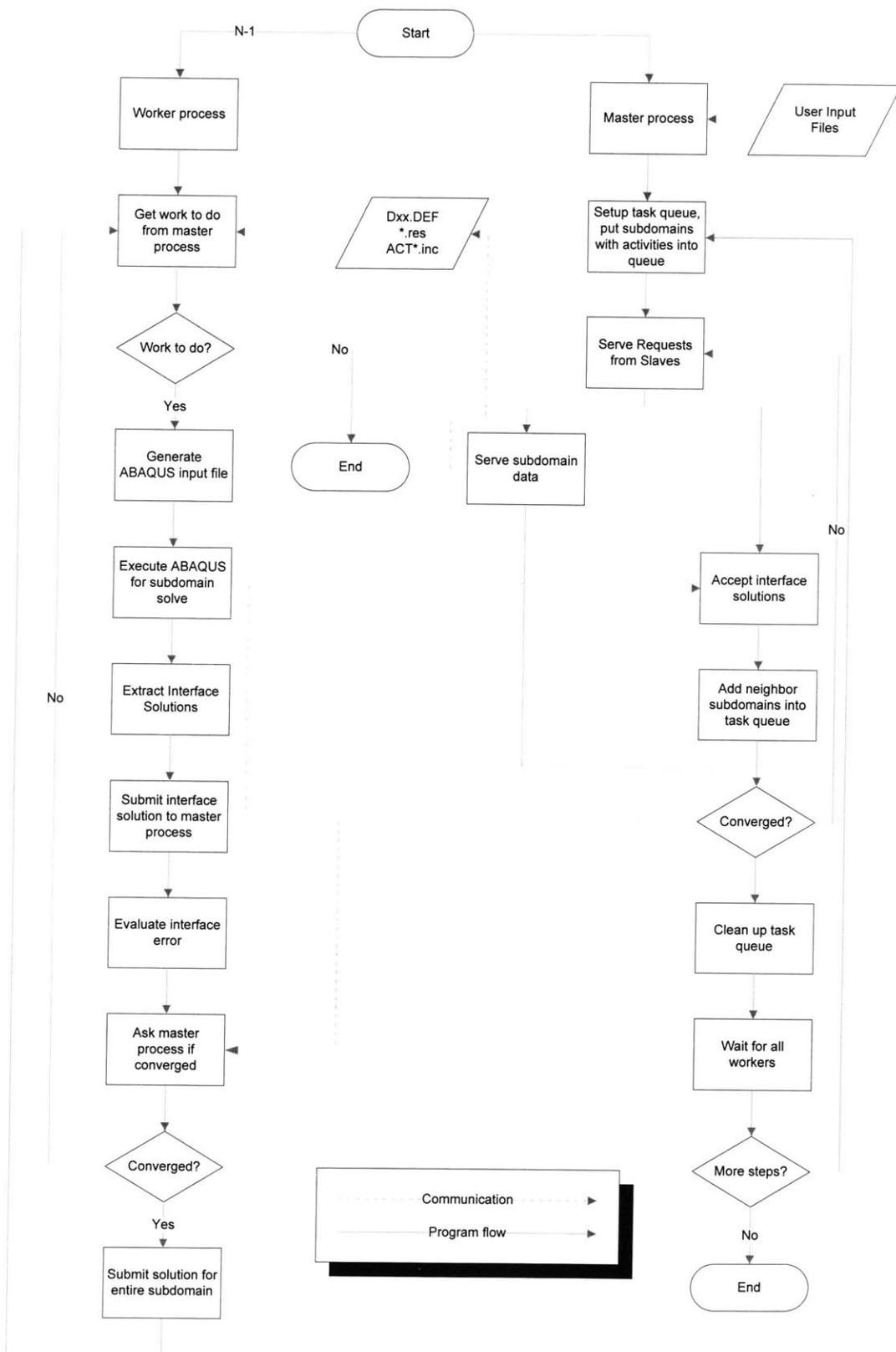


Figure 4-20 Flowchart for the DD Program

```

` Defines a list of valid node numbers
` The file ValidNode.DEF starts with the number of nodes, and
`   each line is contains a node number
*ValidNodeList=19622
ValidNode.DEF

` Defines 8 subdomains, and each subdomain is defined through definition
`   files: D1.DFF, D2.DEF, ..., D8.DEF
*NumberOfSubdomains=8
D1.DEF
D2.DEF
D3.DEF
D4.DEF
D5.DEF
D6.DEF
D7.DEF
D8.DEF
` Dxx.DEF
` Define subdomains and its connectivities
` ID=n                ; define ID of the subdomain currently defining
` INIT=FILENAME       ; define restart file containing initial
`                    ; conditions for the subdomain
` NumberOfNeighbors=n ; Define number of neighboring subdomains
` Neighbor:[DomainID] ; Define neighbor subdomains' "domainID"
` InterfaceElement=n  ; define number of elements at interface
`   Name=[Name]       ; define the name of element set for output
`   followed by [local element ID], [mapped element ID]
` InterfaceNode=n     ; define interface nodes
`   name=[nodeset name] ; define the name of nodeset to be output
`   followed by [local node ID], [mapped node ID]
`   initialGuess=a,b  ; DOFs [a - b] to be fixed
` PermanentBC
`   Node, DOF1, DOF2

` Define 16 action files ACT1.INC, ACT2.INC, ..., ACT16.INC
` Each action is defined with ABAQUS syntax and will be
`   included in generated ABAQUS input file
*NumberOfActions=16
ACT1.INC
ACT2.INC
ACT3.INC
ACT4.INC
ACT5.INC
ACT6.INC
ACT7.INC
ACT8.INC
ACT9.INC
ACT10.INC
ACT11.INC
ACT12.INC
ACT13.INC
ACT14.INC
ACT15.INC
ACT16.INC
` ACT01.INC
` *MODEL CHANGE, REMOVE
` EXCA01,

` Define 2 analysis steps, and each step specifies which action is
`   applied to which subdomain
*NumberOfSTEPS=2
*STEP, 1                ' Define *STEP, and ONE action
  1, 1                  ' [Action ID] on [Domain ID]

*STEP, 1                ' Define *STEP, and one action
  2, 1                  ' [Action ID] on [Domain ID]

```

Figure 4-21 Input File Specification for the *DD* Program

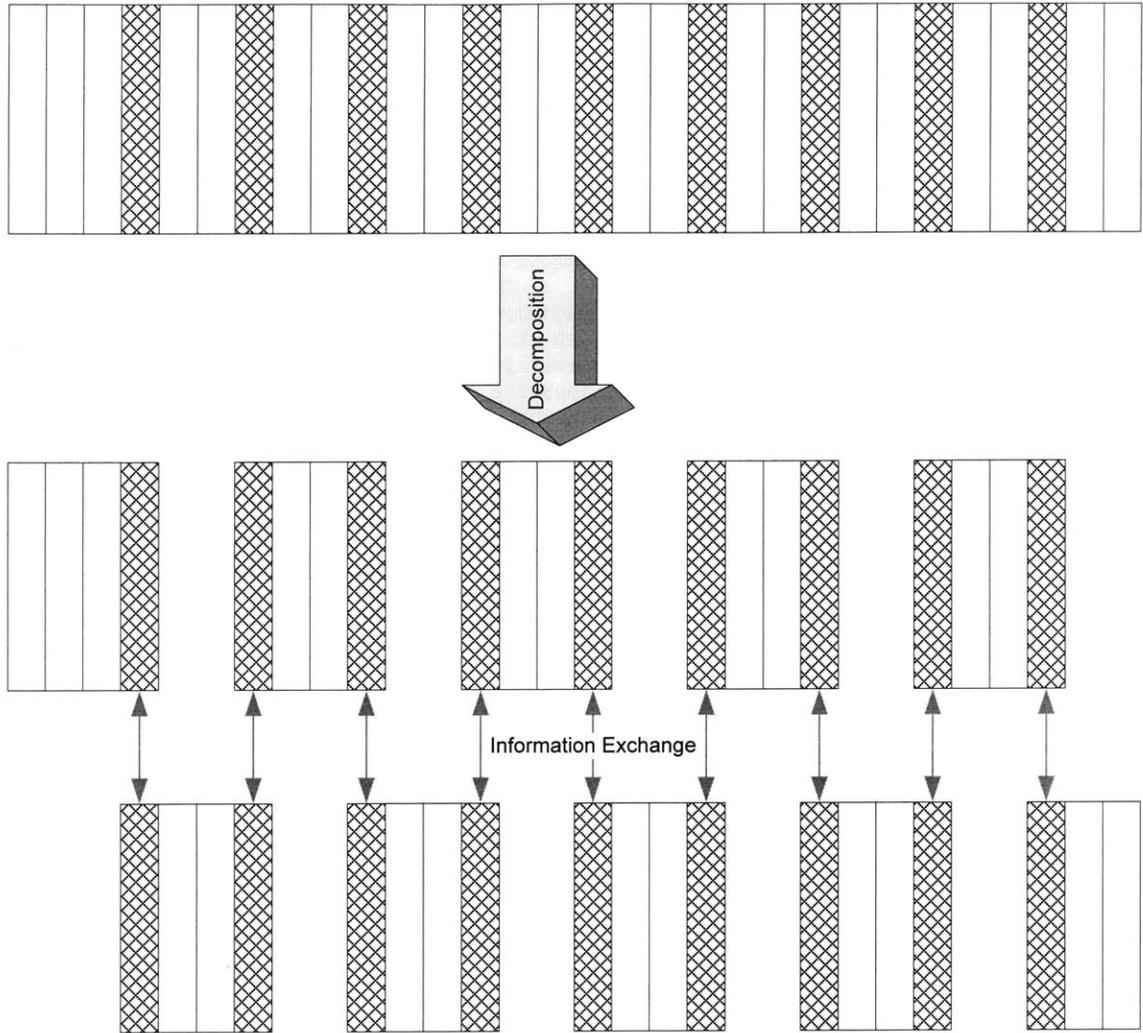


Figure 4-22 Overlapping Domain Decomposition for Extruded 3-D Tunnel Mesh

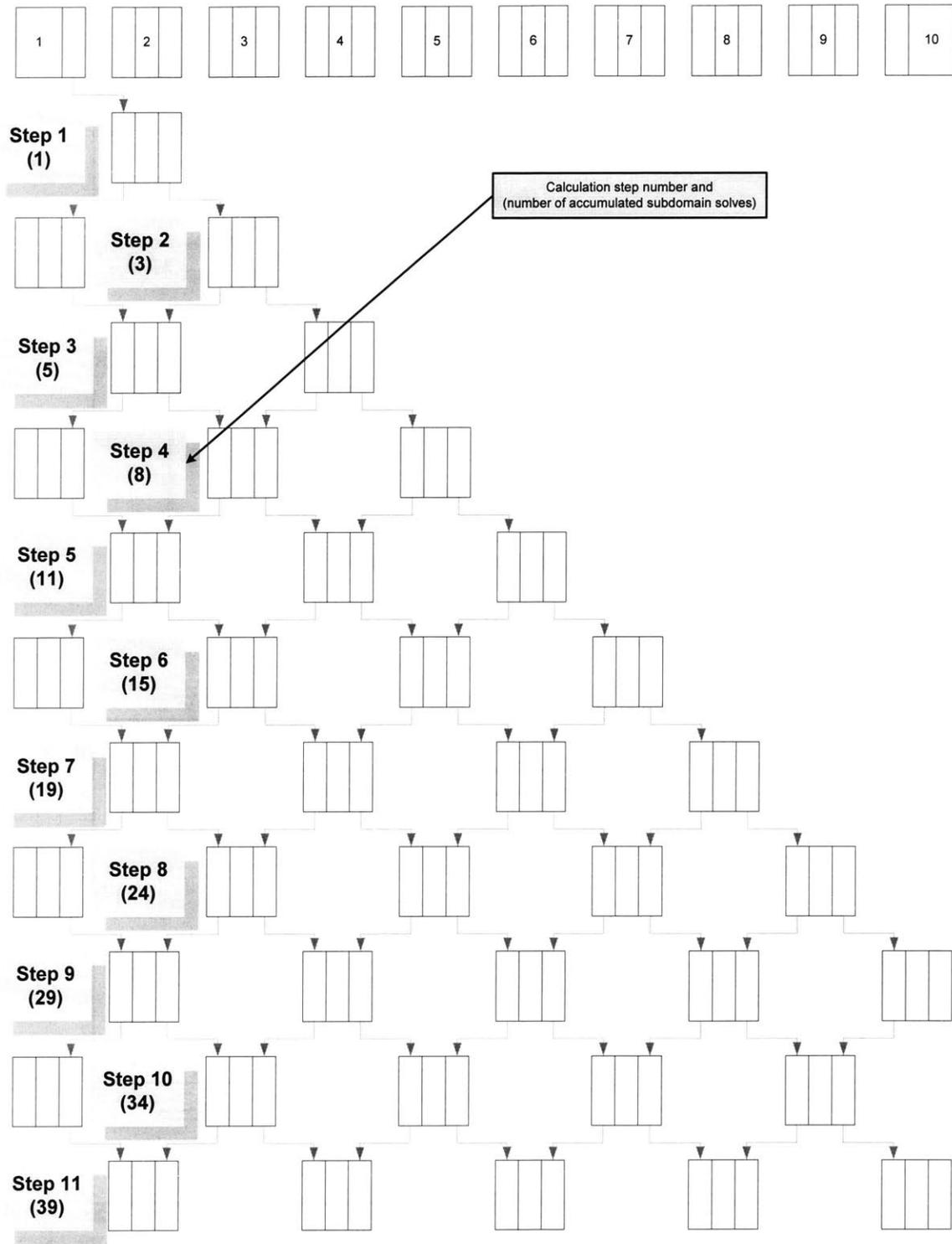


Figure 4-23 Calculation Steps of Implemented Overlapping Domain Decomposition

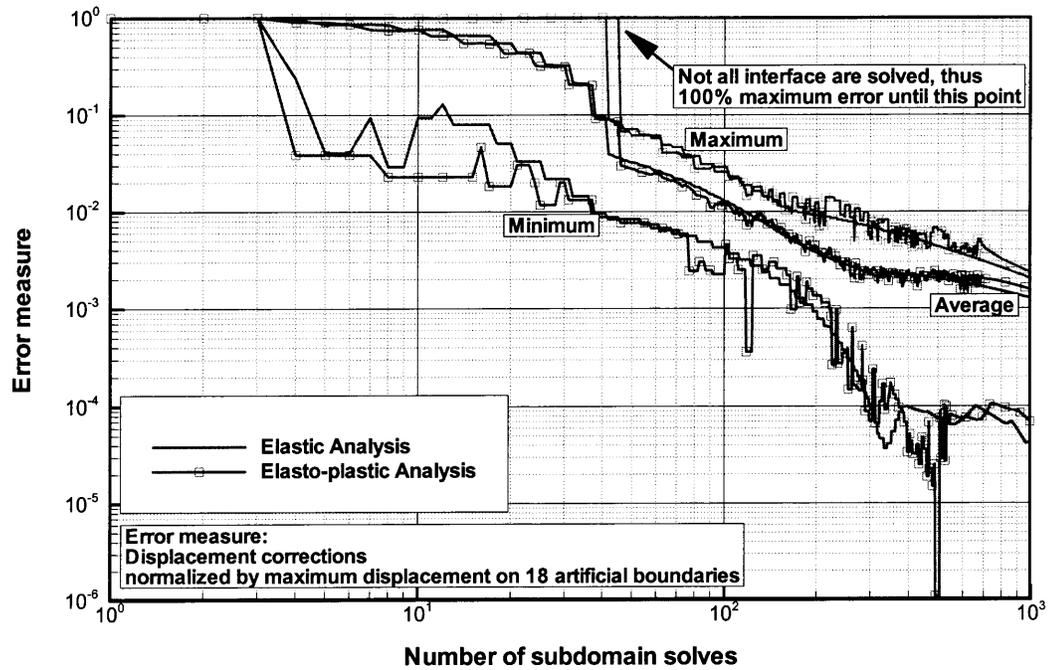


Figure 4-24 Convergence of Overlapping Domain Decomposition

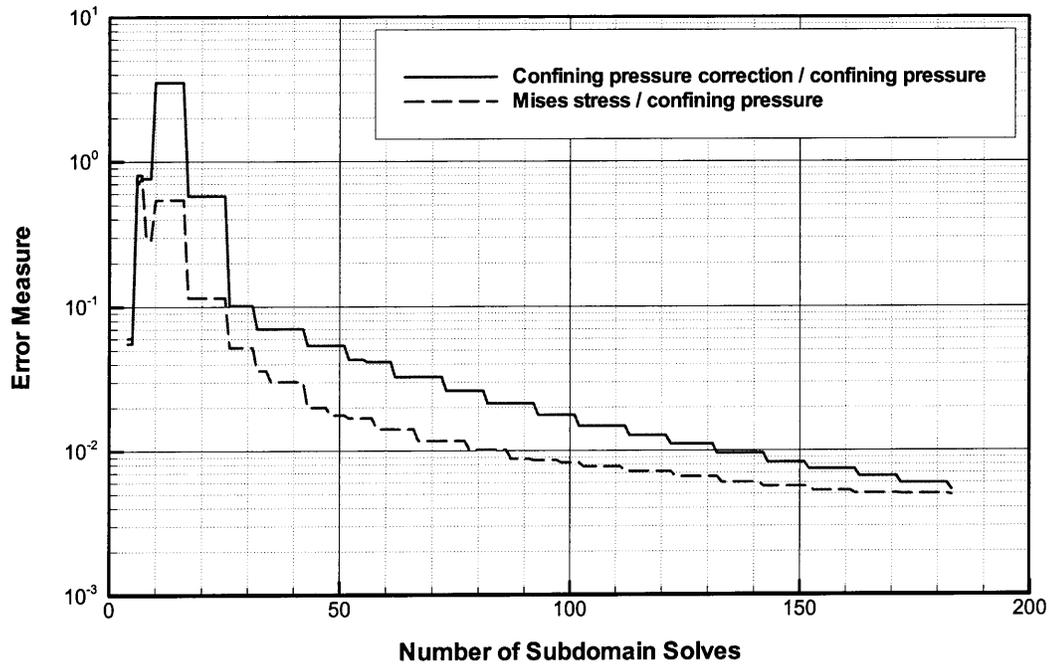


Figure 4-25 Alternative Error Measures for Overlapping Domain Decomposition

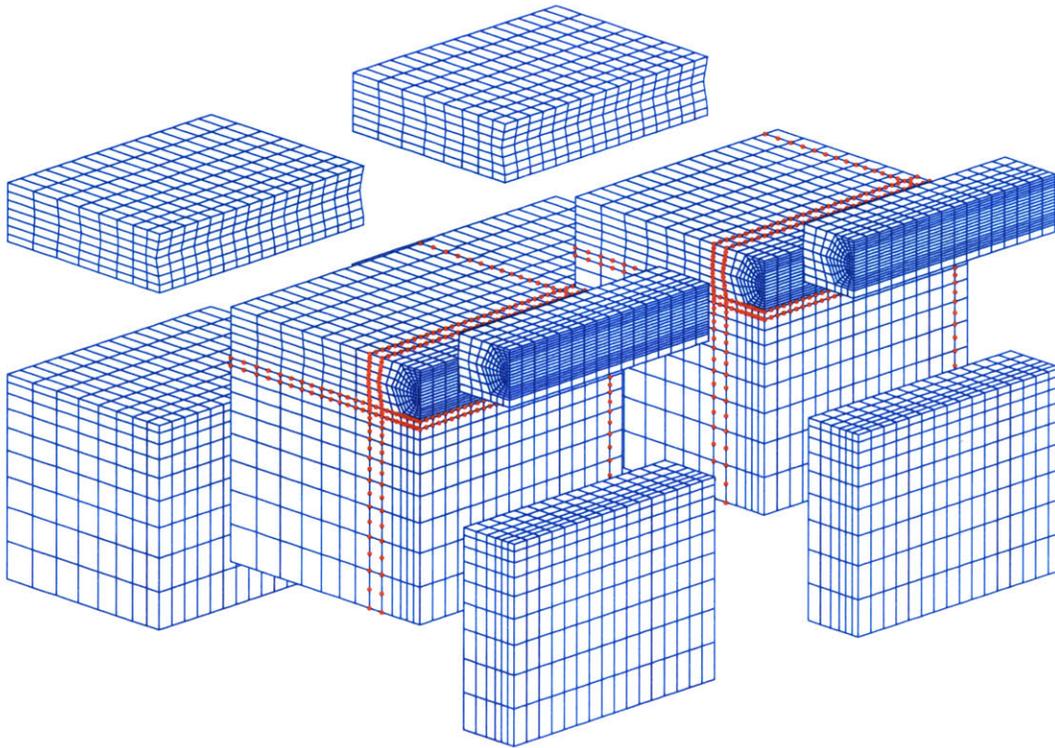


Figure 4-26 Decomposed Finite Element Mesh

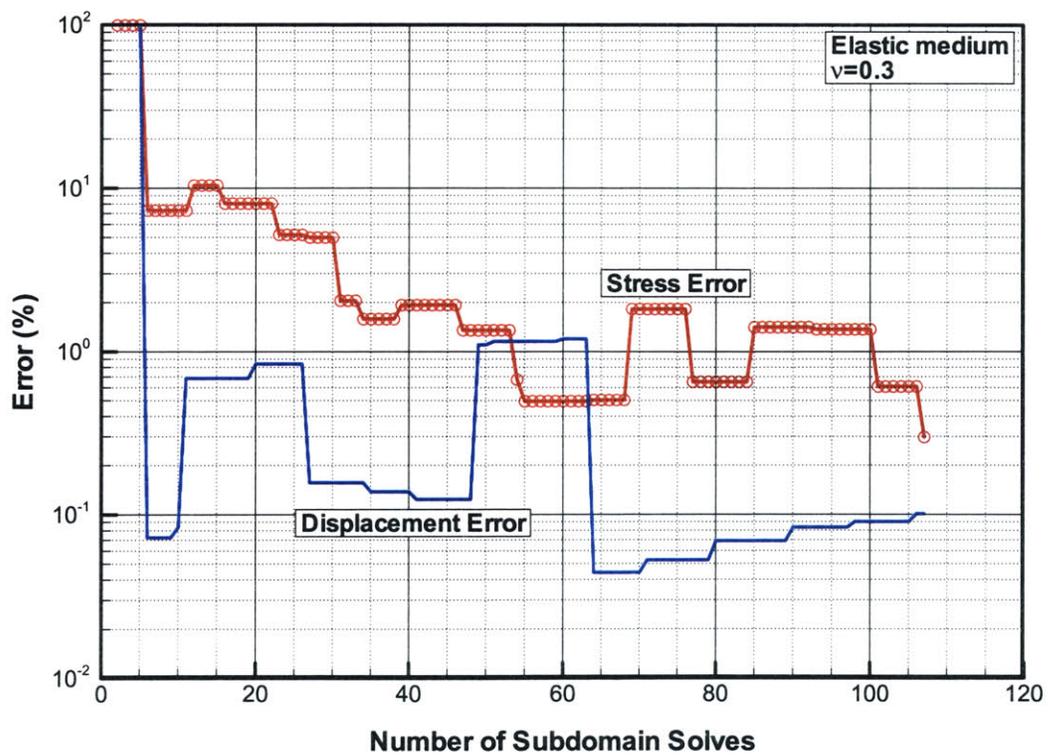
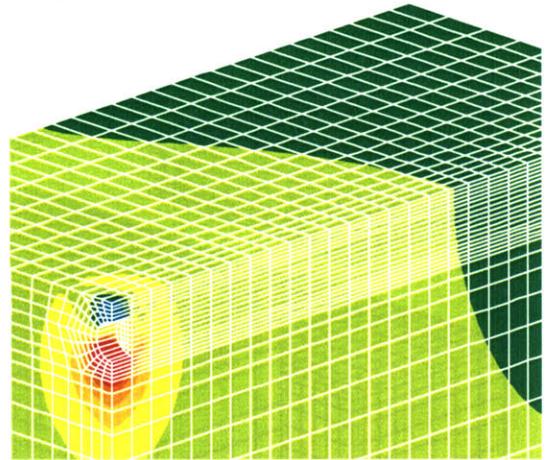
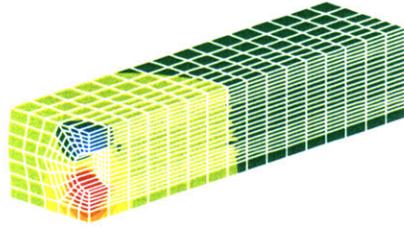


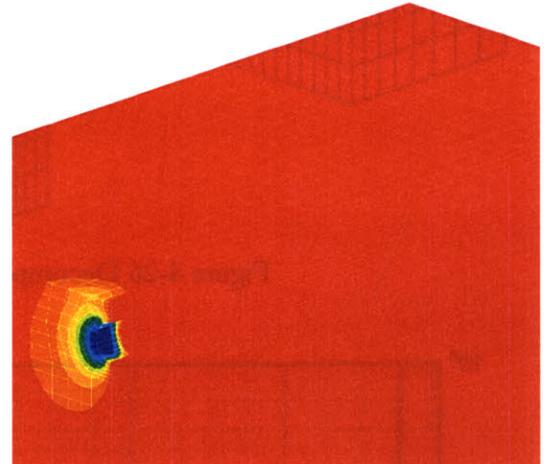
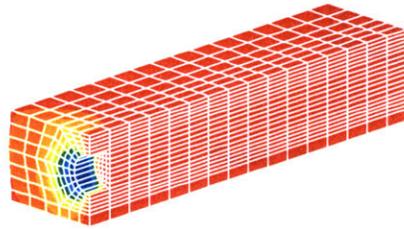
Figure 4-27 Convergence History of Schwarz Alternation for the 8-sub-domain problem

DECOMPOSED SOLUTION

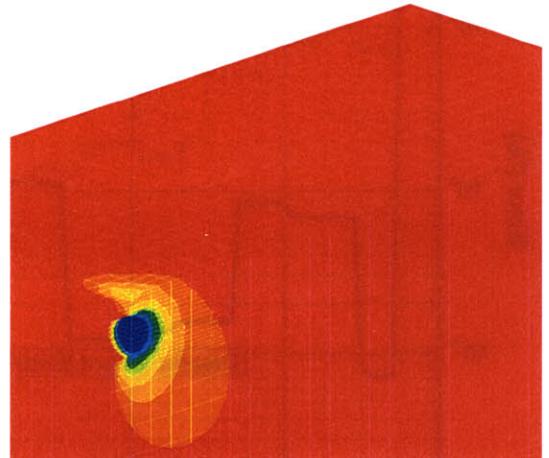
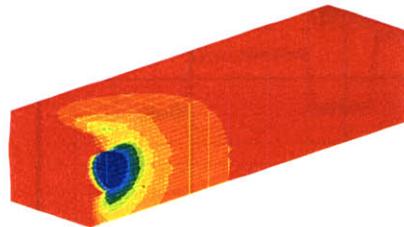
WHOLE FINITE ELEMENT SOLUTION



(a) Vertical displacement contours



(b) Horizontal displacement contours



(c) Tunnel axis displacement contours

Figure 4-28 Comparisons between decomposed problem and full finite element solution

```

*EL FILE, FREQ=1
NFORC,
*NODE FILE, FREQ=1
CF,
*ELEMENT MATRIX OUTPUT, DLOAD=YES, STIFFNESS=YES, FREQ=1, ELSET=GROUND

```

Figure 4-29 Extracting Information from ABAQUS for FETI Algorithm

1. *for* $i = 1 : N_{col}$
2. *if* $(\mathbf{U}_{K^+}(i, i) = 0.0)$
3. $\mathbf{R}_i := -(\mathbf{U}_{K^+}(:, i) + \mathbf{L}_{K^+}(:, i))$
4. $\mathbf{R}_i := \mathbf{L}_{K^+} \setminus \mathbf{R}_i$
5. $\mathbf{U}_{K^+}(:, i) = \mathbf{0}$
6. $\mathbf{L}_{K^+}(:, i) = \mathbf{0}$
6. $\mathbf{U}_{K^+}(i, i) = 1.0$
7. *endif*
8. *endfor*

Figure 4-30 Rigid Body Mode Extraction Algorithm

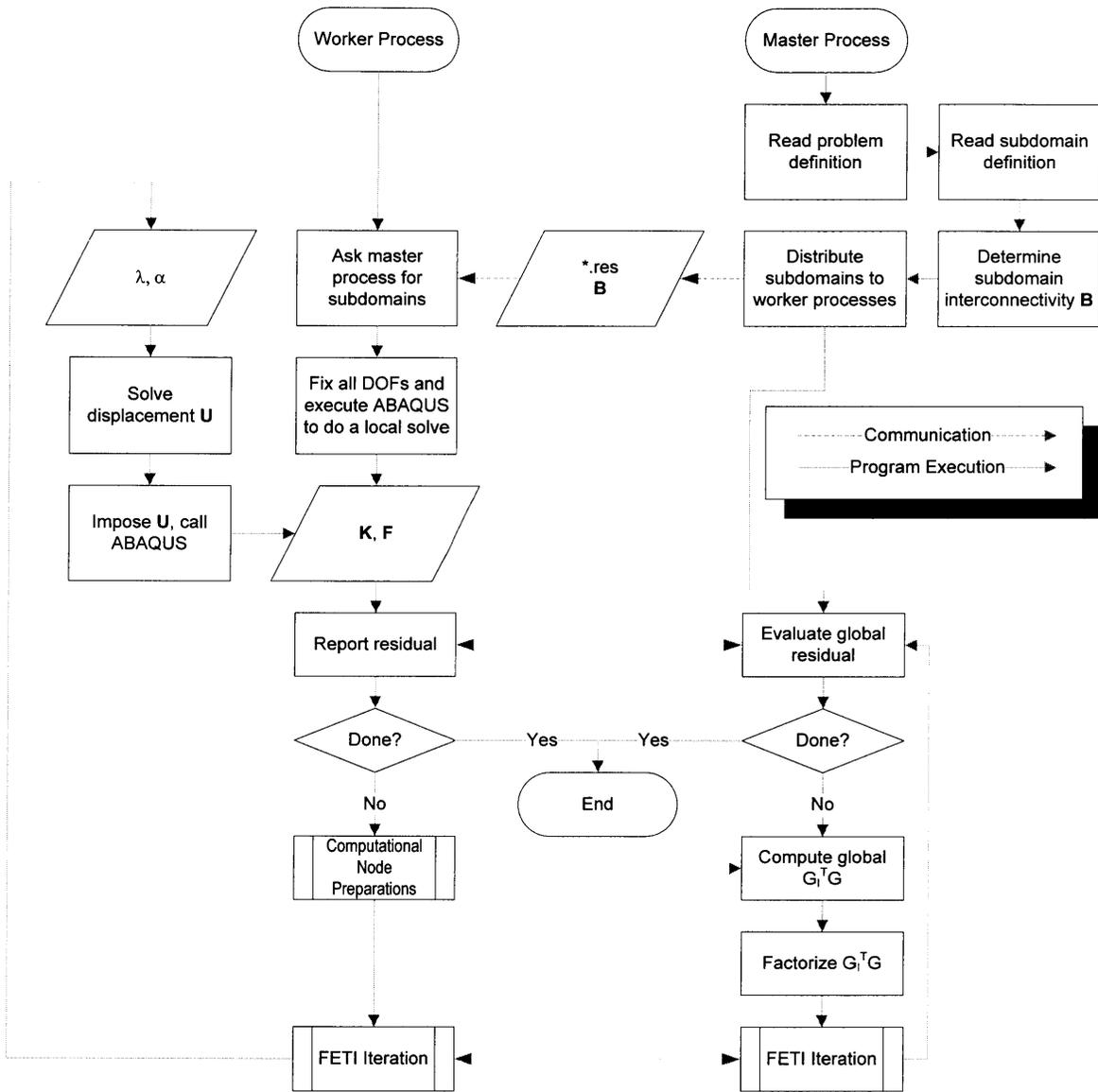


Figure 4-31 Flowchart for *FETI-DD*
 (Please Refer to Section 3.8.2.2)

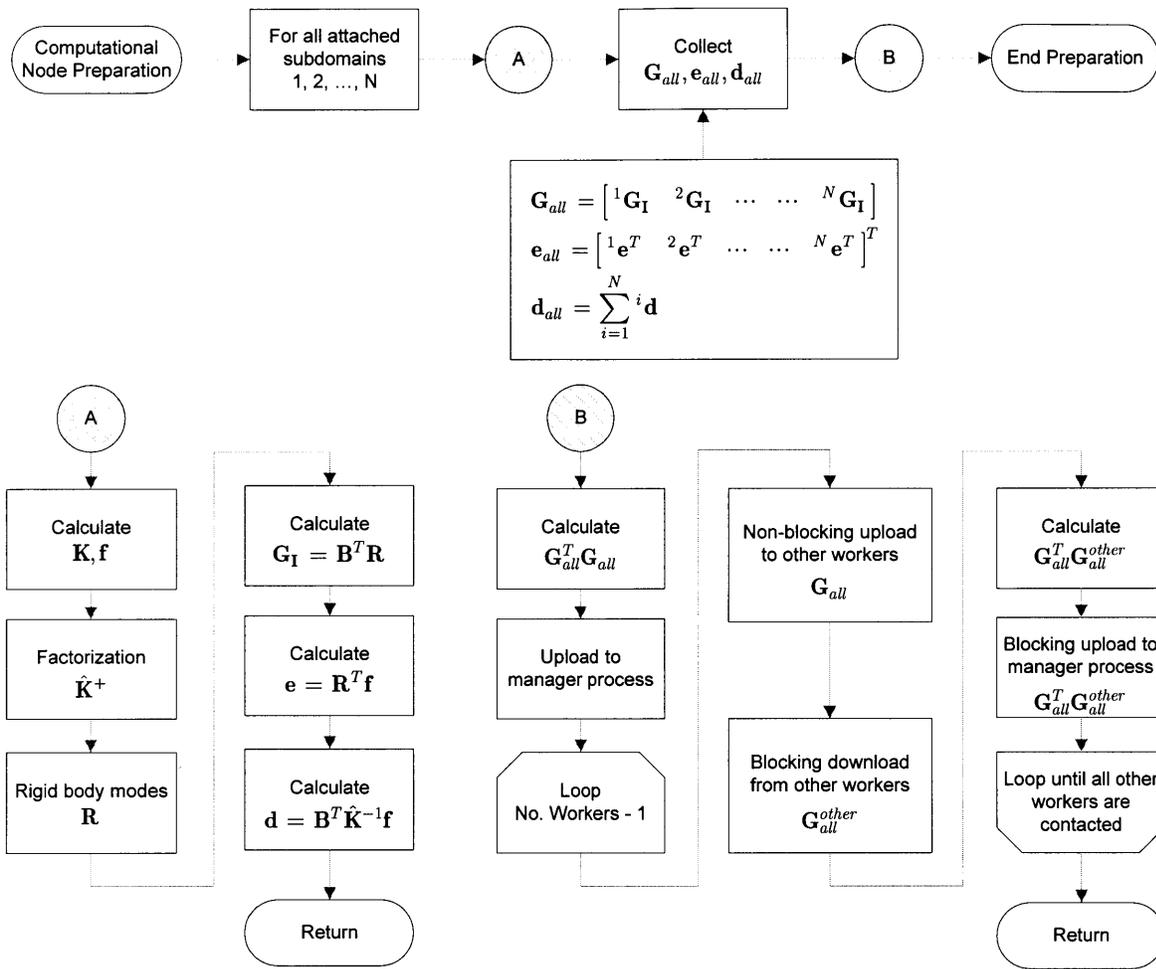


Figure 4-32 Flowchart of Preparations on Worker Processes
(Please Refer to Section 3.8.2.2)

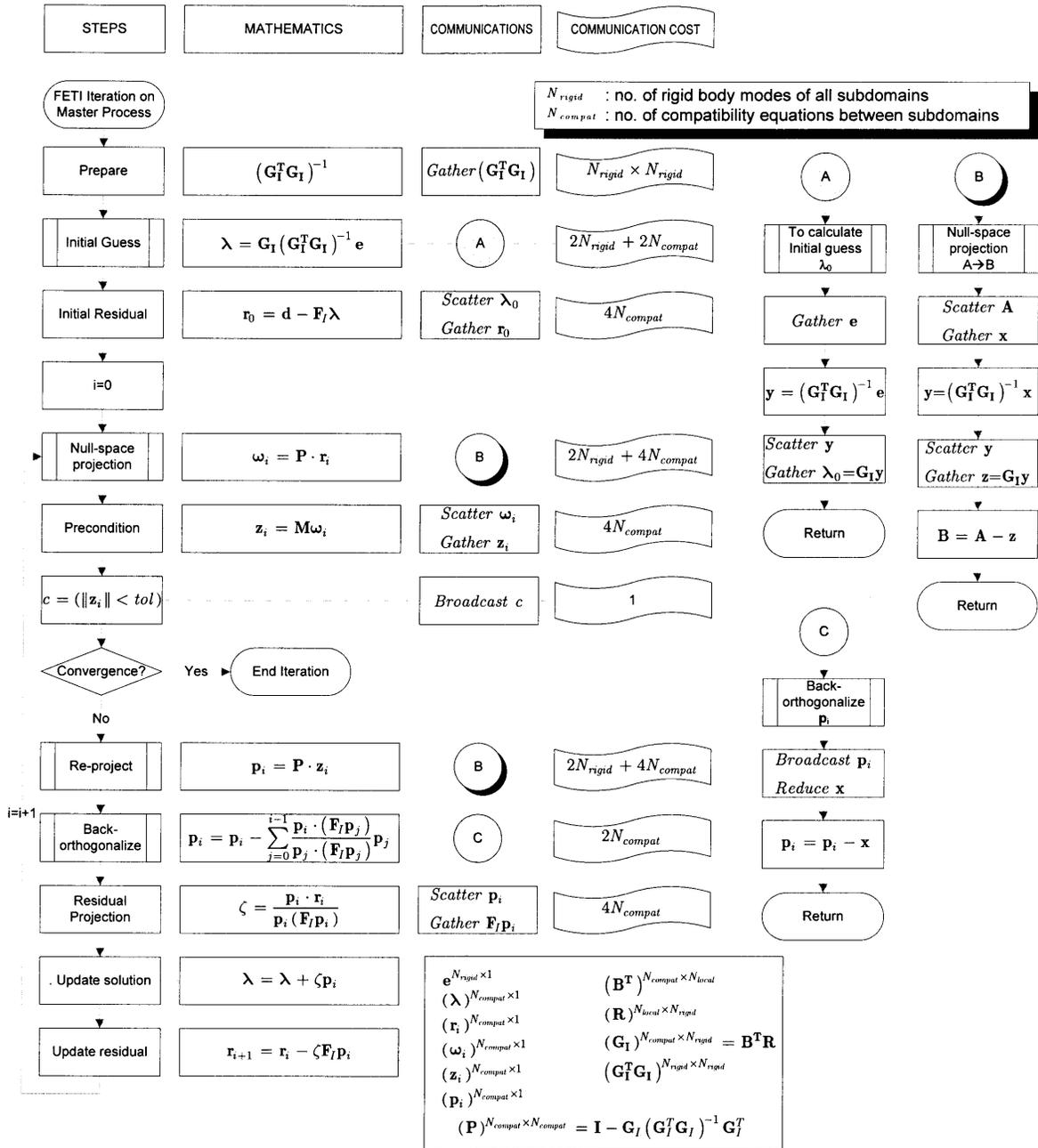
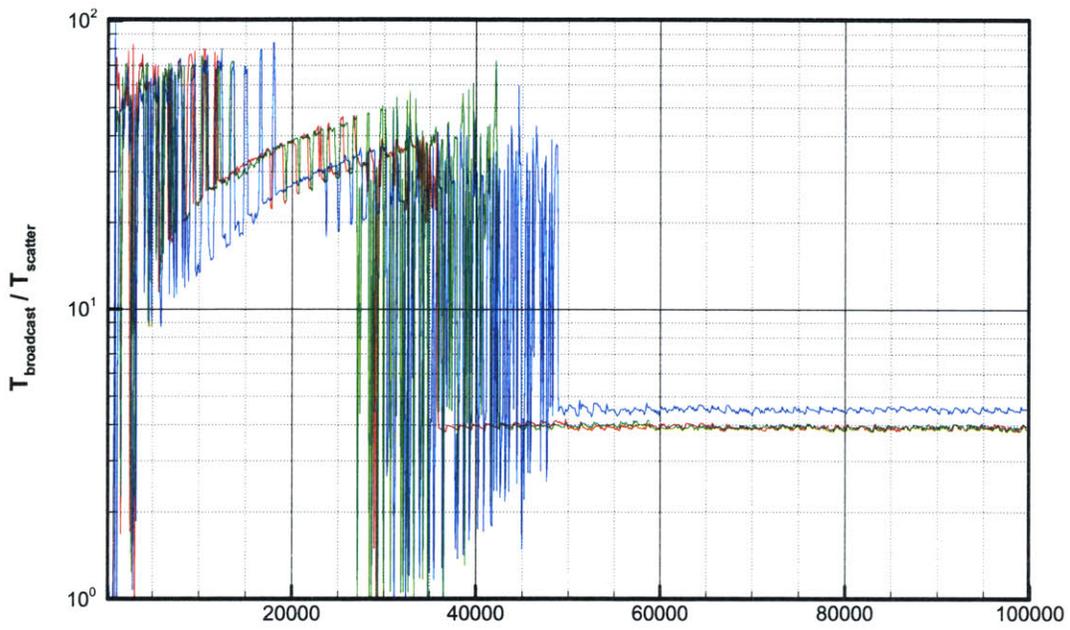
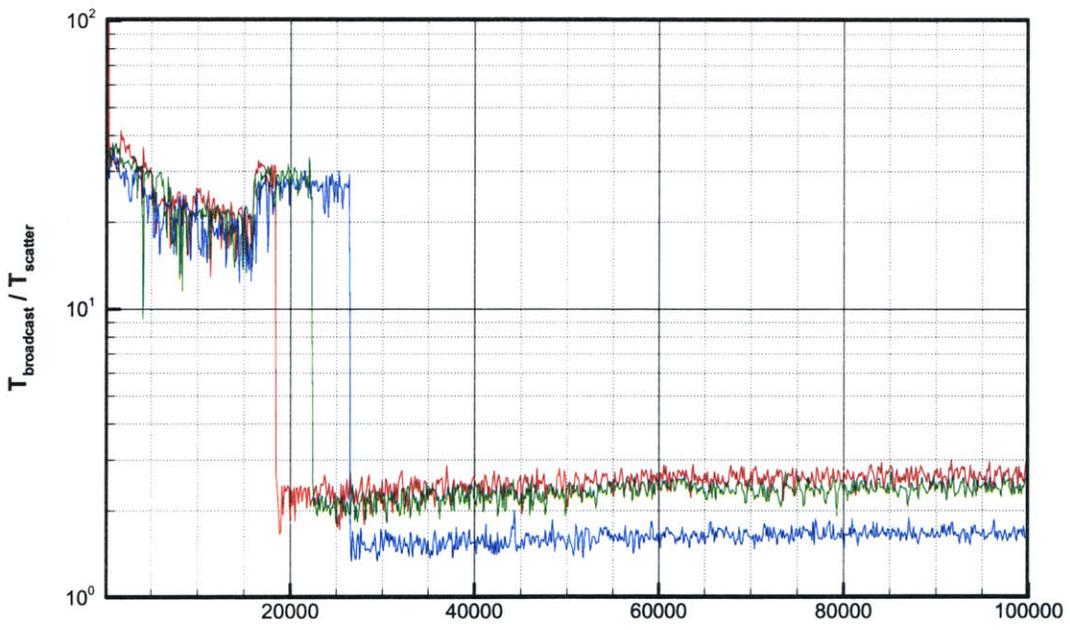


Figure 4-33 Flowchart of FETI Iteration Implementation on the Master Process
(Please Refer to Section 3.8.2.2)



Vector Length (double precision)
 (a) Fast Ethernet, 12, 14, and 16 nodes



Vector Length (double precision)
 (b) Myrinet, 10, 12, and 14 nodes

Figure 4-34 Speed Differences between Scatter and Broadcast Operations

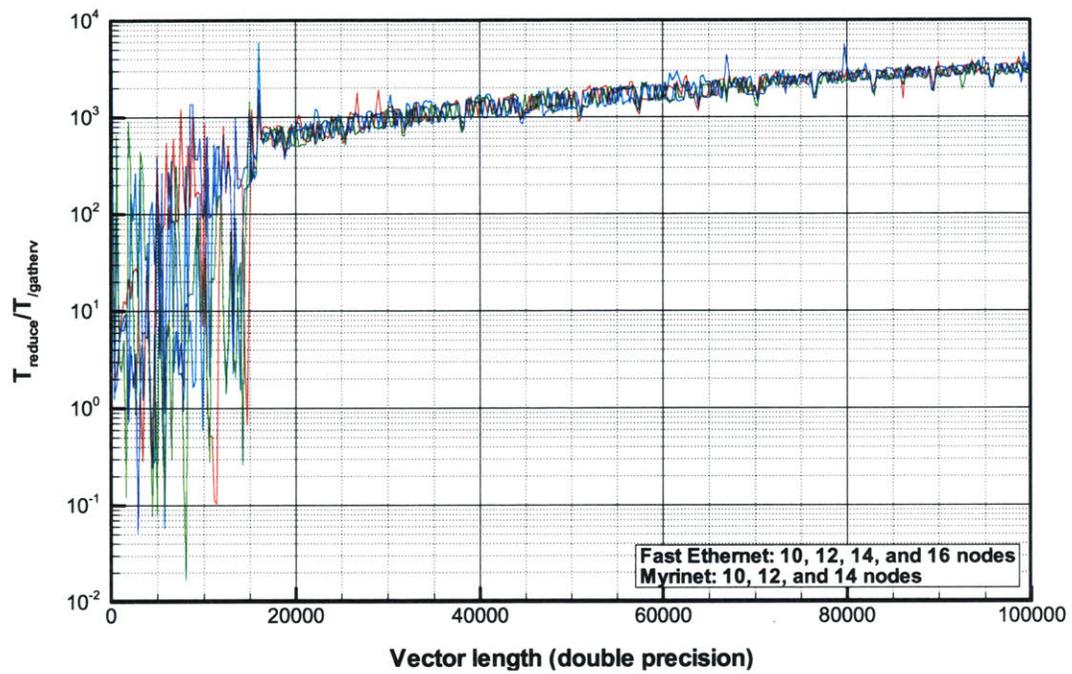


Figure 4-35 Speed Differences between Gather and Reduction Operations

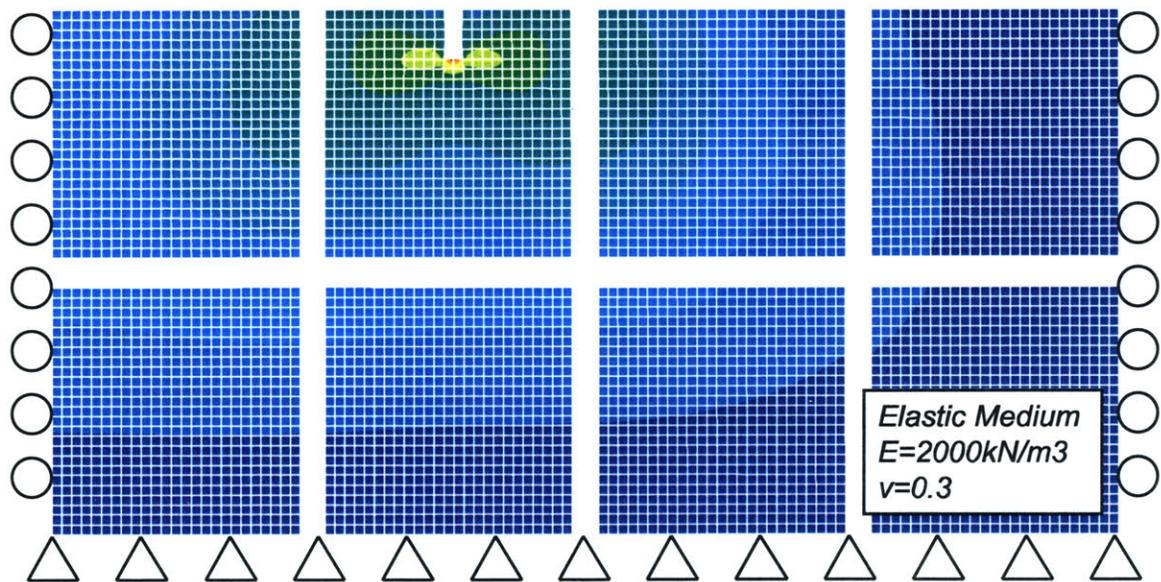
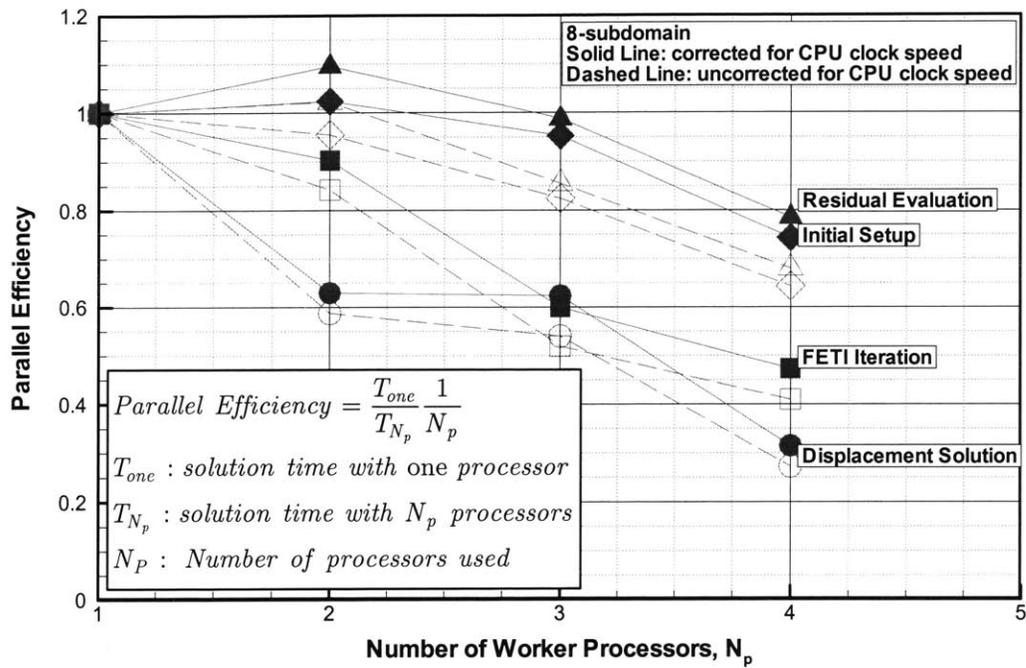
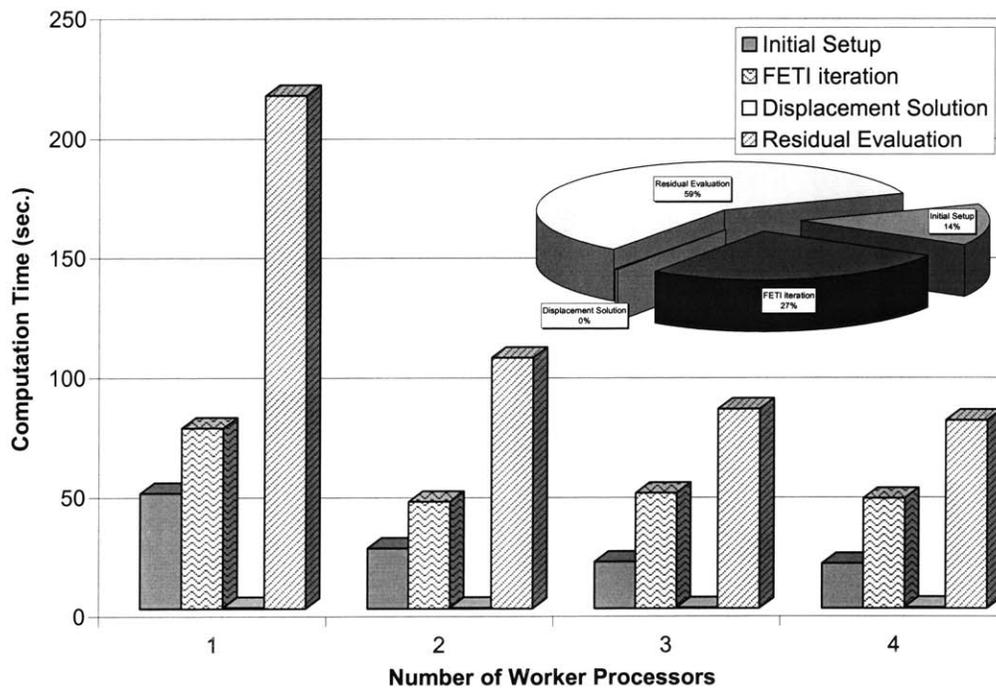


Figure 4-36 Mesh Decomposition for Preliminary Performance Evaluation for FETI-DD Algorithm with ABAQUS

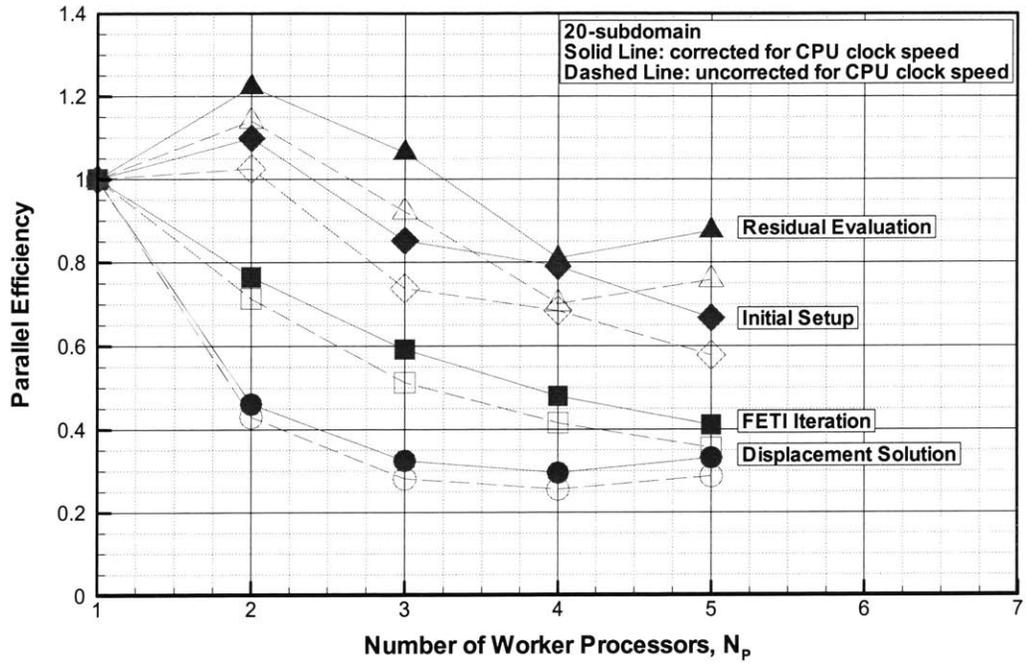


(a) Parallel efficiency

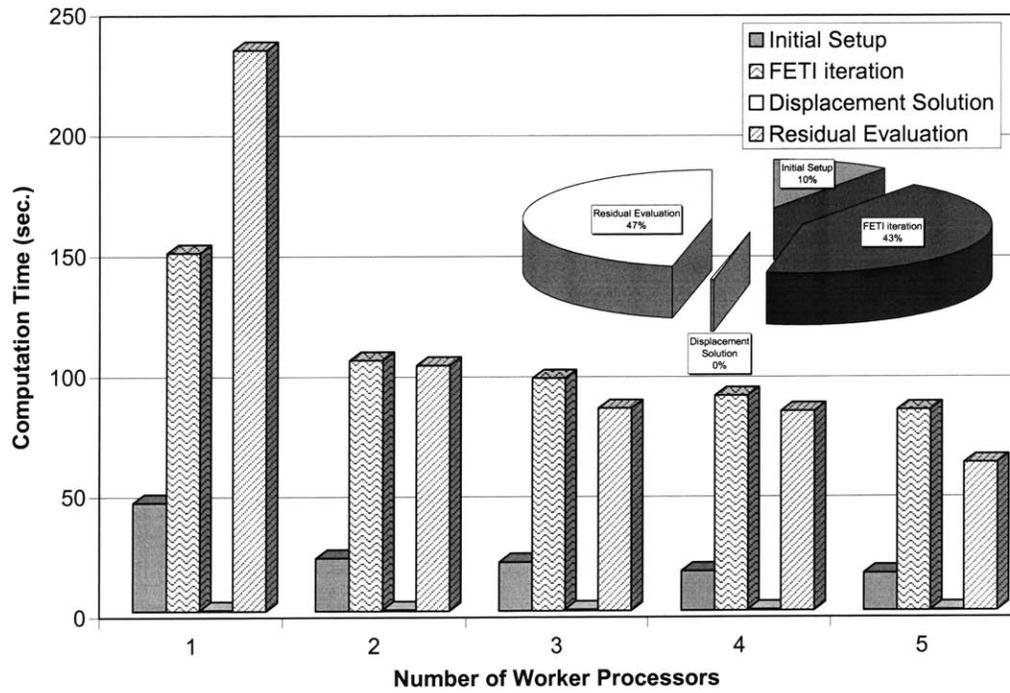


(b) Calculation time

Figure 4-37 Parallel Efficiency for 8-Sub-domain Calculation Using Shared Fast Ethernet



(a) Parallel efficiency



(b) Calculation time

Figure 4-38 Parallel Efficiency for 20-Sub-domain Calculation Using Shared Fast Ethernet



Figure 4-39 Input Files for FETI_FEM

```

*NODE, NSET=NALL
2, 0.0, 1.0, 0.0,
5, 1.0, 1.0, 0.0,
6, 1.0, 2.0, 0.0,
3, 0.0, 2.0, 0.0,
52, 0.5, 1.0, 0.0,
65, 1.0, 1.5, 0.0,
63, 0.5, 2.0, 0.0,
32, 0.0, 1.5, 0.0,
*ELEMENT, TYPE=CPE8, ELSET=CBEAM
2, 2, 5, 6, 3, 52, 65, 63, 32,
*NSET, NSET=SUPPORT
2, 3, 32,
*SOLID SECTION, ELSET=CBEAM, MATERIAL=ELAS
1.0,
*MATERIAL, NAME=ELAS
*ELASTIC
20.0, 0.3,
*DENSITY
1.0,
*RESTART, WRITE
*STEP
*STATIC
*DLOAD
CBEAM, GRAV, 1.0, 0.0, -1.0, 0.0,
*END STEP

```

Figure 4-40 Example Finite Element Definition File for Sub-domains

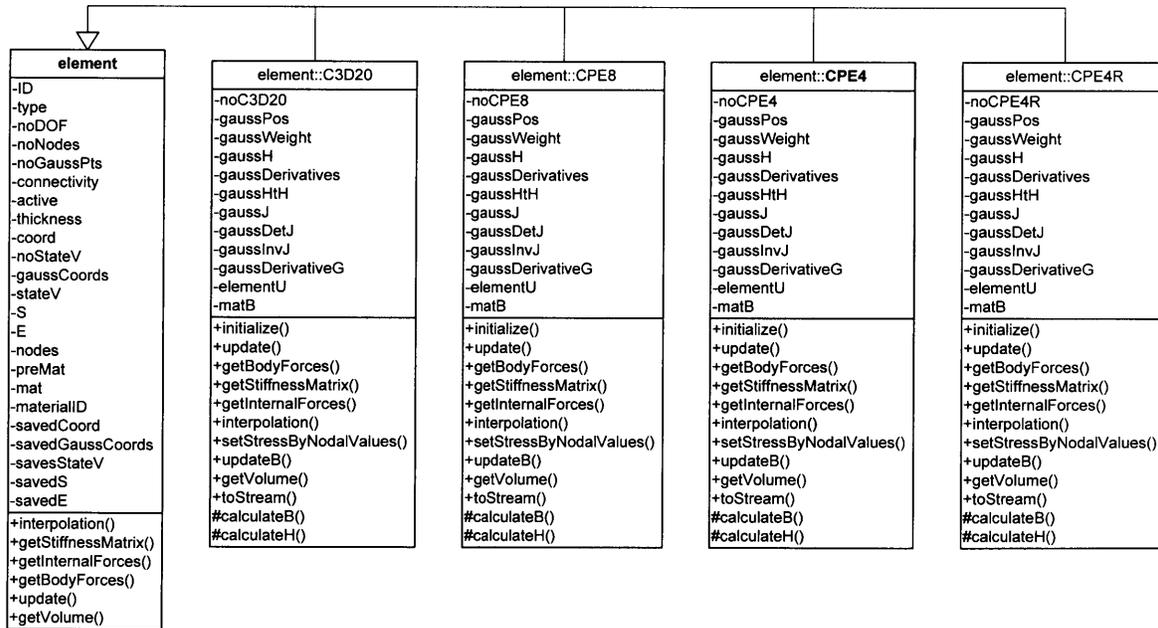


Figure 4-41 Simplified Class Diagram for Element Classes

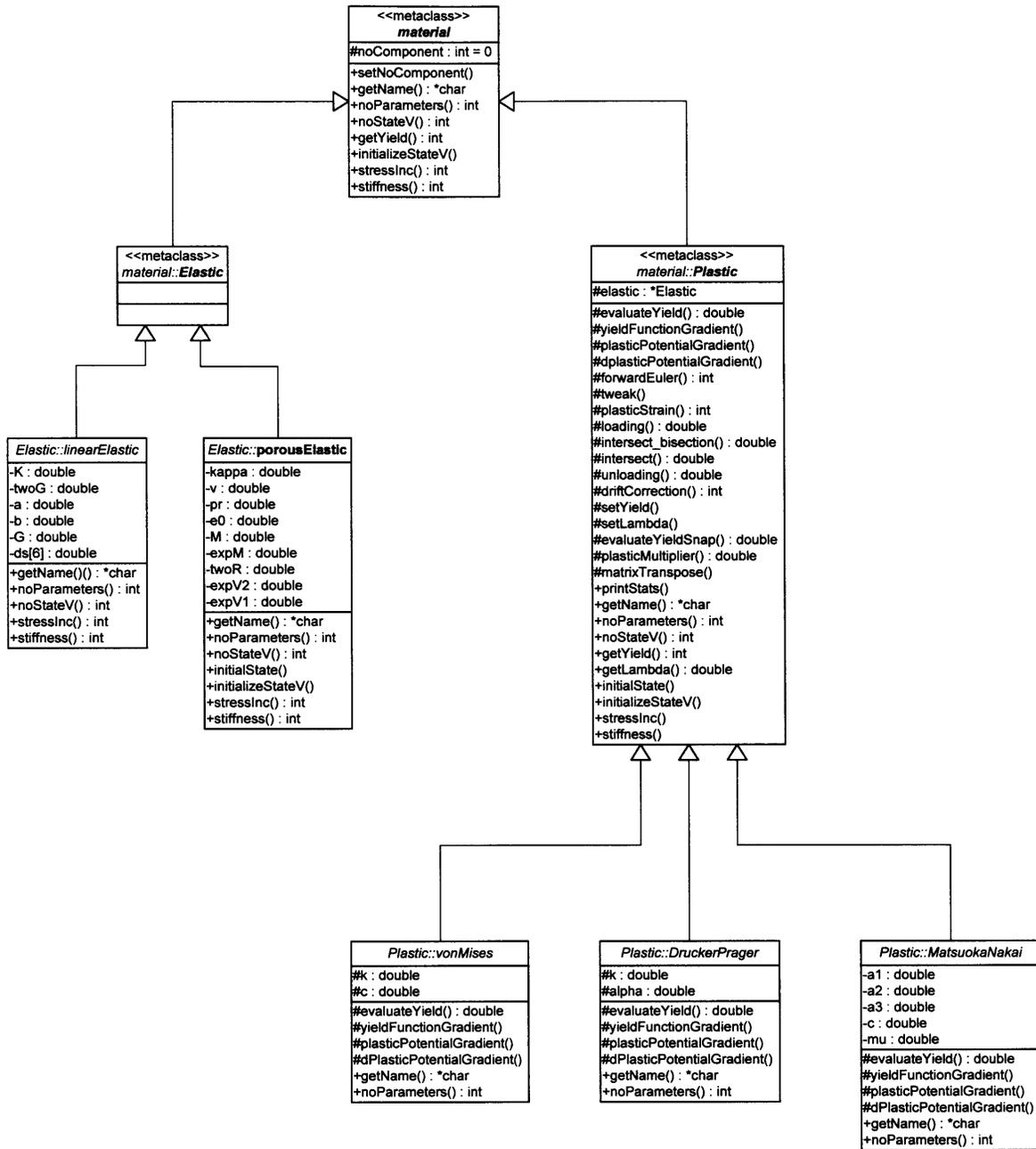
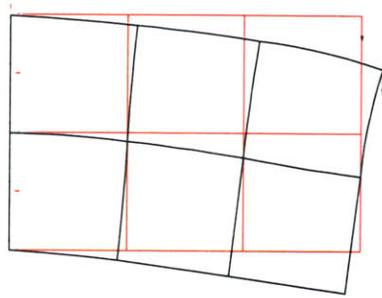
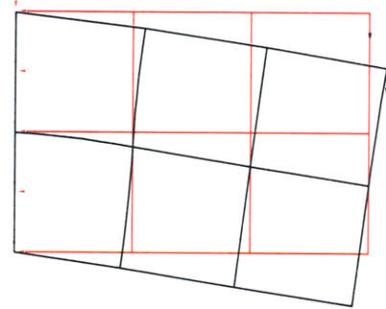


