

MERLIN

Theory Manual



Prepared by:

Victor Saouma
Department of Civil Engineering,
University of Colorado, Boulder
Boulder, CO 80309-0428

Under Contract from:
Tokyo Electric Power Service Company
3-3-3 Higashiueno, Taito-ku, Tokyo 110-0015

Electric Power Research Institute
3412 Hillview Avenue
Palo Alto, California 94304

July 10, 2010

Contents

I THEORY	19
1 OVERVIEW of SEISMIC EVALUATION	23
1.1 Design earthquake criteria	23
1.2 Method of analysis	23
1.2.1 Simplified procedures	24
1.2.2 Response-spectrum modal analysis	24
1.2.3 Time-history analysis	24
1.2.3.1 Concrete Gravity Dams	25
1.2.3.1.1 2-D gravity dam model	25
1.2.3.1.2 3-D gravity dam model	25
1.2.3.2 Arch Dams	25
1.2.3.2.1 Dam model	26
1.2.3.2.2 Foundation model	26
1.2.3.2.3 Reservoir water	26
1.3 Load combinations	26
1.3.1 Deconvolution	27
1.4 Development of structural models	27
1.4.1 Dam Models	27
1.4.1.1 Concrete gravity dams	27
1.4.1.2 Concrete arch dams	28
1.4.2 Fluid Structure interaction	28
1.4.2.1 Simplified Added Hydrodynamic Mass Model	28
1.4.2.1.1 Westergaard added mass	28
1.4.2.1.2 Generalized Westergaard added mass	28
1.4.2.2 Finite Element Added Hydrodynamic Mass Model	29
1.4.2.3 Compressible Water with Absorptive Boundary Model	29
1.4.2.4 Reservoir Boundary Absorption	30
1.4.3 Foundation Structure Interaction	30
1.4.3.1 Massless finite element foundation model	30
1.4.3.2 Viscoelastic foundation rock model	31
1.5 Material properties	31
1.5.1 Concrete properties	31
1.5.2 Foundation rock properties	32
1.5.3 Reservoir bottom absorption	32
1.5.4 Damping	32
1.6 Numerical analysis procedures	32
1.6.1 Analysis in the time domain	33
1.6.2 Analysis in the frequency domain	33
1.7 Structural performance and damage criteria	33
1.7.1 Gravity dams	34
1.7.2 Arch dams	34
2 CONSTITUTIVE MODEL FOR ALKALI AGGREGATE REACTIONS	35
2.1 INTRODUCTION	35
2.2 Chemical Reactions	35
2.3 LITERATURE SURVEY	36
2.4 MODEL	38
2.5 VALIDATION	43

2.6 APPLICATION	47
2.6.1 Dam Analysis Data Preparation	47
2.6.2 Dam Analysis Results	47
2.7 CONCLUSIONS	50
2.8 Appendices	50
2.8.1 Example of Weight Determination	50
2.8.2 Derivation of Kinetics Relation	51
3 PSEUDO-HYDRODYNAMIC FORCES	55
3.1 Westergaard	55
3.1.1 Static Analysis; Pseudo Hydrodynamic Forces	55
3.1.2 Dynamic Analysis; Added Masses	56
3.2 Zangar	58
4 NEARLY INCOMPRESSIBLE ELEMENTS	61
4.1 Consequences of Material Incompressibility	61
4.2 Displacement Based Formulation	62
5 FOUNDATION MODELLING	63
5.1 Wave Equation	63
5.2 Viscous Boundary Conditions; Lysmer Model	64
5.3 Finite Element Implementation	64
5.3.1 Passive/Rigid Boundary; Lysmer	64
5.3.1.1 Modeling	66
5.3.1.2 Reservoir Model	66
5.3.2 Active/Flexible Boundary; Miura	66
5.3.2.1 Finite Element Implementation	70
6 DECONVOLUTION	75
6.1 Introduction	75
6.2 Fourier Transform	75
6.3 Butterworth Filter	76
6.4 Transfer Function	76
6.5 Deconvolution	76
6.5.1 1-D	76
6.5.2 3-D	78
6.5.2.1 Simplification	79
6.5.3 Example	79
7 HU-WASHIZU; MIXED ITERATIVE METHODS	83
7.1 Multifield Variational Principles	83
7.2 General Hu-Washizu Variational Principle	83
7.3 Discretization of the Variational Statement for the HW Variational Principle	85
7.4 Element Formulation	88
7.5 Strain Recovery	89
7.5.1 C-lumping	89
7.5.2 Strain smoothing	90
7.5.3 C-splitting	90
7.6 Uniqueness and Existence of a Solution	91
8 MATERIAL NONLINEARITIES	93
8.1 Introduction	93
8.1.1 Linearization	93
8.1.2 Solution Strategies	94
8.2 Load Control	95
8.2.1 Newton-Raphson	95
8.2.1.1 Newton-Raphson/Tangent Stiffness Method	95
8.2.1.2 Modified Newton-Raphson	96
8.2.1.3 Secant Newton	98
8.2.2 Acceleration of Convergence, Line Search Method	98

8.2.3 Convergence Criteria	99
8.3 Direct Displacement Control	101
8.4 Indirect Displacement Control	102
8.4.1 Partitioning of the Displacement Corrections	102
8.4.2 Arc-Length	105
8.4.3 Relative Displacement Criterion	106
8.4.4 IDC Methods with Approximate Line Searches	106
9 TRANSIENT ANALYSIS; Direct Integration Schemes	109
9.1 Implicit	109
9.1.1 Newmark's β Method	109
9.1.2 Hughes α Method	111
9.1.2.1 Algorithm	113
9.2 Explicit	114
9.2.1 Preliminaries	114
9.2.2 Algorithm	116
9.2.3 Dynamic Relaxation	117
9.3 Rayleigh Damping	118
10 EXPLICIT PARALLEL	121
10.1 Introduction	121
10.1.1 Parallel Computational Models	121
10.1.2 Solution Strategies Based on Message Passing	122
10.1.2.1 Domain Partitioning	123
10.1.3 Parallelization of the Solution Method	123
10.2 MPI – Message Passing Interface	124
10.2.1 Point to Point Communication	124
10.2.2 Collective Communication	125
10.2.3 Groups, Contexts, and Communicators	125
10.2.4 Datatypes	126
10.2.5 Binding to Fortran 77	126
10.2.6 Example of Parallel Fortran Program	126
10.3 Explicit	128
10.3.1 Preliminaries	128
10.3.2 Parallelization Concept based on Node-Cut MeshPartitioning	129
10.3.3 Algorithm	129
10.3.3.1 Note About Interface Elements	132
11 HOURGLASS STABILIZATION	133
12 EMBEDDED REINFORCEMENT	135
13 SINGULAR ELEMENT	137
13.1 Introduction	137
13.2 Quarter Point Singular Elements	137
13.3 Review of Isoparametric Finite Elements	137
13.4 How to Distort the Element to Model the Singularity	139
13.5 Order of Singularity	140
13.6 Stress Intensity Factors Extraction	140
13.6.1 Isotropic Case	141
13.6.2 Anisotropic Case	142
14 RECIPROCAL WORK INTEGRALS	145
14.1 General Formulation	145
14.2 Volume Form of the Reciprocal Work Integral	148
14.3 Surface Tractions on Crack Surfaces	149
14.4 Body Forces	149
14.5 Initial Strains Corresponding to Thermal Loading	150
14.6 Initial Stresses Corresponding to Pore Pressures	151
14.7 Combined Thermal Strains and Pore Pressures	151

14.8 Field Equations for Thermo- and Poro-Elasticity	152
15 J INTEGRAL BASED METHODS	155
15.1 Numerical Evaluation	155
15.2 Mixed Mode SIF Evaluation	157
15.3 Equivalent Domain Integral (EDI) Method	158
15.3.1 Energy Release Rate J	158
15.3.1.1 2D case	158
15.3.1.2 3D Generalization	160
15.3.2 Extraction of SIF	162
15.3.2.1 J Components	162
15.3.2.2 σ and u Decomposition	162
16 HILLERBORG'S MODEL	165
17 LOCALIZED FAILURE	169
17.1 Fictitious Crack Model; FCM (MM: 7)	169
17.1.1 Introduction	169
17.1.2 Weak Form of Governing Equations	169
17.1.3 Discretization of Governing Equations	171
17.1.4 Penalty Method Solution	173
17.1.5 Incremental-Iterative Solution Strategy	174
17.2 Interface Crack Model; ICM-1 Original (MM: 8)	175
17.2.1 Introduction	176
17.2.2 Interface Crack Model	176
17.2.2.1 Relation to fictitious crack model	181
17.2.3 Finite Element Implementation	182
17.2.3.1 Interface element formulation	182
17.2.3.2 Constitutive driver	184
17.2.3.3 Non-linear solver	187
17.2.3.4 Secant-Newton method	187
17.2.3.5 Element secant stiffness	188
17.2.3.6 Line search method	189
17.2.4 Mixed Mode Crack Propagation	190
17.2.4.1 Griffith criterion and ICM	190
17.2.4.2 Criterion for crack propagation	191
17.3 Interface Crack Model; ICM-2 Cyclic (MM: 21)	192
17.3.1 Introduction	193
17.3.2 Cyclic behavior of quasi brittle interfaces	193
17.3.3 Červenka 1994 hyperbolic model	194
17.3.4 Proposed extension to cyclic loading	195
17.3.4.1 Analytical formulation	196
17.3.4.1.1 Asperity definition	196
17.3.4.1.2 Asperity degradation	198
17.3.4.1.3 Rotated activation function	199
17.3.4.1.4 Remarks	200
17.3.5 Computational tests	200
17.3.5.1 Comparison with Kutter and Weissbach test	200
17.3.5.2 Comparison with Červenka's model	202
17.3.6 Conclusions	202
17.3.7 Acknowledgements	202
17.4 Notation	204
17.5 Interface Crack Model; ICM-3-Mohr-Coulomb (MM: 22)	205
17.6 Interface Crack Model; ICM-3-Hyperbolic-Light (MM: 23)	205
18 DISTRIBUTED FAILURE; Fracture Plastic Model (MM:15, 16, 18, 19)	209
18.1 Material Model Formulation	209
18.2 Rankine-Fracturing Model for Concrete Cracking	209
18.3 Plasticity Model for Concrete Crushing	211
18.4 Combination of Plasticity and Fracture model	214

18.5 Confinement Sensitive Fracture-Plastic Model, MM: 18	215
18.5.1 Summary of Main Improvements over MM 19	221
18.5.1.1 Confinement Sensitivity of Stress-Strain response	221
18.5.1.2 Shear retention factor	221
18.5.1.3 Elements of compression field theory	221
18.5.1.4 Improved model stability	221
18.6 Validation Test Problems, MM-19	221
18.6.1 Descrizione del provino	224
18.6.2 Prova uniaxiale di trazione	224
18.6.2.1 L'effetto della mesh	225
18.6.2.2 L'effetto della localizzazione del difetto	225
18.6.2.3 Effetto dell'energia di frattura sui risultati delle prove uniaxiali di trazione	226
18.6.3 Prova uniaxiale di compressione	226
18.6.4 Imposizione di un carico termico	228
18.6.5 Prova di carico ciclico	229
18.6.6 Conclusioni	231
19 NONLINEAR ROCK MODELS	233
19.1 Model	233
19.2 Test Results	234
II SYSTEM	237
20 GETTING READY	239
20.1 Preliminary Considerations; Dam Analysis	239
20.1.1 LEFM	239
20.1.2 Strength Based	239
20.1.3 NLFM	240
20.1.3.1 Incremental NLFM	240
20.1.3.2 Failure/Post-Peak	241
20.1.4 Uplift Pressures	242
20.1.5 Dynamic Analysis	243
20.2 Material Properties	243
20.2.1 Concrete	243
20.2.1.1 Basic Properties	243
20.2.1.2 Linear Elastic Fracture Properties	243
20.2.1.3 Nonlinear Fracture Properties	244
20.2.1.4 Dam Concrete	245
20.2.2 Rock	245
20.2.2.1 Basic Properties	245
20.2.2.2 Linear Elastic Fracture Properties	245
20.2.2.3 Nonlinear Fracture Properties	245
20.2.3 Interface	246
20.2.3.1 Basic Properties	246
20.2.3.2 Linear Elastic Fracture Properties	246
20.2.3.3 Nonlinear Fracture Properties	246
20.3 Load	246
20.3.1 Gravity	246
20.3.2 Thermal	247
20.3.3 Water and Silt Pressures	247
20.3.4 Uplift Pressure	248
20.3.4.1 Cracked Zone	248
20.3.4.2 Uncracked Zone	250
20.4 Finite Element Discretization	255
20.4.1 Mesh Dimensions	255
20.4.2 Boundary Conditions	255
20.4.3 Preliminary Cracks	255
20.4.3.1 Horizontal Crack	255

20.4.3.2 Rock Tensile Zone Cracks	256
20.4.4 Element Types, and Mesh Density	256
20.5 Stress Analysis	256
20.5.1 Linear versus Nonlinear Analysis	256
20.5.2 Two versus Three-Dimensional Analysis	257
20.5.3 Stress Intensity Factor Extraction	257
20.6 Seepage Analysis	257
20.6.1 Material Properties	257
20.6.2 Finite Element Discretization	257
20.7 Thermal Analysis	257
20.7.1 Material Properties	258
20.7.2 Heat Transfer	258
20.7.3 Boundary Conditions	258
20.7.4 Seepage Analysis	259
20.7.5 Boundary Conditions	259
20.8 Units & Conversion Factors	260
20.9 Metric Prefixes and Multipliers	261
21 PROGRAMMER's MANUAL	263
21.1 Introduction	263
21.1.1 Scope of Document	263
21.1.2 Organization of Document	263
21.1.3 File Naming Conventions For Source Code	263
21.1.4 Creating an Executable	263
21.1.5 Coding Standards	263
21.1.5.1 Include Files	264
21.1.5.2 Case Sensitivity	264
21.1.5.3 Variable Declarations	264
21.1.5.4 DO Loops	264
21.1.5.5 RETURN Statement	264
21.1.5.6 Statement Labels	264
21.1.5.7 ANSI Standard Features	264
21.2 File I/O Utilities	265
21.2.1 I/O Utilities Written In C	265
21.2.1.1 File Attribute Data Structure	265
21.2.1.2 Open Function	265
21.2.1.3 Close Function	266
21.2.1.4 Error Output	266
21.2.1.5 Read Functions	266
21.2.1.6 Write Functions	268
21.2.1.7 Seek Function	268
21.2.1.8 Flush Function	269
21.2.2 I/O Utilities Written In FORTRAN	269
21.2.3 Usage of the File I/O Utilities	269
21.3 Program Memory Management	272
21.3.1 Program Memory	272
21.3.2 Data Structures	272
21.3.3 Memory Management Utilities	272
21.3.3.1 Partitioned Program Memory	272
21.3.3.2 Memory Management Routines	273
21.3.3.3 MERLIN Implementation	275
21.3.3.4 Usage of Memory Management Routines	275
21.4 Finite Element Attribute Tables	278
21.4.1 Element Type Attribute Table	278
21.4.2 Element Class Attributes Table	279
21.4.3 Element Surface Definition Table	280
21.4.4 Element Nodal DOF Table	280
21.4.5 Element Integration Rules	280
21.4.6 Surface Integration Rules	280

21.4.7 Constitutive Model State Variable Table	281
21.5 Element Information Tables	281
21.5.1 Interface Element Information Table	281
21.6 Nodal Tables	281
21.6.1 Nodal Attribute Table	281
21.7 Crack Information Tables	281
21.7.1 Crack Front Attributes Table	284
21.7.2 Crack Front List	284
21.7.3 Crack Surface Attribute Table	284
21.7.4 Crack Surface Information	284
21.8 Uplift Information Arrays	284
Bibliography	285
References	294

List of Figures

2.1	Normalized Expansion Curve ($\xi(t) = \varepsilon_{F,Vol}^{AAR}(t)/\varepsilon_{AAR}^\infty$)	36
2.2	Effect of Temperature on AAR Expansion	37
2.3	Stress Induced Cracks with Potential Gel Absorption, (Scrivener 2003)	39
2.4	Graphical Representation of Γ_c and Γ_t	39
2.5	Weight of Volumetric AAR Redistribution in Selected Cases	40
2.6	Weight Regions	42
2.7	Relative Weights	43
2.8	Degradation of E and f'_t	44
2.9	Multon's Test Parameter Identification Results for Free Expansion; Longitudinal and Corresponding Transversal Strains. Init	48
2.10	Comparison between Experimental Results of Multon and Numerical Calculation (After parameter identifications)	46
2.11	Yearly Variation of Hydrostatic and Thermal Load	48
2.12	Yearly Variation of Vertical Crest Displacements; Upper Curve based on Charlwood's Model, Lower Curve based on Proposed	49
2.13	Principal Stress Field Comparison Between proposed and State of the Practice Model (without joints)	49
2.14	Dam/Foundation Interface Joint Characteristics; Uplift; tangential and normal stresses; Crack sliding and opening displacement	49
2.15	Expansion Curve	52
2.16	Determination of the Activation Energies	53
3.1	Hydrostatic and hydrodynamic forces during earthquake excitation	55
3.2	Hydrodynamic water pressure and force	56
3.3	Sloped upstream dam face - definition of θ angle	56
3.4	Westergaard's Added Mass concept	57
3.5	Electric Analog Tray Model used by Zangar (1953)	59
3.6	Increase Pressure Coefficients for Constant Sloping Faces (Zangar 1953)	60
3.7	Pressure Coefficient Distribution Comparison of Experimental and Empirical Curves (Zangar 1953)	60
5.1	Infinitesimal Element Subjected to Elastic Wave	63
5.2	Elastic Waves in an Infinite Medium	64
5.3	Dashpot Boundary Conditions	65
5.4	Foundation Model, Radiating Fixed Foundation	65
5.5	Equivalent Spring Stiffness	65
5.6	Lysmer Modeling, 2D, Modeling for Lateral and Vertical Excitation	66
5.7	Lysmer Modeling, 2D, Modeling for Lateral Excitation	67
5.8	Lysmer Modeling, 2D, Alternative Modeling for Lateral Excitation	67
5.9	Reservoir Model Boundary Conditions	68
5.10	Foundation Model, Radiating Flexible Foundation	68
5.11	Finite Element Discretization of the free field	69
5.12	Finite Element Discretization of Dam Foundation in Account of Free Field Velocities	70
5.13	Finite Element Discretization of the free field	70
5.14	Finite Element Discretization of the Corner free field	71
5.15	Finite Element Discretization of the Side free field	71
5.16	Finite Element Discretization of the Side free field, X Acceleration	72
5.17	Finite Element Discretization of the Side free field, Y Acceleration	72
5.18	Finite Element Discretization of the Side free field, Z Acceleration	73
5.19	Finite Element Discretization of the free field; Transfer of Velocities	73
5.20	Finite Element Discretization of the free field; Rock Foundation	74
5.21	Finite Element Discretization of the free field; Outline of Procedure	74
6.1	Deconvolution	75
6.2	Low Pass (25); High Pass (50); Band Pass (25-50); Band Stop (25-50) Filters, $N = 4$	76

6.3	Low Pass (25) Filter; $N = 2, 4, 6, 8, 10, 12$	77
6.4	Transfer Function	77
6.5	Deconvolution	78
6.6	Finite Element Mesh Example for Deconvolution	79
6.7	Accelerograms of the Input and Output	80
6.8	Transfer Functions	80
6.9	Deconvoluted Signals	80
6.10	Comparison between Original and Deconvoluted Signals	81
6.11	Results of Deconvolution Analysis	81
7.1	Tonti Diagram for Hu-Washizu, (Červenka, J. 1994)	85
7.2	Patch test	92
8.1	Test Controls	94
8.2	Newton-Raphson Method	96
8.3	Modified Newton-Raphson Method, Initial Tangent in Increment	97
8.4	Modified Newton-Raphson Method, Initial Problem Tangent	97
8.5	Incremental Secant, Quasi-Newton Method	98
8.6	Schematic of Line Search, (Reich 1993)	99
8.7	Flowchart for Line Search Algorithm, (Reich 1993)	100
8.8	Divergence of Load-Controlled Algorithms	101
8.9	Hydrostatically Loaded Gravity Dam	102
8.10	Load-Displacement Diagrams with Snapback	103
8.11	Flowchart for an incremental nonlinear finite element program with indirect displacement control	104
8.12	Two points on the load-displacement curve satisfying the arc-length constraint	105
8.13	Flow chart for line search with IDC methods	107
9.1	Secant and Tangent Stiffnesses for α Method	112
9.2	Central Difference Scheme (Explicit Method); Basic Definitions	115
9.3	Algorithm for Central Difference Scheme (Explicit Method)	117
9.4	Rayleigh Damping	119
12.1	Embedded Reinforcement Across a Crack/Joint	135
13.1	Isoparametric Quadratic Finite Element: Global and Parent Element	138
13.2	Singular Element (Quarter-Point Quadratic Isoparametric Element)	140
13.3	Finite Element Discretization of the Crack Tip Using Singular Elements	141
13.4	Displacement Correlation Method to Extract SIF from Quarter Point Singular Elements	141
13.5	Nodal Definition for FE 3D SIF Determination	143
14.1	Contour integral paths around crack tip for reciprocal work integral	145
15.1	Numerical Extraction of the J Integral (Owen and Fawkes 1983)	155
15.2	Simply connected Region A Enclosed by Contours $\Gamma_1, \Gamma_0, \Gamma_+$, and Γ_- , (Anderson 1995)	159
15.3	Surface Enclosing a Tube along a Three Dimensional Crack Front, (Anderson 1995)	160
15.4	Interpretation of q in terms of a Virtual Crack Advance along ΔL , (Anderson 1995)	161
15.5	Inner and Outer Surfaces Enclosing a Tube along a Three Dimensional Crack Front	161
16.1	Hillerborg's Fictitious Crack Model	165
16.2	Concrete Strain Softening Models	166
17.1	Body Consisting of Two Sub-domains	170
17.2	Mixed mode crack propagation	176
17.3	Wedge splitting tests for different materials, (V.E., Červenka, Slowik and Chandra 1994)	177
17.4	Interface idealization and notations	178
17.5	Interface fracture	178
17.6	Failure function	179
17.7	Bi-linear softening laws	180
17.8	Stiffness degradation in the equivalent uniaxial case	181
17.9	Interface element numbering	183

17.10 Local coordinate system of the interface element.	183
17.11 Algorithm for interface constitutive model.	184
17.12 Definition of inelastic return direction.	186
17.13 Secant relationship.	188
17.14 Line search method.	190
17.15 Griffith criterion in NLFM.	190
17.16 Mixed mode crack propagation.	192
17.17 Asperity curves	197
17.18 Cyclic model: yield criterion and plastic potential	198
17.19 Comparison with Kutter-Weissbach test results	201
17.20 Červenka model vs. cyclic model	203
17.21 ICM-3-Mohr-Coulomb (MM:22)	206
17.22 ICM-3-Mohr-Coulomb (MM:23)	207
18.1 Tensile Softening and Characteristic Length, (Červenka, V. and Jendele, L. and Červenka, Jan 2002)	210
18.2 Failure Surface	212
18.3 Compressive Hardening and Softening, (van Mier 1986)	212
18.4 Plastic Predictor-Corrector Algorithm, (Červenka, V. and Jendele, L. and Červenka, Jan 2002)	213
18.5 Schematic Description of the Iterative Process in 2D, (Červenka, V. and Jendele, L. and Červenka, Jan 2002)	215
18.6 Failure Surface	216
18.7 Plastic Potential of Model 18	217
18.8 Exponential Crack Opening Law	217
18.9 Compressive Hardening/Softening	218
18.10 Shear Retention Factor	219
18.11 Compressive Strength Reduction of Cracked Model	220
18.12 Comparison between analytical and experimental results for normal concrete under triaxial compression and various confinement lateral stresses	221
18.13 Comparison between analytical and experimental results for high-strength concrete under triaxial compression and various confinement lateral stresses	221
18.14 Laterally Confined Cube (in x and y while monotonically Loaded in the z Direction	221
18.15 Stress-strain response of the triaxial test for different confinement lateral stresses (0, 4.2, 8.4 MPa)	222
18.16 Geometry of the Leonhardt Beam	222
18.17 Analysis of Leonhardt Shear Beam with Model 18 and 19	223
18.18 Comparison of the Responses of a Three Point Beand Beam Analysis with Models 18 and 19	223
18.19 In figura sono mostrate da sinistra verso destra le immagini della mesh "Coarse" e "Medium"	224
18.20 Curva carico spostamento per la mesh "Coarse" e "Medium"	225
18.21 Mesh del cubo di calcestruzzo artificialmente indebolito prima e dopo la prova uniaxiale di trazione. La resistenza a trazione	226
18.22 Curve carico spostamento per il cubo senza imperfezioni e per il cubo artificialmente indebolito	227
18.23 Curve carico spostamento relative alla prove di trazione uniaxiale con differenti valori dell'energia di frattura	227
18.24 Curve carico-spostamento relative ad una prova di compressione ottenute per la mesh "Coarse" e "Medium"	228
18.25 Mesh deformate al termine delle prove di espansione termica in assenza di vincoli di contenimento (immagine a sinistra) o in	229
18.26 Curva forza spostamenti della prova di carico ciclico	230
18.27 Particolare della curva presente in Figura 18.26 carico e scarico del provino a trazione	230
19.1 Kawamoto Model, all input parameters are shown in red	233
19.2 Kawamoto Model, Compression Test	234
19.3 Kawamoto Model, Compression Test with Unloading	234
19.4 Kawamoto Model, Tension Test with Unloading	235
19.5 Kawamoto Model, Shear Test	235
19.6 Kawamoto Model, Shear Test Cyclic	235
20.1 Uplift Pressures in a Dam	242
20.2 Uplift Pressures for Permeable and Impermeable Rock	242
20.3 Concrete Strain Softening Models	244
20.4 Forces Acting on an Element at the Foundation Surface Subjected to Internal Pressure and Normal Surface Tractions	249
20.5 Uplift Model with Impervious Rock, Concrete, and Uncracked Interface	251
20.6 Uplift Model with Impervious Rock and Concrete and Pervious Uncracked Interface	252
20.7 Pipe Analogy for Flow Along a Pervious Uncracked Interface	253
20.8 Uplift Model with Impervious Concrete and Pervious Rock and Uncracked Interface	254
20.9 Boundary Conditions for Thermal Analysis	259
20.10 Boundary Conditions for Seepage Analysis	260

21.1 Program Memory with Three Arrays	273
---	-----

List of Tables

2.1 System Identification for Multon's Tests	44
2.2 Triaxial Weights	51
4.1 Elastic Properties of Steel, Concrete and Water, (Aslam, Wilson, Button and Ahlgren 2002)	61
7.1 Functionals in Linear Elasticity	83
7.2 Comparison Between Total Potential Energy and Hu-Washizu Formulations	85
7.3 Polynomial orders of the shape functions.	88
7.4 Table of α coefficients and spectral radii for CS technique.	91
17.1 Parameters for the analyses	205
18.1 Caratteristiche del calcestruzzo utilizzato durante le prove di validazione del legame costitutivo	224
18.2 Caratteristiche delle mesh utilizzate nelle prove sul cubo di calcestruzzo	224
18.3 Descrizione della prova uniassiale di trazione	225
18.4 Energia di frattura teorica e calcolata in base alle prove di trazione simulate con il programma MERLIN	226
18.5 Descrizione della prova uniassiale di compressione	228
18.6 Descrizione della prima prova di carico ciclico	229
20.1 Fixed Water Elevation Fracture and Uplift Models	242
20.2 Required Material Parameters	243
20.3 Heat of Hydration for Concrete	247
20.4 Summary of three cases for uplift on uncracked ligament	250
20.5 Required Material Properties for Seepage Analysis	257
20.6 Material Parameters Required for a Thermal Analysis	258
21.1 File Open Modes	266
21.2 State Variables for FCM Model	281
21.3 State Variables for ICM Model	281
21.4 Interface Element Information Table (INTELM)	281
21.5 Nodal Attribute Table (nodatr)	282
21.6 Nodal ID Table (id)	282
21.7 Crack Front Attribute Table (cfatr)	282
21.8 Crack Front List (cflist)	282
21.9 Crack Surface Attribute Table (csatr)	283
21.10 Crack Surface Information (csinfo)	283
21.11 Uplift function limits (fnclim)	284
21.12 Uplift function coefficient (fncoef)	284

Part I

THEORY

Nonlinear analysis

The analysis based on nonlinear material behavior represents the greatest possible refinement and it produces the most accurate results.

However, it is also the most complex and the most costly. It requires time-history ground motion input, direct integration solution, a large main frame computer, specialized computer programs, and a considerable amount of computer time. As such, it is the last recourse in the attribute refining process. The nonlinear analysis should only be undertaken under the guidance of an expert in the field of fracture mechanics and finite element methods.

Engineering and Design - Seismic Design Provisions for Roller Compacted Concrete Dams, EP 1110-2-12, US Army Corps of Engineers, 1995.

Chapter 1

OVERVIEW of SEISMIC EVALUATION

Seismic design and evaluation of hydraulic structures generally consist of the following steps:

- Selection of design/or evaluation earthquakes.
- Selection of method of analysis.
- Development of acceleration time-histories.
 - item Definition of load combinations.
- Development of structural models.
- Definition of material properties and damping.
- Selection of numerical analysis procedures.
- Determination of performance and probable level of damage, if any.

1.1 Design earthquake criteria

Design and safety evaluation earthquakes for concrete hydraulic structures are the operating basis earthquake (OBE) and the maximum design earthquake (MDE) as required by ER 1110-2-1806.

Operating Basis Earthquake (OBE). The OBE is defined in ER 1110-2-1806 as an earthquake that can reasonably be expected to occur within the service life of the project, that is, with a 50 percent probability of exceedance during the service life. This corresponds to a return period of 144 years for a project with a service life of 100 years. The associated performance requirement is that the project function with little or no damage, and without interruption of function. The purpose of the OBE is to protect against economic losses from damage or loss of service; therefore, alternative choices of return period for the OBE may be based on economic considerations. The OBE is determined by probabilistic seismic hazard analysis (PSHA). The response spectrum method of analysis is usually adequate for the OBE excitation, except for the severe OBE ground motions capable of inducing damage. In these situations, the time-history analysis described in this manual may be required.

Maximum Design Earthquake (MDE). The MDE is defined in ER 1110-2-1806 as the maximum level of ground motion for which a structure is designed or evaluated. The associated performance requirement is that the project performs without catastrophic failure, such as uncontrolled release of a reservoir, although severe damage or economic loss may be tolerated.

1.2 Method of analysis

Seismic analysis of concrete hydraulic structures, whenever possible, should start with simplified methods and progress to a more refined analysis as needed. A simplified analysis establishes a baseline for comparison with the refined analyses, as well as providing a practical method to determine if seismic loading controls the design, and thereby offers useful information for making decisions about how to allocate resources. In some cases, it may also provide a preliminary indication of the parameters significant to the structural response. The simplified methods for computation of stresses and section forces consist of the pseudo-static or single-mode response-spectrum analysis. The simplified method for sliding and rotational stability during earthquake excitation is usually based on the seismic coefficient method. The permanent sliding displacements may be computed using Newmarks rigid block model or its numerous variants. The response-spectrum mode superposition described in EM 1110-2-6050 is the next level in the progressive method of dynamic analysis. The response-spectrum mode superposition fully accounts for the multimode dynamic behavior of the structure, but it is limited to the linear-elastic range of behavior and provides only the maximum values of the response quantities. Finally, the time-history method of analysis is used to compute deformations, stresses, and section forces more accurately by considering the time-dependent nature of the dynamic response to earthquake ground motion. This method also better represents the foundation-structure and fluid-structure interaction effects.

1.2.1 Simplified procedures

Simplified procedures are used for preliminary estimates of stresses and section forces and sliding and rotational stability due to earthquake loading. The traditional seismic coefficient is one such procedure employed primarily for the analysis of rigid or nearly rigid hydraulic structures. In this procedure the inertia forces of the structures and the added mass of water due to the earthquake shaking are represented by the equivalent static forces applied at the structure center of gravity and at the resultant location of the hydrodynamic pressures. The inertia forces are simply computed from the product of the structural mass or the added mass of water times an appropriate seismic coefficient in accordance with ER 1110-2-1806. The static equilibrium analysis of the resulting inertia forces together with the customary static forces will then provide an estimate of the stresses and section forces.

The sliding stability is determined on the basis of the limit equilibrium analysis. The sliding factor of safety is computed from the ratio of the resisting to driving forces along a potential failure surface and compared against the allowable values given in ER 1110-2-1806. The resisting forces are obtained from the cohesion and frictional forces and driving forces from the resultant of static and seismic forces in the tangential direction of the sliding surface. When the factor of safety against sliding is not attainable, the sliding may occur as the ground acceleration exceeds a critical acceleration a_c and diminish as the acceleration falls below a_c . If a hydraulic structure is treated as a rigid block, the critical acceleration a_c is estimated from the seismic inertia forces necessary to initiate sliding. The upper bound estimate of permanent sliding displacement may be obtained using Newmarks charts (Figure 2-11 of EM 1110-2-6050).

1.2.2 Response-spectrum modal analysis

The maximum linear elastic response of concrete hydraulic structures can be estimated using the response-spectrum mode superposition method described in EM 1110-2- 6050. The procedure is suitable for the design, but it can also be used for the evaluation of hydraulic structures subjected to low or moderate ground motions that are expected to produce linear elastic response. In responsespectrum analysis, the maximum values of displacements, stresses, and section forces are first computed separately for each individual mode and then combined for all significant modes and multicomponent earthquake input. The modal responses due to each component of ground motion are combined using either the square root of the sum of the squares (SRSS) or the complete quadratic combination (CQC) method. The SRSS combination method is adequate if the vibration modes are well separated. Otherwise the CQC method may be required to account for the correlation of the closely spaced modes. Finally the maximum response values for each component of ground motion are combined using the SRSS or percentage methods in order to obtain the maximum response values due to multicomponent earthquake excitation. The response-spectrum method of analysis, however, has certain limitations that should be considered in the evaluation of results. All computed maximum response values including displacements, stresses, forces, and moments are positive and generally nonconcurrent. Therefore, a plot of deformed shapes and static equilibrium checks cannot be performed to validate the results. For computation of section forces from element stresses, appropriate signs should be assigned to the stresses by careful examination of deflected shapes of the predominant response modes. Alternatively, section forces may be computed first for each individual mode and then combined for the selected modes and multicomponent earthquake input, a capability that may not exist in most finite-element computer programs. Other limitations of the response-spectrum method are that the structure-foundation and structure-water interaction effects can be represented only approximately and that the time-dependent characteristics of the ground motion and structural response are ignored.

1.2.3 Time-history analysis

Time-history earthquake analysis is conducted to avoid many limitations of the response-spectrum method and to account for the time-dependent response of the structure and better representation of the foundation-structure and fluid-structure interaction effects. The earthquake input for timehistory analysis is usually in the form of acceleration time-histories that more accurately characterize many aspects of earthquake ground motion such as the duration, number of cycles, presence of high-energy pulse, and pulse sequencing. Time-history analysis is also the only appropriate method for estimation of the level of damage as described in 1-7h and Chapter 4. Response history is computed in the time domain using a step-by-step numerical integration or in the frequency domain by applying Fourier transformation described in 1-7g.

In the standard finite element approach, the complete system consisting of the structure, the water, and the foundation region is modeled and analyzed as a single composite structural system (Figure 2-3b). Similar to the substructure approach, the structure is modeled as an assemblage of beams or finite elements. The water and the foundation are generally represented by simplified models that only approximately account for their interactions with the structure. In most cases the water is modeled by an equivalent added hydrodynamic mass, and the foundation rock region is represented by a finite element system accounting for the flexibility of the foundation only. Based on these assumptions the equations of motion for the complete system become

The added hydrodynamic mass generally includes nonzero terms for x-, y- and z-DOFs, because they arise from the hydrodynamic pressures acting normal to the structure-water interface. For the structure-water interface with simple geometry, the added hydrodynamic mass terms associated with certain DOFs may be zero. For example, only the added hydrodynamic mass terms corresponding to the x-DOFs (horizontal direction) are nonzero for a gravity dam having vertical upstream face.

1.2.3.1 Concrete Gravity Dams

Conventional concrete gravity dams are constructed as monoliths (blocks) separated by transverse contraction joints. Oriented normal to the dam axis, these vertical joints extend from the foundation to the top of the dam and from the upstream face to the downstream face. For the amplitude of motion expected during strong earthquakes, the shear forces transmitted through the contraction joints are small compared with the inertia forces of the monoliths. For this condition, the monoliths in a long and straight gravity dam tend to vibrate independently, and their responses to earthquakes can be evaluated on the basis of a 2-D model. However, curved gravity dams and those built in narrow canyons need to be analyzed using a 3-D model.

1.2.3.1.1 2-D gravity dam model A 2-D model of a gravity dam for the time-history earthquake analysis consists of a monolith section supported on the flexible foundation rock and impounding a reservoir of water. The tallest monolith or dam cross section is usually selected and modeled using plane stress finite elements. The 2-D model of the selected monolith and the associated foundation rock and the impounded water may be developed as separate systems using the substructure method (Figure 2-3a), or as a complete structural system employing the standard finite element procedures (Figure 2-3b).

The viscoelastic half plane model discussed in (1) above is applicable to a homogeneous foundation where identical rock properties are assumed to exist for the entire unbounded foundation region. In general, foundation rock properties vary with depth and along the footprint of the dam. The effective modulus of the jointed rock within the shallow depths may significantly differ from that at greater depths. In these situations the viscoelastic half plane model is not appropriate and needs to be replaced by a finite element foundation model that can account for the variation of rock properties. The standard procedure is to develop a complete finite element model, which consists of the dam and an appropriate portion of the foundation region, as shown in Figure 2-3b. The foundation model, however, is assumed to be massless in order to simplify the application of the seismic input and avoid the use of large foundation models (paragraph 2-24a). The foundation mesh needs to be extended a distance at least equal to the dam height in the upstream, downstream, and downward directions. The nodal points at the base of the foundation mesh are fixed both in the vertical and horizontal directions. The side nodes, however, are attached to horizontal roller supports for the horizontal excitation and to vertical roller supports for the vertical excitation of the dam. The earthquake ground motions recorded at the ground surface are directly used as the seismic input and are applied at the base of the foundation model. The impounded water is also assumed to be incompressible so that the dam-water interaction effects can be represented by the equivalent added-mass concept. The added mass is obtained using either the simplified procedure developed by Fenves and Chopra (1986) or the generalized Westergaard method described in paragraph 2-19b.

1.2.3.1.2 3-D gravity dam model Sometimes monolith joints are keyed to interlock two adjacent blocks, or the dam is built in narrow canyons or is curved in plan to accommodate the site topography and to transfer part of the water load to the abutments. In these situations, the dam behaves as a 3-D structure and its response especially to earthquake loading should be evaluated using 3-D idealization similar to that described for arch dams in paragraph 2-13.

1.2.3.2 Arch Dams

Because concrete arch dams are 3-D structures, their responses to earthquake loading must be evaluated using a 3-D model. The 3-D model for an arch dam is developed using the finite element procedures and includes the concrete arch, the foundation rock, and the impounded water (Ghanaat 1993a, 1993b). The arch damwater-foundation system may be analyzed using the substructure method or the standard finite element procedures. Both methods use the same mathematical model to represent the concrete arch, except that the substructure method permits more rigorous analysis of the dam-foundation and the dam-water interaction effects (Tan and Chopra 1995). The standard method employs a massless foundation rock with an incompressible finite element model for the impounded water (Ghanaat 1993a, 1993b). The substructure method considers not only the foundation flexibility but also the damping and inertial effects of the foundation rock, and also includes a reservoir water model that accounts for the effects of water compressibility and the reservoir boundary absorption.

1.2.3.2.1 Dam model Concrete arch dams are usually idealized as an assemblage of finite elements, as shown in Figures 2-4 and 2-5. The finite element model of the dam should closely match the dam geometry and be suitable for application of the various loads and presentation of the stress results. To the extent possible, the finite element model of an arch dam should be developed using a regular mesh with elements being arranged on a grid of vertical and horizontal lines (Figure 2-4). This way the gravity loads can easily be applied to the individual cantilever units, and the stresses computed with respect to local axes of the element surfaces would directly relate to the familiar arch and cantilever stresses. The finite elements appropriate for modeling an arch dam include 3-D solid and shell elements available in the computer program GDAP (Ghanaat 1993a) or a general 3-D solid element with 8 to 21 nodes (Bathe and Wilson 1976). A thin or medium-thick arch dam can be modeled adequately using a single layer of shell elements through the dam thickness. A thick arch dam may require three or more layers of solid elements through the dam thickness to better represent its dynamic behavior. The level of finite element mesh refinement depends on the type of elements used. In general, a finite element mesh using the linear 8-node solid elements needs to be finer than that employing shell elements whose displacements and geometry are represented by quadratic functions.

1.2.3.2.2 Foundation model The standard foundation model for analysis of arch dams is the massless foundation discussed in paragraph 2-24a, in which only the effects of foundation flexibility are considered. Such a foundation model should extend to a distance beyond which its effects on deflections, stresses, and natural frequencies of the dam become negligible. The size of the foundation model should be determined based on the modulus ratio of the foundation to the concrete E_f/E_c . For a competent foundation rock with $E_f/E_c \geq 1$, a foundation mesh extending one dam height in the upstream, downstream, and downward directions is adequate. For a more flexible foundation rock with E_f/E_c in the range of 1/2 to 1/4, the foundation model should extend at least twice the dam height in all directions and include more elements. In general, the foundation model can be developed to match the natural topography of the foundation rock region. Such a refined model, however, is not usually required in practice. Instead, a prismatic model employed in the GDAP program (Ghanaat 1993a) and shown in Figure 2-5 may be used. The seismic input for the massless foundation model includes three-component ground acceleration time-histories applied at the fixed boundary nodes of the foundation mesh. Since no wave propagation takes place in the massless foundation model, the seismic input is obtained from the earthquake motions recorded on the ground surface using scaling or spectrum-matching procedures described in Chapter 5.

1.2.3.2.3 Reservoir water The standard dam-water interaction analysis for arch dams is based on the finite element added hydrodynamic mass model described in paragraph 2-20a (Ghanaat 1993a). Assuming the water is incompressible, the hydrodynamic pressures acting on the dam-water interface are first obtained from the finite element solution of wave equation and then converted into equivalent added-mass terms. The resulting added-mass terms are subsequently combined with the mass of concrete nodal points on the dam-water interface. In most cases a prismatic finite element fluid mesh similar to that shown in Figure 2-22 (paragraph 2-20) is adequate for computation of the added hydrodynamic mass. However, for reservoirs with irregular topography and shape, a fluid mesh that matches the actual reservoir topography is recommended (Figure 2-6). A rigorous analysis of the dam-water interaction may be required when the fundamental frequency of the reservoir water is relatively close to fundamental frequency of the dam. Such an analysis, which includes the effects of water compressibility and reservoir boundary absorption on the response of the dam, is performed as described in paragraph 2-21.

1.3 Load combinations

Concrete hydraulic structures should be designed and evaluated for three basic usual, unusual, and extreme loading combinations in accordance with EM 1110-2-2100 and the referenced guidance for specific structures. In general, the usual loading combinations are formulated based on the effects of all applicable static loads that may exist during the normal operation of the structure such as the usual concrete temperatures and the most probable water level, with dead loads, tailwater, ice, uplift, and silt. The unusual static loading combinations refer to all applicable static loads at the floodwater pool elevation with the effects of mean concrete temperatures, dead loads, and silt. For other unusual static loading combinations, refer to EM 1110-2-2100. The unusual dynamic loading combination includes the OBE loading plus any of the usual loading combinations. Extreme loading combinations consist of the effects of the MDE loading plus any of the usual loading combinations.

1. Combination with usual static loads. Time-history dynamic analysis is conducted mainly for the MDE loading conditions but also for the OBE if seismic demand is severe. At each time-step, results of such analyses should be combined with results of any of the usual loading combinations in order to obtain total displacements, stresses, and section forces needed for design or evaluation of structures.
2. Combination for multicomponent earthquake input. Modeled as two- or three-dimensional (2-D or 3-D) structural systems, time-history analysis of concrete hydraulic structures should consider two or three orthogonal

components of acceleration time-histories of earthquake ground motions. At each time-step, response quantities of interest are first computed for each component of the earthquake input and then combined algebraically to obtain the total responses due to two or all three components. Only scalar and similarly oriented response quantities are combined algebraically. After the initial algebraic combination, the resulting displacements, shear forces, and moments in orthogonal directions need to be combined vectorially if the absolute maximum values of such response quantities are required.

3. Combination for earthquake input direction (phase relation). Seismic waves of identical amplitudes, but travelling in two opposite directions, could lead to different structural response. The opposite of acceleration time-histories (i.e., all values multiplied by minus one) should also be considered as a simple way to account for some directional effects. In general, a complete permutation of all three components with positive and negative signs may be required to obtain the most critical directions that would cause the largest structural response.

1.3.1 Deconvolution

Whereas the recorded earthquake signal is on the free-field (surface), and yet the excitation must be applied at the base of the rock, we need to perform a deconvolution analysis.

In such an analysis free-field surface motions are deconvolved to determine the motions at the rigid base boundary. The deconvolution analysis is performed on a horizontally uniform layer of deformable rock using the one-dimensional wave propagation theory. The resulting rigid base motion is then applied at the base of the 3-D foundation structure system, in which the foundation model is assumed to have its normal mass as well as stiffness properties. This procedure permits the wave propagation in the foundation rock, but requires an extensive model for the foundation rock, which computationally is inefficient.

1.4 Development of structural models

Meaningful time-history analysis of probable seismic behavior of a concrete hydraulic structure for design and evaluation requires thorough understanding of the system components, their interaction, and their material properties. Modeling of the structural system and its interaction with the foundation and water are summarized in this section. The required material properties for the analysis are specified in f below. In general, structural models for the time-history analysis should be developed to capture the main dynamic characteristics of the structure and represent the effects of fluidstructure interaction and foundation-structure interaction accurately. Depending on the geometry and mass and stiffness distributions, a particular hydraulic structure may be idealized using a simple beam, a 2-D finite element, or a 3-D finite element model. The structural model should provide an accurate representation of the mass and stiffness distributions, and in the case of existing structures it should account for the effects of any existing cracks, deteriorated concrete, or any deficiency that might affect the stiffness. The fluid-structure interaction effects may be adequately represented by simple added hydrodynamic mass coefficients, or may require a finite element (or boundary element) solution with or without the effects of water compressibility and boundary absorption. Modeling of the foundation-structure effects may range from a simplified massless finite element mesh to more elaborate formulations involving soil-structure or soil-pile-structure interaction analyses. For embedded structures, the effects of dynamic backfill pressures on the structure can also be significant and should be considered.

1.4.1 Dam Models

1.4.1.1 Concrete gravity dams

Relatively long and straight concrete gravity dams built as independent monoliths separated by transverse joints may be idealized using a 2-D finite element model including the foundation rock and the impounded water. The 2-D dam-water-foundation model, usually of the tallest cross section, may be analyzed as three separate systems in the frequency domain using the substructure method (2-12a(1)) or as a single complete system in the time domain using the standard finite element procedures (2-12a(2)). The substructure method may be employed if the assumption of homogeneous material properties for the foundation region can be judged reasonable and a more rigorous formulation of the dam-water interaction including water compressibility and reservoir bottom absorption is desirable. Otherwise the standard finite element method with much simpler added-mass representation of the dam-water interaction should be used in order to account for variation of the foundation rock properties.

Curved concrete gravity dams and those built in narrow canyons should be analyzed using 3-D finite element models similar to those described for arch dams in (2) below.

1.4.1.2 Concrete arch dams

The complicated 3-D geometry of an arch dam requires a rather refined 3-D model of the dam, its foundation, and the impounded water for evaluation of its response to all three components of seismic input (2-13). The arch dam-water-foundation system may be formulated in the time domain using the standard finite element procedures or in the frequency domain using the substructure method. The standard method employs a massless foundation rock included as part of the dam finite element model in conjunction with an incompressible liquid mesh representing the impounded water. Treating each system separately, the substructure method considers the same dam model as the standard method, but employs the flexibility as well as the damping and inertial effects of the foundation rock, with a reservoir water that accounts for the effects of water compressibility and the reservoir boundary absorption. In both methods the seismic input consists of three components of the free-field acceleration time-histories applied uniformly along the dam-foundation interface in the substructure method and at the fixed boundary of the massless foundation in the standard method. The standard method provides reasonable results for small dams and those built on a competent foundation rock having a deformation modulus at least equal that of the concrete and with impounded water whose fundamental resonance frequency is at least twice that of the arch dam. Otherwise, the more rigorous formulation of the dam-water and dam-foundation interaction effects offered by the substructure method might be required.

1.4.2 Fluid Structure interaction

A hydraulic structure and water interact through hydrodynamic pressures at the structure-water interface. In the case of concrete dams, the hydrodynamic pressures are affected by the energy loss at the reservoir boundary. Generated by the motions of the structure and the ground, hydrodynamic pressures affect deformations of the structure, which in turn influence the pressures. The complete formulation of the fluidstructure interaction produces frequency-dependent hydrodynamic pressures that can be interpreted as an added force, an added mass, and an added damping (Chopra 1987). The added hydrodynamic mass influences the structure response by lengthening the period of vibration, which in turn changes the response spectrum ordinate and thus the earthquake forces. The added hydrodynamic damping arises from the radiation of pressure waves and, for dams, also from the refraction or absorption of pressure waves at the reservoir bottom. The added damping reduces the amplitude of the structure response especially at the higher modes.

1.4.2.1 Simplified Added Hydrodynamic Mass Model

If the water is assumed to be incompressible, the fluid-structure interaction for a hydraulic structure can be represented by an equivalent added mass of water. This assumption is generally valid in cases where the fluid responses are at frequencies much greater than the fundamental frequency of the structure. Following sections describe the simplified added-mass procedures including original and generalized Westergaard methods, velocity potential method for Housner's water sloshing model, and Chopras procedure for intake-outlet towers and submerged piers and shafts.

1.4.2.1.1 Westergaard added mass According to Westergaard (1933) the hydrodynamic forces exerted on a gravity dam due to earthquake ground motion are equivalent to inertia forces of a volume of water attached to the dam and moving back and forth with the dam while the rest of the reservoir water remains inactive. For analysis of gravity dams idealized as a 2-D rigid monolith with vertical upstream face, Westergaard proposed a parabolic shape for this body of water as shown in Figure 2-15. The added mass of water at location mai is therefore obtained by multiplying the mass density of water w by the volume of water tributary to point i : $i \ i \ w \ ai \ A \ z \ H \ m \) \ (8 \ 7 \ . \ = (2-6)$ where H = depth of water zi = height above the base of the dam Ai = tributary surface area at point i

1.4.2.1.2 Generalized Westergaard added mass Westergaards original added-mass concept described in a above is directly applicable to the earthquake analysis of gravity dams and other hydraulic structures having a planar vertical contact surface with the water. For structures having sloped or curved contact surfaces, a generalized formulation of the added mass should be employed. The generalized formulation assumes that the pressure is still expressed by Westergaards original parabolic shape, but the fact that the orientation of the pressure is normal to the face of the structure and its magnitude is proportional to the total normal acceleration at that point is recognized. In general, the orientation of pressures on a 3-D surface varies from point to point; and if expressed in Cartesian coordinate components, it would produce added-mass terms associated with all three orthogonal axes. Following this description the generalized Westergaard added mass at any point i on the face of a 3-D structure is expressed (Kuo 1982) by

INCOMPLETE CUT

1.4.2.2 Finite Element Added Hydrodynamic Mass Model

The simplified added hydrodynamic mass concept described in paragraph 2-19 is generally not appropriate for refined analysis of hydraulic structures having complex geometry such as arch dams and irregular intake/outlet towers. For such structures a finite element idealization of the fluid domain permits a more realistic treatment of the complicated geometry of the structure-water interface as well as the reservoir bottom. Assuming water is incompressible, inviscid, and irrotational, the small-amplitude motion of water is governed by the wave equation

(2-16)

where $p(x,y,z)$ is hydrodynamic pressure in excess of the static pressure generated by acceleration of the structure-water contact surface and acceleration of the reservoir bottom. The hydrodynamic pressures acting on the structure-water interface are obtained by solving Equation 2-16 using appropriate boundary conditions. Neglecting the effects of surface waves, which are known to be small, the boundary condition at the free surface is:

$\partial p / \partial z = 0$ (2-17)

On the structure-water contact surface, where the normal acceleration u_n (Figure 2-22) is prescribed, the boundary condition becomes:

Arch dams. For arch dams the solution of Equation 2-16 for hydrodynamic pressures is obtained numerically using the finite element method (Kuo 1982; Ghanaat 1993b), but the reservoir bottom and a truncating vertical plane at the upstream end are assumed to be rigid. This means that the ground motion g_u is not applied to the reservoir bottom (i.e., $u_{nb} = 0$ in Equation 2-19) and that the radiation damping due to propagation of pressure waves in the upstream direction is not considered. The analysis involves development of a finite element discretization of the fluid domain with the truncating upstream plane located a distance at least three times the water depth from the face of the dam. At such distance, parameter studies show that the acceleration at the truncated plane has a small effect on the hydrodynamic pressures at the face of the dam and, thus, can be assumed zero in practical applications (Clough et al. 1984a, 1984b). In most cases a prismatic fluid mesh generated by translating the dam-water interface nodes in the upstream direction is adequate for practical purposes (Figure 2-22). However, if the actual reservoir topography is substantially different from a prismatic model, a fluid mesh that closely matches the reservoir topography may be required. In either case, the distance between the successive surfaces or planes arranged approximately parallel to the dam axis should be such that the fluid layers closer to the dam face contain finer elements. The finite element solution of Equation 2-16 results in nodal pressures on the upstream face of the dam, which after conversion into nodal forces gives the added hydrodynamic mass matrix for earthquake analysis of the dam. The resulting addedmass matrix is a full square matrix with a dimension equal to the number of degrees of freedom on the damwater interface nodes.

1.4.2.3 Compressible Water with Absorptive Boundary Model

a. The added-mass representation of hydrodynamic pressure previously described ignores the effects of water compressibility and water-foundation interaction. Refined dam-water interaction analysis including these factors (Hall and Chopra 1980; Fenves and Chopra 1984b; Fok and Chopra 1985) indicates that water compressibility and the water-foundation interaction can significantly affect the hydrodynamic pressures and hence the response of concrete dams to earthquakes. The effects of water compressibility are generally significant when the fundamental frequency of the dam without the water is relatively close to the fundamental resonant frequency of the impounded water, $\omega_r = \sqrt{C/H}$, where C is the velocity of sound in water and H is the water depth. The water compressibility and the water-foundation interaction effects can be considered by solving the wave equation for compressible water

$\partial^2 p / \partial z^2 + \omega_r^2 p = 0$ (2-20)

subjected to the boundary conditions given in Equations 2-17 to 2-19. The water-foundation interaction, as indicated by Equation 2-19, can be considered by using finite elements to represent the flexible foundation or modeling the foundation material as a viscoelastic half space. This effect has also been accounted for in an approximate manner by using a simplified boundary condition that models the energy dissipated at the waterfoundation interface, as described in paragraph 2-22. The most extensive study of concrete dams with compressible reservoir water has been carried out by Chopra and his co-workers (Hall and Chopra 1980; Fenves and Chopra 1984b; Fok and Chopra 1985) using the substructure method of analysis. Assuming the reservoir water can be idealized as a fluid domain with constant depth and infinite length in the upstream direction, the hydrodynamic pressures for 2-D analysis of gravity dams is obtained from a continuum solution (Fenves and Chopra 1984b). For irregular reservoir boundaries, the fluid domain is usually assumed to consist of an irregular portion adjacent to the dam and a uniform section of infinite length in the upstream direction (Figure 2-25). The irregular portion is represented by a finite element discretization (Hall and Chopra 1980) or boundary element method (Humar and Jablonski 1988), whereas the uniform portion is analyzed by a continuum solution. The equal pressure conditions at the interface then enforce the coupling between the two regions. Such formulation of the hydrodynamic pressure results in frequency-dependent hydrodynamic terms that are best treated in the frequency domain. This procedure has been implemented in the computer program EACD-3D (Fok, Hall, and Chopra 1986) for the earthquake analysis of arch dams. b. The hydrodynamic pressure in the reservoir, as given by Equation 2-20, is generated by the acceleration of the upstream face of the dam and vertical

accelerations of the reservoir bottom. The solution in frequency domain produces the frequency response functions for the hydrodynamic pressures in the impounded water. The computed pressure frequency response functions at the face of the dam and at the reservoir bottom are then converted into statically equivalent nodal forces (\mathbf{R}_h) and (\mathbf{Q}_h) and are substituted into the system equations of motion (Equation 2-1).

1.4.2.4 Reservoir Boundary Absorption

a. The energy loss capability of the reservoir bottom materials is approximately modeled by a boundary that partially absorbs (refracts) the incident pressure waves (Hall and Chopra 1980). In the boundary condition for the reservoir bottom, this energy loss is represented by the damping coefficient q , which is related to the wave reflection coefficient by

$$(2-21)$$

where ρ and C_s are the density and sound velocity for water, respectively, and ρ_s and $C_{s,b}$ are the density and sound velocity for the bottom materials, respectively. The reflection coefficient provides a measure of the level of absorption of the reservoir bottom materials. It is defined as the ratio of the amplitude of the reflected pressure wave to the amplitude of incident pressure wave impinging on the reservoir bottom. The values of α vary between 1 and -1 where $\alpha = 1$ represents a nonabsorptive rigid boundary with 100 percent reflection, $\alpha = 0$ corresponds to a complete absorption with no reflection, and $\alpha = -1$ characterizes 100 percent reflection from a free surface with an attendant phase reversal (water surface). Recent field investigations have indicated that the average values of α for the reservoir bottom materials measured at several concrete damsites varied over a range from -0.55 to 0.66 (Ghanaat and Redpath 1995). Three of the measured values were negative and the largest (-0.66) was much less than 1. the value corresponding to a rigid boundary. The results also showed that some sites had thick layers of soft and muddy sediments with propagation velocities less than that of water, thus leading to negative values of α , a situation never before considered analytically.

b. All hydrodynamic pressure terms (i.e., added mass, added damping, and added force) are affected by the reservoir bottom absorption. Previous studies (Hall and Chopra 1980; Fenves and Chopra 1984b; Fok and Chopra 1985) indicate that the reservoir bottom absorption increases the effective damping, hence reduces the dam response to earthquake loading. The reduction of dam response due to reservoir bottom absorption, however, is much larger for the response to vertical ground motion than to horizontal. Considering that the dam responses due to the vertical and horizontal components of the ground motion are not usually in phase, the effect of reservoir bottom absorption on the total response of the dam is less than that for the vertical ground motion.

1.4.3 Foundation Structure Interaction

Foundation-structure interaction introduces flexibility at the base of the structure and provides additional damping mechanisms through material damping and radiation. The interaction with the flexible foundation affects the earthquake response of the structure by lengthening the period of vibration and increasing the effective damping of the system. The increase in the damping arises from the energy radiation and material damping in the foundation region. However, interaction with the flexible foundation also tends to reduce the structural damping that the structure would have had in the case of a rigid foundation (Novak and El Hifnawy 1983). For lightly damped hydraulic structures (less than 10 percent damping) the reduction in structural damping is usually more than compensated for by the added damping of the flexible foundation. Such interaction effects introduce frequency-dependent interacting forces at the structure-foundation interface, which are represented by the dynamic stiffness (or impedance) matrix for the foundation rock region, as described previously.

1.4.3.1 Massless finite element foundation model

The effects of dam-foundation interaction can most simply be represented by including, in the finite element idealization, foundation rock or soil region above a rigid horizontal boundary. The response to the earthquake excitation applied at the rigid base (bedrock) is then computed by the standard procedures. Such an approach, however, can lead to enormous foundation models where similar materials extend to large depths and there is no obvious "rigid" boundary to select as a fixed base. Although the size of foundation model can be reduced by employing viscous or transmitting boundaries to absorb the wave energy radiating away from the dam (Lysmer and Kuhlemeyer 1969), such viscous boundaries are not standard features of the general-purpose structural analysis programs.

These difficulties can be overcome by employing a simplified massless foundation model, in which only the flexibility of the foundation rock is considered while its inertia and damping effects are neglected. The size of a massless foundation model need not be very large so long as it provides a reasonable estimate of the flexibility of the foundation rock region. A foundation model that extends one dam height in the upstream, downstream, and downward directions usually suffices in most cases. Unlike the homogeneous viscoelastic half plane model described previously, this approach permits different rock properties to be assigned to different elements, so that the variation of rock characteristics with depth can be considered. The massless foundation model also permits the earthquake motions

recorded on the ground surface to be applied directly at the fixed boundaries of the foundation model. This is because in the absence of wave propagation, the motions of the fixed boundaries are transmitted to the base of the dam without any changes.

1.4.3.2 Viscoelastic foundation rock model

The stiffness and damping characteristics of foundation-structure interaction in a viscoelastic halfplane (2-D) or half space (3-D) model are described by the impedance function. Mathematically, an impedance function is a matrix that relates the forces (i.e., shear, thrust, and moment) at the base of the structure to the displacements and rotations of the foundation relative to the free field. The terms in an impedance function are complex and frequency dependent with the real component representing the stiffness and inertia of the foundation and the imaginary component characterizing the radiation and material damping.

Viscoelastic half plane model. For sites where essentially similar rocks extend to large depths, the foundation rock for 2D analyses may be idealized as a viscoelastic half plane. In other situations where soft or fractured rock overlies harder rock at shallow depth, a finite element idealization (a above) that permits for material nonhomogeneity and structural embedment would be more appropriate. In viscoelastic half plane idealization, foundation-structure interaction is represented by a complex valued impedance or dynamic stiffness matrix ($S_f(\omega)$) in Equation 2-2). Assuming that the structure is supported on a horizontal ground surface with homogeneous material properties, the dynamic stiffness matrix $S_f(\omega)$ is evaluated using the approach by Dasgupta and Chopra (1979) or other approaches that use boundary element and Green's functions to analyze the problem (Wolf and Darbre 1984; Alarcon, Dominguez, and Del Cano 1980).

Viscoelastic half space model. The foundation rock for 3-D analyses of concrete hydraulic structures supported on essentially similar rocks with homogeneous material properties may be represented by viscoelastic half space. Employed in the substructure method of analysis, the half space model leads to an impedance matrix for the foundation rock region defined at the structure-foundation interface. A variety of boundary element methods using different Green's functions, finite element techniques in frequency domain using transmitting boundaries, finite element method in time domain, infinite elements, and hybrid methods are available to compute impedance matrices for surface and embedded foundations. Without certain simplifying assumptions, these techniques are computationally demanding and are usually unsuitable for practical applications. One such assumption applied to the analysis of arch dams is to assume that the dam is supported in an infinitely long canyon of arbitrary but uniform cross section and thus break down the problem into a series of two-dimensional boundary problems (Zhang and Chopra 1991). In situations where soft or fractured rock overlies harder rock at shallow depth, a finite element idealization accounting for the material nonhomogeneity should be used.

1.5 Material properties

Concrete hydraulic structures are built using both plain and lightly reinforced forms of concrete construction and may be supported by rock, soil, or pile foundations. Concrete condition, function, age, and properties for existing structures and concrete mix and properties for new designs usually vary widely from structure to structure. These factors and geotechnical information of the subsurface conditions have potentially significant influence on the seismic performance of concrete hydraulic structures. It is essential that the time-history seismic evaluation effort conform to guidelines for determination of material properties and assessment of physical condition described in other references. The primary material properties relevant to time-history dynamic analysis are summarized in the following paragraphs.

1.5.1 Concrete properties

The primary material properties of interest in a concrete structure are those that affect prediction of the structural response and those that are required for evaluation of the structural performance. The structural response is predicted on the basis of unit weight and elastic properties of the concrete including modulus of elasticity and Poisson's ratio. Many laboratory and field measurements have shown that modulus of elasticity is affected by the rate of loading and generally is higher for the dynamic than it is for the static loading. Under the sustained static loading conditions, the effects of creep on the mass concrete may be important and generally can be considered by determining a sustained modulus of elasticity taken as 60 to 70 percent of the laboratory value of the instantaneous modulus of elasticity. For seismic analyses the measured or estimated dynamic modulus is more appropriate and should be used. In the absence of measured data, dynamic modulus of elasticity should be obtained by increasing the laboratory value of the instantaneous modulus by 20 to 30 percent. Compressive and tensile strengths of concrete are properties used to evaluate acceptability of new designs or seismic performance of the existing structures. Like modulus of elasticity, concrete strength parameters are also affected by the rate of loading. Seismic design and performance evaluation of concrete hydraulic structures should therefore be based on the measured or estimated dynamic strength of concrete. Other material properties such as shear strength of concrete, tensile and shear strengths of construction joints, yield

strength and modulus of elasticity of reinforcing steel, and reinforcing steel bond strength and ductility may also be required. In general tensile strength across the deteriorated or poorly constructed joints could significantly be lower than that of the parent concrete. Determination of tensile and shear strengths across such joints may be warranted under severe earthquake loading.

1.5.2 Foundation rock properties

Foundation rock properties for use in structural analyses include shear strength and rock mass modulus of deformation. Procedures for estimating shear strength and modulus of deformation are described in Chapter 10 of EM 1110-2-2201. Shear strength parameters provide a measure of shearing resistance to sliding at the structure-rock interface or within the foundation and abutments, when potential sliding wedges or planes of rock that could cause instability have been identified. The modulus of deformation is a measure of foundation deformations for the rock mass as a whole including the effects of its discontinuities. In contrast, modulus of elasticity is determined for an intact specimen of the rock.

1.5.3 Reservoir bottom absorption

Studies of the dam-water interaction indicate that the earthquake response of concrete dams is sensitive to the water energy loss at the reservoir boundaries. If the reservoir boundary materials are relatively soft, an important fraction of the reservoir water energy can be absorbed, leading to a major reduction in the dynamic response of the dam. An earthquake-generated hydrodynamic pressure wave impinging on the reservoir boundary is partly reflected in the water, and partly refracted (absorbed) into the boundary materials. The energy loss or partial absorption at the reservoir boundary is approximately represented by a reflection coefficient ρ , which is the ratio of reflected to incident wave amplitudes (Hall and Chopra, 1980; Fenves and Chopra 1984b). The reflection coefficient varies between 1 and -1, where $\rho = 1$ corresponds to a total reflection (nonabsorptive or rigid boundary), $\rho = 0$ represents a complete absorption or transmission into the boundary materials, and $\rho = -1$ characterizes 100 percent reflection from a boundary with an attendant phase reversal. The in situ values of ρ for the seismic safety evaluation of concrete dams can be measured using three independent approaches developed and employed at several dams in the United States and abroad. These include the seismic reflection and refraction techniques (Ghanaat and Redpath 1995) and a technique based on the acoustic reverberation (Ghanaat et al. 1999).

1.5.4 Damping

In practice, damping characteristics of typical structures are generally expressed in terms of equivalent viscous damping ratios. The velocity-proportional viscous damping is commonly used because it leads to convenient forms of equations of motion. The energy-loss mechanism for the viscous damping, however, depends on the frequency of excitation, a phenomenon that has not been observed experimentally. As a result it is desirable to remove this frequency dependency by using the so-called hysteretic form of damping. The hysteretic damping is defined as a damping force proportional to the strain or deflection amplitudes but in phase with the velocity. The structural response provided by hysteretic damping can be made identical to that with viscous damping if the hysteretic damping factor is selected as

$\rho = 2(1-\zeta)$ where ζ = hysteretic damping factor $\zeta = \text{viscous damping ratio}$ $\zeta = \text{ratio of the excitation frequency to the natural free-vibration frequency}$ To remove the frequency dependency term from Equation 1-1, the value of hysteretic damping ρ is computed at resonance by setting $\zeta = 1$. The hysteretic damping computed in this manner provides identical response to that of the viscous damping at the resonance and nearly identical response at all other frequencies for $\zeta \geq 0.2$.

Viscous damping is commonly used in the time-domain solution, whereas the hysteretic damping factor taken as twice the viscous damping ratio is usually employed in the frequency domain solution. Linear time-history analysis of concrete hydraulic structures should employ a damping equivalent to a 5 percent viscous damping ratio. However, in situations where a moderate level of nonlinear behavior such as joint opening and cracking is predicted by a linear analysis, a higher damping ratio in the range of 7 to 10 percent could be used to account somewhat for the energy loss due to nonlinear behavior.

1.6 Numerical analysis procedures

Computation of earthquake response history for typical concrete hydraulic structures involves solution of coupled sets of equations of motion that include large numbers of equations or degrees of freedom. In linear response analyses the system equations of motion can be formulated either in the time domain or in the frequency domain. Only time domain formulation is suited to analysis of nonlinear response. These formulations and the corresponding response analysis procedures are described in Chapters 2 and 3, respectively. Following is a brief summary to provide a general

idea of how these techniques are applied in the solution of the earthquake response behavior of concrete hydraulic structures.

1.6.1 Analysis in the time domain

In practice, time-domain response analyses are generally based on some forms of step-by-step methods using numerical integration procedures to satisfy the equations of motion. In all the step-by-step methods the loading and the response history are divided into a sequence of time intervals or steps. The response during each step is computed from the initial conditions (displacement and velocity) at the beginning of the step and from history of loading during the step. The structural properties within each step are assumed to remain constant, but could vary from one step to another (nonlinear behavior) or remain the same during all time-steps (linear behavior). The step-by-step methods may be classified as explicit or implicit. In an explicit method, the new response values calculated in each step depend only on the response quantities available at the beginning of the step. The analysis therefore proceeds directly from one step to the next. In an implicit method, on the other hand, the new response values for a given step include one or more values pertaining to the same step, so that trial values and successive iterations are necessary. The iteration within a step makes implicit formulations inconvenient and in some cases even prohibitive. Only explicit methods such as those described in Chapter 3 may be considered. The primary factors to be considered in selecting a step-by-step method include efficiency, round-off and truncation errors, instability, phase shift or apparent change of frequency, and artificial damping in accordance with Chapter 3.

(a) Mode superposition method. In linear response analysis, the mode superposition techniques can be used to uncouple the system equations of motion, so that the dynamic response can be obtained separately for each mode of vibration and then superimposed for all significant modes to obtain the total response. This way the step-by-step integration discussed in (1) above is applied separately to a number of independent SDOF equations and then the resulting modal response histories are superimposed to compute the total response of the structure. The main effort in this method includes computation of eigenvalue problems followed by modal coordinate transformation to uncouple the MDOF dynamic analysis to the solution of a series of SDOF systems. It is important to note that the equations of motion will be uncoupled only if the damping can be represented by a mass proportional and stiffness proportional damping matrix known as Rayleigh damping. The Rayleigh damping is suitable when the damping mechanism is distributed rather uniformly throughout the structure.

(b) Direct step-by-step method. In this method, the step-by-step integration is applied directly to the original equations of motion with no need for modal coordinate transformation to uncouple them. Thus there is no need to obtain natural mode shapes and frequencies or to limit damping to the proportional type. The method can be used for both the linear and nonlinear response analyses.

1.6.2 Analysis in the frequency domain

An alternative approach to solving the modal equations of motion for linear systems is to perform the analysis in the frequency domain. In particular, when the equation of motion contains frequency-dependent parameters such as foundation stiffness and damping, the frequency domain approach is much superior to the time domain approach. In simple terms the frequency domain solution involves expressing the ground motion in terms of its harmonic components; evaluating the response of the structure to each harmonic component; and superposing the harmonic responses to obtain total structural response. In this process, the harmonic amplitudes of the ground motion in the first step and superposition of harmonic responses in the third step are obtained using the Fast Fourier Transform (FFT) algorithm.

1.7 Structural performance and damage criteria

Chapter 4 describes methodologies and procedures for evaluation of earthquake performance and qualitative estimation of the probable level of damage using the results from linear time-history analyses. The overall process involves describing the results in terms of the demand-capacity ratios, cumulative inelastic duration of excessive stresses or forces, and spatial extent and distribution of high-stress or high-force regions, and then comparing them with a set of acceptance criteria set forth for each type of structure. Another important factor in the evaluation process is consideration of probable nonlinear mechanisms and modes of failure that might develop in a concrete hydraulic structure. The damage in a particular structure is considered to be minor and the linear time-history analysis will suffice if estimated level of damage meets the acceptance requirements established for that structure. Otherwise the damage is considered to be severe, in which case a nonlinear time-history analysis would be required to estimate damage more accurately.

1.7.1 Gravity dams

The dam response to the MDE is considered to be within the linear elastic range of behavior if the computed stress demand-capacity ratios are less than or equal to 1.0. The level of nonlinear response or cracking is considered acceptable if demand-capacity ratios are less than 2, overstressed regions are limited to 15 percent of the dam surface area, and the cumulative duration of stress excursions beyond the tensile strength of the concrete falls below the performance curve shown in Figure 4-2.

1.7.2 Arch dams

The dam response to the MDE is considered to be nearly within the linear elastic range if the computed stress demand-capacity ratios are less than or equal to 1.0. The dam is considered to exhibit nonlinear response in the form of opening and closing of contraction joints and cracking of lift lines if the estimate demand-capacity ratios exceed 1.0. The amount of joint opening and cracking is considered acceptable if demand-capacity ratios are less than 2, overstressed regions are limited to 20 percent of the dam surface area, and the cumulative inelastic duration falls below the performance curve given in Figure 4-18.

Chapter 2

CONSTITUTIVE MODEL FOR ALKALI AGGREGATE REACTIONS

This chapter will address the important problem of AAR in dams and how to model this phenomenon in a nonlinear analysis. We will not address the multitude of tests which can be performed to assess the xxxxx

2.1 INTRODUCTION

As massive concrete structures are ageing, some of them are showing troublesome signs of structural degradation preceded by excessive wrinkles in the form of random cracks. This internal degradation of concrete with time is most often attributed to Alkali Aggregate Reactions (AAR) which is the cause of internal cracks similar to bone osteoporosis. Hence, it is not surprising that only recently has this problem acquired major importance, as structures built over twenty years ago were not properly screened to avoid the fatal combination of reactive aggregates with cement.

AAR problems have been reported all over the world, and no country seems to be immune from this disease (though structures in relatively colder climates have been slower to develop this reaction). Hence, much research has been undertaken recently, and there is an explosion of publications related to AAR. Broadly speaking they can be categorized in three types: a) Chemical reaction description; b) Effect on mechanical properties; and c) Symptoms or effects on structures. What is missing is remedy. Unfortunately, there is not yet known proven remedy to this slow evolving and irrevocable process, other than addressing the symptoms through cutting the structures to relieve the build-up of compressive stresses. On the comforting side, there is strong indication that the reaction does not proceed indefinitely, and that at some point all the reactive aggregates would have been consumed. Hence, structural monitoring, and future expansion revision are of paramount importance. This is crucial, as the internal stress redistribution caused by the AAR (and possibly the cutting) may in turn cause major structural cracks which could jeopardize the structural integrity.

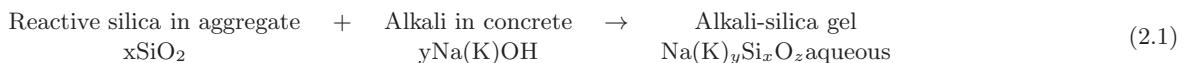
Dams, by their very size, the role they play in modern society, and the damage which can be caused by even a partial failure, are to be particularly monitored for AAR.

This report presents a literature survey on AAR. It is certainly not an exhaustive one, yet it attempts to present the major findings that an Engineer confronted with AAR should be concerned with. To some extent, it reflects the background, biases and opinions of its authors, yet it could constitute a first reading which could lead to other more detailed work.

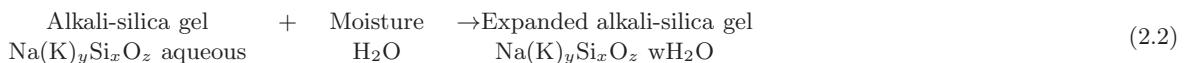
2.2 Chemical Reactions

Alkali Aggregate Reaction (AAR), which includes Alkali Silica Reaction (ASR) is the leading cause of dam concrete deterioration. This slow evolving internal concrete damage is causing millions of dollars in damage worldwide, and whereas there is no (economically) feasible method to stop the reaction, it can to some extent be mitigated. This has been accomplished primarily through an expensive slicing of the dam to relieve the reaction induced compressive stresses. Hence, given the need to plan this complex mitigation procedure, and keeping in mind that in some drastic cases the dam may have to be decommissioned, there is an urgent need to provide the Engineering profession with solid, sound and practical predictive tools for the dam structural response evolution.

Alkali-silica reaction (ASR) in concrete is a chemical reaction involving alkali cations and hydroxyl ions from concrete pore solutions, and certain metastable or strained forms of silica present within aggregate particles. This chemical reaction will produce ASR gel which swells with the absorption of moisture. Hence, in a simplified manner, ASR can be described as a two-step reaction between alkalis (sodium and potassium) in concrete and silica reactive aggregates. The first step is the chemical reaction between the reactive silica in the aggregate with the alkali present in concrete to produce alkali-silica gel:



The second step is the expansion of the alkali-silica gel when it comes in contact with moisture:



It is precisely this second reaction which causes the well known swelling of the concrete resulting in a major internal stress redistribution inside the dam which manifests itself either through large compressive stresses, and/or more dramatically through the formation of structural cracks or the sliding across critical joints. Hence the structural integrity of the structure can certainly be seriously jeopardized by the pernicious and slow evolution of the reaction.

2.3 LITERATURE SURVEY

AAR was first identified by Stanton (1940) as a cause for concrete deterioration. Whereas there were few initial related papers, and probably triggered by an ever increasing manifestation of the reaction in major structures, there has been recently numerous investigations on AAR. In the context of the presented work, only few related work will be examined. More information can be found in (Saouma and Xi 2004).

One of the most extensive and rigorous investigation of AAR has been conducted by Larive (1998) who tested more than 600 specimens with various mixes, ambiental and mechanical conditions. Not only did the author conduct this extensive experimental investigation, but a numerical model has also been proposed for the time expansion of the concrete. In particular, a thermodynamical based model for the expansion evolution is developed, and then calibrated with the experimental data, Fig. 2.1.

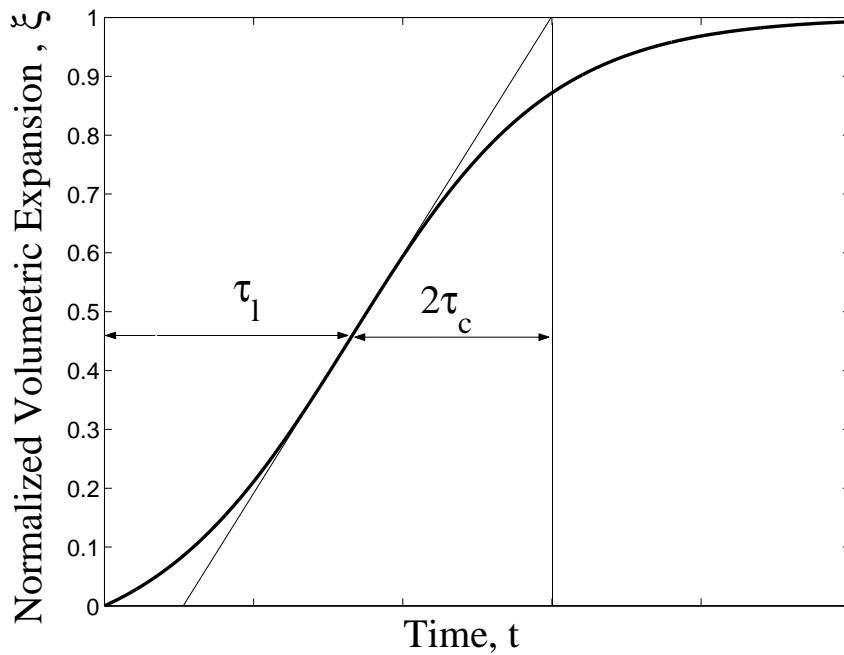


Figure 2.1: Normalized Expansion Curve ($\xi(t) = \varepsilon_{F,Vol}^{AAR}(t)/\varepsilon_{AAR}^\infty$)

$$\xi(t, \theta) = \frac{1 - e^{-\frac{t}{\tau_c(\theta)}}}{1 + e^{-\frac{(t-\tau_l(\theta))}{\tau_c(\theta)}}} \quad (2.3)$$

where τ_l and τ_c are the latency and characteristic times respectively. The first corresponds to the inflexion point, and the second is defined in terms of the intersection of the tangent at τ_L with the asymptotic unit value of ξ . In a subsequent work, Ulm, Coussy, Kefei and Larive (2000) have shown the thermal dependency of those two coefficients:

$$\begin{aligned} \tau_l(\theta) &= \tau_l(\theta_0) \exp \left[U_l \left(\frac{1}{\theta} - \frac{1}{\theta_0} \right) \right] \\ \tau_c(\theta) &= \tau_c(\theta_0) \exp \left[U_c \left(\frac{1}{\theta} - \frac{1}{\theta_0} \right) \right] \end{aligned} \quad (2.4)$$

expressed in terms of the absolute temperature ($\theta^oK = 273 + T^oC$) and the corresponding activation energies. U_l and U_c are the activation energies minimum energy required to trigger the reaction for the latency and characteristic times respectively, and were determined (for Larive's test) to be

$$U_l = 9,400 \pm 500K \quad (2.5)$$

$$U_c = 5,400 \pm 500K \quad (2.6)$$

To the best of the authors knowledge, the only other tests for these values were performed by Scrivener (2005) who obtained values within 20% of Larive's, and dependency on types of aggregates and alkali content of the cement has not been investigated. Hence, in the absence of other tests, those values can be reasonably considered as representative of dam concrete also. The temperature dependance is highlighted by Fig. 2.2 where the expansion curve determined in the laboratory at 38°C is compared with the corresponding one at a dam average temperature of 7°C

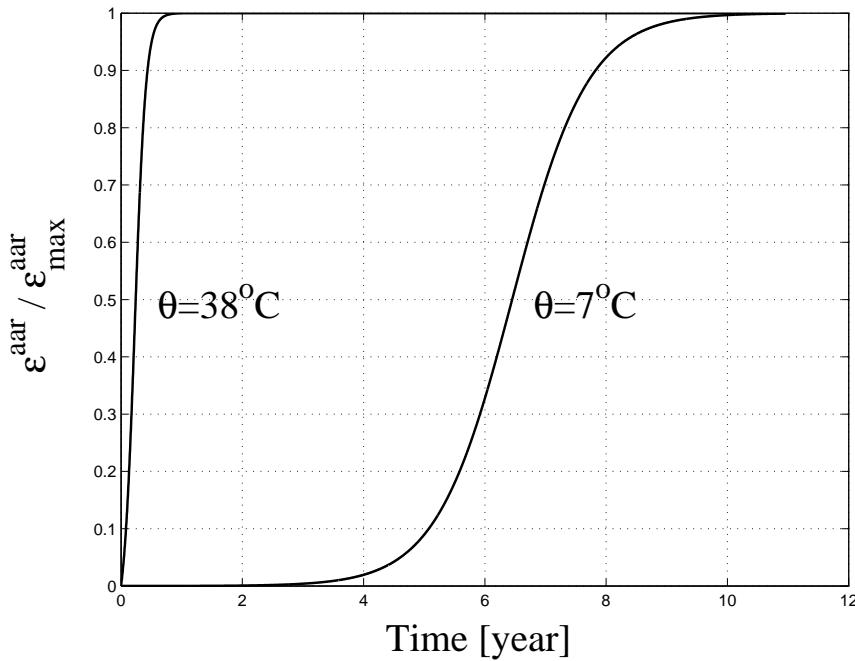


Figure 2.2: Effect of Temperature on AAR Expansion

Beside temperature, other parameters strongly affecting AAR expansion are humidity and confining stresses.

Most recently, Multon (2004) tested AAR expansion under triaxial constraint. Axial traction was applied along one direction of concrete cylinders constrained in the radial directions by steel cylinders. As reported first by Larive (1998) (uniaxial confinement) and later confirmed by Multon, Leclainche, Bourdarot and Toutlemonde (2004) (for triaxial confinement), there is strong evidence of an expansion transfer such that the total volumetric AAR induced strain is almost constant irrespective of the confinement. In other word, the expansion is largest in the direction of "least resistance". In uniaxially or biaxially loaded cylinders, this results in substantially reduced expansion in the loaded directions, and increased expansion in the unconstrained ones. On the other hand, under compressive triaxial confinement, there is nearly equal expansion in all three directions, however the total volumetric expansion is slightly reduced. Final, there are strong indications that high compressive hydrostatic stresses retard the reaction.

Accompanying AAR expansion, there is often a degradation in tensile strength and elastic modulus, (Swamy and Al-Asali 1988). However, one should exercise some caution as the degradation observed in laboratory specimens is often much higher than the one recorded in the field.

Whereas a good model for AAR should start with the gel induced pressure, this is a notoriously complex problem (due to scale), and in that context the work of Struble and Diamond (1981a) and Struble and Diamond (1981b) remains most pertinent.

Modelling of AAR expansion has been undertaken by various researchers. Broadly speaking, this modelling falls into one of three categories:

Micro Models: in which aggregate and cement paste are separately modelled and the transport equation is used to model gel formation through a two stage process, (Suwito, Jin, Xi and Meyer 2002) and (Lemarchand and Dormieux 2000). While essential to properly understand the underlying phenomenon causing AAR, this level of modelling, is of little relevance to structural analysis of AAR affected structures as emphasis is on the transport equation for the reactants.

Meso Models: Where emphasis is on the determination of pessimum size effect, (Furusawa, Ohga and Uomoto 1994) and (Bažant, Z.P and Steffens, A. 2000).

Macro Models: Where one stay clear from the transport modelling, and emphasis is on a global numerical model for the analysis of a structure. Some of the models fully decouple structural modelling from the reaction kinetics, others couple those two effects (and some ignore all together the kinetics).

One of the earliest model is the one of Charlwood, Solymar and Curtis (1992) and Thompson, Charlwood, Steele and Curtis (1994) who identified critical issues related to AAR, namely the stress dependency, that is there is no AAR expansion under a compressive stress of around 8 MPa, and that the expansion is akin of a thermal one. Subsequently more refined models have been proposed by Léger, Côte and Tinawi (1996) and Huang and Pietruszczak (1999) which focus on the kinetics of the reaction, albeit through empirical models. Models which address both the kinetics and the mechanical model of AAR have been proposed by Bournazel and Moranville (1997), Capra and Bournazel (1998), Capra and Sellier (2003), Ulm et al. (2000) and Li and Coussy (2002). It is worth noting that the kinetics model (built into a coupled thermo-chemo-mechanical one) of Ulm et al. (2000) (based on the work of Larive (1998)) departs from other empirical models and is probably the most scientifically correct one. It is the one adopted in this work. Bangert, D. and Meschken (2004) recently proposed a coupled model applied to reinforced concrete, and finally, Farage, Alves and Fairbairn (2004) seems to have finally bridged the gap between scientific rigor and practical applicability to real structures.

Numerous dams worldwide have suffered from AAR, in particular as reported by, Wagner and Newell (1995) (Fontana dam, United States), Gilks and Curtis (2003) (Mactaquac dam, Canada), Shayan, Wark and Moulds (2000) (Canning dam, Australia), (Peyras, Royet and Laleu 2003) (Chambon dam, France), Jabarooti and Golabtoonchi (2003) (Iran), Bon, Chille, Masarati and Massaro (2001) (Pian Telessio dam, Italy), Portugese National Committee on Large Dams (2003) (Pracana dam, Portugal), Malla and Wieland (1999) (a Swiss dam). A comprehensive list of dams suffering from AAR can be found in (Acres 2004).

It is worth noting that dams built in general, dams built in (relatively) hot climate appear to suffer from AAR at an earlier age than those built in high altitudes and colder temperatures. Furthermore, when dam rehabilitation did occur it included one or more of the following: cutting (to relieve the compressive stresses, though accelerating the expansion rate), post-tensioning, or placing an impermeable membrane (which benefits are not yet well proven).

2.4 MODEL

Two different aspects of mathematical modelling of ASR in concrete may be distinguished: 1) The kinetics of the chemical reactions and diffusion processes involved, and 2) The mechanics of fracture that affects volume expansion and causes loss of strength, with possible disintegration of the material, (Bažant, Z.P. and Zi, G. and Meyer, C. 2000).

The proposed model, (Saouma and Perotti 2004c) is driven by the following considerations:

1. AAR is a volumetric expansion, and as such can not be addressed individually along a principal direction without due regard to what may occur along the other two orthogonal ones.
2. Kinetics component is taken from the work of Larive (1998) and Ulm et al. (2000).
3. AAR is sufficiently influenced by temperature to account its temporal variation in an analysis.
4. AAR expansion is constrained by compression, and is redirected in other less constrained principal directions. This will be accomplished by assigning "weights" to each of the three principal directions.
5. Relatively high compressive or tensile stresses inhibit AAR expansion due to the formation of micro or macro cracks which absorb the expanding gel.
6. High compressive hydrostatic stresses slow down the reaction.
7. Triaxial compressive state of stress reduces but does not eliminate expansion.
8. Accompanying AAR expansion is a reduction in tensile strength and elastic modulus.

Hence, the general (uncoupled) equation for the incremental free volumetric AAR strain is given by

$$\dot{\varepsilon}_V^{AAR}(t) = \Gamma_t(f'_t|w_c, \sigma_I|COD_{max})\Gamma_c(\bar{\sigma}, f'_c)g(h)\xi(t, \theta) \varepsilon^\infty|_{\theta=\theta_0} \quad (2.7)$$

where COD is the crack opening displacement, $\xi(t, \theta)$ is a sigmoid curve expressing the volumetric expansion in time as a function of temperature and is given by Eq. 2.3, ε^∞ is the laboratory determined (or predicted) maximum free volumetric expansion at the reference temperature θ_0 , Fig. 2.1.

The retardation effect of the hydrostatic compressive stress manifests itself through τ_l . Hence, Eq. 2.4 is expanded as follows

$$\tau_l(\theta, \theta_0, I_\sigma, f'_c) = f(I_\sigma, f'_c)\tau_l(\theta_0) \exp \left[U_l \left(\frac{1}{\theta} - \frac{1}{\theta_0} \right) \right] \quad (2.8)$$

where

$$f(I_\sigma, f'_c) = \begin{cases} 1 & \text{if } I_\sigma \geq 0 \\ 1 + \alpha \frac{I_\sigma}{3f'_c} & \text{if } I_\sigma < 0. \end{cases} \quad (2.9)$$

and I_σ is the first invariant of the stress tensor, and f'_c the compressive strength. Based on a careful analysis of Multon (2004), it was determined that $\alpha = 4/3$. It should be noted, that the stress dependency (through I_σ) of the kinetic parameter τ_t makes the model a truly coupled one between the chemical and mechanical phases. Coupling with the thermal component, is a loose one (hence a thermal analysis can be separately run),

$0 < g(h) \leq 1$ is a reduction function to account for humidity given by

$$g(h) = h^m \quad (2.10)$$

where h is the relative humidity, (Capra and Bournazel 1998). However, one can reasonably assume that (contrarily to bridges) inside a dam $g(h) = 1$ for all temperatures.

$\Gamma_t(f'_t|w_c, \sigma_I|COD_{max})$ accounts for AAR reduction due to tensile cracking (in which case gel is absorbed by macro-cracks), Fig. 2.3. A hyperbolic decay, with a non-zero residual value is adopted, Fig. 2.4:

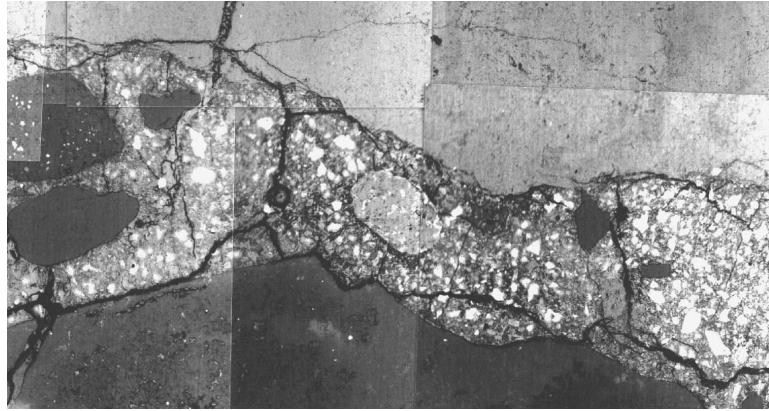


Figure 2.3: Stress Induced Cracks with Potential Gel Absorption, (Scrivener 2003)

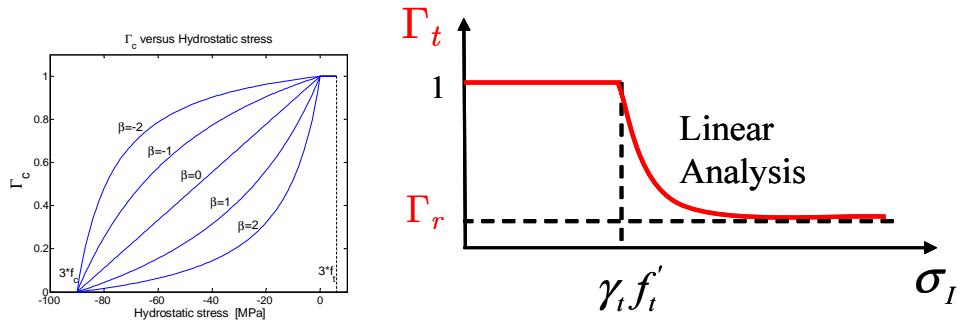


Figure 2.4: Graphical Representation of Γ_c and Γ_t

$$\text{Smeared Crack} \left\{ \begin{array}{l} \text{No } \Gamma_t = \begin{cases} 1 & \text{if } \sigma_I \leq \gamma_t f'_t \\ \Gamma_r + (1 - \Gamma_r) \gamma_t \frac{f'_t}{\sigma_I} & \text{if } \gamma_t f'_t < \sigma_I \end{cases} \\ \text{Yes } \Gamma_t = \begin{cases} 1 & \text{if } COD_{max} \leq \gamma_t w_c \\ \Gamma_r + (1 - \Gamma_r) \gamma_t \frac{w_c}{COD_{max}} & \text{if } \gamma_t w_c < COD_{max} \end{cases} \end{array} \right. \quad (2.11)$$

where γ_t is the fraction of the tensile strength beyond which gel is absorbed by the crack, Γ_r is a residual AAR retention factor for AAR under tension. If an elastic model is used, then f'_t is the tensile strength, σ_I the

maximum principal tensile stress. On the other hand, if a smeared crack model is adopted, then COD_{max} is the maximum crack opening displacement at the current Gauss point, and w_c the maximum crack opening displacement in the tensile softening curve, (Wittmann, Rokugo, Brühwiler, E., Mihashi and Simonin 1988). Concrete pores being seldom interconnected, and the gel viscosity relatively high, gel absorption by the pores is not explicitly accounted for. Furthermore, gel absorption by the pores is accounted for by the kinetic equation through the latency time which depends on concrete porosity. The higher the porosity, the larger the latency time.

Γ_c in turns accounts for the reduction in AAR volumetric expansion under compressive stresses (in which case gel is absorbed by diffused micro-cracks), (Multon 2004):

$$\Gamma_c = \begin{cases} 1 & \text{if } \bar{\sigma} \leq 0. \quad \text{Tension} \\ 1 - \frac{e^{\beta \bar{\sigma}}}{1 + (e^{\beta} - 1) \bar{\sigma}} & \text{if } \bar{\sigma} > 0. \quad \text{Compression} \end{cases} \quad (2.12)$$

$$\bar{\sigma} = \frac{\sigma_I + \sigma_{II} + \sigma_{III}}{3f'_c} \quad (2.13)$$

Whereas this expression will also reduce expansion under uniaxial or biaxial confinement, Fig. 2.4, these conditions are more directly accounted for below through the assignment of weights.

The third major premise of the model, is that the volumetric AAR strain must be redistributed to the three principal directions according to their relative propensity for expansion on the basis of a weight which is a function of the respective stresses. Whereas the determination of the weight is relatively straightforward for triaxial AAR expansion under uniaxial confinement (for which some experimental data is available), it is more problematic for biaxially or triaxially confined concrete.

Given principal stress vector defined by $\sigma_k, \sigma_l, \sigma_m$, we need to assign a weight to each of those three principal directions. These weights will control AAR volumetric expansion distribution. For instance, with reference to Fig. 2.5, we consider three scenarios.

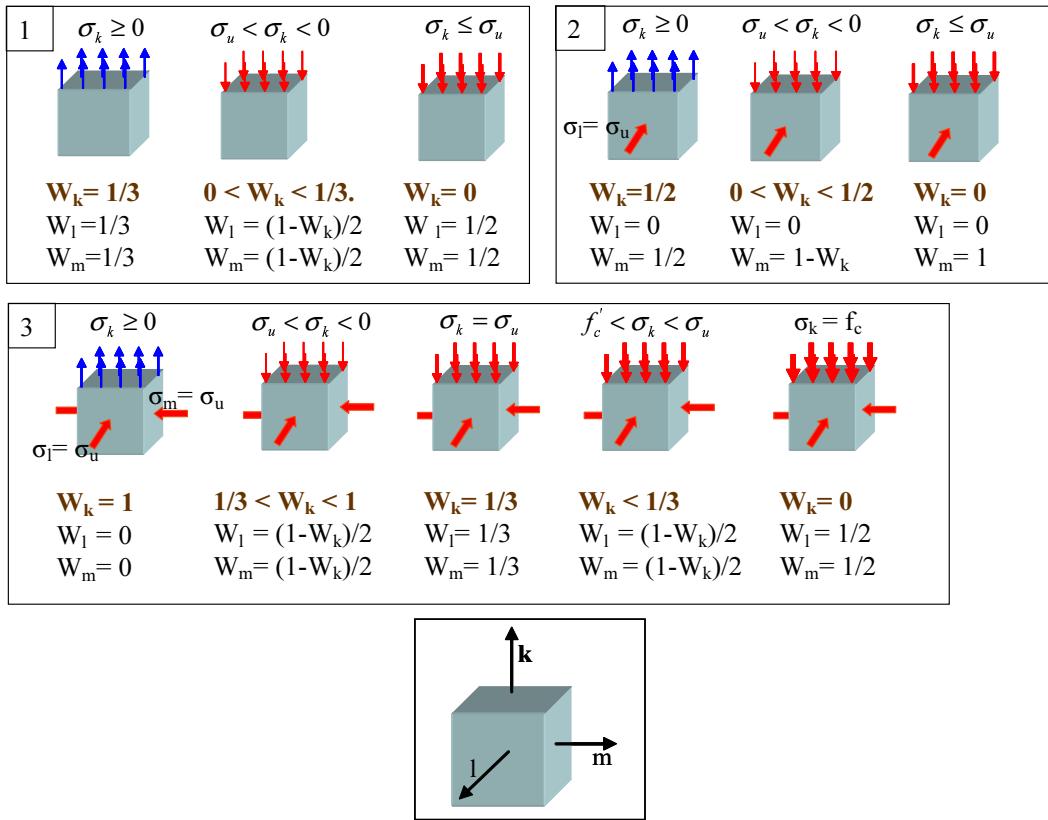


Figure 2.5: Weight of Volumetric AAR Redistribution in Selected Cases

Uniaxial State of stress, where we distinguish the following three cases:

1. In the first case, we have uniaxial tension, and hence, the volumetric AAR strain is equally redistributed in all three directions.
2. Under a compressive stress greater than the limiting one (σ_u), the weight in the corresponding (k) direction should be less than one third. The remaining AAR has to be equally redistributed in the other two directions.
3. If the compressive stress is lower than σ_u , than AAR expansion in the corresponding direction is prevented (weight equal zero), and thus the other two weights must be equal to one half.

Biaxial state of stress in which we have a compressive stress equal to σ_u in one of the three principal directions. In this case, the corresponding weight will always be equal to zero. As to the possible three combinations:

1. Tension in one direction, equal weights of one half.
2. Compression greater than σ_u in one direction, then the corresponding weight must be less than one half, and the remaining weight is assigned to the third direction.
3. Compression less to σ_u , then the corresponding weight is again zero, and a unit weight is assigned to the third direction.

Triaxial state of stress in which we have σ_u acting on two of the three principle directions. We identify the following five cases:

1. Tension along direction k , then all the expansion is along k .
2. Compressive stress greater than σ_u , then we have a triaxial state of compressive stress, and the corresponding weight will be between one and one third. The remaining complement of the weight is equally distributed in the other two directions.
3. Compression equal to σ_u , hence we have a perfect triaxial state of compressive stress. In this case we have equal weights of one third. It should be noted that the overall expansion is reduced through Γ_c .
4. Compression less than σ_u but greater than the compressive strength. In this case, the weight along k should be less than one third, and the remaining equally distributed along the other two directions.
5. Compression equal to the compressive strength. In this case, the corresponding weight is reduced to zero, and the other two weights are equal to one half each.

Based on the preceding discussion, we generalize this weight allocation scheme along direction k as follows

1. Given σ_k , identify the quadrant encompassing σ_l and σ_m , Fig. 2.6¹. Weight will be determined through a bilinear interpolation for those four neighboring nodes.
2. Determine the weights of the neighboring nodes from Table 2.2 through proper linear interpolation of σ_k .
3. Compute the weight from:

$$W_k(\sigma_k, \sigma_l, \sigma_m) = \sum_{i=1}^4 N_i(\sigma_l, \sigma_m) W_i(\sigma_k) \quad (2.14)$$

where N_i is the usual two bilinear shape function used in finite element and is given by

$$\mathbf{N}(\sigma_l, \sigma_m) = \frac{1}{ab} \begin{bmatrix} (a - \sigma_l)(b - \sigma_m) & \sigma_l(b - \sigma_m) & \sigma_l\sigma_m & (a - \sigma_l)\sigma_m \end{bmatrix} \quad (2.15)$$

$$\mathbf{W}(k) = \begin{bmatrix} W_1(\sigma_k) & W_2(\sigma_k) & W_3(\sigma_k) & W_4(\sigma_k) \end{bmatrix}^t \quad (2.16)$$

$$a = (a_1|a_2|a_3) \quad b = (b_1|b_2|b_3) \quad (2.17)$$

$$\sigma_l = (\sigma_l|f'_c - \sigma_l) \quad \sigma_m = (\sigma_m|f'_c - \sigma_m) \quad (2.18)$$

The $i - j$ stress space is decomposed into nine distinct regions, Fig. 2.6, where σ_u is the upper (signed) compressive stress below which no AAR expansion can occur along the corresponding direction (except in triaxially loaded cases). Hence, a and b are the dimensions of the quadrant inside which σ_i and σ_j reside.

¹ Since compressive stresses are quite low compared to the compressive strength, we ignore the strength gained through the biaxiality or triaxiality of the stress tensor (Kupfer and Gerstle 1973). Furthermore, the strength gain is only about 14% for equibiaxial compressive stresses, (CEB 1983).

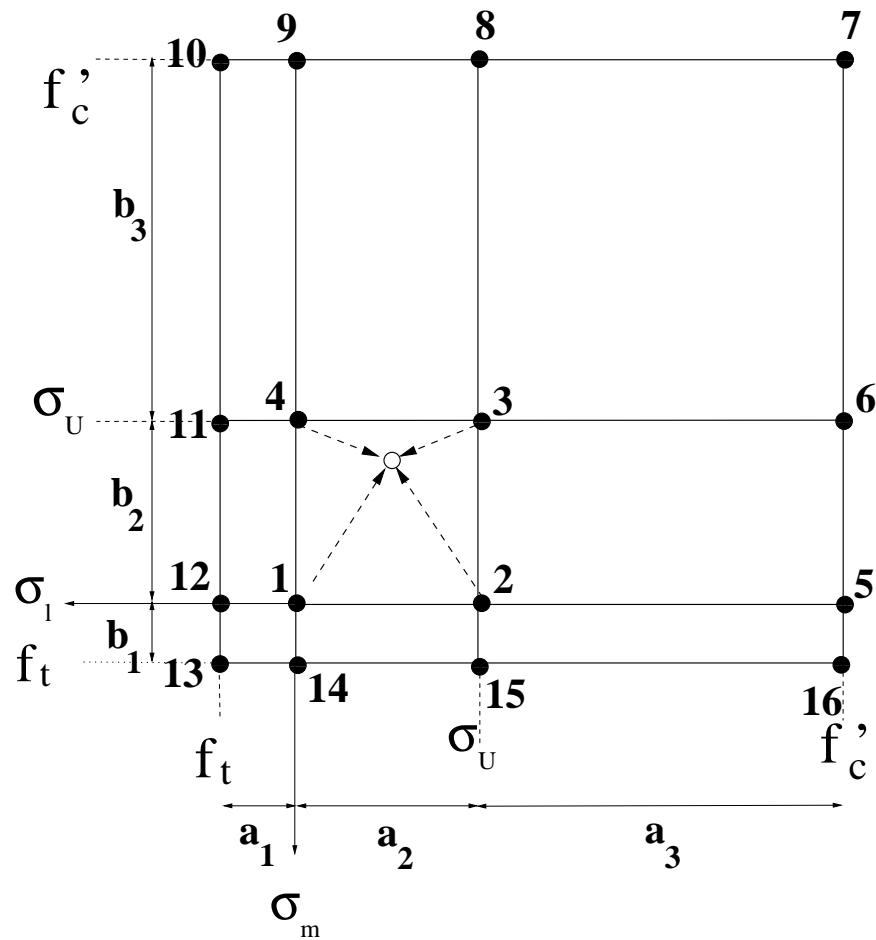


Figure 2.6: Weight Regions

Weights of the individual nodes are in turn interpolated according to the principal stress component in the third direction σ_k , Table 2.2. It should be noted that those weights are for the most part based on the work of Larive (1998) and Multon (2004), but in some cases due to lack of sufficient experimental data, based on simple “engineering common sense”. A simple example for the evaluation of the weight is shown in the appendix.

Based on the earlier work of Struble and Diamond (1981a), in which it was reported that no gel expansion can occur at pressures above 11 MPa (though for a synthetic gel), σ_u is taken as -10 MPa. This value was also confirmed by Larive (1998). f'_t and f'_c are the concrete tensile and compressive strengths respectively.

Individual strain is given by

$$\dot{\varepsilon}_i^{AAR} = W_i \dot{\varepsilon}_V^{AAR} \quad (2.19)$$

and the resulting relative weights are shown in Fig. 2.7.

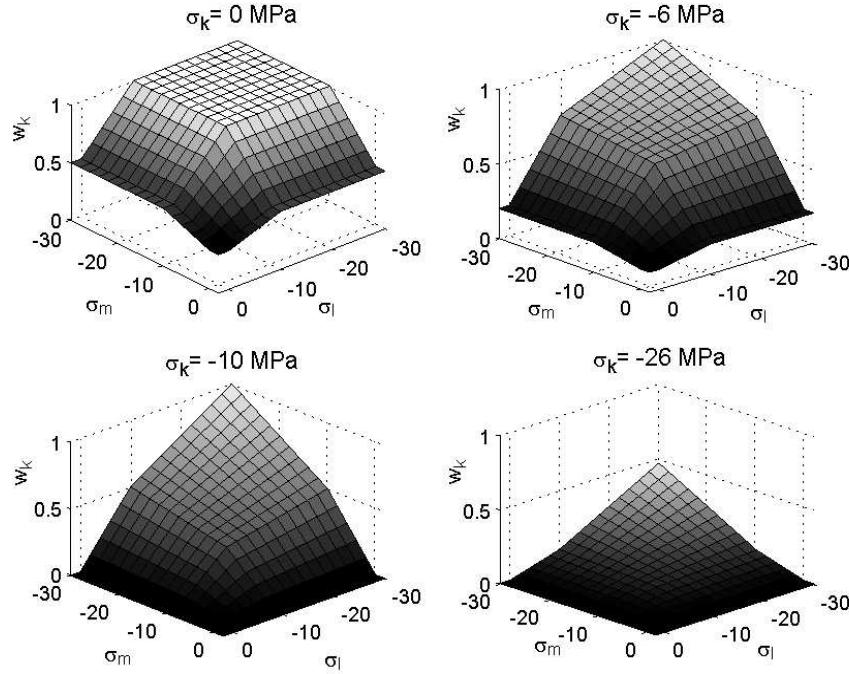


Figure 2.7: Relative Weights

It should be noted that the proposed model will indeed result in an anisotropic AAR expansion. While not explicitly expressed in tensorial form, the anisotropy stems from the different weights assigned to each of the three principal directions.

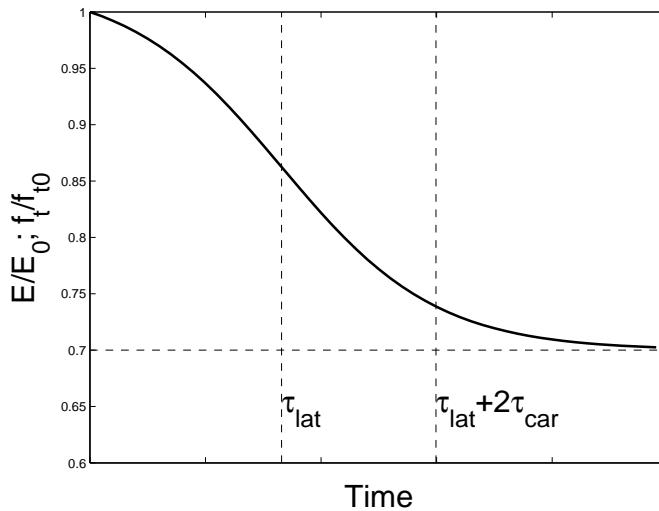
This deterioration being time dependent, the following time dependent nonlinear model is considered, Fig. 2.8.

$$E(t, \theta) = E_0 [1 - (1 - \beta_E) \xi(t, \theta)] \quad (2.20)$$

$$f'_t(t, \theta) = f'_{t,0} [1 - (1 - \beta_f) \xi(t, \theta)] \quad (2.21)$$

where E_0 and $f'_{t,0}$ are the original elastic modulus and tensile strength, β_E and β_f are the corresponding residual fractional values when ε_{AAR} tends to ε_{AAR}^∞ .

Finally, the possible decrease in compressive strength with AAR was ignored. Most of the literature dwelling on the mechanical properties of concrete subjected to AAR show little evidence of a decrease in compressive strength (as one would expect since the stresses will be essentially closing the AAR induced cracks). Furthermore, in dams (gravity and arch) compressive stresses are well below the compressive strength, which is quite different from the tensile stresses.

Figure 2.8: Degradation of E and f'_t

		Time (days)	ε^∞	Iterations
Characteristic	Latency			
Longitudinal Expansion				
Initial	100.0	100.0	1.00e-03	-
Final	82.9	146.5	3.63e-03	8
Transversal Expansion				
Initial	100.0	100.0	1.00e-03	-
Final	68.9	111.0	2.62e-3	7

Table 2.1: System Identification for Multon's Tests

2.5 VALIDATION

Validation and parameter identification was accomplished by analyzing tests of Multon (2004). In those tests, 130 by 240 mm concrete was cast inside a steel cylinder with 3 or 5 mm thickness and subjected to 0, 10 or 20 MPa compressive stress.

Fig. 2.9 shows the 3D finite element mesh adopted (in addition to an axisymmetric one) along with the results of the parameter identification study under free expansion for τ_l , τ_c , and ε^∞ . Starting and final parameters are also shown in Table 2.1. Having determined this initial set of kinetic parameters, another parameter identifications for the parameter β in Eq. 2.12 for the constrained specimens yielded a value of 0.5, Fig. 2.10.

Finally the parameter β was used in the subsequent dam analysis. Other kinetic parameters were determined through laboratory experiments of concrete specimens recovered from the dam.

2.6 APPLICATION

2.6.1 Dam Analysis Data Preparation

Finally, a typical application to a 2D analysis of an arch gravity dam is presented. The model has been used in the 3D nonlinear predictive analysis of an actual arch gravity dam, and it was shown that 50 years after dam construction, the reaction will be exhausted, (Saouma and Perotti 2004b).

The comprehensive incremental AAR analysis of a concrete dam is relatively complex, irrespective of the selected AAR model, as data preparation for the load can be cumbersome.

First the seasonal pool elevation variation (for both thermal and stress analysis), and the stress free temperature, T_{ref} (typically either the grouting temperature, or the average yearly temperature) must be identified, along with the external temperature, Figs. 2.11.

