

Gdańsk University of Technology
Civil and Environmental Engineering Department



Jan Kozicki

Application of Discrete Models to Describe the Fracture Process in Brittle Materials

Doctoral thesis

promoter: prof. dr hab. inż. Jacek Tejchman

Gdańsk, March 6, 2007

Streszczenie

Praca doktorska „Zastosowanie modeli dyskretnych do opisu propagacji rys w materiałach kruchych” została podjęta z uwagi na fakt, że bardzo istotne jest poznanie mechanizmu powstawania lokalizacji odkształceń w materiałach kruchych takich jak beton (rysy i strefy ścinania) na poziomie mikroskopowym. Poznanie to daje możliwość bardziej realistycznego wymiarowania elementów konstrukcji betonowych i żelbetowych oraz przewidywania stanów zniszczenia istniejących konstrukcji.

W pracy został opracowany trójwymiarowy dyskretny model opisujący zachowanie się elementów kruchych (beton) poddanych różnym stanom obciążenia. Model uwzględnia wszystkie najważniejsze składniki betonu: kruszywo, cement, oraz połączenia kruszywa z cementem. W opracowanym modelu składniki te są opisane elementami belkowymi z uwzględnieniem ściskania, rozciągania, ścinania, skręcania i zginania. Elementy posiadają trzy parametry opisujące ich własności (sztywność wzdłużna, giętna i skrętna) oraz jeden parametr wytrzymałościowy (maksymalne odkształcenie elementu przy rozciąganiu). Belka jest usuwana z materiału, jeżeli przekroczony zostanie parametr wytrzymałościowy. W drugim modelu, opisanym w załączniku, beton jest modelowany elementami sferycznymi, które są połączone elementami belkowymi. Elementy sferyczne działają wg. zmodyfikowanego prawa Mohra-Coulomba z uwzględnieniem kohezji.

Praca składa się z 6 rozdziałów i 3 załączników. Rozdział drugi prezentuje przegląd modeli dyskretnych wykorzystywanych do modelowania betonu. Przedstawiono opis sferycznej metody dyskretniej DEM, metodę beleczkową, model z nieliniowym interfejsem oraz model z ograniczonym ścinaniem.

Trzeci rozdział prezentuje zastosowanie metody fotogrametrycznej DIC podczas zginania belek z nacięciami do pomiaru odkształceń na powierzchni betonu.

W kolejnym, czwartym rozdziale zamieszczono szczegółowe wzory oraz metodę obliczeniową w beleczkowym modelu dyskretnym. Omówiono szczegółowo aspekty obliczeniowe oraz właściwości nowo powstałego modelu.

Rozdział piąty prezentuje wyniki weryfikacji modelu. Weryfikacja obejmowała sprawdzenie zbieżności wyników numerycznych z wynikami eksperymentalnymi własnymi i z literatury. Przeprowadzono symulacje numeryczne na testy ściskania, rozciągania, zginania oraz łączonego ścinania z rozciąganiem.

Rozdział szósty podsumowuje wyniki osiągnięte w pracy doktorskiej.

Pracę kończą trzy załączniki prezentujące wyniki modelowania przepływu materiałów granulowanych przy zastosowaniu aparatu komórkowego, wyniki symulacji betonu metodą dyskretną DEM oraz opis kwaternionów zastosowanych do modelowania obrotu.

Contents

1	Introduction	1
1.1	Phenomenon and Problem	1
1.2	Aims	2
1.3	Structure of thesis	2
1.4	Novelties	3
2	Literature review	5
2.1	Classical spherical discrete elements model	5
2.2	Classical lattice–beam fracture model	8
2.3	Nonlinear interfacial zone model	12
2.4	Confinement–shear lattice model	16
2.5	Summary	20
3	Experiments	23
3.1	Introduction	23
3.2	Digital Image Correlation technique	24
3.3	Precision of DIC measurements	29
3.4	Experimental setup	34
3.5	Experimental results	34
4	Description of the Lattice Geometrical Model	43
4.1	Notation used	43
4.2	Introduction	44
4.3	Algorithm for the Lattice Geometrical Model in 3D	47
4.3.1	Beam and node elements used in the simplified algorithm	47
4.3.2	Simplified 3D algorithm without bending and torsion	49
4.3.3	Flexural–torsional spring element used in full algorithm	55
4.3.4	Full 3D algorithm	57
4.4	Example calculation	64
4.5	Model discretization	65
4.5.1	Mesh generation	65
4.5.2	Distribution of aggregates	66
4.6	Non–local approach to calculate strain	66
4.7	Other fracture parameters	67
4.8	Stress and force	68
4.9	Model properties	68
4.9.1	Conservation of motion and shape	68

4.9.2	Poisson's ratio	70
4.9.3	Effect of mesh type and density	70
4.9.4	Effect of strain increment	76
4.10	CPU and memory consumption	76
5	Verification of the model	79
5.1	Calibration of model parameters	79
5.2	Uniaxial tension (one and two-phase material)	89
5.3	Uniaxial tension (three-phase material)	91
5.4	Size effect (uniaxial tension, three-phase material)	92
5.5	Uniaxial tension with non-local approach	96
5.6	Uniaxial compression (three-phase material)	96
5.7	Three-point bending (comparison with experiments)	99
5.8	Uniaxial extension in 3D (one-phase material)	99
6	Conclusions and future work	105
	Appendices	107
A	Cellular automaton	107
A.1	Introduction	107
A.2	Description of the simplified model	108
A.3	Simulation results for the simplified model	109
A.4	Description of the improved model	122
A.5	Simulation results for improved model	125
A.6	Conclusions	130
B	Discrete Element Method in the simulation framework	133
B.1	Introduction	133
B.2	The YADE framework: DEM and other models	134
B.3	Software design applied to numerical simulations	138
C	Quaternion rotations	145
C.1	Definition	145
C.2	Properties	146
C.3	Operations concerning rotation	147
C.4	Matrix representation	148
C.5	Conversions	148
C.5.1	Euler axis angle	148
C.5.2	Rotation matrix	149
C.5.3	Euler angle	150
C.6	Efficiency	150
	Bibliography	153
	List of publications	164

List of Figures

2.1	Interaction of spherical discrete elements [54, 71–73]	6
2.2	Normal and shear springs in a spherical element [54]	6
2.3	A view on example DEM specimen [54, 71–73]	7
2.4	Rupture criterion used in the DEM model [54]	7
2.5	DEM results: experiment vs. simulation [38, 49]	7
2.6	A DEM specimen of reinforced concrete subject to bending [71]	8
2.7	Normal, shearing force and momentum on lattice beam [154]	9
2.8	Classical lattice types; [74, 137, 154]	9
2.9	Construction of a random lattice [133]	10
2.10	Approximation of the grading curve [40]	10
2.11	Heterogeneity in the classical lattice–beam model [137, 154]	11
2.12	Result for lattice–beam model: four point shear test [138]	12
2.13	Discretization process for nonlinear interfacial zone model [27]	13
2.14	Ending the discretization process for nonlinear interfacial zone model [27] .	14
2.15	Delaunay triangulation vs. Voronoi tessellation	14
2.16	Nonlinear interfacial zone model: the interface law [27]	14
2.17	Example results for nonlinear interfacial zone model [26]	15
2.18	Discretization process for confinement–shear lattice model [42]	17
2.19	Constitutive law for confinement–shear lattice model [40, 42]	18
2.20	Confinement–shear lattice model: four point shear test; [42]	19
2.21	Levels of material discretization presented by D’Addetta [46]	21
2.22	Various discretization methods of concrete; [15, 25, 27, 38, 42, 74]	21
3.1	Digital Image Correlation: images with analyzed pixel subsets	25
3.2	Comparison between the correlation functions	27
3.3	Correlation plane and its interpolation	28
3.4	Interpolation functions used to achieve a sub–pixel precision	28
3.5	Effect of sub–pixel interpolation on the strain field	29
3.6	Images used in validation experiments	30
3.7	Comparison between two correlation and two interpolation functions	31
3.8	DIC precision against the pixel subset size for zero strain	31
3.9	DIC precision against the pixel subset size for oscillating strain	32
3.10	Strain fluctuations along single line of the image (zero strain)	32
3.11	Strain fluctuations along single line (oscillating strain, small subset)	33
3.12	Strain fluctuations along single line (oscillating strain, large subset)	33
3.13	Geometry of concrete specimens subjected to three-point bending	34
3.14	Grading curve for sand and gravel used for concrete	35

3.15	Instrumentation used in the experimental tests	35
3.16	Concrete beams after failure	36
3.17	Vertical force at mid-span versus beam deflection (sand concrete)	37
3.18	Vertical force at mid-span versus beam deflection (gravel concrete)	37
3.19	Size effect in three-point bending tests	38
3.20	Width of localized zone in medium size beam (sand concrete)	38
3.21	Evolution of width of fracture zone (medium beam)	39
3.22	Evolution of width fracture zone (large beam)	39
3.23	Evolution of strain field during experiments from DIC (sand concrete)	40
3.24	Evolution of strain field during experiments from DIC (gravel concrete)	41
4.1	Meaning of subscripts and superscripts	43
4.2	General scheme to calculate displacements of elements in the lattice	45
4.3	Vectors that describe a beam	48
4.4	Simplified block diagram in Lattice Geometrical Model with rods	50
4.5	Simulation steps 6–7 (simplified algorithm): calculate current geometry of the beam	51
4.6	Simulation steps 1–2–6: set temporary variables to zero	52
4.7	Simulation steps 7–8: check fracture criterion in the current beam	52
4.8	Simulation steps 8–9 (simplified algorithm): increment displacement of nodes connected to current beam	53
4.9	Simulation steps 9–10: update node positions, apply boundary conditions	54
4.10	Vectors and angles that describe beams and a spring	55
4.11	Scenarios of order of nodes between the connected beams	56
4.12	Block diagram of full algorithm of Lattice Geometrical Model	58
4.13	Simulation steps 1a–2 (full algorithm): set temporary variables to zero	59
4.14	Simulation steps 2–3 (full algorithm): calculate number of beam neighbors	59
4.15	Simulation steps 6c–6d (full algorithm): calculate orientation of the beam, taking its torsion into account	59
4.16	Plot showing the two possible functions for projecting twist angle ϑ^i	60
4.17	Simulation steps 3–4 (full algorithm): calculate current geometry of the flexural-torsional spring	61
4.18	Simulation steps 4–5–6 (full algorithm): increment bending and torsional rotation of beams	62
4.19	Simulation steps 8b–8c (full algorithm): increment displacement of nodes connected to current beam due to bending and torsion between beams	63
4.20	A simplified frame composed of 4 beams	64
4.21	Scheme to assume a non-uniform distribution of beams in the lattice	66
4.22	Range of averaging using gaussian distribution; [19]	67
4.23	Conservation of movement	69
4.24	Conservation of rotational movement	69
4.25	Shape conservation in tension	69
4.26	Shape conservation in bending	69
4.27	Uniaxial extension with rough edges: Poisson's ratio	70
4.28	Sample irregular mesh used for examination of Poisson's ratio	70
4.29	Influence of bending stiffness on Poisson's ratio	71
4.30	Meshes used to demonstrate the effect of mesh type and density	72

4.31	Effect of mesh randomness (Delaunay) on stress–strain curve	72
4.32	Effect of mesh randomness (non–Delaunay) on stress–strain curve	73
4.33	Effect of mesh density (non–Delaunay) on stress–strain curve	74
4.34	Effect of number of beams transversing the cross section on normal stress	74
4.35	Effect of strain increment on the stress–strain curve	75
5.1	Effect of p on uniaxial compression with smooth edges	80
5.2	Effect of p on the crack pattern in uniaxial compression with smooth edges	81
5.3	Effect of p on uniaxial compression with extra parameter ϵ_{\max} , with smooth edges	81
5.4	Effect of p on the crack pattern in uniaxial compression with extra parameter ϵ_{\max} , with smooth edges	82
5.5	Effect of p on fracture process in compression with rough edges	83
5.6	Effect of p on uniaxial tension	83
5.7	Effect of p on fracture process in simple shear	84
5.8	Specimen subject to shearing, depending on critical shearing and bending angle	84
5.9	Experimental setup in Nooru–Mohamed test; [42]	85
5.10	Experimental crack pattern in loading condition 4c; [42]	85
5.11	Influence of p on Nooru–Mohamed test	86
5.12	Effect of ϵ_{\min} on the stress–strain curve in uniaxial tension	87
5.13	Change of parameter p in function of element strain $i\epsilon$	88
5.14	Fracture pattern in Nooru–Mohamed test [111] using three–phase material	89
5.15	2D specimen subject to uniaxial extension (one–phase material)	90
5.16	2D specimen subject to uniaxial extension (two–phase material)	90
5.17	2D concrete specimen subject to uniaxial extension (three–phase material), aggregate area percentage 50%, $d_{50} = 3.5$ mm, material parameters as in Table 5.5 (σ_{22} – vertical normal stress, ϵ_{22} – vertical normal strain)	92
5.18	2D concrete specimen subject to uniaxial extension (three–phase material) aggregate area percentage 50%, $d_{50} = 12$ mm, material parameters as in Table 5.5	93
5.19	2D concrete specimen subject to uniaxial extension (three–phase material) aggregate area percentage 25%, $d_{50} = 3.5$ mm, material parameters as in Table 5.5	93
5.20	2D concrete specimen subject to uniaxial extension (three–phase material) aggregate area percentage 25%, $d_{50} = 12$ mm, material parameters as in Table 5.5	94
5.21	The size effect for 2D concrete specimens subject to uniaxial extension	94
5.22	Experimental force–deformation diagram for dog–bone specimens	95
5.23	Fracture in 2 concrete specimens of different size	95
5.24	Effect of the characteristic length l_c on the stress–strain curve	96
5.25	Effect of the characteristic length l_c on the fracture process	97
5.26	Stress–strain curve in uniaxial compression with smooth edges (three–phase material)	98
5.27	Evolution of Poisson’s ratio in uniaxial compression with smooth edges (three–phase material)	98
5.28	Fracture pattern in 10×10 cm ² specimen (uniaxial compression)	99

5.29	Load–displacement curves in three–point bending	100
5.30	Crack pattern in three–point bending, small specimen	100
5.31	Evolution of strain field during three–point bending simulation	101
5.32	Crack pattern in three–point bending, medium specimen	101
5.33	Three–dimensional $5 \times 5 \times 5 \text{ cm}^3$ specimen subject to uniaxial extension . .	102
5.34	Three–dimensional $10 \times 10 \times 10 \text{ cm}^3$ specimen subject to uniaxial extension	103
5.35	The size effect for 3D concrete specimens subject to uniaxial extension . .	104
A.1	Example propagation of cells in the simplified cellular automaton	109
A.2	Migration rules assumed for quadratic cells	110
A.3	The geometry of silo used in tests	111
A.4	Flow patterns observed in model experimental silo tests	112
A.5	Flow patterns during granular flow in a mass flow silo (rule A)	112
A.6	Flow patterns during granular flow in a mass flow silo (rule B)	113
A.7	Flow patterns during granular flow in a mass flow silo (rule C)	113
A.8	Flow patterns during granular flow in a mass flow silo (rule D)	113
A.9	Flow patterns during granular flow in a funnel flow silo (rule A)	114
A.10	Flow patterns during granular flow in a funnel flow silo (rule B)	114
A.11	Flow patterns during granular flow in a funnel flow silo (rule C)	114
A.12	Flow patterns during granular flow in a funnel flow silo (rule D)	114
A.13	Calculated flow rate during granular mass and funnel flow	115
A.14	Flow patterns during granular disturbed mass and funnel flow (rule B) . . .	116
A.15	Flow patterns with controlled outflow velocity (rule B)	116
A.16	Flow during granular mass and funnel flow with rough walls (rule B)	116
A.17	Mass and funnel flow with $4 \times$ the number of cells (rule B)	117
A.18	Flow patterns and migration rule in 3D simulation	118
A.19	Mass flow patterns and migration rule for simulation in 2D hexagonal grid .	119
A.20	Funnel flow patterns for simulation in 2D hexagonal grid	119
A.21	Flow patterns in a silo with and without inserts	120
A.22	Flow patterns in a silo with and without internal hopper	120
A.23	Flow patterns in a silo with inclined tubes and perforated tube	121
A.24	Vectors of particle velocities at each site for improved model	122
A.25	Example traces of single particles in the gravity field	123
A.26	Examples of collision rules for 4 particles flowing to one random site	124
A.27	Flow pattern in a mass flow silo with default parameters (improved model) .	125
A.28	Mass flow silo: (a) $g = 0.95$, (b) $g = 0.05$	125
A.29	Mass flow silo: (a) $h = 0.8$, (b) $h = 0.0$	126
A.30	Mass flow silo: (a) $k = 0.8$, (b) $k = 0.0$	126
A.31	Mass flow silo: (a) $b = 0.8$, (b) $b = 0.0$	126
A.32	Mass flow silo: (a) $p = 0.8$, (b) $p = 0.0$	127
A.33	Mass flow silo: (a) $q = 0.8$, (b) $q = 0.0$	127
A.34	Flow pattern in a funnel flow silo with default parameters (improved model)	127
A.35	Funnel flow silo: (a) $g = 0.95$, (b) $g = 0.05$	127
A.36	Funnel flow silo: (a) $h = 0.8$, (b) $h = 0.0$	128
A.37	Funnel flow silo: (a) $k = 0.8$, (b) $k = 0.0$	128
A.38	Funnel flow silo: (a) $b = 0.8$, (b) $b = 0.0$	128
A.39	Funnel flow silo: (a) $p = 0.8$, (b) $p = 0.0$	128

A.40	Funnel flow silo: (a) $q = 0.8$, (b) $q = 0.0$	128
A.41	Formation of a granular pile	129
A.42	Evolution of density waves during granular flow in a vertical channel	130
B.1	Layered structure of YADE framework.	134
B.2	Distribution of forces in a two dimensional DEM, as calculated in YADE .	135
B.3	A three dimensional DEM sample in YADE subject to compression	136
B.4	A DEM beam subject to three point bending in YADE	136
B.5	A FEM beam subject to uniaxial extension in YADE	137
B.6	A lattice beam model specimen subject to uniaxial extension in YADE . . .	139
B.7	A view on YADE's main graphical interface	139
B.8	Class <i>Body</i> , and abstract interface classes contained by it	140
B.9	Examples of concrete classes contained by <i>Body</i>	140
B.10	Class <i>Interaction</i> , and abstract interface classes contained by it	141
B.11	Examples of concrete classes contained by <i>Interaction</i>	141
B.12	Abstract class <i>PhysicalAction</i> , with examples of concrete derived classes. .	142
B.13	Class <i>MetaBody</i> and container classes owned by it	142
B.14	One dimensional <i>MetaEngine</i>	142
B.15	Two dimensional <i>MetaEngine</i>	142

List of Tables

2.1	Beam strengths for different material phases; [154]	10
3.1	Receipts for concrete used in experiments	35
4.1	Main parameters used to describe the Lattice Geometrical Model	46
4.2	The variables stored in each node element	47
4.3	The variables stored in each beam element	48
4.4	The variables stored in each flexural–torsional spring element	56
4.5	Mesh generation parameters used in the Lattice Geometrical Model	65
4.6	Additional fracture parameters	67
4.7	Parameters used to demonstrate the effect of mesh type and density	73
5.1	Assumed parameter values used to examine their effect on material behavior	80
5.2	Material parameters used for testing the effect of ϵ_{\min}	86
5.3	The calibrated material parameters	88
5.4	Parameters used in calculation with two–phase material	89
5.5	Parameters used in calculation with three–phase material	91
5.6	The calibrated material parameters (three–phase material)	97

Chapter 1

Introduction

Engineering materials such as sand, concrete, rock, ceramics and polymers have common properties like heterogeneity, anisotropy, discrete structure and nonlinear behavior. To describe them, two kinds of numerical models have been developed: continuum models (within fracture, damage, softening plasticity mechanics) and discrete models (molecular dynamics, discrete element method, lattice models). The fracture which is a typical discontinuity in brittle materials, poses big difficulties when using classical constitutive laws without microstructure due to their inability to correctly predict the mechanism of its occurrence.

This thesis will present a discrete model for simulating a fracture process in brittle materials which contributes to a better description and understanding of the mechanism of fracture in brittle materials at the meso-level.

1.1 Phenomenon and Problem

Fracture process is a fundamental phenomenon in brittle materials [8]. It is a major reason of damage in brittle materials under mechanical loading contributing to a significant degradation of the material strength. It is highly complex due to a heterogeneous structure of brittle materials over many different length scales, changing e.g. in concrete from the few nanometers (hydrated cement) to the millimeters (aggregate particles). Therefore, the material heterogeneity should be taken into account when modelling the material behavior. At the meso-level, concrete can be considered as a three-phase material consisting of aggregate, cement matrix and interfacial transition zone (bond). A realistic description of the fracture process is of major importance to ensure safety of the structure and to optimize the behavior of material.

The discrete models currently used to describe the behavior of concrete are very complex which allows them to operate at a higher scale of discretization but makes it difficult to understand the underlying nature of the material. Another possibility is to use a simple constitutive law at a smaller scale of discretization, but it has prohibitive requirements with respect to memory and computational time. This second possibility was taken advantage of in this thesis.

1.2 Aims

The goals are:

- a) to create a three-dimensional discrete lattice model to describe the behavior of quasi-brittle materials at the meso-level (eg. concrete and reinforced concrete) under different load conditions (tension, compression and combined shear-extension),
- b) to compare the results of this model with results of experiments (own and from literature),
- c) to examine the effect of aggregate density and interfacial zones on the fracture process,
- d) to investigate a size effect (dependence of nominal strength on the specimen size).

The new model is of an explicit type which allows for an easy distribution of calculations on several computers and multiple processors to reduce the computation time. The constitutive law used is simple elastic to further reduce the computation effort.