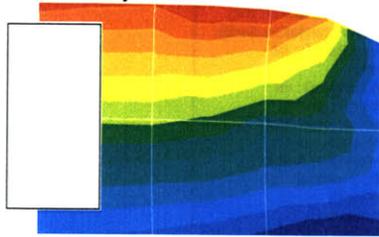
Figure 4-42 Simplified Class Diagram for Material Classes



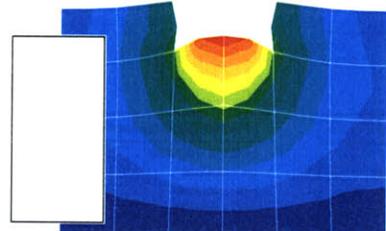
Test0: boundary conditions and *CLOAD



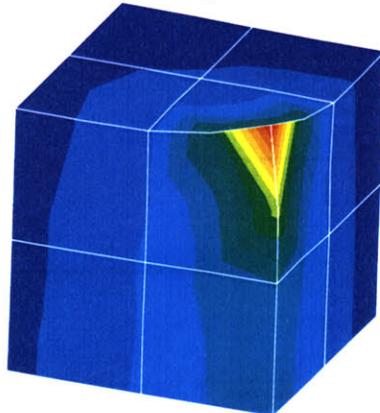
Test1: *DLOAD



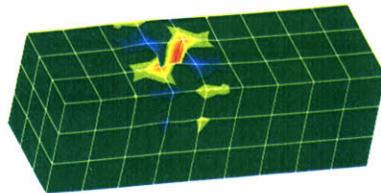
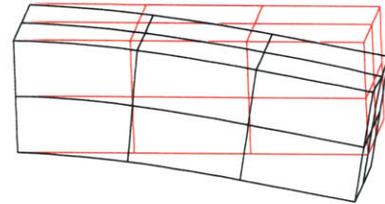
Test2: *INITIAL CONDITIONS, TYPE=STRESS, GEOSTATIC



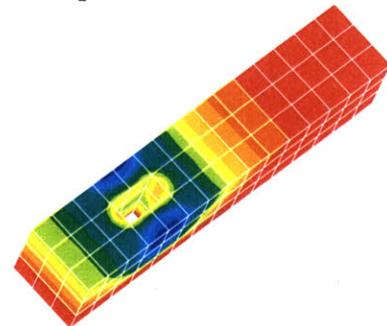
Test3-5: *MODEL CHANGE, REMOVE; multiple steps, multiple elements/domain



Test6: Test 3-D analyses, and C3D20 implementation



Test7: Test 3-D excavation, and change of interconnectivity between sub-domains



Test8: Test 3-D excavation, and change of boundary conditions

Figure 4-43 Initial Validation Problems

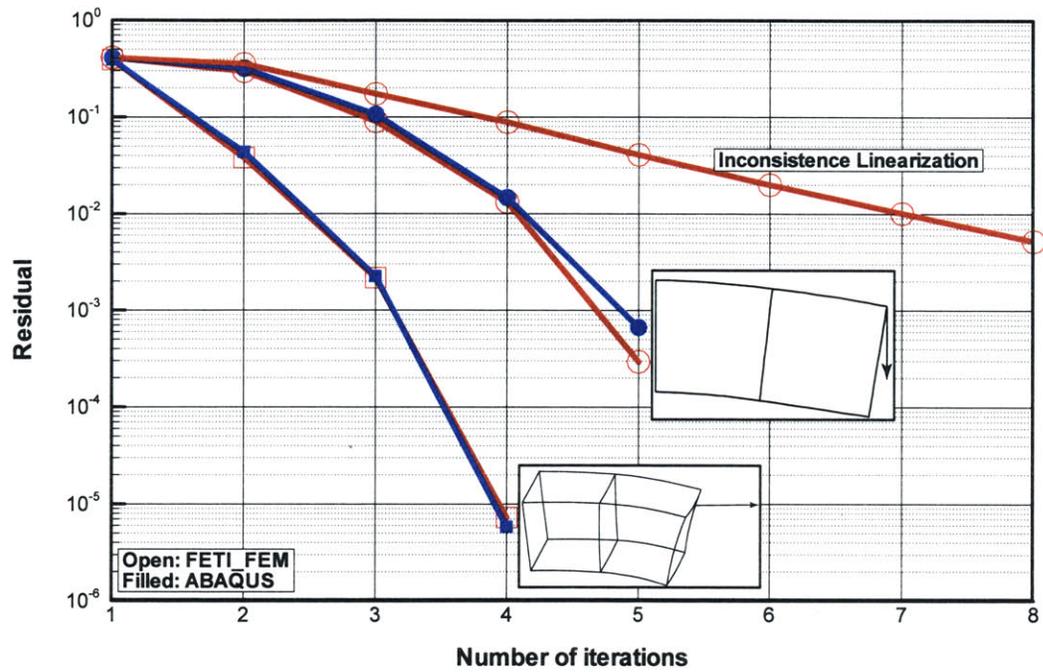


Figure 4-44 Convergence History of Two Test Problems with von Mises yield criterion

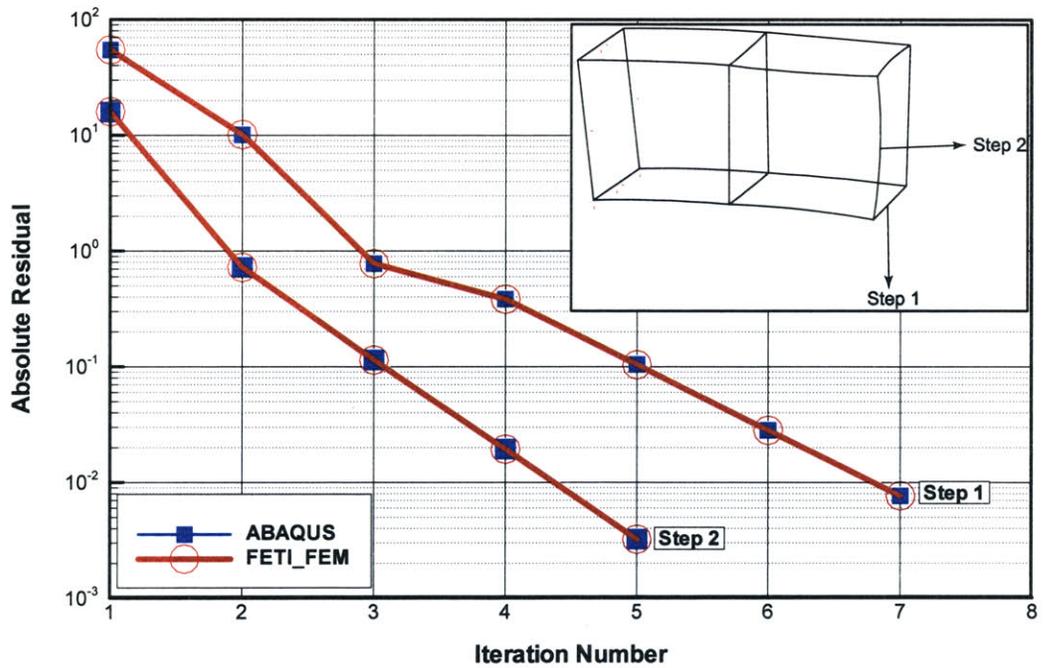


Figure 4-45 Convergence History of a Test Problem with Porous Elasticity

```

*Layers = 20 ; defines number of layers
*Length = 48 ; defines the total depth of extrusion
; therefore, each layer is Length /Layers thickness

*noSteps = 1 ; defines how many empty steps are to be generated
; in the generated input files

! FE mesh numbering
*nodeInc = 5000 ; defines the increment of node numbers between
; layers of nodes generated
; in the extrusion direction
*elementInc = 3000 ; defines the increment of element number between
; layers of elements generated

! Boundary conditions ; defining global boundary conditions
*noBoundaries = 5 ; number of boundary conditions to be define
*Boundary = 1, 0.0, 1, 1 ; *Boundary=[component spec .], [coord.], [DOF1], [DOF2]
; [component spec .]: 1-3, define which component the [coord.] is
*Boundary = 1, 48.0, 1, 1 ; [coord.]: the value of coordinate of be specifying BCs
*Boundary = 2, -40.0, 2, 2 ; [DOF1]: the first DOF to be fixed
*Boundary = 2, 40.0, 2, 2 ; [DOF2]: the second DOF to be fixed
*Boundary = 3, -30.0, 1, 3

*noDomains = 12 ; defines number of 2D meshes to extrude
*Domain = D01.INP, 3 ; *Domain=[filename ], [noLayers ]
; [filename]: the filename contains 2D mesh to extrude
*Domain = D02.INP, 3 ; [noLayers]: number of layers for each subdomain
*Domain = D03.INP, 3 ; therefore, there will be 20/3 ~ 7 subdomains in 3D
*Domain = D04.INP, 3 ; for each extrude 2D mesh
*Domain = D05.INP, 3
*Domain = D06.INP, 3
*Domain = D07.INP, 3
*Domain = D08.INP, 3
*Domain = D09.INP, 3
*Domain = D10.INP, 3
*Domain = D11.INP, 3
*Domain = D12.INP, 3

```

Figure 4-46 Input file EXTRUDE.INI for EXTRUDE Performing Uniform Extrusion

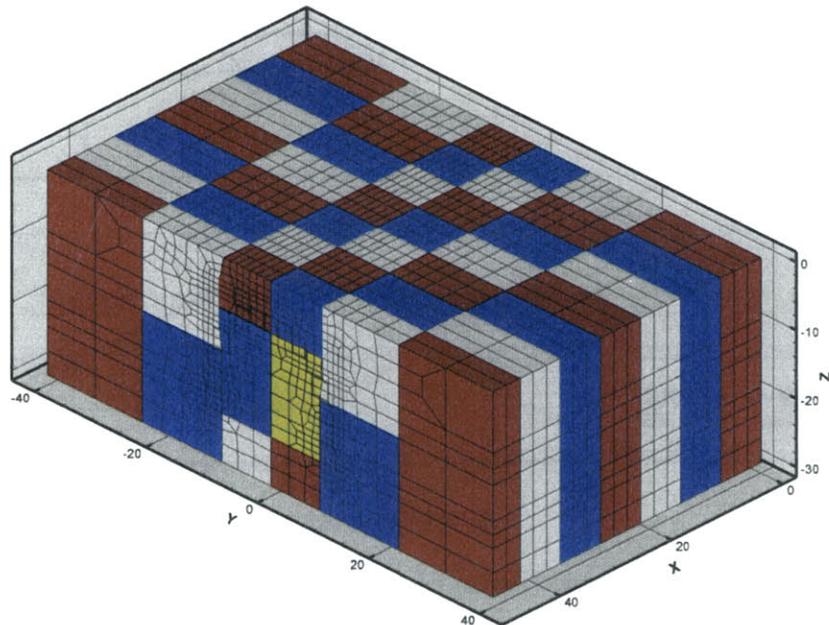


Figure 4-47 A Uniformly Extruded Mesh

```

*Layers = 20 ; defines number of layers
*Length = 48 ; defines the total depth of extrusion
; therefore , each layer is Length /Layers thickness

*noSteps = 1 ; defines how many empty steps are to be generated
; in the generated input files

' FE mesh numbering
*nodeInc = 5000 ; defines the increment of node numbers between
; layers of nodes generated
; in the extrusion direction
*elementInc = 3000 ; defines the increment of element number between
; layers of elements generated

' Boundary conditions ; defining global boundary conditions
*noBoundaries = 5 ; number of boundary conditions to be define
*Boundary = 1, 0.0, 1, 1 ; *Boundary=[component spec .], [coord.], [DOF1], [DOF2]
; [component spec .]: 1-3, define which component the [coord.] is
*Boundary = 1, 48.0, 1, 1 ; [coord.]: the value of coordinate of be specifying BCs
*Boundary = 2, -40.0, 2, 2 ; [DOF1]: the first DOF to be fixed
*Boundary = 2, 40.0, 2, 2 ; [DOF2]: the second DOF to be fixed
*Boundary = 3, -30.0, 1, 3

*noDomains = 12 ; defines number of 2D meshes to extrude
*Dmain = D01.INP, 6 ; *Domain=[filename ], [noLayers ]
; [filename ]: the filename contains 2D mesh to extrude
*Domain = D02.INP, 3 ; [noLayers ]: number of layers for each subdomain
*Domain = D03.INP, 3 ; therefore , there will be 20/3 ~ 7 subdomains in 3D
*Domain = D04.INP, 3 ; for each extrude 2D mesh
*Domain = D05.INP, 3
*Dmain = D06.INP, 6
*Domain = D07.INP, 3
*Domain = D08.INP, 3
*Dmain = D09.INP, 5
*Domain = D10.INP, 3
*Domain = D11.INP, 3
*Dmain = D12.INP, 5

```

Figure 4-48 Non-Uniform Extrusion Input File

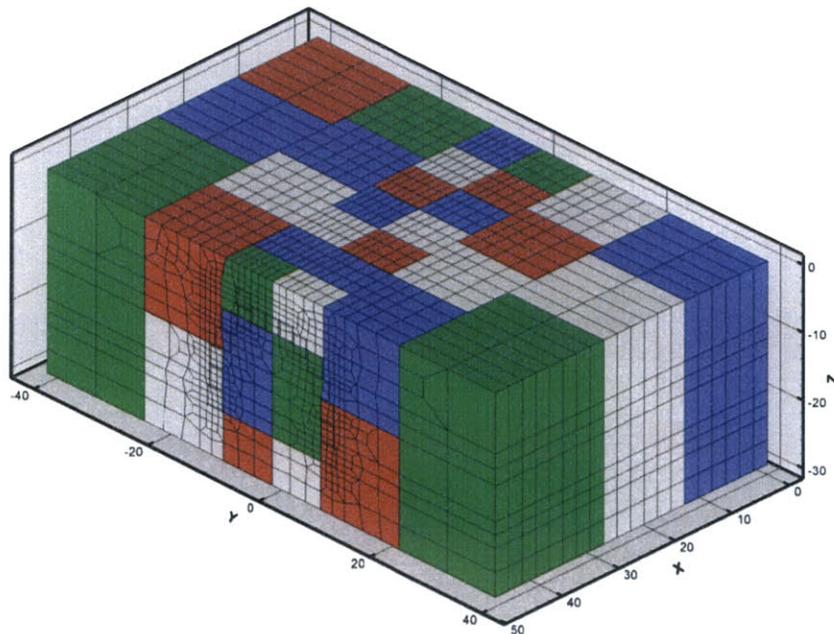


Figure 4-49 A Non-uniformly Extruded Mesh

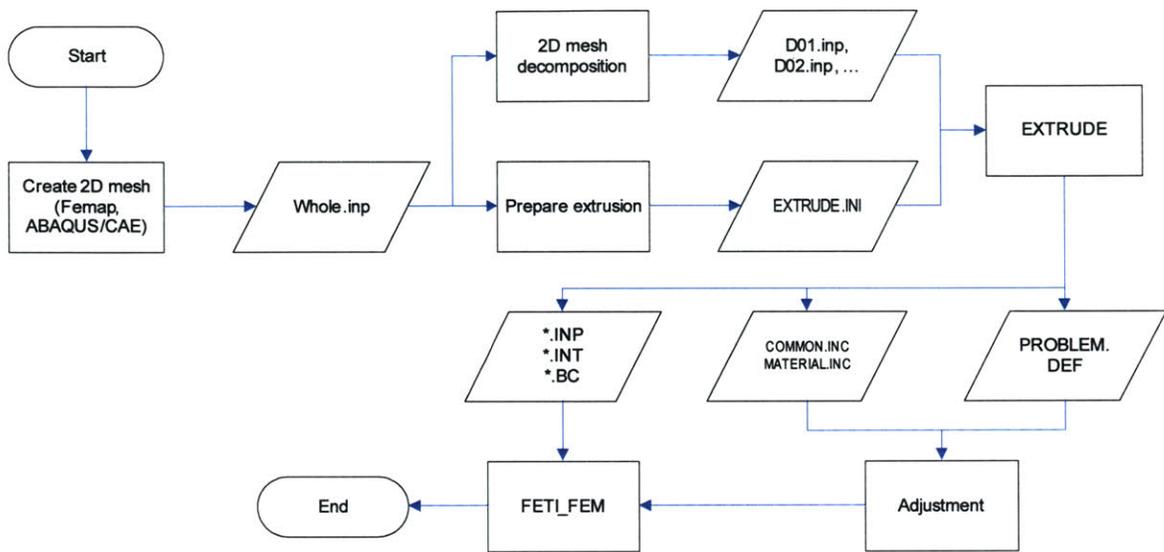


Figure 4-50 Flowchart from 2-D to 3-D Domain Decomposed Analyses

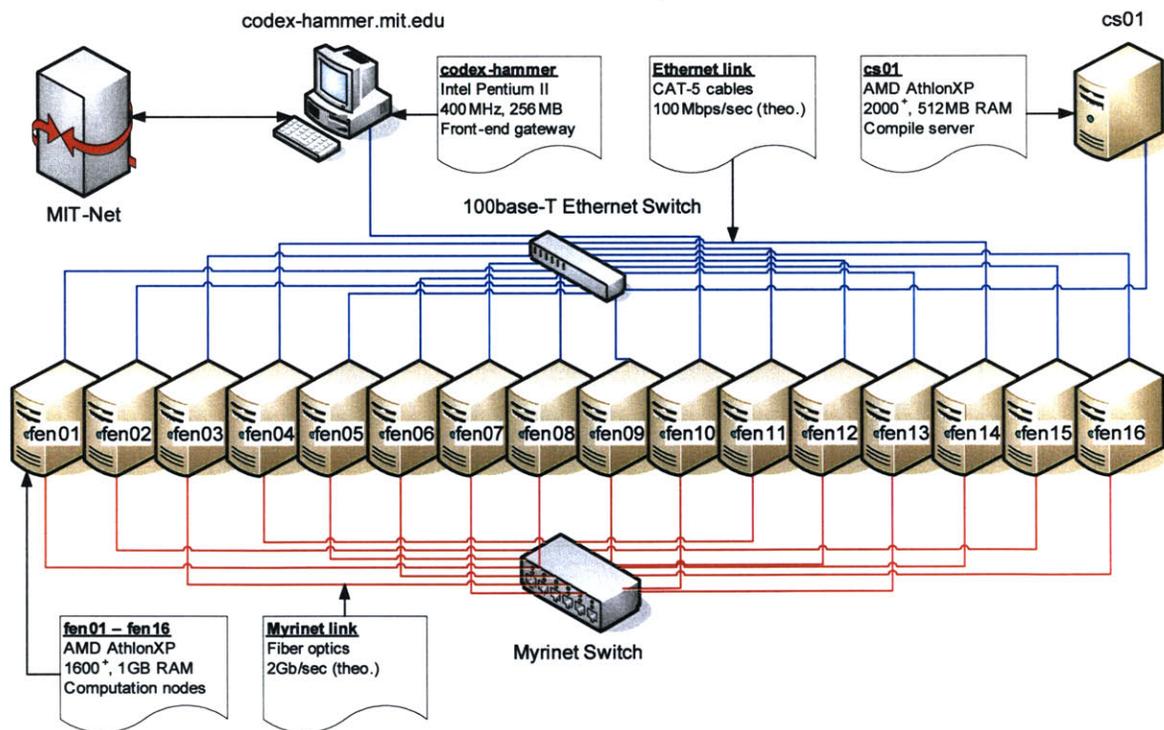
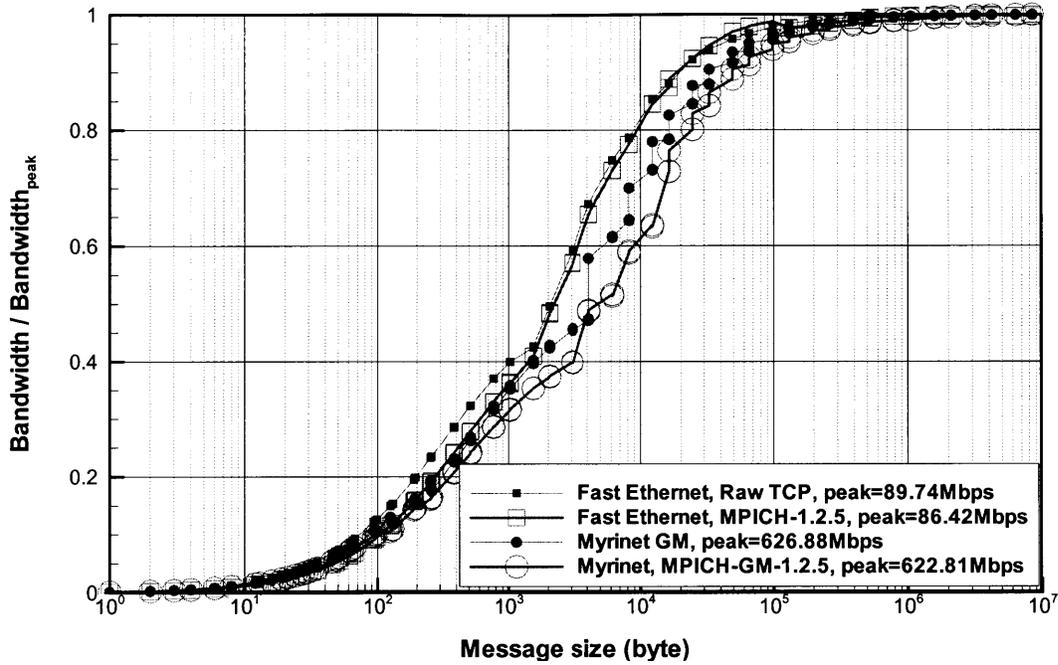
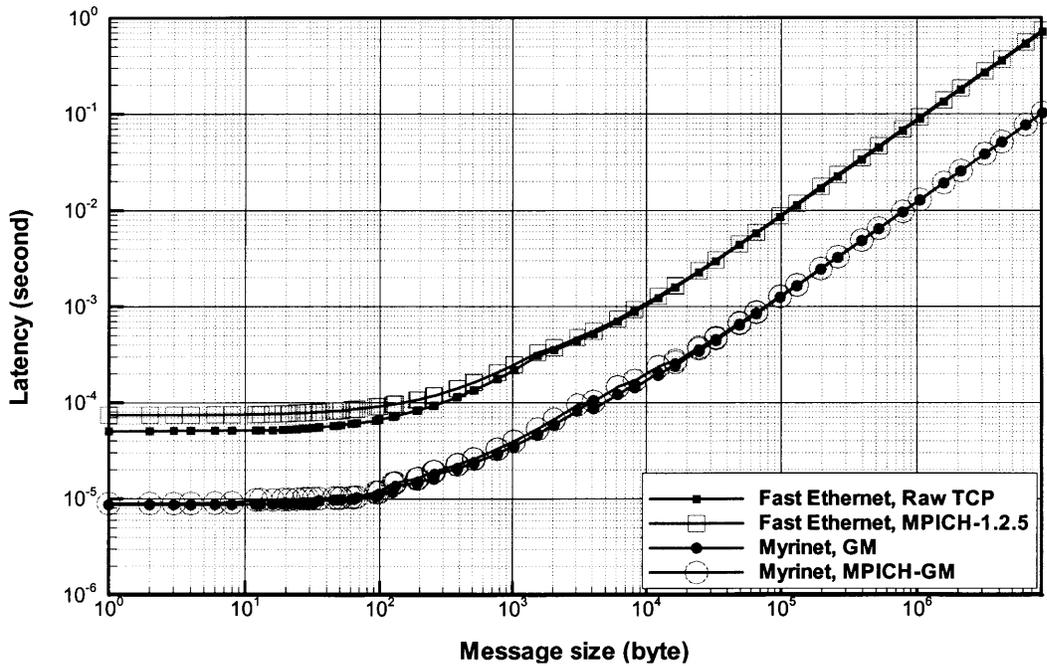


Figure 4-51 Hardware Setup of Codex-hammer



(a) Bandwidth



(b) Latency

Figure 4-52 NetPIPE Performance on Fast Ethernet and on Myrinet in **Codex-hammer**

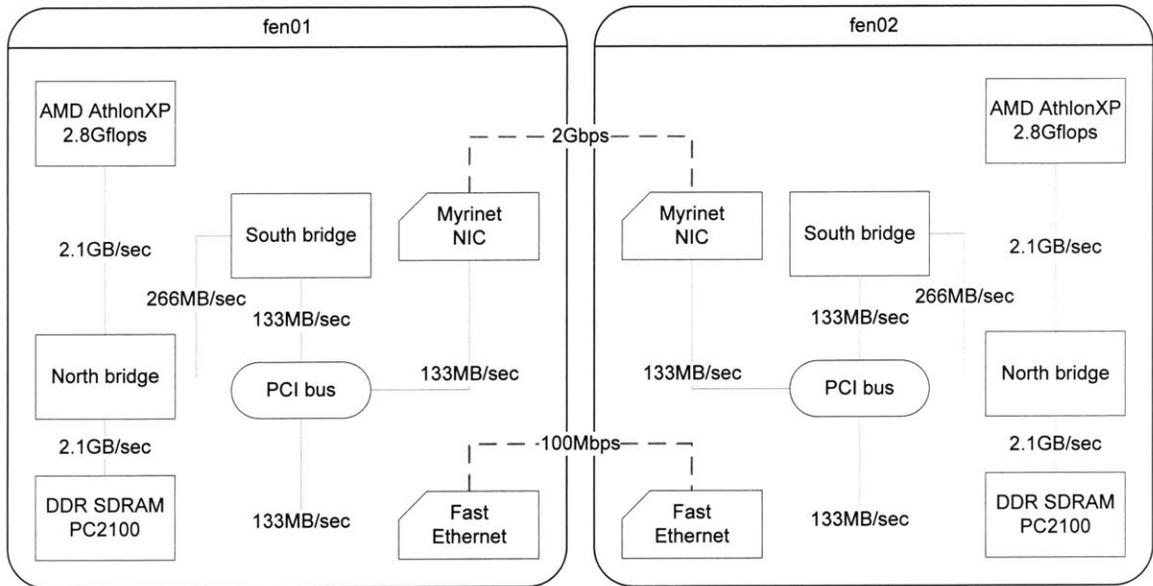


Figure 4-53 System Diagram of Two Interconnected Computers

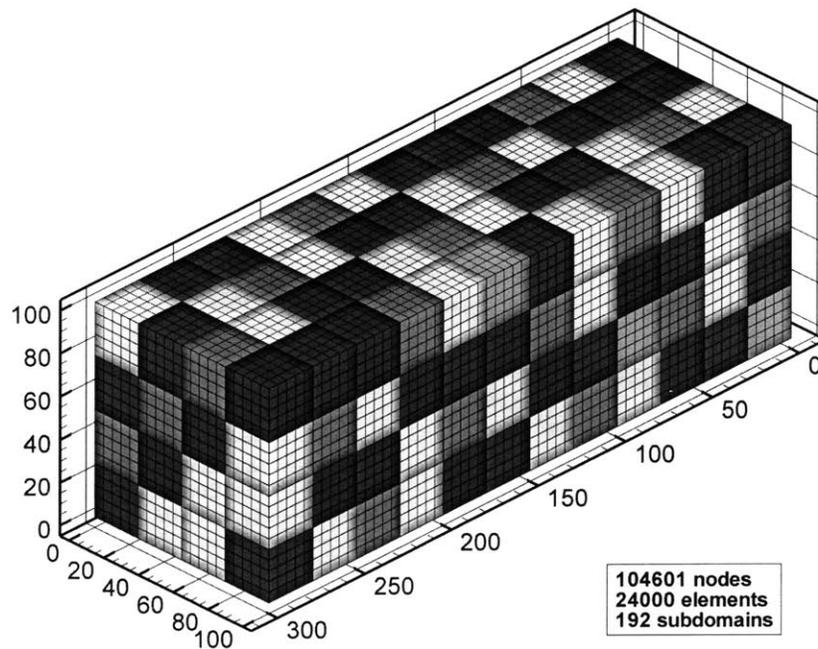


Figure 4-54 Mesh Decomposition for Performance Evaluation on a Structured Mesh

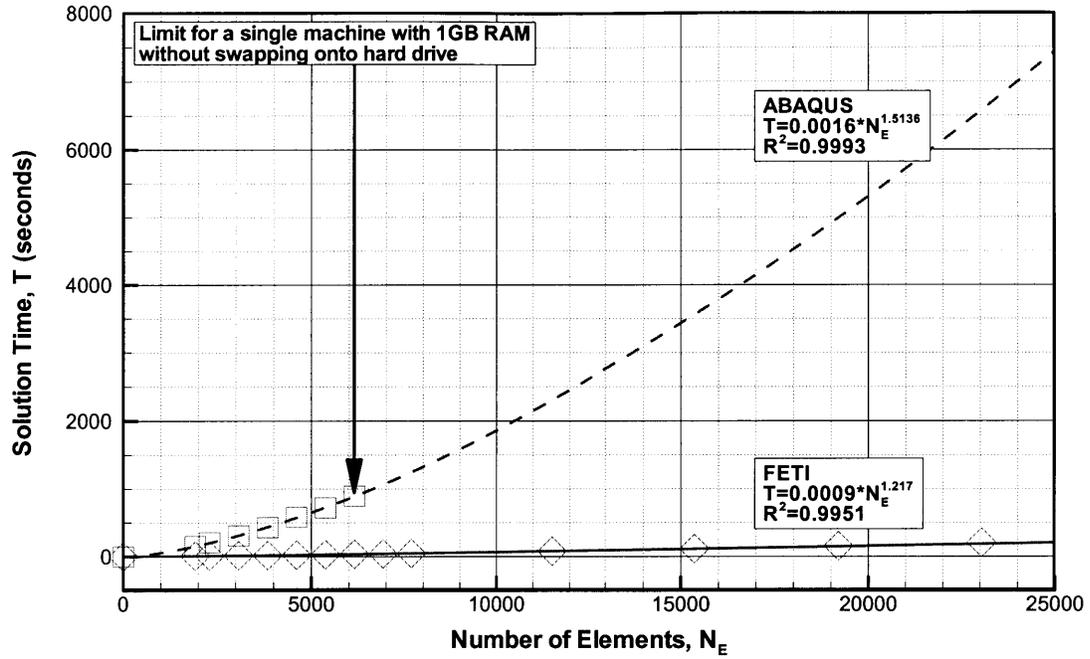


Figure 4-55 Observed Solution Time Scaling with Problem Size for the Structured Mesh

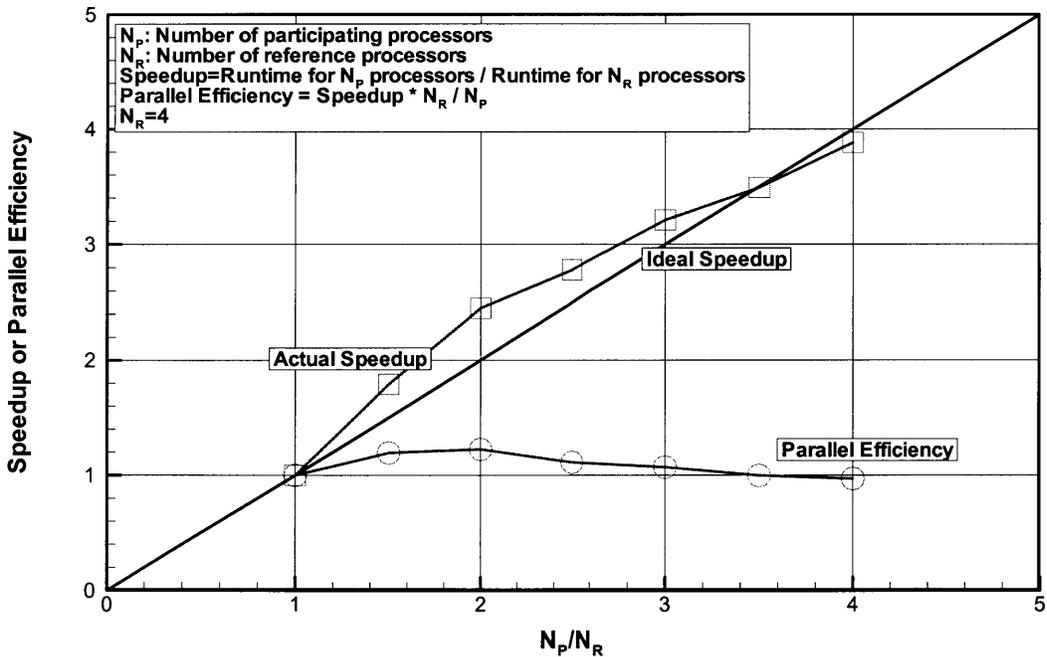


Figure 4-56 Observed Speedup and Parallel Efficiency for the Structured Mesh

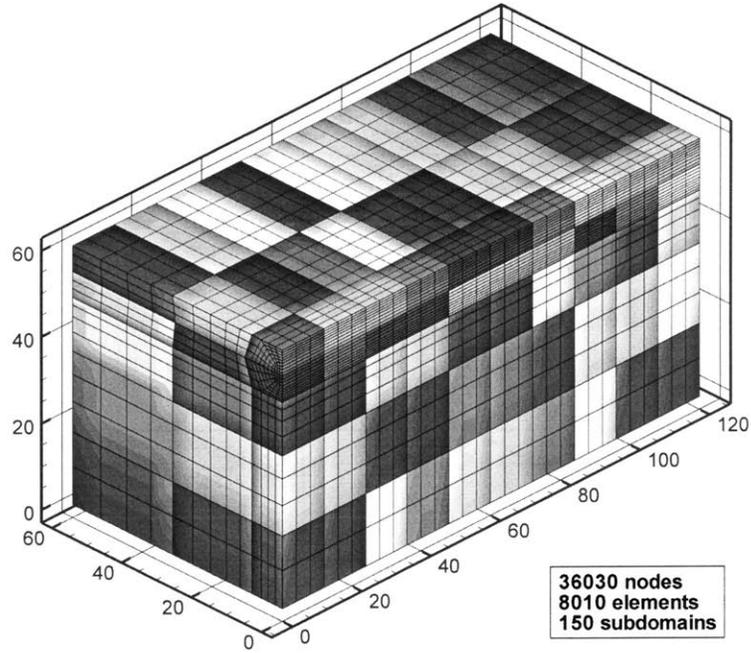


Figure 4-57 Mesh Decomposition for Performance Evaluation for Tunneling

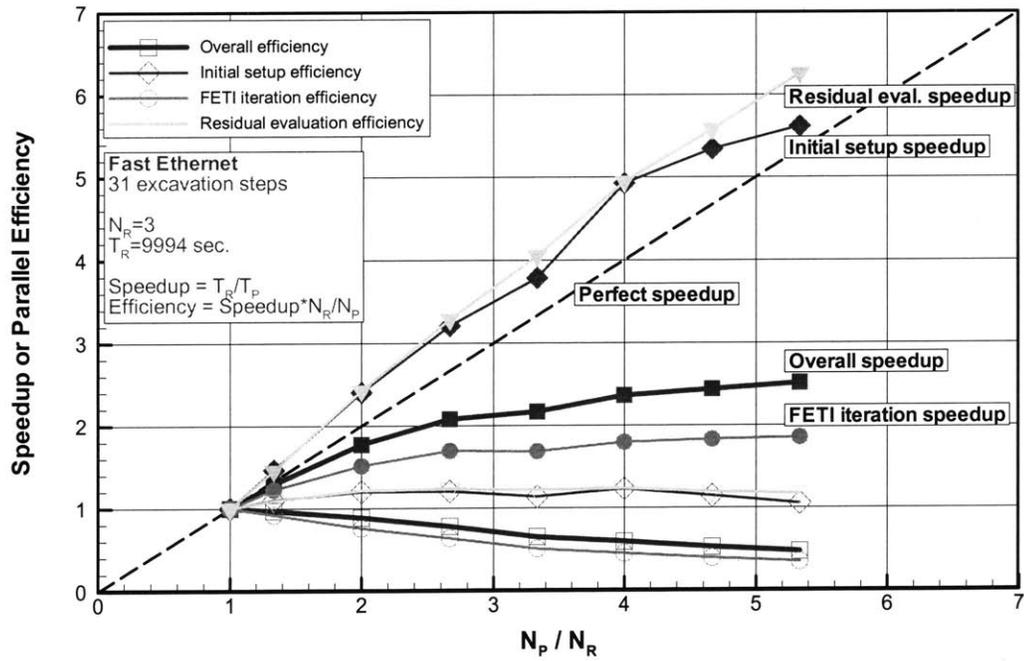


Figure 4-58 Measured Parallel Efficiency and Speedup for an Unstructured Mesh

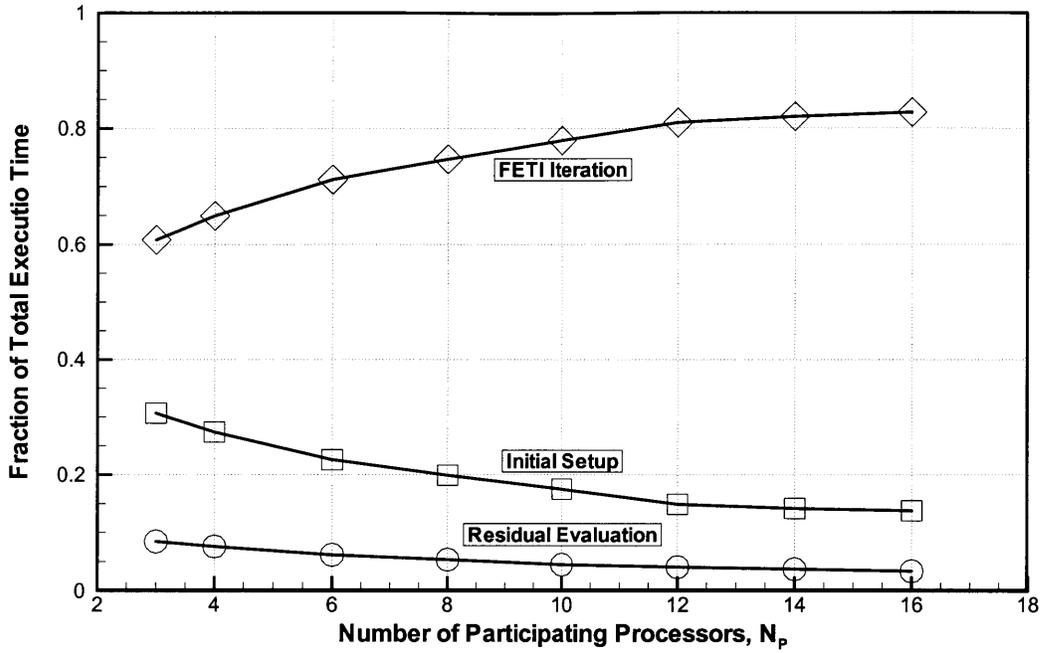


Figure 4-59 Components of Total Execution Time versus Number of Processors

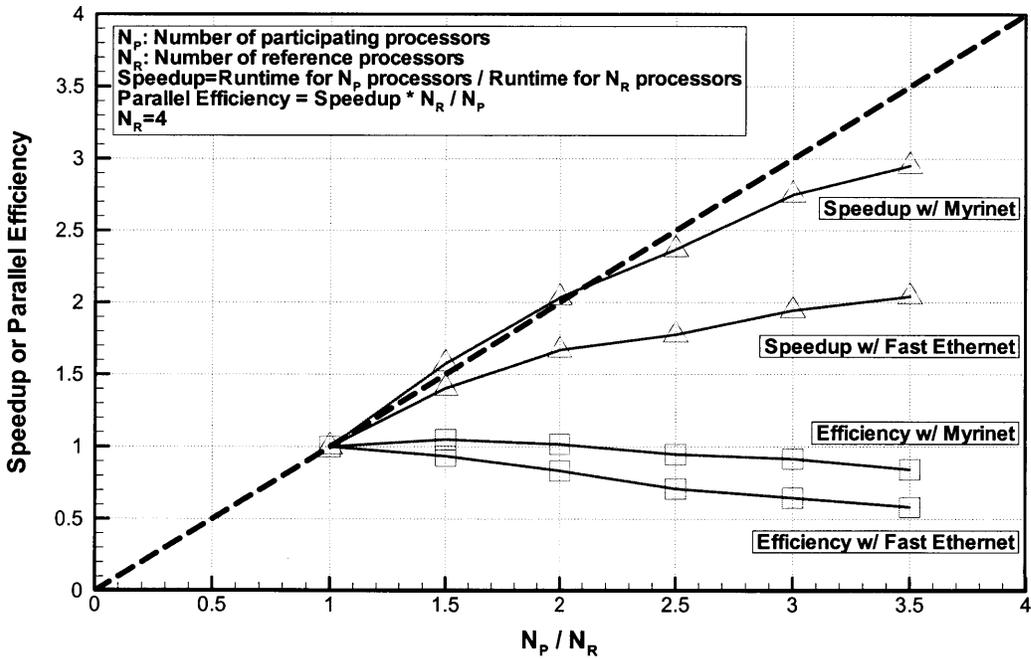


Figure 4-60 Effect of Fast Ethernet and Myrinet on Parallel Performance

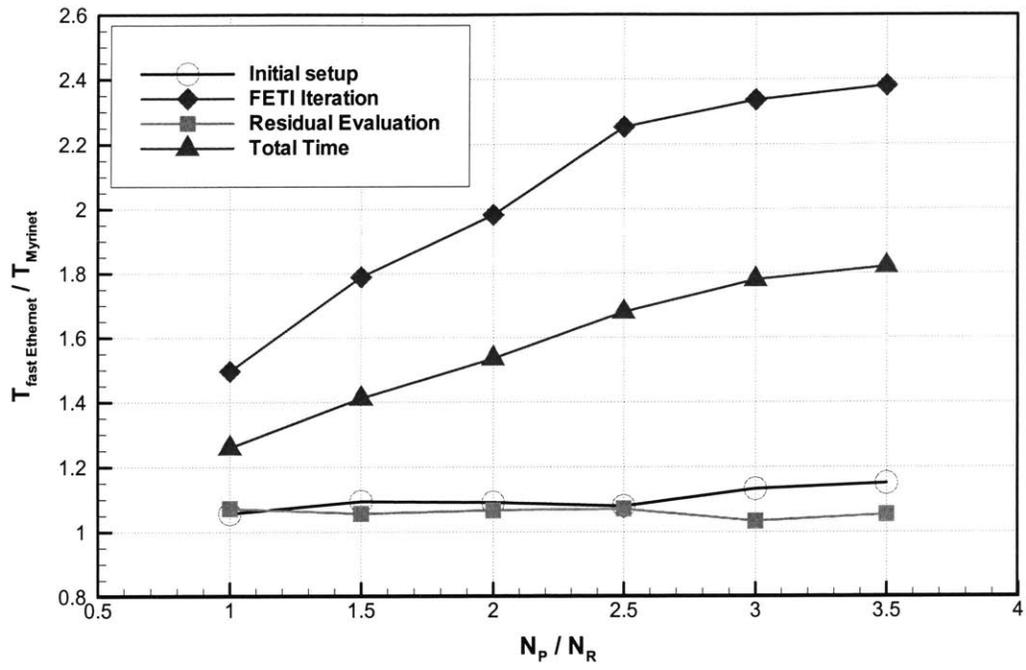


Figure 4-61 Speedups Using Myrinet on Various Components in FETI_FEM

References

- Abu-Farsakh, M. Y., and Voyiadjis, G. Z. (1999). "Computational model for the simulation of the shield tunneling process in cohesive soils." *International Journal for Numerical & Analytical Methods in Geomechanics*, 23(1), 23-44.
- Augarde, C. E., Wissler, C., and Burd, H. J. (1999). "Numerical Modelling of Tunnel Installation Procedures." *Numerical Methods in Geomechanics*, Pande, Pietruszczak, and Schweiger, eds., 329-334.
- Babuska, I., and Rheinboldt, W. C. (1978). "A posteriori error estimates for the finite element method." *International Journal for Numerical Methods in Engineering*, 12, 1597-1615.
- Bathe, K. J. (1996). *Finite Element Procedures*, Prentice Hall.
- Charpis, D. C., and Papadrakakis, M. (2002). "Enhancing the performance of the FETI method with preconditioning techniques implemented on clusters of networked computers." *Computational Mechanics*, 30(1), 12-28.
- Cheng, J.-H., and Kikuchi, N. (1986). "Mesh Re-zoning technique for finite element simulations of metal forming processes." *International Journal for Numerical Methods in Engineering*, 23(2), 219-228.
- Demmel, J. W., Gilbert, J. R., and Li, X. S. (1999). "SuperLU Users' Guide." 58.
- Farhat, C., and Geradin, M. (1998). "On the general solution by a direct method of a large-scale singular system of linear equations: Application to the analysis of floating structures." *International Journal for Numerical Methods in Engineering*, 41(4), 675-696.
- Farhat, C., and Roux, F. X. (1991). "A Method of Finite-Element Tearing and Interconnecting and Its Parallel Solution Algorithm." *International Journal for Numerical Methods in Engineering*, 32(6), 1205-1227.
- Farhat, C., and Roux, F.-X. (1994). "Implicit Parallel Processing in Structural Mechanics." *Computational Mechanics Advances* 2, 1-124.
- Hu, Y., and Randolph, M. F. (1998a). "H-adaptive FE analysis of elasto-plastic non-homogeneous soil with large deformation." *Computers and Geotechnics*, 23(1-2), 61-83.
- Hu, Y., and Randolph, M. F. (1998b). "Practical numerical approach for large deformation problems in soil." *International Journal for Numerical and Analytical Methods in Geomechanics*, 22(5), 327-350.
- Hu, Y., Randolph, M. F., and Watson, P. G. (1999). "Bearing response of skirted foundation on nonhomogeneous soil." *Journal of Geotechnical and Geoenvironmental Engineering*, 125(11), 924-935.
- Jeremic, B., and Sture, S. (1997). "Implicit integrations in elastoplastic geotechnics." *Mechanics of Cohesive-Frictional Materials*, 2(2), 165-183.
- John, V. (2000). "A numerical study of a posteriori error estimators for convection-diffusion equations." *Computer Methods in Applied Mechanics and Engineering*, 190(5-7), 757-781.
- Karypis, G., and Kumar, V. (1998). "Multilevel-k-way Partitioning Scheme for Irregular Graphs." *Journal of Parallel and Distributed Computing*, 48(1), 96-129.
- Komiya, K., Soga, K., Akagi, H., Hagiwara, T., and Bolton, M. D. (1999). "Finite Element Modelling of Excavation and Advancement Processes of a Shield Tunnelling Machine." *Soils and Foundations*, 39(3), 37-52.
- Lee, K. M., and Rowe, R. K. (1990). "Finite element modelling of the three-dimensional ground deformations due to tunnelling in soft cohesive soils. Part 2 - results." *Computers & Geotechnics*, 10(2), 111-138.

- Li, X. S. (1996). "Sparse Gaussian elimination on high performance computers," Ph.D. dissertation, U. C. Berkeley.
- Malkus, D. S., and Hughes, T. J. R. (1978). "Mixed finite element methods and reduced and selective integration techniques: a unification of concepts." *Computer Methods in Applied Mechanics and Engineering*, 15(1), 63-81.
- Park, K. C., R., J. M., and Felippa, C. A. (1997). "An Algebraically Partitioned FETI Method for Parallel Structural Analysis: Algorithm Description." *International Journal for Numerical Methods in Engineering*, 40(15), 2717-2737.
- Rixen, D. J., and Farhat, C. (1999). "A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems." *International Journal for Numerical Methods in Engineering*, 44(4), 489-516.
- Smith, B., Bjorstad, P., and Gropp, W. (1996). *Domain Decomposition : Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press; (June 13, 1996).
- Susila, E., and Hryciw, R. D. (2003). "Large displacement FEM modelling of the cone penetration test (CPT) in normally consolidated sand." *International Journal for Numerical and Analytical Methods in Geomechanics*, 27(7), 585-602.
- Vereecke, B., Bavestrello, H., and Dureisseix, D. (2003). "An extension of the FETI domain decomposition method for incompressible and nearly incompressible problems." *Computer Methods in Applied Mechanics and Engineering*, 192(31-32), 3409-3429.
- Vermeer, P. A. (2001). "On a smart use of 3D-FEM in Tunneling." *Plaxis Bulletin*, 11, 2-6.
- Zienkiewicz, O. C., Boroomand, B., and Zhu, J. Z. (1999). "Recovery procedures in error estimation and adaptivity Part I: Adaptivity in linear problems." *Computer Methods in Applied Mechanics and Engineering*, 176(1-4), 111-125.
- Zienkiewicz, O. C., Emson, C., and Bettess, P. (1983). "Novel Boundary Infinite Element." *International Journal for Numerical Methods in Engineering*, 19(3), 393-404.
- Zienkiewicz, O. C., Kelly, D. W., and Bettess, P. (1977). "Coupling of finite-element method and boundary solution procedures." *International Journal for Numerical Methods in Engineering*, 11(2), 355-375.
- Zienkiewicz, O. C., Taylor, R., and Too, J. (1971). "Reduced integration technique in general analysis of plates and shells." *International Journal of Numerical Methods in Engineering*, 3(2), 275-290.
- Zienkiewicz, O. C., and Zhu, J. Z. (1987). "A simple error estimator and adaptive procedure for practical engineering analysis." *International Journal for Numerical Methods in Engineering*, 24(2), 337-357.

Chapter 5

Finite Element Analyses for a Stacked-Drift Tunnel Construction

5.1 The Río Piedras Station of the Tren Urbano Project

Section 7 of the Tren Urbano project comprises approximately 1500m of underground construction through the town of Río Piedras in San Juan, Puerto Rico. The decision to construct underground was based on the need to preserve historic sites and to avoid disruptions to the business district. Three distinctive tunneling technologies were used in this contract: 1) EPB shield (Earth Pressure Balance shield), 2) NATM (New Austrian Tunneling Method), and 3) Stacked-Drift cavern construction. The new Río Piedras Station (Figure 5-1) houses two levels of platforms, with less than 5.5m cover and a large horseshoe shaped cavern (17m wide, 16m high, and 150m long). The stacked-drift tunneling technique (to be described in 5.1.2) was used to construct a structural arch to support the the underground opening.

This chapter briefly describes the site conditions¹, the construction procedure used for the construction of the Río Piedras station, and the instrumentation program for monitoring the construction performance. The instrumentation data, specifically the surface benchmarks installed on buildings, are synthesized and correlated to the construction activities recorded including the excavation, concreting, and grout injections, Section 5.2.

Simplified 2-D (plane-strain) finite element models are used to assess the ground deformation, and parametric studies are performed to identify important factors affecting the predicted ground movement, Section 5.3. Full three-dimensional finite element calculations are conducted using the parallel finite element code, *FETI_FEM* (Chapter 4), to model closely the advancing tunnel heading, lining installation, and drift concreting. The

¹ Detailed information on the geology, laboratory experiment programs, and engineering properties can be found in Zhang (2002).

calculated 2-D and 3-D ground deformations are compared with the in-situ measurements (Section 5.5), and applied to evaluate damage potentials to the existing buildings, Section 5.6.

5.1.1 Ground Conditions

The underground cavern for the Río Piedras Station was excavated within deposits of weathered old alluvium known locally as the Hato Rey formation (Deere 1955). The old alluvium rests on the Aquada Limestone Formation and consists of thoroughly decomposed sands and gravels with variable thickness exceeding 25m to 30m (Zhang 2002). Project engineers conveniently subdivided the soil profile into three substrates for engineering purposes: upper clay (UC), middle stratified zone (MZ), and lower sand (LS):

- Upper Clay (UC), with an average thickness of 9 m, is medium stiff and brittle, and consists mostly of red, or mottled red and white, silty clays with complex patterns of white veins. These white veins are introduced to the deposit through the dissolution and leaching of the red or brown Fe-oxides by acidic subsurface water along old cracks and root paths (Kaye 1959; Zhang et al. 2003b).
- Middle Zone (MZ), 10 m thick, is very stiff and brittle, and consists of light brown to yellowish sandy clay, silt, or clayey sand with inter-bedded layers of sands. Due to its occurrence at depth, it is less weathered than the UC. White veins also exist in this layer, but are less densely distributed than those in the UC.
- Lower Sand (LS), is a layer containing relatively clean sand or silty sand. Weathering is not so advanced in this layer because of its occurrence at greater depth and under the groundwater table. Therefore, it is not considered as weathered. In addition, due to the difficulty in obtaining undisturbed samples from cohesionless (or highly cemented sand). There has been little laboratory investigation of this layer.

This subdivision is of great convenience for engineering design purposes, but masks the pronounced spatial variability in conventional material classifications and properties indicated from the profiles and boring logs in Figure 5-2.

The groundwater table is located at a depth of 22m, and the UC and MZ layers are partially saturated. Local perched water conditions occur in both UC and MZ layers where there are lower permeability materials.

5.1.2 Underground Construction for the Río Piedras Station

The new Río Piedras station comprises a horseshoe shaped underground cavern 17m wide, 16m high, and 150m long spanning between two 30m deep access shafts. The crown is located less than 5.5m below existing residential and commercial buildings along the Avenida Ponce De Leon (Figure 5-1). The cavern is supported by a structural arch composed of 15 hand-mined square drifts ($3m \times 3m$) in-filled with concrete, prior to the excavation of the main cavern. In order to mitigate the potential damage to overlying buildings due to tunneling-induced ground movements, the original design incorporated a compensation grouting scheme: A shallow grouting gallery (located west of the tunnel centerline and immediately above the main drifts) provided access for installing an array of sub-horizontal tubes-a-manchettes (TAM's) that extended beneath adjacent buildings (Figure 5-3). Grout injections were used to pre-treat the surficial soils (e.g. infilling pre-existing voids), prior to drift excavation. Subsequent grouting was intended to compensate for the ground loss caused by the drift excavation and was controlled using feedback from an automated roof-mounted leveling system (CYCLOPS; LaFonta and Person, 1999).

The designers anticipated ground movements up to 40mm due to drift construction and established a limiting threshold of 25mm for the allowable settlement. When it became apparent that the ground movements were much larger than originally expected, a second consolidation grouting system was installed. This system included a dense array of more steeply inclined TAM's, enabling grout injections to stabilize soils ahead of the advancing drift excavation face.

The excavation of drifts started on May 4, 1998 (Construction Day 0, CD0) and finished in January 2000 (CD640). Figure 5-4 summarizes the recorded drift excavation and concreting. Initially, the construction proceeded following a bottom-up sequence (i.e. 1E, 1W followed by 2E, 2W), but this plan was later modified to accelerate the construction. Thus, drift 8 was excavated at the same time as drifts 3E and 3W, and drifts 7E and 7W were excavated in parallel with drifts 4E and 4W, followed by drift 5 excavations and drift 6 excavations.

Concreting of drifts was typically done immediately after full excavation of each drift, except for 2E and 2W, which were left open (supported by steel sets and timber lagging) to facilitate the transportation of material and equipment for the excavation of drifts 3E and 3W. The schedule of concreting for drifts 1E and 1W is estimated based on the known beginning and ending time recorded. The concreting data for drifts 5 and 6 occurred soon after CD640, is not shown in Figure 5-4.

The original compensation grouting program took place in four phases (Morrison et al. 1999):

- 1) Preconditioning: to stiffen the ground by compressing the soils and by filling voids (in order to improve response at later grouting phases).
- 2) Pre-lifting: to produce a 2 – 3mm heave prior to the excavation to ensure the soil is preconditioned and will react quickly to grouting.
- 3) Real-time compensation: controlled grout injection based on real-time field measurement to control ground surface subsidence during construction.
- 4) Post-grouting: to further compensate ground movements to adjust building response to be within pre-specified tolerances.

The consolidation grouting program was designed to improve the engineering properties of soil (e.g. increase cohesion) for deep drifts, and thus improve the stability during drift excavation.

The stacked-drift construction was conducted between two access shafts. The south shaft is located at the intersection of Arzuaga and Ponce De Leon Avenue and the north shaft is at the intersection of Robles street and Ponce De Leon Avenue. Both shafts are approximately 22m deep and supported by soldier piles and wood lagging, with internal bracing of pipe struts, diagonals, and wales. Excavation of access shafts began on Feb. 1998 and finished in early months of 1998. During the shaft construction, the grouting gallery was excavated from the south shaft in Oct. 1997. Upon 36% completion of the grouting gallery, the excavation was stopped (Nov. 26, 1997) and resumed two days later from the north shaft.

After a period of ground pre-treatment (by injecting grout), the drift excavations started on May 4th, 1998 (first construction day, CD0).

Figure 5-5 illustrates the location and time of grouting activities throughout the stacked-drift construction. Compensation grouting was performed extensively for the first 200 days, and during the construction of drifts 4E, 4W, 7E, and 7W. It can be seen that the compensation grouting closely follows the excavation of drifts 4W, 7E, and 7W. Consolidation grouting was conducted during the construction of drifts 2E, 2W, 3E, and 3W. No grouting was performed after CD500, or during excavations of drifts 5E, 5W, 6E, and 6W.

The drifts were excavated by hand mining assisted in places by a Technicore tunnel digging machine. Morrison et al. (1999) summarizes the typical excavation sequence:

- 1) Advance poling plates in the crown using hydraulic jacks. (A square shield was used for drifts 3, 4).
- 2) Excavate top heading with spaders and installing poling plates off the bench.
- 3) Excavate the bench and install steel-set and lagging support. (Drifts 1 through 4 used timber lagging while the remaining drifts used shotcrete.)
- 4) The tail void is back-filled using excavated materials.

5.2 Measured Surface Settlement during Construction

Measurements of ground deformations within the zone influenced by the tunnel construction are essential to monitor the impact on the overlying structures and underground utilities. A comprehensive array of geotechnical instrumentation and survey points was used to monitor the vertical and lateral ground deformations, to verify settlement predictions, to implement settlement control measures, to adjust the construction work plan as required, to monitor damage to the existing buildings, and to modify the grouting program schedule as needed.

Most of the structures above the stacked-drift tunneling section (SDT) correspond to two or three level concrete and/or masonry buildings, although some older structures such as the

old Río Piedras city hall are a combination of brick and concrete members. The buildings are supported on spread footings cast at shallow depths.

Three main streets (Figure 5-3) are located within the tunneling area: the Ponce De Leon Avenue runs parallel to the SDT longitudinal axis directly above the grouting gallery; and two cross-streets, the Paseo de Diego and Arzuaga, are on the eastern side of the tunnel axis. The underground utilities in the SDT area include water lines, sanitary sewer lines, electric lines, storm drainage, and telephone lines.

Figure 5-6(a) shows the geotechnical instrumentation installed around the SDT section including the following:

- a) Ground surface settlement points (SS) are reference points established on either the sidewalks, curbs, or the pavement surface, and read by optical survey methods.
- b) Subsurface settlement rods (SR), provide more reliable measurements of vertical displacements at a depth of approximately 2m below the ground surface.
- c) Exterior and interior building monitoring arrays (BMA) are fixed points installed at selected places, such as columns of buildings. These are also monitored by optical survey and measure vertical displacements.
- d) Inclinerometers (I) used to measure the lateral deflection profile within the soil mass during excavation. Three inclinometers were installed at the SDT section: I-5 and I-6 located in front of building 732 and 740, respectively, on the west side of the drifts; and I-7 located 2.6m from the edge of the east drifts close to Paseo De Diego.
- e) Crack gauges used to measured changes in wall cracks during construction.
- f) Observation wells to measure positions of groundwater.
- g) Building movements have also been monitored by Soletanche in conjunction with the grouting systems using an automated (roof mounted) leveling system referred to as Cyclops (La Fonta & Person, 1999).

In this study, emphasis is put on the readings of BMA for its extensive coverage along the Ponce De Leon Avenue, which is the primary influence area of construction activities of the stacked-drift tunneling. The BMA benchmarks (Figure 5-6(b)) are sub-divided into nine clusters based on their spatial locations and similarities in the measured performance, and Figure 5-7(a) – Figure 5-7(d) show the settlement-time histories for all nine clusters arranged at different offset distances from the centerline of the SDT. Benchmarks in Figure 5-7(a) are located 12m west of the centerline, those in Figure 5-7(b) are 5m west, while Figure 5-7(c) and Figure 5-7(d) show results from 5m to 34m east of the centerline of the construction. The results show the following:

- 1) All benchmarks show a cycle of settlement followed by heave during the first 50 construction days¹. The settlement in this first phase are caused by the excavation of the grouting gallery, while the heave is the result of initial preconditioning and pre-lifting (Morrison et al. 1999).
- 2) Benchmarks in clusters BM_S1 and BM_S2 start showing more variability than other clusters of benchmarks after CD100. This is caused by extensive compensation grouting performed in the southern half of the SDT (Figure 5-5). Figure 5-8 summarizes the number of compensation grouting injections along the SDT alignment during this period, and significantly more grout injections were applied between sets 60 – 120, (i.e. the southern part of the stacked-drift tunneling section).
- 3) No significant ground movement occurred between CD200 – CD280 (excavation timeframe for drifts 2E, 2W), except a 10 - 20mm movement registered CD207. The movement corresponds to a water main break under building 745 in the northern part of the construction. This movement was quickly compensated by grouting.
- 4) After CD280, a sudden heave is registered in the middle section of the construction (observed in benchmarks of BM_S3, BM_4, and BM_SN). These movements are most likely caused by the consolidation grouting performed toward drifts 2E and 2W,

¹ According to the construction records in Figure 5-4 and Figure 5-5, the drifts 1E and 1W were excavated from the north shaft during this period but did not advance to any of the benchmark points shown in Figure 5-6.

and are quite surprising given the depths of these drifts below the ground surface (~ 16m) .

- 5) A larger extent of ground heave is observed between CD350 – CD400 in nearly all benchmarks. These movements are clearly identified as the result of consolidation grouting.
- 6) The largest movements are observed in clusters BM_N2 and BM_S2 which are close to the centerline of the SDT, directly above the grouting gallery (thus, with the least cover).
- 7) Between CD450 – CD500, a significant amount of settlement is observed by most benchmarks. An average of 16 – 32mm settlement is observed in BM_N1, BM_SN, BM_S1, BM_S3, and BM_N3 groups; 40mm settlement is observed in BM_S2 and BM_N3 group; 20mm settlement in BM_4 group, and 12 mm in BM_5 group. In the same period, compensation grouting is performed during this period closely following the construction of drift 4 and drift 7 as shown in Figure 5-5.
- 8) At the end of CD500 (end of controlled grouting), most benchmarks¹ with similar distance to the centerline have comparable magnitudes of surface settlement. Subsequent movements are due directly to the excavation of drifts 5 and 6.
- 9) Large settlements of cluster BM_N2 appear to be linked to excavation of drifts 7E and 7W, while subsequent surface settlements are linked to 5W and 6W. More aggressive compensation grouting appears to have controlled effects of 7E/7W cluster BM_S2 (CD460-500). However, both BM_N2 and BM_S2 undergo at least 20mm settlement due to excavations of drifts 5W and 6W (immediately beneath these monitoring points).
- 10) The final settlements of BM_N2 are in the range 120 – 130mm, while BM_S2 reaches 70 – 90mm.

¹ The only exception is cluster BM_S1 noted previously.

Figure 5-10 shows the interpolated surface settlement distributions at CD180 corresponding to the completion of drifts 1E and 1W and at CD703, corresponding to the end of stacked-drift construction. From the contour lines, two surface settlement depressions are developed during the construction. One is close to the south access shaft and above the end of the grouting gallery (more compensation grouting was performed close to the south access shaft as shown in Figure 5-9); the other is near the end of existing buildings (#743, 745, 747) close to the north access shaft¹.

The surface settlement depression close to the south shaft is linked to the geometry of the grouting gallery. The grouting gallery is not connected to the south access shaft but makes a 90-degree turn with vertical access aligned with the drift 8. As a result, there is less soil cover is above the excavation and more settlement. The other depression of surface settlement north of the Paseo de Diego is likely caused by the water main break on CD207. The event occurred under building 745 close to the center of the depression, and caused 10mm – 20mm settlement throughout the alignment of the stacked-drift tunnel.

The excavation procedure, consolidation grouting, and recorded surface movement are summarized together in Figure 5-11. The consolidation grouting activities applied to the East and the West of centerline are separated, and the surface settlement is chosen from benchmarks in BM_S2 and BM_N2 that show the largest movements. The following events (noted in Figure 5-11) can be observed:

- A. A sinkhole (estimated 10ft deep from the ground surface) developed on CD77 above set 12 of drift 1W due to seepage conditions combined with sandy ground condition (Urquiza 2000). As a result, the excavation of drift 1W stopped for 20 days, and consolidation grouting was applied to stabilize the open face.
- B. A break up of a 20 inches iron pipe beneath Ponce De Leon Avenue nearby building 745 on CD207. The pipe collapse induced surface subsidence varying from -5mm at the tunnel south end increasing to -20mm at the north end (Urquiza 2000). Compensation grouting was used to compensate the ground loss. The event can be

¹ More surface settlement contours can be found in Appendix H.

seen as a spike on all benchmarks shown in Figure 5-7 (except group BM_5, which was less frequently monitored).