Then, a transient thermal analysis is performed since the reaction is thermodynamically activated, and the total

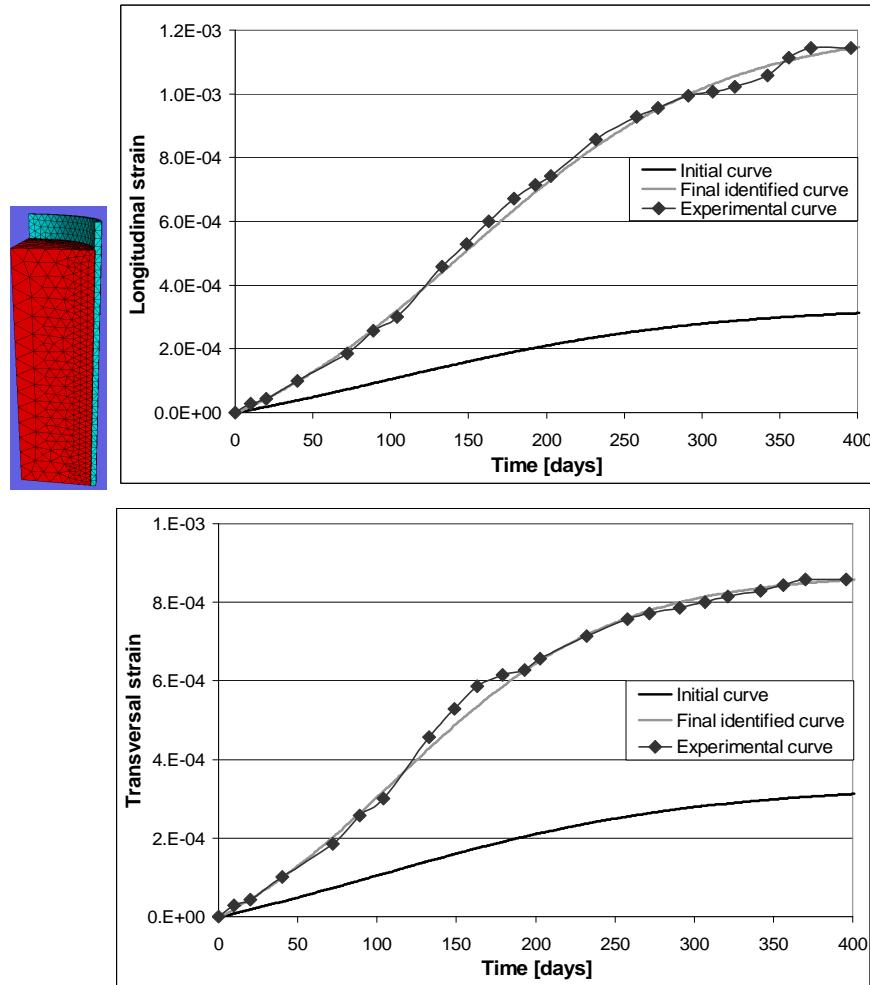
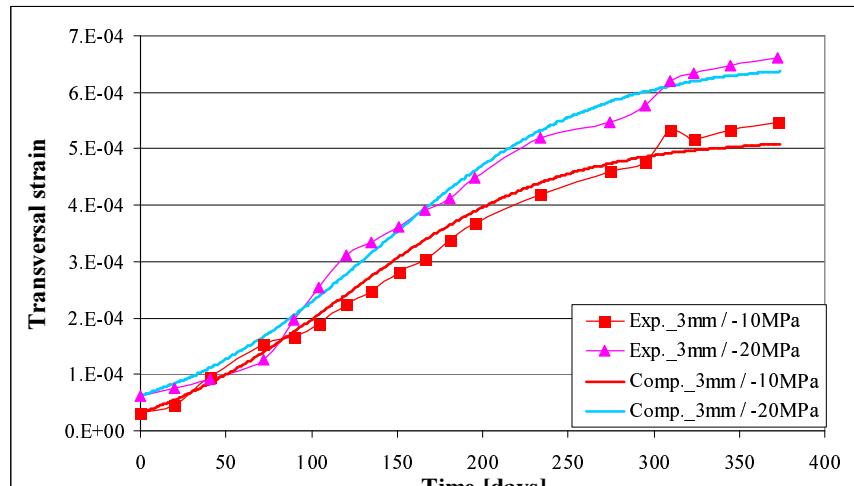
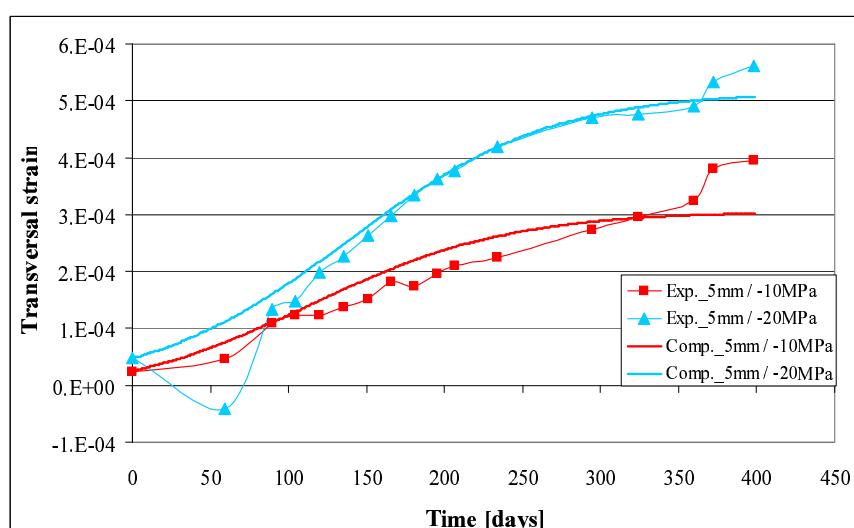


Figure 2.9: Multon's Test Parameter Identification Results for Free Expansion; Longitudinal and Corresponding Transversal Strains. Initial Curve corresponds to Initial Guess in System identification

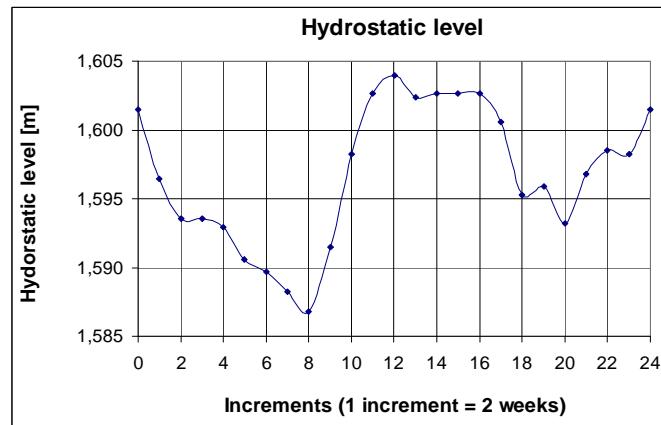


(a) 3 mm Cylinders

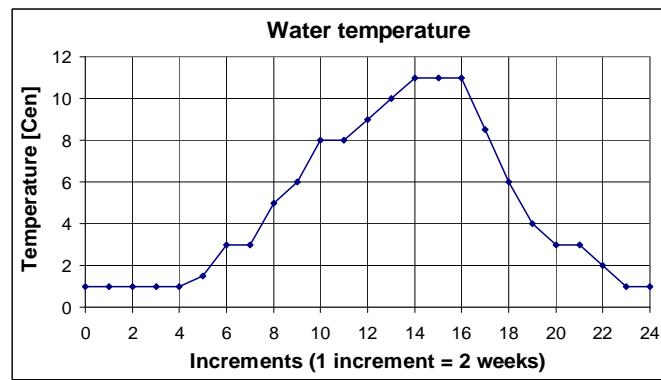


(b) 5 mm Cylinders

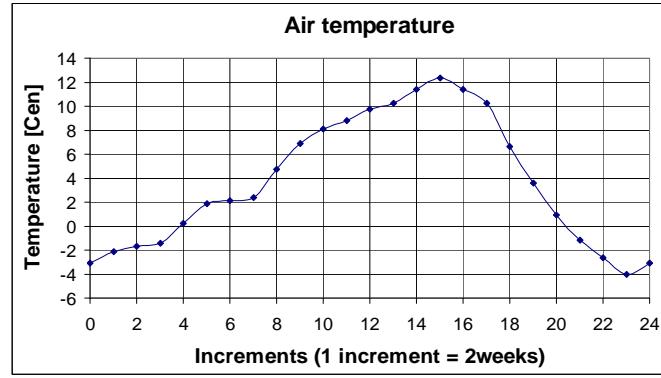
Figure 2.10: Comparison between Experimental Results of Multon and Numerical Calculation (After parameter identifications)



(a) Pool Elevation.



(b) Water Temperature



(c) Air Temperature

Figure 2.11: Yearly Variation of Hydrostatic and Thermal Load

temperature is hence part of the constitutive model. Heat transfer by conduction only is accounted for. Convection and radiation are approximated through an additional temperature, (Malla and Wieland 1999).

The selected incremental time was two weeks, and the initial reference temperature set to zero. Given the external air temperature, the pool elevation, and the water temperature boundary conditions were set to this initial boundary value problem. Analysis is performed with Merlin, and temperature fields examined. It was determined that after four years the temperature field is harmonic with a one year frequency. At that point, the analysis is interrupted and $T_{thermal}(x, y, t)$ saved.

Following the thermal analysis, $T_{thermal}(x, y, t)$ must be transferred to $T_{stress}(x, y, t)$ as in general we do not have the same finite element mesh (foundations, joints and cracks are typically not modelled in the thermal analysis). Following this, a comprehensive input data file must be prepared for the stress analysis. It includes:

1. Gravity load (first increment only).
2. $\Delta\dot{T}(x, y, t) = \dot{T}_{stress}(x, y, t) - T_{ref}$ in an incremental format. This is a delicate step which can not be overlooked. In particular the stress analysis is based on the difference between actual and stress free temperature. In addition an incremental analysis, requires this set of data to be given in an incremental form.
3. Stress free referenced temperature which will be added to the temperature data to determine the total absolute temperature needed for AAR.
4. Cantilever and dam/foundation joint characteristics. The first must be accounted for in an arch dam, as the expansion may lead to upstream joint opening. The second must be accounted for as the AAR induced swelling may result in separation of the dam from the foundation in the central portion of the foundation.
5. Uplift load characteristics (typically in accordance with the upstream hydrostatic load).
6. AAR data as described above It should be noted that a first order approximation of the AAR kinetics parameters may be recovered from laboratory tests of dam cores or through an inverse analysis of the dam crest displacement.

Finally, the assembled set of data must be looped over at least fifty years to provide a complete and correct set of natural and essential boundary conditions. For a 2D problem, this will result in files approximately 45 MB.

2.6.2 Dam Analysis Results

For this preliminary plane strain analysis, a 2D central section of an arch gravity dam is selected. Results based on the proposed model will be contrasted with those obtained using current State of the Practice model, (Charlwood et al. 1992) with a linear kinetics expansion. In this analysis, creep is not accounted for, and the laboratory determined Young's modulus is retained throughout both analyses (Whereas Charlwood tends to substantially reduce E to account for the creep, which in turn may yield potentially lower stresses).

In order to compare both analysis, final volumetric expansion has been calibrated to yield identical vertical crest displacement after 50 years, Fig. 2.12, where the proposed model nonlinearity in the crest displacement is caused by the kinetics model, and its latency time in particular. Despite equal final crest displacements, internal field stresses are drastically different as those determined from Charlwood's model are substantially lower than those predicted by the proposed model, Fig. 2.13. It should be noted that the large discrepancy in stresses is, partially, caused by the plane strain (which inhibits redistribution in the third direction) assumption of the authors model. However, undoubtedly the lack of stress redistribution in Charlwood's model will lead to an underestimation of the stress field.

Furthermore, due to the influence of the thermal load, the proposed model causes tensile stresses inside the concrete dam, and a lift off along the central portion of the dam-foundation interface, Fig. 2.14. These internal tensile stresses can possibly explain the formation of the crack observed inside the gallery in the analyzed dam. More details can be found in (Saouma and Perotti 2004a).

Finally, no attempt is made to correlate computed crest displacements with the (available) field measurements. The two-dimensional plane strain analysis conducted preclude such a realistic comparison which is performed in a separate publication, (?). Furthermore, it should be noted that any model, irrespective of its scientific merits, can be calibrated with field measurements. However, only those models solidly based on the chemistry, physics and mechanics of AAR are likely to yield realistic stress field which is what ultimately Engineers worry about.

2.7 CONCLUSIONS

A new constitutive model for AAR expansion is presented.

This thermo-chemo-mechanical model is rooted in the chemistry (kinetic of the reaction), physics (crack gel absorption, effect of compression), and mechanics of concrete. The major premises of the model is the assumption of a

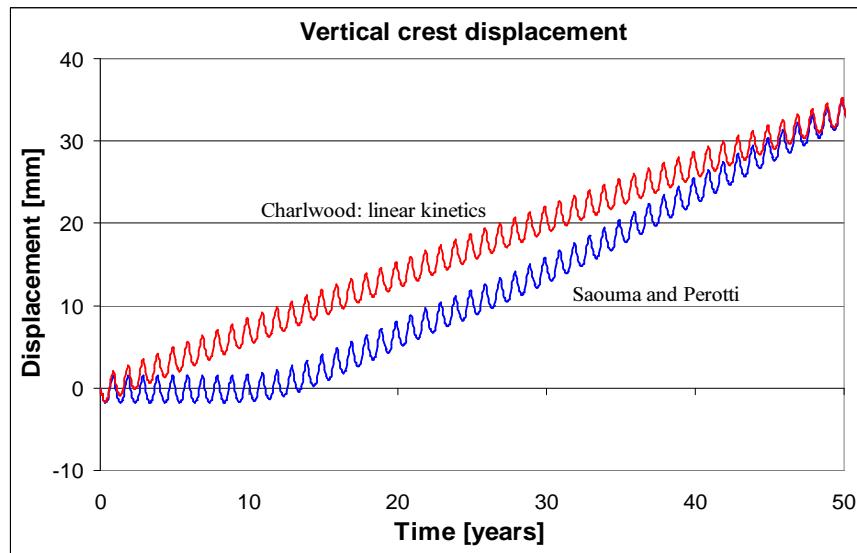


Figure 2.12: Yearly Variation of Vertical Crest Displacements; Upper Curve based on Charlwood's Model, Lower Curve based on Proposed Model

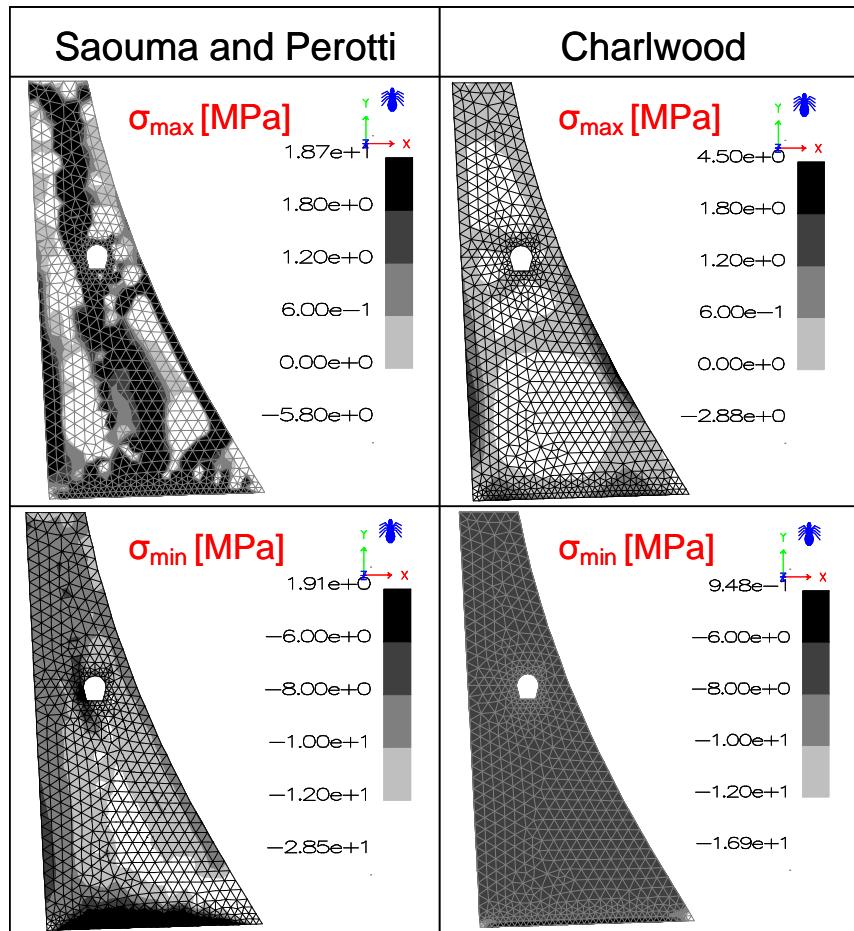


Figure 2.13: Principal Stress Field Comparison Between proposed and State of the Practice Model (without joints)

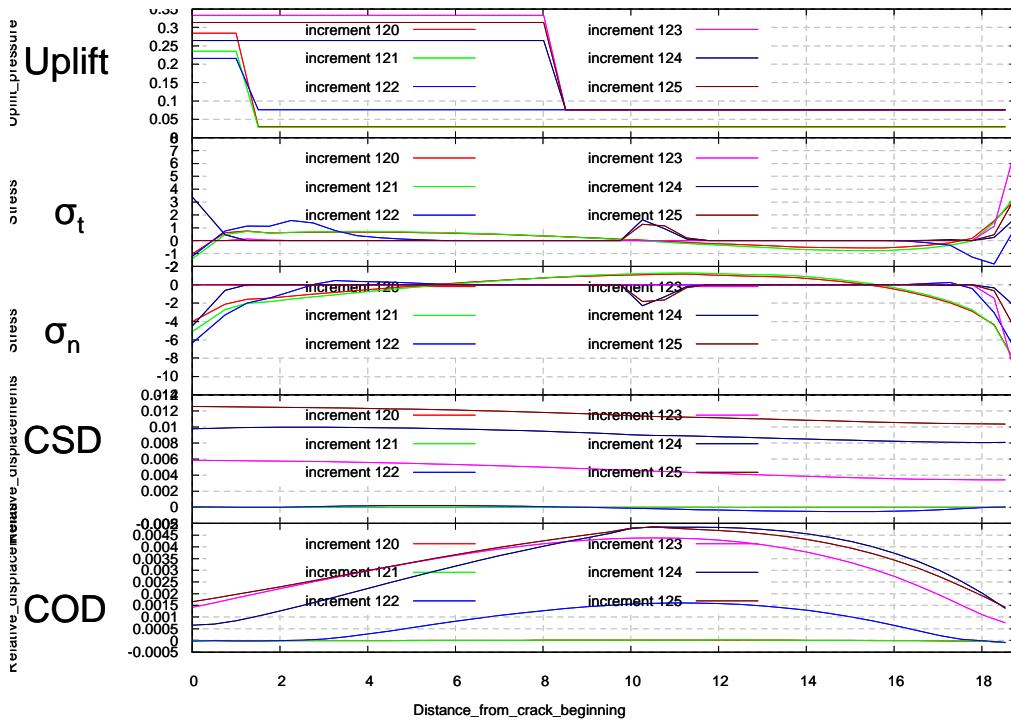


Figure 2.14: Dam/Foundation Interface Joint Characteristics; Uplift; tangential and normal stresses; Crack sliding and opening displacements along the joint.

volumetric expansion, redistribution on the basis of weights related to the stress tensor, and contrarily to previous models the stress field affects reaction kinetics which is a slight modification of the model of Larive (1998).

The model has been used, in conjunction with a formal parameter identification paradigm, to analyze the three dimensional tests of (Multon 2004). Detailed 2D analysis of an arch gravity dam is presented.

2.8 Appendices

2.8.1 Example of Weight Determination

A simple example for weight determination is shown here. Assuming that the principal stresses are given by $\begin{bmatrix} \sigma_1 & \sigma_m & \sigma_k \end{bmatrix} = \begin{bmatrix} -5.0 & -8.0 & -5.0 \end{bmatrix}$ MPa, and that f_c , f'_t and σ_u are equal to -30.0, 2.0, and -10.0 MPa respectively, we seek to determine W_k .

The stress tensors places us inside the quadrant defined by nodes 1-2-3-4 whose respective weights are equal to: $W_1 = \frac{1}{2}(\frac{1}{3}) = \frac{1}{6}$, $W_2 = \frac{1}{2}(\frac{1}{2}) = \frac{1}{4}$, $W_3 = \frac{1}{3} + \frac{1}{2}(1.0 - \frac{1}{3}) = \frac{2}{3}$, and $W_4 = \frac{1}{2}(\frac{1}{2}) = \frac{1}{4}$ a and b are both equal to -10 MPa, and the “shape factors” will be $N_1 = \frac{1}{100} [(-10 + 5)(-10 + 8)] = \frac{1}{10}$, $N_2 = \frac{1}{100} [-5(-10 + 8)] = \frac{1}{10}$, $N_3 = \frac{1}{100} [(-5)(-8)] = \frac{4}{10}$, $N_4 = \frac{1}{100} [-8(-10 + 5)] = \frac{4}{10}$, and finally $W_k = \frac{1}{10} \times \frac{1}{6} + \frac{1}{10} \times \frac{1}{4} + \frac{4}{10} \times \frac{2}{3} + \frac{4}{10} \times \frac{1}{4} = 0.40833$

2.8.2 Derivation of Kinetics Relation

Following is the derivation of the kinetics law by Larive (1998).

For a closed isotropic and isothermal system under constant pressure, the free energy is defined as

$$\Psi(\varepsilon, \theta_0, \xi) = \frac{1}{2} K \varepsilon v^2 - \alpha K \varepsilon v \xi - A_0 \xi + \frac{1}{2} L \xi^2 \quad (2.22)$$

Next we derive the state equations, starting with the stress.

$$\sigma \equiv \frac{\partial \Psi}{\partial \varepsilon} \Rightarrow \sigma = K \varepsilon - K \underbrace{\alpha \xi}_{\varepsilon_{ch}} \Rightarrow \sigma = K(\varepsilon - \varepsilon_{ch}) \quad (2.23)$$

hence K corresponds to the bulk modulus, and α to a coefficient of chemical expansion.

Node			Weights		
No.	σ_l	σ_m	$\sigma_k \geq 0$	$\sigma_k = \sigma_u$	$\sigma_k = f'_c$
1	0.	0.	1/3	0.	0.
2	σ_u	0.	1/2	0.	0.
3	σ_u	σ_u	1.	1/3	0.
4	0.	σ_u	1/2	0.	0.
5	f'_c	0.	1/2	0.	0.
6	f'_c	σ_u	1.	1/2	0.
7	f'_c	f'_c	1.	1.	1/3
8	σ_u	f'_c	1.	1/2	0.
9	0.	f'_c	1/2	0.	0.
10	f'_t	f'_c	1/2	0.	0.
11	f'_t	σ_u	1/2	0.	0.
12	f'_t	0.	1/3	0.	0.
13	f'_t	f'_t	1/3	0.	0.
14	0.	f'_t	1/3	0.	0.
15	σ_u	f'_t	1/2	0.	0.
16	f'_c	f'_t	1/2	0.	0.

Table 2.2: Triaxial Weights

Next, we compute the thermodynamic force

$$F \equiv -\frac{\partial \Psi}{\partial \xi} \Rightarrow F \equiv \mathcal{A} = \mathcal{A}_0 + \alpha K \varepsilon - L \xi \quad (2.24)$$

where \mathcal{A} is the affinity of the reaction and is assumed to be linearly proportional to the reaction velocity $\dot{\xi}$. Since there is ample experimental evidence that the reaction is thermodynamically activated, we consider Arrhenius law

$$\mathcal{A} = k_d \exp\left(\frac{E_a}{RT}\right) \dot{\xi} \quad (2.25)$$

which combined with the previous equation yields

$$\frac{\mathcal{A}}{\alpha} = \left(\sigma + \frac{\mathcal{A}_0}{\alpha} \right) - \kappa \varepsilon_{ch} = \eta \dot{\varepsilon}_{ch} \quad (2.26)$$

where $\kappa = \frac{L}{\alpha^2} - K$, and $\eta = \frac{k_d}{\alpha^2} \exp\left(\frac{E_a}{RT}\right)$. This equation highlight the chemical-mechanical coupling present in an AAR reaction. However, if $\sigma \ll \frac{\mathcal{A}_0}{\alpha}$, we can uncouple the two equations. Hence, for constant σ , and η , we obtain

$$\varepsilon = \frac{\sigma}{K} + \varepsilon^\infty \left(1 - \exp\left(-\frac{t}{\tau}\right) \right) \quad (2.27)$$

where

$$\varepsilon^\infty = \frac{\sigma}{K} + \frac{\mathcal{A}_0}{\alpha \kappa} \quad (2.28)$$

$$\tau = \frac{\eta}{\kappa} \quad (2.29)$$

Next, assuming free expansion, we have the following relation

$$\left. \begin{aligned} \varepsilon &= \varepsilon_{ch} = \alpha \xi \\ \mathcal{A} &= \mathcal{A}_0 - \alpha \kappa \varepsilon_{ch} = \alpha \eta \dot{\varepsilon}_{ch} \\ \varepsilon^\infty &\equiv \frac{\mathcal{A}_0}{\alpha \kappa} \\ \eta &= \frac{k_d}{\alpha^2} \exp\left(\frac{E_a}{RT}\right) \end{aligned} \right\} \varepsilon^\infty - \varepsilon_{ch} = \alpha \lambda \dot{\varepsilon}_{ch} \quad (2.30)$$

The reaction rate must decrease as ε_{ch} increases (based on laboratory experiments, and the fact that there is a limited supply of reactive agents). Hence, η (and thus λ) must also decrease in terms of the chemical reaction strain ε_{ch} , we assume

$$\lambda(\varepsilon_{ch}) = \frac{a}{b + \varepsilon_{ch}} \quad (2.31)$$

thus

$$\left. \begin{aligned} \varepsilon^\infty - \varepsilon_{ch} &= \lambda(\varepsilon_{ch})\dot{\varepsilon}_{ch} \\ \lambda(\varepsilon_{ch}) &= \frac{a}{b + \varepsilon_{ch}} \end{aligned} \right\} \Rightarrow \varepsilon_{ch}(t) = \frac{1 - e^{-\frac{t}{\tau_{car}}}}{1 + \frac{\varepsilon^\infty}{b} e^{-\frac{t}{\tau_{car}}}} \varepsilon^\infty \quad (2.32)$$

where

$$\tau_{car} \equiv \frac{a}{b + \varepsilon^\infty} \quad (2.33)$$

Defining

$$\tau_{lat} = \tau_{car} \ln \left(\frac{\varepsilon^\infty}{b} \right) \quad (2.34)$$

we finally obtain, Fig. 2.15.

$$\varepsilon_{ch}(t) = \frac{1 - e^{-\frac{t}{\tau_{car}}}}{1 + e^{-\frac{t-\tau_{lat}}{\tau_{car}}}} \varepsilon^\infty \quad (2.35)$$

Note: this equation is a generalization of the well known sigmoid curve $P = \frac{1}{1+e^{-t}}$, which is the solution of the

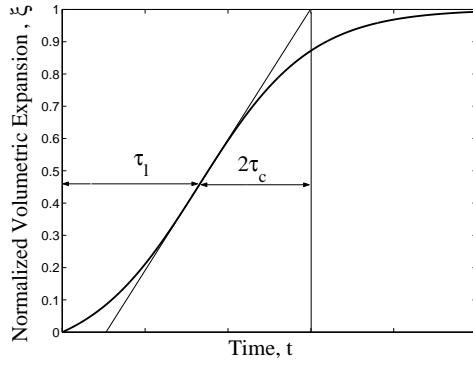


Figure 2.15: Expansion Curve

differential equation $\frac{dP}{dt} = kP(C - P)$, where k is a proportionality constant, C is a carrying capacity, and expresses the fact that a population growth is jointly proportional to the present population size and the amount by which that size falls short of the carrying capacity.

In our problem, the “population” is the affinity of the reaction \mathcal{A} .

Reexamining Eq. 2.35, the equation being thermodynamically activated, we can rewrite it as

$$\varepsilon(t) = \frac{1 - \exp(-t/\tau_{car}(\theta))}{1 + \exp(-t/\tau_{car}(\theta) + \tau_{lat}(\theta)/\tau_{car}(\theta))} \varepsilon^\infty \quad (2.36)$$

where θ is the absolute temperature. Hence, again, from Arrhenius law (Eq. 2.25) we can write

$$\tau = Ae^{U/\theta} \quad (2.37)$$

where a positive exponent is given since $\theta \nearrow, \tau \searrow$ hence, $\ln(\tau) = \ln(A) + \frac{U}{\theta}$ or $\tau = \tau_0 \exp \left[U \left(\frac{1}{\theta} - \frac{1}{\theta_0} \right) \right]$ Thus,

$$\tau_l = \tau_{l0} \exp \left[U_l \left(\frac{1}{\theta} - \frac{1}{\theta_0} \right) \right] \quad (2.38)$$

$$\tau_c = \tau_{c0} \exp \left[U_c \left(\frac{1}{\theta} - \frac{1}{\theta_0} \right) \right] \quad (2.39)$$

Activation energies can then be obtained by determining the characteristic or latency time (τ_c, τ_l) for different temperatures, and then plotting their log values in terms of the inverse of the absolute temperature, Fig. 2.16.

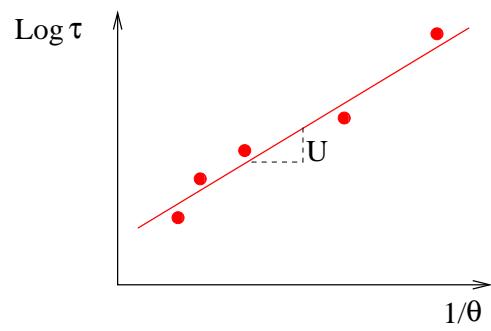


Figure 2.16: Determination of the Activation Energies

Chapter 3

PSEUDO-HYDRODYNAMIC FORCES

3.1 Westergaard

3.1.1 Static Analysis; Pseudo Hydrodynamic Forces

During an earthquake, the interaction between the gravity dam and the reservoir creates additional pressures on the upstream face of the dam. These hydrodynamic pressures may be approximated by the Westergaard (1933) formula, which uses a parabolic approximation for the additional pressures due to earthquake motion. Fig. 3.1 illustrates the forces due to the total water pressures during an earthquake. Note that the hydrodynamic forces act in both

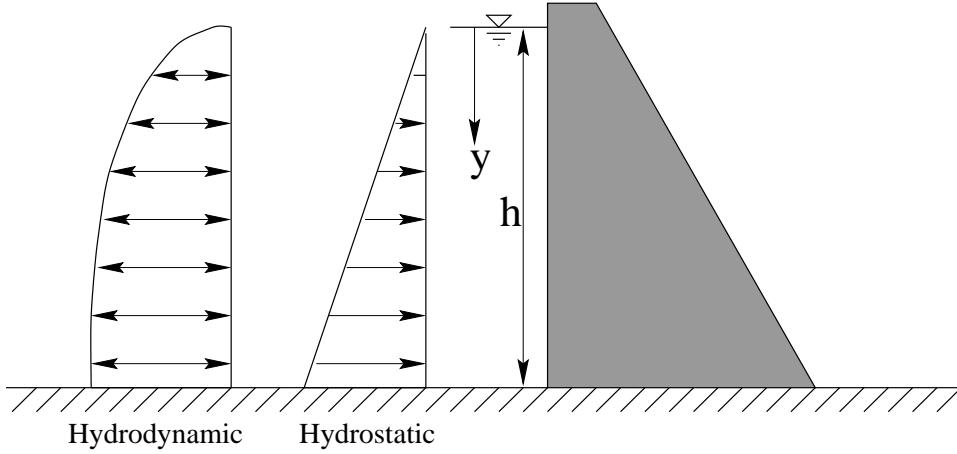


Figure 3.1: Hydrostatic and hydrodynamic forces during earthquake excitation

directions.

Westergaard defines the hydrodynamic pressure p and force q at depth y below the reservoir surface, with total reservoir height h , as

$$p = C_e \alpha \sqrt{hy} \quad (3.1)$$

$$q = \frac{2}{3} C_e \alpha y \sqrt{hy} \quad (3.2)$$

where α measures the intensity of the earthquake by the relation $a = \alpha g$, where a is the maximum horizontal acceleration and g the acceleration of gravity. At the bottom of the reservoir, the maximum hydrodynamic pressure p_o is

$$p_o = C_e \alpha h \quad (3.3)$$

and the total resultant force, q_o is

$$q_o = \frac{2}{3} C_e \alpha h^2 \quad (3.4)$$

These parameters are illustrated in Fig. 3.2.

The coefficient C_e is a correction factor to account for water compressibility. ? define this parameter for both SI and English units as

$$\text{SI: } C_e = \left(\frac{0.543}{0.583} \right) \left(\frac{7}{8} \right) \left(9.81 \frac{\text{kN}}{\text{m}^3} \right) C_c = 7.99 C_c \quad C_c = \frac{1}{\sqrt{1 - 7.75 \left(\frac{h}{1000T} \right)^2}} \quad (3.5-a)$$

$$\text{English: } C_e = \left(\frac{0.543}{0.583} \right) \left(\frac{7}{8} \right) \left(0.0624 \frac{\text{kip}}{\text{ft}^3} \right) C_c = 0.051 C_c \quad C_c = \frac{1}{\sqrt{1 - 0.72 \left(\frac{h}{1000T} \right)^2}} \quad (3.5-b)$$

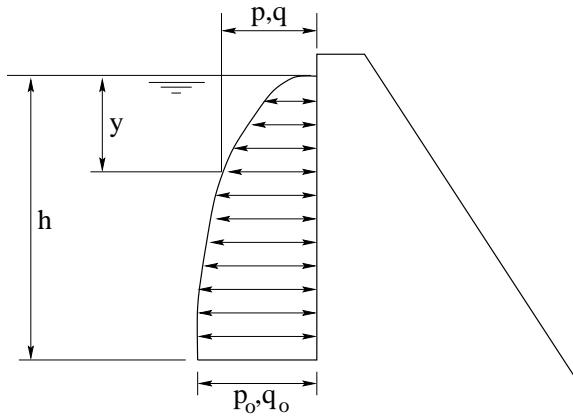
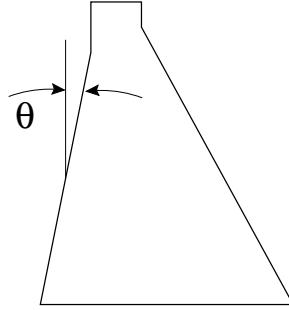


Figure 3.2: Hydrodynamic water pressure and force

where T is the period to characterize the ground seismic acceleration imposed on the dam, in seconds. The SI expression expects units of kilonewton, meter, and seconds, while the English expression uses kips, feet, and seconds.

If the upstream face of the dam is sloped, as in Fig. 3.3, a correction factor K_θ is applied to the p and q relations in Eqs. 3.1 and 3.2. For an angle of slope θ from the vertical, the correction factor is simply $K_\theta = \cos^2 \theta$. The final

Figure 3.3: Sloped upstream dam face - definition of θ angle

expressions for the hydrodynamic pressure and force are then

$$p = C_e \alpha K_\theta \sqrt{hy} \quad [\text{force/length}^2] \quad (3.6)$$

$$q = \frac{2}{3} C_e \alpha K_\theta y \sqrt{hy} \quad [\text{force/length}] \quad (3.7)$$

3.1.2 Dynamic Analysis; Added Masses

According to Westergaard (1933) one can visualize the dynamic action of water on the upstream face of a dam, by thinking of a certain body of water in the reservoir as moving with the dam while the remainder of the reservoir remains inactive, Fig. 3.4. Westergaard has shown that the shape of this body of water is parabolic, with the vertex of the parabola located at the reservoir surface.

Fig. 3.4 shows the parabolic shape of the water that may be considered as contributing to the mass of the dam during earthquake excitation. In the figure, h is the reservoir depth, y is the distance from the reservoir surface to a point under the water, and b is the distance from the dam face to the parabola at the depth y . Westergaard states that at the depth y , the corresponding added mass per unit area of the upstream face of the dam is

$$\gamma_{lump} = \frac{b \rho_w}{g} \quad (3.8)$$

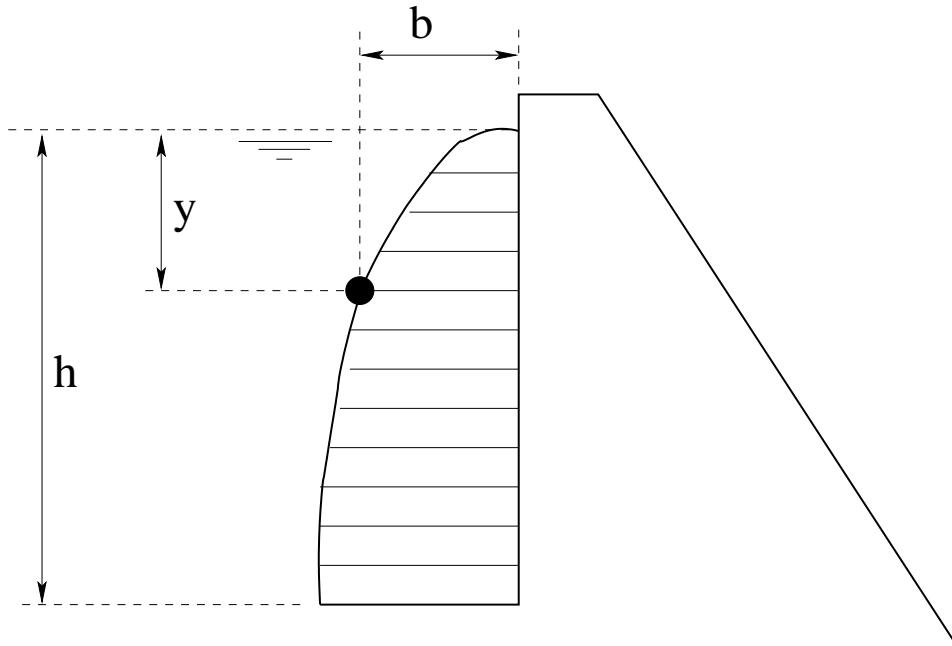


Figure 3.4: Westergaard's Added Mass concept

where ρ_w is the weight per unit volume of water, and g is the acceleration of gravity. b is defined as

$$b = \frac{p}{\alpha \rho_w} \quad (3.9)$$

where p is the hydrodynamic pressure of the dam (Eq. 3.6), and α is the ratio measuring the intensity of the earthquake (a fraction of g). This hydrodynamic pressure (which also assumes a parabolic distribution on the dam face) is defined by

$$p = C\alpha\sqrt{hy} \quad (3.10)$$

where C is a constant defined by

$$C = \frac{K}{\sqrt{1 - \frac{16\rho_w h^2}{gkT^2}}} \quad (3.11)$$

where K is a constant defined by Westergaard as $K = 51 \text{ lb/ft}^3$ ($8,011.4 \text{ N/m}^3$), k is the elastic modulus of water, and T is the period of ground horizontal vibration.

Combining these relations and canceling when possible results in the final relation to determine the lumped mass per unit area of the upstream dam face due to dynamic action of the water on the dam

$$\gamma_{lump} = \frac{K\sqrt{hy}}{g} \frac{1}{\sqrt{1 - \frac{16\rho_w h^2}{gkT^2}}} \quad (3.12)$$

It should be noted that in his original paper, Westergaard did go through an additional simplification of the preceding equation (removing the dependency on T) yielding:

$$\gamma_{lump} = \frac{7}{8} \rho_w \sqrt{hy} \quad (3.13)$$

which is most often referenced in the literature, yet it is less exact than Eq. 3.12.

The following Matlab code highlights the difference between those two equations.

```
%==== Westergaard

clear all
clc
figure (1)
clf
figure (2)
clf
figure (3)
clf
%==== Water data
w=9.8E3;
k=2.068E9;

g=9.8;

K=8011.4;
%==== Approximate formula
hh=100;
yy=[0:1:hh];
TT=8;
west_app1=7*w*sqrt(hh*(hh-yy))/(8*g);
mm=K*sqrt(hh*(hh-yy))*(1/sqrt(1-16*w*hh^2/(g*k*TT^2)))/g;
figure (1)
plot(west_app1,yy,'green')
hold on
plot(mm,yy,'red')
%==== C formula

T=[0.3:0.1:2]';
h=sqrt(g*k*T.^2/(16*w));
s=size(T);
s=s(1,1);

for i=1:s
y=[0:1:100]';
ss=size(y);
ss=ss(1,1);
for j=1:ss
m(i,j)=K*sqrt(y(ss)*(y(ss)-y(j)))*(1/sqrt(1-16*w*y(ss)^2/(g*k*T(i)^2)))/g;
TT(i,j)=T(i);
west_app(i,j)=7*w*sqrt(y(ss)*(y(ss)-y(j)))/(8*g);
end
M(i)=m(i,ss-1);
W(i)=west_app(i,ss-1);
figure (2)
hold on
plot3(m(i,:),TT(i,:),y,'red')
plot3(west_app(i,:),TT(i,:),y,'green')

xlabel('added mass (kg)')
ylabel('Period T')
zlabel('Elevation')
legend('Kumo','Westergaard Approximation')
grid
end
%hold on
%plot3(zeros(s),T,h,'blue')
figure (3)
plot(M,T,'red')
hold on
plot(W,T,'green')
legend('Kumo','Westergaard Approximation')
 xlabel('Added mass')
 ylabel('T')
title('h constant, added mass at the bottom of the dam')
grid
```

3.2 Zangar

Using an electric analog, Fig. 3.5, Zangar (1953) determined experimentally the hydrodynamic effect of horizontal earthquake action on dams having upstream faces with either constant or compound slopes.

The pressure is given by

$$P_{Hyd} = C\alpha wh \quad (3.14)$$

where P_{Hyd} is the increase in pressure (F/L^2 or M/LT^2), α is the intensity of the horizontal earthquake (a/g), w is

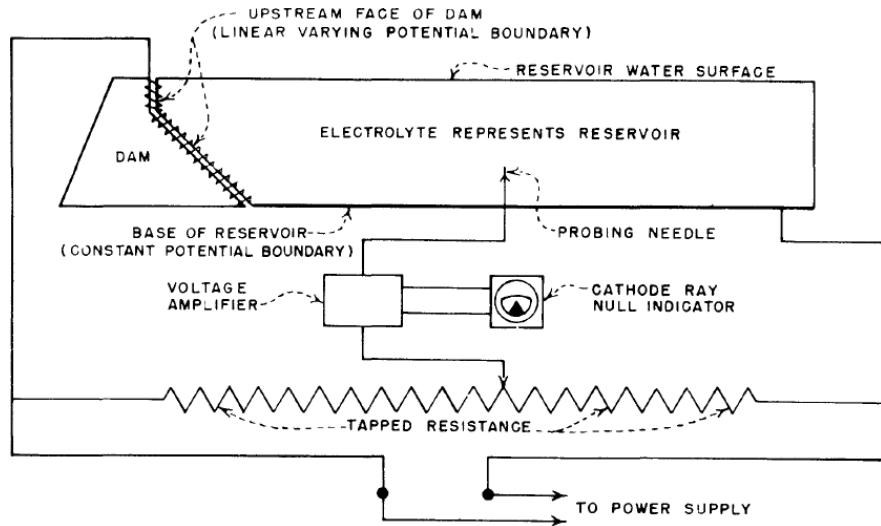


Figure 3.5: Electric Analog Tray Model used by Zangar (1953)

the mass density of water (M/L^3), h (L) is the height of the water reservoir, and C is a coefficient given by

$$C = \frac{C_m}{2} \left[\frac{y}{h} \left(2 - \frac{y}{h} \right) + \sqrt{\frac{y}{h} \left(2 - \frac{y}{h} \right)} \right] \quad (3.15)$$

and C_m is the maximum value of C given by Fig. 3.6.

Finally, the lumped mass will be given by

$$\gamma_{lump} = \frac{C \alpha w h}{\alpha g} \quad (3.16)$$

Fig. 3.7 shows the comparison between experimental and empirical curves.

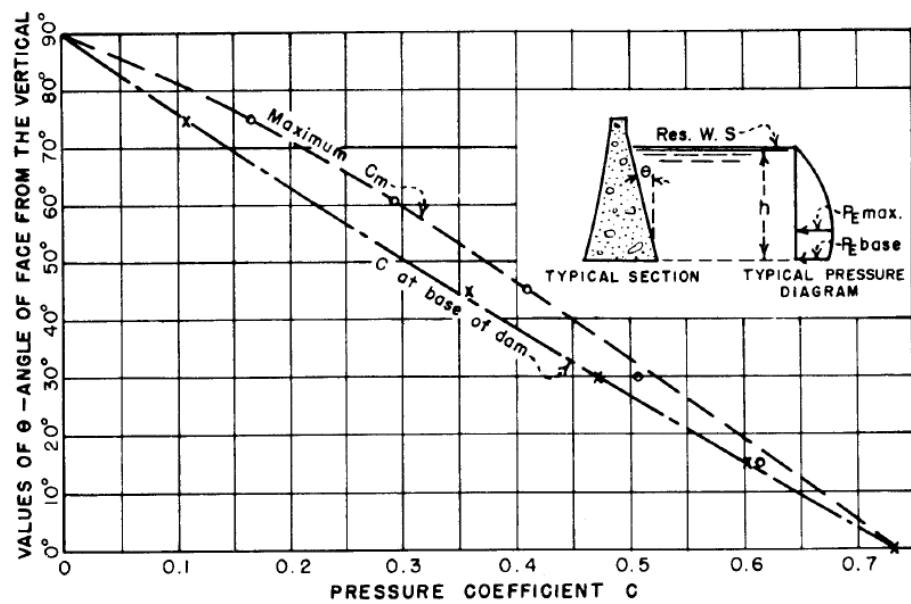


Figure 3.6: Increase Pressure Coefficients for Constant Sloping Faces (Zangar 1953)

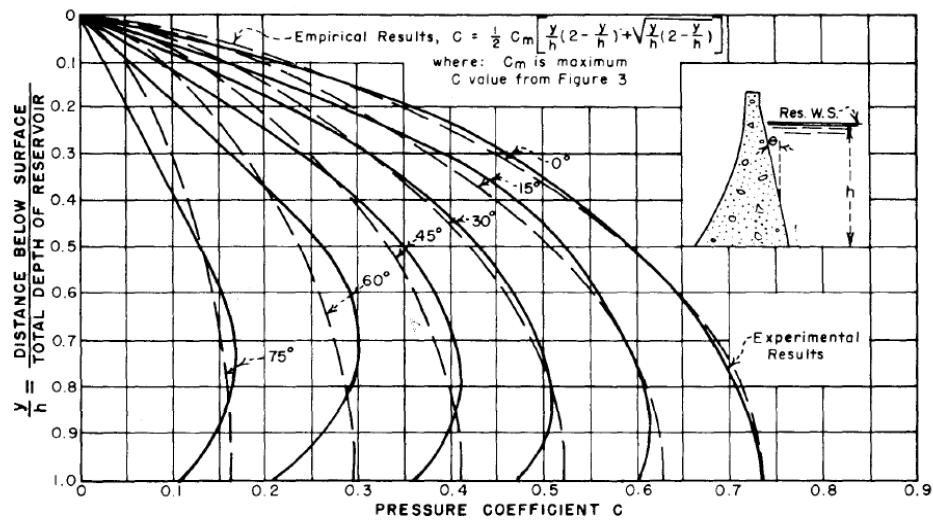


Figure 3.7: Pressure Coefficient Distribution Comparison of Experimental and Empirical Curves (Zangar 1953)

Chapter 4

NEARLY INCOMPRESSIBLE ELEMENTS

4.1 Consequences of Material Incompressibility

From elasticity we have the following fundamental relations

$$\varepsilon_V = \frac{dV}{V} = \varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz} \quad (4.1)$$

$$B = \frac{E}{3(1-2\nu)} \quad (4.2)$$

$$p = -B\varepsilon_V = -\frac{\sigma_{xx} + \sigma_{yy} + \sigma_{zz}}{3} \quad (4.3)$$

The stress strain relation is given by

$$\sigma_{ij} = \lambda u_{k,k} + 2G\varepsilon'_{ij} \quad (4.4)$$

where ε'_{ij} is the deviatoric strain

$$\varepsilon'_{ij} = \varepsilon_{ij} - \frac{\varepsilon_V}{3}\delta_{ij} \quad (4.5)$$

As ν approaches 0.5, resistance to volume change greatly increases assuming that the shear resistance remains constant

$$\frac{B}{G} = \frac{2(1+\nu)}{3(1-2\nu)} \quad (4.6)$$

also since

$$\lambda = \frac{2\nu G}{1-2\nu} \quad (4.7)$$

it is clear that as ν approaches 0.5, the stress becomes unbounded and we need to use an alternative formulation

$$\sigma_{ij} = \underbrace{-B\varepsilon_V}_{-p}\delta_{ij} + 2G\varepsilon'_{ij} \quad (4.8)$$

and now p becomes part of the solution as an additional unknown leading to a mixed formulation.

Table 4.1 gives the elastic properties of water and other engineering materials. It should be noted that shear

Material	E GPa	ν	G GPa	B GPa
Steel	207	0.25	82.8	138
Concrete	27.6	0.20	11.5	15.3
Water	0	0.50	0	2.1
“Water”	6.0×10^{-4}	0.49995	2.1×10^{-4}	2.1

Table 4.1: Elastic Properties of Steel, Concrete and Water, (Aslam et al. 2002)

modulus is zero, however under dynamic loading viscosity and boundary layer effects allow fluids to resist shear (Aslam et al. 2002).

4.2 Displacement Based Formulation

The stiffness matrix is given by

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega \quad (4.9)$$

in terms of G and B , the constitutive matrix is given by

$$\mathbf{D} = B \underbrace{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{D}_B} + G \underbrace{\begin{bmatrix} 4/3 & -2/3 & -2/3 & 0 & 0 & 0 \\ -2/3 & 4/3 & -2/3 & 0 & 0 & 0 \\ -2/3 & -2/3 & 4/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{D}_G} \quad (4.10)$$

where the first term corresponds to the volumetric state, and the second to the deviatoric one. Hence, we can rewrite

$$\mathbf{K} = B \int_{\Omega} \mathbf{B}^T \mathbf{D}_B \mathbf{B} d\Omega + G \int_{\Omega} \mathbf{B}^T \mathbf{D}_G \mathbf{B} d\Omega \quad (4.11)$$

or

$$(G\mathbf{K}_G + B\mathbf{K}_B)\bar{\mathbf{u}} = \bar{\mathbf{p}} \quad (4.12)$$

as ν approaches 0.5, the bulk modulus B approaches infinity, and therefore $B\mathbf{K}_B$ acts as a penalty matrix that enforces the constraint of incompressibility, we will have numerical problems, and finally the mesh “locks” unless \mathbf{K}_B is singular.

\mathbf{K}_B is made singular by reducing the order of numerical quadrature employed to evaluate \mathbf{K}_B below that “normal” used.

Hence a selective reduced numerical integration is performed, regular one for \mathbf{K}_G , and reduced for \mathbf{K}_B .

Note: Because the linear triangular element uses a one point numerical integration scheme, it can not be used for fluid element.

Finally, we can have a viscous damping matrix \mathbf{C}

$$\mathbf{C} = \int_{\Omega} \mathbf{B}^T \mathbf{D}_G \mathbf{B} \mu d\Omega \quad (4.13)$$

where μ is the dynamic coefficient of viscosity.

Chapter 5

FOUNDATION MODELLING

5.1 Wave Equation

Considering an infinitesimal element at rest, with elastic modulus E , and mass density ρ , we seek to determine the governing differential equation under dynamic condition.

- Thinking in terms of equilibrium of forces, it is more appealing to invoke D'Alembert's principle of **dynamic equilibrium** rather than Newton's second law of motion. This principle is based on the notion of a fictitious **inertia force**, equal to the product of mass times acceleration and acting in a direction opposite to the acceleration. Hence, the element force equilibrium requirements of a typical differential element are, using d'Alembert's principle, which states that with inertia forces included, a system is in equilibrium at each time instant.

$$\frac{\partial \sigma_{xx}}{\partial x} dx - \rho \frac{\partial^2 u_x}{\partial t^2} dx = 0 \quad (5.1)$$

Since $\sigma_{xx} = \lambda \varepsilon_{xx} = \lambda \frac{\partial u}{\partial x}$, substituting, we obtain

$$\boxed{\frac{\partial^2 u}{\partial t^2} - V_p^2 \frac{\partial^2 u_x}{\partial x^2} = 0} \quad (5.2)$$

where $V_p = \sqrt{\frac{\lambda}{\rho}}$

The solution of this equation, for harmonic wave propagation in the positive x-direction, is

$$u(t, x) = U \left[\sin(\omega t - \frac{\omega x}{V_p}) + \cos(\omega t - \frac{\omega x}{V_p}) \right] \quad (5.3)$$

where ω is the arbitrary frequency of the harmonic motion. The velocity, $\frac{\partial u}{\partial t}$ of a particle at location x is

$$\dot{u}(t, x) = U \omega \left[\cos(\omega t - \frac{\omega x}{V_p}) - \sin(\omega t - \frac{\omega x}{V_p}) \right] \quad (5.4)$$

and the strain in the x direction is

$$\varepsilon(x, t) = \frac{\partial u}{\partial x} = -\frac{\dot{u}(x, t)}{V_p} \quad (5.5)$$

The corresponding stress is now

$$\sigma(x, t) = \lambda \varepsilon(x, t) = -V_p \rho \dot{u}(x, t) \quad (5.6)$$

Thus the compressive stress is equal to the force on a viscous damper with constant damping coefficient equal to $V_p \rho$ per unit area of boundary.

It can be easily shown that the shear wave radiation boundary condition parallel to a free boundary, is satisfied if damping value is equal to $V_s \rho$.

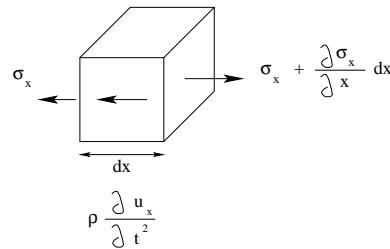


Figure 5.1: Infinitesimal Element Subjected to Elastic Wave

5.2 Viscous Boundary Conditions; Lysmer Model

When modelling a dynamic problem involving soil structure interaction, particular attention must be given to the soil boundary conditions. Ideally, infinite boundary conditions should be surrounding the excited zone, Fig. 5.2. Propagation of energy will occur from the interior to the exterior region. Since the exterior region is nonreflecting, it absorbs all the incoming energy. Yet, in a finite element analysis, we are constrained into applying finite size boundaries for the foundations. Those boundaries in turn will reflect the elastic waves which is contrary to the physics of the problem.

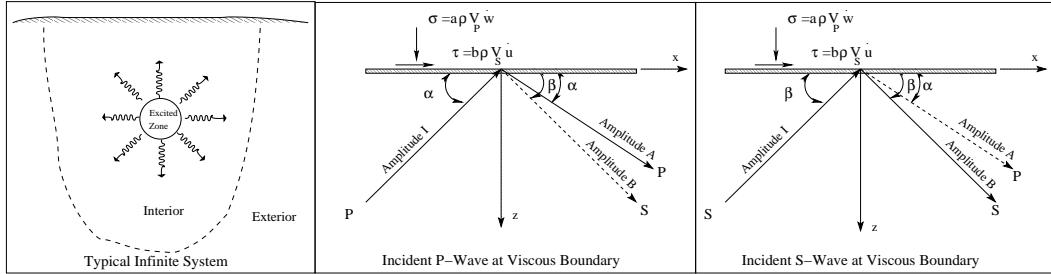


Figure 5.2: Elastic Waves in an Infinite Medium

Lysmer and Kuhlemeyer (1969) was the first to investigate this problem, and he proposed a model through which the boundary of a finite element mesh is surrounded by (energy absorbent) dashpots where

$$\sigma = a\rho V_P \dot{w} \quad (5.7-a)$$

$$\tau = b\rho V_S \dot{u} \quad (5.7-b)$$

where σ and τ are the normal and shear stresses respectively; \dot{w} and \dot{u} are the normal and tangential velocities; ρ is the mass density; V_S and V_P are the shear and pressure waves respectively given by

$$V_S = \sqrt{\frac{G}{\rho}} \quad (5.8-a)$$

$$V_P = \frac{1}{s} V_S \text{ where } s^2 = \frac{1-2\nu}{2(1-\nu)} \quad (5.8-b)$$

where G and ν are the shear modulus and the Poisson ratio respectively.

The directions of the incident and reflected waves are related through Snell's Law

$$\cos \beta = s \cos \alpha \quad (5.9)$$

Lysmer determined the ratio of the reflected energy to incident energy (of the P waves per unit time per unit area) as

$$\frac{E_r}{E_i} = A^2 + s \frac{\sin \beta}{\sin \alpha} B^2 \quad (5.10)$$

where a unit ratio corresponds to a perfect reflection (undesired), while a zero ratio corresponds to complete absorption (desired). A similar equation was determined for S waves.

In both cases, it was found that a viscous boundary defined by $a = b = 1$ is: a) 95% effective in absorbing S waves; and b) absorbs nearly all waves for $\alpha > 30^\circ$ (some reflection occurs at smaller angles).

Hence, in general dashpots should be placed around the boundary, Fig. 5.3.

5.3 Finite Element Implementation

5.3.1 Passive/Rigid Boundary; Lysmer

In this first approach, we indeed assume the boundaries of the foudations to be rigid, and applying Lysmer's model we adopt the discretization shown in Fig. 5.4.

where the spring stiffnesses are set equal to

$$K_{spring} = \frac{EA}{h} \quad (5.11)$$

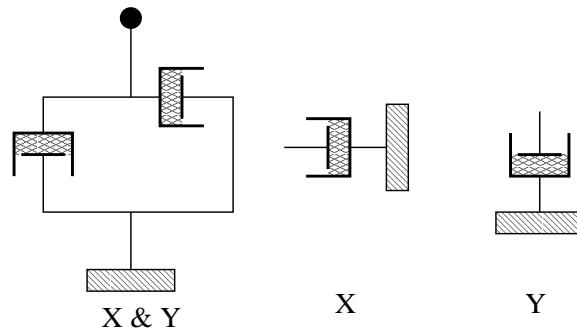


Figure 5.3: Dashpot Boundary Conditions

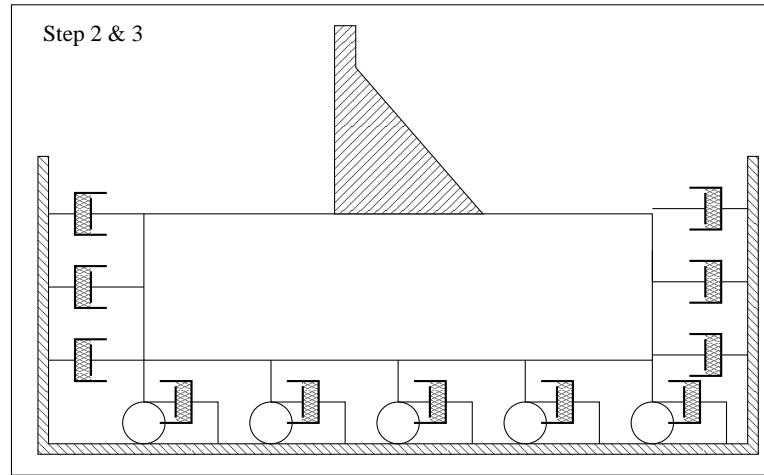


Figure 5.4: Foundation Model, Radiating Fixed Foundation

Step 1

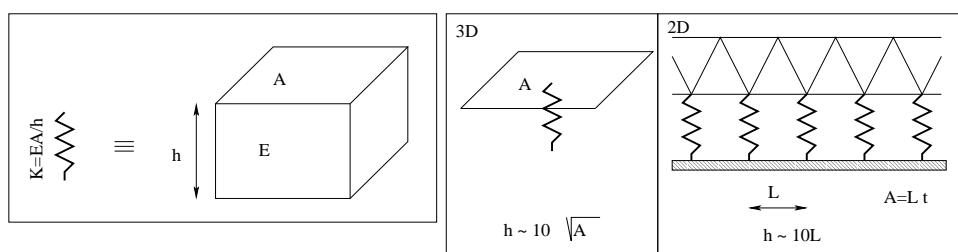


Figure 5.5: Equivalent Spring Stiffness

where E is the Young's modulus of the foundation, A the tributary area of the node connected to the spring, and h is a representative equivalent depth of the foundation, Fig. 5.5.

The analysis proceeds as follows:

1. Perform static analysis with all the body forces and hydrostatic one.
2. Through a restart, initiate a dynamic analysis from the preceding static one. We again apply all the loads for each time increment, in addition to the ground acceleration in the horizontal direction.

Kumo/Merlin are set up to greatly facilitate this analysis in both 2D and 3D.

5.3.1.1 Modeling

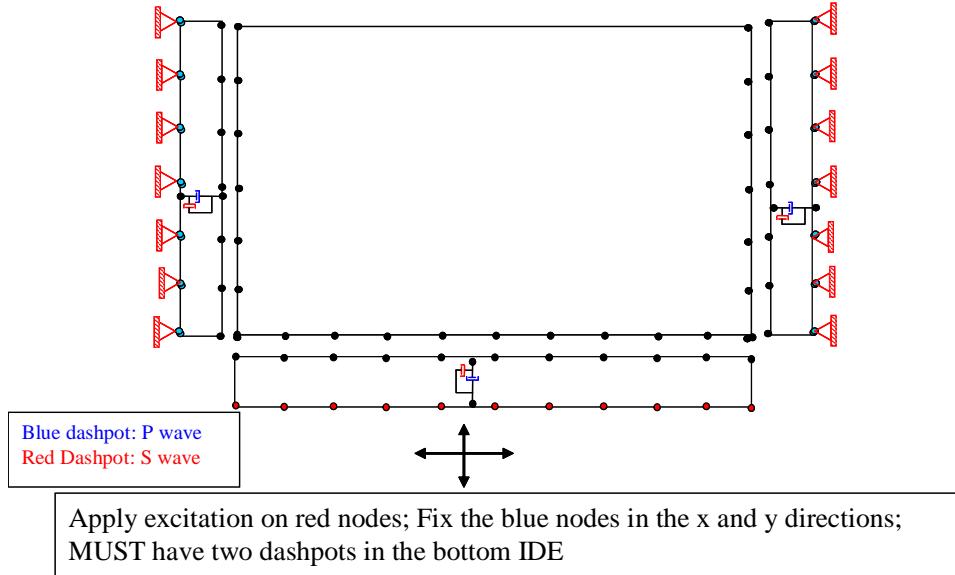


Figure 5.6: Lysmer Modeling, 2D, Modeling for Lateral and Vertical Excitation

5.3.1.2 Reservoir Model

5.3.2 Active/Flexible Boundary; Miura

Recognizing that in practice we do not have a rigid support for the foundation, but rather a flexible one, we need to account for this added variability, Fig. 5.10.

The methodology here adopted here is based on the work of (Miura and Toki 1987).

The governing equation for a dam foundation system in an infinite flexible medium is given by

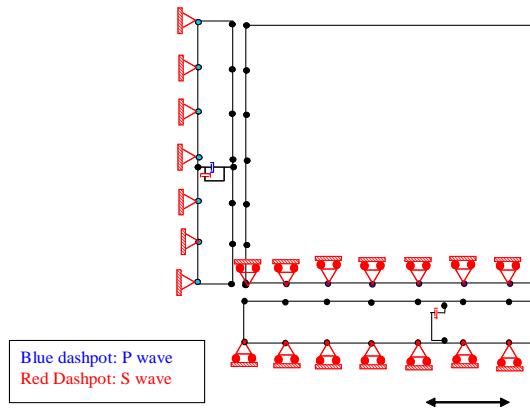
$$\begin{aligned} & [\mathbf{M}] \{\ddot{\mathbf{x}}\} + ([\mathbf{C}] + [\mathbf{C}_B] + [\mathbf{C}_L] + [\mathbf{C}_R]) \{\dot{\mathbf{x}}\} + [\mathbf{K}] \{\mathbf{x}\} \\ & = \{\mathbf{f}\} + [\mathbf{C}_L] \{\dot{\mathbf{x}}_L\} + [\mathbf{C}_R] \{\dot{\mathbf{x}}_R\} + [\mathbf{G}_{CL}] \{\dot{\mathbf{x}}_L\} + [\mathbf{G}_{CR}] \{\dot{\mathbf{x}}_R\} + [\mathbf{G}_L] \{\mathbf{x}_L\} + [\mathbf{G}_R] \{\mathbf{x}_R\} \end{aligned} \quad (5.12)$$

This equation can be rewritten as

$$\left[\begin{array}{cc|cc} M_{II} & M_{IB} & M_{IL} & M_{IR} \\ M_{BI} & M_{BB} & M_{BL} & M_{BR} \\ \hline M_{LI} & M_{LB} & M_{LL} & 0 \\ M_{RI} & M_{RB} & 0 & M_{RR} \end{array} \right] \left\{ \begin{array}{c} \dot{\mathbf{x}}_B \\ \dot{\mathbf{x}}_L \\ \dot{\mathbf{x}}_R \end{array} \right\} + \left[\begin{array}{cc|cc} C_{II} & C_{IB} & C_{IL} & C_{IR} \\ C_{BI} & C_{BB} & C_{BL} & C_{BR} \\ \hline C_{LI} & C_{LB} & C_{LL} & 0 \\ C_{RI} & C_{RB} & 0 & C_{RR} \end{array} \right] \left\{ \begin{array}{c} \dot{\mathbf{x}}_B \\ \dot{\mathbf{x}}_L \\ \dot{\mathbf{x}}_R \end{array} \right\} + \left[\begin{array}{cc|cc} K_{II} & K_{IB} & K_{IL} & K_{IR} \\ K_{BI} & K_{BB} & K_{BL} & K_{BR} \\ \hline K_{LI} & K_{LB} & K_{LL} & 0 \\ K_{RI} & K_{RB} & 0 & K_{RR} \end{array} \right] \left\{ \begin{array}{c} \mathbf{x}_B \\ \mathbf{x}_L \\ \mathbf{x}_R \end{array} \right\} \\ = \left\{ \begin{array}{c} f_B \\ f_L \\ f_R \end{array} \right\} + \left[\begin{array}{c|cc} 0 & C_{LL} + G_{CL} \\ \hline C_{RR} + G_{CR} & G_L & G_R \end{array} \right] \left\{ \begin{array}{c} \dot{\mathbf{x}}_I \\ \dot{\mathbf{x}}_L \\ \dot{\mathbf{x}}_R \end{array} \right\} + \left[\begin{array}{c|cc} 0 & G_L & G_R \\ \hline G_L & G_R & \end{array} \right] \left\{ \begin{array}{c} \mathbf{x}_B \\ \mathbf{x}_L \\ \mathbf{x}_R \end{array} \right\} \end{aligned}$$

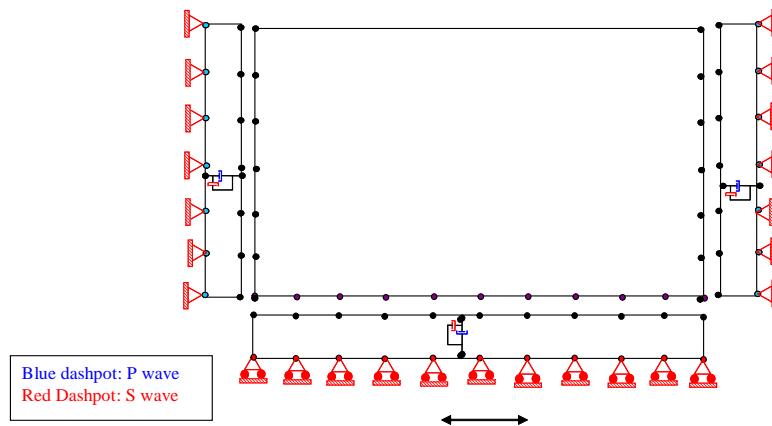
where $[\mathbf{M}]$ is the mass matrix, $[\mathbf{C}]$ damping matrix, $[\mathbf{K}]$ stiffness matrix and subscripts I, B, L, R refer to interior, bottom, left and right nodes; $\{\mathbf{x}\}, \{\dot{\mathbf{x}}\}, \{\ddot{\mathbf{x}}\}$ are the nodal displacements, velocities and accelerations.

Lateral Excitation; Ir Kumo; work



Apply horizontal excitation on red nodes (bottom of IDE) direction. Fix the blue nodes in the x and y directions;

Figure 5.7: Lysmer Modeling, 2D, Modeling for Lateral Excitation



Apply horizontal excitation on red nodes (bottom of IDE); fix red nodes in the y direction. Fix the blue nodes in the x and y directions; No need to fix internal nodes if there is a vertical dashpot.

Figure 5.8: Lysmer Modeling, 2D, Alternative Modeling for Lateral Excitation

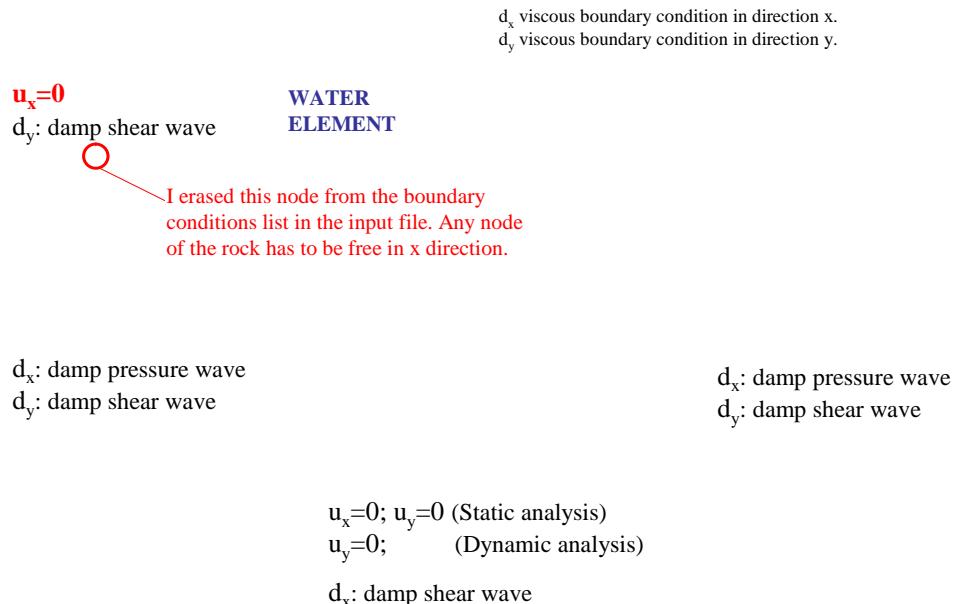


Figure 5.9: Reservoir Model Boundary Conditions

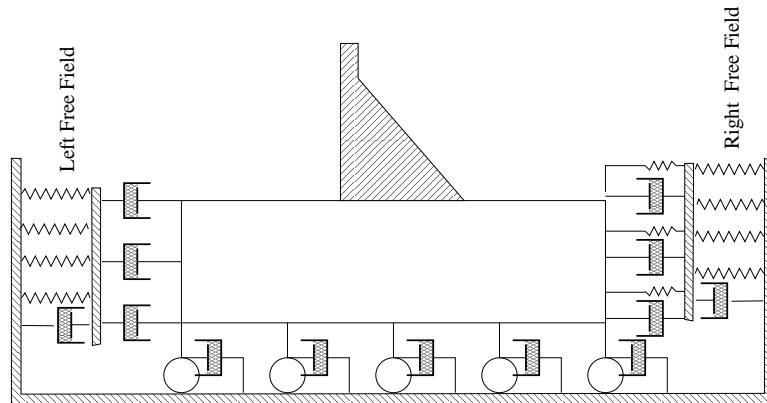


Figure 5.10: Foundation Model, Radiating Flexible Foundation

$[\mathbf{C}_B]$ is Lysmer (dashpot) viscous boundary conditions at the bottom (tuned to shear wave for lateral excitation and to pressure waves for vertical excitation).

$$[\mathbf{C}_B] = \frac{\rho L}{2} \begin{bmatrix} V_H & 0 & 0 & 0 \\ 0 & V_V & 0 & 0 \\ 0 & 0 & V_H & 0 \\ 0 & 0 & 0 & V_V \end{bmatrix} \quad (5.13-a)$$

$$\begin{cases} V_H = V_S \cos \theta + V_P \sin \theta \\ V_V = V_P \cos \theta + V_S \sin \theta \end{cases} \quad (5.13-b)$$

$[\mathbf{C}_L], [\mathbf{C}_R]$ are Lysmer (dashpot) left and right boundary conditions, tuned to pressure wave for lateral excitation and shear waves for vertical excitation. $[\mathbf{G}_L], [\mathbf{G}_R]$ are the boundary stiffness matrices associated with the displacement of the free field.

$$[\mathbf{G}] = \frac{1}{2} \begin{bmatrix} 0 & -\lambda & 0 & \lambda \\ -\mu & 0 & \mu & 0 \\ 0 & -\lambda & 0 & \lambda \\ -\mu & 0 & \mu & 0 \end{bmatrix} \quad (5.14)$$

where λ and μ are the Lame parameters, $\lambda = \frac{\nu E}{(1-2\nu)(1+\nu)} = K - \frac{2}{3}G$, and $\mu = \frac{E}{2(1+\nu)} = G$. For symmetric foundation ($\mathbf{x}_L = -\mathbf{x}_R$) we can ignore this term. $[\mathbf{G}_{CL}], [\mathbf{G}_{CR}]$ are the boundary damping matrices associated with the free field. Their effect do also cancel out for symmetric cases.

Hence, for symmetric boundary conditions, we can ignore $[\mathbf{G}_R], [\mathbf{G}_L], [\mathbf{G}_{CR}], [\mathbf{G}_{CL}], \mathbf{x}_R, \mathbf{x}_L$, and the resulting governing partial differential equation to be solved is reduced to:

$$[\mathbf{M}] \{ \ddot{\mathbf{x}} \} + ([\mathbf{C}] + [\mathbf{C}_b] + [\mathbf{C}_L] + [\mathbf{C}_R]) \{ \dot{\mathbf{x}} \} + [\mathbf{K}] \{ \mathbf{x} \} = \{ \mathbf{f} \} + [\mathbf{C}_L] \{ \dot{\mathbf{x}}_L \} + [\mathbf{C}_R] \{ \dot{\mathbf{x}}_R \} \quad (5.15)$$

or

$$\left[\begin{array}{c|c|c|c} M_{II} & M_{IB} & M_{IL} & M_{IR} \\ \hline M_{BI} & M_{BB} & M_{BL} & M_{BR} \\ \hline M_{LI} & M_{LB} & M_{LL} & 0 \\ \hline M_{RI} & M_{RB} & 0 & M_{RR} \end{array} \right] \left\{ \begin{array}{c} \dot{\mathbf{x}}_I \\ \dot{\mathbf{x}}_B \\ \dot{\mathbf{x}}_L \\ \dot{\mathbf{x}}_R \end{array} \right\} + \left[\begin{array}{c|c|c|c} C_{II} & C_{IB} & C_{IL} & C_{IR} \\ \hline C_{BI} & C_{BB} & C_{BL} & C_{BR} \\ \hline C_{LI} & C_{LB} & C_{LL} & 0 \\ \hline C_{RI} & C_{RB} & 0 & C_{RR} \end{array} \right] \left\{ \begin{array}{c} \mathbf{x}_I \\ \mathbf{x}_B \\ \mathbf{x}_L \\ \mathbf{x}_R \end{array} \right\} + \left[\begin{array}{c|c|c|c} K_{II} & K_{IB} & K_{IL} & K_{IR} \\ \hline K_{BI} & K_{BB} & K_{BL} & K_{BR} \\ \hline K_{LI} & K_{LB} & K_{LL} & 0 \\ \hline K_{RI} & K_{RB} & 0 & K_{RR} \end{array} \right] \left\{ \begin{array}{c} \mathbf{x}_I \\ \mathbf{x}_B \\ \mathbf{x}_L \\ \mathbf{x}_R \end{array} \right\}$$

$$= \left\{ \begin{array}{c} \mathbf{f}_B \\ 0 \\ \mathbf{f}_L \\ \mathbf{f}_R \end{array} \right\} + \left[\begin{array}{c|c|c|c} & 0 & C_{LL} & C_{RR} \\ \hline & \hline & C_{LL} & C_{RR} \end{array} \right] \left\{ \begin{array}{c} \dot{\mathbf{x}}_I \\ \dot{\mathbf{x}}_B \\ \dot{\mathbf{x}}_L \\ \dot{\mathbf{x}}_R \end{array} \right\}$$

In order to solve this equation, we still need some quantities on the right hand side of the equation, namely $\dot{\mathbf{x}}_L$ and $\dot{\mathbf{x}}_R$. These can be obtained from two separate (one if we take advantage of symmetry) analyses of the free field which can be discretized as shown in Fig. 5.11. We note the vertical restraint for lateral excitation, and the lateral restraint

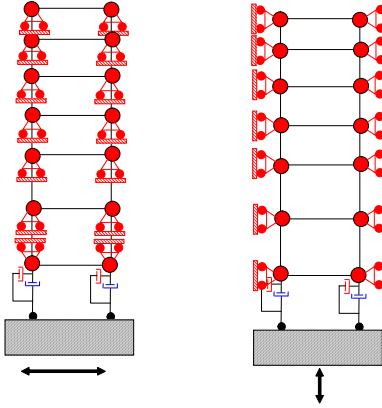


Figure 5.11: Finite Element Discretization of the free field

for vertical excitation in order to respect the far field boundary conditions. Thus the governing differential equations for these analyses are

$$[\mathbf{M}_L] \{ \ddot{\mathbf{x}}_L? \} + [\mathbf{C}_L] \{ \dot{\mathbf{x}}_L? \} + [\mathbf{K}_L] \{ \mathbf{x}_L? \} = \{ \mathbf{f}_L \checkmark \} \quad (5.16-a)$$

$$[\mathbf{M}_R] \{ \ddot{\mathbf{x}}_R? \} + [\mathbf{C}_R] \{ \dot{\mathbf{x}}_R? \} + [\mathbf{K}_R] \{ \mathbf{x}_R? \} = \{ \mathbf{f}_R \checkmark \} \quad (5.16-b)$$

from which we solve for $\dot{\mathbf{x}}_L$ and $\dot{\mathbf{x}}_R$.

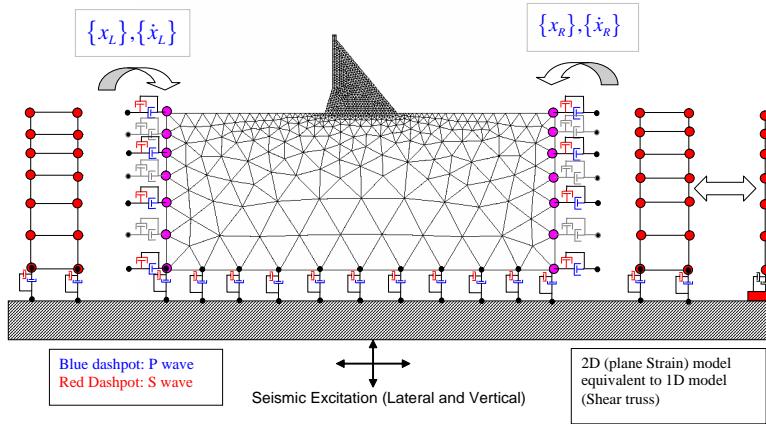


Figure 5.12: Finite Element Discretization of Dam Foundation in Account of Free Field Velocities

Once the free field velocities have been obtained, they can in turn be used in the full 2D analysis of the dam/foundation discretization shown in 5.12.

$$\begin{bmatrix}
 M_{II} & M_{IB} & M_{IL} & M_{IR} \\
 M_{BI} & M_{BB} & M_{BL} & M_{BR} \\
 M_{LI} & M_{LB} & M_{LL} & 0 \\
 M_{RI} & M_{RB} & 0 & M_{RR}
 \end{bmatrix}
 \begin{Bmatrix}
 \dot{x}_I? \\
 \dot{x}_B? \\
 \dot{x}_L? \\
 \dot{x}_R?
 \end{Bmatrix}
 +
 \begin{bmatrix}
 C_{II} & C_{IB} & C_{IL} & C_{IR} \\
 C_{BI} & C_{BB} & C_{BL} & C_{BR} \\
 C_{LI} & C_{LB} & C_{LL} & 0 \\
 C_{RI} & C_{RB} & C_{RL} & C_{RR}
 \end{bmatrix}
 \begin{Bmatrix}
 \dot{x}_B? \\
 \dot{x}_L? \\
 \dot{x}_R?
 \end{Bmatrix}
 +
 \begin{bmatrix}
 K_{II} & K_{IB} & K_{IL} & K_{IR} \\
 K_{BI} & K_{BB} & K_{BL} & K_{BR} \\
 K_{LI} & K_{LB} & K_{LL} & 0 \\
 K_{RI} & K_{RB} & K_{RL} & K_{RR}
 \end{bmatrix}
 \begin{Bmatrix}
 x_I? \\
 x_B? \\
 x_L? \\
 x_R?
 \end{Bmatrix}
 =
 \begin{Bmatrix}
 f_B \\
 f_L \\
 f_R
 \end{Bmatrix}
 +
 \begin{bmatrix}
 0 & 0 & C_{LL} & C_{RR} \\
 0 & 0 & C_{LR} & C_{RL}
 \end{bmatrix}
 \begin{Bmatrix}
 \dot{x}_I? \\
 \dot{x}_B? \\
 \dot{x}_L? \\
 \dot{x}_R?
 \end{Bmatrix}$$

5.3.2.1 Finite Element Implementation

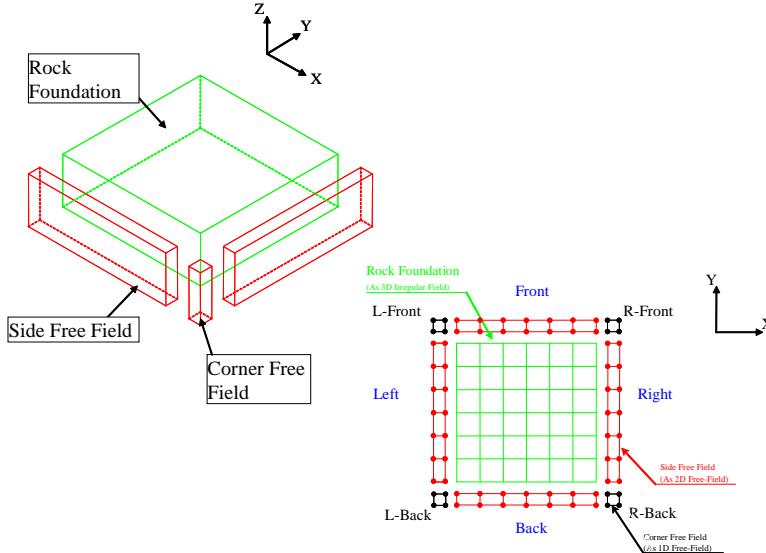


Figure 5.13: Finite Element Discretization of the free field

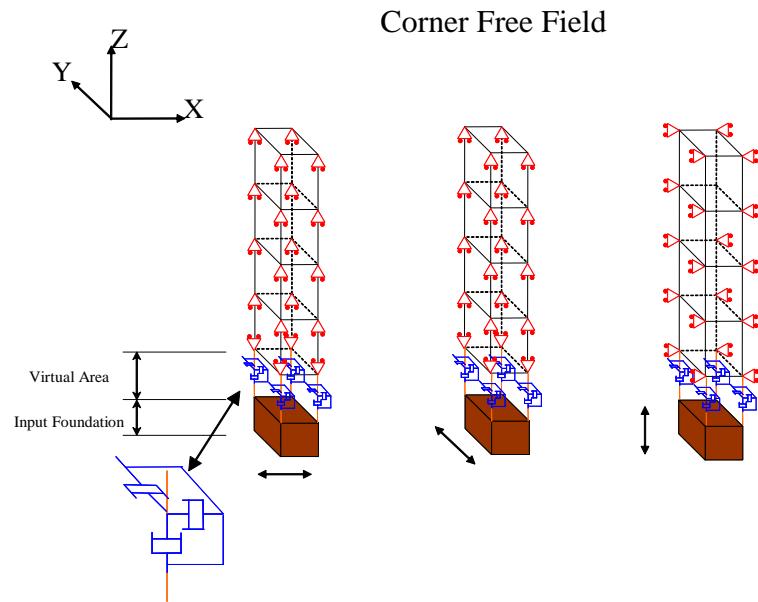


Figure 5.14: Finite Element Discretization of the Corner free field

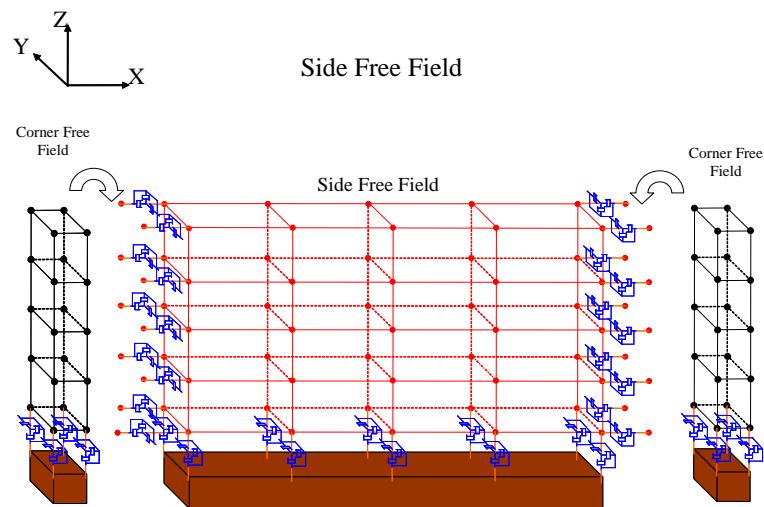


Figure 5.15: Finite Element Discretization of the Side free field

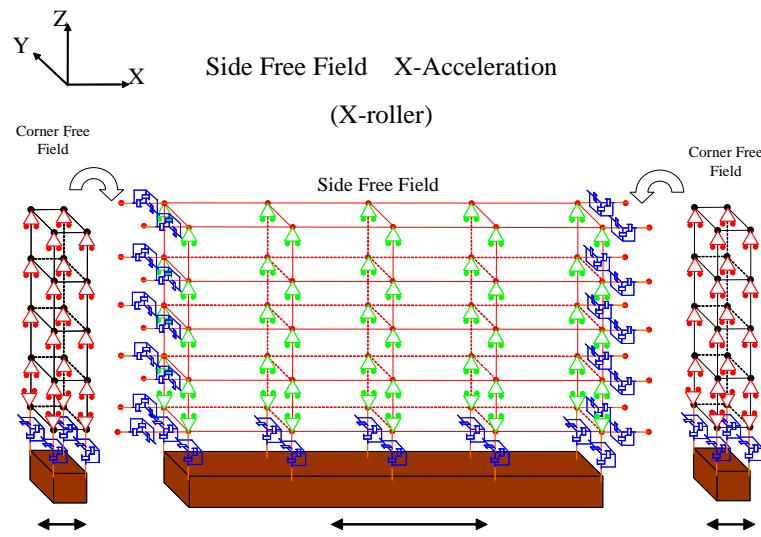


Figure 5.16: Finite Element Discretization of the Side free field, X Acceleration

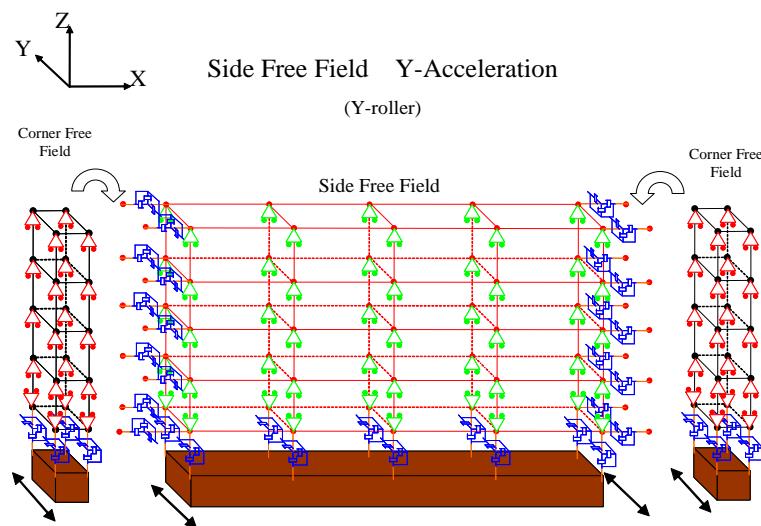


Figure 5.17: Finite Element Discretization of the Side free field, Y Acceleration

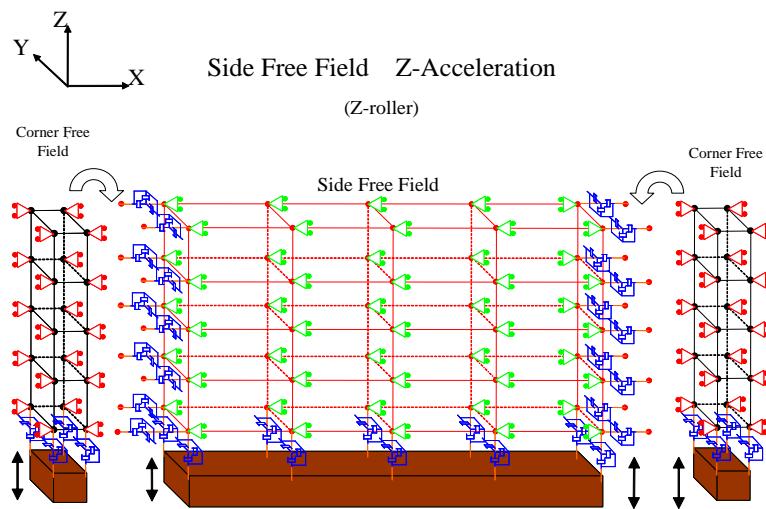


Figure 5.18: Finite Element Discretization of the Side free field, Z Accelreation

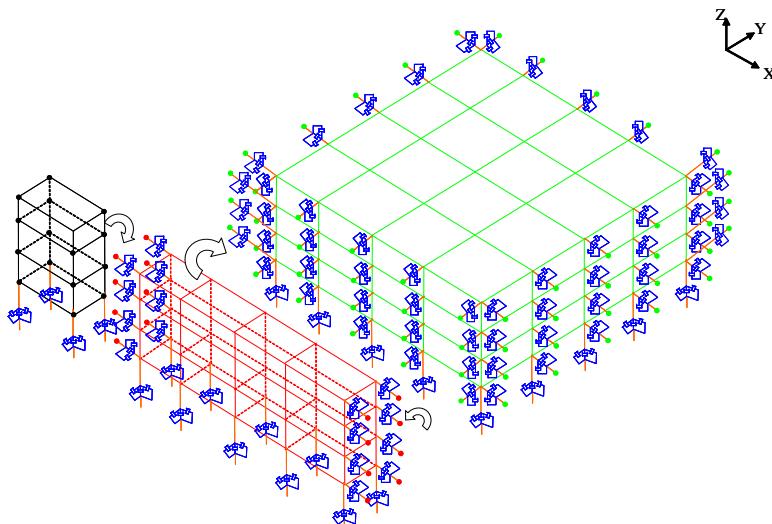


Figure 5.19: Finite Element Discretization of the free field; Transfer of Velocities

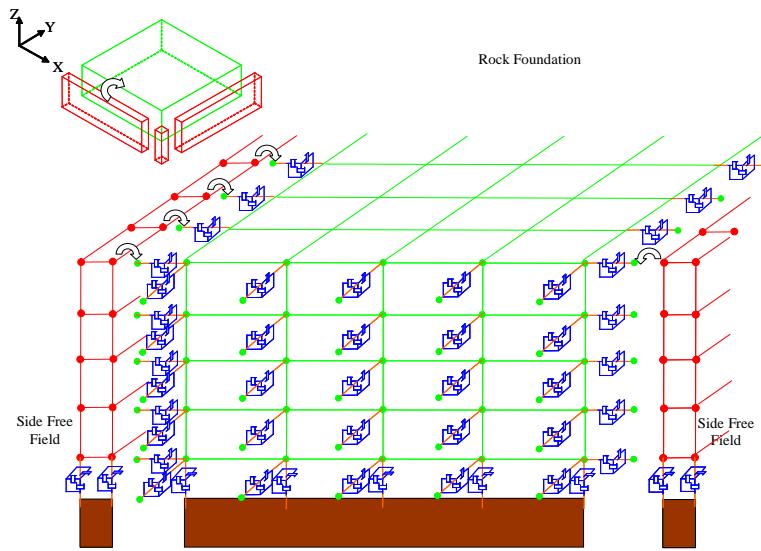


Figure 5.20: Finite Element Discretization of the free field; Rock Foundation

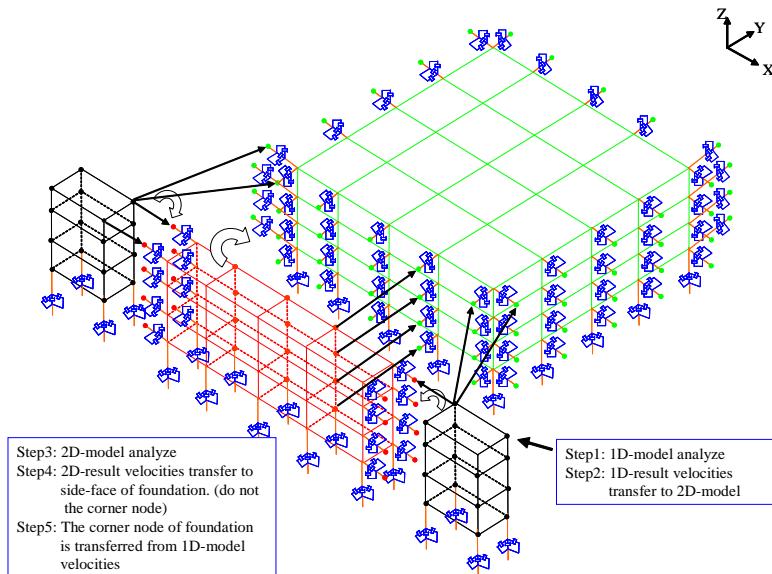


Figure 5.21: Finite Element Discretization of the free field; Outline of Procedure

Chapter 6

DECONVOLUTION

6.1 Introduction

Seismic events originate through tectonic slips and elastic waves (p and s) traveling through rock/soil foundation up to the surface. Hence, the seismographs (usually installed at the foot of the dam) record only the manifestation of the event.

On the other hand, modelling the foundation is essential for proper and comprehensive analysis of the dam, and as such the seismic excitation will have to be applied at the base of the foundation.

However, Fig. 6.1, if we were to apply at the base the accelerogram recorded on the surface $I(t)$, the output signal $A(t)$ at the surface will be different than the one originally recorded (unless we have rigid foundation).

Hence, the accelerogram recorded on the surface must be deconvolved into a new one $I'(t)$, such that when the new signal is applied at the base of the foundation, the computed signal at the dam base matches the one recorded by the accelerogram.

6.2 Fourier Transform

Fourier transforms enables us to transfer a signal from the time domain to the frequency domain.

Hence, the FFT takes us from the time domain to the frequency domain through the following equation:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi\omega t} dt \quad (6.1)$$

$$x(t) \xrightarrow{\text{FFT}} X(\omega) \quad (6.2)$$

while the inverse FFT takes us back from the frequency domain to the time domain through:

$$x(t) = \int_{-\infty}^{\infty} X(\omega) e^{i2\pi\omega t} d\omega \quad (6.3)$$

$$X(\omega) \xrightarrow{\text{FFT}^{-1}} x(t) \quad (6.4)$$

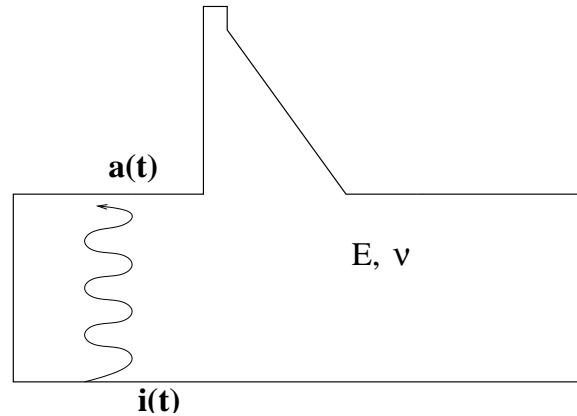


Figure 6.1: Deconvolution

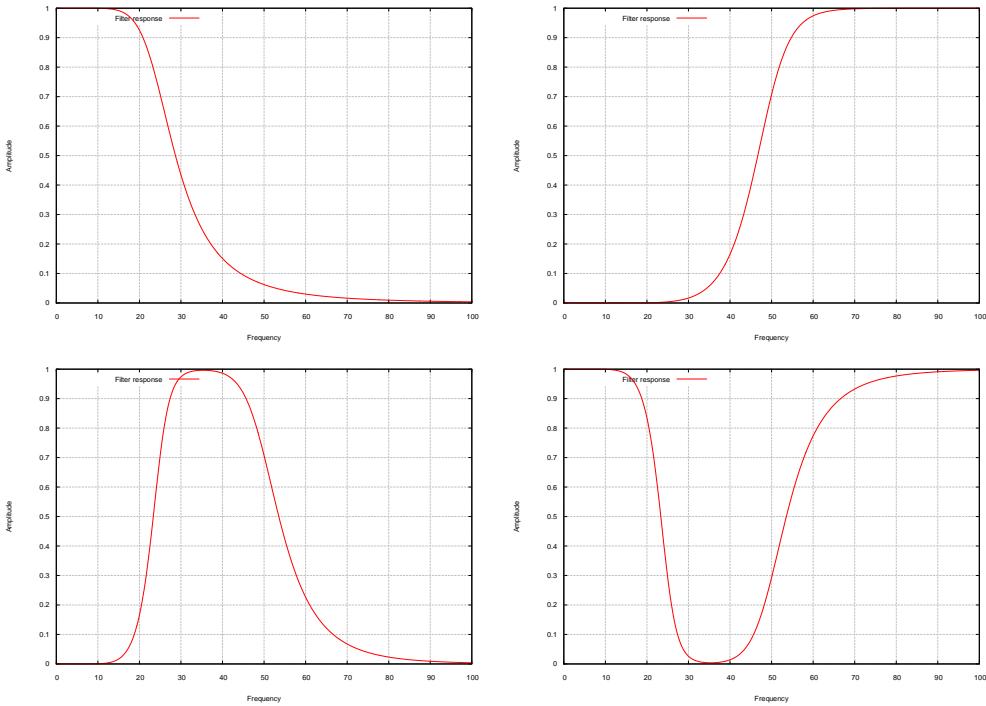


Figure 6.2: Low Pass (25); High Pass (50); Band Pass (25-50); Band Stop (25-50) Filters, $N = 4$

6.3 Butterworth Filter

Spider has the following filters implemented in its Deconvolution feature, Fig. 6.2-6.3.

$$|H(j\omega)|^2 = \begin{cases} \text{Low pass} & \frac{1}{1 + (\frac{\omega}{\omega_L})^{2n}} \\ \text{High pass} & \frac{1}{1 + (\frac{\omega}{\omega_U})^{2n}} \\ \text{Band pass} & \frac{1}{1 + (\frac{\omega}{\omega_L})^{2n}} \frac{1}{1 + (\frac{\omega_{upper}}{\omega})^{2n}} \\ \text{Band stop} & \frac{1}{1 + (\frac{\omega_L}{\omega})^{2n}} \frac{1}{1 + (\frac{\omega_{lower}}{\omega})^{2n}} \end{cases} \quad (6.5)$$

where ω , ω_L , ω_U and n are the frequency, the lower and upper filter frequency, and the order of the filter respectively.

6.4 Transfer Function

In dynamic event, we can define an input record $i(t)$ which is amplified by $h(t)$ resulting in an output signal $o(t)$, Fig. 6.4. Similarly, the operation can be defined in the frequency domain. This output to input relationship is of major importance in many disciplines.

The transfer function is the Laplace transform of the output divided by the Laplace transform of the input.

Hence, in 1D, we can determine the transfer function as follows:

1. $i(t) \xrightarrow{\text{FFT}} I(\omega)$
2. $o(t) \xrightarrow{\text{FFT}} O(\omega)$
3. Transfer Function is $TF_{I-O} = O(\omega)/I(\omega)$

6.5 Deconvolution

6.5.1 1-D

Extending our discussion one step further, we introduce the concept of deconvolution which addresses the dilemma posed above, and will now require one (or more) finite element analyses.

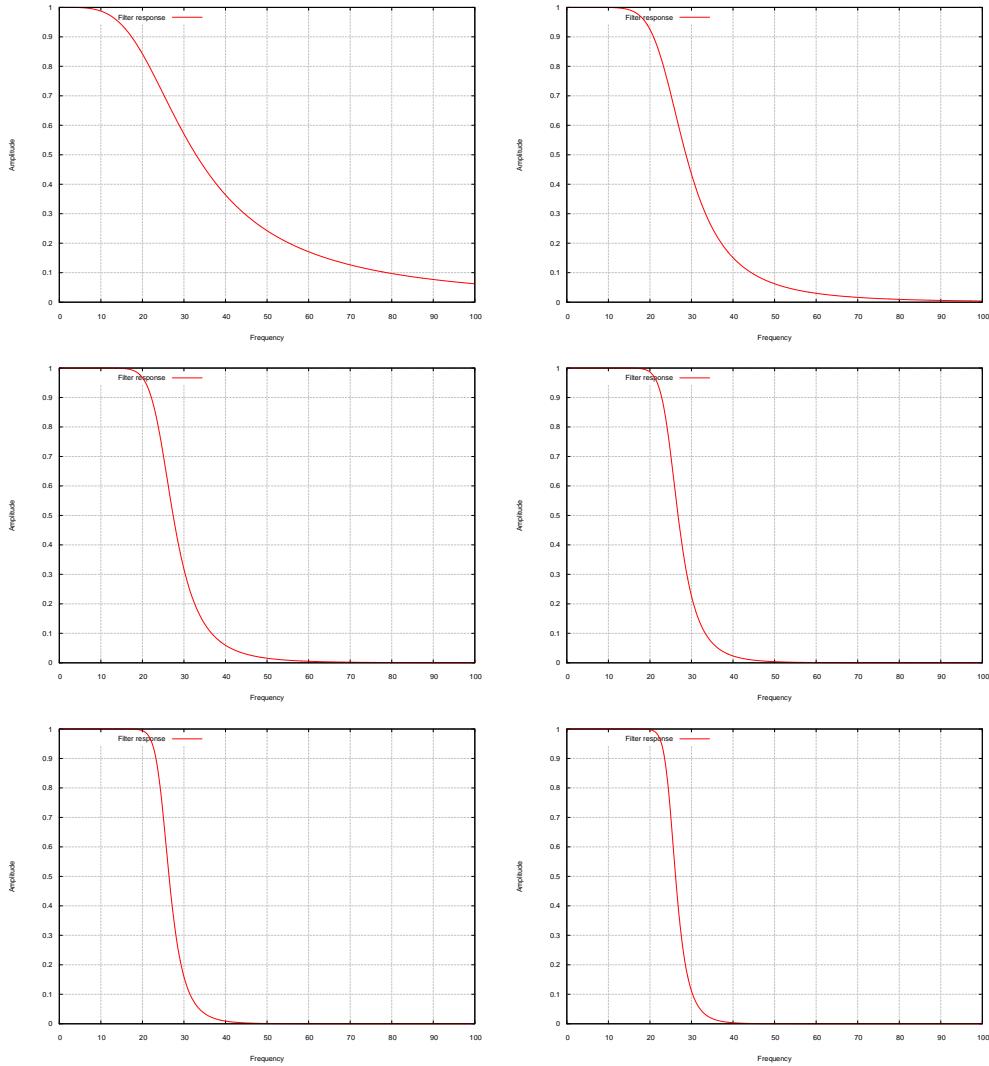
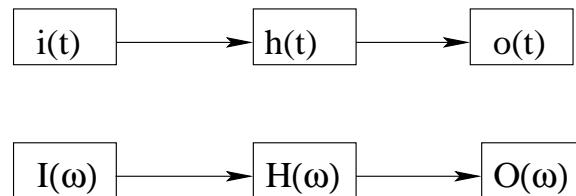
Figure 6.3: Low Pass (25) Filter; $N = 2, 4, 6, 8, 10, 12$ 

Figure 6.4: Transfer Function

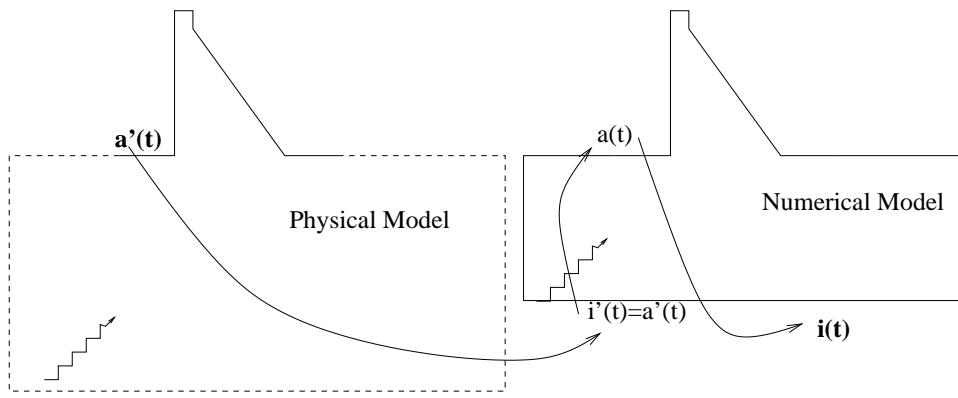


Figure 6.5: Deconvolution

With reference to Fig. 6.5

1. We record the earthquake induced acceleration on the surface $a'(t)$. and apply it as $i'(t)$ at the base of the foundation.
2. Perform a transient finite element analysis.
3. Determine the surface acceleration $a(t)$ (which is obviously different from $i(t)$).
4. Compute:

$$i'(t) \xrightarrow{\text{FFT}} I'(\omega) = A'(\omega) \quad (6.6-a)$$

$$a(t) \xrightarrow{\text{FFT}} A(\omega) \quad (6.6-b)$$

5. Compute transfer function from base to surface as $TF_{I'-A} = A(\omega)/I'(\omega)$.

6. Compute the inverse transfer function $TF_{I'-A}^{-1}$.

7. Determine the updated excitation record in the frequency domain

$$I(\omega) = TF_{I'-A}^{-1} A'(\omega) = \frac{I'(\omega)}{A(\omega)} A'(\omega) \quad (6.6-c)$$

8. Determine the updated excitation in the time domain

$$i(t) \xrightarrow{\text{FFT}^{-1}} I(\omega) \quad (6.6-d)$$

6.5.2 3-D

In 3-D applications, the transfer function is a 3x3 matrix, each row corresponds to the response to an excitation in a given direction, and each column corresponds to the response in a given direction. Hence, three separate analysis must be performed $\begin{bmatrix} I'_x & I'_y & I'_z \end{bmatrix}$ and for each excitation, we must determine the three components of the surface acceleration. Then we will compute the 3D transfer function:

$$[TF] = \underbrace{\begin{bmatrix} TF_{xx} & TF_{xy} & TF_{xz} \\ TF_{yx} & TF_{yy} & TF_{yz} \\ TF_{zx} & TF_{zy} & TF_{zz} \end{bmatrix}}_{TF_{I'-A}} = \begin{bmatrix} \frac{A_{xx}(\omega)}{I'_x(\omega)} & \frac{A_{xy}(\omega)}{I'_x(\omega)} & \frac{A_{xz}(\omega)}{I'_x(\omega)} \\ \frac{A_{yx}(\omega)}{I'_y(\omega)} & \frac{A_{yy}(\omega)}{I'_y(\omega)} & \frac{A_{yz}(\omega)}{I'_y(\omega)} \\ \frac{A_{zx}(\omega)}{I'_z(\omega)} & \frac{A_{zy}(\omega)}{I'_z(\omega)} & \frac{A_{zz}(\omega)}{I'_z(\omega)} \end{bmatrix} \quad (6.5)$$

Hence, the excitation to be applied in the frequency domain is given by:

$$\begin{Bmatrix} I_x(\omega) \\ I_y(\omega) \\ I_z(\omega) \end{Bmatrix} = [TF]^{-1} \begin{Bmatrix} A'_x(\omega) \\ A'_y(\omega) \\ A'_z(\omega) \end{Bmatrix} \quad (6.6)$$

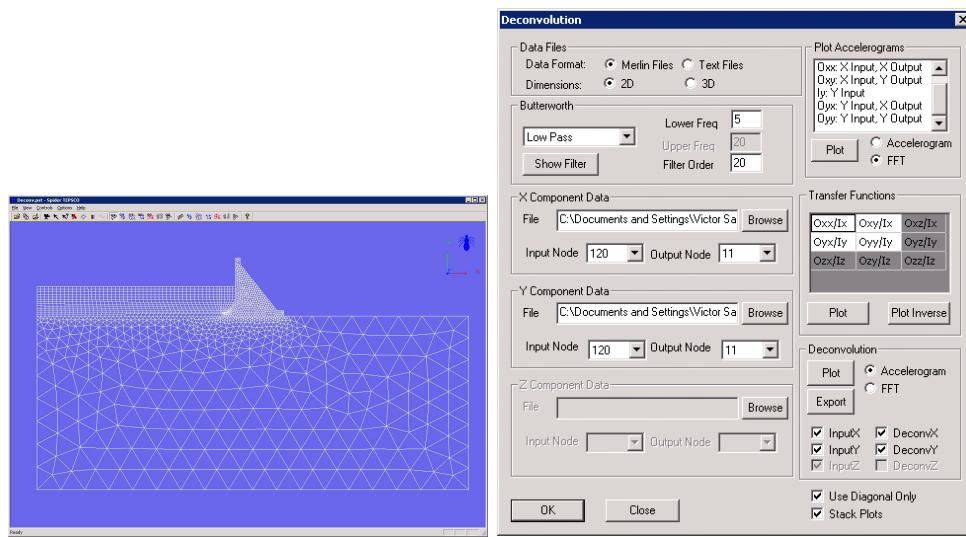


Figure 6.6: Finite Element Mesh Example for Deconvolution

while in the time domain it is

$$\begin{Bmatrix} I_x(\omega) \\ I_y(\omega) \\ I_z(\omega) \end{Bmatrix} \xrightarrow{\text{FFT}^{-1}} \begin{Bmatrix} I_x(t) \\ I_y(t) \\ I_z(t) \end{Bmatrix} \quad (6.7)$$

6.5.2.1 Simplification

The preceding 3D generalized procedure can be simplified if we were to ignore the off diagonal terms

$$[TF] = \begin{bmatrix} TF_{xx} & 0 & 0 \\ 0 & TF_{yy} & 0 \\ 0 & 0 & TF_{zz} \end{bmatrix} = \begin{bmatrix} \frac{A_{xx}(\omega)}{I'_x(\omega)} & 0 & 0 \\ 0 & \frac{A_{yy}(\omega)}{I'_y(\omega)} & 0 \\ 0 & 0 & \frac{A_{zz}(\omega)}{I'_z(\omega)} \end{bmatrix} \quad (6.8)$$

which will greatly simplify the inversion of the transfer function.

$$\begin{Bmatrix} I_x(\omega) \\ I_y(\omega) \\ I_z(\omega) \end{Bmatrix} = [TF_{I'-A}]^{-1} \begin{Bmatrix} A'_x(\omega) \\ A'_y(\omega) \\ A'_z(\omega) \end{Bmatrix} \quad (6.9)$$

$$\begin{Bmatrix} I_x(\omega) \\ I_y(\omega) \\ I_z(\omega) \end{Bmatrix} \xrightarrow{\text{FFT}^{-1}} \begin{Bmatrix} I_x(t) \\ I_y(t) \\ I_z(t) \end{Bmatrix} \quad (6.10)$$

6.5.3 Example

Considering the dam model shown in Fig. 6.6. The recorded ground excitation (at the base of the dam) is first applied at the base of the foundation and analyzed, Fig. 6.7. The Transfer functions and their inverse are shown in Fig. 6.9. The deconvolved signals are then computed, Fig. ?? The Input signal and the deconvolved ones are then compared, Fig. ?? Finally, we reanalyze the dam subjected to the deconvolved signal, and we compare the computed accelerations at the base of the dam with those recorded, Fig. 6.11.

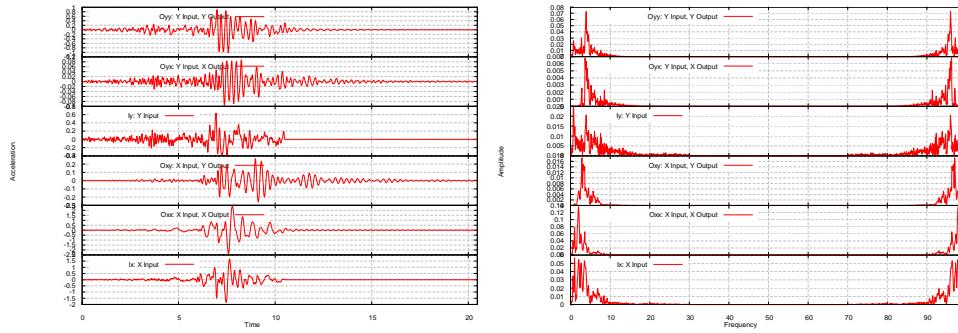


Figure 6.7: Accelerograms of the Input and Output

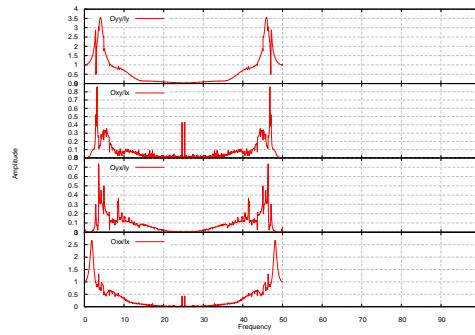


Figure 6.8: Transfer Functions

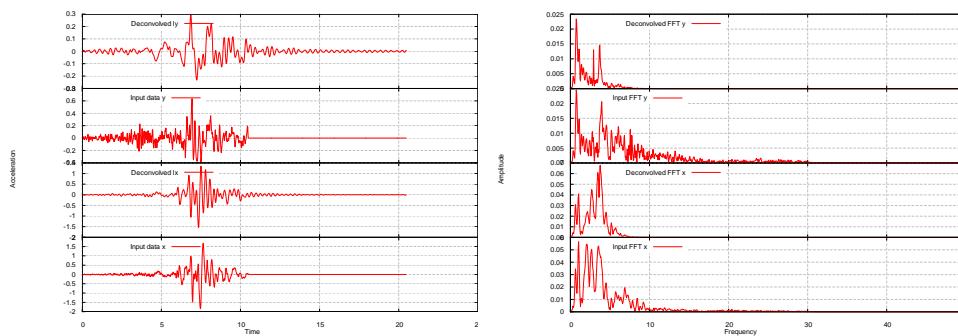


Figure 6.9: Deconvoluted Signals

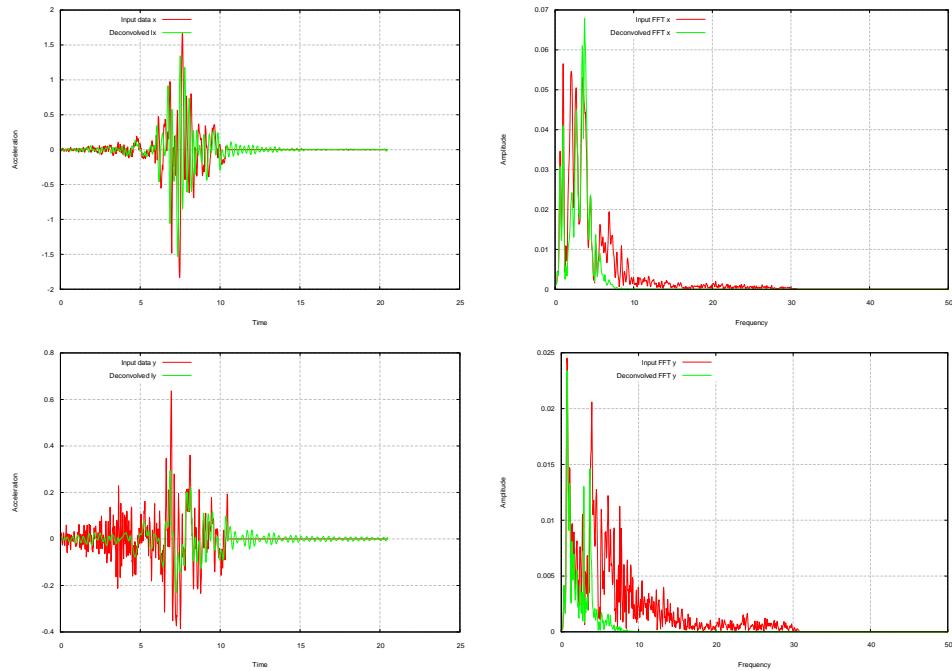


Figure 6.10: Comparison between Original and Deconvoluted Signals

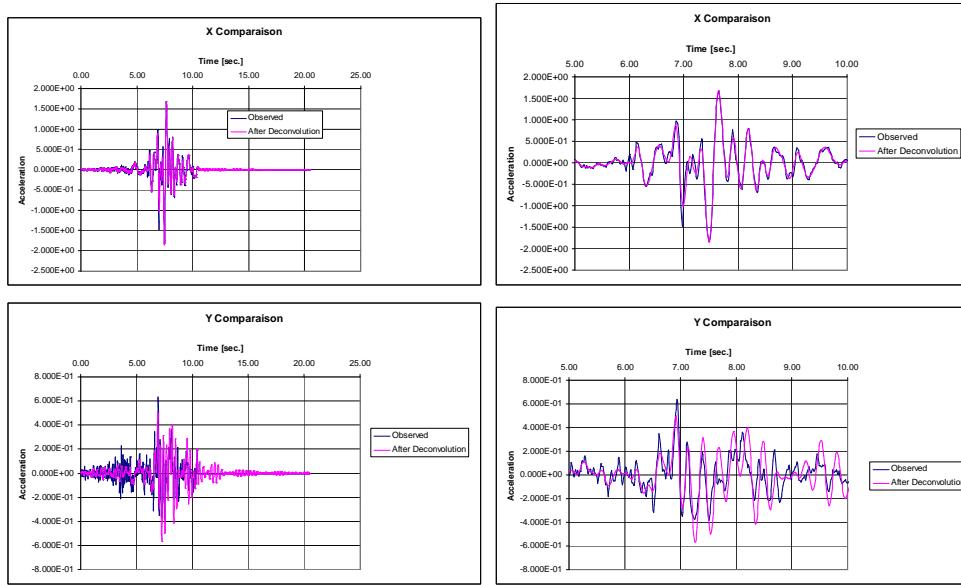


Figure 6.11: Results of Deconvolution Analysis

Chapter 7

HU-WASHIZU; MIXED ITERATIVE METHODS

7.1 Multifield Variational Principles

A **Multifield** variational principle is one that has more than one master field (or state variable), that is more than one unknown field is subject to independent variations. In linear elastostatics, we can have displacement, \mathbf{u} , strains $\boldsymbol{\epsilon}$, or stress $\boldsymbol{\sigma}$ as potential candidates for master fields. Hence seven combinations are possible, (Felippa 2000), Table 7.1.