1.3 Structure of thesis

This thesis contains 6 chapters and 3 appendices. Chapter 2 reviews already existing discrete models used to describe the behavior of concrete. Presented models are the discrete element method, the classical lattice-beam model, the nonlinear interfacial zone model and the confinement-shear lattice model.

Chapter 3 presents a digital image correlation technique for measuring the surface strain in concrete. A novel software was written to perform the digital image correlation technique. The experimental results were compared with the numerical results of the discrete model.

Chapter 4 presents a novel lattice model developed by the author. In contrast to other lattice models, a geometric type lattice model was used. Owing to that, the computational effort was reduced. The method of generating the aggregate structure and one of two mesh generation methods were taken from literature. The aspects of computation and the properties of the model are discussed.

Chapter 5 presents the verification of the model. The results of simulations were compared with own experimental results and experimental results from literature. The numerical tests of uniaxial tension, uniaxial compression and combined extension-shear were performed. The influence of aggregate structure on the results was examined.

Chapter 6 includes conclusions.

Appendix A describes two discrete cellular automaton models to calculate the kinematics of non-cohesive granular materials. The results were compared with mass and funnel flow experiments in a model silo made of perspex.

Appendix B deals with the problems of the modern software design applied to numerical simulations such as a discrete element method DEM, finite element method or a lattice model presented in Chapter 4. A new software framework was written in C++ by the author of this thesis in cooperation with University Joseph-Fourier in Grenoble. The software currently is in the development stage.

Appendix C briefly explains the basics of quaternion rotations used in the lattice model.

1.4 Novelties

The novelties in this thesis are following:

- a) a novel lattice model of a geometric type (according to the own idea),
- b) application of a DIC technique to measure strain on the surface of concrete beams (according to the own idea),
- c) application of a cellular automaton to investigate the kinematics of granular flow in silos.

Chapter 2

Literature review

The phenomenon of the propagation of the fracture process in brittle materials can be modelled with continuous and discrete models. Continuum models describing the mechanical behavior of concrete were formulated within among others non-linear elasticity [87, 101, 114], rate-independent plasticity [21, 86, 104, 108, 121], damage theory [23, 53, 55, 117], endochronic theory [9, 14], coupled damage and plasticity [48, 77, 86] and microplane theory [12]. To model the thickness and spacing of strain localization properly, continuum models require an extension in the form of a characteristic length. Such an extension can be done with strain gradient [33, 109, 115–117, 163], viscous [144, 145] and non-local terms [7, 10, 20, 122]. Other numerical technique which also enables to remedy the drawbacks of a standard FE-method and to obtain mesh-independent results during the description of the formation of strain localization is a strong discontinuity approach allowing a finite element with a displacement discontinuity [3, 142, 161].

Within discrete methods, the most popular ones are: classical particle DEM [38, 45, 54, 72, 73, 151, 164], interface element [26–29], lattice methods [74, 99, 136, 154–159] and confinement-shear lattice models [39–43, 148]. Those models are a realistic alternative to nonlocal continuum models for a fracture phenomenon. They automatically exhibit the size effect, and eliminate problems with mesh sensitivity of local continuum models. Compared to nonlocal continuum models, the random particle or lattice models have several advantages: allow a straightforward implementation of the material heterogeneity which is projected on a lattice and corresponding properties are assigned to relevant lattice elements and they allow to simulate the development and propagation of fracture in brittle materials consisting of a main crack with various branches, secondary cracks and microcracks.

2.1 Classical spherical discrete elements model

The DEM¹ [38, 45, 54, 72, 73, 151, 164] method was initially developed by Cundall in 1979 for the analysis of rock. It is a numerical model capable of describing the mechanical behavior of assemblies of discs and spheres. It uses an explicit numerical scheme in which the interaction of the particles is monitored contact by contact with states of equilibrium. The equilibrium contact forces and displacements of a stressed assembly of spheres are found through a series of calculations tracing the movements of the individual particles. These movements are the result of the propagation through the medium of disturbances originating at the boundaries (a

¹Distinct Element Method or Discrete Element Method or Spherical Discrete Element Code (SDEC)

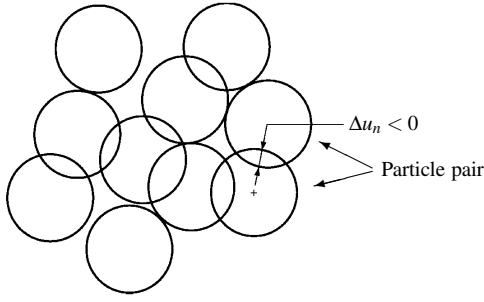


Figure 2.1: An interaction between spherical discrete elements, overlaps exaggerated for clarity [54, 71–73]

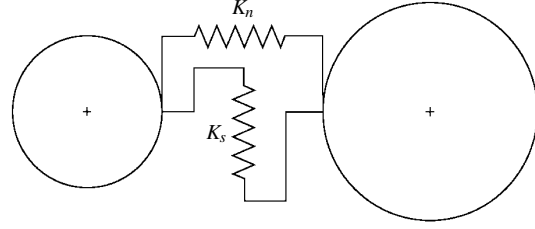


Figure 2.2: Elastic interaction, normal and shear springs (K_n, K_s respectively) [54]

dynamic process). The motion of the particles is modelled particle by particle. The DEM is based upon the idea that the time step chosen may be so small that during a single time step disturbances cannot propagate from any sphere further than its immediate neighbors. Then at all times the resultant forces on any sphere are determined exclusively by its interaction with the spheres with which it is in contact. It is the key feature of DEM, which makes it possible to follow a non-linear interaction of a large number of particles without excessive memory requirements or the need for an iterative procedure.

The calculations performed by DEM alternate between the application of the Newton's second law to spheres:

$$F = ma, \quad (2.1)$$

and force–displacement law at contacts:

$$\begin{aligned} F_n &= K_n \Delta u_n, \\ F_s &= K_s \Delta u_s, \end{aligned} \quad (2.2)$$

where K_n – normal stiffness at contact, K_s – tangential stiffness, Δu_n – relative particle normal displacement, u_s – incremental tangential displacement, a – acceleration (Figs. 2.1 and 2.2). The Newton's second law gives the acceleration of a particle resulting from the forces acting on it. The force–displacement law is used to find contact forces from displacements. In general case of an assembly of many spheres, the force–displacement law is applied at each contact of any sphere and the vectorial sum of these contact forces is determined to yield the resultant force acting on that sphere. When this has been accomplished for every sphere, new accelerations are calculated from Newton's second law. This procedure is repeated in a loop.

The deformations of individual particles are small in comparison with the deformation of a granular assembly as a whole. The latter deformation is due primarily to the movements of the particles as rigid bodies. Therefore a precise modelling of the particle deformation is not necessary to obtain a good approximation of the mechanical behavior. Consequently the particles are allowed to overlap one another at contact points, and this behavior takes the place of the deformation of individual particles. The magnitude of the overlap is related directly to the contact force. These overlaps are small in relation to particle sizes. The boundary conditions are of a strain controlled type. The movement of the wall is defined in terms of the velocity and angular velocity. For the purpose of assigning physical properties to the spheres and walls, they are divided into groups such that all entries of a group have the



Figure 2.3: View on a cylindrical DEM specimen of concrete [54, 71–73]

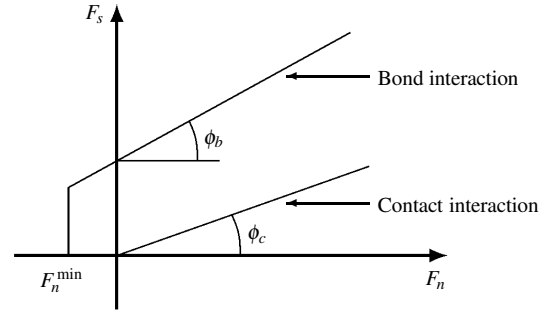


Figure 2.4: Rupture criterion used in the DEM model [54]

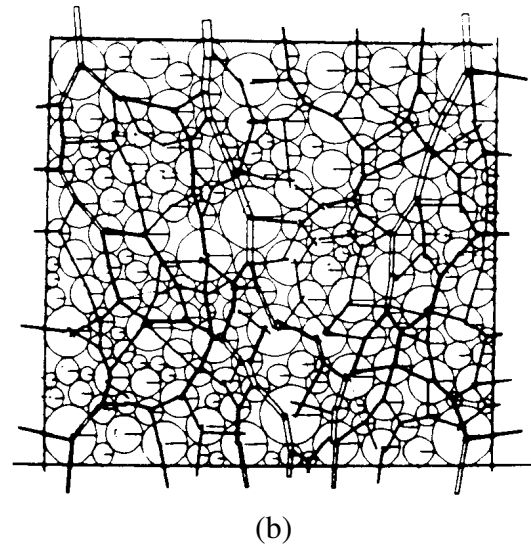
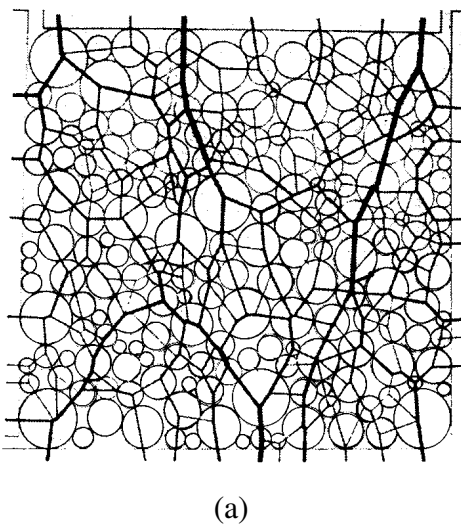


Figure 2.5: Distribution of forces in a two dimensional DEM sample subject to compression: (a) results obtained from experiments by means of photoelastic analysis [49]; (b) results from DEM simulation [38]

same properties. Assigning a single type number to each sphere or wall is then sufficient to associate with it the appropriate set of properties. The physical properties treated in this way include sphere radius, density, cohesion, inter-particle friction coefficient, shear stiffness and normal stiffness. There are other parameters that apply to field as a whole, defining the contact damping, global damping and the fraction of the critical time step.

Donzé and Hentz [54, 73] have extended a classical DEM model by introducing an interaction range, which allows setting of an interaction between two spheres which are not in contact. A pair of particles sustains tensile forces, which can be used for simulating a bond between them, which in turn allows to simulate concrete behavior (Fig. 2.3). The bond can be broken when a tensile force exceeds a maximum tensile force F_n^{\min} . Figure 2.4 shows a modified Mohr–Coulomb rupture criterion used in the model. It is assumed that linked spheres (forming a body of concrete) can sustain higher forces than those of the non-linked contact. After the initial interactions have broken, new ones are identified, which are not cohesive any more: they are merely contact interactions, and cannot undergo any tension force. Then a classical Coulomb criterion is used with a contact friction angle, ϕ_c . This allows simulating complex macroscopic behaviors such as strain softening or fracture that

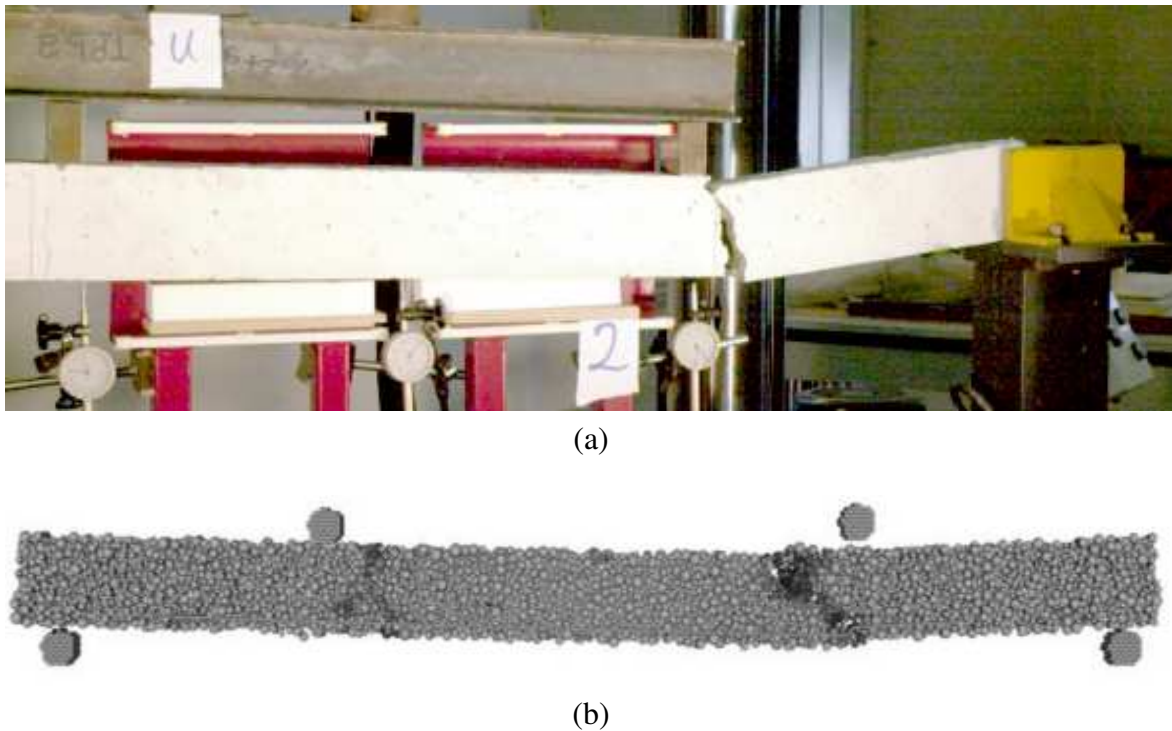


Figure 2.6: A DEM specimen of reinforced concrete subject to bending, an experiment performed by Hentz [71]: (a) the experiment; (b) result from DEM simulation

arises from micro-cracking throughout the assembly.

When DEM was first introduced it was validated by De Josselin de Jong and Verruijt [49] using photo-elastic analysis on the assembly of discs (Fig. 2.5). Those results showed a lot of promises for DEM and the model was pursued in years after with many enhancements. Nevertheless, this is a rather simple way to model the complexity of the microscopic phenomena, and in particular, particle rotations are not realistically constrained. To simulate granular and cohesive materials, different solutions were proposed: Potyondi proposed elements of irregular shape [124, 125], Calvetti et al. proposed to inhibit the rotations [30]. Iwashita et al. proposed to convey the rotation between particles by the introduction of a law couple/rotation [78–80] to obtain a reasonable friction angle. Hentz used the first two methods [71]. Figure 2.6 shows his experiment with a beam subject to asymmetric bending.

2.2 Classical lattice-beam fracture model

The lattice model was first introduced by Hrennikoff [75] in 1941. He demonstrated that a regular triangular lattice of bars (a truss) with Poisson's ratio fixed at $1/3$ is capable of solving problems of continuum elasticity. At that time the computer's power was insufficient to continue investigation thus the work stalled till 1989 when Herrmann [74] proposed to use beam elements instead of bars and to use a regular or random lattice. He laid foundations for a classical lattice model introduced by van Mier, Schlangen, Lilliu and van Vliet [15, 99, 133–138, 153–158].

The main purpose of a classical lattice model is to describe the fracture process in concrete or reinforced concrete. Each quasi-brittle material is discretized as a network of two

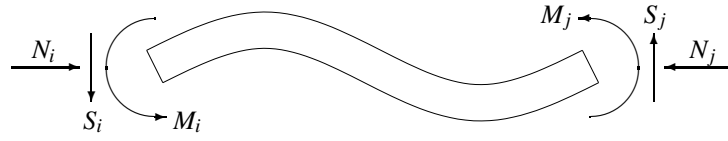


Figure 2.7: Normal, shearing force and momentum on lattice beam [154]

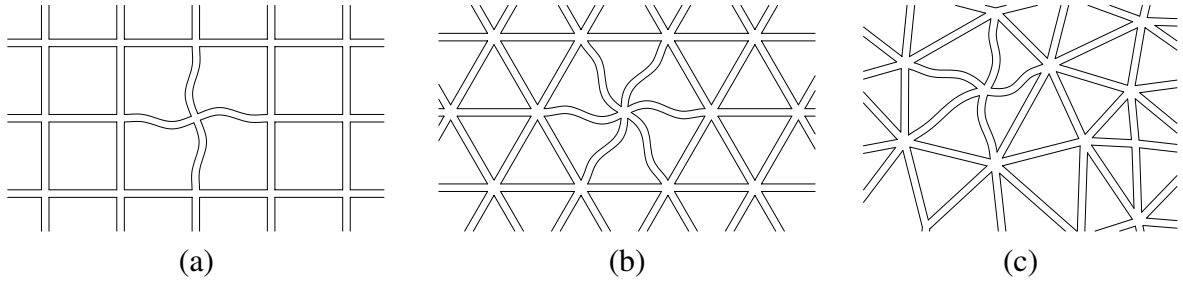


Figure 2.8: Classical lattice types: (a) regular square lattice presented in [74]; (b) regular triangular lattice presented in [137]; (c) random lattice presented in [154]

noded Bernoulli beams (Fig. 2.7) that transfer normal forces, shear forces and bending moments. In this model the size of the beams must be adjusted to achieve desired Poisson's ratio. For an orthogonal lattice (Fig. 2.8 a), the Poisson's ratio is always zero, but for a regular triangular or random lattice (Fig. 2.8 b,c) a different Poisson's ratio (excluding negative values) can be set depending on the height over length ratio of the lattice beams.

Figure 2.9 shows the generation of a random lattice. First a square grid with spacing g is made. In each box of the grid, a point is selected at random inside a sub-square of the size s . The points are connected in quadrilaterals and each quadrilateral has an additional connection along the shorter diagonal. The lattice randomness can be varied by changing s . For $s = 0$, an orthogonal lattice is obtained, when $s > g$ the obtained mesh is highly irregular. The generated lattice has a spatial directional bias and in fact has nothing to do with concrete, or any other brittle material. To perform realistic computations of concrete, the material structure must be directly projected on the lattice.

Concrete is a multi-phase material consisting of aggregates, pores, cement matrix and an interface between aggregate and matrix. For computational reasons, it is impossible to model the concrete down to the smallest particle in the material structure. A widely accepted limit of the smallest aggregate diameter in the lattice is 2 mm [154]. The aggregates are assumed to be spherical and are embedded in a homogeneous matrix. First, a particle structure is generated. Currently there are two widely used methods. The first one allows to obtain a dense packing of aggregates in three dimensional body of concrete by using a Fuller distribution [154]:

$$p = 100 \sqrt{\left(\frac{D}{D_{\max}} \right)}, \quad (2.3)$$

where p is the weight percentage of particles passing a sieve with diameter D , and D_{\max} is the size of a largest particle. Furthermore by using a cumulative distribution for a two-dimensional cross section derived by Walraven [154, 160], the circle diameters for a concrete sample can be generated. The second method of particle generation used by Eckard [60]

Table 2.1: Beam strengths for different material phases; [154]

Phase	Young's modulus	Maximum stress	Maximum strain
Interface bond	$E_b = 25 \text{ GPa}$	$f_b = 1.25 \text{ MPa}$	$\varepsilon_b = 0.005\%$
Cement matrix	$E_m = 25 \text{ GPa}$	$f_m = 5 \text{ MPa}$	$\varepsilon_m = 0.02\%$
Aggregate	$E_a = 70 \text{ GPa}$	$f_a = 10 \text{ MPa}$	$\varepsilon_a = 0.0143\%$

is more straightforward. First, a grading curve is chosen (based on experimental measurements). Next, the certain amounts of particles with defined diameters $d_1, d_2 \dots d_n$ are generated according to this curve (Fig. 2.10). The author of this thesis used the latter method in his work (Sect. 4.5.1). No matter which method is used, the particle structure of concrete can be generated as shown on Fig. 2.11. The circles (or spheres in three-dimensions) are randomly placed in the prescribed area starting with the largest ones and preserving a certain mutual distance [154]:

$$D > 1.1 \frac{D_1 + D_2}{2}, \quad (2.4)$$

where D is the distance between two neighboring particle centers and D_1 and D_2 are the diameters of two particles. The next step is to overlay the generated particle structure with a lattice as shown in Fig. 2.11 b. The lattice beam length should be less than $l_b < d_a^{\min}/3$, where d_a^{\min} is the minimum aggregate diameter. With assumed $d_a^{\min} = 2 \text{ mm}$, the maximum lattice length is 0.7 mm . Otherwise the computational effort would become large. Moreover, to restrict the computational effort, a lattice model is applied only in the area where cracks are expected to occur (Fig. 2.12).

After the lattice is projected on top of the particle structure, different properties (Tab. 2.1) are assigned to beams appearing in different material phases: aggregate, matrix or bond (Fig. 2.11 c). A bond is defined as a beam that has one node inside an aggregate and other node inside cement matrix or other aggregates. Van Mier has assumed [154] that the modulus of elasticity for the bond is equal or smaller to that of the cement matrix (Tab. 2.1). The bond properties influence significantly the results (Sect. 5.3).