- C. Due to sandy ground conditions and observed seepage, consolidation grouting was conducted to help stabilize the heading of drifts 2 and drifts 3. This consolidation grouting generated heave of the ground surface that can be clearly seen from benchmark group BM_SN in Figure 5-7(a) on about CD280.
- D. Excessive heave observed from benchmarks in BM_S2 group. The heave is clearly related to the consolidation grouting performed ahead of drift 2, as heave almost immediately occurred after the consolidation grouting is performed on both eastern and western side of the construction. The heave, however, is not observed in BM_N2 during the drift 2 excavation, as the consolidation grouting was not performed under those benchmarks.
- E. The grouting-induced heave observed during the drift 2 excavation is not seen during drift 3 excavations. This is because a) consolidation grouting are performed mostly on the east-side of the drifts during the excavation under the northern group of benchmarks, thus this benchmark array (above grouting gallery, thus biased by the western side of the construction) is less sensitive to the events on the east-side of the tunnel. b) Drifts 2E and 2W are open during the excavation of drift 3E and 3W, thus the ground can move toward drifts 2E and 2W to accommodate the additional material (injected grout). In contrast, drifts 2E and 2W were excavated after concreting was completed for drifts 1E and 1W. c) Almost twice as much of grout (east-side and west-side combined) was injected under the BM_S2 benchmarks (792 m^3 between sets 75-105) than under the BM_N2 benchmarks (444 m^3 between sets 25-45). All these factors contributed to the different response of the ground to the consolidation grouting.

5.3 2D Finite Element Analyses for the Río Piedras Station

5.3.1 Introduction

The development of a realistic finite element model for the stacked-drift tunnel represents a challenging task due to the following issues:

- 1) The in-situ soil (weathered alluvium) is highly variable and exhibits unusual engineering properties associated with breakdown of aggregates that comprise the intact structure (Zhang et al. 2003a). The current modeling work is based on a simplified interpretation of the soil profile and simulates the shear behavior of the intact soil based on lab tests on block samples (Zhang 2002).
- 2) The construction sequence shown in Figure 5-4 involves complex three-dimensional interactions between individual drifts constructed concurrently from different directions.
- 3) The compensation and consolidation grouting systems have been used to control ground movements and stabilize the drift headings. Multiple injections were carried out through TAM injection points using a bentonite-cement silicate mixture with average injection pressures, $p = 6bar$ ¹. This type of grouting may induce fracturing of the intact soil mass. Other effects of the grouting are very difficult to assess and may induce de-structure of the intact soil, generating changes in the stiffness and shear strength of the soil mass. The complexity of these grouting mechanisms complicates significantly numerical modeling.
- 4) The model must represent accurately stress conditions at three distinct scales: a) the individual grout injection b) the individual drifts, and c) the complete cavern. This type of comprehensive model is beyond the scope of current computation capabilities. The current analyses focus on modeling the drift construction and do not simulate the grouting system.

5.3.2 Description of 2-D Model

Figure 5-12 shows the finite element mesh and boundary condition used to model the stacked-drift construction using the ABAQUS finite element program. The mesh assumes

¹ Records show injection pressure varying widely.

the bedrock occurs at 30m¹. The mesh contains 733 8-noded quadratic plane-strain elements (CPE8) with 2264 nodes. This same mesh is later extruded for three-dimensional finite element analyses. Fully drained conditions are assumed based on 1) the ground water table is below the base of drifts 1E and 1W, and 2) high permeability is observed in laboratory tests on intact materials in UC and MZ.

There are great uncertainties regarding the ground conditions and hence, parametric studies have been performed to understand how selected parameters influence predictions of ground movements. Table 5-1 lists the parameters chosen as the reference configuration, those are later varied to understand the effects of K_0 , lining stiffness, soil stiffness, concreting of drifts, friction angle of Middle-Zone, and yield criteria. The Lower-Sand layer is assumed to have the same engineering property as the Middle-Zone soil due to uncertainties in identifying the elevation of the LS-MZ interface.

The stiffness and shear strength parameters c' and ϕ' are estimated from laboratory drained triaxial shear tests on specimens trimmed from intact block samples (Zhang 2002). Figure 5-13 summarizes the stress paths and estimated Mohr-Coulomb yield envelopes for UC and MZ. Normalized stiffness data from these tests are plotted versus shear strain level in Figure 5-14. A representative normalized stiffness G_{sec} / σ'_{oct} is chosen at shear strain, $\gamma \cong 0.1\%$, based on prior experience. It should be noted that both UC and MZ soils have apparent cohesion $c' = 25 \pm 1kPa$, but large differences in internal friction angle ($\phi' = 24^\circ$ for UC vs. 39.5° for MZ).

The base case analysis assumes an earth pressure coefficient at rest, $K_0 = 0.5$. This is consistent with empirical equations such as Jaky's formula:

$$K_0 = 1 - \sin \phi' \quad (5.1)$$

¹ The depth to the bedrock is highly variable, and most of the boreholes did not encounter the bedrock interface (see Figure 5-2).

where ϕ' is the effective friction angle¹. In principle, K_0 should be related to the geologic history of the old alluvium, but there is very little reliable data for estimating K_0 for a residual soil such as the weathered alluvium. The current analyses consider $K_0 = 0.5 - 1.0$ as a base-line range, ($K_0 > 1$ is only expected from materials that have been heavily over-consolidated). Grouting activities used to improve engineering properties of soils in this site may also change the initial stress condition by changing both the unit weight of soils and the K_0 condition. This effect has been ignored in the current numerical models.

The drifts are supported by a lining system composed of steel sets (mostly W8x24) and timber lagging, and Table 5-2 summarizes the assumed elastic properties for the FE models. Steel sets are spaced at 1.2m and are installed immediately after excavation of the tunnel heading. The lining system is represented using continuum elements in both two-dimensional and subsequent three-dimensional FE modeling of stacked-drift construction. This is because the subsequent parallel computations (using the one-level FETI method) for solving three-dimensional FE models are more efficient using continuum elements. Finite element models with beam or shell elements could be more efficiently solved with a two-level FETI method (Farhat et al. 1998a) or FETI-DP formulation (Farhat et al. 2000). The adequacy of using continuum elements for modeling linings of tunnels has been discussed recently by Augarde et al. (1999) through numerical experiments and comparisons to analytical solutions.

5.3.3 Simulation Procedure for Stacked-Drift Construction

Three major construction activities are modeled in the FE simulations: drift excavations, installation of the lining system, and concrete infilling. The recorded progress of these activities for the stacked-drift construction has been summarized in Figure 5-4. Table 5-3 summarizes the calculation steps for simulating the stacked-drift construction in 2-D plane-strain FE model.

Soil excavation is modeled by removing finite elements within each drift and hence, eliminating the contribution of the finite elements to the global stiffness matrix and force

¹ According to this expression, $K_0 \cong 0.35 - 0.65$ with $\phi' = 20 - 40$

vector (due to the self-weight). The analyses assume full stress release prior to installation of the lining. (i.e., there is full re-distribution of stress within the soil mass as the drift cavity is internally stable).

Concreting of drifts is modeled by increasing the weight of the lining elements to account for the weight of the in-filled concrete, while the stiffening effect of concreting is ignored. Ignoring the stiffening effect of concreting should generate conservative results, and it is shown later changing lining stiffness has trivial effect on computed ground deformations

Increasing the weight of the lining elements is achieved in ABAQUS by increasing the reference magnitude in the keyword *DLOAD. Consider typical $3m \times 3m$ drifts in-filled by concrete with unit weight of $23kN/m^3$, thus the weight per unit length of the concreted drift is:

$$23 \times 3 \times 3 = 207 (kN/m) \quad (5.2)$$

The volume per unit length of the continuous lining elements:

$$2 \times 3 \times 0.2 + 2 \times 2.6 \times 0.2 = 2.24 (m^3/m) \quad (5.3)$$

Thus the unit weight of lining elements to account for the weight of in-filled concrete should be:

$$207/2.24 = 92.4 (kN/m^3) \quad (5.4)$$

Thus the concreting of drifts is simulated by increasing the reference magnitude of *DLOAD from 1.0 to 316.4 ($92.4/0.292$).

The advantage of simulating the effect of drift-concreting by the proposed procedure is simplicity and efficiency. Using the proposed procedure, the finite element mesh requires no modification to add elements to represent concrete inside drifts, thus the number of unknowns is not increased compared to the finite element model without considering concreting. Furthermore, the procedure avoids numerical difficulties of re-activating elements to represent infill (after distortion of the linings).

5.3.4 Results for 2-D Models

Parametric studies are conducted to by varying several parameters listed in Table 5-4, and the rest of this section documents the results and findings based on the parametric study.

Case P-1: Figure 5-15 summarizes the predicted surface settlement troughs due to drift construction based on the reference properties in Table 5-1. Figure 5-15(a) includes the effect of the grouting gallery while Figure 5-15(b) subtracts the effects of grouting gallery excavation from subsequent movements. This figure shows that the 15 drifts generate a maximum settlement of approximately 115mm.

The grouting gallery induces a maximum surface settlement of 17mm, while 80% of the surface settlement is due to construction of shallow drifts 4 – 8. The model predicts 12mm settlement due to drift 1E and 1W, while drifts 2 and 3 each cause about 5mm.

Excavation of the grouting gallery reduces the net settlement (Figure 5-15(b)) on the west side of the cavern compared to the east, while the rest of the excavation sequence is approximately symmetrical. The reduced settlement on the west side of SDT is likely the combined effect of buoyancy and ground stiffened by the lining of the grouting gallery.

Case P-2 and P-13: Figure 5-16 examines the effect of the assumed K_0 condition. The K_0 value affects not only the initial stress state, but also the soil stiffness (which is assumed proportional to the in-situ confining pressure). Figure 5-16(a) shows the settlement troughs (the effects of the grouting gallery have been removed) using stiffness parameters evaluated for $K_0 = 1.0$ (while the initial stress is maintained at $K_0 = 0.5$). Comparing Figure 5-16(a) and Figure 5-15(a), it is seen that this small change in stiffness reduces the calculated maximum surface settlement by 30% (115mm to 80mm). The maximum surface settlement is further reduced to 60mm assuming $K_0 = 1.0$ as shown in Figure 5-16(b).

Figure 5-16 suggests that K_0 is an important factor for considering the stacked-drift tunneling induced surface settlements as it can affect the computed surface settlements by as much as 50% (115mm to 60mm). The significance of the K_0 condition is because: 1) the change of confining pressure affects the soil stiffness, 2) smaller shear strains are needed to yield the soil in certain shear modes, and 3) $K_0 = 0.5$ implies smaller release of horizontal

forces acting on the sides of the lining structure (drifts 1 – 4), thus it is easier for the upper structure (drifts 7 and 8) to deform vertically, and hence larger settlements are expected for $K_0 = 0.5$ than $K_0 = 1.0$.

Case P-5 and P-6: The effect of soil stiffness has been investigated by varying the normalized stiffness ratios G_{sec} / σ_{oct} using parameters corresponding to shear strain levels $\gamma = 0.01\%$ and $\gamma = 1.0\%$ (Figure 5-14). Figure 5-17 compares the settlement troughs for these two extreme stiffness parameter sets.

The maximum surface settlements due to different soil stiffness assumption range between 70mm (P-5), 115mm (P-1), and 400mm (P-6) respectively. The shape of settlement troughs, however, is similar for all three set of parameters. It is difficult to choose one single appropriate stiffness value that depends on both stress levels and subjected strains for linear elasticity based soil models, especially for practical engineering problems that the ground is sheared non-uniformly and the stress levels can vary greatly.

Case P-3 and P-4: Lining stiffness has minimal effect on the computed surface settlement in these 2-D FE calculations as demonstrated in Figure 5-18. The lining stiffness is increased by a factor of 10 – 100 (Table 5-6, Table 5-7) from the reference configuration in Figure 5-15, but the predicted surface settlement troughs are practically identical. This result occurs as there is full stress redistribution prior to lining installation and hence, the lining acts as stiffening element for subsequent excavations.

These simulations regarding lining stiffness also suggest the chosen lining stiffness for the reference configuration is appropriate for the modeling purpose, as the chosen lining stiffness is based on the average of in-plane stiffness and the out-of-plane stiffness, and the averaged lining stiffness is much lower than in-plane stiffness.

Case P-10, P-11, and P-12: Due to the limited number of triaxial experiments, there are great uncertainties involved in determining shear strength parameters for the Middle Zone soil (MZ). As shown in Figure 5-13, only three drained tests and two undrained tests are available for MZ, and two different sets of parameters (Zhang 2002) for Mohr-Coulomb failure criteria are derived based on different interpretations of the available experimental data.

Figure 5-19 shows the effect of different friction angles for MZ. Decreasing the friction angle by $\Delta\phi = 5^\circ$ surface settlement is increased by roughly 5mm throughout all stages of construction (cf. Figure 5-15(b) and Figure 5-19(a)), and by another 10-20mm increase of settlement is seen due to another 5-degree reduction of friction angle according to Figure 5-19. At the same construction stage, the shapes of settlement troughs are similar between different friction angles. The increase of settlement is the result of decreased friction angle, and thus, an increased amount of plasticity in the ground.

Case P-7: Figure 5-20 shows the surface settlement predictions with the weight of infill excluded from the FE model. By comparing Figure 5-20 with the base case, Figure 5-15, it can be seen that concrete infilling accounts for a 30mm difference in the maximum surface settlement $\delta_{v,max}$, and also affects the trough shape for the last two stages (drifts 5 and 6). The comparisons confirm that the concrete infilling must not be ignored in the stacked-drift simulations.

Even though stiffness of the concrete infilling is excluded from the FE model, it should have similar effect of increasing the lining stiffness (case P-3 and P-4), and should have minimal effect on the computed surface settlements.

Case P-8 and P-9: The Drucker-Prager failure criterion is considered as an alternative to Mohr-Coulomb failure for three-dimensional analyses. The Drucker-Prager failure criterion features a smooth yield cone in the principal stress space, while Mohr-Coulomb failure criterion exhibits an irregular hexagonal cone. Therefore, these two failure criteria cannot be matched perfectly, and Drucker-Prager effectively describes a friction angle that varies with the mode of shearing (i.e. is a function of the intermediate principal stress, σ_2 , and direction of the major principal stress, δ). Table 5-12 summarizes the equations for matching shear strength parameters for Drucker-Prager failure criterion with Mohr-Coulomb failure criterion in three major shear modes: triaxial compression, triaxial extension, and plane-strain¹. Matching parameters for Drucker-Prager failure criterion in the triaxial compression

¹ ABAQUS/Standard User Manual

mode with Mohr-Coulomb failure criterion will result in a higher friction angle in the triaxial extension mode as follows:

$$\sin \phi^{TXE} = \frac{3 \sin \phi^{TXC}}{3 - 2 \sin \phi^{TXC}} \quad (5.5)$$

Note that the resultant friction angle from Drucker-Prager failure criterion in triaxial extension mode is always larger than in triaxial compression mode.

Figure 5-21(a) shows the calculated surface settlement troughs with Drucker-Prager failure criterion used for the Middle Zone soil. The Drucker-Prager failure criterion is matched with Mohr-Coulomb failure criterion in plane-strain condition using parameters shown in Table 5-11. It is seen in Figure 5-21(a) that the resultant surface settlement troughs are affected only slightly (less than 5mm) due to the change of failure criterion used for the middle zone soil.

Table 5-10 lists the parameters for Drucker-Prager parameters matching with Mohr Coulomb in the triaxial compression mode with $\phi'_{TC} = 26^\circ$. In this case, the Drucker-Prager failure criterion generates $\phi'_{TE} = 39^\circ$ (i.e. matching the laboratory data shown in Figure 5-13). The surface settlement troughs based on Table 5-10 are shown in Figure 5-21. The magnitude of the surface is reduced by 10% (103mm max.) compared to the reference configuration (115mm max.). This behavior is related to the reduced friction angle in the compression mode of shearing.

5.3.5 The Effect of Construction Sequences and Concreting

Different construction sequences have been considered using 2-D (plane-strain) FE simulations with full stress release as summarized in Figure 5-22. Three different sequences were considered: 1) the actual sequence applied in the Tren Urbano project, 2) a bottom-up sequence (drift 1E/W, 2E/W, ... 7E/W, 8), and 3) a top-down sequence (8, 7E/W, ..., 1E/W). For each construction sequence, the weight of concrete inside drifts is either

considered or completely ignored. The resulting six settlement troughs are shown in Figure 5-22, while Figure 5-23 shows the surface ground loss, ΔV_L ¹, derived from these simulations.

It is seen that different construction sequences can produce noticeable differences in the predicted surface settlement troughs. The bottom-up construction sequence with concreting generates the most settlement, with a narrower and deeper settlement trough than the other cases. Although the bottom-up sequence produces the deepest trough, the top-down sequence with concreting generates the largest volume loss, $\Delta V_L = 2.5\%$ (compared to 2.25% for the bottom-up sequence).

It is interesting to note that the actual construction sequence with concreting considered shows great resemblance to the top-down construction sequence in the trough shape, but the volume loss ratio is closer to the bottom-up sequence. These results suggest the shape of the final settlement trough is mostly determined by the shallow drifts, as the actual construction sequence is initially identical to the bottom-up sequence up to drift 3. As a result, the final settlement trough with the actual construction sequence is close to the top-down construction sequence, with similar magnitude of movement and similar location of maximum surface settlement.

These results also show clearly the effect of the weight of the concrete infilling. It is seen in Figure 5-22 that by excluding the weight of concreting, the maximum magnitude of surface settlement reduces 30mm – 50mm depends on the construction sequence used, while the volume loss is reduced by more than 0.6% by ignoring the presence of concreting according to Figure 5-23.

The top-down sequence is more influenced by the presence of concreting than the other two cases. The weight of concrete increases the magnitude of surface settlement by 83% (50mm) and the volume loss by more than 1%. Figure 5-24 shows the upward movement contours

¹ The ground loss is defined as $\Delta V_L = \frac{V_{trough}}{V_{exca}} \times 100\%$, where V_{trough} is the volume of the surface settlement trough, and V_{exca} is the volume of excavation. The calculation of volume loss from finite element solutions is detailed in Appendix I.

of plane-strain finite element calculations using top-down sequence with and without concreting. The figure suggests the major difference of ground movement with and without concreting is due to the different behavior of the core. For simulations with concreting, the weight of concrete suppresses the heave of the core by applying additional load to the core (because the unit weight of concrete is greater than the soil), and only the bottom of drift 1 shows upward movement. On the other hand, simulations without concreting show the entire core area undergoes heave due to unloading. As a result, the net movement for top-down construction without concreting on the ground surface is significantly less than using the same sequence with concreting.

The contours of upward movement are also shown for the bottom-up sequence in Figure 5-25. With concrete infilling, vertical heave displacements are minimal inside the core for both top-down and bottom-up sequence. The following differences are observed for top-down and bottom-up sequences without concrete infilling:

- 1) The bottom-up construction sequence shows mostly downward movements near the deep drifts (1, 2, and 3) inside the core. This is because when a drift is excavated, the materials on the roof of the drift move downward due to gravity and “drag” down its surrounding soils. For top-down sequence, however, the material above the roof of deep drifts has been removed prior to drift excavations, thus upward movements are observed instead. (e.g. material on the roof of drift 2 is removed prior to the drift 2 excavation due to the excavation of drift 3). The upward movement seen in top-down sequence is due to the buoyancy effect.
- 2) Larger heave is seen on top of the core with bottom-up sequence than with top-down sequence. This is because different soil-structure interactions between the lining system and the ground. For top-down sequence, the lining forms a stiff mat structure on top of the core, and the mat structure causes uniform vertical movements for subsequent ground movements, and the heave from the core need to push the entire mat upwards. On the other hand, the bottom-up sequence initially has two completely separated structures (formed by lining) on the sides of the core, thus there are no constraints on heave inside the core.

The surface settlement troughs of the actual construction sequence are more similar to the top-down sequence than the bottom up sequence (even though the actual construction was based on a bottom-up construction sequence). After drifts 2E and 2W were excavated, drifts 3E, 3W, and 8 were excavated simultaneously, followed by drift 4E, 4W, 7E, and 7W, forming a mat-like structure on top of the core. When concreting of drift 7E and 7W occurs, the load from concreting is immediately spread throughout the mat-like structure, including the lining for drift 7E, 7W, 8, and grouting gallery. As a result, the actual construction sequence results in a flatter and shallower settlement trough than the bottom-up construction sequence, but also generates a maximum settlement to the east-side of the cavern centerline.

5.3.6 Summary of 2-D FE Models

- 1) The reference finite element model produces settlements of comparable magnitude to the field measurements. The major contributions to surface settlements are due to excavations for drifts 4 – 8.
- 2) The presence of the grouting gallery reduces the computed settlements on the west side of the cavern compared to the east side.
- 3) Uncertainties in the initial K_0 condition have a significant influence on the predicted surface settlement. Reducing K_0 from 1.0 to 0.5 increases the maximum surface settlement from 60mm to 115mm.
- 4) The assumed elastic shear stiffness affects the magnitude of surface settlement, but the shape of settlement troughs is largely unaffected. A normalized settlement trough $\delta_v / \delta_{v_{\max}}$ can describe results for different values of elastic soil modulus.
- 5) The effect of lining stiffness is minimal in these simulations because excavation steps achieve full-stress release, and lining elements only affect subsequent changes in drift geometries.
- 6) Uncertainties in the internal friction angle of the Middle Zone soil can affect predictions of settlements by 10 – 20 mm.

- 7) The weight of concrete in-filled in the drifts contributes roughly 35% of the surface settlement, while the stiffness of the concrete should have no effect on the predicted ground movement based on the calculations with different lining stiffness.
- 8) Limited laboratory data suggest that the Middle Zone soil can exhibit a larger friction angle in the triaxial extension mode than the triaxial compression mode. The Drucker-Prager yield criterion can describe this variation of friction angle better than Mohr-Coulomb.
- 9) Using the Drucker-Prager or Mohr-Coulomb yield criteria makes little difference on computed ground movement when the strength parameters are matched in plane-strain condition.

5.4 Three-Dimensional Finite Element Analyses with FETI_FEM

5.4.1 Finite Element Mesh and Domain Decomposition Strategy

The three-dimensional finite element mesh is prepared by extruding the plane-strain model shown in Figure 5-12 with an extrusion length of 147.6m, corresponding to the distance between the North and South access shafts for the stacked-drift cavern.

The two-dimensional decomposition with 12 sub-domains is shown in Figure 5-26, with the following considerations (these all attempt to reduce the iteration count for solving interface problems:

- 1) The artificial boundaries (interfaces between sub-domains), whenever possible, should not coincide with interfaces of different materials, such as interfaces between soil layers and interfaces between soils and linings. Overlying material boundaries with artificial boundaries deteriorate convergence rate of the iterative interface solution
- 2) The aspect ratio of sub-domains should be close to unity. The aspect ratio requirement not only helps to reduce the iteration count, but also improves the accuracy of rigid body extraction using factorization.

- 3) Figure 5-27 shows two possibilities for decomposing sub-domains with lining elements. The decomposition in Figure 5-27(b) reduces the number of interface unknowns and improves the homogeneity of the interface property. As a result, the iteration count is reduced by more than 50% based on numerical experiments.

Once the two-dimensional decomposition is determined, the mesh is extruded with 41 layers of elements, each layer is 3.6m in length, and every sub-domain has three layers of elements. Therefore, 168 sub-domains (12×14) constitute the entire 3-D finite element model, as shown in Figure 5-28.

The resultant global finite element model contains 30,053 20-noded 2nd-order elements, 126,494 nodes, and 379,842 degrees of freedom¹.

5.4.2 Simulation Sequences and Assumptions

The construction history of the stacked-drift tunnel shown in Figure 5-4 is closely modeled using 60 steps shown in Table 5-16, including the initial equilibrium step (to verify the initial stress condition and remove all lining elements) and grouting gallery excavation. The number of steps is determined by limiting the maximum length of excavation in any single drift to be less than 25m (and hence each drift requires at least six steps to be tunneled through completely). The length is chosen to avoid lengthy computation time associated with detailed modeling of every single excavation step, while preserving reasonable representation of the complex construction history.

The material properties shown in Table 5-10 are used for the three-dimensional calculation. The yield of the Middle Zone soil is described with the Drucker-Prager yield criterion that shows variable Mohr-Coulomb friction angles in different modes of shearing due to different shape of the yield function. Matsuoka-Nakai yield criterion, which uses identical parameters with the Mohr-Coulomb and predicts the same friction angles in triaxial

¹ The estimated amount of memory (using data-check option in ABAQUS) needed to solve the finite element model with ABAQUS 6.2 is 3.1GB of physical memory to solve with minimal I/O overhead, and at least 1.55GB in order to solve the finite element model initially. Due to the introduction of lining elements at later stages of construction, more physical memory may be needed.

compression shearing and triaxial extension shearing as Mohr-Coulomb, is used for describing the yield for the Upper Clay soil.

5.4.3 Computation Time and Solution History

Figure 5-29 summarizes the solution history of the three-dimensional finite element model shown in Figure 5-28. The model is solved using 14 computers in the **codex-hammer** cluster computer with Myrinet interconnection, and was computed in 67.3 hours.

The solution process involves 721 linear equation solves and 220,112 FETI iterations. As seen in Figure 5-29, the number of FETI iterations per solve is not constant. The first step removes all lining elements and is in equilibrium condition, assuming the correct initial stress condition is imposed. Therefore, FETI solver (as any residual minimization based iterative method) requires only one iteration step to converge. Direct methods, on the other hand, will always require approximately the same amount of time to solve each step of the analyses (as the most time consuming step in the solution process is to factorize the stiffness, and the factorization time depends mostly on the matrix size).

The iteration count for each solve steadily increases between step 1 and 10, and then continue to increase at a slower rate. The increase in the number of iterations per solve is due to the increasing heterogeneity in the ground caused by both the introduction of linings (that are stiffer than the surrounding ground), and the plasticity that reduces the stiffness of the near-field soil. In addition to changes of stiffness, the change of geometries of the sub-domains can also contribute to the increase of iteration count per solve due to the changed geometry and aspect ratio.

The changing number of iteration steps per solve (thus the solution time) is unique to iterative methods, and the iteration count depends on both the stiffness matrix and the residual force vector. In contrast, the solution time of direct methods depends mainly on the size of the system to be solved regardless the residual force vector, thus solution time will remain nearly constant for similar sized problems.

The number of solves per step significantly increased from 4 – 5 to more than 10 – 20 solves per step after step 23, the beginning of excavation of drift 8. Drift 8 started excavation from the South access shaft, opposite to all previous excavated drifts in this construction. The

increasing number of solves (the number of iterations of Newton-Raphson process) needed to satisfy the equilibrium condition indicates increasing nonlinearity in the system.

Close correlation between numbers of non-linear solves and the total excavation length in a single step is observed in Figure 5-29 after step 20. This behavior reflects the larger extent of excavation, and larger volume of material that is initially out of equilibrium. Furthermore, both deep and shallow drifts (drift 2, 3, 7, and 8) are excavated at the same time after step 20, thus it could be more difficult for non-linear solvers to find solutions.

Based on previous experience with a three-dimensional tunneling finite element model with 36,030 nodes, the solution time for a single increment is roughly 10 minutes¹. To extrapolate the solution time for the stacked-drift tunneling construction model, Figure 4-65 can be used. The solution time based on Figure 4-65 for ABAQUS to solving a single increment:

$$T = 0.0016 \times N_E^{1.5136} \quad (5.6)$$

Thus, it can be estimated that one single increment will take ABAQUS 9,600 seconds (160 minutes or 2.6 hours) to solve, assuming virtual memory is not involved. A computation time of 80 days is estimated in order to finish the entire analysis with 721 increments (equation solves) due to 60 computation steps and nonlinearity. The actual solution time can be even longer due to the use of the virtual memory, hardware failure, power failure, etc.

It should be noted that the model for modeling the stacked-drift construction is at least one order of magnitude larger than previous finite element models for modeling underground construction (Figure 5-30) according to the MOMIS database (Mestat et al. 2004). Yet, the current finite element model still lacks enough resolution to model each individual steel set, and more analysis steps are needed in order to more faithfully model the construction procedure.

5.4.4 Computation Results

Figure 5-31 shows the incremental vertical displacements, Δu_z , between steps. (North is on the right in these figures, and ticks are 20m apart). Initially, the grouting gallery (step 2)

¹ on Geohazard with 600MHz Alpha 21164 processor and 1088MB of RAM

causes roughly 10mm of settlement on the ground surface that can be well described with plane-strain analyses. Between step 3 – step 22, it is seen that deep drift excavations (drift 1, 2, and 3) do not affect surface settlement significantly, and drift excavations form nearly perfect circular contour lines on the ground surface unless 1) headings interactions, 2) boundary effects, and 3) drift concreting are involved.

Excavation of drift 8, the shallowest drift, starts at step 23 from the south shaft (left of the figure). The incremental settlement caused by drift 8 is nearly 15mm, significantly larger than the incremental movements caused by deep drifts.

Figure 5-32 summarizes the calculated surface settlement in the longitudinal direction above the grouting gallery between the North and the South access shafts (comparable to benchmarks BM_S2 and BM_N2 in Figure 5-6). It is seen the surface settlements can be subdivided into eight stages listed in Table 5-17 based on their contribution to the surface settlement. The direction of drift excavations can be identified by observing settlement curves shown in Figure 5-32 (e.g. excavations in Stage 1 and 2 start from North to South, and excavations in the remaining stages start from South to North).

Figure 5-33 compares the settlement trough of the 2-D plane-strain analysis conducted in Section 5.3 and the full 3-D analyses. It is seen from Figure 5-33 that the settlement trough of 2-D plane-strain calculation with 100% load reduction encloses all settlement troughs (from different sections) calculated using full 3-D analyses, thus 2-D analyses yield conservative estimation of ground movement in terms of vertical surface displacement. The variation of surface settlement troughs of different sections from the 3-D analysis can be attributed to the irrecoverable deformation caused by plasticity around tunnel headings. This is because the entire construction process is represented in the numerical model using 60 discrete steps, and drifts are advanced in different rates, thus the 60-step representation results in unevenly distribution of locations of tunnel face, causing variation of surface settlement troughs in all transverse sections.

Figure 5-33 also suggests the final settlement trough from 3-D finite element calculation can be well bracketed by 2-D plane-strain calculations with load reduction factors between 80% and 90%. With reasonable assumed load reduction factors, plane-strain analyses can achieve

reasonable agreement with full 3-D calculation with similar trough shape, trough width, and magnitude of settlement. The result also suggests that with appropriately assumed load reduction factor, 2-D plane-strain analyses with the load-reduction technique can be applied to predict surface movements for complex construction sequences.

Similar significance of concreting as in Section 5.3.5 is also seen in the three-dimensional finite element analyses. Figure 5-34 presents the volume of excavation, concreting, and the ground loss versus time. It is seen the ground loss is closely linked to the volume of concrete infilling, and is less correlated with the excavation itself. Between step 12 and step 26, the volume loss is decreasing while the concreting is not performed. It is also seen that the increase of concreting volume usually accompanies an increase of volume loss ratio.

5.5 Comparisons of Computed and Measured Surface Settlement

Figure 5-35(a) and (b) compare the computed and the measured surface movements versus time from surface benchmarks in nine groups introduced in Figure 5-6. Figure 5-35 also includes the construction activities (excavation and concreting of drifts, and compensation and consolidation grouting) either beneath (BM_S1, BM_SN, BM_N1, BM_S2, BM_N2) or near by (BM_S3, BM_N3, BM_4, BM_5) the benchmarks. The predictions are compared with measurements at locations north of Paseo de Diego, Figure 5-35(a) and south of Paseo de Diego, Figure 5-35(b). The numerical results are averaged values of movements from nodes within the range of respective group of benchmarks. The following points should be noted from these figures:

- 1) For the first 200 days, the measured and computed settlements are in reasonable agreement except the benchmarks in the BM_S2 group, where there is relatively large variation of settlements (between -10mm to -40mm) due to local compensation grouting that is not simulated in the FE models.
- 2) Drifts 1E and 1W are excavated and concreted during the first 200 days, and finite element models suggest very little resultant movement on the ground surface from these activities. Numerical models also suggest that excavation of drift 1E causes minimal movement of benchmarks in BM_N2 and BM_S2 groups, as drift 1E is deep and far from benchmarks in BM_S2 and BM_N2 groups.

- 3) Between CD200 – CD280, very little ground movement is observed in either the numerical simulation or in-situ measurements (except a sudden movement of 10 – 20mm occurred due to the water pipe break, CD207).
- 4) Incremental heave of 5 – 40mm observed in BM_S2, BM_S3, BM_SN, and BM_S1 benchmarks at about CD280 is related to the consolidation grouting. This behavior is not seen in the numerical model, as grouting is not simulated in these analyses.
- 5) Between CD280 – CD400, the ground movement is mostly controlled by the consolidation grouting, as suggested in Figure 5-35. The ground heaves and settles with the amount of consolidation grouting performed; the trend is more pronounced in the southern benchmarks (BM_S1, BM_S2, BM_S3, and BM_SN) than the northern benchmarks.
- 6) After CD400, most benchmarks experience settlement. From finite element calculations, major contributions to the surface settlement come from shallow drift excavations and concreting (e.g. drift 7 and 8), while deep drifts (e.g. drift 1 – 4) contribute little to the surface movement.
- 7) Between CD450 – CD480, all benchmark groups in Figure 5-35 show downward movement. Groups BM_4 and BM_5 are relatively far from constructions, and they also showed downward movements with similar magnitude to other benchmark groups. It is possible the BM_4 and BM_5 benchmarks settle because of the rigidity of the building foundations.
- 8) At the same time, compensation grouting is performed in order to compensate the settlement. It is seen from Figure 5-5 that between CD400 – CD540, the compensation grouting follows drift 4 and drift 7 excavations closely. Based on the finite element calculations, drifts 4 and 7 excavations produce the most incremental ground movement seen in BM_N3, BM_S3, and BM_S2.
- 9) Around CD500, an upward movement of magnitude of about 20mm is seen in BM_N1 and BM_N2 benchmark group. The heave is likely the product of compensation grouting, and is not predicted in finite element analyses because

grouting is not modeled. However, it seems the compensation grouting is not effective, as the ground quickly settles to its original elevation, and continues the pre-existing settlement trend.

- 10) The three-dimensional finite element analyses produce less settlement than plane-strain finite element analyses throughout the modeled construction history.

5.6 Assessment of Building Damages by Stacked-Drift Tunneling

Boscardin and Cording (1989) proposed a “deep beam model” (based on the work of Burland and Wroth, 1974) for assessing building damage due to ground subsidence. The deep beam model evaluates building damage by calculating tensile strains in a deformed beam, and considers the combined effects of bending and shearing from the beam deflection and horizontal straining. A simplified chart shown in Figure 5-36 that considers buildings with $H/L = 1$ (building height / building length) was presented by Boscardin and Cording (1989) to simplify the evaluation of damage.

Mair et al. (1996) adopted the deep beam model and suggested the need to evaluate separately a section of a surface structure that may sag due to underlying ground movement from these that undergo hogging mode, as shown in Figure 5-37. A similar design chart, Figure 5-38, was proposed with considerations for both shearing and bending tensile strain. In addition, the angular distortion β used by Boscardin and Cording (1989) is replaced by the deflection ratio Δ/L (Figure 5-38).

Both Boscardin and Cording (1989) and Mair et al. (1996) consider only tensile horizontal strain, and ignore the compressive horizontal strains, in estimating building damage. However, buildings directly above tunnel excavations can be subjected to mainly compressive horizontal strains rather than tensile horizontal strains. According to their equations, damage can then occur due to the tensile strains generated in the vertical direction (the effect of Poisson’s ratio). In this case, the simplified design charts of Boscardin and Cording (1989) or Mair et al. (1996) cannot be used, and the equations in their papers must be used instead. It should also be noted that although the equations in Mair et al. (1996) predict tensile strain in the vertical direction due compressive horizontal strains, the tensile cracks are unlike to occur or open due to gravity.

Figure 5-39 shows the surface displacements computed beneath the east-side buildings of the stacked-drift construction. It is seen these buildings are subjected to relatively complex deformation patterns. Following the recommendation of Mair et al. (1996), four inflection points are identified from the computed deformations. As a result, the building is divided into five zones of sagging and hogging, and damage potentials are evaluated independently.

The result of 3-D FE analyses is used to assess possible building damages with the damage assessment chart shown in Figure 5-40 based on Mair et al. (1996) summarized in Appendix J. The dimensional characteristics H/L of buildings is assumed to be 0.36^1 , and the stiffness ratio of E/G is assumed to be 2.6 (assuming Poisson's ratio of 0.3) to construct the chart in Figure 5-40. Both compressive and tensile horizontal strains are considered in the chart, and the assessment considers both the transversal and longitudinal sections relative to the SDT. The following points should be noted from Figure 5-40:

- 1) The cross-sectional direction of tunneling is more critical than the longitudinal direction. Most data points in the cross-sectional direction fall between “moderate” to “severe” damage categories (tensile strains in the structure exceeds 0.15%), while data points in longitudinal direction falls in the category of “very slight” and “slight” (tensile strains at 0.05% and 0.075%, respectively).
- 2) In contrast to the east-side buildings, the west-side buildings above the stacked-drift constructions deform entirely in the hogging mode, and the data points for these buildings tightly clustered together in the damage assessment charts shown in Figure 5-40.
- 3) The five zones separated by inflection points in Figure 5-39 can be categorized into three groups. Group I has relatively small deflection with nearly indistinctive mode of deformation; Group II is distinctively in sagging mode with large deflections, and Group III is in hogging a mode also with large deflection. These three groups correspond to the three clusters of data points in Figure 5-40 for east-side buildings.

¹ Buildings above the Río Piedras station are 2-3 floors, thus 9m is assumed for the building height. The length of these buildings is roughly 25m.

Group I corresponds to the data points clustered close to the negative horizontal axis with both hogging and sagging modes; Group II corresponds to the data points in the sagging mode with deflection ratio Δ/L over 0.5%; Group III corresponds to the data points in the hogging mode.

- 4) The envelopes of damage criteria in Figure 5-40 are composed of a linear portion defined by the bending tensile strains; and a non-linear portion defined by diagonal shearing.
- 5) Figure 5-40 suggests east-side buildings not only suffers moderate damage in hogging mode, but also slight to moderate damage in sagging mode. The hogging mode with tensile horizontal strains (Group III) should yield vertical cracks due to bending (as the damage envelope is linear in the entire range of tensile horizontal strain in Figure 5-40).
- 6) Group II should suffer cracking less than 45° from horizon due to combination of tensile strain in the vertical direction from horizontal strain and diagonal shearing due to building deflection.
- 7) Group I in Figure 5-39 is subjected to little deflection, and should have near horizontal crack due to vertical tensile strain from Poisson effect of compressive horizontal strains.

In this particular analysis, the buildings above the stacked-drift construction align with the construction, thus walls of the buildings are perpendicular or parallel to the tunneling direction. Figure 5-40 suggests the building damage in the cross-sectional direction is more significant than in the longitudinal direction, thus two-dimensional analysis can be used to assess potential building damages.

Figure 5-41 evaluates building damage with plane-strain finite element calculations using different stress release ratio of 85% and 100% throughout all simulated construction stages. Great similarities between the end of Figure 5-41(a) with 85% stress release and Figure 5-40 can be observed, with similar assessment of damage categories. Not surprisingly, increasing stress release ratio in plane-strain analyses elevated damage as shown in Figure 5-41(b).

However, if the building under assessment is out of alignment with the tunnel construction, than the plane-strain analyses cannot be applied because deep beam model requires deformation information in the plane of the building. Therefore, three-dimensional finite element calculations will be needed in order to define completely the deformation pattern in the plane of the building walls under assessment.

5.7 Summary

This chapter presents detailed analyses of the new Río Piedras station using 1) parametric studies with conventional 2-D FE models, 2) comprehensive 3-D FE model using the *FETI_FEM* program developed in Chapter 4, and 3) synthesized in-situ measurements and construction records.

From In-situ measurements and construction records:

- 1) At the end of the stacked-drift construction, the surface settlement ranges between 70 – 90mm in the southern half of the SDT, and 120 – 130mm in the northern half of the SDT above the grouting gallery. Most of these settlements occur after CD400, corresponding to the excavation of shallow drifts (e.g. Drifts 4 – 8).
- 2) The differences between the north section and the south section may be attributed to the spatial variation of the ground condition and/or the different construction procedure (e.g. grout injection).
- 3) Consolidation grouting causes heave of about 50mm between CD280 – CD400 on some benchmarks. The heave is observed mostly on the southern part of the construction and less observed on the northern part of the construction due to the construction sequence. The drift 2E and 2W remained opened during the consolidation grouting applied to the northern part of the construction, thus possibly accommodated the additional material (e.g. grout).
- 4) While compensation grouting system appears to be ineffective, the consolidation grouting system significantly affects the measured surface movement, and excessive amount of heave can be closely correlated to the consolidation grout injections. However, the mechanism behind these heaves remains unknown and more detailed modeling of the consolidation grouting process and grout-soil interaction is required.

From the numerical modeling:

- 1) The stacked-drift construction process is modeled with both 2-D and 3-D finite element models. The 3-D FE model is at least one order of magnitude larger than previous FE models for the underground work (Mestat et al. 2004). The solution is performed using in-house developed program *FETI_FEM* with 14 interconnected computers, and the entire analysis is completed in 67.3 hours. The estimated calculation time using ABAQUS is roughly 80 days assuming enough physical memory (at least 3.5GB) is available in the computer system.
- 2) K_0 is an important factor for considering the surface settlement of the stacked-drift construction, and 50% (115mm vs. 60mm) difference of maximum settlement is seen between $K_0 = 1.0$ and $K_0 = 0.5$. The differences are due to: 1) the change of soil stiffness due to different confining pressure; 2) more plasticity is involved with $K_0 \neq 1$; and 3) $K_0 < 1$ results in less lateral forces acting on the sides of the entire lining structure, and less lateral resistance for lining structure allowing it to develop more vertical movements on top of the structure.
- 3) Drift infilling has significant influence on the surface settlement. By ignoring the weight of concrete infilling, the magnitude of surface settlement can be reduced by as much as 30 – 50mm (115mm vs. 65mm). The ground loss, depending on the construction sequence assumed, can be varied by 0.5% due to concreting. Therefore, the performance of the construction may be improved by adopting light-weight concrete to in-fill the drifts and a construction sequence that is close to top-down design that effectively flattens the surface settlement trough and reduces the maximum settlement.
- 4) Alternative construction sequences are analyzed using 2-D FE models. The results suggest different construction sequences, combined with different infilling condition, may induce a maximum surface settlement ranging between 70mm – 150mm, and the ground loss may vary up to 1%. These analyses also suggest the actual construction sequence in the construction of Río Piedras station produces smallest

surface settlement and volume loss, ΔV_L compared to the other two construction sequences with concrete infilling.

- 5) Based on the analyses of alternative construction sequences, constructing shallow drifts 6 – 8 in early stages of the construction helps to reduce the maximum surface movement and produces a shallower and wider settlement trough than the bottom-up sequence by forming a mat-like structure with the lining of shallow drifts.
- 6) The three-dimensionally calculated ground deformation is consistently smaller than plane-strain models with full stress release, and plane-strain model with load reduction of 85% (lining is activated at 85% of stress release) provides reasonable agreement between 2-D and 3-D finite element calculations. Even though 2-D calculation with 85% stress release provides comparable results as the 3-D solution, this ratio could not have been selected rationally in advance (i.e., without the 3-D results).
- 7) The ground loss calculated from 3-D analyses varies between 1% and 1.7% throughout the entire construction, and variation of the ground loss is more strongly dependent on the volume of the concreting infilling than the volume of excavated materials.
- 8) The calculated ground deformations are used to assess building damage using the deep beam model following the procedure outlined by Mair et al. (1996). The assessment suggests moderate to severe damage is expected for buildings on both east-side and west-side of the stacked-drift cavern. It is also predicted that the rear side of the east-side buildings (i.e. further from the centerline of the construction) is more susceptible to damage than their front-side due to the different mode of imposed ground deformations (hogging vs. sagging).
- 9) Plane-strain FE models with an appropriate load-reduction factors can be appropriate for damage assessment for this project because the buildings are well aligned with the tunneling direction (either in parallel or in perpendicular to the building walls) with the tunneling direction, and transverse sections are more critical than the longitudinal sections based on 3-D analyses. Thus, the plane-strain finite

element calculation yields satisfactory assessment. If buildings are out-of-alignment with tunnels, 3-D analyses are necessary.

Table 5-1 Parameters for P-1

Soil Layer	Material Model	c [kPa]	ϕ [°]	γ [kN/m ³]	$\frac{G}{\sigma_{oct}}$ ⁽²⁾	E ⁽³⁾ [kN/m ³]	ν
UC	LE ⁽⁵⁾ -MC ⁽⁶⁾	26	24.2	17.5	90	10.4×10^3	0.1
MZ	LE-MC	24	39.5	18.5	110	40.3×10^3	0.1
LS ⁽¹⁾	LE-MC	24	39.5	18.5	110	71.7×10^3	0.1
Lining	LE	---	---	0.292	---	1.9×10^6 ⁽⁴⁾	0.3

⁽¹⁾ Identical to MZ

⁽²⁾ Based on Figure 5-14 with shear strain at 0.1%

⁽³⁾ $K_0 = 0.5$

⁽⁴⁾ Table 5-2

⁽⁵⁾ Linear Elasticity

⁽⁶⁾ Mohr-Coulomb Yield

Table 5-2 Determination of Lining Parameters for the Stacked-drift construction

1. W8x24 bending stiffness	$EI = 6890 (kN \cdot m^2)$
2. Equivalent Young's modulus in transversal plane of tunneling $b = 1.2 (m)$: spacing between steel sets $h = 0.2 (m)$: the physical thickness of linings $I = \frac{1}{12}bh^3$	$E'_{TR} = 8.6 (GPa)$
3. Bending stiffness in longitudinal direction $E_{timber} = 10000 (MPa)$ $b_{drift\ width} = 300 (cm)$ $h_{timber\ lagging} = 7.62 (cm)$	$EI = 1.106 (MN \cdot m^2)$
4. Equivalent Young's modulus in longitudinal direction $b = 3 (m)$ $h = 0.2 (m)$	$E'_{LN} = 0.553 (GPa)$
5. Weighted average stiffness Width of steel set: 20cm Spacing between steel sets: 120cm Steel set bending in trans. dir.: $W_{TR} = \frac{20}{120}$ Timber lagging for longi. dir.: $W_{LN} = \frac{100}{120}$	$E'_{avg} = W_{TR} \times E_{TR}$ $+ W_{LN} \times E_{LN}$ $= 1.9 (GPa)$

Table 5-3 Simulation Procedure for Stacked-Drift Tunnel in 2-D FE Models

Step	Lining Activation	Removal/Excavation	Concreting	CD
1	---	All linings	---	
2	---	Grouting Gallery	---	
3	G.G.	---	---	0
4	---	1E, 1W	---	
5	1E, 1W	---	---	
6	--	2E, 2W	1E, 1W	160
7	2E, 2W	---	---	
8	---	3E, 3W, 8	2E, 2W	340
9	3E, 3W, 8	---	---	
10	---	4E, 4W, 7E, 7W	3E, 3W, 8	430
11	4E, 4W, 7E, 7W	---	---	
12	---	5E, 5W	4E, 4W, 7E, 7W	545
13	5E, 5W	---	---	
14	---	6E, 6W	5E, 5W	600
15	6E, 6W	---	---	
16	---	---	6E, 6W	645

Table 5-4 Summary of Performed Parametric Studies

Case	Parameters		Remark
P-1	Table 5-1	Reference parameters	Reference configuration
P-2	Table 5-5	$K_0 = 1.0$	Effect of initial stress condition
P-3	Table 5-6	$E_{lining} = 1.9 \times 10^7$	Effect of lining stiffness
P-4	Table 5-7	$E_{lining} = 1.9 \times 10^8$	
P-13	Table 5-5	$K_0 = 0.5$ & stiffness evaluated at $K_0 = 1.0$	Effect of soil stiffness
P-5	Table 5-8	Soil stiffness @ 0.01% shear strain	
P-6	Table 5-9	Soil stiffness @ 1% shear strain	
P-8	Table 5-10	DP match MC at TXC	Effect of yield criterion
P-9	Table 5-11	DP match MC at PS	
P-10	Table 5-13	$\phi_{MZ} = 34.5$	
P-11	Table 5-14	$\phi_{MZ} = 29.5$	
P-12	Table 5-15	$\phi_{MZ} = 24.5$	
P-7	Table 5-1	Weight of concreting	Effect of construction activities
P-15	Table 5-1	Bottom-up construction	
P-16	Table 5-1	Top-down construction	

Table 5-5 Parameters for P-2 & P-13

Soil Layer	Material Model	c [kPa]	ϕ	γ [kN/m ³]	$E^{(3)}$ [kN/m ³]	ν
UC	LE-MC	26	24.2	17.5	15.6×10^3	0.1
MZ	LE-MC	24	39.5	18.5	60.5×10^3	0.1
LS ⁽¹⁾	LE-MC	24	39.5	18.5	108×10^3	0.1
Lining	LE	---	---	0.292	1.9×10^6	0.3

Table 5-6 Parameters for P-3

Soil Layer	Material Model	c [kPa]	ϕ	γ [kN/m ³]	E [kN/m ³]	ν
UC	LE-MC	26	24.2	17.5	10.4×10^3	0.1
MZ	LE-MC	24	39.5	18.5	40.3×10^3	0.1
LS ⁽¹⁾	LE-MC	24	39.5	18.5	71.7×10^3	0.1
Lining	LE	---	---	0.292	1.9×10^7	0.3

Table 5-7 Parameters for P-4

Soil Layer	Material Model	c [kPa]	ϕ	γ [kN/m ³]	E [kN/m ³]	ν
UC	LE-MC	26	24.2	17.5	10.4×10^3	0.1
MZ	LE-MC	24	39.5	18.5	40.3×10^3	0.1
LS ⁽¹⁾	LE-MC	24	39.5	18.5	71.7×10^3	0.1
Lining	LE	---	---	0.292	1.9×10^8	0.3

Table 5-8 Parameters for P-5

Soil Layer	Material Model	c [kPa]	ϕ	γ [kN/m ³]	E [kN/m ³]	ν
UC	LE-MC	26	24.2	17.5	13.9×10^3	0.1
MZ	LE-MC	24	39.5	18.5	183×10^3	0.1
LS ⁽¹⁾	LE-MC	24	39.5	18.5	326×10^3	0.1
Lining	LE	---	---	0.292	1.9×10^6	0.3

Table 5-9 Parameters for P-6

Soil Layer	Material Model	c [kPa]	ϕ	γ [kN/m ³]	E [kN/m ³]	ν
UC	LE-MC	26	24.2	17.5	2.89×10^3	0.1
MZ	LE-MC	24	39.5	18.5	11.0×10^3	0.1
LS ⁽¹⁾	LE-MC	24	39.5	18.5	19.6×10^3	0.1
Lining	LE	---	---	0.292	1.9×10^6	0.3

Table 5-10 Parameters for P-8
(Matching in Triaxial Compression Shear Modes)

Soil Layer	Material Model	c, d [kPa]	ϕ, β	γ [kN/m ³]	E [kN/m ³]	ν
UC	LE-MC	26	24.2	17.5	10.4×10^3	0.1
MZ	LE-DP ⁽¹⁾	164.8 ⁽²⁾	45.8 ⁽²⁾	18.5	40.3×10^3	0.1
LS	LE-DP	164.8 ⁽²⁾	45.8 ⁽²⁾	18.5	71.7×10^3	0.1
Lining	LE	---	---	0.292	1.9×10^6	0.3

¹⁾ Drucker-Prager
²⁾ Drucker-Prager matching Mohr-Coulomb at Triaxial Compression Mode, with $c'_{TXC} = 84$, $c'_{TXE} = 19$, $\phi'_{TXC} = 26$, $\phi'_{TXE} = 39$

Table 5-11 Parameters for P-9
(Matching in Plane-Strain Shear Modes)

Soil Layer	Material Model	c, d [kPa]	ϕ, β	γ [kN/m ³]	E [kN/m ³]	ν
UC	LE-MC	26	24.2	17.5	10.4×10^3	0.1
MZ	LE-DP	50.7 ⁽¹⁾	47.8 ⁽¹⁾	18.5	40.3×10^3	0.1
LS	LE-DP	50.7 ⁽¹⁾	47.8 ⁽¹⁾	18.5	71.7×10^3	0.1
Lining	LE	---	---	0.292	1.9×10^6	0.3

(1) Drucker-Prager matching Mohr-Coulomb at plane-strain mode with parameters shown in Table 5-1, $c = 24$, $\phi = 39.5^\circ$

Table 5-12 Matching Parameters between Mohr-Coulomb and Drucker-Prager Yield Criteria

Shearing Mode	Cohesion	Frictional
Triaxial Compression ($\delta = 0^\circ, b = 0$)	$d = c \cdot \frac{6 + \tan \beta}{3 - \tan \beta} \cos \phi$	$\tan \beta = \frac{6 \sin \phi}{3 - \sin \phi}$
Triaxial Extension ($\delta = 90^\circ, b = 1$)	$d = c \cdot \frac{6 - \tan \beta}{3 - \tan \beta} \cos \phi$	$\tan \beta = \frac{6 \sin \phi}{3 + \sin \phi}$
Plane-Strain (δ varies, $b \cong 0.5$)	$d = c \cdot \frac{3\sqrt{3}}{3 - \tan \beta} \cos \phi$	$\tan \beta = \sqrt{3} \sin \phi$

¹ Mohr-Coulomb: $f(\sigma_1, \sigma_3) = (\sigma_1 - \sigma_3) + (\sigma_1 + \sigma_3) \sin \phi - 2c \cos \phi$

² Drucker-Prager: $f(I_1, J_2) = \sqrt{3}J_2 - \frac{1}{3} \cdot I_1 \cdot \tan \beta - d$

³ σ_1, σ_3 : major and minor principal stresses

⁴ $I_1 = \sigma_{ii}, J_2 = \frac{1}{2} s_{ij} s_{ij}, s_{ij} = \sigma_{ij} - \frac{1}{3} I_1 \delta_{ij}$

⁵ For non-dilatant flow

Table 5-13 Parameters for P-10

Soil Layer	Material Model	c [kPa]	ϕ	γ [kN/m ³]	E [kN/m ³]	ν
UC	LE-MC	26	24.2	17.5	10.4×10^3	0.1
MZ	LE-MC	24	34.5	18.5	40.3×10^3	0.1
LS ⁽¹⁾	LE-MC	24	34.5	18.5	71.7×10^3	0.1
Lining	LE	---	---	0.292	1.9×10^6	0.3

Table 5-14 Parameters for P-11

Soil Layer	Material Model	c [kPa]	ϕ	γ [kN/m ³]	E [kN/m ³]	ν
UC	LE-MC	26	24.2	17.5	10.4×10^3	0.1
MZ	LE-MC	24	29.5	18.5	40.3×10^3	0.1
LS ⁽¹⁾	LE-MC	24	29.5	18.5	71.7×10^3	0.1
Lining	LE	---	---	0.292	1.9×10^6	0.3

Table 5-15 Parameters for P-12

Soil Layer	Material Model	c [kPa]	ϕ	γ [kN/m ³]	E [kN/m ³]	ν
UC	LE-MC	26	24.2	17.5	10.4×10^3	0.1
MZ	LE-MC	24	24.5	18.5	40.3×10^3	0.1
LS ⁽¹⁾	LE-MC	24	24.5	18.5	71.7×10^3	0.1
Lining	LE	---	---	0.292	1.9×10^6	0.3

Table 5-16 Simulation Steps for Three-Dimensional Finite Element Model

Step	Start Day	End Day	D1E	D1W	D2E	D2W	D3E	D3W	D8	D4E	D4W	D7E	D7W	D5W	D5E	D6W	D6E	Total Length of Excavation
			Start from north shaft								Start from south shaft							
1	0	0	Initial equilibrium															0.0
2	0	0	Grout gallery excavation															147.6
3	0	26	25.2	7.2														32.4
4	30	51	25.2	3.6														28.8
5	52	73	25.2															25.2
6	73	87	25.2	3.6														28.8
7	87	98	25.2															25.2
8	98	116	21.6	25.2														46.8
9	116	129		25.2														25.2
10	129	135		25.2														25.2
11	135	151		25.2														25.2
12	151	157		25.2														25.2
13	157	199		7.2	3.6	25.2												36
14	199	214			25.2	18												43.2
15	214	225			25.2													25.2
16	225	283			14.4	25.2												39.6
17	283	292			7.2	25.2												32.4
18	292	304			25.2	14.4												39.6
19	304	309			25.2	3.6												28.8
20	309	338			21.6	21.6												43.2
21	338	354				14.4	25.2											39.6
22	354	364					25.2											25.2
23	364	372					25.2	10.8										36
24	372	379					25.2	10.8	14.4									50.4
25	379	382					25.2	7.2	10.8									43.2
26	382	388					21.6	25.2	10.8									57.6
27	388	397						25.2	7.2									32.4
28	397	401						25.2	3.6	3.6								32.4
29	401	408						25.2										25.2
30	408	415						25.2	14.4	10.8								50.4
31	415	434						3.6	21.6	25.2								50.4
32	434	442							25.2	14.4	18							57.6
33	442	451							25.2	3.6	25.2							54
34	451	458							3.6	3.6	25.2							32.4
35	458	479									10.8	10.8	25.2					46.8
36	479	485									21.6	21.6	25.2					68.4
37	485	493									10.8	21.6	25.2					57.6
38	493	500								14.4	28.8	21.6	21.6					86.4
39	500	507								18	7.2	28.8	25.2					79.2

Table 5-16 (Cont'd) Simulation Steps for Three-Dimensional Finite Element Model

Step	Start Day	End Day	D1E	D1W	D2E	D2W	D3E	D3W	D8	D4E	D4W	D7E	D7W	D5W	D5E	D6W	D6E	Total Length of Excavation	
			Start from north shaft						Start from south shaft										
40	507	514							10.8			25.2	21.6						57.6
41	514	532							25.2			14.4						39.6	
42	532	541							18					25.2				43.2	
43	541	544												25.2				25.2	
44	544	549												25.2				25.2	
45	549	553												25.2	3.6			28.8	
46	553	556												25.2	14.4			39.6	
47	556	565												21.6	25.2			46.8	
48	565	574											3.6		25.2			28.8	
49	574	581													21.6	25.2		46.8	
50	581	585										3.6			21.6	14.4		39.6	
51	585	591													18	25.2		43.2	
52	591	596													18	25.2		43.2	
53	596	619													25.2	14.4		39.6	
54	619	624													18	25.2		43.2	
55	624	626													7.2	25.2		32.4	
56	626	631													7.2	25.2		32.4	
57	631	634														25.2		25.2	
58	634	638														25.2		25.2	
59	638	638														7.2		7.2	
60	638	???	Final lining and concreting																

Table 5-17 Summary of Contributions of Surface Settlement from 3D Analyses

Stage	Step	Excavation	Concreting	Settlement
1	1 – 12	1E, 1W, Grouting Gallery	1E, 1W	20mm
2	13 – 20	2E, 2W	N/A	3mm
3	21 – 34	3E, 3W, 8 4E, 4W, 7E, 7W	2E, 2W, 8 3E, 3W	10mm
4	35 – 41	4W, 7E, 7W, 4E, 5W	3E, 3W, 4W, 7W	15mm
5	42 – 48	5E, 5W, 6W, 7E, 7W	4E, 4W, 7E	16mm
6	49 – 56	5E, 6E, 6W	N/A	8mm
7	57 – 59	6E	N/A	1mm
8	60	N/A	5E, 5W, 6E, 6W	10mm

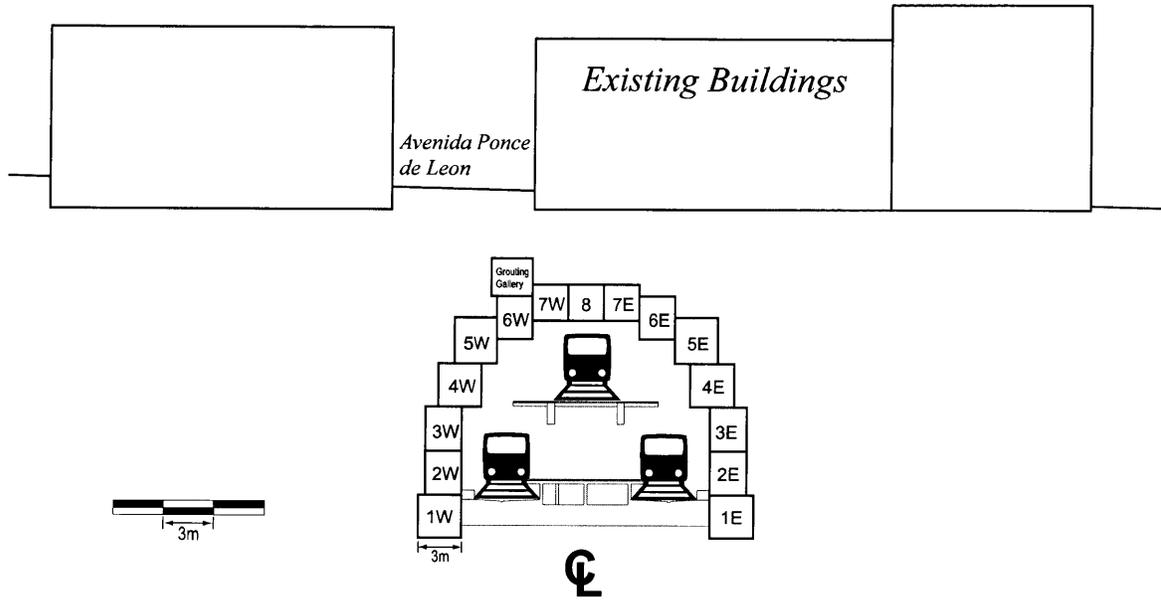
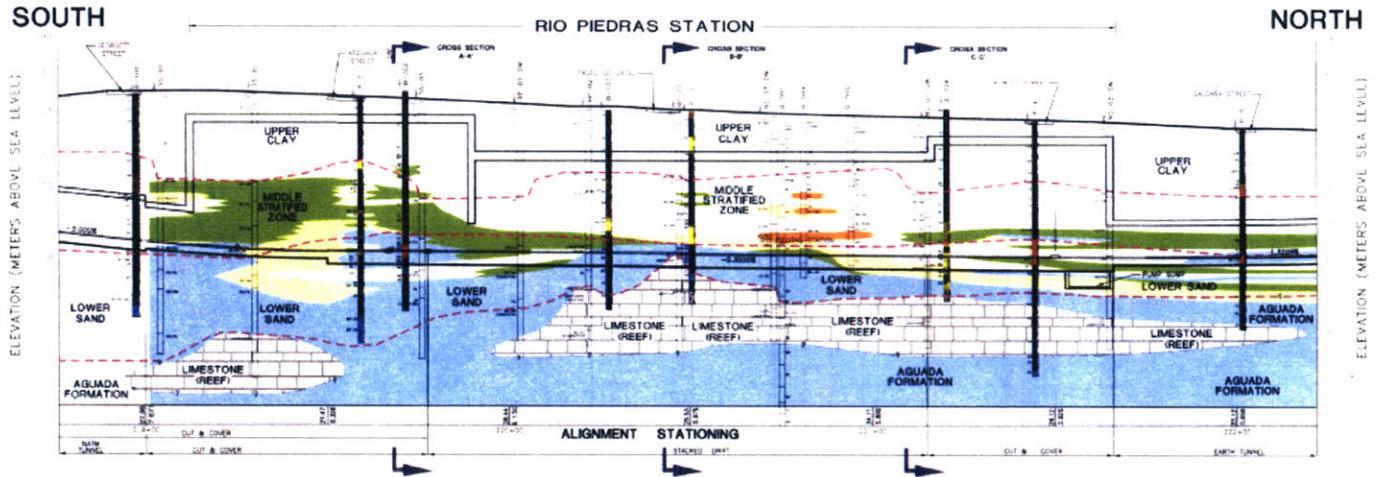
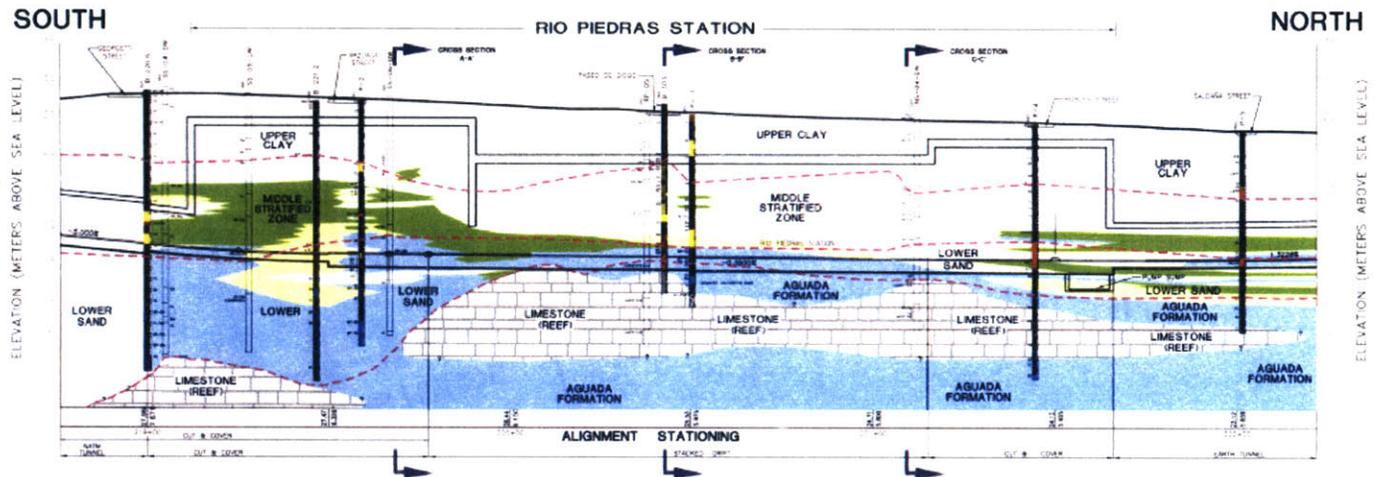