7.2 General Hu-Washizu Variational Principle

Adapted from (Reich 1993)

The Hu-Washizu (HW) variational principle is a three-field variational principle in which the **displacements, strains, and stresses** are treated as **independent fields** (as opposed to only the displacement in the total potential energy principle). Naturally, the two additional field variables, with respect to the TPE variational principle, appear not only in the functional, but also in the discretized system of equations. Consequently, for a domain with a given discretization the discrete system of equations derived from the HW variational principle will be much larger than the discrete system of equations derived from the TPE variational principle. With the increased number of equations, significant improvements in accuracy can be observed for the solution obtained from the discrete form of the HW variational principle compared to the solution obtained from the discrete form of the TPE variational principle for the same discretization. This means that coarse discretizations can be used with the discrete form of the HW variational principle to obtain the same degree of accuracy that would be observed with much finer discretizations using the TPE variational principle. The functional for the HW variational principle is derived from the functional for the TPE variational principle by imposing the strain-displacement equation as a finite subsidiary condition using the method of Lagrange multipliers. The finite subsidiary condition or constraint is written in residual form as

$$\mathbf{L} \mathbf{u} - \boldsymbol{\epsilon} = \mathbf{0} \quad (7.1)$$

and enforced in an average sense over the entire body Ω . By imposing the strain-displacement equation as a constraint C^0 continuous strains and stresses are obtained in the discrete form of the variational statements, as opposed to the discontinuous strains and stresses obtained in the discrete form of the variational statement for the TPE variational principle. The constrained functional is written as

$$\Pi_{HW} = \Pi_{TPE} + \int_{\Omega} \boldsymbol{\lambda}^T (\mathbf{L} \mathbf{u} - \boldsymbol{\epsilon}) d\Omega \quad (7.2)$$

Where $\boldsymbol{\lambda}$ is the Lagrange multiplier and to be consistent with the integrals in the TPE functional (i.e. Equation ??) the Lagrange multiplier must have the units of stress. Since this is the case, $\boldsymbol{\sigma}$ will be used for the Lagrange

\mathbf{u}	$\boldsymbol{\epsilon}$	$\boldsymbol{\sigma}$	Name
Single Field			
Y		Y	Total Potential Energy
	Y		Total Complementary Potential Energy
Two Fields			
Y	Y	Y	Hellinger-Reissner
Y		Y	de Veubeke
	Y		No name
Three Fields			
Y	Y	Y	Hu-Washizu

Table 7.1: Functionals in Linear Elasticity

multiplier instead of the more typical λ such that the physical meaning of the Lagrange multiplier is more apparent. The functional for the HW variational principle thus becomes

$$\begin{aligned} \Pi_{HW} = & \underbrace{\frac{1}{2} \int_{\Omega} \boldsymbol{\epsilon}^T \mathbf{D} \boldsymbol{\epsilon} d\Omega - \int_{\Omega} \boldsymbol{\epsilon}^T \mathbf{D} \boldsymbol{\epsilon}_0 d\Omega + \int_{\Omega} \boldsymbol{\epsilon}^T \boldsymbol{\sigma}_0 d\Omega}_{-W_e} \\ & - \underbrace{\int_{\Omega} \mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \mathbf{u}^T \hat{\mathbf{t}} d\Gamma}_{U} + \underbrace{\int_{\Omega} \boldsymbol{\sigma}^T (\mathbf{L} \mathbf{u} - \boldsymbol{\epsilon}) d\Omega}_{\text{Constraint}} \end{aligned} \quad (7.3)$$

A variational statement is obtained by taking the first variation of the functional and setting this scalar quantity equal to zero. The first variation of the HW functional, with terms arranged according to which field variable is varied, is

$$\begin{aligned} \delta \Pi_{HW} = & \int_{\Omega} \delta(\mathbf{L} \mathbf{u})^T \boldsymbol{\sigma} d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \hat{\mathbf{t}} d\Gamma \\ & + \int_{\Omega} \delta \boldsymbol{\epsilon}^T \mathbf{D} \boldsymbol{\epsilon} d\Omega - \int_{\Omega} \delta \boldsymbol{\epsilon}^T \mathbf{D} \boldsymbol{\epsilon}_0 d\Omega + \int_{\Omega} \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma}_0 d\Omega - \int_{\Omega} \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} d\Omega \\ & + \int_{\Omega} \delta \boldsymbol{\sigma}^T (\mathbf{L} \mathbf{u} - \boldsymbol{\epsilon}) d\Omega = 0 \end{aligned} \quad (7.4)$$

Note that the 4th and 7th term were added and cancell each others, and that we are not using Eq. ?? in this formulation. Since \mathbf{u} , $\boldsymbol{\epsilon}$, and $\boldsymbol{\sigma}$ are independent field variables, terms involving $\delta \mathbf{u}$, $\delta \boldsymbol{\epsilon}$, and $\delta \boldsymbol{\sigma}$ must add up to zero individually and are, therefore grouped together to form three separate variational statements (analogous to the method of separation of variables in the solution of partial differential equations)

$$\int_{\Omega} \delta(\mathbf{L} \mathbf{u})^T \boldsymbol{\sigma} d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \hat{\mathbf{t}} d\Gamma = 0 \quad (7.5-a)$$

$$\int_{\Omega} \delta \boldsymbol{\epsilon}^T [\mathbf{D}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0) + \boldsymbol{\sigma}_0 - \boldsymbol{\sigma}] d\Omega = 0 \quad (7.5-b)$$

$$\int_{\Omega} \delta \boldsymbol{\sigma}^T (\mathbf{L} \mathbf{u} - \boldsymbol{\epsilon}) d\Omega = 0 \quad (7.5-c)$$

To obtain the corresponding Euler equations for the general form of the HW variational principle the volume integral in Equation 7.5-a containing the variation of the strains $\delta(\mathbf{L} \mathbf{u})$ defined in terms of the displacements \mathbf{u} must be integrated by parts using Green's theorem in order to obtain a form of the variational statement in terms of the variation of the displacements $\delta \mathbf{u}$. Integration by parts (Eq. ??) of this integral yields

$$\int_{\Omega} \delta(\mathbf{L} \mathbf{u})^T \boldsymbol{\sigma} d\Omega = \oint_{\Gamma} \delta \mathbf{u}^T \mathbf{G} \boldsymbol{\sigma} d\Gamma - \int_{\Omega} \delta \mathbf{u}^T \mathbf{L}^T \boldsymbol{\sigma} d\Omega \quad (7.6)$$

where \mathbf{G} is a transformation matrix containing the direction cosines for a unit normal vector such that the surface tractions \mathbf{t} are defined as $\mathbf{t} = \mathbf{G} \boldsymbol{\sigma}$ and the surface integral is over the entire surface of the body Γ . Substituting Equation 7.6 into Equation 7.5-a, the first variational statement becomes

$$-\int_{\Omega} \delta \mathbf{u}^T (\mathbf{L}^T \boldsymbol{\sigma} + \mathbf{b}) d\Omega + \int_{\Gamma_t} \delta \mathbf{u}^T (\mathbf{G} \boldsymbol{\sigma} - \hat{\mathbf{t}}) d\Gamma = 0 \quad (7.7)$$

Since $\delta \mathbf{u}$ is arbitrary the expressions in the integrands within the parentheses must both be equal to zero for the sum of the integrals to be equal to zero. Likewise, $\delta \boldsymbol{\epsilon}$ and $\delta \boldsymbol{\sigma}$ are also arbitrary and the expressions within the braces in the second variational statement (i.e. Equation 7.5-b) and within the parentheses in the third variational statement (i.e. Equation 7.5-c) must both be equal to zero for the integral to be equal to zero. The Euler equations for the HW functional are

(BE): Equilibrium	$\mathbf{L}^T \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0}$	on Ω
(CE): Stress-Strain	$\mathbf{D}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0) + \boldsymbol{\sigma}_0 - \boldsymbol{\sigma} = \mathbf{0}$	on Ω
(KE): Strain-Displacement	$\mathbf{L} \mathbf{u} - \boldsymbol{\epsilon} = \mathbf{0}$	on Ω
(NBC): Natural B.C.	$\mathbf{G} \boldsymbol{\sigma} - \hat{\mathbf{t}} = \mathbf{0}$	on Γ_t

(7.8)

where the first Euler equation is the equilibrium equation; the second Euler equation is the stress-strain relationship; the third Euler equation is the strain-displacement equation; and the fourth Euler equation defines the natural boundary conditions. The natural boundary conditions are defined on Γ_t rather than Γ because both the applied surface tractions $\hat{\mathbf{t}}$ and the matrix-vector product $\mathbf{G} \boldsymbol{\sigma}$ are identically zero outside Γ_t . Starting from the Euler equations, it is possible to derive the HW functional by performing the operations just presented in reverse order.

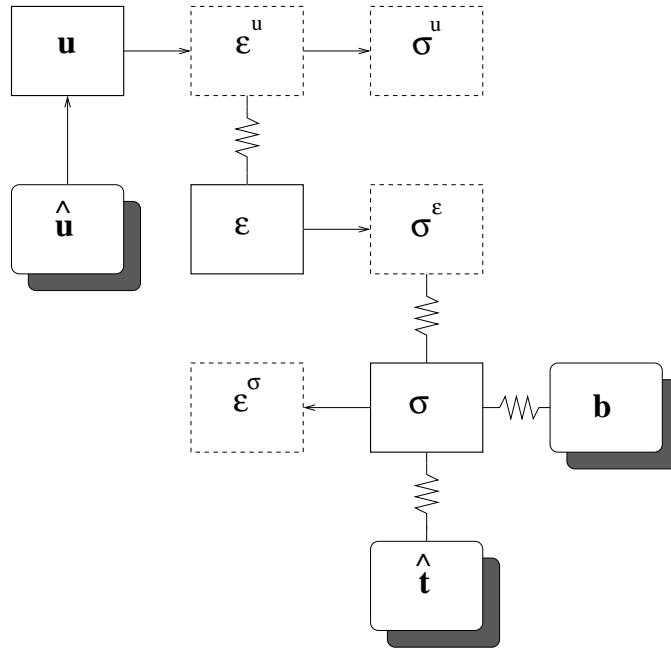


Figure 7.1: Tonti Diagram for Hu-Washizu, (Červenka, J. 1994)

This last set of four Euler equations, should be compared with the two (Eq. ?? and ??) obtained from the original TPE. The additional two equations bring into play stress-strain and strain displacement. Also, whereas the original formulation (Eq. ?? and ??) was in terms of the displacement only (\mathbf{u}), the Hu-Washizu formulation is in terms of three independent variables (\mathbf{u} , $\boldsymbol{\sigma}$ and $\boldsymbol{\epsilon}$), Table 7.2.

		TPE	HW	
Euler Equations				
Equilibrium	$\mathbf{L}^T \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0}$	Ω	Y	Y
Stress-Strain	$\mathbf{D}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0) + \boldsymbol{\sigma}_0 - \boldsymbol{\sigma} = \mathbf{0}$	Ω	N	Y
Strain-Displacement	$\mathbf{L}\mathbf{u} - \boldsymbol{\epsilon} = \mathbf{0}$	Ω	N	Y
Natural B.C.	$\mathbf{G}\boldsymbol{\sigma} - \hat{\mathbf{t}} = \mathbf{0}$	Γ_t	Y	Y
Variables				
Displacement	\mathbf{u}		Y	Y
Strain	$\boldsymbol{\epsilon}$		N	Y
Stress	$\boldsymbol{\sigma}$		N	Y

Table 7.2: Comparison Between Total Potential Energy and Hu-Washizu Formulations

The Tonti diagram for the HW is shown in Fig. ??.

7.3 Discretization of the Variational Statement for the HW Variational Principle

Adapted from (Reich 1993)

The discretization of the three variational statements defined in Equation ?? will be performed on an element domain Ω_e using the procedures described in Chapter 2 of (?) assembly of the discrete element equations into a discrete global system of equations is straightforward and will be omitted from this discussion.

The surface of the element subjected to surface tractions Γ_t comprises one or more surfaces of the element boundary Γ_e . For the present time this discussion will be kept on a very general level with no mention of the dimensionality of the elements; the number of nodes defining the elements; or the nature of constitutive law.

The first step in the discretization process is to define the displacements \mathbf{u} , strains $\boldsymbol{\epsilon}$, and stresses $\boldsymbol{\sigma}$ at a point inside the element in terms of the shape functions \mathbf{N}_u , \mathbf{N}_ϵ , and \mathbf{N}_σ , respectively, and the element nodal displacements

$\bar{\mathbf{u}}_e$, strains $\bar{\boldsymbol{\epsilon}}_e$, and stresses $\bar{\boldsymbol{\sigma}}_e$

$$\begin{aligned} \mathbf{u} &= \mathbf{N}_u \bar{\mathbf{u}}_e \\ \boldsymbol{\epsilon} &= \mathbf{N}_\epsilon \bar{\boldsymbol{\epsilon}}_e \\ \boldsymbol{\sigma} &= \mathbf{N}_\sigma \bar{\boldsymbol{\sigma}}_e \end{aligned} \quad (7.9)$$

We note that contrarily to the previous case (Eq. ??) we now have three discretizations (instead of just one). The virtual displacements $\delta\mathbf{u}$, virtual strains $\delta\boldsymbol{\epsilon}$, and virtual stresses $\delta\boldsymbol{\sigma}$ at a point inside the element can also be defined in terms of the shape functions \mathbf{N}_u , \mathbf{N}_ϵ , and \mathbf{N}_σ , respectively, and the nodal virtual displacements $\delta\bar{\mathbf{u}}_e$, virtual strains $\delta\bar{\boldsymbol{\epsilon}}_e$, and virtual stresses $\delta\bar{\boldsymbol{\sigma}}_e$ for the element

$$\delta\mathbf{u} = \mathbf{N}_u \delta\bar{\mathbf{u}}_e \quad (7.10-a)$$

$$\delta\boldsymbol{\epsilon} = \mathbf{N}_\epsilon \delta\bar{\boldsymbol{\epsilon}}_e \quad (7.10-b)$$

$$\delta\boldsymbol{\sigma} = \mathbf{N}_\sigma \delta\bar{\boldsymbol{\sigma}}_e \quad (7.10-c)$$

We now need to discretize each one of the corresponding Euler equations:

In order to discretize the volume integral in the first variational statement (i.e. Equ. 7.5-a) defining the virtual strain energy for the element, Equation 7.10-a is substituted into the virtual strain-displacement relationship (i.e. Equation ??) to define the virtual strains $\delta\boldsymbol{\epsilon}$ at a point inside the element in terms of the nodal virtual displacements $\delta\bar{\mathbf{u}}_e$

$$\delta(\mathbf{L}\mathbf{u}) = \mathbf{L}\delta\mathbf{u} = \mathbf{L}\mathbf{N}_u \delta\bar{\mathbf{u}}_e \quad (7.11)$$

Defining the discrete strain-displacement operator \mathbf{B}_u as

$$\mathbf{B}_u = \mathbf{L}\mathbf{N}_u \quad (7.12)$$

and substituting Equation 7.9 into the integrand, the virtual strain energy for an element is written as

$$\int_{\Omega_e} \delta(\mathbf{L}\mathbf{u})^T \boldsymbol{\sigma} d\Omega = \delta\bar{\mathbf{u}}_e^T \int_{\Omega_e} \mathbf{B}_u^T \mathbf{N}_\sigma d\Omega \bar{\boldsymbol{\sigma}}_e \quad (7.13)$$

Defining an element operator matrix \mathbf{F}_e as

$$\mathbf{F}_e^T = \int_{\Omega_e} \mathbf{B}_u^T \mathbf{N}_\sigma d\Omega \quad (7.14)$$

Equation 7.13 can be rewritten as

$$\int_{\Omega_e} \delta(\mathbf{L}\mathbf{u})^T \boldsymbol{\sigma} d\Omega = \delta\bar{\mathbf{u}}_e^T \mathbf{F}_e^T \bar{\boldsymbol{\sigma}}_e \quad (7.15)$$

In order to discretize the volume integral defining the work done by the body forces and the surface integral defining the work done by the surface tractions in the first variational statement (i.e. the first equation in Equation ??), Equation 7.10-a is substituted into the integrands

$$\int_{\Omega_e} \delta\mathbf{u}^T \mathbf{b} d\Omega = \delta\bar{\mathbf{u}}_e^T \int_{\Omega_e} \mathbf{N}_u^T \mathbf{b} d\Omega \quad (7.16)$$

$$\int_{\Gamma_t} \delta\mathbf{u}^T \hat{\mathbf{t}} d\Gamma = \delta\bar{\mathbf{u}}_e^T \int_{\Gamma_t} \mathbf{N}_u^T \hat{\mathbf{t}} d\Gamma \quad (7.17)$$

Defining the applied force vector \mathbf{f}_e as

$$\mathbf{f}_e = \int_{\Omega_e} \mathbf{N}_u^T \mathbf{b} d\Omega + \int_{\Gamma_t} \mathbf{N}_u^T \hat{\mathbf{t}} d\Gamma \quad (7.18)$$

the sum of the internal and external virtual work is

$$\int_{\Omega_e} \delta\mathbf{u}^T \mathbf{b} d\Omega + \int_{\Gamma_t} \delta\mathbf{u}^T \hat{\mathbf{t}} d\Gamma = \delta\bar{\mathbf{u}}_e^T \mathbf{f}_e \quad (7.19)$$

Having defined the discretization of the various integrals in the first variational statement for the HW variational principle (i.e. Equ. 7.5-a), it is now possible to define the discrete system of equations. Substituting Equations 7.15 and 7.19 into the variational statement and rearranging terms, the discretized Principle of Virtual Work is

$$\delta\bar{\mathbf{u}}_e^T \mathbf{F}_e^T \bar{\boldsymbol{\sigma}}_e = \delta\bar{\mathbf{u}}_e^T \mathbf{f}_e \quad (7.20)$$

where the left-hand side is the virtual strain energy and the right-hand side is the internal and external virtual work. Since $\delta\bar{\mathbf{u}}_e$ is an arbitrary (i.e. non-zero) vector appearing on both sides of Equation 7.20, the discrete system of equations can be simplified into

$$\boxed{\mathbf{F}_e^T \bar{\boldsymbol{\sigma}}_e = \mathbf{f}_e} \quad (7.21)$$

as the discrete system of equations for an element.

In order to discretize the second variational statement (i.e. Equ. 7.5-b), Equations 7.9, 7.9, and 7.10-b are substituted into the integrand

$$\begin{aligned} \int_{\Omega} \delta\boldsymbol{\epsilon}^T [\mathbf{D}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0) + \boldsymbol{\sigma}_0 - \boldsymbol{\sigma}] d\Omega &= \delta\bar{\boldsymbol{\epsilon}}_e^T \int_{\Omega_e} \mathbf{N}_e^T \mathbf{D} \mathbf{N}_e d\Omega \bar{\boldsymbol{\epsilon}}_e - \delta\bar{\boldsymbol{\epsilon}}_e^T \int_{\Omega_e} \mathbf{N}_e^T \mathbf{D} \boldsymbol{\epsilon}_0 d\Omega \\ &+ \delta\bar{\boldsymbol{\epsilon}}_e^T \int_{\Omega_e} \mathbf{N}_e^T \boldsymbol{\sigma}_0 d\Omega - \delta\bar{\boldsymbol{\epsilon}}_e^T \int_{\Omega_e} \mathbf{N}_e^T \mathbf{N}_{\sigma} d\Omega \bar{\boldsymbol{\sigma}}_e = 0 \end{aligned} \quad (7.22)$$

Defining a pair of element operator matrices \mathbf{A}_e and \mathbf{C}_e as

$$\mathbf{A}_e = \int_{\Omega_e} \mathbf{N}_e^T \mathbf{D} \mathbf{N}_e d\Omega \quad (7.23)$$

$$\mathbf{C}_e = \int_{\Omega_e} \mathbf{N}_e^T \mathbf{N}_{\sigma} d\Omega \quad (7.24)$$

and the initial strain/stress vector \mathbf{g}_e as

$$\mathbf{g}_e = \int_{\Omega_e} \mathbf{N}_e^T \mathbf{D} \boldsymbol{\epsilon}_0 d\Omega - \int_{\Omega_e} \mathbf{N}_e^T \boldsymbol{\sigma}_0 d\Omega \quad (7.25)$$

Equation 7.22 can be rewritten as

$$\int_{\Omega} \delta\boldsymbol{\epsilon}^T [\mathbf{D}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0) + \boldsymbol{\sigma}_0 - \boldsymbol{\sigma}] d\Omega = \delta\bar{\boldsymbol{\epsilon}}_e^T \mathbf{A}_e \bar{\boldsymbol{\epsilon}}_e - \delta\bar{\boldsymbol{\epsilon}}_e^T \mathbf{g}_e - \delta\bar{\boldsymbol{\epsilon}}_e^T \mathbf{C}_e \bar{\boldsymbol{\sigma}}_e = 0 \quad (7.26)$$

Since the nodal virtual strains $\delta\bar{\boldsymbol{\epsilon}}$ are arbitrary they can be eliminated from Equation 7.26 yielding

$$\boxed{\mathbf{A}_e \bar{\boldsymbol{\epsilon}}_e - \mathbf{C}_e \bar{\boldsymbol{\sigma}}_e = \mathbf{g}_e} \quad (7.27)$$

as the discretized form of the second variational statement.

In order to discretize the third variational statement (i.e. Eq. 7.5-c), Equations 7.9, 7.9, and 7.10-c are substituted into the integrand

$$\int_{\Omega_e} \delta\boldsymbol{\sigma}^T (\mathbf{L}\mathbf{u} - \boldsymbol{\epsilon}) d\Omega = \delta\bar{\boldsymbol{\sigma}}_e^T \int_{\Omega_e} \mathbf{N}_{\sigma}^T \mathbf{B}_u d\Omega \bar{\mathbf{u}}_e - \delta\bar{\boldsymbol{\sigma}}_e^T \int_{\Omega_e} \mathbf{N}_{\sigma}^T \mathbf{N}_e d\Omega \bar{\boldsymbol{\epsilon}}_e = 0 \quad (7.28)$$

Recognizing that

$$\int_{\Omega_e} \mathbf{N}_{\sigma}^T \mathbf{B}_u d\Omega = \mathbf{F}_e \quad (7.29)$$

$$\int_{\Omega_e} \mathbf{N}_{\sigma}^T \mathbf{N}_e d\Omega = \mathbf{C}_e^T \quad (7.30)$$

Equation 7.28 can be rewritten as

$$\int_{\Omega_e} \delta\boldsymbol{\sigma}^T (\mathbf{L}\mathbf{u} - \boldsymbol{\epsilon}) d\Omega = \delta\bar{\boldsymbol{\sigma}}_e^T \mathbf{F}_e \bar{\mathbf{u}}_e - \delta\bar{\boldsymbol{\sigma}}_e^T \mathbf{C}_e^T \bar{\boldsymbol{\epsilon}}_e = 0 \quad (7.31)$$

Since the nodal virtual stresses $\delta\bar{\boldsymbol{\sigma}}_e$ are arbitrary they can be eliminated from Equation 7.31 yielding

$$\boxed{\mathbf{F}_e \bar{\mathbf{u}}_e - \mathbf{C}_e^T \bar{\boldsymbol{\epsilon}}_e = 0} \quad (7.32)$$

as the discretized form of the third variational statement.

Having defined the discretized form of all three variational statements, it is now possible to define the discrete mixed system of equations for an element. Assembling Equations 7.21, 7.27, and 7.32 in matrix form adopting the classic arrangement for a constrained system of equations

$$\left[\begin{array}{ccc} \mathbf{A}_e & -\mathbf{C}_e & \mathbf{0} \\ -\mathbf{C}_e^T & \mathbf{0} & \mathbf{F}_e \\ \mathbf{0} & \mathbf{F}_e^T & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \bar{\boldsymbol{\epsilon}}_e \\ \bar{\boldsymbol{\sigma}}_e \\ \bar{\mathbf{u}}_e \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{g}_e \\ \mathbf{0} \\ \mathbf{f}_e \end{array} \right\} \quad (7.33)$$

yields a symmetric system of equations. Although $\boldsymbol{\epsilon}$ is technically an intermediate variable in the field equations indirectly relating $\boldsymbol{\sigma}$ to \mathbf{u} , $\bar{\boldsymbol{\epsilon}}_e$ is the primary variable and $\bar{\boldsymbol{\sigma}}_e$ to $\bar{\mathbf{u}}_e$ are constraint variables in Equation 7.33.

Since it would be computationally expensive to solve the system of equations in Eq. 7.33 using direct method, an indirect or iterative procedure (i.e. Gauss-Seidel instead of Gauss-Jordan) is often selected, (?).

$$\begin{aligned} \text{Step 1: } \bar{\mathbf{u}}_n^{k+1} &= \bar{\mathbf{u}}_n^k + \mathbf{K}^{-1} \bar{\mathbf{r}}_n^k \\ \text{Step 2: } \bar{\boldsymbol{\epsilon}}_n^{k+1} &= \mathbf{C}^{-T} \mathbf{F} \bar{\mathbf{u}}_n^{k+1} \\ \text{Step 3: } \bar{\boldsymbol{\sigma}}_n^{k+1} &= \mathbf{C}^{-1} \mathbf{A} \bar{\boldsymbol{\epsilon}}_n^{k+1} \\ \text{Step 4: } \bar{\mathbf{r}}_n^{k+1} &= \mathbf{f} - \mathbf{F}^T \bar{\boldsymbol{\sigma}}_n^{k+1} \end{aligned} \quad (7.34)$$

for $k = 0, 1, 2, \dots$, where k is an iteration index and \mathbf{r}_n^{k+1} is the residual force vector. It should be noted that this procedure is solved on the structural level, meaning that steps 1 to 3 require a solution of a system of linear equations.

Step 1, \mathbf{K} corresponds to the classical standard displacement stiffness matrix, and this step is used as a preconditioner. This implies that at the beginning of the first iteration, when $\mathbf{u}_n^0 = \mathbf{0}$ and $\mathbf{r}_n^0 = \mathbf{0}$, step corresponds to the standard displacement-based formulation of the finite element method. Steps 1, 2, and 3 above require the solution of simultaneous linear equations. Step 3, however, may be reduced by nodal quadrature and assuming same interpolation functions for strains and stresses to

$$\boldsymbol{\sigma}_i = \mathbf{D} \boldsymbol{\epsilon}_i \quad (7.35)$$

In this equation, $\boldsymbol{\sigma}_i$ and $\boldsymbol{\epsilon}_i$ are the stresses at node i , respectively, and \mathbf{D} is the stress-strain constitutive matrix. Then, Step 3 is nothing else but direct computation of nodal stresses from nodal strains using the constitutive matrix \mathbf{D} . Finally, the uniqueness and the existence of a solution has been addressed by the so-called Babuška-Brezzi (BB) condition (Babuška 1973, Brezzi 1974). Details of the algorithmic implementation will be covered in a later chapter.

7.4 Element Formulation

Taken from (Červenka, J. 1994)

It is necessary to select appropriate interpolation functions for all three elastic fields (i.e. \mathbf{u} , $\boldsymbol{\epsilon}$ and $\boldsymbol{\sigma}$). The choice of these shape functions must be such that the BB condition is satisfied (Appendix ??). In this work, the same interpolation functions are used for all three fields (i.e. displacements, strains and stresses), which implies that there is a full number of unknowns in each node.

$$\dim(\mathbf{u}_n) = N \times \dim - R, \quad \dim(\boldsymbol{\epsilon}_n) = \dim(\boldsymbol{\sigma}_n) = N \times \dim(\boldsymbol{\sigma}) \quad (7.36)$$

where N denotes the number of nodes, \dim is the problem dimension and R is the number of rigid body modes. In Section 7.6, it will be shown that this formulation guarantees the satisfaction of the BB condition.

The polynomial orders of the field approximations are given in Table 7.3.

Table 7.3: Polynomial orders of the shape functions.

field	2D		3D	
	T3	T6	T4	T10
displacement \mathbf{u}	linear	quadratic	linear	quadratic
strain $\boldsymbol{\epsilon}$	linear	quadratic	linear	quadratic
stress $\boldsymbol{\sigma}$	linear	quadratic	linear	quadratic

In general case a variable x is interpolated over a finite element using the expression:

$$x = \sum_i^{N_{en}} \Phi_i x_i \quad (7.37)$$

where N_{en} is the number of element nodes, Φ_i is an interpolation function associated with node i , and x_i is the value of variable x at element node i . For the linear triangular element (T3) the interpolation functions are:

$$\Phi_i = l_i, \quad i = 1, 2, 3 \quad (7.38)$$

and for the six noded triangular element (T6) with three corner nodes and three mid-side nodes the interpolation functions are:

$$\Phi_i = l_i(2l_i - 1), \quad i = 1, 2, 3, \quad \Phi_4 = 4l_1l_2, \quad \Phi_5 = 4l_2l_3, \quad \Phi_6 = 4l_3l_1, \quad (7.39)$$

where element nodes 1 to 3 indicate the element corner nodes and 4 to 6 are the mid-side nodes. Symbols l_i denote the natural area coordinates of the element, which are related to the element natural coordinates ξ and η by relations:

$$l_1 = \xi, \quad l_2 = \eta, \quad l_3 = 1 - \xi - \eta \quad (7.40)$$

For three-dimensional finite elements, the interpolation functions are similar. For the linear tetrahedron T4 element they are:

$$\Phi_i = l_i, \quad i = 1, 2, 3, 4 \quad (7.41)$$

and for the T10 element with four corner nodes and six mid-edge nodes they are defined analogically to the six noded triangular T6 element as:

$$\begin{aligned} \Phi_i &= l_i(2l_i - 1), & i &= 1, 2, 3, 4, & \Phi_5 &= 4l_1l_2, & \Phi_6 &= 4l_2l_3, \\ \Phi_7 &= 4l_3l_1, & \Phi_8 &= 4l_1l_4, & \Phi_9 &= 4l_2l_4, & \Phi_{10} &= 4l_3l_4 \end{aligned} \quad (7.42)$$

Similarly to the two-dimensional elements, symbols l_i denotes the volumetric natural coordinates, which are again related to the element natural coordinates by relations:

$$l_1 = \xi, \quad l_2 = \eta, \quad l_3 = \zeta, \quad l_4 = 1 - \xi - \eta - \zeta \quad (7.43)$$

7.5 Strain Recovery

Taken from (Červenka, J. 1994)

In Equation ??, Step 2 is essentially an expression for the computation of nodal strains from nodal displacements. It involves the inversion of a symmetric matrix \mathbf{C}^T , or in other words the solution of a system of linear simultaneous equations. Three strain recovery techniques, described below, represent an attempt to avoid the direct solution of a large linear system of equations in this step, as its assembly and factorization is computationally expensive.

Three algorithms are discussed and compared: (1) C-lumping (CL), (2) Strain smoothing (SS) and (3) C-splitting (CS). The CL as the simplest algorithm, in which a lumped form of matrix \mathbf{C} is constructed, and the inversion of the resulting diagonal matrix is trivial. The other two algorithms are different iterative techniques to solve the system of equations. Between them, the CS method is tailored for the fastest convergence for linear elements.

In the sequel, the MIM iteration index k is omitted as it focuses on faster solution techniques for the Step 2 only.

7.5.1 C-lumping.

The inversion of \mathbf{C} for the C-lumping (CL) technique is simplified by forming a diagonalized \mathbf{C} matrix. This lumped \mathbf{C} matrix ¹ is evaluated by the following expression:

$$\mathbf{C}_L = \int_V (\mathbf{I} \Phi) dV \quad (7.44)$$

where \mathbf{I} is the identity matrix and Φ is the shape function matrix. Since identical shape functions are used for all three primary fields, the subscript at Φ is no longer necessary. Replacing \mathbf{C} by \mathbf{C}_L Step 2 of Equation ?? reduces to:

$$\boldsymbol{\epsilon}_n = \mathbf{C}_L^{-1} \mathbf{E} \mathbf{u}_n \quad (7.45)$$

At the element level, the lumped \mathbf{C} matrix for the four node linear tetrahedron (T4) is:

$$\mathbf{C}_L^e = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} \psi \quad (7.46)$$

where ψ is a constant based on the element volume.

CL is by far the simplest strain recovery method since no iterations are required to compute the nodal strains. However, the numerical experiments reported in Section ?? indicate that the displacement solution converges to an erroneous value. Hence, the C-lumping technique is kinematically inconsistent.

¹Note: The diagonalized matrix \mathbf{C}_L and the consistent matrix \mathbf{C} described in Section 7.5.2 and 7.5.3 are determined in an analogous way as the standard lumped and consistent mass matrices in dynamics. The only difference is the exclusion of the weight density of the material which is replaced by unity.

7.5.2 Strain smoothing.

Strain Smoothing (Zienkiewicz, Vilotte, Toyoshima and Nakazawa 1985) (SS) is an indirect procedure within *Step 2* that avoids the direct decomposition of the \mathbf{C} matrix. Nodal strains are iteratively evaluated until the ratio of the Euclidean norms of strain correction to total strains satisfies a prescribed limit.

This technique exploits the diagonal matrix \mathbf{C}_L previously described and the consistent matrix \mathbf{C} defined below. Iteratively the nodal strains are evaluated by:

$$\boldsymbol{\epsilon}_n^{j+1} = \boldsymbol{\epsilon}_n^j + \mathbf{C}_L^{-1}(\mathbf{E}\mathbf{u}_n - \mathbf{C}\boldsymbol{\epsilon}_n^j). \quad (7.47)$$

where $j = 0, 1, 2, \dots$ is the strain-iteration count. Note that this represents an internal iteration, not to be confused with the MIM iteration of (??). The iteration process involves the nodal strains in the whole mesh since *Step 2* is equivalent to the least square fit of the nodal based strain field to the strain field derived from the displacement field (?).

For a four noded linear tetrahedral element (T4), the consistent matrix \mathbf{C} is given at the element level by:

$$\mathbf{C}^e = \int_{V_e} \Phi \Phi^T dV = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} \psi \quad (7.48)$$

where ψ is again a constant based on the element volume.

The correction of nodal strain $\Delta\boldsymbol{\epsilon}_n^j$ during one iteration is:

$$\Delta\boldsymbol{\epsilon}_n^j = \boldsymbol{\epsilon}_n^{j+1} - \boldsymbol{\epsilon}_n^j = \mathbf{A}(\rho)\Delta\boldsymbol{\epsilon}_n^{j-1}, \quad (j \geq 1) \quad (7.49)$$

where:

$$\mathbf{A}(\rho) = \mathbf{I} - \mathbf{C}_L^{-1}\mathbf{C} \quad (7.50)$$

where $\mathbf{A}(\rho)$ is a fixed amplification matrix having a spectral radius $\rho = \frac{4}{5}$. The spectral radius ρ is defined as the largest eigenvalue of amplification matrix $\mathbf{A}(\rho)$. Since Equation 7.50 involves a product of \mathbf{C} and inverse of \mathbf{C}_L , the constants ψ are cancelled out. By Banach's fixed point theorem (Haser and Sullivan 1991) it is necessary for the spectral radius ρ to be less than 1 to ensure convergence of the iterative process given by Equation 7.47. Thus, this value of the spectral radius indicates an error decay of $\frac{1}{5}$.

7.5.3 C-splitting.

A new iterative process was recently developed by (Červenka, Keating and Felippa 1993) to solve *Step 2*. This new technique guarantees faster convergence for linear triangular and tetrahedral elements (T3 and T4). This technique is referred to as C-splitting (CS). This method “splits” the consistent matrix \mathbf{C} of Equation 7.48 into two matrices. One matrix is diagonalized and the second is formed such that their algebraic sum is equivalent to the original \mathbf{C} matrix:

$$\mathbf{C} = \mathbf{C}_D + \mathbf{C}_R \quad (7.51)$$

where:

$$\mathbf{C}_D = \alpha \text{diag}(\mathbf{C}) \quad \mathbf{C}_R = \mathbf{C} - \mathbf{C}_D \quad (7.52)$$

α is a “splitting” coefficient controlling the splitting of the matrix \mathbf{C} . Using this method *Step 2* in Equation ?? is modified to:

$$\boldsymbol{\epsilon}_n^{j+1} = \mathbf{C}_D^{-1}(\mathbf{E}\mathbf{u}_n - \mathbf{C}_R\boldsymbol{\epsilon}_n^j). \quad (7.53)$$

For the C-splitting method, the per-iteration strain correction is:

$$\Delta\boldsymbol{\epsilon}_n^j = \boldsymbol{\epsilon}_n^{j+1} - \boldsymbol{\epsilon}_n^j = \mathbf{A}(\rho)\Delta\boldsymbol{\epsilon}_n^{j-1}, \quad (j \geq 1) \quad (7.54)$$

where the amplification matrix $\mathbf{A}(\rho)$ is given by:

$$\mathbf{A}(\rho) = -\mathbf{C}_D^{-1}\mathbf{C}_R, \quad (7.55)$$

Table 7.4: Table of α coefficients and spectral radii for CS technique.

Element type	splitting coef.	spectral radius
	α	ρ
three node linear triangle T3	5/4	3/5
four node quadrilateral Q4	5/4	4/5
tetrahedral element T4	3/2	2/3
eight node brick element B8	7/4	0.929

It is possible to select the coefficient α such that the spectral radius of the amplification matrix is minimal. For four-node tetrahedron elements using $\alpha = \frac{3}{2}$, which as shown below minimizes the spectral radius of the amplification matrix, \mathbf{C} splits at the element level into the following matrices:

$$\mathbf{C}_D^e = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \psi \quad (7.56)$$

and

$$\mathbf{C}_R^e = \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \psi \quad (7.57)$$

Then the spectral radius ρ is equal to $\frac{2}{3}$. Thus, CS has an error decay rate of $\frac{1}{3}$ allowing for a faster convergence than the SS method. For example, 10 steps of CS can be expected to reduce the initial strain errors by $(\frac{2}{3})^{10} \approx 0.0173$ whereas 10 steps of SS would reduce those errors by only $(\frac{4}{5})^{10} \approx 0.1074$.

This technique was investigated also for other low order element types. The best splitting coefficients α and resulting spectral radii of the operator $A(\rho)$ (Eq. 7.55) for other element types are summarized in Table 7.4. The coefficients α for four node quadrilateral and eight node brick element are however valid only for elements with parallel or almost parallel sides. Therefore, the (SS) technique would be probably more reliable for these element types.

7.6 Uniqueness and Existence of a Solution

The BB condition for uniqueness and existence of a solution of the three-field variational principle is stated in Appendix ???. This condition was derived by (Babuška 1971), (Babuška 1973) and (Brezzi 1974). Xue and Atluri (1985) extended the condition to a general three-field problem, and derived its discrete form. The continuous and discrete forms of the BB conditions are again described in Appendix ???, and it is shown that they are equivalent to the following three conditions:

$$\begin{aligned} \text{rank}(\mathbf{E}) &= n_u \leq n_\sigma \\ \text{rank}(\mathbf{C}) &= n_\sigma \leq n_\epsilon + n_u \\ \mathbf{A} &\text{ is positive definite} \end{aligned} \quad (7.58)$$

where matrices \mathbf{E} , \mathbf{C} and \mathbf{A} are derived in Appendix ?? and are given by the following integrals:

$$\mathbf{E} = \int_V \Phi \mathbf{B} dV, \quad \mathbf{C} = \int_V \Phi \Phi^T dV, \quad \mathbf{A} = \int_V \Phi \mathbf{D} \Phi^T dV \quad (7.59)$$

The third condition is satisfied as long as the material does not exhibit softening. This is always guaranteed in the discrete crack approach, since softening is modeled only along the interface elements which are not included in the mixed iterative solution.

If identical shape functions are used for all three fields, then the number of unknowns for each field is given by Equation 7.36, and the inequalities in the first and second condition of Equation 7.58 are always satisfied.

$$\begin{aligned} n_u &= N \times \dim - R \leq n_\sigma &= N \times \dim(\boldsymbol{\sigma}) \\ n_\sigma &= N \times \dim(\boldsymbol{\sigma}) \leq n_\epsilon + n_u &= N \times \dim(\boldsymbol{\sigma}) + N \times \dim - R \end{aligned} \quad (7.60)$$

We note that the rank condition of matrix \mathbf{C} is also satisfied, as it is analogous to the consistent mass matrix of isoparametric elements, which has always full rank, as can be seen from Equation 7.48.

More complex is the verification of the rank condition of matrix \mathbf{E} . In Equation 7.59, \mathbf{E} is given by the integral:

$$\mathbf{E} = \int_V \Phi \mathbf{B} dV \quad (7.61)$$

where matrix \mathbf{B} is the matrix relating strains at a certain point to the nodal displacements and Φ is the matrix of shape functions relating strains or stresses at a certain point to their nodal counterparts. Matrix \mathbf{E} will have a rank equal to n_u if the following two conditions are satisfied.

$$\begin{array}{lll} \forall \mathbf{u}_n \neq \mathbf{0}, & \exists \mathbf{x} \in V & : \quad \mathbf{B}(\mathbf{x})\mathbf{u}_n \neq \mathbf{0} \\ \forall \boldsymbol{\epsilon}_n, & \exists \boldsymbol{\epsilon}(\mathbf{x}) = \Phi^T(\mathbf{x})\boldsymbol{\epsilon}_n & : \quad \boldsymbol{\epsilon}(\mathbf{x}) \text{ is unique} \end{array} \quad (7.62)$$

The first condition is equivalent to the requirement that a nonzero vector of nodal displacements must cause non-zero strain field. It should be noted that the rigid body modes are excluded from vector \mathbf{u}_n . They would be the only modes allowed to produce a zero strain field. In this case, matrix \mathbf{B} corresponds to that of a standard isoparametric triangular or tetrahedral element, and will therefore satisfy this condition.

The second condition is also satisfied, since the shape functions in matrix Φ are those of a standard isoparametric element, and the uniqueness of the interpolation is guaranteed.

For higher order triangular and tetrahedral elements (i.e. 6 noded triangle and 10 noded tetrahedron), it would seem preferable to select interpolation functions for strains and stresses, which are one order lower than those for the displacements. This would correspond to the mathematical relation between strains and displacements, since the strains are determined by differentiation of the displacement field. For this formulation, there would be unknown displacements, strains and stresses at each element corner node, but only displacement unknowns at the midside element nodes. The strains and stresses would be interpolated using linear shape functions and displacement using quadratic shape functions. Now, it is possible to show that this formulation would not guarantee the satisfaction of the inequality in the first condition of Equation 7.58.

We consider a patch of two six-noded triangular element as shown in Figure 7.2. From the previous assumptions

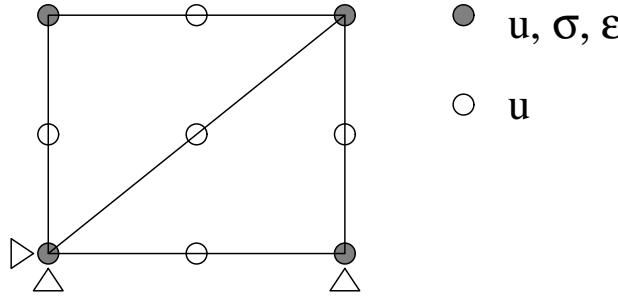


Figure 7.2: Patch test.

of linear stress and strain interpolation and quadratic displacement interpolation, the unknown stresses and strains are only at the element corner nodes, while all nodes have unknown displacements. The total number of stress and displacement unknowns is then given by:

$$n_u = 9 \times 2 - 3 = 15 > n_\sigma = 4 \times 3 = 12 \quad (7.63)$$

and clearly the important inequality $n_u \leq n_\sigma$ is not satisfied and the existence and uniqueness of a solution cannot be guaranteed. This should be contrasted by the previous formulation, in which the same interpolation functions are used for all three fields, and the satisfaction of the BB is guaranteed by Equation 7.60 and 7.62.

Chapter 8

MATERIAL NONLINEARITIES

8.1 Introduction

8.1.1 Linearization

We define a **constitutive operator** as

$$\sigma = \tilde{\sigma}(\varepsilon)$$

(8.1)

where $\tilde{\sigma}$ denotes the constitutive operator (analogous to the \mathbf{L}).

Given a strain state $\bar{\varepsilon}$, the corresponding stress will be $\bar{\sigma} = \tilde{\sigma}(\bar{\varepsilon})$. The constitutive operator $\tilde{\sigma}$ can be expanded into a Taylor series with respect to $\bar{\varepsilon}$

$$\tilde{\sigma}(\bar{\varepsilon} + \delta\varepsilon) = \tilde{\sigma}(\bar{\varepsilon}) + \left. \frac{\partial\tilde{\sigma}}{\partial\varepsilon} \right|_{\varepsilon=\bar{\varepsilon}} \delta\varepsilon + \dots \quad (8.2)$$

Neglecting quadratic and higher order terms leads to a **linearized** constitutive law

$$\sigma \approx \tilde{\sigma}(\bar{\varepsilon}) + \mathbf{D}(\bar{\varepsilon})\delta\varepsilon \quad (8.3)$$

which approximates Eq. 8.1 for strains in the neighborhood of $\bar{\varepsilon}$, and

$$\mathbf{D} \equiv \frac{\partial\tilde{\sigma}}{\partial\varepsilon}$$

(8.4)

is the **tangent stiffness matrix** which is a function of the *current* strain.

We rewrite Eq. ??, ?? and ?? in terms of the newly defined constitutive operator

$$\underbrace{\int_{\Omega_e} \mathbf{B}^T \tilde{\sigma}(\mathbf{B}\bar{\mathbf{u}}) d\Omega}_{\bar{\mathbf{f}}^{int}} = \underbrace{\int_{\Omega_e} \mathbf{B}^T \mathbf{D} \mathbf{e}_0 d\Omega - \int_{\Omega_t} \mathbf{B}^T \sigma_0 d\Omega}_{\underbrace{\bar{\mathbf{f}}_{0,e}}_{\bar{\mathbf{f}}^{ext}}} + \underbrace{\int_{\Omega_e} \mathbf{N}^T \mathbf{b} d\Omega + \int_{\Gamma_t} \mathbf{N}^T \hat{\mathbf{t}} d\Gamma}_{\bar{\mathbf{f}}_e} \quad (8.5)$$

or

$$\bar{\mathbf{f}}^{int}(\bar{\mathbf{u}}) = \bar{\mathbf{f}}^{ext}$$

(8.6)

We now develop a linearized expression for the internal forces. Given $\bar{\mathbf{u}}$ as nodal displacements yielding strain field $\bar{\varepsilon} = \mathbf{B}\bar{\mathbf{u}}$, and the stress field $\bar{\sigma} = \tilde{\sigma}(\bar{\varepsilon}) = \tilde{\sigma}(\mathbf{B}\bar{\mathbf{u}})$. Then, the Taylor expansion of the internal forces around $\bar{\mathbf{u}}$ yields

$$\bar{\mathbf{f}}^{int}(\bar{\mathbf{u}} + \Delta\bar{\mathbf{u}}) = \bar{\mathbf{f}}^{int}(\bar{\mathbf{u}}) + \left. \frac{\partial\bar{\mathbf{f}}^{int}}{\partial\bar{\mathbf{u}}} \right|_{\bar{\mathbf{u}}=\bar{\mathbf{u}}} \Delta\bar{\mathbf{u}} + \dots \quad (8.7)$$

We again neglect the quadratic and higher order terms, leading to

$$\bar{\mathbf{f}}^{int}(\bar{\mathbf{u}} + \Delta\bar{\mathbf{u}}) \approx \bar{\mathbf{f}}^{int}(\bar{\mathbf{u}}) + \mathbf{K}_T(\bar{\mathbf{u}})\Delta\bar{\mathbf{u}} \quad (8.8)$$

where

$$\mathbf{K}_T \equiv \frac{\partial\bar{\mathbf{f}}^{int}}{\partial\bar{\mathbf{u}}}$$

(8.9)

is the **tangent stiffness matrix** of the structure Differentiating Eq. 8.5

$$\mathbf{K}_T = \frac{\partial\bar{\mathbf{f}}^{int}}{\partial\bar{\mathbf{u}}} = \frac{\partial}{\partial\bar{\mathbf{u}}} \int_{\Omega} \mathbf{B}^T \tilde{\sigma} d\Omega = \int_{\Omega} \mathbf{B}^T \frac{\partial\tilde{\sigma}}{\partial\bar{\mathbf{u}}} d\Omega = \int_{\Omega} \mathbf{B}^T \frac{\partial\tilde{\sigma}}{\partial\varepsilon} \frac{\partial\varepsilon}{\partial\bar{\mathbf{u}}} d\Omega = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega \quad (8.10)$$

which is the well known formula for the stiffness matrix, however \mathbf{D}_{el} is now replaced by the tangent moduli \mathbf{D}

8.1.2 Solution Strategies

Before we discuss solution strategies, it may be helpful to point out the parallelism which exists between (numerical) solution strategies, and (experimental) testing methods. Modern testing equipment can be programmed to apply a pre-determined rate of load (as measured by a load cell), of displacement (as measured by an internal displacement transducer), or of strain (or relative displacement such as crack mouth opening displacement) measured by a strain/clip gage or other instruments, Fig. 8.1

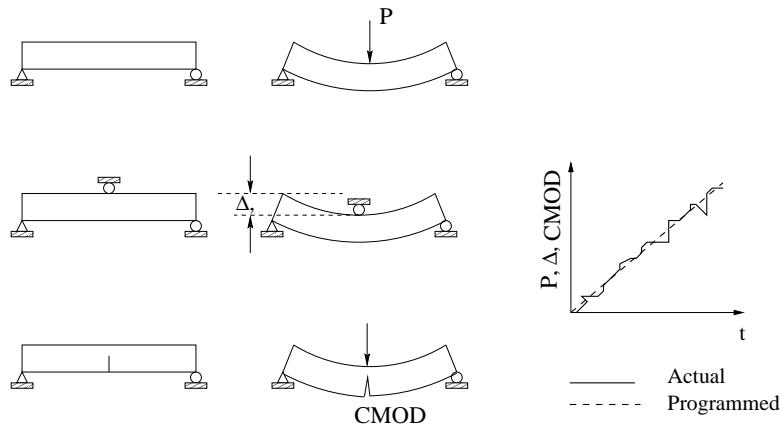


Figure 8.1: Test Controls

Load Control: the cross-head applies an increasing load irrespective of the specimen deformation or response. For all materials, when the tensile strength is reached, there is a sudden and abrupt brittle failure. The strain energy accumulated in the specimen is suddenly released once the ultimate load of the specimen is reached, thus the sudden failure can be explosive.

Displacement/Stroke Control: the cross-head applies an increasing displacement to the specimen. For softening material there will be a post-peak response with a gradual decrease in stress accompanying an increase in displacement. In this case, there is a gradual release of strain energy which is then transferred to surface energy during crack formation.

Strain Control: is analogous to displacement control, except that the feedback is provided by (“strategically positioned”) strain gage or a clip gage or an arbitrary specimen deformation (not necessarily corresponding to the loading direction). To accomplish this test a clip gage or a strain gage has to provide the feedback signal to the testing equipment in order to accordingly adjust the stroke.

Similarly, the objective of a nonlinear finite element analysis is to **trace** the (nonlinear) response of a structure subjected a given **load history**. This is best done in an **incremental-iterative** procedure where the load (or the displacement) is applied through several increments, and within each increment we seek to satisfy equilibrium through an iterative procedure (caused by the nonlinearity of the problem). The incremental analysis can be performed under

1. Load control; Load is incrementally applied on the structure.
2. Direct displacement control; An imposed displacement is applied.
3. Indirect displacement control (such as relative displacements between two degrees of freedom)
4. Arc-Length control

Alternatively, iterative techniques include

1. Newton-Raphson
2. Modified Newton-Raphson
3. Initial Stiffness
4. Secant Newton

Finally, an essential ingredient of an incremental-iterative solution strategy are

1. Convergence Criteria
2. Convergence Accelerators (such as line-search or step-size adjustments).

8.2 Load Control

8.2.1 Newton-Raphson

For the sake of discussion, we will assume in the following sections that the incremental analysis is under load control, with increments of loads $\Delta \bar{\mathbf{f}}^{ext}$. At the end of each load increment, internal forces must be in equilibrium with the external ones. Hence, we define the vector of **residual forces** $\bar{\mathcal{R}}$ as

$$\bar{\mathcal{R}}_{n+1} \equiv \bar{\mathcal{R}}(\bar{\mathbf{u}}_{n+1}) = \bar{\mathbf{f}}^{int}(\bar{\mathbf{u}}_{n+1}) - \bar{\mathbf{f}}^{ext} = 0 \quad (8.11)$$

where $\bar{\mathbf{f}}^{int}$ is the vector of internal forces, also commonly known as **reaction vector**. For equilibrium to be satisfied, the vector of **reactions** internal forces $\bar{\mathbf{f}}^{int}$ must be equal to the one of external ones $\bar{\mathbf{f}}^{ext}$. This is automatically satisfied in linear elastic analysis, but not necessarily so in nonlinear analyses. We start the analysis from an equilibrium configuration, at the end of increment n such that

$$\bar{\mathbf{u}} = \bar{\mathbf{u}}_n, \quad \bar{\mathcal{R}}_n = 0 \quad (8.12)$$

and apply an increment of load $\Delta \bar{\mathbf{f}}_n^{ext}$ such that

$$\bar{\mathbf{f}}_{n+1}^{ext} = \bar{\mathbf{f}}_n^{ext} + \Delta \bar{\mathbf{f}}_n^{ext} \quad (8.13)$$

and we seek to determine the corresponding change in displacement

$$\bar{\mathbf{u}}_{n+1} = \bar{\mathbf{u}}_n + \Delta \bar{\mathbf{u}}_n \quad (8.14)$$

We will keep $\Delta \bar{\mathbf{f}}_n^{ext}$ reasonably small to capture the full nonlinear response.

8.2.1.1 Newton-Raphson/Tangent Stiffness Method

This is the most rapidly convergent process (albeit computationally expensive) of non-linear problems. At the beginning of each step $n+1$, we start from the displacement $\bar{\mathbf{u}}_n$ that were computed in the previous step through equilibrium $\bar{\mathcal{R}}_n \approx 0$ or $\bar{\mathbf{f}}_n^{int} \approx \bar{\mathbf{f}}_n^{ext}$. The external forces are now increased from $\bar{\mathbf{f}}_n^{ext}$ to $\bar{\mathbf{f}}_{n+1}^{ext} = \bar{\mathbf{f}}_{n+1}^{ext} + \Delta \bar{\mathbf{f}}_n^{ext}$, and we seek to determine the corresponding displacements $\bar{\mathbf{u}}_{n+1}$ through equilibrium $\bar{\mathcal{R}}_{n+1} \approx 0$ or $\bar{\mathbf{f}}_{n+1}^{int} \approx \bar{\mathbf{f}}_{n+1}^{ext}$. Within the current step (identified through the subscript n), we will be iterating (through superscript k) in order to achieve equilibrium. As initial guess for $\bar{\mathbf{u}}_{n+1}^0$ we take it to be $\bar{\mathbf{u}}_n$ and based on the linearization around this initial state we have

$$\bar{\mathbf{f}}_{int}(\bar{\mathbf{u}}_{n+1}^0) + \mathbf{K}_T(\bar{\mathbf{u}}_{n+1}^0)\Delta \bar{\mathbf{u}}_{n+1}^1 = \bar{\mathbf{f}}_{n+1}^{ext} \quad (8.15)$$

where $\Delta \bar{\mathbf{u}}_{n+1}^1$ is the first approximation for the unknown displacement increment $\Delta \bar{\mathbf{u}}_{n+1} = \bar{\mathbf{u}}_{n+1} - \bar{\mathbf{u}}_n$.

Alternatively, we begin from a linearization of Eq. 8.11, Fig. 8.2

$$\bar{\mathcal{R}}(\bar{\mathbf{u}}_{n+1}^{i+1}) \approx \bar{\mathcal{R}}(\bar{\mathbf{u}}_n^i) + \left(\frac{\partial \bar{\mathcal{R}}}{\partial \bar{\mathbf{u}}} \right)_{n+1}^i \delta \bar{\mathbf{u}}_n^i = 0 \quad (8.16)$$

where i is a counter starting from $\bar{\mathbf{u}}_{n+1}^1 = \bar{\mathbf{u}}_n$. Observing that

$$\frac{\partial \bar{\mathcal{R}}}{\partial \bar{\mathbf{u}}} = \frac{\partial \bar{\mathbf{f}}^{int}}{\partial \bar{\mathbf{u}}} = \mathbf{K}_T \quad (8.17)$$

assuming that $\bar{\mathbf{f}}^{ext}$ is constant, and \mathbf{K}_T is the tangent stiffness matrix. Thus, Eq. 8.16 yields

$$\mathbf{K}_T^i \delta \bar{\mathbf{u}}_n^i = -\bar{\mathcal{R}}_{n+1}^i \quad (8.18)$$

or

$$\delta \bar{\mathbf{u}}_n^i = -(\mathbf{K}_T^i)^{-1} \bar{\mathcal{R}}_{n+1}^i \quad (8.19)$$

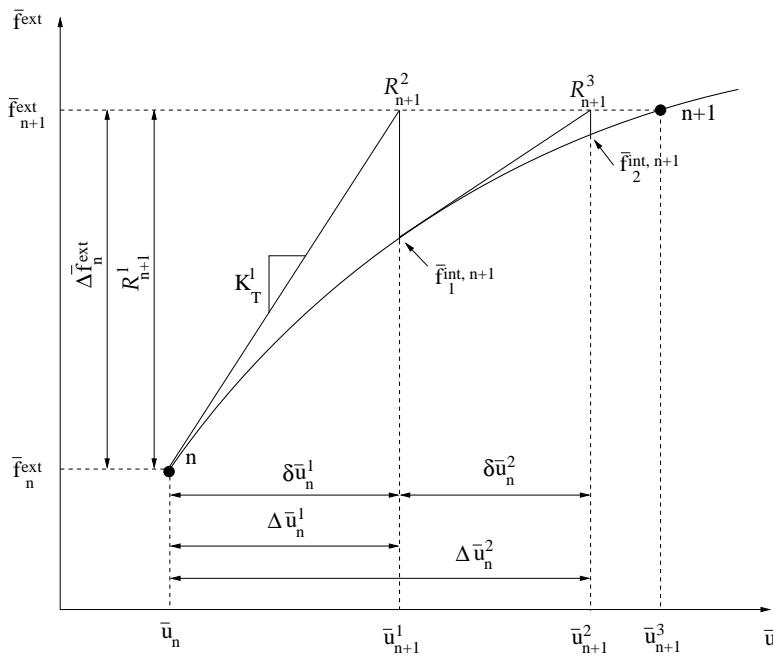


Figure 8.2: Newton-Raphson Method

Thus, a series of successive approximations yields

$$\bar{\mathbf{u}}_{n+1}^{i+1} = \bar{\mathbf{u}}_n + \Delta \bar{\mathbf{u}}_n^i = \bar{\mathbf{u}}_{n+1}^i + \delta \bar{\mathbf{u}}_n^i \quad (8.20)$$

with

$$\Delta \bar{\mathbf{u}}_n^i = \sum_{k \leq i} \delta \bar{\mathbf{u}}_n^k \quad (8.21)$$

very rapidly. It should be noted that each iteration involves three computationally expensive steps:

1. Evaluation of internal forces $\bar{\mathbf{f}}^{int}$ (or reactions)
2. Evaluation of the global tangent stiffness matrix \mathbf{K}_T
3. Solution of a system of linear equations

8.2.1.2 Modified Newton-Raphson

This method is essentially the same as the Newton-Raphson however in Eq. 8.23 (\mathbf{K}_T^i) is replaced by $\bar{\mathbf{K}}_T$ which is the tangent stiffness matrix of the first iteration of either 1) the first increment $\bar{\mathbf{K}}_T = \mathbf{K}_{T,0}^1$, Fig. 8.4, or 2) current increment, Fig. 8.3 $\bar{\mathbf{K}}_T = \mathbf{K}_{T,n}^1$ Fig. 8.3

$$\delta \bar{\mathbf{u}}_n^i = -(\bar{\mathbf{K}}_T)^{-1} \bar{\mathcal{R}}_{n+1}^i \quad (8.22)$$

In general the cpu time required for the extra iterations required by this method is less than the one saved by the assembly and decomposition of the stiffness matrix for each iteration. It should be mentioned that the tangent stiffness matrix does not necessarily have to be the true tangent stiffness matrix; an approximation of the true tangent stiffness matrix or even the initial stiffness matrix will generally produce satisfactory results, albeit at the cost of additional iterations.

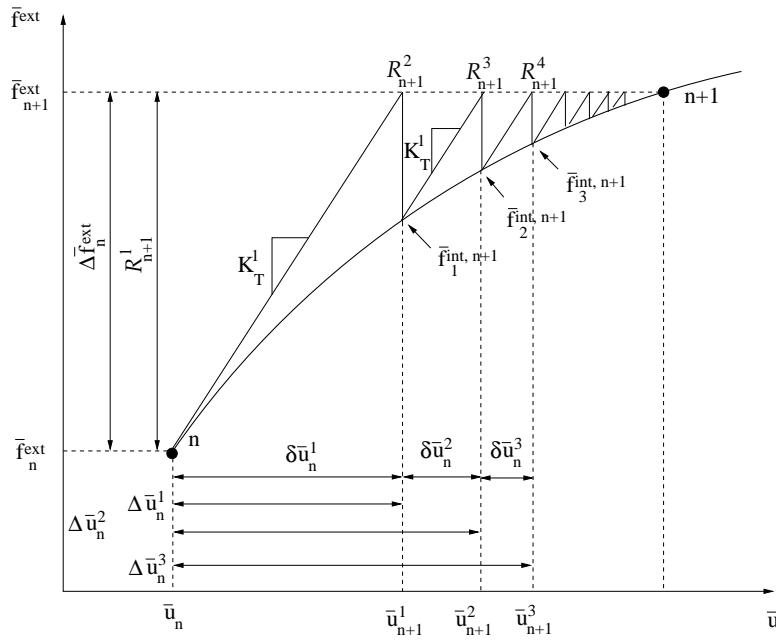


Figure 8.3: Modified Newton-Raphson Method, Initial Tangent in Increment

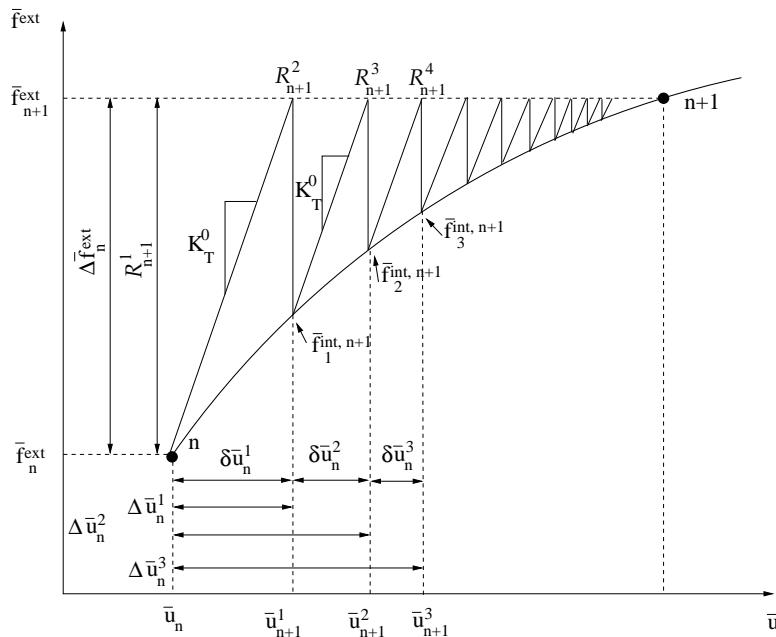


Figure 8.4: Modified Newton-Raphson Method, Initial Problem Tangent

8.2.1.3 Secant Newton

This method is a compromise between the first two. First we seek two displacements by two cycles of modified Newton-Raphson, then a secant to the curve is established between those two points, and a step taken along it, Fig. 8.5.

$$\delta \bar{u}_n^i = -(\bar{K}_T)^{-1} \bar{\mathcal{R}}_{n+1}^i \quad (8.23)$$

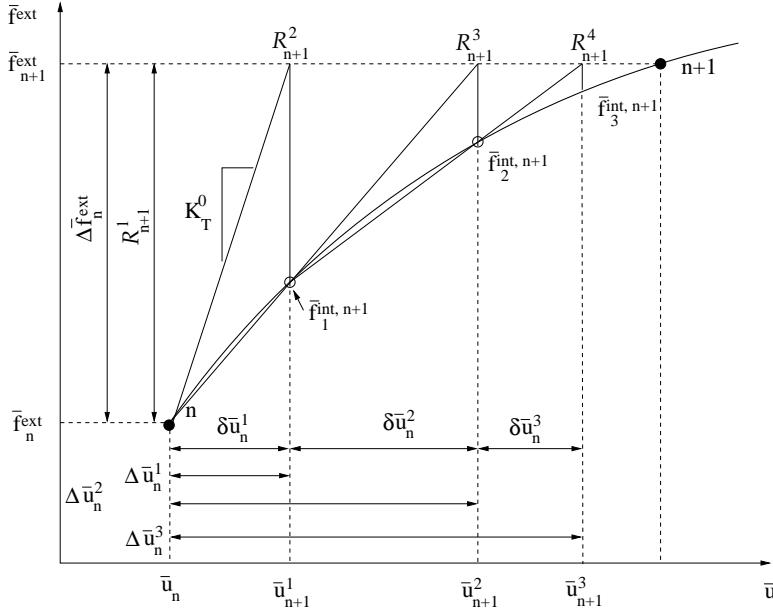


Figure 8.5: Incremental Secant, Quasi-Newton Method

Subsequently, each step will be taken along a secant connecting the previous two points. Hence, starting with

$$\delta \bar{u}_n^1 = -\bar{K}_T^{-1} \bar{\mathcal{R}}_{n+1}^1 \quad (8.24)$$

the secant slope can be determined

$$(\bar{K}_S^2)^{-1} = -\frac{\delta \bar{u}_n^1}{(\bar{\mathcal{R}}_{n+1}^1 - \bar{\mathcal{R}}_{n+1}^2)} \quad (8.25)$$

and then

$$\delta \bar{u}_n^2 = -(\bar{K}_S^2)^{-1} \bar{\mathcal{R}}_{n+1}^2 \quad (8.26)$$

This process can be generalized to

$$\begin{aligned} \delta \bar{u}_n^i &= -(\bar{K}_S^i)^{-1} \bar{\mathcal{R}}_{n+1}^i \\ (\bar{K}_S^i)^{-1} &= -\frac{\delta \bar{u}_n^i}{(\bar{\mathcal{R}}_{n+1}^{i-1} - \bar{\mathcal{R}}_{n+1}^i)} \end{aligned} \quad (8.27)$$

8.2.2 Acceleration of Convergence, Line Search Method

Adapted from (Reich 1993)

The line search is an iterative technique for automatically under- or over-relaxing the displacement corrections $\delta \bar{u}^j$ so as to accelerate the convergence of nonlinear solution algorithms. The amount of under- or over-relaxation is determined by enforcing an orthogonality condition between the displacement corrections $\delta \bar{u}^j$ and the residual loads $\bar{\mathcal{R}}^{j+1}$, which amounts to forcing the iterative change in energy to be zero. The displacement corrections are

multiplied by a scalar value s^k defining the amount of under- or over-relaxation such that the total displacements $\bar{\mathbf{u}}^{j+1,k}$ are defined as

$$\bar{\mathbf{u}}^{j+1,k} = \bar{\mathbf{u}}^j + s^k \delta\bar{\mathbf{u}}^j \quad (8.28)$$

For $k = 0$ and $k = 1$, the values of s^k are 0.0 and 1.0, respectively. Therefore, $\bar{\mathbf{u}}^{j+1,0} = \bar{\mathbf{u}}^j$ and $\bar{\mathbf{u}}^{j+1,1} = \bar{\mathbf{u}}^{j+1}$. The orthogonality condition is quantified by a scalar value g^k representing the iterative change in energy, which is defined as

$$g^k = \delta\bar{\mathbf{u}}^j \cdot \bar{\mathcal{R}}^{j+1,k} \quad (8.29)$$

where

$$\bar{\mathcal{R}}^{j+1,k} = \bar{\mathbf{f}}^{ext} - \bar{\mathbf{f}}^{int}(\bar{\mathbf{u}}^{j+1,k}) \quad (8.30)$$

are the residual loads at the end of solution iteration j and line search iteration k . g^k can be expressed as a function of s^k (see Figure 8.6) and the object of the line search is to find s^k such that g^k is zero. An estimate of s^{k+1} such that g^{k+1} is zero can be computed using a simple extrapolation procedure based on similar triangles

$$\frac{s^{k+1}}{g^0} = \frac{s^k}{g^0 - g^k} \quad (8.31)$$

On rearranging terms, s^{k+1} is defined as

$$s^{k+1} = s^k \left(\frac{g^0}{g^0 - g^k} \right) \quad (8.32)$$

As a preventative measure, s^{k+1} is assigned a value of 5.0 for all $s^{k+1} > 5.0$ so that unrestrained over-relaxation is inhibited. Once s^{k+1} is estimated, $\bar{\mathbf{u}}^{j+1,k+1}$, $\bar{\mathbf{f}}^{int}_{j+1,k+1}$, and $\bar{\mathcal{R}}_{j+1,k+1}$ are computed for the next line search iteration, Fig. 8.6. The line search terminates after three iterations or when

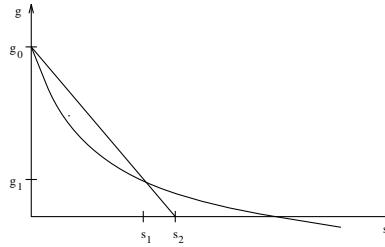


Figure 8.6: Schematic of Line Search, (Reich 1993)

$$\frac{|g^0|}{|g^k|} \leq 0.8 \quad (8.33)$$

and $|g^0| g^k \leq 0.001 |g^0|$. Smaller tolerances may be used to determine if the line search has converged, ? prefer to use 0.6, but Crisfield, M.A. (1979) concluded that there was little advantage to be gained by doing such.

The flowchart illustrating the Line Search algorithm is shown in Fig. 8.7.

8.2.3 Convergence Criteria

In all preceding methods, iterations are performed until one or all of a variety of convergence criteria are satisfied. Relative convergence criteria are optionally enforced on the displacements, loads, and/or incremental energy to define the termination conditions. The relative displacement criteria is defined in terms of the displacement corrections $\delta\bar{\mathbf{u}}^j$ and the updated incremental displacements $\delta\bar{\mathbf{u}}^{j+1}$ as

$$\epsilon_u = \frac{\|\delta\bar{\mathbf{u}}^j\|_2}{\|\Delta\bar{\mathbf{u}}^{j+1}\|_2} \quad (8.34)$$

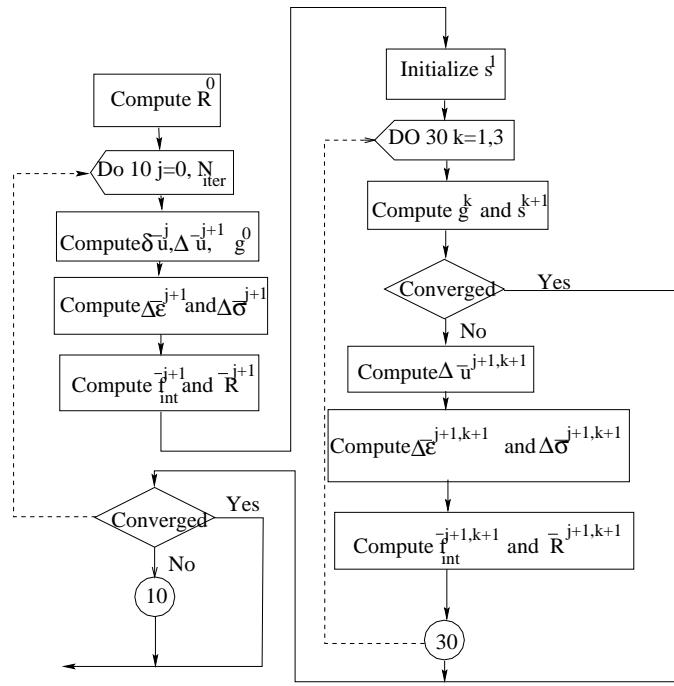


Figure 8.7: Flowchart for Line Search Algorithm, (Reich 1993)

where $\| \dots \|_2$ is the Euclidean norm. The Euclidean norm, which is also known as the L_2 norm, of a vector \mathbf{v} is defined as

$$\| \mathbf{v} \|_2 = \left[\sum_{i=1}^N v_i^2 \right]^{1/2} \quad (8.35)$$

where N is the size of \mathbf{v} . The relative load criteria is defined in terms of the updated residual loads $\bar{\mathcal{R}}^{j+1}$ and the reactions $\bar{\mathbf{f}}_{j+1}^{int}$ as either

$$\epsilon_r = \frac{\| \bar{\mathcal{R}}^{j+1} \|_2}{\| \bar{\mathbf{f}}_{j+1}^{int} \|_2} \quad (8.36)$$

or

$$\epsilon_r = \frac{\| \bar{\mathcal{R}}^{j+1} \|_\infty}{\| \bar{\mathbf{f}}_{j+1}^{int} \|_\infty} \quad (8.37)$$

where $\| \dots \|_\infty$ is the infinity norm. The infinity norm of a vector \mathbf{v} is defined as

$$\| \mathbf{v} \|_\infty = \max \left(\sum_{i=1}^N |v_i| \right) \quad (8.38)$$

where N is the size of \mathbf{v} . The relative incremental energy criteria is defined in terms of displacement corrections $\delta\bar{\mathbf{u}}^j$, the updated residual loads $\bar{\mathcal{R}}^{j+1}$, the updated incremental displacements $\Delta\bar{\mathbf{u}}^{j+1}$, and the updated reactions $\bar{\mathbf{f}}_{j+1}^{int}$ as

$$\epsilon_W = \frac{\delta\bar{\mathbf{u}}^j \cdot \bar{\mathcal{R}}^{j+1}}{\Delta\bar{\mathbf{u}}^{j+1} \cdot \bar{\mathbf{f}}_{j+1}^{int}} \quad (8.39)$$

where the numerator is the change in the incremental energy for iteration j and denominator is the incremental energy.

8.3 Direct Displacement Control

Adapted from (Jirasek and Bažant 2001)

Independently of the choice of iterative algorithm, any solution strategy using load control fails if the prescribed external loads cannot be maintained in equilibrium by the internal forces. This would typically occur if the load is monotonically increased until the load-carrying capacity of the structure is exhausted, Fig. 8.8 In most engineering

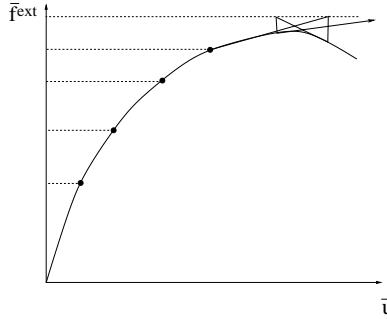


Figure 8.8: Divergence of Load-Controlled Algorithms

analyses, it is simply required to determine the maximum load carrying capacity, and the corresponding displacements. As such, divergence of the iterative process is often taken as an indicator of structural failure, and the last converged step provides information on the state prior to collapse. However, finite element simulations of complex engineering problems can diverge for a number of other reasons, many of which are purely numerical and have nothing to do with the real structural failure.

If the load-displacement diagram is to be followed beyond the peak, i.e. **post-peak** response is required, then alternative solution strategy to the load-control one must be devised. Post-peak response may be of interest not only in problems in structures with imposed displacements (such as initial settlements), but also to assess the ductility of the structure (specially when cracks are present).

To outline the displacement controlled algorithm, we divide the displacements into two groups: one with unknown displacements at nodes that are left “free”, and the second with prescribed displacements at nodes that are controlled. Accordingly, we partition the displacement vector into $\{\bar{u}_f, \bar{u}_p\}^T$ and the internal and external force vectors into $\{\bar{f}_{int,f}, \bar{f}_{int,p}\}^T$ and $\{\bar{f}_{ext,f}, \bar{f}_{ext,p}\}^T$, respectively. External forces $\bar{f}_{ext,f}$ (corresponding to the unknown displacements \bar{u}_f) are prescribed, and for simplicity we will assume that they are equal to zero. All external forces acting on the structure are represented by reactions $\bar{f}_{ext,p}$ at the supports with prescribed displacements \bar{u}_p . Hence, the equilibrium equations are partitioned as

$$\bar{f}_{int,f}(\bar{u}_f, \bar{u}_p) = \mathbf{0} \quad (8.40-a)$$

$$\bar{f}_{int,p}(\bar{u}_f, \bar{u}_p) = \bar{f}_{ext,p} \quad (8.40-b)$$

For given \bar{u}_p , the unknown displacements \bar{u}_f can be computed by solving Eq. 8.40-a. After that, the reactions $\bar{f}_{ext,p}$ are obtained by simple evaluation of the left-hand side in (8.40-b).

In a typical incremental step number n , we start from the converged displacements $\bar{u}_f^{(n-1)}$ and $\bar{u}_p^{(n-1)}$ from the previous step, and we replace Eq. 8.40-a by the linearized equations

$$\bar{f}_f^{(n-1)} + \mathbf{K}_{11}^{(n-1)} \Delta \bar{u}_f^{(n,1)} + \mathbf{K}_{12}^{(n-1)} \Delta \bar{u}_p^{(n,1)} = \mathbf{0} \quad (8.41)$$

where $\mathbf{K}_{11} \equiv \frac{\partial \bar{f}_{int,f}}{\partial \bar{u}_f}$ and $\mathbf{K}_{12} \equiv \frac{\partial \bar{f}_{int,f}}{\partial \bar{u}_p}$ are blocks of the global tangent stiffness matrix

$$\mathbf{K} \equiv \frac{\partial \bar{f}_{int}}{\partial \bar{u}} = \begin{bmatrix} \frac{\partial \bar{f}_{int,f}}{\partial \bar{u}_f} & \frac{\partial \bar{f}_{int,f}}{\partial \bar{u}_p} \\ \frac{\partial \bar{f}_{int,p}}{\partial \bar{u}_f} & \frac{\partial \bar{f}_{int,p}}{\partial \bar{u}_p} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \quad (8.42)$$

The increment of the prescribed displacements \bar{u}_p is known in advance, and so we set $\Delta \bar{u}_p^{(n,1)} = \Delta \bar{u}_p^{(n)} = \bar{u}_p^{(n)} - \bar{u}_p^{(n-1)}$ and rewrite (8.41) as

$$\mathbf{K}_{11}^{(n-1)} \Delta \bar{u}_f^{(n,1)} = -\bar{f}_{int,f}^{(n-1)} - \mathbf{K}_{12}^{(n-1)} \Delta \bar{u}_p^{(n)}$$

Having solved for $\Delta\bar{\mathbf{u}}_f^{(n,1)}$, we construct the first approximation $\bar{\mathbf{u}}_f^{(n,1)} = \bar{\mathbf{u}}_f^{(n-1)} + \Delta\bar{\mathbf{u}}_f^{(n,1)}$ and $\bar{\mathbf{u}}_p^{(n,1)} = \bar{\mathbf{u}}_p^{(n-1)} + \Delta\bar{\mathbf{u}}_p^{(n,1)} = \bar{\mathbf{u}}_p^{(n)}$. Equations (8.40-a) are then linearized around $(\bar{\mathbf{u}}_f^{(n,1)}, \bar{\mathbf{u}}_p^{(n,1)})$, corrections of displacements $\bar{\mathbf{u}}_f$ are computed, and the procedure is repeated until the convergence criteria are satisfied. The iterative process can be described by recursive formulas

$$\left. \begin{aligned} \mathbf{K}_{11}^{(n,i-1)} \delta\bar{\mathbf{u}}_f^{(n,i)} &= -\bar{\mathbf{f}}_{int}^{(n,i-1)} - \mathbf{K}_{12}^{(n,i-1)} \delta\bar{\mathbf{u}}_p^{(n,i)} \\ \bar{\mathbf{u}}_f^{(n,i)} &= \bar{\mathbf{u}}_f^{(n,i-1)} + \delta\bar{\mathbf{u}}_f^{(n,i)} \end{aligned} \right\} \quad i = 1, 2, 3, \dots$$

where

$$\bar{\mathbf{u}}_f^{(n,0)} = \bar{\mathbf{u}}_f^{(n-1)} \quad (8.43-a)$$

$$\bar{\mathbf{u}}_p^{(n,0)} = \bar{\mathbf{u}}_p^{(n-1)} \quad (8.43-b)$$

$$\delta\bar{\mathbf{u}}_p^{(n,1)} = \bar{\mathbf{u}}_p^{(n)} - \bar{\mathbf{u}}_p^{(n-1)} \quad (8.43-c)$$

$$\delta\bar{\mathbf{u}}_p^{(n,i)} = \mathbf{0} \quad \text{for } i = 2, 3, \dots \quad (8.43-d)$$

Note that, starting from the second iteration, the correction $\delta\bar{\mathbf{u}}_p$ is zero, and so the term with \mathbf{K}_{12} on the right-hand side of (??) vanishes. This term is present only in the first iteration. It might seem that one could start immediately from $\bar{\mathbf{u}}_p^{(n,0)} = \bar{\mathbf{u}}_p^{(n)}$ instead of $\bar{\mathbf{u}}_p^{(n,0)} = \bar{\mathbf{u}}_p^{(n-1)}$, and then the correction $\delta\bar{\mathbf{u}}_p$ would be zero already in the first iteration and the matrix \mathbf{K}_{12} would never have to be evaluated. However, this is in general not a good idea because such an initial approximation would be too far from the equilibrium path and the process might diverge.

8.4 Indirect Displacement Control

Direct displacement control can be applied only on structures loaded only at one point, or when the load is transmitted by a stiff platen so that all points on the loaded surface exhibit the same displacements. However, this is not always the case. As an example, consider a dam loaded by hydrostatic pressure due to reservoir overflow; see Fig. 8.9. Here,

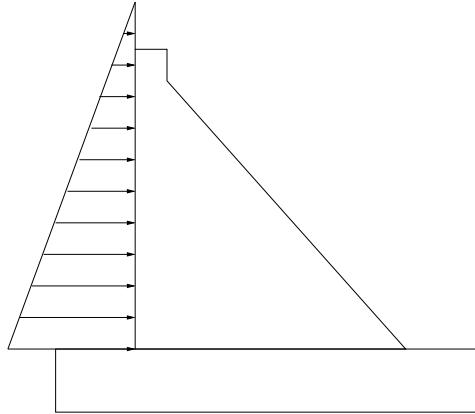


Figure 8.9: Hydrostatically Loaded Gravity Dam

the load is applied along a large portion of the boundary, and the shape of the corresponding displacement profile is not known in advance. Another case in which direct displacement control fails is very brittle failure characterized by a load-displacement diagram with a snapback, Fig. 8.10.

Advanced incrementation control techniques abandon the assumption that the values of external loads and/or displacements at supports after each incremental step are prescribed in advance. Instead, the loading program is **parameterized by a scalar load multiplier**.

8.4.1 Partitioning of the Displacement Corrections

Adapted from (Reich 1993)

Restricting the applied loading to be proportional, a scalar load parameter β can be used to scale an arbitrary set of applied loads $\bar{\mathbf{f}}^{ext}$. The applied loads at the start of increment i are defined as the scalar-vector product $\beta_i \bar{\mathbf{f}}^{ext}$, where β_i is the load parameter at the start of increment i . β_i is zero at the start of the first increment. The applied

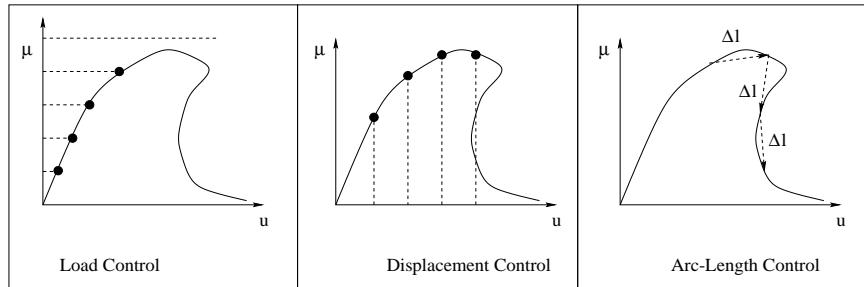


Figure 8.10: Load-Displacement Diagrams with Snapback

incremental loads for increment i are defined as the scalar-vector product $\Delta\beta_i \bar{\mathbf{f}}^{ext}$, where $\Delta\beta_i$ is the incremental load parameter for increment i . The updated load parameter β_{i+1} at the end of increment i is

$$\beta_{i+1} = \beta_i + \Delta\beta_i \quad (8.44)$$

The incremental displacements due to the applied incremental loads are obtained using the standard modified-Newton algorithm, as described in Zienkiewicz & Taylor (1991). The incremental displacements $\Delta\bar{\mathbf{u}}^{j+1}$ at the end of iteration j for a generic increment are defined as

$$\Delta\bar{\mathbf{u}}^{j+1} = \Delta\bar{\mathbf{u}}^j + \delta\bar{\mathbf{u}}^j \quad (8.45)$$

where $\Delta\bar{\mathbf{u}}^j$ are the incremental displacements at the start of iteration j and $\delta\bar{\mathbf{u}}^j$ are the incremental displacement corrections for iteration j . The incremental load parameter $\Delta\beta^{j+1}$ at the end of iteration j is defined in an analogous manner as

$$\Delta\beta^{j+1} = \Delta\beta^j + \delta\beta^j \quad (8.46)$$

where $\Delta\beta^j$ is the incremental load parameter at the start of iteration j and $\delta\beta^j$ is the incremental load parameter correction for iteration j . At the start of the first iteration $\Delta\bar{\mathbf{u}}^j$ and $\Delta\beta^j$ are identically zero. Incremental displacement corrections are determined by solving

$$\mathbf{K} \delta\bar{\mathbf{u}}^j = (\beta \bar{\mathbf{f}}^{ext} + \Delta\beta^j \bar{\mathbf{f}}^{ext} + \delta\beta^j \bar{\mathbf{f}}^{ext} - \bar{\mathbf{f}}_{int}^j) \quad (8.47)$$

where \mathbf{K} is the global stiffness matrix and

$$\bar{\mathbf{f}}_{int}^j = \sum_{e=1}^{N_{elem}} \int_{\Omega_e} \mathbf{B}^T \mathbf{D}(\boldsymbol{\epsilon} + \Delta\boldsymbol{\epsilon}^j) \delta\Omega \quad (8.48)$$

are the reactions for the state of stress at the start of iteration j . Defining the residual forces $\bar{\mathcal{R}}^j$ at the start of iteration j as

$$\bar{\mathcal{R}}^j = \beta \bar{\mathbf{f}}^{ext} + \Delta\beta^j \bar{\mathbf{f}}^{ext} - \bar{\mathbf{f}}_{int}^j \quad (8.49)$$

Equation 8.47 can be written more simply as

$$\delta\bar{\mathbf{u}}^j = \mathbf{K}^{-1} (\delta\beta^j \bar{\mathbf{f}}^{ext} + \bar{\mathcal{R}}^j) \quad (8.50)$$

The matrix-vector product $\mathbf{K}^{-1} \bar{\mathbf{f}}^{ext}$ is invariant for the increment and, therefore, can be treated as a vector constant $\delta\bar{\mathbf{u}}_T$, which Crisfield (1981) referred to as the tangent displacements

$$\delta\bar{\mathbf{u}}_T = \mathbf{K}^{-1} \bar{\mathbf{f}}^{ext} \quad (8.51)$$

The matrix-vector product $\mathbf{K}^{-1} \bar{\mathcal{R}}^j$ defines the displacement corrections $\delta\bar{\mathbf{u}}_r^j$ due to the residual forces

$$\delta\bar{\mathbf{u}}_r^j = \mathbf{K}^{-1} \bar{\mathcal{R}}^j \quad (8.52)$$

but they are obviously not invariant for the increment. The displacement corrections for iteration j are then defined as

$$\delta\bar{\mathbf{u}}^j = \delta\beta^j \delta\bar{\mathbf{u}}_T + \delta\bar{\mathbf{u}}_r^j \quad (8.53)$$

Figure 8.11 shows a flowchart for an incremental nonlinear finite element program based on the modified-Newton algorithm with indirect displacement control capabilities. The numbers in the boxes in Figure 8.11 correspond to those appearing in Figure ??.

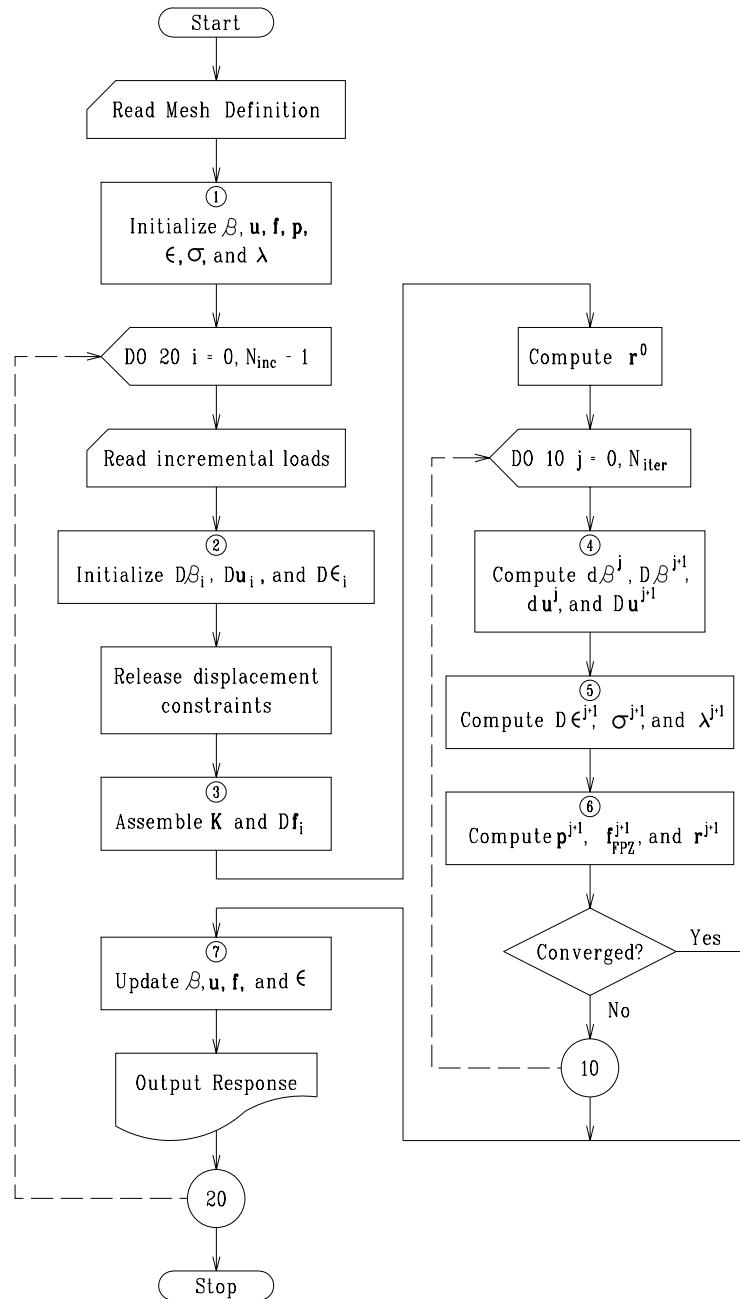


Figure 8.11: Flowchart for an incremental nonlinear finite element program with indirect displacement control

Figure 8.12: Two points on the load-displacement curve satisfying the arc-length constraint

8.4.2 Arc-Length

Adapted from (Jirasek and Bažant 2001)

The basic idea of a flexible incrementation control technique is that the step size is specified by a constraint equation that involves the unknown displacements as well as the load multiplier. The original motivation was provided by the requirement that the size of the step measured as the geometric distance between the initial and final state in the load-displacement space should be equal to a prescribed constant, Fig. 8.10.

Despite the apparent simplicity of the condition of a constant arc length, it must be used with caution. First of all, it is important to realize that forces and displacements have completely different units, and so the purely geometrical measure of length in the load-displacement space does not make a good sense. It is necessary to introduce at least one scaling factor, denoted as c , that multiplies the load parameter and converts it into a quantity with the physical dimension of displacement. The length of a step during which the load parameter changes by $\Delta\mu$ and the displacements change by $\Delta\bar{\mathbf{u}}$ is then defined as

$$\Delta l = \sqrt{\Delta\bar{\mathbf{u}}^T \Delta\bar{\mathbf{u}} + (c\Delta\mu)^2} \quad (8.54)$$

By adjusting the scaling factor we can amplify or suppress the relative contribution of loads and displacements. One reasonable choice is derived from the condition that the contributions should be equal as long as the response remains linear elastic, which leads to $c = \sqrt{\bar{\mathbf{u}}_e^T \bar{\mathbf{u}}_e}$ where $\bar{\mathbf{u}}_e$ is the solution of $\mathbf{K}_e \bar{\mathbf{u}}_e = \bar{\mathbf{f}}$. In some cases, e.g., for frame, plate, and shell models that use both translational and rotational degrees of freedom, the components of the generalized displacement vector $\bar{\mathbf{u}}$ do not have the same physical dimension. It is then necessary to apply scaling also to the vector $\Delta\bar{\mathbf{u}}$.

Consider an incremental solution process controlled by the *arc-length method*. In a typical step number n , we start with displacements $\bar{\mathbf{u}}^{(n-1)}$ and load parameter $\mu^{(n-1)}$ computed in the previous step, and we search for displacements $\bar{\mathbf{u}}^{(n)}$ and load parameter $\mu^{(n)}$. The state at the end of the step must satisfy the equations of equilibrium between the internal forces $\bar{\mathbf{f}}_{int}(\bar{\mathbf{u}}^{(n)})$ and external forces $\bar{\mathbf{f}}_{ext}(\mu^{(n)})$. Compared to the load control or direct displacement control, the load parameter is an additional unknown. The corresponding additional equation is provided by the constraint that fixes the size of the step. For example, we can require that the length of the step evaluated from formula (8.54) be equal to a prescribed value, $\bar{\Delta}l$. We could treat the problem as a system of $N_{df} + 1$ nonlinear equations, where N_{df} is the number of unknown displacement components (degrees of freedom), and solve it by Newton-Raphson iteration. However, a more elegant and computationally more efficient procedure treats the equilibrium equations and the constraint equation to a certain extent separately. Assume for simplicity that the loading program is described by (??). The linearized equations of equilibrium in the i -th iteration read

$$\mathbf{K}^{(n,i-1)} \delta\bar{\mathbf{u}}^{(n,i)} = \bar{\mathbf{f}}_0 + \mu^{(n,i-1)} \bar{\mathbf{f}} - \bar{\mathbf{f}}_{int}^{(n,i-1)} + \delta\mu^{(n,i)} \bar{\mathbf{f}} \quad (8.55)$$

where $\delta\bar{\mathbf{u}}^{(n,i)}$ is the unknown displacement correction, and $\delta\mu^{(n,i)}$ is the unknown correction of the load parameter. The first three terms on the right-hand side are known, and the last term is an unknown scalar multiple of a given vector $\bar{\mathbf{f}}$. We can therefore separately solve equations

$$\mathbf{K}^{(n,i-1)} \delta\bar{\mathbf{u}}_0 = \bar{\mathbf{f}}_0 + \mu^{(n,i-1)} \bar{\mathbf{f}} - \bar{\mathbf{f}}_{int}^{(n,i-1)} \quad (8.56-a)$$

$$\mathbf{K}^{(n,i-1)} \delta\bar{\mathbf{u}}_f = \bar{\mathbf{f}} \quad (8.56-b)$$

and then express the displacement correction as

$$\delta\bar{\mathbf{u}}^{(n,i)} = \delta\bar{\mathbf{u}}_0 + \delta\mu^{(n,i)} \delta\bar{\mathbf{u}}_f \quad (8.57)$$

When this expression is substituted into the constraint condition,

$$(\Delta\bar{\mathbf{u}}^{(n,i-1)} + \delta\bar{\mathbf{u}}^{(n,i)})^T (\Delta\bar{\mathbf{u}}^{(n,i-1)} + \delta\bar{\mathbf{u}}^{(n,i)}) + c^2 (\Delta\mu^{(n,i-1)} + \delta\mu^{(n,i)})^2 = (\bar{\Delta}l)^2 \quad (8.58)$$

we obtain a quadratic equation for a single unknown, $\delta\mu^{(n,i)}$. This equation usually has two real roots, corresponding to the two points of the equilibrium path that have the prescribed distance from point $(\bar{\mathbf{u}}^{(n-1)}, \mu^{(n-1)})$; see Fig. 8.12. The correct root is selected depending on the sense in which we march on the equilibrium path (Crisfield, M.A. 1981), and the displacement correction is determined from (8.57). After standard updates of the displacement vector and the load parameter, the iteration cycle is repeated until the convergence criteria are satisfied.

8.4.3 Relative Displacement Criterion

Adapted from (Reich 1993)

The standard arc-length control performs well if the entire structure or its large portion participates in the failure mechanism. In cases when the failure pattern is highly localized, robustness of the technique may deteriorate. The remedy is to adapt the constraint equation to the particular problem and control the incrementation process by a few carefully selected displacement components. Motivation is again provided by the physical background. If the load-displacement diagram of a brittle structure exhibits snapback, direct displacement control applied in an experiment leads to sudden catastrophic failure. When the displacement imposed by the loading device reaches a critical value, fracture starts propagating even though the imposed displacement at the load point is kept fixed. However, opening of the crack monotonically increases during the entire failure process, and so it can be used as a control variable. If the experimental setup is arranged such that the applied force is continuously adjusted depending on the currently measured value of the crack opening, the response can be traced in a stable manner even after the point at which the load-displacement diagram snaps back. The same idea can be exploited by a numerical simulation. It suffices to select a suitable linear combination of displacement components that increases monotonically during the entire failure process, and to use this combination as the control variable.

de Borst (1985,1986) concluded that arc-length methods (Riks 1979, Ramm 1981, Crisfield 1981), which were the original IDC methods, were not satisfactory for analyses involving cracking accompanied by softening. The main problem with the arc-length methods, when used in this context, was that the constraint involved all displacement components equally when, in fact, only a few displacement components were dominant. The dominant displacement components were typically those for nodes at or near the crack mouth. This being the case, de Borst proposed using a transformed relative displacement component between two nodes as the constraint. The transformed relative displacement component can define the crack mouth opening displacement (*CMOD*), crack mouth sliding displacement (*CMSD*), or some arbitrary displacement Δu between two points on a structure. The arbitrary displacement Δu may correspond to a relative displacement measured during an experiment such as the relative vertical displacement between a point on the neutral axis of a 3-point bend beam over a support and the bottom of the beam at mid-span.

As it is the most general case, the relative displacement criterion will be described in terms of the arbitrary relative displacement Δu . A pair of nodes, m and n , are selected to define Δu , with their total displacements being $(\bar{\mathbf{u}})_m$ and $(\bar{\mathbf{u}})_n$, respectively. The direction associated with Δu is defined by a unit vector \mathbf{v} . Δu is thereby defined as

$$\Delta u = \mathbf{v}^T [(\bar{\mathbf{u}})_n - (\bar{\mathbf{u}})_m] \quad (8.59)$$

If m and n are nodes on opposite sides of a discrete crack $\Delta u \equiv CMOD$ if \mathbf{v} is normal to the crack surface and $\Delta u \equiv CMSD$ if \mathbf{v} is tangent to the crack surface. The value for Δu is prescribed for an increment and the applied loads are scaled such that the total displacements at the end of each iteration reflect that value. Recalling that the total displacements $\bar{\mathbf{u}}^{j+1}$ at the end of iteration j are defined as

$$\bar{\mathbf{u}}^{j+1} = \bar{\mathbf{u}}^j + \delta\beta^j \delta\bar{\mathbf{u}}_T + \delta\bar{\mathbf{u}}_r^j \quad (8.60)$$

the load parameter correction $\delta\beta^j$ for iteration j is

$$\delta\beta^j = \frac{\Delta u - \mathbf{v}^T [(\bar{\mathbf{u}}^j)_n - (\bar{\mathbf{u}}^j)_m] - \mathbf{v}^T [(\delta\bar{\mathbf{u}}_r^j)_n - (\delta\bar{\mathbf{u}}_r^j)_m]}{\mathbf{v}^T [(\delta\bar{\mathbf{u}}_T)_n - (\delta\bar{\mathbf{u}}_T)_m]} \quad (8.61)$$

8.4.4 IDC Methods with Approximate Line Searches

Employing a procedure proposed by Crisfield (1983) for use with the arc-length method, the convergence of the solution algorithm can be accelerated by performing approximate line searches; approximate line searches under fixed (i.e. non-scalable) loads are described in Section ???. This procedure requires an extra iterative loop at the beginning of the line search loop in which a combination of $\delta\beta^j$ and s^{k+1} satisfying the constraint conditions (i.e. Equations ?? and 8.59) is computed. As $\delta\beta^j$ is initially computed for $s^1 = 1.0$, any change in s^k requires a corresponding change in $\delta\beta^j$ for the IDC constraint to remain satisfied. Consequently, an iterative loop, in which $\delta\beta^j$ is recomputed based on the estimated value of s^{k+1} , is required to obtain a compatible combination of $\delta\beta^j$ and s^{k+1} . After recomputing $\delta\beta^j$, the values of g^0 and g^k are also recomputed using Equation 8.29 to reflect the change in the residual loads $\bar{\mathcal{R}}^{j+1,k}$ caused by the new value of $\delta\beta^j$. $\bar{\mathbf{F}}_{int}^{j+1,k}$ is not updated to reflect the changes in s^{k+1} when recomputing $\bar{\mathcal{R}}^{j+1,k}$, which is strictly not correct, but it does significantly reduce the number of computations without causing any difficulties (Crisfield 1983). Finally, from the new values of g^0 and g^k , s^{k+1} is re-estimated using Equation 8.32. The loop is terminated when

$$\frac{|s_{new}^{k+1} - s^{k+1}|}{|s_{new}^{k+1}|} \leq 0.05 \quad (8.62)$$

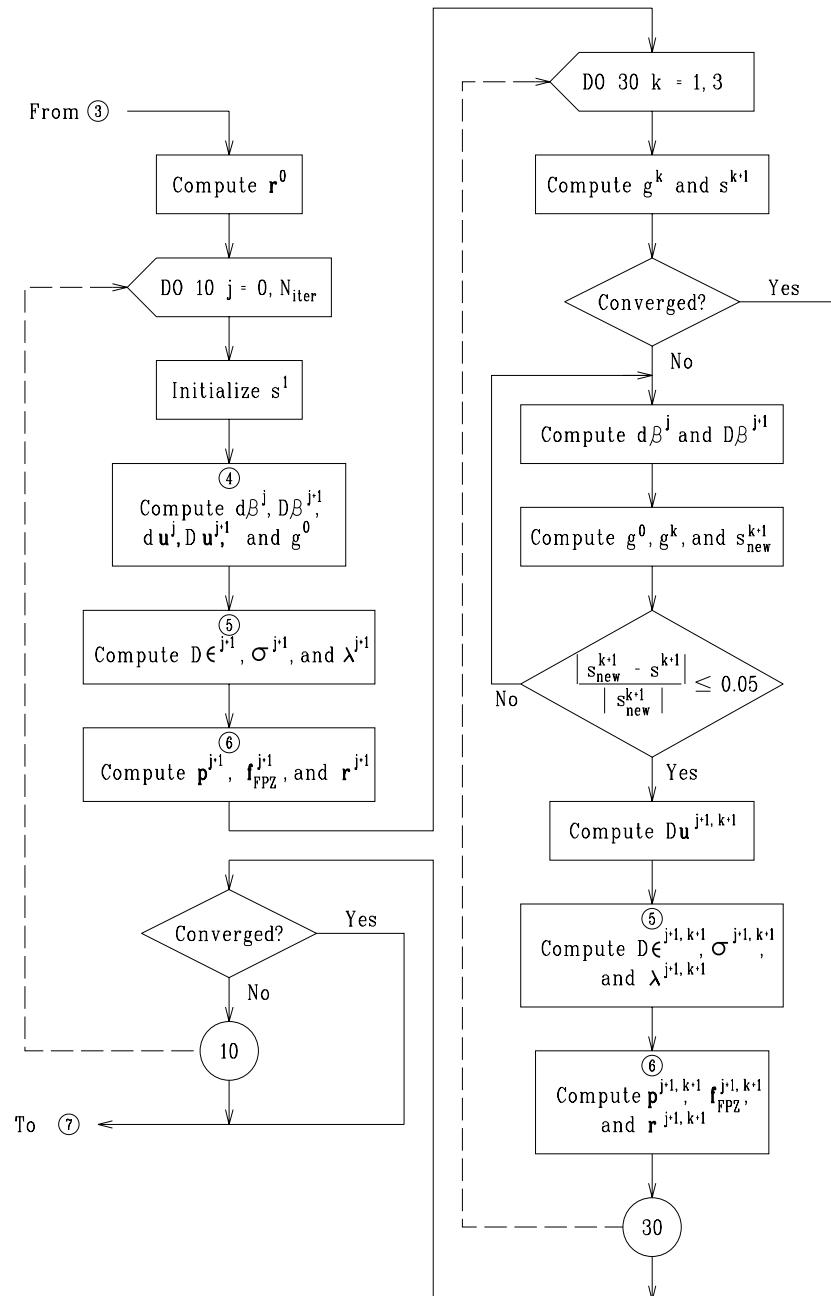


Figure 8.13: Flow chart for line search with IDC methods

which generally requires only a few iterations. A flow chart of this procedure is shown in Figure 8.13.

Since the total displacements $\bar{\mathbf{u}}^{j+1,k}$ are now defined as

$$\bar{\mathbf{u}}^{j+1,k} = \bar{\mathbf{u}}^j + s^k (\delta\bar{\mathbf{u}}_r^j + \delta\beta^j \delta\bar{\mathbf{u}}_T) \quad (8.63)$$

reflecting the introduction of the relaxation parameter s^k , the IDC constraint equations must be modified accordingly. $\delta\beta^j$ for the stress criterion is now defined as

$$\delta\beta^j = \min \left\{ \frac{f_t - [(\bar{\lambda}^j)_n + s^k (\delta\bar{\lambda}_r^j)_n] (\mathbf{n})_n}{s^k (\delta\bar{\lambda}_T)_n (\mathbf{n})_n} \right\} \quad (8.64)$$

and $\delta\beta^j$ for the relative displacement criterion is now defined as

$$\delta\beta^j = \frac{\Delta u - \mathbf{v}^T [(\bar{\mathbf{u}}^j)_n - (\bar{\mathbf{u}}^j)_m] - s^k \mathbf{v}^T [(\delta\bar{\mathbf{u}}_r^j)_n - (\delta\bar{\mathbf{u}}_r^j)_m]}{s^k \mathbf{v}^T [(\delta\bar{\mathbf{u}}_T)_n - (\delta\bar{\mathbf{u}}_T)_m]} \quad (8.65)$$

It is these general forms of the constraint equations that are implemented in MERLIN.

Chapter 9

TRANSIENT ANALYSIS; Direct Integration Schemes

9.1 Implicit

9.1.1 Newmark's β Method

Consider the Taylor series expansions of the displacement and velocity terms about the values at t

$$\bar{\mathbf{u}}^{t+\Delta t} = \bar{\mathbf{u}}^t + \frac{\partial \bar{\mathbf{u}}^t}{\partial t} \Delta t + \frac{\partial^2 \bar{\mathbf{u}}^t}{\partial t^2} \frac{\Delta t^2}{2!} + \frac{\partial^3 \bar{\mathbf{u}}^{t+\Delta t}}{\partial t^3} \frac{\Delta t^3}{3!} \quad (9.1)$$

$$\dot{\bar{\mathbf{u}}}^{t+\Delta t} = \dot{\bar{\mathbf{u}}}^t + \frac{\partial^2 \bar{\mathbf{u}}^t}{\partial t^2} \Delta t + \frac{\partial^3 \bar{\mathbf{u}}^{t+\Delta t}}{\partial t^3} \frac{\Delta t^2}{2!} \quad (9.2)$$

The above two equations represent the approximate displacement and velocity ($\bar{\mathbf{u}}^{t+\Delta t}$ and $\dot{\bar{\mathbf{u}}}^{t+\Delta t}$) by a truncated Taylor series. Looking at the remainder term (last term above),

$$\begin{aligned} \mathbf{R}_1 &= \frac{\partial^3 \bar{\mathbf{u}}^{t+\Delta t}}{\partial t^3} \frac{\Delta t^3}{3!} \simeq \frac{\frac{\partial^2 \bar{\mathbf{u}}^{t+\Delta t}}{\partial t^2} - \frac{\partial^2 \bar{\mathbf{u}}^t}{\partial t^2}}{\Delta t} \frac{\Delta t^3}{3!} \\ &\simeq (\ddot{\bar{\mathbf{u}}}^{t+\Delta t} - \ddot{\bar{\mathbf{u}}}^t) \frac{\Delta t^2}{3!} \\ &\simeq \beta(\ddot{\bar{\mathbf{u}}}^{t+\Delta t} - \ddot{\bar{\mathbf{u}}}^t) \Delta t^2 \end{aligned} \quad (9.3)$$

Similarly,

$$\begin{aligned} \mathbf{R}_2 &= \frac{\partial^3 \bar{\mathbf{u}}^{t+\Delta t}}{\partial t^3} \frac{\Delta t^2}{2!} \simeq \frac{\frac{\partial^2 \bar{\mathbf{u}}^{t+\Delta t}}{\partial t^2} - \frac{\partial^2 \bar{\mathbf{u}}^t}{\partial t^2}}{\Delta t} \frac{\Delta t^2}{2!} \\ &\simeq \gamma(\ddot{\bar{\mathbf{u}}}^{t+\Delta t} - \ddot{\bar{\mathbf{u}}}^t) \Delta t \end{aligned} \quad (9.4)$$

β and γ are parameters representing numerical approximations. Those parameters will account for \mathbf{R}_1 and \mathbf{R}_2 plus additional terms which were dropped from our Taylor series approximation. Substituting Equations 9.3 and 9.4 into Equations ?? and ?? respectively, we obtain,

$$\bar{\mathbf{u}}^{t+\Delta t} = \bar{\mathbf{u}}^t + \dot{\bar{\mathbf{u}}}^t \Delta t + \dot{\bar{\mathbf{u}}}^t \frac{\Delta t^2}{2} + \beta(\ddot{\bar{\mathbf{u}}}^{t+\Delta t} - \ddot{\bar{\mathbf{u}}}^t) \Delta t^2 \quad (9.5)$$

$$\dot{\bar{\mathbf{u}}}^{t+\Delta t} = \dot{\bar{\mathbf{u}}}^t + \ddot{\bar{\mathbf{u}}}^t \Delta t + \gamma(\ddot{\bar{\mathbf{u}}}^{t+\Delta t} - \ddot{\bar{\mathbf{u}}}^t) \Delta t \quad (9.6)$$

By rearrangement, we obtain,

$\dot{\bar{\mathbf{u}}}^{t+\Delta t} = \dot{\bar{\mathbf{u}}}^t + [(1-\gamma)\ddot{\bar{\mathbf{u}}}^t + \gamma\ddot{\bar{\mathbf{u}}}^{t+\Delta t}] \Delta t$	$\bar{\mathbf{u}}^{t+\Delta t} = \bar{\mathbf{u}}^t + \dot{\bar{\mathbf{u}}}^t \Delta t + [(1/2 - \beta)\ddot{\bar{\mathbf{u}}}^t + \beta\ddot{\bar{\mathbf{u}}}^{t+\Delta t}] \Delta t^2$
$\mathbf{f}_e^{t+\Delta t} = \mathbf{M}\ddot{\bar{\mathbf{u}}}^{t+\Delta t} + \mathbf{C}\dot{\bar{\mathbf{u}}}^{t+\Delta t} + \mathbf{K}\bar{\mathbf{u}}^{t+\Delta t}$	

(9.7)

Where the last equation is the equation of equilibrium, Eq. ?? expressed at time $t + \Delta t$. It can be shown that Newmark's forward difference assumes constant average acceleration over the time step. β and γ are parameters that can be determined to obtain integration accuracy and stability.

If $\gamma = 1/2$ and $\beta = 1/6$, corresponds to a linear acceleration, and $\gamma = 1/2$ and $\beta = 1/4$, correspond to a constant acceleration during the time increment. The scheme is explicit when β is 0. When γ is 1/2, this explicit form has the same numerical properties as the central difference method.

It can be shown that the Newmark method is

1. unconditionally stable if

$$\gamma \geq \frac{1}{2} \quad (9.8)$$

$$\beta \geq \frac{\gamma}{2} \quad (9.9)$$

2. conditionally stable if

$$\gamma \geq \frac{1}{2} \quad (9.10)$$

$$\beta < \frac{\gamma}{2} \quad (9.11)$$

with the following stability limit:

$$\omega\Delta t_{crit} = \frac{\xi(\gamma - 1/2) + [\gamma/2 - \beta + \xi^2(\gamma - 1/2)^2]^{1/2}}{(\gamma/2 - \beta)} \quad (9.12)$$

where ξ is the damping parameter.

Or, a constant acceleration is always stable, however for a linear acceleration to be stable

$$\frac{\Delta t}{T_n} \leq \frac{1}{\pi\sqrt{2}} \frac{1}{\sqrt{\gamma - 2\beta}} \quad (9.13-a)$$

$$\leq 0.551 \quad (9.13-b)$$

Solving from Eq. 9.7 for $\ddot{\bar{\mathbf{u}}}^{t+\Delta t}$ in terms of $\bar{\mathbf{u}}^{t+\Delta t}$ and then substituting for $\ddot{\bar{\mathbf{u}}}^{t+\Delta t}$ into Eq. 9.7, we obtain equations for $\dot{\bar{\mathbf{u}}}^{t+\Delta t}$ and $\ddot{\bar{\mathbf{u}}}^{t+\Delta t}$ each in terms of the unknown displacements $\bar{\mathbf{u}}^{t+\Delta t}$ only. These two equations for $\dot{\bar{\mathbf{u}}}^{t+\Delta t}$ and $\ddot{\bar{\mathbf{u}}}^{t+\Delta t}$ are then substituted in Eq. 9.7 to solve for $\bar{\mathbf{u}}^{t+\Delta t}$, after which, using Eq. 9.7 and 9.7, $\dot{\bar{\mathbf{u}}}^{t+\Delta t}$, and $\ddot{\bar{\mathbf{u}}}^{t+\Delta t}$ can be determined. This leads to the following algorithm:

1. Form the stiffness matrix \mathbf{K} , mass matrix \mathbf{M} , and damping matrix \mathbf{C} .
2. Initialise $\bar{\mathbf{u}}^0, \dot{\bar{\mathbf{u}}}^0, \ddot{\bar{\mathbf{u}}}^0$ at time $t = 0$
3. Select the time step Δt and parameters β and γ .
4. Determine the constants $a_0 = \frac{1}{\beta\Delta t^2}, a_1 = \frac{\gamma}{\beta\Delta t}, a_2 = \frac{1}{\beta\Delta t}, a_3 = \frac{1}{2\beta} - 1, a_4 = \frac{\gamma}{\beta} - 1, a_5 = \frac{\Delta t}{2} \left(\frac{\gamma}{\beta} - 2 \right), a_6 = \Delta t(1 - \gamma), a_7 = \gamma\Delta t$.
5. Form the effective stiffness matrix $\hat{\mathbf{K}}$: $\hat{\mathbf{K}} = \mathbf{K} + a_0\mathbf{M} + a_1\mathbf{C}$
6. Triangularize $\hat{\mathbf{K}} = \mathbf{LDL}^T$
7. For each time step:
 - a) Determine the effective load at time $t + \Delta t$

$$\hat{\mathbf{f}}_e^{t+\Delta t} = \mathbf{f}_e^{t+\Delta t} + \mathbf{M}(a_0\bar{\mathbf{u}}^t + a_2\dot{\bar{\mathbf{u}}}^t + a_3\ddot{\bar{\mathbf{u}}}^t) + \mathbf{C}(a_1\bar{\mathbf{u}}^t + a_4\dot{\bar{\mathbf{u}}}^t + a_5\ddot{\bar{\mathbf{u}}}^t) \quad (9.14)$$

- b) Solve for the displacement at time $t + \Delta t$

$$\mathbf{LDL}^T \bar{\mathbf{u}}^{t+\Delta t} = \hat{\mathbf{f}}_e^{t+\Delta t} \quad (9.15)$$

- c) Compute the accelerations and velocities

$$\ddot{\bar{\mathbf{u}}}^{t+\Delta t} = a_0(\bar{\mathbf{u}}^{t+\Delta t} - \bar{\mathbf{u}}^t) - a_2\dot{\bar{\mathbf{u}}}^t - a_3\ddot{\bar{\mathbf{u}}}^t \quad (9.16-a)$$

$$\dot{\bar{\mathbf{u}}}^{t+\Delta t} = \dot{\bar{\mathbf{u}}}^t + a_6\ddot{\bar{\mathbf{u}}}^t + a_7\ddot{\bar{\mathbf{u}}}^{t+\Delta t} \quad (9.16-b)$$

- d) If necessary solve for the stresses

$$\boldsymbol{\sigma}^{t+\Delta t} = f(\bar{\mathbf{u}}^{t+\Delta t}) \quad (9.17)$$

- e) Increase time step $t = t + \Delta t$

9.1.2 Hughes α Method

A major drawback of the Newmark β method is the tendency for high frequency noise to persist in the solution. On the other hand, when linear damping or artificial viscosity is added via the parameter γ , the accuracy is markedly degraded. The α method, (Hilber, Hughes and Taylor 1977) improves numerical dissipation for high frequency without degrading the accuracy as much, (Belytschko, Liu and Moran 2000).