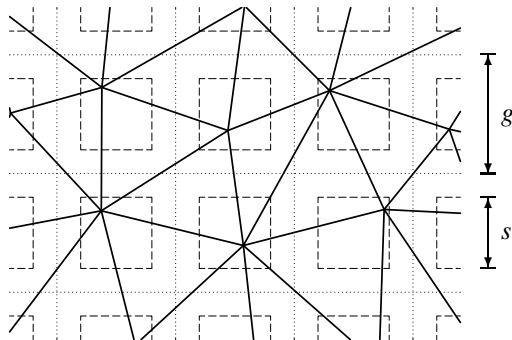


Figure 2.9: Construction of a random lattice [133]

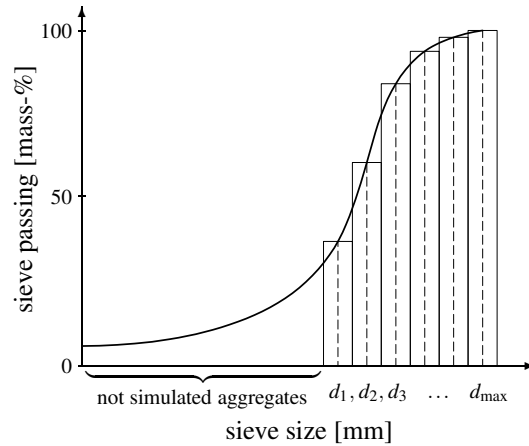


Figure 2.10: Approximation of the grading curve with discrete number of aggregate sizes [40]

Normal forces, shear forces and moments are calculated using a conventional simple beam theory. The stiffness matrix is constructed for the entire lattice. The displacement vector is calculated using conventional FEM. Fracture is simulated by performing a linear elastic analysis up to the failure under loading and removing a beam element that exceeds maximum effective stress (Eq. 2.5, Tab. 2.1). To solve FEM, a gradient method is found to be most useful [135], because using it, the beam removal is a local effect and only few iterations are needed to relax the system. Then the next critical beam is identified and the above procedure is repeated. The effective stress first proposed by van Mier then extended by van Vliet [154, 156] is a combination of normal stress and flexural stress in the beams:

$$\sigma = \frac{N}{A} \pm \alpha \frac{\max(|M_i|, |M_j|)}{W}, \quad (2.5)$$

where N is the normal force in the beam, M_i, M_j are bending moments in the nodes i and j , $A = bh$ is the beam cross-section and $W = bh^2/6$ is the section modulus. The α parameter is used for varying the amount of bending. When it decreases, the compressive behavior changes from brittle to ductile one. The value of $\alpha = 0.005$ is usually used [156].

An example simulation performed by Schlangen [138] (Fig. 2.12) shows a specimen in a four point shear test used to study the crack growth under a combination of mode I (tensile) and mode II (in-plane shear) loading. The experimentally obtained crack has a curved shape (Fig. 2.12 a) due to rotation of the direction of the maximum tensile stress. To perform computations, only a middle 10×10 cm part of the specimen is discretized with lattice beams, the rest is a regular FEM mesh. The final crack pattern is shown on Fig. 2.12 b, and agrees with the experimental one.

Summarizing, the classical lattice model [15, 99, 133–138, 153–158] can identify micro-cracking, crack branching, crack tortuosity and bridging which leads to the fracture process to be followed until complete failure. It enables also to capture a size effect during tension [159]. The advantages of this approach are simplicity and a direct insight in the fracture process on the level of the micro-structure. A complex crack patterning can be reproduced. Therein a limited number of parameters is needed. By applying an elastic–purely brittle local fracture law at the particle level, global softening behavior is observed. The fracture process is realistically described, in particular, when the mode I failure prevails. The disadvantages of this model are the following: when calibrated to correctly predict tensile cracking, it over-

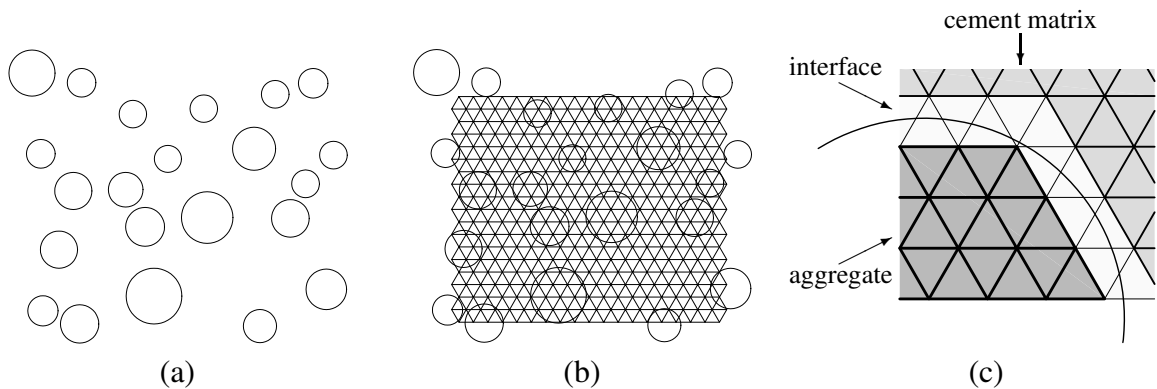


Figure 2.11: Particle structure in the classical lattice-beam model: (a) generated particle structure of concrete; (b) example of the projection of regular triangular lattice on the generated structure; (c) definition of aggregate, cement matrix and interface bond [137, 154]

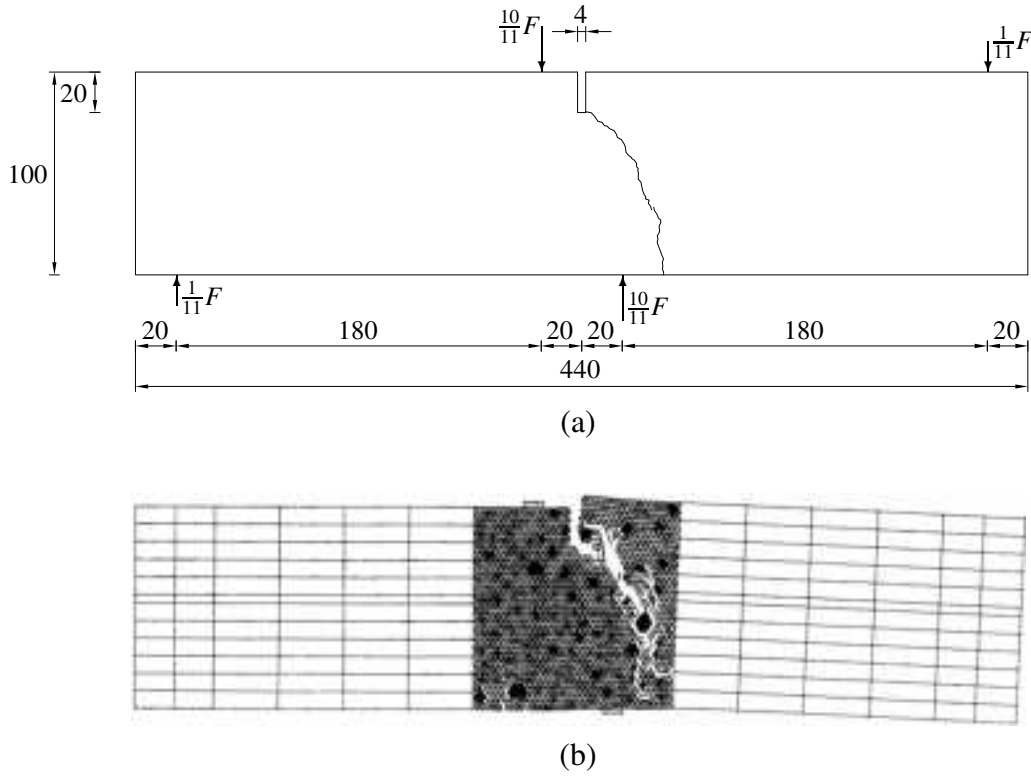


Figure 2.12: Four point shear test: (a) initial configuration and experimental crack pattern; (b) simulation performed by Schlangen and van Mier [138]

predicts the uniaxial compressive strength, the response of the material is too brittle (due to the assumed brittleness of single beams), the compressed beam elements overlap each other and a big computational effort is needed. The first and second drawback can be improved by applying a non-local approach for beam deformations, as proposed by Schlangen [136], and also by a 3D version by Lilliu [99] which increases the amount of the crack face bridging and dissipated energy. The computational effort was recently reduced by using a conjugate gradient solver [136]. In this algorithm, breaking an element and removing it from the lattice was a local effect, and the solution required only a few iterations.

2.3 Nonlinear interfacial zone model

The three-dimensional nonlinear interfacial zone model was developed by Caballero, Carol and López [26–29], based on earlier work by Carol and López [32]. In this approach, the heterogeneous concrete is discretized as a composite material of the larger aggregates embedded in a matrix phase representing mortar plus smaller aggregates. The polyhedral geometry is numerically generated by a standard Voronoi/Delaunay tessellation. Both continuum-type components are considered linear elastic and are represented as tetrahedron FEM elements. The possibilities of failure are provided by zero-thickness interface elements (that use a cohesive fracture constitutive law) which are inserted between all particle-matrix interfaces and also along selected matrix-matrix inter-element boundaries representing the main potential crack patterns. The purpose of nonlinear interfacial zone model is to reproduce in 3D the experimental behavior of concrete: representation of crack patterns, stress-strain curves

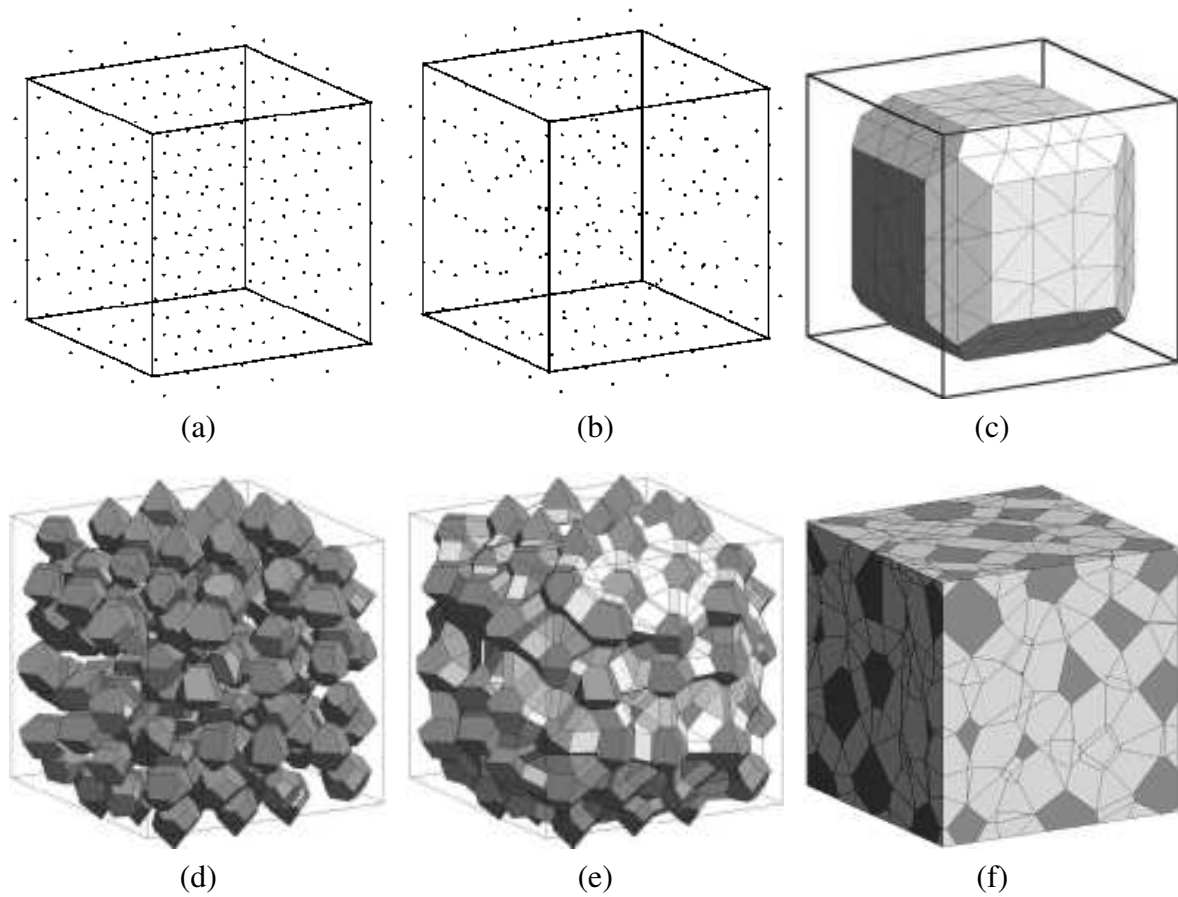


Figure 2.13: Discretization process: (a) regular distribution of points; (b) each point displaced by a random vector; (c) Delaunay mesh obtained from points; (d) aggregates obtained from Voronoi polyhedrons; (e) filling space between polyhedrons by mortar; (f) specimen after cutting [27]

and material failure mechanisms.

Figure 2.13 a,b shows the first step of model generation [26–29]: a 3D space is filled by regular distribution of points, then each point is randomly displaced (compare with Fig. 2.9). On those points, a Delaunay tetrahedral mesh is generated (Fig. 2.13 c). The duality between Delaunay and Voronoi spaces allows to uniquely obtain Voronoi polyhedrons by connecting centers of each tetrahedron with all its neighbors (see 2D example on Fig. 2.15). Those polyhedrons are guaranteed to be convex and each of them contains inside exactly one initially generated point. Next step is to shrink each of them by a certain factor (random or based on grading curve, Fig. 2.10), from now on those convex polyhedrons will be the aggregates in concrete (Fig. 2.13 d). Next, the space between aggregates is filled with the volume of mortar and smaller aggregates (Fig. 2.13 e). From generated mesh a smaller specimen is cut out so that its faces are planar (Fig. 2.13 f). Then using the aggregate's central point and points lying on the center of all its surfaces, the polyhedron is subdivided into tetrahedron volumes. Next the zero-thickness failure interface planes are inserted along all potential fracture surfaces, those include: the aggregate surfaces, new subdivisions of mortar defined by two parallel edges belonging to opposite aggregates (Fig. 2.14 a, top) and new subdivisions of mortar by means of each aggregate edge and a point located in the center of space between two polyhedrons facing each other (Fig. 2.14 a, bottom). Figure 2.14 b shows a view of the part of generated specimen.

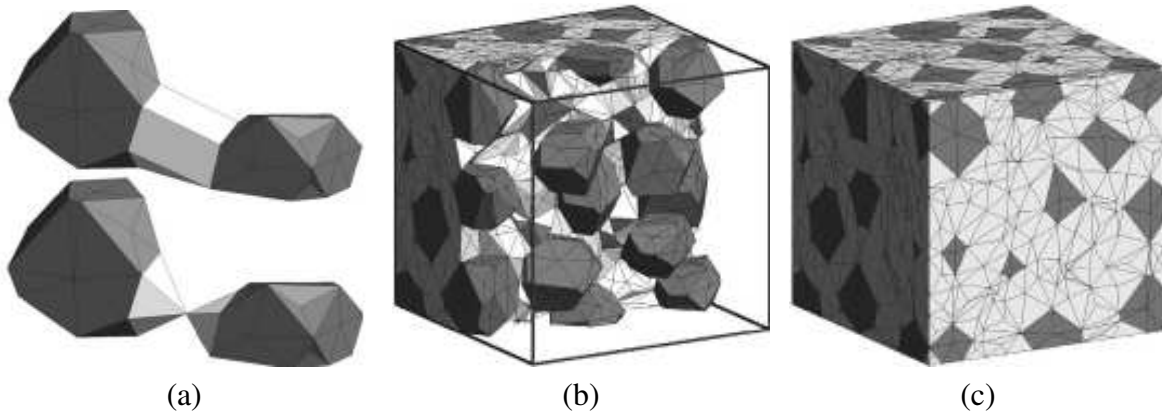


Figure 2.14: Discretization process: (a) generation of potential failure surfaces (b) view of the part of generated specimen (c) final result of specimen generation [27]

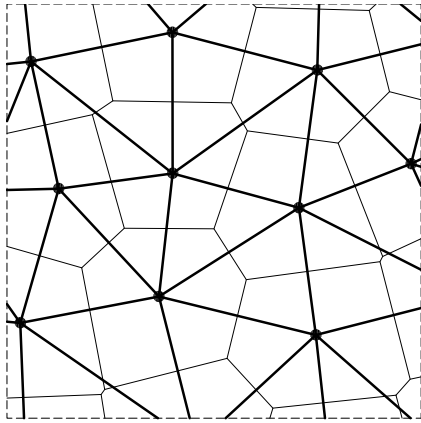


Figure 2.15: The Delaunay triangulation (thick lines) and Voronoi tessellation (thin lines) are dual to each other in the graph theoretical sense.

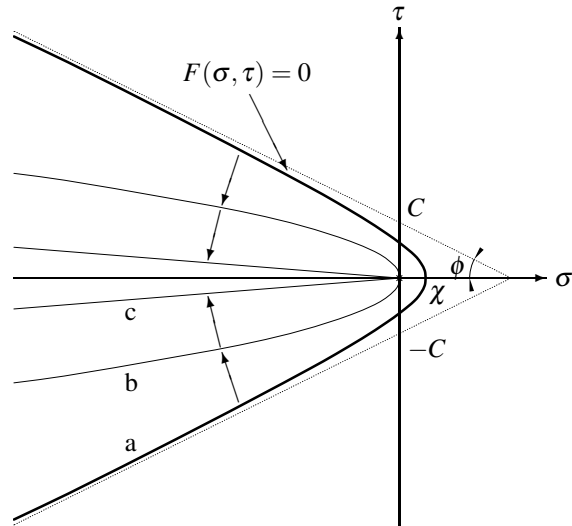


Figure 2.16: Interface cracking law: fracture surface F (a) and its evolution (a,b,c) [27]

At this stage the geometry is already subdivided in tetrahedrons, but in the process some extremely short edges, surfaces with very small angles or excessively flat volumes were generated. To fix this, a „geometry collapse” process is performed by merging all vertices that are very close to each other and eliminating disproportional volumes. The cleansed mesh is still too coarse for straightforward turning each tetrahedron into a FEM element, and first an additional tetrahedron subdivision is performed. When specimen discretization with continuum elements is done, the zero-thickness interface elements are introduced along all potential fracture surfaces. This is done by simple duplication of nodes and changes in FEM element nodal connectivities. This process increases the number of nodes considerably, while the number of continuum elements remains unchanged. Figure 2.14 c shows the final result of specimen generation.

The continuum tetrahedron FEM elements show the elastic behavior. The zero-thickness interface FEM elements have the nonlinear behavior formulated in terms of one normal and

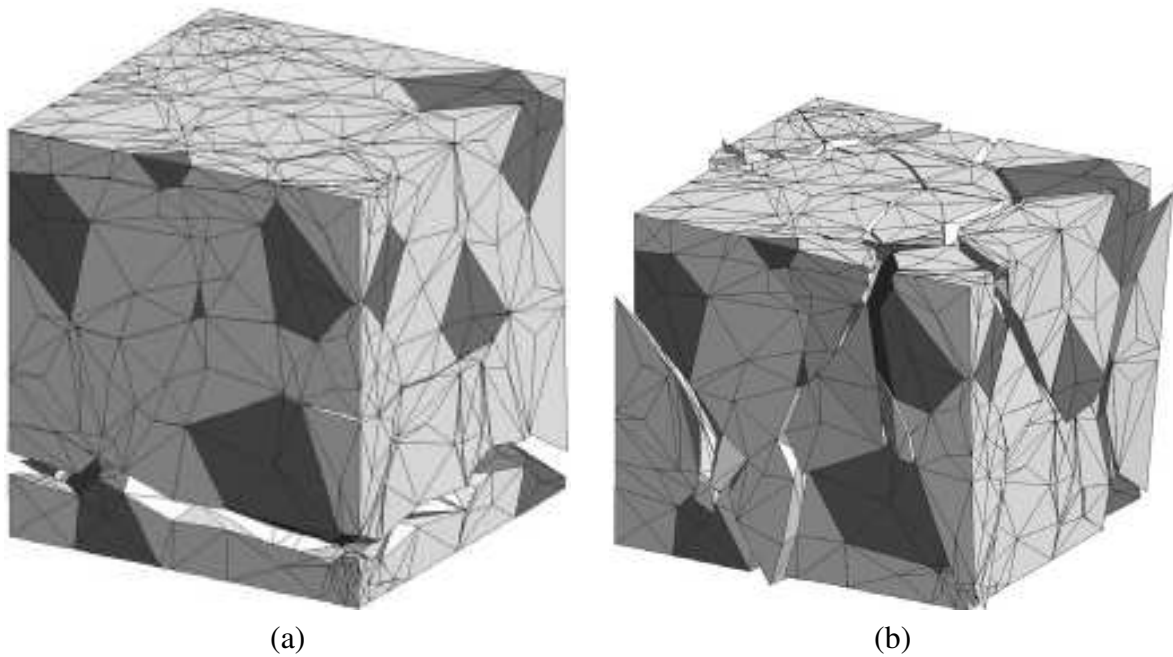


Figure 2.17: The deformation and crack pattern for concrete specimen under uniaxial: (a) tension; (b) compression [26]

two tangential traction components on the contact surface, $\sigma = (\sigma_N, \sigma_{T1}, \sigma_{T2})$. The constitutive formulation is related to work-softening elasto-plasticity, in which plastic relative displacements can be identified as crack openings [26–29]. The initial yield surface $F = 0$ is given as a three parameter hyperboloid (Fig. 2.16): χ is the maximum tensile strength, C is asymptotic cohesion and ϕ is a friction angle. When cracking starts, the loading surface begins to shrink. This is achieved by means of softening laws in which the surface parameters are functions of the work of the fracture process. Under pure tension, the loading surface (Fig. 2.16 a) moves to become another hyperboloid starting at the coordinate origin (Fig. 2.16 b). Under mixed-mode it degenerates further, asymptotically becoming a cone (Fig. 2.16 c) that represents the residual friction after all interface roughness has been eliminated.

An example simulation performed by Caballero [26] with a specimen consisting of 14 aggregates (5755 tetrahedra and 3991 interface elements). The material parameters used by Caballero are similar to those proposed by van Mier (Tab. 2.1): $E_a = 70$ GPa, $E_m = 25$ GPa and Poisson's ratio $\nu = 0.2$ for mortar and aggregate. The specimen is subject to uniaxial tension and compression. Figure 2.17 shows the crack pattern in the deformed specimens subject to load. The results are in good agreement with laboratory experiments.

Summarizing, the nonlinear interfacial zone model has following advantages: a realistic representation of stress-strain curve in tension and compression as well as crack patterns and their evolution, the localization process is spontaneously starting from disturbed micro-cracking and leading to macro-cracks that cross the specimen. The disadvantages of the model are following: the zero-thickness elements multiply the number of nodes by a factor of five, which enhances the calculation cost [27], the model exhibits a mesh dependency and is restricted by deformation capabilities of finite elements.