Figure 5-1 Stacked-Drift Construction for the Río Piedras Station

RIO PIEDRAS STATION BAYAMON NORTHBOUND ALIGNMENT

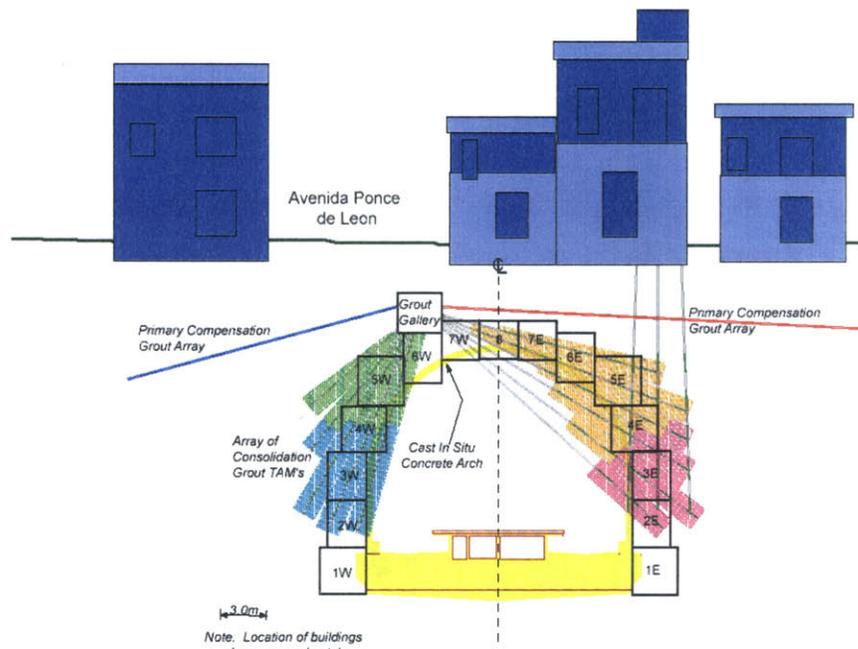


RIO PIEDRAS STATION BAYAMON SOUTHBOUND ALIGNMENT

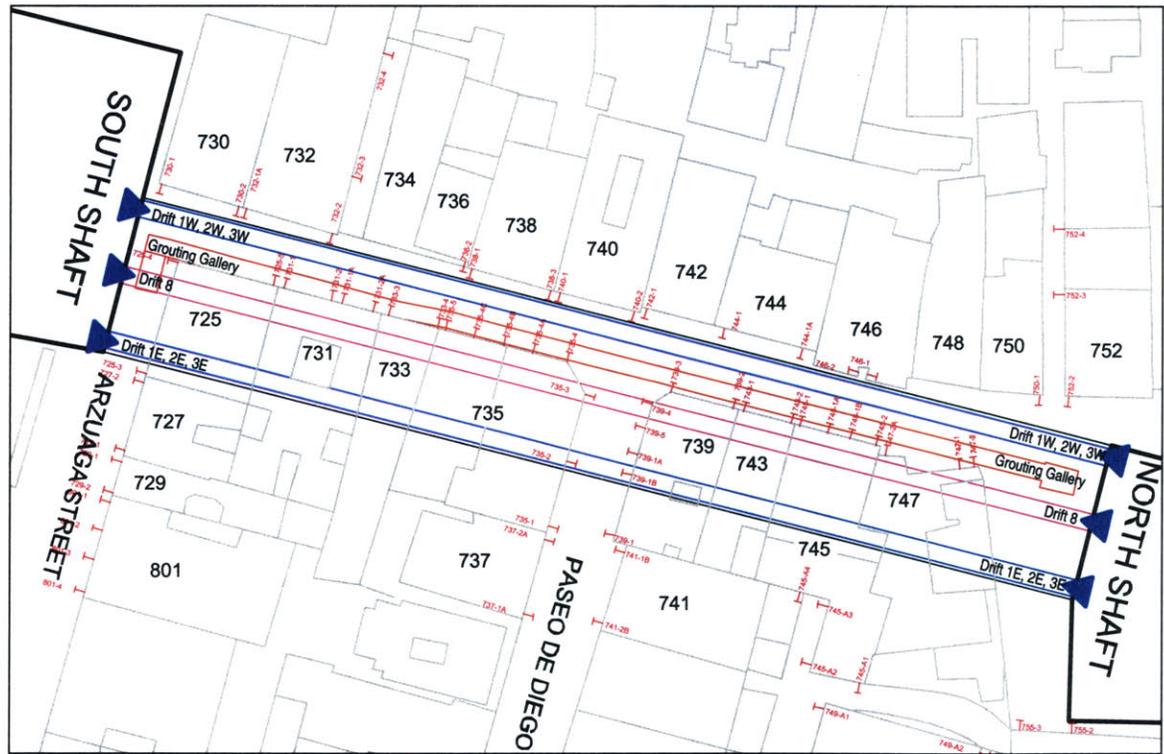


- LEGEND**
- ANTICIPATED GROUND BEHAVIOR (SHOWN ON UPPER CROSS SECTION)
 - █ FILL
 - █ FINE GRAINED IMPERVIOUS SOILS
 - █ FINE GRAINED SOILS SUBJECT TO RAVELING
 - █ COARSE GRAINED SOILS SUBJECT TO FLOWING
 - █ COARSE GRAINED PERVIOUS SOILS EASILY DRAINED BY DEWATERING
 - INTERPRETED GEOLOGIC UNITS (NOT SHOWN IN COLOR ON THIS DRAWING)
 - █ UPPER CLAY
 - █ MIDDLE STRATIFIED ZONE
 - █ LOWER SAND
 - █ AGUADA FORMATION
 - █ LIMESTONE
 - GEOLOGIC CONTACT (DASHED WHERE POORLY KNOWN)
 - BORINGS
 - BOREHOLE DESIGNATION (INCLUDES PREFIXES S, N, R, B AND P)
 - ▬ SAMPLE INTERVAL
 - USCS DESIGNATION
 - W/L HIGHEST WATER LEVEL READING
 - A/L AVERAGE WATER LEVEL READING
 - L/W LOWEST WATER LEVEL READING
 - RECORDED PENETRATION RESISTANCE (BLOWS PER FOOT) (SHOWN ON LOWER CROSS SECTION)
 - █ N<20
 - █ N=20-29
 - █ N=30-39
 - █ N=40-49
 - █ N>50
- SCALES**
- HORIZONTAL SCALE IN METERS: 0, 20, 40
- VERTICAL SCALE IN METERS: 0, 10, 20

Figure 5-2 Soil Profile for the Rio Piedras Station Site



(a) Section view (Whittle and Bernad, 2003)



(b) Top view

Figure 5-3 Stacked-Drift Construction and Surface Benchmarks of Río Piedras Station

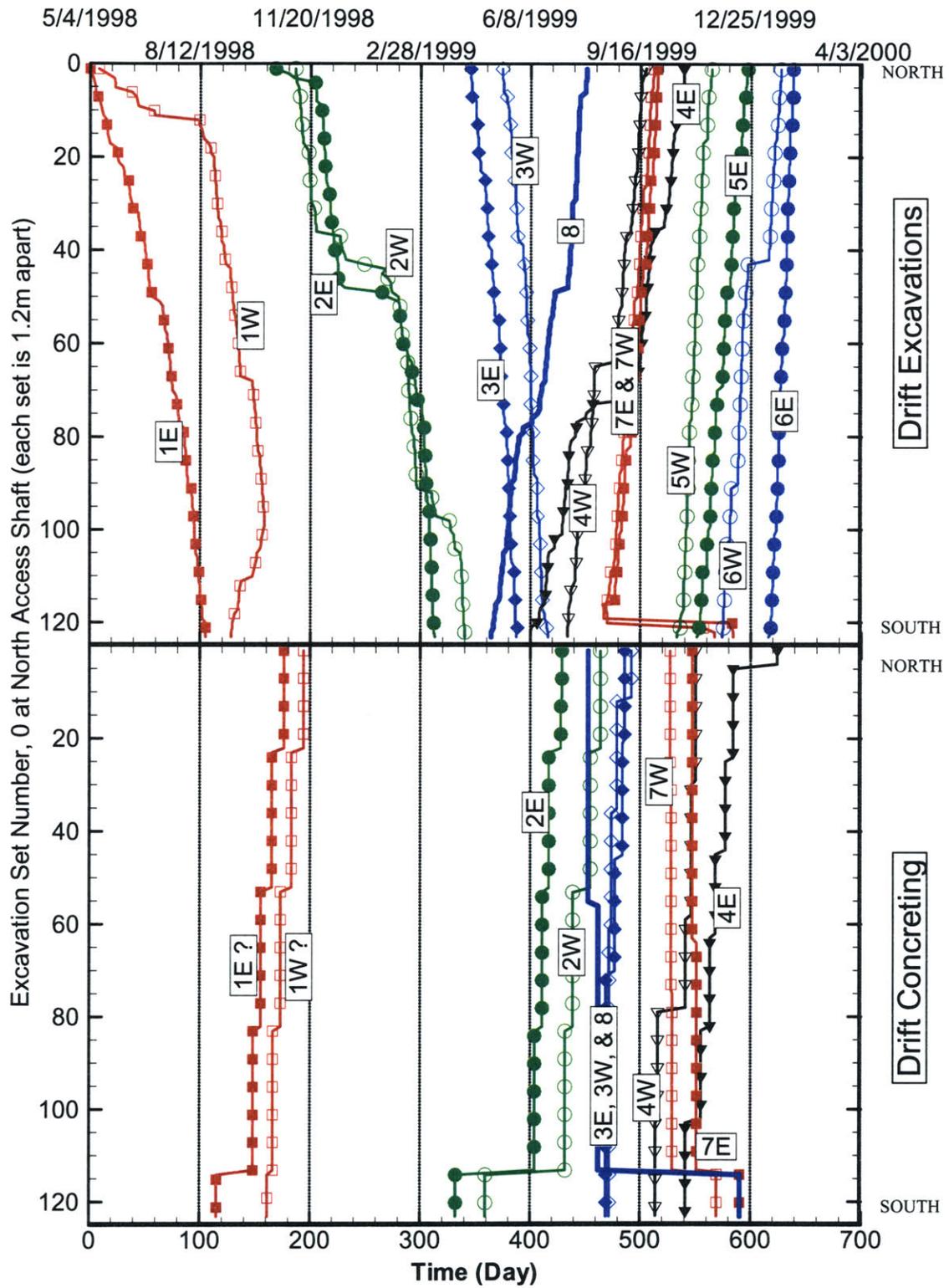
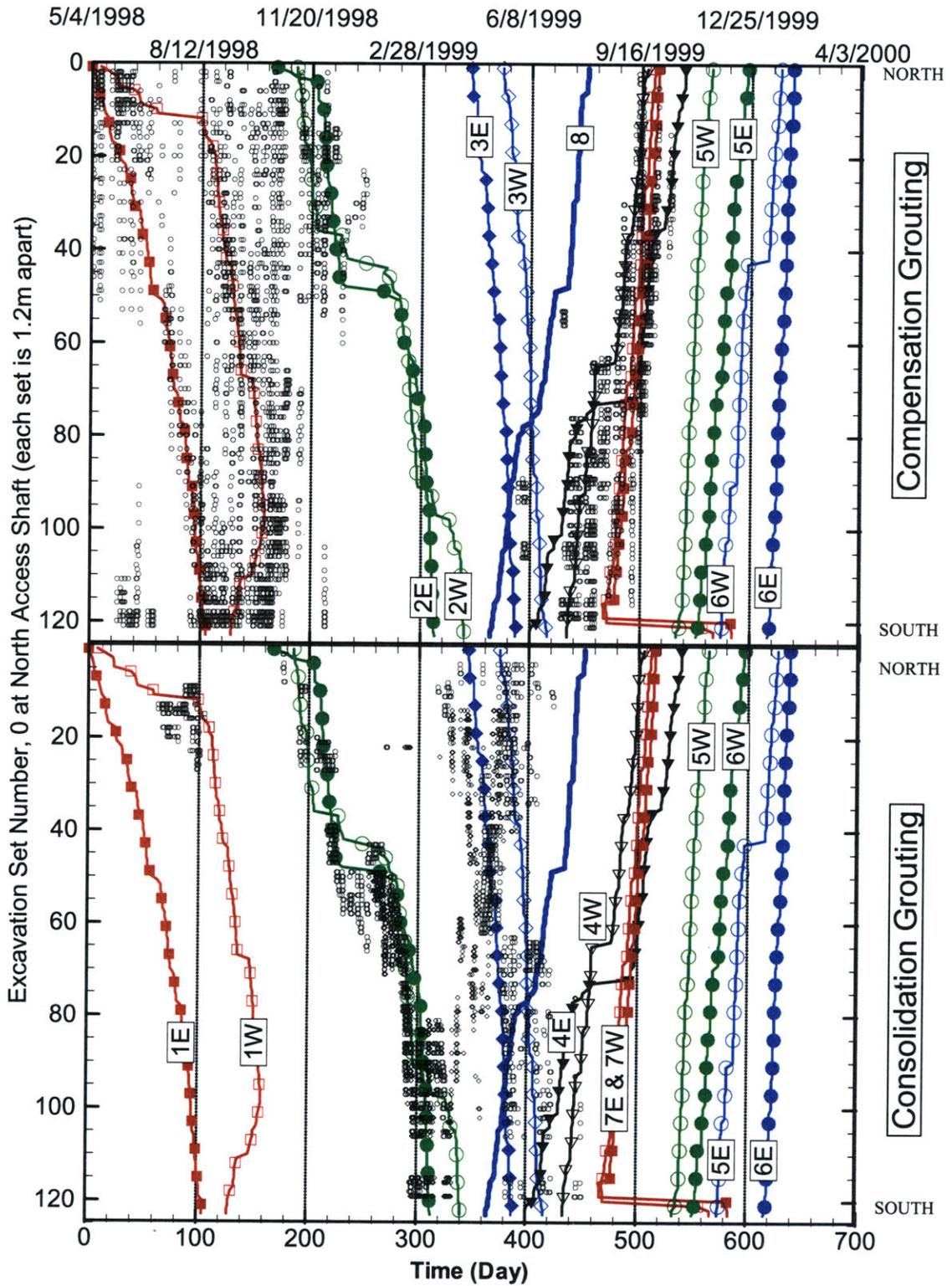
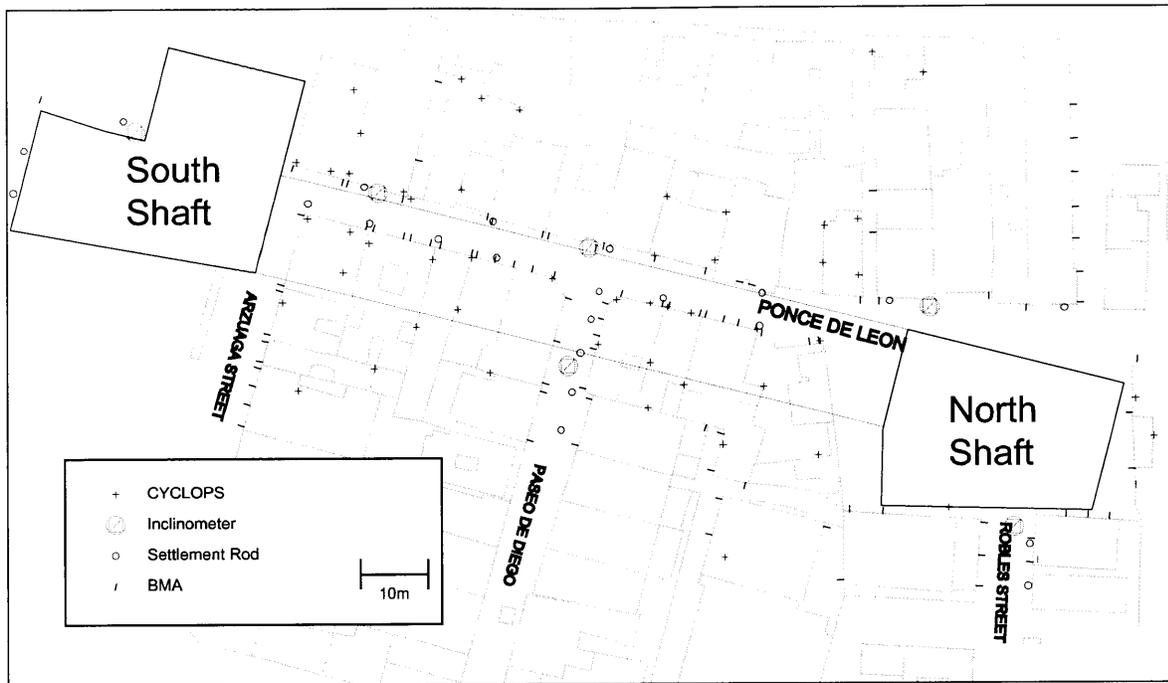


Figure 5-4 Excavation and Construction Sequence of the Stacked-Drift Construction



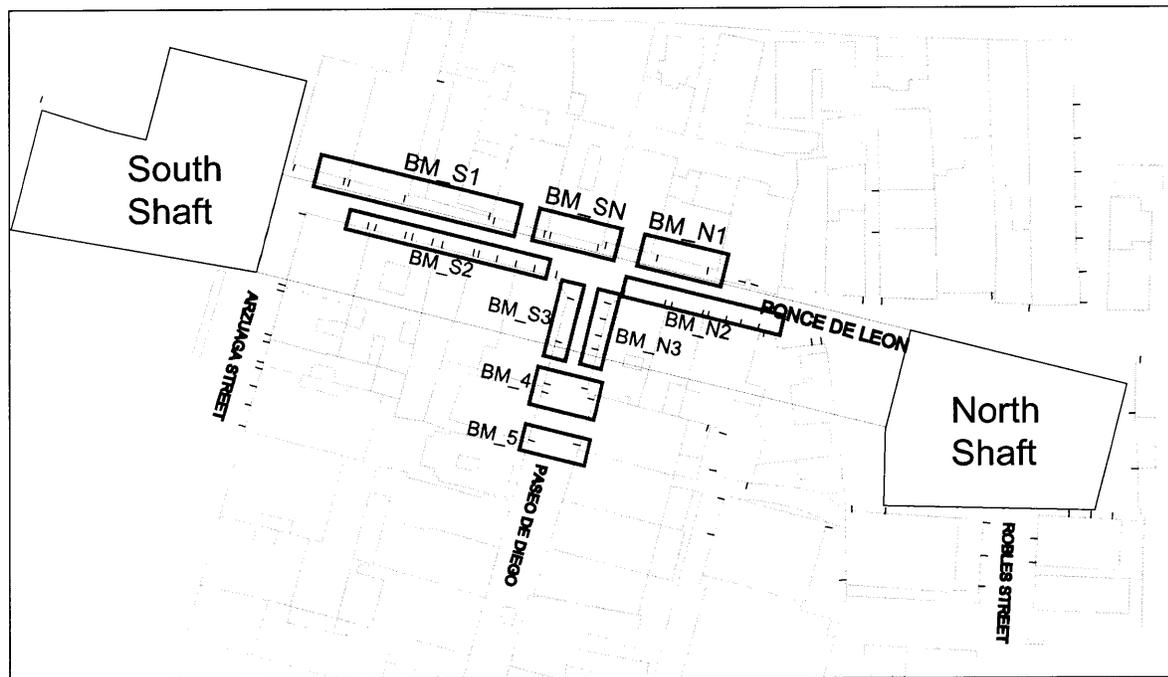
- Each black circle represents a grout injection event, including time and its location along the tunnel axis.

Figure 5-5 Grouting Activities during Stacked-Drift Construction



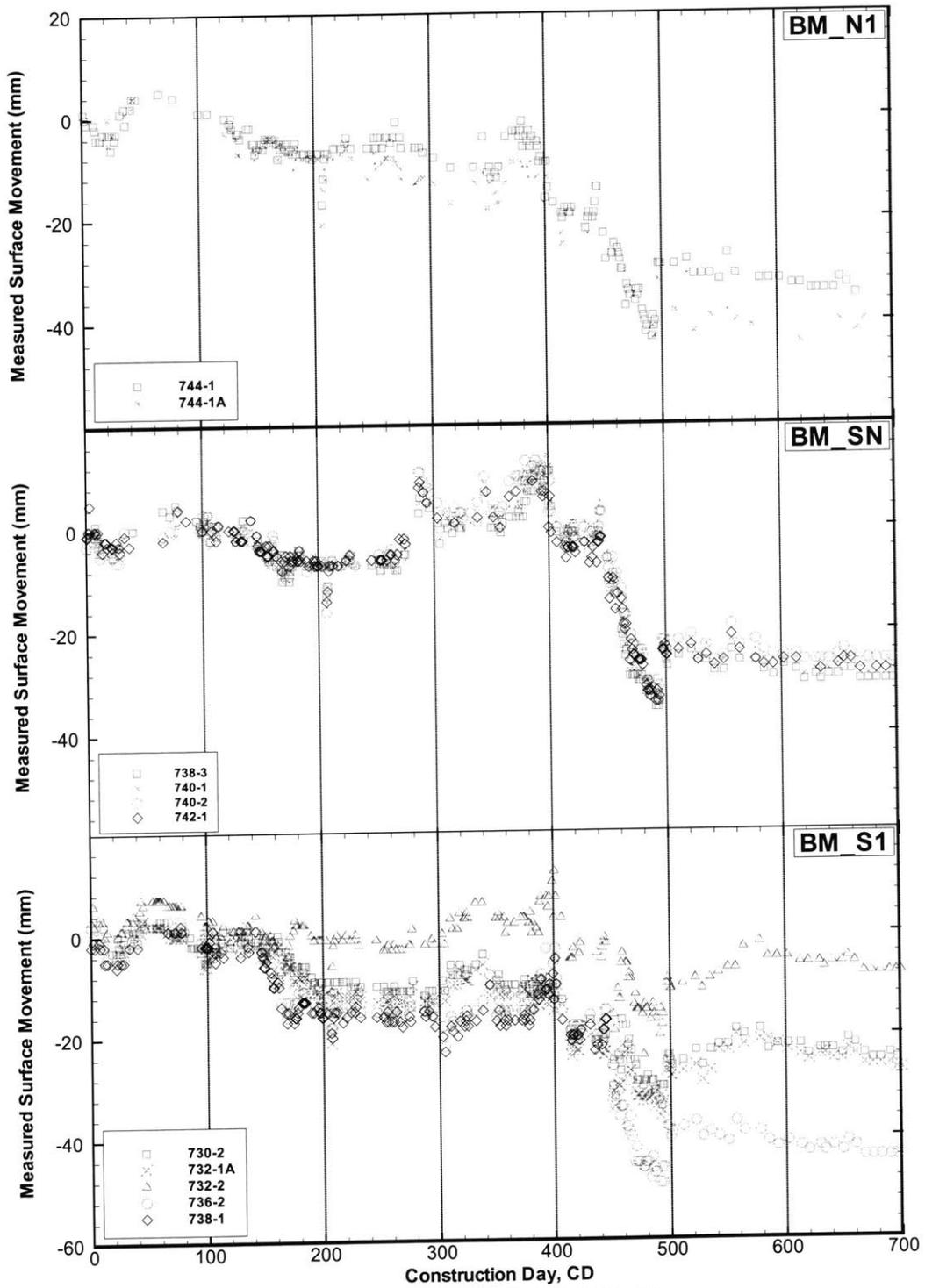
(a) All instrumentations for the SDT section

(b)



(b) Clusters of BMAs

Figure 5-6 Groups of Surface Benchmarks



(a) BM_N1, BM_SN, BM_S1

Figure 5-7 Settlement Records from Surface Benchmark Clusters
 a) Offset at 12m west of SDT centerline

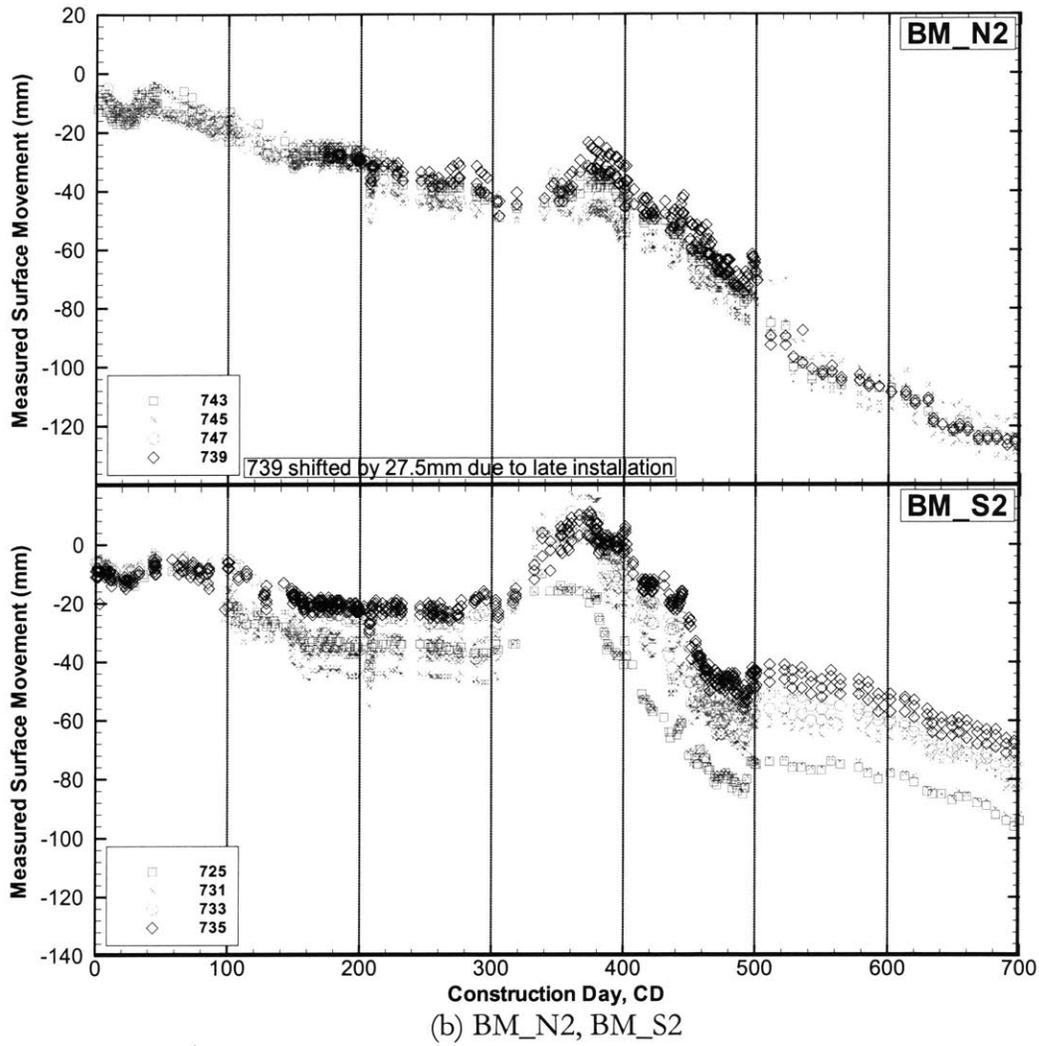


Figure 5-7 (cont'd) Settlement Records from Surface Benchmark Clusters
 b) Offset 5m west of the SDT centerline

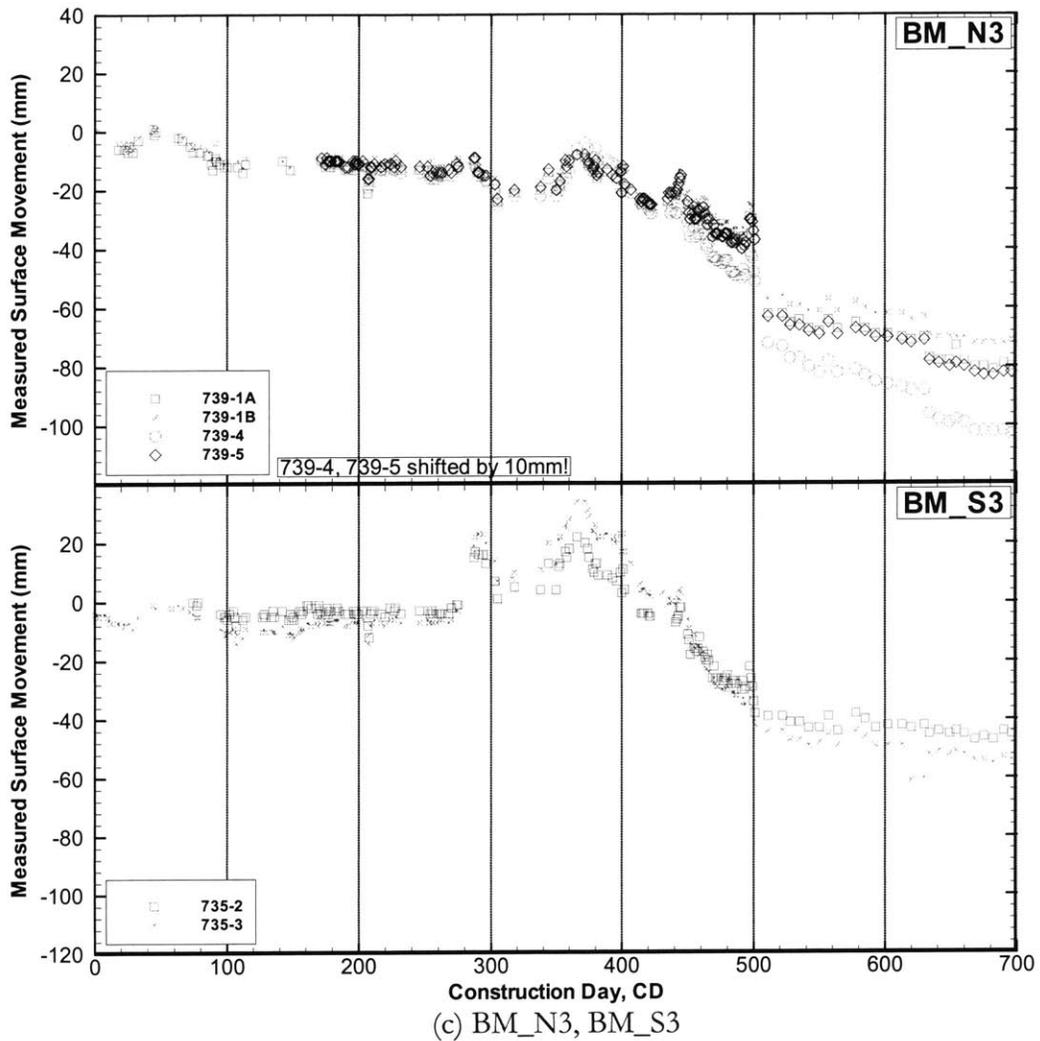
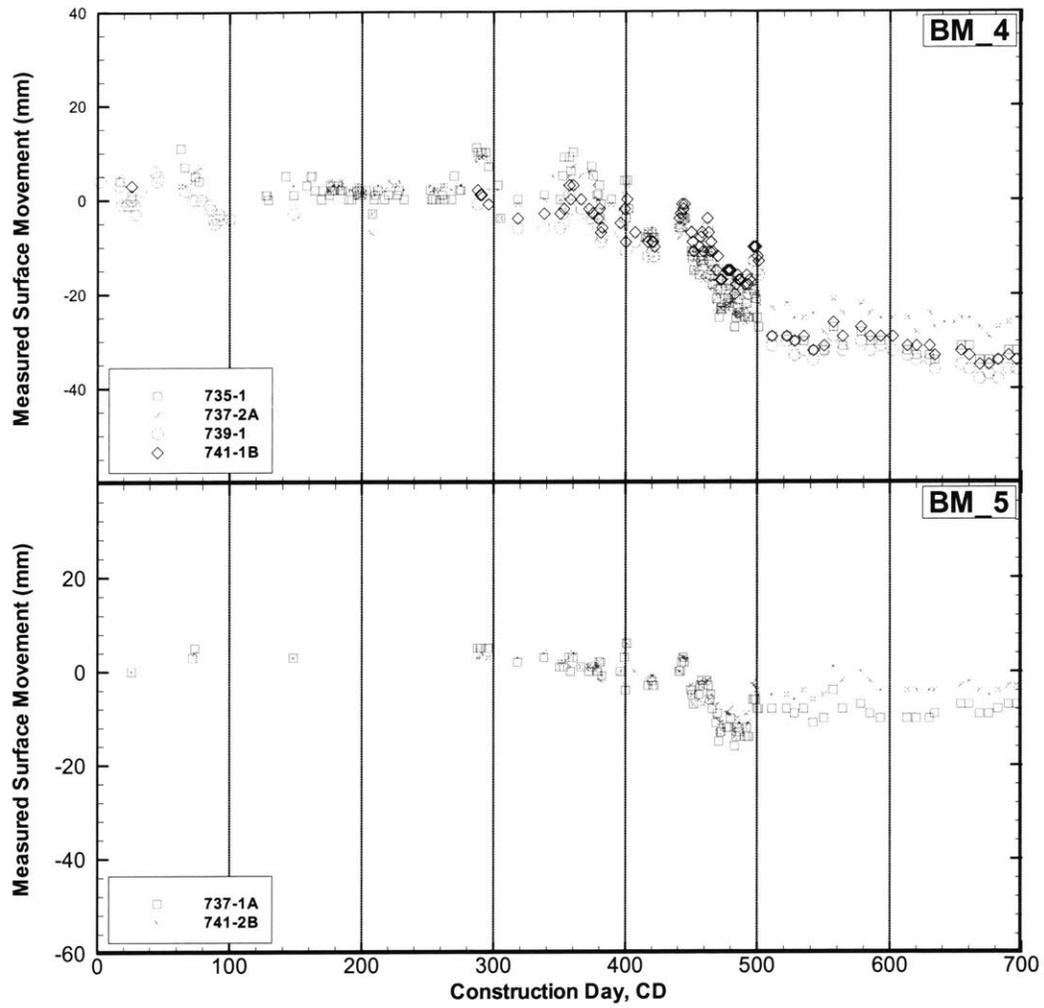


Figure 5-7 (cont'd) Records from Surface Benchmark Clusters
 c) Offset 0 – 10m east of the SDT centerline



(d) BM_4, BM_5

Figure 5-7 (cont'd) Records from Surface Benchmark Clusters
 d) Offset 20 – 30m east of the SDT centerline

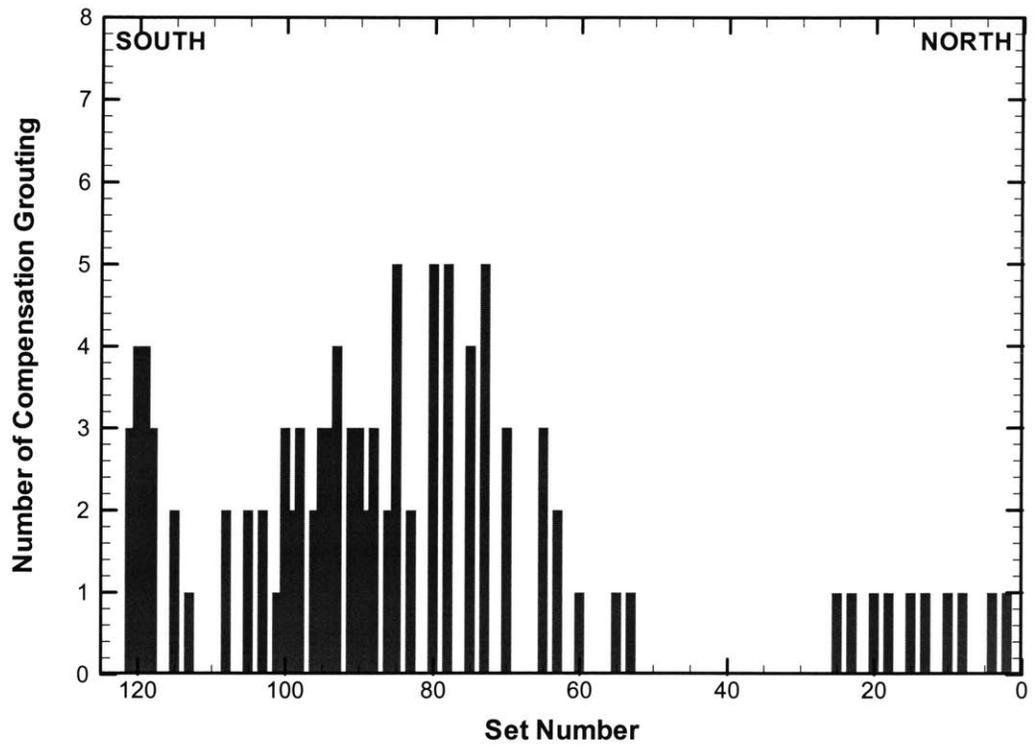


Figure 5-8 Number of Compensation Grouting Performed between CD80 – CD100

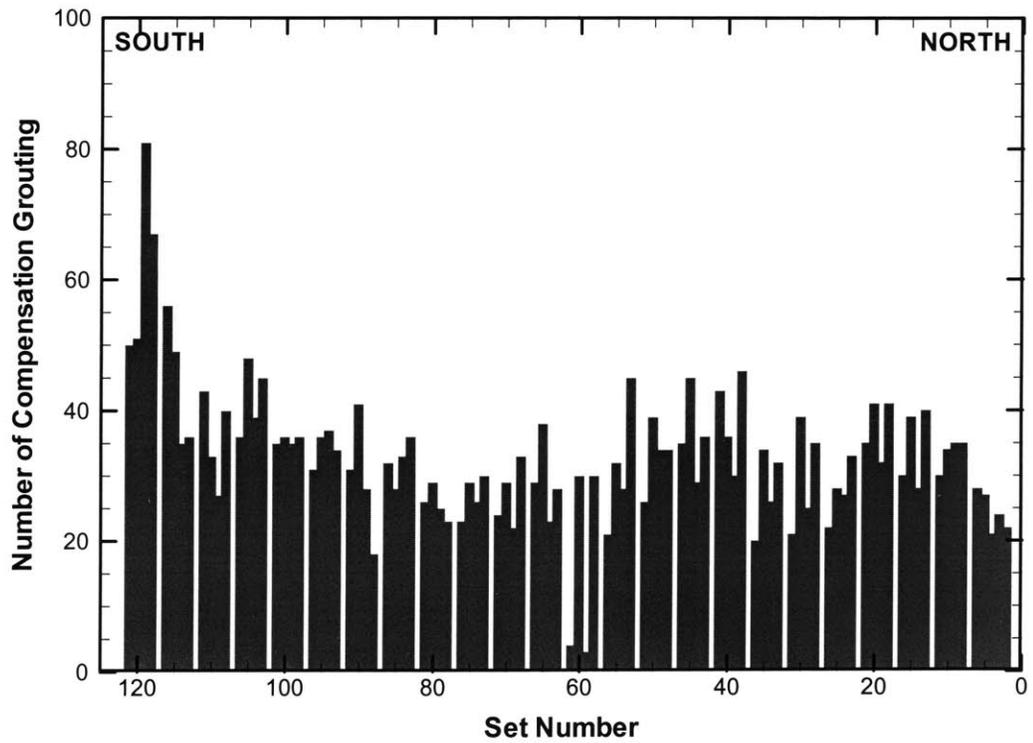
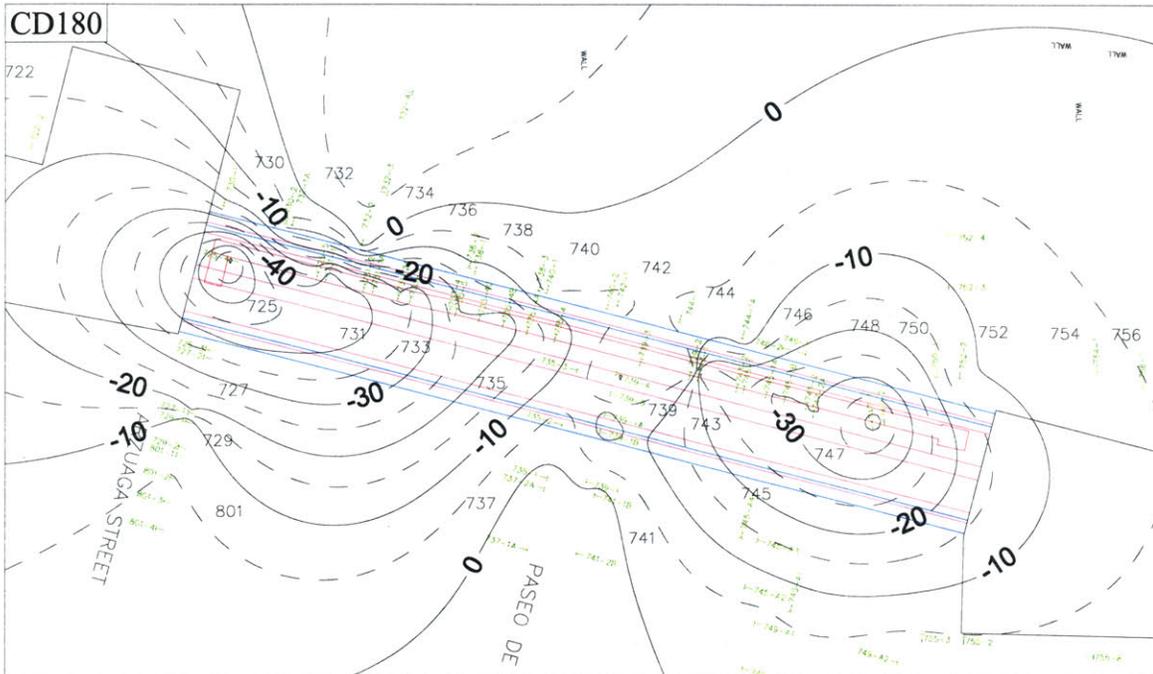
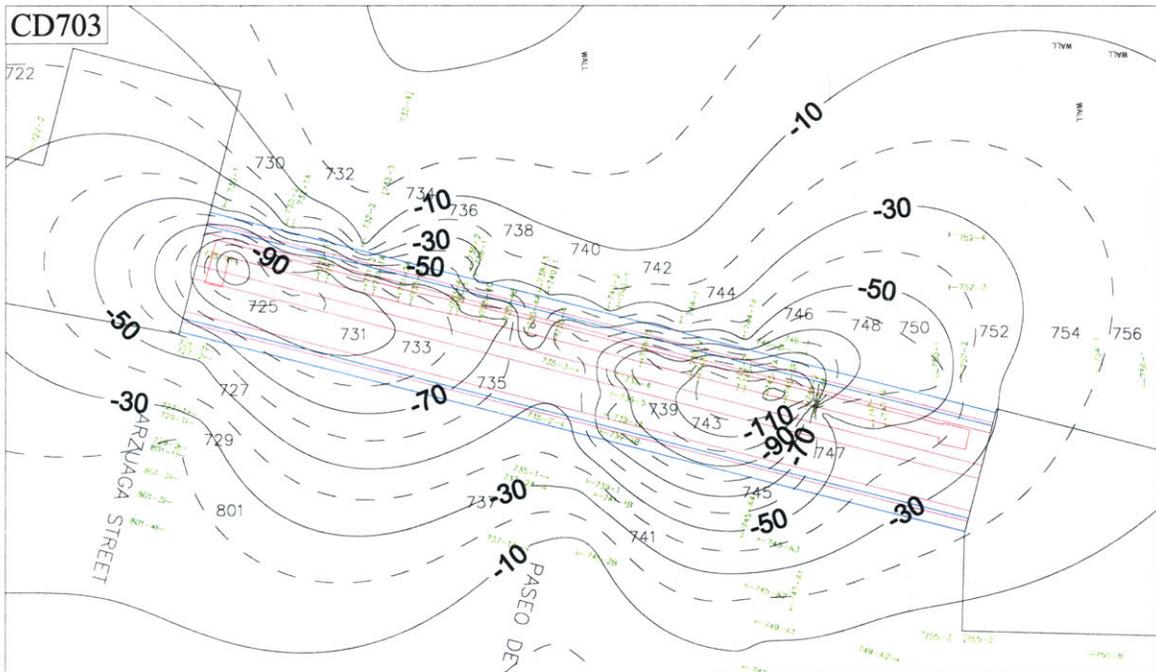


Figure 5-9 Total Number of Compensation Grouting Performed



(a) Surface Settlement Contour on CD180, End of Drift 1E and 1W Construction



(b) Surface Settlement Contour on CD703, End of Drift Construction

Figure 5-10 Measured Surface Settlement at the End of Excavation

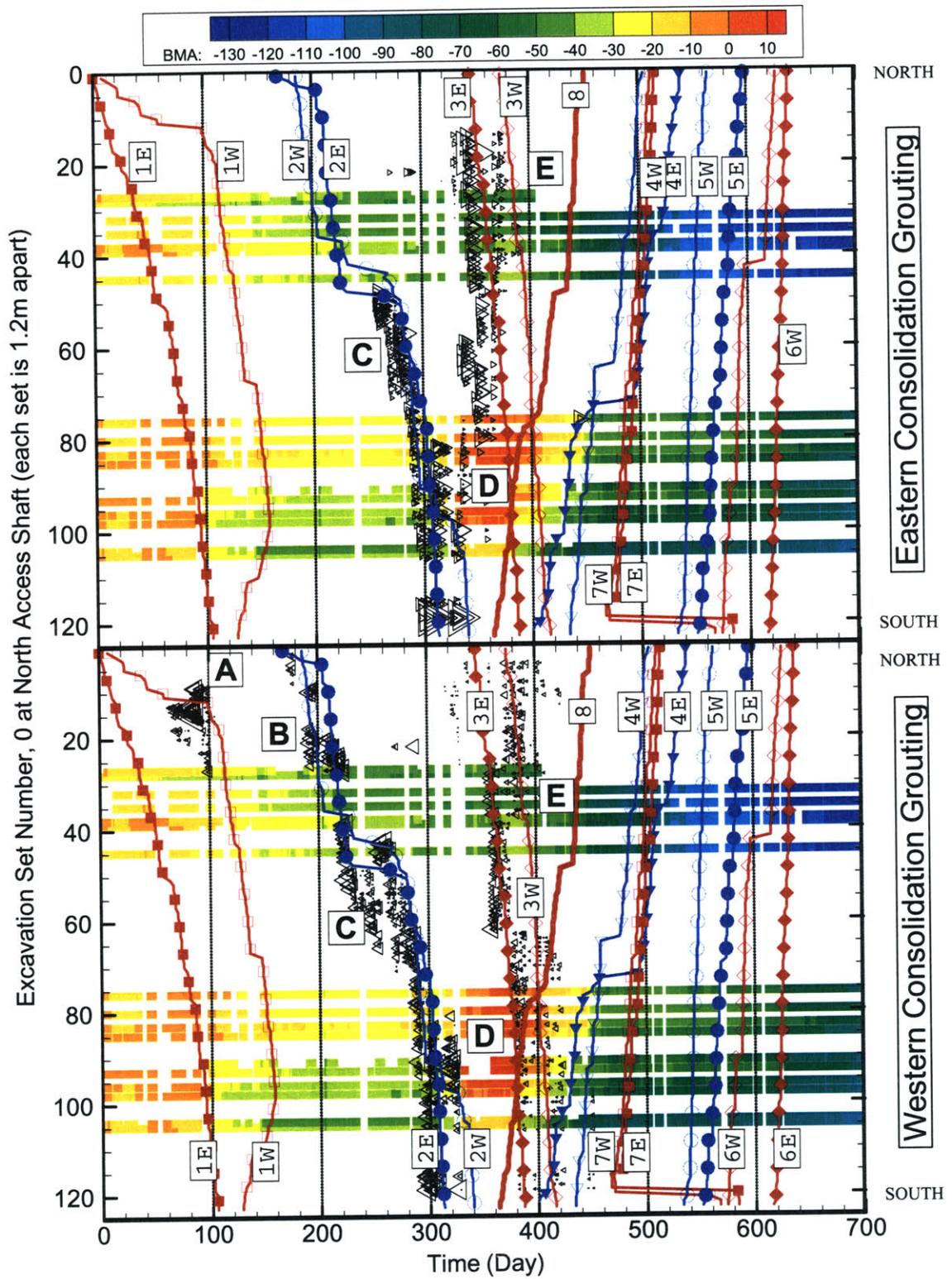


Figure 5-11 Consolidation Grouting, Excavation, and Settlement above the Grouting Gallery

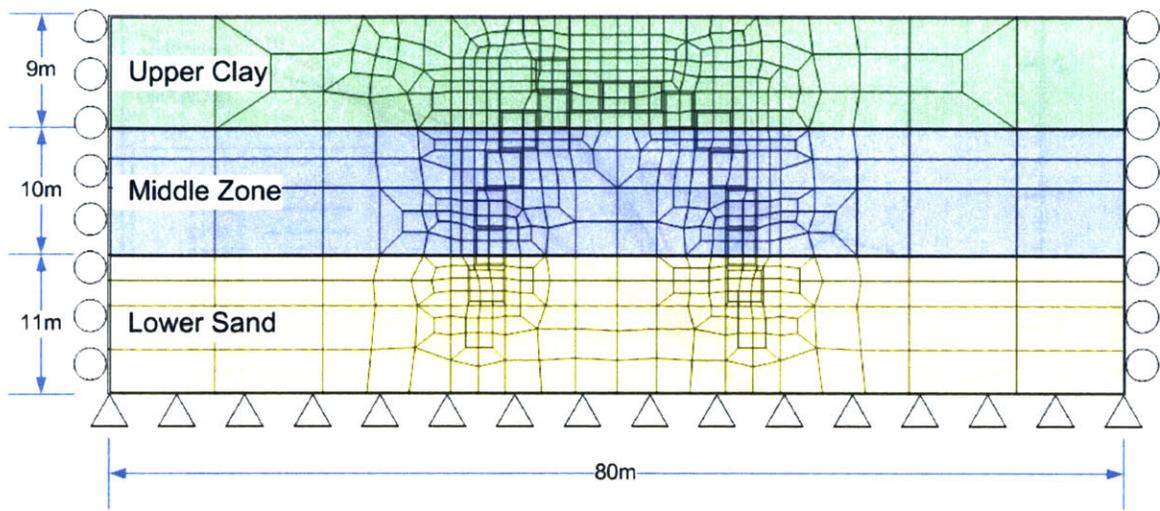
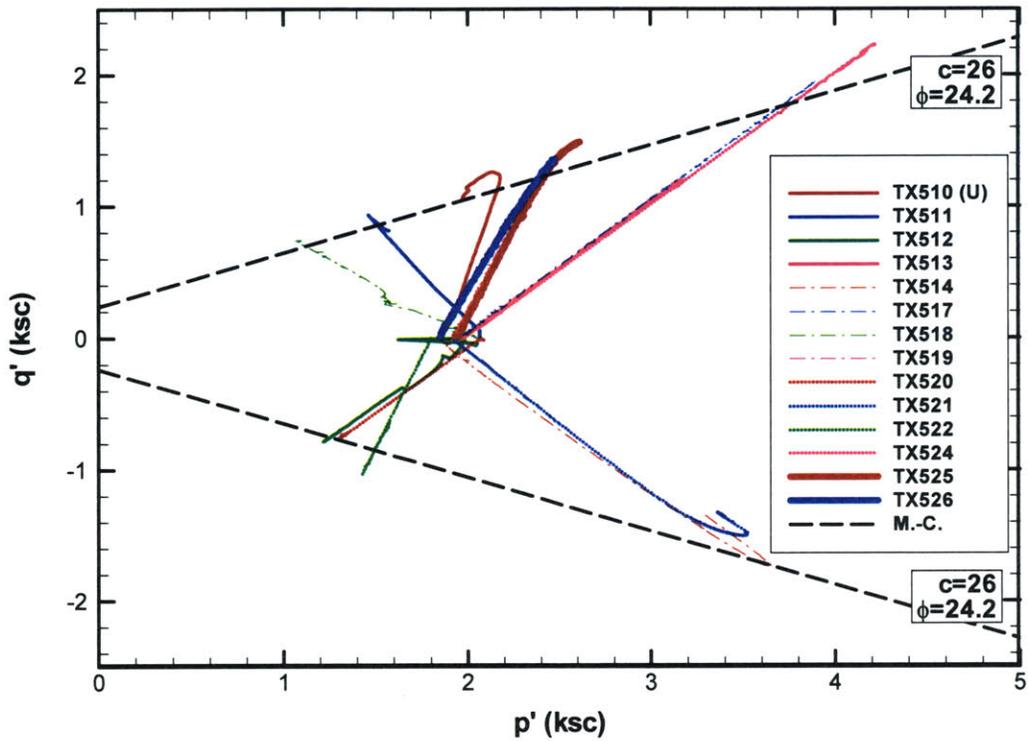
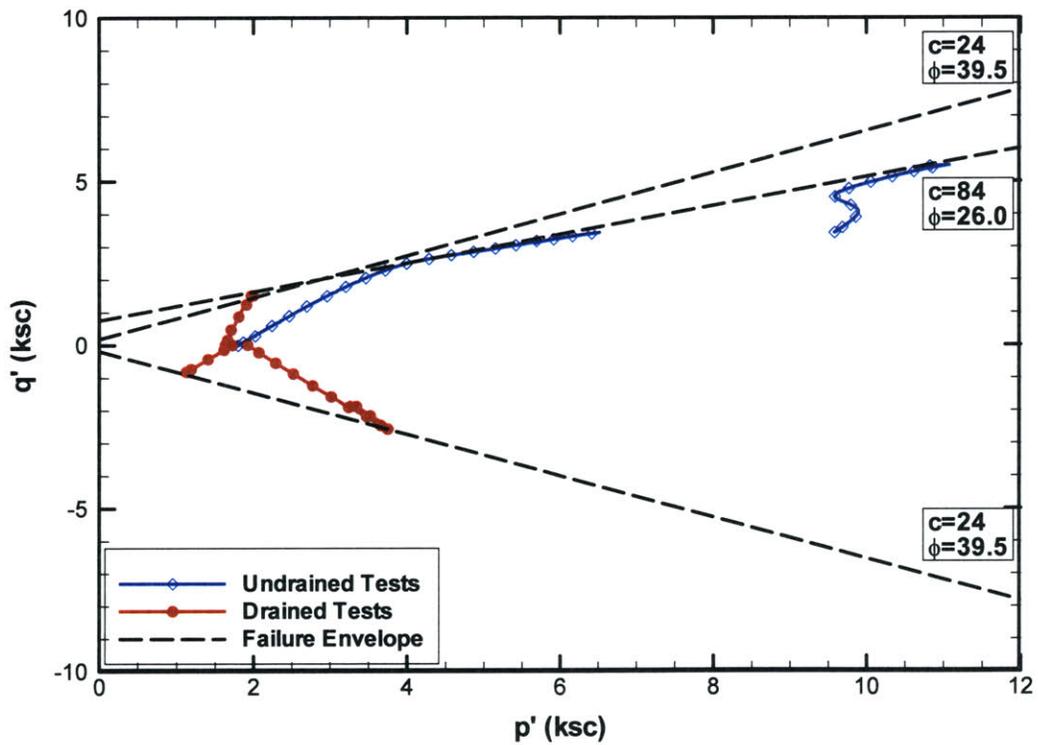


Figure 5-12 2-D Finite Element Model for Stacked-Drift Tunnel Analyses



(a) UC (data of Zhang, 2002)



(b) MZ (Data of Zhang, 2002)

Figure 5-13 Stress Paths of Lab Experiments Conducted at MIT for UC and MZ

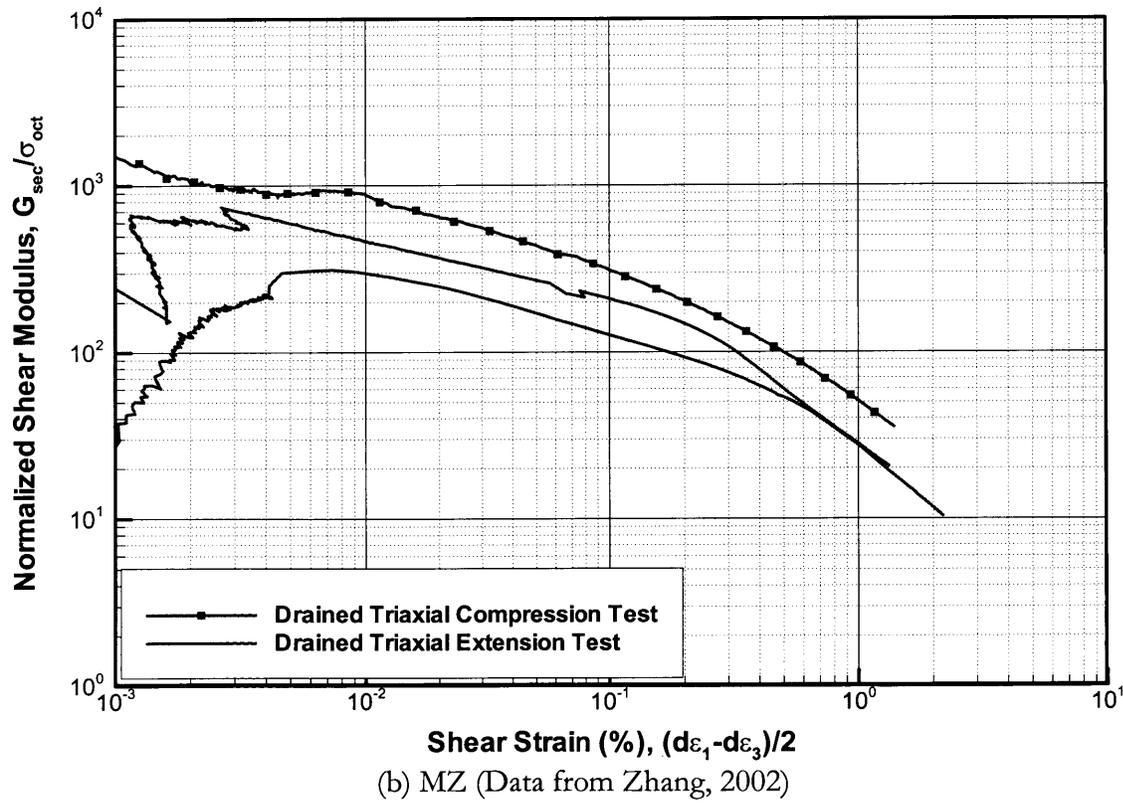
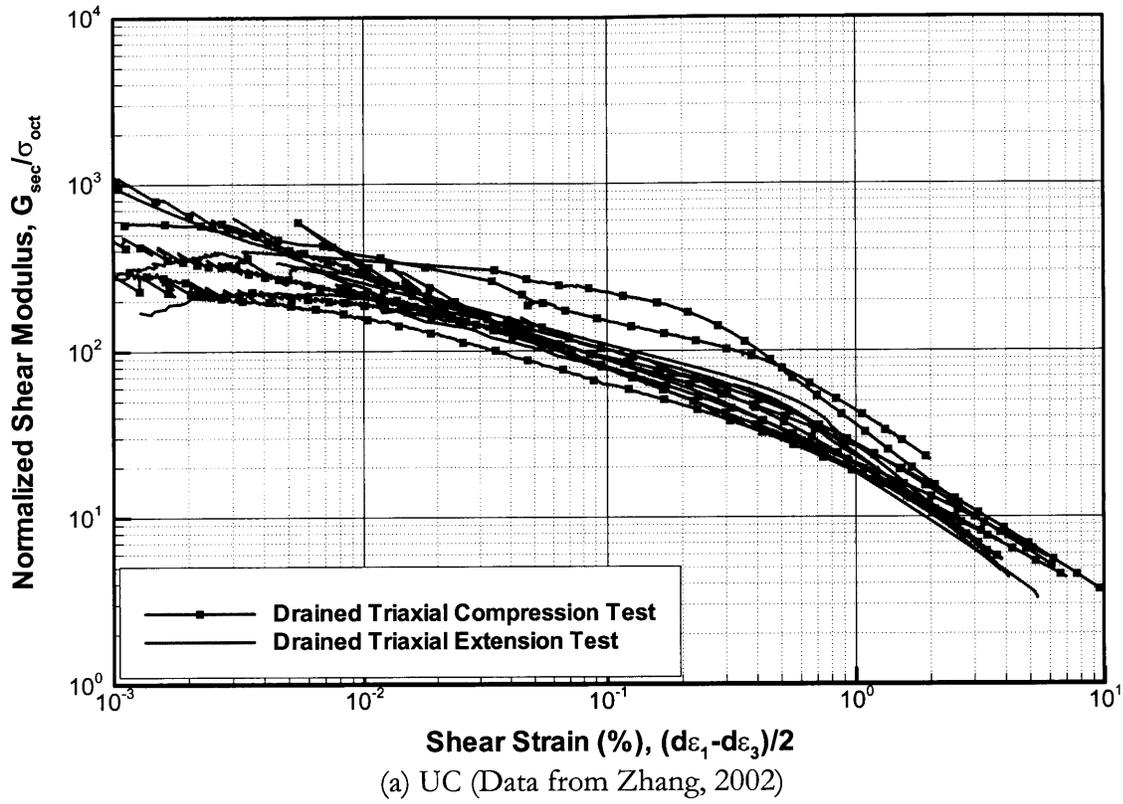
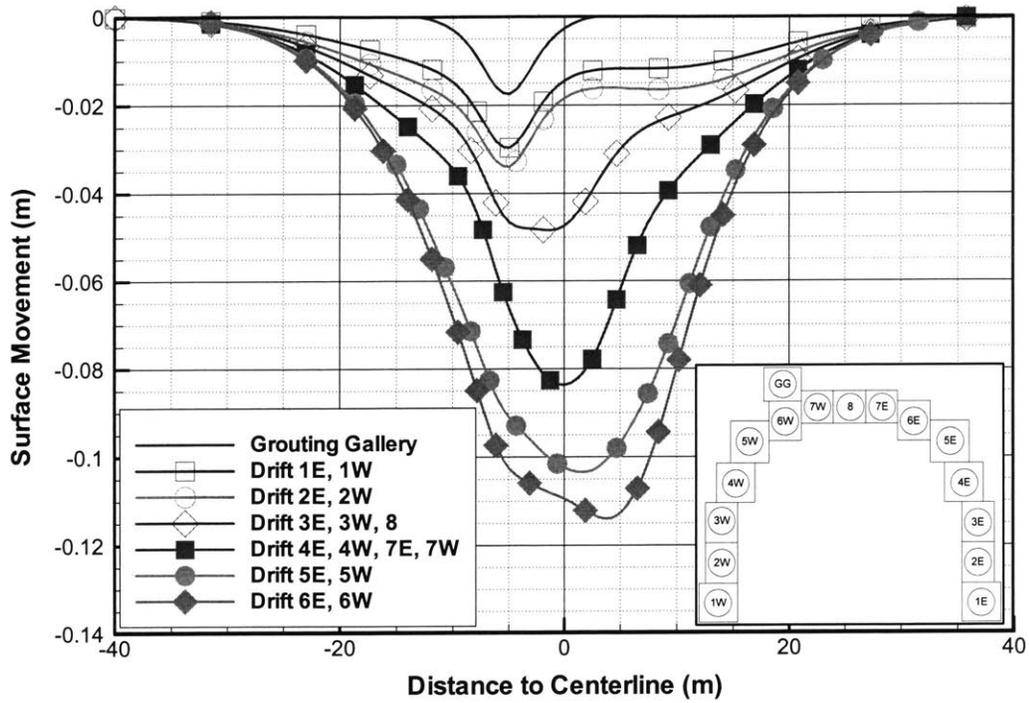
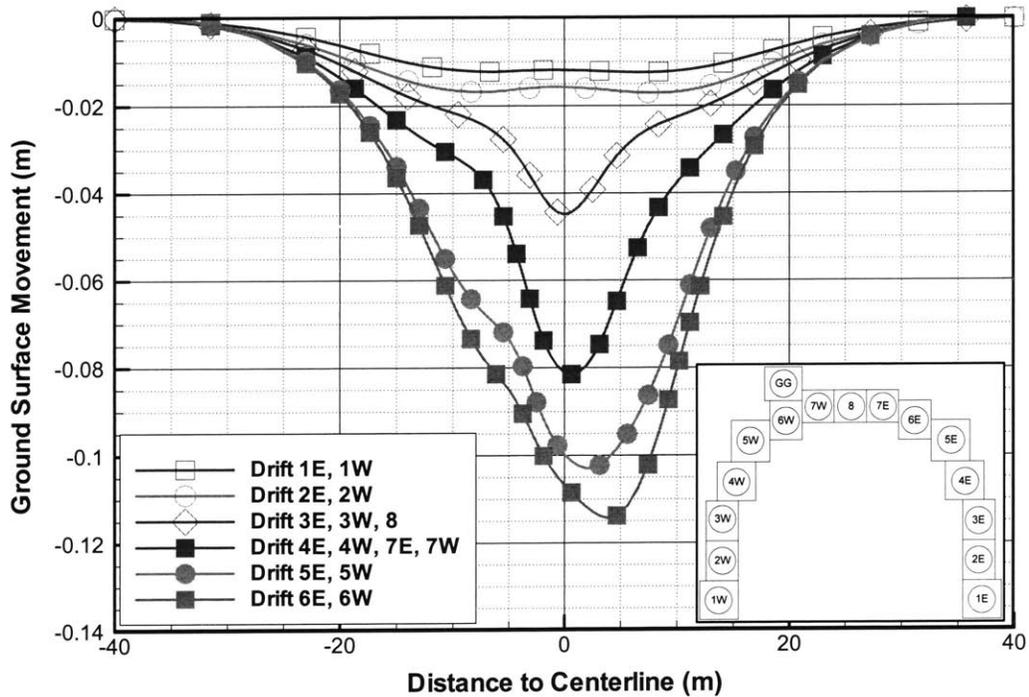