Hughes α method (Hilber et al. 1977) is an implicit method in which the equation of motion (ignoring damping for now) is written at time $t + \Delta t$ (forward difference):

$$\mathbf{M}\ddot{\mathbf{u}}^{t+\Delta t} + \mathbf{Ku}^{t+\Delta t} = \mathbf{f}_e^{t+\Delta t} \quad (9.18)$$

Seeking an approximate solution of this equation by one-step difference, we write

$$\mathbf{M}\ddot{\mathbf{u}}^{t+\Delta t} + (1 + \alpha)\mathbf{Ku}^{t+\Delta t} - \alpha\mathbf{Ku}^t = \mathbf{f}_e^{t+\Delta t} \quad (9.19)$$

with

$$\mathbf{u}^{t+\Delta t} = \mathbf{u}^t + \Delta t\dot{\mathbf{u}}^t + \Delta t^2 \left[\left(\frac{1}{2} - \beta \right) \ddot{\mathbf{u}}^t + \beta \ddot{\mathbf{u}}^{t+\Delta t} \right] \quad (9.20-a)$$

$$\dot{\mathbf{u}}^{t+\Delta t} = \dot{\mathbf{u}}^t + \Delta t \left[(1 - \gamma) \ddot{\mathbf{u}}^t + \gamma \ddot{\mathbf{u}}^{t+\Delta t} \right] \quad (9.20-b)$$

We note that the α method introduces $\alpha\mathbf{K}(\mathbf{u}^{t+\Delta t} - \mathbf{u}^t)$ which is akin of stiffness proportional damping.

If the above equation is expanded, (Hughes 1983) effect of damping introduced, and possible material nonlinearity introduced, we obtain:

$$\mathbf{M}\ddot{\mathbf{u}}^{t+\Delta t} + (1 + \alpha)\mathbf{C}\dot{\mathbf{u}}^{t+\Delta t} - \alpha\mathbf{C}\dot{\mathbf{u}}^t + (1 + \alpha)\mathbf{f}_i^{t+\Delta t} - \alpha\mathbf{f}_i^t = (1 + \alpha)\mathbf{f}_e^{t+\Delta t} - \alpha\mathbf{f}_e^t \quad (9.21-a)$$

$$\mathbf{u}^{t+\Delta t} = \tilde{\mathbf{u}}^{t+\Delta t} + \beta\Delta t^2 \ddot{\mathbf{u}}^{t+\Delta t} \quad (9.21-b)$$

$$\dot{\mathbf{u}}^{t+\Delta t} = \tilde{\mathbf{u}}^{t+\Delta t} + \gamma\Delta t \ddot{\mathbf{u}}^{t+\Delta t} \quad (9.21-c)$$

where

$$\tilde{\mathbf{u}}^{t+\Delta t} = \mathbf{u}^t + \Delta t\dot{\mathbf{u}}^t + \Delta t^2 \left(\frac{1}{2} - \beta \right) \ddot{\mathbf{u}}^t \quad (9.22-a)$$

$$\tilde{\mathbf{u}}^{t+\Delta t} = \dot{\mathbf{u}}^t + \Delta t(1 - \gamma) \ddot{\mathbf{u}}^t \quad (9.22-b)$$

and \mathbf{f}_i , is the internal restoring force, and \mathbf{f}_e the external driving force.

Hence, the 3 equations, and the (possibly nonlinear) constitutive equation must all be simultaneously satisfied through an iterative method.

Assuming that we have obtained the response at time t , i.e. $\mathbf{u}^t, \dot{\mathbf{u}}^t$ and $\ddot{\mathbf{u}}^t$ which satisfy the equation of motion, we now seek to determine the solution at time $t + \Delta t$ by iteration (since $\mathbf{f}_i^{t+\Delta t} = \mathbf{K}(\mathbf{u}^{t+\Delta t})\mathbf{u}^{t+\Delta t}$). Given iteration step k , the trial solution is $\mathbf{u}_k^{t+\Delta t}, \dot{\mathbf{u}}_k^{t+\Delta t}$ and $\ddot{\mathbf{u}}_k^{t+\Delta t}$, it does not yet satisfy the equation of motion, Eq. 9.21-a. Hence, for this particular step we can write:

$$\mathbf{M}\ddot{\mathbf{u}}_k^{t+\Delta t} + (1 + \alpha)\mathbf{C}\dot{\mathbf{u}}_k^{t+\Delta t} - \alpha\mathbf{C}\dot{\mathbf{u}}^t + (1 + \alpha)\mathbf{f}_{i,k}^{t+\Delta t} - \alpha\mathbf{f}_i^t = (1 + \alpha)\mathbf{f}_e^{t+\Delta t} - \alpha\mathbf{f}_e^t - \mathbf{R}_k^{t+\Delta t} \quad (9.23)$$

where $\mathbf{f}_{i,k}^{t+\Delta t}$ is evaluated from the trial displacement $\mathbf{u}_k^{t+\Delta t}$ and $\mathbf{R}_k^{t+\Delta t}$ is the residual force error. If we subtract this equation from the exact equilibrium equation (Eq. 9.21-a), we have:

$$\mathbf{R}_k^{t+\Delta t} = \mathbf{M}\Delta\dot{\mathbf{u}}_k^{t+\Delta t} + (1 + \alpha)\mathbf{C}\Delta\dot{\mathbf{u}}^{t+\Delta t} + (1 + \alpha)\Delta\mathbf{f}_{i,k}^{t+\Delta t} \quad (9.24)$$

where

$$\Delta\ddot{\mathbf{u}}_k^{t+\Delta t} = \ddot{\mathbf{u}}^{t+\Delta t} - \ddot{\mathbf{u}}_k^{t+\Delta t} \quad (9.25-a)$$

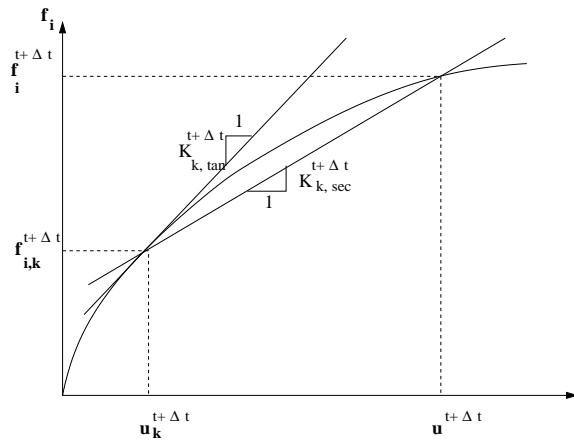
$$\Delta\dot{\mathbf{u}}_k^{t+\Delta t} = \dot{\mathbf{u}}^{t+\Delta t} - \dot{\mathbf{u}}_k^{t+\Delta t} \quad (9.25-b)$$

$$\Delta\mathbf{u}_k^{t+\Delta t} = \mathbf{u}^{t+\Delta t} - \mathbf{u}_k^{t+\Delta t} \quad (9.25-c)$$

$$\Delta\mathbf{f}_{i,k}^{t+\Delta t} = \mathbf{f}_i^{t+\Delta t} - \mathbf{f}_{i,k}^{t+\Delta t} \quad (9.25-d)$$

Hence, if we know the exact secant stiffness $\mathbf{K}_{k,sec}^{t+\Delta t}$ we can solve for $\mathbf{u}^{t+\Delta t}, \dot{\mathbf{u}}^{t+\Delta t}$ and $\ddot{\mathbf{u}}^{t+\Delta t}$ directly. However, in the context of a nonlinear model, the secant stiffness will have to be determined at each time step, Fig. 9.3. The solution process can now proceed as follows:from Fig. 9.3 we have

$$\Delta\mathbf{f}_{i,k}^{t+\Delta t} = \mathbf{K}_{k,sec}^{t+\Delta t} \Delta\mathbf{u}_k^{t+\Delta t} \quad (9.26)$$

Figure 9.1: Secant and Tangent Stiffnesses for α Method

however from Eq. 9.21-b and 9.21-c

$$\mathbf{u}_k^{t+\Delta t} = \tilde{\mathbf{u}}^{t+\Delta t} + \beta \Delta t^2 \ddot{\mathbf{u}}_k^{t+\Delta t} \quad (9.27-a)$$

$$\dot{\mathbf{u}}_k^{t+\Delta t} = \tilde{\mathbf{u}}^{t+\Delta t} + \gamma \Delta t \ddot{\mathbf{u}}_k^{t+\Delta t} \quad (9.27-b)$$

Subtracting the above equations from the corresponding exact ones

$$\Delta \ddot{\mathbf{u}}_k^{t+\Delta t} = \frac{\Delta \mathbf{u}_k^{t+\Delta t}}{\beta \Delta t^2} \quad (9.28-a)$$

$$\Delta \dot{\mathbf{u}}_k^{t+\Delta t} = \gamma \Delta t \Delta \ddot{\mathbf{u}}_k^{t+\Delta t} = \frac{\gamma}{\beta \Delta t} \Delta \mathbf{u}_k^{t+\Delta t} \quad (9.28-b)$$

Substituting the above equation into the residual equation (Eq. 9.24), we obtain

$$\mathbf{R}_k^{t+\Delta t} = \frac{\Delta \mathbf{u}_k^{t+\Delta t}}{\beta \Delta t^2} \mathbf{M} + (1 + \alpha) \frac{\gamma}{\beta \Delta t} \Delta \mathbf{u}_k^{t+\Delta t} \mathbf{C} + (1 + \alpha) \mathbf{K}_{k,sec}^{t+\Delta t} \Delta \mathbf{u}_k^{t+\Delta t} \quad (9.29)$$

which can be rearranged as

$$\tilde{\mathbf{K}}_k^{t+\Delta t} \Delta \mathbf{u}_k^{t+\Delta t} = \mathbf{R}_k^{t+\Delta t} \quad (9.30)$$

where the effective stiffness matrix is given by

$$\tilde{\mathbf{K}}_{k,sec}^{t+\Delta t} = \frac{1}{\beta \Delta t^2} \mathbf{M} + \frac{\gamma(1 + \alpha)}{\beta \Delta t} \mathbf{C} + (1 + \alpha) \mathbf{K}_{k,sec}^{t+\Delta t} \quad (9.31)$$

It should be noted that Eq. 9.30 is analogous to the simple static equilibrium equation, and we can directly evaluate $\Delta \mathbf{u}_k^{t+\Delta t}$ and obtain the exact solution. However, since we do not know $\mathbf{u}^{t+\Delta t}$ *a priori*, we can not evaluate the secant stiffness matrix $\mathbf{K}_{k,sec}^{t+\Delta t}$. This can be numerically estimated from Fig. 9.3 by the tangent stiffness matrix if need be. In this later case, we will need to iterate to converge to the exact solution.

Finally, we note that:

1. Alpha introduces a damping that grows with the ratio of time increment to the period of vibration of a node.
2. Negative values of α provide damping.
3. If $\alpha = 0$, we have no artificial damping (energy preserving) and is exactly the trapezoidal rule (Newmark's method if $\beta = 1/4$ and $\gamma = 1/2$).
4. Maximum value is $\alpha = -1/3$ which provides the maximum artificial damping. This results in a damping ratio of about 6% when the time increment is 40% of the period of oscillation of the mode being studied and smaller if the oscillation period increases.
5. This artificial damping is not very substantial for realistic time increment and low frequencies, but is non-negligible for high frequencies.
6. A default value of -0.05 is recommended.
- 7.

9.1.2.1 Algorithm

Adapted from (Hughes 1983).

Initialization:

1. Initialize at time $t = 0$: $\mathbf{u}^0, \dot{\mathbf{u}}^0, \mathbf{f}_i^0$ and $\ddot{\mathbf{u}}^0$.

2. Evaluate

$$\tilde{\mathbf{u}}^{t+\Delta t} = \mathbf{u}^t + \Delta t \dot{\mathbf{u}}^t + \Delta t^2 \left(\frac{1}{2} - \beta \right) \ddot{\mathbf{u}}^t \quad (9.32-a)$$

$$\tilde{\mathbf{u}}^{t+\Delta t} = \dot{\mathbf{u}}^t + \Delta t (1 - \gamma) \ddot{\mathbf{u}}^t \quad (9.32-b)$$

3. Set $k = 0$, and select a trial solution

$$\mathbf{u}_k^{t+\Delta t} = \tilde{\mathbf{u}}^{t+\Delta t} \quad (9.33-a)$$

$$\dot{\mathbf{u}}_k^{t+\Delta t} = \tilde{\mathbf{u}}^{t+\Delta t} \quad (9.33-b)$$

$$\mathbf{f}_{i,k}^{t+\Delta t} = \mathbf{f}_i(\mathbf{u}_k^{t+\Delta t}) \quad (9.33-c)$$

$$\ddot{\mathbf{u}}_k^{t+\Delta t} = 0 \quad (9.33-d)$$

Inner Loops

A Current increment t , set $k = 0$

1. Evaluate the tangent stiffness matrix and then the effective stiffness

$$\tilde{\mathbf{K}}_k^{t+\Delta t} = \frac{1}{\beta \Delta t^2} \mathbf{M} + \frac{\gamma(1+\alpha)}{\beta \Delta t} \mathbf{C} + (1+\alpha) \mathbf{K}_{k,tan}^{t+\Delta t} \quad (9.34)$$

2. Evaluate the residual error

$$\mathbf{R}_k^{t+\Delta t} = (1+\alpha) \mathbf{f}_e^{t+\Delta t} - \alpha \mathbf{f}_e^t - \mathbf{M} \ddot{\mathbf{u}}_k^{t+\Delta t} - (1+\alpha) \mathbf{C} \dot{\mathbf{u}}_k^{t+\Delta t} + \alpha \mathbf{C} \dot{\mathbf{u}}^t - (1+\alpha) \mathbf{f}_{i,k}^{t+\Delta t} + \alpha \mathbf{f}_i^t \quad (9.35)$$

3. Solve for $\Delta \mathbf{u}_k^{t+\Delta t}$

$$\tilde{\mathbf{K}}_k^{t+\Delta t} \Delta \mathbf{u}_k^{t+\Delta t} = \mathbf{R}_k^{t+\Delta t} \quad (9.36)$$

4. Evaluate the new trial displacements

$$\mathbf{u}_{k+1}^{t+\Delta t} = \mathbf{u}_k^{t+\Delta t} + \Delta \mathbf{u}_k^{t+\Delta t} \quad (9.37)$$

5. Solve for $\mathbf{f}_{i,k+1}^{t+\Delta t}$

6. Evaluate the new trial acceleration and velocity

$$\ddot{\mathbf{u}}_{k+1}^{t+\Delta t} = \frac{1}{\beta \Delta t^2} (\mathbf{u}_{k+1}^{t+\Delta t} - \tilde{\mathbf{u}}^{t+\Delta t}) \quad (9.38-a)$$

$$\dot{\mathbf{u}}_{k+1}^{t+\Delta t} = \tilde{\mathbf{u}}^{t+\Delta t} + \gamma \Delta t \ddot{\mathbf{u}}_{k+1}^{t+\Delta t} \quad (9.38-b)$$

7. If $|\Delta \mathbf{u}_k^{t+\Delta t}| > \epsilon$, set $k = k + 1$ and go to step 1.

B Update the displacement, velocity and accelerations

$$\mathbf{u}^{t+\Delta t} = \mathbf{u}_{k+1}^{t+\Delta t} \quad (9.39-a)$$

$$\dot{\mathbf{u}}^{t+\Delta t} = \dot{\mathbf{u}}_{k+1}^{t+\Delta t} \quad (9.39-b)$$

$$\ddot{\mathbf{u}}^{t+\Delta t} = \ddot{\mathbf{u}}_{k+1}^{t+\Delta t} \quad (9.39-c)$$

C Set $t = t + \Delta t$ and go to step A.

It can be shown that the method is unconditionally stable if

$$\alpha \in \left[-\frac{1}{3}, 0 \right]; \quad \gamma = \frac{(1-2\alpha)}{2}; \quad \beta = \frac{(1-\alpha)^2}{4} \quad (9.40)$$

9.2 Explicit

Adapted from Belytschko et al. (2000)

9.2.1 Preliminaries

Time step definitions, Fig. 9.2:

$$\Delta t^{n+1/2} = t^{n+1} - t^n \quad (9.41-a)$$

$$t^{n+1/2} = \frac{1}{2}(t^{n+1} + t^n) \quad (9.41-b)$$

$$\Delta t^n = t^{n+1/2} - t^{n-1/2} \quad (9.41-c)$$

The central difference formulae for velocity is

$$\dot{\mathbf{d}}^{n+1/2} \stackrel{\text{def}}{=} \mathbf{v}^{n+1/2} = \frac{\mathbf{d}^{n+1} - \mathbf{d}^n}{\Delta t^{n+1/2}} = \frac{1}{\Delta t^{n+1/2}}(\mathbf{d}^{n+1} - \mathbf{d}^n) \quad (9.42)$$

This difference formula can be converted into an integration formula by rearranging terms

$$\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t^{n+1/2} \mathbf{v}^{n+1/2} \quad (9.43)$$

Similarly, the acceleration and the corresponding integration formula are

$$\begin{aligned} \ddot{\mathbf{d}}^n &\stackrel{\text{def}}{=} \mathbf{a}^n \\ \mathbf{v}^{n+1/2} &= \frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{\Delta t^{n+1/2} - \Delta t^{n-1/2}} \\ &= \mathbf{v}^{n-1/2} + \Delta t^n \mathbf{a}^n \end{aligned} \quad (9.44)$$

Hence, velocities are defined at the midpoints of the time steps.

Substituting Eq. 10.2 (expressed at $t^{n+1/2}$ and $t^{n-1/2}$) into Eq. 10.4, the acceleration can be expressed directly in terms of the displacements

$$\ddot{\mathbf{d}}^n \stackrel{\text{def}}{=} \mathbf{a}^n = \frac{\Delta t^{n-1/2}(\mathbf{d}^{n+1} - \mathbf{d}^n) - \Delta t^{n+1/2}(\mathbf{d}^n - \mathbf{d}^{n-1})}{\Delta t^{n+1/2} \Delta t^n \Delta t^{n-1/2}} \quad (9.45)$$

For equal time steps, this reduces to

$$\ddot{\mathbf{d}}^n \stackrel{\text{def}}{=} \mathbf{a}^n = \frac{(\mathbf{d}^{n+1} - 2\mathbf{d}^n + \mathbf{d}^{n-1})}{(\Delta t^n)^2} \quad (9.46)$$

which is the well known central difference formula for the second derivative of a function.

We now consider the time integration of the motion equation at time step n

$$\mathbf{M}\ddot{\mathbf{a}}^n = \mathbf{f}^n = \mathbf{f}^{\text{ext}}(\mathbf{d}^n, t^n) - \mathbf{f}^{\text{int}}(\mathbf{d}^n, t^n) \quad (9.47)$$

subjected to the essential boundary condition

$$g_I(\mathbf{v}^n) = 0 \quad I = 1 \text{ to } n_c \quad \text{on } \Gamma_v \quad (9.48)$$

which is an ordinary differential equation of second order in time.

The internal forces are functions of the nodal displacements (and thus on time), the external forces are function both of time and displacement (uplift forces). and time.

Substituting Eq. 10.7 into 10.4 gives

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \Delta t^n \mathbf{M}^{-1} \mathbf{f}^n \quad (9.49)$$

Energy balance must be satisfied, as numerical instability in nonlinear problems may only manifests itself in a pernicious and subtle manner which will lead uncorrect results. It may also cause exponential growth which may cause localized premature failures.

Hence Energy must be computed as follows:

$$W_{int}^{n+1} = W_{int}^n + \frac{\Delta t^{n+1/2}}{2} \left(\mathbf{v}^{n+1/2} \right)^T (\mathbf{f}_{int}^n + \mathbf{f}_{int}^{n+1}) = W_{int}^n + \frac{1}{2} \Delta t \mathbf{d}^T (\mathbf{f}_{int}^n + \mathbf{f}_{int}^{n+1}) \quad (9.50-a)$$

$$W_{ext}^{n+1} = W_{ext}^n + \frac{\Delta t^{n+1/2}}{2} \left(\mathbf{v}^{n+1/2} \right)^T (\mathbf{f}_{ext}^n + \mathbf{f}_{ext}^{n+1}) = W_{ext}^n + \frac{1}{2} \Delta t \mathbf{d}^T (\mathbf{f}_{ext}^n + \mathbf{f}_{ext}^{n+1}) \quad (9.50-b)$$

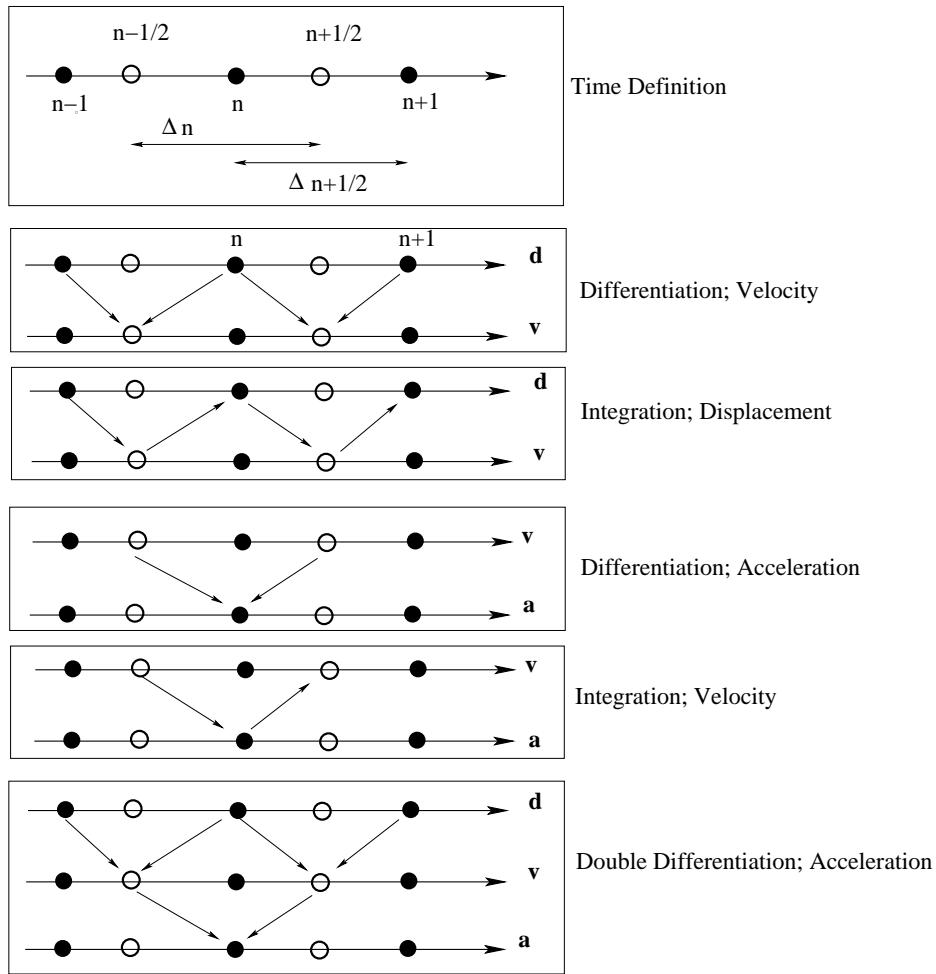


Figure 9.2: Central Difference Scheme (Explicit Method); Basic Definitions

The kinetic energy is given by

$$W_{kin}^n = \frac{1}{2}(\mathbf{v}^n)^T \mathbf{M} \mathbf{v}^n \quad (9.51)$$

where $\Delta \mathbf{d} = \mathbf{d}^{n+1} - \mathbf{d}^n$

Energy conservation requires that

$$|W_{kin} + W_{int} - W_{ext}| \leq \varepsilon \max(W_{kin}, W_{int}, W_{ext}) \quad (9.52)$$

where ε is a tolerance of the order of 10^{-2}

9.2.2 Algorithm

1. **Initialization** Loop over elements and determine
 - a) External load vectors \mathbf{f}^{ext} (includes gravity, hydrostatic, uplift, etc...)
 - b) $\mathbf{a}^0, \mathbf{v}^0, \boldsymbol{\sigma}^0$ and other state variables.
 - c) Compute lumped mass matrix \mathbf{M} (vector stored)
 - d) Compute damping matrix $\mathbf{C} = \alpha \mathbf{M}$ (vector stored)
 - e) Set $\mathbf{f}^{ext,0} = \mathbf{M} \mathbf{a}^0$
 - f) If \mathbf{a}^0 is prescribed, set $\mathbf{v}^0 = \mathbf{C}^{-1} (\mathbf{f}^{ext,0} - \mathbf{f}^{int,0} - \mathbf{M} \mathbf{a}^0)$
 - g) $\mathbf{d}^0 = \mathbf{0}; n = 0, t = 0$.
 2. **Get Forces**
 3. Compute accelerations $\mathbf{a}^n = \mathbf{M}^{-1} (\mathbf{f}^{ext,n} - \mathbf{f}^{int,n} - \mathbf{C}^{damp} \mathbf{v}^{n-1/2})$
 4. Update time $t^{n+1} = t^n + \Delta t^{n+1/2}$, and $t^{n+1/2} = \frac{1}{2}(t^n + t^{n+1})$.
 5. First partial update of the velocities $\mathbf{v}^{n+1/2} = \mathbf{v}^n + (t^{n+1/2} - t^n) \mathbf{a}^n$
 6. Enforce velocity boundary conditions on Γ_v $\mathbf{v}^{n+1/2} = \frac{t^{n+1} - t^{n-1/2}}{t^{n+1} - t^n} \tilde{\mathbf{v}}^n + \frac{t^{n+1/2} - t^n}{t^{n+1} - t^n} \tilde{\mathbf{v}}^{n+1}$
 7. Update nodal displacements $\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t^{n+1/2} \mathbf{v}^{n+1/2}$
 8. Enforce displacement boundary condition over Γ_d : $\mathbf{d}^{n+1} = \tilde{\mathbf{d}}^{n+1}$
 9. **Get Force**
 10. Compute $\mathbf{a}^{n+1} = \mathbf{M}^{-1} (\mathbf{f}^{ext,n+1} - \mathbf{f}^{int,n+1} - \mathbf{C}^{damp} \mathbf{v}^{n+1/2})$
 11. Enforce acceleration boundary conditions over Γ_a : $\mathbf{a}^{n+1} = \tilde{\mathbf{a}}^{n+1}$
 12. Second partial update of nodal velocities $\mathbf{v}^{n+1} = \mathbf{v}^{n+1/2} + (t^{n+1} - t^{n+1/2}) \mathbf{a}^{n+1}$
 13. Enforce velocity boundary conditions on Γ_v $\tilde{\mathbf{v}}^{n+1} = \frac{t^{n+1/2} - t^{n-1/2}}{t^{n+1} - t^n} \tilde{\mathbf{v}}^n + \frac{t^{n+1/2} - t^n}{t^{n+1} - t^n} \tilde{\mathbf{v}}^{n+1}$
 14. Check energy balance every k time steps where $k \simeq ??$ $W_{int}^{n+1} = W_{int}^n + \frac{1}{2} \Delta \mathbf{d}^T (\mathbf{f}_{int}^n + \mathbf{f}_{int}^{n+1})$, $W_{ext}^{n+1} = W_{ext}^n + \frac{1}{2} \Delta \mathbf{d}^T (\mathbf{f}_{ext}^n + \mathbf{f}_{ext}^{n+1})$, $W_{kin}^{n+1} = \frac{1}{2} (\mathbf{v}^{n+1})^T \mathbf{M} \mathbf{v}^{n+1}$, and $|W_{kin} + W_{int} - W_{ext}| \leq \varepsilon \max(W_{kin}, W_{int}, W_{ext})$ where $\varepsilon \simeq 10^{-2}$
 15. Update counter $n \leftarrow n + 1$
 16. Output or go to 4
- Algorithm Get Forces**
1. Initialize $\mathbf{f}^n = \mathbf{0}$;
 2. Set $\Delta t_{crit} = \infty$

3. Compute global external nodal forces \mathbf{f}_{ext}^n (includes uplift)
4. Loop over elements (e).
 - a) Gather element nodal displacements and velocities from global array
 - b) Set $\mathbf{f}_{(e)}^{int,n} = \mathbf{0}$
 - c) Loop over Gauss points ξ_Q
 - i. If $n = 0$ go to 3(c)ii
 - ii. Compute stresses $\sigma^n(\xi_Q)$ through the constitutive equation.
 - iii. $\mathbf{f}_{(e)}^{int,n} \leftarrow \mathbf{f}_{(e)}^{int,n} + \mathbf{B}^T \bar{\mathbf{w}}_Q J|_Q$
 - d) Compute Rayleigh damping stiffness proportional terms $\mathbf{f}_{(e,dc)}^{int,n} = -\beta \frac{\Delta \mathbf{f}_{(e)}^{int,n}}{\Delta t}$
 - e) Update internal forces $\mathbf{f}_{(e)}^{int,n} \leftarrow \mathbf{f}_{(e)}^{int,n} + \mathbf{f}_{(e,dc)}^{int,n}$
 - f) Compute external nodal forces $\mathbf{f}_{(e)}^{ext,n}$
 - g) $\mathbf{f}_{(e)}^n = \mathbf{f}_{(e)}^{ext,n} - \mathbf{f}_{(e)}^{int,n}$
 - h) Compute $\Delta t_{crit}^{(e)} = \alpha \min \frac{L_{(e)}}{c} = \alpha \min \frac{L_{(e)}}{\sqrt{\frac{E}{\rho}}}$
 - i) If $\Delta t_{crit}^{(e)} < \Delta t_{crit}$, then $\Delta t_{crit} = \Delta t_{crit}^{(e)}$
 - j) Scatter $\mathbf{f}_{(e)}^n$ back to global \mathbf{f}^n
5. $\Delta t = \alpha \Delta t_{crit}$

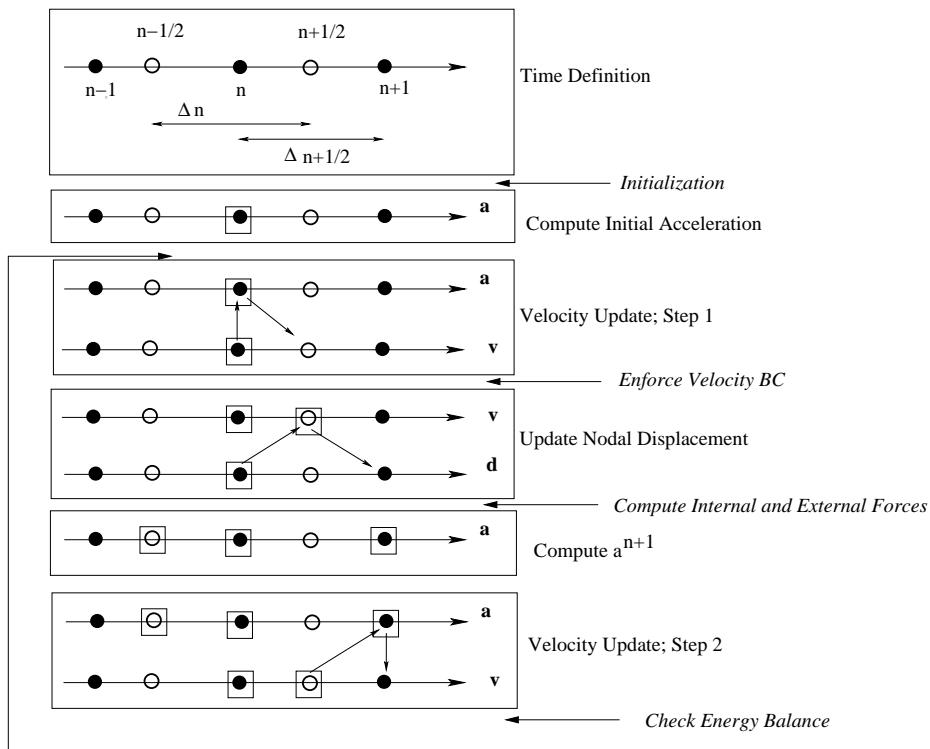


Figure 9.3: Algorithm for Central Difference Scheme (Explicit Method)

9.2.3 Dynamic Relaxation

Adapted from (Pandolfi 2003)

9.3 Rayleigh Damping

Rayleigh damping, is the most widely used (but not only) model for damping. It assumes that

$$\mathbf{C} = a\mathbf{M} + b\mathbf{K} \quad (9.53)$$

where the coefficients a and b are calculated based upon two circular frequencies (ω_1 and ω_2 , radians/sec.) to be damped at ξ_1 and ξ_2 respectively.

We recall that the damping ratio for a single degree of freedom (SDF) for mode n is given by

$$\zeta_n = \frac{C_n}{2M_n\omega_n} \quad (9.54)$$

Thus for mass proportional damping of multi degree of freedom (MDF) system, with $\mathbf{C}_n = a\mathbf{M}_n$, this would lead to

$$\zeta_n = \frac{a}{2} \frac{1}{\omega_n} \quad (9.55)$$

The damping ratio is thus inversely proportional to the natural frequency and a can be selected to obtain a specified damping ratio in any one mode i or

$$a = 2\zeta_i\omega_i \quad (9.56)$$

Similarly, and recalling that $\mathbf{K}\phi_n = \omega_n^2\mathbf{M}\phi_n$, a stiffness proportional damping $\mathbf{C}_n = b\mathbf{K}_n$ combined with Eq. 9.54 will lead to

$$\zeta_n = \frac{b}{2} \omega_n \quad (9.57)$$

In this case the damping ratio is proportional to the natural frequency and b can be selected to obtain a specified damping ratio in any one mode j or

$$b = \frac{2\zeta_j}{\omega_j} \quad (9.58)$$

Combining Eq. 9.56 and 9.58 leads to the following linear equations

$$\frac{1}{2} \begin{bmatrix} \frac{1}{\omega_i} & \omega_i \\ \frac{1}{\omega_j} & \omega_j \end{bmatrix} \begin{Bmatrix} a \\ b \end{Bmatrix} = \begin{Bmatrix} \zeta_1 \\ \zeta_2 \end{Bmatrix} \quad (9.59)$$

If one assumes the same damping ratio ζ for both modes (reasonable practical assumption), then

$$\begin{cases} a = \zeta \frac{2\omega_i\omega_j}{\omega_i + \omega_j} \\ b = \zeta \frac{2}{\omega_i + \omega_j} \end{cases} \quad (9.60)$$

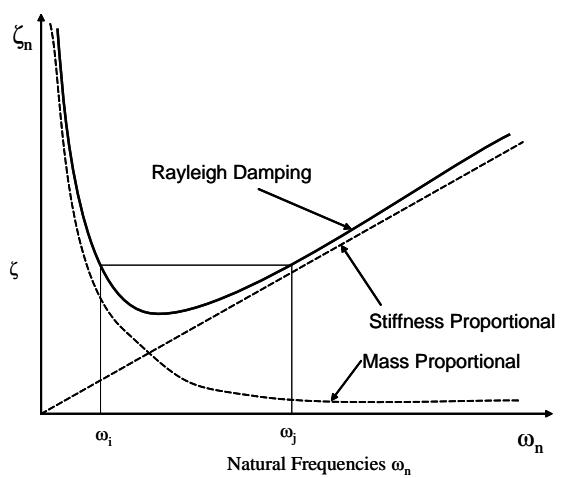


Figure 9.4: Rayleigh Damping

Chapter 10

EXPLICIT PARALLEL

10.1 Introduction

Parallel computing is nowadays considered a very efficient tool to overcome bottlenecks of traditional serial computing. These bottlenecks relate to both lack of resources (memory, disk space, etc.) and long computational times. Typical parallel application decreases the demands on memory and other resources by spreading the task over several mutually interconnected computers and speeds up the response of the application by distributing the computation to individual processors. Note however that parallel computing is worth also for applications that require almost no resources but consume an excessive amount of time and for applications that cannot be performed on a single (even well equipped) computer regardless of the computational time. It is important to realize that from engineering point of view the scalability of the algorithm is not the only criterion to judge efficiency of parallel application. In many cases, the ability to analyze extremely large problems not solvable on single machine is of primary interest.

The solution of complex sophisticated problems to model various phenomena with sufficiently high accuracy and in reasonable time makes the parallel processing attractive for a large family of applications, including structural analysis. However it is important to realize that most of traditional algorithms are inherently not suitable for parallelization because of their development for sequential processing. The most natural way for parallelization is the decomposition of the problem being solved in time or space. The individual domains are then mapped on individual processors and are solved separately ensuring the proper response of the whole system by appropriate communication between the domains. An efficient parallel algorithm requires a balance of the work (performed on individual domains) between the processors while maintaining the interprocessor communication (typical bottleneck of parallel computation) at a minimum.

Since the last decade the parallel computation has become quite feasible due to the following three aspects. Firstly, a lot of new algorithms, suitable for parallel processing, have been developed (including efficient algorithms for domain decomposition). Secondly, the parallel computation ceased to be limited to parallel supercomputers (equipped with high technology for even higher price) but can be performed on ordinary computers interconnected by network into computer cluster. Such a parallel cluster can even outperform the supercomputers (as IBM SP2, SGI Origin etc) while keeping the investment and maintenance costs substantially lower ! And thirdly, several message passing libraries (typically MPI, PVM), portable to various hardware and operating system platforms, have been developed, which allows to port the parallel applications almost to any platform (including multiplatform parallel computing cluster).

10.1.1 Parallel Computational Models

There are several parallel computational models available depending on whether the memory is physically shared or distributed, or whether it is virtually shared if physically distributed. In this view, one possible classification¹ of parallel computational models is

- data parallelism

In this model, the parallelism comes entirely from implicit data independence, the program itself looks very much like a sequential program. The early application of this approach is the vectorization of a code using a vector machine. More recent applications perform the partitioning of data by the compiler (e.g. HPF – High Performance Fortran). This model is typical representative of SIMD (Single Instruction Multiple Data) applications. The efficiency of this model is strongly dependent on the problem without the possibility to be too influenced by the programmer.

- shared memory

On shared memory architecture, each processor has access to all of a single shared address space. However, the parallelism in this case must be explicitly specified by the programmer. Coordination of access to memory by multiple processes is done by some form of locking (that might be hidden in high level languages). Shared memory model is also sometimes known as SMP (Symmetric Multiprocessing). The thread technology, based on fast switching between individual threads of a single process with a separate address space, can be also classified into this model. Note that shared memory parallel computational model can accommodate both paradigms SIMD and MIMD (Multiple Instruction Multiple Data).

The shared memory machines with high number of processor are quite difficult (and expensive) to build. Since

¹Note that there are other classifications based on other criteria available.

the capacity of the communication bus is shared between all the processors, the memory access time increases with the number of processors. To make the memory access more effective, each processor has fast cache memory (very expensive, mostly more than processor itself) which, if properly used, significantly reduces the memory access time.

- message passing

In this model, the memory is physically distributed and each processor has access only to its local memory. The off-processor data are accessed via communication with other processors using sending and receiving messages over a network. It is a defining feature of the message passing that data transfer from the local memory of one process to the local memory of another one requires operations to be performed by both processes. This model enables besides SIMD and MIMD also to process MPMD (Multiple Program Multiple Data) applications. Note, that most of the MPMD applications can be converted to MIMD applications (that can be much easily debugged) by simply branching the code using *if-then-else* construction. The way in which the individual processors are interconnected defines the topology of the model. For some common topologies as grid, torus, or hypercube, there might be a significant support in the particular message passing.

The communication network can be a special high-speed network (e.g. high performance switch in IBM SP2 machine) or general purpose network as fast Ethernet or even in near future the Internet (so called P2P – pier to pier architecture). The performance of the network communication is given by the latency and the bandwidth. The latency is the time necessary to start an interaction between two processors and the bandwidth is the number of bytes that can be transferred via the network within one second. Since a distributed memory computer has no shared resources like a bus, the number of processors is virtually unlimited.

The advantage of message passing model consists in the fact that it gives to the programmer full control over the parallelism. It is well suited to the adaptive, self-scheduling algorithms and to programs that are to be made tolerant of the imbalance in process speed on shared networks. The message passing paradigm is attractive because of its wide portability and scalability. It is easily compatible with both distributed memory computers and shared memory multiprocessors, and their combination. The most common message passing libraries are PVM (Parallel Virtual Machine) and MPI (Message Passing Interface), but there are many others.

- combined model

This model is based on combination of shared memory and message passing models. A typical example is a cluster (distributed memory) of shared memory multiprocessor workstations. Because of the memory distribution, the message passing concept is typically used. However, the overall performance can be improved by the fact that the message passing can take advantage of hardware services for accessing shared memory without explicitly communicating the messages.

- virtual shared memory model

In this architecture, the physical distribution of the memory (either on memory distributed or combined model) is hidden to the user and all the memory is accessible as a (virtually) shared memory. It is the responsibility of the system to make the access to the remote memory transparent as it would be local.

10.1.2 Solution Strategies Based on Message Passing

As described above, message passing is parallel paradigm applicable especially on memory distributed computing platforms. In this view, the problem to be solved needs to be distributed accordingly. Then each processor is working with its local data and the requests of remote data are resolved using the message passing. Most of the distribution strategies are based on the decomposition of spatial domain over which the solution is searched for. To achieve high level of parallelism, the domain decomposition should satisfy the following conditions:

- the individual subdomains should be approximately of the same complexity
- the interface between the subdomains should be minimized.

While the first condition is related to the load balancing of the parallel analysis in order to minimize the idle time (during which the processor is waiting for remote data, that are not yet available), the second condition takes into account that the data transfer between the processors is usually on much lower performance level than the CPU performance of the processor itself.

The load balancing can be either static, where the actual domain decomposition is persistent through the whole solution process (taking into account the heterogeneity of the computing cluster and the heterogeneity of the computational domain), or dynamic, in which the domains are changing during the solution to maintain the load balance during the whole analysis. The load balance can be disturbed for example by the analysis itself (e.g. change of the constitutive law from linear to nonlinear in part of the computational domain) or by the computing environment (if the analysis is run on non-dedicated computers). Dynamic load balancing strategy, involving large data migration

and complex changes in the data management, is however very difficult to implement. This is why the dynamic load balancing is sometimes replaced by pseudo-dynamic load balancing concept. In this strategy, the static domain partitioning into much higher number of partitions than number of available processors is performed. The individual domains are then assigned to the processors whenever the processor has completed its job.

10.1.2.1 Domain Partitioning

The domain partitioning is depending on the dimension and regularity of the domain and on the way in which the discretization of the spatial domain is accomplished. Different strategies may be used for 1D and multidimensional problems, for regular and irregular or even noncontinuous domains, and for domains discretized by finite differences and finite elements (volumes) or or discretizations based on element free concept. In the following it is assumed that the computational domain is represented by a 2D or 3D mesh of finite elements.

A mesh decomposer should distribute the mesh across the individual processors so that the computational load is evenly balanced and the amount of interprocessor communication is minimized. However, the numerical experience has shown that several other issues, as the subdomain shape and connectivity, in addition to load balancing and communication costs, need to be addressed. In recent years, a considerable attention has been focused on developing suitable techniques to solve the mesh partitioning problem and several powerful methods have been devised. The greedy algorithm is based on a successive expansion of a subdomain, initially formed by one appropriately chosen element, until it comprises a sufficiently large number of elements. The expansion is usually driven by neighbourhood search schemes using the depth-first or breadth-first search. The basic disadvantage of this very fast technique resides in the fact that the final partitioning is often very far from the “optimal” one. However, the speed makes this technique very suitable for an initial decomposition subjected to further optimization based, for example, on the relative gain concept or simulated annealing. The recursive bisection methods utilize the spatial distribution of a mesh. While the coordinate recursive bisection (Cartesian, polar, or spherical) exploits only the dimensional properties of the mesh with respect to a given coordinate system, the inertial recursive bisection accounts for principal inertial properties of the mesh which are invariant with respect to the coordinate system. The spectral recursive bisection is based on the finding that the second largest eigenvalue of the Laplacian matrix of an undirected graph associated with a mesh provides a good measure of the connectivity of the mesh and that the components of the corresponding eigenvector can be conveniently used for the mesh bisection. Although this approach provides decomposition of a high quality, computationally complexity makes its use problematic when large meshes are under consideration. This deficiency was partially eliminated by a multilevel implementation of this technique.

Note that the obtained domain partitioning must be sometimes further processed to allow for overlapping of individual subdomains for example when dealing with nonlocal material models based on nonlocal averaging, or when using solution methods based on domain overlapping (e.g. Schwartz additive methods).

10.1.3 Parallelization of the Solution Method

The complexity related to the parallelization of a particular solution method is strongly depending on the properties of a global system of linear algebraic equations that has to be solved. If the system has a canonical form (e.g. in explicit integration methods using lumped mass matrix and Rayleigh form of damping), the actual parallelization is very straightforward. Since each of the equations can be solved independently and locally, on the partition owning the corresponding mesh node, the communication via the message passing is more or less needed only for assembling contributions to the particular equation if these contributions originate on different partitions. This is usually the case if the equation is related to a mesh node being shared by several partitions. Then the exchange of values corresponding to contributions to the (lumped) mass matrix and vectors of internal and external forces is needed. To ensure synchronization of the integration of the equilibrium equations in time, all partitions have to use the same time increment. Therefore any change in time step must be communicated between all subdomain in order to agree on the common value still ensuring stability of the method.

If the global system of linear algebraic equations is coupled (implicit methods, or explicit methods with full mass matrix), the most crucial step is the parallelization of the linear equation solver. This depends on many aspects, for example on the solver itself (iterative, direct, sparse direct), on the matrix storage format (skyline, compressed row, symmetric compressed row), etc. Some of the methods are based on the solution of a reduced system that is assembled from partially eliminated matrices corresponding to individual subdomains. This elimination can be typically performed in parallel without any communication. The final reduced system is then solved either on one dedicated processor using a sequential solver or in parallel based on message passing. Typical representatives of this approach is the Schur complement method and FETI (Finite Element Tearing and Interconnection). Note that the physical meaning of the unknowns in the reduced system may be different from the physical meaning of unknowns in the original system of equations. For example, in dual FETI method, the unknowns of the final reduced problem are Lagrange multipliers representing the forces used to ensure compatibility between adjacent subdomains.

There are several packages available for parallel solution of system of equations, for example PETSc, Spooles, SuperLU, BlockSolve and others.

10.2 MPI – Message Passing Interface

MPI is a portable message passing standard that facilitates the development of parallel applications and libraries. MPI itself is a library, not a language. It is used to specify the communication between a set of processes forming a concurrent program that is efficient and highly functional and that is portable to different computing platforms including heterogeneous networks of computers that have different lengths and formats for various fundamental datatypes.

The current MPI standards includes (except others) the following:

- Point to point communications: messages between pairs of processes.
- Collective communications: messages and synchronization operations that involve entire groups of processes.
- Process groups: manipulation of groups of processes.
- Communicators: a mechanism for providing separate communication scope for modules or libraries.
- Process topologies: functions that allow the convenient manipulation of processes forming a particular topology.
- Datatypes: a mechanism for handling existing and user defined data types.
- Bindings for C and Fortran 77: specifications of names, calling sequences and results of subroutines calls from Fortran 77 and functions called from C programs.

The following aspects (except others) are not covered by the current standard:

- shared memory operations
- thread support
- process and task management
- input and output functions
- debugging support

MPI is a rich library offering over 100 functions. But many parallel programs can be written using just 6 basic functions:

- MPI_INIT – initialize MPI
- MPI_COMM_SIZE – find how many processes there are
- MPI_COMM_RANK – find out which process I am
- MPI_SEND – Send a message
- MPI_RECV – Receive a message
- MPI_FINALIZE – Terminate MPI

The other functions all add flexibility, robustness, efficiency, modularity, and convenience.

10.2.1 Point to Point Communication

MPI provides send and receive functions that allow the communication of typed data with an associated tag. Typing of the message contents is necessary for heterogeneous support – the type information is needed so that correct data representation conversions can be performed as data is sent from one architecture to another. The tag allows selectivity of messages at the receiving end; one can receive on a particular tag, or one can use wildcard tag allowing reception of messages with any tag. Both the send and receive message contains the address of the buffer from/to which the data is communicated. The amount of data is represented by number of elements of given type (not number of bytes). While the send message specifies the destination process, the receive message specifies the source process, that can be also used for message selectivity. The processes must belong to the same communicator, that is also parameter of both calls. The receive message has one more argument, which is the status of the message, from which the tag and source of the received message can be obtained (if wildcards has been used in the call to it).

MPI provides a whole set of send and receive functions for different communications modes. The basic mode is the standard blocking mode. In this mode, the send call does not return until the message data are safely stored and user can reuse the send buffer. It is the choice of MPI whether it will copy the message to a local buffer (which causes some overhead) or whether it will wait for the matching receive call (which causes idle time). Similarly the receive function blocks until the receive buffer actually contains the contents of the message. There are 3 additional communication blocking modes. In the buffered mode (prefix B), MPI always copies the send data to its or user supplied buffer, and immediately returns from the send call. In the synchronous mode (prefix S), the send call does

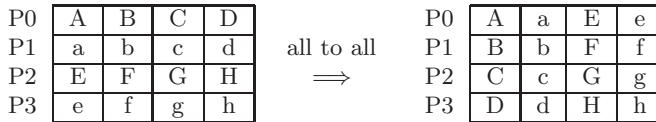
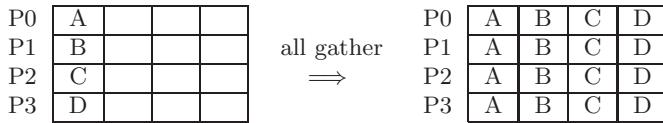
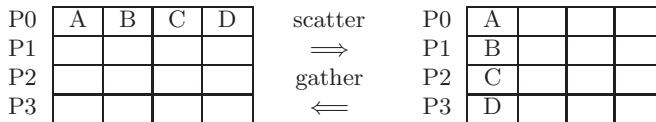
not return until the matching receive call was posted. And finally, in the ready mode (prefix R) the send can be called only if the matching receive has already been called, otherwise error occurs. Note that the ready mode results in improved performance and can be also used for synchronization of the code.

For all 4 above communication modes there exist also their non-blocking variant (prefix I). Since the send call in non-blocking communication returns immediately this type of communications is appropriate for overlapping of computation and communication. A separate calls to MPI wait and test functions are needed to complete the send and receive calls, in other words to be sure that the buffers for message data can be safely reused. Note that the non-blocking send can be matched with blocking receive and vice versa.

10.2.2 Collective Communication

Collective communications transmit data among all the processes specified by a communicator object. The only exception is the barrier function, that serves to synchronize processes without passing data. Generally these functions can be classified as

- collective synchronization – barrier
- collective data movement – broadcast, gather, scatter, allgather, alldtoall (see below)
- collective computation – sum, max, min, logical and bitwise and, or, xor, and user defined function



Some of the collective functions come in the variable variant (suffix V), whereby different amounts of data can be sent to or received from different processes. Keep in mind however that in contrast to point-to-point communication the amount of data sent must exactly match the amount of data specified by the receiver. A major simplification is that collective functions come in blocking versions only in the communication mode that can be regarded as analogous to the standard mode of point-to-point communication. Thus, a collective communication (except the barrier) may, or may not, have the effect of synchronizing all calling processes.

10.2.3 Groups, Contexts, and Communicators

A key feature needed to support the creation of robust, parallel libraries is to guarantee that communication within a library routine does not conflict with communication extraneous to the routine. Clearly, tagging the messages is not enough to ensure that. Therefore MPI introduces a communicator, a data object that specifies the scope of communication operation in terms of the group of processes involved and the communication context. A message sent in one context cannot be received in another context. Process ranks are interpreted with respect to the process group associated with a communicator. MPI applications begin with a default communicator MPI_COMM_WORLD, which has as process group the entire set of processes. New communicators are created from existing communicators and the creation of a communicator is a collective operation.

10.2.4 Datatypes

All MPI communication functions take a datatype argument. In the simplest case this will be a basic primitive type, such as an integer or floating-point number. The basic datatypes and their equivalent in Fortran (if available) are

- MPI_INTEGER – INTEGER
- MPI_REAL – REAL
- MPI_DOUBLE_PRECISION – DOUBLE_PRECISION
- MPI_COMPLEX – COMPLEX
- MPI_CHARACTER – CHARACTER(1)
- MPI_BYTE
- MPI_PACKED

An important feature of MPI is that it allows to create user-defined types consisting of basic primitive types. Through user-defined types, MPI supports the communication of complex data structures such as array sections (noncontiguous data) and structures containing combinations of primitive datatypes (e.g. an integer count, followed by a sequence of real numbers) without the necessity to pack them in advance in a local buffer.

10.2.5 Binding to Fortran 77

An example of binding MPI send and receive functions to Fortran code is given by

```
MPI_SEND(BUF, COUNT, DATATYPE, DEST, TAG, COMM, IERROR)
<type> BUF(*)
INTEGER COUNT, DATATYPE, DEST, TAG, COMM, IERROR

MPI_RECV(BUF, COUNT, DATATYPE, SOURCE, TAG, COMM, STATUS, IERROR)
<type> BUF(*)
INTEGER COUNT, DATATYPE, SOURCE, TAG, COMM, IERROR
INTEGER STATUS(MPI_STATUS_SIZE)
```

10.2.6 Example of Parallel Fortran Program

In this example, the value of π is calculated by numerical integration of

$$\int_0^1 \frac{4}{1+x^2} dx = 4(\arctan(1) - \arctan(0)) = 4\frac{\pi}{4} = \pi$$

```
*****
c pi.f - compute pi by integrating f(x) = 4/(1 + x**2)
c
c Each node:
c   1) receives the number of rectangles used in the approximation.
c   2) calculates the areas of its rectangles.
c   3) Synchronizes for a global summation.
c Node 0 prints the result.
c
c Variables:
c
c   pi   the calculated result
c   n    number of points of integration.
c   x    midpoint of each rectangle's interval
c   f    function to integrate
c   sum,pi area of rectangles
c   tmp   temporary scratch space for global summation
c   i    do loop index
*****
program main
  include 'mpif.h'
```

```

double precision PI25DT
parameter      (PI25DT = 3.141592653589793238462643d0)

double precision mypi, pi, h, sum, x, f, a
integer n, myid, numprocs, i, rc
c                           function to integrate
f(a) = 4.d0 / (1.d0 + a*a)

call MPI_INIT( ierr )
call MPI_COMM_RANK( MPI_COMM_WORLD, myid, ierr )
call MPI_COMM_SIZE( MPI_COMM_WORLD, numprocs, ierr )
print *, "Process ", myid, " of ", numprocs, " is alive"

sizetype    = 1
sumtype     = 2

10   if ( myid .eq. 0 ) then
      write(6,98)
98   format('Enter the number of intervals: (0 quits)')
      read(5,99) n
99   format(i10)
      endif

      call MPI_BCAST(n,1,MPI_INTEGER,0,MPI_COMM_WORLD,ierr)
c                           check for quit signal
      if ( n .le. 0 ) goto 30
c                           calculate the interval size
      h = 1.0d0/n

      sum = 0.0d0
      do 20 i = myid+1, n, numprocs
         x = h * (dble(i) - 0.5d0)
         sum = sum + f(x)
20   continue
      mypi = h * sum
c                           collect all the partial sums
      call MPI_REDUCE(mypi,pi,1,MPI_DOUBLE_PRECISION,MPI_SUM,0,
+      MPI_COMM_WORLD,ierr)
c                           node 0 prints the answer.
      if (myid .eq. 0) then
         write(6, 97) pi, abs(pi - PI25DT)
97   format(' pi is approximately: ', F18.16,
+           ' Error is: ', F18.16)
         endif

      goto 10

30   call MPI_FINALIZE(rc)
      stop
      end

```

10.3 Explicit

10.3.1 Preliminaries

Time step definitions, Fig. 9.2:

$$\Delta t^{n+1/2} = t^{n+1} - t^n \quad (10.1-a)$$

$$t^{n+1/2} = \frac{1}{2}(t^{n+1} + t^n) \quad (10.1-b)$$

$$\Delta t^n = t^{n+1/2} - t^{n-1/2} \quad (10.1-c)$$

The central difference formulae for velocity is

$$\dot{\mathbf{d}}^{n+1/2} \stackrel{\text{def}}{=} \mathbf{v}^{n+1/2} = \frac{\mathbf{d}^{n+1} - \mathbf{d}^n}{\Delta t^{n+1/2}} = \frac{1}{\Delta t^{n+1/2}}(\mathbf{d}^{n+1} - \mathbf{d}^n) \quad (10.2)$$

This difference formula can be converted into an integration formula by rearranging terms

$$\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t^{n+1/2} \mathbf{v}^{n+1/2} \quad (10.3)$$

Similarly, the acceleration and the corresponding integration formula are

$$\begin{aligned} \ddot{\mathbf{d}}^n &\stackrel{\text{def}}{=} \mathbf{a}^n \\ \mathbf{v}^{n+1/2} &= \frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{\Delta t^{n+1/2} - \Delta t^{n-1/2}} \\ &= \mathbf{v}^{n-1/2} + \Delta t^n \mathbf{a}^n \end{aligned} \quad (10.4)$$

Hence, velocities are defined at the midpoints of the time steps.

Substituting Eq. 10.2 (expressed at $t^{n+1/2}$ and $t^{n-1/2}$) into Eq. 10.4, the acceleration can be expressed directly in terms of the displacements

$$\ddot{\mathbf{d}}^n \stackrel{\text{def}}{=} \mathbf{a}^n = \frac{\Delta t^{n-1/2}(\mathbf{d}^{n+1} - \mathbf{d}^n) - \Delta t^{n+1/2}(\mathbf{d}^n - \mathbf{d}^{n-1})}{\Delta t^{n+1/2} \Delta t^n \Delta t^{n-1/2}} \quad (10.5)$$

For equal time steps, this reduces to

$$\ddot{\mathbf{d}}^n \stackrel{\text{def}}{=} \mathbf{a}^n = \frac{(\mathbf{d}^{n+1} - 2\mathbf{d}^n + \mathbf{d}^{n-1})}{(\Delta t^n)^2} \quad (10.6)$$

which is the well known central difference formula for the second derivative of a function.

We now consider the time integration of the motion equation at time step n

$$\mathbf{M}\ddot{\mathbf{a}}^n = \mathbf{f}^n = \mathbf{f}^{\text{ext}}(\mathbf{d}^n, t^n) - \mathbf{f}^{\text{int}}(\mathbf{d}^n, t^n) \quad (10.7)$$

subjected to the essential boundary condition

$$g_I(\mathbf{v}^n) = 0 \quad I = 1 \text{ to } n_c \quad \text{on } \Gamma_v \quad (10.8)$$

which is an ordinary differential equation of second order in time.

The internal forces are functions of the nodal displacements (and thus on time), the external forces are function both of time and displacement (uplift forces).

Substituting Eq. 10.7 into 10.4 gives

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \Delta t^n \mathbf{M}^{-1} \mathbf{f}^n \quad (10.9)$$

Energy balance must be satisfied, as numerical instability in nonlinear problems may only manifests itself in a pernicious and subtle manner which will lead uncorrect results. It may also cause exponential growth which may cause localized premature failures.

Hence Energy must be computed as follows:

$$W_{\text{int}}^{n+1} = W_{\text{int}}^n + \frac{\Delta t^{n+1/2}}{2} \left(\mathbf{v}^{n+1/2} \right)^T (\mathbf{f}_{\text{int}}^n + \mathbf{f}_{\text{int}}^{n+1}) = W_{\text{int}}^n + \frac{1}{2} \Delta \mathbf{d}^T (\mathbf{f}_{\text{int}}^n + \mathbf{f}_{\text{int}}^{n+1}) \quad (10.10-a)$$

$$W_{\text{ext}}^{n+1} = W_{\text{ext}}^n + \frac{\Delta t^{n+1/2}}{2} \left(\mathbf{v}^{n+1/2} \right)^T (\mathbf{f}_{\text{ext}}^n + \mathbf{f}_{\text{ext}}^{n+1}) = W_{\text{ext}}^n + \frac{1}{2} \Delta \mathbf{d}^T (\mathbf{f}_{\text{ext}}^n + \mathbf{f}_{\text{ext}}^{n+1}) \quad (10.10-b)$$

The kinetic energy is given by

$$W_{\text{kin}}^n = \frac{1}{2} (\mathbf{v}^n)^T \mathbf{M} \mathbf{v}^n \quad (10.11)$$

where $\Delta \mathbf{d} = \mathbf{d}^{n+1} - \mathbf{d}^n$.

Energy conservation requires that

$$|W_{\text{kin}} + W_{\text{int}} - W_{\text{ext}}| \leq \varepsilon \max(W_{\text{kin}}, W_{\text{int}}, W_{\text{ext}}) \quad (10.12)$$

where ε is a tolerance of the order of 10^{-2}

10.3.2 Parallelization Concept based on Node-Cut Mesh Partitioning

In the node-cut mesh partitioning, the cut runs through element sides and corresponding nodes. The nodes lying on partition boundaries are marked as shared nodes. These nodes are shared by all adjacent partitions. On each partition, the shared nodes have assigned unique local code (equation) numbers. The elements are uniquely assigned to particular partitions. In order to guarantee the correctness of the solution of the partitioned problem, a modification of the single processor algorithm is necessary. The equilibrium equations at local partition nodes are solved without any change. However, at shared nodes, one is confronted with the necessity to assemble contributions from two or more adjacent partitions. The correctness has to be enforced by exchange of contributions of shared node internal and external forces between partitions. Each partition has to add the contributions received from neighbouring partitions to the locally assembled shared node internal and external force and to send its shared node contributions to neighbouring partitions. Since the partitioned domains contain only the local elements, the correct mass matrix has to be established by an analogous data exchange operation before the time-stepping algorithm starts. As far as the damping is assumed in the Rayleigh form, there is no need for transfer of contributions to the correct damping matrix.

The process of mutual exchange of internal nodal force contributions must be repeated for each time step to guarantee the correctness of the solution. In order to efficiently handle this exchange, each partition assembles its send and receive communication maps for all partitions. While the send map contains the shared node numbers, for which the exchange, in terms of sending the local contributions to a particular remote partition, is required, the receive map contains the shared node numbers, for which the exchange, in terms of receiving the contributions from a particular remote partition, is required. The nice property of node-cut approach is that the send and receive maps are identical.

10.3.3 Algorithm

0. **MPI:** Build communication maps (note that this step can be done without communication if in each input file will be for each shared node list of partitions sharing it)
 - a) get the maximum number of shared nodes on each partition using `MPI_ALLREDUCE` function
 - b) broadcast (`MPI_BCAST`) the list of global ids of shared nodes to other partitions
 - c) receive the broadcasted list (`MPI_BCAST`) from other partitions
 - d) setup the communication map for each remote partition (array of shared node local ids sorted by their global id)
1. Initial allocation of work arrays, \mathbf{a}^0 , \mathbf{v}^0 , $\boldsymbol{\sigma}^0$ and other state variables.
2. Assemble lumped diagonal mass matrix \mathbf{M} (vector stored).
3. **MPI:** Appropriate distribution of mass matrix \mathbf{M} to individual nodes and the summation of individual contributions (`MPI_ISEND`, `MPI_IRECV`)
4. Assemble the first part (mass proportional) of the damping matrix $\mathbf{C} = a_1 \mathbf{M}$.
5. Assemble initial external load vectors $\mathbf{f}^{\text{ext},0}$ (includes gravity, hydrostatic, uplift, etc...)
6. **MPI:** Communicate the local contributions to $\mathbf{f}^{\text{ext},0}$ to other partitions (`MPI_ISEND`, `MPI_IRECV`)
7. Calculate internal force vector $\mathbf{f}^{\text{int},0}$ and critical time step $\Delta t^{\text{crit}} = \alpha \min \frac{l_e}{\sqrt{E/\rho}}$
8. **MPI:** Appropriate distribution of internal forces $\mathbf{f}^{\text{int},0}$ to individual nodes and their summation, plus distribution of Δt^{crit} .
9. Initialization of
 - a) \mathbf{d}^0 based on input data
 - b) If \mathbf{a}^0 is prescribed: $\mathbf{v}^0 = \mathbf{C}^{-1} (\mathbf{f}^{\text{ext},0} - \mathbf{M}\mathbf{a}^0 - \mathbf{f}^{\text{int},0})$
 - c) If \mathbf{v}^0 is prescribed: $\mathbf{a}^0 = \mathbf{M} (\mathbf{f}^{\text{ext},0} - \mathbf{C}\mathbf{v}^0 - \mathbf{f}^{\text{int},0})$
10. Start loop over time steps $t^{k+1} = t^k + \Delta \hat{t}$.
11. Read loading and prescribed $\tilde{\mathbf{a}}^{k+1}$, $\tilde{\mathbf{v}}^{k+1}$, and $\tilde{\mathbf{d}}^{k+1}$

12. Assemble external load vector $\mathbf{f}^{\text{ext},k+1}$
13. **MPI:** Appropriate distribution of external forces $\mathbf{f}^{\text{ext},k+1}$ to individual nodes and the summation of individual contributions.
14. Initialize load, displacement, velocity and acceleration vectors

$$\left\{ \begin{array}{lcl} \mathbf{f}^{\text{ext},n} & = & \mathbf{f}^{\text{ext},k} \\ \mathbf{d}^n & = & \mathbf{d}^k \\ \mathbf{v}^n & = & \mathbf{v}^k \\ \mathbf{a}^n & = & \mathbf{a}^k \end{array} \right. \quad (10.13)$$

Note: k refers to the user defined time stepping, whereas n (defined below) refers to the substeps necessary to satisfy $\Delta t \leq \Delta t_{\text{crit}}$.

- a) Start loop over substeps

$$\Delta t = \min(\Delta \hat{t}, \Delta t^{\text{crit}}) \quad (10.14-\text{a})$$

$$\Delta \hat{t} = \Delta \hat{t} - \Delta t \quad (10.14-\text{b})$$

- i. Time update

$$t^{n+1} = t^n + \Delta t, \quad t^{n+1/2} = \frac{1}{2}(t^n + t^{n+1}) \quad (10.15)$$

- ii. Partial velocity update

$$\mathbf{v}^{n+1/2} = \mathbf{v}^n + (t^{n+1/2} - t^n) \mathbf{a}^n \quad (10.16)$$

- iii. Enforce prescribed velocities on Γ_v

$$\mathbf{v}^{n+1/2} = \frac{t^{k+1} - t^{n+1/2}}{t^{k+1} - t^n} \mathbf{v}^n + \frac{t^{n+1/2} - t^n}{t^{k+1} - t^n} \tilde{\mathbf{v}}^{k+1} \quad (10.17)$$

- iv. Update nodal displacements

$$\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t \mathbf{v}^{n+1/2} \quad (10.18)$$

- v. Enforce prescribed displacements on Γ_d

$$\mathbf{d}^{n+1} = \frac{t^{k+1} - t^{n+1}}{t^{k+1} - t^n} \mathbf{d}^n + \frac{t^{n+1} - t^n}{t^{k+1} - t^n} \tilde{\mathbf{d}}^{k+1} \quad (10.19)$$

- vi. **GET FORCES** (Calculate internal force vector $\mathbf{f}^{\text{int},n+1}$ and critical time Δt^{crit}).

- vii. **MPI:** Appropriate distribution of internal forces $\mathbf{f}^{\text{int},n+1}$ to individual nodes and their summation, plus distribution of Δt^{crit} .

- viii. Interpolate external loads for step $n + 1$

$$\mathbf{f}^{\text{ext},n+1} = \frac{t^{k+1} - t^{n+1}}{t^{k+1} - t^n} \mathbf{f}^{\text{ext},n} + \frac{t^{n+1} - t^n}{t^{k+1} - t^n} \mathbf{f}^{\text{ext},k+1} \quad (10.20)$$

- ix. Compute acceleration

$$\mathbf{a}^{n+1} = \mathbf{M}^{-1} \left(\mathbf{f}^{\text{ext},n+1} - \mathbf{f}^{\text{int},n+1} - a_2 \frac{\mathbf{f}^{\text{int}n+1} - \mathbf{f}^{\text{int}n}}{\Delta t} - \mathbf{C} \mathbf{v}^{n+1/2} \right) \quad (10.21)$$

Note third term is the stiffness proportional damping factor.

- x. Enforce prescribed acceleration on Γ_a

$$\mathbf{a}^{n+1} = \frac{t^{k+1} - t^{n+1}}{t^{k+1} - t^n} \mathbf{a}^n + \frac{t^{n+1} - t^n}{t^{k+1} - t^n} \tilde{\mathbf{a}}^{k+1} \quad (10.22)$$

xi. Update velocities

$$\mathbf{v}^{n+1} = \mathbf{v}^{n+1/2} + (t^{n+1} - t^{n+1/2})\mathbf{a}^{n+1} \quad (10.23)$$

xii. Enforce prescribed velocities on Γ_v

$$\mathbf{v}^{n+1} = \frac{t^{k+1} - t^{n+1}}{t^{k+1} - t^{n+1/2}}\mathbf{v}^{n+1/2} + \frac{t^{n+1} - t^{n+1/2}}{t^{k+1} - t^{n+1/2}}\tilde{\mathbf{v}}^{k+1} \quad (10.24)$$

xiii. Check energy balance every m time steps where $m \simeq XX$

A. Compute

$$W_{int}^{n+1} = W_{int}^n + \frac{1}{2}\Delta\mathbf{d}^T(\mathbf{f}_{int}^n + \mathbf{f}_{int}^{n+1}) \quad (10.25-a)$$

$$W_{ext}^{n+1} = W_{ext}^n + \frac{1}{2}\Delta\mathbf{d}^T(\mathbf{f}_{ext}^n + \mathbf{f}_{ext}^{n+1}) \quad (10.25-b)$$

$$W_{kin}^{n+1} = \frac{1}{2}(\mathbf{v}^{n+1})^T \mathbf{M} \mathbf{v}^{n+1} \quad (10.25-c)$$

B. **MPI:** Sum the contributions to W_{int}^{n+1} , W_{ext}^{n+1} , and W_{kin}^{n+1} from all partitions using **MPI_ALLREDUCE** (every k time steps)

C. Check if

$$|W_{kin} + W_{int} - W_{ext}| \leq \varepsilon \max(W_{kin}, W_{int}, W_{ext}) \quad (10.26)$$

where $\varepsilon \simeq 10^{-2}$

b) End loop over substeps, end loop if $\Delta\hat{t} \leq 0$.

15. Update incremental data plus other stuff to be done at the end of increments.

16. End loop over time steps

Algorithm Get Forces

1. Initialize $\mathbf{f}^n = \mathbf{0}$;
2. Set $\Delta t_{crit} = \infty$
3. Loop over elements (e).
 - a) Gather element nodal displacements and velocities from global array
 - b) Set $\mathbf{f}_{(e)}^{int,n} = \mathbf{0}$
 - c) Loop over Gauss points ξ_Q
 - i. Compute stresses $\boldsymbol{\sigma}^n(\xi_Q)$ through the constitutive equation.
 - ii. $\mathbf{f}_{(e)}^{int,n} \leftarrow \mathbf{f}_{(e)}^{int,n} + \mathbf{B}^T \boldsymbol{\sigma}^n(\xi_Q) \bar{w}_Q J|_Q$
 - d) Update internal forces $\mathbf{f}_{(e)}^{int,n} \leftarrow \mathbf{f}_{(e)}^{int,n} + \mathbf{f}_{(e,dc)}^{int,n}$
 - e) Compute $\Delta t_{crit}^{(e)} = \alpha \min \frac{L_{(e)}}{c} = \alpha \min \frac{L_{(e)}}{\sqrt{E/\rho}}$ (We can ignore zero thickness interface elements with cohesive stresses because these are zero mass elements).
 - f) If $\Delta t_{crit}^{(e)} < \Delta t_{crit}$, then $\Delta t_{crit} = \Delta t_{crit}^{(e)}$
 - g) Element internal force vector is added to global internal force vector

10.3.3.1 Note About Interface Elements

The evaluation of the Δt_{cr} is of paramount importance in the explicit method. Δt_{cr} corresponding to the time it takes a seismic wave to cross an element can be estimated by

$$\Delta t_{cr} = \alpha \min \frac{L_{(e)}}{\sqrt{\frac{E}{\rho}}} \quad (10.27)$$

where $\alpha < 1.0$.

We recall that zero thickness interface elements are

1. Formulated in terms of the relative displacements ($\Delta \mathbf{u}$).
2. Have zero thickness.
3. Are assigned a normal thickness

$$K_n = \frac{E}{t} \quad (10.28)$$

where E is the elastic modulus of the adjacent material, and t is an estimate of the actual physical thickness of the interface in the prototype.

Hence, for the evaluation of Δt_{cr} , we should consider this physical thickness of the prototype t in lieu of $L_{(e)}$.

Furthermore, we can define

$$\rho = \frac{\rho_i}{t} \quad (10.29)$$

where ρ is the actual mass density, and ρ_i is the interface mass density. Substituting in Eq. 10.27, explicit reference to the thickness t cancels out

$$\Delta t_{cr} = \alpha \frac{t}{\sqrt{\frac{K_n t}{\frac{\rho_i}{t}}}} = \alpha \sqrt{\frac{\rho_i}{K_n}} \quad (10.30)$$

where ρ_i and K_n are material properties assigned for the interface element

Chapter 11

HOURGLASS STABILIZATION

Adapted from (Belytschko et al. 2000)

In order to accelerate explicit analysis, it is beneficial to reduce the element order of integration. However, it is well known that underintegration will result in hourglass modes (specially for linear elements) and mesh locking. To mitigate this unpleasant effect, special measures must be taken to eliminate this effect. One such approach, implemented in Merlin, is the so-called Perturbation Hourglass Stabilization procedure, (Belytschko et al. 2000).

In this method, (Belytschko, Ong, Liu and Kennedy 1984), a small correction is added to the discretization in order to restore the (lost through the reduced integration) rank of the element stiffness matrix. However, it is important to augment the rank without disturbing the linear completeness of the isoparametric element. Hence, one approach is to augment the one point quadrature (linear) element by two rows which are orthogonal to the other three. This orthogonalization ensures that the additional rows are linearly independent of the first three and that the correction does not affect the response to linear fields.

The additional rows of the \mathbf{B} matrix are the $\boldsymbol{\gamma}$ vector given by

$$\boldsymbol{\gamma} = \frac{1}{4} [\mathbf{h} - (\mathbf{h}^T \mathbf{x}) \mathbf{b}_x - (\mathbf{h}^T \mathbf{y}) \mathbf{b}_y] \quad (11.1)$$

where $\mathbf{h} = [1 \ -1 \ 1 \ -1]$ and

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_x^T & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_y^T \\ \mathbf{b}_y^T & \mathbf{b}_x^T \end{bmatrix} \quad (11.2)$$

for the Q4 element.

Hence, the \mathbf{B} matrix is augmented as follows

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{b}_x^T & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_y^T \\ \mathbf{b}_y^T & \mathbf{b}_x^T \\ \boldsymbol{\gamma}^T & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\gamma}^T \end{bmatrix} \quad (11.3)$$

and the corresponding stress vectors are now given by

$$\{\boldsymbol{\sigma}\} = [\sigma_x \ \sigma_y \ \sigma_{xy} \ Q_x \ Q_y]^T \quad (11.4)$$

The constitutive matrix is also correspondingly augmented:

$$\tilde{\mathbf{E}} = \begin{bmatrix} E_{11} & E_{12} & E_{13} & 0 & 0 \\ E_{21} & E_{22} & E_{23} & 0 & 0 \\ E_{31} & E_{32} & E_{33} & 0 & 0 \\ 0 & 0 & 0 & E^Q & 0 \\ 0 & 0 & 0 & 0 & E^Q \end{bmatrix} \quad (11.5)$$

Finally, using this stabilization procedure, the linear stiffness matrix is given by

$$\mathbf{K}_{(e)} = \mathbf{K}_{(e)}^{1pt} + E^Q A \begin{bmatrix} \boldsymbol{\gamma} \boldsymbol{\gamma}^T & \boldsymbol{\gamma} \boldsymbol{\gamma}^T \\ \boldsymbol{\gamma} \boldsymbol{\gamma}^T & \boldsymbol{\gamma} \boldsymbol{\gamma}^T \end{bmatrix} \quad (11.6)$$

where A is the element area (to be replaced by V for brick elements) and the rank of the element stiffness matrix is again 5 which is the correct one for the Q4 element.

E^Q is given by

$$E^Q = \frac{1}{2} \alpha_s c^2 \rho A \mathbf{b}_i^T \mathbf{b}_i \quad (11.7)$$

where α_s is a scaling parameter, and it is recommended that it be about equal to 0.1, α_s the elastic dilational wave speed, and ρ the specific mass density.

In Merlin four elements incorporate this hourglass control:

- a) Element 71: 4 noded quadrilateral for plane stress 2D analysis.
- b) Element 72: 4 noded quadrilateral for plane strain 2D analysis.
- c) Element 73: 4 noded quadrilateral for axisymmetric 2D analysis.
- d) Element 74: 8 noded brick element for 3D analysis Literature.

Chapter 12

EMBEDDED REINFORCEMENT

The stiffness matrix of a rod element is given by the classical equation

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega \quad (12.1)$$

where \mathbf{B} is really composed of two parts,

$$\underbrace{\mathbf{B}}_{1 \times 4} = \underbrace{\mathbf{B}_1}_{1 \times 2} \underbrace{\mathbf{B}_2}_{2 \times 4} \quad (12.2)$$

where \mathbf{B}_2 transforms the displacements from global to local coordinates, and \mathbf{B}_1 determines the derivative for the strain.

The stiffness matrix is a 4×4 matrix in 2D, and 6×6 in 3D for an arbitrarily oriented element in space. The embedded reinforcement is going to intersect edges of continuum solid elements. At each one of those points, we can relate the (bar) displacements to those of the element through the shape functions. Considering a 3 noded triangle,

$$\left\{ \begin{array}{c} \mathbf{u}_1 \\ \mathbf{u}_2 \end{array} \right\} = \left[\begin{array}{c} \mathbf{N}(\xi_1) \\ \mathbf{N}(\xi_2) \end{array} \right] \mathbf{u}_s = \underbrace{\mathbf{B}^*}_{4 \times 6} \mathbf{u}_s \quad (12.3)$$

where \mathbf{u}_s is the nodal displacement vector of the solid element, \mathbf{u}_i are the nodal displacements of the bar node i , and $\mathbf{N}(\xi_i)$ are shape functions values of the solid element that are evaluated at bar node i with natural coordinates ξ_i .

The stiffness matrix of the embedded element is thus added to the one of the solid element

$$\mathbf{K}^{total} = \mathbf{K}^{solid} + \mathbf{K}^{bar} \quad (12.4)$$

where

$$\underbrace{\mathbf{K}^{bar}}_{6 \times 6} = \int_{\Omega} \mathbf{B}'^T \mathbf{D} \mathbf{B}' d\Omega \quad (12.5)$$

where

$$\underbrace{\mathbf{B}'}_{1 \times 6} = \underbrace{\mathbf{B}}_{1 \times 4} \underbrace{\mathbf{B}^*}_{4 \times 6} \quad (12.6-a)$$

$$\mathbf{D} = \frac{AE}{L} \quad (12.6-b)$$

When an embedded reinforcement crosses a crack, Fig. 12.1, then

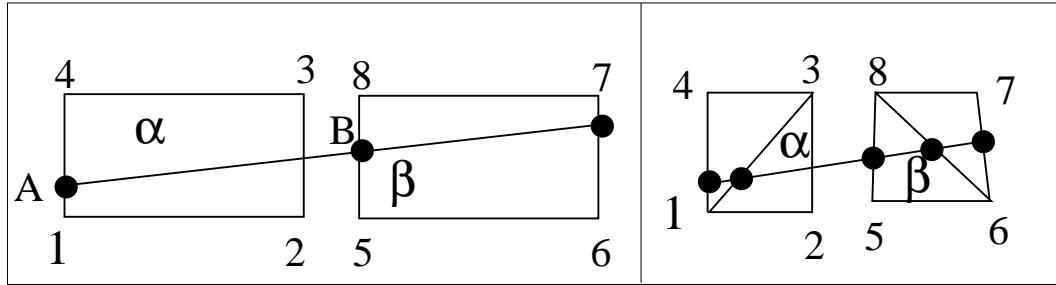


Figure 12.1: Embedded Reinforcement Across a Crack/Joint

1. The element stiffness matrices of the two adjacent elements are computed and added to the global stiffness matrix.

2. A virtual super element, obtained by adding the connectivity vectors of the two elements α and β (1-2-3 and 5-6-8 or 1-2-3-4 and 5-6-7-8) is created. The \mathbf{B}^* matrix is now given by

$$\left\{ \begin{array}{c} \mathbf{u}_1 \\ \mathbf{u}_2 \end{array} \right\} = \underbrace{\left[\begin{array}{cc} \mathbf{N}(\xi_\alpha) & 0 \\ 0 & \mathbf{N}(\xi_\beta) \end{array} \right]}_{\substack{\mathbf{B}^* \\ 4 \times 12}} \left\{ \begin{array}{c} \mathbf{u}_\alpha \\ \mathbf{u}_\beta \end{array} \right\} \quad (12.7)$$

where ξ_α are the natural coordinates of the first bar node in the first element, and ξ_β the natural coordinates of the second bar node in the second element (the first bar node (1) contributes to the stiffness of element α , and the second node (2) contributes to the adjacent element β); \mathbf{u}_α and \mathbf{u}_β are the nodal displacements of the two virtual elements.

3. Hence, the size of \mathbf{B}' is

$$\underbrace{\mathbf{B}'}_{1 \times 12} = \underbrace{\mathbf{B}}_{1 \times 4} \underbrace{\mathbf{B}^*}_{4 \times 12} \quad (12.8)$$

and

$$\underbrace{\mathbf{K}^{bar}}_{12 \times 12} = \int_{\Omega} \mathbf{B}'^T \mathbf{D} \mathbf{B}' d\Omega \quad (12.9)$$

4. If the two elements have identical nodes (i.e. are not intersected by a crack), then we essentially recover Eq. 12.3 since the stiffness terms of the second bar node will contribute to the same stiffness term in the global stiffness matrix (and the second bar node does not contribute to the stiffness terms corresponding to node 6).

Chapter 13

SINGULAR ELEMENT

Theoretical background for determination of SIF in Merlin using singular elements

13.1 Introduction

For most practical problems, either there is no analytical solution, or the handbook ((Tada, Paris and Irwin 1973)) ones are only crude approximation. Hence numerical techniques should be used. Whereas Boundary Element Methods are increasingly being used, (Aliabadi and Rooke 1991), they are far behind in sophistication the Finite Element Methods which will be exclusively covered in this chapter. For an overview of early finite element techniques in finite elements the reader should consult (Owen and Fawkes 1983), some of the more recent methods are partially covered in (Anderson 1995). Finally, the Ph.D. thesis of Reich (1993) and of Červenka, J. (1994) contain some of the major extensions of modern techniques to include thermal load, body forces, surface tractions in 2D and 3D respectively.

Numerical methods for fracture mechanics can be categorized in many different ways, in this chapter we shall use three criteria:

1. Those in which the singularity is modelled, that is the $r^{-\frac{1}{2}}$ stress field at the tip of the crack is properly represented.
2. Techniques in which the SIF are directly evaluated as part of the augmented global stiffness matrix.
3. Techniques through which the SIF can be computed *a post priori* following a standard finite element analysis via a special purpose post-processor.

13.2 Quarter Point Singular Elements

This section discusses the easiest and most powerful technique used in finite elements to model a stress singularity.

Barsoum (Barsoum 1974) and Henshell and Shaw (Henshell and Shaw 1975) independently demonstrated that the inverse square root singularity characteristic of linear elastic fracture mechanics can be obtained in the 2D 8-noded isoparametric element (Q8) when the mid-side nodes near the crack tip are placed at the quarter point. Thus, in order to model a stress singularity without altering the finite element code, the mid-side nodes adjacent to the crack tip must be shifted to their quarter-point position. Since then this element became known as the quarter-point element. In light of the simplicity and accuracy achieved by this element, this section will:

1. cover a brief review of the isoparametric element formulation
2. show how the element can be distorted in order to achieve a stress singularity
3. determine the order of the stress singularity
4. provide a brief review of all the historical developments surrounding this element
5. discuss the effect on numerical accuracy of element size, order of integration, and local meshing around the crack tip
6. briefly mention references to other singular elements

13.3 Review of Isoparametric Finite Elements

In the isoparametric finite element representation, both the internal displacement and coordinates are related to their nodal values through the shape functions:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \sum_{i=1}^8 \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} \begin{Bmatrix} \bar{x}_i \\ \bar{y}_i \end{Bmatrix} \quad (13.1)$$

$$\{d\} = \begin{Bmatrix} u \\ v \end{Bmatrix} = \sum_{i=1}^8 \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} \begin{Bmatrix} \bar{u}_i \\ \bar{v}_i \end{Bmatrix} \quad (13.2)$$

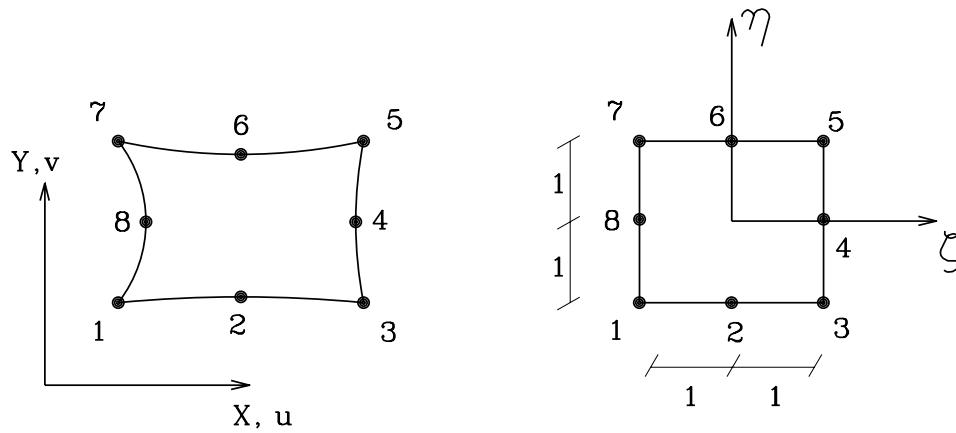


Figure 13.1: Isoparametric Quadratic Finite Element: Global and Parent Element

where the N_i are the assumed shape functions. For quadratic isoparametric serendipity elements (Fig. 13.1) the shape functions are given by:

$$N_i = \frac{1}{4} (1 + \xi \xi_i) (1 + \eta \eta_i) (\xi \xi_i + \eta \eta_i - 1), \quad i = 1, 3, 5, 7 \quad (13.3)$$

$$N_i = \frac{1}{2} (1 - \xi^2) (1 + \eta \eta_i), \quad i = 2, 6 \quad (13.4)$$

$$N_i = \frac{1}{2} (1 + \xi \xi_i) (1 - \eta^2), \quad i = 4, 8 \quad (13.5)$$

In Fig. 13.1, x_i, y_i are the nodal coordinates, u_i, v_i are the nodal displacements.

As the strain is the derivative of the displacement, we will need later to define $\frac{\partial N}{\partial x}$ and $\frac{\partial N}{\partial y}$. N has been defined in Eq. 13.3 - 13.5 in terms of the natural coordinates ξ and η . Thus the chain rule will have to be invoked and the inverse of the jacobian will be needed. In this case, the jacobian matrix is:

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (13.6)$$

$$= \begin{bmatrix} \sum_{i=1}^8 \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^8 \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix} \quad (13.7)$$

The inverse jacobian is then evaluated from:

$$[J]^{-1} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} \quad (13.8)$$

$$= \frac{1}{\text{Det} J} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \quad (13.9)$$

The strain displacement relationship is:

$$\{\epsilon\} = \sum_{i=1}^8 [B_i] [\bar{d}_i] \quad (13.10)$$

where $[B_i]$ is the strain matrix given by:

$$[B_i] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix} \quad (13.11)$$

where the following chain rule is invoked to determine the coefficients of $[B]$:

$$\left\{ \begin{array}{c} \frac{\partial N}{\partial x} \\ \frac{\partial N}{\partial y} \end{array} \right\} = [J]^{-1} \left\{ \begin{array}{c} \frac{\partial N}{\partial \xi} \\ \frac{\partial N}{\partial \eta} \end{array} \right\} \quad (13.12)$$

Finally, it can be shown that the element stiffness matrix of an element is given by (Gallagher 1975), (Zienkiewicz 1967):

$$[K] = \int_{-1}^1 \int_{-1}^1 [B(\xi, \eta)] [D] [B(\xi, \eta)] \det J d\xi d\eta \quad (13.13)$$

where the natural coordinates ξ and η are shown in Fig. 13.1 and $[D]$ is the stress-strain or constitutive matrix.

The stress is given by:

$$\{\sigma\} = [D] [B] \left\{ \begin{array}{c} \bar{u}_i \\ \bar{v}_i \end{array} \right\} \quad (13.14)$$

13.4 How to Distort the Element to Model the Singularity

In Eq. 13.14, if the stresses are to be singular, then $[B]$ has to be singular as the two other components are constants. Consequently, if $[B]$ is to be singular then the determinant of J must vanish to zero (Eq. 13.6) at the crack tip.

Now considering a rectangular element of length L along its first side (1-2-3, in Fig. 13.1), we can readily see that both off-diagonal terms ($\frac{\partial u}{\partial \xi}$ and $\frac{\partial v}{\partial \eta}$) are zero. Thus, for the determinant of the jacobian to be zero we must have either one of the diagonal terms equal to zero. It will suffice to force $\frac{\partial x}{\partial \xi}$ to be zero. Making the proper substitution for $\frac{\partial x}{\partial \xi}$ at $\eta = -1$ we have:

$$\begin{aligned} \frac{\partial x}{\partial \xi} \Big|_{\eta=-1} &= \sum_{i=1}^8 N_i \bar{x}_i \\ &= \frac{1}{4} [-1 + 2\xi + 2\xi + 1](0) \\ &\quad + \frac{1}{4} [1 + 2\xi + 2\xi + 1](L) \\ &\quad + \frac{1}{4} [-1 + 2\xi - 2\xi + 1](L) \\ &\quad + \frac{1}{4} [1 - 2\xi + 2\xi - 1](0) \\ &\quad + \frac{1}{2} (-2\xi - 2\xi)(x_2) \\ &\quad + \frac{1}{2} (-2\xi + 2\xi) \left(\frac{L}{2} \right) \\ &\quad + \frac{1}{2} (1 - 1)(L) \\ &\quad + \frac{1}{2} (-1 + 1)(0) \\ &= \frac{1}{4} (2 + 4\xi)L + \frac{1}{2} (-4\xi)x_2 \end{aligned} \quad (13.15)$$

After simplification, and considering the first corner node (where $\eta = \xi = -1$), we would have:

$$\frac{\partial x}{\partial \xi} \Big|_{\substack{\xi=-1 \\ \eta=-1}} = 0 \Leftrightarrow (1 - 2) \frac{L}{2} + 2x_2 = 0 \quad (13.16)$$

$x_2 = \frac{L}{4}$

(13.17)

Thus all the terms in the jacobian vanish if and only if the second node is located at $\frac{L}{4}$ instead of $\frac{L}{2}$, and subsequently both the stresses and strains at the first node will become singular.

Thus singularity at the crack tip is achieved by shifting the mid-side node to its quarter-point position, see Fig. 13.2.

We should observe that instead of enforcing $\frac{\partial x}{\partial \xi}$ along edge 1-3 to vanish at the crack tip, we could have enforced $\frac{\partial y}{\partial \eta}$ along edge 1-7 to be zero at the crack tip.

A similar approach will show that if node 8 is shifted to its quarter-point position the same radial strain variation would be obtained along sides 1-7. However, along rays within the element emanating from node 1 the strain variation is not singular. The next section will discuss this issue and other variation of this distorted element in more detail.

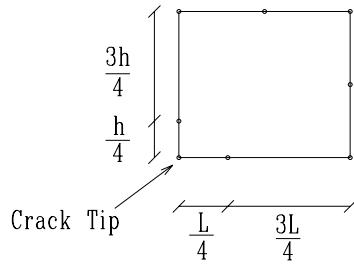


Figure 13.2: Singular Element (Quarter-Point Quadratic Isoparametric Element)

13.5 Order of Singularity

Having shown that the stresses at the first node are singular, the obvious question is what is the degree of singularity.

First let us solve for ξ in terms of x and L at $\eta = -1$ (that is, alongside 1-2-3):

$$\begin{aligned} x &= \sum_{i=1}^8 N_i \bar{x}_i \\ &= \frac{1}{2} (1 - \xi^2) (1 + 1) \frac{L}{4} + \frac{1}{4} (1 + \xi) (1 + 1) (\xi) L \\ &= \frac{1}{2} \xi (1 + \xi) L + (1 - \xi^2) \frac{L}{4} \end{aligned} \quad (13.18)$$

$$\Rightarrow \xi = -1 + 2\sqrt{\frac{x}{L}} \quad (13.19)$$

Recalling that in isoparametric elements the displacement field along $\eta = -1$ is given by:

$$u = -\frac{1}{2} \xi (1 - \xi) \bar{u}_1 + \frac{1}{2} \xi (1 + \xi) \bar{u}_2 + (1 - \xi^2) \bar{u}_3 \quad (13.20)$$

we can rewrite Eq. 13.20 by replacing ξ with the previously derived expression, Eq. 13.19):

$$\begin{aligned} u &= -\frac{1}{2} \left(-1 + 2\sqrt{\frac{x}{L}} \right) \left(2 - 2\sqrt{\frac{x}{L}} \right) \bar{u}_1 \\ &+ \frac{1}{2} \left(-1 + 2\sqrt{\frac{x}{L}} \right) \left(2\sqrt{\frac{x}{L}} \right) \bar{u}_2 \\ &+ \left(4\sqrt{\frac{x}{L}} - 4\frac{x}{L} \right) \bar{u}_3 \end{aligned} \quad (13.21)$$

This complex equation can be rewritten in the form:

$$u = A + Bx + C\sqrt{\frac{x}{L}}$$

(13.22)

We thus note that the displacement field has had its quadratic term replaced by $x^{\frac{1}{2}}$, which means that when the derivative of the displacement is taken, the strain (and stresses) are of the form:

$$\epsilon_x = -\frac{1}{2} \left(\frac{3}{\sqrt{xL}} - \frac{4}{L} \right) \bar{u}_1 + \frac{1}{2} \left(\frac{-1}{\sqrt{xL}} + \frac{4}{L} \right) \bar{u}_2 + \left(\frac{2}{\sqrt{xL}} - \frac{4}{L} \right) \bar{u}_3 \quad (13.23)$$

Thus the strength of the singularity is of order $\frac{1}{2}$, just as we wanted it to be for linear elastic fracture mechanics !

13.6 Stress Intensity Factors Extraction

A number of techniques (including the ones discussed in the subsequent section) can be used to determine the SIF when quarter-point elements are used, Fig. 13.3 but by far the simplest one to use and implement is the one based on the nodal displacement correlation technique.

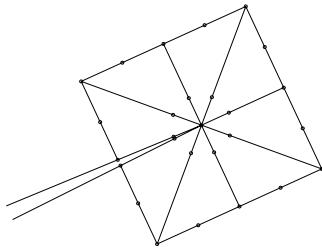


Figure 13.3: Finite Element Discretization of the Crack Tip Using Singular Elements

This technique, first introduced by Shih *et al.* (Shih, de Lorenzi and German 1976), equates the displacement field in the quarter-point singular element with the theoretical one. This method was subsequently refined by Lynn and Ingraffea (Lynn and Ingraffea 1977) who introduced the transition elements, and extended by Manu and Ingraffea to three-dimensional isotropic problems (Ingraffea and Manu 1980).

This method was finally extended to full three-dimensional anisotropic cases by Saouma and Sikiotis (Saouma and Sikiotis 1986).

13.6.1 Isotropic Case

For the quarter-point singular element, in two dimensions, and with reference to Fig. 13.4, the displacement field is

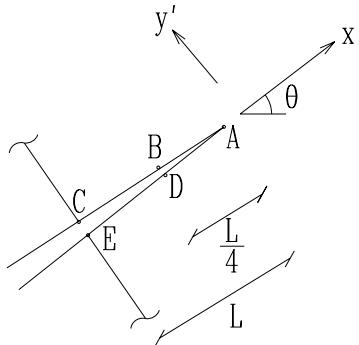


Figure 13.4: Displacement Correlation Method to Extract SIF from Quarter Point Singular Elements

given by:

$$u' = \bar{u}'_A + \left(-3\bar{u}'_A + 4\bar{u}'_B - \bar{u}'_C \right) \sqrt{\frac{r}{L}} + \left(2\bar{u}'_A + 2\bar{u}'_C - 4\bar{u}'_B \right) \frac{r}{L} \quad (13.24)$$

$$v' = \bar{v}'_A + \left(-3\bar{v}'_A + 4\bar{v}'_B - \bar{v}'_C \right) \sqrt{\frac{r}{L}} + \left(2\bar{v}'_A + 2\bar{v}'_C - 4\bar{v}'_B \right) \frac{r}{L} \quad (13.25)$$

where \bar{u}' and \bar{v}' are the local displacements (with x' aligned with the crack axis) of the nodes along the crack in the singular elements.

On the other hand, the analytical expression for v is given by Eq. ?? with $\theta = 180$, yielding:

$$v = K_I \frac{\kappa + 1}{2G} \sqrt{\frac{r}{2\pi}} \quad (13.26)$$

Equating the terms of equal power ($\frac{1}{2}$) in the preceding two equations, the \sqrt{r} term vanishes, and we obtain:

$$K_I = \frac{2G}{\kappa + 1} \sqrt{\frac{2\pi}{L}} \left(-3\bar{v}'_A + 4\bar{v}'_B - \bar{v}'_C \right) \quad (13.27)$$

If this approach is generalized to mixed mode problems, then the two stress intensity factors are given by:

$$\left\{ \begin{array}{c} K_I \\ K_{II} \end{array} \right\} = \frac{1}{2} \frac{2G}{\kappa+1} \sqrt{\frac{2\pi}{L}} \left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right] \left[\begin{array}{c} -3\bar{u}'_A + 4(\bar{u}'_B - \bar{u}'_D) - (\bar{u}'_C - \bar{u}'_E) \\ -3\bar{v}'_A + 4(\bar{v}'_B - \bar{v}'_D) - (\bar{v}'_C - \bar{v}'_E) \end{array} \right] \quad (13.28)$$

Thus it can be readily seen that the extraction of the SIF can be accomplished through a “post-processing” routine following a conventional finite element analysis in which the quarter-point elements have been used.

13.6.2 Anisotropic Case

Following a similar procedure to the one previously described, for the anisotropic case,¹ Saouma and Sikiotis (Saouma and Sikiotis 1986) have shown that the three stress intensity factors can be evaluated from:

$$\left\{ \begin{array}{c} K_I \\ K_{II} \\ K_{III} \end{array} \right\} = [B]^{-1} [A] \sqrt{\frac{2\pi}{L}} \quad (13.29)$$

where $[A]$ is obtained from the displacements of those nodes along the crack in the singular quarter-point wedge element, as shown in Fig. 13.5:

$$[A] = \left[\begin{array}{c} 2\bar{u}_B - \bar{u}_C + 2\bar{u}_E - \bar{u}_F + \bar{u}_D + \frac{1}{2}\eta(-4\bar{u}_B + \bar{u}_C + 4\bar{u}_E - \bar{u}_F) + \frac{1}{2}\eta^2(\bar{u}_F + \bar{u}_C - 2\bar{u}_D) \\ 2\bar{v}_B - \bar{v}_C + 2\bar{v}_E - \bar{v}_F + \bar{v}_D + \frac{1}{2}\eta(-4\bar{v}_B + \bar{v}_C + 4\bar{v}_E - \bar{v}_F) + \frac{1}{2}\eta^2(\bar{v}_F + \bar{v}_C - 2\bar{v}_D) \\ 2\bar{w}_B - \bar{w}_C + 2\bar{w}_E - \bar{w}_F + \bar{w}_D + \frac{1}{2}\eta(-4\bar{w}_B + \bar{w}_C + 4\bar{w}_E - \bar{w}_F) + \frac{1}{2}\eta^2(\bar{w}_F + \bar{w}_C - 2\bar{w}_D) \end{array} \right] \quad (13.30)$$

and $[B]$ is obtained from the analytical solution to the displacements around the crack tip in homogeneous anisotropic solids:

$$[B]^{-1} = \left[\begin{array}{ccc} \text{Re} \left[\frac{i}{s_1-s_2} (q_2 - q_1) \right] \frac{1}{D} & \text{Re} \left[\frac{-i}{s_1-s_2} (p_2 - p_1) \right] \frac{1}{D} & 0 \\ \text{Re} \left[\frac{-i}{s_1-s_2} (s_1 q_2 - s_2 q_1) \right] \frac{1}{D} & \text{Re} \left[\frac{i}{s_1-s_2} (s_1 p_2 - s_2 p_1) \right] \frac{1}{D} & 0 \\ 0 & 0 & \frac{1}{(c_{44} c_{55} - c_{45}^2)} \end{array} \right] \quad (13.31)$$

¹Anisotropic modeling is important for either roller compacted concrete dams or layered rock foundations.

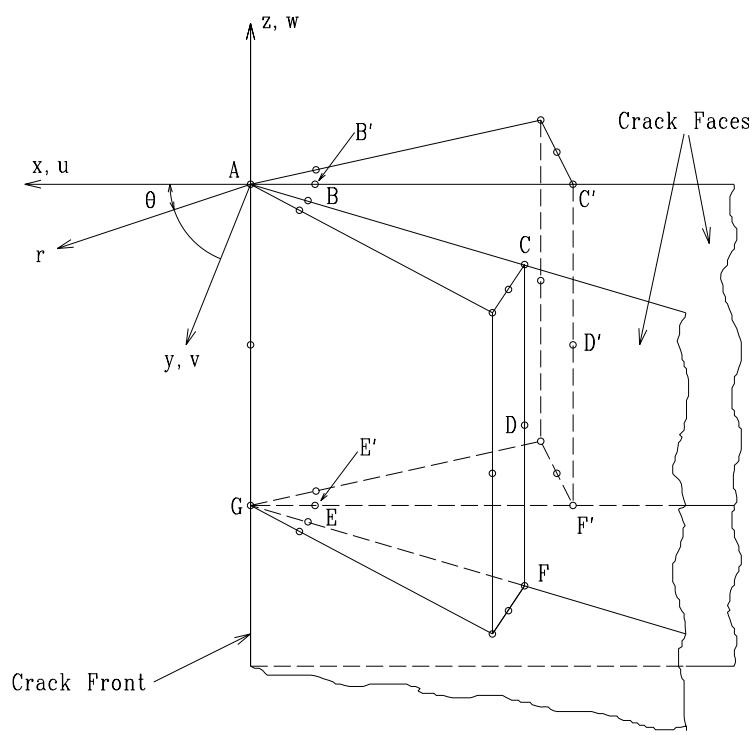


Figure 13.5: Nodal Definition for FE 3D SIF Determination

Chapter 14

RECIPROCAL WORK INTEGRALS

Theoretical background for evaluation of SIF in 2D using the S Integral in Merlin
 Chapter adapted from (Reich 1993)

14.1 General Formulation

In addition to conservation laws, a form of Betti's reciprocal work theorem (Sokolnikoff 1956) can also be exploited to directly compute stress intensity factors (Stern 1973). The reciprocal work theorem defines the relationship between two equilibrium states for a solid. For a solid free of body forces and initial strains and stresses the reciprocal work theorem is defined as

$$\oint_{\Gamma} t_i \tilde{u}_i d\Gamma = \oint_{\Gamma} \tilde{t}_i u_i d\Gamma \quad (14.1)$$

where Ω is any simply connected region within the solid and Γ is the contour of that region; u_i and t_i are the displacements and surface tractions, respectively, associated with one equilibrium state and \tilde{u}_i ; and \tilde{t}_i are the displacements and surface tractions, respectively, associated with another equilibrium state. The equilibrium state defined by u_i and t_i is called the primary state and the equilibrium state defined by \tilde{u}_i and \tilde{t}_i is called the complementary or auxiliary state.

To apply the reciprocal work theorem to a cracked solid the simply connected region Ω must be defined such that the singularity at the crack tip is avoided. This is accomplished by defining a pair of surfaces, Γ and Γ_ϵ , that begin on one crack surface and end on the other. Γ is an arbitrary surface defined in the counter-clockwise direction around the crack tip but far away from it. Γ_ϵ is a circle of radius ϵ centered on the crack tip that is defined in the clockwise direction around the crack tip completely inside Γ . Another pair of surfaces, Γ_t^+ and Γ_t^- , corresponding to the crack surfaces complete the definition of Γ , as is shown in Figure 14.1. Γ_t^+ is defined on the upper crack surface between Γ and Γ_ϵ and Γ_t^- is defined on the lower crack surface between Γ_ϵ and Γ . Naturally, Ω is the region inside this closed path through the solid. Since the material inside Γ_ϵ is not included in the definition of Ω the singularity at the crack tip has been excluded.

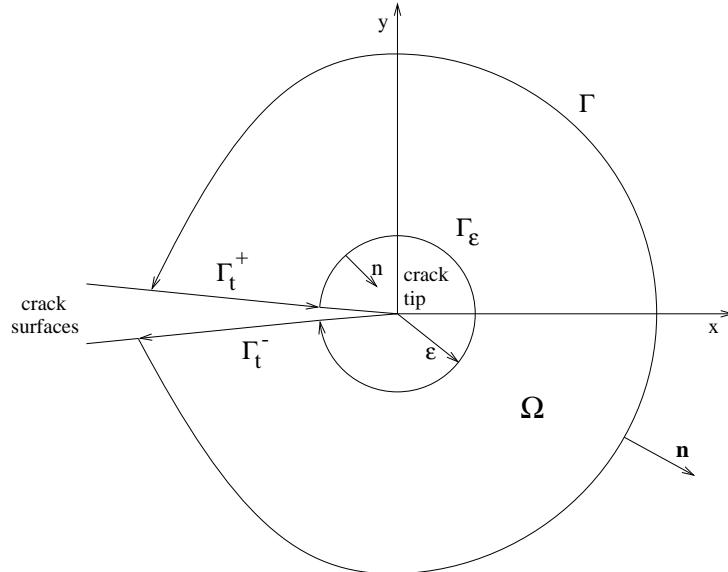


Figure 14.1: Contour integral paths around crack tip for reciprocal work integral

Assuming that Γ_t^+ and Γ_t^- are traction free the definition of the reciprocal work theorem can be rewritten as

$$\int_{\Gamma} t_i \tilde{u}_i d\Gamma + \int_{\Gamma_\epsilon} t_i \tilde{u}_i d\Gamma = \int_{\Gamma} \tilde{t}_i u_i d\Gamma + \int_{\Gamma_\epsilon} \tilde{t}_i u_i d\Gamma \quad (14.2)$$

in which the contributions from Γ and Γ_ϵ are clearly separated. This expanded expression is then rewritten in the form of Somigliana's identity to obtain

$$\int_{\Gamma} (t_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma + \int_{\Gamma_\epsilon} (t_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma = 0 \quad (14.3)$$

The displacements u_i and the stresses σ_{ij} for the primary state can be decomposed into

$$\begin{aligned} u_i &= u_i^s + u_i^e + u_i^0 \\ \sigma_{ij} &= \sigma_{ij}^s + \sigma_{ij}^e \end{aligned} \quad (14.4)$$

where u_i^s and σ_{ij}^s are the displacements and stresses, respectively, for the singular elastic state at the crack tip; u_i^e and σ_{ij}^e are the displacements and stresses, respectively, for the elastic state required to insure that boundary conditions on u_i and σ_{ij} are satisfied; and u_i^0 are the displacements of the crack tip.

Recognizing that the product $u_i^0 \tilde{t}_i$ has no contribution to the integral since the tractions \tilde{t}_i are self equilibrating due to the lack of body forces and taking into account the orders of the displacements and stresses in the various elastic states, (Stern 1973) determined that

$$\int_{\Gamma_\epsilon} (t_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma = \int_{\Gamma_\epsilon} (t_i^s \tilde{u}_i - \tilde{t}_i u_i^s) d\Gamma + o(1) \quad (14.5)$$

As ϵ is decreased the elastic singular state u_i^s and t_i^s becomes more dominant and the $o(1)$ terms can be ignored allowing the integrals over Γ_ϵ and Γ to be related in the following manner

$$I_\epsilon = \lim_{\epsilon \rightarrow 0} \int_{\Gamma_\epsilon} (t_i^s \tilde{u}_i - \tilde{t}_i u_i^s) d\Gamma = - \int_{\Gamma} (t_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma \quad (14.6)$$

Based on this relationship a singular elastic state u_i^s and t_i^s can be assumed; an auxiliary singular state \tilde{u}_i and \tilde{t}_i can be constructed from the assumed singular elastic state; and the value I_ϵ can be determined from the auxiliary singular state and far field displacements and tractions, u_i and t_i , computed using a suitable numerical method. Perhaps the most attractive feature of this approach is that the singularity at the crack tip need not be rigorously modeled in the numerical method used to obtain u_i and t_i .

Auxiliary singular states have been constructed for a crack in a homogeneous isotropic medium (Stern, Becker and Dunham 1976), a crack in a homogeneous orthotropic medium (Stern and M.L. 1975), and a crack on the interface between dissimilar isotropic media (Hong and Stern 1978). The procedure for constructing an auxiliary singular state will be outlined here using the homogeneous isotropic medium for this discussion. Once the singular elastic state has been assumed, the auxiliary singular state is constructed by taking λ as the negative of the value used in the singular elastic state, $\lambda = -\frac{1}{2}$ in this case. This of course means that the strain energy for the auxiliary singular state is unbounded at the crack tip, but since the integral is evaluated well away from the crack tip this is of no concern (Stern 1973). The value of the complex constant A for the auxiliary singular state is determined to be

$$A = \frac{2\mu}{(2\pi)^{\frac{1}{2}}(1+\kappa)} (c_1 + i c_2) \quad (14.7)$$

where c_1 and c_2 are arbitrary constants. This choice for A normalizes the integrand for Γ_ϵ involving the singular elastic state and the auxiliary singular state (Stern et al. 1976). Having determined A , the product of this integral is

$$I_\epsilon = c_1 K_I + c_2 K_{II} + o(1) \quad (14.8)$$

with the $o(1)$ term going to zero as ϵ is decreased. The stress intensity factors, K_I and K_{II} , can therefore be directly related to the integral over Γ

$$c_1 K_I + c_2 K_{II} = \int_{\Gamma} [(u_i - u_i^0) \tilde{t}_i + \tilde{u}_i t_i] d\Gamma \quad (14.9)$$

as was shown in Equation 14.6. When the integral is evaluated using u_i and t_i obtained from the numerical method the constants associated with the coefficients c_1 and c_2 are the stress intensity factors.

For the isotropic case, in the neighborhood of the crack tip, the displacements and the stresses, in polar coordinate system, are given by Westergaard as:

$$\begin{aligned} u_r - u_r^0 &= \frac{1}{4\mu} \left(\frac{r}{2\pi} \right)^{\frac{1}{2}} \left\{ \left[(2\kappa - 1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right] K_I \right. \\ &\quad \left. - \left[(2\kappa - 1) \sin \frac{\theta}{2} - 3 \sin \frac{3\theta}{2} \right] K_{II} \right\} + O(r^{\frac{1}{2}}) \end{aligned} \quad (14.10)$$

$$\begin{aligned} u_\theta - u_\theta^0 &= \frac{1}{4\mu} \left(\frac{r}{2\pi} \right)^{\frac{1}{2}} \left\{ \left[-(2\kappa + 1) \sin \frac{\theta}{2} + \sin \frac{3\theta}{2} \right] K_I \right. \\ &\quad \left. - \left[(2\kappa + 1) \cos \frac{\theta}{2} - 3 \cos \frac{3\theta}{2} \right] K_{II} \right\} + O(r^{\frac{1}{2}}) \end{aligned} \quad (14.11)$$

$$\sigma_r = \frac{1}{4(2\pi r)^{\frac{1}{2}}} \left\{ \left(5 \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right) K_I - \left(5 \sin \frac{\theta}{2} - 3 \sin \frac{3\theta}{2} \right) K_{II} \right\} + O(r^{-\frac{1}{2}}) \quad (14.12)$$

$$\sigma_\theta = \frac{1}{4(2\pi r)^{\frac{1}{2}}} \left\{ \left(3 \cos \frac{\theta}{2} + \cos \frac{3\theta}{2} \right) K_I - \left(3 \sin \frac{\theta}{2} + 3 \sin \frac{3\theta}{2} \right) K_{II} \right\} + O(r^{-\frac{1}{2}}) \quad (14.13)$$

$$\sigma_{r\theta} = \frac{1}{4(2\pi r)^{\frac{1}{2}}} \left\{ \left(\sin \frac{\theta}{2} + \sin \frac{3\theta}{2} \right) K_I + \left(\cos \frac{\theta}{2} + 3 \cos \frac{3\theta}{2} \right) K_{II} \right\} + O(r^{-\frac{1}{2}}) \quad (14.14)$$

where u_r^0 and u_θ^0 are the radial and tangential components, respectively, of the displacements u^0 of the crack tip, and

$$K_I = \lim_{r \rightarrow 0} (2\pi r)^{\frac{1}{2}} \sigma_\theta|_{\theta=0} \quad (14.15)$$

$$K_{II} = \lim_{r \rightarrow 0} (2\pi r)^{\frac{1}{2}} \sigma_{r\theta}|_{\theta=0} \quad (14.16)$$

are the usual stress intensity factors.