2.4 Confinement–shear lattice model

The confinement shear lattice model developed by Cusatis et al. [39–43] simulates the concrete meso–structure by means of a three–dimensional lattice in which the nodes are the centers of coarse aggregates pieces, and the adjacent aggregates are connected by struts. The topology of strut connections is obtained by a three–dimensional Delaunay triangulation. The struts reproduce interaction between connected aggregates through the embedding matrix using a nonlinear damage–like constitutive relation which exhibits fracture, friction and cohesion at the meso–level.

The confinement shear lattice model can describe a nonlinear uniaxial, biaxial, and tri-axial response in compression, strain localization and mixed–mode crack propagation in concrete.

A specimen generation begins with a random placing of spheres (aggregates) from large to small in the prescribed 3D space. The diameters are given by a grading curve (Fig. 2.10), using a Fuller distribution (Eq. 2.3). If a newly placed sphere overlaps an already existing sphere or crosses the specimen’s boundary, another random coordinate is chosen. Additionally the zero-radius spheres are placed at specimen boundary with a similar spacing as spacing inside the specimen. This enables discretizing the surface without effort because the boundary zero–radius aggregates are treated similarly as other aggregates. Next a Delaunay triangulation is performed treating each aggregate’s center as a tessellation node (Fig. 2.13 c). Each tetrahedron’s edge is a strut that connects the aggregates and its cross-section area must be determined somehow. This cross–section is also a contact area between the two aggregates. The Voronoi tessellation (Fig. 2.15) is unsuitable for this purpose, because the cross–section would be always at the midpoint of the tetrahedron edge and could intersect an aggregate. Therefore to obtain the area, a different tessellation method is used. Figure 2.18 a shows this tessellation performed in 2D. First the midpoints are calculated: on the strut (an edge) it is a midpoint of mortar between the two aggregates, inside the Delaunay triangle it is a center of mass of triangle without the aggregate volumes. The cross–section of the strut (Fig. 2.18 a,b) is defined by points A , B and C . In three dimensions along with previously defined points A , B and C on the tetrahedron’s triangular face, an additional point D must be used: the center of mass of the tetrahedron without sphere volumes (Fig. 2.18 c) Figure 2.18 d shows an example three dimensional cell containing an aggregate.

For the sake of simplicity, the constitutive law is imposed on a projection of this area on a plane orthogonal to the strut and containing point B of the tessellation (Fig. 2.18 e,f). The contact point E is the the center of mass of the projected area, which implies extra momentum in the contact and is necessary for a realistic simulation [42].

The strut simulates a contact layer of mortar (and smaller aggregates) between two adjacent aggregate particles and can transmit both normal and shear stresses. The shear and normal displacement are calculated in the middle of the contact zone from the relative rigid body motion of two adjacent particles. The displacement is then divided by the strut length assuming uniform normal (ϵ_N) and shear strain (ϵ_T) over its whole length.

The constitutive relation is described in [40]. In short, the elastic behavior can be characterized as $\sigma = E\epsilon$ or equivalently as $\sigma_N = E_N\epsilon_N$ and $\sigma_T = E_T\epsilon_T$ where $E_N = E$ and $E_T = \alpha E$. The constant α is the ratio between normal and tangential stiffness of the strut and controls macroscopic Poisson’s ratio ν . The value of $\alpha = 0.25$ causes $\nu = 0.18$.

The inelastic behavior, simulating damage, fracture and plasticity at the meso–level is formulated assuming the effective stress σ to be governed by a strain–dependent function

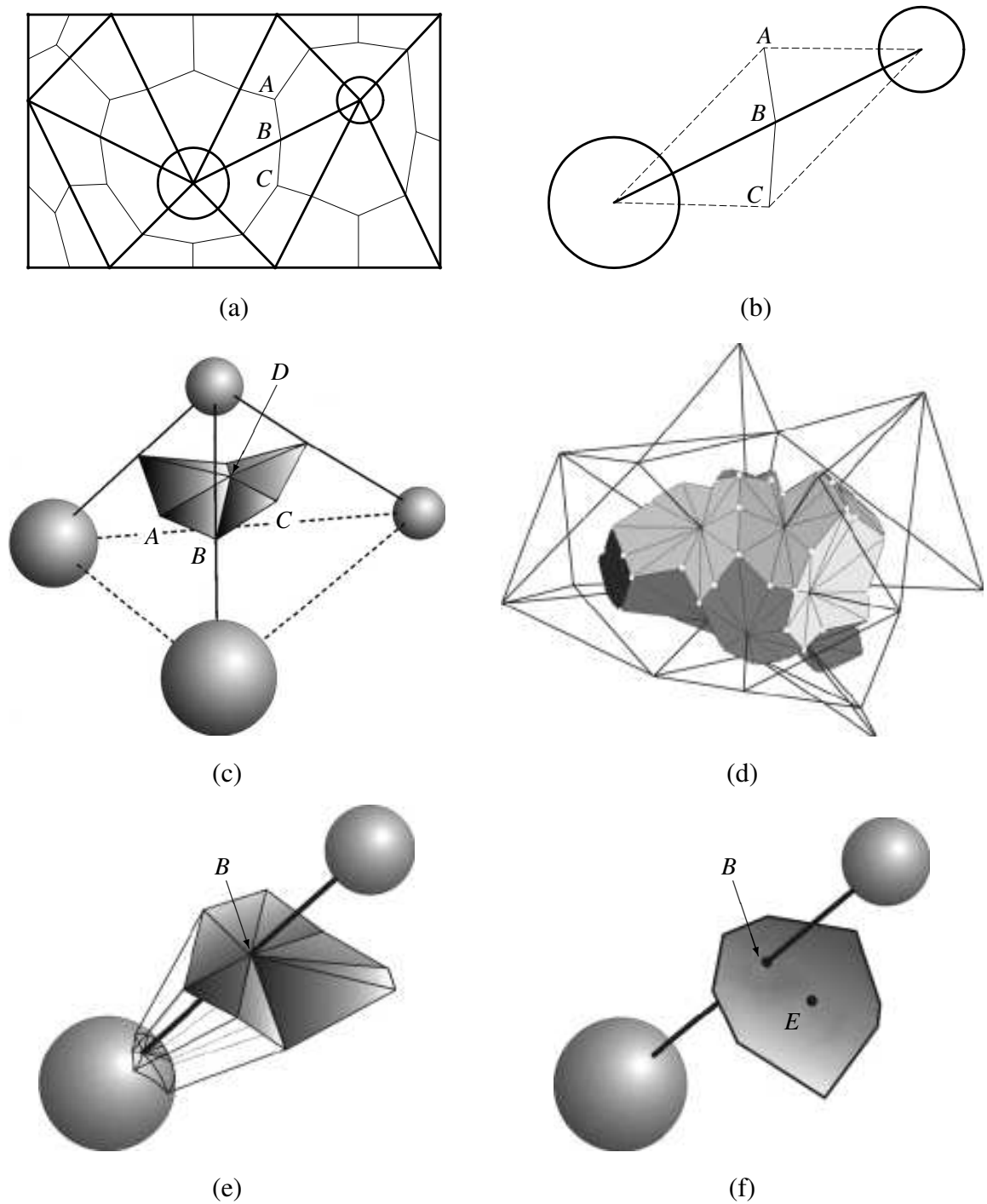


Figure 2.18: Discretization process: (a) example of two-dimensional tessellation (thin lines) of Delaunay triangulation (thick lines) with two aggregates in the middle, and boundary aggregates with zero radius around; (b) the geometry of the strut; (c) cross-section surface generated for upper aggregate of tetrahedron; (d) three dimensional cell containing an aggregate; (e) cross-section surface of the strut; (f) a simplified cross-section obtained by projecting it on a plane orthogonal to strut's direction [42]

which for non-cyclic loads is expressed as:

$$\sigma_b(\varepsilon, \omega) = \sigma_0(\omega) \exp \left(\frac{K(\omega)}{\sigma_0(\omega)} \left\langle \varepsilon - \frac{\sigma_0(\omega)}{E} \right\rangle \right), \quad (2.6)$$

in which $\langle \bullet \rangle = \max(\bullet, 0)$. Function $\sigma_0(\omega)$ is the initial effective strength function which delimits the elastic domain in the stress space (Fig. 2.19 a). It is a hyperbola with an ellipsoidal cap in compression and is defined by: tensile strength σ_t , shear strength σ_s , compressive strength σ_c , slope μ of the hyperbola asymptote and aspect ratio β of the ellipsoidal cap.

For $\omega < \omega_0$ there is hardening, and for $\omega > \omega_0$ softening, while $\omega = \omega_0$ is the transition from a hyperbola to an ellipsoidal cap. The initial hardening–softening modulus is given by the initial slope function $K(\omega)$ defined as (Fig. 2.19 b):

$$K(\omega) = \begin{cases} K_c \left(1 - \left(\frac{\omega + \pi/2}{\omega_0 + \pi/2} \right)^{n_c} \right) & \text{for } \omega \leq 0, \\ -K_t f(\lambda) \left(1 - \left(\frac{\omega - \pi/2}{\omega_0 - \pi/2} \right)^{n_t} \right) & \text{for } \omega > 0. \end{cases} \quad (2.7)$$

where K_c and n_c govern the nonlinear compressive and shear–compressive behavior at meso level, and parameters K_t and n_t govern the nonlinear tensile and shear–tensile behavior.

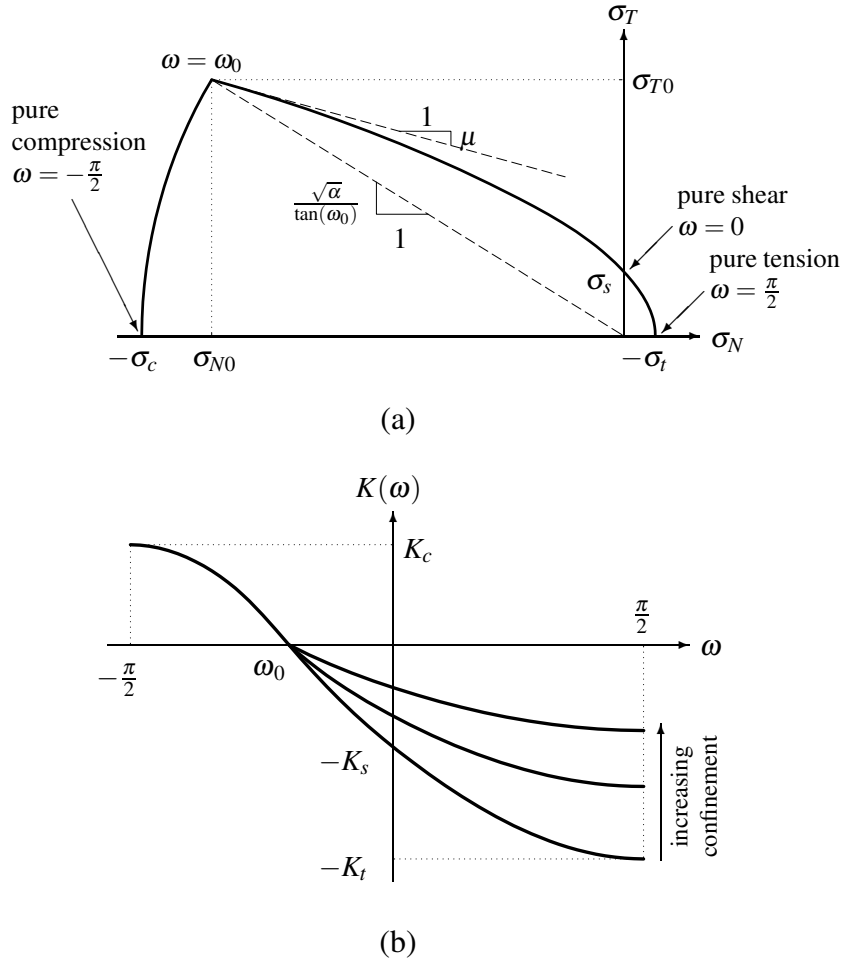


Figure 2.19: Constitutive law: (a) elastic domain at the strut's cross-section; (b) initial slope function [40, 42]

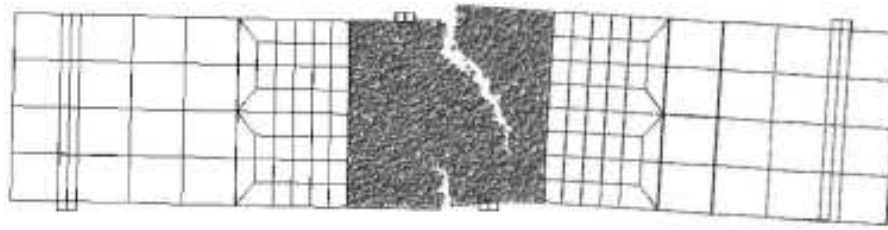


Figure 2.20: Four point shear test, the specimen configuration at failure [42]

For $\omega > \omega_0$, which includes pure shear ($\omega = 0$) and pure tension ($\omega = \pi/2$), the post-peak slope is made sensitive to the confinement transversal to the connection (up arrow on Fig. 2.19 b). This dependence is justified by the fact that the degree of the transversal confinement must affect the resistance to the frictional pullout of crack-bridging fragments and small aggregate pieces, which is known to cause softening to be gradual rather than sharp. The confinement effect is introduced in Eq. 2.7 through the function:

$$f(\lambda) = \frac{1}{1 + \langle -\lambda/\lambda_0 \rangle}, \quad (2.8)$$

where λ is the confinement strain in each strut obtained by projecting orthogonally to the connection the average strain of the adjacent tetrahedra. The strain tensor in each tetrahedron is computed assuming a linear distribution of displacements and neglecting the effect of particle rotations. The characteristic strain parameter λ_0 determines the sensitivity to confining strain.

In overall the constitutive behavior of each strut is softening for pure tension, tension with shear, and shear with low compression. On the contrary, it is hardening for pure compression and shear with high compression. The tensile and shear response of the connecting strut is sensitive to the lateral confinement in directions orthogonal to the strut. The shear response of the connecting struts exhibits friction and cohesion. The transmission of shear forces between particles is effected without postulating any flexural resistance of the struts. It should be noted that shear transmission is also obtained in the classical lattice model by van Mier [155], but at the cost of treating the lattice struts as beams subject to bending, which is considered unrealistic (not a physical phenomena in the microstructure). A different method of a shear transmission using lattice elements is shown in Sect. 4.3.

An example simulation performed by Cusatis [42] in Fig. 2.20 shows a specimen in a four point shear test (a combination of mode I and mode II). The crack pattern agrees with the experimentally obtained one (Fig. 2.12 a), the load–deflection curve also agrees with the experimental one [42].

Summarizing the confinement shear lattice model [39–43] can realistically simulate the behavior of concrete in different types of deformation. The calculation is performed using a mid–point explicit integration scheme. The advantages of the model are: it is suitable for the failure mode I and II, it can realistically simulate many aspects of the material response such as tensile fracturing, cohesive fracture and size effect or hydrostatic compression. The disadvantages of the model are: it uses a complex macroscopic nonlinear stress–strain relationship to describe a microscopic local behavior, it is suitable only for quasi–static loading. The last drawback is currently being worked on by Cusatis [43] by introduction of inertia forces and creep.

2.5 Summary

The above approaches to model concrete are not the only possibilities. To distinguish them let us introduce a level of material representation L expressed as the average distance between discretization points. Figure 2.21 shows material representation on different scales, as proposed by D’Addetta et al. [46]. Our scale of interest lies between „homogeneity” and „discontinuity” represented by lattice and particle models. At this scale two approaches are possible: a finer discretization ($L \approx 1$ mm) with a simpler microscopic law, and a coarse discretization with a complex nonlinear macroscopic law ($L \approx 10$ mm).

Figure 2.22 shows various discretization methods for $L \in \langle 1, 10 \rangle$ mm, each of them uses the same random distribution of points to allow a better comparison. The DEM (Fig. 2.22 a) uses solely spherical elements. Starting from this basic discretization method the two directions are possible with respect to filling a 3D space: using polygons (polyhedrons) that inherently allow to completely fill the prescribed space or ignoring the problem and using beams (or rods) that merely connect the points. Caballero [27] (Fig. 2.22 b) and Cusatis [42] (Fig. 2.22 c) went the first way using a sophisticated generation method and a complex nonlinear law, which allowed to use $L \approx 10$ mm. An intermediate approach was proposed by Bazant [15], and Beranek [16] to use spheres and beams together (Fig. 2.22 d). In turn van Mier chose the second way using a simple lattice (Fig. 2.22 e). Burt [25] and Berg [17] proposed a more densely connected mesh while using an even simpler law (Fig. 2.22 f).

In general it can be observed that the smaller the L , the simpler law can be used, while $L \approx 10$ mm requires the softening to be introduced in the fracture law, making it more complex. Additionally in terms of the computational efficiency there are two methods: implicit and explicit. Using explicit method is preferred because it eases the calculation parallelization to take advantage of recently popular multi-core processors. In turn, implicit methods are inherently incompatible with distributed methods of calculation which poses additional problems. Because of that, the current research trends indicate that when using a complex law at $L \approx 10$ mm, the law used is explicit [42]. In this aspect there is exactly one advantage: explicit law (computation speed scales well with number of processors) and one disadvantage: the law is complex. To author’s knowledge no one has yet tried an explicit formulation at $L \approx 1$ mm, which would have both advantages: a simple and explicit law.

Moreover a simple law at a smaller scale allows to better understand the underlying mechanisms. The average length of beams is smaller than the smallest aggregate, thus the material behavior is modelled directly by elements used for discretization.

For those reasons the author of this thesis decided to create a model which takes into account a simple fracture law at a smaller level of representation ($L \approx 1$ mm) with an explicit formulation. The new model can be used in multi-scale simulations by coupling it with higher scale models [42, 138].

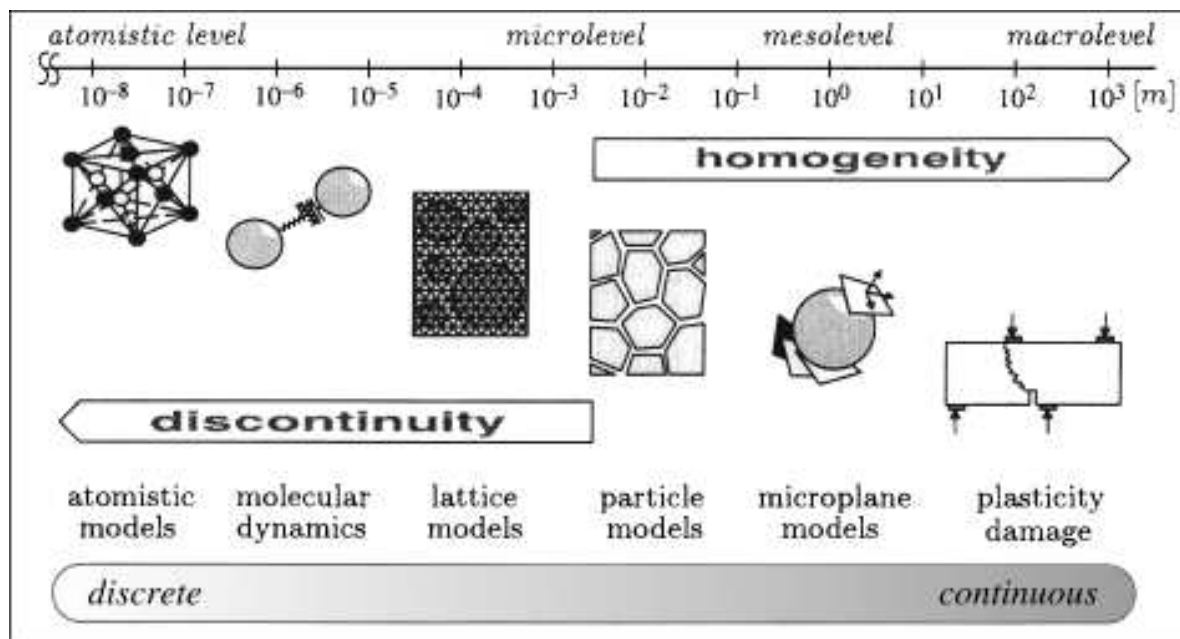


Figure 2.21: Levels of material representation at various scales as presented by D'Addetta [46]

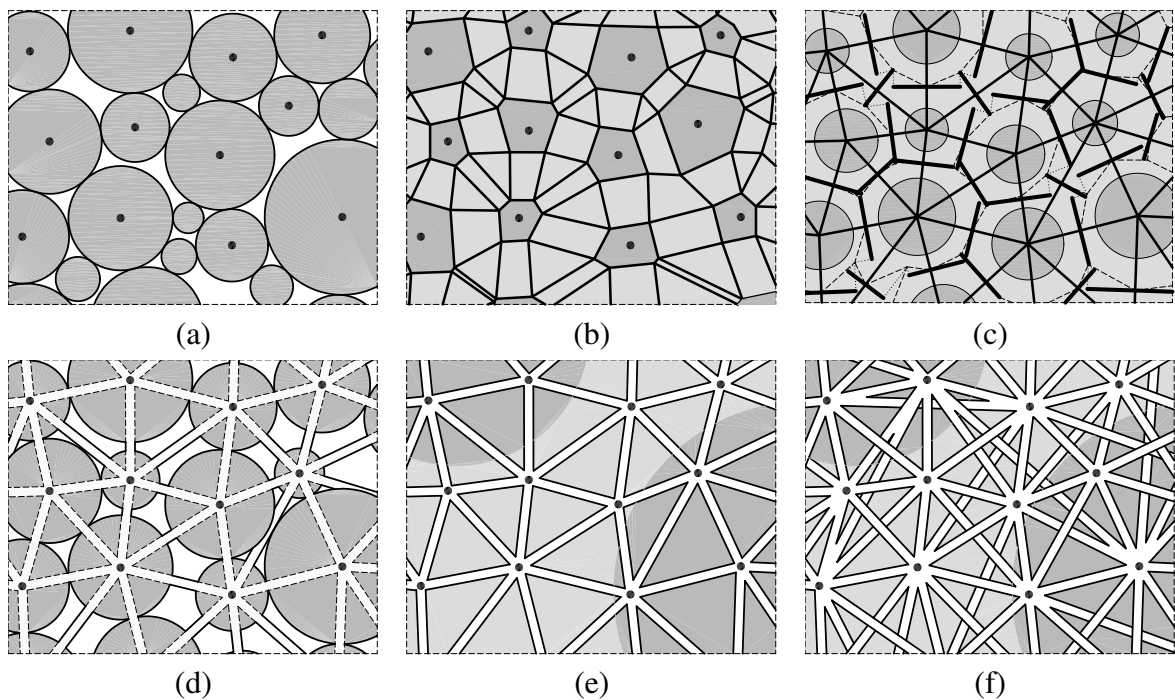


Figure 2.22: Various discretization methods at different material scales using the same random distribution of points (dark gray – aggregate, light gray – matrix or mortar, L – average point distance):

(a) discrete element method (Cundall [38], Donze [54]), $L \approx 10$ mm;

(b) nonlinear interfacial zone model (Caballero et al. [27]), black lines: an interface $L \approx 10$ mm;

(c) confinement-shear lattice model (Cusatis et al. [42]), bold black lines indicate a projected strut cross-section: the interface between two aggregates, $L \approx 10$ mm;

(d) particle-lattice model (Bazant et al. [15], Beranek [16]), $L \approx 5$ mm;

(e) classical lattice-beam fracture model (Herrmann [74], van Mier [154]), beams that cross matrix/aggregate boundary are interface, $L \approx 1$ mm;

(f) random lattice-beam fracture model (Burt et al. [25], Berg et al. [17]), $L \approx 1$ mm;

Chapter 3

Experiments using digital image correlation technique DIC

This chapter presents results of experimental investigations of strain localization in concrete using a non-destructive method called Digital Image Correlation (DIC) technique. This technique measures surface displacements from digital images. The model tests were carried out with notched concrete specimens under three-point bending. 3 different beam sizes and 2 different concrete mixes were used. During experiments, load-deflection curves and evolution of fracture process zone were determined. The measured size effect in strength was compared with the size effect law by Bazant [6] and experimental results by Le Bellégo [98].