Figure 5-14 Measured Drained Shear Modulus versus Axial Strain for Middle Zone Soil

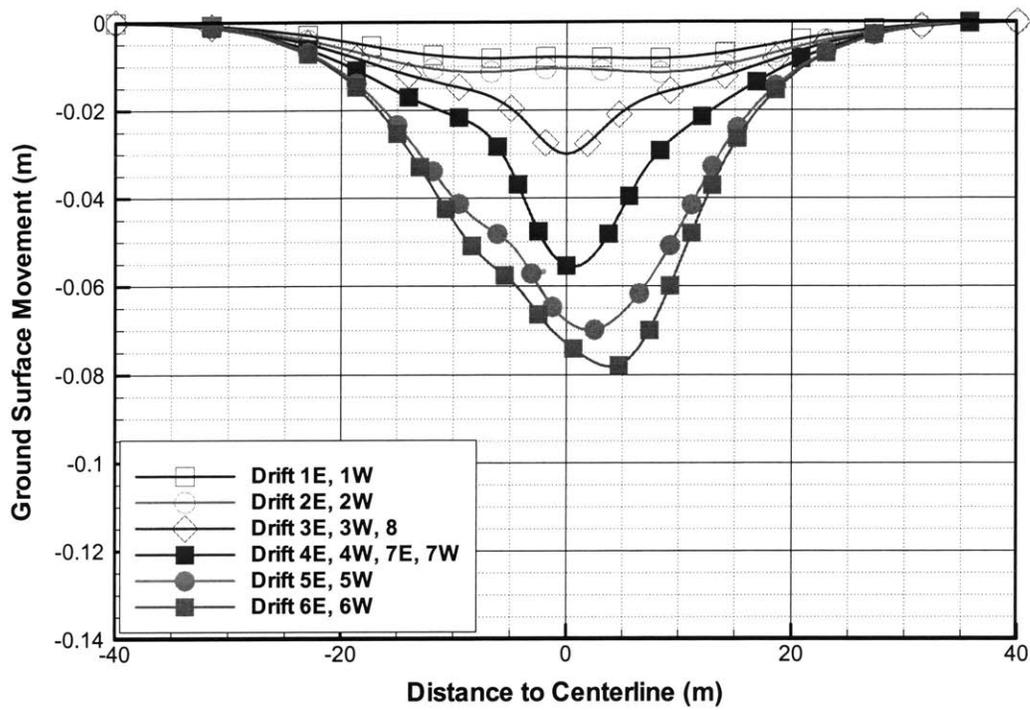


(a) P-1: Surface Movement due to Grouting Gallery and Drift Excavations

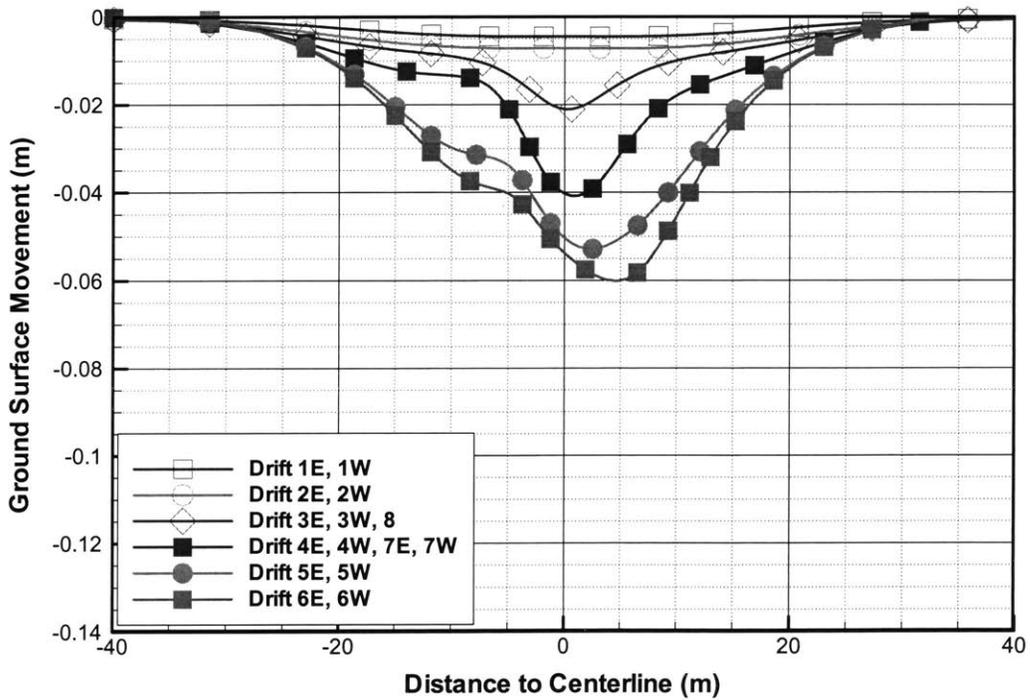


(b) P-1: Surface Movement due to Drift Excavations

Figure 5-15 Calculated Surface Movement, Reference Configuration

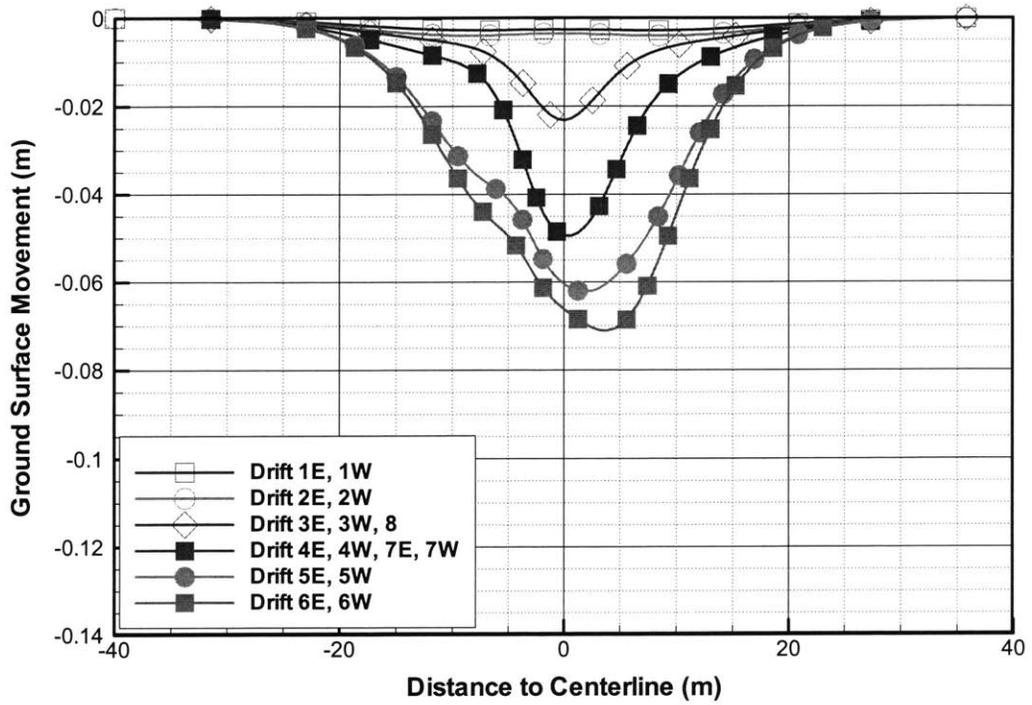


(a) P-13: $K_0 = 0.5$, Stiffness Evaluated at $K_0 = 1.0$

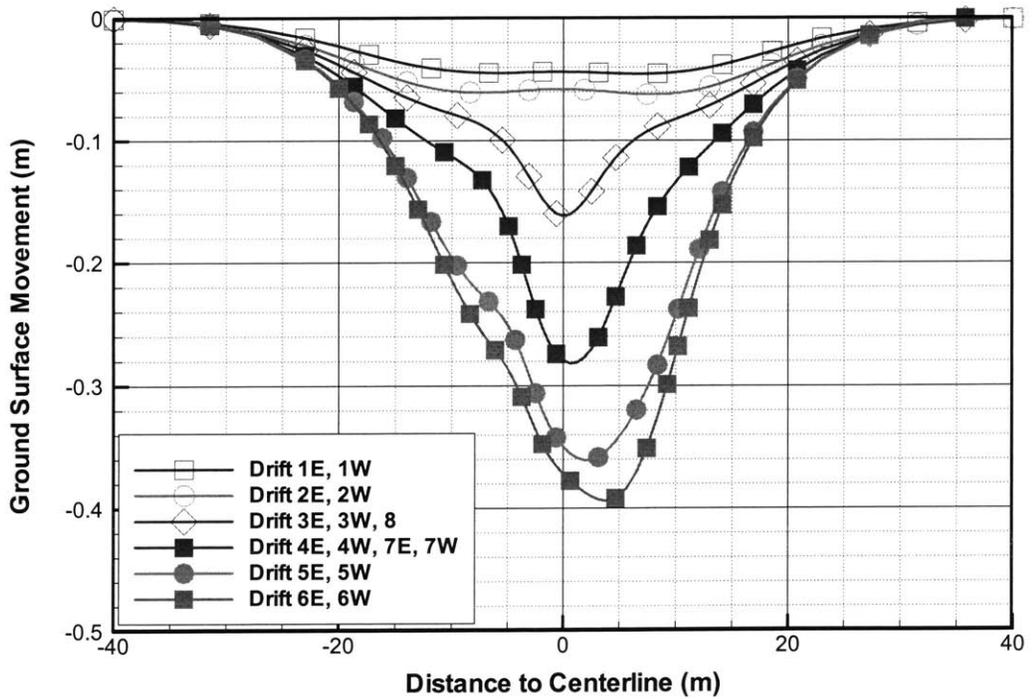


(b) P-2: $K_0 = 1.0$, Stiffness Evaluated at $K_0 = 1.0$

Figure 5-16 Effect of K_0 on Settlement Troughs

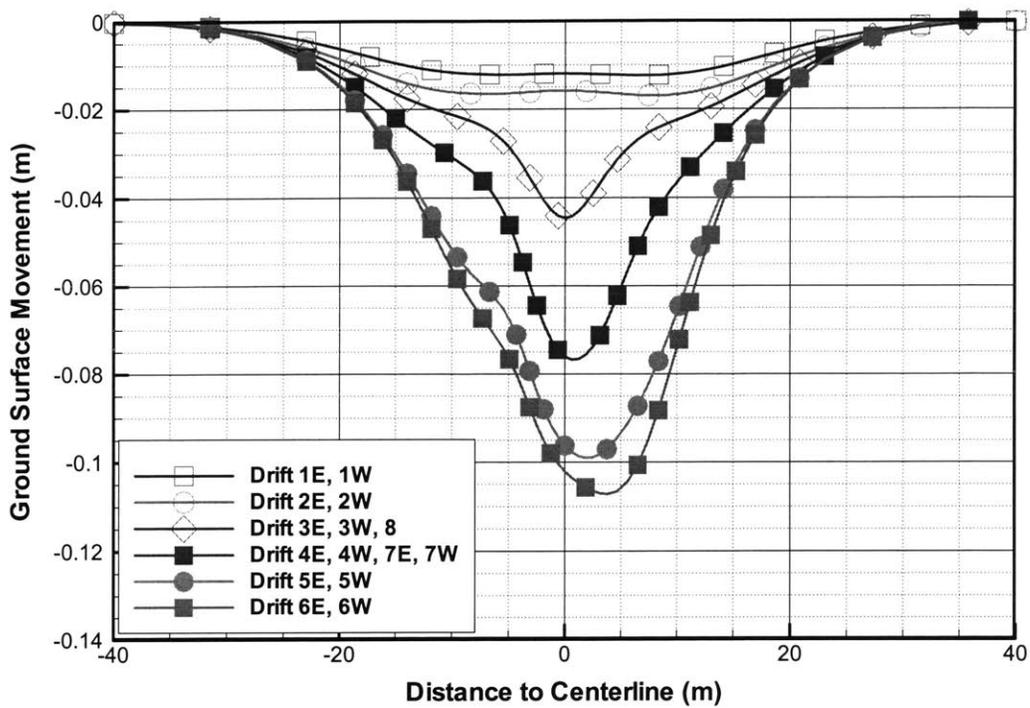


(a) P-5: Soil Stiffness Determined with Shear Strain $\gamma = 0.01\%$

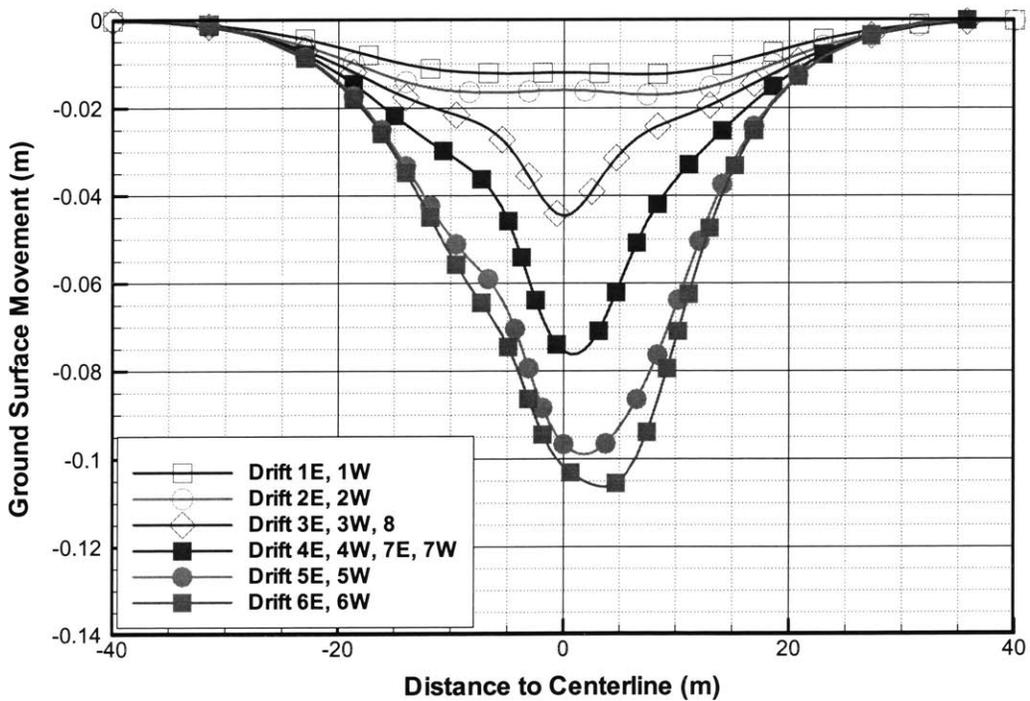


(b) P-6: Soil Stiffness Determined at Shear Strain $\gamma = 1.0\%$

Figure 5-17 Effect of Soil Stiffness on Surface Settlement Troughs

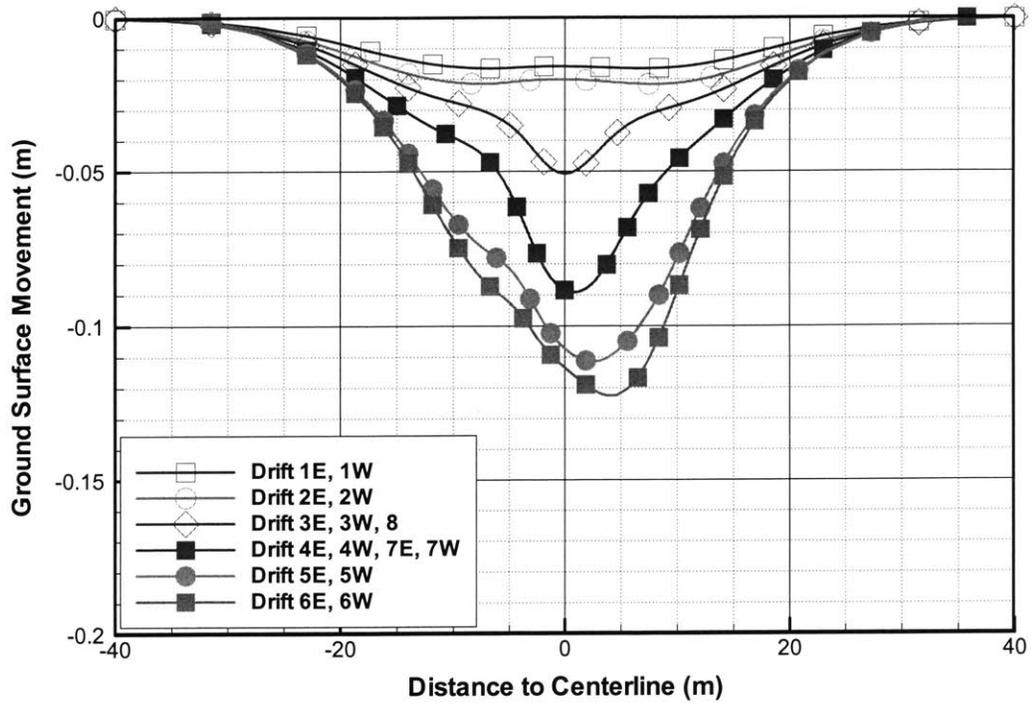


(a) P-3: Lining Stiffness Increased by a factor of 10

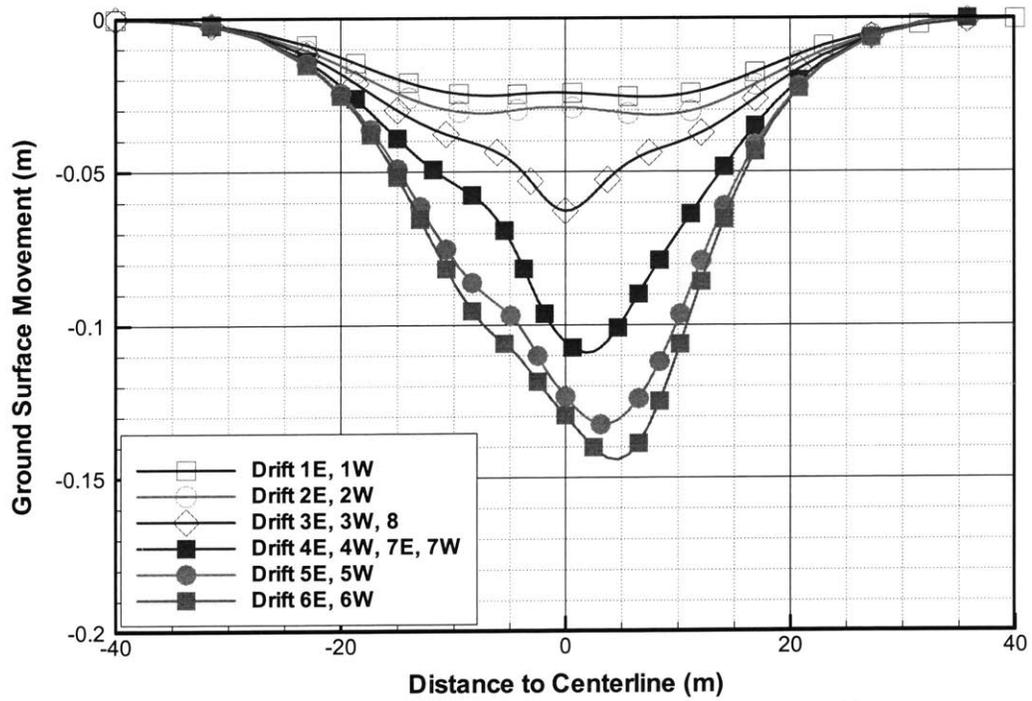


(b) P-4: Lining Stiffness Increased by a factor of 100

Figure 5-18 Effect of Lining Stiffness on Surface Settlement Troughs



(a) P-10: Decrease Friction Angle of Middle-Zone by 5 Degrees



(b) P-11: Decrease Friction Angle of Middle-Zone by 10 Degrees

Figure 5-19 Effect of Friction Angle of Middle Zone Soil on Surface Settlement Troughs

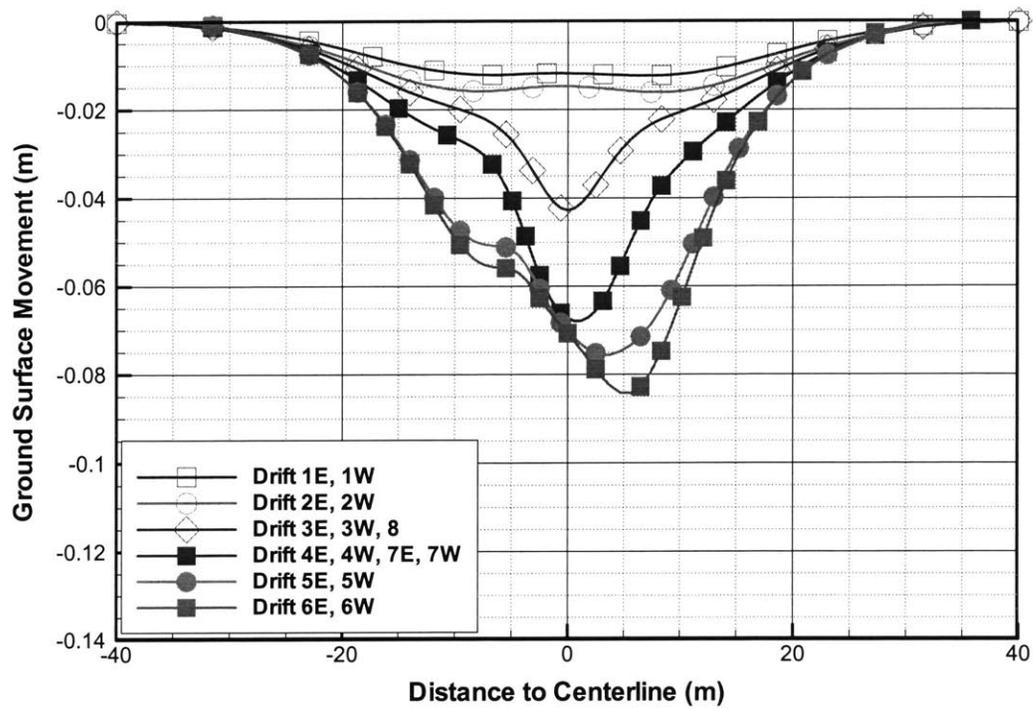
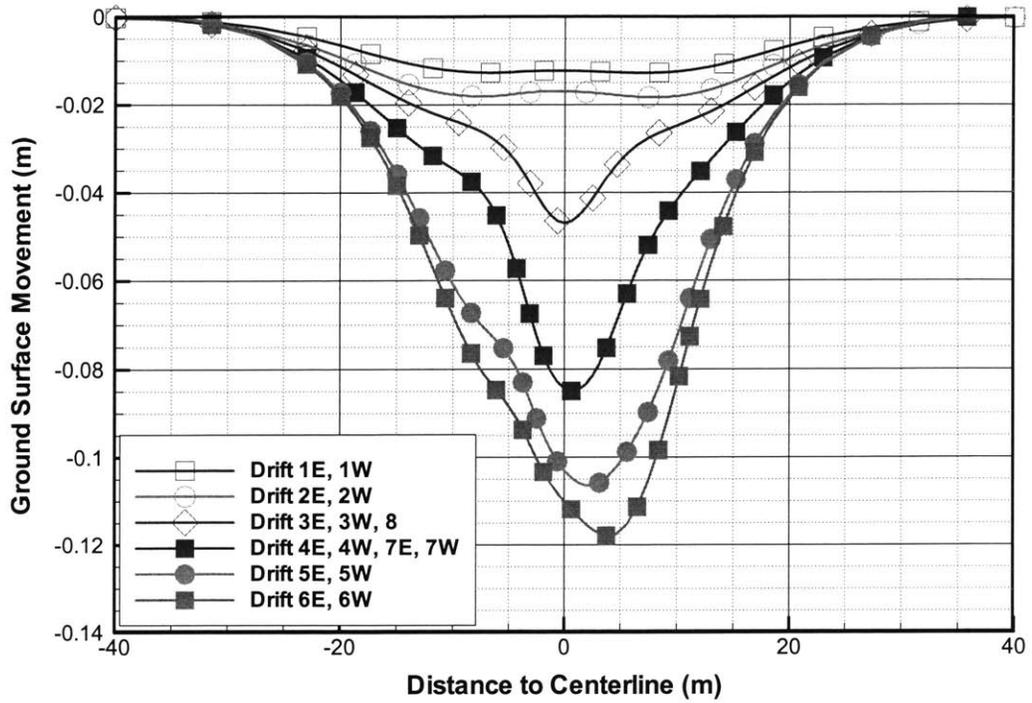
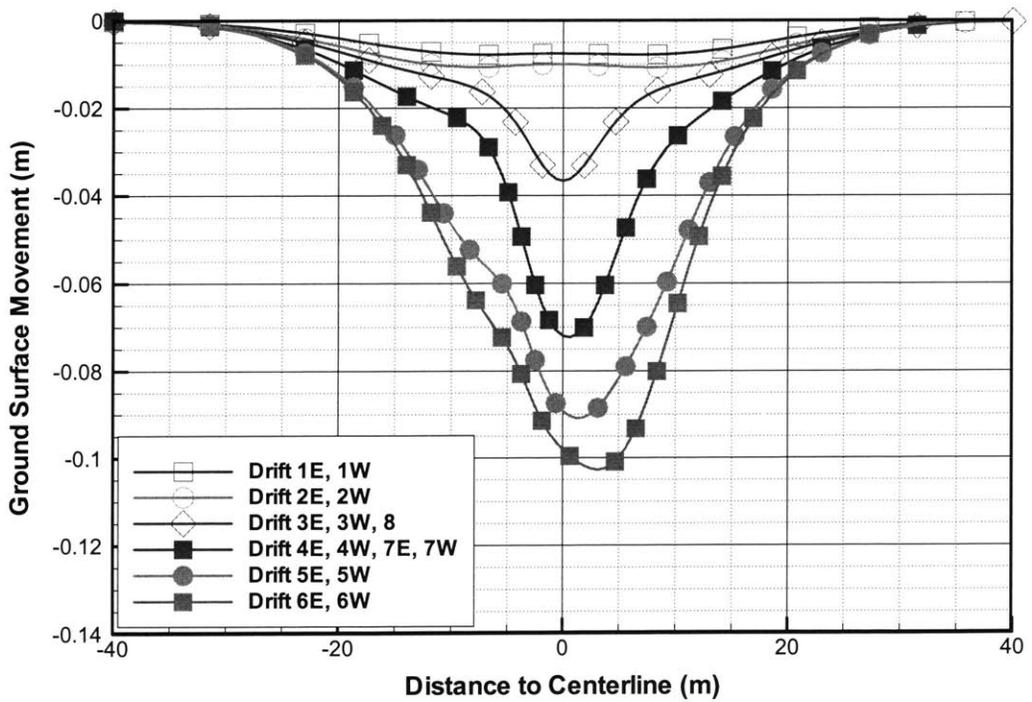


Figure 5-20 P-7: Effect of Concreting on the Surface Settlement Troughs



(a) P-9: Drucker-Prager Matching Mohr-Coulomb in Plane-Strain Shearing



(b) P-8: Drucker-Prager Matching Mohr-Coulomb in Triaxial Compression Shearing

Figure 5-21 Effect of Different Soil Yield Criteria for the Middle-Zone Soil

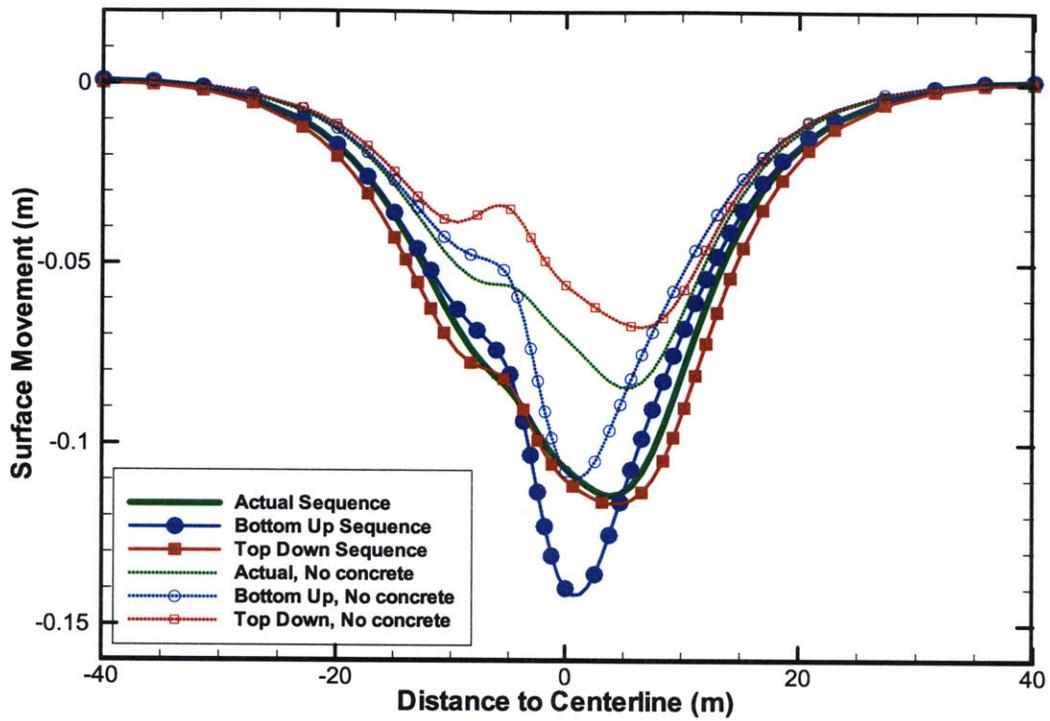


Figure 5-22 Plane-Strain Finite Element Calculations with Alternative Sequences

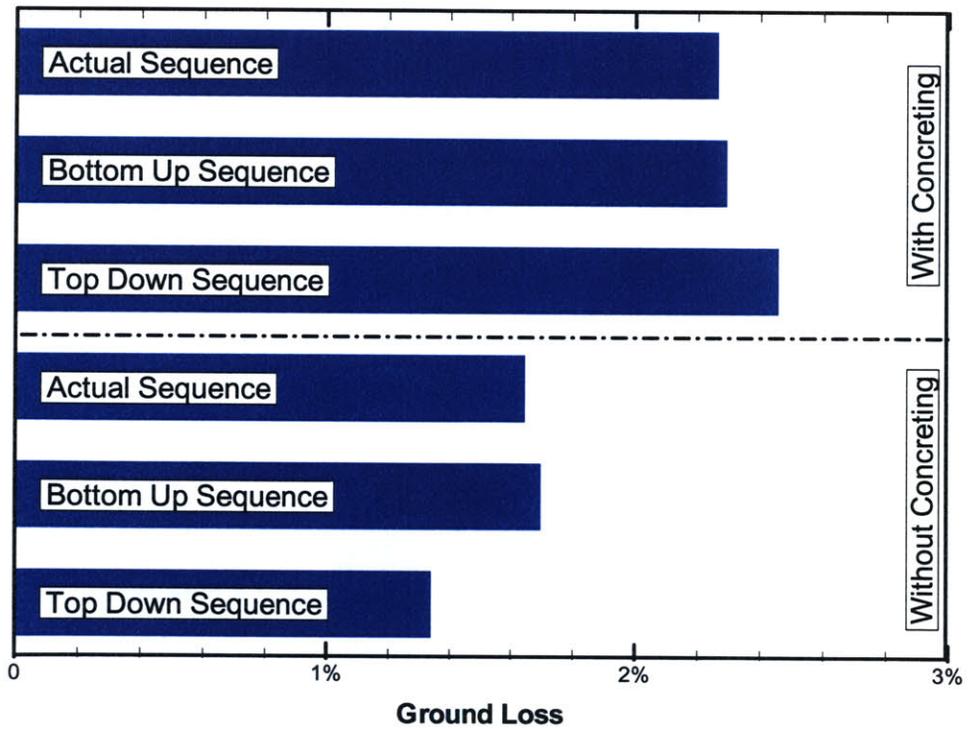


Figure 5-23 Volume Losses of Plane-Strain Finite Element Calculations

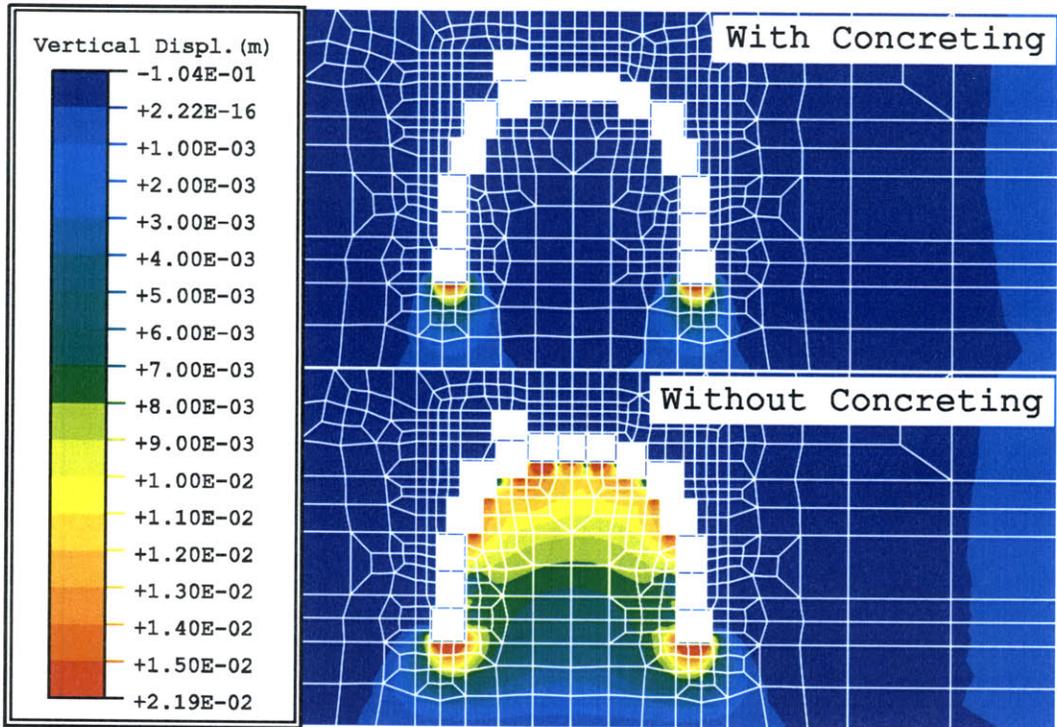


Figure 5-24 Upward Displacement Contours from 2D FEA with Top-Down Sequence

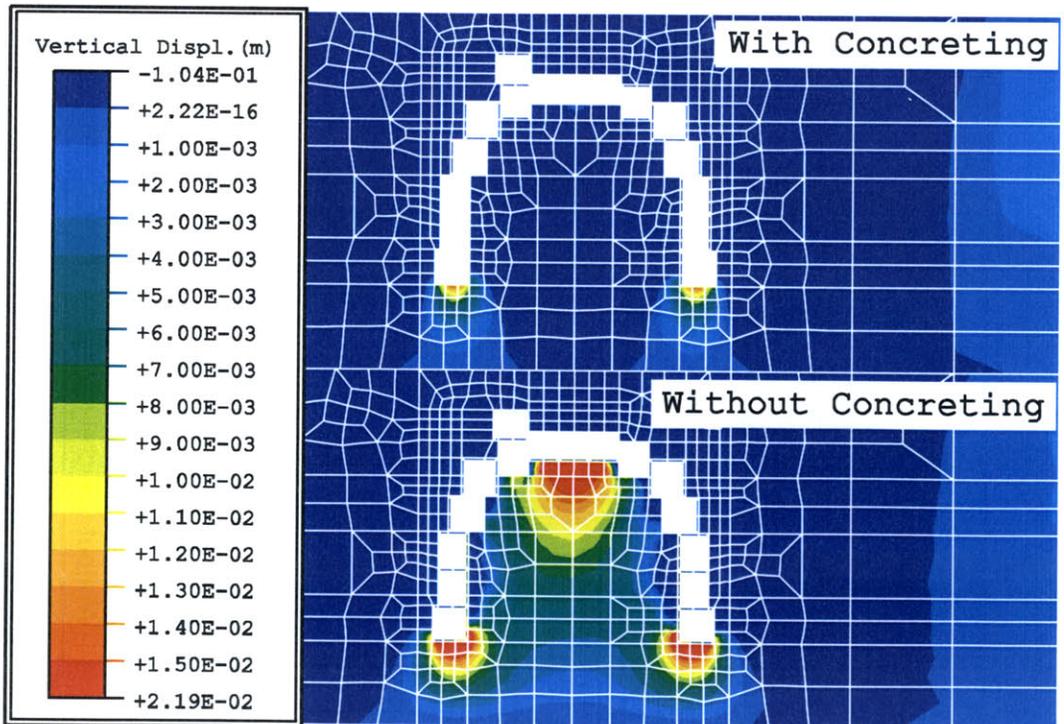


Figure 5-25 Upward Displacement Contours from 2D FEA with Bottom-Up Sequence

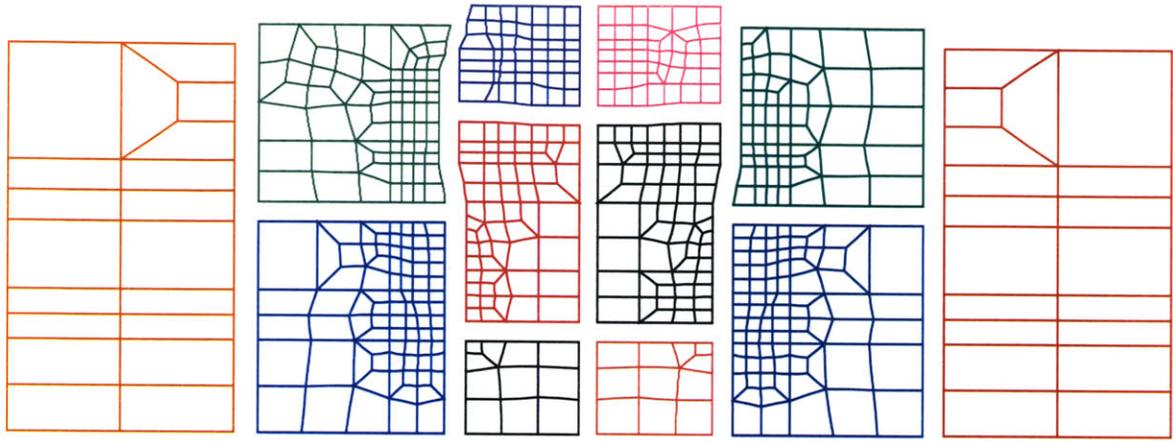
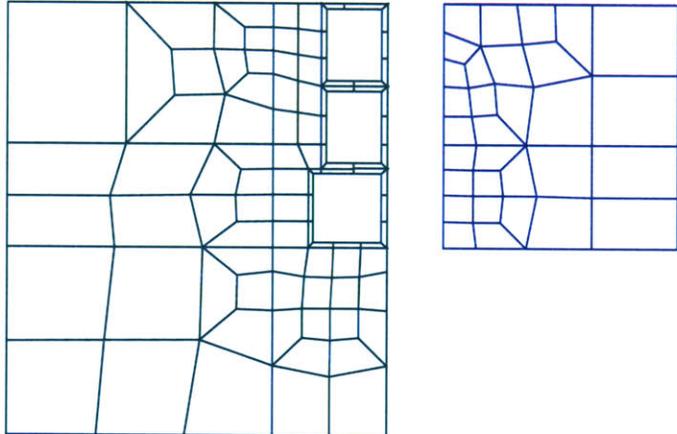
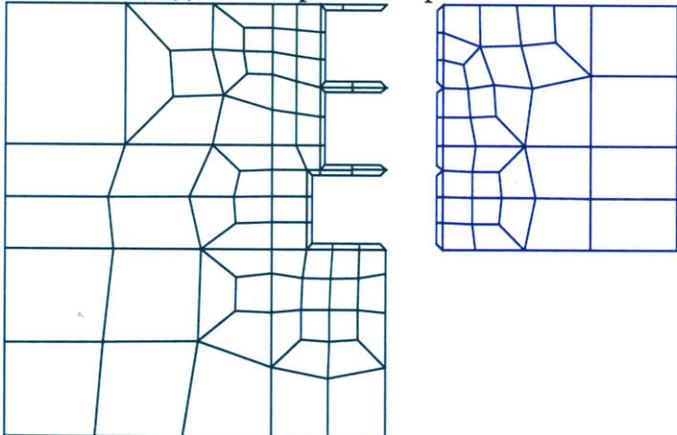


Figure 5-26 Two-Dimensional Decomposition of the Finite Element Model



(a) Decomposition option 1



(b) Decomposition option 2

Figure 5-27 Special Consideration of Decomposition with Lining Elements

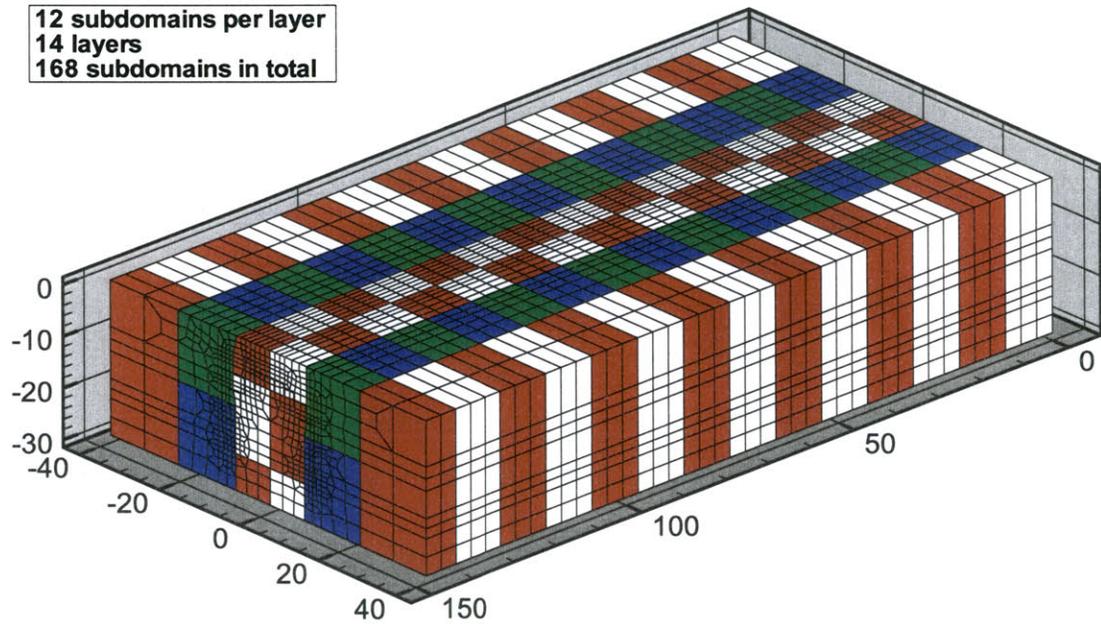


Figure 5-28 Three-Dimensional Mesh Decomposition used in FETI_FEM for SDT Model

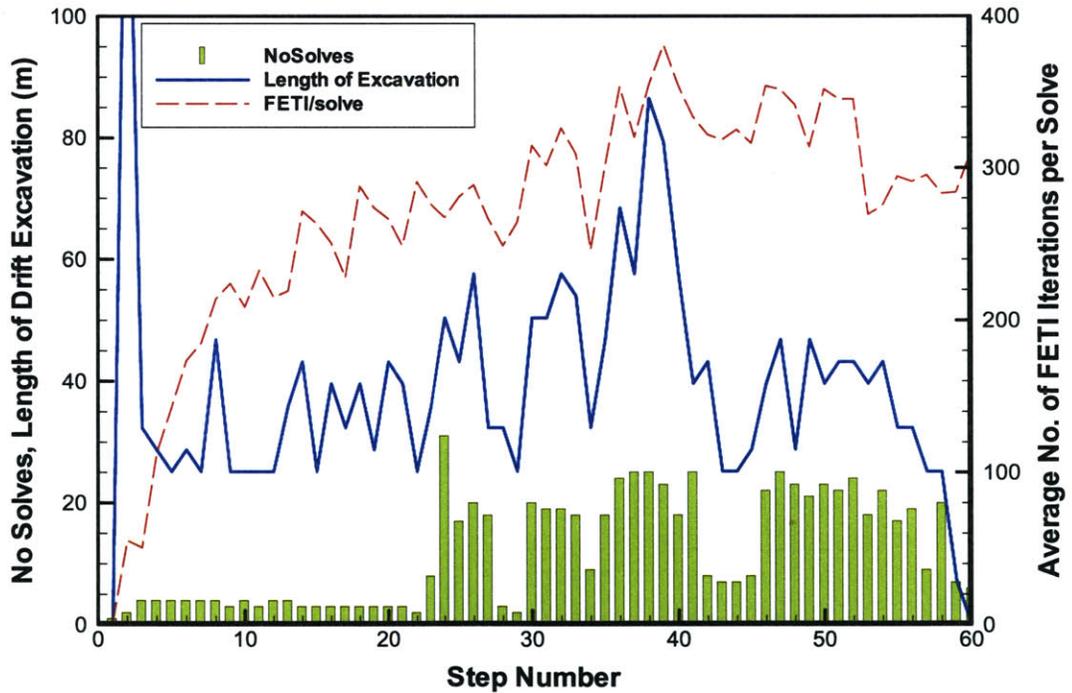


Figure 5-29 Solution History for Three-Dimensional Analysis of SDT using FETI_FEM

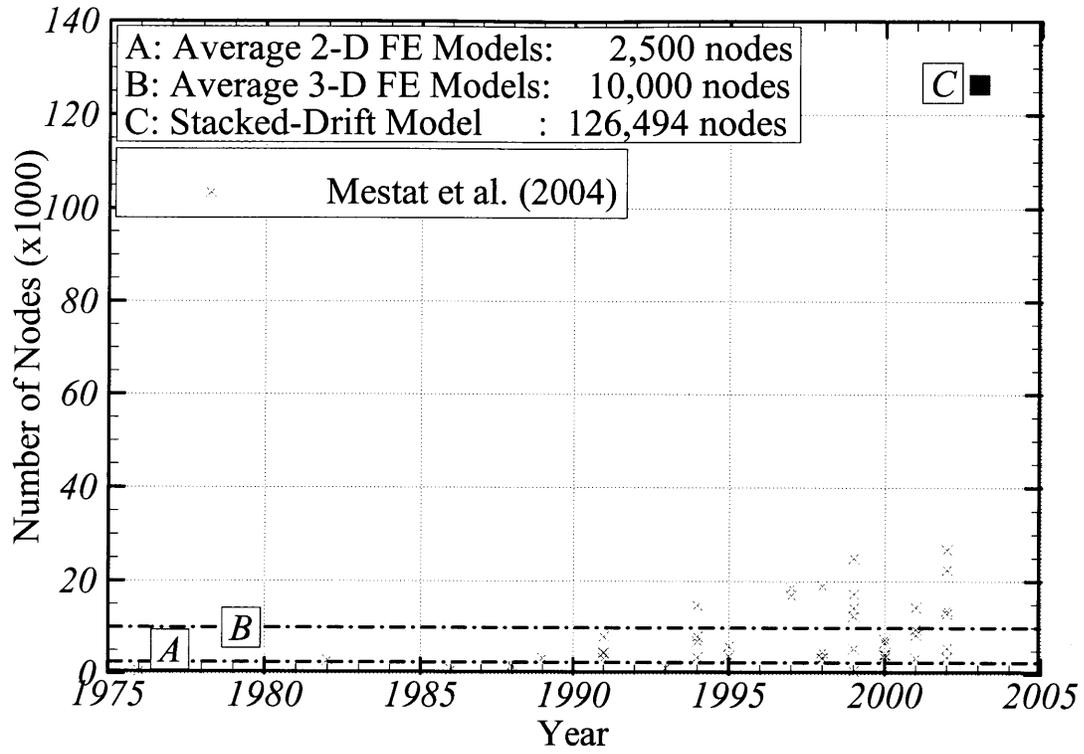
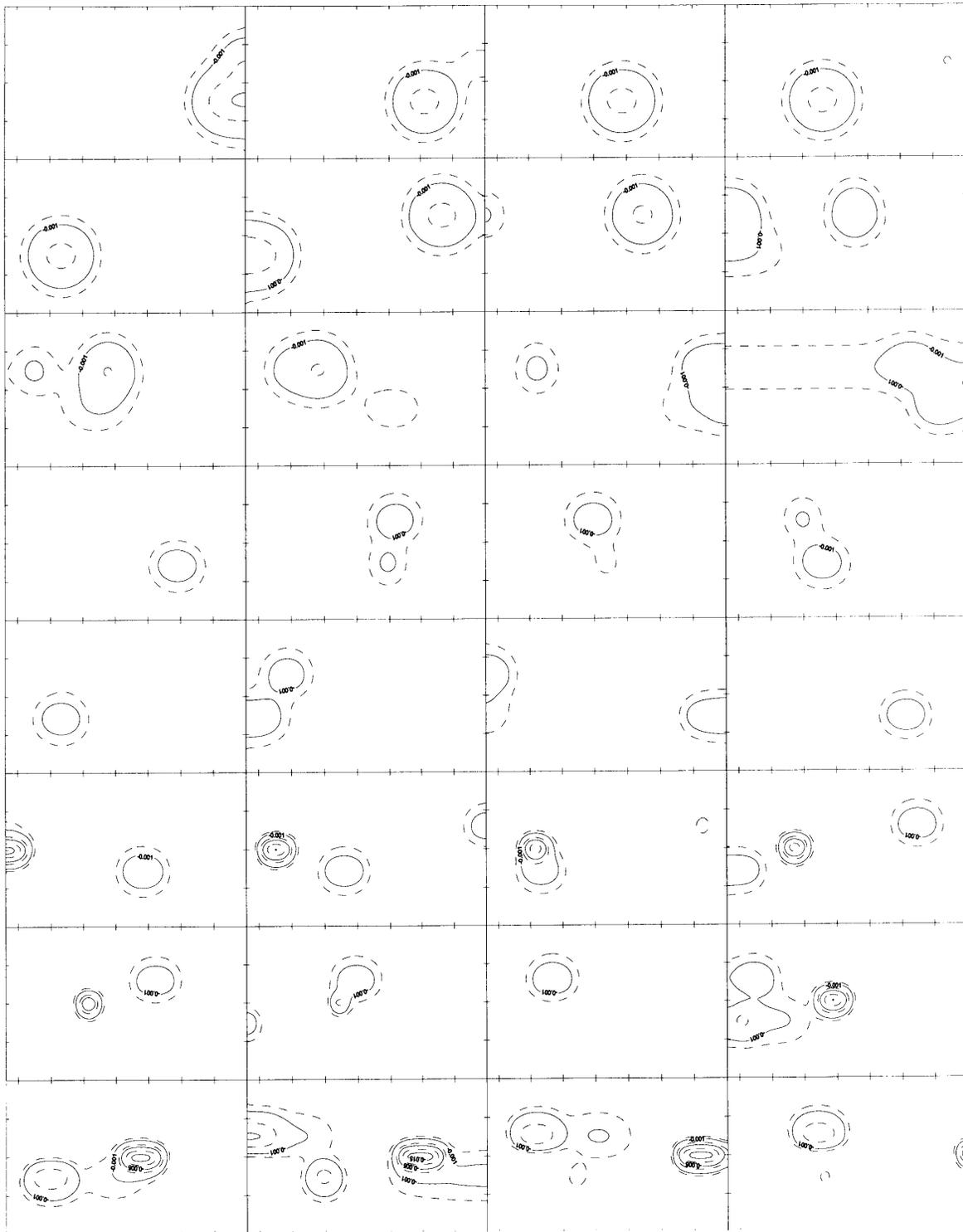
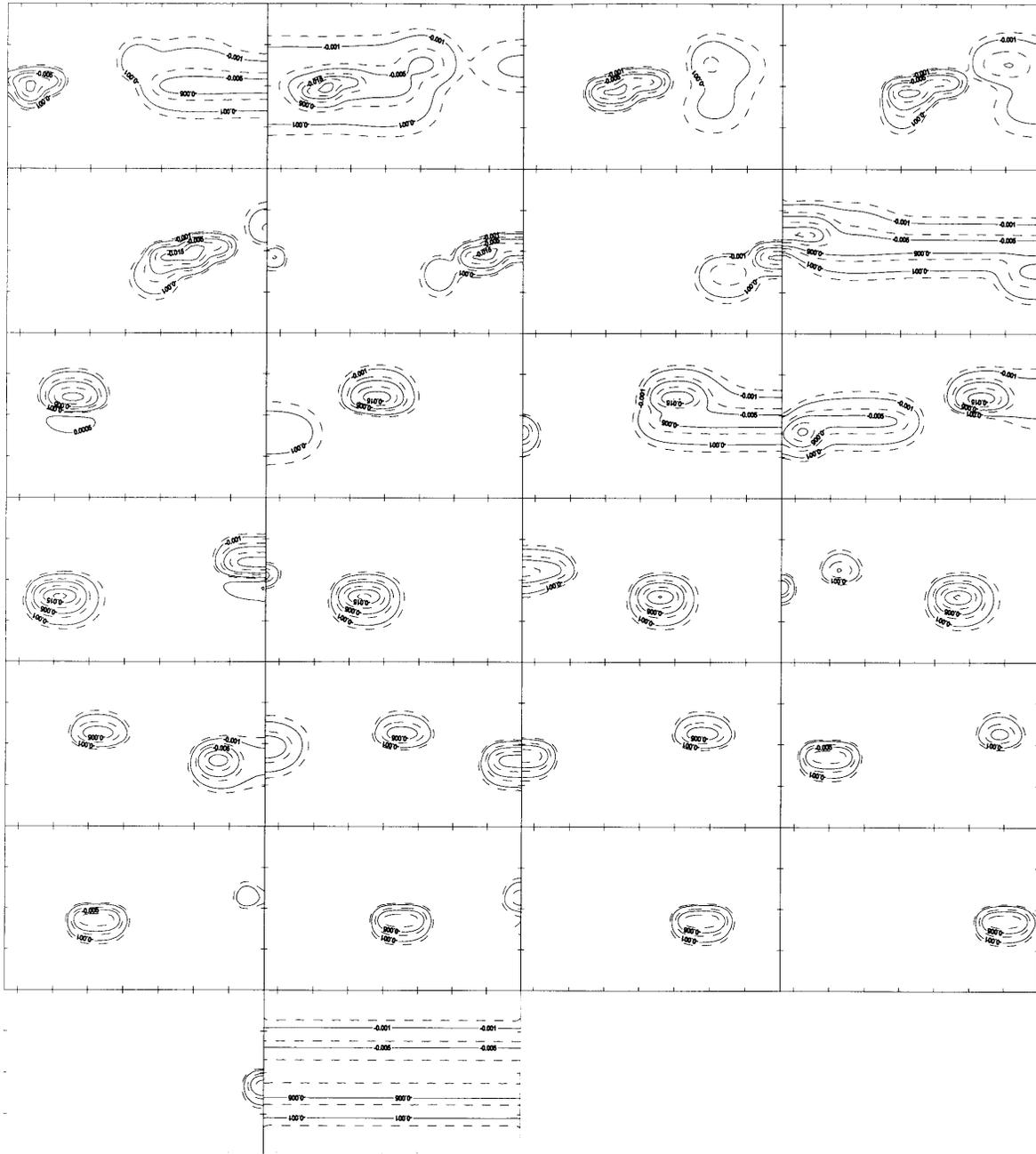


Figure 5-30 Comparison of the SDT Model Size with MOMIS Database (Mestat et al. 2004)



(a) Step 3 – Step 34

Figure 5-31 Incremental Surface Movement during Stacked-Drift Construction



(b) Step 35 – 60

Figure 5-31 (Cont'd) Incremental Surface Movement during Stacked-Drift Construction

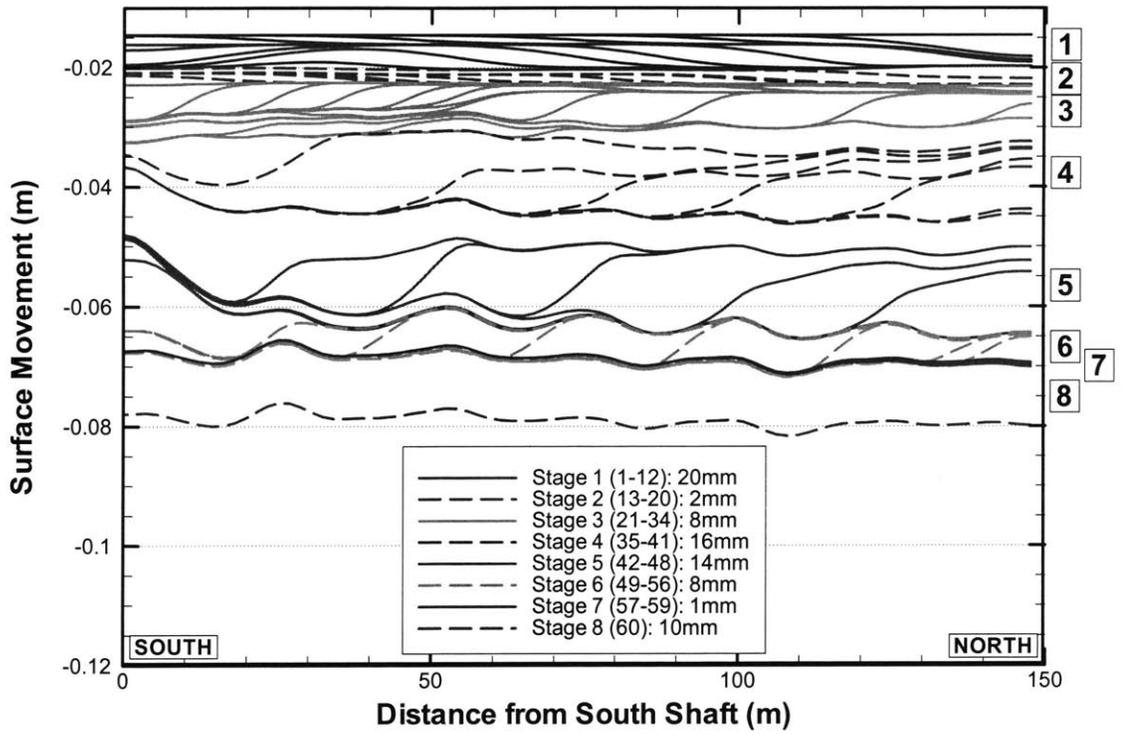


Figure 5-32 Evolution of Surface Settlement above Grouting Gallery

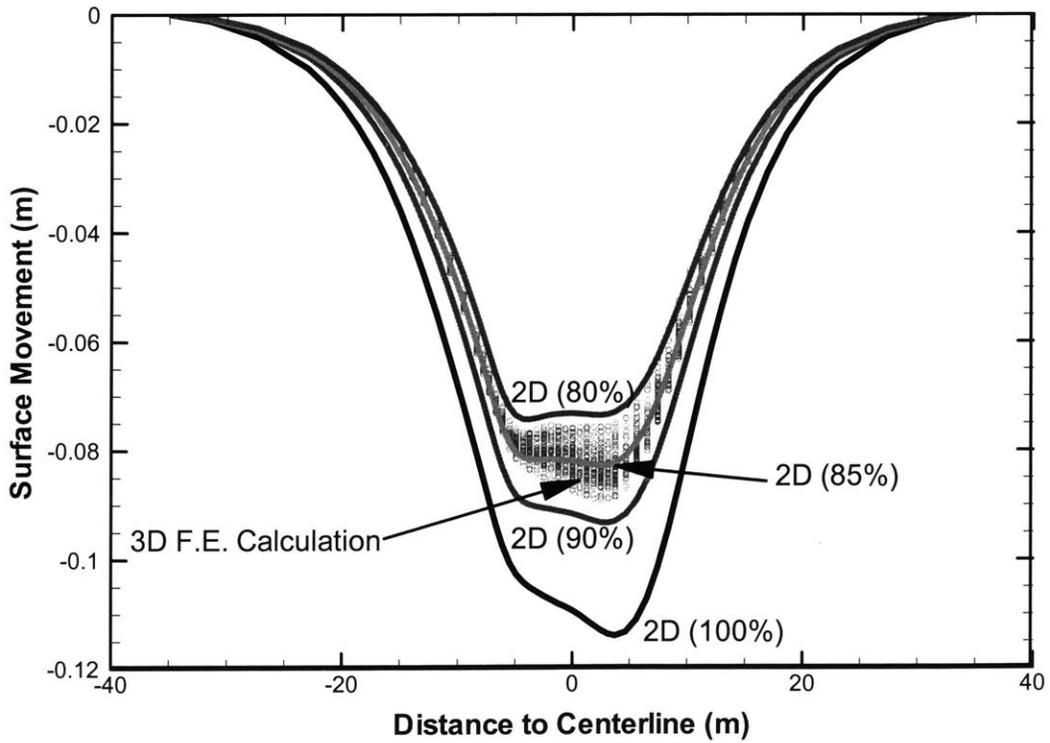


Figure 5-33 Comparison of 2D and 3D Settlement Trough

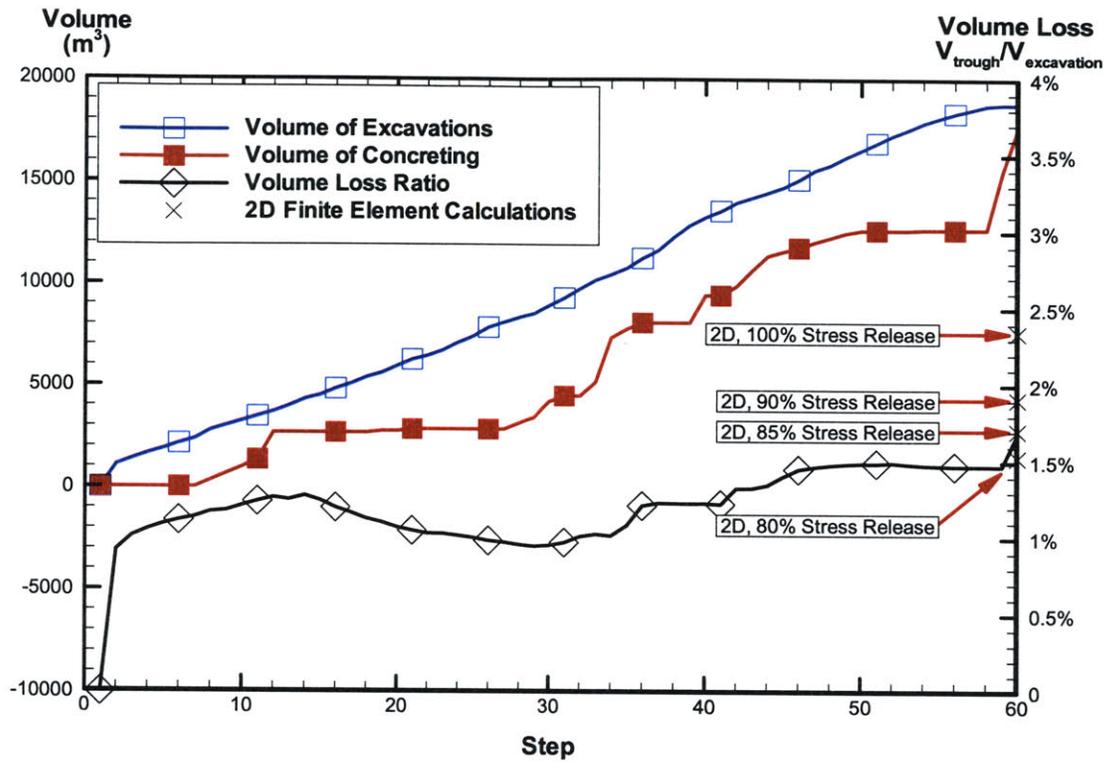
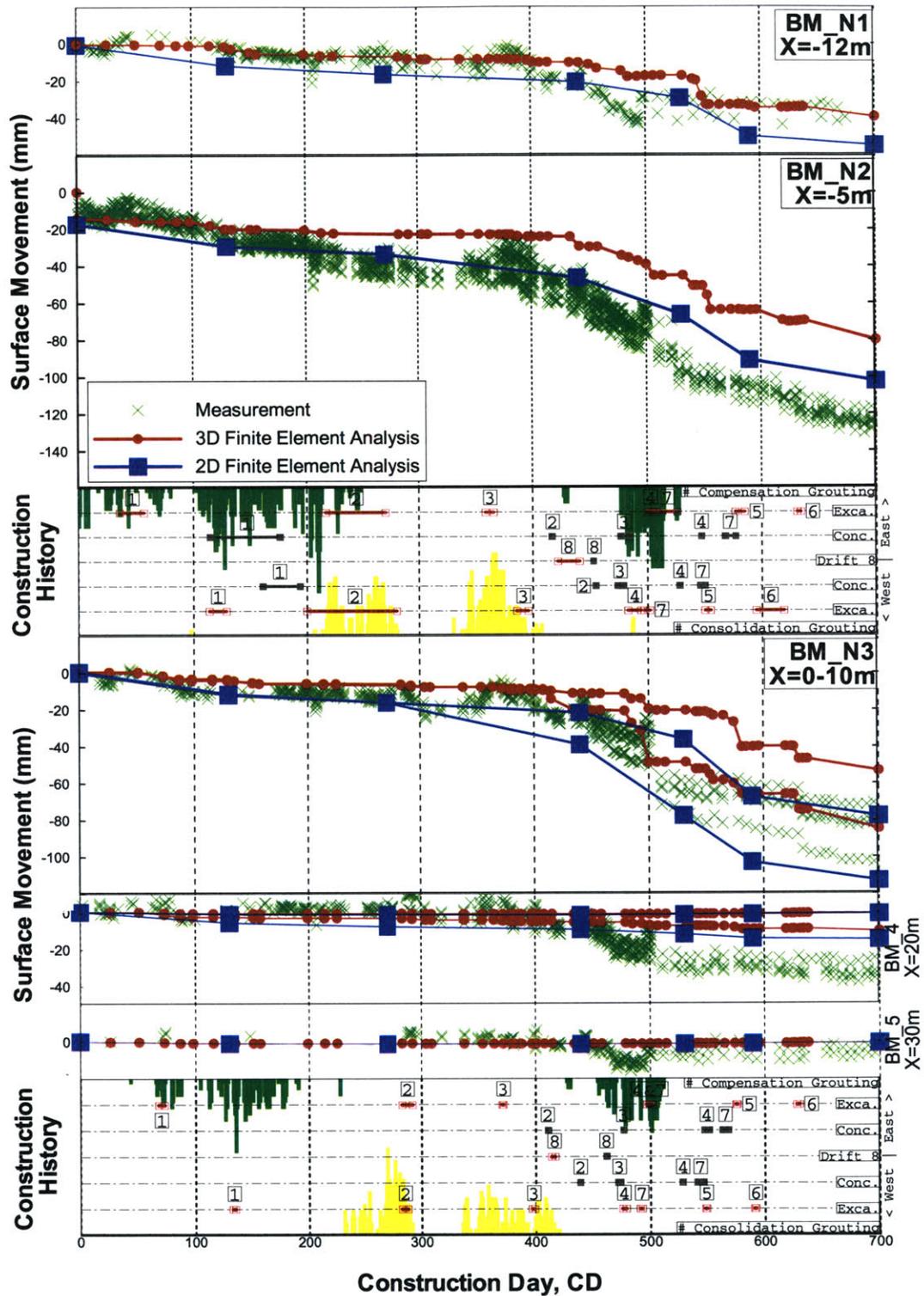
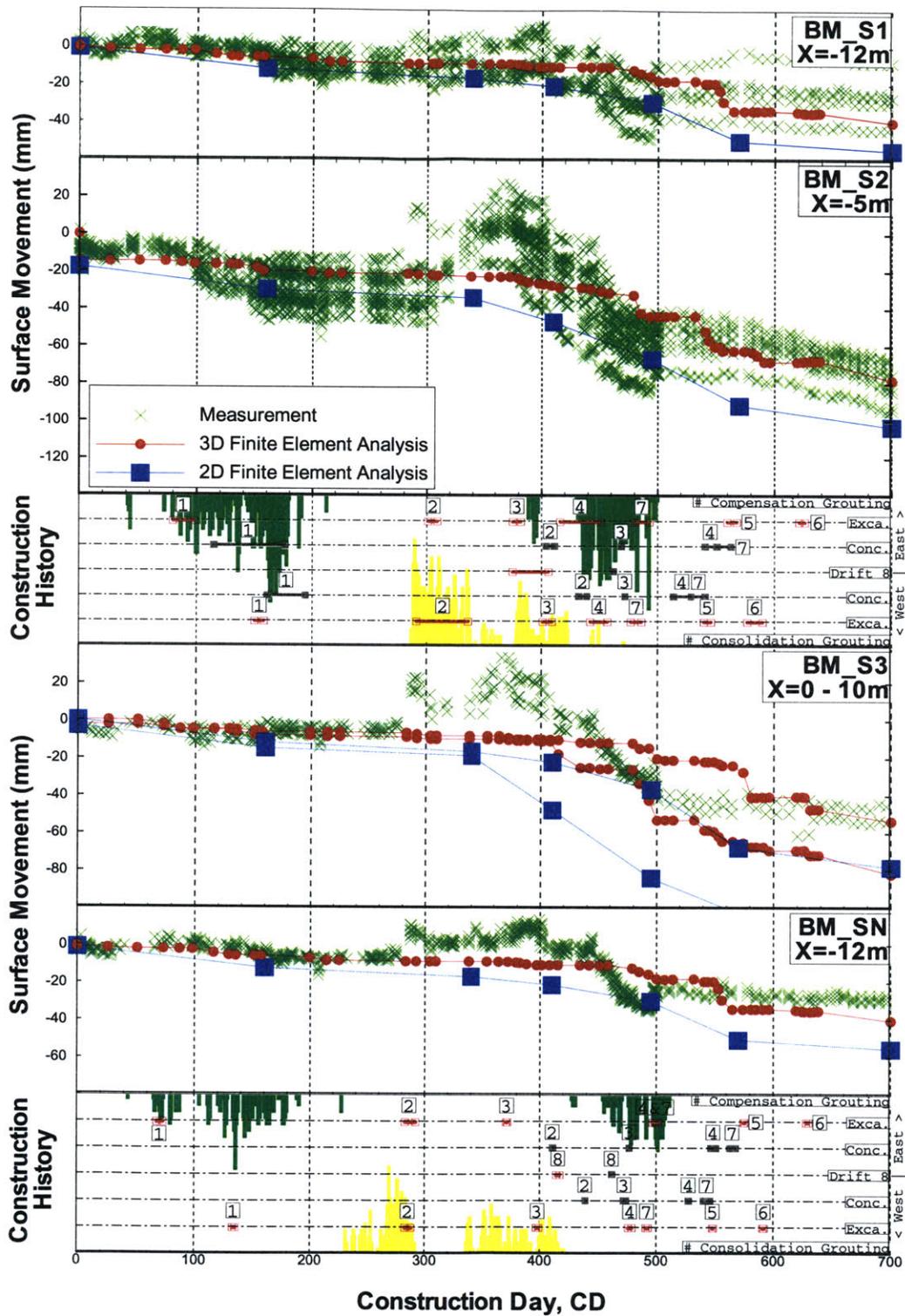