The auxiliary solution to be used in the reciprocal work relation is based on Williams solution (Stern et al. 1976):

$$\begin{aligned} \tilde{u}_r &= \frac{1}{2(2\pi r)^{\frac{1}{2}}(1+\kappa)} \\ &\quad \left\{ \left[(2\kappa + 1) \cos \frac{3\theta}{2} - 3 \cos \frac{\theta}{2} \right] c_1 + \left[(2\kappa + 1) \sin \frac{3\theta}{2} - \sin \frac{\theta}{2} \right] c_2 \right\} \end{aligned} \quad (14.17)$$

$$\begin{aligned} \tilde{u}_\theta &= \frac{1}{2(2\pi r)^{\frac{1}{2}}(1+\kappa)} \\ &\quad \left\{ \left[-(2\kappa - 1) \sin \frac{3\theta}{2} + 3 \sin \frac{\theta}{2} \right] c_1 + \left[(2\kappa - 1) \cos \frac{3\theta}{2} - \cos \frac{\theta}{2} \right] c_2 \right\} \end{aligned} \quad (14.18)$$

$$\tilde{\sigma}_r = -\frac{\mu}{2(2\pi r^3)^{\frac{1}{2}}(1+\kappa)} \left\{ \left[7 \cos \frac{3\theta}{2} - 3 \cos \frac{\theta}{2} \right] c_1 + \left[7 \sin \frac{3\theta}{2} - \sin \frac{\theta}{2} \right] c_2 \right\} \quad (14.19)$$

$$\tilde{\sigma}_\theta = -\frac{\mu}{2(2\pi r^3)^{\frac{1}{2}}(1+\kappa)} \left\{ \left[\cos \frac{3\theta}{2} + 3 \cos \frac{\theta}{2} \right] c_1 + \left[\sin \frac{3\theta}{2} + \sin \frac{\theta}{2} \right] c_2 \right\} \quad (14.20)$$

$$\begin{aligned} \tilde{\sigma}_{r\theta} &= -\frac{\mu}{2(2\pi r^3)^{\frac{1}{2}}(1+\kappa)} \left\{ 3 \left[\sin \frac{3\theta}{2} + \sin \frac{\theta}{2} \right] c_1 - \left[3 \cos \frac{3\theta}{2} - \cos \frac{\theta}{2} \right] c_2 \right\} \end{aligned} \quad (14.21)$$

where c_1 and c_2 are arbitrary constants. Now, on the inner circular boundary, the evaluation of the contour integral in terms of traction and displacement takes the form:

$$\begin{aligned} I_\epsilon &= - \int_{C_\epsilon} ((\mathbf{u} - \mathbf{u}^0) \cdot \tilde{\mathbf{t}}) - \tilde{\mathbf{u}} \cdot \mathbf{t} ds \\ &= \int_{-\pi}^{\pi} [\tilde{\sigma}_r(u_r - u_r^0) \tilde{\sigma}_{r\theta}(u_\theta - u_\theta^0) - \sigma_r \tilde{u}_r + \sigma_{r\theta} \tilde{u}_\theta] r d\theta \end{aligned} \quad (14.22)$$

When the two solutions are substituted into the preceding equation, we obtain:

$$I_\epsilon = c_1 K_I - c_2 K_{II} \quad (14.23)$$

Thus it can be readily seen that Eq. ?? now reduces to:

$$\boxed{c_1 K_I - c_2 K_{II} = \int_C [(\mathbf{u} - \mathbf{u}^0) \cdot \tilde{\mathbf{t}} - \tilde{\mathbf{u}} \cdot \mathbf{t}] ds} \quad (14.24)$$

From this equation an algorithm for the SIF determination emerges:

1. Perform a linear elastic finite element analysis.
2. Extract \mathbf{u} and \mathbf{t} (displacements and traction) from the analysis.
3. Substitute into Eq. 14.24 along with the auxiliary solution.
4. The components of c_1 in Eq. 14.24 yield K_I .
5. The components of c_2 in Eq. 14.24 yield K_{II} .

In addition to demonstrating the reciprocal work integral for cracks in homogeneous isotropic (Stern et al. 1976), homogeneous orthotropic (Stern and M.L. 1975), and on the interface between dissimilar isotropic materials (Hong and Stern 1978), Stern also proposed extensions for treating body forces (Stern et al. 1976) and thermal strains (Stern 1979). Unfortunately, the description of the extension for body forces was rather superficial, being limited to a footnote, and no example problems were presented for either of these extensions. However, for the case of thermal strains it was clearly shown that there is no need to consider thermal loading in the auxiliary state, meaning that the reciprocal work integral can also be extended to include initial stresses without modifying the auxiliary solution. More recent developments include the treatment of dynamic crack propagation (Atkinson, Bastero and Miranda 1986, Bastero, Atkinson and Martinez-Esnaola 1989), sharp notches (Atkinson, Bastero and Martinez-Esnaola 1988, Atkinson and Bastero 1991), and cracks in coupled poro-elastic media (Atkinson and Craster 1992).

14.2 Volume Form of the Reciprocal Work Integral

The first step to be taken when formulating extensions to the reciprocal work integral is the definition of the reciprocal work theorem accounting for the applied loads in the two equilibrium states. Unfortunately, it is not always obvious how the reciprocal work theorem should be defined to account for the applied loads, particularly when they are the result of initial strains or stresses. It will be shown here that the line integrals in the reciprocal work theorem can be converted to volume integrals using Green's theorem and that the form of the integrand for the volume integrals is such that the appropriate form of the reciprocal work theorem can be determined quite simply.

(Sokolnikoff 1956) defined the reciprocal work theorem relating two separate equilibrium states for a solid, both including body forces, as

$$\int_{\Gamma} t_i \tilde{u}_i d\Gamma + \int_{\Omega} b_i \tilde{u}_i d\Omega = \int_{\Gamma} \tilde{t}_i u_i d\Gamma + \int_{\Omega} \tilde{b}_i u_i d\Omega \quad (14.25)$$

where u_i , t_i , and b_i are the displacements, surface tractions, and body forces, respectively, for one equilibrium state; \tilde{u}_i , \tilde{t}_i , and \tilde{b}_i are the displacements, surface tractions, and body forces, respectively, for the other equilibrium state; Ω corresponds to the volume of the solid; and Γ corresponds to the entire surface of the solid. The equilibrium state defined by u_i , t_i , and b_i is referred to as the primary state and the equilibrium state defined by \tilde{u}_i , \tilde{t}_i , and \tilde{b}_i is referred to as the auxiliary state. Recalling from the equilibrium equation that $b_i = -\sigma_{ij,j}$ and $\tilde{b}_i = -\tilde{\sigma}_{ij,j}$ the reciprocal work theorem can be rewritten as

$$\int_{\Gamma} t_i \tilde{u}_i d\Gamma - \int_{\Omega} \sigma_{ij,j} \tilde{u}_i d\Omega = \int_{\Gamma} \tilde{t}_i u_i d\Gamma - \int_{\Omega} \tilde{\sigma}_{ij,j} u_i d\Omega \quad (14.26)$$

where σ_{ij} and $\tilde{\sigma}_{ij}$ are the stress tensors for the two equilibrium states. Adopting a counter-clockwise path around Γ the expressions relating $d\Gamma$ to dx_1 and dx_2 given in Equation ?? are still valid, allowing the line integrals to be written in a form compatible with Green's theorem

$$\begin{aligned} \int_{\Gamma} t_i \tilde{u}_i d\Gamma &= \int_{\Gamma} (-\sigma_{i2} \tilde{u}_i dx_1 + \sigma_{i1} \tilde{u}_i dx_2) \\ \int_{\Gamma} \tilde{t}_i u_i d\Gamma &= \int_{\Gamma} (-\tilde{\sigma}_{i2} u_i dx_1 + \tilde{\sigma}_{i1} u_i dx_2) \end{aligned} \quad (14.27)$$

by expanding t_i and \tilde{t}_i in terms of σ_{ij} , $\tilde{\sigma}_{ij}$, and n_i and collecting terms. Applying Green's theorem (Kreyszig 1979) to convert the line integrals to volume integrals yields

$$\begin{aligned} \int_{\Gamma} t_i \tilde{u}_i d\Gamma &= \int_{\Omega} \sigma_{ij,j} \tilde{u}_i d\Omega + \int_{\Omega} \sigma_{ij} \tilde{u}_{i,j} d\Omega \\ \int_{\Gamma} \tilde{t}_i u_i d\Gamma &= \int_{\Omega} \tilde{\sigma}_{ij,j} u_i d\Omega + \int_{\Omega} \tilde{\sigma}_{ij} u_{i,j} d\Omega \end{aligned} \quad (14.28)$$

and the reciprocal work theorem clearly simplifies to

$$\int_{\Omega} \sigma_{ij} \tilde{u}_{i,j} d\Omega = \int_{\Omega} \tilde{\sigma}_{ij} u_{i,j} d\Omega \quad (14.29)$$

In the absence of body forces the volume integrals are not included in the definition of the reciprocal work theorem and the expression shown above is still valid since $\sigma_{ij,j} = 0$ and $\tilde{\sigma}_{ij,j} = 0$. Knowing that the line integral form of the reciprocal work theorem can be rewritten in the volume integral form shown in Equation 14.29, the appropriate definition of the reciprocal work theorem to account for initial strains and stresses in the primary state can be obtained quite easily. This is accomplished by simply writing the volume form of the reciprocal work theorem such that there is a direct relationship between the stresses and displacements in the primary state.

14.3 Surface Tractions on Crack Surfaces

The extension to the reciprocal work integral to include the effect of surface tractions on the crack surfaces in the primary state parallels the approach proposed by (Karlsson and Bäcklund 1978) for the J integral. For a primary state free of body forces with surface tractions on the crack surfaces the reciprocal work theorem is defined as

$$\int_{\Gamma} t_i \tilde{u}_i d\Gamma = \int_{\Gamma} \tilde{t}_i u_i d\Gamma \quad (14.30)$$

This expression can be rewritten such that a separate integral is given for each portion of the contour path

$$\int_{\Gamma} t_i \tilde{u}_i d\Gamma + \int_{\Gamma_{\epsilon}} t_i \tilde{u}_i d\Gamma + \int_{\Gamma_t} \hat{t}_i \tilde{u}_i d\Gamma = \int_{\Gamma} \tilde{t}_i u_i d\Gamma + \int_{\Gamma_{\epsilon}} \tilde{t}_i u_i d\Gamma \quad (14.31)$$

where $\Gamma_t = \Gamma_t^+ \cup \Gamma_t^-$ and \hat{t}_i is the applied surface traction vector on the crack surfaces in the primary state. This expression for the reciprocal work theorem can be rewritten in the form of Somigliana's identity as

$$\int_{\Gamma} (t_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma + \int_{\Gamma_t} \hat{t}_i \tilde{u}_i d\Gamma + \int_{\Gamma_{\epsilon}} (t_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma = 0 \quad (14.32)$$

Clearly, the integrand of the integral over Γ_{ϵ} is identical to that for the case of a primary state free of surface tractions on the crack surfaces, which means that Equation 14.5 still holds and the solution for the auxiliary singular state is still valid. The value I_{ϵ} is then defined as

$$I_{\epsilon} = - \int_{\Gamma} (t_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma - \lim_{\epsilon \rightarrow 0} \int_{\Gamma_t} \hat{t}_i \tilde{u}_i d\Gamma \quad (14.33)$$

Provided that \hat{t}_i is not expressed in powers of r less than $-\frac{1}{2}$, the limit exists and the stress intensity factors are defined as

$$c_1 K_I + c_2 K_{II} = \int_{\Gamma} [\tilde{t}_i (u_i - u_i^0) - t_i \tilde{u}_i] d\Gamma - \int_{\Gamma_t} \hat{t}_i \tilde{u}_i d\Gamma \quad (14.34)$$

where u_i^0 are the displacements of the crack tip. However, when the integral over Γ_t is evaluated using numerical integration techniques, quadratures based on sampling points that coincide with the nodal locations should be avoided since \tilde{u}_i is singular at the crack tip.

14.4 Body Forces

For a primary state with body forces but free surface tractions on the crack surfaces and initial strains and stresses the reciprocal work theorem is defined as

$$\int_{\Gamma} t_i \tilde{u}_i d\Gamma + \int_{\Omega} b_i \tilde{u}_i d\Omega = \int_{\Gamma} \tilde{t}_i u_i d\Gamma \quad (14.35)$$

where b_i is the body force vector. It should be noted that since the line integrals are defined over Γ this form of the reciprocal work integral could also account for surface tractions on the crack surfaces. The expression for the reciprocal work theorem can be rewritten such that a separate integral is given for each portion of the contour path

$$\int_{\Gamma} t_i \tilde{u}_i d\Gamma + \int_{\Gamma_{\epsilon}} t_i \tilde{u}_i d\Gamma + \int_{\Omega} b_i \tilde{u}_i d\Omega = \int_{\Gamma} \tilde{t}_i u_i d\Gamma + \int_{\Gamma_{\epsilon}} \tilde{t}_i u_i d\Gamma \quad (14.36)$$

This expression can be rewritten in the form of Somigliana's identity as

$$\int_{\Gamma} (t_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma + \int_{\Omega} b_i \tilde{u}_i d\Omega + \int_{\Gamma_{\epsilon}} (t_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma = 0 \quad (14.37)$$

Clearly, the integrand of the integral over Γ_{ϵ} is identical to that for the case of a primary state free of body forces, which means that Equation 14.5 still holds and the solution for the auxiliary singular state described is still valid. The value I_{ϵ} is then defined as

$$I_{\epsilon} = - \int_{\Gamma} (t_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma - \lim_{\epsilon \rightarrow 0} \int_{\Omega} b_i \tilde{u}_i d\Omega \quad (14.38)$$

Provided that b_i is not expressed in powers of r less than $-\frac{1}{2}$, the limit exists and the stress intensity factors are defined as

$$c_1 K_I + c_2 K_{II} = \int_{\Gamma} [\tilde{t}_i (u_i - u_i^0) - t_i \tilde{u}_i] d\Gamma - \int_{\Omega} b_i \tilde{u}_i d\Omega \quad (14.39)$$

where u_i^0 are the displacements of the crack tip. However, when the integral over Ω is evaluated using numerical integration techniques, quadratures based on sampling points that coincide with the nodal locations should be avoided since \tilde{u}_i is singular at the crack tip.

14.5 Initial Strains Corresponding to Thermal Loading

For problems in thermo-elasticity the constitutive law defines net effective stresses $\bar{\sigma}'_{ij}$ in terms of the total strains ε_{ij} and the thermal strains ε_{ij}^0 , as is shown in Equation 14.53. $\bar{\sigma}'_{ij}$ can be decomposed into effective stresses σ'_{ij} and thermal stresses σ''_{ij} . σ'_{ij} are the result of ε_{ij} , which are, in turn, defined by the displacements u_i . Therefore, the effective stresses σ'_{ij} are then directly related to the displacements u_i and should be used in the definition of the reciprocal work theorem rather than the net effective stresses $\bar{\sigma}'_{ij}$. The volume form of the reciprocal work theorem for a primary state that includes thermal strains is

$$\int_{\Omega} \sigma'_{ij} \tilde{u}_{i,j} d\Omega = \int_{\Omega} \tilde{\sigma}_{ij} u_{i,j} d\Omega \quad (14.40)$$

The relationship between the line and volume integral forms of the reciprocal work theorem can be readily obtained by applying Green's theorem to the volume integral with σ'_{ij} in the integrand

$$\int_{\Gamma} t'_i \tilde{u}_i d\Gamma = \int_{\Omega} \sigma'_{ij,j} \tilde{u}_i d\Omega + \int_{\Omega} \sigma'_{ij} \tilde{u}_{i,j} d\Omega \quad (14.41)$$

where $t'_i = \sigma'_{ij} n_j$ is the effective surface traction vector. Recalling from the equilibrium equation that $\sigma'_{ij,j} = \alpha C_{ijkl} \mathcal{T}_{j,k} \delta_{kl}$ in the absence of body forces, it is clearly evident that a volume integral is required to complete the definition of the reciprocal work theorem. Therefore, the appropriate form of the reciprocal work theorem for a primary state with thermal strains but no body forces is

$$\int_{\Gamma} t'_i \tilde{u}_i d\Gamma - \int_{\Omega} \alpha (C_{ijkl} \mathcal{T}_{j,k} \delta_{kl}) \tilde{u}_i d\Omega = \int_{\Gamma} \tilde{t}_i u_i d\Gamma \quad (14.42)$$

where $\mathcal{T}_{j,k}$ is the gradient of the temperatures. A more general form of the reciprocal work theorem would be

$$\int_{\Gamma} t'_i \tilde{u}_i d\Gamma + \int_{\Omega} b'_i \tilde{u}_i d\Omega = \int_{\Gamma} \tilde{t}_i u_i d\Gamma \quad (14.43)$$

where b'_i is the effective body force vector, as defined in Equation 14.58, which in this particular case does not include a true body force vector b_i .

Recalling that the natural boundary conditions are defined in terms of the total stresses, $t'_i \neq 0$ on the crack surfaces. Therefore, the form of the reciprocal work theorem in which the line integrals on Γ have been separated is

$$\int_{\Gamma} t'_i \tilde{u}_i d\Gamma + \int_{\Gamma_{\epsilon}} t'_i \tilde{u}_i d\Gamma + \int_{\Gamma_t} \hat{t}'_i \tilde{u}_i d\Gamma + \int_{\Omega} b'_i \tilde{u}_i d\Omega = \int_{\Gamma} \tilde{t}_i u_i d\Gamma + \int_{\Gamma_{\epsilon}} \tilde{t}_i u_i d\Gamma \quad (14.44)$$

where \hat{t}'_i is the applied effective surface traction vector, as defined in Equation 14.60, which, much like the effective body force vector b'_i , does not include a true applied surface traction vector in this case. This expression for the reciprocal work theorem can be rewritten in the form of Somigliana's identity as

$$\int_{\Gamma} (t'_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma + \int_{\Gamma_t} \hat{t}'_i \tilde{u}_i d\Gamma + \int_{\Omega} b'_i \tilde{u}_i d\Omega + \int_{\Gamma_{\epsilon}} (t'_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma = 0 \quad (14.45)$$

Clearly, the integrand of the integral over Γ_ϵ is identical to that for the case of a primary state free of initial strains, which means that Equation 14.5 still holds and the solution for the auxiliary singular state described in Section ?? is still valid. The value I_ϵ is then defined as

$$I_\epsilon = - \int_{\Gamma} (t'_i \tilde{u}_i - \tilde{t}_i u_i) d\Gamma - \lim_{\epsilon \rightarrow 0} \int_{\Gamma_t} \tilde{t}'_i \tilde{u}_i d\Gamma - \lim_{\epsilon \rightarrow 0} \int_{\Omega} b'_i \tilde{u}_i d\Omega \quad (14.46)$$

Provided that the temperature \mathcal{T} is not expressed in powers of r less than $\frac{1}{2}$, the limit exists and the stress intensity factors are defined as

$$c_1 K_I + c_2 K_{II} = \int_{\Gamma} [\tilde{t}_i (u_i - u_i^0) - t'_i \tilde{u}_i] d\Gamma - \int_{\Gamma_t} \tilde{t}'_i \tilde{u}_i d\Gamma - \int_{\Omega} b'_i \tilde{u}_i d\Omega \quad (14.47)$$

where u_i^0 are the displacements of the crack tip. However, when the integrals over Γ_t and Ω are evaluated using numerical integration techniques, quadratures based on sampling points that coincide with the nodal locations should be avoided since \tilde{u}_i is singular at the crack tip.

14.6 Initial Stresses Corresponding to Pore Pressures

The stress-strain relationship for poro-elasticity, which is obtained by substituting the constitutive law defining the effective stresses σ'_{ij} into the principle of effective stress, defines the total stresses σ_{ij} in terms of the total strains ε_{ij} and the pore pressures p , as was shown in Equation 14.62. As was the case for problems in thermo-elasticity, σ_{ij} can again be decomposed, but in this instance the constituent stresses are σ'_{ij} and the initial stresses σ_{ij}^0 corresponding to the pore pressures. Since ε_{ij} is defined in terms of the displacements u_i , the effective stresses σ'_{ij} are then directly related to the displacements u_i and the reciprocal work theorem is again defined in terms of the effective stresses in the primary state. Therefore, Equations 14.40 and 14.41 also apply when the primary state includes initial stresses. Recalling from the equilibrium equation that $\sigma'_{ij,j} = p_{,j} \delta_{ij}$ in the absence of body forces, it is clearly evident that a volume integral is required to complete the definition of the reciprocal work theorem. Therefore, the appropriate form of the reciprocal work theorem for a primary state with pore pressures but no body forces is

$$\int_{\Gamma} t'_i \tilde{u}_i d\Gamma - \int_{\Omega} p_{,i} \tilde{u}_i d\Omega = \int_{\Gamma} \tilde{t}_i u_i d\Gamma \quad (14.48)$$

where $p_{,i}$ is the gradient of the pore pressures. A more general form of the reciprocal work integral is

$$\int_{\Gamma} t'_i \tilde{u}_i d\Gamma + \int_{\Omega} b'_i \tilde{u}_i d\Omega = \int_{\Gamma} \tilde{t}_i u_i d\Gamma \quad (14.49)$$

where b'_i is the effective body force vector, as defined in Equation 14.64, which does not include a true body force vector b_i in this case. Recognizing that the general form of the reciprocal work theorem accounting for initial stresses is identical to that accounting for initial strains (i.e. Equation 14.43), Equations 14.44 through 14.47 apply for initial stresses as well. However, the applied effective surface traction vector is defined by Equation 14.65 and the pore pressure p rather than the temperature \mathcal{T} must be expressed in a power of r greater than $\frac{1}{2}$ in order for the limits in Equation 14.46 to exist. Naturally, the restrictions on the choice of numerical integration techniques are also still in effect.

14.7 Combined Thermal Strains and Pore Pressures

Recalling that in the absence of initial strains and stresses that the total stresses σ_{ij} and the effective stresses σ'_{ij} are equivalent, it is quite clear that Equation 14.43 also defines the reciprocal work theorem for solids that are free of initial strains and stresses. Due to the general definitions of the applied effective surface traction vector \tilde{t}'_i and the effective body force vector b'_i , the cases of a primary state with true surface tractions \hat{t}_i on the crack surfaces and true body forces b_i are also addressed by Equation 14.43. Therefore, the stress intensity factors for a primary state which includes any combination of surface tractions on the crack surfaces, body forces, and initial strains and stresses are defined by Equation 14.47. The relationship between the stress intensity factors and the reciprocal work theorem is obtained by substituting the expressions for \tilde{t}'_i and b'_i defined by Equation 14.69 into Equation 14.47

$$\begin{aligned} c_1 K_I + c_2 K_{II} &= \int_{\Gamma} [\tilde{t}_i (u_i - u_i^0) - (t_i + p n_i + \alpha \mathcal{T} C_{ijkl} n_i \delta_{kl}) \tilde{u}_i] d\Gamma \\ &- \int_{\Gamma_t} (\hat{t}_i + p n_i + \alpha \mathcal{T} C_{ijkl} n_i \delta_{kl}) \tilde{u}_i d\Gamma \end{aligned}$$

$$- \int_{\Omega} (b_i - p_{,i} - \alpha C_{ijkl} \mathcal{T}_{,l} \delta_{kl}) \tilde{u}_i d\Omega \quad (14.50)$$

Naturally, the restrictions imposed on the power of r for \hat{t}_i , b_i , \mathcal{T} , and p are still in effect, as are the restrictions on the choice of numerical integration techniques.

14.8 Field Equations for Thermo- and Poro-Elasticity

In thermo or poro-elasticity the thermal strains and pore pressures are usually treated as initial strains and initial stresses, respectively. The general stress-strain relationship obtained by substituting the constitutive law into the effective stress principle is

$$\sigma_{ij} = C_{ijkl} (\varepsilon_{kl} - \varepsilon_{kl}^0) + \sigma_{ij}^0 \quad (14.51)$$

Clearly, in the absence of initial stresses $\sigma_{ij} = \bar{\sigma}'_{ij}$ and in the absence of both initial strains and stresses $\sigma_{ij} = \sigma'_{ij}$.

The thermal strains for an isotropic material are defined in terms of the temperature \mathcal{T} and the coefficient of thermal expansion α as

$$\varepsilon_{ij}^0 = \alpha \mathcal{T} \delta_{ij} \quad (14.52)$$

where δ_{ij} is the Kronecker delta. Substituting this expression for the thermal strains into Equation 14.51, the resulting constitutive law for thermo-elasticity is

$$\bar{\sigma}'_{ij} = C_{ijkl} (\varepsilon_{kl} - \alpha \mathcal{T} \delta_{kl}) \quad (14.53)$$

The thermal stresses σ''_{ij} are defined as

$$\sigma''_{ij} = \alpha \mathcal{T} C_{ijkl} \delta_{kl} \quad (14.54)$$

and the net strains \bar{E}_{ij} are defined as

$$\bar{E}_{ij} = \varepsilon_{kl} - \alpha \mathcal{T} \delta_{kl} \quad (14.55)$$

Adopting the standard form of the effective stress principle the equilibrium equation and natural boundary conditions, respectively, can be rewritten in terms of the effective stresses

$$\begin{aligned} \sigma'_{ij,j} + b'_i &= 0 \\ \sigma'_{ij} n_j - \hat{t}'_i &= 0 \end{aligned} \quad (14.56)$$

where b'_i is the effective body force vector and \hat{t}'_i is the applied effective surface traction vector. The effective body force vector b'_i is defined as

$$b'_i = b_i - \sigma''_{ij,j} \quad (14.57)$$

and may be rewritten in terms of the temperature gradient vector $\mathcal{T}_{,i}$

$$b'_i = b_i - \alpha C_{ijkl} \mathcal{T}_{,i} \delta_{kl} \quad (14.58)$$

based on the definition of the thermal stresses given in Equation 14.54 and the assumption of a homogeneous material. The applied effective surface traction vector \hat{t}'_i is defined as

$$\hat{t}'_i = \hat{t}_i + \sigma''_{ij} n_j \quad (14.59)$$

and may be rewritten in terms of the temperature \mathcal{T}

$$\hat{t}'_i = \hat{t}_i + \alpha \mathcal{T} C_{ijkl} n_i \delta_{kl} \quad (14.60)$$

based on the definition of the thermal stresses given in Equation 14.54.

Pore pressures are typically defined using the sign convention for soil mechanics in which compression is positive, but in the sign convention for standard solid mechanics tension is considered to be positive. Therefore, the initial stresses corresponding to a pore pressure are defined as

$$\sigma_{ij}^0 = -p \delta_{ij} \quad (14.61)$$

where p is the pore pressure defined using the compression positive sign convention; the minus sign corrects the discrepancy in the sign conventions; and δ_{ij} is the Kronecker delta. In the classical interpretation of the behavior of a

porous material (Terzaghi and Peck 1967), the pore pressures p act only in the voids of the material and the effective stresses act only on the skeleton of the material. It must be noted that the pore pressures p being considered in this discussion and throughout the remainder of this chapter are the steady state pore pressures; excess pore pressures resulting from dilatant behavior in the skeleton of the material are not considered. The stress-strain relationship for poro-elasticity

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl} - p \delta_{ij} \quad (14.62)$$

is obtained by substituting the expression for the initial stresses into Equation 14.51. Adopting the standard form of the principle of effective stress the equilibrium equation and the natural boundary conditions, respectively, can be rewritten in terms of the effective stresses

$$\begin{aligned} \sigma'_{ij,j} + b'_i &= 0 \\ \sigma'_{ij} n_j - \hat{t}'_i &= 0 \end{aligned} \quad (14.63)$$

where b'_i is the effective body force vector and \hat{t}'_i is the applied effective surface traction vector. The effective body force vector b'_i is defined as

$$b'_i = b_i - p_{,i} \quad (14.64)$$

where $p_{,i}$ is the pore pressure gradient vector. The applied effective surface traction vector \hat{t}'_i is defined as

$$\hat{t}'_i = \hat{t}_i + p n_i \quad (14.65)$$

Since $\sigma'_{ij} = 0$ on surfaces exposed to hydrostatic pressures but no other surface tractions, $\hat{t}_i = -p n_i$ on these surfaces.

When thermal strains and pore pressures are considered in combination the constitutive law is defined as a simple combination of Equations 14.53 and 14.62

$$\sigma_{ij} = C_{ijkl} (\varepsilon_{kl} - \alpha \mathcal{T} \delta_{kl}) - p \delta_{ij} \quad (14.66)$$

The equilibrium equation and natural boundary conditions, respectively, can be rewritten in terms of either the effective stresses σ'_{ij}

$$\begin{aligned} \sigma'_{ij,j} + b'_i &= 0 \\ \sigma'_{ij} n_j - \hat{t}'_i &= 0 \end{aligned} \quad (14.67)$$

or the net effective stresses $\bar{\sigma}'_{ij}$

$$\begin{aligned} \bar{\sigma}'_{ij,j} + b'_i &= 0 \\ \bar{\sigma}'_{ij} n_j - \hat{t}'_i &= 0 \end{aligned} \quad (14.68)$$

The field equations defined in terms of σ'_{ij} are obtained by adopting the standard form of the principle effective stress and the field equations defined in terms of $\bar{\sigma}'_{ij}$ are obtained by adopting the alternate form of the principle effective stress.

When the equilibrium equation and natural boundary conditions are written in terms of σ'_{ij} the effective body forces b'_i and applied effective surface tractions \hat{t}'_i are defined as

$$\begin{aligned} b'_i &= b_i - p_{,i} - \alpha C_{ijkl} \mathcal{T}_{,i} \delta_{kl} \\ \hat{t}'_i &= \hat{t}_i + p n_i + \alpha \mathcal{T} C_{ijkl} n_i \delta_{kl} \end{aligned} \quad (14.69)$$

However, when the equilibrium equation and natural boundary conditions are written in terms of $\bar{\sigma}'_{ij}$ the effective body forces b'_i and applied effective surface tractions \hat{t}'_i are identical to those for poro-elasticity (i.e. Equations 14.64 and 14.65, respectively).

Chapter 15

J INTEGRAL BASED METHODS

Theoretical background for *J* integral evaluation, and 3D SIF calculations in MERLIN

15.1 Numerical Evaluation

Within linear elastic fracture mechanics, the *J* integral is equivalent to *G* and we have:

$$G = J = -\frac{\partial \Pi}{\partial a} = \int_r (wdy - \mathbf{t} \cdot \frac{\partial \mathbf{d}}{\partial x} ds) \quad (15.1)$$

Thus it is evident that we do have two methods of evaluating *J*: the first one stems from its equivalence to the energy released rate, and the second one from its definition as an integral along a closed contour. Evaluation of *J* according to the first approach is identical to the one of *G* and has been previously presented.

In this chapter we shall present the algorithm to evaluate *J* on the basis of its contour line integral definition. Whereas derivation will be for *J* integral only, its extension to *J_i* is quite straightforward.

If the stresses were to be determined at the nodes, than the numerical evaluation of *J* will be relatively simple. However, most standard finite element codes only provide Gauss point stresses, and hence care must be exercised in properly determining the *J* integral along a path passing through them.

The algorithm for the *J* calculation closely follows the method presented in (Owen and Fawkes 1983), and is as follows:

1. First let us restrict ourselves to the more general case in which isoparametric elements are used. Because the stresses are most accurately evaluated at the gauss points, the path can be conveniently chosen to coincide with $\xi = \xi_{cst}$ and/or $\eta = \eta_{cst}$. For the sake of discussion, let us assume that the element connectivity is such that the path is along $\xi = \xi_{cst}$, as in Fig. 15.1. We note that for corner elements the integration will have to be performed twice along the two directions.

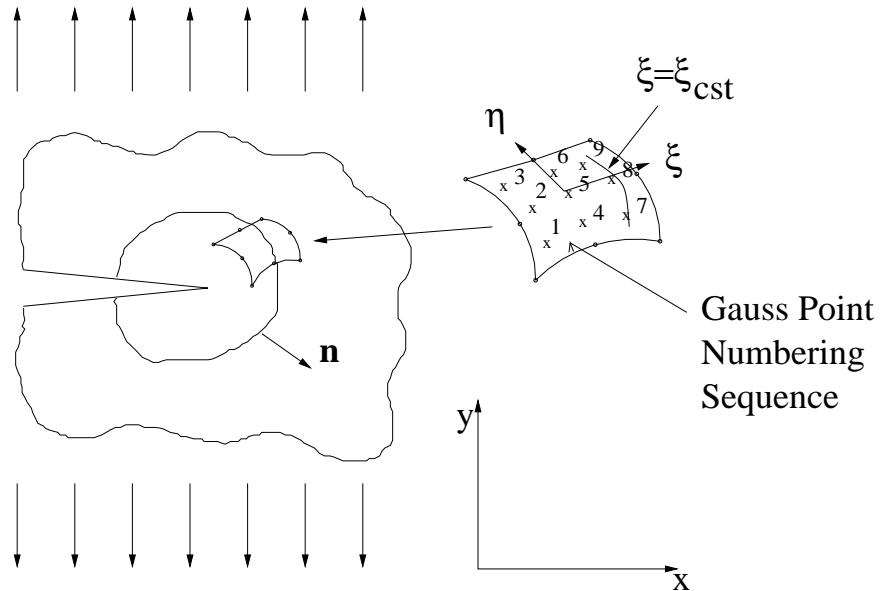


Figure 15.1: Numerical Extraction of the *J* Integral (Owen and Fawkes 1983)

2. Now let us start from the basic definition of *J*:

$$J = \int_{\Gamma} wdy - \mathbf{t} \cdot \frac{\partial \mathbf{d}}{\partial x} ds \quad (15.2)$$

where \mathbf{t} is the traction vector along \mathbf{n} , which is normal to the path; \mathbf{d} is the displacement vector; ds is the element of arc along path Γ ; and w is the strain energy density. We note that the crack is assumed to be along the x axis. If it is not, stresses and displacements would first have to be rotated. Let us now determine each term of Eq. 15.2.

3. The traction vector is given by:

$$t_i = \sigma_{ij} n_j \Rightarrow \mathbf{t} = \begin{Bmatrix} \sigma_x n_1 + \tau_{xy} n_2 \\ \tau_{xy} n_1 + \sigma_y n_2 \end{Bmatrix} \quad (15.3)$$

4. The displacement vector is:

$$\mathbf{d} = \begin{Bmatrix} u \\ v \end{Bmatrix} \quad (15.4)$$

5. The strain energy density w is:

$$\begin{aligned} w &= \frac{1}{2}(\sigma_x \varepsilon_x + 2\tau_{xy} \gamma_{xy} + \sigma_y \varepsilon_y) \\ &= \frac{1}{2}[\sigma_x \frac{\partial u}{\partial x} + \tau_{xy}(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) + \sigma_y \frac{\partial v}{\partial y}] \end{aligned} \quad (15.5)$$

6. The arc length ds and dy are given by:

$$ds = \sqrt{dx^2 + dy^2} = \sqrt{\left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2} d\eta \quad (15.6)$$

$$dy = \frac{\partial y}{\partial \eta} d\eta \quad (15.7)$$

7. Next we can evaluate part of the second term of J :

$$\mathbf{t} \cdot \frac{\partial \mathbf{d}}{\partial x} = (\sigma_x n_1 + \tau_{xy} n_2) \frac{\partial u}{\partial x} + (\tau_{xy} n_1 + \sigma_y n_2) \frac{\partial v}{\partial x} \quad (15.8)$$

where n_1 and n_2 are the components of \mathbf{n} , which is a unit vector normal to the contour line at the Gauss point under consideration.

8. Having defined all the terms of J , we substitute in Eq. 15.2 to obtain the contribution to J from a particular Gauss point within an element.

$$\begin{aligned} J^e &= \int_{-1}^1 \left\{ \underbrace{\frac{1}{2} \left[\sigma_x \frac{\partial u}{\partial x} + \tau_{xy} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \sigma_y \frac{\partial v}{\partial y} \right]}_{w} \underbrace{\frac{\partial y}{\partial \eta}}_{dy} \right. \\ &\quad \left. - \underbrace{\left[(\sigma_x n_1 + \tau_{xy} n_2) \frac{\partial u}{\partial x} + (\tau_{xy} n_1 + \sigma_y n_2) \frac{\partial v}{\partial x} \right]}_{\mathbf{t} \cdot \frac{\partial \mathbf{d}}{\partial x}} \right. \\ &\quad \left. \underbrace{\sqrt{\left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2}}_{ds} d\eta \right\} \\ &= \int_{-1}^1 I d\eta \end{aligned} \quad (15.9)$$

9. Since the integration is to be carried out numerically along the path (using the same integration points used for the element stiffness matrix), we have:

$$J^e = \sum_{q=1}^{NGAUS} I(\xi_p, \eta_q) W_q \quad (15.10)$$

where W_q is the weighting factor corresponding to η_q and $NGAUS$ is the order of integration (2 or 3).

10. Stresses σ_x , σ_y , τ_{xy} are readily available at the Gauss points.
11. $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$, $\frac{\partial v}{\partial x}$, and $\frac{\partial v}{\partial y}$ are obtained through the shape function. For instance $\frac{\partial u}{\partial x} = [\frac{\partial N_i}{\partial x}] \{u_i\}$ where the u_i are the nodal displacements and $\frac{\partial N_i}{\partial x}$ is the cartesian derivative of the shape function stored in the $[B]$ matrix:

$$[B] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix} \quad (15.11)$$

where i ranges from 1 to 8 for quadrilateral elements.

12. Another term not yet defined in Eq. 15.9 is $\frac{\partial y}{\partial \eta}$. This term is actually stored already in the Gauss point Jacobian matrix:

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (15.12)$$

13. Finally we are left to determine n_1 and n_2 (components of \mathbf{n}).

- a) Define two arbitrary vectors: \mathbf{A} along $\xi = \xi_{cst}$ and \mathbf{B} along $\eta = \eta_{cst}$ such that:

$$\mathbf{A}^t = [\frac{\partial x}{\partial \eta}, \frac{\partial y}{\partial \eta}, 0] \quad (15.13)$$

$$\mathbf{B}^t = [\frac{\partial x}{\partial \xi}, \frac{\partial y}{\partial \xi}, 0] \quad (15.14)$$

Note that we have defined the three-dimensional components of those two vectors.

- b) Now we define a third vector, which is normal to the plane defined by the preceding two: $\mathbf{C} = \mathbf{A} \times \mathbf{B}$, or:

$$\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & 0 \\ \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & 0 \end{bmatrix} \quad (15.15)$$

This leads to:

$$\mathbf{C} = [0, 0, \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} - \frac{\partial y}{\partial \eta} \frac{\partial x}{\partial \xi}] \quad (15.16)$$

- c) With \mathbf{C} defined, we can now return to the original plane and define

$$\mathbf{D} = \mathbf{C} \times \mathbf{A} \Rightarrow \mathbf{D} = [\underbrace{\frac{\partial y}{\partial \eta}(\frac{\partial y}{\partial \eta} \frac{\partial x}{\partial \xi} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi})}_{D_1}, \underbrace{\frac{\partial x}{\partial \eta}(\frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} - \frac{\partial y}{\partial \eta} \frac{\partial x}{\partial \xi})}_{D_2}, 0] \quad (15.17)$$

- d) The unit normal vector is now given by:

$$\mathbf{n} = \left\{ \begin{array}{l} n_1 \\ n_2 \\ 0 \end{array} \right\} = \left\{ \begin{array}{l} \frac{D_1}{N} \\ \frac{D_2}{N} \\ 0 \end{array} \right\} \quad (15.18)$$

where $N = \sqrt{D_1^2 + D_2^2}$ and all terms are taken from the Jacobian matrix.

15.2 Mixed Mode SIF Evaluation

In subsection 15.1 we have outlined two procedures to extract the J integral from a finite element analysis. Based on this technique, at best only K_I may be determined. In this section, we shall generalize the algorithm to extract both J_1 and J_2 through a postprocessing for our finite element analysis, and subsequently determine K_I and K_{II} from Eq. ???. Once again the outlined procedure is based on the method outlined in (Owen and Fawkes 1983). First let us redefine the two contour integrals according to (Knowles and Sternberg 1972) as:

$$J_k = \int \{w n_k - \mathbf{t} \cdot \frac{\partial \mathbf{d}}{\partial x_k}\} ds \quad (15.19)$$

combining with Eq. ?? we obtain

$$J_1 = \int \{wdy - \mathbf{t} \cdot \frac{\partial \mathbf{d}}{\partial x}\} ds = \frac{K_I^2 + K_{II}^2}{H} + \frac{K_{III}^2}{\mu} \quad (15.20)$$

$$J_2 = \int \{wdx - \mathbf{t} \cdot \frac{\partial \mathbf{d}}{\partial y}\} ds = \frac{-2K_I K_{II}}{H} \quad (15.21)$$

where

$$H = \begin{cases} E & \text{plane strain} \\ \frac{E}{1-\nu^2} & \text{plane stress} \end{cases} \quad (15.22)$$

We note that the original definition of J is recovered from J_1 .

The procedure to determine J_1 and J_2 will be identical to the one outlined in 15.1 and previously presented with the addition of the following equations:

$$dx = -n_2 ds \quad (15.23)$$

$$dx = -\frac{\partial x}{\partial \eta} d\eta \quad (15.24)$$

15.3 Equivalent Domain Integral (EDI) Method

In this section, we shall derive an alternative expression for the energy release rate. Contrarily to the virtual crack extension method where two analyses (or a stiffness derivative) had to be evaluated, in this method, we have to perform only one analysis. The method is really based on Rice's J integral. However, it is recognized that evaluation of J in 2D involves a line integral only and a line integral plus a volume integral if body forces are present, (deLorenzi, H.G. 1985). For 3D problems, the line integral is replaced by a surface integral (and a volume integral for body forces).

Recognizing that surface integrals may not be easily evaluated in 3D, Green's theorem is invoked, and J will be evaluated through a volume integral in 3D and a surface integral in 2D. Thus computationally, this method is quite attractive.

Again as for the previous case, we will start by evaluating the energy release rate, and only subsequently we shall derive expressions for the SIF.

The essence of the method consists in replacing the contour integral, by a closed integral (outer and inner) while multiplying the expression of J by a function q equal to zero on the outer surface and unity on the inner one. We adopt the expression of J derived for a propagating crack (thus determined around a path close to the crack tip). Having defined a closed path, we then apply Green's theorem, and replace a contour integral by a surface integral.

15.3.1 Energy Release Rate J

15.3.1.1 2D case

Recalling the expression for the energy release rate of a propagating crack, Eq. ??

$$J = \lim_{\Gamma_0 \rightarrow 0} \int_{\Gamma_0} \left[(w + T) \delta_{1i} - \sigma_{ij} \frac{\partial u_j}{\partial x_1} \right] n_i d\Gamma \quad (15.25)$$

where w is the strain energy density, T is the kinetic energy

$$T = \frac{1}{2} \rho \frac{\partial u_i}{\partial t} \frac{\partial u_i}{\partial t} \quad (15.26)$$

and δ the Kronecker delta. An alternative form of this equation (Anderson 1995) is

$$J = \lim_{\Gamma_0 \rightarrow 0} \int_{\Gamma_0} \left[(w + T) dy - \sigma_{ij} n_i \frac{\partial u_j}{\partial x} d\Gamma \right] \quad (15.27)$$

Unlike the conventional J integral, the contour path for this equation can not be arbitrarily selected.

This equation is derived from an energy balance approach, and is thus applicable to all types of material models. However, this J integral is path independent only if Γ is within an elastic zone; if it is taken within the plastic zone than it will be path dependent.

This equation is not well suited for numerical evaluation as the path would have to be along a vanishingly small one where the stresses and strains could not be determined. As such, (Li, F. Z. and Shih, C. F. and Needleman,

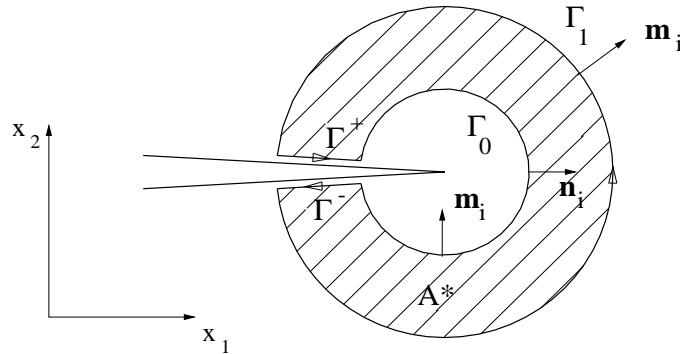


Figure 15.2: Simply connected Region A Enclosed by Contours Γ_1 , Γ_0 , Γ_+ , and Γ_- , (Anderson 1995)

A. 1985), we will be rewriting an alternative form of this equation, by considering the contour shown in Fig. 15.2 where Γ_1 , is the outer finite contour, Γ_0 is the inner vanishingly small contour, and Γ_+ , and Γ_- are respectively the upper and lower crack surfaces along the contour. For quasi-static cases ($T = 0$), let us construct a closed contour by connecting inner and outer ones. The outer one Γ_1 is finite, while the inner one Γ_0 is vanishingly small. For linear (or nonlinear) elastic material J can be evaluated along either one of those two contours, but only the inner one gives the exact solution in the general case. Thus, we can rewrite Eq. 15.25 around the following closed contour

$$\Gamma^* = \Gamma_1 + \Gamma_+ + \Gamma_- - \Gamma_0 \quad (15.28)$$

yielding (and assuming that the crack faces are traction free)

$$J = \int_{\Gamma^*} \left[\sigma_{ij} \frac{\partial u_j}{\partial x_1} - w \delta_{1i} \right] q m_i d\Gamma - \int_{\Gamma_+ \cup \Gamma_-} \sigma_{2j} \frac{\partial u_j}{\partial x_1} q d\Gamma \quad (15.29)$$

where m_i is the outward normal to Γ^* (thus $m_i = n_i$ on Γ_1 , and $m_i = -n_i$ on Γ_0 , $m_1 = 0$ and $m_2 = \pm 1$ on Γ_+ and Γ_-), and q is an arbitrary but smooth function which is equal to unity on Γ_0 and zero on Γ_1 . Note that since the integral is taken along the contours, by explicitly specifying $q = 0$ on the outer one, and $q = 1$ on the inner one, Eq. 15.25 and 15.29 are identical. Furthermore, in the absence of crack surface tractions, the second term is equal to zero.

Applying the divergence theorem to Eq. 15.29

$$\oint_{\Gamma} \mathbf{v} \cdot \mathbf{n} = \int_A \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \right) dx dy \quad (15.30)$$

we obtain

$$J = \int_{A^*} \frac{\partial}{\partial x_1} \left\{ \left[\sigma_{ij} \frac{\partial u_j}{\partial x_1} - w \delta_{1i} \right] q \right\} dA \quad (15.31)$$

$$= \int_{A^*} \left[\left(\sigma_{ij} \frac{\partial u_j}{\partial x_1} - w \delta_{1i} \right) \frac{\partial q}{\partial x_1} + \left(\frac{\partial}{\partial x_1} \left(\sigma_{ij} \frac{\partial u_j}{\partial x_1} \right) - \frac{\partial w}{\partial x_1} \right) q \right] dA \quad (15.32)$$

where A^* is the area enclosed by Γ^* .

Let us show that the second term is equal to zero:

$$\frac{\partial}{\partial x_i} \left(\sigma_{ij} \frac{\partial u_j}{\partial x_1} \right) = \underbrace{\sigma_{ij} \frac{\partial}{\partial x_i} \left(\frac{\partial u_j}{\partial x_1} \right)}_{\frac{\partial w}{\partial x}} + \underbrace{\frac{\partial \sigma_{ij}}{\partial x_i} \frac{\partial u_j}{\partial x_1}}_0 \quad (15.33)$$

however from equilibrium we have

$$\frac{\partial \sigma_{ij}}{\partial x_i} = 0 \quad (15.34)$$

Furthermore, the derivative of the strain energy density is

$$\frac{\partial w}{\partial x} = \frac{\partial w}{\partial \varepsilon_{ij}} \frac{\partial \varepsilon_{ij}}{\partial x} = \sigma_{ij} \frac{\partial \varepsilon_{ij}}{\partial x} \quad (15.35)$$

substituting

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (15.36)$$

we obtain

$$\frac{\partial w}{\partial x} = \frac{1}{2} \sigma_{ij} \left[\frac{\partial}{\partial x} \left(\frac{\partial u_i}{\partial x_j} \right) + \frac{\partial}{\partial x} \left(\frac{\partial u_j}{\partial x_i} \right) \right] = \sigma_{ij} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x} \right) \quad (15.37)$$

Hence, it is evident that the second term of Eq. 15.32 vanishes and that we are left with

$$J = \int_{A^*} \left[\sigma_{ij} \frac{\partial u_i}{\partial x_1} - w \delta_{1i} \right] \frac{\partial q}{\partial x_i} dA \quad (15.38)$$

This expression, is analogous to the one proposed by Babuska for a surface integral based method to evaluate stress intensity factors, (Babuska and Miller 1984).

We note that deLorenzi (deLorenzi, H.G. 1985) has shown that the energy release rate is given by

$$G = \frac{1}{\Delta A} \int \left(\sigma_{ij} \frac{\partial u_j}{\partial x_1} - w \delta_{1i} \right) \frac{\partial \Delta x_1}{\partial x_i} dA \quad (15.39)$$

for a unit crack growth extension along x_1 . Thus comparing Eq. 15.38 with 15.39, we observe that the two expressions are identical for $q = \frac{\Delta x_1}{\Delta a}$, and thus q can be interpreted as a normalized virtual displacement. In this context it was merely a mathematical device.

In summary, we have replace a contour integral by an equivalent area integral to determine J .

15.3.1.2 3D Generalization

In this section, we shall generalize to 3D our previous derivation, (Anderson 1995). From Fig. 15.3 we define a local coordinate system such that x_1 is normal to the crack front, x_2 normal to the crack plane, and x_3 tangent to the crack front. For an arbitrary point, the J integral is given by Eq. 15.25. We now consider a tube of length ΔL and

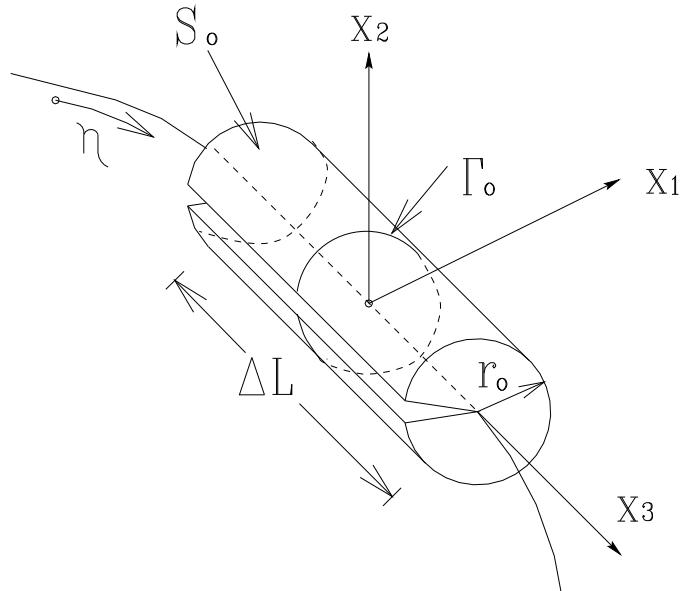


Figure 15.3: Surface Enclosing a Tube along a Three Dimensional Crack Front, (Anderson 1995)

radius r_0 that surrounds the segment of the crack front under consideration. We now define a weighted average J over the crack front segment of length ΔL as

$$\bar{J} \Delta L = \int_{\Delta L} J(\eta) q d\eta \quad (15.40)$$

$$= \lim_{r_0 \rightarrow 0} \int_{S_0} \left[w \delta_{1i} - \sigma_{ij} \frac{\partial u_j}{\partial x_1} \right] q n_i dS \quad (15.41)$$

where $J(\eta)$ is the point-wise value of J , S_0 is the vanishingly small surface area of the tube, q is the weight function previously introduced. q can be again interpreted as a virtual crack advance and Fig. 15.4 illustrates an incremental crack advance over ΔL where q is defined as

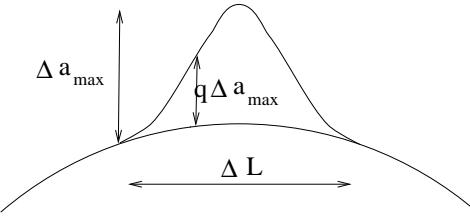


Figure 15.4: Interpretation of q in terms of a Virtual Crack Advance along ΔL , (Anderson 1995)

$$\Delta a(\eta) = q(\eta) \Delta a_{max} \quad (15.42)$$

and the corresponding incremental area of the virtual crack is

$$\Delta A_c = \Delta a_{max} \int_{\Delta L} q(\eta) d\eta \quad (15.43)$$

As in the previous case, this expression of J can not be numerically determined for a vanishingly small radius r_0 , as such and as in the previous 2D case, we define a second tube of radius r_1 around the crack front, Fig. 15.5.

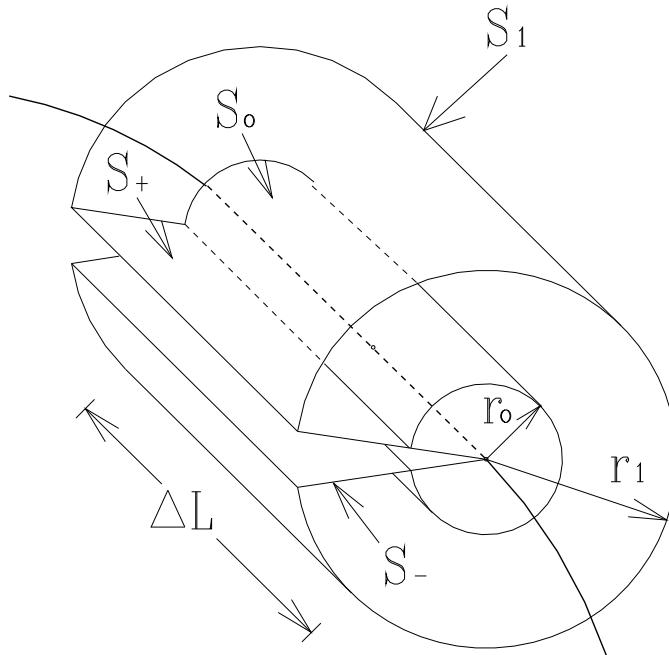


Figure 15.5: Inner and Outer Surfaces Enclosing a Tube along a Three Dimensional Crack Front

$$\bar{J}\Delta L = \oint_{S^*} \left[\sigma_{ij} \frac{\partial u_i}{\partial x_1} - w \delta_{1i} \right] q m_i dS - \oint_{S_- \cup S_+} \sigma_{2j} \frac{\partial u_j}{\partial x_1} q dS \quad (15.44)$$

where

$$S^* = S_1 + S_+ + S_- - S_0 \quad (15.45)$$

and S_+ and S_- are the upper and lower crack surfaces respectively, S_0 and S_1 the inner and outer tube surfaces. Note that this equation is the 3D counterpart of Eq. 15.29 which was written in 2D.

Applying the divergence theorem, this equation reduces to a volume integral

$$\begin{aligned} \bar{J}\Delta L &= \int_{V^*} \left\{ \left[\sigma_{ij} \frac{\partial u_j}{\partial x_i} - w \delta_{1i} \right] \frac{\partial q}{\partial x_i} + \left[-\frac{\partial w}{\partial x_1} + \frac{\partial}{\partial x_j} \left(\sigma_{ij} \frac{\partial u_i}{\partial x_1} \right) \right] q \right\} dV \\ &\quad + \int_{A_1 \cup A_2} \left(w \delta_{1i} - \sigma_{ij} \frac{\partial u_i}{\partial x_1} \delta_{1i} \right) q dA \end{aligned} \quad (15.46)$$

and q must be equal to zero at either end of ΔL that is on A_1 and A_2 . In (Nikishkov, G. P. and Atluri, S. N. 1987) it is shown that in the absence of non-elastic (thermal and plastic) deformations the second term would be equal to zero. The third term will also be equal to zero because q is arbitrarily selected to be zero at each end.

15.3.2 Extraction of SIF

From Eq. 15.46 it is impossible to extract the 3 distinct stress intensity factors. Hence we shall generalize this equation and write it as (Nikishkov, G. P. and Atluri, S. N. 1987) (ignoring the second and third terms)

$$\bar{J}_k \Delta L = \int_{V^*} \left(\sigma_{ij} \frac{\partial u_i}{\partial x_k} \frac{\partial q}{\partial x_j} - w \frac{\partial q}{\partial x_k} \right) dV \quad (15.47)$$

Note that $k = 1, 2$ only thus defining $G_1 = J_1$ and $G_2 = J_2$. However, (Nikishkov, G. P. and Atluri, S. N. 1987) have shown that G_3 has a similar form and is equal to

$$G_{III} = \int_{V^*} \left(\sigma_{3j} \frac{\partial u_3}{\partial x_1} \frac{\partial q}{\partial x_j} - w^{III} \frac{\partial q}{\partial x_1} \right) dV \quad (15.48)$$

With G_1 , G_2 and G_3 known we need to extract the three stress intensity factors K_I , K_{II} and K_{III} . Again there are two approaches.

15.3.2.1 J Components

Based on the solution by Nikishkov, (Nikishkov and Vainshtok 1980)

$$\begin{aligned} K_I &= \frac{1}{2} \sqrt{E^*} \left(\sqrt{(J_1 - J_2 - G_3)} + \sqrt{(J_1 + J_2 - G_3)} \right) \\ K_{II} &= \frac{1}{2} \sqrt{E^*} \left(\sqrt{(J_1 - J_2 - G_3)} - \sqrt{(J_1 + J_2 - G_3)} \right) \\ K_{III} &= \sqrt{2\mu G_3} \end{aligned} \quad (15.49)$$

where, (Nikishkov, G. P. and Atluri, S. N. 1987)

$$E^* = E \left[\frac{1}{1 - \nu^2} + \left(\frac{\nu}{1 + \nu} \right) \frac{\varepsilon_{33}}{\varepsilon_{11} + \varepsilon_{22}} \right] \quad (15.50)$$

which is a weighted value of E such that we retrieve $E^* = \frac{E}{1 - \nu^2}$ for plane strain and $E^* = E$ for plane stress.

15.3.2.2 σ and u Decomposition

As for the solution by Shah, we can decompose the displacement field as

$$\begin{aligned} \{u\} &= \left\{ u^I \right\} + \left\{ u^{II} \right\} + \left\{ u^{III} \right\} \\ &= \frac{1}{2} \left\{ \begin{array}{c} u_1 + u'_1 \\ u_2 - u'_2 \\ u_3 + u'_3 \end{array} \right\} + \frac{1}{2} \left\{ \begin{array}{c} u_1 - u'_1 \\ u_2 + u'_2 \\ 0 \end{array} \right\} + \frac{1}{2} \left\{ \begin{array}{c} 0 \\ 0 \\ u_3 - u'_3 \end{array} \right\} \end{aligned} \quad (15.51)$$

similarly the stresses are decomposed as

$$\{\sigma\} = \left\{ \sigma^I \right\} + \left\{ \sigma^{II} \right\} + \left\{ \sigma^{III} \right\}$$

$$= \frac{1}{2} \begin{Bmatrix} \sigma_{11} + \sigma'_{11} \\ \sigma_{22} + \sigma'_{22} \\ \sigma_{33} + \sigma'_{33} \\ \sigma_{12} - \sigma'_{12} \\ \sigma_{23} - \sigma'_{23} \\ \sigma_{31} - \sigma'_{31} \end{Bmatrix} + \frac{1}{2} \begin{Bmatrix} \sigma_{11} - \sigma'_{11} \\ \sigma_{22} - \sigma'_{22} \\ 0 \\ \sigma_{12} + \sigma'_{12} \\ 0 \\ 0 \end{Bmatrix} + \frac{1}{2} \begin{Bmatrix} 0 \\ 0 \\ \sigma_{33} - \sigma'_{33} \\ 0 \\ \sigma_{23} + \sigma'_{23} \\ \sigma_{31} + \sigma'_{31} \end{Bmatrix} \quad (15.52)$$

where

$$u'_i(x_1, x_2, x_3) = u_i(x_1, -x_2, x_3) \quad (15.53)$$

$$\sigma'_{ij}(x_1, x_2, x_3) = \sigma_{ij}(x_1, -x_2, x_3) \quad (15.54)$$

and the stress intensity factors are then determined from

$$K_I = \sqrt{E'G_I} \quad K_{II} = \sqrt{E'G_{II}} \quad K_{III} = \sqrt{2\mu G_{III}} \quad (15.55)$$

where

$$G_k = \int_{V^*} \left(\sigma_{kj} \frac{\partial u_k}{\partial x_1} \frac{\partial q}{\partial x_j} - w^k \frac{\partial q}{\partial x_1} \right) dV \quad (15.56)$$

Whereas this method may be difficult to use in conjunction with a 3D finite element mesh generated by triangulation (due to the lack of symmetry around the crack front), it has been successfully used by Cervenka (1994) in conjunction with a unit volume integration in the FE code MERLIN (Saouma, Červenka and Reich 2008).

Chapter 16

HILLERBORG'S MODEL

From the previous discussion, it is clear that concrete softening is characterized by a stress-crack opening width curve (and not stress-strain). The exact characterization of the softening response should ideally be obtained from a uniaxial test of an uncracked specimen. However, it has been found (Li and Liang 1986, Hordijk, Reinhardt and Cornelissen 1989) that not only are those tests extremely sensitive, but drastically different results can be obtained from different geometries, sizes, and testing machines. Hence, the softening curve is often indirectly determined by testing notched specimens.

In what is probably the most referenced work in the nonlinear fracture of concrete literature, Hillerborg (Hillerborg, Modéer and Petersson 1976) presented in 1976 a very simple and elegant model which has been previously described qualitatively. In this model, the crack is composed of two parts, Fig. 16.1:

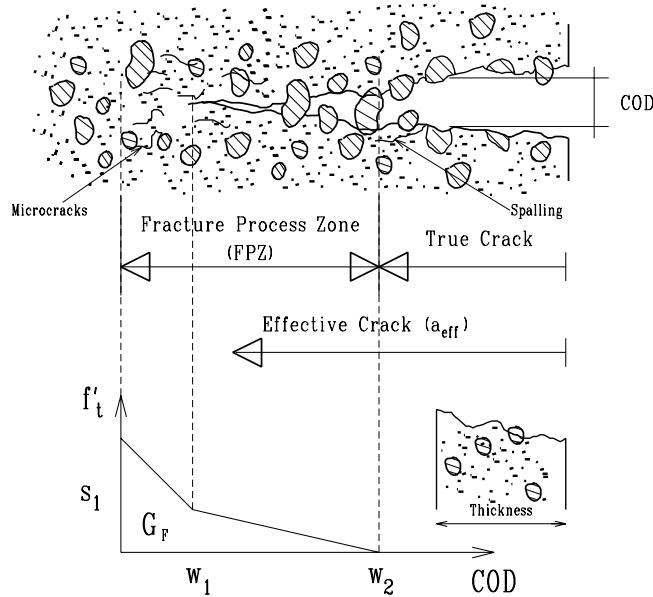


Figure 16.1: Hillerborg's Fictitious Crack Model

1. True or physical crack across which no stresses can be transmitted. Along this zone we have both displacement and stress discontinuities.
2. Fictitious crack, or Fracture Process Zone (FPZ) ahead of the previous one, characterized by:
 - a) peak stress at its tip equal to the tensile strength of concrete
 - b) decreasing stress distribution from f_t' at the tip of the fictitious crack to zero at the tip of the physical crack

It should be noted that along the FPZ, we have displacement discontinuity and stress continuity.

This model is among the most widely used in non-linear fracture mechanics finite element analysis, however due to the computational complexity, few “engineering” structures have been analyzed. In addition,

1. There is an inflection point in the descending branch.
 - a) The first part has been associated with (unconnected) microcracking ahead of the stress-free crack
 - b) The second part with bridging of the crack by aggregates

2. The area under the curve, termed the fracture energy G_F (not to be confused with G_c or critical energy release rate), is a measure of the energy that needs to be spent to generate a unit surface of crack.
3. By analyzing numerous test data, Bažant and Oh (Bažant, Z.P. 1984) found that G_F may be predicted (with a coefficient of variation of about 16%) from the following empirical equation:

$$G_F = 0.0214(f'_t + 127)f_t'^2 \frac{d_a}{E_c} \quad (16.1)$$

where E_c and f'_t are in pounds per square inch, d_a is the aggregate size in inches.

Using extensive nonlinear optimization studies based on the Levenberg-Marquardt algorithm, Bažant and Becq-Giraudon (2001) obtained two simple approximate formulae for the means of G_f and G_F as functions of the compressive strength f'_c , maximum aggregate size d_a , water-cement ratio $/c$, and shape of aggregate (crushed or river);

$$\begin{aligned} G_f &= \alpha_0 \left(\frac{f'_c}{0.051} \right)^{0.46} \left(1 + \frac{d_a}{11.27} \right)^{0.22} \left(\frac{w}{c} \right)^{-0.30} & \omega_{G_f} = 17.8\% \\ G_F &= 2.5\alpha_0 \left(\frac{f'_c}{0.051} \right)^{0.46} \left(1 + \frac{d_a}{11.27} \right)^{0.22} \left(\frac{w}{c} \right)^{-0.30} & \omega_{G_F} = 29.9\% \\ c_f &= \exp \left[\gamma_0 \left(\frac{f'_c}{0.022} \right)^{-0.019} \left(1 + \frac{d_a}{15.05} \right)^{0.72} \left(\frac{w}{c} \right)^{0.2} \right] & \omega_{c_f} = 47.6\% \end{aligned} \quad (16.2)$$

Here $\alpha_0 = \gamma_0 = 1$ for rounded aggregates, while $\alpha_0 = 1.44$ and $\gamma_0 = 1.12$ for crushed or angular aggregates; ω_{G_f} and ω_{G_F} are the coefficients of variation of the ratios G_f^{test}/G_f and G_F^{test}/G_F , for which a normal distribution may be assumed, and ω_{c_f} is the coefficient of variation of c_f^{test}/c_f , for which a lognormal distribution should be assumed (ω_{c_f} is approximately equal to the standard deviation of $\ln c_f$).

4. G_F : or fracture energy. For gravity dams, a value of 1.35×10^{-3} kip/in. is recommended, (Saouma, Broz, Brühwiler and Boggs 1991). Note that for arch dams, this value could probably be increased on the basis of laboratory tests. Also, laboratory tests could be performed on recovered cores to obtain a better indication of G_F , (Brühwiler, E. 1988).
5. Shape of the softening diagram ($\sigma - COD$), and in general a bi-linear model for the strain softening should be used. With reference to Fig. 20.3, A topic of much research lately has been the experimental determination of the

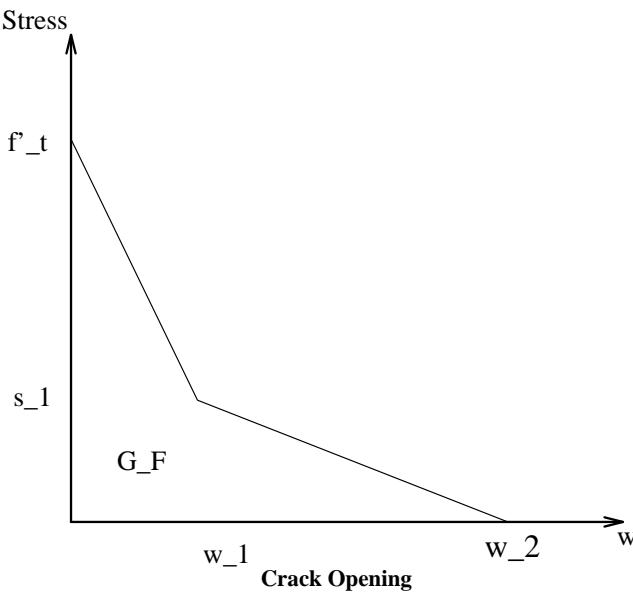


Figure 16.2: Concrete Strain Softening Models

fracture energy G_F , and the resulting shape of the softening diagram (Cedolin, Dei Poli and Iori 1987, Petersson 1981, Wittmann et al. 1988, Jeang and Hawkins 1985, Gopalaratnam and Shah 1985, Duda 1990, Giuriani and Rosati 1986). In order to assess the relevance of the exact value of G_F and the softening curve shape on numerical

simulations, three different set of fracture experiments are analysed using the average reported fracture energy. The shape of the softening diagram is assumed to be the bilinear one proposed in (Wittmann et al. 1988), Fig. 20.3. This simple model can be uniquely defined in terms of the tensile strength f'_t , and the fracture energy G_F . In (Brühwiler and Wittmann 1990), it was found that the optimal points for concrete with 1" maximum size aggregate are:

$$s_1 = 0.4f'_t \quad (16.3)$$

$$w_1 = 0.8 \frac{G_F}{f'_t} \quad (16.4)$$

$$w_2 = 3 \frac{G_F}{f'_t} \quad (16.5)$$

whereas for structural concrete, (Wittmann et al. 1988), the corresponding values are:

$$s_1 = \frac{f'_t}{4} \quad (16.6)$$

$$w_1 = 0.75 \frac{G_F}{f'_t} \quad (16.7)$$

$$w_2 = 5 \frac{G_F}{f'_t} \quad (16.8)$$

where f'_t is the uniaxial tensile strength. Within the context of a nonlinear fracture mechanics analysis, this tensile strength can not be taken as zero, otherwise there will be no fracture process zone. As f'_t is seldom determined experimentally, it is assumed to be 9% of f'_c , (Mindess and Young 1981).

6. In lieu of a direct tension test, a flexural test can be performed under strain control, and the fracture energy G_F could still be determined from the area under the load and corresponding displacement curve.
7. For dynamic analysis, the fracture properties of dam concrete depend on both rate of loading and preloadings. Test results (Brühwiler and Wittmann 1990) show that the fracture properties generally increase with increasing loading rate. However, dynamic compressive preloading leads to a reduction of the fracture properties at both quasi-static and high loading rates.

Chapter 17

LOCALIZED FAILURE

17.1 Fictitious Crack Model; FCM (MM: 7)

Originally published as:

Implementation and Validation of a nonlinear fracture model in a 2D/3D finite element code by Reich, Plizzari, Cervenka and Saouma; in *Numerical Models in Fracture of Concrete*; Wittman Ed., Balkema (1993).

17.1.1 Introduction

An incremental formulation for the Fictitious Crack Model (FCM) will be presented. The computational algorithm treats the structure as a set of sub-domains bonded along assumed crack paths. The crack paths are defined by interface elements that initially act as constraints enforcing the bond between adjacent sub-domains, but change state to function as standard interface elements as the crack propagates. Constraints are enforced on the global system of equations using a penalty approach. A load scaling strategy, which allows for load controlled analyses in the post-peak regime, is used to enforce stress continuity at the tip of the Fracture Process Zone (FPZ). To demonstrate the accuracy of the computational algorithm, a series of three wedge-splitting (WS) test specimens are analyzed. Specimen sizes are 31, 91, and 152 cm (1, 3, and 5 ft). Material properties for the concrete are taken as the mean values of the observed experimental results for all specimen sizes. The computed results are compared to the envelopes of the experimental response for each specimen size.

The most commonly implemented nonlinear fracture model for concrete using the discrete crack approach is the FCM (Hillerborg et al. 1976). In the FCM the zone of micro-cracking and debonding ahead of the crack front is modeled as a cohesive stress that acts to close the crack. The magnitude of the cohesive stresses on the crack surface are determined by a softening law that relates the stress to the relative displacement of the crack surfaces through the fracture energy. Many implementations of the FCM have been reported (Ingraffea and Gerstle 1984, Roelfstra and Sadouki 1986, Dahlblom and Ottosen 1990, Bocca, Carpinteri and Valente 1990, Gopalaratnam and Ye 1991, Gerstle and Xie 1992), but none of the implementations based on a discrete crack approach claim to be based on the standard incremental formulation normally associated with nonlinear analyses. Only the implementation by Dahlblom and Ottosen (Dahlblom and Ottosen 1990), which is based on a smeared crack approach, uses an incremental formulation.

In this chapter, an incremental solution algorithm for the FCM based in the discrete crack approach will be presented and its performance evaluated by comparing the computed response of WS test specimens against known experimental results.

Treatment of the structure as a set of bonded sub-domains results in a system of mixed equations with the unknowns being displacements and surface tractions on the interface between the sub-domains. The weak form of the system of mixed equations will be derived from the Principle of Virtual Work. The weak form equations will then be discretized for solution using the finite element method. The penalty method solution for the mixed system of equations will be discussed; particularly the automatic selection of the penalty number. Finally, an incremental-iterative solution strategy based on the modified-Newton algorithm that includes load scaling and allows for load control in the post-peak regime will be discussed.

17.1.2 Weak Form of Governing Equations

Figure 17.1 shows a body consisting of two sub-domains, Ω_1 and Ω_2 that intersect on a surface Γ_I without penetration. Each sub-domain may be subject to body forces \mathbf{b}_m or to prescribed surface tractions $\hat{\mathbf{t}}_m$ on Γ_{t_m} . Defining the volume of the body as

$$\Omega = \Omega_1 \cup \Omega_2 \quad (17.1)$$

and the surface of the body subject to prescribed surface tractions as

$$\Gamma_t = \Gamma_{t_1} \cup \Gamma_{t_2}, \quad (17.2)$$

the Principle of Virtual Work for the body is

$$\int_{\Omega} \delta \epsilon^T \sigma d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \hat{\mathbf{t}} d\Gamma = 0 \quad (17.3)$$

where

$$\delta\boldsymbol{\epsilon} = \mathbf{L}\delta\mathbf{u} \quad (17.4-a)$$

$$\boldsymbol{\epsilon} = \mathbf{Lu} \quad (17.4-b)$$

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon} \quad (17.4-c)$$

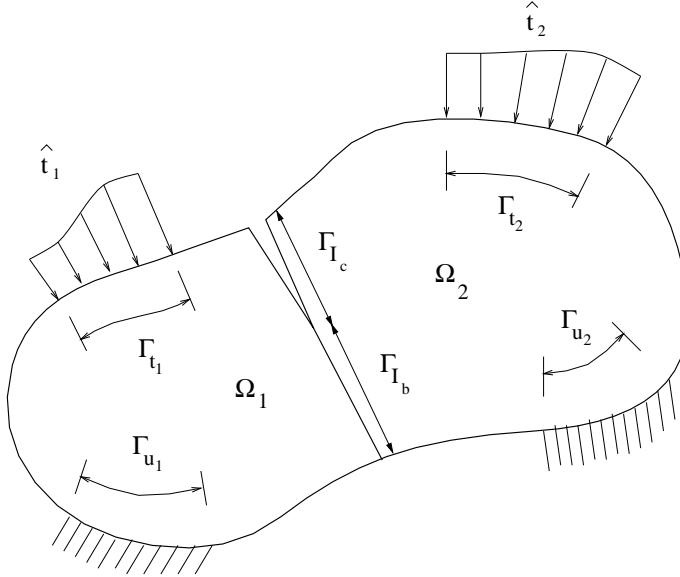


Figure 17.1: Body Consisting of Two Sub-domains

Within each sub-domain of the body Ω_m the Principle of Virtual Work must also hold, but additional integrals are required to account for the work performed by the surface tractions \mathbf{t}_{I_m} on the interface Γ_I . Surface tractions on the interface are due to bonding of the sub-domains \mathbf{t}_{b_m} or to cohesive stresses in the FPZ \mathbf{t}_{c_m} . In either case, stress continuity on Γ_I requires that

$$\mathbf{t}_{b_2} = -\mathbf{t}_{b_1} \quad (17.5-a)$$

$$\mathbf{t}_{c_2} = -\mathbf{t}_{c_1} \quad (17.5-b)$$

Defining the interface surface as

$$\Gamma_I = \Gamma_{I_b} \cup \Gamma_{I_c}, \quad (17.6)$$

where Γ_{I_b} is the bonded interface surface and Γ_{I_c} is the interface surface subject to cohesive stresses, the external work on the interface is written as

$$\int_{\Gamma_I} \delta\mathbf{u}_1^T \mathbf{t}_{I_1} d\Gamma = \int_{\Gamma_{I_b}} \delta\mathbf{u}_1^T \mathbf{t}_b d\Gamma + \int_{\Gamma_{I_c}} \delta\mathbf{u}_1^T \mathbf{t}_c d\Gamma \quad (17.7-a)$$

$$\int_{\Gamma_I} \delta\mathbf{u}_2^T \mathbf{t}_{I_2} d\Gamma = - \int_{\Gamma_{I_b}} \delta\mathbf{u}_2^T \mathbf{t}_b d\Gamma - \int_{\Gamma_{I_c}} \delta\mathbf{u}_2^T \mathbf{t}_c d\Gamma \quad (17.7-b)$$

Both \mathbf{t}_b and \mathbf{t}_c are unknown, but as \mathbf{t}_b acts on the bonded, or constrained, interface it will be treated as a Lagrange multiplier

$$\lambda = \mathbf{t}_b \quad (17.8)$$

Substituting λ into Equations 17.3 and 17.4-a-17.4-c and including the external work performed by the surface tractions on the interface surface, the Principle of Virtual Work for sub-domains Ω_1 and Ω_2 is written as

$$\boxed{\begin{aligned} \int_{\Omega_1} \delta\boldsymbol{\epsilon}_1^T \boldsymbol{\sigma}_1 d\Omega - \int_{\Omega_1} \delta\mathbf{u}_1^T \mathbf{b}_1 d\Omega - \int_{\Gamma_{t_1}} \delta\mathbf{u}_1^T \hat{\mathbf{t}}_1 d\Gamma - \int_{\Gamma_{I_b}} \delta\mathbf{u}_1^T \lambda d\Gamma - \int_{\Gamma_{I_c}} \delta\mathbf{u}_1^T \mathbf{t}_c d\Gamma &= 0 \\ \int_{\Omega_2} \delta\boldsymbol{\epsilon}_2^T \boldsymbol{\sigma}_2 d\Omega - \int_{\Omega_2} \delta\mathbf{u}_2^T \mathbf{b}_2 d\Omega - \int_{\Gamma_{t_2}} \delta\mathbf{u}_2^T \hat{\mathbf{t}}_2 d\Gamma + \int_{\Gamma_{I_b}} \delta\mathbf{u}_2^T \lambda d\Gamma + \int_{\Gamma_{I_c}} \delta\mathbf{u}_2^T \mathbf{t}_c d\Gamma &= 0 \end{aligned}} \quad (17.9)$$

On Γ_{I_b} the displacements for the two sub-domains, $\mathbf{u}_1|_{\Gamma_{I_b}}$ and $\mathbf{u}_2|_{\Gamma_{I_b}}$, must be equal. This condition can be written as a constraint in the strong form

$$\mathbf{u}_2|_{\Gamma_{I_b}} - \mathbf{u}_1|_{\Gamma_{I_b}} = \mathbf{0}, \quad (17.10)$$

but a weak form is required to be compatible with Equation 17.5-a-17.5-b. The following weak form

$$\int_{\Gamma_i} \delta \boldsymbol{\lambda}^T (\mathbf{u}_2 - \mathbf{u}_1) d\Gamma = 0$$

(17.11)

was chosen for the constraint equation as it makes the system of mixed equations symmetric.

17.1.3 Discretization of Governing Equations

Discretization of Equations 17.9-17.9 and 17.11 will be presented as if each sub-domain were an element (?); the extension to multi-element sub-domains is straightforward and will be omitted from this discussion. Each sub-domain Ω_m is discretized for displacements \mathbf{u}_m such that nodes on Γ_{t_m} and Γ_I are included in the vector of discrete displacements $\bar{\mathbf{u}}_m$. The number of nodes on Γ_I in Ω_1 is equal to the number of nodes on Γ_I in Ω_2 . For each node on Γ_I in Ω_1 there is a node on Γ_I in Ω_2 with the same coordinates. The nodes at which the surface tractions due to bonding $\boldsymbol{\lambda}$ on Γ_{I_b} are discretized are at the same locations as those for the displacements.

Displacements \mathbf{u}_m within the sub-domains Ω_m and the surface tractions $\boldsymbol{\lambda}$ on the bonded interface Γ_{I_b} are defined in terms of their discretized counterparts using shape functions

$$\mathbf{u}_m = \mathbf{N}_{u_m} \bar{\mathbf{u}}_m \quad (17.12-a)$$

$$\boldsymbol{\lambda} = \mathbf{N}_{\lambda} \bar{\boldsymbol{\lambda}} \quad (17.12-b)$$

$$\delta \mathbf{u}_m = \mathbf{N}_{u_m} \delta \bar{\mathbf{u}}_m \quad (17.12-c)$$

$$\delta \boldsymbol{\lambda} = \mathbf{N}_{\lambda} \delta \bar{\boldsymbol{\lambda}} \quad (17.12-d)$$

\mathbf{N}_{u_m} and \mathbf{N}_{λ} are standard shape functions in that for each node there is a corresponding shape function whose value is one at that node and zero at all other nodes.

To discretize the integral defining the virtual strain energy, the stresses and the virtual strains defined in Equation 17.4-a-17.4-c must be expressed in terms of the discrete displacements and virtual displacements using Equations 17.12-a-17.12-b and 17.12-c-17.12-d

$$\delta \boldsymbol{\epsilon}_m = \mathbf{L} \mathbf{N}_{u_m} \delta \bar{\mathbf{u}}_m \quad (17.13-a)$$

$$\boldsymbol{\sigma}_m = \mathbf{D}_m \mathbf{L} \mathbf{N}_{u_m} \bar{\mathbf{u}}_m \quad (17.13-b)$$

Defining the discrete strain-displacement operator \mathbf{B}_m as

$$\mathbf{B}_m = \mathbf{L} \mathbf{N}_{u_m}, \quad (17.14)$$

the virtual strain energy can be written as

$$\int_{\Omega_m} \delta \boldsymbol{\epsilon}_m^T \boldsymbol{\sigma}_m d\Omega = \delta \bar{\mathbf{u}}_m^T \int_{\Omega_m} \mathbf{B}_m^T \mathbf{D}_m \mathbf{B}_m d\Omega \bar{\mathbf{u}}_m \quad (17.15)$$

Recognizing that

$$\mathbf{K}_m = \int_{\Omega_m} \mathbf{B}_m^T \mathbf{D}_m \mathbf{B}_m d\Omega \quad (17.16)$$

is the standard stiffness matrix for the finite element method, Equation 17.15 can be rewritten as

$$\int_{\Omega_m} \delta \boldsymbol{\epsilon}_m^T \boldsymbol{\sigma}_m d\Omega = \delta \bar{\mathbf{u}}_m^T \mathbf{K}_m \bar{\mathbf{u}}_m \quad (17.17)$$

Discretization of the integrals for the internal virtual work due to body forces and the external virtual work due to prescribed surface tractions simply involves expressing the virtual displacements in terms of the discrete virtual displacements using Equation 17.12-c-17.12-d

$$\int_{\Omega_m} \delta \mathbf{u}_m^T \mathbf{b}_m d\Omega = \delta \bar{\mathbf{u}}_m^T \int_{\Omega_m} \mathbf{N}_{u_m}^T \mathbf{b}_m d\Omega \quad (17.18-a)$$

$$\int_{\Gamma_{t_m}} \delta \mathbf{u}_m^T \hat{\mathbf{t}}_m d\Gamma = \delta \bar{\mathbf{u}}_m^T \int_{\Gamma_{t_m}} \mathbf{N}_{u_m}^T \hat{\mathbf{t}}_m d\Gamma \quad (17.18-b)$$

Recognizing that

$$\mathbf{f}_m = \int_{\Omega_m} \mathbf{N}_{u_m}^T \mathbf{b}_m d\Omega + \int_{\Gamma_{t_m}} \mathbf{N}_{u_m}^T \hat{\mathbf{t}}_m d\Gamma \quad (17.19)$$

is the standard applied load vector for the finite element method, the sum of the internal virtual work and the external virtual work is

$$\int_{\Omega_m} \delta \mathbf{u}_m^T \mathbf{b}_m d\Omega + \int_{\Gamma_{t_m}} \delta \mathbf{u}_m^T \hat{\mathbf{t}}_m d\Gamma = \delta \bar{\mathbf{u}}_m^T \mathbf{f}_m \quad (17.20)$$

To discretize the external virtual work due to surface tractions on the interface, the surface tractions and the virtual displacements must be expressed in terms of the discrete surface tractions and virtual displacements using Equations 17.12-a-17.12-b and 17.12-c-17.12-d

$$\int_{\Gamma_{I_b}} \delta \mathbf{u}_m^T \boldsymbol{\lambda} d\Gamma = \delta \bar{\mathbf{u}}_m^T \int_{\Gamma_{I_b}} \mathbf{N}_{u_m}^T \mathbf{N}_{\lambda} d\Gamma \bar{\boldsymbol{\lambda}} \quad (17.21-a)$$

$$\int_{\Gamma_{t_c}} \delta \mathbf{u}_m^T \mathbf{t}_c d\Gamma = \delta \bar{\mathbf{u}}_m^T \int_{\Gamma_{t_c}} \mathbf{N}_{u_m}^T \mathbf{t}_c d\Gamma \quad (17.21-b)$$

Defining the operator matrix for the load vector due to surface tractions on the bonded interface as

$$\mathbf{Q}_m = \int_{\Gamma_{I_b}} \mathbf{N}_{u_m}^T \mathbf{N}_{\lambda} d\Gamma \quad (17.22)$$

and the load vector for the cohesive stresses as

$$\mathbf{f}_{c_m} = \int_{\Gamma_{t_c}} \mathbf{N}_{u_m}^T \mathbf{t}_c d\Gamma \quad (17.23)$$

the external work due to surface tractions on the interface is

$$\int_{\Gamma_{I_b}} \delta \mathbf{u}_m^T \boldsymbol{\lambda} d\Gamma + \int_{\Gamma_{t_c}} \delta \mathbf{u}_m^T \mathbf{t}_c d\Gamma = \delta \bar{\mathbf{u}}_m^T (\mathbf{Q}_m \bar{\boldsymbol{\lambda}} + \mathbf{f}_{c_m}) \quad (17.24)$$

To discretize the weak constraint equation, the displacements and the virtual surface tractions must be expressed in terms of the discrete displacements and the discrete virtual surface tractions using Equations 17.12-a-17.12-b and 17.12-c-17.12-d

$$\int_{\Gamma_{I_b}} \delta \boldsymbol{\lambda}^T \mathbf{u}_1 d\Gamma = \delta \bar{\boldsymbol{\lambda}}^T \int_{\Gamma_{I_b}} \mathbf{N}_{\lambda}^T \mathbf{N}_{u_1} d\Gamma \bar{\mathbf{u}}_1 \quad (17.25-a)$$

$$\int_{\Gamma_{I_b}} \delta \boldsymbol{\lambda}^T \mathbf{u}_2 d\Gamma = \delta \bar{\boldsymbol{\lambda}}^T \int_{\Gamma_{I_b}} \mathbf{N}_{\lambda}^T \mathbf{N}_{u_2} d\Gamma \bar{\mathbf{u}}_2 \quad (17.25-b)$$

Recognizing that

$$\mathbf{Q}_m^T = \int_{\Gamma_{I_b}} \mathbf{N}_{\lambda}^T \mathbf{N}_{u_m} d\Gamma \quad (17.26)$$

is the transpose of the operator matrix for the load vector due to surface tractions on the bonded interface defined in Equation 17.23, the weak constraint equation can be rewritten as

$$\int_{\Gamma_{I_b}} \delta \boldsymbol{\lambda}^T (\mathbf{u}_2 - \mathbf{u}_1) d\Gamma = \delta \bar{\boldsymbol{\lambda}}^T (\mathbf{Q}_2^T \bar{\mathbf{u}}_2 - \mathbf{Q}_1^T \bar{\mathbf{u}}_1) = 0 \quad (17.27)$$

Having defined the discretized form of all integrals in the governing equations, it is now possible to define the discrete system of mixed equations. Substituting Equations 17.17, 17.20, and 17.24 into Equation 17.9-17.9 and rearranging terms, the discrete Principle of Virtual Work is written as

$$\delta \bar{\mathbf{u}}_1^T (\mathbf{K}_1 \bar{\mathbf{u}}_1 - \mathbf{Q}_1 \bar{\boldsymbol{\lambda}}) = \delta \bar{\mathbf{u}}_1^T (\mathbf{f}_1 + \mathbf{f}_{c_1}) \quad (17.28-a)$$

$$\delta \bar{\mathbf{u}}_2^T (\mathbf{K}_2 \bar{\mathbf{u}}_2 + \mathbf{Q}_2 \bar{\boldsymbol{\lambda}}) = \delta \bar{\mathbf{u}}_2^T (\mathbf{f}_2 - \mathbf{f}_{c_2}) \quad (17.28-b)$$

As $\delta \bar{\mathbf{u}}_m^T$ appears in both sides of Equation 17.28-a-17.28-b, it can be eliminated, leaving

$$\mathbf{K}_1 \bar{\mathbf{u}}_1 - \mathbf{Q}_1 \bar{\boldsymbol{\lambda}} = \mathbf{f}_1 + \mathbf{f}_{c_1} \quad (17.29-a)$$

$$\mathbf{K}_2 \bar{\mathbf{u}}_2 + \mathbf{Q}_2 \bar{\boldsymbol{\lambda}} = \mathbf{f}_2 - \mathbf{f}_{c_2} \quad (17.29-b)$$

In a similar fashion, $\delta\bar{\lambda}^T$ can be eliminated from Equation 17.27, leaving

$$\mathbf{Q}_2^T \bar{\mathbf{u}}_2 - \mathbf{Q}_1^T \bar{\mathbf{u}}_1 = 0 \quad (17.30)$$

as the discrete constraint equation. The discrete system of mixed equations is defined by Equations 17.29-a-17.29-b and 17.30, which can be written in matrix form as

$$\begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & -\mathbf{Q}_1 \\ \mathbf{0} & \mathbf{K}_2 & \mathbf{Q}_2 \\ -\mathbf{Q}_1^T & \mathbf{Q}_2^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{u}}_1 \\ \bar{\mathbf{u}}_2 \\ \bar{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 + \mathbf{f}_{c_1} \\ \mathbf{f}_2 - \mathbf{f}_{c_2} \\ \mathbf{0} \end{Bmatrix} \quad (17.31)$$

17.1.4 Penalty Method Solution

The penalty method (?) was chosen for the solution of the discrete system of mixed equations because it reduces the problem to that of a single-field. Reducing the system of mixed equations to a single-field equation decreases the number of unknowns that must be solved for and simplifies the use of direct solution methods. Direct solution methods can be used with the system of mixed equations, but interlacing of the equations is required to avoid singularities (Wiberg 1974). Another troublesome aspect related to the use of direct solution methods with the system of mixed equations is that since crack propagation is simulated by the release of constraints on the interface, the total number of unknowns would change as the crack propagates. Interlacing a system of mixed equations with an ever changing number of unknowns would certainly create major bookkeeping problems in a finite element code.

To obtain the penalty form of the system of mixed equations, Equation 17.31 is rewritten as

$$\begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & -\mathbf{Q}_1 \\ \mathbf{0} & \mathbf{K}_2 & \mathbf{Q}_2 \\ -\mathbf{Q}_1^T & \mathbf{Q}_2^T & -\frac{1}{\alpha}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{u}}_1 \\ \bar{\mathbf{u}}_2 \\ \bar{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 + \mathbf{f}_{c_1} \\ \mathbf{f}_2 - \mathbf{f}_{c_2} \\ \mathbf{0} \end{Bmatrix} \quad (17.32)$$

where α is the penalty number. α should be sufficiently large that $\frac{1}{\alpha}\mathbf{I}$ is close to zero. It is now possible to express $\bar{\lambda}$ in terms of $\bar{\mathbf{u}}_1$ and $\bar{\mathbf{u}}_2$

$$\bar{\lambda} = \alpha(\mathbf{Q}_2 \bar{\mathbf{u}}_2 - \mathbf{Q}_1 \bar{\mathbf{u}}_1) \quad (17.33)$$

Substituting Equation 17.33 into Equation 17.32, a single-field penalized stiffness matrix equation is obtained

$$\begin{bmatrix} (\mathbf{K}_1 + \alpha\mathbf{Q}_1\mathbf{Q}_1^T) & -\alpha\mathbf{Q}_1\mathbf{Q}_2^T \\ -\alpha\mathbf{Q}_2\mathbf{Q}_1^T & (\mathbf{K}_2 + \alpha\mathbf{Q}_2\mathbf{Q}_2^T) \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{u}}_1 \\ \bar{\mathbf{u}}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 + \mathbf{f}_{c_1} \\ \mathbf{f}_2 - \mathbf{f}_{c_2} \end{Bmatrix} \quad (17.34)$$

The selection of a good penalty number is a rather difficult task. If the penalty number is too small the computed displacements will yield a substantial error when inserted into the constraint equation

$$\mathbf{Q}_2 \bar{\mathbf{u}}_2 - \mathbf{Q}_1 \bar{\mathbf{u}}_1 = \epsilon \gg 0 \quad (17.35)$$

As the penalty number is increased the error ϵ approaches zero, but the character of the system of equations changes as the effect of \mathbf{K}_1 and \mathbf{K}_2 is diminished. When the effect of \mathbf{K}_1 and \mathbf{K}_2 is significantly diminished the computed displacements away from the interface, which are not included in the constraint equation, will lose accuracy due to round off errors. The goal is to select a penalty number that yields an acceptable error when the computed displacements are inserted in the constraint equation without sacrificing the accuracy of the displacements away from the interface. The author's experience is that a penalty number selected using

$$\alpha = \frac{\max(\text{diag}(\mathbf{K}_m))}{\max(\text{diag}(\mathbf{Q}_m \mathbf{Q}_m^T))} \times 10^6 \quad (17.36)$$

yields very good results for the class of problems being considered. Penalty numbers selected in this fashion result in computed values of $\bar{\mathbf{u}}_1$ and $\bar{\mathbf{u}}_2$ on the interface that tend to be identical for the first five or six digits when the penalized stiffness matrix is assembled in double precision.

17.1.5 Incremental-Iterative Solution Strategy

An incremental-iterative solution strategy is used to obtain the equilibrium configuration for each crack length. At zero load, the entire interface is constrained (i.e., fully bonded). As load is applied, surface tractions on the constrained interface violate a strength criteria and the corresponding constraints are released. On that portion of the interface where constraints have been released, cohesive stresses act until the relative displacements of the unconstrained interface surfaces become large enough to dictate otherwise. In this solution strategy crack propagation occurs after every increment.

The use of a strength criteria to detect the onset of crack propagation requires that the magnitude of the applied loads be such that the surface tractions at a node on the constrained interface are precisely equal to the maximum allowable stress. In this case, equality is required between the normal surface traction and the uniaxial tensile strength. However, as the magnitude of the applied loads that causes the strength criteria to be satisfied exactly is not known *a priori*, some form of automatic load scaling must be included in the solution strategy. Assuming that the applied loads are proportional, a load factor β can be used to scale an arbitrary set of applied load vector \mathbf{f} of some arbitrary magnitude. At the beginning of each load increment i , the load factor is β_i and the applied load vector is

$$\beta_i \mathbf{f} = \beta_i \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \quad (17.37)$$

The value of β_i is zero at the beginning of the first increment. The incremental load factor for increment i is $\Delta\beta_i$ and the applied incremental load vector is

$$\Delta\beta_i \mathbf{f} = \Delta\beta_i \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \quad (17.38)$$

The load factor at the end of increment i is

$$\beta_{i+1} = \beta_i + \Delta\beta_i \quad (17.39)$$

The modified-Newton algorithm (Zienkiewicz, Taylor and Nithiarasu 2005) is used to solve for incremental displacements due to the applied incremental loads. The incremental displacements for a generic increment are defined as

$$\Delta\bar{\mathbf{u}}^{n+1} = \Delta\bar{\mathbf{u}}^n + \delta\bar{\mathbf{u}}^n \quad (17.40)$$

where

$$\bar{\mathbf{u}} = \begin{Bmatrix} \bar{\mathbf{u}}_1 \\ \bar{\mathbf{u}}_2 \end{Bmatrix} \quad (17.41)$$

and $\Delta\bar{\mathbf{u}}^n$ is the incremental displacement vector at the beginning of iteration n and $\delta\bar{\mathbf{u}}^n$ is the correction to the incremental displacement vector for iteration n . In a similar fashion, the incremental load factor is defined as

$$\Delta\beta^{n+1} = \Delta\beta^n + \delta\beta^n \quad (17.42)$$

where $\Delta\beta^n$ is the incremental load factor at the beginning of iteration n and $\delta\beta^n$ is the correction to the incremental load factor for iteration n . At the beginning of the first iteration both $\Delta\bar{\mathbf{u}}^n$ and $\Delta\beta^n$ are zero. Displacement corrections are computed by solving

$$\boxed{\mathbf{K}_\alpha d\bar{\mathbf{u}}^n = (\beta\mathbf{f} + \Delta\beta^n \mathbf{f} + d\beta^n \mathbf{f} + \mathbf{f}_c^n - \mathbf{p}^n)} \quad (17.43)$$

where

$$\mathbf{K}_\alpha = \begin{bmatrix} (\mathbf{K}_1 + \alpha\mathbf{Q}_1\mathbf{Q}_1^T) & -\alpha\mathbf{Q}_1\mathbf{Q}_2^T \\ -\alpha\mathbf{Q}_2\mathbf{Q}_1^T & (\mathbf{K}_2 + \alpha\mathbf{Q}_2\mathbf{Q}_2^T) \end{bmatrix} \quad (17.44)$$

is the penalized stiffness matrix;

$$\mathbf{f}_c^n = \begin{Bmatrix} \mathbf{f}_{c1}^n \\ -\mathbf{f}_{c2}^n \end{Bmatrix} \quad (17.45)$$

is the load vector due to cohesive stresses on the interface at the beginning of iteration n ; and

$$\mathbf{p}^n = \sum_{i=1}^{n_{elem}} \int_{\Omega_{e_i}} \mathbf{B}^T \mathbf{D}(\boldsymbol{\epsilon} + \Delta\boldsymbol{\epsilon}^n) d\Omega \quad (17.46)$$

is the reaction vector for the state of stress at iteration n . Recognizing that

$$\mathbf{r}^n = \beta\mathbf{f} + \Delta\beta^n\mathbf{f} + \mathbf{f}_c^n - \mathbf{p}^n \quad (17.47)$$

is the residual force vector at the beginning of iteration n , Equation 17.43 can be written in a more compact fashion as

$$d\bar{\mathbf{u}}^n = \mathbf{K}_\alpha^{-1}(\delta\beta^n\mathbf{f} + \mathbf{r}^n) \quad (17.48)$$

Since the $\mathbf{K}_\alpha^{-1}\mathbf{f}$ term does not change throughout the course of the iterative process it can be defined as a constant value for the increment

$$\delta\bar{\mathbf{u}}_T = \mathbf{K}_\alpha^{-1}\mathbf{f} \quad (17.49)$$

The displacement vector $\delta\bar{\mathbf{u}}_T$ is commonly called the tangent displacement vector (Crisfield, M.A. 1981). At this point, the iterative displacement correction can be defined as

$$\delta\bar{\mathbf{u}}^n = \delta\beta^n\delta\bar{\mathbf{u}}_T + \mathbf{K}_\alpha^{-1}\mathbf{r}^n \quad (17.50)$$

Having shown how the load factor is implemented within the incremental-iterative solution strategy, the last detail left to explain is the procedure for computing $\delta\beta^n$ such that the strength criteria is exactly satisfied. Since the surface tractions on the constrained interface are used to determine the magnitude of the applied load, the total surface tractions for iteration n must be expressed in terms of its various contributions

$$\bar{\boldsymbol{\lambda}}^{n+1} = \bar{\boldsymbol{\lambda}} + \Delta\bar{\boldsymbol{\lambda}}^n + \delta\bar{\boldsymbol{\lambda}}_r^n + \delta\beta^n\delta\bar{\boldsymbol{\lambda}}_T \quad (17.51)$$

where $\bar{\boldsymbol{\lambda}}$ is the surface traction vector at the beginning of the increment; $\Delta\bar{\boldsymbol{\lambda}}^n$ is the incremental surface traction vector at the beginning of iteration n ; $\delta\bar{\boldsymbol{\lambda}}_r^n$ is correction to the incremental surface traction vector due to the residual load vector \mathbf{r}^n for iteration n ; and $\delta\bar{\boldsymbol{\lambda}}_T$ is the surface traction vector due to the tangent displacement vector $\delta\bar{\mathbf{u}}_T$. $\delta\bar{\boldsymbol{\lambda}}_r$ and $\delta\bar{\boldsymbol{\lambda}}_T$ are defined as

$$\delta\bar{\boldsymbol{\lambda}}_r^n = \alpha(\mathbf{Q}_2^T \delta\bar{\mathbf{u}}_{r_2}^n - \mathbf{Q}_1^T \delta\bar{\mathbf{u}}_{r_1}^n) \quad (17.52-a)$$

$$\delta\bar{\boldsymbol{\lambda}}_T = \alpha(\mathbf{Q}_2^T \delta\bar{\mathbf{u}}_T - \mathbf{Q}_1^T \delta\bar{\mathbf{u}}_T) \quad (17.52-b)$$

The strength criteria is applied to $\bar{\boldsymbol{\lambda}}^{n+1}$ on a node-by-node basis such that

$$\max((\bar{\boldsymbol{\lambda}}^{n+1})_i(\mathbf{n})_i) = f_t \quad (17.53)$$

where $(\mathbf{n})_i$ is the normal vector at node i and f_t is the uniaxial tensile strength. Recognizing that $\bar{\boldsymbol{\lambda}}$, $\Delta\bar{\boldsymbol{\lambda}}^n$, and $\delta\bar{\boldsymbol{\lambda}}_r^n$ are fixed for iteration n , the iterative load factor correction is defined as

$$\delta\beta^n = \min \left\{ \frac{f_t - [(\bar{\boldsymbol{\lambda}})_i + (\Delta\bar{\boldsymbol{\lambda}}^n)_i + (\delta\bar{\boldsymbol{\lambda}}_r^n)_i](\mathbf{n})_i}{(\delta\bar{\boldsymbol{\lambda}}_T)_i(\mathbf{n})_i} \right\} \quad (17.54)$$

Provided that the cohesive stresses on the interface are treated as forces and no stiffness matrix is assembled for those interface elements, this solution strategy allows for load control in the post peak regime. The use of stiffness matrices for the interface elements subject to softening is avoided because their presence in the global stiffness matrix will eventually cause it to become non-positive definite.

17.2 Interface Crack Model; ICM-1 Original (MM: 8)

Chapter 6 of (Červenka, J. 1994)

This section discusses the nonlinear modeling of concrete using a discrete crack fracture mechanics based model. It addresses two important issues: mixed mode fracture in homogeneous materials and interface fracture. A new three-dimensional interface crack model is derived. The model is a generalization of classical Hillerborg's fictitious crack model, which can be recovered if shear displacements are set to zero. Several examples are used to validate the applicability of the proposed interface crack model. First, direct shear tests on mortar joints are used to test the model performance in the shear-compression regime. The more complicated combination of shear-tension is investigated using large biaxial tests of concrete-rock interfaces. The applicability to mixed mode cracking in homogeneous concrete is tested using experiments on modified Iosipescu's shear beam and anchor bolt pull-out tests.

17.2.1 Introduction

The assumption of singular stresses at the crack tip is mathematically correct only within the framework of linear elastic fracture mechanics, but physically unrealistic.

In concrete materials, a fracture process zone (Section ??) exists ahead of the crack tip. The most popular model simulating this behavior is Hillerborg's fictitious crack model (FCM) described in Section ?? and Figure ???. In a previous work, the classical FCM model was implemented by (Reich 1993) for mode I crack propagation, and extended to account for the influence of water pressure inside the crack.

The classical FCM model, Chapter 17.1, defines a relationship between normal crack opening and normal cohesive stresses, and assumes that there are no sliding displacements nor shear stresses along the process zone. This assumption is only partially valid for concrete materials. Based on experimental observations, it is indeed correct that a crack is usually initiated in pure mode I (i.e. opening mode) in concrete, even for mixed mode loading. However, during crack propagation, the crack may curve due to stress redistribution or non-proportional loading, and significant sliding displacements develop along the crack as schematically shown in Figure 17.2. Therefore, it is desirable

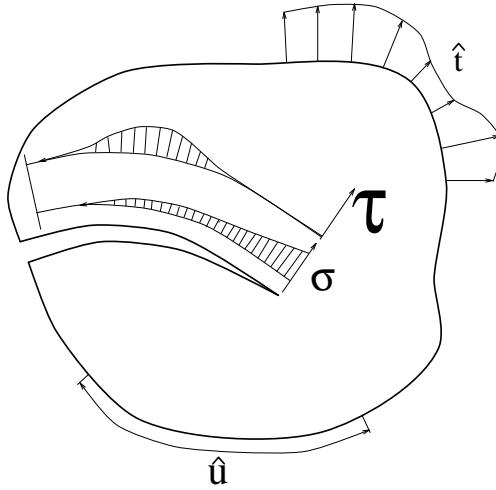


Figure 17.2: Mixed mode crack propagation.

to incorporate these shear effects into the proposed crack model.

Finally for concrete dams, it is well accepted that the weakest part of the structure is the dam-foundation interface, which is also the location of highest tensile stresses and lowest tensile strength. Given the scope of this work, as described in Chapter ??, it is necessary to address this problem.

Hence, the two major objectives of this chapter are:

- (1) Modification of the FCM model to account for shear effects along both the fracture process zone and the true crack.
- (2) Development of an interface model based on fracture mechanics to simulate cracking along rock-concrete interfaces.

The FCM model, within the framework of a discrete crack implementation, can be visualized as an interface between two identical materials. Therefore, we can develop a general model which addresses both objectives.

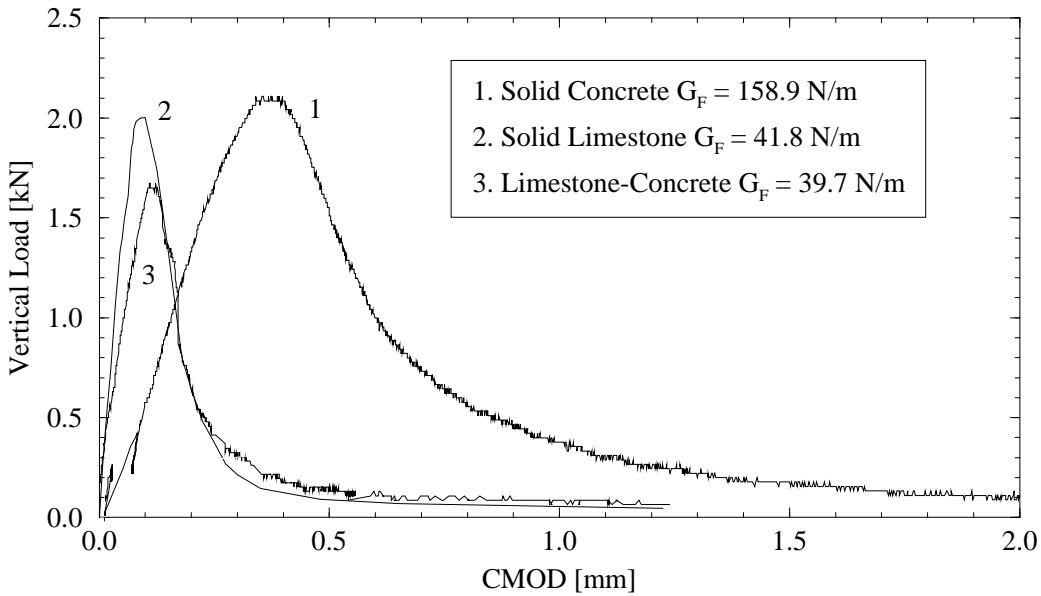


Figure 17.3: Wedge splitting tests for different materials, (V.E. et al. 1994)

Interface elements were first proposed by (Goodman, R.E. and Taylor, R.C. and Brekke, T.C. 1968) to model non-linear behavior of rock joints. Since then, numerous interface constitutive models have been proposed for a wide range of applications such as rock-joints (Goodman, R.E. and Taylor, R.C. and Brekke, T.C. 1968) masonry structures (Lotfi 1992) and concrete fracture (Stankowski 1990) (Feenstra, de Borst and Rots 1991) and (Carol, I. and Bažant, Z.P. and Prat, P.C. 1992).

In the following section an interface crack model will first be proposed, and then it will be used to simulate cracking both in homogeneous concrete and along a rock-concrete interface. The presented model is a modification of the one first proposed by (Carol, I. and Bažant, Z.P. and Prat, P.C. 1992).

17.2.2 Interface Crack Model

The objective is to develop a physically sound model, yet simple enough so that all its parameters can be easily derived from laboratory tests. The model should be capable of simulating the behavior of rock-concrete and concrete-concrete interfaces.

Experimental data (V.E. et al. 1994) on rock-concrete interfaces show (Figure 17.3) that the decrease in tensile strength is not abrupt, but is rather gradual. This is caused by the presence of the fracture process zone, along which the energy of the system is gradually dissipated.

In the present model, the rock-concrete contact is idealized as an interface between two dissimilar materials with zero thickness. Thus, the objective is to define relationships between normal and tangential stresses with opening and sliding displacements. The notation used in the interface model is illustrated in Figure 17.2.2. The major premises upon which the model is developed are:

- (1) Shear strength depends on the normal stress.
- (2) Softening is present both in shear and tension.
- (3) There is a residual shear strength due to the friction along the interface, which depends on the compressive normal stress.
- (4) Reduction in strength, i.e. softening, is caused by crack formation.
- (5) There is a zero normal and shear stiffness when the interface is totally destroyed.
- (6) Under compressive normal stresses neither the shear nor the normal stiffnesses decrease to zero. In addition, should a compressive stress be introduced in the normal direction following a full crack opening, two faces of the interface come to contact, and both tangential and normal stiffnesses become nonzero.
- (7) Irreversible relative displacements are caused by broken segments of the interface material and by friction between the two crack surfaces.
- (8) Roughness of the interface causes opening displacements (i.e. dilatancy) when subjected to sliding displacements.
- (9) The dilatancy vanishes with increasing sliding or opening displacements.

Figure 17.5 illustrates the probable character of the fracturing process along an interface.

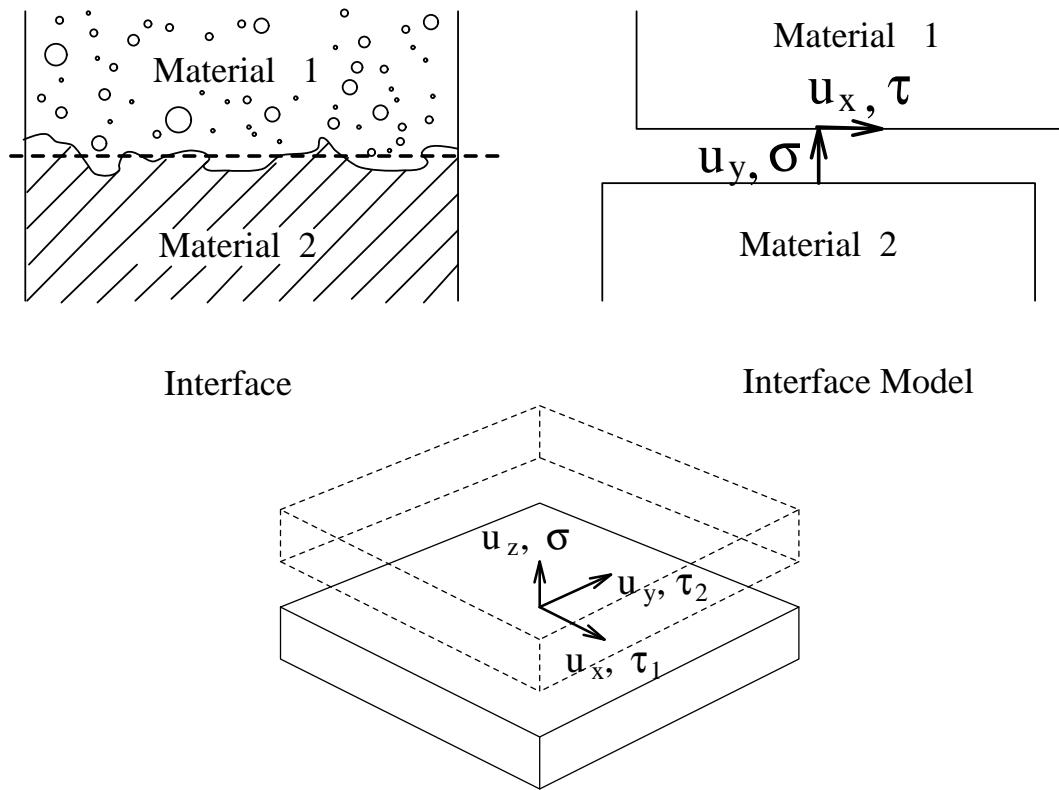


Figure 17.4: Interface idealization and notations.

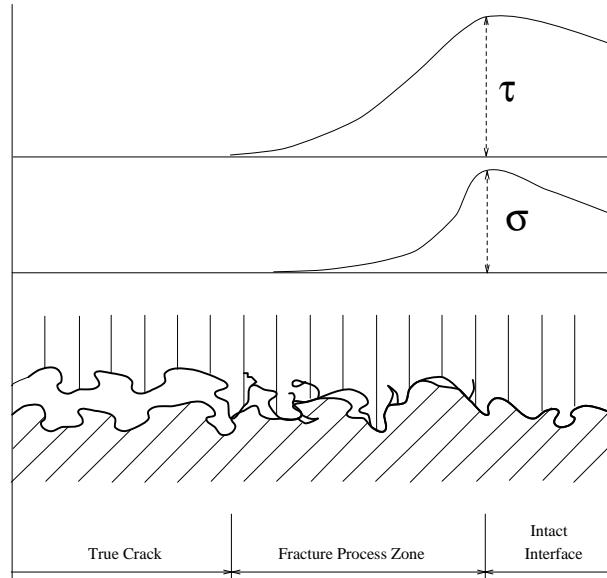


Figure 17.5: Interface fracture.

In the proposed model the strength of an interface is described by a failure function:

$$F = (\tau_1^2 + \tau_2^2) - 2c \tan(\phi_f)(\sigma_t - \sigma) - \tan^2(\phi_f)(\sigma^2 - \sigma_t^2) = 0 \quad (17.55)$$

where:

- c is the cohesion.
- ϕ_f is the angle of friction.
- σ_t is the tensile strength of the interface.
- τ_1 and τ_2 are the two tangential components of the interface traction vector.
- σ is the normal traction component.

The shape of the failure function in two-dimensional case is shown in Figure 17.6, and it corresponds to the failure criteria first proposed by (Carol, I. and Bažant, Z.P. and Prat, P.C. 1992). The general three-dimensional failure function is obtained by mere rotation around the σ -axis.

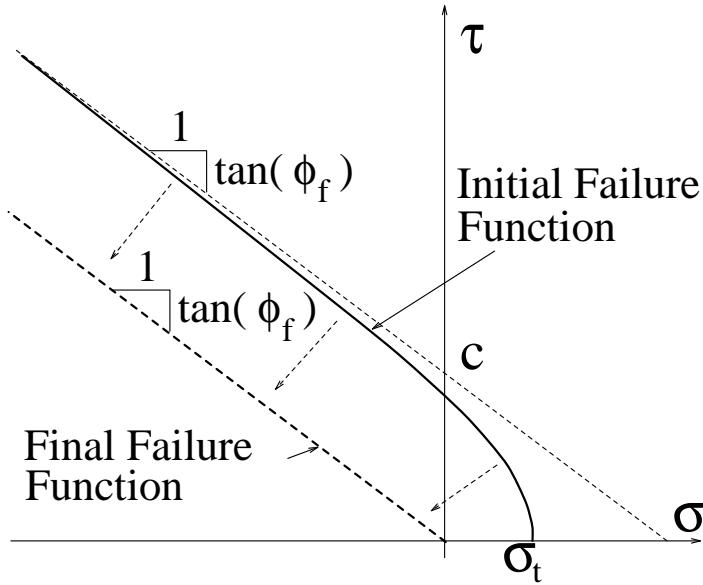


Figure 17.6: Failure function.