3.1 Introduction

Strain localization in the form of fracture process zones (FPZ) is a fundamental phenomenon in quasi-brittle materials like concrete [8, 13, 157]. The determination of the width and spacing of strain localization is crucial to evaluate the material strength at peak and in the post-peak regime. To properly describe strain localization (width and spacing), continuum constitutive models should include a characteristic length of micro-structure [8]. There are several approaches within continuum mechanics to include a characteristic length and to regularize the ill-posedness of the underlying incremental boundary value problem [47] caused by strain-softening material behavior and localization formation (differential equations of motion do not change their elliptic type during quasi-static calculations and hyperbolic type during dynamic calculations), and to avoid a pathological mesh-sensitivity of numerical solutions for quasi-brittle materials as: second-gradient [33, 115, 118], non-local [8, 20, 122] and viscous ones [144, 145]. Owing to them, objective and properly convergent numerical solutions for localized deformation (mesh-insensitive load-displacement diagram and mesh-insensitive deformation pattern) are achieved. Otherwise, FE-results are completely controlled by the size and orientation of the mesh and thus produce unreliable results, i.e. strain localization becomes narrower upon mesh refinement (element size becomes the characteristic length) and computed force-displacement curves change considerably depending on the width of the calculated localization. In addition, a premature divergence of incremental FE-calculations is often met. The presence of a characteristic length allows also to take into account microscopic inhomogeneities triggering shear localization (e.g. aggregate size) and to capture a deterministic size effect of a specimen (dependence of strength and other mechan-

ical properties on the size of the specimen) observed experimentally on softening granular and brittle specimens [8]. This is made possible since the ratio l_c/D governs the response of the model (l_c is a characteristic length, D is the specimen size). The characteristic length in brittle materials is usually related to the aggregate size. According to [122] and [11], it is approximately $3 \times d_a$, where d_a is the maximum aggregate size. It is usually determined with an inverse identification process of experimental [66, 102] since it cannot be directly measured. Recently Le Bellégo [98] presented a calibration method of non-local models containing a characteristic length on the basis of 3 size effect bending tests. However, such a method is not consistent [22]. In general, the determination of one representative characteristic length of micro-structure is very complex in concrete since strain localization can include a mixed mode (cracks and shear zones [10]), a characteristic length is one-dimensional but is related to the fracture process zone with a certain area or volume [10] which increases during deformation (e.g. on the basis of acoustic emission measurements by [123]). In turn, other researchers conclude that the characteristic length is not a constant, and it depends on the type of the boundary value problem and the current level of damage [62]. The only way to determine a characteristic length are simultaneous measurements of load-displacement curves and widths of the fracture process zones in experiments with the same concrete for different boundary value problems and specimen sizes [22].

This chapter shows a capability of a non-invasive method called Digital Image Correlation technique [18] (called also Particle Image Velocimetry (PIV)) to measure directly surface displacements in model concrete beams subject to three-point bending. The technique which was originally developed in the field of experimental fluid and gas mechanics [126] has been nowadays frequently used in granular materials [76, 107, 112, 127, 143, 162]. The deformation quantities is, however, significantly larger than these in brittle materials.

The model tests were carried out with notched concrete specimens subject to three-point bending. Three different beam sizes and two different concrete mixes were used. During experiments, load-deflection curves and evolution of fracture process zone were determined. The measured size effect results were compared with the size effect law by [6] and experimental results by [98].

This chapter is organized as follows. In Sect. 3.2, the DIC technique is described in detail. Section 3.3 discusses the accuracy of the method. The information about the experimental setup is given in Sect. 3.4. The experimental results are depicted in Sect. 3.5. Finally, some conclusions are given in Sect. 6.

To use DIC, the software written in C++ by the author of this thesis was applied.

3.2 Digital Image Correlation technique

DIC is a powerful optical surface velocity-measuring tool to visualize two dimensional deformations. The colored surface serves as tracers. A high resolution monitoring of different deformations can be obtained by processing successive digital images taken with a constant time between frames from a usual digital camera. Owing to that strain localization can be visualized and analysed in detail. Digital camera sensors are comprised of tiny, light-sensitive elements called pixels.

When an image is captured, each pixel reflects three numbers (called the color components $YCbCr$) in proportion to the amount of the light reflected from the imaged object, where Y is the luminosity (brightness), C_b is the blueness and C_r is the redness. The DIC

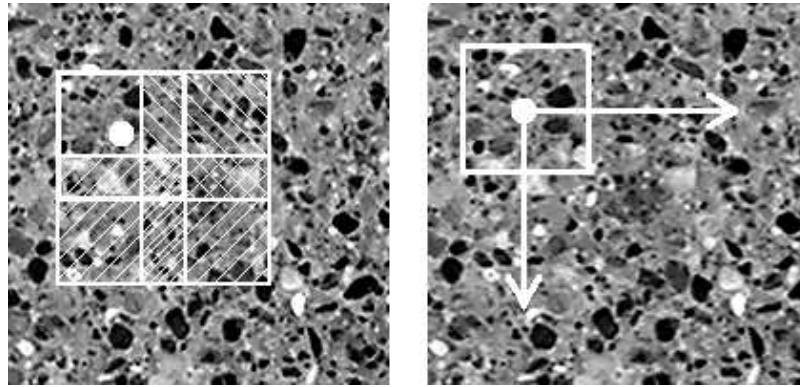


Figure 3.1: DIC analysis: two digital images with the marked square pixel subset (white square) and other overlapping pixel subsets (hashed squares)

system interprets differences in light intensity as a gray-scale pattern recorded at each pixel on CCD-camera (Charge Coupled Device). Three functions are of a major importance for DIC: image field intensity, cross-correlation function and interpolation function. The image intensity field assigns to each point in the image plane a scalar value which reflects the light intensity (Y -color component) of the corresponding point in the physical space. The gray levels range numerically between 0 (black) and 255 (white) for an 8-bit image. A so-called area of interest (AOI) is cut out of the digital image and small overlapping sub-areas called pixel subsets are chosen (Fig. 3.1). Each cell comprises a particular number of square sub-sets of digital pixels from 15×15 pixels up to 240×240 pixels. If the deformation between two images is sufficiently small, the patterns of the interrogation cells are supposed not to change their characteristics (only their locations).

A deformation pattern is detected by comparing two consecutive images captured by a camera which remains in a fixed position with its axis oriented perpendicular to the plane of deformation. To find a local displacement between images 1 and 2, a search zone is extracted from the second image. A correct local displacement vector for each interrogation cell is accomplished by means of a cross-correlation function between two consecutive brightness distributions (Y color component) in two digital images. The function calculates simply possible displacements by correlating all gray values from the first image with all gray values from the second image. The correlation plane is evaluated at single pixel intervals, this means that the resolution is equal to one pixel. By fitting an interpolation function to the region close to the peak, the displacement vector is established with a high accuracy (equal to the correlation offset). The peak in the correlation function indicates that the two images are overlaying each other (thus, it indicates the „degree of match” between two images). The Charged Coupled Device operates in the color space $YCbCr$, as opposed to the widely known color space Red Green Blue RGB (R is the red, G is the green and B is the blue color component). The first step in the image preprocessing is to convert the RGB color space into the $YCbCr$ color space according to the ISO/TC42N 4378 TIFF/EP standard, assuming $R, G, B \in \langle 0, 255 \rangle$ and $Y, C_b, C_r \in \langle 0, 255 \rangle$:

$$\begin{aligned} Y &= 0.2989 \cdot R + 0.5866 \cdot G + 0.1145 \cdot B, \\ C_b &= 128 - 0.168736 \cdot R - 0.331264 \cdot G + 0.5 \cdot B, \\ C_r &= 128 + 0.5 \cdot R - 0.418688 \cdot G - 0.081312 \cdot B. \end{aligned} \quad (3.1)$$

To calculate a strain field on the specimen surface, two successive digital images have to

be compared with each other. First, one pixel is selected on the first digital image, then a square pixel subset of a certain size is chosen around it. Next a search for this pixel subset on the second image is done by using a correlation function (Fig. 3.1). Two different correlation functions were used: first using the Spearman rank correlation coefficient R_1 [18] and the Pearson's product-moment correlation coefficient R_2 . The first method is faster but less stable during calculations with large pixel subset sizes:

$$R_1(x, y) = \frac{(\sum x_i y_i)^2}{\sum x_i^2 \sum y_i^2}. \quad (3.2)$$

The parameters x and y refer to the first and second image, respectively and i stands for the number of the pixel. In turn, the Pearson's product-moment correlation method produces numerically more stable results although it needs more time (ca. 20% more) for calculations. It is obtained by dividing the covariance of two variables by the product of their standard deviations (n - number of pixels in the pixel subset):

$$R_2(x, y) = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}}. \quad (3.3)$$

Equation 3.3 was implemented using following algorithm (written in the pseudocode):

```

sum_sq_x = 0
sum_sq_y = 0
sum_coproduct = 0
mean_x = x[1]
mean_y = y[1]
for i from 2 to N:
    sweep = (i - 1.0) / i
    delta_x = x[i] - mean_x
    delta_y = y[i] - mean_y
    sum_sq_x += delta_x * delta_x * sweep
    sum_sq_y += delta_y * delta_y * sweep
    sum_coproduct += delta_x * delta_y * sweep
    mean_x += delta_x / i
    mean_y += delta_y / i
pop_sd_x = sqrt( sum_sq_x / N )
pop_sd_y = sqrt( sum_sq_y / N )
cov_x_y = sum_coproduct / N
correlation = cov_x_y / (pop_sd_x * pop_sd_y).

```

This algorithm¹ proved to be approximately ten times faster than the calculations with a straightforward implementation of the Pearson's formula (Eq. 3.3). It does not include numerical instabilities when a huge amount of correlation points is large, or their scalar values are big. The 8 byte float precision type was used in calculations since it turned out to be the fastest. Figure 3.2 shows a comparison between two different correlation functions and pixel subset sizes. The Pearson's method with the subset of 200 pixels had smallest amount of noise but also a less steep peak. A too small subset produced noise. In turn, the Spearman's method generated too small correlation peak with respect to the correlation plane.

¹see <http://en.wikipedia.org/wiki/Correlation>

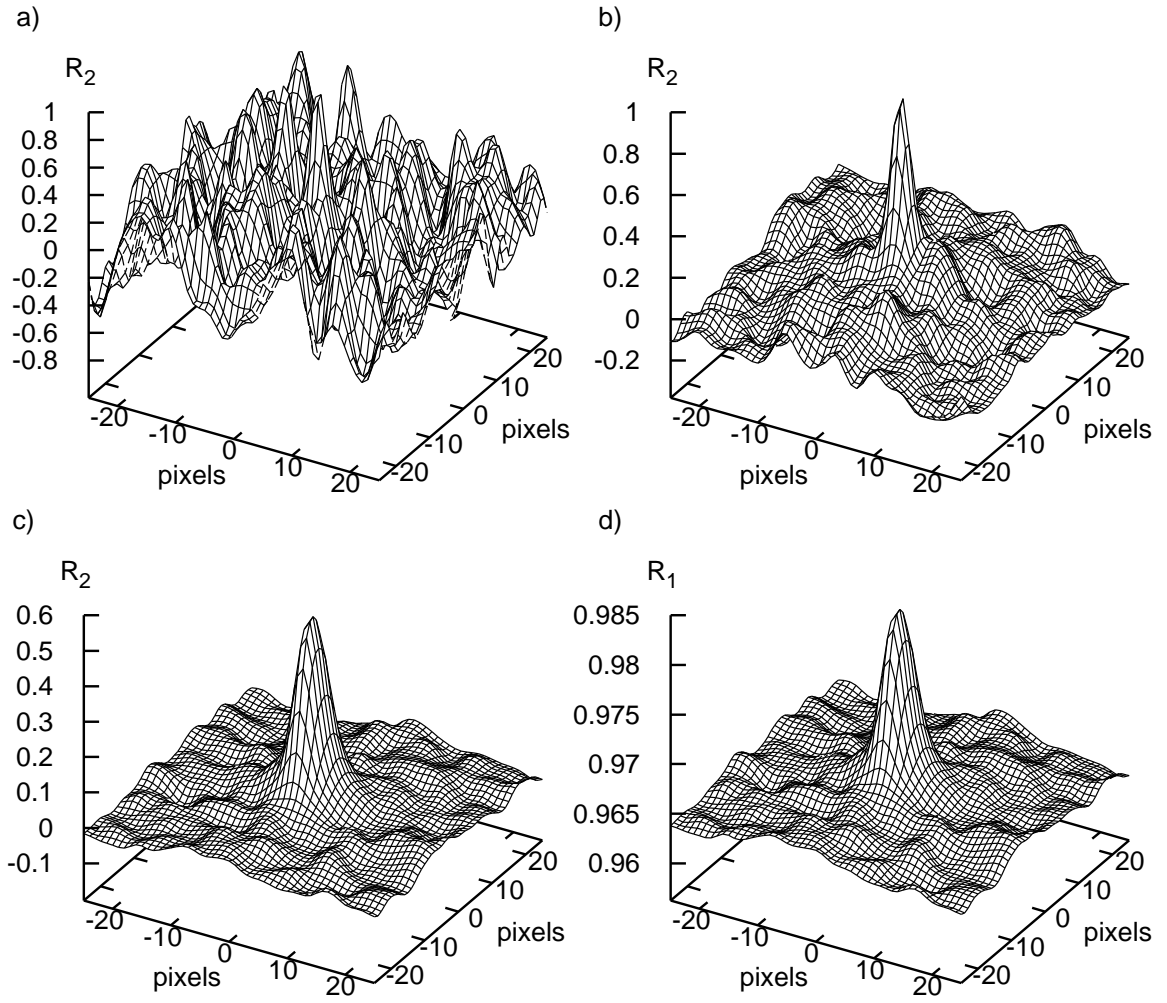


Figure 3.2: Comparison between the correlation functions and pixel subset sizes: (a) pixel subset of 10 pixels, Pearson's formula (Eq. 3.3); (b) pixel subset of 50 pixels, Pearson's formula (Eq. 3.3); (c) pixel subset of 200 pixels, Pearson's formula (Eq. 3.3); (d) pixel subset of 200 pixels, Spearman's formula (Eq. 3.2)

The top of the correlation peak is usually not clearly distinguished as the correlation function generates results for each pixel of the image. To determine the top of the peak, a sub-pixel interpolation has to be performed. Figure 3.3 a,b shows a correlation plane used for interpolating the correlation peak for 2 different interpolation functions [50].

The first one was the spline36 interpolation (Fig. 3.4) which used the 6×6 pixel image area (Fig. 3.3 b) to calculate the interpolated value:

$$w_1(a) = \begin{cases} \left(\left(\frac{13}{11}a - \frac{453}{209} \right) a - \frac{3}{209} \right) a + 1 & \text{for } 0 < a < 1, \\ \left(\left(-\frac{6}{11}(a-1) + \frac{270}{209} \right) (a-1) - \frac{156}{209} \right) (a-1) & \text{for } 1 < a < 2, \\ \left(\left(\frac{1}{11}(a-2) - \frac{45}{209} \right) (a-2) + \frac{26}{209} \right) (a-2) & \text{for } 2 < a < 3. \end{cases} \quad (3.4)$$

The second interpolation function was sinc256 (Figs. 3.3 a and 3.4) which used 16×16

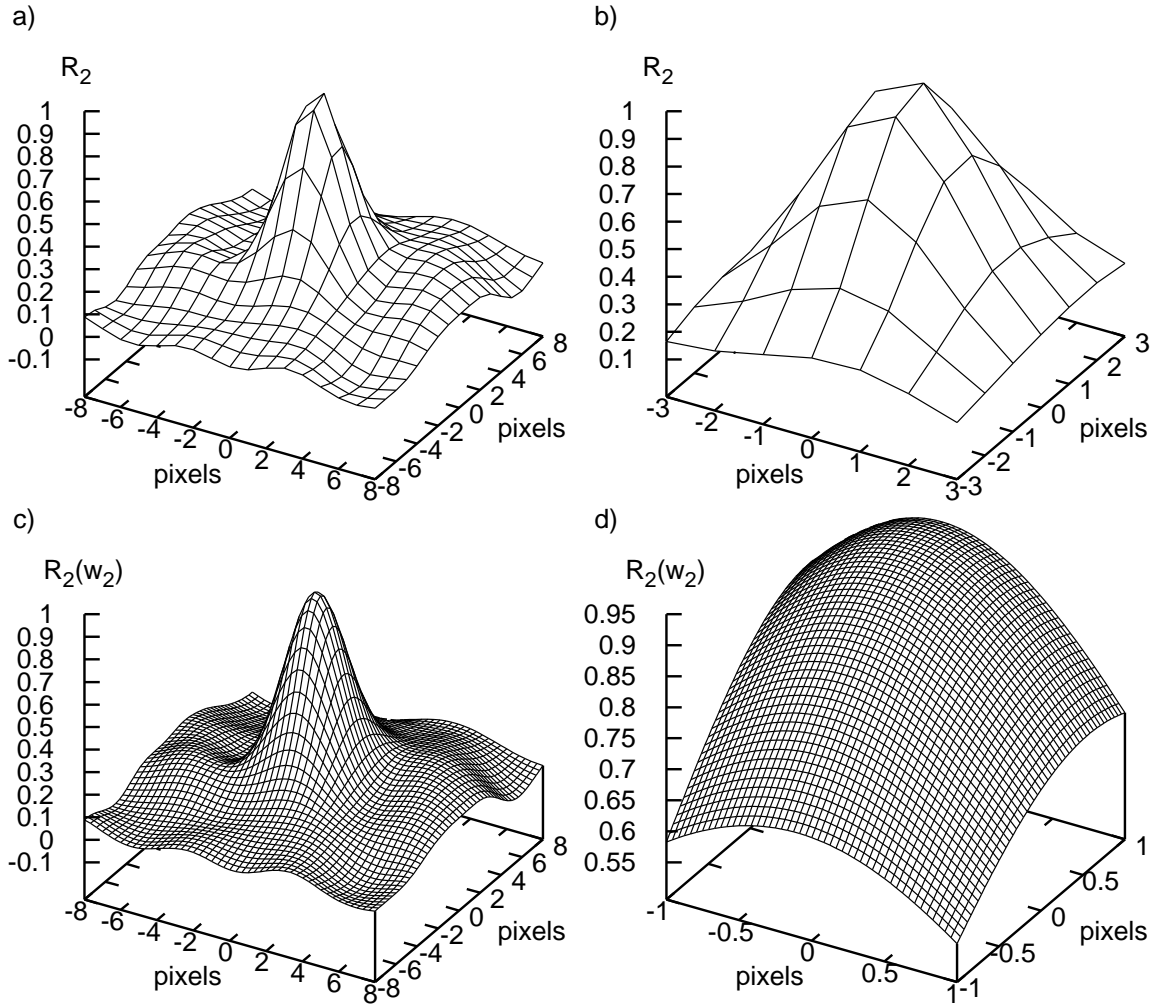


Figure 3.3: Correlation plane with correlation peak: (a) 16×16 pixel area used in interpolation by *sinc256*; (b) 6×6 pixel area used in interpolation by *spline36*; (c) surface after sub-pixel interpolation; (d) zoom on the peak after sub-pixel interpolation