Figure 5-34 Observed Ground Loss from Three-Dimensional Finite Element Analyses



(a) North Section benchmarks in BM_N1, BM_N2, BM_N3, BM_4, and BM_5

Figure 5-35 Comparisons between Calculation Results and In-situ Measurements at Selected Offset Locations



(b) South section benchmarks in BM_S1, BM_S2, BM_S3, and BM_SN

Figure 5-35 (Cont'd) Comparisons between Calculation Results and In-situ Measurements at Offset Select Locations

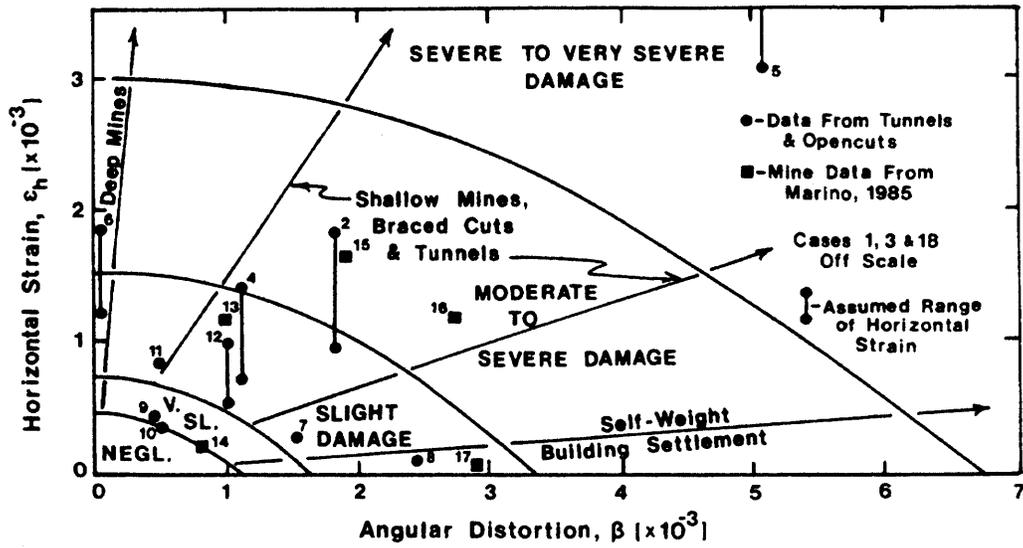


Figure 5-36 Building Damage Evaluation Chart (Boscardin and Cording 1989)

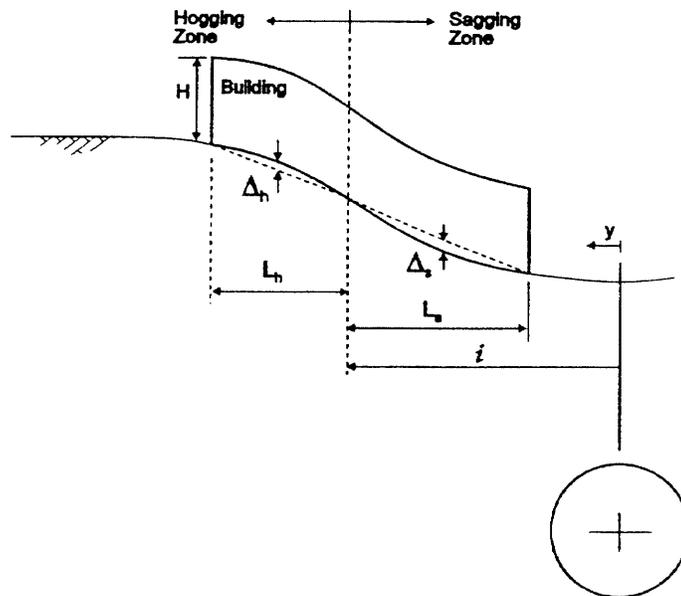


Figure 5-37 Sagging and Hogging of Building Deformation due to Tunneling (Mair et al. 1996)

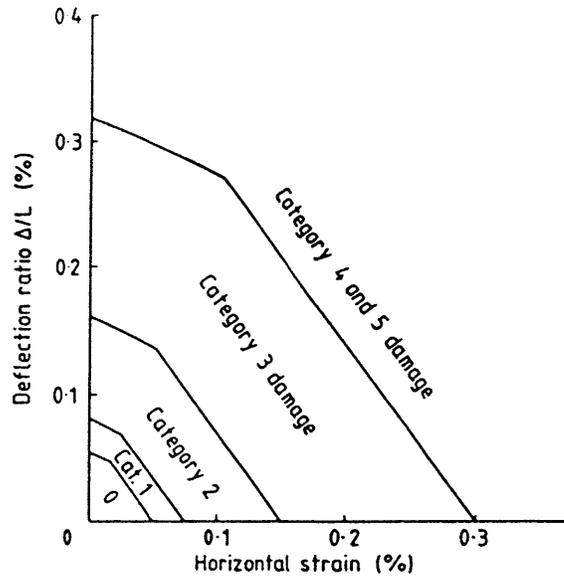
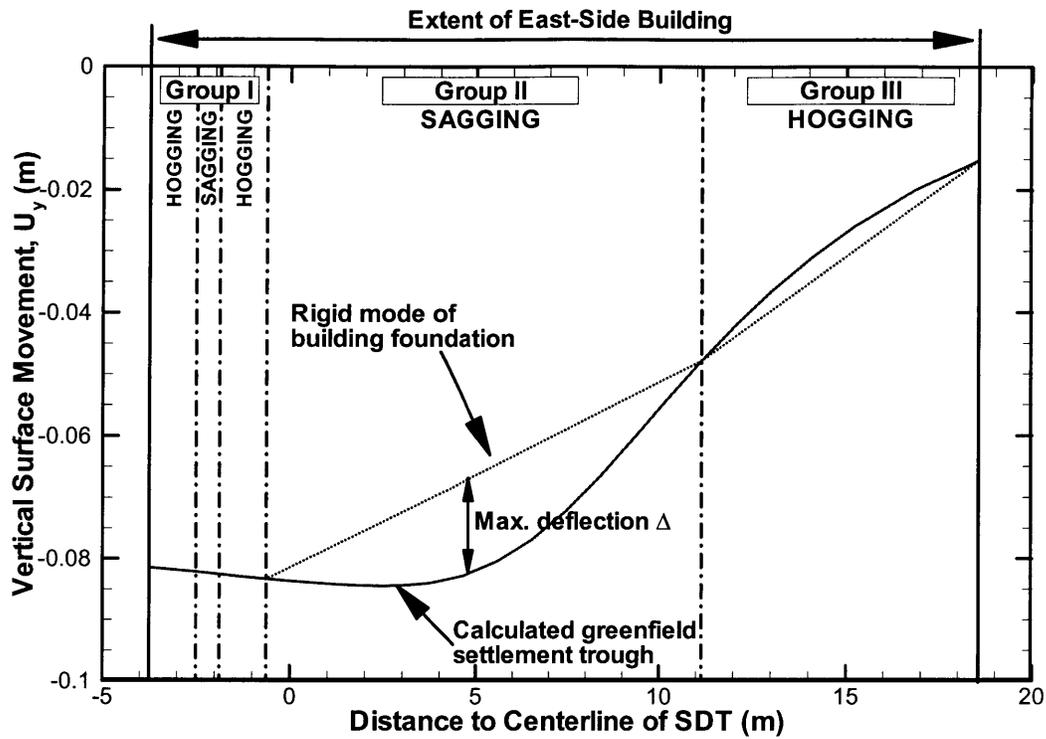
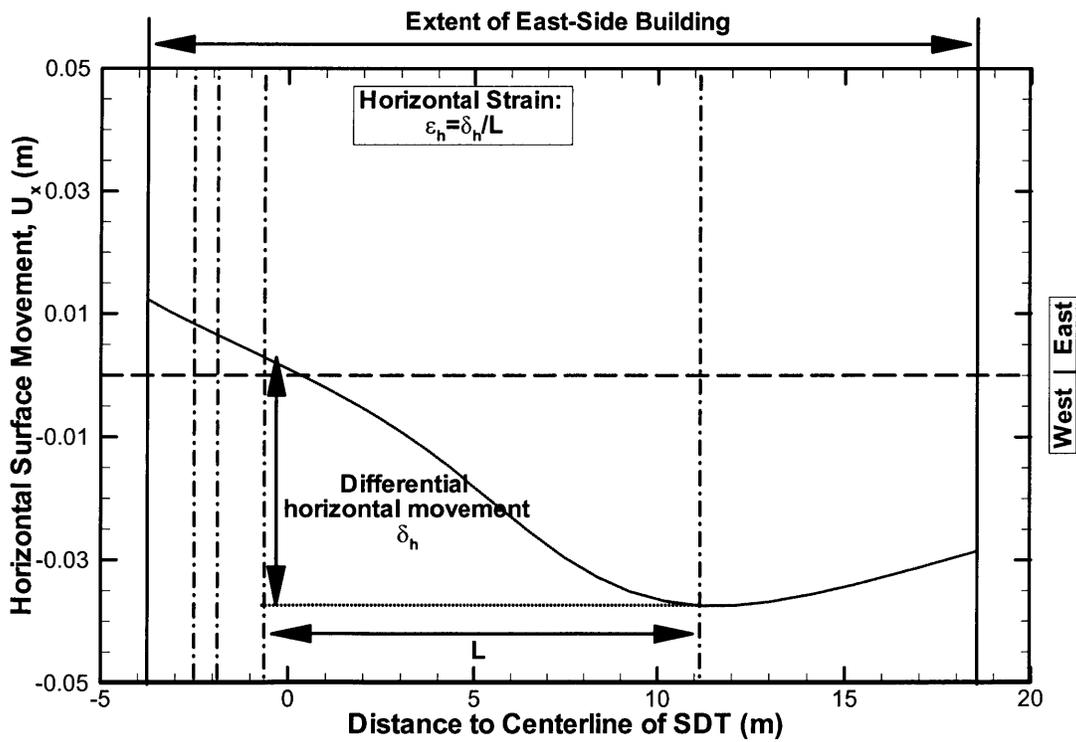


Figure 5-38 Damage Category Chart for $L/H = 1$, Hogging Mode
(Burland 1995)

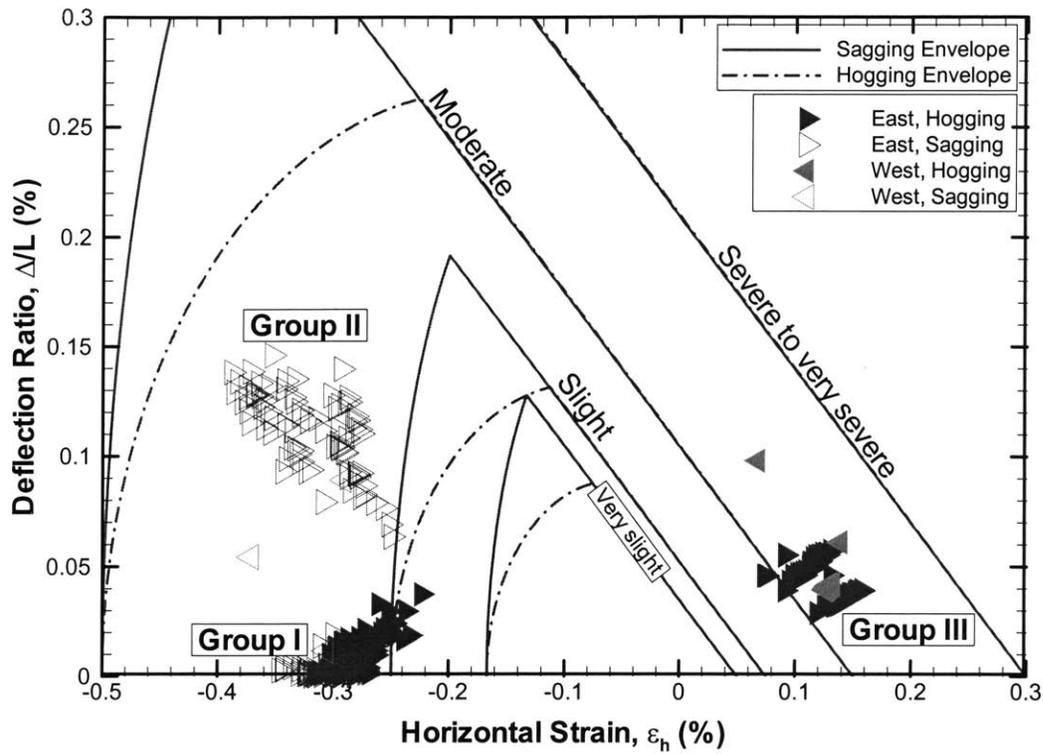


(a) Vertical ground movement

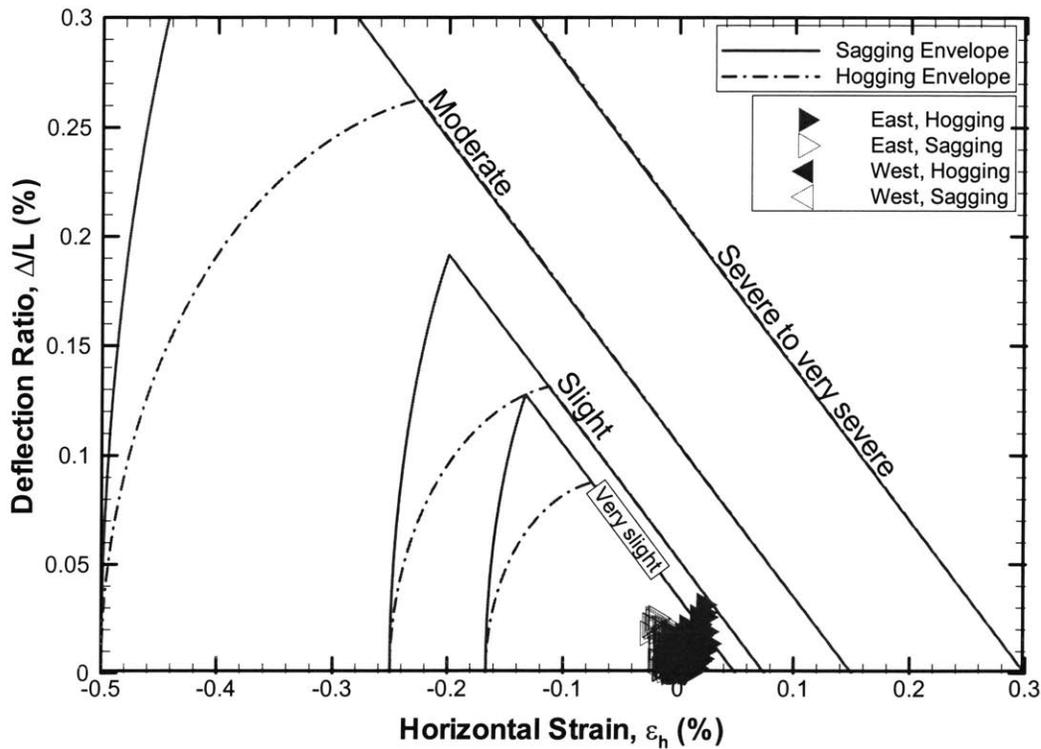


(b) Horizontal ground movement

Figure 5-39 Settlement Trough under East-Side Buildings for a Particular Section

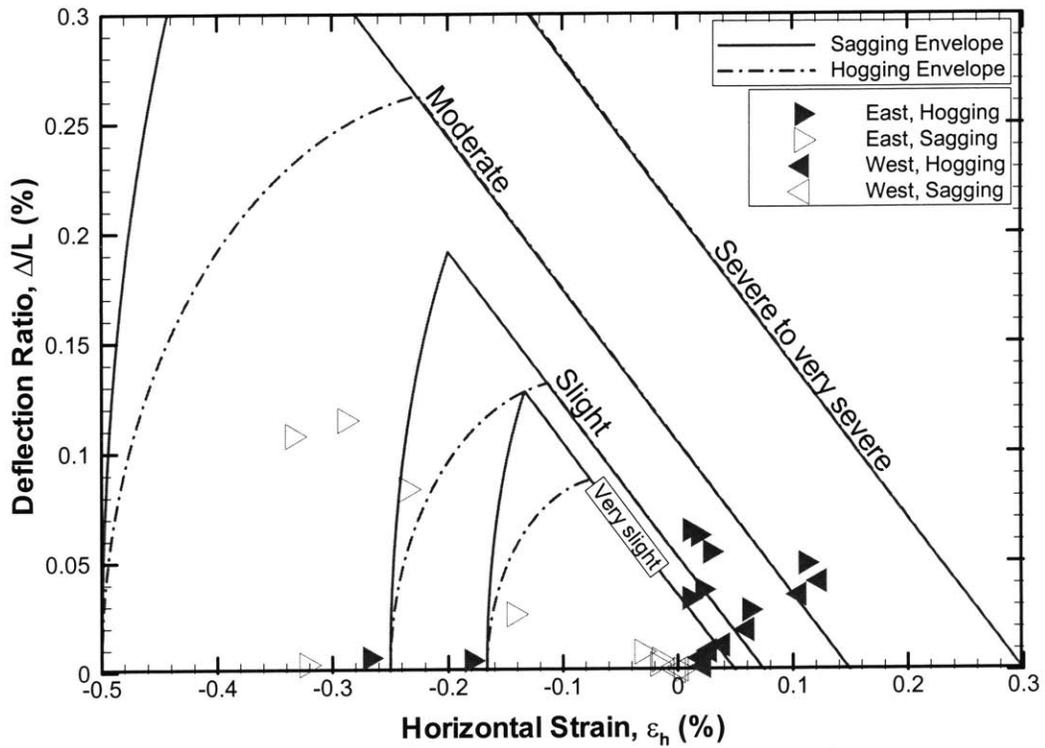


(a) Building Damage Assessment in Cross-Sectional Direction of Tunneling

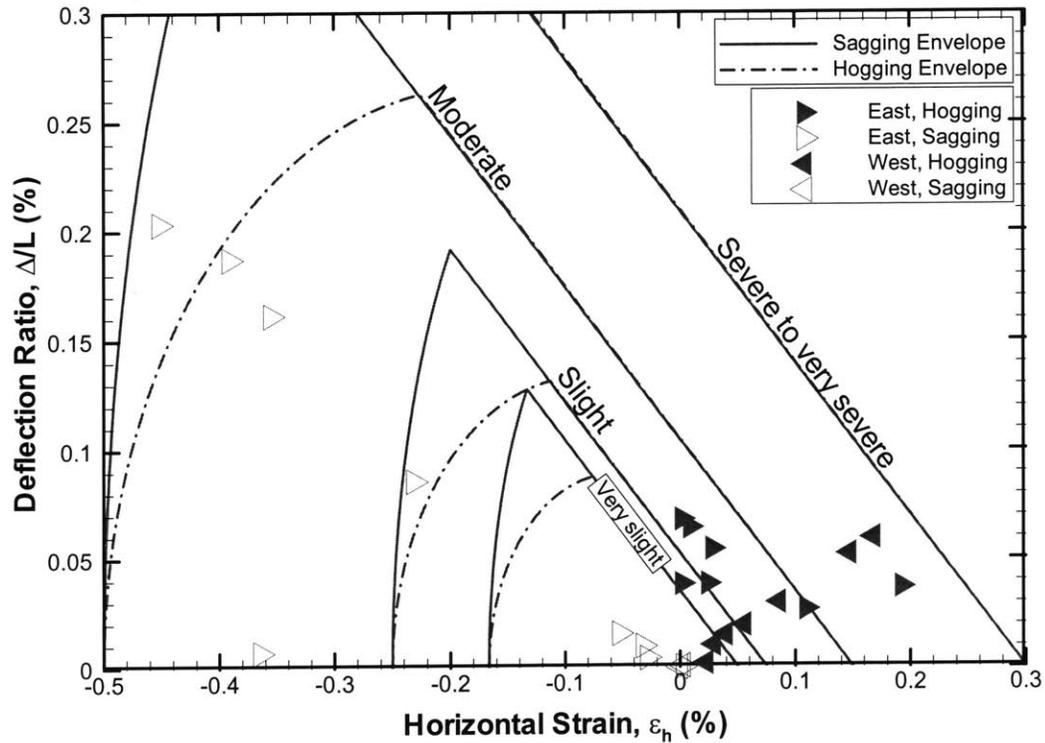


(b) Damage Assessment in Longitudinal Direction of Tunneling

Figure 5-40 Damage Assessment of Stacked-Drift Tunneling with 3-D FEA in the End of Construction



(a) 2D Plane-Strain Analysis with 85% Stress Release



(b) 2D Plane-Strain Analysis with 100% Stress Release

Figure 5-41 Damage Assessment of Building Damages with 2D Finite Element Analyses throughout the Construction

References

- Augarde, C. E., Wisser, C., and Burd, H. J. (1999). "Numerical Modelling of Tunnel Installation Procedures." *Numerical Methods in Geomechanics*, Pande, Pietruszczak, and Schweiger, eds., 329-334.
- Boscardin, M. D., and Cording, E. J. (1989). "Building response to excavation-induced settlement." *Journal of Geotechnical Engineering*, 115(1), 1-21.
- Burland, J. B. "Assessment of risk of damage to buildings due to tunnelling and excavations." *1st Int. Conf. on Earthquake Geotechnical Engineering*, Tokyo, Japan.
- Burland, J. B., and Wroth, C. P. "Settlement of buildings and associated damage." *Conf. Settlement of Structures*, Cambridge, 611-654.
- Deere, D. U. (1955). "Engineering Properties of the Pleistocene and recent sediments of the San Juan Bay area, Puerto Rico," University of Illinois, Urbana.
- Farhat, C., Chen, P. S., Mandel, J., and Roux, F. X. (1998). "The two-level FETI method Part II: Extension to shell problems, parallel implementation and performance results." *Computer Methods in Applied Mechanics and Engineering*, 155(1-2), 153-179.
- Farhat, C., Lesoinne, M., and Pierson, K. (2000). "A scalable dual-primal domain decomposition method." *Numerical Linear Algebra with Applications*, 7(7-8), 687-714.
- Kaye, C. A. (1959). "Coastal Geology of Puerto Rico: (A) Geology of the San Juan Metropolitan Area." US Government Printing Office, Washington, D.C.
- LaFonta, G., and Person, T. (1999). "Puerto Rico real-time control of compensation grouting with the CYCLOPS system." *Geotechnical News*, 27-33.
- Mair, R. J., Taylor, R. N., and Burland, J. B. (1996). "Prediction of Ground Movements and Assessment of Risk of Building Damage due to Bored Tunnelling." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 713-718.
- Mestat, P., Bourgeois, E., and Riou, Y. (2004). "Numerical Modelling of embankments and underground works." *Computers and Geotechnics*, 31, 227-236.
- Morrison, J. A., Madsen, P. H., and Carayol, S. "Ground Control Program for the Río Piedras Project, Tren Urbano Program, San Juan, Puerto Rico." *Rapid Excavation and Tunneling Conference*, Orlando, U.S.A.
- Urquiza, L. J. (2000). "Ground deformations caused by the construction of the Río Piedras underground station using the stacked drift tunneling method," University of Puerto Rico, Mayaguez.
- Zhang, G. (2002). "Laboratory Characterization of a Highly Weathered Old Alluvium in San Juan, Puerto Rico," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge.
- Zhang, G., Germaine, J. T., and Whittle, A. J. (2003a). "Effects of Fe-oxides cementation on the deformation characteristics of a highly weathered old alluvium in San Juan, Puerto Rico." *Soils and Foundations*, 43(1), 119-130.
- Zhang, G., Germaine, J. T., and Whittle, A. J. "Site characteristics of a highly weathered old alluvium in San Juan, Puerto Rico." *The 12th Panamerican Conference on Soil Mechanics and Geotechnical Engineering*, Cambridge, Massachusetts, USA.

Chapter 6

Summary, Conclusions, and Recommendations

6.1 Summary

Reliable predictions of ground deformations are important in evaluating the potential damage caused by underground construction on existing structures and utilities and effectiveness of mitigate strategies. The Río Piedras station of the new Tren Urbano transit system in San Juan, Puerto Rico presents an interesting example of a recent underground construction project involving a relative complex construction method (stacked-drift tunneling) in marginally stable ground, for which there is very little prior experience. During the construction, unexpectedly large ground deformations occurred and caused damage to several overlying buildings. Three-dimensional finite element analyses were necessary to simulate the stacked-drift construction process without making major judgment and assumptions such as the stress release ratio, as opposed to conventional 2-D (plane-strain) models. The construction sequence involved hand excavation of 16 separate drifts (including the grouting gallery) primarily supported by steel sets and timber-lagging, concrete infilling, and mitigation of ground movements through consolidation and compensation grouting. An initial 3-D FE model (using ABAQUS) for a single drift required nearly 3 days computation on a modern workstation¹ with an extravagant 1GB of RAM. There was little improvement in computational efficiency using a Cray supercomputer with eight processors due to highly uneven utilization of available processors, and 25% overall CPU utilization.

Alternative approaches to reduce computational requires, such as reducing the model size, have limited applicability if accurate solutions are to be achieved for problems such as tunnel construction, which contain features with multiple length scales from O[cm] (corresponding to the scale of grout injection or lining), through O[m] corresponding to drift heading dimensions, to O[100m] for the settlement trough. An efficient remeshing scheme (Section 4.3) was developed to improve discretization around the tunnel heading. This technique improves solution accuracy while maintaining the finite element model size, but has limited

¹ Year 2001, Digital Personal Workstation 600au, Alpha 21164 600MHz processor,

applicability when dealing with complex excavation sequences (such as multiple headings for the stacked drifts).

The underlying bottleneck for large-scale finite element solutions lies in the solution of the system of linearized equilibrium equations. Direct solution techniques are robust and mature, but cannot practically be applied to large-scale computations due to memory requirement and unfavorable scaling with problem sizes. Computations for tunnel problems indicate power law scaling of computation time with the number of elements, N_E , and the exponent $p = 1.514$. The standard alternatives are Krylov subspace-based, iterative solution methods. These are memory efficient and less sensitive to problem size ($p = 1.17 - 1.33$ for preconditioned conjugate gradient method), but do not guarantee convergent solutions. The preconditioning technique can sometimes accelerate the iteration process and/or help convergence. Therefore, direct solution methods are universally used in commercial finite element packages for their robustness, while iterative methods are becoming more popular (e.g. ABAQUS 6.4-1, 2003; ANSYS 5.1, 1995; Plaxis 3D-Tunnel 1.2, 2002).

Parallel computation is increasingly practical through Beowulf cluster computing, which offers an alternative to traditional supercomputers by aggregating computational power from dedicated interconnected commodity computers. Although the hardware can be easily set up, there is little available software which can make efficient use of these resources.

This thesis has made a detailed study of Domain Decomposition methods that offer a divide-and-conquer solution strategy for parallel computing platforms. An overlapping Domain Decomposition scheme commonly known as the Schwarz alternating scheme (Schwarz 1869) was first proposed over 100 years ago for solving partial differential equations in irregular-shaped domains (Section 4.4). Using this method, the original finite element model is decomposed into several smaller sub-domains which share overlapped regions. Calculations are performed by alternating between adjacent sub-domains and information is transmitted via shared overlaps. This scheme was implemented for its simplicity and can be readily integrated within existing finite element software. However, the method suffers from slow convergence, and is cumbersome for designing domain boundaries in tunnel excavation problems.

Substructuring (Przemieniecki 1963) is a non-overlapping domain decomposition that solves interface displacements between sub-domains. The method subdivides a complete structure into several smaller substructures, identifies interface between substructures, calculates interface stiffness contributions from each substructure (the Schur complement) through a condensation process, computes forces on these interfaces, and solves the displacement. This method is unsuitable for geotechnical engineering applications because the interface stiffness forms a dense matrix, and substructuring of a continuous soil mass produces large interface areas.

This research has implemented FETI (Finite Element Tearing and Interconnecting; Farhat and Roux, 1991), a non-overlapping domain decomposition algorithm developed for solving large-scale finite element models with parallel computers. In contrast to substructuring, forces instead of displacements are used to formulate the interface problem, which is then solved using iterative methods. The algorithm can be seen as a hybrid solution method that applies direct solution techniques on individual sub-domains, and iterative methods on the reduced interface problem formulated from mesh decomposition. The FETI method was initially evaluated by creative use of ABAQUS to compute stiffness matrices and force vectors of sub-domains, enabling the development effort to focus on the implementation of FETI and utilize existing capabilities of ABAQUS. This generated a program, *FETI-DD* (Section 4.5), which generates input files for ABAQUS to solve, parses ABAQUS output files to extract needed matrices, and performs FETI iterations on loosely interconnected computers using MPI for the explicit inter-processor communications. The *FETI_DD* program was evaluated by analyzing excavations in elastic soils; the solution was accurate, and reasonable parallel efficiency was achieved using five PCs interconnected through non-dedicated network (MIT-Net). However, the solution efficiency was hindered by interfacing with ABAQUS, which imposed unnecessary equation solves and I/O overhead.

Finally the FETI method was implemented and evaluated with an in-house developed non-linear finite element code *FETI_FEM*. (Section 4.6) In order to cope with material non-linearity, *FETI_FEM* couples the FETI method with Newton-Raphson non-linear iterative schemes and uses heuristics based on ABAQUS enhanced with feedback from the FETI solver (such as the iteration count for interface solutions).

By enabling the independence of the number of sub-domains from the number of processors, the *FETI_FEM* program provides great flexibility in the design of mesh decomposition (great influence on parallel computational efficiency). Load balancing of processors in *FETI_FEM* is achieved based on the apparent factorization computational efficiency during execution. This method does not assume the homogeneity of the participating processors, and it can therefore be used efficiently in a shared heterogeneous cluster. The program has no built-in requirements or limits on the number of participating processors, thus can be used on computer clusters or parallel computers with any arrangement of processors.

Two auxiliary programs known as *EXTRUDE* and *FEMPost* were developed to help in preparing the input files for *FETI_FEM* and generate outputs for visualization. *EXTRUDE* reads decomposed plane-strain finite element meshes written with ABAQUS syntax, extrudes plane-strain mesh, and performs decomposition in the extrusion direction. *FEMPost* reads the computational results scattered amongst processors participated in the calculation and generates output files for visualization with Tecplot¹, a commercially available visualization package.

The non-linear finite element code *FETI_FEM* demonstrates good parallel efficiency (Figure 4.58) and efficiently utilizes the Beowulf cluster, in the department of Civil and Environment Engineering (*Codex-hammer*), composed of 16 off-the-shelf computers. The program significantly reduces the solution time for modeling Río Piedras stacked-drift construction from 80 days (projected) to less than 3 days using 14 processors. To the Author's knowledge, this is the first application of the FETI method in the field of geomechanics, and the only application incorporating material non-linearity and evolving geometries (due to material removal and lining installation) for the FETI method. Chapter 5 presents the application of the non-linear 3-D analyses for the stacked-drift tunnel using the *FETI_FEM* program.

¹ Product of Tecplot, Inc. <http://www.tecplot.com>

Field measurements of surface settlements (particularly the benchmarks installed on the buildings above the construction) were closely examined and synthesized to correlate the measured surface settlements with the progression of the construction, including excavation, concrete infilling, and grouting activities.

Based on a simplified soil profile and material properties (obtained from extensive laboratory experiments), 2-D (plane-strain) finite element analyses were conducted to simulate drift construction procedures including drift-excavation, lining installation, and drift infilling (i.e. excluding grouting activities). Parametric studies were performed to identify important factors affecting the ground movement such as concrete drift infilling, construction sequences, yield condition, etc.

Three-dimensional finite element analyses were then performed with *FETI_FEM*. The finite element model contains more than 30,000 20-noded quadratic displacement-based elements with nearly 400,000 degrees of freedom, which makes the model at least one order of magnitude larger than any prior model in the literature of geotechnical engineering (Mestat et al. 2004). The large-scale FE model is subdivided into 168 sub-domains, with 60 stages to simulate the complex construction sequence. The damage potential to overlying structures was then evaluated from the computed distribution of ground movements.

6.2 Conclusions

The main conclusions of this research can be summarized as follows:

- Three-dimensional finite element analyses with reasonable resolution and modeling large-extent of soil domains can easily exceed practical computational limits of single processor computers. Efficient solutions can be obtained through parallel computation techniques. Beowulf computer clusters provide an affordable computing platform for such purposes.
- Direct solution methods are unsuitable for large-scale finite element modeling. For tunnel excavation problem, the computation time, T , can be correlated with the number of finite elements, N_E by a power law scaling with an empirical exponent $p = 1.514$. For direct methods, parallel efficiency is limited, and memory

consumption is significant. Therefore, the applicability of direct solution techniques is limited to small to medium sized problems.

- The *FETI_FEM* program provides an efficient parallel computation method for large-scale finite element analyses in geomechanics involving material non-linearities and evolving problem geometries. Solution efficiency is improved by 1) reducing the size of matrices (small sub-domains) factorized using direct methods, 2) using iterative solution techniques for solving interface forces, and 3) aggregating computing resources from interconnected computers. This solution strategy efficiently uses multiple processors and is insensitive to problem sizes, thus it is ideal for solving large-scale problems. The independence of the number of sub-domains from the number of processors is an important feature of *FETI_FEM*. This feature gives freedom to design mesh decompositions, enables one to control the size of interfaces (this controls the problem size feed into the direct and iterative solvers), and helps to achieve load balance in shared heterogeneous clusters.
- Mesh decomposition can significantly affect the computational efficiency by reducing the FETI iterations. The current application for the stacked-drift tunnel achieves high efficiency by avoiding sub-domain interfaces between dissimilar materials.
- The choice of preconditioner can also affect dramatically the convergence rate of the iterative interface solution. It has been found mechanical consistency condition (Rixen and Farhat 1999), always accelerates the convergence of the FETI iterations and should always be used as minimal overhead is imposed by incorporating mechanical consistency with other preconditioners. In contrast to prior recommendations (which favor lumped preconditioners), Dirichlet preconditioners generate shorter computation time than lumped preconditioners. This is due to different characteristics of the computing platform. Beowulf clusters tend to have higher latency or lower communication bandwidth than supercomputers, and FETI iteration counts become the critical component controlling the overall solution time. In contrast, supercomputers feature very efficient intercommunication links, and reduction in the amount of computation are more important to achieve lower

solution time. Therefore, prior recommendations are suitable for supercomputers, while Dirichlet preconditioners are recommended for Beowulf clusters.

- Although the thesis has solved one of the largest finite element models for an underground construction project, the resolution of the finite element mesh is still insufficient to address all aspects of the construction progress such as grout injection and individual support installation.
- Based on numerical simulations performed using plane-strain finite element models, the principal parameters affecting 2-D predictions of ground movements due to the stacked-drift construction are initial K_0 condition, drift infilling, and construction sequence.
 - There are significant uncertainties in the initial stress conditions at the Río Piedras site due to the complex stress history, weathering of the old alluvium, and spatial variability. The predicted ground settlements were found to vary by almost a factor of 2 (60 to 115mm) for assumed K_0 values in a modest range, $K_0 = 0.5 - 1.0$.
 - The concrete infilling is a characteristic feature of stacked-drift construction. The weight can account for 20 – 30mm of the total surface settlement (115mm). Clear correlations between the concreting and the surface ground loss were also observed in 3-D FE models.
 - The bottom-up construction sequence of the stacked drifts produced a deeper but narrower settlement trough than either the actual (or top-down) construction sequences. The differences reflect soil-structure interactions that were not appreciated during initial designs.
 - The top-down and the actual construction sequence produce shallower and wider settlement trough because the construction of drifts 7 and 8 essentially created a mat foundation.

- One major uncertainty in 2-D FE modeling for the stacked-drift construction is the representation of concurrently advancing drifts at various depths in opposite directions. It is a common practice to assume that only certain specified fraction of the initial stress is released prior to lining activation. Such considerations require major engineering judgment. It is also very difficult to provide a rational stress release ratio according to drifts advancing concurrently at various depths in opposite directions. Thus, three-dimensional simulations are essential to understand the soil-structure interactions for stacked-drift construction. The end-of-construction field settlements predicted by the 3-D FE analyses were found to be essentially equivalent to 2-D plane-strain model using a stress release ratio of approximately 85%.
- The computed surface settlement troughs have been used to evaluate the potential damage to several buildings. It was found that buildings on both sides of the Ponce de Leon Avenue are susceptible to moderate damage based on FE predictions. The most critical locations are found on the distant side of buildings on the east-side of the cavern.

6.3 Recommendations

There are various improvements, extensions, and refinements that can be proposed for the current FETI_FEM implementation:

- Undrained analyses of low permeability clays are routinely performed in geotechnical engineering. Currently, material incompressibility severely degrades the convergence characteristics of the iterative interface solution in FETI. Based on numerical experiments, finite element models with close to incompressible medium (i.e. $\nu \rightarrow 0.5$) require 3 – 5 more iterations to achieve convergent solutions than with compressible medium (i.e. $\nu < 0.45$). Vereecke et al. (2003) have recently proposed an extension to the FETI method to improve the solution efficiency with incompressible medium.
- Many geotechnical applications involve coupling of flow and deformation within the soil mass. There have been very few prior studies linking flow and deformation for tunneling in 3-D FE analyses using either simplified soil model (e.g. linear elasticity)

or very crude FE mesh, most likely due to enormous computation time. FETI_FEM can be enhanced with deformation-flow coupled elements to improve the modeling capabilities involving flow-deformation coupled analyses.

- The current *FETI_FEM* implementation uses a Newton-Raphson iterative scheme. However, it may be advantageous to use explicit non-linear solution techniques, because the number of iterations for interface problems depends on the residual forces to be equilibrated. Explicit non-linear solution techniques tend to adopt smaller sub-steps (thus smaller residual forces) than Newton-Raphson schemes (and can even be adjusted to control residual errors; Abbo and Sloan, 1996). Therefore, it is possible coupling explicit non-linear solution technique with FETI method will improve the computational efficiency.
- For finite element models with evolving geometries, a single sub-domain can be separated into two or more entities due to material removal. This condition is currently avoided by careful design of the mesh decomposition, and is not currently considered in the programming. If sub-domain separation conditions are incorporated into the program, this will allow extended flexibility in the design of mesh decompositions and enable the use of automated mesh decomposition algorithms within the current programming model.
- Manual mesh decomposition can be a time-consuming task for unstructured meshes, and prohibits the optimization of computational efficiency through improving mesh decompositions. There are freely available libraries for mesh decompositions (e.g. METIS, Karypis and Kumar 1995), but these do not consider the inhomogeneous material properties. By improving automated mesh decomposition techniques with explicit considerations of the material heterogeneity, the transition from traditional finite element modeling to domain decomposed finite element modeling can be eased, and solution efficiency can be further improved through refinements of the mesh decomposition.

- Long calculations are prone to accidents (i.e. power failure, component failure, etc.) Therefore, it is highly recommended that the *FETI_FEM* program includes the functionality to continue an interrupt analysis.
- Currently the load balance is done during the initial task distribution phase, (which distributes all sub-domains to participating processors), based on the apparent factorization efficiency of sub-domains. Load imbalance can develop during the course of calculations due to 1) changed of load distribution from tasks of other users on the same computing system, and 2) changed sub-domain sizes due to material addition or removal. Therefore, it is recommended that load rebalancing be carried out for every few analysis steps to ensure the maximum parallel efficiency. This functionality can be implemented in conjunction with the continuation capability.
- Incomplete Cholesky preconditioners have been proposed for the FETI interface solution (Charmpis and Papadrakakis 2002). They appear to provide better convergence than lumped preconditioners, and are more computationally efficient and less memory demanding than Dirichlet preconditioners. Thus, further testing of these preconditioners should be carried out for *FETI_FEM*.
- To improve the modeling of the stacked-drift construction for the new Río Piedras station, better constitutive models for the weathered alluvium soil and understanding the impact of grout injections are necessary.
 - The Río Piedras station construction incorporates two grouting programs to control the ground movement: consolidation grouting (fracture grouting) and compensation grouting (permeation and/or fracture grouting). Representation of these processes requires the development of realistic models that can represent the spatial distribution of the injected grout through single injection and multiple injections, and their effects on the nature soil.
 - Further systematic research is needed to understand the immediate change and the time-dependent change (e.g. consolidation) of mechanical properties

of the grouted soil, potentially controlled by 1) the mechanical properties of the grout mixture, 2) the mechanical properties of the original soil, 3) the spatial distance to the grouting point, 4) injection variables.

- It was observed by Zhang (2002) that destructured weathered alluvium soils swell with access to water. Therefore, it is important to identify if the grouting process destructures the alluvium soils and provides water that make disintegrated weathered alluvium swell.

- The weathered alluvial soil in Río Piedras showed unusual characteristics and properties as reported by Zhang (2002). The weathered alluvium is structured, and exhibits a large transition in mechanical properties due to mechanical (or chemical) dis-aggregation. The current analysis considers the alluvium in its intact state. The intact alluvium comprises a very stiff cemented soil, inter-aggregated with large pore space and high hydraulic conductivity. Once the intact structure is broken down, the pore space collapses and reduces hydraulic conductivity dramatically. Other changes on mechanical properties are also possible due to destructuring (e.g. from brittle behavior to ductile behavior). On going research (Nikolinakou, 2004) aims to develop a constitutive model that can describe realistically these transitions in material behavior.

References

- Abbo, A. J., and Sloan, S. W. (1996). "Automatic load stepping algorithm with error control." *International Journal for Numerical Methods in Engineering*, 39(10), 1737-1759.
- Abu-Farsakh, M. Y., and Voyiadjis, G. Z. (1999). "Computational model for the simulation of the shield tunneling process in cohesive soils." *International Journal for Numerical & Analytical Methods in Geomechanics*, 23(1), 23-44.
- Addenbrooke, T. I., Potts, D. M., and Puzrin, A. M. (1997). "Influence of pre-failure soil stiffness on the numerical analysis of tunnel construction." *Geotechnique*, 47(3), 693-712.
- Akagi, H., and Komiya, K. (1996). "Finite Element Simulation of Shield Tunnelling Processes in Soft Ground." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 447-452.
- Amestoy, P., Davis, T. A., and Duff, I. S. (1996). "An approximate minimum degree ordering algorithm." *SIAM Journal on Matrix Analysis and Applications*, 17, 886-905.
- Amestoy, P. R., Duff, I. S., and L'Excellent, J.-Y. (2000). "Multifrontal parallel distributed symmetric and unsymmetric solvers." *Comput. Methods Appl. Mech. Eng.*, 501-520.
- Arnoldi, W. E. (1951). "The principle of minimized iteration in the solution of the matrix eigenvalue problem." *Quart. Appl. Math.*, 9, 17-29.
- Ashcraft, C., Grimes, R., Lweis, J., Peyton, B., and Simon, H. (1987). "Progress in sparse matrix methods for large sparse linear systems on vector supercomputers." *International Journal of Supercomputer Applications*, 1, 10-30.
- Atkinson, J. H., and Potts, D. M. (1977). "Subsidence above shallow tunnels in soft ground." *Journal of the Geotechnical Engineering Division - American Society of Civil Engineers*, 103(4), 307-325.
- Attewell, A. P., and Woodman, J. P. (1982). "Predicting the dynamics of ground settlement and its derivatives caused by tunnelling in soil." *Ground Engineering*, 15(8), 13-36.
- Atzl, G. V., and Mayr, J. K. "FEM-analysis of Heathrow NATM trial tunnel." *Numerical Methods in Geotechnical Engineering*, Rotterdam, 195-201.
- Augarde, C. E., and Burd, H. J. (2001). "Three-dimensional finite element analysis of lined tunnels." *International Journal for Numerical and Analytical Methods in Geomechanics*, 25, 243-262.
- Augarde, C. E., Wisser, C., and Burd, H. J. (1999). "Numerical Modelling of Tunnel Installation Procedures." *Numerical Methods in Geomechanics*, Pande, Pietruszczak, and Schweiger, eds., 329-334.
- Aydan, O., Kyoya, T., Ichikawa, Y., Kawamoto, T., Ito, T., and Shimizu, Y. (1988). "Three-dimensional Simulation of an Advancing Tunnel Supported with Forepoles, Shotcrete, Steel Ribs and Rockbolts." *Numerical Methods in Geomechanics*, G. Swoboda, ed, 1481-1486.
- Babuska, I., and Rheinboldt, W. C. (1978). "A posteriori error estimates for the finite element method." *International Journal for Numerical Methods in Engineering*, 12, 1597-1615.
- Bathe, K. J. (1996). *Finite Element Procedures*, Prentice Hall.
- Bernat, S., Cambou, B., and Santosa, P. (1997). "Modeling of Soil-Structure Interaction During Tunnelling in Soft Soil." *Computer Methods and Advances in Geomechanics*, 1377-1382.
- Bhardwaj, M., Day, D., Farhat, C., Lesoinne, M., Pierson, K., and Rixen, D. (2000). "Application of the FETI method to ASCI problems - scalability results on 1000

- processors and discussion of highly heterogeneous problems." *International Journal for Numerical Methods in Engineering*, 47(1-3), 513-535.
- Blackford, L. S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongara, J., Hammarling, S., Henry, G., Petitet, A., Stanley, K., Walker, D., and Whaley, R. C. (1997). *Scalapack Users' Guide*, Society for Industrial & Applied Mathematics.
- Bobet, A. (2001). "Analytical solutions for shallow tunnels in saturated ground." *Journal of Engineering Mechanics*, 127(12), 1258-1266.
- Boscardin, M. D., and Cording, E. J. (1989). "Building response to excavation-induced settlement." *Journal of Geotechnical Engineering*, 115(1), 1-21.
- Bowers, K. H. (1997). "An appraisal of the New Austrian Tunnelling Method in soil and weak rock," PhD Thesis, The University of Leeds.
- Burd, H. J., Houlsby, G. T., Augarde, C. E., and Liu, G. (2000). "Modelling tunnelling-induced settlement of masonry buildings." *Proceedings of the Institution of Civil Engineers Geotechnical Engineering*, 143(1), 17-29.
- Burland, J. B. (1989). "Small is Beautiful - the stiffness of soils at small strains." *Canadian Geotechnical Journal*, 26(4), 499-516.
- Burland, J. B. "Assessment of risk of damage to buildings due to tunnelling and excavations." *1st Int. Conf. on Earthquake Geotechnical Engineering*, Tokyo, Japan.
- Burland, J. B., and Wroth, C. P. "Settlement of buildings and associated damage." *Conf. Settlement of Structures*, Cambridge, 611-654.
- Canetta, G., Cavagna, B., and Nova, R. (1996). "Experimental and Numerical Tests on the Excavation of a Railway Tunnel in Grouted Soil in Milan." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 479-484.
- Charpis, D. C., and Papadrakakis, M. (2002). "Enhancing the performance of the FETI method with preconditioning techniques implemented on clusters of networked computers." *Computational Mechanics*, 30(1), 12-28.
- Chatzigiannelis, Y., and Whittle, A. J. (2003). "Deformation caused by shallow tunneling in cross-anisotropic soil." *submitted for publication*.
- Cheng, J.-H., and Kikuchi, N. (1986). "Mesh Re-zoning technique for finite element simulations of metal forming processes." *International Journal for Numerical Methods in Engineering*, 23(2), 219-228.
- Clough, G. W., and Schmidt, B. (1981). "Design and performance of excavations and tunnels in soft clay." *Soft Clay Engineering*, Elsevier Scientific Publishing Company, Amsterdam, The Netherlands, 569-634.
- Cuthill, E., and McKee, J. "Reducing the bandwidth of sparse symmetric matrices." *24th ACM National Conference*, 157-172.
- Deere, D. U. (1955). "Engineering Properties of the Pleistocene and recent sediments of the San Juan Bay area, Puerto Rico," University of Illinois, Urbana.
- Deming, P. W., Lacy, H. S., and Chang, D. K. "Ground freezing for tunnel face stabilization." *North American Tunneling '00*.
- Demmel, J. W., Gilbert, J. R., and Li, X. S. (1999). "SuperLU Users' Guide." 58.
- Dongarra, J. (1997). "Online Lecture Notes, <http://www.cs.utk.edu/~dongarra/WEB-PAGES/SPRING-2000/lect08.pdf>."
- Duff, I., and Reid, J. (1983). "The multifrontal solution of indefinite sparse symmetric linear equations." *ACM Trans. Mathematical Software*, 9, 302-325.
- Duff, I. S. (1996). "Sparse numerical linear algebra: direct methods and preconditioning." *Technical Report TR/PA/96/22*.

- Dureisseix, D., and Farhat, C. (2001). "A numerically scalable domain decomposition method for the solution of frictionless contact problems." *International Journal for Numerical Methods in Engineering*, 50(12), 2643-2666.
- Eberhardt, E. (2001). "Numerical modelling of three-dimension stress rotation ahead of an advancing tunnel face." *International Journal of Rock Mechanics and Mining Sciences*, 38(4), 499-518.
- Einstein, H. H., and Schwartz, C. W. (1979). "Simplified analysis for tunnel supports." *Journal of the Geotechnical Engineering Division-ASCE*, 105(4), 499-518.
- Farhat, C., Chen, P. S., Mandel, J., and Roux, F. X. (1998a). "The two-level FETI method Part II: Extension to shell problems, parallel implementation and performance results." *Computer Methods in Applied Mechanics and Engineering*, 155(1-2), 153-179.
- Farhat, C., Crivelli, L., and Roux, F. X. (1994). "Extending Substructure Based Iterative Solvers to Multiple Load and Repeated Analyses." *Computer Methods in Applied Mechanics and Engineering*, 117(1-2), 195-209.
- Farhat, C., and Geradin, M. (1998). "On the general solution by a direct method of a large-scale singular system of linear equations: Application to the analysis of floating structures." *International Journal for Numerical Methods in Engineering*, 41(4), 675-696.
- Farhat, C., Lacour, C., and Rixen, D. (1998b). "Incorporation of linear multipoint constraints in substructure based iterative solvers. Part 1: A numerically scalable algorithm." *International Journal for Numerical Methods in Engineering*, 43(6), 997-1016.
- Farhat, C., Lesoinne, M., and Pierson, K. (2000). "A scalable dual-primal domain decomposition method." *Numerical Linear Algebra with Applications*, 7(7-8), 687-714.
- Farhat, C., and Mandel, J. (1998). "The two-level FETI method for static and dynamic plate problems Part I: An optimal iterative solver for biharmonic systems." *Computer Methods in Applied Mechanics and Engineering*, 155(1-2), 129-151.
- Farhat, C., and Roux, F. X. (1991). "A Method of Finite-Element Tearing and Interconnecting and Its Parallel Solution Algorithm." *International Journal for Numerical Methods in Engineering*, 32(6), 1205-1227.
- Farhat, C., and Roux, F.-X. (1994). "Implicit Parallel Processing in Structural Mechanics." *Computational Mechanics Advances* 2, 1-124.
- Fish, J., and Shek, K. L. (1999). "Computational Aspects of Incrementally Objective Algorithms for Large Deformation Plasticity." *International Journal for Numerical Methods in Engineering*, 44, 839-851.
- Fujita, K. "Underground construction, tunnel, underground transportation." *Proceedings 12th International Conference on Soil Mechanics and Foundation Engineering*, Rio de Janeiro, 2159-2176.
- George, A. (1973). "Nested dissection of a regular finite element mesh." *SIAM Journal Numer. Anal.*, 10, 345-363.
- Gosselet, P., Rey, C., and Rixen, D. J. (2003). "On the initial estimate of interface forces in FETI methods." *Computer Methods in Applied Mechanics and Engineering*, 192(25), 2749-2764.
- Grant, R. J., and Taylor, R. N. (2000). "Tunnelling-induced ground movements in clay." *Proceedings of the Institution of Civil Engineers Geotechnical Engineering*, 143(1), 43-55.
- Harris, D. I., Pooley, A. J., Menkiti, C. O., and Stephenson, J. A. "Construction of low level tunnels below Waterloo Station with compensation grouting for Jubilee line extension." *Geotechnical aspects of underground construction in soft ground*, 361-366.
- Henn, R. W. (1996). *Practical Guide to Grouting of Underground Structures*, American Society of Civil Engineers.

- Hsieh, Y.-M., and Whittle, A. J. "A computational strategy for solving three-dimensional tunnel excavation problems." *Second M.I.T. Conference on Computational Fluid and Solid Mechanics*, Cambridge, Massachusetts, USA.
- Hu, Y., and Randolph, M. F. (1998a). "H-adaptive FE analysis of elasto-plastic non-homogeneous soil with large deformation." *Computers and Geotechnics*, 23(1-2), 61-83.
- Hu, Y., and Randolph, M. F. (1998b). "Practical numerical approach for large deformation problems in soil." *International Journal for Numerical and Analytical Methods in Geomechanics*, 22(5), 327-350.
- Hu, Y., Randolph, M. F., and Watson, P. G. (1999). "Bearing response of skirted foundation on nonhomogeneous soil." *Journal of Geotechnical and Geoenvironmental Engineering*, 125(11), 924-935.
- Jeremic, B., and Sture, S. (1997). "Implicit integrations in elastoplastic geotechnics." *Mechanics of Cohesive-Frictional Materials*, 2(2), 165-183.
- John, V. (2000). "A numerical study of a posteriori error estimators for convection-diffusion equations." *Computer Methods in Applied Mechanics and Engineering*, 190(5-7), 757-781.
- Karakus, M., and Fowell, R. J. (2003). "Effects of different tunnel face advance excavation on the settlement by FEM." *Tunnelling and Underground Space Technology*, 18(5), 513-523.
- Karypis, G., and Kumar, V. (1995). "A fast and high quality multilevel scheme for partitioning irregular graphs." 95-035, Department of Computer Science, University of Minnesota, Minneapolis.
- Karypis, G., and Kumar, V. (1998). "Multilevelk-way Partitioning Scheme for Irregular Graphs." *Journal of Parallel and Distributed Computing*, 48(1), 96-129.
- Kaye, C. A. (1959). "Coastal Geology of Puerto Rico: (A) Geology of the San Juan Metropolitan Area." US Government Printing Office, Washington, D.C.
- Kimura, T., and Mair, R. J. "Centrifugal testing of model tunnels in soft clay." *Proceedings of 10th International Conference on Soil Mechanics and Foundation Engineering*, Stockholm, 319-322.
- Komiya, K., Soga, K., Akagi, H., Hagiwara, T., and Bolton, M. D. (1999). "Finite Element Modelling of Excavation and Advancement Processes of a Shield Tunnelling Machine." *Soils and Foundations*, 39(3), 37-52.
- LaFonta, G., and Person, T. (1999). "Puerto Rico real-time control of compensation grouting with the CYCLOPS system." *Geotechnical News*, 27-33.
- Lawson, C. L., Hanson, R. J., Kincaid, D., and Krogh, F. T. (1979). "Basic linear algebra subprograms for FORTRAN usage." *{ACM} Transactions on Mathematical Software*, 5(3), 308-323.
- Lee, K. M., and Rowe, R. K. (1989). "Deformations caused by surface loading and tunnelling: the role of elastic anisotropy." *Geotechnique*, 39(1), 125-140.
- Lee, K. M., and Rowe, R. K. (1990a). "Finite element modelling of the three-dimensional ground deformations due to tunnelling in soft cohesive soils. Part 2 - results." *Computers & Geotechnics*, 10(2), 111-138.
- Lee, K. M., and Rowe, R. K. (1990b). "Finite element modelling of the three-dimensional ground deformations due to tunnelling in soft cohesive soils. Part I - method of analysis." *Computers & Geotechnics*, 10(2), 87-109.
- Lee, K. M., Rowe, R. K., and Lo, K. Y. (1992). "Subsidence Owing to Tunnelling. I. Estimating the Gap Parameter." *Can. Geotech. J.*, 29, 929-940.
- Li, X. S. (1996). "Sparse Gaussian elimination on high performance computers," Ph.D. dissertation, U. C. Berkeley.

- Li, X. S., and Demmel, J. W. "A scalable sparse direct solver using static pivoting." *Ninth SLAM Conference on Parallel Processing for Scientific Computing*, San Antonio, Texas, U.S.A.
- Li, X. S., and Demmel, J. W. (2003). "SuperLU DIST: A Scalable Distributed-Memory Sparse Direct Solver for Unsymmetric Linear Systems." *ACM Transactions on Mathematical Software*, 29(2), 110-140.
- Liu, J. W.-H. (1985). "Modification of the minimum degree algorithm by multiple elimination." *ACM Trans. Mathematical Software*, 11, 141-153.
- Loganathan, N., and Poulos, H. G. (1998). "Analytical prediction for tunneling-induced ground movements in clays." *Journal of Geotechnical and Geoenvironmental Engineering*, 124(9), 846-856.
- Macklin, S. R. (1999). "Prediction of volume loss due to tunnelling in overconsolidated clay based on heading geometry and stability number." *Ground Engineering*, 32(4), 30-33.
- Mair, R. J., and Taylor, R. N. "Bored tunnelling in the urban environment." *14th International Conference on Soil Mechanics and Foundation Engineering*, Hamburg.
- Mair, R. J., Taylor, R. N., and Bracegirdle, A. (1993). "Subsurface Settlement Profiles above Tunnels in Clays." *Geotechnique*, 43(2), 315-320.
- Mair, R. J., Taylor, R. N., and Burland, J. B. (1996). "Prediction of Ground Movements and Assessment of Risk of Building Damage due to Bored Tunnelling." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 713-718.
- Malkus, D. S., and Hughes, T. J. R. (1978). "Mixed finite element methods em dash reduced and selective integration techniques: a unification of concepts." *Computer Methods in Applied Mechanics and Engineering*, 15(1), 63-81.
- Matthies, H., and Strang, G. (1979). "Solution of non-linear finite element equations." *International Journal for Numerical Methods in Engineering*, 14(11), 1613-1626.
- Mestat, P., Bourgeois, E., and Riou, Y. (2004). "Numerical Modelling of embankments and underground works." *Computers and Geotechnics*, 31, 227-236.
- Moh, Z.-C., Hwang, R. N., and Ju, D. H. (1996). "Ground Movements Around Tunnels in Soft Ground." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 725-730.
- Morrison, J. A., Madsen, P. H., and Carayol, S. "Ground Control Program for the Rio Piedras Project, Tren Urbano Program, San Juan, Puerto Rico." *Rapid Excavation and Tunneling Conference*, Orlando, U.S.A.
- Mroueh, H., and Shahrou, I. (2002). "Three-dimensional finite element analysis of the interaction between tunneling and pile foundations." *International Journal for Numerical and Analytical Methods in Geomechanics*, 26(3), 217-230.
- Mroueh, H., and Shahrou, I. (2003). "A full 3-D finite element analysis of tunneling-adjacent structures interaction." *Computers and Geotechnics*, 30(3), 245-253.
- Nakai, T., and Xu, L. (1995). "3D and dilatancy effects on settlements and earth pressures over lowering basement." *Soils & Foundations*, 35(4), 73-84.
- Nakai, T., Xu, L. M., and Yamazaki, H. (1995). "Prediction of Surface Settlement Profiles due to Tunnel Excavation: Model Tests and Numerical Analyses." *Numerical Methods in Geomechanics*, Pande and Pietruszczak, eds., 495-500.
- Ng, C. W. W., and Lee, G. T. K. (2002). "A three-dimensional parametric study of the use of soil nails for stabilising tunnel faces." *Computers and Geotechnics*, 29(8), 673-697.
- Oettl, G., Stark, R. F., and Hofstetter, G. (1998). "Comparison of elastic-plastic soil models for 2D FE analyses of tunnelling." *Computers & Geotechnics*, 23(1-2), 19-38.