The evolution of the failure function is based on a softening parameter u^{eff} which is the norm of the inelastic displacement vector \mathbf{u}^i . The inelastic displacement vector is obtained by decomposition of the displacement vector \mathbf{u} into an elastic part \mathbf{u}^e and an inelastic part \mathbf{u}^i . The inelastic part can subsequently be decomposed into plastic (i.e. irreversible) displacements \mathbf{u}^p and fracturing displacements \mathbf{u}^f . The plastic displacements are assumed to be caused by friction between crack surfaces and the fracturing displacements by the formation of microcracks.

$$\begin{aligned} F &= F(c, \sigma_t, \phi_f), \quad c = c(u^{\text{eff}}), \quad \sigma_t = \sigma_t(u^{\text{eff}}) \\ \mathbf{u} &= \mathbf{u}^e + \mathbf{u}^i, \quad \mathbf{u}^i = \mathbf{u}^p + \mathbf{u}^f \\ u^{\text{eff}} &= \|\mathbf{u}^i\| = (u_x^{i2} + u_y^{i2} + u_z^{i2})^{1/2} \end{aligned} \quad (17.56)$$

In this work both linear and bilinear relationship are used for $c(u^{\text{eff}})$ and $\sigma_t(u^{\text{eff}})$.

$$\left. \begin{aligned} c(u^{\text{eff}}) &= c_0 \left(1 - \frac{u^{\text{eff}}}{w_c}\right) & \forall u^{\text{eff}} < w_c \\ c(u^{\text{eff}}) &= 0 & \forall u^{\text{eff}} \geq w_c \\ w_c &= \frac{2G_F^{IIa}}{c_0} \end{aligned} \right\} \text{linear for cohesion} \quad (17.57)$$

$$\left. \begin{aligned} c(u^{\text{eff}}) &= c_0 + u^{\text{eff}} \frac{s_{1c} - c_0}{w_{1c}} & \forall u^{\text{eff}} < w_{1c} \\ c(u^{\text{eff}}) &= s_c \left(1 - \frac{u^{\text{eff}} - w_{1c}}{w_c - w_{1c}}\right) & \forall u^{\text{eff}} \in (w_{1c}, w_c) \\ c(u^{\text{eff}}) &= 0 & \forall u^{\text{eff}} > w_c \\ w_c &= \frac{2G_F^{IIa} - c_0 w_{1c}}{s_{1c}} \end{aligned} \right\} \text{bi-linear for cohesion}$$

$$\left. \begin{array}{l}
 \sigma_t(u^{\text{ieff}}) = \sigma_{t0}(1 - \frac{u^{\text{ieff}}}{w_\sigma}) \quad \forall u^{\text{ieff}} < w_\sigma \\
 \sigma_t(u^{\text{ieff}}) = 0 \quad \forall u^{\text{ieff}} \geq w_{\sigma_t} \\
 w_\sigma = \frac{2G_F^I}{\sigma_{t0}}
 \end{array} \right\} \text{linear for tensile strength} \quad (17.58)$$

$$\left. \begin{array}{l}
 \sigma_t(u^{\text{ieff}}) = \sigma_{t0} + u^{\text{ieff}} \frac{s_{1\sigma} - \sigma_{t0}}{w_{1\sigma}} \quad \forall u^{\text{ieff}} < w_{1\sigma} \\
 \sigma_t(u^{\text{ieff}}) = s_{1\sigma}(1 - \frac{u^{\text{ieff}} - w_{1\sigma}}{w_{\sigma_t} - w_{1\sigma}}) \quad \forall u^{\text{ieff}} \in (w_{1\sigma}, w_\sigma) \\
 \sigma_t(u^{\text{ieff}}) = 0 \quad \forall u^{\text{ieff}} > w_\sigma \\
 w_\sigma = \frac{2G_F^I - \sigma_{t0} w_{1\sigma}}{s_{1\sigma}}
 \end{array} \right\} \text{bi-linear for tensile strength}$$

where G_F^I and G_F^{IIa} are mode I and II fracture energies. s_{1c} , w_{1c} and $s_{1\sigma}$, $w_{1\sigma}$ are the coordinates of the breakpoint in the bi-linear softening laws for cohesion and tensile strength respectively. The critical opening and sliding corresponding to zero cohesion and tensile strength are denoted by w_σ and w_c respectively, and they are determined from the condition that the area under the linear or bilinear softening law must be equal to G_F^I and G_F^{IIa} respectively. The significance of these symbols can be best explained through Figure 17.7. It should be noted that G_F^{IIa} is not

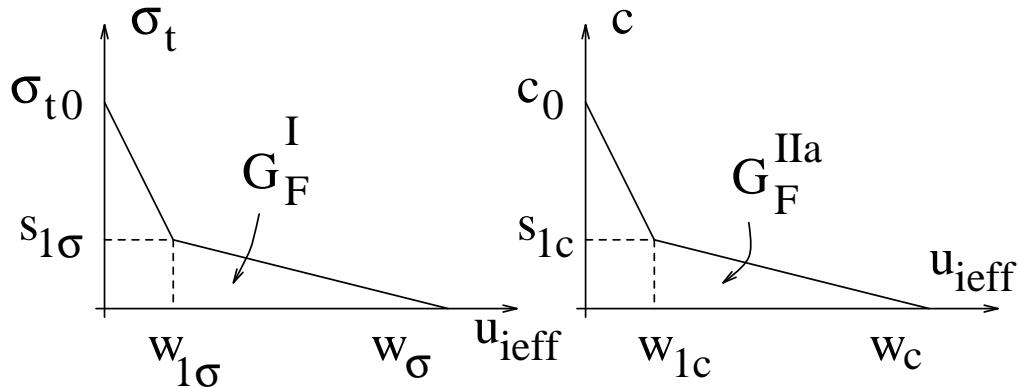


Figure 17.7: Bi-linear softening laws.

the pure mode II fracture energy (i.e. the area under a $\tau-u_x$ curve), but rather is the energy dissipated during a shear test with high confining normal stress. This parameter was first introduced by (Carol, I. and Bažant, Z.P. and Prat, P.C. 1992) in their microplane model. This representation seems to be more favorable to the pure mode II fracture energy G_F^{II} . The determination of G_F^{II} would require a pure shear test without confinement, which is extremely difficult to perform. Alternatively, a G_F^{IIa} test requires a large normal confinement, and is therefore easier to accomplish. Furthermore, if G_F^{II} is used, the whole shear-compression region of the interface model would be an extrapolation from the observed behavior, whereas the second approach represents an interpolation between the upper bound G_F^{IIa} and the lower bound G_F^I .

The residual shear strength is obtained from the failure function by setting both c and σ_t equal to 0, which corresponds to the final shape of the failure function in Figure 17.6 and is given by:

$$\tau_1^2 + \tau_2^2 = \tan^2(\phi_f) \sigma^2 \quad (17.59)$$

Stiffness degradation is modeled through a damage parameter, $D \in \langle 0, 1 \rangle$, which is a relative measure of the fractured surface. Thus, D is related to the secant of the normal stiffness K_{ns} in the uniaxial case:

$$D = \frac{A_f}{A_o} = 1 - \frac{K_{ns}}{K_{no}} \quad (17.60)$$

where K_{no} is the initial normal stiffness of the interface; A_o and A_f are the total interface area and the fractured area respectively. It is assumed, that the damage parameter D can be determined by converting the mixed mode problem into an equivalent uniaxial one (Figure 17.8). In the equivalent uniaxial problem the normal inelastic displacement is set equal to u^{ieff} . Then, the secant normal stiffness can be determined from:

$$K_{ns} = \frac{\sigma}{u - u^p} = \frac{\sigma_t(u^{\text{ieff}})}{u^e + u^p + u^f - u^p} = \frac{\sigma_t(u^{\text{ieff}})}{\sigma_t(u^{\text{ieff}})/K_{no} + (1 - \gamma)u^{\text{ieff}}} \quad (17.61)$$

where γ is the ratio of irreversible inelastic normal displacement to the total value of inelastic displacement. Experimentally, γ can be determined from a pure mode I test through:

$$\gamma = \frac{u^p}{u^i} \quad (17.62)$$

where u^p is the residual displacement after unloading and u^i is the inelastic displacement before unloading. (Figure 17.8). For concrete, γ is usually assumed equal to 0.2 (Dahlblom and Ottosen 1990) or 0.3 (Alvaredo and

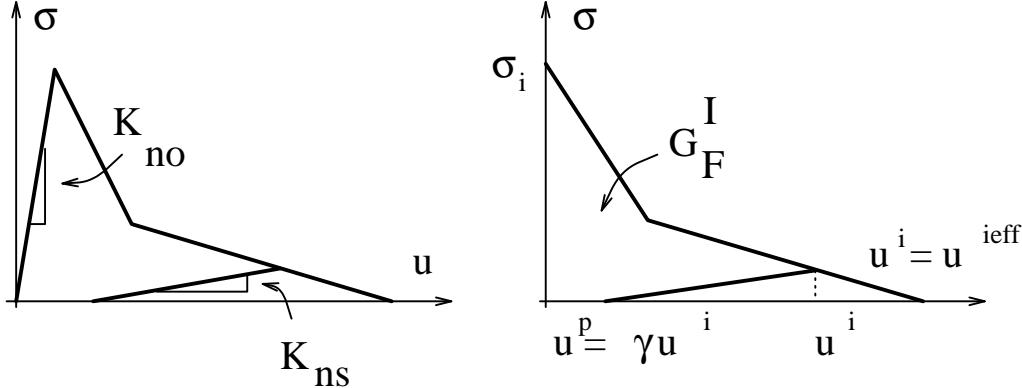


Figure 17.8: Stiffness degradation in the equivalent uniaxial case.

Wittman 1992). Then, the evolution of the damage parameter D is defined by formula:

$$D = 1 - \frac{\sigma_t(u^{i\text{eff}})}{\sigma_t(u^{i\text{eff}}) + (1 - \gamma)u^{i\text{eff}}K_{no}} \quad (17.63)$$

which is obtained by substituting Equation 17.61 into Eq. 17.60.

The stress-displacement relationship of the interface is expressed as:

$$\boldsymbol{\sigma} = \alpha \mathbf{E}(\mathbf{u} - \mathbf{u}^p) \quad (17.64)$$

where: (a) $\boldsymbol{\sigma}$ is the vector of tangential and normal stress at the interface.

$$\boldsymbol{\sigma} = \{\tau_1, \tau_2, \sigma\}^T \quad (17.65)$$

(b) α is the integrity parameter defining the relative active area of the interface, and it is related to the damage parameter D .

$$\alpha = 1 - \frac{|\sigma| + \sigma}{2|\sigma|} D \quad (17.66)$$

It should be noted that α can be different from 1 only if the normal stress σ is positive (i.e. the interface is in tension). In other words, the damage parameter D is activated only if the interface is in tension. In compression, the crack is assumed to be closed, and there is full contact between the two crack surface. The activation of D is controlled through the fraction $\frac{|\sigma| + \sigma}{2|\sigma|}$, which is equal to one if σ is positive, and is zero otherwise.

(c) \mathbf{E} is the elastic stiffness matrix of the interface.

$$\mathbf{E} = \begin{bmatrix} K_{to} & 0 & 0 \\ 0 & K_{to} & 0 \\ 0 & 0 & K_{no} \end{bmatrix} \quad (17.67)$$

It should be noted, that the off-diagonal terms in the elastic stiffness matrix \mathbf{E} of the interface are all equal to zero, which implies that no dilatancy is considered in the elastic range. The dilatancy is introduced later after the failure limit has been reached through the iterative solution process. The dilatancy of the interface is given by dilatancy angle ϕ_d , which is again assumed to be a function of $u^{i\text{eff}}$. In the proposed model, a linear relationship is assumed:

$$\begin{aligned} \phi_d(u^{i\text{eff}}) &= \phi_{d0}(1 - \frac{u^{i\text{eff}}}{u_{dil}}) & \forall u^{i\text{eff}} \leq u_{dil} \\ \phi_d(u^{i\text{eff}}) &= 0 & \forall u^{i\text{eff}} > u_{dil} \end{aligned} \quad (17.68)$$

where u_{dil} is the critical relative displacement after which, the interface does not exhibit the dilatancy effect any more, and ϕ_{d0} is the initial value of the dilatancy angle.

17.2.2.1 Relation to fictitious crack model.

It is possible to prove that the proposed interface crack model (ICM) reduces to Hillerborg's fictitious crack model in the case of zero sliding displacements.

PROOF 17.1 (FCM a special case of ICM.) We assume that all shear displacements are zero. Then, the interface stresses develop only along the σ -axis in the $\sigma \times \tau_1 \times \tau_2$ space (Figure 17.6). After the tensile strength σ_t is reached, softening starts, and the stress in the interface is given by:

$$\sigma = \sigma(u_z^i) \quad (17.69)$$

Normal traction σ is now a function of the normal inelastic displacement u_z^i only, since for zero sliding displacements, u^{eff} is equivalent to u_z^i . The total opening u_z of the interface is given by:

$$u_z = \frac{\sigma(u_z^i)}{K_{no}} + u_z^i$$

If the limiting case of K_{no} equal to infinity is considered, then u_z^i becomes equivalent to u_z , and the normal stress in Equation 17.69 becomes a function of the interface opening only:

$$\lim_{K_{no} \rightarrow \infty} \sigma = \sigma(u_z^i) = \sigma(u_z) = \sigma(COD) \quad (17.70)$$

which is precisely the definition of Hillerborg's fictitious crack model.

17.2.3 Finite Element Implementation

The finite element implementation of the interface crack model previously presented will be discussed in this section. The implementation of a nonlinear model into a finite element code consists of three major subtasks:

1. Interface element formulation.
2. Constitutive driver for the computation of internal forces.
3. Non-linear solution algorithm on the structural level.

17.2.3.1 Interface element formulation.

Standard interface elements are used in this work. The element stiffness matrix is computed using the well known relation:

$$\mathbf{K}^e = \int_{A_e} \mathbf{B}^T \mathbf{E} \mathbf{B} dA \quad (17.71)$$

where \mathbf{E} is the interface material stiffness matrix, given by Equation 17.67, and \mathbf{B} is the matrix relating element nodal displacements \mathbf{u}_e to slidings and openings along the interface:

$$\mathbf{u} = \sum_i^{1/2N_{en}} N_i(\bar{\mathbf{u}}_i^+ - \bar{\mathbf{u}}_i^-) = \mathbf{B} \mathbf{u}_e \quad (17.72)$$

where $\bar{\mathbf{u}}_i^+$ and $\bar{\mathbf{u}}_i^-$ denote the element nodal displacements in the local coordinate system of the interface on the upper and lower interface surface respectively. Given this definition, matrix \mathbf{B} is equal to:

$$\mathbf{B} = [-\mathbf{B}_1 \mathbf{T}, \quad \dots, \quad -\mathbf{B}_{p+1} \mathbf{T}, \quad +\mathbf{B}_1 \mathbf{T}, \quad \dots, \quad +\mathbf{B}_{p+1} \mathbf{T}] \quad (17.73)$$

where submatrix \mathbf{B}_i is a diagonal matrix of shape functions $N_i(\zeta, \eta)$ corresponding to node i . In three-dimensional case it has the form:

$$\mathbf{B}_i = \begin{bmatrix} N_i(\zeta, \eta) & 0 & 0 \\ 0 & N_i(\zeta, \eta) & 0 \\ 0 & 0 & N_i(\zeta, \eta) \end{bmatrix} \quad (17.74)$$

and in two-dimensional case it is given by:

$$\mathbf{B}_i = \begin{bmatrix} N_i(\zeta) & 0 \\ 0 & N_i(\zeta) \end{bmatrix} \quad (17.75)$$

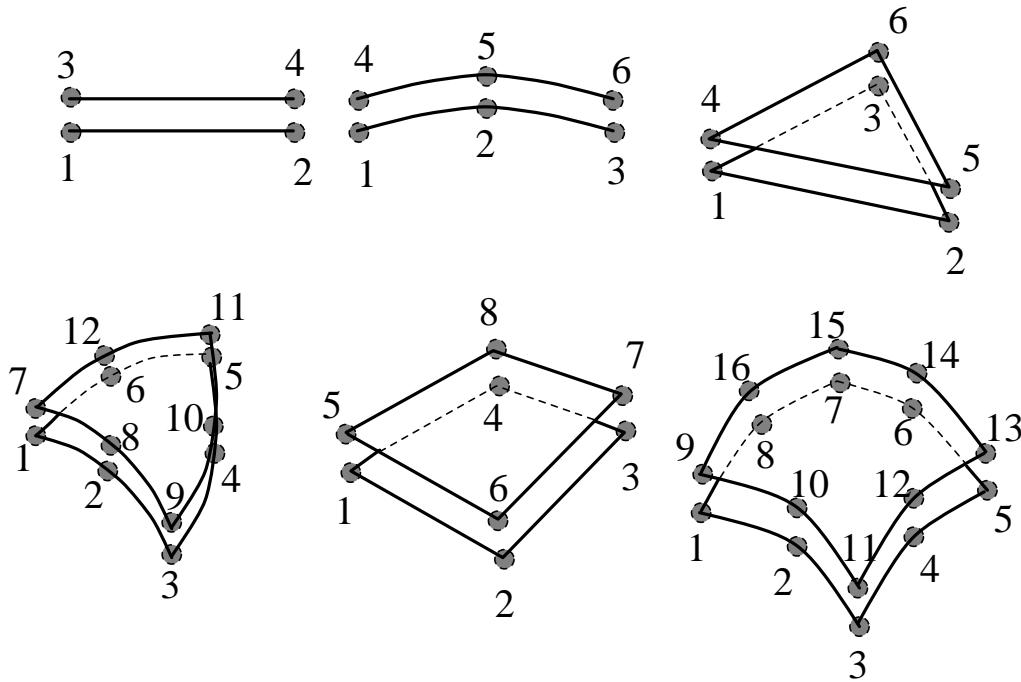


Figure 17.9: Interface element numbering.

Subscript i is a node numbering index on one element surface ranging from 1 to $\frac{N_{en}}{2}$, where N_{en} is the total number of element nodes, p is the order of the interface element, and is equal to $(\frac{N_{en}}{2} - 1)$. Finally, ζ and η are the natural coordinates of the interface element.

This definition of matrix \mathbf{B} corresponds to the element numbering shown in Figure 17.9 for several two- and three-dimensional interface elements. The transformation from global to local coordinate system of the interface element is accomplished through the transformation matrix \mathbf{T} , which in general three-dimensional case is:

$$\mathbf{T} = \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \mathbf{v}_3^T \end{bmatrix} \quad (17.76)$$

The rows of the transformation matrix \mathbf{T} are formed by vectors \mathbf{v}_i defined by following formulas:

$$\mathbf{v}_1 = \frac{\frac{\partial \mathbf{x}}{\partial \zeta}}{\left\| \frac{\partial \mathbf{x}}{\partial \zeta} \right\|}, \quad \mathbf{v}_3 = \frac{\frac{\partial \mathbf{x}}{\partial \zeta} \times \frac{\partial \mathbf{x}}{\partial \eta}}{\left\| \frac{\partial \mathbf{x}}{\partial \zeta} \times \frac{\partial \mathbf{x}}{\partial \eta} \right\|}, \quad \mathbf{v}_2 = \mathbf{v}_3 \times \mathbf{v}_1, \quad (17.77)$$

The two-dimensional case can be recovered from the two preceding formulas by deleting the last row in matrix \mathbf{T} and considering the following definition of vectors \mathbf{v}_i .

$$\mathbf{v}_1 = \frac{\frac{\partial \mathbf{x}}{\partial \zeta}}{\left\| \frac{\partial \mathbf{x}}{\partial \zeta} \right\|}, \quad \mathbf{v}_2 = \{-v_{1y}, v_{1x}\} \quad (17.78)$$

Local coordinate systems defined by these transformations are shown in Figure 17.10.

17.2.3.2 Constitutive driver.

The mathematical theory of plasticity is used in the development of the constitutive driver for the interface crack model. On the constitutive level in the sense of finite element implementation, the problem can be stated as follows:

For a given stress state $\boldsymbol{\sigma}_n$, softening parameter u_n^{eff} and displacement increment $\Delta \mathbf{u}_n$, determine a new stress state $\boldsymbol{\sigma}_{n+1}$ and corresponding value of softening parameter u_{n+1}^{eff} . In both states n and $n + 1$, the failure criterion must be satisfied:

$$F_n(\boldsymbol{\sigma}_n, u_n^{\text{eff}}) = 0 \quad \wedge \quad F_{n+1}(\boldsymbol{\sigma}_{n+1}, u_{n+1}^{\text{eff}}) = 0 \quad (17.79)$$

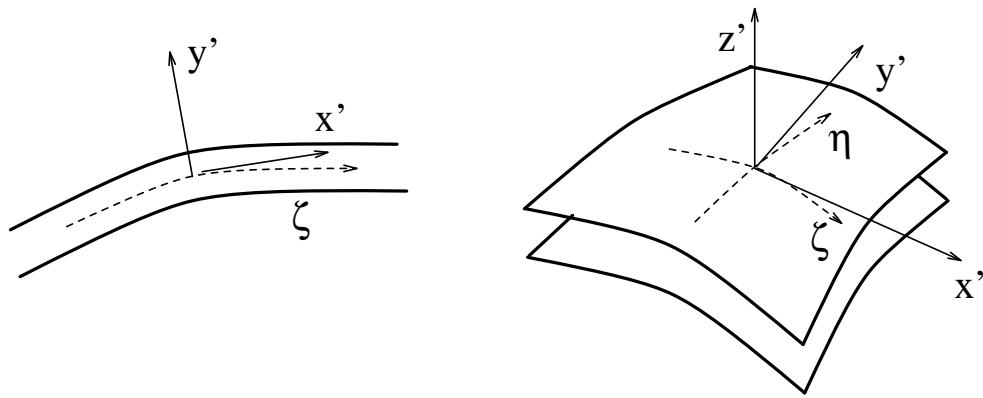


Figure 17.10: Local coordinate system of the interface element.

These two conditions are equivalent to an incremental form of the consistency condition (Equation ??):

$$\Delta F = F_{n+1} - F_n = 0 \quad (17.80)$$

Because the failure function is assumed to be satisfied for state n , it is necessary to also ensure the satisfaction of the failure function at state $n + 1$. In this work, plasticity theory is used to describe the evolution of the failure function based on the softening parameter u^{eff} , which is the euclidean norm of the inelastic displacement vector. The inelastic displacements are subsequently decomposed according to Equation 17.62. Thus, plastic and fracturing effects can be separated.

The elastic predictor is given by:

$$\boldsymbol{\sigma}_e = \boldsymbol{\sigma}_n + \mathbf{E} \Delta \mathbf{u}_n \quad (17.81)$$

where $\boldsymbol{\sigma}_e$ are the trial tractions outside the failure surface if a totally elastic behavior is considered. The inelastic corrector returns the trial stress state back to the failure surface:

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_e - \Delta \lambda \mathbf{E} \mathbf{m} \quad (17.82)$$

where $\Delta \lambda$ is the inelastic multiplier and \mathbf{m} is the direction of the inelastic displacements. Inelastic multiplier $\Delta \lambda$ is determined from the failure condition at state $n + 1$.

$$F_{n+1}(\boldsymbol{\sigma}_e - \Delta \lambda \mathbf{E} \mathbf{m}, u_{n+1}^{\text{eff}}) = 0 \quad (17.83)$$

In the three dimensional space $\sigma \times \tau_1 \times \tau_2$, the geometrical interpretation of this condition is the determination of an intersection of a line emanating from point $\boldsymbol{\sigma}_e$ in the direction $\mathbf{E} \mathbf{m}$ with the moving failure surface (Figure 17.11). The failure surface, $F = 0$, expands or shrinks depending on the softening introduced through u^{eff} . This is schematically shown in Figure 17.11 for a two-dimensional case. The increment of the plastic multiplier $\Delta \lambda$ is computed by solving a quadratic equation obtained by considering the particular form of the failure function 17.55 in Equation 17.83.

For this case, the failure function is equal to:

$$F = (\tau_{1,n+1}^2 + \tau_{2,n+1}^2) - 2c \tan(\phi_f)(\sigma_t - \sigma_{n+1}) - \tan^2(\phi_f)(\sigma_{n+1}^2 - \sigma_t^2) = 0 \quad (17.84)$$

To this equation, we substitute the expression for the new stress state $\boldsymbol{\sigma}_{n+1}$, which are equal to (Equation 17.82):

$$\begin{aligned} \tau_{1,n+1} &= \tau_{1,e} - \Delta \lambda K_{t,o} m_1 &= \tau_{1,e} - \Delta \lambda l_1 \\ \tau_{2,n+1} &= \tau_{2,e} - \Delta \lambda K_{t,o} m_2 &= \tau_{2,e} - \Delta \lambda l_2 \\ \sigma_{n+1} &= \sigma_e - \Delta \lambda K_{n,o} m_3 &= \sigma_e - \Delta \lambda l_3 \end{aligned} \quad (17.85)$$

The result of this substitution is a quadratic equation with roots:

$$\Delta \lambda_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (17.86)$$

where

$$\begin{aligned} A &= l_1^2 + l_2^2 - \mu^2 l_3^2 \\ B &= 2\mu^2 \sigma_e l_3 - 2l_1 \tau_{1,e} - 2l_2 \tau_{2,e} - 2c\mu l_3 \\ C &= \tau_{1,e}^2 + \tau_{2,e}^2 - 2c\mu(\sigma_t - \sigma_e) - \mu^2(\sigma_e^2 - \sigma_t^2) \end{aligned}$$

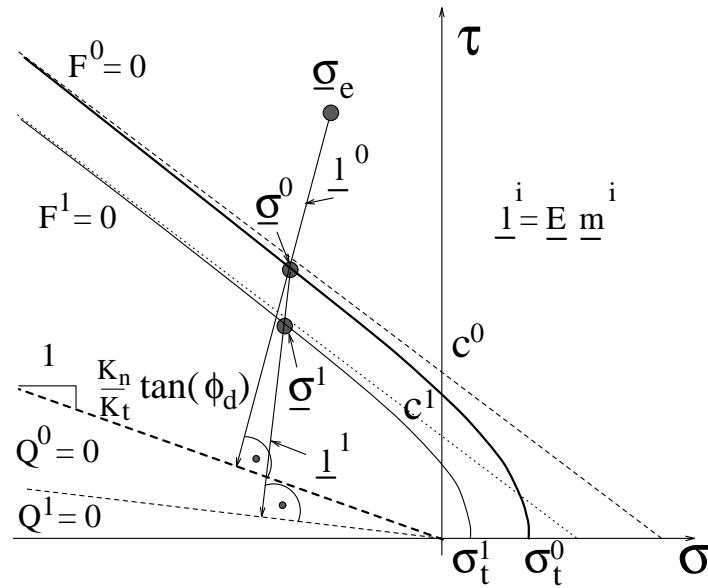


Figure 17.11: Algorithm for interface constitutive model.

The required solution must satisfy the following conditions.

$$\Delta\lambda > 0 \quad \wedge \quad \Delta\lambda = \min(\Delta\lambda_1, \Delta\lambda_2) \quad (17.87)$$

In the previous equations, l_1 , l_2 and l_3 are components of vector \mathbf{l} indicating the direction of inelastic return in the stress space, and they are related to the direction of inelastic displacements \mathbf{m} through the stiffness matrix \mathbf{E} .

$$\mathbf{l} = \mathbf{Em} \quad (17.88)$$

The direction of inelastic displacements \mathbf{m} is defined as the normal vector to the plastic potential Q (Figure 17.11), which is defined using the dilatancy angle $\phi_d(u^{\text{eff}})$ as:

$$Q = \tau_1^2 + \tau_2^2 - \left(\frac{K_n}{K_t} \tan \phi_d \right)^2 \sigma^2 = 0 \quad (17.89)$$

For the definition of \mathbf{m} , we must distinguish between the case, when the return direction \mathbf{m} can be determined on the basis of Q , and the pathological case of the apex of Q , when the normal \mathbf{m} cannot be constructed. For this case, \mathbf{m} is defined by connecting the trial tractions σ_e with the origin of the $\sigma \times \tau_1 \times \tau_2$ space (Figure 17.12):

$$\begin{aligned} \mathbf{m} &= \begin{cases} \tau_1/K_{to} \\ \tau_2/K_{to} \\ \sigma/K_{no} \end{cases} && \text{if } \frac{\|\tau\|}{\sigma} \leq \frac{1}{\tan \phi_d} \frac{K_{to}}{K_{no}} \wedge \sigma > 0 \\ \mathbf{m} &= \begin{cases} \tau_1 \\ \tau_2 \\ \sqrt{\tau_1^2 + \tau_2^2} \tan \phi_d \end{cases} && \text{otherwise} \end{aligned} \quad (17.90)$$

At this stage, we can identify three major steps to the proposed algorithm:

1. Elastic predictor:

$$\sigma_e = \sigma_n + \mathbf{E}\Delta\mathbf{u}_n \quad (17.91)$$

2. Inelastic corrector simultaneously satisfying the following two equations:

$$\begin{aligned} F_{n+1}(\sigma_e - \Delta\lambda \mathbf{Em}, u_{n+1}^{\text{eff}}) &= 0 \\ u_{n+1}^{\text{eff}} &= u_n^{\text{eff}} + \|\Delta\lambda \mathbf{m}\| \end{aligned} \quad (17.92)$$

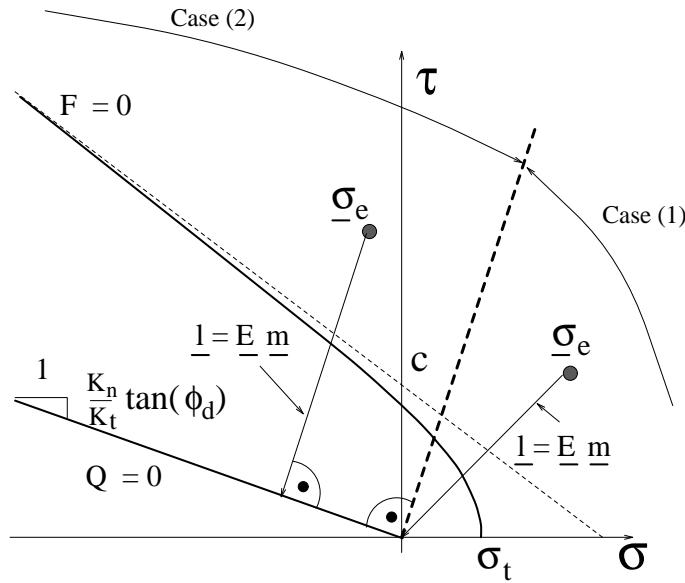


Figure 17.12: Definition of inelastic return direction.

3. Fracturing corrector:

$$\begin{aligned} \mathbf{E}_s &= \alpha \mathbf{E} \\ \mathbf{u}^p &= \mathbf{u} - \mathbf{E}_s^{-1} \boldsymbol{\sigma}_{n+1} \end{aligned} \tag{17.93}$$

In the fracturing corrector, the inelastic displacements due to friction and microcracks development are separated. This separation is controlled by the damage parameter D defined by Equation 17.63. The evolution of damage parameter D is defined by converting the mixed mode problem into an equivalent uniaxial case as described in Section 17.2.2.

The complete algorithm of the interface constitutive driver is described in Algorithm 17.1 and is shown schematically on Figure 17.11.

ALGORITHM 17.1 (ICM constitutive driver.)

- Input: σ_n , u_n^{ieff} and Δu_n
- $\sigma_{n+1} = \sigma_n + \alpha E \Delta u_n$
- if $F(\sigma_{n+1}, u_n^{ieff}) > 0$
 - Update σ_n , and Δu_n such that $F(\sigma_n, u_n^{ieff}) = 0$.
 - Elastic predictor: $\sigma_{n+1} = \sigma_n + E \Delta u_n$
 - Inelastic corrector:
 - * $u_{n+1}^{ieff} = u_n^{ieff}$
 - * Do
 - Evaluate return direction m
 - Determine $d\lambda$ such that $F(\sigma_{n+1} - d\lambda Em, u_{n+1}^{ieff}) = 0$
 - $u_{n+1}^{ieff} = u_{n+1}^{ieff} + ||d\lambda m||$
 - $\sigma_{n+1} = \sigma_{n+1} - d\lambda Em$
 - * While $d\lambda < \varepsilon$
 - Fracturing corrector:

$$\alpha = 1 - \frac{|\sigma| + \sigma}{2|\sigma|} D(u_{n+1}^{ieff})$$

$$E_s = \alpha E$$

$$u^p = u - E_s^{-1} \sigma_{n+1}$$
- Output: σ_{n+1} , u_{n+1}^{ieff}

17.2.3.3 Non-linear solver.

The proposed interface crack model is clearly a nonlinear material formulation, and therefore, a finite analysis including this material formulation involves a system of nonlinear equations. Such system can be solved, for instance, by the Newton-Raphson method. To exploit the full Newton-Raphson method a tangent stiffness matrix would have to be computed at each iteration. The incremental tangent stiffness matrix for the proposed material formulation can be computed from the incremental stress-displacement relationship:

$$\Delta \sigma_n = E \Delta u_n - \Delta \lambda Em \quad (17.94)$$

when multiply the last term by a fraction which is equal to unity:

$$\Delta \sigma_n = E \Delta u_n - \Delta \lambda Em \frac{n^T E \Delta u_n}{n^T E \Delta u_n} \quad (17.95)$$

where n is the normal vector to the failure surface passing through the trial stress state σ_e (Equation 17.81). From this equation it is possible to derive a formula for an incremental tangent material stiffness matrix E_T :

$$\Delta \sigma_n = E_T \Delta u_n \quad (17.96)$$

where:

$$E_T = E \left(I - \Delta \lambda \frac{Em n^T E}{n^T E \Delta u_n} \right) \quad (17.97)$$

In this particular case, the new stress state is computed using the iterative process described in Algorithm 17.1. Therefore, the incremental stress-displacement is given by a sum:

$$\Delta \sigma_n = E \Delta u_n - \sum_{i=1}^{N_{iter}} (\Delta \lambda_i Em_i) \quad (17.98)$$

where N_{iter} is the number of iterations in the inelastic corrector part of Algorithm 17.1. Following similar arguments leading to equation 17.97, the incremental tangent stiffness is computed by the following expression:

$$E_T = E \left[I - \sum_{i=1}^{N_{iter}} \left(\Delta \lambda_i \frac{Em_i n_i^T E}{n_i^T E \Delta u_n} \right) \right] \quad (17.99)$$

During softening, the tangent matrix \mathbf{E}_T becomes negative. In addition, the matrix becomes also unsymmetric due to the dilatancy, which is introduced in the softening regime of the interface model. This would imply the need to store the full stiffness matrix on the structural level, and a method for solving unsymmetric and non-positive system of equations would have to be adopted. This is clearly not an efficient approach, since only few elements will be affected by the non-linear behavior (i.e. interface elements), and therefore, only small portions of the structural stiffness matrix will be unsymmetric.

On the other hand, it can be expected that the initial stiffnesses of the interface elements are very large, and in some cases, they represent penalty numbers modeling a rigid contact. This means that it is not possible to use the initial structural stiffness throughout the whole iterative process, as it would result in an excessive number of iterations.

In this work two approaches are suggested to mitigate this problem:

(1) Use of secant-Newton method to accelerate the convergence on the structural level.

(2) Use of secant interface stiffness on the element level while preserving its positiveness and symmetry.

Both methods are supplemented with the line-search technique (Crisfield 1991).

17.2.3.4 Secant-Newton method.

The secant-Newton method is described in detail in (Crisfield 1991). In this method, it is not necessary to recompute the structural stiffness matrix at each iteration, but rather the vector of iterative displacement corrections is updated to satisfy the secant relationship.

$$\frac{d\mathbf{u}_i^*}{r_i} = \frac{d\mathbf{u}_{i-1}^*}{r_i - r_{i-1}} \quad (17.100)$$

For one-dimensional case, the meaning of this formula is illustrated by Figure 17.13. In this work, (Davidon, W.C.

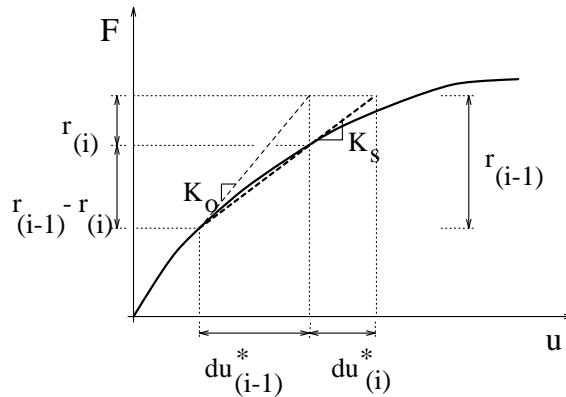


Figure 17.13: Secant relationship.

1968) rank-one quasi-Newton update is used, and the corrected iterative update of the displacement vector in iteration i is equal to:

$$d\mathbf{u}_i^* = Ad\mathbf{u}_i + Bd\mathbf{u}_{i-1}^* + Cd\mathbf{u}_{i-1} \quad (17.101)$$

where $d\mathbf{u}_i$ is the iterative update of the displacement vector computed in iteration i by solving:

$$d\mathbf{u}_i = \mathbf{K}^{-1}\mathbf{r}_i \quad (17.102)$$

where \mathbf{K} is the structural stiffness matrix, and \mathbf{r}_i are residual forces at iteration i . The stared symbols, \mathbf{u}_i^* and \mathbf{u}_{i-1}^* , represent the displacement vector updates based on the secant-Newton corrections (Equation 17.101), and coefficients A , B and C are given by (Davidon, W.C. 1968):

$$C = \frac{(d\mathbf{u}_{i-1}^* + d\mathbf{u}_i - d\mathbf{u}_{i-1})^T \mathbf{r}_i}{(d\mathbf{u}_{i-1}^* + d\mathbf{u}_i - d\mathbf{u}_{i-1})^T (\mathbf{r}_i - \mathbf{r}_{i-1})} \quad (17.103)$$

$$A = 1 - C, \quad B = -C$$

17.2.3.5 Element secant stiffness.

It is also possible to employ the secant formula (Eq. 17.100) on the element level. Considering the diagonal form of the material stiffness matrix \mathbf{E} , it is possible to determine its secant form from the stress and displacement corrections in each iteration.

$$\begin{aligned} K_{t1}^i &= \frac{\tau_{1n+1}^i - \tau_1^{i-1}}{\Delta u_{x,n+1}^i - \Delta u_x^{i-1}} \\ K_{t2}^i &= \frac{\tau_{2n+1}^i - \tau_2^{i-1}}{\Delta u_{y,n+1}^i - \Delta u_y^{i-1}} \\ K_n^i &= \frac{\sigma_{n+1}^i - \sigma_{n+1}^{i-1}}{\Delta u_{z,n+1}^i - \Delta u_z^{i-1}} \end{aligned} \quad (17.104)$$

To preserve the positiveness of the material stiffness matrix a minimal value for shear and normal stiffnesses must be specified. In this work the shear and normal stiffnesses cannot be less than 10^{-8} times their original value. This number is based on the assumption that the ratio of the lowest elastic modulus to the largest interface stiffness is below 10^{-4} . This ratio should be sufficient in most practical problems, since the interface stiffness can be estimated from:

$$K_n = \frac{E}{t} \quad (17.105)$$

where t is the interface thickness. Thus, the ratio $\frac{E}{K_{\text{interface}}} \approx 10^{-4}$ corresponds to the assumption of interface thickness being equal to 10^{-4} times a unit length of the problem. This should be adequate for the types of problems under consideration in this work. Alternatively, we consider an extreme case of $K_{\text{interface}}$ of the same order as E (i.e. $t \approx$ problem unit). Then after cracking, the interface stiffness will be reduced to 10^{-8} times its original value, and it is possible to estimate the condition number of the system using the elastic modulus, maximal and minimal element sizes.

$$\kappa \approx \frac{K_{\max}^{\text{elem}}}{K_{\min}^{\text{interface}}} \approx \frac{Eh_{\max}}{(Eh_{\min}^2)/t \times 10^{-8}} \approx 10^{14} \quad (17.106)$$

In the formula, the element sizes were assumed to be in the range of the order $\langle 10^{-2}, 10^2 \rangle$.

The loss of accuracy due to finite precision arithmetic is given by:

$$s = p - \log(\kappa) \quad (17.107)$$

where p is the number of significant digits in the computer representation of real numbers and s is the accuracy of the solution. The system will become ill-conditioned when:

$$s \leq 0 \quad (17.108)$$

A real number f is internally represented in a computer memory by three integers m , β and e .

$$f = .m \times \beta^e \quad (17.109)$$

The mantissa m gives the number of significant digits. For double precision data type, m is usually stored in 52 bits, which corresponds to approximately 16 significant digits. Therefore, the accuracy after decomposition is in the worst possible scenario equal to 2 (Equation 17.107), which is of course an unacceptable level of accuracy. However, it should be kept in mind that this is a worst case scenario, and it would be unrealistic to have a ratio of largest to smallest element of the order of 10^4 , as was assumed in Equation 17.106.

17.2.3.6 Line search method.

Numerical experiments showed, that often the diagonal approximation of the secant stiffness underestimates the true stiffness of the interface and allows for excessive interface sliding. The excessive sliding in turn introduces large dilatancy effects and high compressive stresses in the normal direction in the subsequent iteration. These high compressive stresses and the frictional properties of the interface combined with the excessive slidings will cause large shear stresses, which may not be in equilibrium with the rest of the finite element mesh. Due to this, the resulting high residual forces attempt to slide the interface backwards, but since the stiffness of the interface is underestimated, the backward sliding is too large, and the iteration process diverges. This problem can be solved by combining the previously discussed secant-methods with line searches.

The fundamental principle behind the line search method (Crisfield 1991) is to determine a scaling factor ω , for the current iterative displacement correction, such that the functional of total potential energy is stationary.

$$\Pi(\omega) = \Pi(\mathbf{u}_{i-1} + \omega d\mathbf{u}_i) = \Pi(\omega) + \frac{\partial \Pi(\omega)}{\partial \mathbf{u}(\omega)} \frac{\partial \mathbf{u}(\omega)}{\partial \omega} \delta \omega \quad (17.110)$$

The functional $\Pi(\omega)$ would be stationary if the last term is equal to zero. It can be shown, (Crisfield 1991), that the partial derivative of total potential energy $\Pi(\omega)$ with respect to displacements $\mathbf{u}(\omega)$ is equal to the vector of residual forces $\mathbf{r}(\mathbf{u})$. Thus, the last term of Equation 17.110 is equivalent to:

$$\frac{\partial \Pi(\omega)}{\partial \mathbf{u}(\omega)} \frac{\partial \mathbf{u}(\omega)}{\partial \omega} \delta \omega = \mathbf{r}(\omega) d\mathbf{u}_i \delta \omega = 0 \quad (17.111)$$

If we introduce a new symbol $s(\omega)$ representing the scalar product of vectors $\mathbf{r}(\omega)$ and $d\mathbf{u}_i$, then the objective is to find a scalar multiplier ω such that $s(\omega)$ is equal to zero. Such ω can be approximately computed from $s(\omega)$ for ω equal to zero and one.

$$s(0) = \mathbf{r}(\mathbf{u}_{i-1}) d\mathbf{u}_i, \quad s(1) = \mathbf{r}(\mathbf{u}_{i-1} + d\mathbf{u}_i) d\mathbf{u}_i \quad (17.112)$$

Then an approximation of ω can be evaluated using the following formulas based on the linear interpolation between $s(0)$ and $s(1)$.

$$\omega = \frac{-s(0)}{s(1) - s(0)} \quad (17.113)$$

A more accurate value of ω can be determined through recursive applications of this formula.

$$\omega_{i+1} = \omega_i \frac{-s(0)}{s(\omega_i) - s(0)} \quad (17.114)$$

Graphically, the line search is illustrated in Figure 17.14. We observe that it corresponds exactly to the divergence

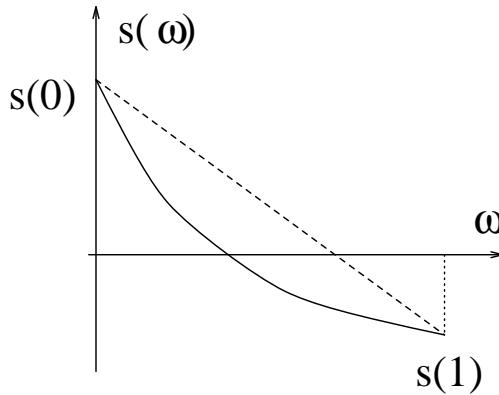


Figure 17.14: Line search method.

problem previously described. Originally, the residual forces acted along the same direction as the iterative displacement correction, and their scalar product $s(0)$ was positive. However, after the iterative correction is considered, the residuals have opposite orientation with respect to the iterative displacement update $d\mathbf{u}_i$, and $s(1)$ is negative. This indicates that the displacements should be smaller, and this is exactly, what the line search method is able to recognize and correct.

The line search method can be implemented in the context of various load control techniques. The implementation of line searches in the context of the arc-length method is discussed in (Crisfield 1991). (Reich 1993) implemented the line search method with an indirect displacement control technique, which is based on crack step control mechanism, and can be therefore easily used for non-linear fracture mechanics analyzes using the FCM model.

17.2.4 Mixed Mode Crack Propagation

In most engineering problems, the crack path is not known a priori, and therefore, must be determined during an analysis. In the context of discrete crack analysis, this is accomplished by modifications of the initial mesh. It is, therefore, necessary to establish appropriate criteria for crack initiation and propagation. The criteria for LEFM analysis were discussed in Section ?? of Chapter ???. In the non-linear fracture mechanics analysis, a crack initiation criterion can be based on tensile stresses, and energy control is conducted through an appropriate softening diagram. This is to be contrasted with LEFM, where the stress based criteria are not applicable, as they are infinite at the crack tip. It can be readily verified that the Griffith energy based criterion is also satisfied in the non-linear fracture mechanics through an appropriate softening law.

17.2.4.1 Griffith criterion and ICM.

Let us consider a cohesive crack with both normal and tangential tractions in a thin plate subjected to far field stresses, and let us assume the crack is to be under general mixed mode conditions, Figure 17.15. To verify if the non-linear model satisfies Griffith criterion, it is necessary to compute the energy released by a unit crack propagation. The J-integral provides a method to evaluate the energy release rate. The J-integral is a path independent integral and in two-dimensional is given by:

$$J = \oint_{\Gamma} (W n_x - \hat{t} \frac{\partial u}{\partial x}) d\Gamma \quad (17.115)$$

Due to its path independent character it is possible to evaluate the J-integral along the crack surfaces.

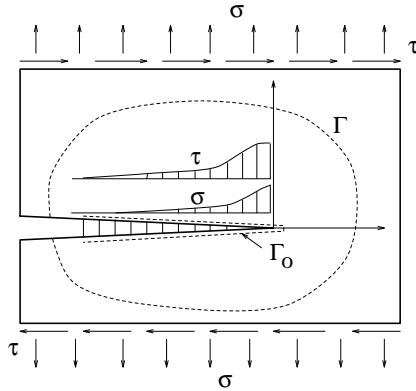


Figure 17.15: Griffith criterion in NLFM.

$$J(\Gamma_o) = - \int_{\Gamma_o} \hat{t} \frac{\partial u}{\partial x} ds = \int_{FPZ} \left(\tau \frac{\partial \Delta_x}{\partial x} + \sigma \frac{\partial \Delta_y}{\partial x} \right) dx \quad (17.116)$$

Applying Leibnitz rule for the differentiation of definite integrals the J-integral is equivalent to:

$$J(\Gamma_o) = \int_{FPZ} \left[\frac{d}{dx} \left(\int_0^{\Delta_x} \tau d\Delta_x \right) \right] dx + \int_{FPZ} \left[\frac{d}{dx} \left(\int_0^{\Delta_y} \sigma d\Delta_x \right) \right] dx \quad (17.117)$$

The expressions in parentheses represent the surface energies dissipated in mode I and II at every point along the fracture process zone normalized with respect to crack surface. Hence, we define:

$$\int_0^{\Delta_x} \tau d\Delta_x = q^{II}(x), \quad \int_0^{\Delta_x} \sigma d\Delta_y = q^I(x) \quad (17.118)$$

$$J(\Gamma_o) = \int_{FPZ} \frac{dq^{II}(x)}{dx} dx + \int_{FPZ} \frac{dq^I(x)}{dx} dx = G_c^{II} + G_c^I = G_c \quad (17.119)$$

where G_c^{II} and G_c^I is the energy dissipated by a unit propagation of the cohesive crack in mode II and I respectively. It should be noted that in general, G_c^{II} and G_c^I are not equivalent to G_F^{II} and G_F^I , but are rather functions of these and the stress state along the interface. However, it is possible to consider two special cases for pure mode I and II cracks.

In the case of pure mode I crack, the J-integral is equal to:

$$J(\Gamma_o) = \int_{FPZ} \left[\frac{d}{dx} \left(\int_0^{\Delta_y} \sigma d\Delta_y \right) \right] dx = \int_0^{w_\sigma} \sigma(\Delta_y) d\Delta_y = G_F^I \quad (17.120)$$

Similarly, in the case of pure mode II crack, the J-integral is equal to:

$$J(\Gamma_o) = \int_{FPZ} \left[\frac{d}{dx} \left(\int_0^{\Delta_x} \tau d\Delta_x \right) \right] dx = \int_0^{w_\tau} \tau(\Delta_x) d\Delta_x = G_F^{II} \quad (17.121)$$

where w_σ and w_τ is the critical crack opening and sliding respectively for which normal and tangent stresses can no longer be transferred across the crack.

The following conclusion can be drawn based on the basis of the previous discussion:

(1) It was shown that a unit extension of a cohesive crack model dissipates energy whose amount depends on the softening laws used by the model. The amount of dissipated energy also depends on the loading conditions in FPZ. In pure mode I and mode II loading, specific fracture energies G_F^I and G_F^{II} are dissipated respectively. If the structural system cannot provide these energies, the crack would not propagate.

(2) In the limiting case, when the dimensions of the analyzed problem increase, the cohesive crack gives identical results as LEFM.

(3) In finite element implementation, errors are introduced due to discretization errors. In large structures, fine mesh would be necessary at the crack tip to model the fracture process zone. If the FPZ is not modeled adequately, the Griffith criterion for crack propagation is violated, and erroneous results will be obtained.

17.2.4.2 Criterion for crack propagation.

In this work a stress based criterion is used for crack initiation and propagation. A crack is initiated when a maximal principal stress σ_1 exceeds the tensile strength of the material. A crack of certain length Δa is inserted into the boundary representation of the model in the direction perpendicular to the direction of the maximal principal stress, and the length of the new crack Δa is arbitrarily selected by the user. The exact solution is approached as this length tends to zero, this is however not feasible, and from author's experience, the crack step size should be:

$$\Delta a \leq \frac{L}{10} \quad (17.122)$$

where L is maximal dimension of the problem. From the updated boundary representation, a new mesh is generated, in which interface elements are placed along the crack. Then, a non-linear analysis is performed, and the maximal principal stresses at crack tips are monitored. When they are found to exceed the tensile strength of the material, the analysis is interrupted, and new crack surfaces are inserted into the boundary representation of the problem. Then, a new mesh is again generated and the problem is reanalyzed from the beginning. In this manner the finite element model is adaptively modified until the structure is fully cracked or the prescribed loading level is reached. This process is described by Algorithm 17.2, and is shown graphically in Figure 17.16.

ALGORITHM 17.2 (Mixed mode crack propagation.)

- (1) *Input: Boundary representation.*
- (2) *Generate finite element model.*
- (3) *Do*
 - (3.1) *Non-linear finite element analysis.*
- (4) *While: maximal principal stresses < f_t' .*
- (5) *If maximal principal stress exceed f_t' .*
 - (4.1) *Add new crack surfaces of length Δa to
the boundary representation in the direction perpendicular to σ_1 .*
 - (4.2) *Goto Step 2.*
- (6) *Output: Boundary representation, Finite element model.*

17.3 Interface Crack Model; ICM-2 Cyclic (MM: 21)

Adapted from ?

17.3.1 Introduction

Joint and interfaces, coupled with cohesive stresses, are present in many structures spanning well over six orders of magnitudes in size (from metallic polycrystals, ceramics to dams and tectonic faults). In all cases, one is confronted with an actual or potential displacement discontinuity where classical continuum mechanics fails to provide a solution, and very often these displacement discontinuities are precisely the main source of nonlinearity.

Depending on the field of study, these discontinuities assume different names: interface, crack, joint, fault or even artificially built joint. A civil engineering application where cracks abound are dams where, let aside AAR, they

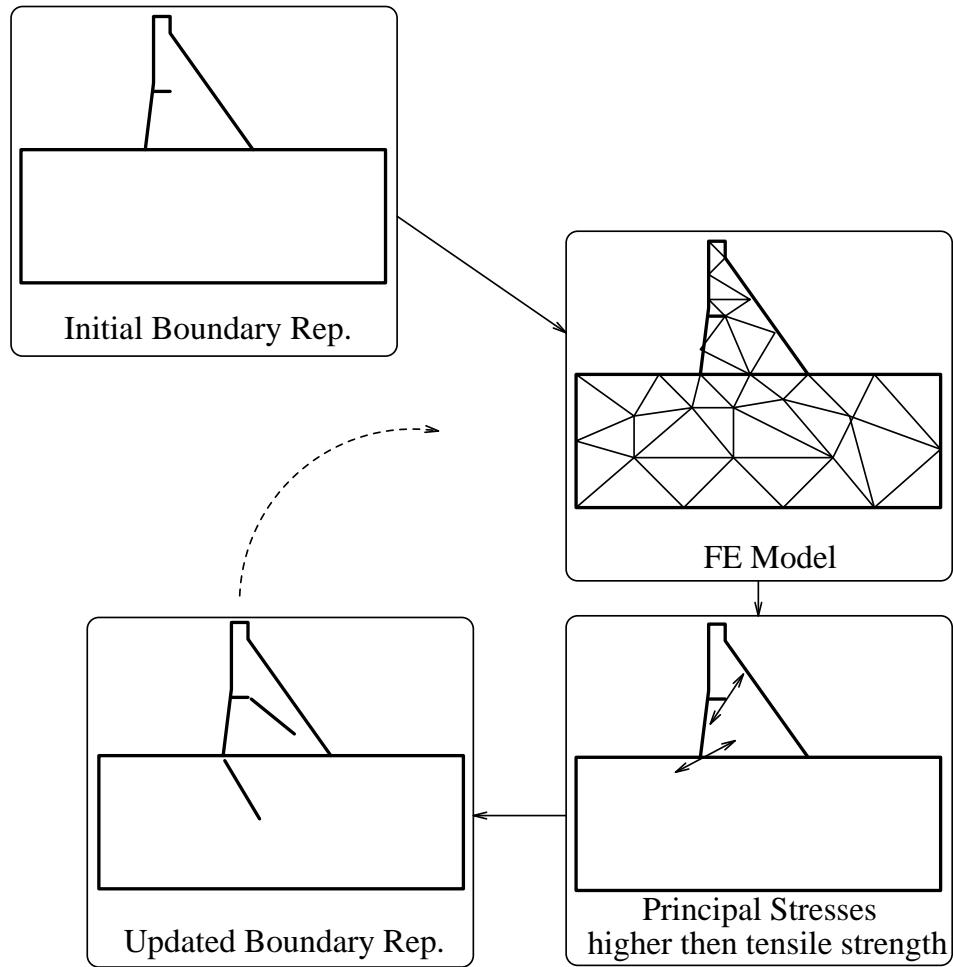


Figure 17.16: Mixed mode crack propagation.

are the major source of nonlinearity. They are present along the rock/concrete interface, lift joints, cantilever joints (with or without shear keys), in plain concrete cracks, or rock joints. Yet, irrespective of their origin, all those cracks can be correctly modelled by the same generalized model provided material parameters are appropriately set. Other civil structures where joints or cracks can be of particular concern are nuclear containment vessels, (Hansen and Saouma 2003). Surprisingly, interface elements have also been used, albeit at a much smaller scale, for improved understanding of ceramics (Saouma, Natekar and Sbaizer 2002), and aluminum (Iesulauro, Ingraffea, Arwade and Wawrynek 2002).

Ever since the pioneering work of Goodman, R.E. and Taylor, R.C. and Brekke, T.C. (1968), numerous joint or interface laws have been proposed for both discrete and smeared crack models in concrete structures, (Hohberg 1992).

Of particular interest to dam engineering are the models of Fenves, Mojtabaei and Reimer (1992), Divoux, Bourdarot and Boulon (1997), Hall (1998), and Ahmadi, Izadnia and Bachmann (2001). Each of those models indeed presents an innovative component, but none appears to have been generalized to account for many phenomena associated with reverse cyclic load, asperity degradation, softening of tensile strength and cohesion, or stiffness degradation.

Some mixed-mode interface models exploit the analogy with the mechanics of irreversible plastic processes to account for unrecoverable joint opening and sliding; to them belongs the proposal made by Plesha (1987) and several successive works based on it. The frictional behaviour of the joint under reversed shear in compression is mainly considered in this approach, widely used in rock mechanics.

Insight and deeper understanding of lower scale surface interactions can be achieved through micro-mechanics based models such as (Fox, Kana and Hsiung 1998), (Grasselli, Wirth and Egger 2002), Misra (2002). The prediction of the joint behaviour results from a statistical description of the surface topography, but this kind of information is seldom available to practical purposes in structural design and overall analysis. In these cases, phenomenologically based models, unburdened by mechanics, often result in easier modelling of experimental results, (Bažant and Gambarova 1980) (Divoux, Boulon and Bourdarot 1997).

Fracture mechanics based models set in an elasto-plasticity framework seem to be the most general formulation in terms of the range of problems they can address; see, e.g.: (Lotfi and Shing 1994), (Carol, Prat and López 1997), (Červenka, Kishen and Saouma 1998), and (Cocchetti, Maier and Shen 2002). In particular, they permit to include and follow strength deterioration leading to the formation and progressive development of natural joints (i.e., cracks).

To the best of the authors' knowledge, the only generalized model addressing cyclic load, though in a displacement based formulation, is the one of Giambanco and Di Gati (1997) based on previous frictional-dilatant models by Snyman and Martin (1992) and Mróz and Giambanco (1996). The model, intended for the structural analysis of masonry blocks, introduces a piece-wise linear yield condition governed by two independent internal variables. In the present model a unique yield function is defined and its evolution in the stress space controlled by a quantity which has a clear mechanical meaning, namely by a norm of inelastic displacement discontinuities. Moreover the present proposal is also different in the definition of roughness characteristics, e.g. through dilatant displacement and not dilatancy angle, and in their connection with the mechanical properties.

In this paper, an existing fracture mechanics based joint model, (Červenka et al. 1998), is extensively modified to account for cyclic loading (and accompanying surface degradation) in a manner similar to the one proposed by Plesha (1987). A suitable idealisation of the joint surface geometry is introduced for describing the macroscopic (overall) behaviour of the joint, more than for reflecting its microscopic structure as in (Fox et al. 1998), (Grasselli et al. 2002), Misra (2002).

A review of the monotonic interface element being extended is summarized first, then the formulation for cyclic load is presented. Finally the response of the generalized model at material point level is analyzed through a comparison with the model of Červenka et al. (1998) and with the cyclic shear experimental results of Kutter and Weissbach (1980).

The formulation presented herein is restricted to two dimensional situations; extension to 3D cases is conceptually straightforward only if isotropy is assumed in the joint plane.

17.3.2 Cyclic behavior of quasi brittle interfaces

Experimental studies on the cyclic behavior of quasi-brittle interfaces have been reported for both concrete and rock. For concrete they are mainly motivated by the investigation of the aggregate interlock phenomenon in which a slightly opened crack is subjected to reversed cyclic slip at given initial confinement, (Paulay and Loeber 1974), (Tassios and Vintzileou 1987) and (Fronteddu, Léger and Tinawi 1998).

Numerous experiments on rock joints have been carried out. Of particular relevance to the present investigation is the work of Hutson and Dowding (1990), Lee, Park, Cho and You (2001), Homand, Belem and Souley (2001), and Jafari, Hosseini, Pellet, Boulon and Buzzi (2003). All of these studies contain also proposals of shear strength or dilatancy degradation laws derived from the tests.

In rock mechanics, a clear distinction is often made between first and second order asperities as those factors have a strong influence on joint response, (Patton 1966). First order asperities (from here on referred to as "asperities"

unless otherwise noted) are associated with roughness of larger amplitude and wave-length and thus they tend to dominate dilatant behavior; second order asperities are associated with smaller amplitude and wave-length surface variations and are primarily responsible for the frictional forces exchanged along the inclined sliding surfaces.

In the case of a smooth joint, i.e. with no relevant first order asperities effects, the quasi-static response is characterized by almost no dilatancy and constant shear stress. This behavior can be captured by a relatively simple non associate Coulomb type frictional law.

On the other hand, in rough joints (characterized by prominent first order asperities) the response depends on the slip direction (forward or backward). Mathematically, forward and backward slip are respectively defined as having an increasing or decreasing absolute value of tangential relative displacement.

In forward slip, not only does a rough joint dilate, but its apparent shear strength is also higher. The opposite is true for backward slip. Furthermore, both of these behaviors are affected by the degradation of the joint surfaces with progressive cycling: the dilatancy angle and the configurational difference in shear strength decrease as asperities are worn out.

Several models for rough interfaces have been published. A particularly effective mechanical interpretation was given by Plesha (1987), who assumed that sliding does not occur parallel to the joint mid-plane but along an inclined angle characterizing the asperities. Hence, writing the Coulomb slip criterion along the inclined slope, and expressing it in terms of joint stress vector components, the essential characteristics of backward and forward slip are captured.

The observed degradation of the joint characteristics with cycling loading is usually ascribed to the decrease of the asperity angle, often exponentially with the tangential work performed. This assumption, herein adopted, has been followed by Hutson and Dowding (1990), Qiu, Plesha, Huang and Haimson (1993), and Stupkiewicz and Mróz (2001).

17.3.3 Červenka 1994 hyperbolic model

Following a broad literature survey, (Punzel 2004), it was determined that the most suitable monotonic interface element for cyclic extension to the present dam-engineering oriented purpose, is the one originally developed by Červenka (1994) and subsequently published by (Červenka et al. 1998).

The formulation developed herein is two dimensional (2D), nevertheless the extension to the 3D case is straightforward provided that isotropy is assumed in the joint plane. Tractions and discontinuities considered are then 2D vectors with a normal and a single tangential component, referred to by subscript n and t respectively.

The strength (alias yield or activation) criterion of the interface is hyperbolic as also assumed by Carol et al. (1997), Lotfi and Shing (1994).

$$\varphi = p_t^2 - (c - p_n \mu)^2 + (c - \chi \mu)^2 \quad (17.123)$$

Three parameters define the interface strength: the two static internal variables, namely tensile strength χ and cohesion c , and the friction coefficient μ . The former two decrease bi-linearly with the effective inelastic displacement w^{ieff} which is the model's softening variable.

$$\chi(w^{ieff}) = \begin{cases} \chi_0 - \frac{\chi_0 - \chi_1}{w_{\chi_1} - w_{\chi_0}} w^{ieff} & 0 \leq w^{ieff} \leq w_{\chi_1} \\ \chi_1 \frac{w_{\chi_0} - w^{ieff}}{w_{\chi_0} - w_{\chi_1}} & w_{\chi_1} \leq w^{ieff} \leq w_{\chi_0} \end{cases} \quad (17.124)$$

$$c(w^{ieff}) = \begin{cases} c_0 - \frac{c_0 - c_1}{w_{c_1} - w_{c_0}} w^{ieff} & 0 \leq w^{ieff} \leq w_{c_1} \\ c_1 \frac{w_{c_0} - w^{ieff}}{w_{c_0} - w_{c_1}} & w_{c_1} \leq w^{ieff} \leq w_{c_0} \end{cases} \quad (17.125)$$

$$w_{\chi_0} = (2G_f^I - \chi_0 w_{\chi_1})/\chi_1 \quad (17.126)$$

$$w_{c_0} = (2G_f^{IIa} - c_0 w_{c_1})/c_1 \quad (17.127)$$

where: χ_0 , c_0 , G_f^I , G_f^{IIa} , w_{χ_1} , χ_1 , w_{c_1} , c_1 , w_{χ_0} , w_{c_0} are the material parameters described in the notation list. Of these ten parameters, only eight are independent to define the bilinear curves; the other two (w_{χ_0} and w_{c_0}) can be determined from equations 17.126 and 17.127.

The rate of w^{ieff} is defined as the norm of the rate of inelastic displacements \dot{w}^i :

$$\dot{w}^{ieff} = \|\dot{w}\| = \left((\dot{w}_n^i)^2 + (\dot{w}_t^i)^2 \right)^{1/2} \quad (17.128)$$

The inelastic displacements \mathbf{sw}^i are the sum of plastic (i.e. unrecoverable) and fracture (i.e. recoverable in tension only) displacements \mathbf{sw}^p and \mathbf{sw}^f respectively; total displacement discontinuities \mathbf{sw} are obtained adding the elastic term \mathbf{sw}^e to the previous ones:

$$\left. \begin{array}{l} \mathbf{sw} = \mathbf{sw}^e + \mathbf{sw}^i \\ \mathbf{sw}^i = \mathbf{sw}^p + \mathbf{sw}^f \end{array} \right\} \Rightarrow \mathbf{sw} = \mathbf{sw}^e + \mathbf{sw}^p + \mathbf{sw}^f \quad (17.129)$$

The distinction between the two inelastic terms is motivated by the considered deterioration of the elastic stiffness in tension due to the damage parameter D : \mathbf{sw}^f enters explicitly in the expression of D , while \mathbf{sw}^p does not. The matrix of initial elastic stiffness coefficients \mathbf{sK}_0 is diagonal with K_{n0} and K_{t0} defined as normal and tangential components respectively. An elastic deterioration coefficient ρ is introduced; ρ is fixed to one in compression, while it ranges from one to zero in tension according to the level of damage D :

$$\rho = 1 - \frac{\langle p_n \rangle}{|p_n|} D \quad (17.130)$$

where the symbol $\langle \bullet \rangle$ indicates the Macaulay brackets:

$$\langle \bullet \rangle = (\bullet + |\bullet|)/2 \quad (17.131)$$

The traction – displacement discontinuity relationship reads:

$$\mathbf{sp} = \rho \mathbf{sK}_0 (\mathbf{sw} - \dot{\mathbf{sw}}^p) \quad (17.132)$$

Damage D can hence be defined as the complement to one of the ratio between the current normal stiffness K_{nc} and the initial one K_{n0} .

$$D = 1 - \frac{K_{nc}}{K_{n0}} = \frac{K_{n0} - K_{nc}}{K_{n0}} \quad (17.133)$$

It can be shown, (Červenka et al. 1998), that D is related to the current normal strength $\chi(w^{\text{eff}})$ by the relationship:

$$D = 1 - \frac{\chi(w^{\text{eff}})}{\chi(w^{\text{eff}}) + (1 - \gamma) w^{\text{eff}} K_{n0}} \quad (17.134)$$

where γ , a new parameter, is introduced to define the irrecoverable (plastic) portion of inelastic displacements:

$$w_n^p = \gamma w_n^i \quad (17.135)$$

Finally, the direction of inelastic displacements is explicitly defined by the gradient of the potential Q :

$$\dot{\mathbf{sw}}^i = \frac{\partial Q}{\partial \mathbf{sp}} \dot{\lambda} \quad , \quad \dot{\lambda} \geq 0 \quad (17.136)$$

$$\frac{\partial Q}{\partial \mathbf{sp}} = \begin{cases} \begin{bmatrix} p_n/K_{n0} \\ p_t/K_{t0} \end{bmatrix} & \text{if } \frac{p_n}{|p_t|} \geq \mu_d \frac{K_{n0}}{K_{t0}} \\ \begin{bmatrix} |p_t| \mu_d \\ p_t \end{bmatrix} & \text{otherwise} \end{cases} \quad (17.137)$$

where μ_d is the dilatancy angle.

Around the origin of the stress space the inelastic return direction is toward the origin if $K_{n0} = K_{t0}$, otherwise it is given by the normal to an ellipse with aspect ratio $\sqrt{K_{t0}/K_{n0}}$; when the tangent to the ellipse equals the tangent μ_d of the dilatancy angle, the direction remains constant for every smaller value of normal traction p_n . However, for $w^{\text{eff}} \leq w_{dil}$ the dilatancy μ_d is not constant but decreases linearly with w^{eff} from its initial value μ_{d0} to zero:

$$\mu_d(w^{\text{eff}}) = \begin{cases} \mu_{d0} (1 - w^{\text{eff}}/w_{dil}) & w^{\text{eff}} \leq w_{dil} \\ 0 & w^{\text{eff}} > w_{dil} \end{cases} \quad (17.138)$$

17.3.4 Proposed extension to cyclic loading

This section presents an extension of the previously described interface model into a generalized one which can also capture the essential characteristics of joint cyclic behavior. This is done preserving the inherent capabilities of the original element, and maintaining its fracture mechanics based origin.

The cyclic model description presented herein is limited to those features that will be added or modified to the original Červenka model, namely: 1) introduction of an asperity function which characterizes joint roughness and governs the dilatancy of the model; 2) consideration of an integrity factor which keeps track of the degradation of the asperities; 3) modification of the yield function, of the friction angle in particular, to account for the sliding along inclined asperities.

Some other aspects of the response of joints to cyclic loading were not considered here for the following reasons:

1. Joint bulking or seating, that is the increase or decrease of joint thickness with asperity degradation respectively, was not included due to apparent lack of consistent experimental results.
2. Configuration rearrangements of third body granular layer particles caused by debris inside the joint (Stupkiewicz and Mróz 2001) is not accounted for due to: its minor relevance in the present context; complexity; paucity of experimental results.
3. Dilatancy associated to second order asperities has been deemed as not essential for the aims of the present model, though it would be easy to insert it in the model and despite the fact that Lee et al. (2001) and Jafari et al. (2003) report their influence on first loading cycles and for tangential relative displacements of small amplitude.
4. Asperity degradation caused by pure compressive stresses was left since it was preferred to describe permanent normal deformations by means of the elasto-plastic strains developing in the bulk material.
5. A fully 3D formulation, including effects such as anisotropic wear, has not been dealt with so far, but represents an important extension and possible subject of future work.

17.3.4.1 Analytical formulation

In what follows, the symbol α refers to quantities related to first order asperities, while β refers to all joint properties not related to first order asperities (such as tensile strength and cohesion) besides frictional quantities. The term “basic” will indicate joint properties associated with second order roughness, while “apparent” will refer to both orders.

17.3.4.1.1 Asperity definition Following the formulation of Plesha (1987), and of Stupkiewicz and Mróz (2001), an asperity curve characterizing first order joint roughness can be defined as follows:

$$w_n^i = f(p_n, L_t^i) \cdot y(w_t^i) \quad (17.139)$$

This has to be intended as an average geometry of the joint surface reflecting the macroscopic (overall) behaviour of the joint rather than its microscopic structure as in (Fox et al. 1998), (Grasselli et al. 2002), Misra (2002). The asperity curve relates the joint irreversible normal (w_n^i) and tangential (w_t^i) relative displacements, and it is the product of the geometric curve y defining the initial asperity shape with an integrity parameter f (to be defined later) which reflects the joint degradation level and ranges from 1 to 0. Integrity (f) is assumed to be a function of normal traction p_n and tangential inelastic shear work L_t^i defined as follows in rate form:

$$\dot{L}_t^i = p_t \cdot \dot{w}_t^i \quad (17.140)$$

It should be noted that roughness degradation affects the asperity height only, while its wavelength remains unchanged.

In this work two particular asperity curves are considered, namely a Gaussian and a hyperbolic one:

$$y(w_t^i) = h_0 \left(1 - \exp \left(-\frac{1}{2} \left(\frac{w_t^i}{s} \right)^2 \right) \right) \quad \text{Gaussian} \quad (17.141)$$

$$y(w_t^i) = \mu_{\bar{\alpha}0} \left(\sqrt{(w_t^i)^2 + (r_0 \mu_{\bar{\alpha}0})^2} - r_0 \mu_{\bar{\alpha}0} \right) \quad \text{hyperbolic} \quad (17.142)$$

The gaussian asperity function reaches a constant value for large sliding displacements, thus implying that asperities are not periodic so that dilatancy cannot be recovered once sliding has exceeded a characteristic asperity length.

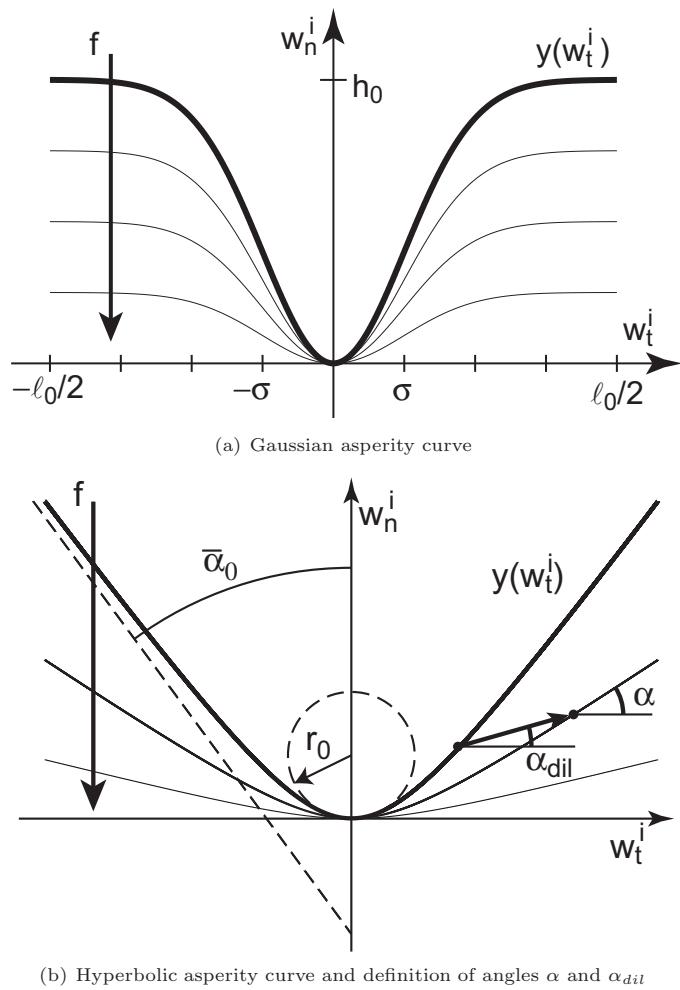


Figure 17.17: Asperity curves

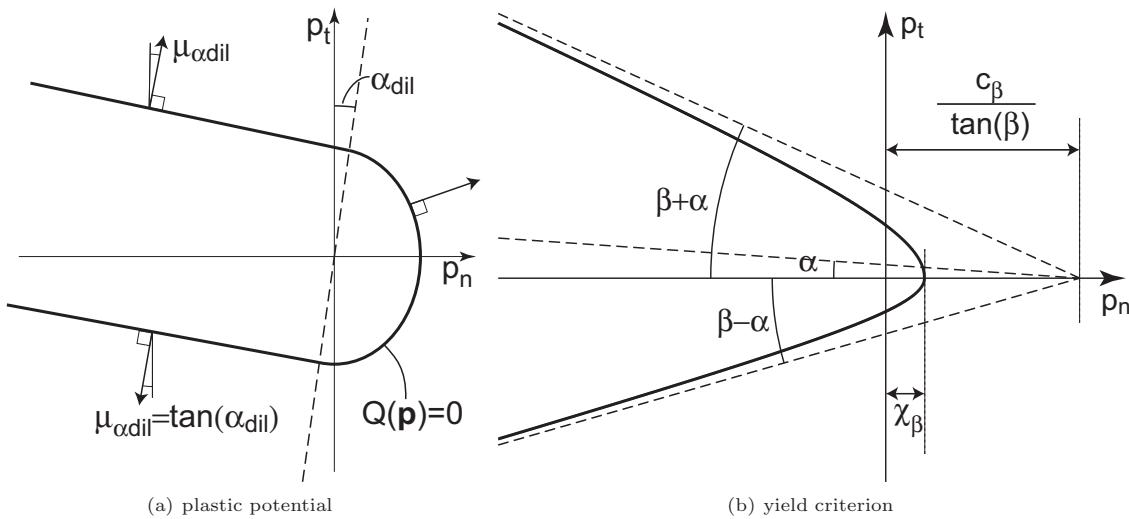


Figure 17.18: Cyclic model: yield criterion and plastic potential

Figure 17.17(a) shows the two parameters of the Gaussian asperity curve: asperity height h_0 and asperity length ℓ_0 . Parameter s in equation 17.141 determines the curve amplitude and is therefore closely connected to ℓ_0 .

The evolution of the Gaussian asperity curve with progressive degradation is shown in fig. 17.17(a). Again it can be noted that only the ordinate (asperity height) is affected and not the abscissa (asperity length).

The hyperbolic asperity function, figure 17.17(b), grows indefinitely, even for large sliding displacements; therefore it is only appropriate for problems in which it is a priori known or assumed that the tangential slip w_t^i will be smaller than a characteristic asperity length. This curve is characterized by two parameters: the inclination angle of the asymptote of the hyperbola, $\bar{\alpha}_0$, and the curvature r_0 at the origin. The tangent of $\bar{\alpha}_0$ is named $\mu_{\bar{\alpha}0}$, according to the rule $\mu_x = \tan(x)$; $\bar{\alpha}_0$ is also the maximum value the asperity angle can attain for hyperbolic asperities. The same symbol will be used to indicate the maximum angle reachable for gaussian asperities; i.e., in this case:

$$\bar{\alpha}_0 = h_0/s \cdot \exp(-1/2) \quad (17.143)$$

Saw-tooth shaped asperities, often adopted in roughness description, were not used for experimental and numerical reasons. As observed by (Hutson and Dowding 1990) and (Sun, Gerrard and Stephansson 1985), an initial amount of joint shearing is necessary to induce maximum dilatancy angle and shear strength. Besides, a curve with continuous derivative is computationally preferable.

Following (Stupkiewicz and Mróz 2001), the dilatancy curve introduced by equation 17.139 is used to explicitly prescribe the joint dilatant behavior as a function of current normal stress, inelastic tangential work and displacement.

Plesha (1987), Giambanco and Di Gati (1997) and other researchers have preferred to prescribe the dilatancy angle, instead of the dilatant displacement, as a function of current tractions, relative displacements and internal variables, but it was realized that this approach can lead to undesirable and uncontrollable joint bulking.

The dilatancy angle α_{dil} , shown in figure 17.17(b), is defined as the total variation of w_n^i with respect to w_t^i ; therefore, from equation 17.139:

$$\mu_{\alpha_{dil}} = \tan(\alpha_{dil}) = \frac{dw_n^i}{dw_t^i} = \frac{d(fy)}{dw_t^i} \quad (17.144)$$

Finally we have to consider that the effect of the asperity curve requires a modification to Červenka's flow rule, which defines the direction of the vector of inelastic displacement discontinuity. This direction is now defined such that under backward slip the joint does not dilate but contracts, figure 17.18(a). Hence, the gradient of the plastic potential Q is given by

$$\frac{\partial Q}{\partial sp} = \begin{cases} \begin{bmatrix} p_n/K_{n0} \\ p_t/K_{t0} \end{bmatrix} & \text{if } p_n \geq \mu_{\alpha_{dil}} \frac{K_{n0}}{K_{t0}} p_t \\ \begin{bmatrix} p_t \mu_{\alpha_{dil}} \\ p_t \end{bmatrix} & \text{otherwise} \end{cases} \quad (17.145)$$

Relationships 17.145 imply that the original formulation is retained under monotonic loading.

17.3.4.1.2 Asperity degradation The integrity parameter f , introduced in equation 17.139 ranges from 1 to 0 and governs asperity degradation. The results of the experimental tests carried out by Lee et al. (2001), Homand et al. (2001), Jafari et al. (2003), Huang, Haimson, Plesha and Qiu (1993) suggest that f will depend not only on the inelastic shear work L_t^i , as proposed by Plesha (1987) and Hutson and Dowding (1990), but also on the normal traction p_n .

In fact:

- a non zero steady state asperity degradation of joints is reached after several shearing cycles at constant confinement.
- the residual, alias asymptotic, degradation depends on the amount of applied compressive stress.

The incremental expression of f is defined in terms of \bar{f} (integrity factor in the known configuration), p_n (normal traction in the ensuing state) and ΔL_t^i (increment of inelastic shear work between the two configurations) as follows :

$$\Delta f \left(\bar{f}, p_n, \Delta L_t^i \right) = \langle \bar{f} - f_{asym}(p_n) \rangle \cdot \left(1 - \exp \left(-C \Delta L_t^i \right) \right) \quad (17.146)$$

$$f_{asym}(p_n) = (-d p_n + 1)^{-1} \quad (17.147)$$

where, once again, the symbol $\langle \cdot \rangle$ indicates the Macaulay brackets (see eq. 17.131).

The asymptotic degradation factor f_{asym} provides a residual value under constant confinement when the increment of inelastic shear work ΔL_t^i tends to infinity.

Under tension the asperities are not worn and f_{asym} is fixed to 1; under increased compressive stress the asymptotic degradation factor decreases, reaching zero when p_n tends to minus infinity. Function $f_{asym}(p_n)$ is a hyperbola with a single parameter, d , which controls the speed rate as the function approaches zero. Its expression is relatively simple due to its derivation from qualitative, though not quantitative, experimental observations.

Looking at equation 17.146, it can be noted that for $\Delta L_t^i = 0$, that is if no tangential inelastic work takes place, then f is equal to \bar{f} for any value of p_n . Furthermore, asperities do not wear when $f_{asym}(p_n)$ is larger than \bar{f} , that is when asperities have already degraded more than they would under the current value of normal stress p_n for any value of ΔL_t^i .

Conversely, if $f_{asym}(p_n)$ is smaller than \bar{f} and p_n is kept constant, the integrity factor f decreases exponentially for increasing ΔL_t^i from f to $f_{asym}(p_n)$. The speed of the exponential decay is controlled by C .

The evolution of f is defined by its gradient with respect to L_t^i and p_n :

$$\frac{\partial f}{\partial L_t^i} = - \langle \bar{f} - f_{asym}(p_n) \rangle \cdot C \quad (17.148)$$

$$\frac{\partial f}{\partial p_n} = 0 \quad (17.149)$$

where \bar{f} is again the current value of f .

17.3.4.1.3 Rotated activation function The total variation of w_t^i with respect to w_t^i reads:

$$\frac{dw_n^i}{dw_t^i} = \left(\frac{\partial f}{\partial p_n} \cdot \frac{\partial p_n}{\partial w_t^i} + \frac{\partial f}{\partial L_t^i} \cdot p_t \right) \cdot y(w_t^i) + f \frac{\partial y}{\partial w_t^i} \quad (17.150)$$

Substituting equations 17.148 and 17.149 in 17.150, we note that the total derivative is contributed by a compaction and a friction term:

$$\frac{dw_n^i}{dw_t^i} = \left(- \langle \bar{f} - f_{asym}(p_n) \rangle \cdot C p_t \right) \cdot y(w_t^i) + f \frac{\partial y}{\partial w_t^i} \quad (17.151)$$

Only the frictional term is retained here to account for inclination of the sliding plane with respect to the joint mid-plane. Hence, the angle α , shown in figure 17.17(b), by which the yield function is rotated with respect to the original configuration is given by:

$$\alpha = \arctan(\mu_\alpha) = \arctan \left(f \frac{\partial y}{\partial w_t^i} \right) \quad (17.152)$$

For the Gaussian and hyperbolic asperity curves introduced in equations 17.141 and 17.142, the expression of μ_α reads:

$$\mu_\alpha = \frac{f h_0}{s^2} w_t^i \exp\left(-\frac{1}{2} \left(\frac{w_t^i}{s}\right)^2\right) \quad \text{Gaussian} \quad (17.153)$$

$$\mu_\alpha = f \mu_{\bar{\alpha}0} \left(\frac{w_t^i}{\sqrt{(w_t^i)^2 + (r_0 \mu_{\bar{\alpha}0})^2}} \right) \quad \text{hyperbolic} \quad (17.154)$$

The last modification which has to be introduced in Červenka model refers to the expression of the activation function for the inelastic displacement discontinuities.

In the model of Plesha (1987), the activation function φ is written in terms of local tractions transferred along inclined asperities. Expressing φ in the joint reference system corresponds to rotating the activation function by an angle α . In the current proposal the hyperbolic activation function of Červenka's model is modified through the rotation of its asymptotes, thus modifying the current (or apparent) friction angle. In this way an asymmetric activation function is obtained, composed of two branches of hyperbola with the same vertex (the tensile strength), but different inclination of the asymptotes.

Recalling that α represents the current slope of the asperity curve and β the basic friction angle (related to second order asperities), we can define the friction coefficients $\mu_{\beta+\alpha}$ and $\mu_{\beta-\alpha}$ in forward and backward slip, respectively, as follows:

$$\mu_{\beta+\alpha} = \tan(\beta + \alpha) \quad (17.155)$$

$$\mu_{\beta-\alpha} = \tan(\beta - \alpha) \quad (17.156)$$

Because of the asymptote rotation, the apparent cohesion is modified:

$$c = \begin{cases} \frac{c_\beta}{\mu_\beta} \mu_{\beta+\alpha} & \text{forward slip} \\ \frac{c_\beta}{\mu_\beta} \mu_{\beta-\alpha} & \text{backward slip} \end{cases} \quad (17.157)$$

Apparent cohesion c depends on basic cohesion c_β and on the asperity angle α . This, often overlooked, dependency of the cohesion on the asperity angle is recognized in FERC (1999).

On the contrary, the tensile strength is not affected by the presence of first order asperities:

$$\chi = \chi_\beta \quad (17.158)$$

The hyperbolic activation function with rotated asymptotes is shown in figure 17.18(b). Its analytical expression is given by:

$$\varphi = \begin{cases} \left(\frac{\mu_\beta}{\mu_{\beta+\alpha}}\right)^2 p_t^2 - (c_\beta - p_n \mu_\beta)^2 + (c_\beta - \chi_\beta \mu_\beta)^2 & \forall p_t \geq 0 \\ \left(\frac{\mu_\beta}{\mu_{\beta-\alpha}}\right)^2 p_t^2 - (c_\beta - p_n \mu_\beta)^2 + (c_\beta - \chi_\beta \mu_\beta)^2 & \forall p_t < 0 \end{cases} \quad (17.159)$$

It can be observed that the proposed expression of φ has no meaning for $\alpha < \beta$. Moreover, if α is larger than β the inelastic work can have negative increments. Hence, the maximum slope of the asperity curve must satisfy the condition $\alpha < \beta$.

17.3.4.1.4 Remarks In the presented extension of Červenka's model most terms retain their original meaning (such as damage D , inelastic effective displacement w^{eff}), the main difference being that the bilinear softening law for cohesion now applies to c_β rather than to the apparent (or perceived) cohesion c .

Four new independent parameters have been introduced to the original model: two related to the asperity curve (initial asperity length ℓ_0 and initial asperity height h_0); two others modelling the asperity degradation (C and d). At the same time, however, two parameters of the monotonic version of Červenka's model have been discarded: the initial dilatancy angle μ_{d0} and the amount of effective inelastic displacement w_{dil} for which dilatancy μ_d reaches zero. These are now related to the chosen asperity representation.

17.3.5 Computational tests

The predictive capabilities of the enhanced model introduced in the previous sections are shown here by comparison with the results of Kutter and Weissbach (1980) experimental test. Comparison with the predictions of the original Červenka formulation allows to highlight the improvements here proposed.

17.3.5.1 Comparison with Kutter and Weissbach test

The Kutter and Weissbach (1980) experimental results obtained from the *IALAD Network for the Integrity Assessment of Large Dams* (2004) web page are considered in which a cyclic slip was imposed under a constant compressive stress of 2.5 MPa. The test, as described in (Plesha 1987), was performed on a joint in sandstone which was artificially produced by line loading. The specimen was 495 cm² in size.

The cyclic model with hyperbolic asperities has been used. Material parameters are selected in order to produce the best fit. The normal and tangential stiffness K_{n0} and K_{t0} equal 8.26 and 50 MPa/mm respectively. The friction angle β is 34.62 degrees, hence μ_β is 0.69. No tensile strength is assumed, while c_β amounts to 1.42 MPa and decreases linearly with w^{ieff} , hence only one additional parameter, namely fracture energy $G_f^{IIa} = 15.57 \text{ MPa mm}$, is required. As for the asperities, the curvature radius r_0 is 38.86 mm and the angle $\bar{\alpha}_0$ is 10.85 degrees, i.e. $\mu_{\bar{\alpha}0} = 0.192$. The asymptotic degradation parameter d is given a fairly large value so that complete degradation is possible under the imposed normal stress. However, since results at only one confinement are taken into account, the role of d is not so relevant in this analysis. Parameter C governing rate of asperity degradation with inelastic shear work is assumed to be 1.5 m/MN, a rather typical value. The tests of Kutter and Weissbach do not seem to start from an initially mated position, hence an initial inelastic tangential displacement of -2.5 mm is adopted.

The comparison of the two shear strength responses is in figure 17.19(a). The overall result is good, though some differences in shear strength degradation can be noticed, especially for positive shear displacements. The main difference is however in the initial stiffness, which is much lower in the experiment because, as mentioned, the joint is not fully in contact. This feature is not accounted for by the model.

The comparison between the cyclic model and experimental result's dilatancy is in figure 17.19(b). As can be seen the essential features of the response are reasonably well captured, however this figure is also good to highlight some of the already mentioned limitations of the model. In fact seating is not accounted for, nor is a possible different inclination of left and right asperities. The rate of first order asperities' degradation differs from the experimental one too. Maybe the inclusion of the dilatancy associated to second order asperities could improve the results, especially in the first cycle.

Anyway the asperity slope and curvature, and the initial offset (first sliding of the joint is accompanied by a slight contraction) are satisfactorily described by the model.

To summarize, this test well exemplifies capabilities and possible deficiencies of the proposed model. It should be noted that, though the model extension concerns precisely the cyclic shear behavior, the simulation of all cycles of Kutter and Weissbach test is a fairly exigent task.

17.3.5.2 Comparison with Červenka's model

The cyclic model results in Kutter and Weissbach (1980) experiment are now compared with Červenka's model response under the same loading conditions. To this end the latter model parameters are aptly chosen. Cohesion $c_0 = 2.01 \text{ MPa}$ includes the contribution of asperities and is thus obtained from the cyclic model parameters multiplying c_β by the factor $\tan(\beta + \bar{\alpha}_0) / \tan(\beta) = 1.47$ similarly to equation 17.157. Fracture energy $G_f^{IIa} = 22.93 \text{ MPa mm}$ is obtained amplifying G_f^{IIa} of the cyclic model by the same coefficient. Dilatancy $\mu_{d0} = 0.149$ is smaller than $\mu_{\bar{\alpha}0}$ to account for asperity curvature. It is obtained averaging μ_α between 0 and 29.1 mm, i.e. the shear displacement at which the first inversion of sliding occurs in Kutter and Weissbach (1980) experiment. Parameter w_{dil} is obtained imposing that the residual integrity of the asperities is equal for the two models at the end of the test, alias $\mu_d / \mu_{d0} = f$. Friction coefficient μ equals μ_β . All other parameters have the same values of those adopted for the cyclic model.

The shear traction – shear displacement plot is displayed in figure 17.20(a). First of all, it can be noticed how Červenka's model reliably conveys the two most important pieces of information: the peak load and the average shear strength. However, the main feature of Červenka model response is that, once cohesion softening is completed, shear strength is constant irrespective of amount and direction of sliding displacement, i.e. of the characteristics of surface roughness and their degradation. Another difference between the models is in the shear displacement at which peak load is attained: smaller for Červenka that mobilizes instantaneously the maximum dilatancy angle; larger, and nearer to the experimental value, for the cyclic model in which asperity inclination α grows gradually with sliding w_t^i .

Looking at the dilatancy plot in figure 17.20(b), a striking discrepancy can be noted in the amount of predicted joint opening. In fact, despite the apt choice of model parameters, Červenka's model response is quantitatively (almost one order of magnitude) and qualitatively incorrect. Unfortunately this overestimation is also unsafe because it induces a greater normal stress in the surrounding material which in turn allows for larger shear stresses to be transferred across the joint.

The reason for this wrong prediction is the constant sign of dilatancy as sliding direction is inverted. More deeply the cause lies in the almost independent description of shear stress and dilatancy phenomena which are in the cyclic model tightly connected through the asperity description. Indeed the importance of this aspect goes beyond the mere improvement in the modelling of cyclic shear tests.

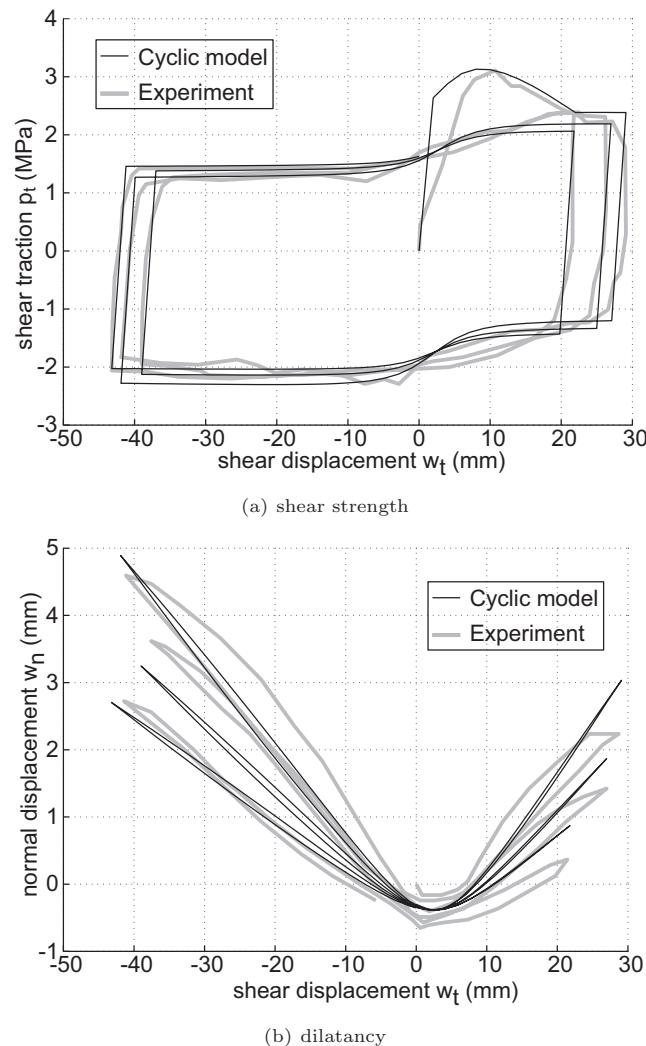
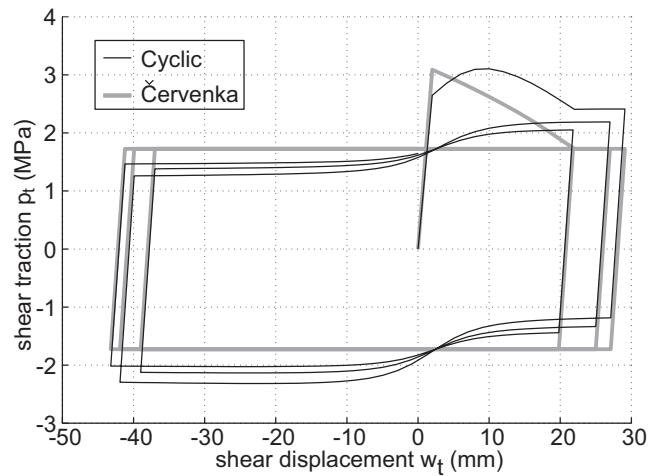
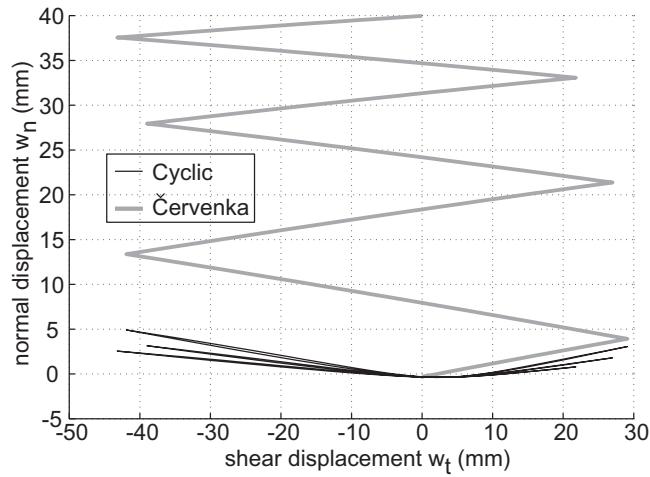


Figure 17.19: Comparison with Kutter-Weissbach test results



(a) shear strength



(b) dilatancy

Figure 17.20: Červenka model vs. cyclic model

Finally it should be remarked how the shortcoming of Červenka's model in figure 17.20(b) is customarily common to all mixed mode quasi brittle joint models devised originally for monotonic analysis, and how a close comparison has been possible in this case due to the similarity of the two models.

17.3.6 Conclusions

A general interface model has been formulated for reproducing the mechanical behavior of joints and cracks in quasi-brittle concrete-like materials under cyclic loading.

The model combines, and enhances, two existing ones: 1) a fracture mechanics based model proposed by Červenka et al. (1998) for concrete cracks, which accounts for loss of tensile strength and normal stiffness in mode I, and for friction and decrease of cohesion in mode II; and 2) an asperity based frictional model proposed by (Plesha 1987) for rough rock joints, which properly models configuration dependent dilation and shear strength of rough joints under cyclic loading conditions.

Numerical simulation of Kutter and Weissbach (1980) experimental tests exhibits encouraging results and provides a useful test to show the model capabilities, while comparison with the (Červenka et al. 1998) model response highlights the extent of the introduced novelties.

The proposed model, suitable for implementation in finite element codes based on either discrete or smeared interpretation of crack, integrates coherently a number of different basic mechanical processes as required by its sought application to dam engineering.

17.3.7 Acknowledgements

This work was largely made possible through the financial support of the Italian Ministry for University and Research (MIUR) which enabled the visit of V.E. Saouma to the Politecnico, along with the accompanying research fund. In addition, the first and second authors would like to acknowledge the support by MIUR through COFIN02 grant on "Concrete dam-foundation-reservoir systems: integrity assessments allowing for interactions".

17.4 Notation

Latin symbols

c	(apparent) cohesion
C	rate of asperity degradation due to inelastic tangential work
c_0	initial cohesion (bilinear softening law)
c_1	cohesion at break point (bilinear softening law)
c_β	joint cohesion in the absence of first order asperities
d	rate of decrease of f_{asym} for increasing compressive stress
D	joint damage in tension
f	asperity integrity (alias degradation) factor
f_{asym}	asymptotic asperity degradation factor (reached for $L_t^i \rightarrow \infty$)
\bar{f}	current value of asperity degradation factor f
G_f^I	mode I fracture energy (softening law)
G_f^{IIa}	mode II fracture energy at high confinement (softening law)
h	asperity height;
h_0	initial asperity height
sK_0	joint stiffness matrix (diagonal)
K_{n0}	initial normal stiffness
K_{nc}	current normal stiffness (degrades in tension)
K_{t0}	initial tangential stiffness
ℓ_0	initial asperity length
L	work
L_t^i	inelastic tangential work
sp	joint stress vector
Q	plastic potential
r	curvature radius in the origin (hyperbolic asperities)
r_0	initial value of curvature radius in the origin (hyperbolic asperities)
s	standard deviation (length) of gaussian asperities
sw	joint displacement discontinuity vector
w_{c0}	value of w^{ieff} at zero cohesion (bilinear softening law)
w_{c1}	value of w^{ieff} at cohesion breakpoint (bilinear softening law)
$w_{\chi 0}$	value of w^{ieff} at zero tensile strength (bilinear softening law)
$w_{\chi 1}$	value of w^{ieff} at tensile strength breakpoint (bilinear softening law)
w_{dil}	value of w^{ieff} at zero dilatancy (Červenka model only)
w^{ieff}	effective inelastic displacement discontinuity (softening variable)
w_n^i	normal inelastic displacement discontinuity
w_t^i	tangential inelastic displacement discontinuity
$y(w_t^i)$	function prescribing the initial asperity shape

Greek symbols

α	first order asperity angle
$\bar{\alpha}_0$	maximum asperity angle (gaussian and hyperbolic asperities)
	asymptotic asperity angle (hyperbolic asperities)
α_{dil}	dilatancy angle
β	second order asperity (friction) angle
γ	irrecoverable portion of total displacement in tension
χ_0	initial tensile strength (bilinear softening law)
χ_1	tensile strength at break point (bilinear softening law)
χ_β	joint tensile strength in the absence of first order asperities
φ	joint activation (alias yield) function
μ	friction coefficient (Červenka model only)
μ_d	tangent of the dilatancy angle (Červenka model only)
μ_{d0}	initial tangent of the dilatancy angle (Červenka model only)
ρ	stiffness reducing coefficient in tension
0	initial value of a quantity
α	quantity related to the first order asperities
β	quantity related to the second order asperities
n	normal component
t	tangential component

Subscripts**Superscripts**

a	joint state at the beginning of the step (numerical implementation)
b	joint state at the end of the elastic portion of the step (numerical implementation)
c	converged joint state at the end of the the step (numerical implementation)
e	elastic quantity (work, displacement or trial stress)
i	inelastic quantity (work or displacement)
p	plastic quantity (irrecoverable displacement in tension)
f	fracture quantity (recoverable displacement in tension)

Operators

μ_\bullet	$\tan(\bullet)$
$\Delta\bullet$	finite increment of \bullet
$\langle \bullet \rangle$	Macaulay brackets $(\bullet + \bullet)/2$

One table for each page at the end of the manuscript

Table 17.1: Summary of parameters adopted for the analyses

par.	units	I	II	III
K_{n0}	MPa/mm	3750	3750	375
K_{t0}	MPa/mm	5000	5000	500
μ		—	0.6	—
μ_β		0.3640	—	0.3640
γ		0.3	0.3	0.3
χ_0	MPa	0.0	0.0	3.0
c_0	MPa	1.5144	2.5	3.0287
G_f^I	MPa mm	0.0	0.0	0.12
G_f^{IIa}	MPa mm	1.1358	1.875	0.1514
χ_1	MPa	0.0	0.0	0.45
c_1	MPa	0.2272	0.375	0.4543
$w_{\chi 1}$	mm	0.05	0.0	0.05
$w_{c 1}$	mm	0.9375	0.9375	0.0625
d	MPa^{-1}	1.0	—	1.0
C	J^{-1}	20	—	50.0
μ_{d0}		—	0.2	—
w_{dil}	mm	—	30	—

17.5 Interface Crack Model; ICM-3-Mohr-Coulomb (MM: 22)

This interface model is a subset of the original ICM-1 (MM: 8) with the following differences:

1. The failure envelope is not hyperbolic, but pure Mohr-Coulomb.
2. There is no dilatancy
3. There is no stiffness degradation

4. There are two different normal stiffnesses for tension and compression
5. There are two different shear stiffnesses for tension and compression
6. There is softening of the tensile strength and cohesion (controlled by G_F^I and G_F^{II} which can be set to zero for perfectly brittle material).
7. Tensile strength can be set to zero.
8. Uplift pressure can be applied along this element.
9. Damping along the crack is possible.

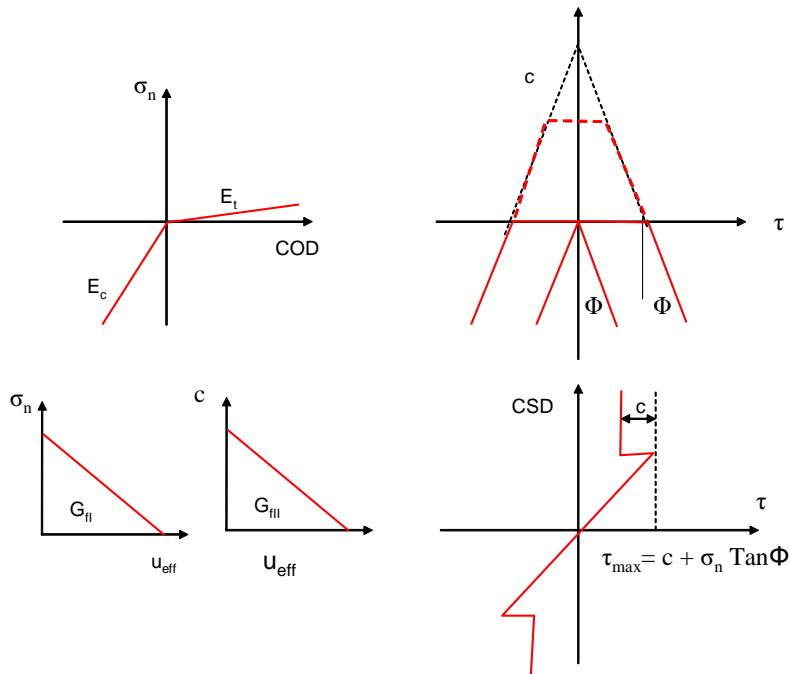


Figure 17.21: ICM-3-Mohr-Coulomb (MM:22)

17.6 Interface Crack Model; ICM-3-Hyperbolic-Light (MM: 23)

This interface model is a subset of the original ICM-1 (MM:8) with the following differences:

1. The failure enveloppe is the hyperbolic one of the original interface crack model (ICM-1). There is no dilatancy
2. There is no stiffness degradation
3. There are two different normal stiffnesses for tension and compression
4. There are two different shear stiffnesses for tension and compression
5. There is softening of the tensile strength and cohesion (controlled by G_F^I and G_F^{II} which can be set to zero for perfectly brittle material).
6. Tensile strength can be set to zero.
7. Uplift pressure can be applied along this element.
8. Damping along the crack is possible.

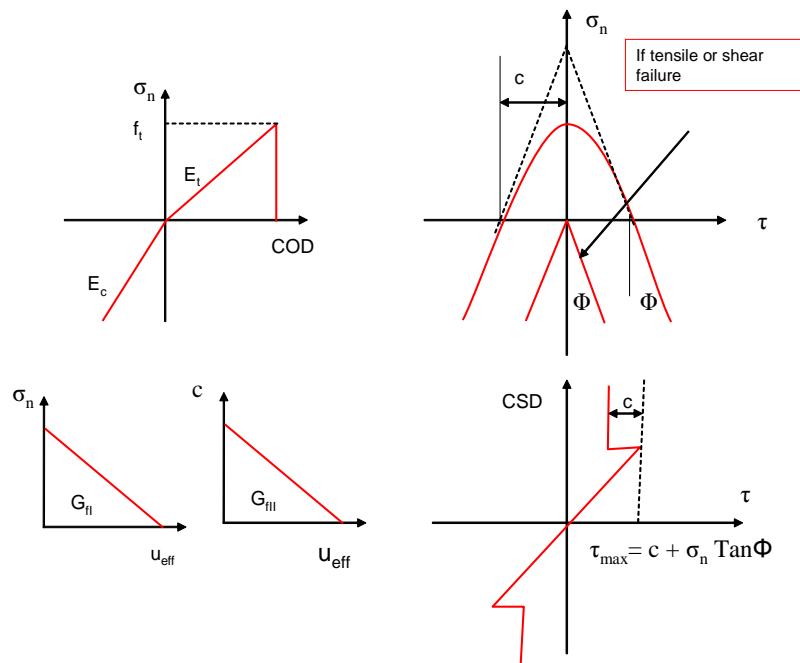


Figure 17.22: ICM-3-Mohr-Coulomb (MM:23)

Chapter 18

DISTRIBUTED FAILURE; Fracture Plastic Model (MM:15, 16, 18, 19)

This chapter covers the implementation of Models 15-16, 18 and 19 in Merlin. It is practically identical to model CC3D ((Červenka, V. and Jendele, L. and Červenka, Jan 2002) in the ATENA Program available from SBETA, Prague.

This fracture-plastic model combines constitutive models for tensile (fracturing) and compressive (plastic) behavior. The fracture model is based on the classical orthotropic smeared crack formulation and crack band model. It employs Rankine failure criterion, exponential (or user defined) softening, and it can be used as rotated or fixed crack model. The hardening/softening plasticity model is based on Menétrey and Willam (1995) failure surface. Both models use return mapping algorithm for the integration of constitutive equations. Special attention is given to the development of an algorithm for the combination of the two models. The combined algorithm is based on a recursive substitution, and it allows for the two models to be developed and formulated separately. The algorithm can handle cases when failure surfaces of both models are active, but also when physical changes such as crack closure occur. The model can be used to simulate concrete cracking, crushing under high confinement, and crack closure due to crushing in other material directions.

The method of strain decomposition, as introduced by de Borst (1986), is used to combine fracture and plasticity models together. Both models are developed within the framework of return mapping algorithm by Wilkins (1964). This approach guarantees the solution for all magnitudes of strain increment. From an algorithmic point of view the problem is then transformed into finding an optimal return point on the failure surface. The combined algorithm must determine the separation of strains into plastic and fracturing components, while it must preserve the stress equivalence in both models. The proposed algorithm is based on a recursive iterative scheme. It can be shown that such a recursive algorithm cannot reach convergence in certain cases such as, for instance, softening and dilating materials. For this reason the recursive algorithm is extended by a variation of the relaxation method to stabilize convergence.

18.1 Material Model Formulation

The material model formulation is based on the strain decomposition into elastic ε_{ij}^e , plastic ε_{ij}^p and fracturing components ε_{ij}^f , (de Borst 1986).

$$\varepsilon_{ij} = \varepsilon_{ij}^e + \varepsilon_{ij}^p + \varepsilon_{ij}^f \quad (18.1)$$

The new stress state is then computed from:

$$\sigma_{ij}^n = \sigma_{ij}^{n-1} + E_{ijkl}(\Delta\varepsilon_{kl} - \Delta\varepsilon_{kl}^p - \Delta\varepsilon_{kl}^f) \quad (18.2)$$

where the increments of plastic strain $\Delta\varepsilon_{kl}^p$ and fracturing strain $\Delta\varepsilon_{kl}^f$ must be evaluated based on the selected material model.

18.2 Rankine-Fracturing Model for Concrete Cracking

Rankine criterion is used for concrete cracking

$$F_i^f = \sigma_{ii}^{tt} - f_{ti}' \leq 0 \quad (18.3)$$

where strains and stresses are expressed in material directions. For rotated cracks those correspond to the principal directions, and for the fixed crack model they correspond to the principal ones at the onset of first cracking. Thus, σ_{ii}^{tt} and f_{ti}' are the trial stress and tensile strength in the local material direction i . Prime symbol denotes quantities in the material directions.

Trial stress is determined from the elastic predictor

$$\sigma_{ij}^{tt} = \sigma_{ij}^{n-1} + E_{ijkl}\Delta\varepsilon_{kl}' \quad (18.4)$$

If Equation 18.3 is violated (i.e. cracking occurs) then the incremental fracturing strain in direction i can be evaluated under the assumption that the final stress state must satisfy

$$F_i^f = \sigma_{ii}^{tn} - f_{ti}' = \sigma_{ii}^{tt} - E_{iikl}\Delta\varepsilon_{kl}'^f - f_{ti}' = 0 \quad (18.5)$$

This equation can be further simplified under the assumption that the increment of fracturing strain is normal to the failure surface, and that always only one failure surface is being checked. Then for surface k the incremental fracturing strain is

$$\Delta\varepsilon'_{ij}^f = \Delta\lambda \frac{\partial F_k^f}{\partial \sigma_{ij}} = \Delta\lambda \delta_{ik} \quad (18.6)$$

substituting into Eq. 18.5, the increment of the fracturing multiplier is recovered as

$$\Delta\lambda = \frac{\sigma_{kk}^{it} - f'_{tk}}{E_{kkkk}} = \frac{\sigma_{kk}^{it} - f'^t(w_k^{max})}{E_{kkkk}} \quad (18.7)$$

where $f'^t(w_k^{max})$ is the softening curve in terms of w which is the current crack opening. The softening diagram adopted in this model is the exponential decay function of Hordijk (1991). The crack opening w is determined from

$$w_k^{max} = L_t(\hat{\varepsilon}_{kk}^f + \Delta\lambda) \quad (18.8)$$

where $\hat{\varepsilon}_{kk}^f$ is the total fracturing strain in direction k , and L_t is the characteristic dimension of the element as introduced by Bažant and Oh (1983), Fig. 18.1. L_t is calculated as a size of the element projected into the crack direction, it is a satisfactory solution for low order linear elements. Equation 18.7 can be solved by recursive

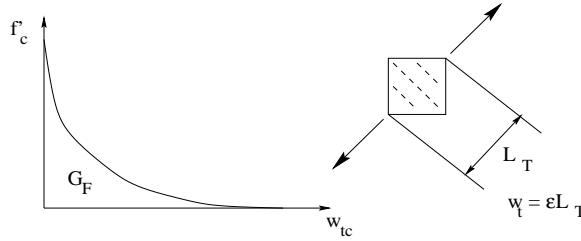


Figure 18.1: Tensile Softening and Characteristic Length, (Červenka, V. and Jendele, L. and Červenka, Jan 2002)

substitution. It can be shown that expanding $f'^t(w_k^{max})$ into a Taylor series, that this iteration scheme converges as long as

$$\left| -\frac{\partial f'^t(w_k^{max})}{\partial w} \right| < \frac{E_{kkkk}}{L_t} \quad (18.9)$$

This equation is violated for softening materials only when snap-back is observed in the stress-strain relationship, which can occur if large finite elements are used. Since in the standard finite element based method, the strain increment is given, therefore, a snap back on the constitutive level can not be captured. Since in the critical region where snap back occurring on the softening curve will be skipped, then the energy dissipated by the system will be over estimated. Because this is undesirable, finite elements smaller than $L < \frac{E_{kkkk}}{\left| \frac{\partial f'^t(0)}{\partial w} \right|}$ should be used, where $\frac{\partial f'^t(0)}{\partial w}$ is

the initial slope of the crack softening curve.

Distinction is made between the total maximum fracturing strain during loading $\hat{\varepsilon}_{kk}^f$ and the current fracturing strain ε'_{ij}^f which is determined according to Rots and Blaauwendraad (1989)

$$\varepsilon'_{kl}^f = (E_{ijkl} + E'_{ijkl})^{-1} E_{klmn} \varepsilon'_{mm} \quad (18.10)$$

$$\sigma'_{ij} = E'_{ijkl} \varepsilon'_{kl}^f \quad (18.11)$$

where E'_{ijkl} is the cracking stiffness in the local material (prime) direction. It is assumed that there is no interaction between normal and shear components thus the crack tensor is given by:

$$E'_{ijkl} = 0 \text{ for } i \neq k \text{ and } j \neq l \quad (18.12)$$

The mode I crack stiffness is

$$E'_{iiii} = \frac{f'^t(w_i^{max})}{\varepsilon'_{ii}^f} \quad (18.13)$$

and mode II and III crack stiffnesses are assumed to be equal to

$$E'_{ijij}^{cr} = \frac{r_g^{ij} G}{1 - r_g^{ij}} \quad (18.14)$$

where $i \neq j$, $r_g^{ij} = \min(r_g^i, r_g^j)$ is the minimum shear retention factors on cracks for the directions i and j and are given by (Kolmar 1986)

$$r_g^i = \frac{-\ln\left(\frac{\varepsilon'_{ii}}{c_1}\right)}{c_2} \quad (18.15)$$

$$c_1 = 7 + 333(\rho - 0.005) \quad (18.16)$$

$$c_2 = 10 - 167(\rho - 0.005) \quad (18.17)$$

where ρ is the reinforcement ratio assuming that it is below 0.002. G is the elastic shear modulus.

For the special cases before the onset of cracking, when the expressions approach infinity. Large penalty numbers are used for crack stiffness in these cases. The shear retention factor is used only in the case of the fixed crack option.

Finally, the secant constitutive matrix in the material direction is analogous to Eq. 18.10 as presented by (Rots and Blaauwendraad 1989)

$$\mathbf{E}'^s = \mathbf{E} - \mathbf{E}(\mathbf{E}'^{cr} + \mathbf{E})^{-1}\mathbf{E} \quad (18.18)$$

which should then be transformed to the global coordinate system $\mathbf{E}^s = \Gamma_\varepsilon^T \mathbf{E}'^s \Gamma_\varepsilon$ where Γ_ε is the strain vector transformation matrix (i.e. global to local strain transformation matrix).

18.3 Plasticity Model for Concrete Crushing

Starting with the predictor-corrector formula, the stress is determined from

$$\sigma_{ij}^n = \sigma_{ij}^{n-1} + E_{ijkl}(\Delta\varepsilon_{kl} - \Delta\varepsilon_{kl}^p) = \sigma_{ij}^t - E_{ijkl}\Delta\varepsilon_{kl}^p = \sigma_{ij}^t - \sigma_{ij}^p \quad (18.19)$$

where σ_{ij}^t is the total stress, and σ_{ij}^p is determined from the yield function via the return mapping algorithm

$$F^p(\sigma_{ij}^t - \sigma_{ij}^p) = F^p(\sigma_{ij}^t - \Delta\lambda l_{ij}) \quad (18.20)$$

The critical component of this equation is l_{ij} which is the return direction defined by

$$l_{ij} = E_{ijkl} \frac{\partial G^p(\sigma_{kl}^t)}{\partial \sigma_{kl}} \quad (18.21)$$

$$\Rightarrow \Delta\varepsilon_{ij}^p = \Delta\lambda \frac{\partial G^p(\sigma_{ij}^t)}{\partial \sigma_{ij}} \quad (18.22)$$

where $G^p(\sigma_{ij})$ is the plastic potential function whose derivative is evaluated at the predictor stress state σ_{ij}^t to determine the return direction.

The adopted failure surface is the one of Menétrey and Willam (1995) which affords much flexibility in its formulation

$$F_{3p}^P = \left[\sqrt{1.5} \frac{\rho}{f'_c} \right]^2 + m \left[\frac{\rho}{\sqrt{6} f'_c} r(\theta, e) + \frac{\xi}{\sqrt{3} f'_c} \right] - c = 0 \quad (18.23)$$

where

$$m = \sqrt{3} \frac{f_c'^2 - f_t'^2}{f_c' f_t'} \frac{e}{e+1} \quad (18.24)$$

$$r(\theta, e) = \frac{4(1-e^2)\cos^2\theta + (2e-1)^2}{2(1-e^2)\cos\theta + (2e-1)\sqrt{4(1-e^2)\cos^2\theta + 5e^2 - 4e}} \quad (18.25)$$

(ξ, ρ, θ) constitute the Heigh-Westergaard coordinates, f'_c and f'_t are the uniaxial compressive and tensile strength respectively. The curvature of the failure surface is controlled by $e \in \langle 0.5, 1.0 \rangle$ (sharp corner for $e = 0.5$, and circular for $e = 1.0$, Fig. 18.2).