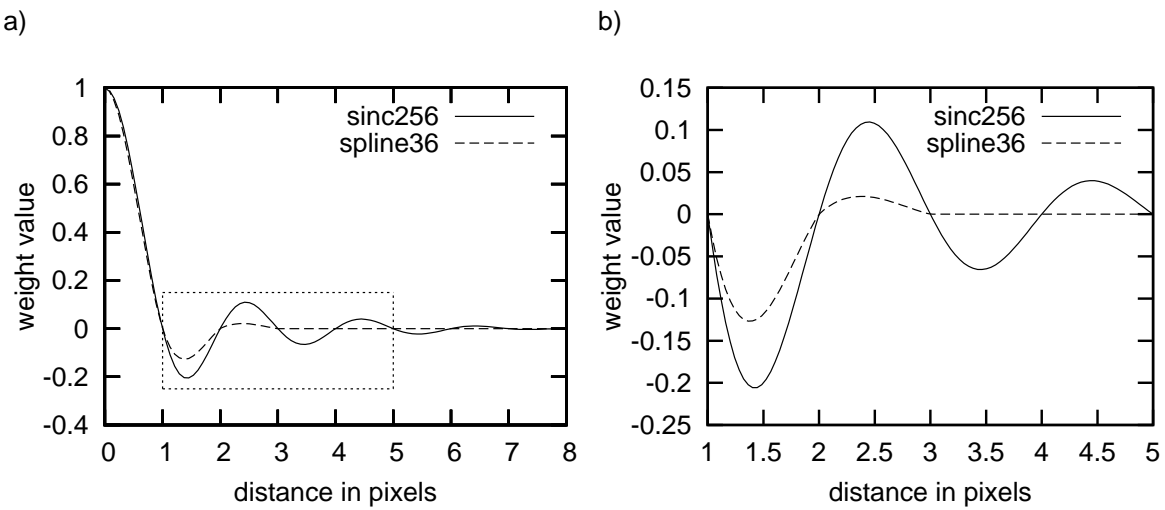


Figure 3.4: Interpolation functions used to achieve a sub-pixel precision

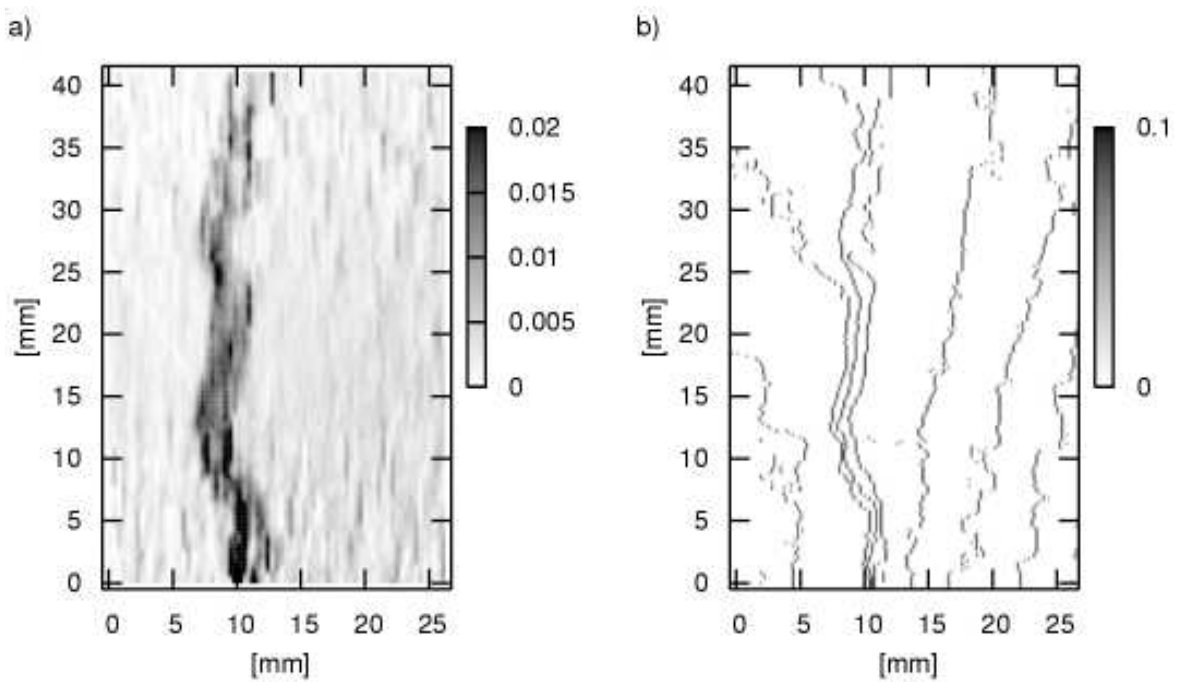


Figure 3.5: Effect of sub-pixel interpolation on the strain field in the concrete specimen: (a) with sub-pixel interpolation; (b) without sub-pixel interpolation

pixel image area and was slower (by ca. 10%) than the first one:

$$w_2(a) = \begin{cases} 1 & \text{for } a = 0, \\ \frac{\sin(\pi a)}{\pi a} \cdot \frac{\sin(\frac{\pi a}{8})}{\frac{\pi a}{8}} & \text{for } a > 0. \end{cases} \quad (3.5)$$

Generally, the interpolation was evaluated at the 1/500th pixel intervals, yielding a system resolution of 0.002 pixels (Fig. 3.3 c,d). Figure 3.5 shows the difference between the obtained results of strain fields with and without a sub-pixel interpolation. The results demonstrate that the application of a sub-pixel interpolation increases the accuracy of calculations.

3.3 Precision of DIC measurements

A series of initial experiments was conducted to assess the precision of DIC. Figure 3.7 shows the standard deviation of the measured strain against the pixel subset size for two different correlation and two interpolation functions. For this purpose, two successive digital images of the concrete specimen were compared wherein the resulting strain was zero. Figure 3.7 e,f demonstrates that using Y component is crucial, because using a typical „gray color” (the matching component calculated as $(R + G + B)/3$, or R alone) approach yields worse results. The most accurate results were obtained with the Pearson’s method used in combination with the interpolation function sinc256 and Y color component. Therefore, all further calculations were performed using this method.

Figures 3.8 and 3.9 show the results for following digital image pairs (as shown on Fig. 3.6):

- a) two subsequent images of the concrete specimen,
- b) random image with 3×3 „grains”,
- c) single image of the concrete specimen compared with itself,
- d) random image with 1×1 „grains”.

For those four cases, Fig. 3.8 shows that the DIC precision is better than 0.0005 for the pixel subsets greater than 60 pixels in all cases.

The precision errors are associated with the asymmetry of the correlation peak. In the case of the random image with 1×1 pixel „grains”, an almost perfect match was found (Fig. 3.8 d) with the highest precision. For 3×3 pixel „grains” and two subsequent images (Fig. 3.8 a,b) no perfect match was found, since the correlation peak was asymmetric. The strain value fluctuations along a single line of the image is shown on Fig. 3.10 for the point 'A' of Fig. 3.8.

Next, the second image for all four cases (Fig. 3.6) was artificially modified by stretching 300 pixel stripes into 301 pixels, so strain oscillated between 0 and 0.0033 every 300 pixels. Figure 3.9 reveals a higher precision for the real case (Fig. 3.6 a,c) than for the random pixel pattern (Fig. 3.6 b,d). In the randomly generated pixel pattern the noise has high amplitude in the frequency domain thus by stretching it (to achieve strain 0.0033), an image interference occurs. The real images have small amplitude in the frequency domain thus no artefacts are generated. Moreover, a large pixel subset size blurs the boundaries between oscillating strain values while a smaller pixel subset size (but not too small) yields more accurate results. Figure 3.11 and 3.12 clearly show this outcome for points 'B' and 'C' of Fig. 3.9.

The precision of DIC strongly depends on the pixel subset size and the quality of speckle pattern in the image. It is recommended that the speckle pattern dots on the concrete surface should include 4 – 10 pixels on the captured image. Thus, no image interference takes place.

The selection of an optimum pixel subset size in the DIC analysis requires two conflicting parameters to be balanced. The larger pixel subsets offer an improved accuracy (Fig. 3.8) but they blur the boundary between strain values (Fig. 3.12). The number of measurement points depends only on the computational time since the distance between the pixel subset centers can be as small as one pixel. In the calculations presented here, this distance was assumed to lie between 10 and 40 pixels. So it was small enough to not influence the resolution of results.

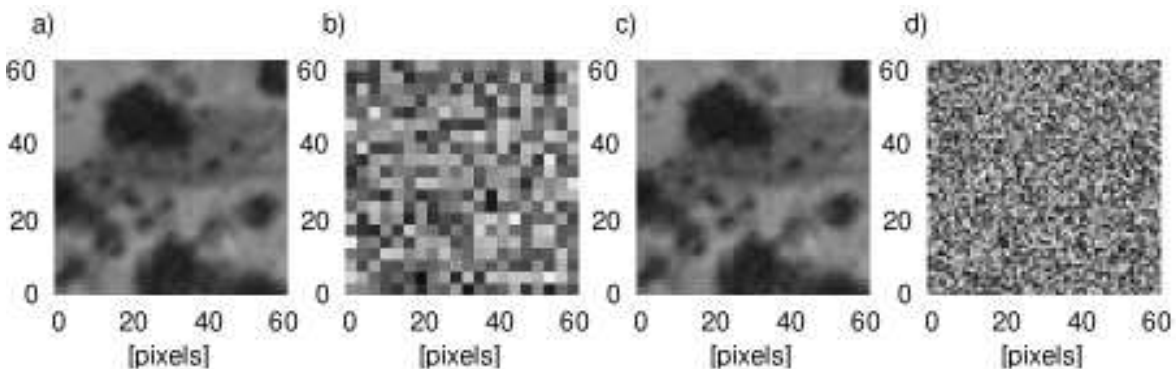


Figure 3.6: Images used in validation experiments: (a) two subsequent images; (b) random image with 3×3 „grains”; (c) single image; (d) random image with 1×1 „grains”.

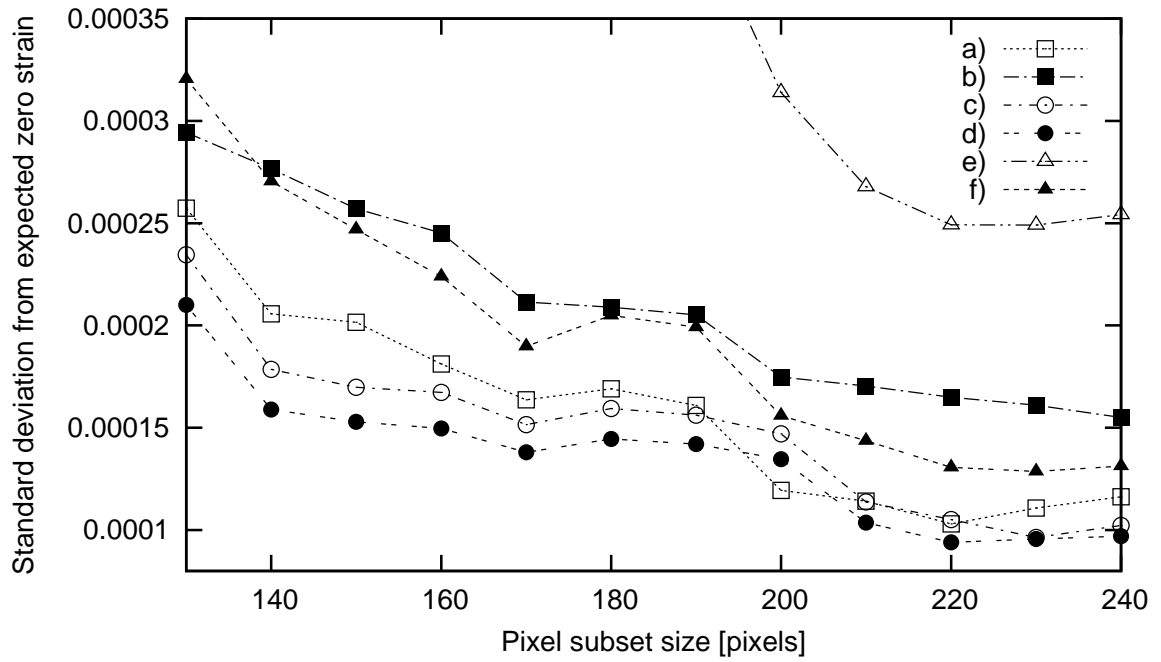


Figure 3.7: Effect of two correlation and two interpolation functions on the standard deviation: (a) Spearman's method with spline36 interpolation; (b) Spearman's method with sinc256 interpolation; (c) Pearson's method with spline36 interpolation; (d) Pearson's method with sinc256 interpolation; (e) Pearson's method with sinc256 interpolation, R component from RGB used instead of Y component; (f) Pearson's method with sinc256 interpolation, $(R + G + B)/3$ component from RGB used instead of Y component

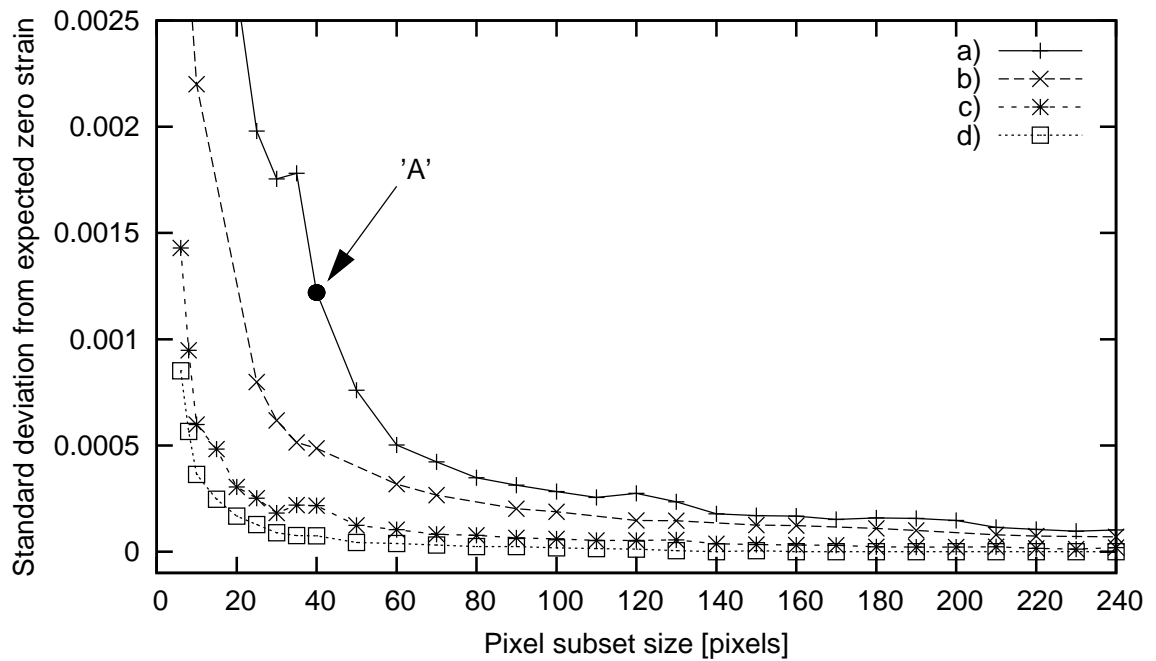


Figure 3.8: DIC precision against the pixel subset size (for strain equal to zero): (a) two subsequent images; (b) random image with 3×3 „grains”; (c) single image; (d) random image with 1×1 „grains”

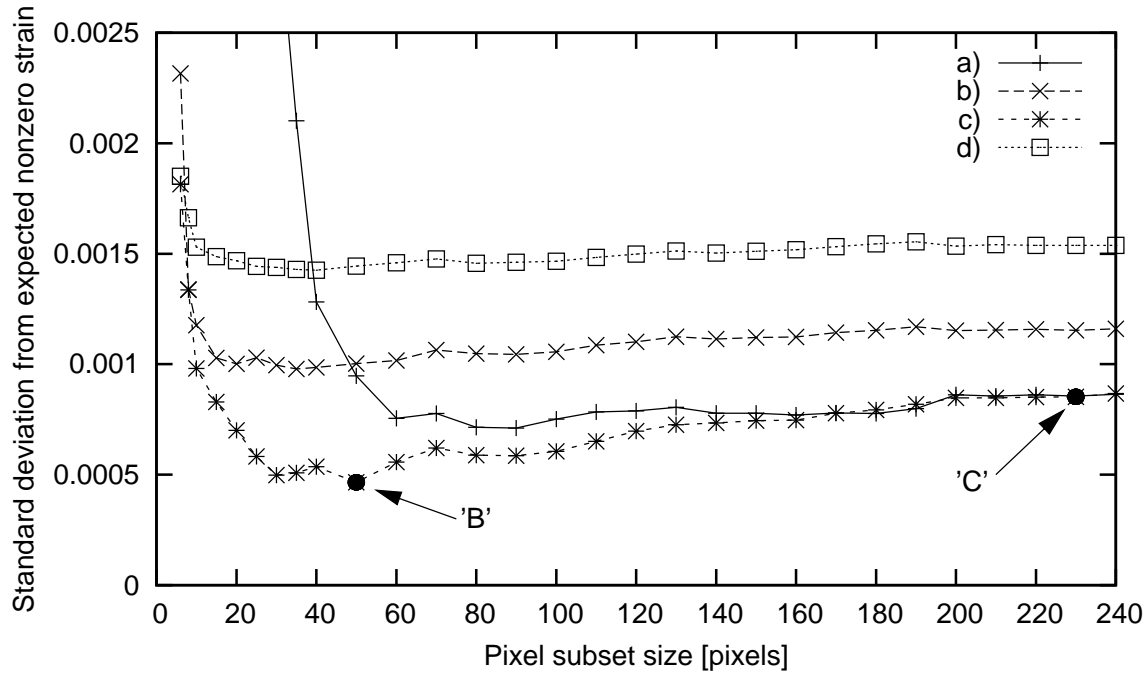


Figure 3.9: DIC precision against the pixel subset size for strain oscillating between 0 and 0.0033 every 300 pixels

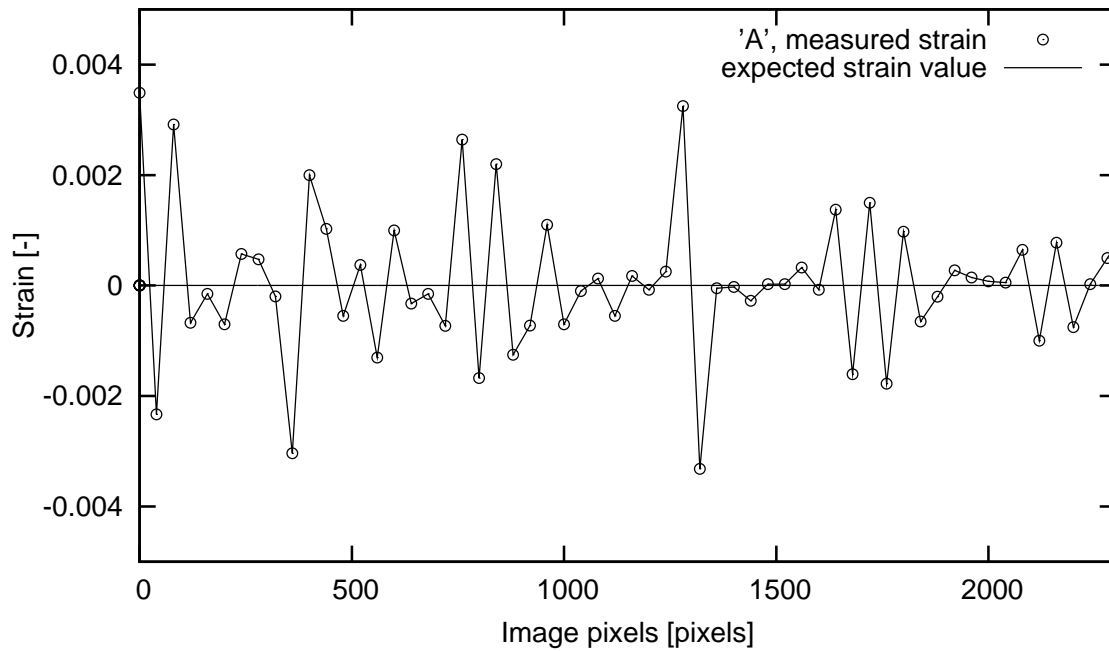


Figure 3.10: Strain fluctuations along single line of the image for the point 'A' of Fig. 3.8 (pixel subset size has 40 pixels, standard deviation from expected zero strain is 0.0012)

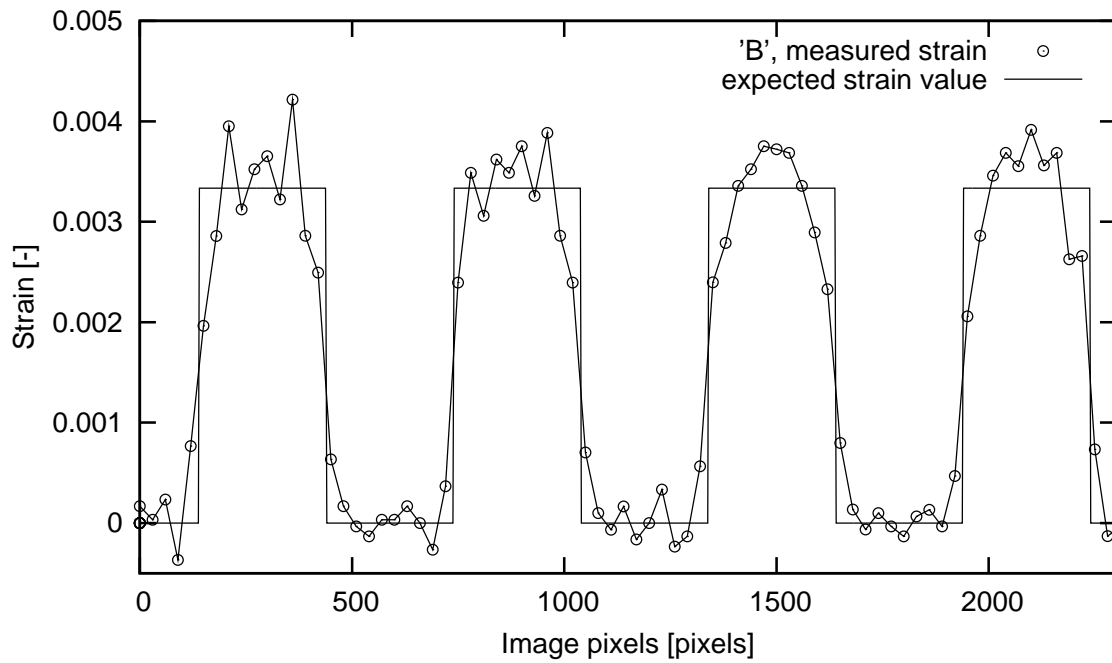


Figure 3.11: Strain fluctuations along single line of the image for the point 'B' of Fig. 3.9 (pixel subset size has 50 pixels, width of the localized zone is estimated with high precision, standard deviation from expected strain is 0.0005)