- Ohtsu, H., Ohnishi, Y., Taki, H., and Kamemura, K. (1999). "A Study on Problems Associated with Finite Element Excavation Analysis by the Stress-flow Coupled Method." *Int. J. Numer. Anal. Meth. Geomech.*, 23, 1473-1492.
- O'Reilly, M. P., and New, M. B. "Settlements above tunnels in the United Kingdom - their magnitudes and prediction." *Tunnelling*, 173-181.
- Panet, M., and Guenot, A. "Analysis of convergence behind the face of a tunnel." *Tunnelling '82, Papers Presented at the 3rd International Symposium.*, Brighton, Engl, 197-204.
- Park, K. C., R., J. M., and Felippa, C. A. (1997). "An Algebraically Partitioned FETI Method for Parallel Structural Analysis: Algorithm Description." *International Journal for Numerical Methods in Engineering*, 40(15), 2717-2737.
- Peck, R. B. (1969). "Deep excavations and tunneling in soft ground." University of Illinois, Urbano, 225-290.
- Pinto, F. (2000). *Analytical methods to interpret ground deformations due to soft ground tunneling.*
- Poole, G., Liu, Y.-C., and Mandel, J. (2003). "Advancing analysis capabilities in Ansys through solver technology." *Electronic Transactions on Numerical Analysis*, 15, 106-121.
- Przemieniecki, J. S. (1963). "Matrix structural analysis of substructures." *Am. Inst. Aero. Astro. J.*, 1, 138-147.
- Rey, C., and Lene, F. "Reuse of Krylov Spaces in the Solution of Large-scale Nonlinear Elasticity Problems." *Ninth International Conference on Domain Decomposition Methods*, 466-470.
- Rixen, D. J., and Farhat, C. (1999). "A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems." *International Journal for Numerical Methods in Engineering*, 44(4), 489-516.
- Rowe, R. K., and Lee, K. M. (1992). "Subsidence Owing to Tunnelling. II. Evaluation of a Prediction Technique." *Can. Geotech. J.*, 29, 941-954.
- Rowe, R. K., Lo, K. Y., and Kack, G. J. (1983). "Method of estimating surface settlement above tunnels constructed in soft ground." *Canadian Geotechnical Journal*, 20(1), 11-22.
- Saad, Y. (1996). *Iterative Methods for Sparse Linear Systems*, PWS Publishing Co.
- Saad, Y., and Vorst, H. A. v. d. (2000). "Iterative solution of linear systems in the 20-th century." *Journal of Computational and Applied Mathematics*, 123(1-2), 1-33.
- Sagaseta, C. (1987). "Analysis of undrained soil deformation due to ground loss." *Geotechnique*, 37(3), 301-320.
- Sagaseta, C., Verruijt, A., and Booker, J. R. (1998). "Surface settlements due to deformation of a tunnel in an elastic half plane." *Geotechnique*, 48(5), 709-713.
- Schwarz, H. A. (1869). "Uber einige Abbildungsaufgaben." *Ges. Math. Abb.*, 11, 65-83.
- Schweiger, H. F., Schuller, H., and Pottler, R. (1997). "Some Remarks on 2-D-models for Numerical Simulation of Underground Constructions with Complex Cross Sections." *Computer Methods and Advances in Geomechanics*, 1303-1308.
- Sheng, D., and Sloan, S. W. (2003). "Time stepping schemes for coupled displacement and pore pressure analysis." *Computational Mechanics*, 31(1-2 SPEC), 122-134.
- Simo, J. C., and Taylor, R. L. (1985). "Consistent tangent operators for rate-independent elastoplasticity." *Computer Methods in Applied Mechanics and Engineering*, 48(1), 101-118.
- Simpson, B., Atkinson, J. H., and Jovicic, V. (1996). "The Influence of Anisotropy on Calculations of Ground Settlements Above Tunnels." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 591-594.
- Sloan, S. W., and Abbo, A. J. (1999). "Biot consolidation analysis with automatic time stepping and error control. Part 2: Applications." *International Journal for Numerical and Analytical Methods in Geomechanics*, 23(6), 493-529.

- Smith, B., Bjorstad, P., and Gropp, W. (1996). *Domain Decomposition : Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press; (June 13, 1996).
- Strack, O. E., and Verruijt, A. (2002). "A complex variable solution for a deforming buoyant tunnel in a heavy elastic half-plane." *International Journal for Numerical and Analytical Methods in Geomechanics*, 26(12), 1235-1252.
- Susila, E., and Hryciw, R. D. (2003). "Large displacement FEM modelling of the cone penetration test (CPT) in normally consolidated sand." *International Journal for Numerical and Analytical Methods in Geomechanics*, 27(7), 585-602.
- Swoboda, G. "Finite element analysis of New Austrian Tunneling Method (NATM)." *Proc. 3rd Int. Conf. Numer. Methods Geomech.*, Aachen, 581-586.
- Taguchi, Y., Hada, M., Kagawa, K., and Hara, T. "Soil nailing technique in tunnel support." *Proceedings of the 6th Engineering Foundation Conference on Shotcrete for Underground Support, May 2-6 1993*, Niagara-on-the-Lake, Ont, Can, 158-165.
- Tsuchiyama, S., Hayakawa, M., Shinokawa, T., and Konno, H. (1988). "Deformation Behaviour of the Tunnel Under the Excavation of Crossing Tunnel." *Numerical Methods in Geomechanics*, G. Swoboda, ed., 1591-1596.
- Urquiza, L. J. (2000). "Ground deformations caused by the construction of the Rio Piedras underground station using the stacked drift tunneling method," University of Puerto Rico, Mayaguez.
- Vassilev, V. H., and Hristov, T. N. (1988). "Influence of the heading face and a two-dimensional calculation model of tunnel linings." *Numerical Methods in Geomechanics*, G. Swoboda, ed., 1551-1555.
- Vereecke, B., Bavestrello, H., and Dureisseix, D. (2003). "An extension of the FETI domain decomposition method for incompressible and nearly incompressible problems." *Computer Methods in Applied Mechanics and Engineering*, 192(31-32), 3409-3429.
- Vermeer, P. A. (2001). "On a smart use of 3D-FEM in Tunneling." *Plaxis Bulletin*, 11, 2-6.
- Verruijt, A. (1997). "Complex variable solution for a deforming circular tunnel in an elastic half-plane." *International Journal for Numerical and Analytical Methods in Geomechanics*, 21(2), 77-89.
- Verruijt, A., and Booker, J. R. (1996). "Surface settlements due to deformation of a tunnel in an elastic half plane." *Geotechnique*, 46(4), 753-756.
- Wagner, H., and Schuster, A. (1988). "Geonumerical Computations for the Determination of Critical Deformations in Shallow Tunnelling." *Numerical Methods in Geomechanics*, G. Swoboda, ed., 1531-1536.
- Watson, W. R., and Storaasli, O. O. (2000). "Application of NASA general-purpose solver to large-scale computations in aeroacoustics." *Advances in Engineering Software*, 31(8), 555-561.
- Weinberg, D. J. (2000). "Performance assessment of NE/Nastran's new sparse direct and iterative solvers." *Advances in Engineering Software*, 31(8), 547-554.
- Whittle, A. J. (1987). "A constitutive model for overconsolidated clays with application to the cyclic loading of friction piles," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge.
- Yoo, C., and Shin, H.-K. (2003). "Deformation behaviour of tunnel face reinforced with longitudinal pipes--laboratory and numerical investigation." *Tunnelling and Underground Space Technology*, 18(4), 303-319.
- Yu, H. S., and Rowe, R. K. (1999). "Plasticity solutions for soil behaviour around contracting cavities and tunnels." *International Journal for Numerical & Analytical Methods in Geomechanics*, 23(12), 1245-1279.

- Zhang, G. (2002). "Laboratory Characterization of a Highly Weathered Old Alluvium in San Juan, Puerto Rico," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge.
- Zhang, G., Germaine, J. T., and Whittle, A. J. (2003a). "Effects of Fe-oxides cementation on the deformation characteristics of a highly weathered old alluvium in San Juan, Puerto Rico." *Soils and Foundations*, 43(1), 119-130.
- Zhang, G., Germaine, J. T., and Whittle, A. J. "Site characteristics of a highly weathered old alluvium in San Juan, Puerto Rico." *The 12th Panamerican Conference on Soil Mechanics and Geotechnical Engineering*, Cambridge, Massachusetts, USA.
- Zienkiewicz, O. C., Boroomand, B., and Zhu, J. Z. (1999). "Recovery procedures in error estimation and adaptivity Part I: Adaptivity in linear problems." *Computer Methods in Applied Mechanics and Engineering*, 176(1-4), 111-125.
- Zienkiewicz, O. C., Emson, C., and Bettess, P. (1983). "Novel Boundary Infinite Element." *International Journal for Numerical Methods in Engineering*, 19(3), 393-404.
- Zienkiewicz, O. C., Kelly, D. W., and Bettess, P. (1977). "Coupling of finite-element method and boundary solution procedures." *International Journal for Numerical Methods in Engineering*, 11(2), 355-375.
- Zienkiewicz, O. C., Taylor, R., and Too, J. (1971). "Reduced integration technique in general analysis of plates and shells." *International Journal of Numerical Methods in Engineering*, 3(2), 275-290.
- Zienkiewicz, O. C., and Zhu, J. Z. (1987). "A simple error estimator and adaptive procedure for practical engineering analysis." *International Journal for Numerical Methods in Engineering*, 24(2), 337-357.

APPENDIX

List of appendices

APPENDIX A	NUMERICAL AND ANALYTICAL MODELING OF GROUND DEFORMATIONS DUE TO SHALLOW TUNNELING IN SOFT SOILS.....	323
APPENDIX B	IMPLEMENTATION OF REMESHING ALGORITHM FOR EFFICIENT TUNNELING ANALYSIS	331
APPENDIX C	RECOMMENDED PRACTICE FOR BEOWULF CLUSTERS.....	337
APPENDIX D	SPARSE MATRIX STORAGE	341
APPENDIX E	ISOPARAMETRIC ELEMENT FORMULATION.....	345
APPENDIX F	SELECTIVE-REDUCED ELEMENT INTEGRATION	353
APPENDIX G	POST PROCESSING PROGRAM FOR FETI_FEM.....	357
APPENDIX H	SURFACE SETTLEMENTS OF RÍO PIEDRAS STATION CONSTRUCTION	361
APPENDIX I	VOLUME LOSS CALCULATION FROM FINITE ELEMENT SOLUTIONS ...	369
APPENDIX J	BUILDING DAMAGE ASSESSMENT CHARTS.....	373
REFERENCE	380

Appendix A

Numerical and Analytical Modeling of Ground Deformations due to Shallow Tunneling in Soft Soils¹

A.1 Abstract

This paper illustrates the role of analytical solutions in evaluating the settlements caused by tunnel excavation in soft soils. Analytical solutions have been obtained for shallow tunnels in isotropic and cross-anisotropic linear soil half-plane. These solutions provide a frame of reference for establishing numerical accuracy in FE analyses and for understanding the role of actual non-linear, inelastic soil behavior on far-field ground movements.

Keywords: Tunnel; Ground deformation; Constitutive model; Elasticity; Anisotropy; Non-linearity

A.2 Introduction

The prediction and mitigation of damage caused by construction-induced ground movements represents a major factor in the design of tunnels in congested urban environments. This is an especially important problem for shallow tunnels excavated in soft soils, where expensive remedial measures such as compensation grouting or structural underpinning must be considered prior to construction.

Ground movements inevitably arise from changes in soil stresses around the tunnel face and overexcavation of the final tunnel cavity. Sources of movements are closely related to the method of tunnel construction ranging from:

- (a) closed-face systems such as tunnel boring machines (with earth pressure or slurry shields), where overcutting occurs around the face and shield ('tail void'), but local

¹ Published in the First MIT Conference on Computational Fluid and Solid Mechanics, June 12 – 15, 2001. By A.J. Whittle L, Y.M. Hsieh, F. Pinto, Y. Chatzigiannelis.

ground loss is constrained by grout injected between the soil and precast lining system; to

- (b) open-face systems (such as the New Austrian Tunneling Method, NATM) where ground loss around the heading is minimized by expeditious installation of lining systems in contact with the soil (typically steel rib or lattice girder and shotcrete) with additional face support provided by a shield or other mechanical reinforcement (soil nails, sub-horizontal jet grouting, etc.).

In all cases, it is easy to appreciate the complexity of the mechanisms causing ground movement and their close relationship with construction details, especially given the non-linear, time-dependent mechanical properties of soils and their linkage to groundwater flows. Indeed, this complexity has encouraged the widespread use of numerical analyses, particularly non-linear finite element methods, over a period of more than 30 years (see review by Gioda and Swoboda, 1999).

However, it is less clear how details of the construction methods or soil properties affect the magnitude and distribution of far field movements and hence, the potential damage to overlying structures. Although there have been some notable advances in numerical modeling of interactions between soil and overlying structures (e.g. Burd et al., 1994), nearly all FE analyses rely on approximate methods or empirical estimates of ground loss to simulate conditions occurring around the tunnel cavity (e.g. Schweiger et al., 1997). In practice, most predictions of ground settlements are still based on empirical methods (Peck 1969). Recent research by Sageseta (1987, 1998) and Verruijt and Booker (1996) strongly suggests that reasonable predictions of far field soil movements can be obtained using analytical solutions that simplify the tunnel geometry and assume linear elastic soil properties.

The authors have investigated and extended analytical solutions for shallow tunnels in soil and have used these results to first establish the numerical accuracy in FE analyses and then to evaluate the role of soil behavior in controlling far field deformations. The following paragraphs summarize some results from this on-going study.

A.3 Analytical solutions for shallow tunnels

By assuming elastic properties for the soil, it is possible to derive closed-form expressions for the ground deformations as functions of displacements occurring at the tunnel cavity. Displacement boundary conditions can be expressed as the summation of three basic modes as shown in Figure A-1: (1) uniform ground loss (u_ε); (2) pure distortion (ovalization, u_δ , with no change in volume of the cavity); and (3) vertical translation (Δu_y).

Following Sagaseta (1987, 1998) and Verruijt and Booker (1996), ground deformations can be obtained using fundamental solutions for a line sink in full-space (located at depth, H , corresponding to the axis of the tunnel). Boundary solutions for the stress-free ground surface are obtained by superimposing solutions for an image sink together with a distribution of corrective surface shear tractions. Solutions for the distortion mode are based on full-space solutions for deformations around a cylindrical cavity acted upon by uniform deviatoric stresses (after Kirsch, 1898). Both solutions produce vertical translations of the tunnel cavity. Pinto and Whittle (2001) have evaluated the accuracy of these solutions by deriving solutions from a more complete formulation that includes the physical dimensions of the tunnel cavity with radius, R (using complex variables and conformal mapping after Verruijt, 1997). The solutions for this ‘exact geometry’ are written in Laurent series format with coefficients that must be solved numerically in order to satisfy the far field boundary conditions. Figure A-2 shows that the approximate (point) solutions provide are in excellent agreement with the solutions obtained using the exact geometry for a tunnel with radius to embedment ratio, $R/H = 0.2$ ($\nu = 0.25$) in both the uniform convergence and ovalization modes of cavity deformation. Not surprisingly, the importance of the cavity geometry becomes more significant as R/H increases (i.e. for very shallow tunnels).

Sagaseta (1987) and Pinto (2000) have reported good agreement between analytical and measured settlement troughs from a number of instrumented tunnel projects. Other authors have reported that the shape of the settlement trough can be further improved by considering elastic anisotropy of the soil mass. Chatzigiannelis and Whittle (2003) have extended the approximate superposition method for tunnels excavated in a cross-anisotropic soil using complex variable methods. Figure A-3 shows that the width of the surface

settlement trough reduces significantly for soils with a low stiffness ratio, G_{vh}/E_h , as reported from FE analyses by Lee and Rowe (1989).

A.4 Soil model and numerical analyses

There is an extensive literature documenting the non-linear and inelastic constitutive behavior of soils. For example, Pestana and Whittle (1999) have presented a generalized effective stress model, referred to as MIT-S1, which is capable of predicting the rate independent, effective stress–strain–strength behavior of uncemented soils over a wide range of confining pressures and densities. Figure A-4 illustrates typical MIT-S1 predictions for a series of five standard, drained triaxial compression shear tests (with parameters derived for a reference cohesionless soil, Toyoura sand) performed at the same confining pressure but a range of initial void ratios (very dense, $e_0 = 0.6$, to very loose, $e_0 = 0.95$). The predicted stress–strain behavior is consistent with measured data, with non-linear stiffness properties clearly seen at axial strains $\varepsilon_a < 0.01\%$, and dilation occurring at $\varepsilon_a = 0.5\%$ for dense samples at high mobilized friction angles.

The role of soil modeling in predictions of tunnel-induced ground movements can now be appreciated by comparing numerical simulations using MIT-S1, with analytical results for an isotropic elastic soil. The numerical analyses have been carried out with prescribed deformation boundary conditions around the tunnel cavity. Figure A-5 compares results for a tunnel with $R/H = 0.45$ in a cohesionless, medium-dense Toyoura sand ($e_0 = 0.75$) in the uniform convergence and ovalization modes of deformation. The distribution of surface settlements predicted by MIT-S1 is very similar to the analytical solutions (linear, isotropic soil) for very small values of u_ε/R (0.002, 0.02%, Figure A-5(a)). However, as the ground loss increases to $u_\varepsilon/R = 0.2\%$, there is a very substantial narrowing of the settlement trough associated with an incipient failure mechanism predicted by MIT-S1. In contrast, Figure A-5(b) shows excellent agreement between the non-linear numerical solutions and analytical solutions for ovalization up to $u_\delta/R = 0.2\%$. These results represent a first step towards a more comprehensive understanding linking soil properties to tunnel-induced ground movements.

A.5 Conclusion

Reliable predictions of ground movements caused by shallow tunnel excavation are important for underground construction projects in urban areas. Recently developed analytical solutions provide a useful framework for estimating these ground movements, but are based on simplified assumptions of linear (isotropic or anisotropic) material behavior. By comparing numerical predictions using realistic soil models with the analytical solutions, the results in this paper provide a first step towards understanding the role of non-linear and inelastic soil behavior on the distribution of ground movements.

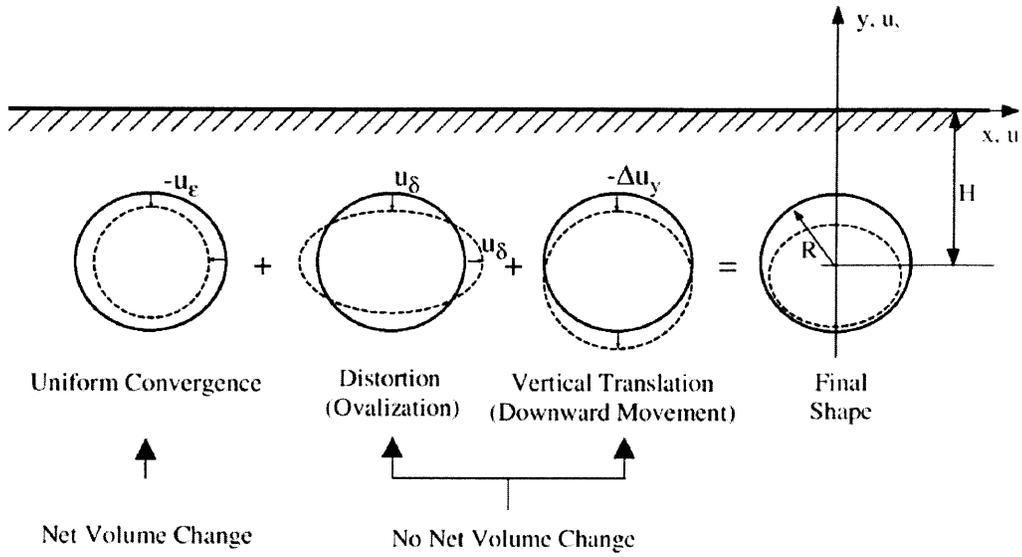


Figure A-1 Deformation Modes for a Shallow Tunnel Cavity

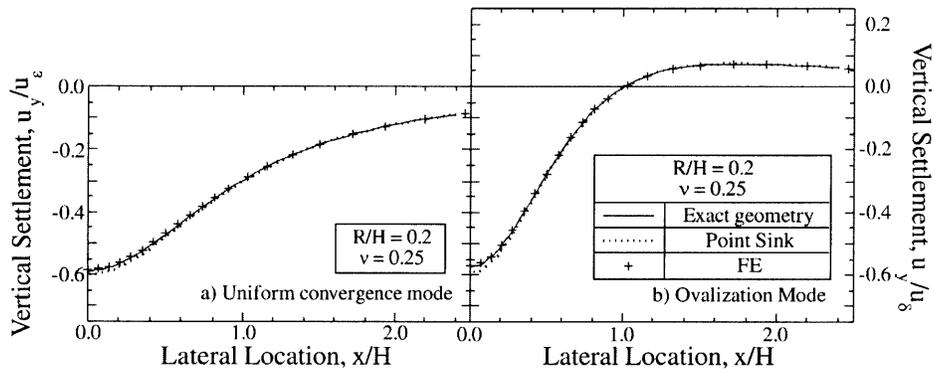


Figure A-2 Comparison of Analytical Solutions for Surface Settlements for a Shallow Tunnel in a Linear, Isotropic soil

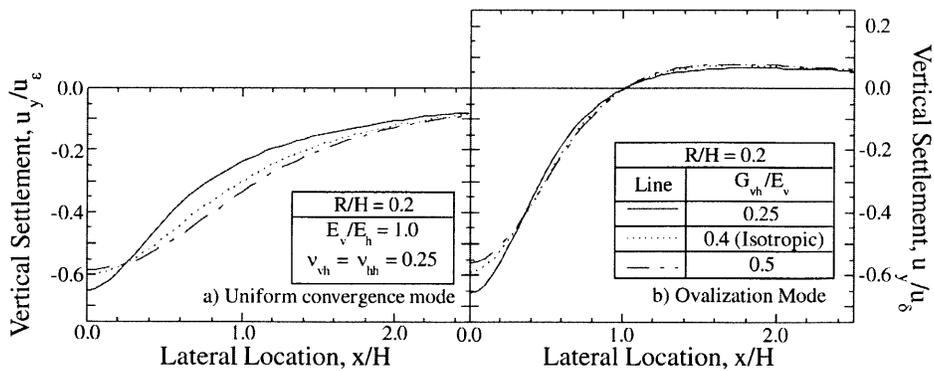


Figure A-3 Analytical Solutions for Surface Settlements in Cross-Anisotropic Elastic Soil

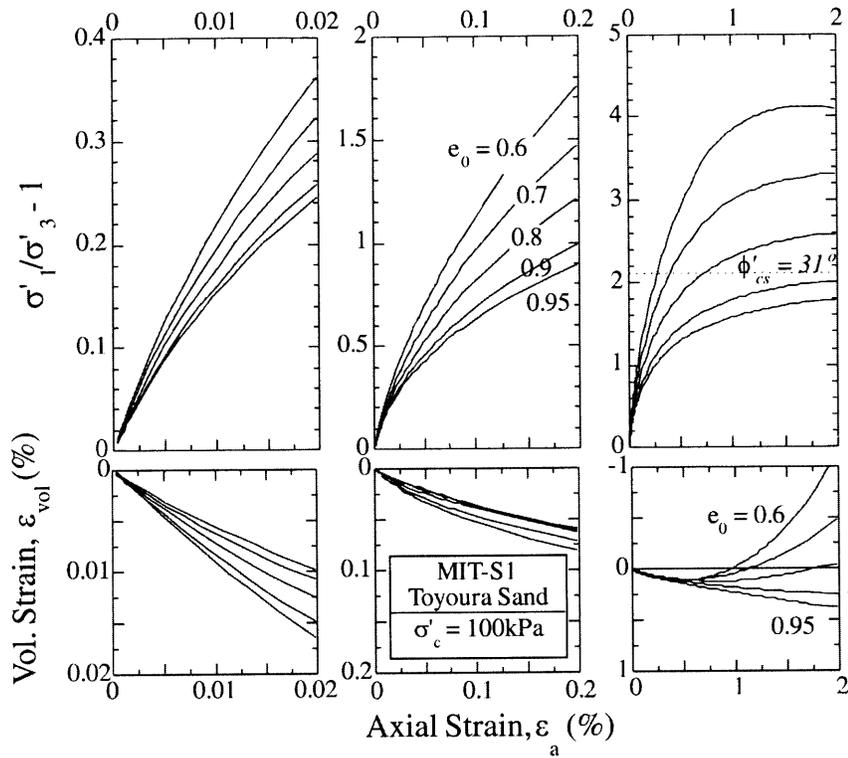


Figure A-4 Typical Predictions of Drained Triaxial Compression Shear Behavior for Sand using MIT-S1 Model

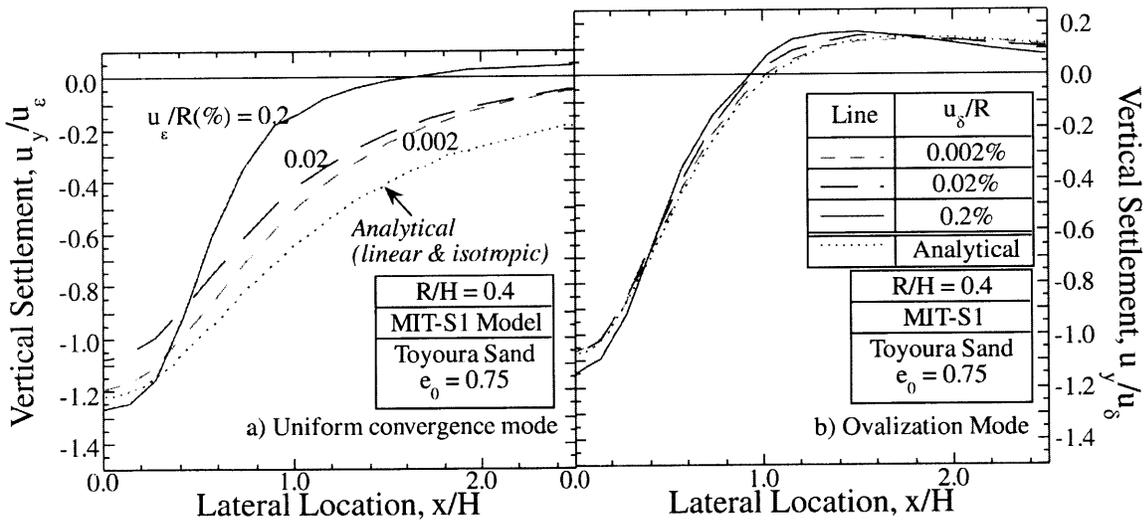


Figure A-5 Role of Soil Model in Predictions of Surface Settlements for Shallow Tunnel in Sand

Appendix B

Implementation of Remeshing Algorithm for Efficient Tunneling Analysis

This appendix documents an implementation of remeshing algorithm for efficient tunneling analyses proposed in Section 4.3.

B.1 A Prototype Implementation

The proposed remeshing procedure is implemented in conjunction with ABAQUS 5.8. The prototype implementation consists of three components: 1) a remeshing program, 2) a post-processing program, and 3) ABAQUS 5.8. The entire analysis is summarized in Figure B-1.

B.2 The Remeshing Program – newMesh

The remeshing program “newMesh” performs two major tasks: 1) reconfigure the finite element mesh based on user specifications, and 2) map solutions between the old mesh and the new mesh.

The mesh reconfiguration is based on a text file prepared by the user using the format shown in Figure B-2. The mode-1 remeshing specified in Figure B-2(a) transforms a uniformly extruded mesh into a user-specified non-uniform mesh, as illustrated on top of Figure B-3. This task is done by overriding nodal coordinates in the extrusion direction with user-specified layer-by-layer coordinates, and no changes are made on nodal connectivity, element stresses, or strains. As a result, the implemented mode-1 remeshing is only valid for level ground conditions.

Mode-2 remeshing, illustrated in Figure B-3 with input file shown in Figure B-2(b), combines two user-specified layers of elements into one layer; and halves a user-specified layer of elements into two layers. The currently implementation considers only second-order brick elements (C3D20²) and the remeshing procedure is illustrated in Figure B-4. The

² ABAQUS™ User Manual

procedures described following are applied to all elements in layers to be divided or combined, and therefore the following descriptions only refer to a single element or adjacent two elements.

Combining two adjacent elements into one element is achieved by eliminating nodes that are not needed in the new configuration, and create the new element by reusing the existing nodes. Variables on the new integration points are interpolated or extrapolated linearly with the closest two integration points along the direction of extrusion in the elements to be combined, as illustrated in Figure B-4(a). The interpolation is computationally efficient, and should be sufficiently accurate if the interpolation is done far from the tunnel face where the distributions of variables should be close to one-dimensional condition along the tunnel axis.

To divide one layer of elements into two layers in mode-2 remeshing, two layers of nodes need to be created. As shown in Figure B-4(b), the new nodes are linearly interpolated from existing nodes along the edges of the original finite element along the extrusion direction, and new integration points are interpolated/extrapolated linearly along the extrusion direction use the existing integration points.

The interpolation scheme proposed here is crude and inconsistent with finite element interpolation, thus should only be employed away from the tunnel face as the linear interpolation along the tunnel axis may introduce error during solution mapping process. More sophisticated interpolation algorithms are needed if the remeshing zones are close to the tunnel face. As the stress/strain distribution along the tunnel axis cannot be interpolated accurately using linear interpolations. Inaccurate interpolation of stress, strain, or state variables will fail the global force equilibrium condition in the system and produce spurious displacements.

B.3 Interfacing with ABAQUS

The remeshing program *newMesh* interfaces with ABAQUS to get information regarding nodes, elements, and solution variables in order to reconfigure the finite element mesh and map solutions between the old and the new meshes

In ABAQUS, nodal and elemental variables can be exported to an external file with filename ended with **.fil* in either the ASCII mode or the binary mode. Use the binary mode is

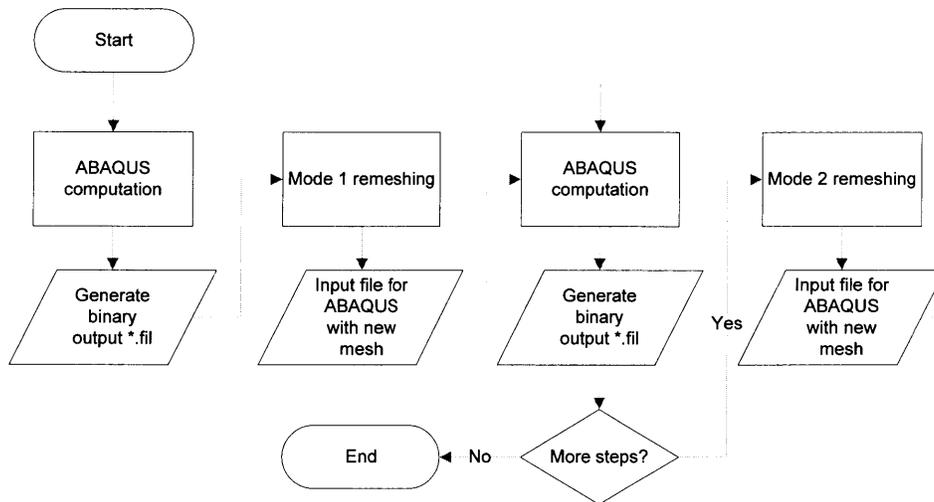
essential to acquire information accurately from ABAQUS. Figure B-5 shows the code to instruct ABAQUS to export nodal displacements, element stresses and coordinates to **.fil* files in binary format, and the file format is documented in ABAQUS/Standard user's manual.

After parsing the binary ABAQUS output file, the *newMesh* program reconfigures the mesh and interpolates variables onto the new mesh. The information regarding the new mesh including nodal coordinates and element connectivity is written to a file called *GEOM-NEW.INC*, and the mapped stresses are written to a separate binary file *NEWSTRESS.DAT*. The input file for ABAQUS in the next step will then need to include the *GEOM-NEW.INC* that defines the new mesh and a user defined subroutine *SIGINI.F* to read the mapped stresses. The prototype only interpolates stresses, as ABAQUS does not provide capabilities to initialize strains and displacements. Therefore, each remeshing step produces incremental results (e.g. displacements), and a post-processing step is required to sum up all increments results to recover the solution.

B.4 The Post Processing Program

Since each new mesh solved in ABAQUS is treated as a standalone problem and stresses are transferred from the previous mesh as an initial stress condition, each computation solves only the incremental displacements. As a result, a post-processing step is needed to obtain total displacement field by summing up the incremental displacement fields from all previous remeshing steps.

In order to calculate displacement field, post-processing needs to take both node number and the remeshing sequence into consideration, as the same node number may correspond to two different nodes at two different meshes. This is achieved by the program *post*.



Mode 1 remeshing: turn uniform meshes into non-uniform ones
 Mode 2 remeshing: move HMD zone with tunnel headings

Figure B-1 Tunneling Analyses with Remeshing

```

Direction=0      ' Tunnel axis aligns with X direction
Sections=61     ' 30 layers of 2nd order element→61 layers of nodes
Mode=1          ' Mode 1: Remesh from initial uniform mesh

Face=0.0       ' reference coordinate for the following
RS= 0.0, 24.0, 1.0 ' start relative position (to face), end position, and thickness
RS=24.0, 72.0, 2.0
RS=72.0,120.0, 4.0
  
```

(a) Input file for mode-1 remeshing

```

Mode=2
C-.....H.....          \ step 5-6
.C-.....H.....          \ step 6-7
..C-.....H.....        \ step 7-8
...C-.....H.....       \ step 8-9
....C-.....H.....      \ step 9-10
.....C-.....H.....     \ step 10-11
.....C-.....H.....     \ step 11-12
.....C-.....H.....     \ step 12-13
C-.....C-.....H...H... \ step 13-14
.....C-.....H.....     \ step 14-15
.C-.....C-.....H...H... \ step 15-16
.....C-.....H.....     \ step 16-17
..C-.....C-.....H...H... \ step 17-18
.....C-.....H.....     \ step 18-19
...C-.....C-.....H...H... \ step 19-20
.....C-.....H.....     \ step 20-21
....C-.....C-.....H...H... \ step 21-22
.....C-.....H.....     \ step 22-23
.....C-.....C-.....H...H... \ step 23-24
.....C-.....H.....     \ step 24-25
.....C-.....H.....     \ step 25-26
.....C-.....H.....     \ step 26-27
.....C-.....H.....     \ step 27-28
.....C-.....H.....     \ step 28-29
  
```

(b) Input file for mode-2 remeshing

Figure B-2 Input files for remeshing program

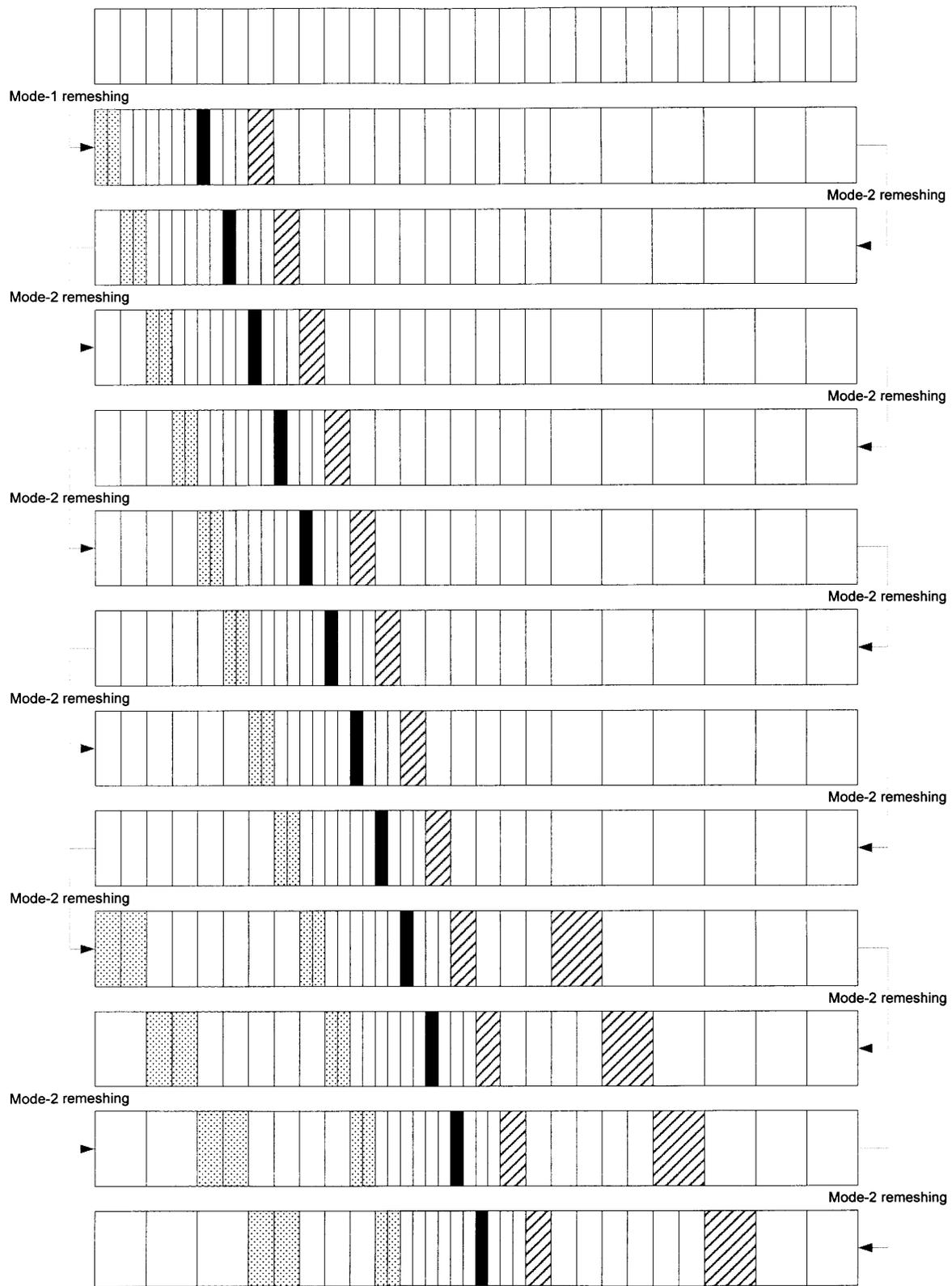
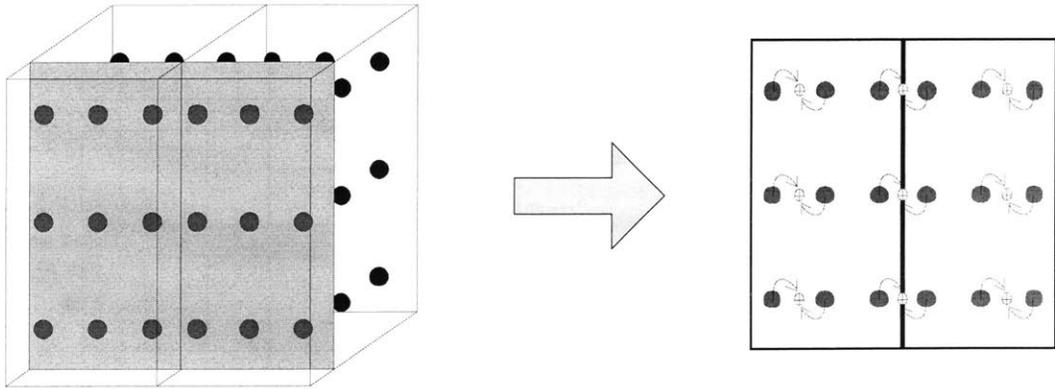
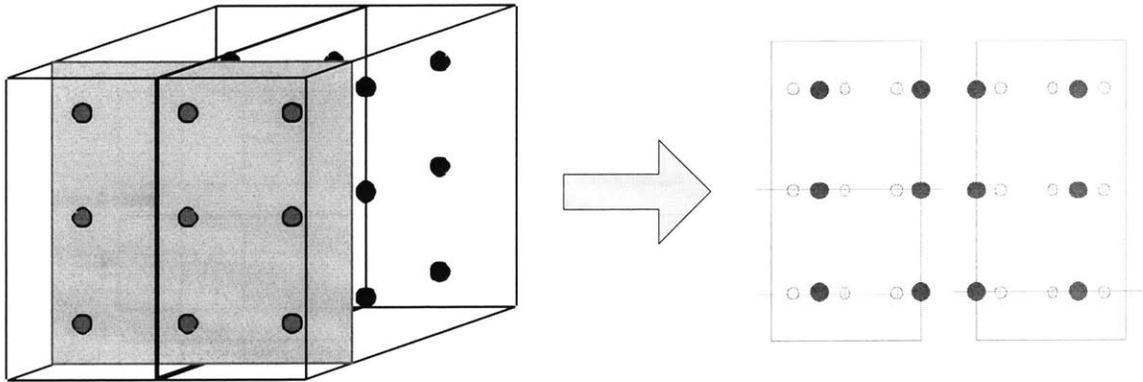


Figure B-3 Two Modes of Remeshing



(a) Combining two layers of elements into one layer



(b) Halving one layer of element into two layers

Figure B-4 Remeshing for C3D20 elements

```

*POST OUTPUT, STEP=5
1
***FILE FORMAT, ASCII
*NSET, NSET=NALL, GEN
1, 53070, 1
*NODE FILE, NSET=NALL
U
*EL FILE, ELSET=GROUND
COORD, S

```

Figure B-5 Code for ABAQUS to Export Information for newMesh

Appendix C

Recommended Practice for Beowulf Clusters

C.1 Observations

Although Beowulf cluster offers significantly lower price/performance ratio than supercomputers, and can be easily set up using COTS (Components Off-The-Shelf). Based on our experience with **codex-hammer**, the Beowulf cluster installed in the department of Civil and Environmental Engineering at MIT, there are several issues that cannot be overlooked:

- 1) **Cooling**: cluster systems need proper cooling, or random crash¹ may occur. This is especially problematic during the development of parallel programs, as it is hard to determine if a crash is caused by a software error or by unstable overheated hardware. Inadequate cooling also prevents long calculations to be performed and renders the computational capability of cluster systems useless.
- 2) **Component compatibility**: using COTS not only reduces the hardware cost but also avoids vendor lock-in. However, components can sometimes be incompatible with each other even though the specification says otherwise. Incompatible components can result in random lockups or data corruption. **Codex-hammer** suffered from the compatibility problem between the motherboard and the memory modules installed, and resulted in random data corruption in memory chips. As a result, different results were generated from identical calculations, and programs crashes unexpectedly. The problem was resolved by replacing all motherboards in the cluster.
- 3) **Component reliability**: recently some nodes of codex-hammer experiences random lock-up under heavy load even with proper cooling. Although the real cause has yet

¹ The console returns message “segmentation fault (11)”, which is a very general error messages that can be triggered by the use of unallocated memory.

to be determined, the stability was improved by increasing the voltage¹ (VCore) supplied to the processor. It is suspected the voltage regulator on the motherboard may be failing as we are reading fluctuating voltage readings from the sensors on the motherboard.

C.2 Recommendations

- **Testing and Validation:** it is important to test and validate the hardware components constituting the Beowulf cluster. Through thorough validation process, the faulty or unreliable components can be exchanged or replaced under warranty of the components and ensures the reliability of the cluster system.
 - Memory: faulty or incompatible memory modules exhibit random data corruption (the data stored is different from the data retrieved). The problem can be identified by programs through writing different patterns into memory and then retrieve the data from memory to check if they are consistent. However, it should be noted due to the caching mechanism of modern computer systems, the data retrieval needs to be done after the data is flushed from the cache memory. Memtest86² is an open source solution for memory testing.
 - Processor: inadequate cooling is usually the cause of unreliable operations of processors. It has to be tested by putting the processor under heavy load (e.g. running computationally intensive tasks) for a long period, and the calculation results need to be checked with a known solution.
- **Monitoring:** modern computer motherboards are equipped with several sensors to monitor the health of the system. The motherboard in **codex-hammer** can monitor processor temperatures, system temperatures, the cooling fan speed, voltage supplied

¹ This is a tweak usually performed by “overclockers” to improve the stability of overclocked (running processors at higher clock speed than their specification) systems.

² A stand-alone memory diagnostics tool available at <http://www.memtest86.com/>

to various components, etc. By monitoring these readings, it can be determine if the system is running within operational parameters, and measures can be taken to prevent the system to become unstable.

Appendix D

Sparse Matrix Storage

The finite element system usually gives rise to sparse equation systems to be solved. There are several standard storage schemes for storing sparse matrices: compressed-row/column, coordinate format, etc. that only store nonzero entries in sparse matrices. Using sparse storage scheme is essential for large-scale finite element analyses, as it is impossible to store large stiffness matrices using dense matrix (array) storage due to excessive memory requirement. Figure D-1 demonstrates the memory requirement for given matrix sizes using the dense matrix storage. It is seen that 1GB of memory can store only a little more than 10000×10000 dense matrix assuming double precision values (64-bit,) and large finite element models can easily give rise to problem sizes one or two orders of magnitude larger. Therefore, it is impractical to use dense matrix storage.

The standard sparse storage schemes such as compressed-column format, as known as Harwell-Boeing sparse matrix format (Duff et al. 1989), are hard to manipulate efficiently. Adding a value to a compressed-row formatted sparse matrix requires a search to be done on pre-existing entries of the matrix. If the entry is not originally present in the matrix, part of the matrix needs to be moved by one element to make room for the new entry. This is not very an efficient scheme for cases where the matrix is being updated frequently.

On the other hand, once the sparse matrix has finished updating, the existing sparse storage schemes are both memory and computationally efficient. They are also required when using existing sparse numerical packages such as SuperLU (Li 1996). Therefore, a sparse matrix class, *sMatrix*, is written in C++ to manipulate and store sparse matrices. The class *sMatrix* manages memory allocation internally, and features methods to add and update entries in sparse matrices, to convert the sparse matrix into standard compressed row/column format, and to perform sparse matrix calculations such as matrix-vector products.

An internal flag *_compact* is used to indicate whether the sparse matrix is compacted. If the sparse matrix is compacted, compressed column storage scheme is used internally to store the sparse matrix, and no updates to the sparse matrix are allowed. Otherwise, the internal

storage scheme is in coordinate format, duplicates are allowed, and memory is dynamically allocated to accommodate the needs to store matrix entries. Once the pre-allocated memory is exhausted for storing data, a new block of memory is allocated with twice the size of previous allocation, data are transferred to the newly allocated block of memory, and then the old allocation is released.

The method `compact()` is used to compact the sparse matrix and transform internal storage scheme from coordinate format to compressed column format. The transformation is done via a quick sort algorithm on both row and column indices, and duplicates entries are merged. Once the transformation is complete, the internal flag `_compact` is set, no more updates to the sparse matrix are allowed, and various calculation methods associated with the sparse matrix are enabled.

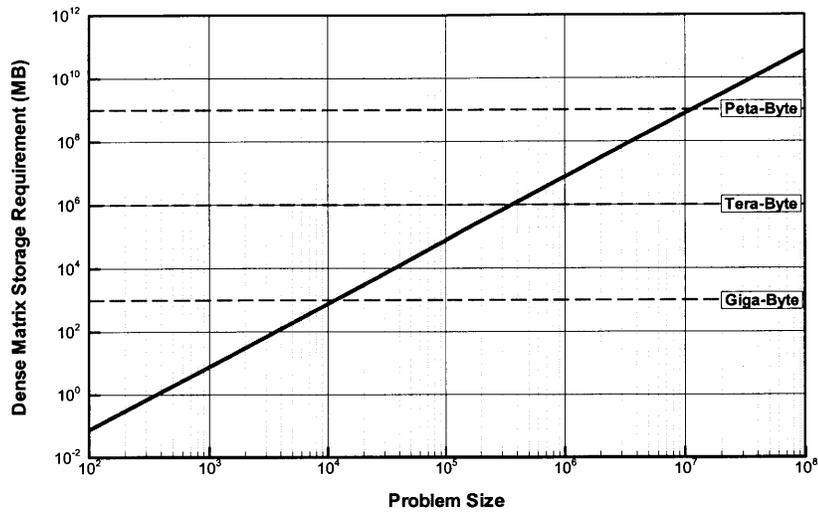


Figure D-1 Memory Requirement for Dense Matrix Storage

Appendix E

Isoparametric Element Formulation

In order to obtain a discrete system with limited number of unknowns to be solved, finite element method uses the nodal values of physical quantity of interest to infer the quantity within each finite element through interpolation functions. Thus, the interpolation function defines the variations within finite elements.

It is convenient and effective to construct finite element interpolation functions using local coordinate system (r, s, t) ranging from -1 to 1 on each axis (also know as the natural coordinate system) instead of using the global coordinate system (x, y, z) , and place the nodes constituting the finite element at special locations in order to evaluate interpolation effectively.

If a finite element is defined by n nodes with its nodal values $\hat{\mathbf{U}}$ (a vector with n components) and interpolation function $\mathbf{H}(r, s, t)$ (a vector of n components) defined. Then the values with identical physical meaning as $\hat{\mathbf{U}}$ within the finite element is completely defined by:

$$\mathbf{U}(r, s, t) = \mathbf{H}(r, s, t)\hat{\mathbf{U}} = \sum_{i=1}^n H_i \hat{U}_i \quad (\text{E.1})$$

Each component H_i of vector $\mathbf{H}(r, s, t)$ is a scalar weighting function, and their values are determined entirely by the location defined in the local coordinate system (r, s, t) . The corresponding global coordinates can be obtained by replacing nodal variables $\hat{\mathbf{U}}$ in Equation (E.1) with nodal global coordinates $\hat{\mathbf{X}}$ defined in global coordinate system:

$$\mathbf{X}(r, s, t) = \mathbf{H}(r, s, t)\hat{\mathbf{X}} \quad (\text{E.2})$$

Derivatives of physical quantities are often encountered in mathematical modeling. From (E.1), the derivatives are readily available by differentiating interpolation functions \mathbf{H} with the help of chain rules. Thus,

$$\begin{bmatrix} \frac{\partial \mathbf{U}}{\partial x} \\ \frac{\partial \mathbf{U}}{\partial y} \\ \frac{\partial \mathbf{U}}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial s}{\partial x} & \frac{\partial t}{\partial x} \\ \frac{\partial r}{\partial y} & \frac{\partial s}{\partial y} & \frac{\partial t}{\partial y} \\ \frac{\partial r}{\partial z} & \frac{\partial s}{\partial z} & \frac{\partial t}{\partial z} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{U}}{\partial r} \\ \frac{\partial \mathbf{U}}{\partial s} \\ \frac{\partial \mathbf{U}}{\partial t} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial \mathbf{U}}{\partial r} \\ \frac{\partial \mathbf{U}}{\partial s} \\ \frac{\partial \mathbf{U}}{\partial t} \end{bmatrix} \quad (\text{E.3})$$

Table E-1 and Table E-2 list the interpolation functions for two-dimensional and three-dimensional interpolation functions used for CPE4, CPS4 (Figure E-1, Figure E-2), CPE8, CPS8 (Figure E-3, Figure E-4), and C3D20 (Figure E-5) elements implemented in FETI_FEM and in ABAQUS. Figure E-1 to Figure E-5 show the node numbering and plots of interpolation functions to check the interpolation functions. These interpolation functions should have a value of 1 when passing the corresponding nodes, and 0 on other nodes.

Table E-1 2-D Weighting Functions and Derivatives

	Node k	Weighting h_k	Derivative-r $\frac{\partial h_k}{\partial r}$	Derivative-s $\frac{\partial h_k}{\partial s}$
4-noded	1	$\frac{1}{4}AC$	$-\frac{1}{4}C$	$-\frac{1}{4}A$
	2	$\frac{1}{4}BC$	$+\frac{1}{4}C$	$-\frac{1}{4}B$
	3	$\frac{1}{4}BD$	$+\frac{1}{4}D$	$+\frac{1}{4}B$
	4	$\frac{1}{4}AD$	$-\frac{1}{4}D$	$+\frac{1}{4}A$
8-noded	1	$-\frac{1}{4}AC(B+s)$	$\frac{1}{4}C(E+s)$	$\frac{1}{4}A(F+r)$
	2	$-\frac{1}{4}BC(A+s)$	$\frac{1}{4}C(E-s)$	$\frac{1}{4}B(F-r)$
	3	$-\frac{1}{4}BD(A-s)$	$\frac{1}{4}D(E+s)$	$\frac{1}{4}B(F+r)$
	4	$-\frac{1}{4}AD(B-s)$	$\frac{1}{4}D(E-s)$	$\frac{1}{4}A(F-r)$
	5	$\frac{1}{2}ABC$	$-\frac{1}{2}CE$	$-\frac{1}{2}AB$
	6	$\frac{1}{2}BCD$	$+\frac{1}{2}CD$	$-\frac{1}{2}BF$
	7	$\frac{1}{2}ABD$	$-\frac{1}{2}DE$	$+\frac{1}{2}AB$
	8	$\frac{1}{2}ACD$	$-\frac{1}{2}CD$	$-\frac{1}{2}AF$

*Coordinates: (r, s) , $r = [-1 \dots 1]$, $s = [-1 \dots 1]$

$$** \begin{cases} A = 1 - r \\ B = 1 + r \end{cases} \begin{cases} C = 1 - s \\ D = 1 + s \end{cases} \begin{cases} E = B - A = 2r \\ F = D - C = 2s \end{cases}$$

Table E-2 3-D Weighting Functions and Derivatives

Node k	Weighting h_k	Derivative-r $\frac{\partial h_k}{\partial r}$	Derivative-s $\frac{\partial h_k}{\partial s}$	Derivative-t $\frac{\partial h_k}{\partial t}$
1	$-[ACE(+r + D + F)]/8$	$CE(+r - A + D + F)/8$	$AE(+r - C + D + F)/8$	$AC(+r + D - E + F)/8$
2	$-[BCE(-r + D + F)]/8$	$CE(+r + B - D - F)/8$	$BE(-r - C + D + F)/8$	$BC(-r + D - E + F)/8$
3	$-[BDE(-r + C + F)]/8$	$DE(+r + B - C - F)/8$	$BE(+r - C + D - F)/8$	$BD(-r + C - E + F)/8$
4	$-[ADE(+r + C + F)]/8$	$DE(+r - A + C + F)/8$	$AE(-r - C + D - F)/8$	$AD(+r + C - E + F)/8$
5	$-[ACF(+r + D + E)]/8$	$CF(+r - A + D + E)/8$	$AF(+r - C + D + E)/8$	$AC(-r - D - E + F)/8$
6	$-[BCF(-r + D + E)]/8$	$CF(+r + B - D - E)/8$	$BF(-r - C + D + E)/8$	$BC(+r - D - E + F)/8$
7	$-[BDF(-r + C + E)]/8$	$DF(+r + B - C - E)/8$	$BF(+r - C + D - E)/8$	$BD(+r - C - E + F)/8$
8	$-[ADF(+r + C + E)]/8$	$DF(+r - A + C + E)/8$	$AF(-r - C + D - E)/8$	$AD(-r - C - E + F)/8$
9	$ABCE / 4$	$CE(A - B) / 4$	$-ABE / 4$	$-ABC / 4$
10	$BCDE / 4$	$+CDE / 4$	$BE(C - D) / 4$	$-BCD / 4$
11	$ABDE / 4$	$DE(A - B) / 4$	$+ABE / 4$	$-ABD / 4$
12	$ACDE / 4$	$-CDE / 4$	$AE(C - D) / 4$	$-ACD / 4$
13	$ABCF / 4$	$CF(A - B) / 4$	$-ABF / 4$	$+ABC / 4$
14	$BCDF / 4$	$+CDF / 4$	$BF(C - D) / 4$	$+BCD / 4$
15	$ABDF / 4$	$DF(A - B) / 4$	$+ABF / 4$	$+ABD / 4$
16	$ACDF / 4$	$-CDF / 4$	$AF(C - D) / 4$	$+ACD / 4$
17	$ACEF / 4$	$-CEF / 4$	$-AEF / 4$	$AC(E - F) / 4$
18	$BCEF / 4$	$+CEF / 4$	$-BEF / 4$	$BC(E - F) / 4$
19	$BDEF / 4$	$+DEF / 4$	$+BEF / 4$	$BD(E - F) / 4$
20	$ADEF / 4$	$-DEF / 4$	$+AEF / 4$	$AD(E - F) / 4$

20-noded

*Coordinates: (r, s, t) , $r = [-1 \dots 1]$, $s = [-1 \dots 1]$, $t = [-1 \dots 1]$

$$** \begin{cases} A = 1 - r \\ B = 1 + r \end{cases}, \begin{cases} C = 1 - s \\ D = 1 + s \end{cases}, \begin{cases} E = 1 - t \\ F = 1 + t \end{cases}$$

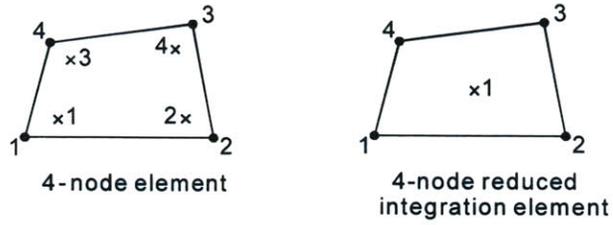


Figure E-1 2-D 4-noded Finite Elements and Integration Points
 (From ABAQUS standard User's Manual, Chap. 13)

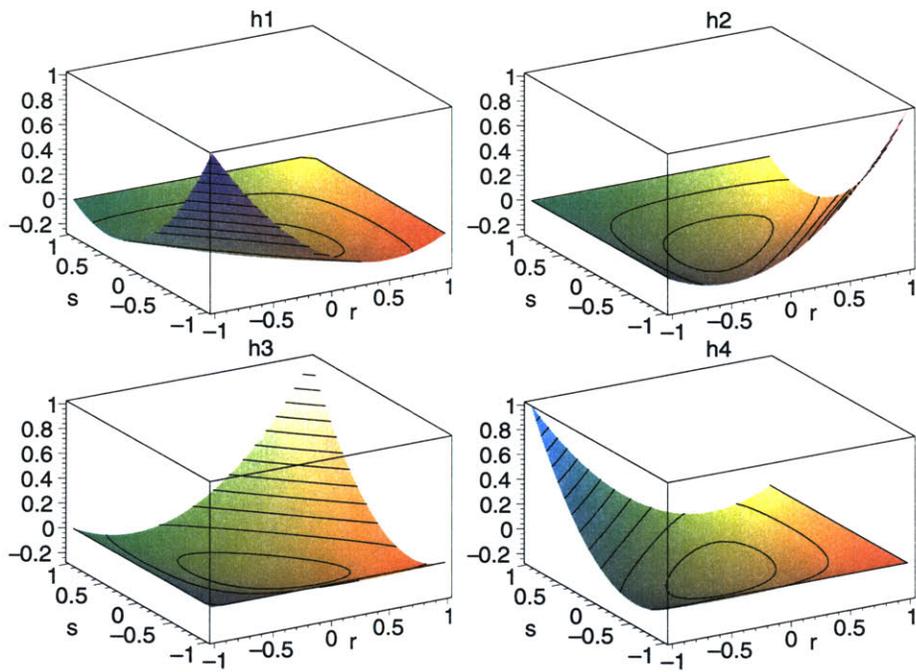


Figure E-2 Weighting Functions for 4-noded elements

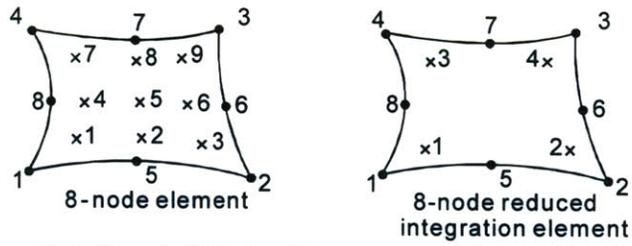


Figure E-3 8-noded Finite Elements and Integration Points
(From ABAQUS standard User's Manual, Chap. 13)

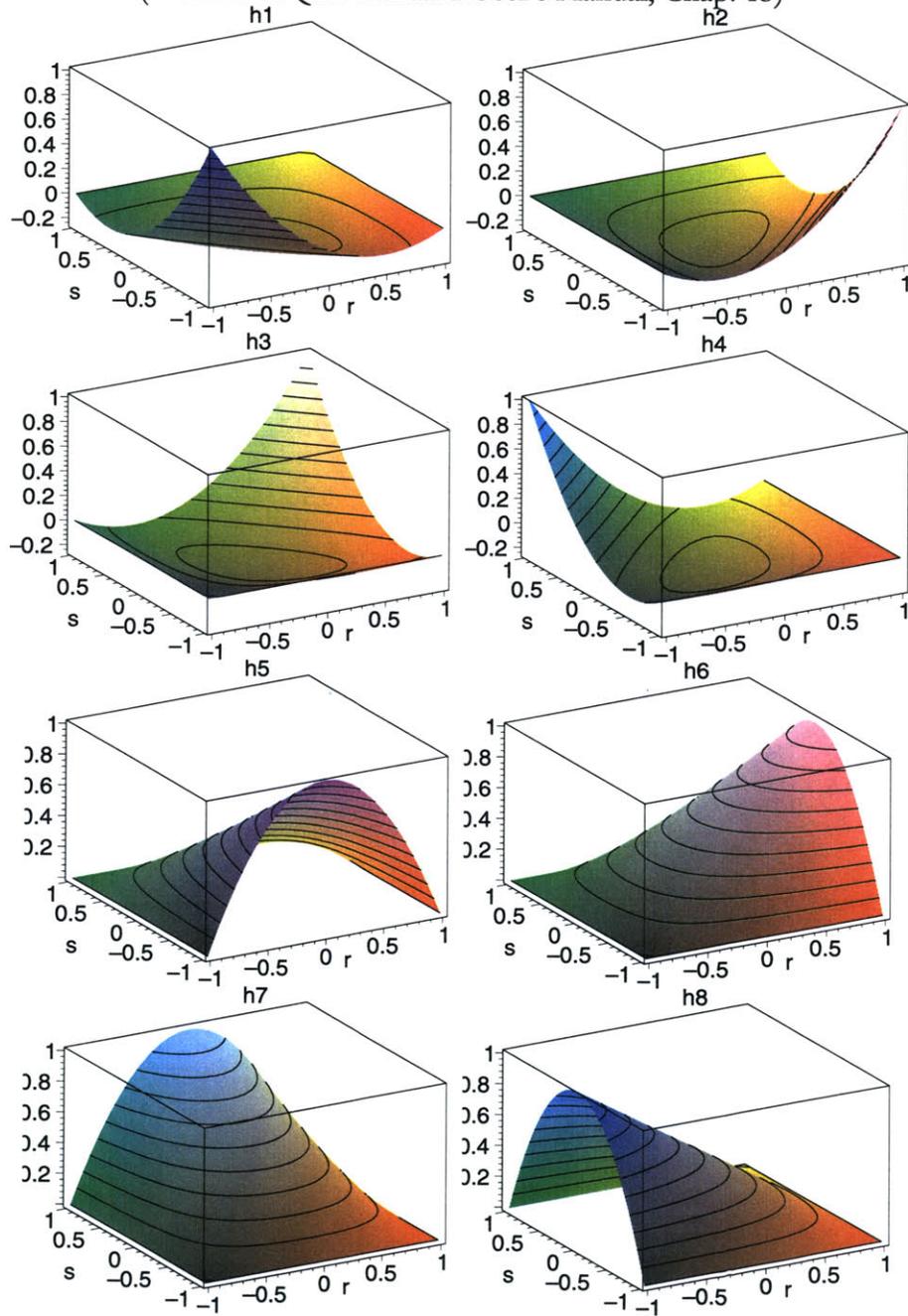


Figure E-4 Weighting Functions for 8-noded elements

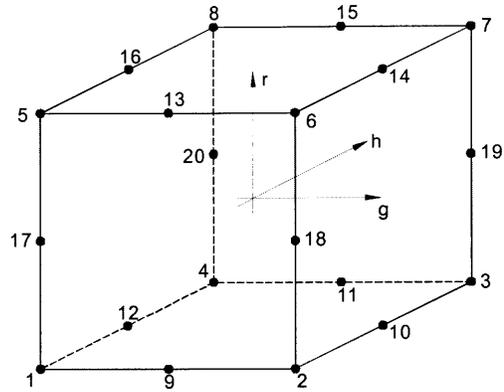


Figure E-5 Nodal Numbering for 20-noded Brick Element
 (From ABAQUS User's Manual)

Appendix F

Selective-Reduced Element Integration

Selective-reduced integration is used in ABAQUS for low-order elements to avoid overly stiff response due to numerical integration. The selective reduced integration for 4-noded elements uses full integration (4 integration points) for integrating deviatoric strains, and reduced integration (1 integration point) for integration of volumetric strains. This is achieved by modifying the standard strain-displacement relationship and the derivatives of interpolation functions:

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{xy} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \mathbf{B}_k & \cdots & \mathbf{B}_n \end{bmatrix} \begin{bmatrix} \hat{U}_1 \\ \hat{U}_2 \\ \vdots \\ \hat{U}_k \\ \vdots \\ \hat{U}_n \end{bmatrix}, \mathbf{B}_k = \begin{bmatrix} \frac{\partial h_k}{\partial x} & 0 & 0 \\ 0 & \frac{\partial h_k}{\partial y} & 0 \\ 0 & 0 & \frac{\partial h_k}{\partial z} \\ \frac{\partial h_k}{\partial y} & \frac{\partial h_k}{\partial x} & 0 \\ 0 & \frac{\partial h_k}{\partial z} & \frac{\partial h_k}{\partial y} \\ \frac{\partial h_k}{\partial z} & 0 & \frac{\partial h_k}{\partial x} \end{bmatrix} \quad (\text{F.1})$$

ε_{ij} is standard strain components, h_i is interpolation function for the i -th node, and \hat{U}_i is nodal displacements, and \mathbf{B} is known as strain-displacement matrix. Because locations of integration points are known a-prior when isoparametric formulation is used, \mathbf{B} can be calculated prior finite element calculations.