The position of the failure surface is not fixed, but rather can move depending on the magnitude of the strain hardening/softening parameter. The strain hardening is based on the equivalent plastic strain which is calculated from $\Delta\varepsilon_{eq}^p = \min(\Delta\varepsilon_{ij}^p)$.

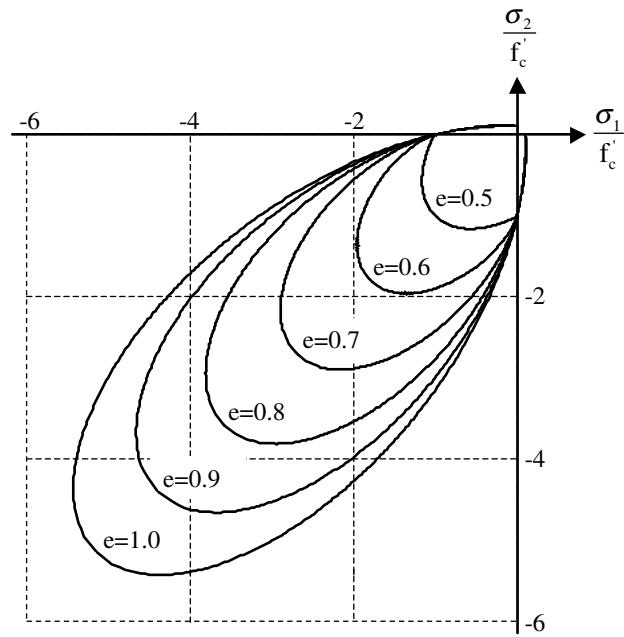


Figure 18.2: Failure Surface

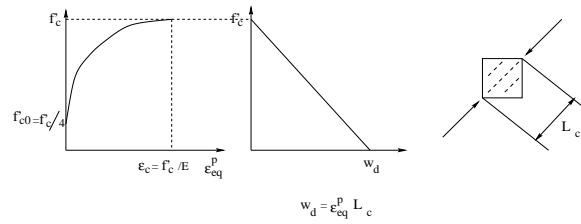


Figure 18.3: Compressive Hardening and Softening, (van Mier 1986)

Hardening/softening is controlled by the parameter $c \in \langle 0, 1 \rangle$, which evolved during the yielding/crushing process according to

$$c = \left(\frac{f'_c(\varepsilon_{eq}^p)}{f'_c} \right)^2 \quad (18.26)$$

where $f'_c(\varepsilon_{eq}^p)$ is the hardening/softening law based on uniaxial test, Fig. 18.3. The law shown in Fig. 18.3 has an elliptical ascending branch and a linear postpeak softening branch after the peak. The elliptical ascending part depends on strains

$$\sigma = f_{c0} + (f_c - f_{c0}) \sqrt{1 - \left(\frac{\varepsilon_c - \varepsilon_{sq}^p}{\varepsilon_c} \right)^2} \quad (18.27)$$

while the descending part is based on relative displacements . In order to introduce mesh objectivity, the descending branch is based on the work of van Mier (1986) where the equivalent plastic strain is transformed into displacements through the length scale L_c . This parameter is defined in an analogous manner to the crack band parameter in the fracture model, Fig. 18.1 and it corresponds to the projection of element size into the direction of minimal principal stresses. The square in Eq. 18.26 is due to the quadratic nature of the Mentrey-Willam surface.

Return direction is given by the following plastic potential

$$G^p(\sigma_{ij}) = \beta \frac{\sqrt{3}}{I_1} + \sqrt{2J_2} \quad (18.28)$$

(2.58) where β determines the return direction. If $\beta < 0$ material is being compacted during crushing, if $\beta = 0$ material volume is preserved, and if $\beta > 0$ material is dilating. In general the plastic model is non-associated, since the plastic flow is not perpendicular to the failure surface. The return mapping algorithm for the plastic model is based on predictor-corrector approach as shown in Fig. 18.4. During the corrector phase of the algorithm the failure surface moves along the hydrostatic axis to simulate hardening and softening. The final failure surface has the apex located at the origin of the Haigh-Westergaard coordinate system. Secant method based Algorithm 1 is used to determine the stress on the surface, which satisfies the yield condition and also the hardening/softening law.

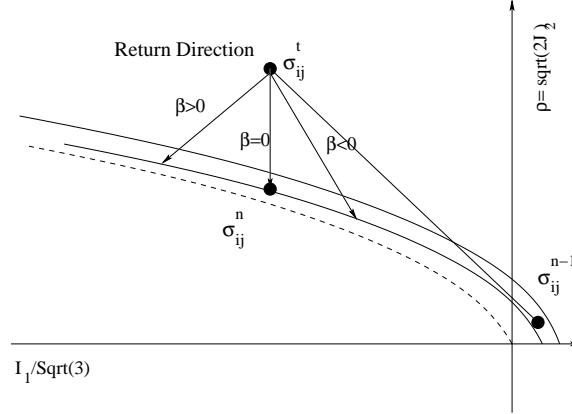


Figure 18.4: Plastic Predictor-Corrector Algorithm, (Červenka, V. and Jendele, L. and Červenka, Jan 2002)

Algorithm 1: Input: $\sigma_{ij}^{n-1}, \varepsilon_{ij}^{p^{n-1}}, \Delta\varepsilon_{ij}^n$

1. Elastic predictor $\sigma_{ij}^t = \sigma_{ij}^{n-1} + E_{ijkl}\Delta\varepsilon_{kl}^n$
2. Evaluate failure criterion: $f_A^p = F^p(\sigma_{ij}^t, \varepsilon_{ij}^{p^{n-1}}, \Delta\lambda_A = 0)$
3. If failure criterion is violated i.e. $f_A^p > 0$
 - a) Evaluate return direction: $m_{ij} = \frac{\partial G^p(\sigma_{ij}^t)}{\partial \sigma_{ij}}$
 - b) Return mapping: $F^p(\sigma_{ij}^t - \Delta\lambda_B E m_{ij}, \varepsilon_{ij}^{p^{n-1}}) = 0 \Rightarrow \Delta\lambda_B$
 - c) Evaluate failure criterion: $f_B^p = F^p(\sigma_{ij}^t - \Delta\lambda_B E m_{ij}, \varepsilon_{ij}^{p^{n-1}}) + \Delta\lambda_B m_{ij}$

- d) Secant iterations as long as $|\Delta\lambda_A - \Delta\lambda_B| < e$
- New plastic multiplier increment: $\Delta\lambda = \Delta\lambda_A - f_A^p \frac{\Delta\lambda_B - \Delta\lambda_A}{f_B^p - f_A^p}$
 - New return direction: $m_{ij}^{(i)} = \frac{\partial G^p(\sigma_{ij}^t - \Delta\lambda E m_{ij}^{(i-1)})}{\partial \sigma_{ij}}$
 - Evaluate failure criterion: $f^p = F^p(\sigma_{ij}^t - \Delta\lambda E m_{ij}^{(i)}, \varepsilon_{ij}^p + \Delta\lambda m_{ij}^{(i)})$
 - New initial values for secant iterations:
- $$\begin{aligned} * f_B^p < 0 &\Rightarrow f_B^p = f^p, \quad \Delta\lambda_B = \Delta\lambda \\ * f_B^p \geq 0 &\Rightarrow f_A^p = f_B^p, \quad \Delta\lambda_A = \Delta\lambda_B, \quad f_B^p = f^p, \quad \Delta\lambda_B = \Delta\lambda \end{aligned} \quad (18.29)$$

e) End of secant iteration loop.

4. End of algorithm update stress and plastic strains. $\varepsilon_{ij}^{p^n} = \varepsilon_{ij}^{p^{n-1}} + \Delta\lambda_B mij^{(i)}$ $\text{chsigma}_{ij}^n = \sigma_{ij}^t - \Delta\lambda_B Em_{ij}^{(i)}$

18.4 Combination of Plasticity and Fracture model

The objective is to combine the above models into a single model such that plasticity is used for concrete crushing and the Rankine fracture model for cracking. This problem can be generally stated as a simultaneous solution of the two following inequalities.

$$F^p(\sigma_{ij}^{n-1} + E_{ijkl}(\Delta\varepsilon_{kl} - \Delta\varepsilon_{kl}^f - \Delta\varepsilon_{kl}^p)) \leq 0 \quad \text{solve for } \Delta\varepsilon_{kl}^p \quad (18.30)$$

$$F^f(\sigma_{ij}^{n-1} + E_{ijkl}(\Delta\varepsilon_{kl} - \Delta\varepsilon_{kl}^p - \Delta\varepsilon_{kl}^f)) \leq 0 \quad \text{solve for } \Delta\varepsilon_{kl}^f \quad (18.31)$$

Each inequality depends on the output from the other one, therefore the following iterative scheme is developed.

Algorithm 2:

- $F^p(\sigma_{ij}^{n-1} + E_{ijkl}(\Delta\varepsilon_{kl} - \Delta\varepsilon_{kl}^{f(i-1)} + b\Delta\varepsilon_{kl}^{cor^{(i-1)}} - \Delta\varepsilon_{kl}^{p(i)})) \leq 0$ solve for $\Delta\varepsilon_{kl}^{p(i)}$
- $F^f(\sigma_{ij}^{n-1} + E_{ijkl}(\Delta\varepsilon_{kl} - \Delta\varepsilon_{kl}^{p(i-1)} - Delta\varepsilon_{kl}^{f(i)})) \leq 0$ solve for $\Delta\varepsilon_{kl}^{f(i)}$
- $\Delta\varepsilon_{ij}^{cor^{(i)}} = \Delta\varepsilon_{ij}^{f(i)} - \Delta\varepsilon_{ij}^{p(i-1)}$
- Iterative correction of the strain norm between two subsequent iterations can be expressed as $\|\Delta\varepsilon_{ij}^{cor^{(i)}}\| = (1-b)\alpha^f\alpha^p\|\Delta\varepsilon_{ij}^{cor^{(i-1)}}\|$ where $\alpha^f = \frac{\|\Delta\varepsilon_{ij}^{f(i)} - \Delta\varepsilon_{ij}^{f(i-1)}\|}{\Delta\varepsilon_{ij}^{p(i)} \Delta\varepsilon_{ij}^{p(i-1)}}$ and $\alpha^p = \frac{\|\Delta\varepsilon_{ij}^{p(i)} - \Delta\varepsilon_{ij}^{p(i-1)}\|}{\Delta\varepsilon_{ij}^{f(i)} \Delta\varepsilon_{ij}^{f(i-1)}}$

b is an iteration correction or relaxation factor, which is introduced in order to guarantee convergence. It is to be determined based on the run-time analysis of α^f and α^p , such that the convergence of the iterative scheme can be assured. The parameters α^f and α^p characterize the mapping properties of each model (i.e. plastic and fracture). It is possible to consider each model as an operator, which maps strain increment on the input into a fracture or plastic strain increment on the output. The product of the two mappings must be contractive in order to obtain a convergence. The necessary condition for the convergence is:

$$|(1-b)\alpha^f\alpha^p| < 1 \quad (18.32)$$

(2.75) If b equals 0, an iterative algorithm based on recursive substitution is obtained. The convergence can be guaranteed only in two cases:

- One of the models is not activated (i.e. implies α^f or $\alpha^p = 0$)
- There is no softening in either of the two models and dilating material is not used in the plastic part, which for the plastic potential in this work means $\beta < 0$, (Eq. 18.28). This is a sufficient but not necessary condition to ensure that α^f and $\alpha^p < 1$.

It can be shown that the values of α^f and α^p are directly proportional to the softening rate in each model. Since the softening model remains usually constant for a material model and finite element, their values do not change significantly between iterations. It is possible to select the scalar b such that the inequality Eq. 18.32 is satisfied always at the end of each iteration based on the current values of α^f and α^p . There are three possible scenarios, which must be handled, for the appropriate calculation of b :

- $|\alpha^f\alpha^p| \leq \chi$, where χ is related to the requested convergence rate. For linear rate it can be set to $\chi = 1/2$. In this case the convergence is satisfactory and $b = -0$.

2. $\chi < |\alpha^f \alpha^p|$, then the convergence would be too slow. In this case b can be estimated as $b = 1 - \frac{|\alpha^f \alpha^p|}{\chi}$ in order to increase the convergence rate.
3. $1 \leq |\alpha^f \alpha^p|$, then the algorithm is diverging. In this case b should be calculated as $b = 1 - \frac{\chi}{|\alpha^f \alpha^p|}$ to stabilize the iterations.

This approach guarantees convergence as long as the parameters does not change drastically between the iterations, which should be satisfied for smooth and correctly formulated models. The rate of convergence depends on material brittleness, dilating parameter β and finite element size. It is advantageous to further stabilize the algorithm by smoothing the parameter b during the iterative process:

$$b = \frac{b^{(i)} + b^{(i-1)}}{2} \quad (18.33)$$

where the superscript i denotes values from two subsequent iterations. This will eliminate problems due to the oscillation of the correction parameter b . Important condition for the convergence of the above Algorithm 2 is that the failure surfaces of the two models are intersecting each other in all possible positions even during the hardening or softening. Additional constraints are used in the iterative algorithm. If the stress state at the end of the first step violates the Rankine criterion, the order of the first two steps in Algorithm 2 is reversed. Also in reality concrete crushing in one direction has an effect on the cracking in other directions. It is assumed that after the plasticity yield criterion is violated, the tensile strength in all material directions is set to zero. On the structural level secant matrix is used in order to achieve a robust convergence during the strain localization process. The proposed algorithm for the combination of plastic and fracture models is graphically shown in Fig. 18.5. When both surfaces are activated,

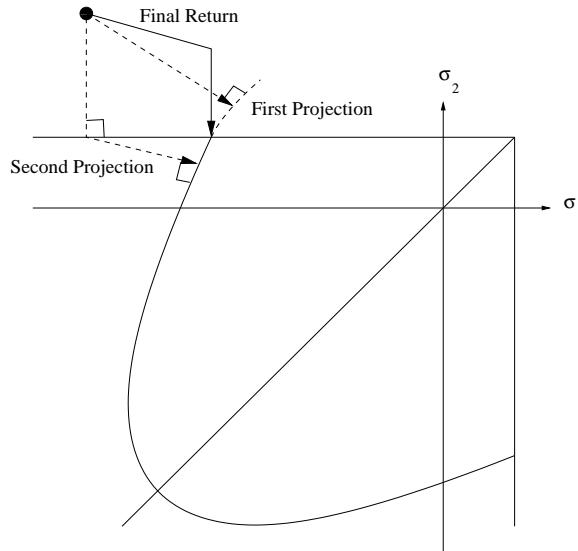


Figure 18.5: Schematic Description of the Iterative Process in 2D, (Červenka, V. and Jendele, L. and Červenka, Jan 2002)

the behavior is quite similar to the multi-surface plasticity (?). Contrary to the multi-surface plasticity algorithm the proposed method is more general in the sense that it covers all loading regimes including physical changes such as for instance crack closure. Currently, it is developed only for two interacting models, and its extension to multiple models is not straightforward.

18.5 Confinement Sensitive Fracture-Plastic Model, MM: 18

Main model features:

Failure Surface of Menétrey and Willam (1995) which affords much flexibility in its formulation

$$F_{3p}^P = \left[\sqrt{1.5} \frac{\rho}{f'_c} \right]^2 + m \left[\frac{\rho}{\sqrt{6} f'_c} r(\theta, e) + \frac{\xi}{\sqrt{3} f'_c} \right] - c = 0 \quad (18.34)$$

where

$$m = \sqrt{3} \frac{f'_c - f'_t}{f'_c f'_t} \frac{e}{e+1} \quad (18.35)$$

$$r(\theta, e) = \frac{4(1-e^2) \cos^2 \theta + (2e-1)^2}{2(1-e^2) \cos \theta + (2e-1) \sqrt{4(1-e^2) \cos^2 \theta + 5e^2 - 4e}} \quad (18.36)$$

(ξ, ρ, θ) constitute the Heigh-Westergaard coordinates, f'_c and f'_t are the uniaxial compressive and tensile strength respectively. The curvature of the failure surface is controlled by $e \in \langle 0.5, 1.0 \rangle$ (sharp corner for $e = 0.5$, and circular for $e = 1.0$, Fig. 18.6).

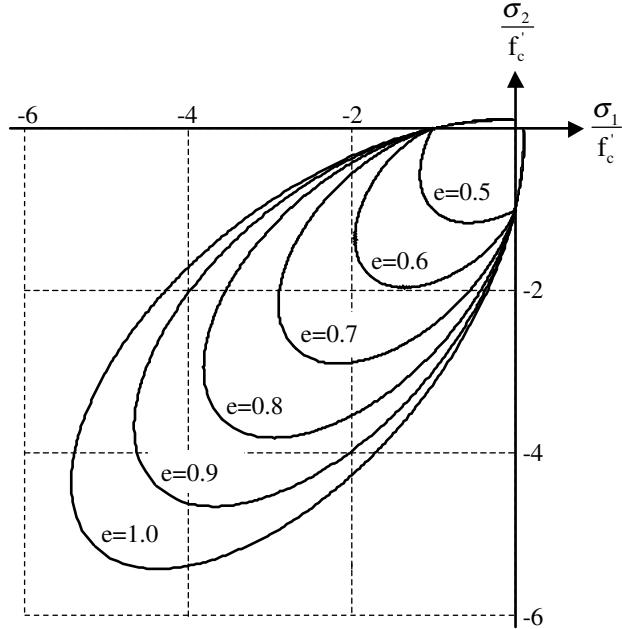


Figure 18.6: Failure Surface

The position of the failure surface is not fixed, but rather can move depending on the magnitude of the strain hardening/softening parameter. The strain hardening is based on the equivalent plastic strain which is calculated from $\Delta\varepsilon_{eq}^p = \min(\Delta\varepsilon_{ij}^p)$.

Plastic Potential is given by, Fig. ??

$$g = A \cdot \left(\frac{\rho}{k \cdot \sqrt{c} \cdot f_c} \right)^n + \left[C + \frac{1}{2}(B-C)(1-\cos 3\theta) \right] \cdot \frac{\rho}{k \cdot \sqrt{c} \cdot f_c} + \frac{\xi}{k \cdot \sqrt{c} \cdot f_c} - a \quad (18.37)$$

where $n = 3$ A , B , and C are constants that can be defined by user. Currently, in the model they are hardcoded for most typical concrete types to be: $A = 5.436$; $B = -6.563$; $C = -3.256$. These parameters were determined by fitting the evolution of plastic strains from many experimental results published in the literature. Parameter a is not needed, since in the model formulation only a derivative of g with respect to ρ and ξ is required.

Tensile softening Exponential Crack Opening Law, Fig. 18.8

The function of crack opening was experimentally derived by (Hordijk 1991)

$$\frac{\sigma}{f_t^{ef}} = \left\{ 1 + \left(c_1 \frac{w}{w_c} \right)^3 \right\} \exp \left(-c_2 \frac{w}{w_c} \right) - \frac{w}{w_c} (1 + c_1^3) \exp(-c_2) \quad (18.38)$$

where

$$w_c = 5.14 \frac{G_f}{f_t^{ef}} \quad (18.39)$$

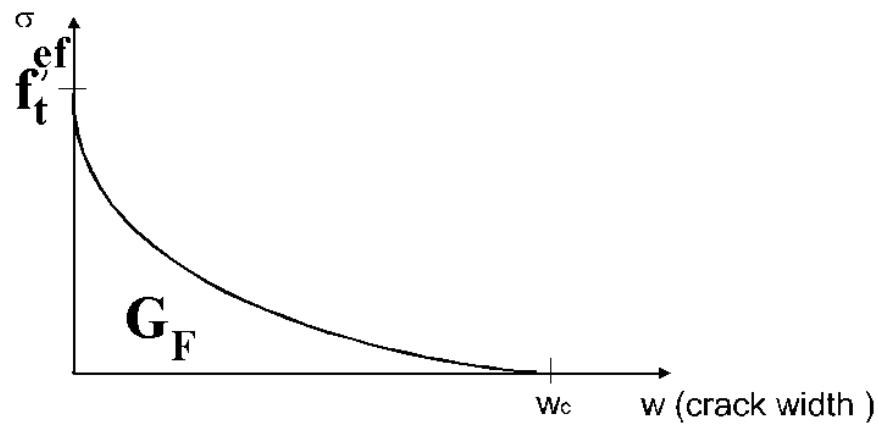


Figure 18.7: Plastic Potential of Model 18

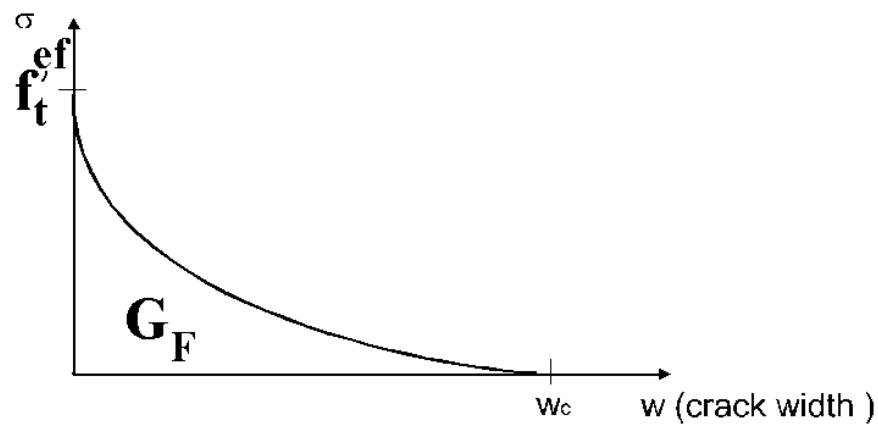


Figure 18.8: Exponential Crack Opening Law

and w is the crack opening, w_c is the crack opening at the complete release of stress, σ is the normal stress in the crack (crack cohesion). Values of the constants are, $c_1 = 3$, $c_2 = 6.93$. G_f is the fracture energy needed to create a unit area of stress-free crack, $f_t'^{ef}$ is the effective tensile strength based on equation.

$$f_t'^{ef} = f_t' r_{et} \quad (18.40)$$

where r_{et} is the reduction factor of the tensile strength in the direction 1 due to the compressive stress in the direction 2. The reduction function has the following form

$$r_{et} = 1 - 0.8 \frac{\sigma_{c2}}{f_c} \quad (18.41)$$

The crack opening displacement w is derived from strains according to the crack band theory of Bažant, Z.P. and Oh, B.H. (1983)

Compressive hardening/softening Hardening/softening parameter in the present model is set equal to the plastic volumetric strain (ε_v^p), suggested by Grassl, Lungren and Gylltoft (2002).

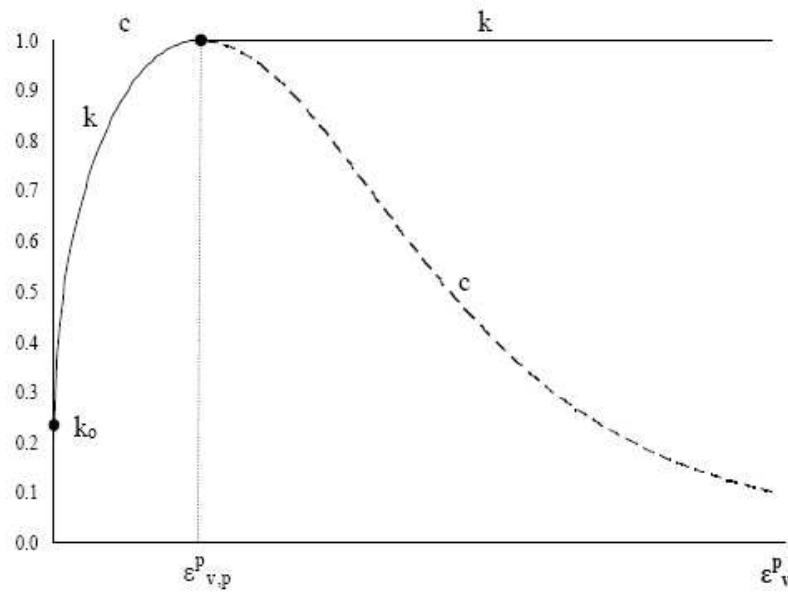


Figure 18.9: Compressive Hardening/Softening

Hardening:

$$k(\varepsilon_v^p) = k_0 + (1 - k_0) \cdot \sqrt{1 - \left(\frac{\varepsilon_{v,p}^p - \varepsilon_v^p}{\varepsilon_{v,p}^p} \right)^2} \quad (18.42)$$

where $k_0 = \frac{f_c^{0.885}}{60}$ based on Com (1990). The value of input parameter $\varepsilon_{v,p}^p$ for typical concrete can be estimated using the formula:

$$\varepsilon_{v,p}^p = \frac{f_c}{E_c} (1 - 2\nu) \quad (18.43)$$

Softening:

$$c(\varepsilon_v^p) = \left(\frac{1}{1 + \left(\frac{n-1}{n_2-1} \right)^2} \right)^2 \quad (18.44)$$

where

$$n = \frac{\varepsilon_v^p}{\varepsilon_{v,p}^p} \quad (18.45-a)$$

$$n_2 = \frac{\varepsilon_{v,p}^p + t}{\varepsilon_{v,p}^p} \quad (18.45-b)$$

Shear retention factor When cracking occurs the shear modulus is reduced according to the law derived by (Kolmar 1986) after cracking. The shear modulus is reduced with growing strain normal to the crack, and this represents a reduction of the shear stiffness due to the crack opening, Fig. 18.10. where

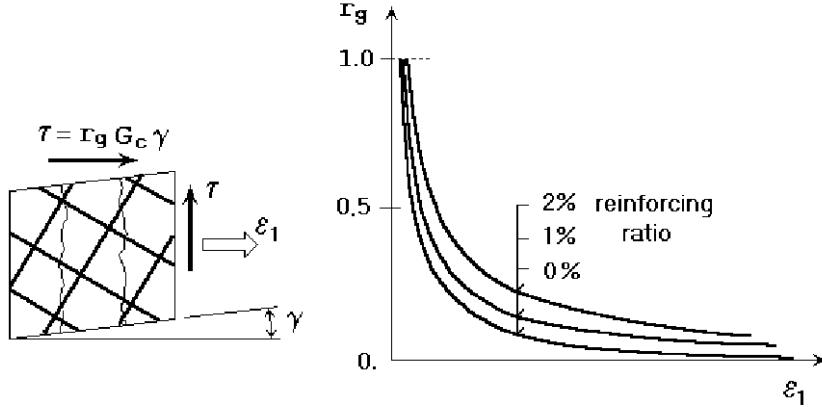


Figure 18.10: Shear Retention Factor

$$G = r_g G_c \quad (18.46-a)$$

$$r_g = \frac{-\ln\left(\frac{1000\varepsilon_u}{c_1}\right)}{c_2} \quad (18.46-b)$$

$$c_1 = 7 + 333(p - 0.005) \quad (18.46-c)$$

$$c_2 = 10 - 167(p - 0.005) \quad (18.46-d)$$

$$0 \leq p \leq 0.02 \quad (18.46-e)$$

where r_g is the shear retention factor, G is the reduced shear modulus and G_c is the initial concrete shear modulus

$$G_c = \frac{E_c}{2(1+\nu)} \quad (18.47)$$

where E_c is the initial elastic modulus and ν is the Poisson's ratio. The strain ε_u is normal to the crack direction (the crack opening strain), c_1 and c_2 are parameters depending on the reinforcing crossing the crack direction, p . The effect of reinforcement ratio is not considered, and p is assumed to be 0.0.

Compressive strength reduction due to cracking A reduction of the compressive strength after cracking in the direction parallel to the cracks is done by a similar way as found from experiments of Vecchio and Collins (1986) and formulated in the Compression Field Theory. However, a different function is used for the reduction of concrete strength here, in order to allow for user's adjustment of this effect. This function has the form of the Gauss's function. The parameters of the function were derived from the experimental data published by (Kollegger and Mehlhorn 1988) which included also data of Collins and (Vecchio and Collins 1986).

$$f_c'^{ef} = r_c f_c', r_c = c + (1 - c)e^{-(128\varepsilon_u)^2} \quad (18.48)$$

For the zero normal strain, ε_u there is no strength reduction, and for the large strains, the strength is asymptotically approaching to the minimum value $f_c'^{ef} = c f_c'$. The constant c represents the maximal strength reduction

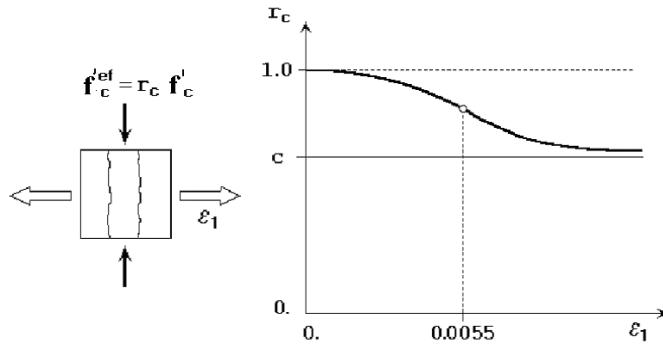


Figure 18.11: Compressive Strength Reduction of Cracked Model

under the large transverse strain. From the experiments by ?, the value $c = 0.45$ was derived for the concrete reinforced with the fine mesh. The other researchers DYNGEELAND 1989 found the reductions not less than $c = 0.8$. The value of c can be adjusted by input data according to the actual type of reinforcing.

Comparison between analytical and experimental results, (Kotsovos and Newman 1980) for normal concrete under triaxial compression and various confinement levels is shown in Fig. 18.12.

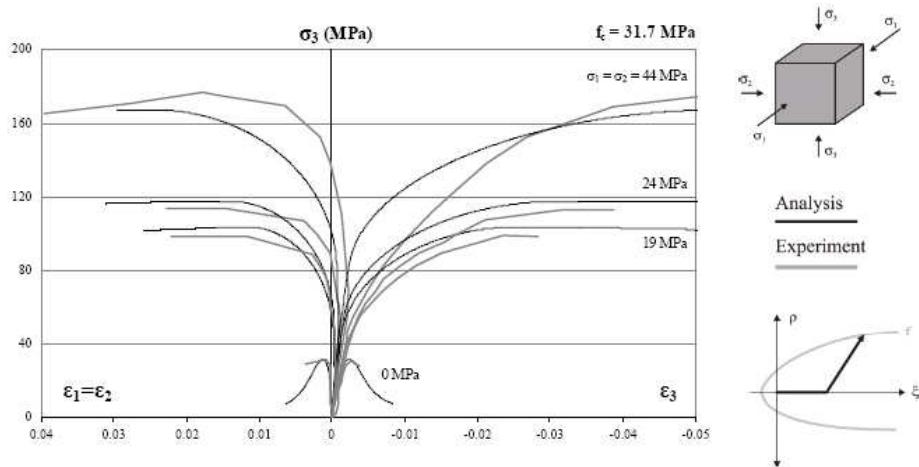


Figure 18.12: Comparison between analytical and experimental results for normal concrete under triaxial compression and various confinement levels

Comparison between analytical and experimental results (Candappa, Sanjayan and Setunge 2000) for high-strength concrete under triaxial compression and various confinement level is shown in Fig. 18.13

18.5.1 Summary of Main Improvements over MM 19

18.5.1.1 Confinement Sensitivity of Stress-Strain response

The previous Model 19 was able to correctly capture the confinement affect on the biaxial or triaxial compressive strength, however the strains at the compressive strength did not show any effect of confinement levels, which was in contradiction to experimental evidence. This behavior is improved in the new model, and it enhances significantly the model behavior in triaxial stress state. This can be demonstrated on an example of a concrete cube. The cube is loaded in vertical direction by gradually increasing deformation, while the tractions (i.e. confinement) in the lateral directions x and y are kept constant, Fig. 18.14 Fig. 18.15 shows the stress-strain response for different confinement levels for a normal concrete with compressive strength 28 MPa. The solid lines show the response of the new model

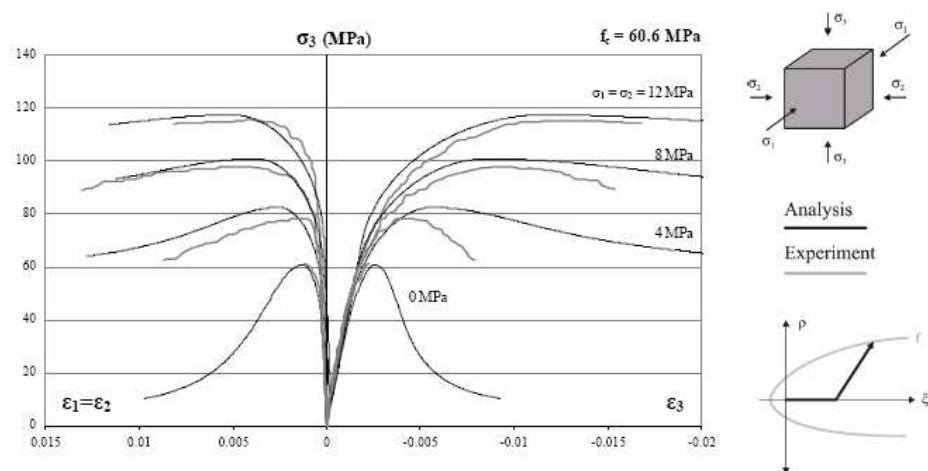


Figure 18.13: Comparison between analytical and experimental results for high-strength concrete under triaxial compression and various confinement level

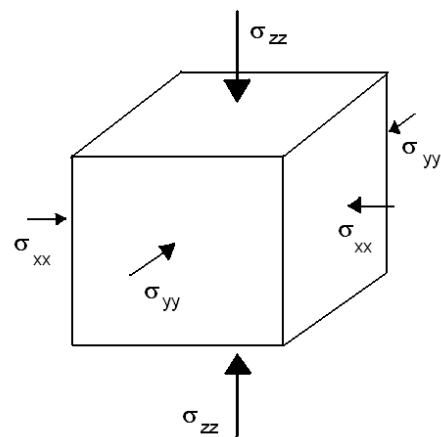


Figure 18.14: Laterally Confined Cube (in x and y while monotonically Loaded in the z Direction)

while the dashed lines represent the behavior of the older model. The graphs clearly show that the peak strength values for the two models are identical, and correctly capture the confinement effect. On the other hand, in the old model 19, the strain values when the strength is reached are almost identical for all confinement levels. This can be contrasted with the behavior of the new model 18, where the peak correctly shifts to higher strain values with increasing confinement.

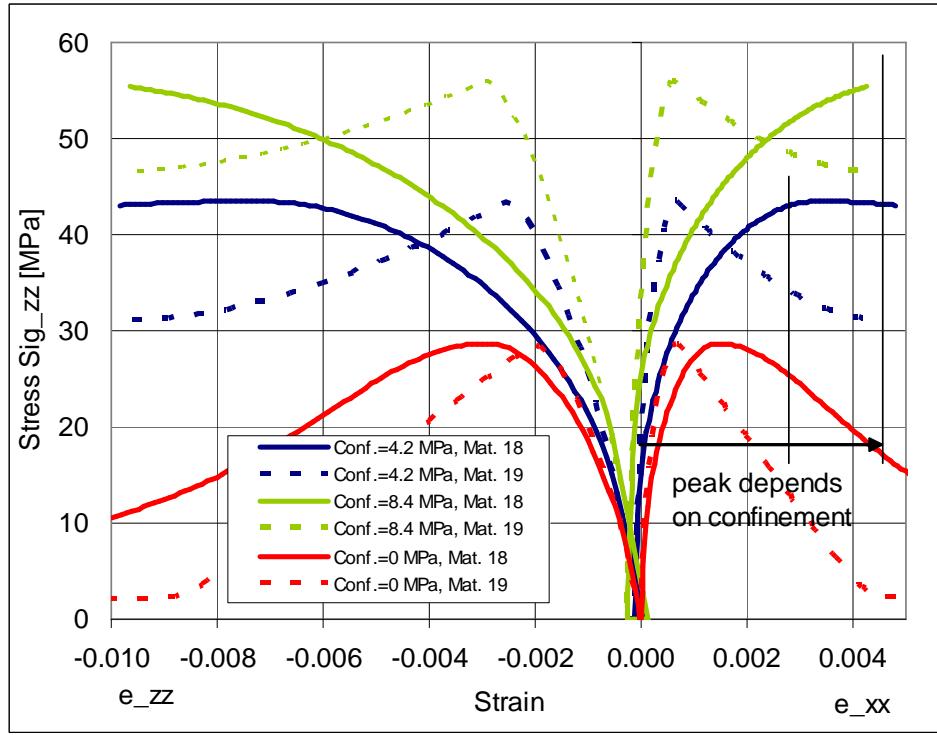


Figure 18.15: Stress-strain response of the triaxial test for different confinement lateral stresses (0, 4.2, 8.4 MPa)

18.5.1.2 Shear retention factor

The new model includes a direct formula for shear retention factor, this means it is possible to directly adjust the shear stiffness of the cracked concrete.

18.5.1.3 Elements of compression field theory

In shear dominated problems, very often the final failure occurs due the compressive crushing of the shear diagonal. This behavior has a phenomenological explanation by the so called “compression” field theory (Vecchio and Collins 1986). The basis of this theory is the decrease of concrete compressive strength which depends on the cracking in perpendicular directions. The new features of the material model such as shear retention factor and compression field theory improves the model behavior in shear problems. This can be demonstrated on the example of a shear beam with four point loading. The geometry and material properties correspond to the test setup of Leonhardt and Walther (1962), Fig. 18.16.

The experiment predicted peak load between 60-70 kN. This is predicted more accurately by the new version of the model as can be seen from Fig. 18.17.

18.5.1.4 Improved model stability

The model stability has been improved as can be seen from the following results for a three point bend beam test, Fig. 18.18.

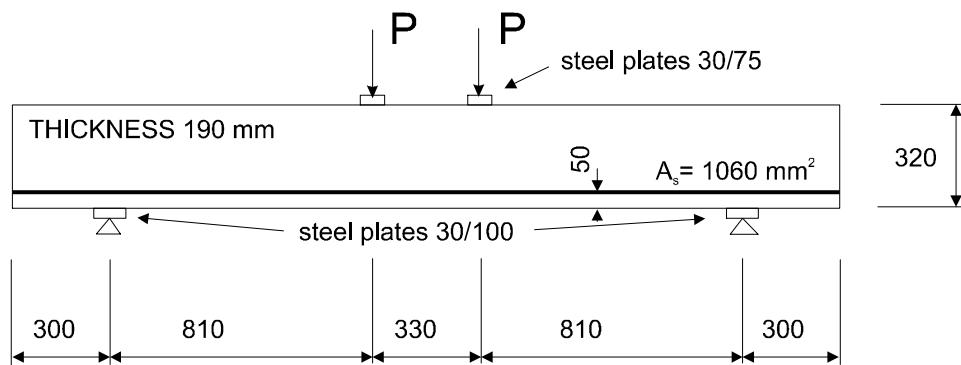


Figure 18.16: Geometry of the Leonhardt Beam

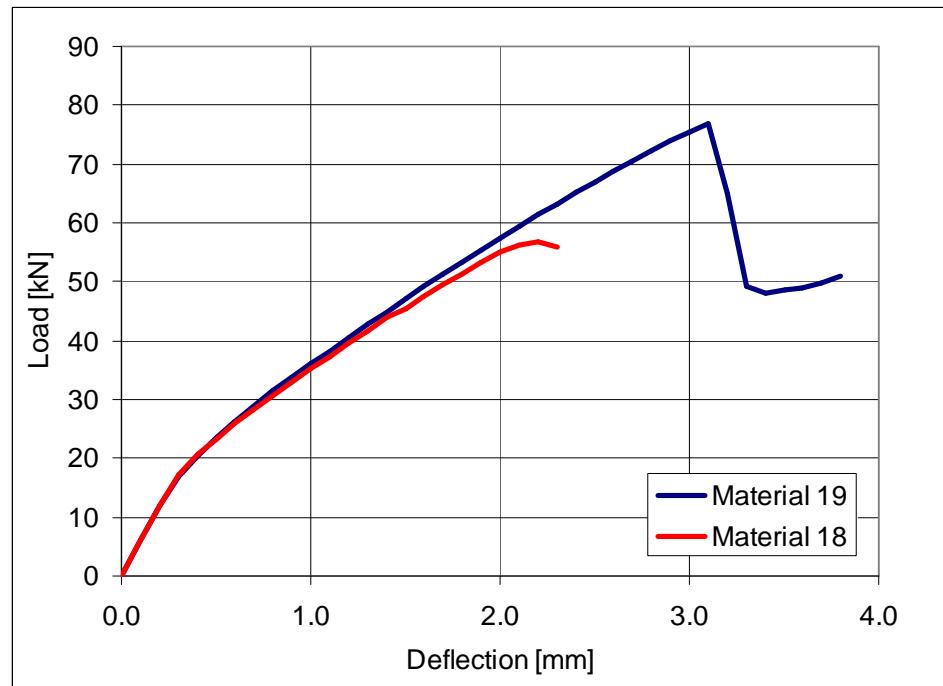
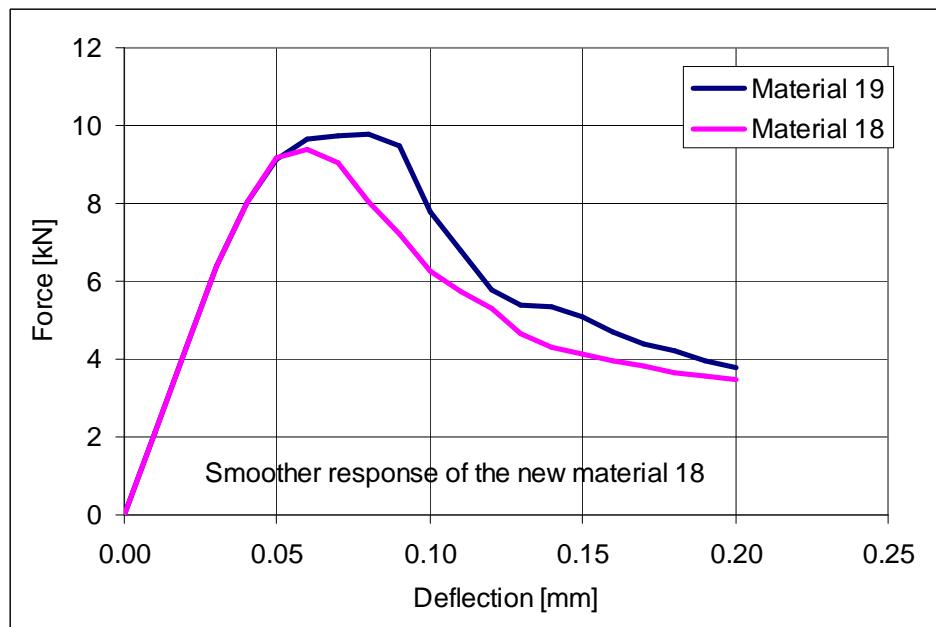


Figure 18.17: Analysis of Leonhardt Shear Beam with Model 18 and 19



18.6 Validation Test Problems, MM-19

This section is taken from a report submitted by Saouma and Perotti to Edison, Italy

18.6.1 Descrizione del provino

La validazione del legame costitutivo non lineare è stata eseguita tramite delle analisi su un cubetto di calcestruzzo con lato di 30 cm. Vengono simulate sul provino delle prove di trazione e di compressione uniaxiale; le caratteristiche del calcestruzzo che compone il provino sono indicate nella seguente Tabella 18.1. Si deve notare che le caratteristiche del calcestruzzo sono identiche a quelle adottate nelle analisi dell'elemento centrale della diga.

Densità di massa	2400 Kg/m ³
Coefficiente di espansione termica	10–5 °C ⁻¹
Modulo elastico	18000 MPa
Coefficiente di Poisson	0,2
Resistenza a trazione	1,5 MPa
Energia specifica di frattura	140 N/m
Resistenza a compressione	-32 MPa
Sforzo a compressione da cui inizia la non linearità	-20 MPa

Table 18.1: Caratteristiche del calcestruzzo utilizzato durante le prove di validazione del legame costitutivo

	No. Nodi	Gradi di libertà	No. Elementi
Coarse	46	138	116
Medium	338	1014	1340

Table 18.2: Caratteristiche delle mesh utilizzate nelle prove sul cubo di calcestruzzo

Nel corso delle analisi sul cubetto di calcestruzzo sono stati utilizzati due tipi di mesh, che differiscono per il numero di elementi. Le due mesh utilizzate sono identificate dai nomi "Medium" e "Coarse" e le loro caratteristiche sono riportate in Tabella 18.2 mentre le loro immagini sono riportate in Figura 18.19

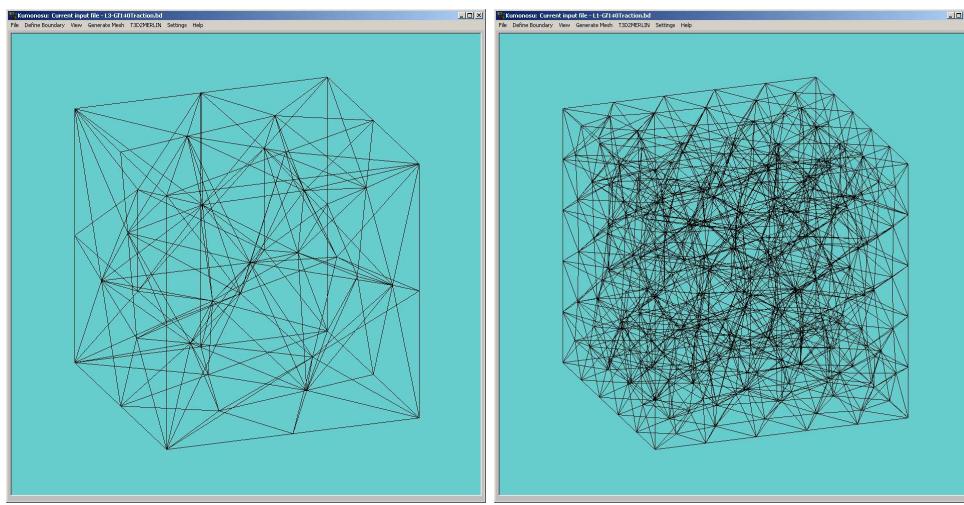


Figure 18.19: In figura sono mostrate da sinistra verso destra le immagini della mesh “Coarse” e “Medium”

18.6.2 Prova uniassiale di trazione

Viene simulata sul cubo di calcestruzzo una prova uniassiale di trazione in controllo di spostamento. La faccia inferiore del cubo è vincolata nella direzione di applicazione del carico e lo spostamento verso l’alto è imposto in 50 incrementi di 0,006 mm ciascuno. In Tabella 18.3 viene descritta nei particolari la prova di trazione a cui è soggetto il cubo di calcestruzzo.

Numero di incrementi	50	
Spostamento per incremento	0.006	mm
Spostamento totale	0.3	mm
Deformazione finale	0.1	%

Table 18.3: Descrizione della prova uniassiale di trazione

18.6.2.1 L’effetto della mesh

L’effetto della densità della mesh è stato analizzato utilizzando nella prova di trazione uniassiale sia la mesh “Medium”, sia la mesh “Coarse”. Dalle curve in Figura 18.20 si nota che in entrambi i casi si riesce a modellare il comportamento “softening” del calcestruzzo. I risultati ottenuti con la mesh “Medium” sono meno precisi poiché il ramo di “softening” del calcestruzzo è composto da due parti distinte, fra le quali esiste una discontinuità attribuibile solo all’algoritmo risolutivo e non al comportamento reale del calcestruzzo.

18.6.2.2 L’effetto della localizzazione del difetto

Il legame costitutivo implementato in MERLIN prevede la nascita di fessure diffuse all’interno del materiale. Questo fatto implica che l’area sottesa alla curva carico spostamento sia maggiore dell’energia di frattura impostata nelle analisi. Se si localizza il difetto si ottiene invece un’area sottesa alla curva carico spostamento che meglio stima l’energia di frattura. Questa approssimazione migliora con il diminuire della zona in cui si inserisce il difetto e quindi con l’aumentare della localizzazione del difetto stesso. In quest’ultimo caso ci si avvicina ad una frattura localizzata e non diffusa.

La seguente prova è stata pensata per mostrare l’effetto di localizzazione del difetto. Due cubi di calcestruzzo sono stati sottoposti alla stessa prova di trazione uniassiale: il primo cubo è composto da un calcestruzzo con proprietà omogenee, mentre il secondo cubo è composto da tre strati di materiale. Gli strati di materiale differiscono solo per il valore della resistenza a trazione, che viene artificialmente diminuita di 0,1 MPa nello strato intermedio di calcestruzzo.

In Figura 18.21 è riportato il confronto fra la mesh del provino con difetto prima e dopo la prova uniassiale di trazione. Si nota che, utilizzando il provino composto da tre strati di materiale, la deformazione si localizza nello strato

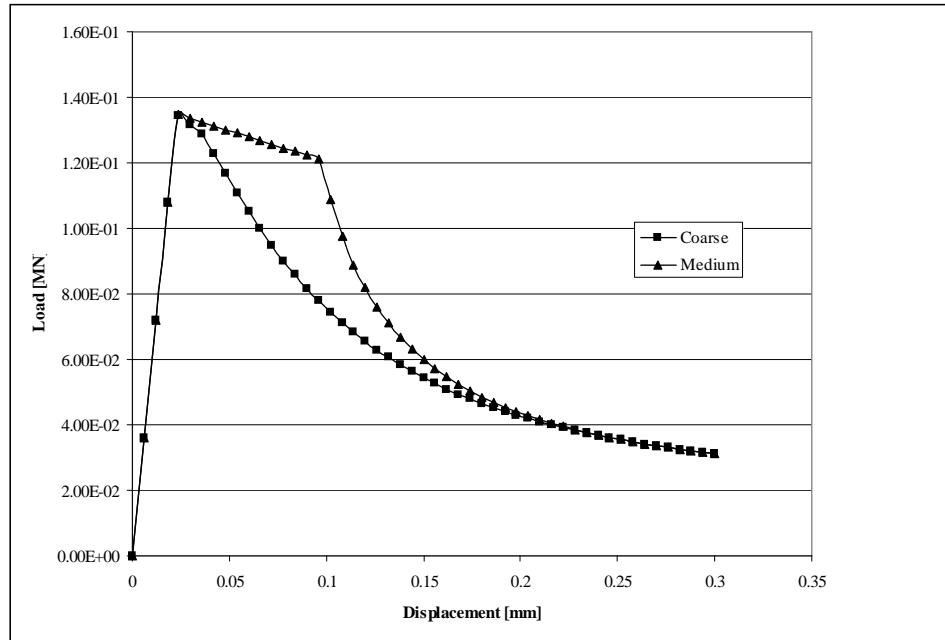


Figure 18.20: Curva carico spostamento per la mesh "Coarse" e "Medium"

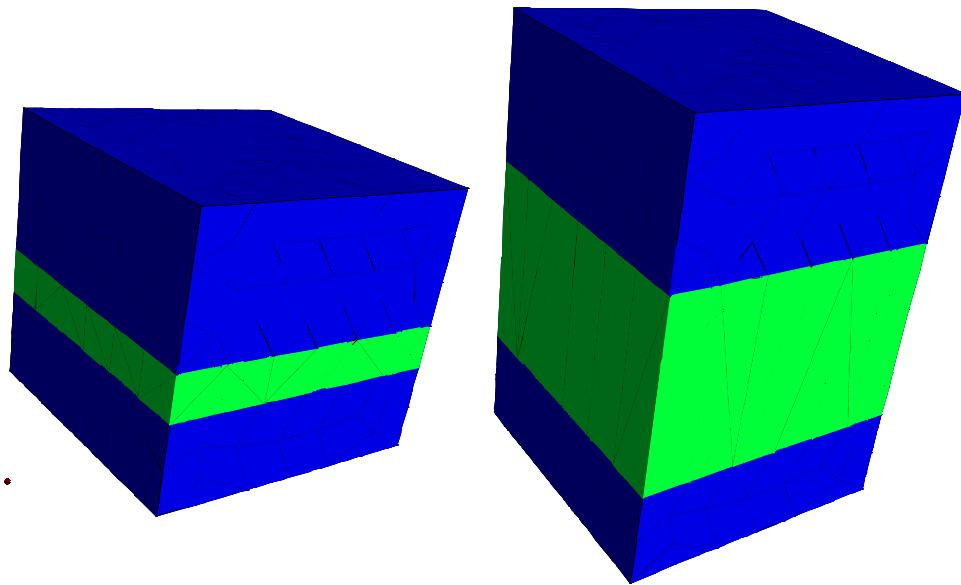


Figure 18.21: Mesh del cubo di calcestruzzo artificialmente indebolito prima e dopo la prova uniaxiale di trazione. La resistenza a trazione viene diminuita di 0,1 MPa all'interno dello strato verde di calcestruzzo.

pi debole. In questo strato del cubo si localizzano le fessure e l'area sottesa alla curva carico spostamento si avvicina maggiormente all'energia di frattura imposta nell'analisi. Il grafico in Figura fig:lp-cer-3-11 mostra chiaramente come la curva relativa al cubo senza imperfezioni sottenda un'area maggiore della curva relativa al cubo artificialmente indebolito

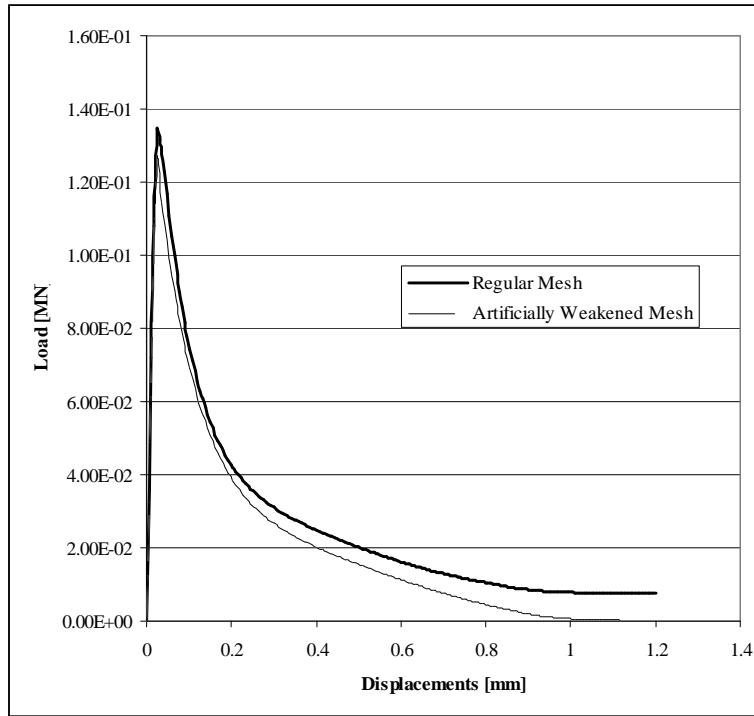


Figure 18.22: Curve carico spostamento per il cubo senza imperfezioni e per il cubo artificialmente indebolito

In Tabella 18.4 viene calcolata l'area sottesa alle due curve in Figura 18.22 e si confrontano i dati teorici con quelli ottenuti dalle prove di trazione simulate in MERLIN. Si nota che l'energia teorica di frattura è sempre inferiore all'energia calcolata numericamente dalla prova di trazione ma questo divario diminuisce utilizzando la mesh con imperfezione.

Tipo di Mesh	Energia teorica ($G_F * 0.3 * 0.3$)	Energia calcolata numericamente
Mesh regolare	12.6 J	29.89 J
Mesh indebolita artificialmente	12.6 J	23.07 J

Table 18.4: Energia di frattura teorica e calcolata in base alle prove di trazione simulate con il programma MERLIN

18.6.2.3 Effetto dell'energia di frattura sui risultati delle prove uniassiali di trazione

La fragilità del materiale dipende dal valore dell'energia di frattura e si ottengono comportamenti più fragili al diminuire di G_f .

Le prove di trazione simulate con MERLIN per cogliere l'infrafilamento del materiale al diminuire di G_f consistono nell'utilizzare due tipi differenti di materiale ed entrambe le mesh "Coarse" e "Medium". Dal grafico in Figura 18.23 notiamo come l'area sottesa alle curve diminuisca passando dal calcestruzzo con G_f uguale a 140 N/m al calcestruzzo con G_f pari a 50 N/m. Inoltre il ramo di "softening" delle diverse curve mostra una diminuzione di resistenza più brusca al diminuire dell'energia di frattura.

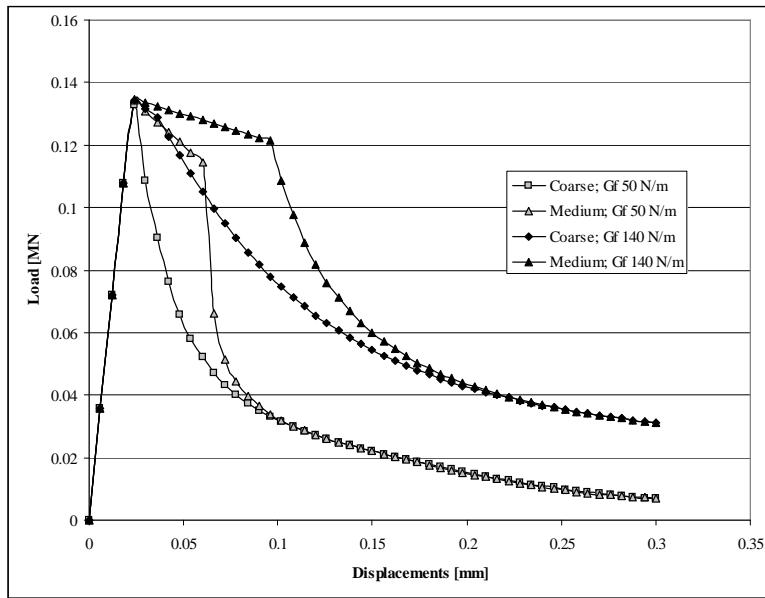


Figure 18.23: Curve carico spostamento relative alla prove di trazione uniaxiale con differenti valori dell'energia di frattura

18.6.3 Prova uniaxiale di compressione

Viene simulata sul cubo di calcestruzzo una prova di compressione uniaxiale in controllo di spostamento. La faccia inferiore del cubo è vincolata nella direzione di applicazione del carico e lo spostamento verso il basso è imposto in 50 incrementi di 0,03 mm ciascuno. In Tabella 18.5 viene descritta nei particolari la prova di compressione a

Numero di incrementi	50	
Spostamento per incremento	0.03	mm
Spostamento totale	1.5	mm
Deformazione finale	0.5	%

Table 18.5: Descrizione della prova uniaxiale di compressione

cui è soggetto il cubo di calcestruzzo. Le prove uniaxiali di compressione vogliono mostrare la capacità del legame costitutivo del calcestruzzo di cogliere due aspetti fondamentali:

1. la non linearità presente nel ramo di carico della curva superato un valore prefissato di sforzo di compressione;
2. il "softening" dopo il raggiungimento della resistenza massima a compressione.

Le curve in Figura 18.24 mostrano entrambi questi aspetti della legge costitutiva del calcestruzzo. Inoltre si nota che, al variare della mesh, le due curve carico spostamento seguono lo stesso ramo di carico, ma si differenziano nel ramo softening post-picco.

18.6.4 Imposizione di un carico termico

L'espansione dovuta alla reazione alcali-aggregati viene simulata nelle analisi applicando una variazione di temperatura, tale da riprodurre uno spostamento verticale del coronamento pari a 30mm. Per testare il legame costitutivo non lineare in presenza di un'espansione termica si sono eseguite due semplici prove:

1. l'espansione libera del cubo di calcestruzzo soggetto ad una crescita progressiva di temperatura per incrementi;
2. l'espansione vincolata del cubo in calcestruzzo soggetto ad una crescita progressiva di temperatura analoga alla precedente. Il cubo presenta due facce vincolate in direzione normale alle facce stesse; le altre facce sono libere.

Dalle mesh deformate al termine delle due prove (Figura 18.25) si nota il diverso comportamento deformativo in assenza ed in presenza di un vincolo di contenimento durante l'espansione termica.

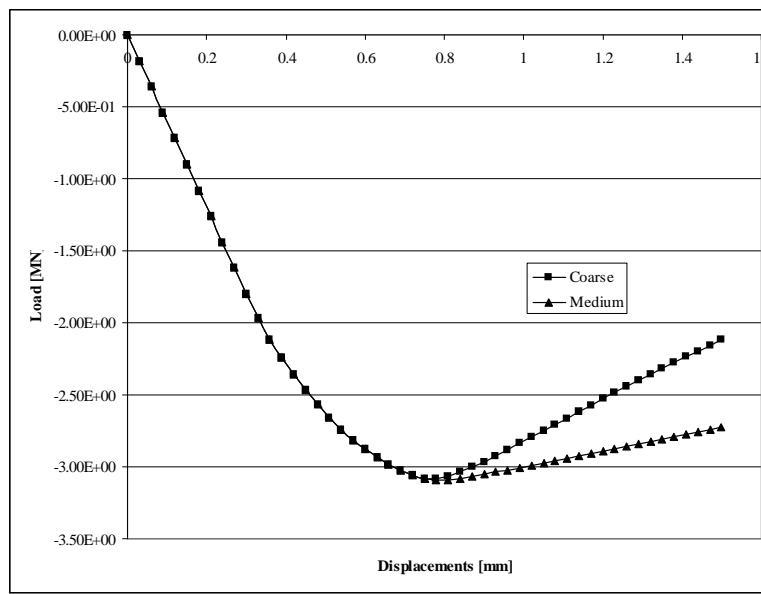


Figure 18.24: Curve carico-spostamento relative ad una prova di compressione ottenute per la mesh “Coarse” e “Medium”

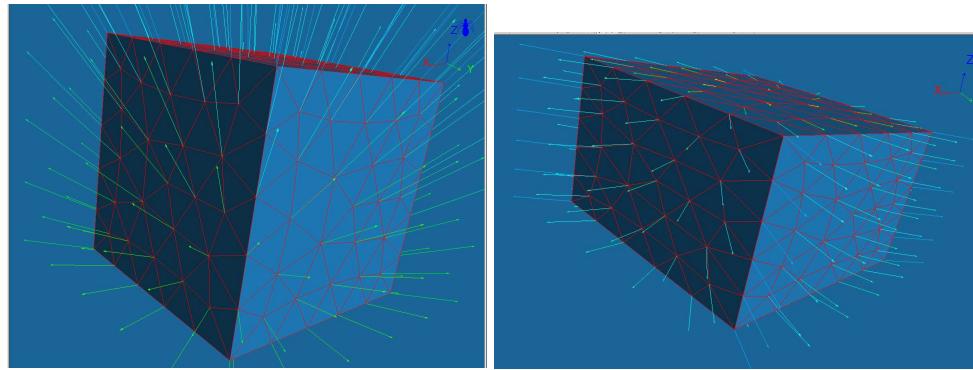


Figure 18.25: Mesh deformate al termine delle prove di espansione termica in assenza di vincoli di contenimento (immagine a sinistra) o in loro presenza (immagine a destra). Nelle precedenti immagini sono riportati i vettori spostamento relativi alle due mesh deformate.

18.6.5 Prova di carico ciclico

Le prova di carico ciclico (Figura 18.26 e 18.27) si svolge in controllo di spostamenti ed è composta da 4 fasi successive (Tabella 18.6):

1. applicazione in 50 incrementi di un allungamento del provino;
2. applicazione in 50 incrementi di uno spostamento opposto al precedente: al termine di questo secondo passo lo spostamento complessivo applicato al provino è nullo;
3. applicazione in 50 incrementi di un accorciamento del provino;
4. applicazione in 50 incrementi di uno spostamento opposto a quello applicato nel terzo passo: al termine di questa fase lo spostamento complessivo del provino è nuovamente nullo.

	Incrementi	Verso dello spostamento	Spostamento per incremento [mm]	Spostamento complessivo [mm]
Prima fase	50	Elongazione	6.00E-03	0.3
Seconda fase	50	Compressione	-6.00E-03	0
Terza fase	50	Compressione	-3.00E-02	-1,5
Quarta fase	50	Elongazione	3.00E-02	0

Table 18.6: Descrizione della prima prova di carico ciclico

Durante l'esecuzione della prova si percorrono:

1. il ramo elastico di carico ed il ramo di softening a trazione durante la prima fase della prova;
2. lo scarico rettilineo con un modulo elastico ridotto fino all'origine durante la seconda fase. La diminuzione del modulo elastico è dovuta al danneggiamento del materiale. Al termine dello scarico, con spostamento complessivo imposto nullo, la forza rilevata è anch'essa nulla (passaggio per l'origine della curva forza-spostamento): questo fatto implica l'assenza di uno spostamento irreversibile e la richiusura delle fessure al termine del processo di scarico;
3. il ramo di carico e di softening a compressione durante la terza fase della prova. Il ramo di carico a compressione è lineare solo nel primo tratto fino ad un limite imposto nelle caratteristiche del materiale (evidenziato da una linea rossa in Figura 18.26). La resistenza a compressione del provino è di poco superiore a quella uniaxiale dichiarata nelle caratteristiche del materiale per effetto delle condizioni di vincolo che generano localmente uno stato tensionale biassiale;
4. lo scarico quasi rettilineo del provino con un modulo elastico non deteriorato durante la quarta fase. L'ultimo tratto sub-orizzontale dello scarico a compressione non deve essere considerato come rappresentativo del comportamento del materiale.

18.6.6 Conclusioni

Dai risultati ottenuti con MERLIN, relativi alle prove di validazione sopra esposte, si nota la capacità del codice ad elementi finiti utilizzato di cogliere in modo soddisfacente la risposta del provino di calcestruzzo sottoposto a prova. Avendo validato il legame costitutivo di Cervenka all'interno del codice MERLIN si può procedere allo svolgimento delle analisi dell'elemento centrale della diga di Poglia utilizzando il suddetto modello costitutivo. I risultati delle analisi saranno esposti nel prossimo capitolo.

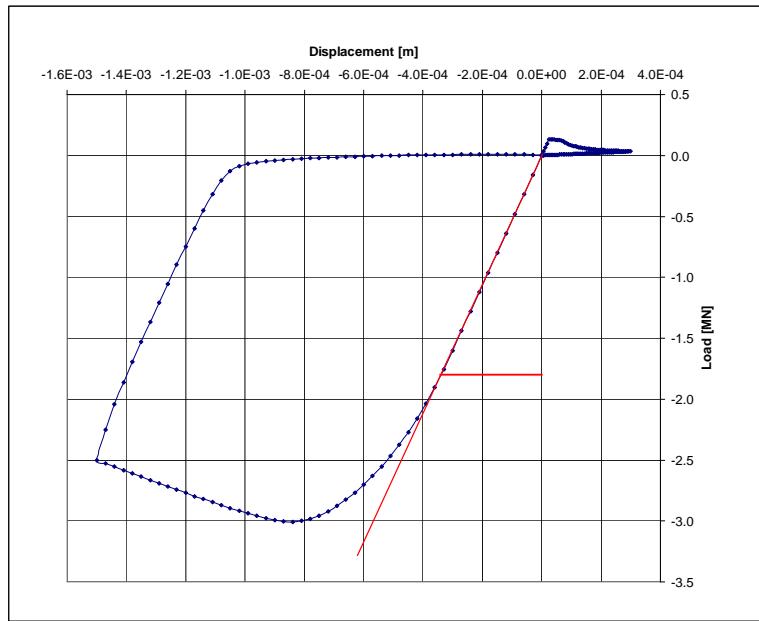


Figure 18.26: Curva forza spostamenti della prova di carico ciclico

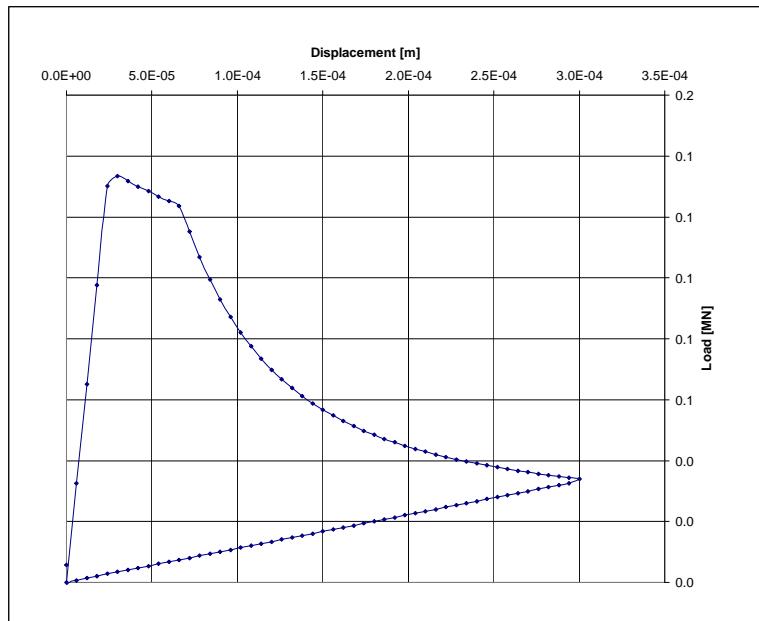


Figure 18.27: Particolare della curva presente in Figura 18.26 carico e scarico del provino a trazione

Chapter 19

NONLINEAR ROCK MODELS

19.1 Model

Many sites, particularly in Japan and Iran, have notoriously weak and fissured rock. These peculiarities must be accounted for in the context of an advanced nonlinear analysis.

The model adopted, (Kawamoto and Ishizuka 1981) is one which proved to be particularly suitable for Japanese rock, and extensively used.

$$\frac{E}{E_0} = a(R)^b \quad (19.1)$$

$$\nu = \nu_f - (\nu_f - \nu_0)A(R)^B \quad (19.2)$$

where E_0 is the initial tangent modulus and nonlinear parameters, ν_0 is the initial Poisson ratio, ν_f is the fracture Poisson ratio, a , b , A and B are nonlinear parameters. Upon failure $E \rightarrow E_f \simeq E_0/10 - 100$. R is the so-called fracture margin and is equal to $\min(d_1/D_1, d_2/D_2)$.

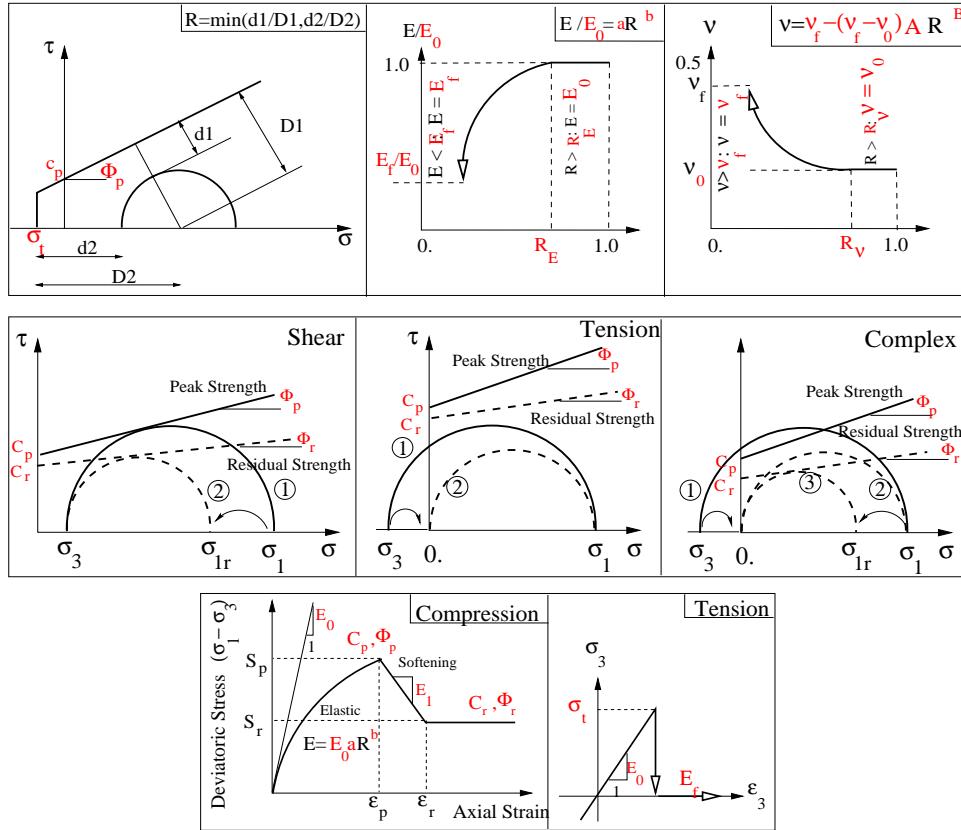


Figure 19.1: Kawamoto Model, all input parameters are shown in red

The stress-strain curve in compression exhibits first a linear softening, followed by a residual value; The tensile response is brittle (if pure tension), while no tension is allowed in biaxial state of stresses.

User input data:

- E_0, E_1, E_f Initial, softening (-ve) and failure modulus
- ν, ν_f Poisson ratio, initial and failure value
- σ_t Tensile strength
- a, b nonlinear parameters for E
- A, B nonlinear parameters for ν
- C_p, C_r Initial and residual cohesion
- Φ_p, Φ_r Initial and residual angle of friction
- R_E, R_ν Threshold values for respective values of R

19.2 Test Results

One element tests were conducted to assess the implementation of the model. Figures 19.2 to 19.6 illustrate the results for: compression, compression/unload, tension, shear, and shear cyclic tests. Results are consistent with the theory.

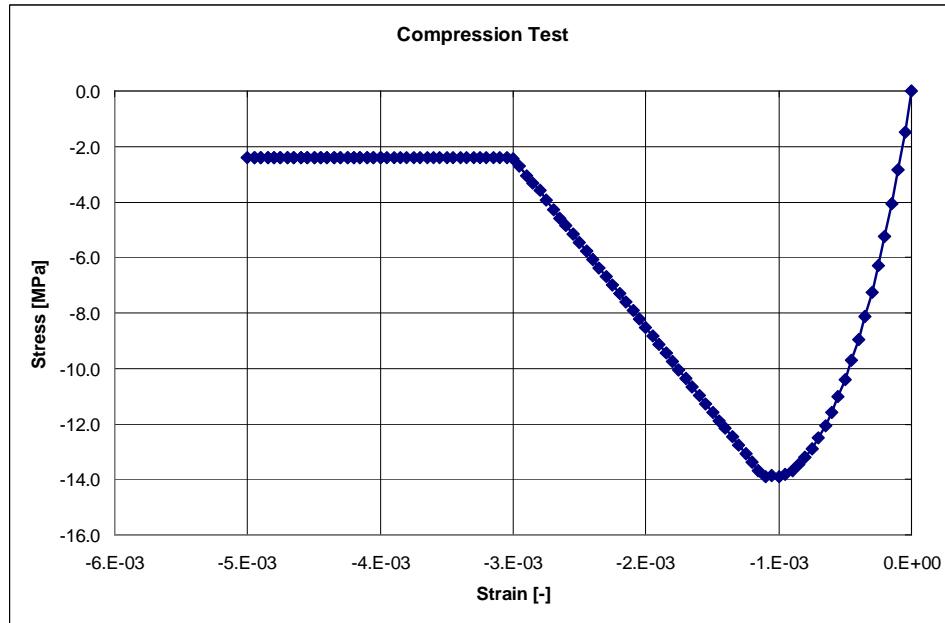


Figure 19.2: Kawamoto Model, Compression Test

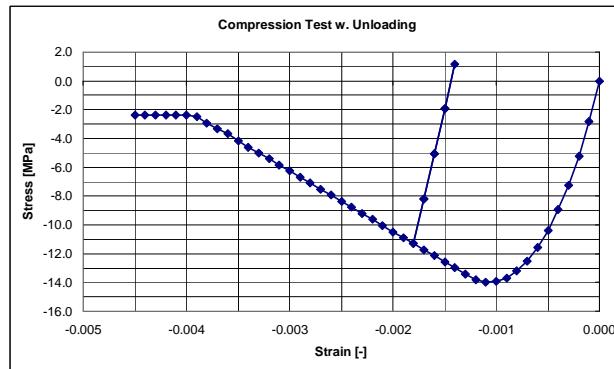


Figure 19.3: Kawamoto Model, Compression Test with Unloading

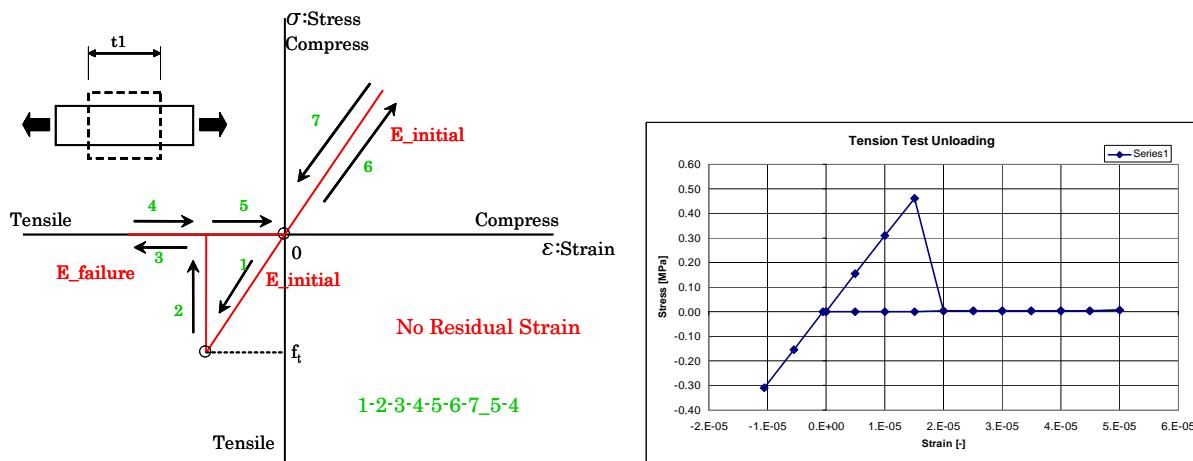


Figure 19.4: Kawamoto Model, Tension Test with Unloading

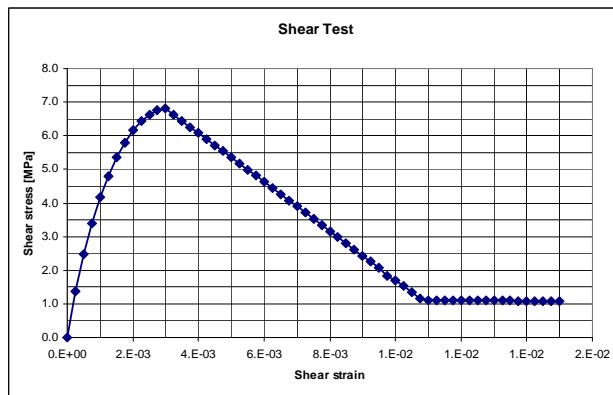


Figure 19.5: Kawamoto Model, Shear Test

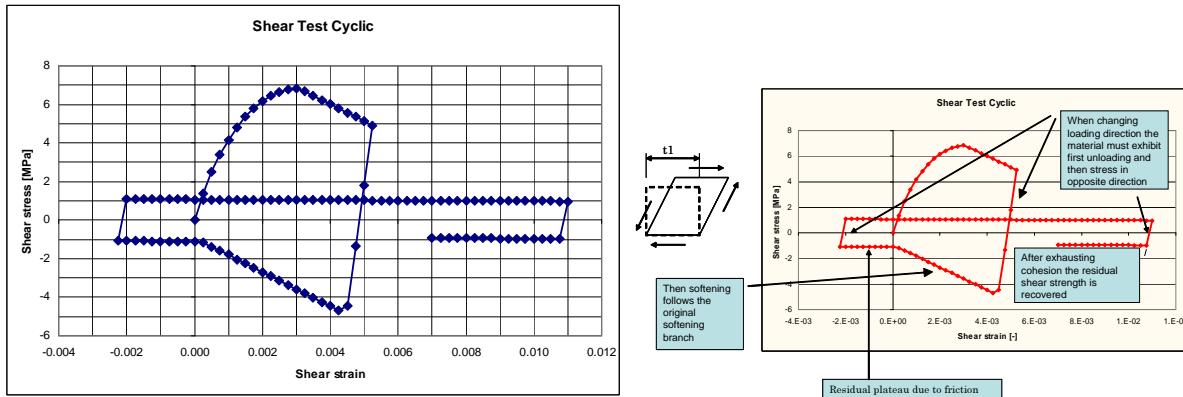


Figure 19.6: Kawamoto Model, Shear Test Cyclic

Part II

SYSTEM

Chapter 20

GETTING READY

This chapter will cover the preparation of the input data file for a fracture mechanics based analysis. More specifically, we shall list all parameters required, and provide the reader with guidelines for their selection.

20.1 Preliminary Considerations; Dam Analysis

This section provides some general guideline on which options of MERLIN the user should activate for different types of analysis.

20.1.1 LEFM

In the LEFM analysis, there must be an initial crack, and the rock/concrete interfaces are connected through master/slave nodes (**RIGID** option in preMERLIN). Uplift in the cracked ligament is handled by the **HYDROSTATIC** option, and along the uncracked ligament by an uplift only if the rock is permeable.

For impermeable rock, if master/slaves nodes are used and **HYDROSTATIC** pressures are applied on both faces at the interfaces, then those forces simply cancel out.

The analysis procedure is as follows

1. Prepare a preMERLIN .bd file with an initial crack which contains as a minimum the following options:
`Dimension, Smoothing, LEFM, S-integral, PrintCrack, GeomModel, MatProperties, IELAST, MeshProperties, IMESHSIZE, Coordinates, Faces, SFACE, Regions, SREGION, BCs, FACE, Loads, HYDROSTATIC, BODYFORCE, Connectors, RIGID, Cracks, ContourPathRadius, EndGeomModel, and EndInput.`
2. Run preMERLIN.
3. Run MERLIN.
4. From MERLIN's Output (or from Spider), inspect the stress intensity factor K_I , if greater than the selected fracture toughness, K_{Ic} , then simply increase the crack length by altering the coordinates of the crack tip node in the preMERLIN input data file (from step 1), and go to 2.
5. If $K_I \leq K_{Ic}$, then the analysis can be terminated, and the final one corresponds to the final crack length from an LEFM analysis.

20.1.2 Strength Based

Using preMERLIN a mesh with interface elements must first be generated. We note that there is no need to have an initial crack, as MERLIN will automatically open the interface crack and propagate it.

Hence, as a minimum the following options should be used in PreMERLIN:

`Dimension, Smoothing, NLFM, PrintCrack, PrintStress, GeomModel, MatProperties, IELAST, ICM, MeshProperties, IMESHSIZE, Coordinates, Faces, SFACE, Regions, SREGION, BCs, FACE, Loads, HYDROSTATIC, UPLIFT, BODYFORCE, Connectors, RIGID, EndGeomModel, and EndInput.`

Note, that in its current version, incremental loads can not be handled by preMERLIN, hence the following incremental load attributes will have to be added to the output file of preMERLIN:

`SecantNewton, LineSearch, RelResidErr, DispError, EnergyError, AbsResidErr`, in several increments with gradually increasing water elevation.

As to the material properties, the interface element should have

1. h : is for the third dimension.
2. ρ : should be zero.
3. α : should be zero.
4. $K_t: \approx G/t \approx E/t$ where E is the Young's modulus of an adjacent material, and t is the physical thickness of the interface. If t is unknown, use $\approx 10E$.

5. K_n : $\approx E/t$ where E is the Young's modulus of an adjacent material, and t is the physical thickness of the interface. If t is unknown, use $\approx 10E$.
6. σ_t : As deemed appropriate. If zero, use instead a very small value to avoid numerical errors (such as 0.1).
7. c : As deemed appropriate. If shear failure is not to be accounted for, use a large value such as 1,000 psi.
8. ϕ_f Use a large value, $\approx 70^0$.
9. ϕ_D : As deemed appropriate. If shear failure is to be neglected, use 0.
10. G_F^I : Should be zero.
11. G_F^{II} : Should be zero.
12. γ : Should be zero.
13. u_{Dmax} : As deemed appropriate. If shear failure is to be neglected, use a large value, ≈ 10 . in.
14. s_1 : Should be zero.
15. w_{s_1} : Should be zero.
16. c_1 : Should be zero.
17. CSD_{cw_1} : Should be zero.

Note that the crack will automatically propagate, and uplift automatically adjusted until equilibrium is reached. To each crack (element) increment, will correspond a load increment (in the fictitious crack model).

20.1.3 NLFM

MERLIN supports two type of nonlinear fracture mechanics analysis:

20.1.3.1 Incremental NLFM

In this mode of analysis, the load is incrementally specified. Increment 0 typically corresponds to the gravity load, and zero water elevation. Subsequent load increment will then correspond to head and or tail water elevation as well as uplift pressures.

The preMERLIN file should first be prepared and would typicaaly include:

`Dimension, Smoothing, NLFM, PrintCrack, PrintStress, GeomModel, MatProperties, IELAST, ICM, MeshProperties, IMESHSIZE, Coordinates, Faces, SFACE, Regions, SREGION, BCs, FACE, Loads, HYDROSTATIC, Uplift, BODYFORCE, Connectors, INTERFACE, EndGeomModel, and EndInput. LoadDspCurve` may be used to tabulate crest displacements.

Noting that the current version of PreMERLIN generates only one single increment, its output should be manually edited to add additional load increments. Each block will contain the following:

`Iterations, EnergyError, RelResidErr, AbsResidErr, DispError, TangentStiff, DispBCs, Hydrostatic, Uplift, EndIncrement`

It should be noted that in the Hydrostatic option associated with a loaded element, the user defines the water elevation. If the water elevation is below the element, than the element is not loaded.

It is suggested that the dam be impounded through at least 6 increments. If the IFF (Imminent Failure Flood) is sought, then the load increments should be reduced as the anticipated IFF is approached. A failure to converge in an increment is a strong indication of instability/failure.

As to the material properties, the interface element should have

1. h : is for the third dimension.
2. ρ : should be zero.
3. α : should be zero.
4. K_t :
5. K_n :
6. σ_t : As deemed appropriate.

7. c : As deemed appropriate.
8. ϕ_f As deemed appropriate.
9. ϕ_D : As deemed appropriate.
10. G_F^I : As deemed appropriate.
11. G_F^{II} : Usually 10 times G_F^I .
12. γ : Usually 0.3
13. u_{Dmax} : As deemed appropriate
14. s_1 : Usually $\frac{1}{4}\sigma_t$.
15. w_{s_1} : Usually $\frac{0.75G_F^I}{\sigma_t}$.
16. c_1 : Usually $\frac{1}{4}c$.
17. CSD_{cw_1} : Usually $\frac{0.75G_F^{II}}{c}$.

20.1.3.2 Failure/Post-Peak

This feature of the program should be exercised only by very experienced users.

Where as in the incremental approach the user specifies increment of water elevation, in this mode of analysis the user would first define a couple of incremental loads corresponding to gravity and water elevation, and then will specify a crest displacement of crack mouth opening displacement. This feature will trigger internal algorithms which will adjust correspondingly the water elevation, and enable MERLIN to automatically seek the IFF. As the failure flood is reached, MERLIN will then decrease the water elevation to prevent failure yet accommodating the increased crest displacement.

As with the NLFM analysis, this procedure is fully automated. First a preMERLIN file should be prepared. This file should contain Dimension, Smoothing, NLFM, PrintCrack, PrintStress, GeomModel, MatProperties, IELAST, ICM, MeshProperties, IMESHSIZE, Coordinates, Faces, SFACE, Regions, SREGION, BCs, FACE, Loads, HYDROSTATIC, Uplift, BODYFORCE, Connectors, INTERFACE, EndGeomModel, and EndInput.

1. Increment 0 (self-weight):
`Iterations, EnergyError, RelResidErr, AbsResidErr, DispError, TangentStiff, DispBCs, BodyForces, EndIncrement`

2. Increment 1 (arc length):

`Iterations, EnergyError, RelResidErr, AbsResidErr, DispError, Arc-Length, TangentStiff, DispBCs, Hydrostatic, Uplift,`

Note that in this increment we specify a unit height of water elevation.

3. Increment 2-i (arc length):

`Iterations, EnergyError, RelResidErr, AbsResidErr, DispError, Arc-Length, TangentStiff, Uplift, EndIncrement`

In all the increments, we use the same value for `Arc-length`, and i is the increment number for which the crest displacement is positive (i.e. downstream). Once this has been reached, then we can have

4. Increment i+1 where we specify the COD

`Iterations, EnergyError, RelResidErr, AbsResidErr, DispError, TangentStiff, SpecifyCOD, Uplift, EndIncrement`

Note that from now on we are specifying the crack opening displacement (which should be guessed) or the crest displacement, and MERLIN will automatically determine the water elevation which would have to be applied in order to cause such an imposed displacement. This is in actuality a multiplier of the first `Arc-length` increment.

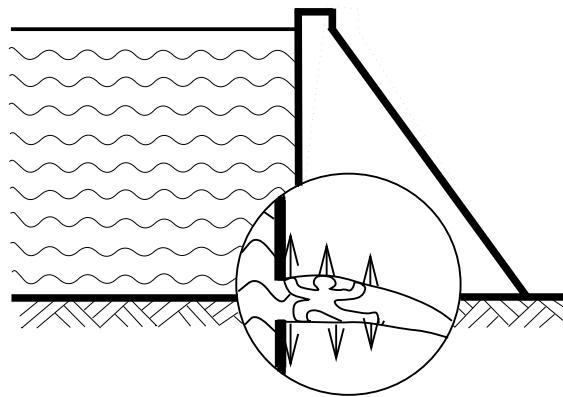


Figure 20.1: Uplift Pressures in a Dam

20.1.4 Uplift Pressures

Whereas the uplift modeling within the context of a gravity concrete dams, Fig 20.1, remains the subject of much discussion, ??, we identify two possible major models:

Permeable Rock: In which case the rock is fully saturated, and thus the uplift forces are to be applied only upward on the dam base, Fig. 20.2.

Impermeable Rock: where seepage along the rock/concrete interface takes place, and the uplift forces is applied both upward on the dam, and downward on the rock, Fig. 20.2.

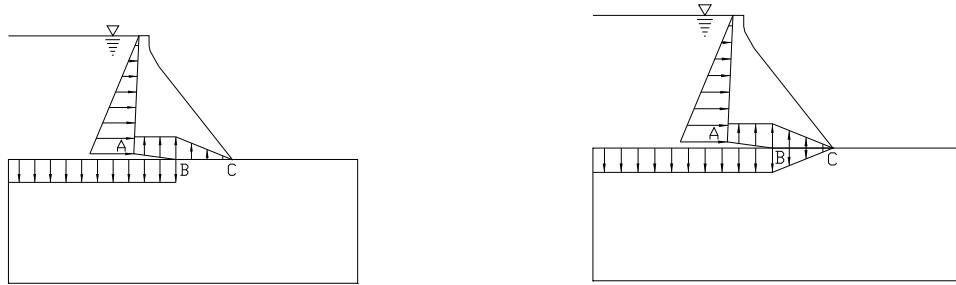


Figure 20.2: Uplift Pressures for Permeable and Impermeable Rock

Hence, depending on the rock permeability and the selected models, there can be six different combinations of uplifts and crack models, Table 20.1.

Model	Master File	Joint	Autom. Crack Prop.	Init.		Ligament	
				Crack	Cracked	Cracked	Uncracked
Permeable Rock							
LEFM	lp.bd	Master/Slave	No	Yes	Hydro.	Hydro.	
Strength	sp.bd	Interface	Yes	No	Uplift		
NLFM	np.bd	Interface	Yes	No	Uplift		
Impermeable Rock							
LEFM	li.bd	Master/Slave	No	Yes	Hydro.	Pore Pressures	
Strength	si.bd	Interface	Yes	No	Uplift		
NLFM	ni.bd	Interface	Yes	No	Uplift		

Table 20.1: Fixed Water Elevation Fracture and Uplift Models

20.1.5 Dynamic Analysis

This feature of MERLIN is currently being revised (streamline input data file, and provide additional features), and should not be exercised until further revisions of the code.

20.2 Material Properties

Prior to the analysis, material properties for the concrete structure, rock foundations, and rock/concrete interface must be determined, and Table 20.2 summarizes the ones which must be determined.

	Concrete	Rock	Interface
Basic Properties	$E, \nu, \alpha, \gamma, f'_t$	$E, \nu, \alpha, \gamma, f'_t$	c, ϕ, K_n, K_t
LEFM	K_{Ic}	K_{Ic}	K_{Ic}
NLFM/FCM	$f't, G_F$	f'_t, G_F	f'_t, G_F
NLFM/ICM			

Table 20.2: Required Material Parameters

20.2.1 Concrete

20.2.1.1 Basic Properties

- E : The elastic modulus of concrete can be either directly evaluated from laboratory tests, or simply derived from ACI-318 equations:

$$E = 57,000\sqrt{f'_c} \quad (20.1)$$

where both E and f'_c (the uniaxial unconfined compressive strength) are expressed in psi.

In general, results will not be too much affected by small variations of E .

- ν : The Poisson's ratio for concrete is commonly taken to be in the range of 0.15 to 0.20.
- α : The coefficient of thermal expansion will be needed only if thermal stresses are present, (Army Corps of Engineers 1990). Those thermal stresses are caused either by an initial stress due to heat of hydration or due to thermal loading (one face of the dam being at a different temperature than the other). For concrete $\alpha = 5.5 \times 10^{-6}$ in/in per deg F is generally accepted for calculating stresses and deformations caused by temperature changes.
- γ : The density of concrete is commonly taken as 150 lb/cu-ft.
- f'_t : The concrete direct tensile strength can be either determined from laboratory tests, or simply estimated at 7% of f'_c , (Mindess and Young 1981). f'_t will be used to determine whether crack nucleation takes place. . Finally, f'_t can also be estimated to be $f'_r/1.8$ where f'_r is the modulus of rupture determined from a flexural test.

20.2.1.2 Linear Elastic Fracture Properties

The simplest form of fracture mechanics analysis which can be performed is a linear elastic one. In this model, the stress singularity (infinite theoretical stress at the crack tip) is recognized, and the criteria for crack propagation is one based on the strengths of the singularity denoted as stress intensity factors.

In this context, the only linear elastic fracture property required is the fracture toughness K_{Ic} . As a first approximation, it is recommended that K_{Ic} be taken equal to zero. Should the results be unacceptable, then a value of $K_{Ic} = 1.0 \text{ ksi}\sqrt{\text{in}}$ (Saouma, Broz, Brühwiler and Boggs 1991) could be used. Note that for subangular aggregates, this value could be increased up to 1.3.

Finally, should this value again result in unacceptable crack lengths, then laboratory experiments could be performed on recovered core specimens (Brühwiler, E. 1988), or in-situ tests (Saouma, Broz and Boggs 1991) could be conducted to determine the fracture properties of the dam concrete in question.

20.2.1.3 Nonlinear Fracture Properties

A more refined fracture mechanics model over the linear elastic one is the nonlinear one based on the presence of a fictitious crack. In this model it is assumed that the “true” crack is preceded by a so called fracture process zone (or fictitious crack) along which stresses can be transmitted.

The nonlinear fracture properties are:

- G_F : or fracture energy. For gravity dams, a value of 1.35×10^{-3} kip/in. is recommended, (Saouma, Broz, Brühwiler and Boggs 1991). Note that for arch dams, this value could probably be increased on the basis of laboratory tests. Also, laboratory tests could be performed on recovered cores to obtain a better indication of G_F , (Brühwiler, E. 1988).
- f'_t : or tensile strength. Within the context of a nonlinear analysis, this value can not be taken as zero, otherwise there will be no fracture process zone. Unless it is experimentally determined, f'_t should be taken as 7% of f'_c , (Mindess and Young 1981), or $f'_r/1.8$ where f'_r is the modulus of rupture.
- Shape of the softening diagram ($\sigma - COD$), and in general a bi-linear model for the strain softening should be used. With reference to Fig. 20.3, This simple model can be uniquely defined in terms of the tensile strength

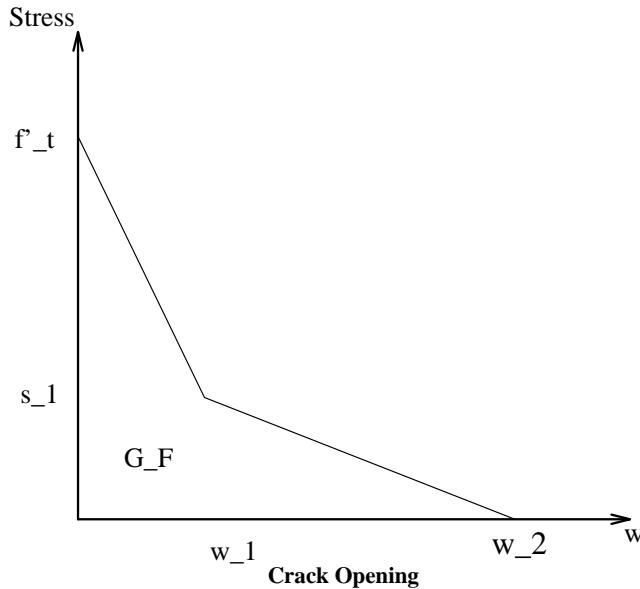


Figure 20.3: Concrete Strain Softening Models

f'_t , and the fracture energy G_F . In (Brühwiler and Wittmann 1990), it was found that the optimal points for concrete with 1" maximum size aggregate are:

$$s_1 = 0.4f'_t \quad (20.2)$$

$$w_1 = 0.8 \frac{G_F}{f'_t} \quad (20.3)$$

$$w_2 = 3 \frac{G_F}{f'_t} \quad (20.4)$$

whereas for structural concrete, (Wittmann et al. 1988), the corresponding values are:

$$s_1 = \frac{f'_t}{4} \quad (20.5)$$

$$w_1 = 0.75 \frac{G_F}{f'_t} \quad (20.6)$$

$$w_2 = 5 \frac{G_F}{f'_t} \quad (20.7)$$

- K_n should be 10 times E
- K_t should be 10 times E
- Φ_F and Φ_D , unless measured, should be taken as 40° and 20° respectively.
- G_{IIF} should be 10 times G_{IF}
- γ , unless measured, should be 0.3 for concrete
- u_{Dmax} , unless measured, should be 0.01 m for concrete

20.2.1.4 Dam Concrete

For dam concrete in metric units: $\rho = 2,400 \text{ Kg/m}^3$, $E = 36 \times 10^9 \text{ Pa}$, $\alpha = 1 \times 10^{-5} \text{ m/m}^\circ\text{C}$, film coefficient for heat transfer by convection: $h_{air} = 34 \text{ W/m}^2 \text{ }^\circ\text{C}$, $h_{water} = 100 \text{ W/m}^2 \text{ }^\circ\text{C}$; Specific heat is about 1,000 J/Kg.K, the thermal conductivity k is 2.7 J/sec.m.K.

Note that Whittman's reports $f'_t = 3.75 \times 10^6 \text{ Pa}$, and $G_{IF} = 400 \text{ N/m}$.

20.2.2 Rock

20.2.2.1 Basic Properties

Basic elastic properties of rocks vary greatly depending on the rock type. Whereas many of those are found in Table 6.1 of (Goodman 1980), field test may be necessary.

It should be noted that those values are strongly affected by the degree of fracturing.

Finally, the engineer should be cautioned about the potentially orthotropic nature of the rock. This orthotropy can be either:

1. "Micro-scopic" due to the intrinsic rock type
2. "Macro-scopic" due to the presence of numerous faults and joints separating otherwise isotropic (or orthotropic) rock masses. When the distance separating those faults is too small compared to the dam base, then it may be easier to model the rock foundation as orthotropic.

20.2.2.2 Linear Elastic Fracture Properties

Two cases should be distinguished:

Fracture of intact rock: In which case a value of zero for the fracture toughness is still recommended for preliminary analysis. However should this value yield unacceptably large cracks, then actual fracture toughness values could be used. The best reference to obtain K_{Ic} for rock is through the work of Ouchterlony in Sweden, (Ouchterloni, Takahashi, Matsuki and Hashida 1991). Should tests be necessary, then either the Wedge Splitting test of Brühwiler and Saouma (Brühwiler, E. and Saouma, V.E. 1990) or the ISRM (International Society of Rock Mechanics) method, (Ouchterlony 1988) can be used.

Fracture along a joint: In this case the fracture toughness should always be taken as zero.

Given the alternative, a "dipping" crack inside the rock is by far preferable to a horizontal crack within the concrete or along the concrete/rock interface. Should the crack dip inside the rock then a different type of analysis would have to be conducted.

Unfortunately, rock foundations are seldom well characterized, and may include numerous joints/faults which make them far from homogeneous. However should they be assumed to be homogeneous (for analysis purpose), then the fracture toughness along with the elastic properties should be known.

Finally, in assessing the final crack length, it should be recognized that not only is it dependent on the fracture toughness but also on the presence of joints/faults, and the presence of *in-situ* stresses which are usually unaccounted for in analysis and which may close the numerically simulated crack.

20.2.2.3 Nonlinear Fracture Properties

Within the context of a stability investigation of a concrete dam, a nonlinear fracture model for rock can not be justified.

Furthermore, and with probable exception of the work of Labuz (Labuz, Shah and Dowding 1985), there has been very limited data on nonlinear fracture properties of rock.

20.2.3 Interface

20.2.3.1 Basic Properties

Joint elements can be used to model the rock/concrete interface and account for its finite stiffness and strength. It should be mentioned that in such an analysis, the criteria for crack opening is based on Mohr-Coulomb, and hence it would preclude fracture mechanics based ones.

Elastic Properties: such as the normal and tangential stiffnesses, K_n and K_t can only be determined through recovered cores. As a guideline for either those elastic properties, or for the recommended testing procedure for interface properties, the reader should consult a recent report published by the EPRI, (Corporation 1992).

Strength Properties: should also be obtained from tests on recovered cores. However, as an indication, the range of values in (Corporation 1992) are:

- Friction angle Φ : 53 to 63 degrees
- Cohesion c : 15-250 psi

20.2.3.2 Linear Elastic Fracture Properties

In a linear elastic fracture mechanics based analysis of crack propagation along the interface, joint elements should not be used. Instead, it is assumed that there is a perfect bond between rock and concrete and the criteria for crack propagation is based on the fracture toughness.

Fracture toughness values along the interface are substantially lower than those found in intact material. Whereas to the best of our knowledge there is no experimental data, limited tests on concrete/concrete interface, (Saouma, Broz, Brühwiler and Boggs 1991) have shown that at least a 50% reduction is expected. As such, a zero value of fracture toughness is recommended for the interface cracks.

20.2.3.3 Nonlinear Fracture Properties

In a nonlinear elastic fracture mechanics based analysis of crack propagation along the interface, joint elements should not be used. Instead, it is assumed that there is a perfect bond between rock and concrete and the criteria for crack propagation is based on f'_t and G_F .

As for LEFM properties, there is not yet any experimental data to allow the quantification of either G_F or the softening curve at the rock/concrete interface.

20.3 Load

20.3.1 Gravity

Concrete: Gravity load should always be considered for the concrete. The applied forces due to gravity loads are specified using the **BodyForces** option in MERLIN. A typical gravity load specification would appear as follows:

```
BodyForces
1.0 0.0 -1.0
```

The first number in the body force specification is the magnitude of the gravitational acceleration, which in this case is 1.0. This indicates that the unit weight has been specified in the material properties rather than the mass density. If the mass density is specified in the material properties, the magnitude of the acceleration should be the actual value of the gravitational acceleration in the appropriate system of units. The next two numbers specify the direction of the acceleration. In this case the acceleration is in the negative y-direction. For 3-D analyses a third component is required for the direction of acceleration.

Rock: In most cases, gravity loads should not be accounted for in the rock, as all deformation caused by them would have taken place prior to construction. This is accomplished by assigning the unit weight/mass density to be zero in the material properties for the rock. The gravity of the rock must be considered when the crack is propagated into the foundation.

20.3.2 Thermal

Thermal load should be considered when:

1. Initial stresses are caused by the heat of hydration during curing of the concrete. For roller compacted concrete structures it is imperative that thermal loading be considered. For other structures such a loading might give an indication of the secondary stresses which may have caused some (limited) initial stresses.
2. The difference in temperature between the downstream face (typically exposed to the sun), and the cooler upstream face (typically under water) results in significant initial stresses.

The applied loads due to thermal effect are specified using the **Temperatures** option in MERLIN. A typical thermal load specification would appear as follows:

```
Temperatures
 9
101 50.0
102 52.5
103 55.0
104 57.5
105 60.0
106 62.5
107 65.0
108 67.5
109 70.0
```

In this case temperatures are specified at nine different nodes in the mesh; the temperatures for all other nodes are automatically assigned a value of zero. It should be noted that it is not the absolute value of the temperature which controls the thermal stresses, but rather the relative difference among them. Specification of nodal temperatures requires two pieces of information: a node number and the value of the temperature at that node. The nodal temperatures should be obtained through either a separate steady-state or transient heat conduction analysis. MERLIN converts the specified nodal temperatures to thermal strains and finally to thermal stresses.

Heat of hydration for concrete can be estimated from Table 20.3 for two different cement contents (180 and 280 Kg/m³).

Age [days]	H_c [J/g]	H_c [J/g.day]	$H_b = H_c m_z / \rho_b$	
			$[J/Kg.day]$	
			m_z	
			180	280
0	0			
1.5		85.000	6,375	9,917
3	255			
5		20.000	1,500	2,333
7	335			
17.5		3.09524	232.14	361.11
28	400			
59		0.48387	36.29	56.45
90	430			
227.5		0.10909	8.18	12.73
365	460			
1,368.5		0.01495	1.12	1.74
2,372	490			

Table 20.3: Heat of Hydration for Concrete

20.3.3 Water and Silt Pressures

Water and silt pressures should be accounted for on both the upstream and downstream faces. When a discrete crack is present and the crack mouth is exposed to water, the water pressure on the crack surfaces should also be

considered. The applied forces due to water and silt pressures are specified using both the **Tractions** and **UTRACT** options in MERLIN. A typical pressure loading specification would appear as follows:

```
UTRACT
Tractions
 5
101  4 100.0 62.5
102  4 100.0 62.5
103  4 100.0 62.5
104  4 100.0 62.5
105  4 100.0 62.5
```

The **Tractions** option allows for the specification of both normal and tangential surface tractions on element surfaces. A normal surface traction is what most engineers would call a pressure and a tangential surface traction would simply be called a traction. To specify the element surface on which pressures and/or tractions will be applied an element number and an element surface number are required; the convention for the numbering of element surfaces is given in Section 2.8 of the MERLIN User's Manual (Saouma et al. 2008). In addition to the element surface specification, magnitudes for the pressure and the traction are required. By default MERLIN assumes constant values of pressure or traction on the specified element surface, but this limitation can be circumvented by using the **UTRACT** option. The presence of the a **UTRACT** option indicates that the pressures and/or tractions will be defined by the user in user subroutine **utract** (see Section 7.2 of the MERLIN User's Manual (Saouma et al. 2008)). When the magnitudes of the pressures and/or tractions are defined by user subroutine **utract**, the values for the pressures and tractions entered in the input file under the **Tractions** option are passed to **utract** as arguments. For the default version of **utract** the value that normally specifies the magnitude of the pressure specifies the elevation of the reservoir or tailwater and the value that normally specifies the magnitude of the traction specifies the unit weight of water. These two quantities must be defined in units consistent with the rest of the input file. For more elaborate loadings the user must reprogram **utract**, compile the source code, and link the object (i.e., the compiled source code) with the MERLIN libraries. Tools are provided with MERLIN to simplify this task for the user.

20.3.4 Uplift Pressure

In modeling the uplift pressure, one must distinguish between the actual crack and the uncracked ligament. In most cases, different techniques are used to model uplift pressures acting on the crack surfaces and those acting along the uncracked ligament. The necessity for these different techniques is due to the assumptions regarding whether or not a given material (i.e., either rock or concrete) is pervious or impervious. For this discussion, the modeling of uplift pressures in the crack surfaces and along the uncracked ligament will be treated separately.

20.3.4.1 Cracked Zone

Along the crack, full uplift pressure should be applied on both sides of the crack as a normal surface traction regardless of the assumptions as to whether the rock or concrete is pervious or impervious. This is necessary because the natural (i.e., stress) boundary conditions are defined in terms of the total stresses σ and, with the effective stresses σ' being identically zero, the presence of a non-zero seepage pressure p at the surface of the material requires an applied surface traction $\hat{\mathbf{t}}$ for the natural boundary conditions to be satisfied

$$(\sigma' - p\mathbf{I})\mathbf{n} - \hat{\mathbf{t}} = 0 \Rightarrow \hat{\mathbf{t}} = -p\mathbf{n} \quad (20.8)$$

Fig. 20.3.4.1 illustrates how the consistent nodal forces for an element on the foundation surface subjected to internal (seepage) pressure combine to create a traction free condition. Should full uplift pressure yield an unacceptably long crack, and should the crack be completely within the concrete rather than along the concrete/rock interface, then a reduction of the uplift pressure may occur as the crack openings become very small (Brühwiler, E. and Saouma, V.E. 1991). For very stiff structures, this may provide a substantial reduction in the total uplift force.

The applied forces due to full uplift pressures on the crack surfaces are specified using the **Tractions** and **UTRACT** options in MERLIN. These were discussed in some detail in Section 20.3.3, so no additional discussion on there usage is required here. Specification of the uplift pressures on the crack surfaces must be made within the same invocation of the **Tractions** option as the water and silt pressures; invocations of load options are not cumulative within a given increment. For both the upper and lower surfaces of the crack the specified normal surface traction is compressive (i.e., the resulting nodal forces act upward on the concrete and downward on the rock).

The applied forces due to an uplift pressure that is a function of the crack opening are specified using the **p_W0-COD_W0** and **Uplift** options in MERLIN. The **p_W0-COD_W0** option must be included in the program control block of the input file to define the relationship between the full uplift pressure p_w0 and the crack opening displacement

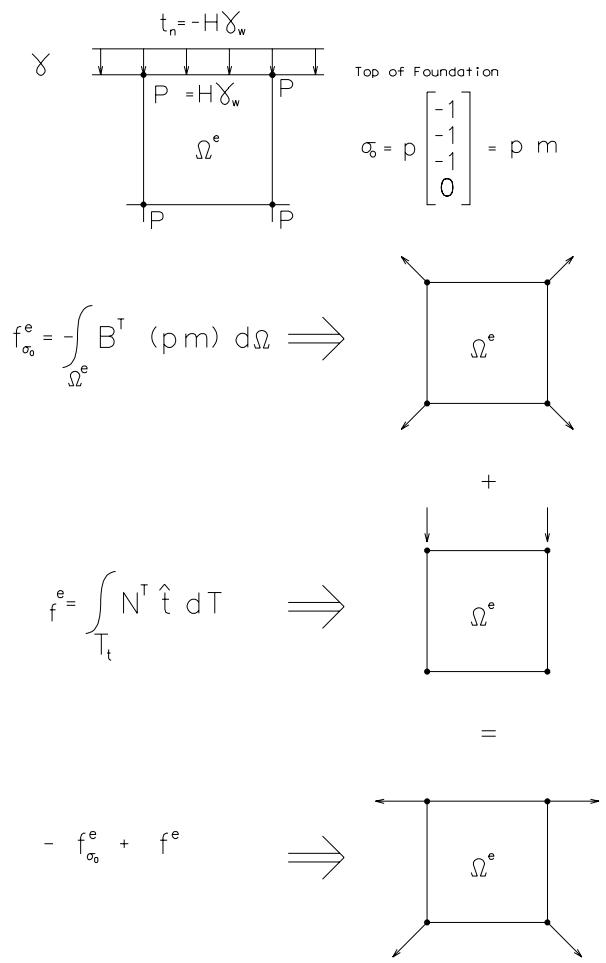


Figure 20.4: Forces Acting on an Element at the Foundation Surface Subjected to Internal Pressure and Normal Surface Tractions

COD_{W0} at above which p_{W0} acts on the crack surfaces. The input for the **p-W0-COD-W0** option would appear as follows:

```
p_W0-COD_W0
 3
 14.30  0.00395
 42.47  0.00206
127.83  0.00064
```

Obviously, the relationship between p_{W0} and COD_{W0} is idealized as piecewise linear. The combinations of p_{W0} and COD_{W0} shown above correspond to those determined experimentally by Brühwiler and Saouma (Brühwiler, E. and Saouma, V.E. 1991). The **Uplift** option is used to define the full uplift pressure p_{W0} acting at the mouth of each crack. The input for the **Uplift** option is as follows:

```
Uplift
 1
 1    2400.0    0.0361
```

The full uplift pressure p_{W0} is defined by head above the crack mouth H_W and the unit weight of the water γ_W . In the example shown above H_W is 2400.0 inches and γ_W pci. The value of p_{W0} defined in the **Uplift** option must be between the end points of the relationship defined in the **p-W0-COD-W0** option or *MERLIN* cannot determine the appropriate value of COD_{W0} .

20.3.4.2 Uncracked Zone

Along the uncracked ligament, uplift should also be modeled in some fashion. The magnitude and distribution of the uplift pressures should be governed by the following considerations:

1. Unless field data is available, and for rock with isotropic hydraulic conductivity, the uplift pressure is assumed to vary linearly from the upstream to the downstream value when the interface between the dam and foundation is pervious.
2. For rock with isotropic hydraulic conductivity, and with field data measurements, the uplift pressure can be assumed to vary linearly between points where the values of the uplift pressure are known.
3. For rock with a known orthotropic hydraulic conductivity, a steady state seepage analysis should be performed to determine the uplift pressure distribution.
4. For rock with isotropic hydraulic conductivity, a steady state seepage analysis is required within the context of the Case 3 (to be outlined below).

Three different approaches for modeling uplift pressures along the uncracked ligament are described here. They are based on assumptions as to whether the rock, concrete, and uncracked interface are pervious or impervious. The combinations for the three cases discussed here are summarized in Table 20.3.4.2.

Case	Rock	Concrete	Uncracked Interface
1	Impervious	Impervious	Impervious
2	Impervious	Impervious	Pervious
3	Pervious	Impervious	Pervious

Table 20.4: Summary of three cases for uplift on uncracked ligament

Case 1: In this model, shown in Fig. 20.3.4.2, the rock, concrete, and uncracked interface are all assumed to be impervious. This case corresponds to a “no flow” situation and, consequently, no uplift pressures are applied along the uncracked interface. The only uplift pressures present in this case are those acting on the crack surfaces.