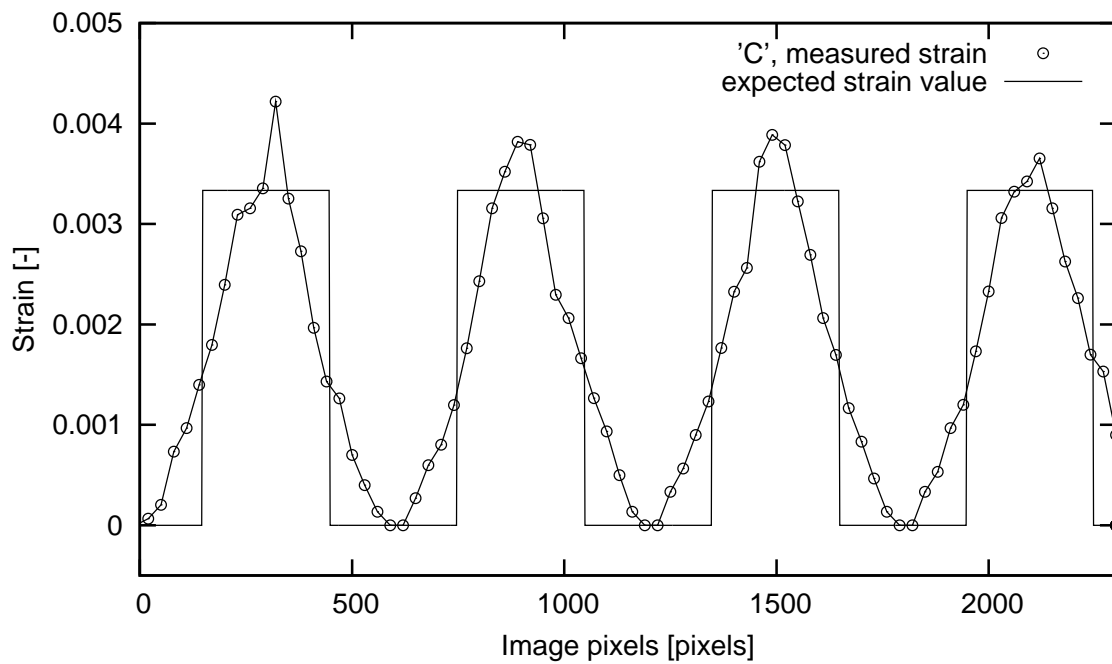


Figure 3.12: Strain fluctuations along single line of the image for the point 'C' of Fig. 3.9 (pixel subset size has 230 pixels, boundaries of localized zone are blurred, standard deviation from expected strain is 0.0009)

3.4 Experimental setup

The model tests were performed with 12 notched concrete beams of different sizes under three-point bending (with free ends). Figure 3.13 shows the geometry of 3 concrete beam specimens with the thickness of 40 mm (the same geometry was assumed in laboratory experiments by [98]). The beam depths D were 80 mm (small size beam), 160 mm (medium size beam) and 320 mm (large size beam), respectively. A notch of depth $D/10$ mm and thickness of 3 mm was placed in the middle of each beam bottom. The deformation in the beams was induced by a vertical displacement prescribed at the mid-span with a rate of 0.01 mm/min. The entire experiment with a single beam lasted approximately 1 hour.

The specimens were made with two different mixes (Tab. 3.1) consisting of ordinary Portland cement, water and sand (mean aggregate diameter $d_{50} = 0.5$ mm) or gravel (mean aggregate diameter $d_{50} = 2.0$ mm). Figure 3.14 shows the grading curves for sand and gravel used for both mixes. 9 beams were made of sand concrete and 3 of gravel concrete. The uniaxial compression strength of beams measured on $10 \times 10 \times 10$ cm³ specimens was $f_c = 64 \pm 3$ MPa (sand concrete) and $f_c = 52 \pm 3$ MPa (gravel concrete). In experiments, the usual digital camera Nikon 8400 ED (8MP) was applied with the image resolution of 3264×2448 pixels (Fig. 3.15 a). The camera was mounted on a tripod with additional fixtures to ensure a stable position. During experiments, load–deflection curves were registered and the fracture zone width was measured with an induction displacement gauge placed above the notch (Fig. 3.15 b). First, the concrete beams were polished to achieve a smooth surface. Then, a speckle pattern was put on the surface using two color sprays: black and yellow. The images were shot once per minute.

3.5 Experimental results

Figure 3.16 shows some beams after failure with a single crack propagated from the notch up to the top. The typical load–deflection curves for sand and gravel concrete beams are presented in Fig. 3.17. In turn, Fig. 3.19 shows the measured size effect for all experiments with sand concrete beams compared with the size effect law by Bazant [6] and experimental results by Le Bellégo [98]. In the case of gravel concrete beams, due to only 3 performed tests, the measured size effect was not shown.

A pronounced size effect was observed in experiments (the larger the beam, the smaller

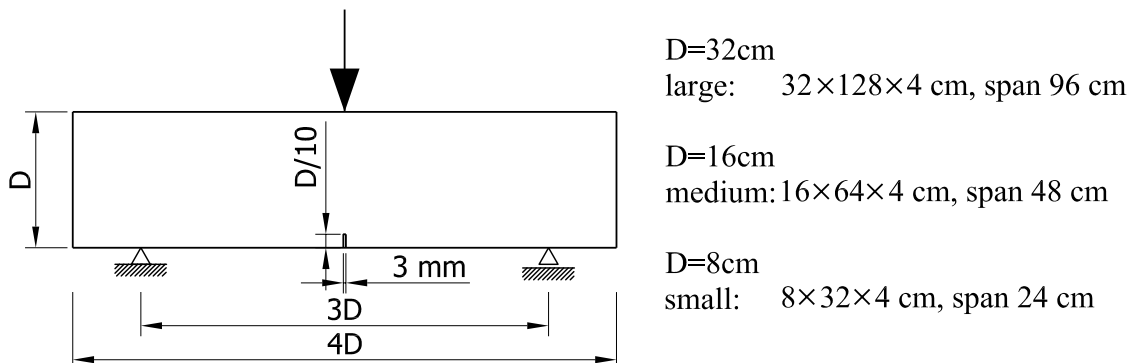


Figure 3.13: Geometry of concrete specimens subjected to three-point bending

Table 3.1: Receipts for concrete used in experiments: (a) with sand; (b) with gravel

(a)			(b)		
No.	Material	Volume	No.	Material	Volume
1.	Sand	62%	1.	Gravel	76%
2.	Cement	27%	2.	Cement	17%
3.	Water	11%	3.	Water	7%

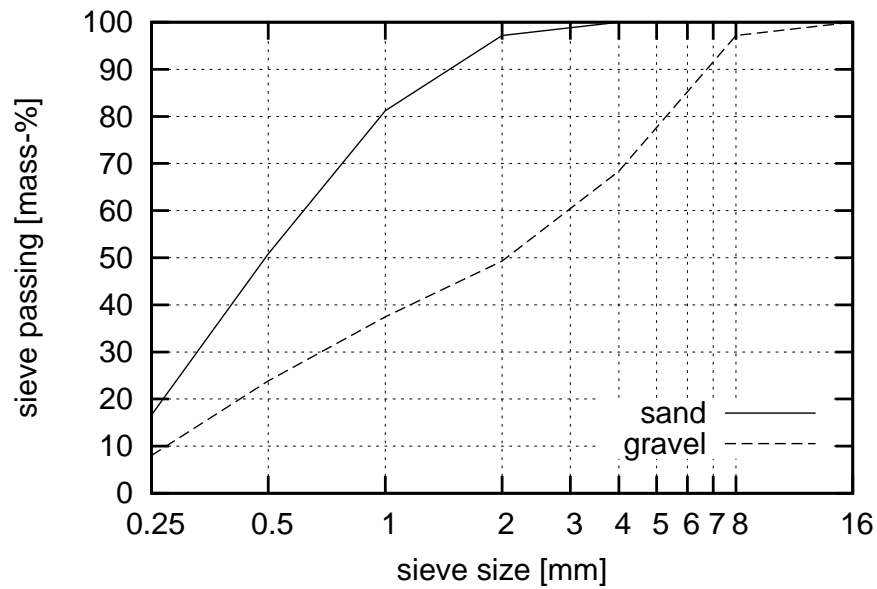


Figure 3.14: Grading curve for sand and gravel used for concrete



a)



b)

Figure 3.15: Instrumentation used in the experimental tests: (a) Nikon 8400 digital camera; (b) induction displacement gauge

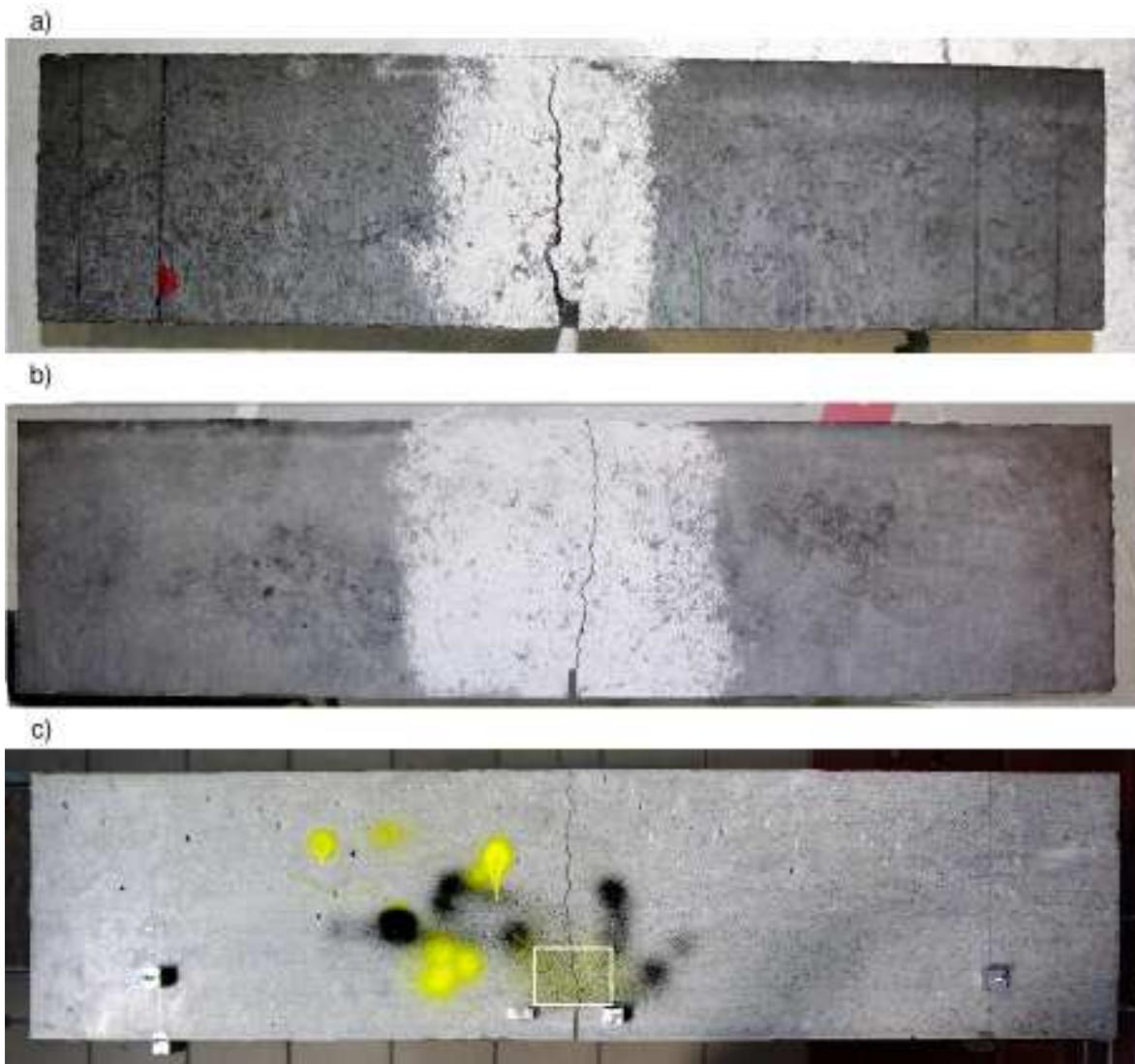


Figure 3.16: Beams after failure: (a) small size beam (back side); (b) medium size beam (back side); (c) large size beam (front side with the black-yellow speckle pattern inside of the image area marked by white rectangle).

the strength). The material ductility increased with decreasing beam size. The results are in satisfactory agreement with the size effect law [6] which is valid for structures with pre-existing notches or large cracks:

$$\sigma = \frac{B f_t}{\sqrt{1 + \frac{D}{D_0}}}, \quad (3.6)$$

where σ – nominal strength, f_t – tensile strength, B – dimensionless geometry-dependent parameter which depends on the geometry of the structure and of the crack, D – specimen size and D_0 – size-dependent parameter called transitional size. To find the parameters B and D_0 from FE-analysis, a nonlinear least-squares (NLLS) Marquardt-Levenberg algorithm was used. In spite of a small amount of sand concrete specimens (9), a satisfactory agreement with the Bazant's size effect law was obtained (Fig. 3.19 a). The measured data match also the experimental results by Le Bellégo [98] (Fig. 3.19 b).

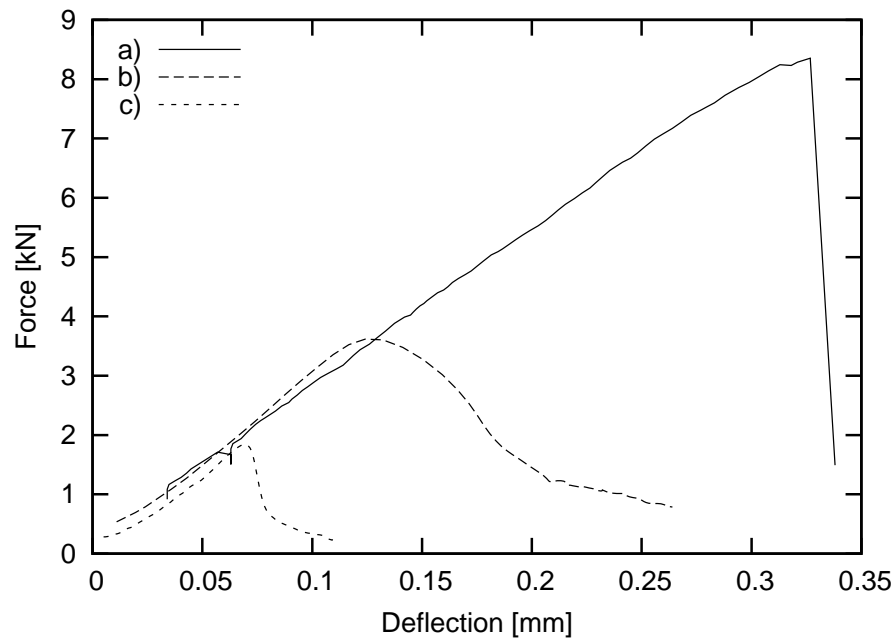


Figure 3.17: Vertical force at mid-span versus beam deflection (sand concrete): (a) large size beam; (b) medium size beam; (c) small size beam

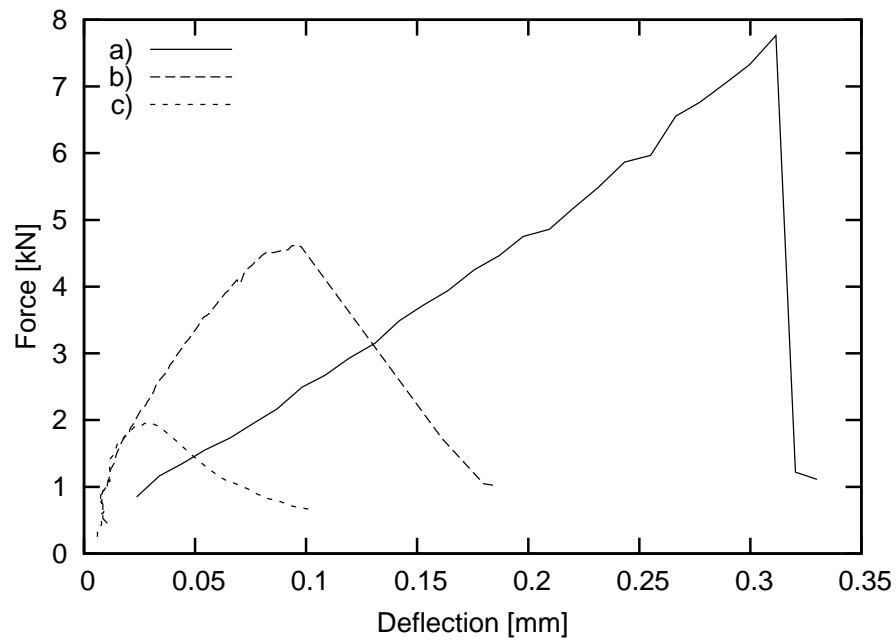


Figure 3.18: Vertical force at mid-span versus beam deflection (gravel concrete): (a) large size beam; (b) medium size beam; (c) small size beam

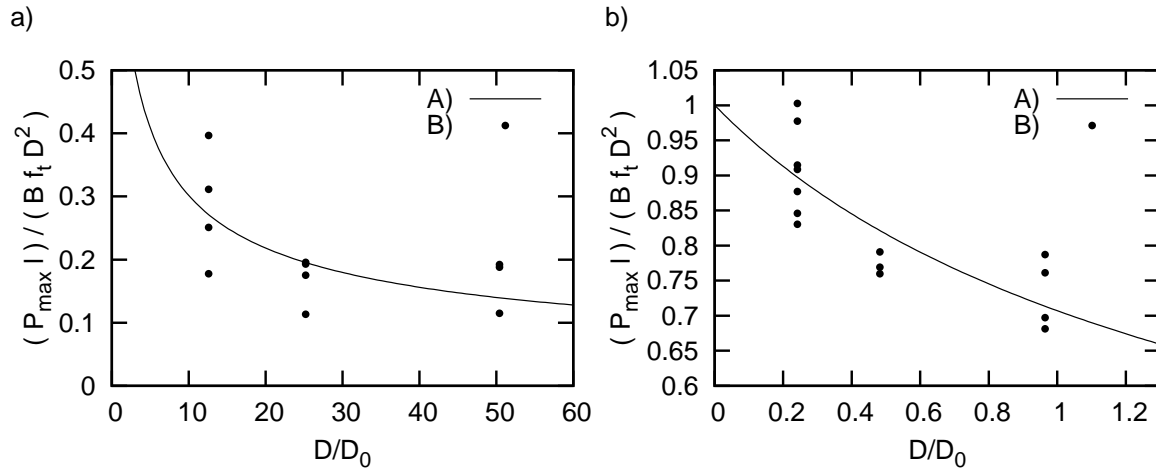


Figure 3.19: Size effect in three-point bending tests (D is beam height, l is beam span, f_t is tensile strength, D_0 and B are parameters): (a) experimental results with sand concrete (B) compared with the size effect law by Bazant (1984) (A) ($D_0 = 0.0063$, $B = 81.89$); (b) experimental results by Le Bellego et al. (2003) (B) compared with the size effect law (A) ($D_0 = 0.332$, $B = 23.93$)

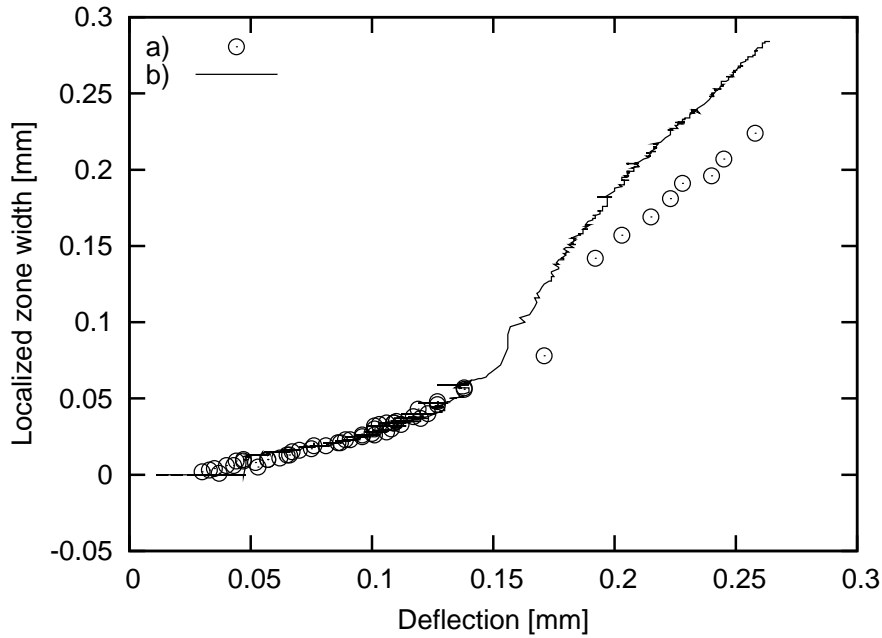


Figure 3.20: Width of localized zone in medium size beam (sand concrete) measured: (a) by DIC; (b) by induction displacement gauge (Eq. 3.7)