For two-dimensional 4-noded finite elements (CPE4, CPS4), a modified strain-displacement matrix $\bar{\mathbf{B}}$ that is different from the one shown in (F.1) is adopted in ABAQUS and in *FETI_FEM*. It is because for low-order elements, full integration scheme have “mesh locking” problems with nearly incompressible materials, and “shear locking” problems showing overly stiff bending response. Reduced integration scheme does not suffer from locking problems. However, reduced integration (1-point integration for 4-noded elements) admits deformations without incurring any strain at integration points as shown in Figure F-1. This zero-strain deformation mode is known as hour-glass mode.

To improve solution for nearly incompressible material behavior for full integration scheme, which ideally gives better integral evaluation, a special integration scheme known as selectively reduced integration is employed for 4-noded elements. Selectively reduce integration integrates the deviatoric strains with full integration (thus four integration points for 2D 4-noded elements), but replaces the volumetric strain with average volumetric strain, which is equivalent to the average of volumetric strain over Gauss points. To achieve selectively reduced integration, the strain-displacement matrix needs to be modified by taking the following steps:

1. For each integration point, calculate deviatoric and volumetric strain-displacement matrices \mathbf{B}^d and \mathbf{B}^{vol} on all integration points:

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{xy} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \end{bmatrix} = \mathbf{B}\hat{\mathbf{U}} = [\mathbf{B}^{vol} + \mathbf{B}^d] \hat{\mathbf{U}} = \begin{bmatrix} \mathbf{B}_1^{vol} \\ \mathbf{B}_2^{vol} \\ \vdots \\ \mathbf{B}_k^{vol} \\ \vdots \\ \mathbf{B}_n^{vol} \end{bmatrix}^T + \begin{bmatrix} \mathbf{B}_1^d \\ \mathbf{B}_2^d \\ \vdots \\ \mathbf{B}_k^d \\ \vdots \\ \mathbf{B}_n^d \end{bmatrix}^T \begin{bmatrix} \hat{\mathbf{U}}_1 \\ \hat{\mathbf{U}}_2 \\ \vdots \\ \hat{\mathbf{U}}_k \\ \vdots \\ \hat{\mathbf{U}}_n \end{bmatrix}$$

$$\mathbf{B}_k^d = \begin{bmatrix} \frac{2}{3} \frac{\partial h_k}{\partial x} & \frac{1}{3} \frac{\partial h_k}{\partial y} & -\frac{1}{3} \frac{\partial h_k}{\partial z} \\ -\frac{1}{3} \frac{\partial h_k}{\partial x} & \frac{2}{3} \frac{\partial h_k}{\partial y} & -\frac{1}{3} \frac{\partial h_k}{\partial z} \\ -\frac{1}{3} \frac{\partial h_k}{\partial x} & -\frac{1}{3} \frac{\partial h_k}{\partial y} & \frac{2}{3} \frac{\partial h_k}{\partial z} \\ \frac{\partial h_k}{\partial y} & \frac{\partial h_k}{\partial x} & 0 \\ 0 & \frac{\partial h_k}{\partial z} & \frac{\partial h_k}{\partial y} \\ \frac{\partial h_k}{\partial z} & 0 & \frac{\partial h_k}{\partial x} \end{bmatrix}, \mathbf{B}_n^{vol} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \frac{1}{3} \frac{\partial h_k}{\partial x} & \frac{1}{3} \frac{\partial h_k}{\partial y} & \frac{1}{3} \frac{\partial h_k}{\partial z} \end{bmatrix} \quad (\text{F.2})$$

2. Calculate the averaged volumetric strain-displacement conversion matrix $\bar{\mathbf{B}}^{vol}$ by averaging \mathbf{B}_k^{vol} on all integration points.
3. Calculate the modified strain-displacement matrix $\bar{\mathbf{B}}$ by using the updated volumetric strain-displacement matrix $\bar{\mathbf{B}}^{vol}$:

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{xy} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \end{bmatrix} = \bar{\mathbf{B}}\hat{\mathbf{U}} = [\bar{\mathbf{B}}^{vol} + \mathbf{B}^d]\hat{\mathbf{U}} \quad (\text{F.3})$$

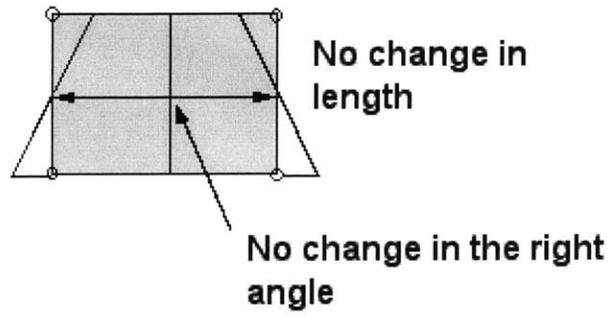


Figure F-1 Hour Glass Mode for 4-noded reduced integration

Appendix G

Post Processing Program for FETI_FEM

After the calculation of *FETI_FEM*, the solutions for sub-domains are scattered across all computation nodes, and the master node only holds the convergence history. Consequently, it is impossible to visualize the overall computation results without gathering all information onto a single computer.

In order to visualize the computational result from *FETI_FEM*, a post-processing program *FEM_POST* is created to process data from *FETI_FEM* following the procedure outlined in Figure G-1. The program then outputs files prepared for use in Tecplot¹. Due to the distributed nature of *FETI_FEM*, the procedure is divided into sub-domain-level operations, computation-node level operations, and master node level operation as shown in Figure G-1. Great similarities are seen between different levels because sub-domains can be regarded as super-elements, thus merging results between sub-domains is no different from merging results between elements.

Merging between sub-domains and between computation nodes is optional. It helps to understand how the workload is distributed between the computation nodes (by merging results of sub-domains computed only by the same computation nodes). Each sub-domain belonging to a specific computation node will be contained in a single “zone” in Tecplot. Similarly, another option enables results from different sub-domains to be merged regardless of which computation node was responsible for them. By not merging results from different sub-domains, each sub-domain has its own zone in Tecplot. Figure G-2 shows results that are merged on each computation node, but not merged across processors. The distribution and geometry of sub-domains can be visualized by not merging sub-domains on a computation node. More importantly, the inaccuracy of interface solvers such as FETI can be visualized by looking for discontinuities across the interfaces, which may not be apparent by merging results across sub-domains.

¹ Visualization software by Amtec Engineering, Inc.

Tecplot can only visualize 8-noded brick elements in 3-D. There are two possibilities to visualize second-order brick elements, which are used in *FETI_FEM*. The first approach is to eliminate all mid-side nodes and only use the 8-corner nodes for each single 20-noded element to form the brick-element supported by Tecplot. This approach is easy to implement, and the data file for Tecplot is small. The resulting visualization model is faster to render and manipulate. However, some information or details are lost due to the filtering.

The second approach is to use finite element interpolation creating mid-face nodes and the mid-volume node, and these additional nodes are then used to transform each single 20-noded element into eight 8-noded elements as illustrated in Figure G-3. This approach provides a more detailed presentation of the finite element calculation results by retaining computation results on all nodes, but at the cost of much larger data file, and requiring more time for rendering and manipulation.

Based on the pros and cons of the two different approaches for visualization finite elements, the coarse presentation (the first approach) is used to search for interesting features of the results, while the fine presentation is used for final rendition and output.

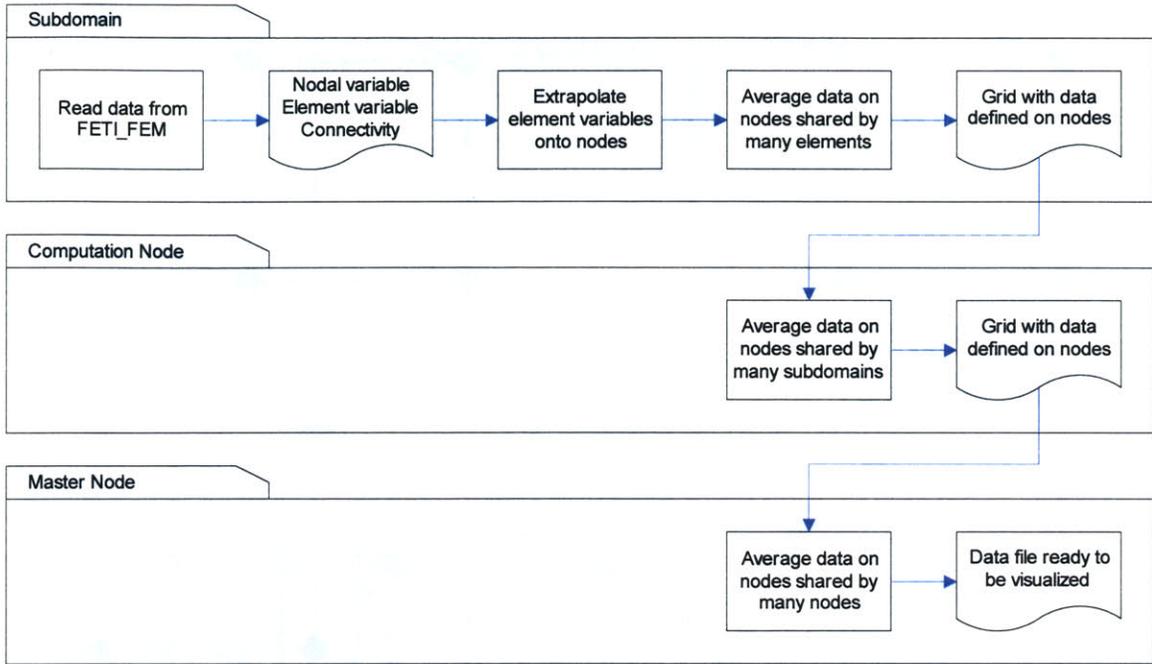


Figure G-1 Data Process for Visualizing Computation Results from FETI_FEM

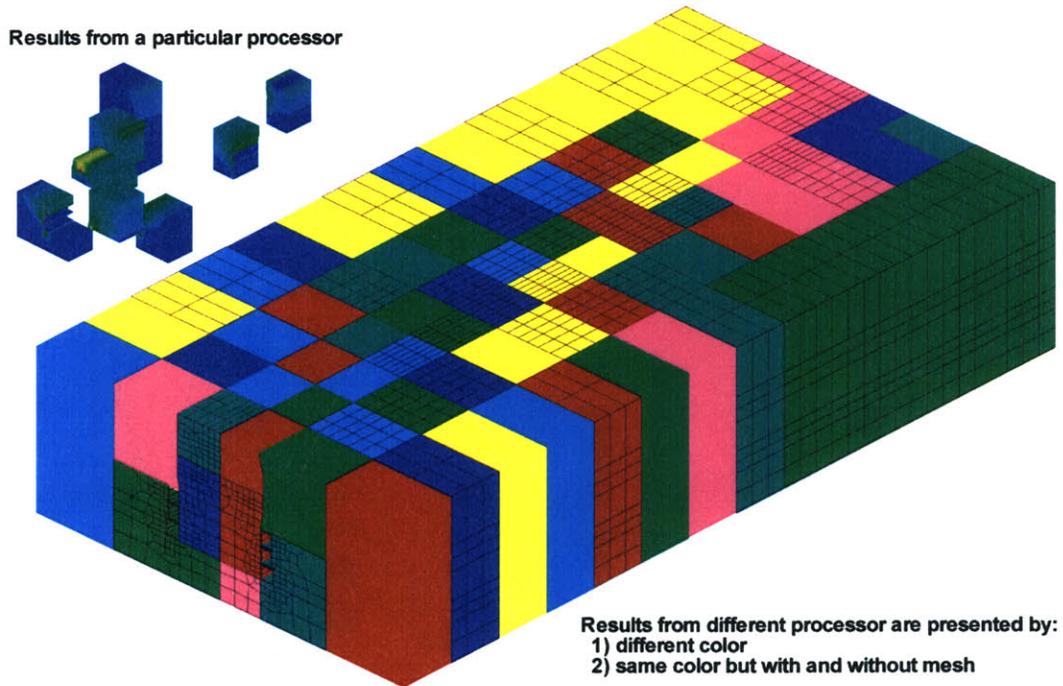


Figure G-2 Visualization of Load Distribution between Computation Nodes

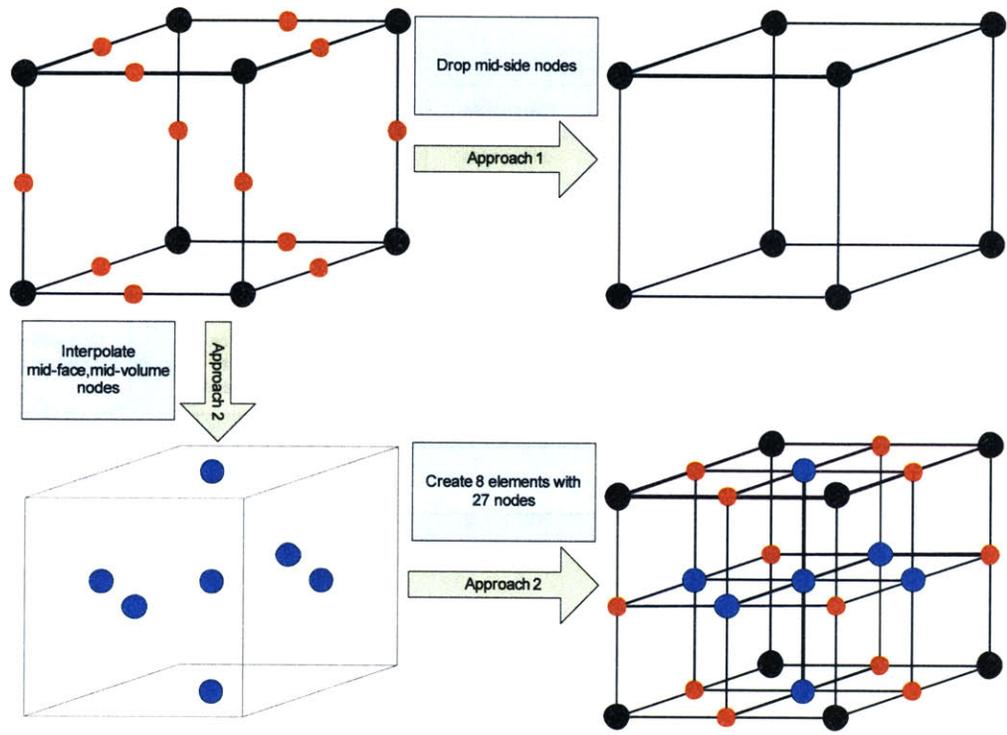
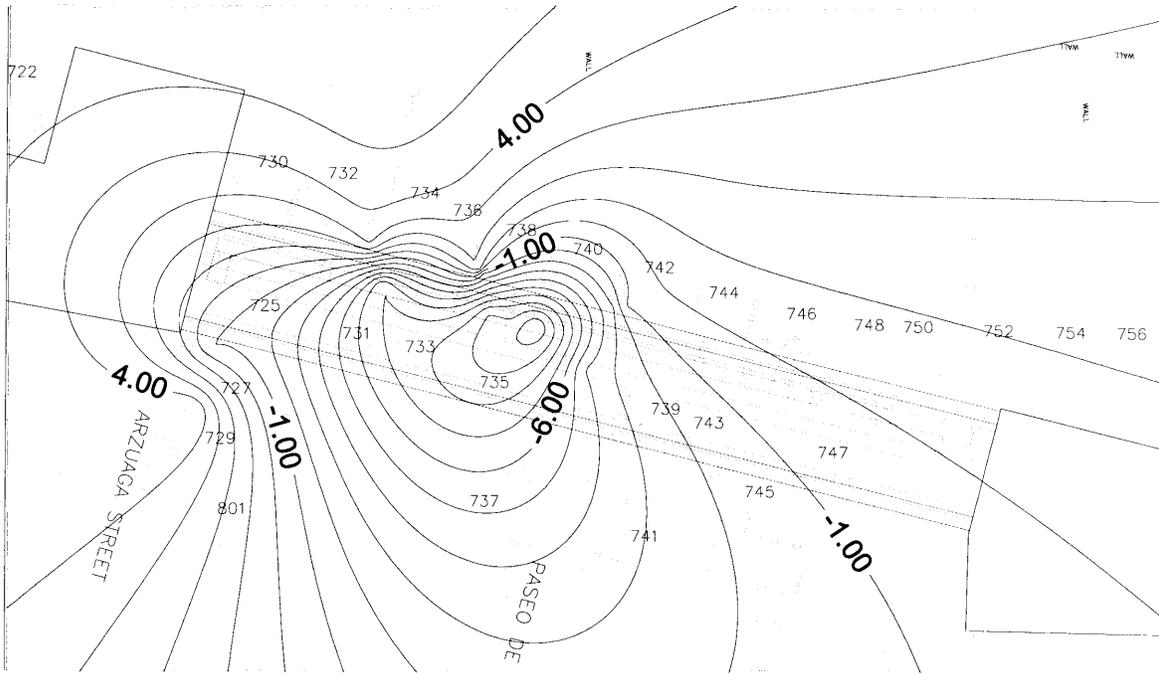


Figure G-3 Process a 20-noded Finite Element for Visualization in Tecplot

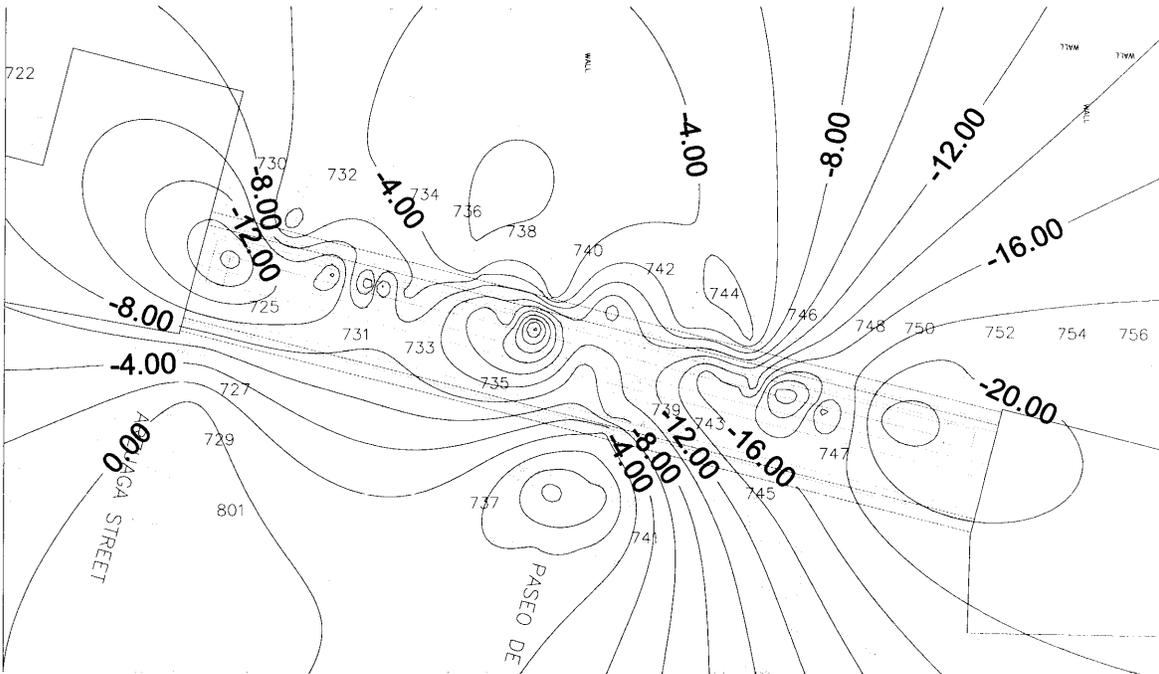
Appendix H

Surface Settlements of Río Piedras Station Construction

This appendix compiles the contours of surface settlements measured from building benchmarks (BMA) at some selected time during the construction.

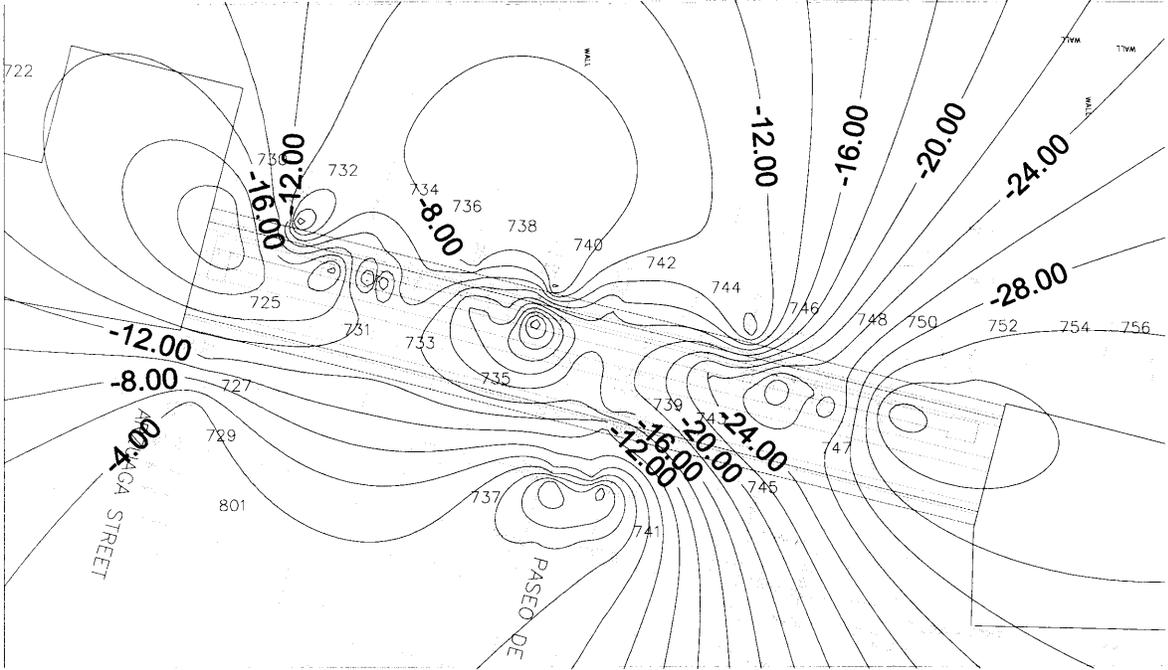


(a) Surface Settlement Contour from Construction Record for CD0

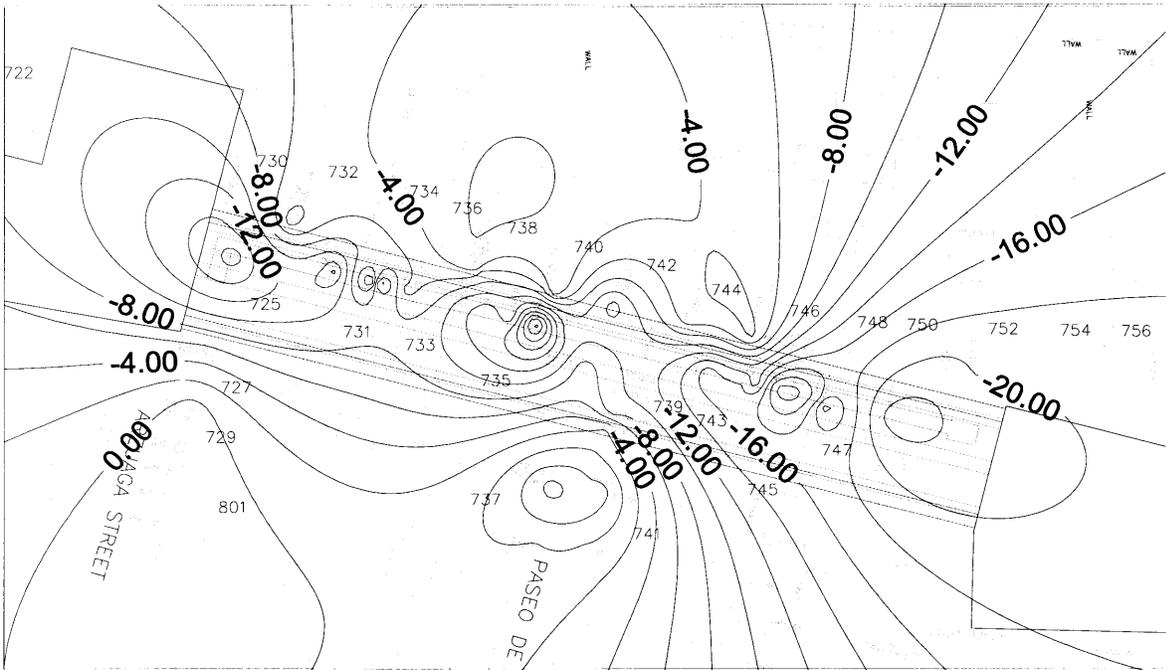


(b) Surface Settlement Contour from Construction Record for CD30

Figure H-1 Surface Settlement Contours from Construction Record

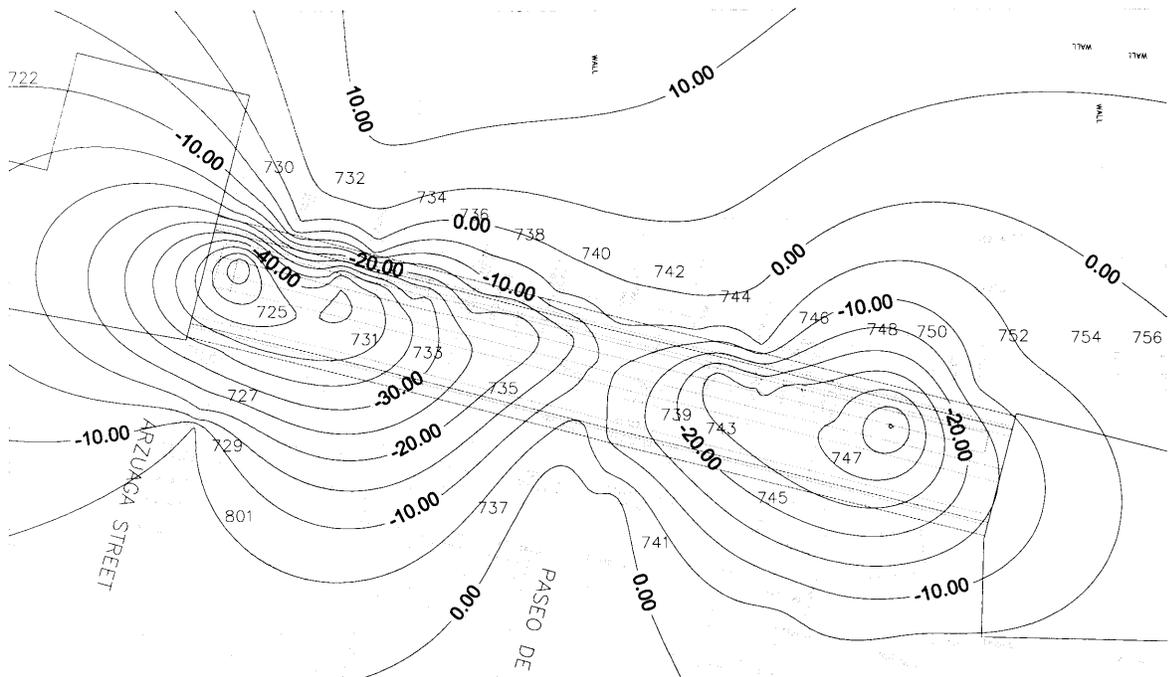


(c) Surface Settlement Contour from Construction Record for CD60

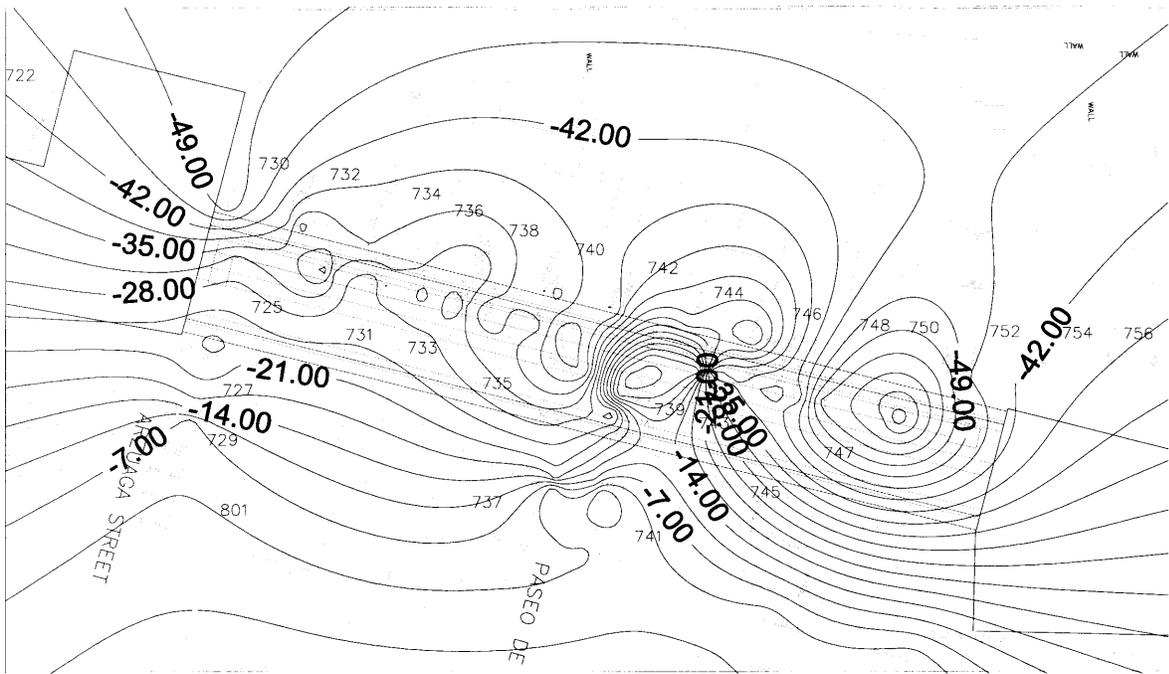


(d) Surface Settlement Contour from Construction Record for CD90

Figure H-1 (Cont'd) Surface Settlement Contours from Construction Record

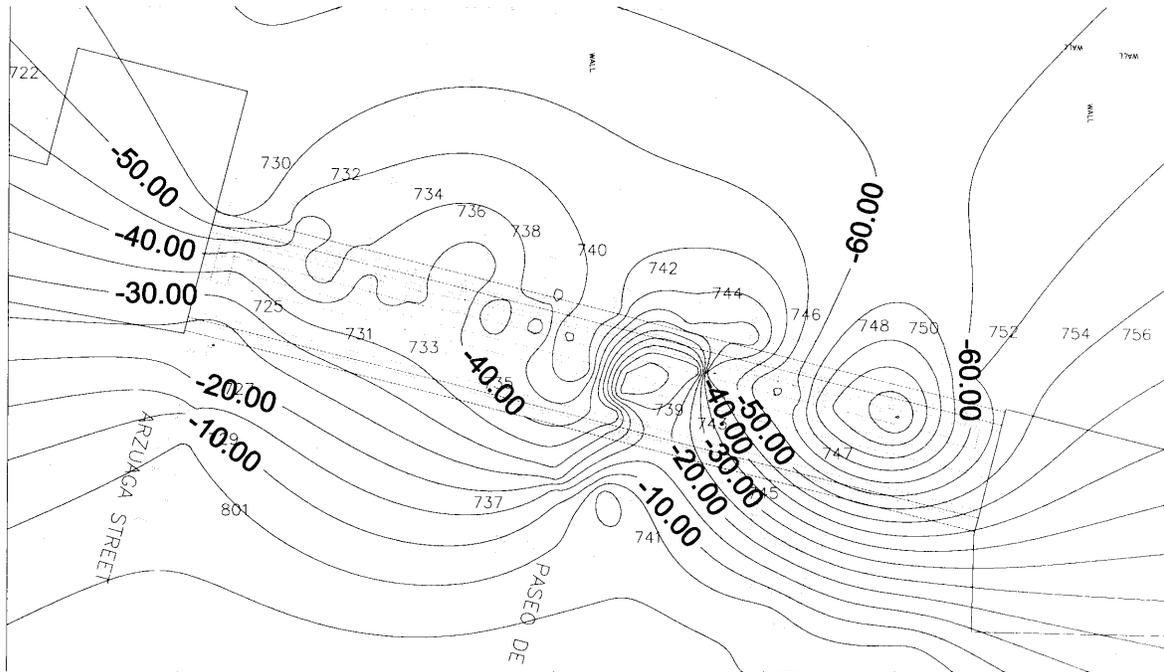


(g) Surface Settlement Contour from Construction Record for CD158

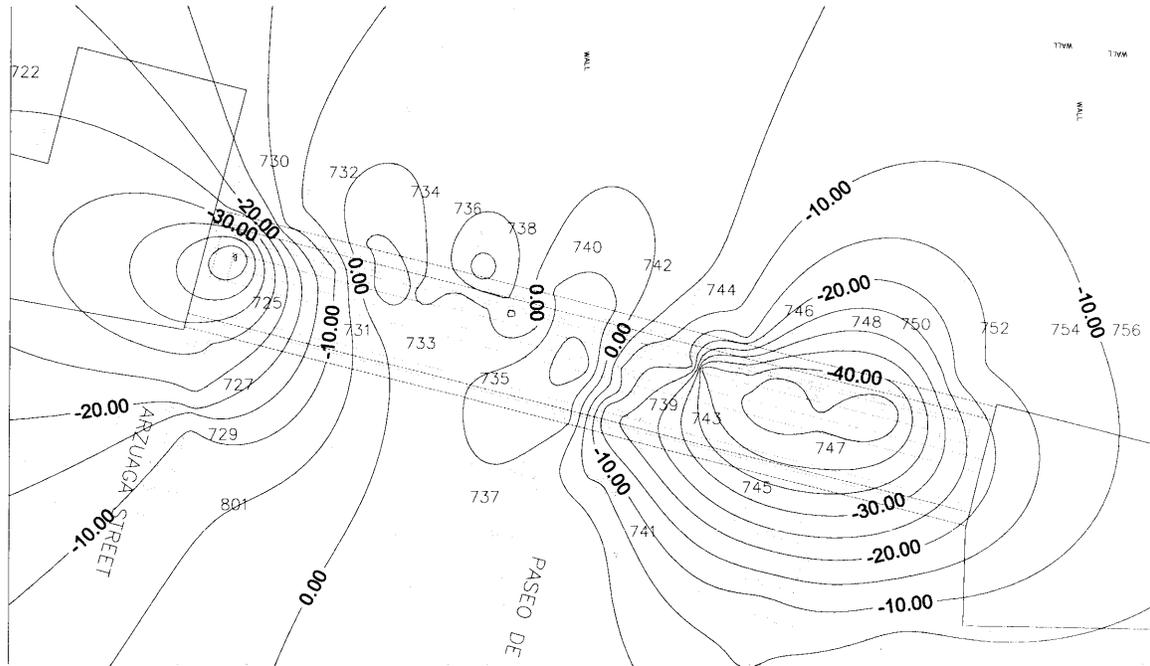


(h) Surface Settlement Contour from Construction Record for CD180

Figure H-1 (Cont'd) Surface Settlement Contours from Construction Record

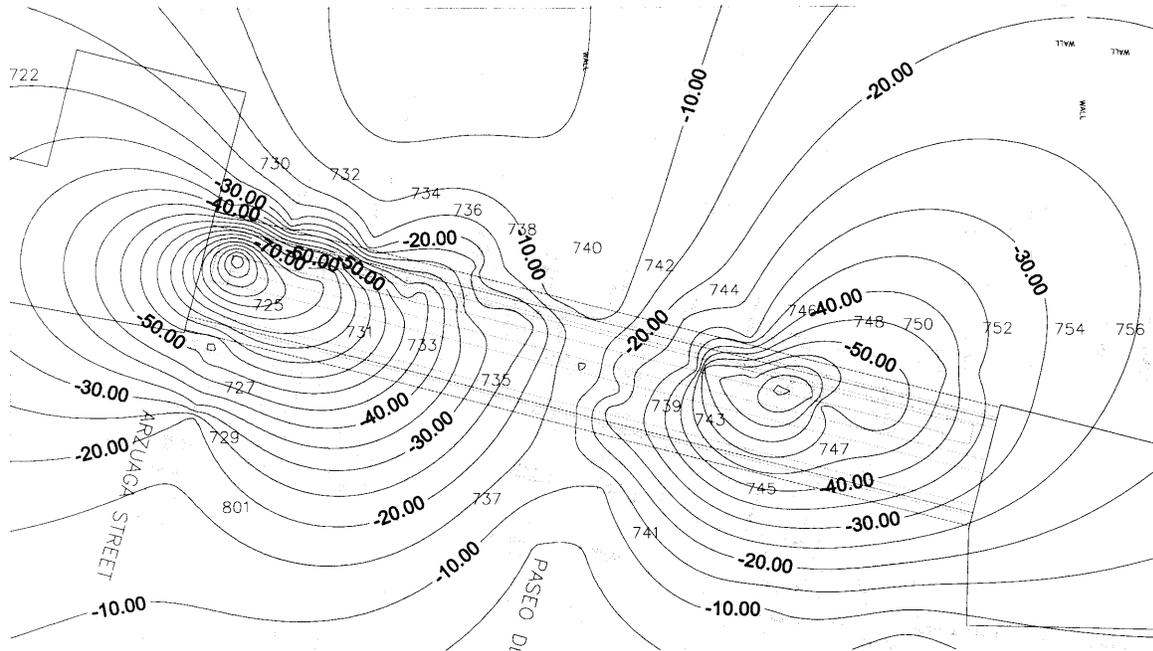


(i) Surface Settlement Contour from Construction Record for CD210

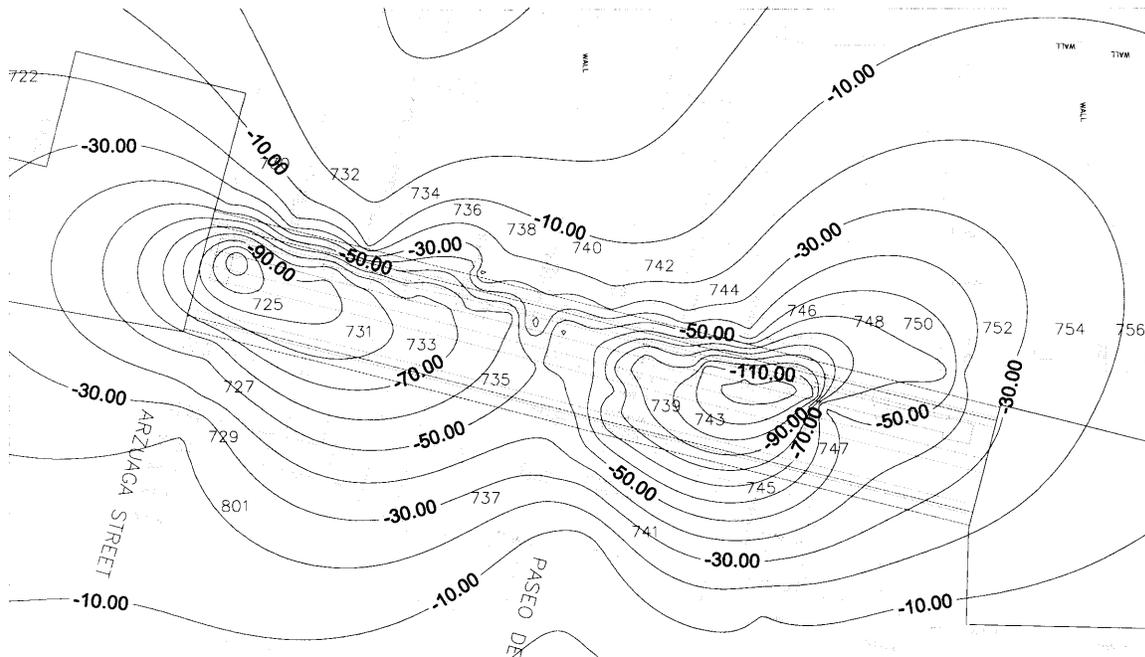


(j) Surface Settlement Contour from Construction Record for CD340

Figure H-1 (Cont'd) Surface Settlement Contours from Construction Record



(k) Surface Settlement Contour from Construction Record for CD451



(l) Surface Settlement Contour from Construction Record for CD640

Figure H-1 (Cont'd) Surface Settlement Contours from Construction Record

Appendix I

Volume Loss Calculation from Finite Element Solutions

The volume loss is defined by the ratio between the volume of surface settlement trough and the volume of excavated material:

$$VL = \frac{V_{trough}}{V_{excavation}} \quad (I.1)$$

Thus V_{trough} and $V_{excavation}$ are needed to calculate the volume loss. $V_{excavation}$ is defined by the excavation sequence, and V_{trough} can be either observed in-situ or calculated from either empirical methods or from numerical models.

V_{trough} is mathematically defined as follows:

$$V_{trough} = \begin{cases} \int_{-\infty}^{\infty} U_y dx & \forall \text{ 2-D plane-strain} \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_z dy dx & \forall \text{ 3-D} \end{cases} \quad (I.2)$$

where U_y and U_z are settlements in 2-D and 3-D analyses, and ∞ is replaced by the bounds of the defined finite element domain. There are several numerical integration methods that can be applied to evaluate (I.2), and each method has its characteristics and requirements. For example, Gauss integration requires the function evaluated at special points rather than end-points, and trapezoidal rule or mid-point rule are only exact for linear functions. Thus, the choice of integration scheme for calculating V_{trough} with finite element calculation is dependent on the finite element type used.

Linear elements (e.g. CPE4) have linear interpolation functions, and trapezoidal rule produces exact integral for linear functions, thus trapezoidal rule can be applied to calculate V_{trough} for linear elements perfectly. Quadratic elements (e.g. CPE8), however, have second-order interpolation functions, and higher-order integration schemes are needed.

The Simpson's three-point integration rule is perfect to compute the V_{trough} for quadratic elements because it is exact up to cubics (Maron and Lopez 1998), and it uses two ends points and one center-point to evaluate the integral, matching node distributions for quadratic elements. The Simpson's three-point integration rule can be expressed as:

$$\int_{x_1}^{x_2} f(x) dx \cong \frac{h}{3} [f(x_1) + 4f(x_1 + h) + f(x_2)], \quad h = \frac{x_2 - x_1}{2} \quad (I.3)$$

Although Gauss integration is also exact for quadratic elements, and is used for integrating elemental quantities such as body forces. It requires quantities (to be integrated) be evaluated on special Gauss points rather than on end-points, and displacements calculated from finite element methods are defined on nodes. Thus, Gauss integration is not the best choice for integration of 2nd order elements for calculating V_{trough} .

Applying Simpson's three-point integration rule to calculate V_{trough} for 2-D quadratic finite elements is straightforward. Three sampling points required for evaluating the integral correspond to the three nodes on the side of quadratic elements, and mid-side node corresponds to the central point in the integration scheme. Thus, the surface settlement trough can be calculated simply by applying the following calculation:

$$V_{trough} = \sum_{i=1}^{N_E} \frac{x_2^i - x_1^i}{6} [U_{z1}^i + 4U_{z3}^i + U_{z2}^i] \quad (I.4)$$

In Equation (I.4), the superscript i is the index looping through N_E finite elements that constitute the ground surface, subscript indicates the local nodal number (1, 2 are corners nodes, and 3 is the mid-side node), x is the nodal horizontal coordinate, and U_z is the calculated vertical movements defined on nodes.

Calculating V_{trough} for 3-D finite element analyses with quadratic elements need to apply Simpson's integration scheme four times. The following steps summarized the procedure to calculate V_{trough} for 3-D finite element solutions:

- 1) Identify the face corresponding to the ground surface from 3-D finite elements. Each face is composed by 8 nodes and is equivalent to 2-D 8-noded quadratic elements (e.g. CPE8).

- 2) For each identified 8-noded face, interpolate the mid-face node and its displacement using interpolation function of quadratic elements such as CPE8. Thus, there are 9 nodes within each face element constitutes the ground surface.
- 3) Assuming each face element aligns with X and Y axes, then the 9 nodes result in three lines of nodes in the X direction, and each line has three nodes.
- 4) The 3-point Simpson's rule is applied on the three lines of nodes determined from previous step, thus results in three integrals. The Simpson's rule is again applied on the three integrals and the volume of settlement trough above the face element is then obtained.

The above steps summarized the calculation procedures for the volume of surface settlement trough with quadratic elements.

Appendix J

Building Damage Assessment Charts

This appendix is based on the work of Mair et al. (1996) for evaluating building damages due to ground surface settlement. The strain of a simply supported beam loaded at center can be associated with the central deflection:

$$\frac{\Delta}{L} = \left[\frac{L}{12t} + \frac{3I}{2tLH} \frac{E}{G} \right] \varepsilon_b = C_b \varepsilon_b \quad (\text{J.1})$$

$$\frac{\Delta}{L} = \left\{ 1 + \frac{HL^2}{18I} \frac{G}{E} \right\} \varepsilon_s = C_s \varepsilon_s \quad (\text{J.2})$$

where Δ is the deflection in the center of the simply supported beam, L is the length of the beam, and H is the height of the beam. t ranges from 0 to H and is the location of the neutral axis, and 0 is on the top of the beam. E is Young's modulus, G is shear modulus, I is the moment of inertia about neutral axis. ε_b is the strain in the beam due to bending, and ε_s is the strain due to shearing.

By imposing a horizontal strain ε_h to the beam, the deflection-induced strains in the beam are altered. The bending induced strain ε_b is in the same direction as the horizontal strain ε_h , thus can be combined by direct summation. Therefore, the relationship between the horizontal strain and deflection ratio Δ/L can be obtained with the aid of Equation (J.1) and a known strain limit ε_{lim} :

$$\begin{aligned} \varepsilon_{\text{lim}} &= \varepsilon_b + \varepsilon_h = \frac{1}{C_b} \frac{\Delta}{L} + \varepsilon_h \\ \Rightarrow \frac{\Delta}{L} &= C_b (\varepsilon_{\text{lim}} - \varepsilon_h) \end{aligned} \quad (\text{J.3})$$

Shear-induced strain ε_s and horizontal strain ε_h are in different direction, thus the combined effect needs to be obtained by considering rotation (e.g. Mohr circle). The combined strain with an assumed Poisson's ratio of 0.3 can be calculated as:

$$\varepsilon'_s = 0.35\varepsilon_h + \left[(0.65\varepsilon_h)^2 + \varepsilon_s^2 \right]^{0.5} \quad (\text{J.4})$$

Substitute Equation (J.2) into Equation (J.4) and let ε'_s be a given limiting strain ε_{lim} , the relationship between ε_h and Δ/L is also defined:

$$\frac{\Delta}{L} = C_s \sqrt{(\varepsilon_{\text{lim}} + 0.30\varepsilon_h)(\varepsilon_{\text{lim}} - \varepsilon_h)} \quad (\text{J.5})$$

Mair et al. (1996) suggests separate assessments of the building in the sagging mode and in the hogging mode. The neutral axis is assumed to be in the bottom of the beam for the hogging mode, and in the center for the sagging mode. Thus, Table J-1 can be obtained for sagging and hogging mode using Equation (J.1) and Equation (J.2) under these assumptions.

Table J-2 lists the suggested limiting strain and associated damage categories by Mair et al. (1996). Damage assessment charts similar to Mair et al. (1996) can be constructed using Equation (J.3), Equation (J.5), Table J-1 and Table J-2 for various E/G and H/L . With isotropic elasticity and Poisson's ratio of 0.3:

$$\frac{E}{G} = 2(1 + \nu) = 2.6 \quad (\text{J.6})$$

For masonry structures, it is recommended to use 12.5 rather than 2.6 (Burland and Wroth 1974) for the ratio E/G .

To determine if the building under assessment is controlled by bending or by shearing, the maximum allowable deflection in bending Δ_b and in shearing Δ_s are compared using (J.3) and (J.5) assuming the limiting strain ε_{lim} and horizontal ε_h are known:

$$\frac{\Delta_b}{\Delta_s} = \frac{C_b (\varepsilon_{\text{lim}} - \varepsilon_h)}{C_s \sqrt{(\varepsilon_{\text{lim}} + 0.30\varepsilon_h)(\varepsilon_{\text{lim}} - \varepsilon_h)}} = \frac{C_b}{C_s} \sqrt{\frac{1 - \frac{\varepsilon_h}{\varepsilon_{\text{lim}}}}{1 + 0.30 \frac{\varepsilon_h}{\varepsilon_{\text{lim}}}}} \quad (\text{J.7})$$

If the ratio Δ_b/Δ_s is greater than one, then shearing and Equation (J.5) determines the building damage; if the ratio Δ_b/Δ_s is less than one, then bending and Equation (J.3) dictates the building damage. Equation (J.8) can then be derived from (J.7). If Equation (J.8) is true, then (J.5) should be used to determine building damage, otherwise (J.3) is used.

$$\frac{\varepsilon_h}{\varepsilon_{\text{lim}}} < \frac{1 - \left(\frac{C_s}{C_b}\right)^2}{1 + 0.30 \left(\frac{C_s}{C_b}\right)^2} \quad (\text{J.8})$$

Based on Equation (J.8), Figure J-1 is developed to help determine if shear damage dominates or bending-damage dominates. For given limiting and horizontal strains (tension

as positive) and dimension information H/L , shear damage occurs earlier than bending damage if a point falls under the curve of choice (based on Poisson's ratio) and vice versa.

It is seen from Figure J-1 that shear damage is more likely to occur in the hogging mode than in the sagging mode. It is also seen from these curves in Figure J-1 converge to either 1 with high H/L ratio or $-1/0.3$ with low H/L ratio. This is because 1 and $-1/0.3$ are two limits of $\varepsilon_h/\varepsilon_{lim}$ ratios, 1 means the horizontal strain is equal to the limiting strain, thus crack occurs vertically due to horizontal tensile strain, and $-1/0.3$ corresponds to tensile vertical strain induced due to Poisson's effect from compressive horizontal strain with an assumed Poisson's ratio of 0.3 in Equation (J.5).

Figure J-2 is a damage assessment chart similar to Mair et al. (1996) with the following additions:

- 1) Both compressive and tensile horizontal strains are included in Figure J-2, while Mair et al. (1996) considers only the tensile horizontal strain. Based on Figure J-2, compressive horizontal strain can cause damage to the building due to tensile vertical strain (caused by Poisson's effect). However, tensile vertical strain may not open horizontal tensile cracks due to gravity.
- 2) Both sagging and hogging modes are considered. It is seen from Figure J-2 that the sagging and hogging mode share the same endpoints of envelopes on the horizontal axis that has zero deflection and the maximum tensile strain is determined entirely by the horizontal strain. It is also seen in the sagging mode, the damage is mostly controlled by bending (the linear portion of the envelope) as expected from Figure J-1.

The damage assessment chart of Boscardin and Cording (1989) considers only the shear mode failure, and uses angular distortion β to describe the shearing of the beam rather than using the deflection ratio Δ/L . Thus, additional equations are needed to convert Δ/L to β :

$$\beta = 3 \left[\frac{1 + 4 \left(\frac{E}{G} \right) \left(\frac{H}{L} \right)^2}{1 + 6 \left(\frac{E}{G} \right) \left(\frac{H}{L} \right)^2} \right] \frac{\Delta}{L} = C_\beta \frac{\Delta}{L} \quad (J.9)$$

With Equation (J.5), (J.9) and Table J-1 in hogging mode and Table J-2, the damage assessment chart of Boscardin and Cording (1989) can be obtained.

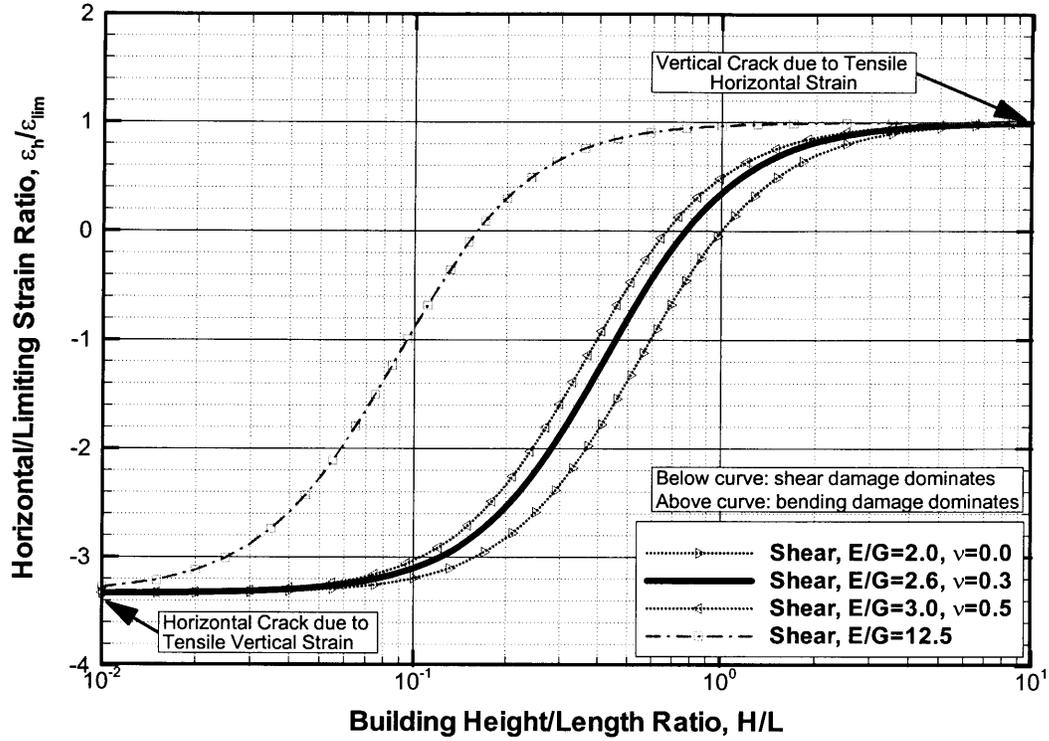
Table J-1 Geometric Factors for Sagging and Hogging

	t	I	C_b	C_s
Hogging	H	$\frac{H^3}{3}$	$\frac{1}{12}\left(\frac{L}{H}\right) + \frac{1}{2}\left(\frac{H}{L}\right)\frac{E}{G}$	$1 + \frac{1}{6}\left(\frac{L}{H}\right)^2\frac{G}{E}$
Sagging	$\frac{H}{2}$	$\frac{H^3}{12}$	$\frac{1}{6}\left(\frac{L}{H}\right) + \frac{1}{4}\left(\frac{H}{L}\right)\frac{E}{G}$	$1 + \frac{2}{3}\left(\frac{L}{H}\right)^2\frac{G}{E}$

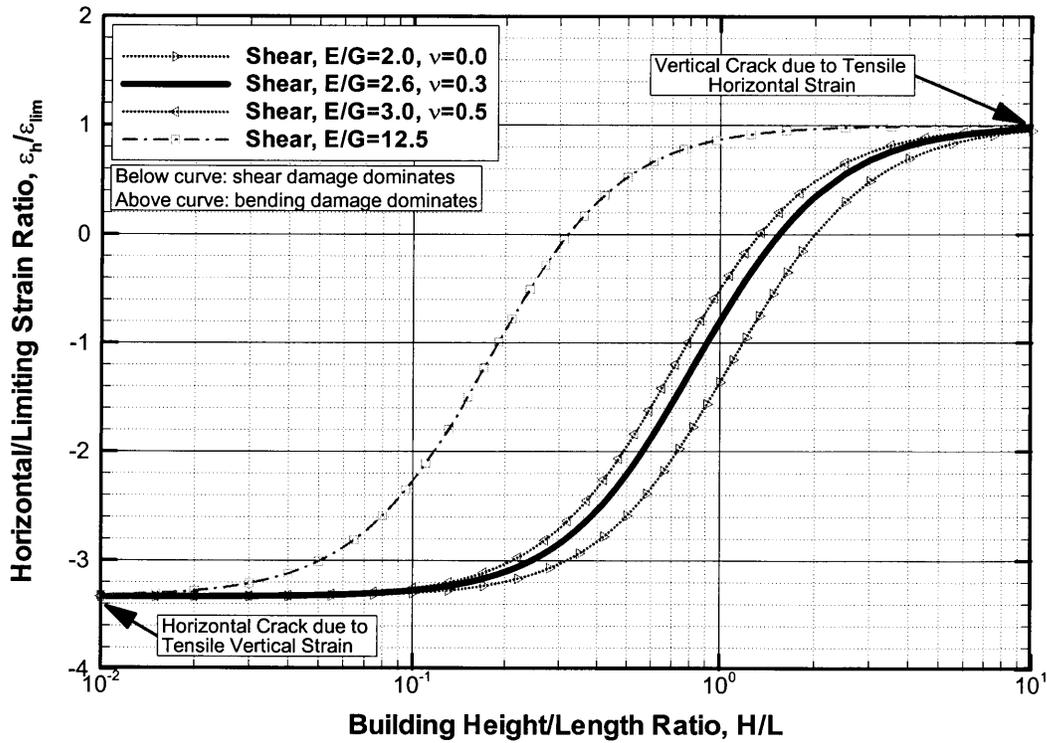
$$C_b = \left[\frac{L}{12t} + \frac{3I}{2tLH} \frac{E}{G} \right], C_s = \left[1 + \frac{HL^2}{18I} \frac{G}{E} \right]$$

Table J-2 Relationship between Category of Damage and Limiting Tensile Strain
(Mair et al. 1996)

Category of damage	Normal degree of severity	Limiting tensile strain ϵ_{lim}
0	Negligible	0 – 0.05
1	Very slight	0.05 – 0.075
2	Slight	0.075 – 0.15
3	Moderate	0.15 – 0.3
4 - 5	Severe to very severe	> 0.3



(a) Hogging mode



(b) Sagging Mode

Figure J-1 Shear Damage Criterion

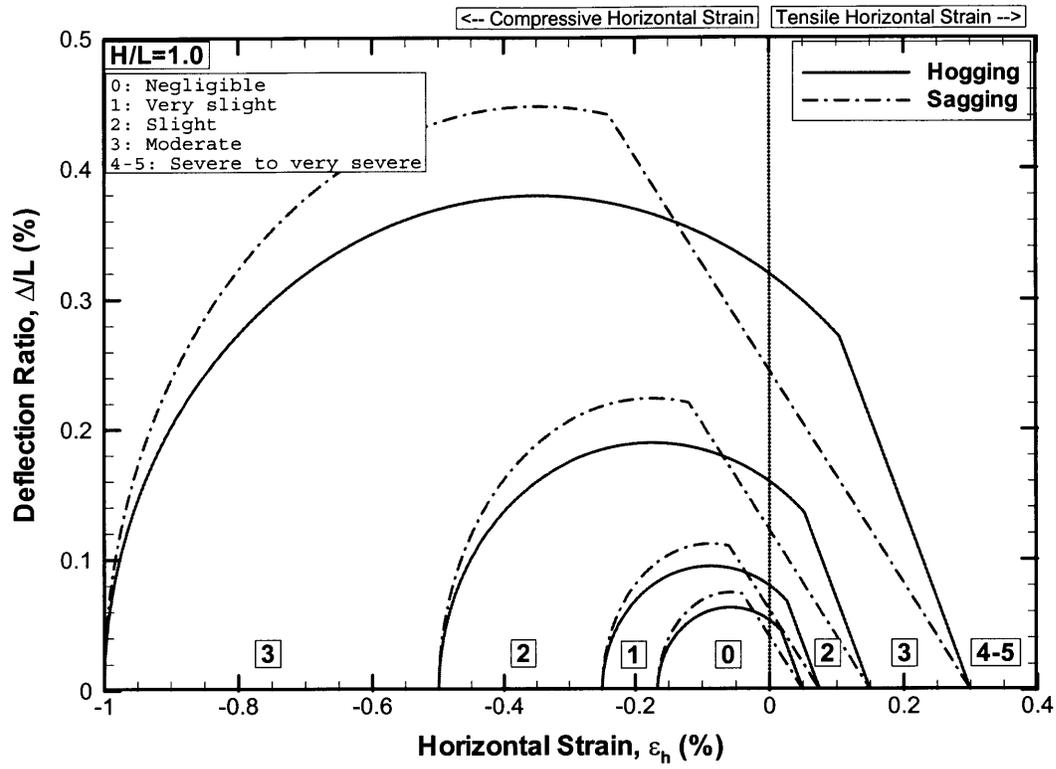


Figure J-2 Damage Assessment Chart for $H/L = 1$

Reference

- Boscardin, M. D., and Cording, E. J. (1989). "Building response to excavation-induced settlement." *Journal of Geotechnical Engineering*, 115(1), 1-21.
- Burd, H. J., Houlsby, G. T., Chow, L., Augarde, C. E., and Liu, G. "Analysis of settlement damage to masonry structures." *Numerical Methods in Geotechnical Engineering*, 203-209.
- Burland, J. B., and Wroth, C. P. "Settlement of buildings and associated damage." *Conf. Settlement of Structures*, Cambridge, 611-654.
- Chatzigiannelis, Y., and Whittle, A. J. (2003). "Deformation caused by shallow tunneling in cross-anisotropic soil." *submitted for publication*.
- Duff, I., Grimes, R., and Lewis, J. (1989). "Sparse matrix test problems." *ACM Transactions on Mathematical Software*, 15, 1-14.
- Gioda, G., and Swoboda, G. (1999). "Developments and Applications of the Numerical Analysis of Tunnels in Continuous Media." *Int. J. Numer. Anal. Meth. Geomech.*, 23, 1393-1405.
- Lee, K. M., and Rowe, R. K. (1989). "Deformations caused by surface loading and tunnelling: the role of elastic anisotropy." *Geotechnique*, 39(1), 125-140.
- Li, X. S. (1996). "Sparse Gaussian elimination on high performance computers," Ph.D. dissertation, U. C. Berkeley.
- Mair, R. J., Taylor, R. N., and Burland, J. B. (1996). "Prediction of Ground Movements and Assessment of Risk of Building Damage due to Bored Tunnelling." *Geotechnical Aspects of Underground Construction in Soft Ground*, Mair and Taylor, eds., 713-718.
- Maron, M. J., and Lopez, R. J. (1998). *Numerical Analysis: A Practical Approach*, Wadsworth Publishing.
- Peck, R. B. (1969). "Deep excavations and tunneling in soft ground." University of Illinois, Urbano, 225-290.
- Pestana, J. M., and Whittle, A. J. (1999). "Formulation of a unified constitutive model for clays and sands." *Int. J. Numer. Anal. Meth. Geomech.*, 23, 1215-1243.
- Pinto, F. (2000). *Analytical methods to interpret ground deformations due to soft ground tunneling*.
- Pinto, F., and Whittle, A. J. "Comparison of analytical solutions for ground movements caused by shallow tunneling in soil." *Journal of Engineering Mechanics*, submitted for publication.
- Sagaseta, C. (1987). "Analysis of undrained soil deformation due to ground loss." *Geotechnique*, 37(3), 301-320.
- Sagaseta, C. "On the role of analytical solutions for the evaluation of soil deformations around tunnels." *Application of Numerical Methods to Geotechnical Problems*, 3-24.
- Schweiger, H. F., Schuller, H., and Hottler, R. "Some remarks on 2-D models for numerical simulation of underground constructions with complex cross-sections." *Computer Methods and Advances in Geomechanics*, 1303-1308.
- Verruijt, A. (1997). "Complex variable solution for a deforming circular tunnel in an elastic half-plane." *International Journal for Numerical and Analytical Methods in Geomechanics*, 21(2), 77-89.
- Verruijt, A., and Booker, J. R. (1996). "Surface settlements due to deformation of a tunnel in an elastic half plane." *Geotechnique*, 46(4), 753-756.