Case 2: In this model, shown in Fig. 20.3.4.2, the rock and concrete are assumed to be impervious, but the uncracked interface is assumed to be impervious. The assumed flow regime for this case is represented in Fig. 20.3.4.2. Uplift pressures are applied upward on the dam and downward on the foundation along the uncracked interface as normal surface tractions using the **Tractions** and **UTRACT** options in *MERLIN*. This approach requires that the uplift pressures be prescribed as initial stresses using the **Pressures** option in *MERLIN*. Either

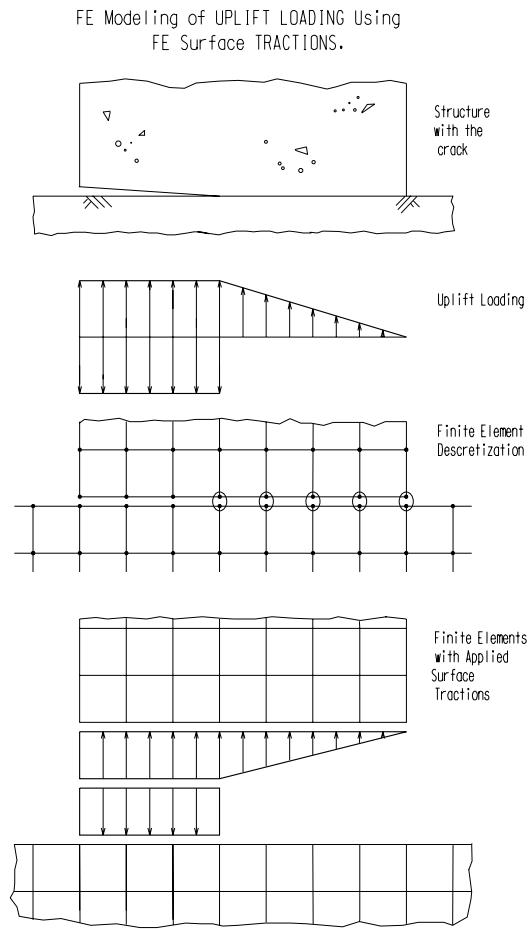


Figure 20.5: Uplift Model with Impervious Rock, Concrete, and Uncracked Interface

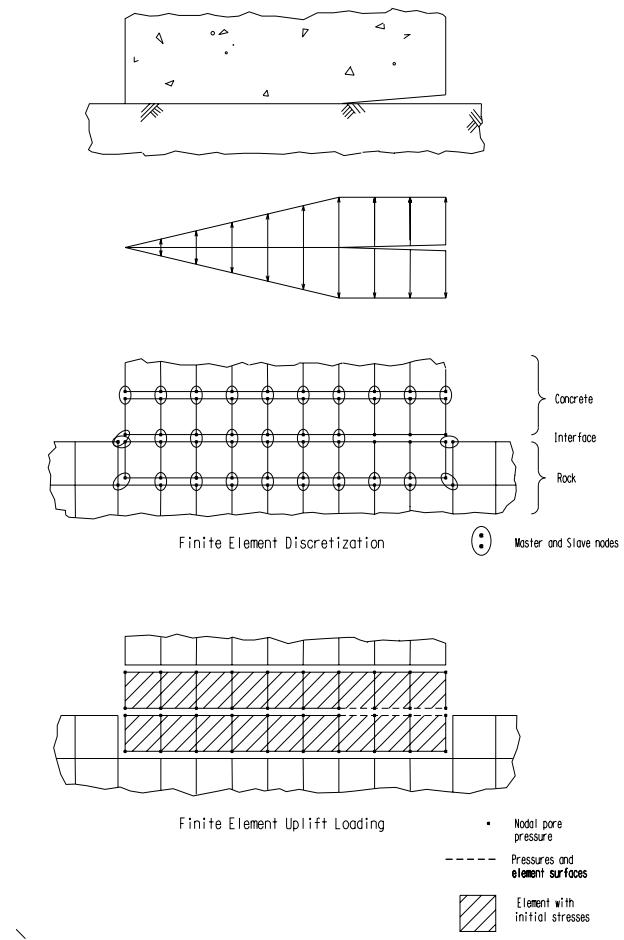


Figure 20.6: Uplift Model with Impervious Rock and Concrete and Pervious Uncracked Interface

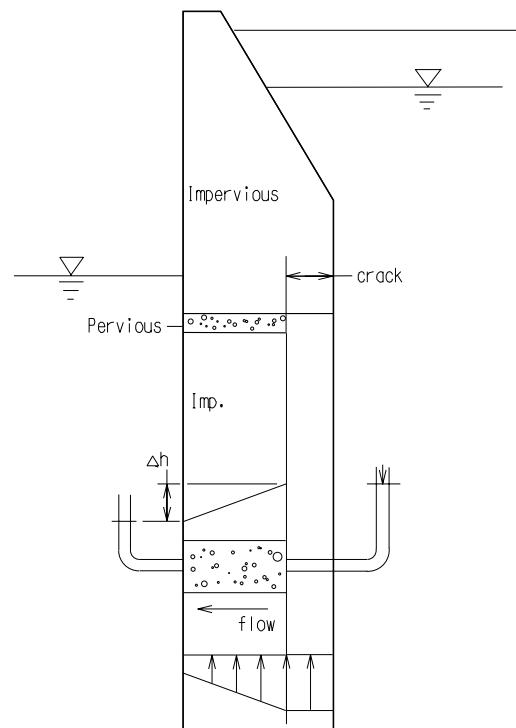


Figure 20.7: Pipe Analogy for Flow Along a Pervious Uncracked Interface

the uncracked interface must be modeled by interface elements or the continuum elements adjacent to the dam/foundation interface must be separated from neighboring elements using duplicate nodes. Fig. 20.3.4.2 represents the latter approach. Using duplicate nodes along the uncracked interface and applying normal surface tractions on the appropriate element surfaces is no acceptable because the equivalent nodal forces corresponding to the applied uplift pressures will cancel on the uncracked interface. Isolating the “pressurized” elements from the neighbors effectively confines the pressure to those elements, otherwise the neighboring elements would also be subjected to a seepage pressure. The mesh construction techniques required to isolate the “pressurized” elements and the **Pressures** option will be discussed below, as parts of this discussion will also apply to the third method of modeling uplift pressures on the uncracked ligament.

Case 3: In this model, shown in Fig. 20.8, the rock and uncracked interface are assumed to be pervious, but the concrete is assumed to be impervious. The uplift pressure acting on the dam is simply modeled by specifying

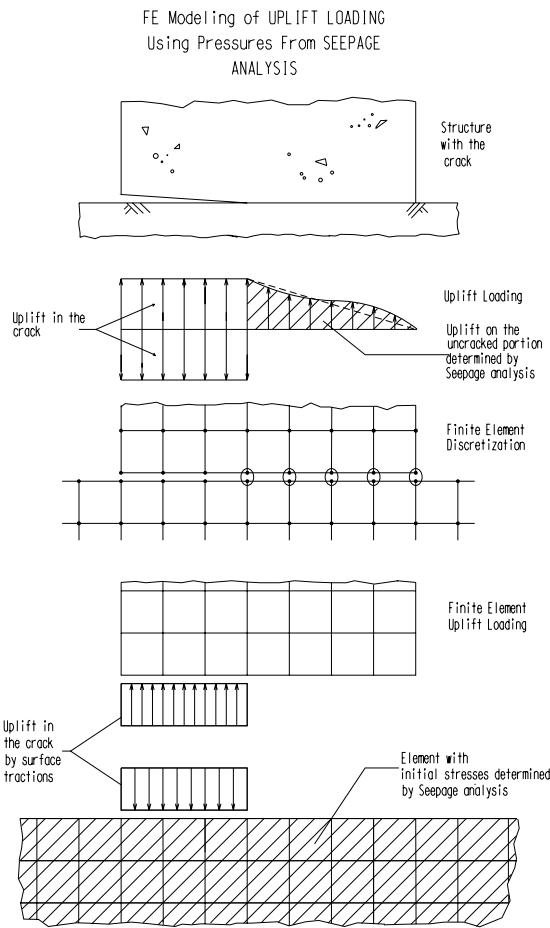


Figure 20.8: Uplift Model with Impervious Concrete and Pervious Rock and Uncracked Interface

the pressure at each node in the foundation using the **Pressures** option in MERLIN. These nodal pressures are computed in a separate steady-state seepage flow analysis. Using nodal pressures from a transient seepage flow analysis should be avoided because the flow conditions are coupled to the stress state for time-dependent poro-elastic problems. Duplicate nodes are used along the uncracked interface to “contain” the effect of the seepage pressures within the foundation.

Additional approaches may be theoretically valid, but those described above should suffice in most cases.

The modeling of uplift pressures on the uncracked ligament often requires a mesh construction technique commonly known as double or master/slave nodes, particularly when nodal pressures are specified. In MERLIN, they are called master/slave nodes, so this terminology will be used in this discussion. Master/Slave nodes are a pair of nodes that

have identical coordinates and displacements. Typically, the nodes of a master/slave node pair belong to elements with different material properties or where a discontinuity in stress is expected due to pore water pressure. For all of the uplift models described above master/slave nodes should be used along the interface between the concrete. For the second uplift model master/slave nodes should be used along the element boundaries shared by pervious and impervious elements; this keeps the applied loads due to hydrostatic pressures isolated within the pervious elements.

Hydrostatic pressures within pervious materials are specified using the **Pressures** option in MERLIN. A internal hydrostatic pressure load specification would appear as follows:

Pressures

9
101 70.0
102 67.5
103 65.0
104 62.5
105 60.0
106 57.5
107 55.0
108 52.5
109 50.0

In this case internal hydrostatic pressures are specified at nine different nodes in the mesh; the pressures for all other nodes are automatically assigned a value of zero. Specification of nodal pressures requires two pieces of information: a node number and the value of the pressure at that node. The nodal pressures can be assumed based on sound engineering judgement or obtained through a separate steady-state seepage flow analysis. MERLIN treats the pressures as hydrostatic initial stresses σ_0 assembling an consistent nodal force vector $\mathbf{f}_{\sigma_0}^e$ for each pressurized element by integrating the gradient of specified nodal pressures over the element domains

$$\mathbf{f}_{\sigma_0}^e = \int_{\Omega^e} \mathbf{B}^T \sigma_0 \, d\Omega \quad (20.9)$$

Finally, it should be noted that an internal hydrostatic pressure is the pressure within a pervious medium due to seepage or pore pressures; it should not be confused with the hydrostatic pressure due to water acting on the exterior of an impervious medium.

20.4 Finite Element Discretization

20.4.1 Mesh Dimensions

The finite element discretization depends on whether it will be used exclusively for a stress analysis, or for a combination of (uncoupled) seepage/stress analyses.

In the former, a mesh comprising the rock foundation extending the dam height H on either side and below the dam is recommended. In the second case, the recommended extension should be at least $2H$.

20.4.2 Boundary Conditions

The results should be insensitive to the choice of the boundary conditions around the foundation. Differing results obtained by placing either rollers or rigid supports around the boundary indicate that the foundation model should be extended.

20.4.3 Preliminary Cracks

20.4.3.1 Horizontal Crack

In the case of the primary horizontal crack which may cause dam instability, two approaches are possible:

1. Start with a mesh with no cracks, perform a linear elastic analysis, identify the node with highest tensile stress, and use the remeshing program to initiate a crack at this particular location.
2. Model a discrete crack as a gap between adjacent elements at the anticipated location of an existing or a potential crack. Unless known, the initial crack length should be no less than three element deep.

20.4.3.2 Rock Tensile Zone Cracks

In many cases, large horizontal tensile stresses occur in the rock under the dam's heel. In early analysis, Zienkiewicz (Zienkiewicz and Cheung 1964, Zienkiewicz and Cheung 1965) recommended the softening of this tensile zone by a reduced modulus of elasticity (one tenth the original value). Within the context of a discrete crack model, this can be equivalently replaced by the insertion of an initial crack. The crack would be vertical for homogeneous isotropic rock, or inclined for orthotropic jointed rock. Its initial length should not exceed $H/20$. Hence, this initial crack will typically relieve the tensile stresses at the base.

20.4.4 Element Types, and Mesh Density

The density of the mesh required to obtain accurate results is a function of both the distribution of the stress field in the structure and the element types used in the mesh. The presence of stress concentrators such as reentrant corners at the heel or toe will require some degree of mesh refinement or densification in those areas. The use of higher order elements allows for greater accuracy with fewer elements, but the lower order, high performance elements in MERLIN may still be computationally more efficient in some cases.

Whereas singular quarter point elements have been very popular to provide a very simple way of modeling the stress singularity and determining the stress intensity factors, their use is found to be cumbersome for the following reasons. Special attention must be paid to discretize the mesh around the crack tip with enough singular triangular elements of a size not exceeding 10% of the total crack length. For crack propagation studies, it was found that a simpler method would be one based on regular discretization around the crack tip, and use contour line integrals away from the crack tip to determine the stress intensity factors. Results were found to be quite robust, and mesh size insensitive.

Accordingly, element type 5 in MERLIN is recommended for 2-D analyses and element type 20 is recommended for 3-D analyses. These elements are low order, high performance elements which give good coarse mesh accuracy. These elements have been enhanced to provide improved bending mode behavior (Reich 1993). When using the mixed-iterative method in MERLIN, these elements also tend to exhibit both better convergence characteristics and higher accuracy than either the standard lower order or higher order elements.

Finally, it should be mentioned that at least two meshes should be prepared, and results between the "coarse" and the "fine" one should be within 10% to 15% to be considered satisfactory.

20.5 Stress Analysis

20.5.1 Linear versus Nonlinear Analysis

In a structural analysis, the choice of the appropriate fracture mechanics model is influenced by parameters such as the uncertainty of loads or material properties, availability of computer codes, computational cost and desired accuracy.

The advantages of a nonlinear analysis over a linear one are: 1) determination the size of the true crack and of the process zone in terms of the applied load, 2) capturing of the pre-peak nonlinear response of the structure, and 3) the post-peak response which for dam structures is of importance when deformations are induced by foundation settlement or temperature change. However, it should be kept in mind that nonlinear analyses are not simple to perform, and very few computer codes (including MERLIN) are capable of properly performing such an analysis.

Hence the following order of analysis should be followed, keeping in mind that should the results of a particular analysis prove satisfactory, then there may not be a need to undertake the subsequent one:

1. Two dimensional linear elastic fracture mechanics, with at least two different mesh sizes.
2. Two dimensional non-linear fracture analysis.
3. Three dimensional linear elastic fracture mechanics
4. Three dimensional nonlinear fracture analysis.

Note that the analysis complexity increases almost exponentially from one type to the other.

Furthermore, in a nonlinear fracture model, analysis should be interrupted at different stages depending on the load type:

1. For structures subjected to directly applied load (such as water pressure and uplift), it could stop once the peak displacement has been reached as this would be synonymous of collapse or failure. Any post-peak response would be purely academic.
2. For structures subjected to imposed displacements, such as foundation settlement or thermal stresses, then the analysis should proceed beyond the peak load.

20.5.2 Two versus Three-Dimensional Analysis

Two-dimensional analyses normally consider the transverse section only, and the crack is thus assumed to span the entire dam width from abutment to abutment. This approach may yield excessive stresses, which would not have occurred had a three-dimensional been performed. Hence, for narrow canyons and slightly curved dams a three dimensional analysis should be preferred.

Three-dimensional (3D) fracture mechanics analyses will be able to model not only a partial crack, but also the side restraining and horizontal beam beneficial effects. The obvious limitation of a 3D analysis is the extensive data preparation associated with it. With current technology, such an analysis can be undertaken but is likely to be expensive.

20.5.3 Stress Intensity Factor Extraction

There are numerous techniques to extract the stress intensity factors, and those can be broadly classified under two categories: The one based on correlation of nodal displacement in singular elements, and those based on contour or surface integrals.

The former requires the use of higher order elements, and in this case, the crack tip should be surrounded by at least 6 singular quarter point elements, and the singular element size should not exceed 15% of the crack length, (Saouma and Schwemmer 1984).

The second category of SIF extraction does not rely on the modeling of the stress singularity, but is based on a contour/surface integral taken around the crack tip, and results are independent of the selected path. This includes: the J -integral formulation of Hellen and Blackburn (Hellen and Blackburn 1975); the reciprocal work intergral of Stern, Becker, and Dunham (Stern et al. 1976); and the surface integral of Babuska and Miller (Babuska and Miller 1984).

Of all methods, the integrals of Stern, Becker, and Dunham and the surface one of Babuska and Miller tend to give the best results, and are recommended.

20.6 Seepage Analysis

20.6.1 Material Properties

In section 20.3.4 three models for the uplift pressure along the uncracked ligament were discussed. The model identified as **Case 3** requires a steady-state seepage flow analysis prior to the stress analysis. Transient seepage flow analysis should not be performed because the flow conditions are coupled to the stress state for time-dependent poro-elastic problems and MERLIN is not capable of performing this type of problem. Whereas a seepage analysis with isotropic rock conductivity is likely to yield a linear steady state pressure distribution, such an analysis should be undertaken for orthotropic cases.

Material properties required for a steady-state seepage flow analysis are summarized in Table 20.6.1.

Property	Isotropic	Orthotropic
Mass density	ρ	ρ
Permeability	k	θ, k_1, k_2

Table 20.5: Required Material Properties for Seepage Analysis

Note that for isotropic hydraulic conductivities the value of the permeability can be arbitrary.

In most cases, individual joints are not modeled, and hence the permeability should be that of a homogeneous continuum equivalent to the jointed rock system.

20.6.2 Finite Element Discretization

20.7 Thermal Analysis

In section 20.3.2 the application of thermal load was discussed, and shown that it is represented by nodal temperatures.

Nodal temperatures can be obtained through a steady state (time independent), or transient (time dependent) thermal analysis.

Such an analysis (as implemented in MERLIN), would enable the user to perform a thermal analysis, and then using the same finite element discretization, and the analysis output (nodal temperatures), perform an uncoupled stress analysis.

For transient analysis one must be very careful in the selection of the time step. Computer codes employing explicit solution techniques yield completely erroneous results when too big time step is used. Program MERLIN (Saouma et al. 2008) is based on an implicit method, therefore no limits on the size of the time step are necessary. However too big time step can cause inaccurate results. Interested reader should consult the MERLIN's example manual.

20.7.1 Material Properties

Material properties required for a thermal analysis are summarized in Table 20.7.1

	Steady-state	Transient
— Material Properties —		
mass density		ρ
Specific Heat		c
conductivity	k	k
— Boundary Conditions —		
Temperature	T	T
film	h	h
flux	q	q

Table 20.6: Material Parameters Required for a Thermal Analysis

Indicative values of concrete and rock conductivities are 1-5 and 1-2 BTU/Hr/Ft/ $^{\circ}$ F; For concrete, the specific heat can be assumed to be 0.22 BTU/lb/ $^{\circ}$ F, (Townsend n.d.).

It should be noted that in a stress analysis, results would be very sensitive to the selected coefficient of thermal expansion α .

20.7.2 Heat Transfer

In heat conduction problems, the primary field variable Φ in Eq. ?? is the temperature T , k is the thermal conductivity, Q (W/m³) is the rate of heat (positive) or sink (negative) generation, and c is the specific heat (J/ $^{\circ}$ C).

There are three fundamental modes of heat transfer:

Conduction: takes place when a temperature gradient exists within a material and is governed by Fourier's Law

$$q_x = -k_x \frac{\partial T}{\partial x} \quad q_y = -k_y \frac{\partial T}{\partial y} \quad (20.10)$$

where $T = T(x, y)$ is the temperature field in the medium, q_x and q_y are the components of the heat flux (W/m² or Btu/h.ft²), k is the thermal conductivity (W/m. $^{\circ}$ C or Btu/h.ft. $^{\circ}$ F) and $\frac{\partial T}{\partial x}$, $\frac{\partial T}{\partial y}$ are the temperature gradients along the x and y respectively. The resultant heat flux $\mathbf{q} = q_x \mathbf{i} + q_y \mathbf{j}$ is at right angles to an isotherm or a line of constant temperature. The minus sign indicates that flux is along the direction of decreasing temperature.

Convection: heat transfer takes place when a material is exposed to a moving fluid which is at different temperature.

It is governed by Newton's Law of Cooling

$$q = h(T_s - T_{\infty}) \quad (20.11)$$

where q is the convective heat flux (W/m²), h is the convection heat transfer coefficient or film coefficient (W/m². $^{\circ}$ C or Btu/h.ft². $^{\circ}$ F). It depends on various factors, such as whether convection is natural or forced, laminar or turbulent flow, type of fluid, and geometry of the body; T_s and T_{∞} are the surface and fluid temperature, respectively.

Radiation: is the energy transferred between two separated bodies at different temperatures by means of electromagnetic waves. The fundamental law is the Stefan-Boltzman's Law of Thermal Radiation for black bodies in which the flux is proportional to the fourth power of the absolute temperature., which causes the problem to be nonlinear. This mode of heat transfer is not considered by MERLIN.

Note that for steady state problems, c can be ignored.

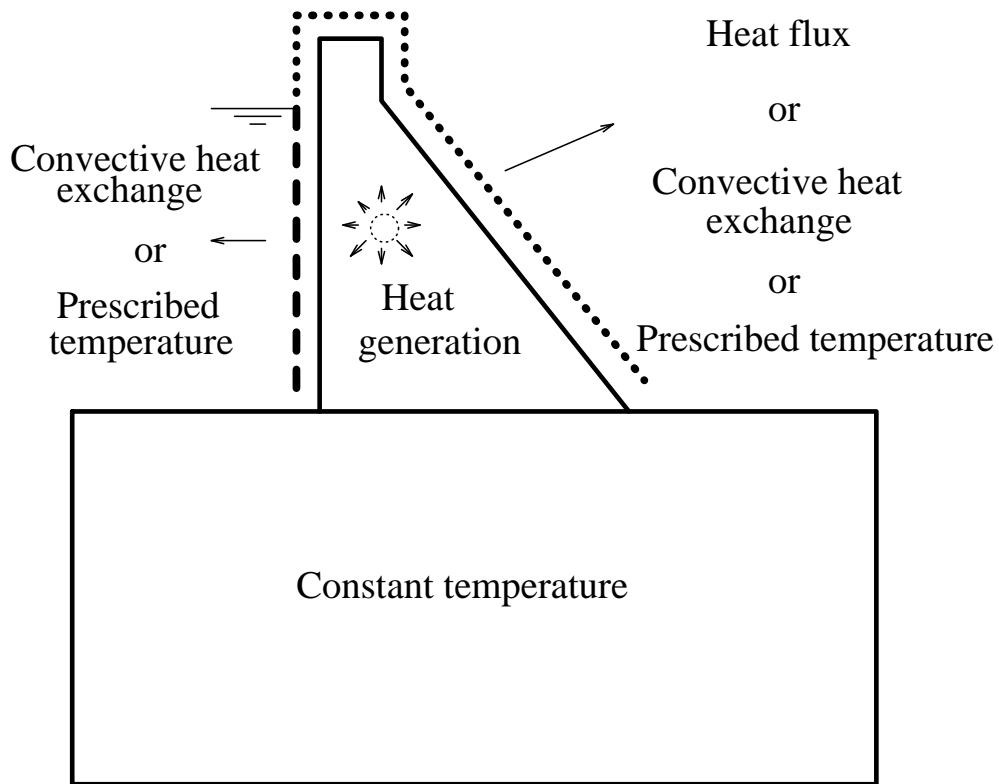


Figure 20.9: Boundary Conditions for Thermal Analysis

20.7.3 Boundary Conditions

The boundary conditions are mainly of three kinds, Fig. 20.9:

1. Specified temperature ($T = T_0$)
2. Specified heat flux ($q_n = q_0$), note an insulated surface will have zero flux across it, thus $q_n = 0$.
3. Specified convection ($q = h(T - T_\infty)$)

20.7.4 Seepage Analysis

In seepage problems, the primary field variable Φ in Eq. ?? is the hydraulic potential (or hydraulic/piezometric head) h , k_x and k_y are the permeabilities (m/day), and c the storativity.

The fluid velocity (or fluxes) components are obtained from Darcy's law as

$$v_x = -k_x \frac{\partial \Phi}{\partial x} \quad v_y = -k_y \frac{\partial \Phi}{\partial y} \quad (20.12)$$

Lines of $\Phi=\text{constant}$ are called equipotential surfaces, across which flow occurs.

20.7.5 Boundary Conditions

Two types of boundary conditions are applicable, Fig. 20.10:

prescribed head :

1. Caused by the known pressure head on the upstream and downstream side.
2. Experimentally measured through piezometer readings.

prescribed flux :

1. Zero flux should be specified around surface of the rock mass.
2. A point flux may be caused by a known flow through a drain.

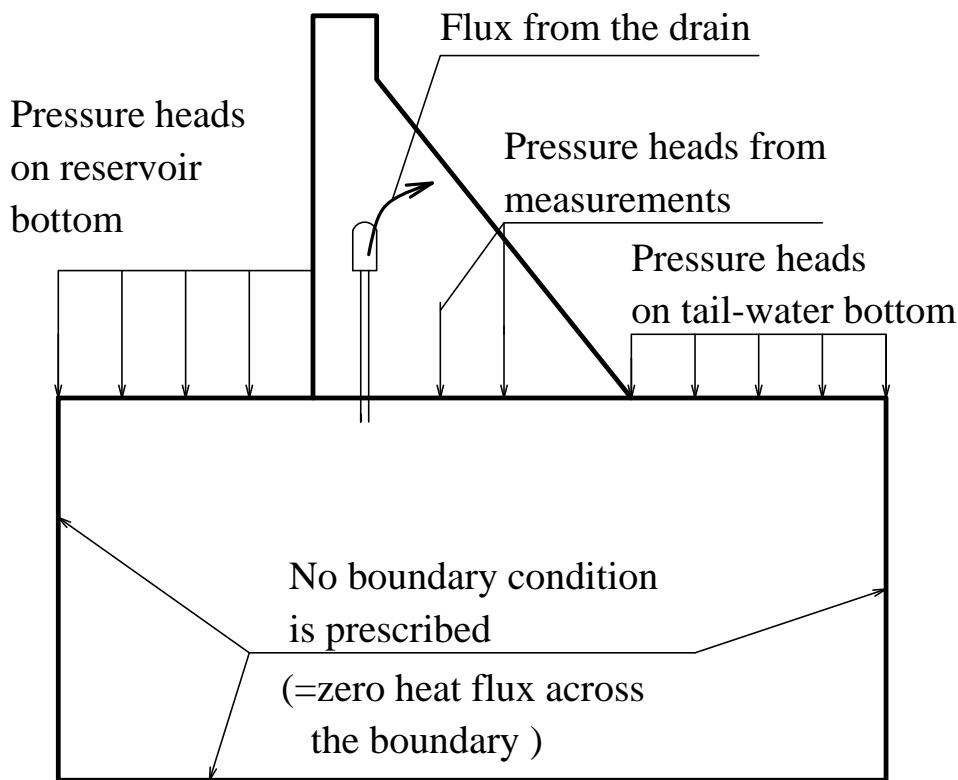


Figure 20.10: Boundary Conditions for Seepage Analysis

20.8 Units & Conversion Factors

length, m (meter)	1 inch = 0.0254 m; 1 m = 39.37 inch
Force, N (Newton)	1 lb = 4.4482 N; 1 N = 0.22481 lb
Mass, Kg (kilogram)	1 lbm = 0.45359 Kg; 1 Kg=2.2046 lb
Density, Kg/m ³	1 lbm/ft ³ = 16.018 Kg/m ³ ; 1 Kg/m ³ =0.062428 lbm/ft ³
Temperature, T	T °F=[(9/5)T°C+32]
Acceleration, m/s ²	1 in/s ² = 0.0254 m/s ² ;
Stiffness, N/m	1 lb/in = 175.1 N/m
Stress, Pa = N/m ²	1 psi = 6,894.8 Pa; 1 MPa = 145.04 psi
Work, energy, N-m=Joule	1 ft-lbf= 1.3558 J; 1 J = 0.73756 ft- lbf
Heat Transfer	
Convection coefficient, <i>h</i>	1 Btu/h.ft ² .°F = 5.6783 W/m ² .°C
Heat, J	1 Btu=1055.06 J; 1 Btu = 778.17 ft-lb
Heat Source/Sink, <i>Q</i>	W/m ³ =
Heat flux (<i>q</i>)	1 Btu/h.ft ² = 3.1546 W/m ²
Specific heat, <i>c</i>	1 Btu/°F = 1,899.108 J/°C
Thermal conductivity, <i>k</i>	1 Btu/h.ft.°F = 1.7307 W/m.°C
Seepage Flow	
permeability, <i>k</i>	
Fracture Mechanics	
Stress intensity factor, <i>K</i>	1 MPa√m=1.099 ksi√in
Fracture energy <i>G_F</i>	1 lb/in =.0057 N/m;

20.9 Metric Prefixes and Multipliers

Prefix	Abbreviation	Multiplier
tera	T	10^{12}
giga	G	10^9
mega	M	10^6
kilo	k	10^3
hecto	h	10^2
deca	da	10
deci	d	10^{-1}
centi	c	10^{-2}
milli	m	10^{-3}
micro	μ	10^{-6}
nano	n	10^{-9}
pico	p	10^{-12}

Chapter 21

PROGRAMMER's MANUAL

This appendix contains information which should be relevant only to those who are licensed to modify MERLIN's source code.

21.1 Introduction

MERLIN is three-dimensional, linear elastic finite element program based on the mixed-iterative method of Zienkiewicz (Zienkiewicz et al. 2005). In the mixed-iterative method all stress and strain quantities are nodal and values on the interior of an element are easily interpolated using shape functions, generally the same shape functions are used for displacements. Since the mixed-iterative method is an extension of the displacement method, it is also possible to perform analyses with MERLIN using the displacement method. When using the displacement method, stress and strain quantities are still projected to the nodes, but the nodal quantities are not used in any subsequent finite element computations. MERLIN also includes capabilities for performing fracture mechanics analyses using a discrete crack model; an implementation of Rice's J-integral (Rice 1968) has been included to compute stress intensity factors for linear elastic fracture mechanics and Hillerborg's fictitious crack model (FCM) (Hillerborg et al. 1976) has been included for nonlinear analysis of cementitious materials.

21.1.1 Scope of Document

This document was written for someone who wishes to modify existing capabilities of MERLIN or add new capabilities to MERLIN. It describes the various components of the program in sufficient detail to allow users to make their modifications with a minimum amount of effort. Subroutine and function argument lists for utilities that a programmer may find useful during the course of their modifications are defined and discussed in detail. Program examples using the utilities are included to clarify these discussions and demonstrate proper usage of the utilities.

21.1.2 Organization of Document

This part is organized in sections which are intended to be relatively independent of one another; in situations where this is not true, the reader will be alerted as to which sections are not independent. First time readers may want to browse the entire document to familiarize themselves with the contents of each section for future reference. The file I/O utilities used for all file handling in the program are described in section 21.2. The memory management utilities that handle the partitioning of the large array that serves as program are discussed in Csection 21.3. The contents and organization of the finite element attribute tables, which are used to define element types in the element library and constitutive models in the material library, are described in section 21.4.

21.1.3 File Naming Conventions For Source Code

The source code for MERLIN is primarily FORTRAN 77 with a few utility routines written in C. File names are constructed such that there is a root file name and a file extension. The root file name generally indicates the functionality of the source in the file and the file extension indicates the source code language. Files containing FORTRAN source code have the extension .f. Files containing C source code have the extension .c.

21.1.4 Creating an Executable

The makefile UNIX utility is used to handle the task of compiling the source code and linking the object modules into an executable code. Dependencies for all the "include" files are defined in the `makefile` so that the programmer need not be concerned with where a particular include file is referenced. If an include file is modified every source code file referencing that include file will be recompiled.

21.1.5 Coding Standards

A consistent coding standard throughout code is necessary to allow a developer to become comfortable with the program more quickly. A short discussion of each point in the MERLIN coding standard is included in the following subsections so that the developer may better understand how these coding standards actually make their life easier.

21.1.5.1 Include Files

All common blocks and sub-system control variables defined via PARAMETER statements are defined in include files. The naming convention for include files is based on their contents; include files containing common blocks have the extension **.cmn** and include files containing parameters have the extension **.par**. The different extensions allow the developer to determine the contents of an include file without using a text editor.

21.1.5.2 Case Sensitivity

FORTRAN 77 is not a case sensitive language. However, lower case code is allowed on all machines on which MERLIN is available, so this feature is taken advantage of in the source code. All source code is in lower case with the exception of the following exceptions:

- Global or common variables begin with an upper case letter; all other characters are lower case.
- Variables defined via PARAMETER statements are all upper case.

This allows the developer to quickly determine where a particular variable comes from.

21.1.5.3 Variable Declarations

All variables are declared; the FORTRAN standard for variable types is ignored. To assure that all variables are declared, the '**-u**' option is used when the source code is compiled on the Sun. Explicit declaration of all variables eliminates typographical errors and the omission of function subroutine type declarations, both of which are fairly tricky bugs to locate.

21.1.5.4 DO Loops

All DO loops end on separate CONTINUE statements and the code between a DO statement and the corresponding CONTINUE statement is indented. This is more a matter of readability than anything else, but often comes in very handy when examining a particularly long DO loop.

21.1.5.5 RETURN Statement

Ideally, a subroutine should have only one RETURN statement. This is particularly true with long subroutines where multiple RETURN statements make it difficult to follow the logic of the subroutine. It can be especially difficult to debug a subroutine with multiple RETURN statements and for these reasons the use of one RETURN statement per subroutine is strongly advocated. If there are multiple conditions that require returning to the calling subroutine the effect of multiple RETURN statements can be duplicated with one labeled RETURN statement and GOTO statements using the label corresponding to the RETURN statement. The statement label will appear in a compiled listing with cross-references easily allowing a programmer to pinpoint the return conditions.

21.1.5.6 Statement Labels

Ideally, the numbers used as statement labels in a given subroutine should appear in increasing order and the numbers should be evenly spaced. The statement labels found in most subroutines are multiples of ten. When there is a statement label associated with the RETURN statement this statement label is generally 999. Statement labels associated with format statements are generally numbers between 7000 and 9990, inclusive. Statement labels between 7000 and 7990 are associated with debug prints; statement labels between 8000 and 8990 are associated with echoing input data; and statement labels between 9000 and 9990 are associated with error, warning, and informational messages written to the formatted output file. Associating a particular number or range of numbers with a certain type of format statement makes it easy to locate and/or recognize these statements when examining a compiled listing with cross-references.

21.1.5.7 ANSI Standard Features

The use of extensions to the FORTRAN 77 standard should be avoided at all costs for the sake of portability. Extensions to the standard are generally identified in some way in the FORTRAN manual for a given compiler. If in doubt, consult the manual to be sure.

21.2 File I/O Utilities

The data that defines a finite element model are read from a file and the results of the analysis are written to files. For the sake of portability, all disk I/O functionality is isolated in utilities that are called in place of the “standard” FORTRAN I/O library. These utilities are written primarily in FORTRAN callable C, but include a few FORTRAN subroutines. Functions from the buffered I/O library of C are used as the basis for the file I/O utilities and the FORTRAN utilities provide additional functionality.

21.2.1 I/O Utilities Written In C

Functions from the standard C buffered I/O library are used to open and close files and to read and write information to and from these files. These utility functions are capable of handling the I/O for a maximum of twenty concurrently open files. Both formatted and sequential binary I/O are supported. All utility functions written in C are FORTRAN callable and include internal error handling. The source code for those I/O utilities written in C are in the file **ioutil.c**.

21.2.1.1 File Attribute Data Structure

An array of data structures is used to store attributes for open files. Access into this array of data structures is facilitated through the use of file identifiers, which are nothing more than indices into the array of data structures ranging in value from 0 to 19. Array elements 0, 1, and 2 are used by standard input, output, and error, respectively. The file identifiers are used so that the various open files can be manipulated through FORTRAN, which does not include pointers to data structures as a standard data type. These attributes are used extensively throughout the utility functions. The template for the file attribute data structure is as follows:

```
typedef struct {
    char name[129];
    FILE *stream;
    FILE *errout;
    int last_op;
} FileInfo;
```

The elements of the file attribute data structure have the following functionality:

- **name** is the name of the file.
- **stream** is the stream from which information is read and to which data is written.
- **errout** is the stream to which any error messages associated with an I/O operation are printed.
- **last_op** is a flag which indicates whether the last operation was a read or a write.

The stream is used for every I/O operation, but the file name and the error output stream are used only when printing error messages. The default error output stream is **stderr**. Changing the stream for error output is discussed in Section 21.2.1.4. The structure element **last_op** is used to automatically synchronize I/O for update or read/write files.

21.2.1.2 Open Function

The opening of files is handled by the FORTRAN callable function **filopn**. **filopn** is of type **int** and, therefore, is called as a function subroutine of type **INTEGER*4**. The value returned by **filopn** is either a file identifier or an error indicator. The file error indicator is a -1; any other value is a valid file identifier. The file identifier returned by **filopn** is the means by which a particular file is specified to the other I/O utilities.

The function and argument declarations for **filopn** in FORTRAN are as follows:

```
integer*4 function filopn( name , mode )
character*(*) name
character*(*) mode
```

where **name** is the filename and **mode** indicates how the file will be opened. Valid values for **mode** are given in Table 21.1; appending a ‘b’ after these file modes indicates a binary file. Both **name** and **mode** must be NULL terminated strings.

NULL termination of FORTRAN character strings is performed by the function subroutine **nulstr**. The function and argument declarations for **nulstr** are as follows:

Open Mode	Function
'r'	Open text file for reading
'w'	Create text file for writing; discard previous contents if any
'a'	Append; open or create text file for writing at end of file
'r+'	Open text file for update (i.e., reading and writing)
'w+'	Create text file for update; discard previous contents if any
'a+'	Append; open or create text file for update, writing at end

Table 21.1: File Open Modes

```
character*(* function nulstr( string )
character*(* string
```

The NULL character is added immediately following the last non-blank character in **string**. Therefore, when **nulstr** is declared in the calling subroutine its length must be at least one character greater than that of **string** or the NULL character cannot be appended to **string**.

Before attempting to open a file, an unused element in the array of file attribute data structures must be located. Failure to locate an unused element will cause **filopn** to return a value of -1. Once an unused element is located, an attempt to open the file is made. If the file opening operation is successful, the attributes for the file are stored in the unused element of the file attribute array and the file identifier, which is the array index corresponding to the previously unused element, is returned as the function value. Otherwise, a value of -1 is returned. Any type of failure within **filopn** is accompanied by an error message written to standard error.

21.2.1.3 Close Function

The closing of files is handled by the FORTRAN callable function **filcls**. **filcls** is of type **void** and, therefore, is called as a subroutine. The function and argument declarations for **filcls** in FORTRAN are as follows:

```
subroutine filcls( fid )
integer*4 fid
```

where **fid** is the file identifier for the file to be closed. **filcls** check the value of **fid** before actually attempting to close the file to make sure it is between 3 and 19, inclusive, and that the file corresponding to that file identifier is indeed open. Upon closing the file, the element in the array of file attribute data structures corresponding to the file identifier is released for reuse.

21.2.1.4 Error Output

Once a file is open, it is possible to have any subsequent error messages that are the result of an illegal I/O operation printed to a file opened by **filopn** rather than to standard error. In MERLIN, all I/O errors associated with read, write, and seek operations are printed to the formatted output file. Redirection of error messages is handled by the FORTRAN callable function **filerr**. **filerr** is of type **void** and, therefore, is called as a subroutine. The function and argument declarations for **filerr** in FORTRAN are as follows:

```
subroutine filerr( fid , fiderr )
integer*4 fid
integer*4 fiderr
```

where **fid** is the file identifier of file for which error messages are to be redirected and **fiderr** the file identifier of file to which the error messages are to be redirected. Both **fid** and **fiderr** must be file identifiers for open files or the attempt to redirect error output will be ignored.

21.2.1.5 Read Functions

The file I/O utilities include a number of FORTRAN callable functions for read operations. Naturally, there are different functions for reading from both text (i.e., formatted or ASCII) files and binary (i.e., unformatted) files. There are currently three functions available for reading from text files:

1. **rdstr** reads a character string,
2. **rdlong** reads an array of 32-bit integers, and

3. **rddble** reads an array of double precision floating points.

However, there is only one function required for reading from binary files, **rdbin**. Each of these functions is of type **int** and, therefore, is called as a function subroutine of type INTEGER*4. The value returned by these functions indicates how many items were in fact read. Any number less than the specified value can generally be regarded as an error.

The function and argument declarations for **rdstr** in FORTRAN are as follows:

```
integer*4 function rdstr( fid , string )
integer*4 fid
character*(*) string
```

where **fid** the file identifier of the file from which to read and **string** is the character string read from the file. A character string is considered to be those characters that fall in between two white space characters (i.e., blank, tab, newline, carriage return, vertical tab, and formfeed). The character string is read into a temporary buffer and then copied to **string**. If **string** is not large enough to accomodate the character string read from file, only those characters that will fit into **string** are returned. In any case, the NULL terminator is stripped from the character string during the copy operation since it is not necessary in FORTRAN. The function value is 1 if the character string was read successfully, otherwise it is 0. A function value of zero is accompanied by an error message to the error output file stream.

The function and argument declarations for **rdlong** in FORTRAN are as follows:

```
integer*4 function rdlong( fid , count , array )
integer*4 fid
integer*4 count
integer*4 array(count)
```

where **fid** the file identifier of the file from which to read, **count** is the number of integers to read, and **array** is the array of integers read from the file. Each integer must be bracketed by white space characters (i.e., blank, tab, newline, carriage return, vertical tab, and formfeed). The function value is **count** if the integer array was read successfully, otherwise it is a value less than **count**. A function value less than **count** is accompanied by an error message to the error output file stream.

The function and argument declarations for **rddble** in FORTRAN are as follows:

```
integer*4 function rddble( fid , count , array )
integer*4 fid
integer*4 count
real*8 array(count)
```

where **fid** the file identifier of the file from which to read, **count** is the number of double precision floating points to read, and **array** is the array of double precision floating points read from the file. Each floating point must be bracketed by white space characters (i.e., blank, tab, newline, carriage return, vertical tab, and formfeed). The function value is **count** if the double precision floating point array was read successfully, otherwise it is a value less than **count**. A function value less than **count** is accompanied by an error message to the error output file stream.

The function and argument declarations for **rdbin** in FORTRAN are as follows:

```
integer*4 function rdbin( fid , count , array )
integer*4 fid
integer*4 count
character array(count)
```

where **fid** the file identifier of the file from which to read, **count** is the number of characters (i.e., bytes) to read, and **array** is the array of characters read from the file. In this case, the use of data type **CHARACTER** is not entirely appropriate for **array**, but it is the only FORTRAN data type available that easily translates to bytes and bytes are the unit of measurement for binary files in Unix. The function value is **count** if the ‘character’ array was read successfully, otherwise it is a value less than **count**. A function value less than **count** is accompanied by an error message to the error output file stream.

The number of bytes associated with any FORTRAN data type can be determined using the function subroutine **sizeof**. The function and argument declarations for **sizeof** are as follows:

```
integer*4 function sizeof( type )
integer*4 type
```

where **type** is data type for which the corresponding number of bytes are required. **sizeof** returns as its value the number of bytes corresponding to a given data type. **type** can be specified using the parameters defined in the include file **pmmkey.par** and described in Section 21.3.3.3.

21.2.1.6 Write Functions

The file I/O utilities includes two FORTRAN callable functions for write operations; one for writing to text files, **wrtstr**, and one for writing to binary files, **wrtbin**. Each of these functions is of type **int** and, therefore, is called as a function subroutine of type INTEGER*4. The value returned by these functions indicates how many items were in fact written. Any number less than the specified value can generally be regarded as an error.

The function and argument declarations for **wrtstr** in FORTRAN are as follows:

```
integer*4 function wrtstr( fid , string )
integer*4 fid
character*(*) string
```

where **fid** the file identifier of the file to write to and **string** is the character string to be written to the file. Since all character string operations in C require that the character string be NULL terminated, **string** is copied into a temporary buffer and the terminating NULL character is inserted immediately following the last non-blank character before it is written to the file. A newline character is also written to the file following the character string. The function value is 1 if the character string was written successfully, otherwise it is 0. A function value of zero is accompanied by an error message to the error output file stream.

The function and argument declarations for **wrtbin** in FORTRAN are as follows:

```
integer*4 function wrtbin( fid , count , array )
integer*4 fid
integer*4 count
character array(count)
```

where **fid** the file identifier of the file to write to, **count** is the number of characters (i.e., bytes) to write, and **array** is the array of characters to write to the file. In this case, the use of data type **CHARACTER** is not entirely appropriate for **array**, but it is the only FORTRAN data type available that easily translates to bytes and bytes are the unit of measurement for binary files in Unix. The function value is **count** if the ‘character’ array was written successfully, otherwise it is a value less than **count**. A function value less than **count** is accompanied by an error message to the error output file stream.

21.2.1.7 Seek Function

In binary files it is often advantageous to move the file pointer around the file without actually reading information into program memory. Manipulation of the file pointer is performed by the FORTRAN callable function **filpos**. **filpos** is of type **int** and, therefore, is called as a function subroutine of type INTEGER*4. A value is returned whether or not the operation succeeded; the function value is 0 if the operation was successful and non-zero if it was not.

The function and argument declarations for **filpos** in FORTRAN are as follows:

```
integer*4 function filpos( fid , offset )
integer*4 fid
integer*4 offset
```

where **fid** is the file identifier of the file for which the file pointer is to be repositioned and **offset** is the number of bytes that the pointer is to be moved from the current position. Negative values are valid for **offset** and they indicate the file position will be moved backward through the file rather than forward. A non-zero function value is accompanied by an error message to the error output file stream.

21.2.1.8 Flush Function

When performing file I/O operations, there will be situations in which it is advantageous to flush the contents of the I/O buffer before it is filled. Flushing of the I/O buffer is handled by the FORTRAN callable function **filclr**. **filclr** is of type **void** and, therefore, is called as a subroutine. The function and argument declarations for **filclr** in FORTRAN are as follows:

```
subroutine filclr( fid )
  integer*4 fid
```

where **fid** is the file identifier for the file for which the I/O buffer is to be flushed. **filclr** checks the value of **fid** before actually attempting to flush the I/O buffer to make sure it is between 3 and 19, inclusive, and that the file corresponding to that file identifier is indeed open. If **fid** corresponds to a file that is not open, there is no I/O buffer to be flushed and the attempt to flush the I/O buffer will be ignored.

21.2.2 I/O Utilities Written In FORTRAN

Currently, there is only one FORTRAN function in the file I/O utilities functions, **frmwrt**. **frmwrt** is a function subroutine of type **INTEGER*4** that writes an array of character strings to a text file one element at a time using **wrtstr**. The function value returned indicates how many elements of the array were successfully written to the file.

The function and argument declarations for **frmwrt** are as follows:

```
integer*4 function frmwrt( fid , count , array )
  integer*4 fid
  integer*4 count
  character*(*) array(count)
```

where **fid** is the file identifier of the file to write to, **count** is the number of character strings to write, and **array** is the array of character strings to write to the file. The function value is **count** if the character strings were written successfully, otherwise it is a value less than **count**. A function value less than **count** is accompanied by an error message to the error output file stream.

21.2.3 Usage of the File I/O Utilities

Because the majority of the I/O utilities are written in C and called from FORTRAN there are some idiosyncracies in these utilities that should be pointed out at this time. A simple program that reads a character string, an array of integers, and an array of double precision floating points from a text file and copies them to a binary file is used to identify these idiosyncracies for the programmer. All error messages associated with the I/O utilities are redirected to a separate text file, which will also contain error messages for any invalid input. Comments in the code will indicate points of special interest.

```
program testio
c
c  This program illustrates the use of the file I/O utilities in MERLIN.
c
c      include 'include/pmmkey.par'
c
c  Local Variable Type Declarations:
c
c      character*80  title , string (      3)
c      integer*4    buffer (      20), inpfid , outfid , logfid , nbytes ,
c      &          numint , numflt , nread ,   ints (   100), nwrite
c      real*8       reals (   100)
c
c  Function Type Declarations:
c
c      character*40 nulstr
c      integer*4    filopn ,  rdstr , wrtbin , rdlong , frmwrt , rddble
c
c      equivalence (buffer (      1),  title)
c
c  Open the text file for input
```

```

c
    inpfid = filopn( nulstr( 'input.dat' ) , nulstr( 'r' ) )
    if (inpfid .lt. 0) goto 999
c
c   Open the binary file for output
c
    outfid = filopn( nulstr( 'output.dat' ) , nulstr( 'wb' ) )
    if (outfid .lt. 0) goto 30
c
c   Open a text file for error output
c
    logfid = filopn( nulstr( 'error.dat' ) , nulstr( 'w' ) )
    if (logfid .lt. 0) goto 20
c
c   Redirect all file I/O error messages to 'error.dat'
c
    call filerr( inpfid , logfid )
    call filerr( outfid , logfid )
c
c   NOTE: Error messages for 'error.dat' were not redirected from
c         standard error to 'error.dat' because chances are if it
c         is not possible to write to 'error.dat' through the
c         utility functions it is not possible to write to it at
c         all.
c
c   Read file title (i.e., a character string with <= 80 characters)
c   from the text file
c
    if (rdstr( inpfid , title ) .ne. 1) goto 10
c
c   Write the title to the binary file; use the integer array that
c   has been equivalenced to the character string containing the
c   title because rdbin does not include the extra argument in its
c   argument list that indicates the length of the character string.
c   The extra argument is necessary for cross-language communication
c   between FORTRAN and C.
c
    nbytes = len( title )
    nwrite = wrtbin( outfid , nbytes , buffer )
    if (nwrite .ne. nbytes) goto 10
c
c   Read the number of integers and floating points to be read
c   from the text file
c
    if (rdlong( inpfid , 1 , numint ) .ne. 1) goto 10
    if (rdlong( inpfid , 1 , numflt ) .ne. 1) goto 10
c
c   Check to make sure both numbers are >= 0 and <= 100; print
c   an error message if they are not. The error message is first
c   written to an array of character strings using an internal
c   write (one line of the error message appears in each character
c   string) and then it is written to 'error.dat' using frmwr.
c
    if (numint .lt. 0 .or. numflt .lt. 0) then
        write( string , 9000 )
        nwrite = frmwr( logfid , 3 , string )
        goto 10
    else if (numint .gt. 100 .or. numflt .gt. 100) then
        write( string , 9010 )

```

```

nwrite = frmwr( logfid , 3 , string )
      goto 10
end if
c
c   Write the number of integers and floating points to be read
c   to the binary file
c
nbytes = sizeof( LONG ) * 2
nwrite = wrtbin( outfid , nbytes , numint )
if (nwrite .ne. nbytes) goto 10
nwrite = wrtbin( outfid , nbytes , numflt )
if (nwrite .ne. nbytes) goto 10
c
      if (numint .gt. 0) then
c
c   Read the integer array from the text file
c
nread = rdlong( inpfid , numint ,    ints )
if (nread .ne. numint) goto 10
c
c   Write the integer array to the binary file
c
nbytes = sizeof( LONG ) * numint
nwrite = wrtbin( outfid , nbytes ,    ints )
if (nwrite .ne. nbytes) goto 10
end if
c
      if (numflt .gt. 0) then
c
c   Read the floating point array from the text file
c
nread = rddble( inpfid , numint ,  reals )
if (nread .ne. numflt) goto 10
c
c   Write the integer array to the binary file
c
nbytes = sizeof( DOUBLE ) * numint
nwrite = wrtbin( outfid , nbytes ,  reals )
if (nwrite .ne. nbytes) goto 10
end if
c
c   All done; close files before exit
c
10 call filcls( logfid )
20 call filcls( outfid )
30 call filcls( inpfid )
c
999 exit( 0 )
c
      stop
9000 format('Both the number of integers and floating points ',
           &          'to read must be greater than',
           &          '/or equal to 0.  File copy has failed.')
9010 format('Both the number of integers and floating points ',
           &          'to read must be less than or',
           &          '/equal to 100.  File copy has failed.')
end

```

21.3 Program Memory Management

All arrays with dimensions dependent on information read from the input file are stored in a partitioned one-dimensional INTEGER*4 array. Henceforth, this array will be referred to as **program memory** and the utility functions that manage program memory are **program memory management utilities**. Section 21.3.1 includes a description of the program memory array and how to increase or decrease the size of program memory. Section 21.3.2 is a discussion of the data structures used for the data stored in program memory. Section 21.3.3 includes a discussion of the memory management subsystem implemented to manage the data stored in program memory.

21.3.1 Program Memory

The program memory array is in common block **memory** and is named **Kmn**. This common block can be found in include file **memory.cmn**. Both the starting and ending array indices are specified by values set via a PARAMETER statement in **memory.cmn**. The starting array index is named **BOM** and the ending array index is named **EOM**. Currently, the values for **BOM** and **EOM** are 0 and 1499999, respectively.

Program memory can be resized simply by modifying the value of **EOM**, recompiling those source code files that include **memory.cmn**, and relinking the object files into a new executable. This operation is performed automatically by the makefile (see Section 21.1.4). In general, the value of **BOM** should not be changed.

21.3.2 Data Structures

As FORTRAN 77 is the language in which the majority of the MERLIN source code is written, the data structures are arrays. One-, two-, and three-dimensional arrays partitioned from program memory are used to store data. Throughout the remainder of this document the third dimension of a three-dimensional array will be referred to as a **page**. Since FORTRAN is a column major language (i.e., the data contained within a given column of an array is stored sequentially in core memory) it is very desirable to store data in arrays such that it is accessed down columns instead of across rows or pages. In fact, it is inefficient to access data in any other manner.

21.3.3 Memory Management Utilities

The memory management utilities used in MERLIN are based on a package found in the CAL structural analysis program. As this software was not completely autonomous, modifications were required to implement an ‘easy-to-use’ memory manager in MERLIN. The following subsections will describe the dynamic memory allocation and memory management utilities implemented in MERLIN.

21.3.3.1 Partitioned Program Memory

The number of partitions (i.e., data arrays) allowed is limited only by the amount of program memory available. Each data array seven attributes associated with it that include:

1. A six character mnemonic name (stored one character per word) identifying the array,
2. The type of data stored in the array,
3. The starting address of the array in program memory,
4. The number of rows,
5. The number of columns,
6. The number of pages, and
7. The size of the array in words.

Twelve words are required to store the seven array attributes for each data array. These attributes are stored contiguously at the end of program memory and the data arrays are stored contiguously at the beginning of program memory; free program memory resides in the space between the data arrays and their attributes. This storage scheme is illustrated by Figure 21.1.

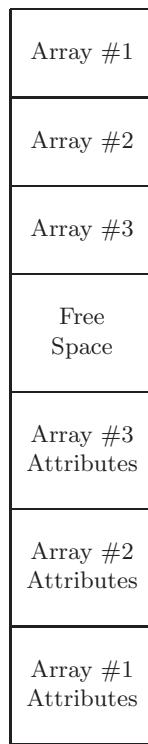


Figure 21.1: Program Memory with Three Arrays

21.3.3.2 Memory Management Routines

Six function subroutines are required to allocate and manage memory: one subroutine to allocate program memory for a data array, **alloc8**; one subroutine to resize an allocated data array (i.e., reallocate a data array), **realloc**; one subroutine to copy the contents of an allocated data array to another allocated data array, **copy**; one subroutine to locate a data array in program memory, **locate**; one subroutine to determine the attributes of a data array, **query**; and one subroutine to delete a data array from program memory, **delete**. An additional subroutine, **pmmmini**, is also required to initialize the global (i.e. common) variables used by the other six memory management utility subroutines.

The function and argument declarations for function subroutine **alloc8** are as follows:

```

integer*4 function alloc8( name , type , nrow , ncol , npag )
character*6 name
integer*4   type
integer*4   nrow
integer*4   ncol
integer*4   npag

```

where **name** is the data array name, **type** is the data type, **nrow** is the number of rows in the data array, **ncol** is the number of columns in the data array, and **npag** is the number of pages in the data array. Naturally, all array dimensions should be greater than zero; a one is used to indicate that a specific array dimension is not required. For example, the number of pages for a two-dimensional array is 1 and the number of columns and pages for a one-dimensional array is 1. The starting address of the array in program memory (i.e., the array index into **Kmn**) is returned as the function value. If there is insufficient free space in program memory to allocate the array an error indicator, **NULL** from include file **pmmkey.par** (see Section 21.3.3.3), is returned as the function value.

The function and argument declarations for function subroutine **realloc** are as follows:

```

integer*4 function realloc( name , nrow , ncol , npag )
character*6 name
integer*4   nrow

```

```
integer*4 ncol
integer*4 npag
```

where bf name is the data array name, **nrow** is the number of rows in the data array, **ncol** is the number of columns in the data array, and **npag** is the number of pages in the data array. Naturally, all array dimensions should be greater than zero; a one is used to indicate that a specific array dimension is not required. For example, the number of pages for a two-dimensional array is 1 and the number of columns and pages for a one-dimensional array is 1. Currently, the size of an array can only be increased; an attempt to decrease the size on an array will fail. The starting address of the array in program memory (i.e., the array index into **Kmn**) is returned as the function value. If the array name cannot be located in the attributes table or there is insufficient free space in program memory to reallocate the array an error indicator, NULL from include file **pmmkey.par** (see Section 21.3.3.3), is returned as the function value.

The function and argument declarations for function subroutine **copy** are as follows:

```
integer*4 function copy( from , to )
character*6 from
character*6 to
```

where bf from is the data array name from which information is to be copied and **to** is the data array name to which information is to be copied. The starting address of the **to** array in program memory (i.e., the array index into **Kmn**) is returned as the function value. If either of the array names cannot be located in the attributes table or the dimensions of the two data arrays are inconsistant (i.e., the number of rows, columns, and pages must all be identical) an error indicator, NULL from include file **pmmkey.par** (see Section 21.3.3.3), is returned as the function value.

The function and argument declarations for function subroutine **delete** are as follows:

```
integer*4 function locate( name )
character*6 name
```

where bf name is the data array name. The starting address of the array in program memory (i.e., the array index into **Kmn**) is returned as the function value. If the array name cannot be located in the attributes table an error indicator, NULL from include file **pmmkey.par** (see Section 21.3.3.3), is returned as the function value.

The function and argument declarations for function subroutine **query** are as follows:

```
integer*4 function query( name , type , nrow , ncol , npag )
character*6 name
integer*4 type
integer*4 nrow
integer*4 ncol
integer*4 npag
```

where bf name is the data array name, **type** is the data type, **nrow** is the number of rows in the data array, **ncol** is the number of columns in the data array, and **npag** is the number of pages in the data array. For this subroutine, only **name** is an input argument; **type**, **nrow**, **ncol**, and **npag** are all output arguments whose values correspond to those stored in the file attribute table. The starting address of the array in program memory (i.e., the array index into **Kmn**) is returned as the function value. If the array name cannot be located in the attributes table an error indicator, NULL from include file **pmmkey.par** (see Section 21.3.3.3), is returned as the function value.

The function and argument declarations for function subroutine **delete** are as follows:

```
integer*4 function delete( name )
character*6 name
```

where bf name is the data array name. When an array is deleted from program memory the contents of program memory are shifted so the the data arrays and the attribute table are contiguous. This means that the starting address of any data array that was allocated after the deleted array now starts at a different address and, therefore, must be located in order to determine what the value of the new address is. The former starting address of the array in program memory (i.e., the array index into **Kmn**) is returned as the function value. If the array name cannot be located in the attributes table an error indicator, NULL from include file **pmmkey.par** (see Section 21.3.3.3), is returned as the function value.

21.3.3.3 MERLIN Implementation

All program memory management utilities return an INTEGER*4 value that may or may not indicate that an error occurred and an INTEGER*4 value is required to specify the type of data to be stored in an array to be allocated. As it may be difficult for the programmer to remember the values that correspond to the various data types or the value that indicates the occurrence of an error, an include file containing **keys** for the memory management routines is provided to assist the MERLIN programmer. A key is a constant defined via a PARAMETER statement with a mnemonic name that the programmer can easily remember. This include file is called **pmmkey.par**. The parameters contained in this include file and their functionality is as follows:

- SHORT specifies an INTEGER*2 data type,
- LONG specifies an INTEGER*4 data type,
- FLOAT specifies a REAL*4 data type,
- DOUBLE specifies a REAL*8 data type, and
- NULL indicates that an error occurred.

These parameters should not be used as an argument to a subroutine that may be reset within the subroutine, as values initialized by the PARAMETER statement cannot be reset. A fatal error resulting in a program crash will result if this happens.

21.3.3.4 Usage of Memory Management Routines

The memory management subsystem is designed so that data arrays are allocated from program memory one or more subroutine levels above where they are needed and passed on to those routines as arguments using the starting address in program memory. The overhead incurred by passing local arrays, such as the element stiffness matrix, to subroutines called from the analysis driver has been reduced by putting their array addresses in a common block separate from global addresses and using this common block along with the program memory array in those subroutines where they are needed.

The following source code example demonstrates the use of the program memory management subsystem and the coding standards discussed in Sections 21.1.5.1 through 21.1.5.4:

```

program testmm
c
c  This program illustrates the use of the memory management
c  utilities in MERLIN.
c
      include 'include/pmmkey.par'
      include 'include/memory.cmn'
c
c  Local Variable Type Declarations:
c
      character*40 string (      5)
      integer*4    NUMROW , NUMCOL , NUMPAG , logfid ,   ptr ,   type ,
      &           ptr1 ,   nrow ,   ncol ,   npag ,   ptr2 ,   ptr3 ,
      &           ptr4
c
c  Function Type Declarations:
c
      character*40 nulstr
      integer*4    filopn , alloc8 ,  query , frmwr , delete
c
c  Define array dimensions
c
      parameter    (NUMROW =      3 , NUMCOL =      3 , NUMPAG =      1)
c
c  Open a text file for error output
c

```

```

logfid = filopn( nulstr( 'error.dat' ) , nulstr( 'w' ) )
if (logfid .lt. 0) goto 999
c
c Initialize program memory
c
call pmmmini
c
c Allocate one array of each type; the name of the array corresponds
c to data type to be stored in the array
c
ptr = alloc8( 'SHORT' , SHORT , NUMROW , NUMCOL , NUMPAG )
if (ptr .eq. NULL) goto 50
ptr = alloc8( 'LONG' , LONG , NUMROW , NUMCOL , NUMPAG )
if (ptr .eq. NULL) goto 40
ptr = alloc8( 'FLOAT' , FLOAT , NUMROW , NUMCOL , NUMPAG )
if (ptr .eq. NULL) goto 30
ptr = alloc8( 'DOUBLE' , DOUBLE , NUMROW , NUMCOL , NUMPAG )
if (ptr .eq. NULL) goto 20
c
c Determine where each array is located and print the attributes
c
ptr1 = locate( 'SHORT' , type , nrow , ncol , npag )
c
write( string , 9000 )
if (frmwr( Logfid , 2 , string ) .ne. 2) goto 10
write( string , 9040 ) type,ptr1,nrow,ncol,npag
if (frmwr( Logfid , 5 , string ) .ne. 5) goto 10
c
ptr2 = locate( 'LONG' , type , nrow , ncol , npag )
c
write( string , 9010 )
if (frmwr( Logfid , 2 , string ) .ne. 2) goto 10
write( string , 9040 ) type,ptr1,nrow,ncol,npag
if (frmwr( Logfid , 5 , string ) .ne. 5) goto 10
c
ptr3 = locate( 'FLOAT' , type , nrow , ncol , npag )
c
write( string , 9020 )
if (frmwr( Logfid , 2 , string ) .ne. 2) goto 10
write( string , 9040 ) type,ptr1,nrow,ncol,npag
if (frmwr( Logfid , 5 , string ) .ne. 5) goto 10
c
ptr4 = locate( 'DOUBLE' , type , nrow , ncol , npag )
c
write( string , 9030 )
if (frmwr( Logfid , 2 , string ) .ne. 2) goto 10
write( string , 9040 ) type,ptr1,nrow,ncol,npag
if (frmwr( Logfid , 5 , string ) .ne. 5) goto 10
c
c Pass the data arrays to a subroutine to be zeroed
c
call zero( nrow , ncol , npag ,
& Kmnn ( ptr1), Kmnn ( ptr2),
& Kmnn ( ptr3), Kmnn ( ptr4))
c
c Delete the data arrays in the reverse order that they were
c allocated to prevent shifting of data in program memory
c
10 ptr4 = delete('DOUBLE')

```

```

20 ptr3 = delete('FLOAT ')
30 ptr2 = delete('LONG ')
40 ptr1 = delete('SHORT ')

c
c   All done; close error output file before exit
c
50 call filcls( logfid )
c
call exit( 0 )

c
999 stop

9000 format(/'Data Array SHORT:')
9010 format(/'Data Array LONG:')
9020 format(/'Data Array FLOAT:')
9030 format(/'Data Array DOUBLE:')
9040 format('Data type = ',i1/'Starting Address = ',i5/
&           'Number of Rows = ',i5/'Number of Columns = ',i5/
&           'Number of Pages = ',i5)
      end

c
c
      subroutine zero ( ncol , nrow , npag , short , long ,
&                   float , double )

c
c This subroutine file arrays of all available data types and
c identical dimensions with zeros. Those arguments that are
c array dimensions are defined first so that the compiler will
c know they are integers; the compiler options are such that
c all variables must be defined.

c
c Declare Argument Types:
c
      integer*4      nrow , ncol , npag
      integer*4      long ( nrow , ncol , npag)
      integer*2      short ( nrow , ncol , npag)
      real*4        float ( nrow , ncol , npag)
      real*8        double ( nrow , ncol , npag)

c
c Declare Local Variable Types:
c
      integer*4      i , j , k

c
c Fill all arrays with zeroes; note the order in which the array
c elements are accessed
c
      do 30 k = 1, npag
        do 20 j = 1, ncol
          do 10 i = 1, nrow
            short(i,j,k) = 0
            long(i,j,k)  = 0
            float(i,j,k) = 0.0
            double(i,j,k) = 0.0d0
10      continue
20      continue
30      continue

c
      return
end

```

21.4 Finite Element Attribute Tables

For each element type or constitutive model there are a number of integer values that define their attributes, such as the number of nodes per element or the number of state variables per constitutive model. These attributes are easily defined in a tabular format using one-, two-, and three-dimensional arrays. The finite element attribute tables used in MERLIN include:

- Element type attributes,
- Element class attributes,
- Element surface definitions,
- Element nodal DOF's,
- Element integration rules,
- Surface integration rules,
- Element constitutive model availability,
- Constitutive model stress components, and
- Constitutive model state variables.

All of these tables are stored in COMMON BLOCK **/fematr/**, which is located in the include file **fematr.cmn** along with the parameters that define the dimensions of these tables. The contents of each of these attribute tables are discussed in detail, in separate subsections, in the remainder of this section.

21.4.1 Element Type Attribute Table

An element type is defined by its configuration (i.e., the number of nodes and the shape of the element) and various formulation parameters. The full list of attributes for an element type is as follows:

1. Element class identifier,
2. Isoparametric formulation flag,
3. Number of element shape functions,
4. Number of coordinates per node,
5. Number of DOF for element,
6. Stress-strain law classification, and
7. Strain-displacement matrix transformation flag.

The array name for this attribute table is **Elmatr** and it is a two-dimensional array. The dimensions of this array are defined by the parameters **NELATR** and **NELTYP**, where **NELATR** is the number of attributes per element type and **NELTYP** is the number of element types in the element library. The number of rows in **Elmatr** is defined by **NELATR** and the number of columns is defined by **NELTYP**.

The element class identifier indicates the basic configuration of the element type as well as information concerning the natural coordinate system in which the element is defined; the list of supported element classes is included in Section 21.4.2. A non-zero value for the isoparametric formulation flag indicates that the element formulation is indeed isoparametric and a value of zero indicates that some other formulation is used. For an isoparametric element, the number of nodes and the number of element shape functions are the same. However, for elements that do not use an isoparametric formulation, those numbers may differ. The number of coordinates per node is the same for all nodes defining an element. An attempt to add an element type to the element library that does not fit this criteria would require the significant modifications be made to the code. The number of DOF's for an element type is the total number of DOF's for all nodes defining the element; DOF's not associated with a node are not supported. However, the nodes defining an element type are not required to have the same number of DOF's. Each element type models only one type of stress-strain idealization. This was done to eliminate IF statements from element matrix formulation utility subroutines, where speed is critical. The stress-strain formulations supported by MERLIN are as follows:

- Truss/spring,
- Plane stress,
- Plane strain,
- Axisymmetric, and
- Three-dimensional continuum.

The strain-displacement matrix formulation flag currently is used only with the high performance 4-node quadrilateral elements, otherwise this value is zero. Perhaps this attribute should be called the element technology or high performance element flag, with the type of element technology technique employed indicated by a unique numeric value.

21.4.2 Element Class Attributes Table

An element class is defined only by its configuration (i.e., the number of nodes and the shape of the element). The full list of attributes for an element type is as follows:

1. Number of nodes defining the element,
2. Number of element surfaces (i.e., edges of 2-D continuum elements and faces of 3-D continuum elements),
3. Column index in the element surface definition table and surface integration rule table where the surface definitions for this element class begin,
4. Number of element natural coordinates, and
5. Natural coordinate system classification.

The array name for this attribute table is **Elmccls** and it is a two-dimensional array. The dimensions of this array are defined by the parameters **NECATR** and **NELCLS**, where **NECATR** is the number of attributes per element class and **NELTYP** is the number of element classes supported by MERLIN. The number of rows in **Elmccls** is defined by **NECATR** and the number of columns is defined by **NELCLS**.

The number of nodes per element class is not necessarily a unique value. For instance, an 8-node quadrilateral and an 8-node brick both are defined by eight nodes, but they are clearly assembled in two distinctly different configurations. The definition of element surfaces, which is dependent on the element topology, for each element class was done to simplify user identification of element surfaces for the application of surface tractions and the definition of discrete cracks and contour paths. The conventions for numbering the nodes defining an element surface are described in Section 21.4.3. The number of natural coordinates used to define an element class depends on the element configuration. The element classes supported in MERLIN are defined in one of the following four natural coordinate systems:

1. One-dimensional,
2. Two-dimensional Cartesian coordinates,
3. Two-dimensional area coordinates, and
4. Three-dimensional Cartesian coordinates.

When adding a new element type to the element library, an element class identifier must be specified. Currently, MERLIN supports 12 element classes for the programmer to choose from:

1. 2-node line,
2. 3-node line,
3. 3-node triangle,
4. 4-node quadrilateral,
5. 6-node triangle,

6. 8-node quadrilateral,
7. 9-node quadrilateral,
8. 4-node 2-D interface,
9. 6-node 2-D interface,
10. 8-node brick,
11. 15-node wedge, and
12. 20-node brick.

Should the desired element configuration not be available in the table of supported element classes, the programmer must add the appropriate element class to the element class attribute table and the corresponding surface definitions (see Section 21.4.3).

21.4.3 Element Surface Definition Table

An element surface is defined by the number of nodes that constitute the surface and the list of nodes ordered in some rational manner. In MERLIN, the nodes that constitute a surface for a two-dimensional element are numbered in a counter-clockwise direction around the element boundary; in a sense, a surface definition in two-dimensions is nothing more than a simple subset of the connectivity. However, for three-dimensional elements, the surfaces are numbered counterclockwise as viewed outside of the element in the direction normal to the element surface.

The array name for this attribute table is **Elmsrf** and it is a two-dimensional array. The dimensions of this array are defined by the parameters **NDPSRF** and **NUMSRF**, where **NDPSRF** is the maximum number of nodes defining an element surface and **NUMSRF** is the number of element surface definitions. The value is the summation of the number of element surfaces for all element classes. The number of rows in **Elmsrf** is defined by **NDPSRF** and the number of columns is defined by **NUMSRF**. The range of indices for the rows in **Elmsrf** has been modified such that the indices begin at 0 and end at **NDPSRF**, with the number of nodes defining a surface being stored in row 0 and the nodes defining the surface stored in rows 1 through **NDPSRF**. This numbering scheme allows for direct indexing of the element surface nodes.

21.4.4 Element Nodal DOF Table

This will list the element nodal dof.

21.4.5 Element Integration Rules

This will list the element integration rules.

21.4.6 Surface Integration Rules

Not all operations within the finite element method requiring integration are over the volume of an element. Occasionally, it is necessary to integrate quantities over element surfaces, as is the case with applied surface tractions. Therefore, a table of integration rules for element surfaces is required. The attributes defining the integration rules for the element surfaces are stored in the two-dimensional array **Srfatt**. The dimensions of this array are defined by the parameters **NSRINT** and **NUMSRF**, where **NSRINT** is the number of integration rule attributes per surface and **NUMSRF** is the number of element surface definitions (see Section 21.4.3). The number of rows in **Srfatt** is defined by **NSRINT** and the number of columns is defined by **NUMSRF**.

Each column of **Srfatt** contains the integration rule for a surface associated with a particular element class. A one to one correspondence exists between the columns in arrays **Srfatt** and **Elmsrf** (i.e., the integration rule defined in column 10 of **Srfatt** corresponds to the element surface defined column 10 of **Elmsrf**). Therefore, the index into **Elmsrf** defined in **Elmccls** also is an index into **Srfatt**. The list of attributes for each element surface integration rule is as follows:

1. Number of the element class for which shape functions will be used for the surface integration. If this number is zero, surface integration is disabled for this surface. For surface tractions, an attempt to apply tractions on such a surface will result in a run-time error.
2. Number of shape functions. This should correspond to the number of nodes defining the element.

3. Number corresponding to the numerical integration scheme (see Section 21.4.5 for the list of numerical integration schemes).
4. Number of integration points. This number should be sufficient to allow for accurate integration of quantities that vary on the order of the shape functions for the surface.

21.4.7 Constitutive Model State Variable Table

Element	Description
1	Damage number due to uplift pressure, $\langle 0, 1 \rangle$.
2	Crack opening before unloading, non-zero only after unloading.
3	Normal cohesive stress before unloading non-zero only after unloading.

Table 21.2: State Variables for FCM Model

Element	Description
---------	-------------

Table 21.3: State Variables for ICM Model

21.5 Element Information Tables

21.5.1 Interface Element Information Table

Table Element	Description
intelm(1,iintel)	Interface element ID.
intelm(2,iintel)	Constraint status 0 = no constraint 1 = sliding constraint 2 = opening constraint 3 = both sliding and opening constraint

Table 21.4: Interface Element Information Table (**INTELIM**)

21.6 Nodal Tables

21.6.1 Nodal Attribute Table

21.7 Crack Information Tables

In MERLIN each crack is defined by its surfaces and crack tips/fronts.

Table Element	Description
nodatr(1, node)	New node number after renumbering
nodatr(2, node)	Number of degrees of freedom at the node
nodatr(3, node)	Element group id associated to the node
nodatr(4, node)	Number of elements using this node
nodatr(5, node)	Pointer to array noduse where the list of elements using this node begins.
nodatr(6, node)	Projection flag. Equal to one if strain projection is to be performed for this node

Table 21.5: Nodal Attribute Table (**nodatr**)

Table Element	Description
id(1, node)	Equation number for the first degree of freedom of this node
id(2, node)	Equation number for the second degree of freedom of this node
...
id(Mndof, node)	Equation number for the last degree of freedom of this node

Table 21.6: Nodal ID Table (**id**)

Table Element	Description
cfatr(1,ifront)	Number of crack front nodes.
cfatr(2,ifront)	Index to array cflist where the crack front list for this crack starts.

Table 21.7: Crack Front Attribute Table (**cfatr**)

Table Element	Description
cflist(offset + 0)	ID of first crack front node.
cflist(offset + 1)	ID of second crack front node.
...	...
cflist(offset + lstlen)	ID of last crack front node. lstlen is equal to cfatr(1,ifront).

Table 21.8: Crack Front List (**cflist**)

Table Element	Description
csatr(1,icrack)	Number of crack surface pairs.
csatr(2,icrack)	Index to array csinfo where the crack surface informations for this crack starts.
csatr(3,icrack)	Crack uplift pressure status flag. -1 = decrease in uplift pressure 0 = no uplift 1 = increase in uplift pressure 2 = no change in pressure
csatr(4,icrack)	Crack traction status flag 0 = no tractions applied 1 = tractions applied
csatr(5,icrack)	Crack hydrostatic load status flag 0 = no hydrostatic load is applied 1 = hydrostatic load is applied

Table 21.9: Crack Surface Attribute Table (**csatr**)

Table Element	Description
Upper Crack Surface	
csinfo(1,1,offset)	Element ID on the crack surface.
csinfo(2,1,offset)	Element surface number on the crack surface.
csinfo(3,1,offset)	Number of nodes on the crack surface.
csinfo(4,1,offset)	Interface element ID connected to this surface.
csinfo(5,1,offset)	Interface surface number connected to this surface.
csinfo(6,1,offset)	ID of the traction applied on this crack surface.
csinfo(7,1,offset)	Traction ID is an index into the array trcinf . ID of the hydrostatic load applied on this crack surface. Hydro ID is an index into the array hydinf .
csinfo(8,1,offset)	First node ID on the upper surface.
...	...
csinfo([9,10,11,15],1,offset)	Last node ID on the upper surface. Depending on the element type there can be up to 8 nodes on an element surface.
Lower Crack Surface	
csinfo(1,2,offset)	Element ID on the crack surfac.
csinfo(2,2,offset)	Element surface number on the crack surface.
csinfo(3,2,offset)	Number of nodes on the crack surface.
csinfo(4,2,offset)	Interface element ID connected to this surface.
csinfo(5,2,offset)	Interface surface number connected to this surface.
csinfo(6,2,offset)	ID of the traction applied on this crack surface.
csinfo(7,2,offset)	Traction ID is an index into the array trcinf . ID of the hydrostatic load applied on this crack surface. Hydro ID is an index into the array hydinf .
csinfo(8,2,offset)	First node ID on the upper surface.
...	...
csinfo([9,10,11,15],2,offset)	Last node ID on the upper surface. Depending on the element type there can be up to 8 nodes on an element surface.

Table 21.10: Crack Surface Information (**csinfo**)

Table Element	Description
fnclim(1)	COD after which the uplift becomes non-zero
fnclim(2)	COD limit for second uplift pressure function
fnclim(3)	COD limit for third uplift pressure function (Equal to COD_{w0} for quadratic relationship)
fnclim(4)	COD limit for fourth uplift pressure function (Equal to COD_{w0} for cubic relationship)

Table 21.11: Uplift function limits (**fnclim**)**21.7.1 Crack Front Attributes Table****21.7.2 Crack Front List****21.7.3 Crack Surface Attribute Table****21.7.4 Crack Surface Information****21.8 Uplift Information Arrays**

Uplift pressure function coefficients are stored in a local array **fncoef** (Table 21.12), and are used to define the pressure-cod relationship:

$$\begin{aligned} \frac{p_w}{p_{w0}} &= a + b * \frac{COD}{COD_{w0}} + c * \left(\frac{COD}{COD_{w0}}\right)^2 && \text{For quadratic relationship} \\ \frac{p_w}{p_{w0}} &= a + b * \frac{COD}{COD_{w0}} + c * \left(\frac{COD}{COD_{w0}}\right)^2 + d * \left(\frac{COD}{COD_{w0}}\right)^3 && \text{For quadratic relationship} \end{aligned} \quad (21.1)$$

Table Element	Description
First pressure function	
fncoef(1,1)	Coefficient a for first pressure function
fncoef(2,1)	Coefficient b for first pressure function
fncoef(3,1)	Coefficient c for first pressure function
fncoef(4,1)	Coefficient d for first pressure function
Second pressure function	
fncoef(1,2)	Coefficient a for second pressure function
fncoef(2,2)	Coefficient b for second pressure function
fncoef(3,2)	Coefficient c for second pressure function
fncoef(4,2)	Coefficient d for second pressure function

Table 21.12: Uplift function coefficient (**fncoef**)

Bibliography

- Acres: 2004. <http://www.hatchacres.com/Company/Services/ServHydroAAR/indent.htm>.
- Ahmadi, M., Izadinia, M. and Bachmann, H.: 2001, A discrete crack joint model for nonlinear dynamic analysis of concrete arch dam, *Computers & Structures* **79**, 403–420.
- Aliabadi, M. and Rooke, D.: 1991, *Numerical Fracture Mechanics*, Kluwer Academic Publishers.
- Alvaredo, A. and Wittman, F.: 1992, Crack formation due to hygral gradients, *Third International Workshop on Behaviour of Concrete Elements Under Thermal and Hygral Gradients*, Weimar, Germany.
- Anderson, T.: 1995, *Fracture Mechanics, Fundamentals and Applications*, CRC Press.
- Army Corps of Engineers: 1990, Special design provisions for massive concrete structures, *ETL 1110-2-324*, Department of the Army, US Army Corps of Engineers, Washington, D.C.
- Aslam, M., Wilson, E., Button, M. and Ahlgren, C.: 2002, Earthquake analysis of radial gates/dam including fluid-structure interaction, *Proceedings, Third U.S.-Japan Workshop on Advanced Research on Earthquake Engineering for Dams*, San Diego, CA.
- Atkinson, C. and Bastero, J.: 1991, On the use of betti's reciprocal theorem for computing the coefficients of stress singularities in anisotropic media, *International Journal of Engineering Science* **29**(6), 727–741.
- Atkinson, C., Bastero, J. and Martinez-Esnaola, J.: 1988, Stress analysis in sharp angular notches using auxiliary fields, *Engineering Fracture Mechanics* **31**(4), 637–646.
- Atkinson, C., Bastero, J. and Miranda, I.: 1986, Path-independent integrals in fracture dynamics using auxiliary fields, *Engineering Fracture Mechanics* **25**(1), 53–62.
- Atkinson, C. and Craster, R.: 1992, The application of invariant integrals in diffusive elastic solids, *Philosophical Transactions of the Royal Society of London, Series A* **399**(1653), 231–263.
- Babuška, I.: 1971, Error-bounds for finite element methods, *Numer. Math.* **20**(3), 179–192.
- Babuška, I.: 1973, The finite element method with lagrange multipliers, *Numer. Math.* **20**(3), 179–182.
- Babuska, I. and Miller, A.: 1984, The post-processing approach to finite element method– part ii, *Int. J. Num. Meth. Engrg.* **20**, 1111–1129.
- Bangert, F., D., K. and Meschken, G.: 2004, Chemo-hygro-mechanical modelling and numerical simulation of concrete deterioration caused by alkalisilica reaction, *Int. J. Numer. Anal. Meth. Geomech.* **28**, 689–714.
- Barsoum, R.: 1974, Application of quadratic isoparametric finite elements in linear fracture mechanics, *Int. J. Fract.* **10**, 603–605.
- Bastero, J., Atkinson, C. and Martinez-Esnaola, J.: 1989, Use of path-independent integral a for mixed-mode fast crack propagation in uncoupled thermoelastic media, *Engineering Fracture Mechanics* **34**(2), 325–335.
- Bažant, Z. and Becq-Giraudon, E.: 2001, Estimation of fracture energy from basic characteristics of concrete, in R. de Borst, J. Mazars, G. Pijaudier-Cabot and J. van Mier (eds), *Fracture Mechanics of Concrete Structures (Proc., FraMCoS-4 Int. Conf., Paris)*, Balkema, pp. 491–495.
- Bažant, Z. and Oh, B.-H.: 1983, Crack band theory for fracture of concrete, *Materials and Structures* **16**, 155–177.
- Bažant, Z.P.: 1984, Size effect in blunt fracture: Concrete, rock, metal, *J. of Engineering Mechanics, ASCE* **110**(4), 518–535.
- Bažant, Z.P. and Oh, B.H.: 1983, Crack band theory for fracture of concrete, *Materials and Structures* **93**, 155–177.
- Bažant, Z.P. and Steffens, A.: 2000, Mathematical model for kinetics of alkali-silica reaction in concrete, *Cement and Concrete Research* **30**, 419–428.

- Bažant, Z.P. and Zi, G. and Meyer, C.: 2000, Fracture mechanics of aar in concretes with waste glass particles of different sizes, *J. of Engineering Mechanics, ASCE* **126**(3), 226–232.
- Bažant, Z. P. and Gambarova, P.: 1980, Rough cracks in reinforced concrete, *Journal of the Structural Division ASCE* **106**(ST4), 819–842.
- Belytschko, T., Liu, W. and Moran, B.: 2000, *Nonlinear Finite Elements for Continua and Structures*, Wiley & Sons, Ltd.
- Belytschko, T., Ong, J., Liu, W. and Kennedy, J. M.: 1984, Hourglass control in linear and nonlinear problems, *Computer methods in applied mechanics and engineering*.
- Bocca, P., Carpinteri, A. and Valente, S.: 1990, Size effects in the mixed mode crack propagation: Softening and snap-back analysis, *Engineering Fracture Mechanics* **35**(1–3), 159–170.
- Bon, E., Chille, F., Masarati, P. and Massaro, C.: 2001, Analysis of the effects induce by alkali-aggregate reaction (aar) on the structural behavior of Pian Telessio dam, *Sixth International Benchmark Workshop on Numerical Analysis of Dams (Theme A)*, Salzburg, Austria.
- Bournazel, J. and Moranville, M.: 1997, Durability of concrete: The crossroad between chemistry and mechanics, *Cement and Concrete Research* **27**(10), 1543–1552.
- Brezzi, F.: 1974, On the existence uniqueness and approximation of saddle-point problems arising from lagrangian multipliers, *RAIRO 8-R2* pp. 129–151.
- Brühwiler, E.: 1988, *Fracture Mechanics of Dam Concrete Subjected to Quasi-Static and Seismic Loading Conditions*, Doctoral Thesis No 739, Swiss Federal Institute of Technology, Lausanne. (in German).
- Brühwiler, E. and Saouma, V.E.: 1990, Fracture testing of rock by the wedge splitting test, in W. Hustrulid and G. Johnson (eds), *Proceedings of the 31st US Symposium on Rock Mechanics*, Balkema, Golden, CO, pp. 287–294.
- Brühwiler, E. and Saouma, V.E.: 1991, Water fracture interaction in cracked concrete dams, in V. Saouma, R. Dungar and D. Morris (eds), *Proceedings of the Int. Conference on Dam Fracture*, Electric Power Research Institute, GS-7491, Palo-Alto, Ca, pp. 551–567.
- Brühwiler, E. and Wittmann, F.: 1990, Failure of Dam Concrete Subjected to Seismic Loading Conditions, *Engineering Fracture Mechanics* **35**(3), 565–572.
- Candappa, D., Sanjayan, J. and Setunge, S.: 2000, Complete triaxial stress-strain curves of high-strength concrete, *Cement and Concrete Research* **13**(3), 209–215.
- Capra, B. and Bournazel, J.: 1998, Modeling of induced mechanical effects of alkali-aggregate reactions, *Cement and Concrete Research* **28**(2), 251–260.
- Capra, B. and Sellier, A.: 2003, Orthotropic modeling of alkali-aggregate reaction in concrete structures: Numerical simulations, *Mechanics of Materials* **35**, 817–830.
- Carol, I. and Bažant, Z.P. and Prat, P.C.: 1992, Microplane type constitutive models for distributed damage and localized cracking in concrete structures, *Proc. Fracture Mechanics of Concrete Structures*, Elsevier, Breckenridge, CO, pp. 299–304.
- Carol, I., Prat, P. C. and López, C. M.: 1997, Normal/shear cracking model: Application to discrete crack analysis, *J. Engng. Mech. Div., ASCE* **123**(8), 765–773.
- CEB: 1983, Concrete under multiaxial states of stress constitutive equations for practical design, *Technical Report Bulletin d'Information 156*, Comité Euro-International du Béton.
- Cedolin, L., Dei Poli, S. and Iori, I.: 1987, Tensile behavior of concrete, *J. of Engineering Mechanics, ASCE* **113**(3).
- Červenka, V., Keating, S. and Felippa, C.: 1993, A comparison of strain recovery techniques for the mixed iterative method, *Communications in Applied Numerical Methods* **9**, 925–932.
- Červenka, V. and Jendele, L. and Červenka, Jan: 2002, *ATENA Program Documentation; Part 1 Theory*, Červenka Consulting, Prague, Czech Republic.

- Charlwood, R. G., Solymar, S. V. and Curtis, D. D.: 1992, A review of alkali aggregate reactions in hydroelectric plants and dams, in CEA and CANCOLD (eds), *Proceedings of the International Conference of Alkali-Aggregate Reactions in Hydroelectric Plants and Dams*, Fredericton, Canada, pp. 1–29.
- Cocchetti, G., Maier, G. and Shen, X.: 2002, Piecewise linear models for interfaces and mixed mode cohesive cracks, *Computer Modelling in Engineering & Sciences* **3**, 279–298.
- Com: 1990, *CEB-FIP Model Code 1990*. First Draft.
- Corporation, S. . W. E.: 1992, Uplift pressures, shear strengths, and tensile strengths for stability analysis of concrete gravity dams, *Technical Report TR 100345*, Electric Power Research Institute.
- Crisfield, M.: 1991, *Non-linear Finite Element Analysis of Solids and Structures, Vol 1*, John Wiley & Sons.
- Crisfield, M.A.: 1979, A faster modified newton-raphson iteration, *Computer Methods in Applied Mechanics and Engineering* **20**(3), 267–278.
- Crisfield, M.A.: 1981, A fast incremental/iterative solution procedure that handles ‘snap through’, *Computers and Structures* **13**(1-3), 55–62.
- Dahlblom, O. and Ottosen, N. S.: 1990, Smoother crack analysis using a generalized fictitious crack model, *Journal of Engineering Mechanics* **116**(1), 55–76.
- Davidon, W.C.: 1968, Variance algorithm for minimisation, *Computer J.* **10**, 406–410.
- de Borst, R.: 1986, *Nonlinear Analysis of Frictional Materials*, PhD thesis, Delft University of Technology, Delft.
- deLorenzi, H.G.: 1985, Energy release rate calculations by the finite element method, *Engineering Fracture Mechanics* **21**, 103–110.
- Divoux, P., Boulon, M. and Bourdarot, E.: 1997, A mechanical constitutive model for rock and concrete joints under cyclic loading, in Rossmanith (ed.), *Damage and Failure of Interfaces*, Rotterdam, Balkema, pp. 443–450.
- Divoux, P., Bourdarot, E. and Boulon, M.: 1997, Use of joint elements in the behaviour analysis of arch dams, in Pietruszczak and Pande (eds), *Numerical Models in Geomechanics*, Rotterdam, Balkema, pp. 569–574.
- Duda, H.: 1990, Stress-crack-opening relation and size effect in high-strength concrete, *Darmastad Concrete* **5**, 35–46.
- Farage, M., Alves, J. and Fairbairn, E.: 2004, Macroscopic model of concrete subjected to alkaliaggregate reaction, *Cement and Concrete Research* **34**, 495–505.
- Feenstra, P., de Borst, R. and Rots, J.: 1991, Numerical study on crack dilatancy. i: Models and stability analysis. ii: Applications, *J. Eng. Mech.* **117**(4), 733–769.
- Felippa, C.: 2000, Lecture notes in advanced finite element methods (asen 5367), *Technical report*, Dept. of Aerospace Engineering, University of Colorado, Boulder. <http://caswww.colorado.edu/courses.d/AFEM.d/Home.html>.
- Fenves, G. L., Mojtabaei, S. and Reimer, R. B.: 1992, Parameter study of joint opening effects on earthquake response of arch dams, *Technical Report 92/05*, Earthquake Engineering Research Center - University of California at Berkeley.
- FERC: 1999, *Engineering Guidelines for the Evaluation of Hydropower Projects*, Federal Energy Regulatory Commission – Division of Dam Safety and Inspection, Washington, USA. www.ferc.gov/industries/hydropower/safety/eng-guide.asp.
- Fox, D. J., Kana, D. D. and Hsiung, S. M.: 1998, Influence of interface roughness on dynamic shear behavior in jointed rock, *Int. J. Rock Mech. Min. Sci.* **35**(7), 923–940.
- Fronteddu, L., Léger, P. and Tinawi, R.: 1998, Static and dynamic behaviour of concrete lift joint interfaces, *Journal of Structural Engineering*, ASCE **124**(12), 1418–1430.
- Furusawa, Y., Ohga, H. and Uomoto, T.: 1994, An analytical study concerning prediction of concrete expansion due to alkali-silica reaction, in Malhotra (ed.), *Proc. of 3rd Int. Conf. on Durability of Concrete*, Nice, France, pp. 757–780. SP 145-40.
- Gallagher, R.: 1975, *Finite Element Analysis Fundamentals*, Prentice Hall, Englewood Cliffs, N.J.

- Gerstle, W. and Xie, M.: 1992, Fem modeling of fictitious propagation in concrete, *Journal of Engineering Mechanics* **118**(2), 416–434.
- Giambanco, G. and Di Gati, L.: 1997, A cohesive interface model for the structural mechanics of block masonry, *Mechanics Research Communications* **24**(5), 503–512.
- Gilks, P. and Curtis, D.: 2003, Dealing with the effects of aar on the water retaining structures at mactaquac gs, *Proceedings of the 21st Congress on Large Dams*, Montreal, Canada, pp. 681–703.
- Giuriani, E. and Rosati, G. P.: 1986, Behaviour of concrete elements under tension after cracking, *Studi e Ricerche, Corso di Perfezionamento per le Costruzioni in Cemento Armato* **8**, 65–82. (in Italian).
- Goodman, R.: 1980, *Introduction to Rock Mechanics*, John Wiley & Sons.
- Goodman, R.E. and Taylor, R.C. and Brekke, T.C.: 1968, A Model for the Mechanics of Jointed Rocks, *J. of the Soil Mechanics and Foundations Division ASCE* **94**, 637–659.
- Gopalaratnam, V. S. and Shah, P. S.: 1985, Softening response of plain concrete in direct tension, *ACI Journal* **82**, 310–323.
- Gopalaratnam, V. S. and Ye, B. S.: 1991, Numerical characterization of the nonlinear fracture process in concrete, *Engineering Fracture Mechanics* **40**(6), 991–1006.
- Grasselli, G., Wirth, J. and Egger, P.: 2002, Quantitative three-dimensional description of a rough surface and parameter evolution with shearing, *Int. J. Rock Mech. Min. Sci.* **39**, 789–800.
- Grassl, P., Lungren, K. and Gylltoft, K.: 2002, Concrete in compression : A plasticity theory with a novel hardening law, *International Journal of Solids and Structures* **39**, 5205–5223.
- Hall, J. F.: 1998, Efficient non-linear seismic analysis of arch dams, *Earthquake Engng. Struct. Dyn.* **27**, 1425–1444.
- Hansen, E. and Saouma, V.: 2003, 3 d nonlinear finite element/fracture mechanics analysis of a pressurized nuclear reactor container ring, *Nuclear Engineering and Design* **225**, 1–10.
- Haser, N. and Sullivan, J.: 1991, *Real Analysis*, Dover Publications, New-York.
- Hellen, T. K. and Blackburn, W. S.: 1975, The calculation of stress intensity factors for combined tensile and shear loading, *International Journal of Fracture* **11**(4), 605–617.
- Henshell, R. D. and Shaw, K. G.: 1975, Crack tip elements are unnecessary, *International Journal of Numerical Methods in Engineering* **9**(3), 495–509.
- Hilber, H., Hughes, T. and Taylor, R.: 1977, Improved numerical dissipation for time integration algorithms in structural dynamics, *Earthquake Engineering and Structural Dynamics* **5**, 283–292.
- Hillerborg, A., Modéer, M. and Petersson, P.: 1976, Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements, *Cement and Concrete Research* **6**(6), 773–782.
- Hohberg, J.-M.: 1992, *A joint element for the nonlinear dynamic analysis of arch dams*, PhD thesis, Institute of Structural Engineering, ETH, Zurich.
- Homand, F., Belem, T. and Souley, M.: 2001, Friction and degradation of rock joint surfaces under shear loads, *Int. J. Numer. Anal. Meth. Geomech.* **25**, 973–999.
- Hong, C.-C. and Stern, M.: 1978, The computation of stress intensity factors in dissimilar materials, *Journal of Elasticity* **8**(1), 21–34.
- Hordijk, D.: 1991, *Local approach to Fatigue of Concrete*, PhD thesis, Delft University of Technology.
- Hordijk, D., Reinhardt, H. and Cornelissen, H.: 1989, Fracture mechanics parameters of concrete from uniaxial tensile tests as influenced by specimen length, in S. Shah and S. Swartz (eds), *RILEM-SEM Int. Conference on Fracture Mechanics of Rock and Concrete*, Springer-Verlag, New-York.
- Huang, M. and Pietruszczak, S.: 1999, Alkali-silica reaction: modeling of thermo-mechanical effects, *Journal of Engineering Mechanics, ASCE* **125**(4), 476–487.

- Huang, X., Haimson, B. C., Plesha, M. E. and Qiu, X.: 1993, An investigation of the mechanics of rock joints.—part I. laboratory investigation, *Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.* **30**(3), 257–269.
- Hughes, T.: 1983, Analysis of transient algorithms with particular reference to stability behavior, in T. Belytschko and T. Hughes (eds), *Computational Methods for Transient Analysis*, Elsevier Science Publishers B.V., pp. 67–155.
- Hutson, R. and Dowding, C.: 1990, Joint asperity degradation during cyclic shear, *Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.* **27**(2), 109–119.
- IALAD Network for the Integrity Assessment of Large Dams*: 2004, internet home page. <http://nw-ialad.uibk.ac.at>.
- Iesulauro, E., Ingraffea, A. R., Arwade, S. and Wawrzynek, P. A.: 2002, Simulation of grain boundary decohesion and crack initiation in aluminum microstructure models, in W. Reuter and R. Piascik (eds), *Fatigue and Fracture Mechanics: 33rd Volume, ASTM STP 1417*.
- Ingraffea, A. and Gerstle, W.: 1984, Non-linear fracture models for discrete crack propagation, in S. Shah (ed.), *Proceedings of the NATO Advanced Workshop on Application of Fracture Mechanics to Cementitious Composites*, Martinus Nijhoff, Hingham, Mass, pp. 171–209.
- Ingraffea, A. and Manu, C.: 1980, Stress intensity factor computation in three dimension with quarter-point elements, *Int. J. num. Meth. Engng.* **15**(10), 1427–1445.
- Jabarooti, M. and Golabtoonchi, I.: 2003, Alkali-aggregate reactivity in south-east of iran, *Proceedings of the 21st Congress on Large Dams*, Montreal, Canada, pp. 53–62.
- Jafari, M., Hosseini, K. A., Pellet, F., Boulon, M. and Buzzi, O.: 2003, Evaluation of shear strength of rock joints subjected to cyclic loading, *Soil Dynamics and Earthquake Engineering* **23**, 619–630.
- Jeang, F. L. and Hawkins, N. M.: 1985, Nonlinear analysis of concrete fracture, *Structures and mechanics report*, Department of Civil Engineering, University of Washington, Seattle, WA.
- Jirasek, M. and Bažant, Z.: 2001, *Inelastic Analysis of Structures*, John Wiley, Chichester.
- Karlsson, A. and Bäcklund, J.: 1978, J-integral at loaded crack surfaces, *International Journal of Fracture* **14**(6), R311–R314.
- Kawamoto, T. and Ishizuka, Y.: 1981, An analysis of excavation in strain-softening rock mass, *Journal of Geotechnical Engineering, (JSCE)* **312**(8), 107–118. In Japanese.
- Knowles, J. and Sternberg, E.: 1972, On a class of conservation laws in linearised and finite elastostatics, *Arch. Rat. Mech. Anal.* **44**, 187–211.
- Kolleger, J. and Mehlhorn, G.: 1988, Experimentelle und analytische untersuchungen zur aufstellung eines materialmodels fr gerissene stahbetonscheiben, *Technical report*, Forschungsbericht, Massivbau, Gesamthochschule, Kassel.
- Kolmar, W.: 1986, *Beschreibung der Kraftuebertragung ueber Risse in nichtlinearen Finite-Element-Berechnungen von Stahlbetontragwerken*, PhD thesis, Darmstadt.
- Kotsovos, M. and Newman, J.: 1980, A mathematical description of deformational behavior of concrete under generalized stress beyond ultimate strength, *ACI Journal* **77**(5), 340–346.
- Kreyszig, E.: 1979, *Advanced Engineering Mathematics, 4 th Edition*, John Wiley and Sons.
- Kupfer, B. and Gerstle, K.: 1973, Behavior of concrete under biaxial stresses, *ASCE Journal of the Engineering Mechanics Division* **99**(4), 853–866.
- Kutter, H. K. and Weissbach, G.: 1980, Der einflues von verformungs- und belastungsgeschichte auf den scherwiderstand von gesteinsklufen unter besonderer berucksichtigung der mylonitbildung., *Technical Report Research Project Ku 361/2/4*, DFG. in German.
- Labuz, J., Shah, S. and Dowding, C.: 1985, Experimental analysis of crack propagation in granite, *Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.* **22**(2), 85–98.

- Larive, C.: 1998, *Apports Combinés de l'Experimentation et de la Modélisation à la Compréhension de l'Alcali-Réaction et de ses Effets Mécaniques*, PhD thesis, Thèse de Doctorat, Laboratoire Central des Ponts et Chaussées, Paris.
- Lee, H. S., Park, Y. J., Cho, T. F. and You, K. H.: 2001, Influence of asperity degradation on the mechanical behavior of rough rock joints under cyclic shear loading, *Int. J. Rock Mech. Min. Sci.* **38**(7), 967–980.
- Léger, P., Côte, P. and Tinawi, R.: 1996, Finite element analysis of concrete swelling due to alkali-aggregate reactions in dams, *Computers & Structures* **60**(4), 601–611.
- Lemarchand, E. and Dormieux, L.: 2000, A micromechanical approach to the modeling of swelling due to alkali-silica reaction, *Proceedings of the 14th. Engineering Mechanics Conference*, Austin (Texas), USA.
- Leonhardt, F. and Walther, R.: 1962, Schubversuche an einfeldrigen stahlbetonbalken mit und ohne schubbewehrung zur ermittlung der schubtragfähigkeit und der oberen schubspannungsgrenze, *Deutscher Ausschuss für Stahlbeton* **151**. Shear tests on a single span reinforced concrete beams without shear reinforcement.
- Li, F. Z. and Shih, C. F. and Needleman, A.: 1985, A comparison of methods for calculating energy release rates, *Engineering Fracture Mechanics* **21**(2), 405–421.
- Li, K. and Coussy, O.: 2002, Concrete ASR degradation: from material modeling to structure assessment, *J. Concrete Science and Engineering* **4**, 35–46.
- Li, V. C. and Liang, E.: 1986, Fracture processes in concrete and fiber reinforced cementitious composites, *Journal of Engineering Mechanics, ASCE* **112**(6), 566–586.
- Lotfi, H.: 1992, *Finite Element Analysis of Fracture of Concrete and Masonry Structures*, PhD thesis, University of Colorado, Boulder.
- Lotfi, H. R. and Shing, P. B.: 1994, Interface model applied to fracture of masonry structures, *Journal of Structural Engineering, ASCE* **120**(1), 63–80.
- Lynn, P. and Ingraffea, A.: 1977, Transition elements to be used with quarter-point crack-tip elements, *Int. J. num. Meth. Engng.* **11**, 1031–1036.
- Lysmer, J. and Kuhlemeyer, R.: 1969, Finite element model for infinite media, *ASCE Journal of Engineering Mechanics* **95**(EM 4), 859–877.
- Malla, S. and Wieland, M.: 1999, Analysis of an arch-gravity dam with a horizontal crack, *Computers & Structures* **72**(1), 267–278.
- Menétrey, P. and Willam, K.: 1995, Triaxial failure criterion for concrete and its generalization, *ACI Structural Journal* **92**(3), 311–318.
- Mindess, S. and Young, F.: 1981, *Concrete*, Prentice-Hall, Inc.
- Misra, A.: 2002, Effect of asperity damage on shear behavior of single fracture, *Engineering Fracture Mechanics* **69**(17), 1997–2014.
- Miura, F. and Toki, K.: 1987, Estimation of natural frequency and damping factor for dynamic soil structure interaction systems, *Proceedings of the 3rd Int. Conf. on Soil Dynamics and Earthquake Engineering*, pp. 73–87.
- Mróz, Z. and Giambanco, G.: 1996, An interface model for analysis of deformation behaviour of discontinuities, *Int. J. Numer. Anal. Meth. Geomech.* **20**(1), 1–33.
- Multon, S.: 2004, *Evaluation expérimentale et théorique des effets mécaniques de l'alcali-réaction sur des structures modèles*, PhD thesis, Université de Marne la Vallée, France. réalisée en partenariat LCPC-EDF. Reprise dans la collection études et recherches des Laboratoires des Ponts et Chausses, série Ouvrages d'Art, OA46, 424 pages, 182 réf., résumé anglais, LCPC, octobre 2004.
- Multon, S., Leclainche, G., Bourdarot, E. and Toutlemonde, F.: 2004, Alkali-silica reaction in specimens under multi-axial mechanical stresses, *Proceedings of CONSEC 4 (Concrete Under Severe Conditions)*, Seoul, pp. 2004–2011.
- Nikishkov, G. P. and Atluri, S. N.: 1987, Calculation of fracture mechanics parameters for an arbitrary three-dimensional crack, by the ‘equivalent domain integral’ method, *International Journal of Numerical Methods in Engineering* **24**(9), 1801–1821.

- Nikishkov, G. and Vainshtok, V.: 1980, Method of virtual crack growth for determining stress intensity factors k_I and k_{II} , *Strength of Materials* **12**, 696–701.
- Ouchterloni, F., Takahashi, H., Matsuki, K. and Hashida, T.: 1991, Experiences from fracture toughness testing of rock according to the isrm suggested methods, *Report ds 1991:2*, Swedish Detonic Research Foundation, SveDeFo, Stockholm, Sweden.
- Ouchterlony, F.: 1988, Suggested methods for determining the fracture toughness of rock, *Int. J. of Rock Mechanics and Mining Sciences* **25**, 71–96.
- Owen, D. and Fawkes, A.: 1983, *Engineering Fracture Mechanics*, Pineridge Press, Swansea, U.K.
- Pandolfi, A.: 2003, Lecture notes on dynamic relaxation, *Technical report*, Caltech.
- Patton, F.: 1966, *Multiple modes of shear failure in rock and related materials*, PhD thesis, University of Illinois.
- Paulay, T. and Loeber, T. J.: 1974, Shear transfer by aggregate interlock, *Shear in Reinforced Concrete*, Vol. 1 of *ACI Special Publication*, American Concrete Institute, Detroit, Michigan, pp. 1–16.
- Petersson, P.: 1981, Crack growth and development of fracture zones in plain concrete and similar materials, *Technical Report TVBM 1006*, Lund Institute of Technology, Lund, Sweden.
- Peyras, L., Royet, P. and Laleu, V.: 2003, Functional modeling of dam performance loss; application to the alkali-aggregate reaction mechanism -chambon dam, *Proceedings of the 21st Congress on Large Dams*, Montreal, Canada, pp. 853–872.
- Plesha, M. E.: 1987, Constitutive models for rock discontinuities with dilatancy and surface degradation, *Int. J. Numer. Anal. Meth. Geomech.* **11**(4), 345–362.
- Portugese National Committee on Large Dams: 2003, Ageing process and rehabilitation of pracaña dam, *Proceedings of the 21st Congress on Large Dams*, Montreal, Canada, pp. 121–138.
- Puntel, E.: 2004, *Experimental and numerical investigation of the monotonic and cyclic behaviour of concrete dam joints*, PhD thesis, Politecnico di Milano.
- Qiu, X., Plesha, M. E., Huang, X. and Haimson, B. C.: 1993, An investigation of the mechanics of rock joints.—part II. analytical investigation, *Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.* **30**(3), 271–287.
- Reich, R.: 1993, *On the Marriage of Mixed Finite Element Methods and Fracture Mechanics: An Application to Concrete Dams*, PhD thesis, University of Colorado, Boulder.
- Rice, J. R.: 1968, A path independent integral and the approximate analysis of strain concentration by notches and cracks, *Journal of Applied Mechanics* **35**(2), 379–386.
- Roelfstra, P. and Sadouki, H.: 1986, Fracture/1: Theory and applications, *Laboratory for Building Materials (LMC)*, EPF Lausanne .
- Rots, J. G. and Blaauwendraad, J.: 1989, Crack models for concrete: Discrete or smeared? fixed, multi-directional or rotating?, *HERON* **34**(1), 1512–1533.
- Saouma, V., Broz, J. and Boggs, H.: 1991, In-situ Field Testing for Fracture Properties of Dam Concrete, *ASCE, Journal of Materials in Civil Engineering* **3**(3), 219–234.
- Saouma, V., Broz, J., Brühwiler, E. and Boggs, H.: 1991, Effect of aggregate and specimen size on fracture properties of dam concrete, *ASCE, Journal of Materials in Civil Engineering* **3**(3), 204–218.
- Saouma, V., Natekar, D. and Sbaizer, O.: 2002, Nonlinear finite element analysis and size effect study of a metal-reinforced ceramics-composite, *Materials Science and Engineering A* **323**, 129–137.
- Saouma, V. and Perotti, L.: 2004a, 2D AAR parametric investigation of a dam, *Technical report*, Report No. 4, Submitted to the Swiss Federal Agency of Water and Geology,, Bienna, Switzerland.
- Saouma, V. and Perotti, L.: 2004b, AAR analysis procedure in kumo/merlin, *Technical report*, Report No. 3, Submitted to the Swiss Federal Agency of Water and Geology,, Bienna, Switzerland.

- Saouma, V. and Perotti, L.: 2004c, Constitutive model for alkali aggregate reaction, *Technical report*, Report No. 2, Submitted to the Swiss Federal Agency of Water and Geology,, Biene, Switzerland.
- Saouma, V. and Schwemmer, D.: 1984, Numerical evaluation of the quarter-point crack tip element, *International Journal of Numerical Methods in Engineering* **20**(9), 1629–1641.
- Saouma, V. and Sikiotis, E.: 1986, Stress intensity factors in anisotropic bodies using singular isoparametric elements, *Engineering Fracture Mechanics* **25**(1), 115–121.
- Saouma, V., Červenka, J. and Reich, R.: 2008, Merlin finite element user's manual, *Technical report*, Electric Power Research Institute (EPRI), Palo Alto; Tokyo Electric Power Service Company (TEPSCO), Tokyo. <http://civil.colorado.edu/~saouma/pdf/users.pdf>.
- Saouma, V. and Xi, Y.: 2004, State of the art survey of alkali-aggregate reactions in dams, *Technical report*, Report No. 1, Submitted to the Swiss Federal Agency of Water and Geology,, Biene, Switzerland.
- Scrivener: 2003. Personal Communication.
- Scrivener: 2005. Personal Communication.
- Shayan, A., Wark, R. and Moulds, A.: 2000, Diagnosis of aar in canning dam, characterization of the affected concrete and rehabilitation of the structure, *Proc. of 11th Int. Conf. on AAR*, Quebec, Canada, pp. 1383–1392.
- Shih, C., de Lorenzi, H. and German, M.: 1976, Crack extension modelling with singular quadratic isoparametric elements, *Int. J. Fract.* **12**, 647–651.
- Snyman, M. F. and Martin, J. B.: 1992, A consistent formulation of a dilatant interface element, *Int. J. Numer. Anal. Meth. Geomech.* **16**(7), 493–527.
- Sokolnikoff, I.: 1956, *Mathematical Theory of Elasticity*, McGraw-Hill, New York, NY.
- Stankowski, T.: 1990, *Numerical Simulation of Progressive Failure in Particle Composites*, PhD thesis, University of Colorado, Boulder.
- Stanton, T.: 1940, Expansion of concrete through reaction between cement and aggregate, *Proceedings of ASCE* **66**, 1781–1811.
- Stern, M.: 1973, A boundary integral representation for stress intensity factors, in T. S. Chang (ed.), *Recent Advances in Engineering Science*, Scientific Publishers, Boston, MA, pp. 125–132. Volume 7.
- Stern, M.: 1979, The numerical calculation of thermally induced stress intensity factors, *Journal of Elasticity* **9**(1), 91–95.
- Stern, M., Becker, E. and Dunham, R.: 1976, A contour integral computation of mixed mode stress intensity factors, *International Journal of Fracture* **12**(3), 359–368.
- Stern, M. and M.L., S.: 1975, The calculation of stress intensity factors in anisotropic materials by a contour integral method, in R. E.F. and S. Benzley (eds), *Computational Fracture Mechanics*, ASME, New York, NY, pp. 161–171.
- Struble, L. and Diamond, S.: 1981a, Swelling properties of synthetic alkali silica gels, *Journal of the American Ceramic Society* **64**(11), 652–655.
- Struble, L. and Diamond, S.: 1981b, Unstable swelling behavior of alkali silica gels, *Cement and Concrete Research* **11**, 611–617.
- Stupkiewicz, S. and Mróz, Z.: 2001, Modelling of friction and dilatancy effects at brittle interfaces for monotonic and cyclic loading, *Journal of Theoretical and Applied Mechanics* **39**, 707–739.
- Sun, Z., Gerrard, C. and Stephansson, O.: 1985, Rock joint compliance tests for compression and shear loads, *Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.* **22**(4), 197–213.
- Suwito, A., Jin, W., Xi, Y. and Meyer, C.: 2002, A mathematical model for the pessimum effect of asr in concrete, *Concrete Science and Engineering*, RILEM . Accepted for Publication.

- Swamy, R. and Al-Asali, M.: 1988, Engineering properties of concrete affected by alkali-silica, *ACI Materials Journal* pp. 367–374.
- Tada, H., Paris, P. and Irwin, G.: 1973, *The Stress Analysis of Cracks Handbook*, Del Research Corporation, Hellertown, PA.
- Tassios, T. P. and Vintzéleou, E. N.: 1987, Concrete-to-concrete friction, *Journal of Structural Engineering*, ASCE **113**(4), 832–849.
- Terzaghi, K. and Peck, R.: 1967, *Soil Mechanics in Engineering Practice, 2nd edition*, John Wiley & Sons, New York, NY.
- Thompson, G., Charlwood, R., Steele, R. and Curtis, D.: 1994, Mactaquac generating station intake and spillway remedial measures, in (ed.), *Proceedings for the Eighteenth International Congress on Large Dams*, Vol. 1, Q-68, R.24, Durban, South-Africa, pp. 347–368.
- Townsend, C.: n.d., Control of cracking in mass concrete structures, *Technical report*, United States Department of the Interior, Bureau of Reclamation, Water Resources Technical Publication, Engineering Monograph No. 34.
- Ulm, F., Coussy, O., Kefei, L. and Larive, C.: 2000, Thermo-chemo-mechanics of asr expansion in concrete structures, *ASCE J. of Engineering Mechanics* **126**(3), 233–242.
- van Mier, J.: 1986, Multi-axial strain softening of concrete; part i: Fracture, *Materials and Structures*, RILEM **19**(111).
- Červenka, J.: 1994, *Discrete Crack Modeling in Concrete Structures*, PhD thesis, University of Colorado, Boulder.
- Červenka, J.: 1994, *Discrete crack modeling in concrete structures*, PhD thesis, University of Colorado, Boulder, Colorado.
- Červenka, J., Kishen, J. M. C. and Saouma, V. E.: 1998, Mixed mode fracture of cementitious bimaterial interfaces; part II: numerical simulation, *Engineering Fracture Mechanics* **60**(1), 95–107.
- V.E., S., Červenka, J., Slowik, V. and Chandra, J.: 1994, Mixed mode fracture of rock-concrete interfaces, in Z. Bažant (ed.), *US-Europe Workshop on Fracture and Damage of Quasi-Brittle Materials: Experiment, Modeling and Computer Analysis*, Prague, Czech Republic.
- Vecchio, F. and Collins, M.: 1986, Modified compression-field theory for reinforced concrete beams subjected to shear, *ACI Journal* **83**(2), 219–231.
- Wagner, C. and Newell, V.: 1995, A review of the history of aar at three of the tva's dam, *Proc. of the Second International Conference on AAR in Hydroelectric Plants and Dams*, Tennessee, pp. 57–66.
- Westergaard, H.: 1933, Water pressures on dams during earthquakes, *Transactions ASCE* **59**(8), 418–433.
- Wiberg, N.-E.: 1974, Matrix structural analysis with mixed variables, *International Journal of Numerical Methods in Engineering* **8**(2), 167–194.
- Wilkins, M.: 1964, Calculation of elasto-plastic flow, *Methods of Computational Physics* **3**, 211–263.
- Wittmann, F., Rokugo, K., Brühwiler, E., Mihashi, H. and Simonin, P.: 1988, Fracture Energy and Strain Softening of Concrete as Determined by Means of Compact Tension Specimens, *Materials and Structures* **21**, 21–32.
- Xue, W. and Atluri, N.: 1985, Existence and stability, and discrete bb and rank conditions for general mixed-hybrid finite elements in elasticity, *Hybrid and Mixed Finite Element Methods* pp. 91–112.
- Zangar, C.: 1953, Hydrodynamic pressures on dams due to horizontal earthquakes, *Proceedings of Society of Experimental Stress Analysis* **10**, 93–102.
- Zienkiewicz, O.: 1967, *The Finite Element in Structural and Continuum Mechanics*, first edn, McGraw-Hill, London.
- Zienkiewicz, O. C. and Cheung, Y. K.: 1964, Buttress dams on complex rock foundations, *Water Power* **16**, 193.
- Zienkiewicz, O. C. and Cheung, Y. K.: 1965, Stresses in buttress dams, *Water Power* **17**, 69.

- Zienkiewicz, O. C., Taylor, R. L. and Nithiarasu, P.: 2005, *The Finite Element Method for Fluid Dynamics*, 6th edn, Elsevier Butterworth-Heinemann.
- Zienkiewicz, O. C., Vilotte, J. P., Toyoshima, S. and Nakazawa, S.: 1985, Iterative method for constrained and mixed approximations. an inexpensive improvement of fem performance, *Computer Methods in Applied Mechanics and Engineering* **51**(1–3), 3–29.