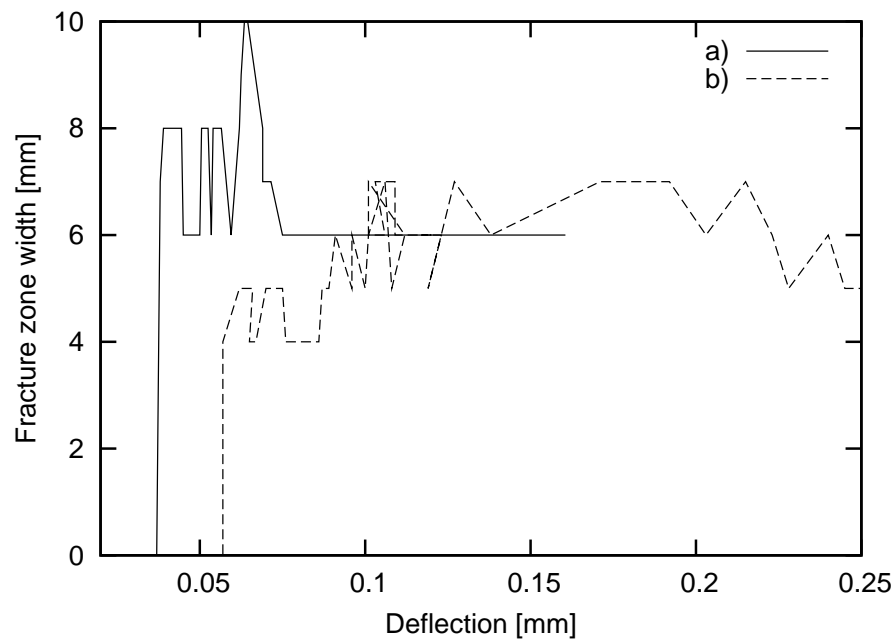


Figure 3.21: Evolution of width of fracture process zone during the experiment: (a) medium size beam (gravel concrete); (b) medium size beam (sand concrete)

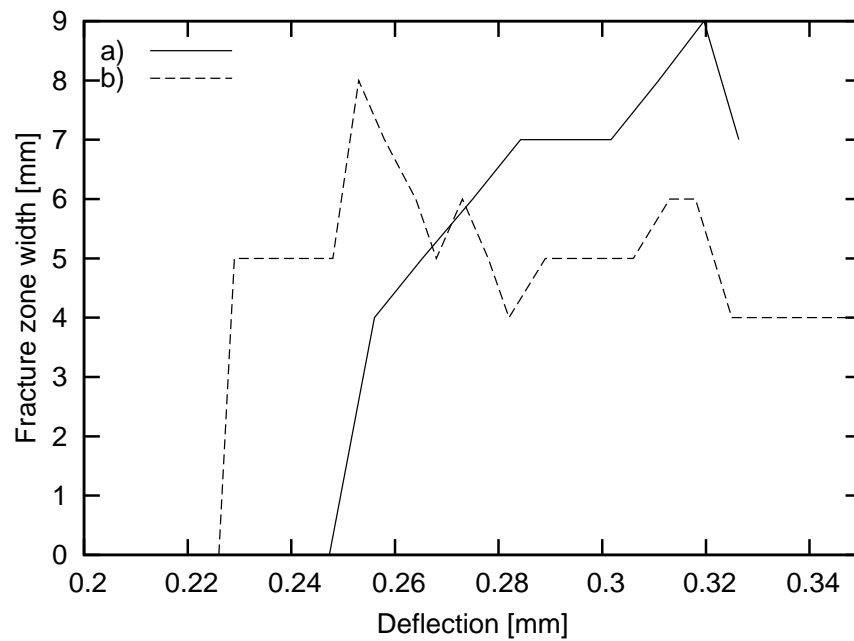


Figure 3.22: Evolution of width fracture zone during the experiment: (a) large size beam (gravel concrete); (b) large size beam (sand concrete)

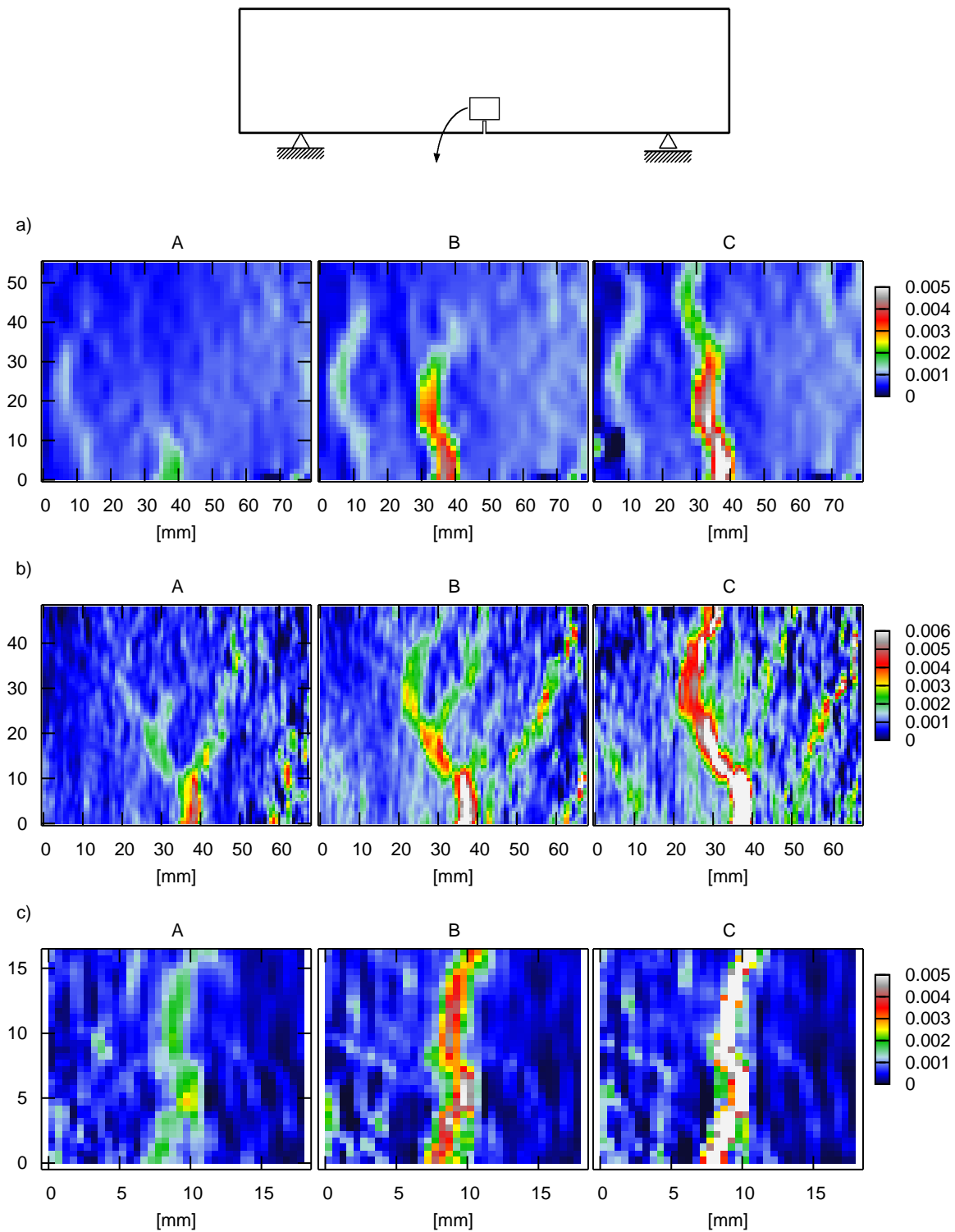


Figure 3.23: Evolution of strain field during experiments from DIC above notch (sand concrete): (a) small size beam (pixel subset 200); (b) medium size beam (pixel subset 210); (c) large size beam (pixel subset 210); (A) before the peak; (B) at peak; (C) after the peak

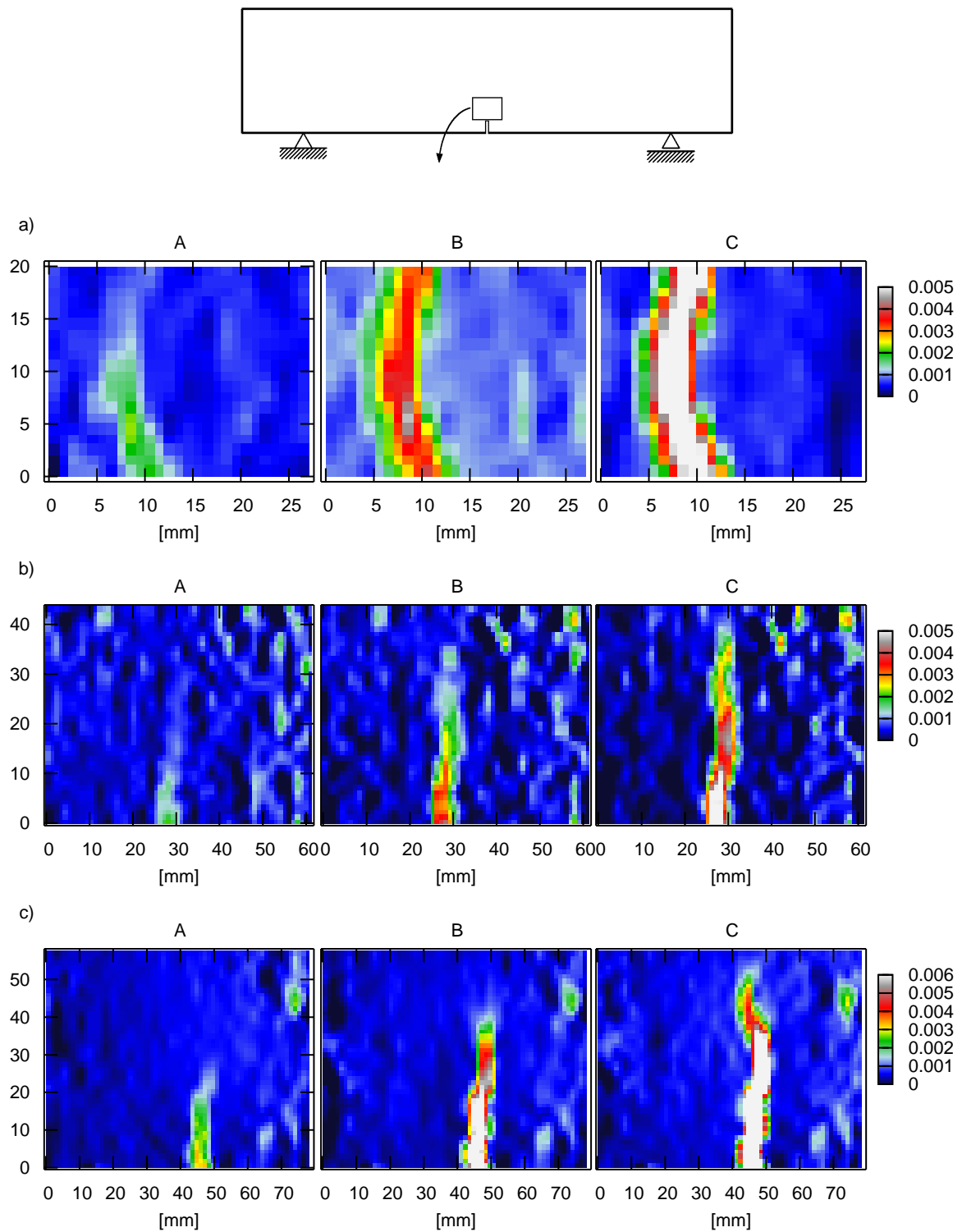


Figure 3.24: Evolution of strain field during experiments from DIC above notch (gravel concrete): (a) small size beam (pixel subset 190); (b) medium size beam (pixel subset 170); (c) large size beam (pixel subset 170); (A) before the peak; (B) at peak; (C) after the peak

The displacement results from the induction displacement gauge were used to verify the width of the fracture process zone obtained from DIC (Fig. 3.20). One assumed that the measured displacement was equal to the integral of strain over the width of the fracture process zone:

$$\Delta L = \int \varepsilon dl \approx \sum \varepsilon d, \quad (3.7)$$

where d is a distance between pixel subsets expressed in millimeters in the DIC method. Before the beam failure, a perfect fit occurred between two curves, after the failure a small discrepancy took place.

Figures 3.23 and 3.24 demonstrate the evolution of the horizontal strain in concrete beams from DIC. The fracture process zone occurs before the peak on the load–deflection diagram. The zone can be almost straight or it can be curved. In some cases, a pattern of localized zones was created at the beginning of loading (Fig. 3.23 b). The width of the fracture process zone was about 6 ± 1 mm ($12d_{50}$, $1.5d_a$) for sand concrete with $d_{50} = 0.5$ mm and $d_a = 4$ mm, and 7 ± 1 mm ($3.5d_{50}$, $0.5d_a$) for gravel concrete with $d_{50} = 2.0$ mm and $d_a = 16$ mm, before the failure. In general, the zone width evolved during the beam loading (Figs. 3.21 and 3.22) (it mainly increased) which was in agreement with other experiments [123].

Chapter 4

Description of the Lattice Geometrical Model

The lattice model developed by the author of this thesis follows the 60 year old idea of discretizing the continuum by line elements [74, 75, 153], uses the already known method for aggregate generation [160] and overlaying it on the lattice [60, 154] (which are nearly 30 years old; Sect. 2.2 and Sect. 4.5.1), but uses a novel approach in performing the computation of lattice elements. The algorithm is of explicit type which allows to easily distribute the calculation onto multiple computers and multiple processors, thus reducing the computation time. Also the formulas are kept simple. This model, if combined with other models by a multi-scale approach, such as FEM or non-local FEM [19] (Figs. 2.12 and 2.20) should give a reliable option for doing lattice-based simulations with discretization scale ≈ 1 mm used in the regions of interest.

To use Lattice Geometrical Model, the software written in C++ by the author of this thesis was applied (Appendix B).

4.1 Notation used

The simulated system consists of numerous elements such as a node, a beam or a flexural-torsional spring. Each of them contains numerous properties listed in Tables 4.2–4.4. Each property is referred to using the index number of the element, where the index placement tells which type of the element is being related to. Figure 4.1 shows the assumed placement for those indices. Each beam connects two nodes, and each spring connects two beams. To simplify the indexing method, they have indices of elements they connect (A, B for a beam and C, D for spring). Thus for example the symbol ${}_C\vec{X}$ used in the context of flexural-torsional spring refers to a beam with index C (first beam of the two beams that the spring connects), and from that beam, its position \vec{X} is addressed.

All variables marked with math ring symbol above them are quaternions (eg. ${}_i\hat{q}$ is the orientation of a beam with index i), all variables with an arrow above are three-dimensional vectors (eg. ${}_i\vec{X}$ is a position of a node with index i) and all other variables are scalars. Quaternions are a very convenient tool for working with rotation due to their interesting features, such as

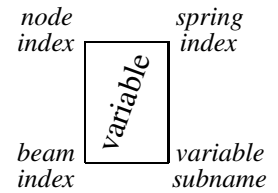


Figure 4.1: Meaning of subscripts and superscripts

the fact that sequences of rotations can be represented by multiplication of two quaternions, or that they do not possess singularity points in the spherical space of all possible rotations. The quaternion algebra is used extensively in the algorithm when dealing with rotations. The necessary information about quaternions is included in Appendix C.

4.2 Introduction

In contrast to the lattice method by van Mier and Schlangen [136,153] (described in Sect. 2.2), the beams are treated as geometric lines (thus, they do not possess any cross-section area) and the calculation is done by examining successive geometric changes of elements, like change of angle between the two beams, or the change of length of a beam. Thus, the model is of kinematic type, the global stiffness matrix is not built and the calculation method has a purely explicit character.

The full 3D version of the model has five types of geometric changes considered: translation ${}_i\Delta\vec{W}$, rotation ${}_i\Delta\vec{R}$, normal deformation ${}_i\Delta\vec{D}$, bending deformation ${}_i\Delta\vec{B}$ and torsional deformation ${}_i\Delta\vec{T}$.

The displacement of each node due to translation ${}_i\Delta\vec{W}$ and rotation ${}_i\Delta\vec{R}$ is calculated by means of performing a simple mathematical average of type $\frac{m_1+m_2+\dots+m_n}{n}$ as shown below:

$${}_j\Delta\vec{N} = \sum_i \frac{{}_i\Delta\vec{W} + {}_i\Delta\vec{R}}{{}_jn_{sum}}, \quad (4.1)$$

where the summation over index i refers to all beams connected to node j , symbol ${}_jn_{sum}$ is the number of beams connected with the node j and ${}_j\Delta\vec{N}$ is the resulting node's displacement. The translation ${}_i\Delta\vec{W}$ and rotation ${}_i\Delta\vec{R}$ are actually used for the conservation of movement. A beam, which is not connected to any other beams (${}^{A,B}n_{sum} = 1$) that moved by ${}_i\Delta\vec{W}$ (or rotated by ${}_i\Delta\vec{R}$) in one calculation step will move (or rotate) the same amount of ${}_i\Delta\vec{W}$ in the next step. Therefore ${}_i\Delta\vec{W}$ and ${}_i\Delta\vec{R}$ do not need any kind of parameters to control the model behavior¹.

In turn, the displacement of each node due to the beam deformation is controlled by model stiffness parameters listed in Tab. 4.1, by means of performing a simple mathematical weighted average of type $\frac{m_1a_1+m_2a_2+\dots+m_na_n}{a_1+a_2+\dots+a_n}$ using weights for longitudinal deformation ${}_ik_l$, bending deformation ${}_ik_b$ and torsional deformation ${}_ik_t$ as shown below:

$${}_j\Delta\vec{K} = \frac{\sum_i \left({}_i\Delta\vec{D}{}_ik_l + {}_i\Delta\vec{B}{}_ik_b + {}_i\Delta\vec{T}{}_ik_t \right)}{\sum_i ({}_ik_l + {}_ik_b + {}_ik_t)}, \quad (4.2)$$

where the summation index i refers to all beams connected to node j and the ${}_j\Delta\vec{K}$ is the resulting node's displacement due to deformation. To simplify the Eq. 4.2, a sum of all weighted average components ${}_jk_{sum}$ of all beams connected with the node j is introduced as:

$${}_jk_{sum} = \sum_i ({}_ik_l + {}_ik_b + {}_ik_t). \quad (4.3)$$

¹the model operates using calculation steps, without time in seconds, therefore the velocity is an artificial (dummy) parameter

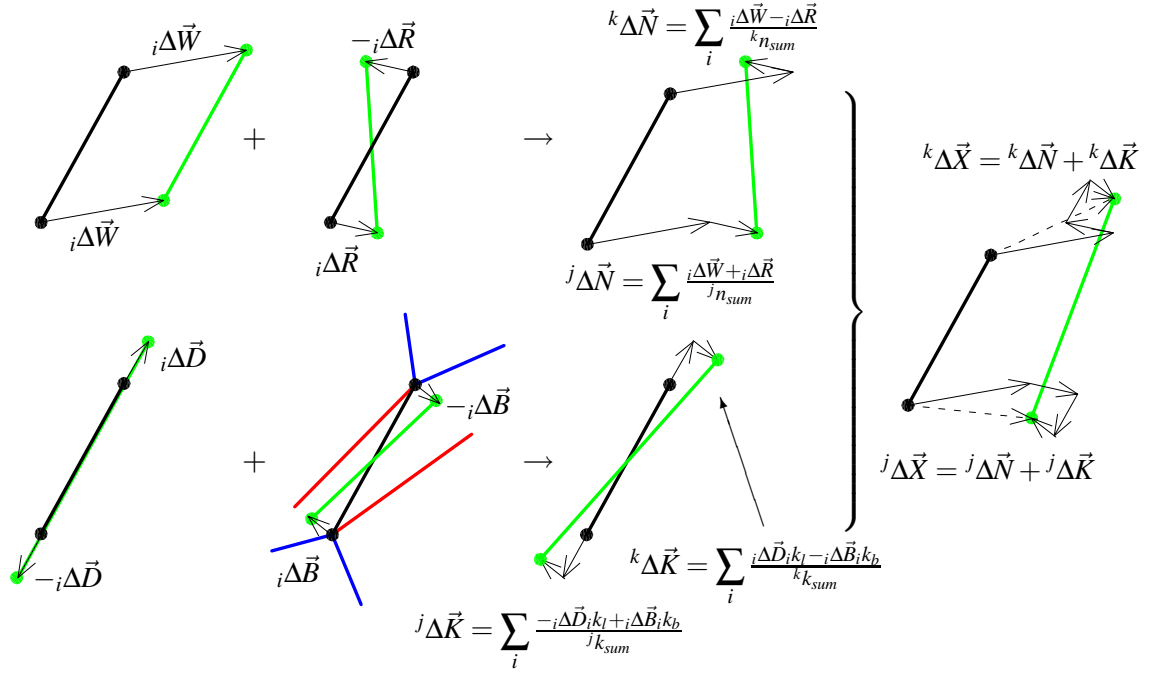


Figure 4.2: General scheme to calculate displacements of elements in the lattice, summation over index i refers to all beams connected with node j for which displacement $j\Delta\vec{X}$ is calculated, similarly with node k

The parameters $i k_l$, $i k_b$ and $i k_t$ govern the deformation response of the material, therefore they are stiffness parameters, but since they are used only to perform a simple mathematical weighted average, they are dimensionless. Consequently the longitudinal stiffness parameter $i k_l$, controls the changes of the beam length, the bending stiffness parameter $i k_b$, controls the changes of the angle between beams, and the torsional stiffness parameter $i k_t$, controls the torsion between beams.

Next, the total node displacement $j\Delta\vec{X}$ is calculated by adding its two components:

$$j\Delta\vec{X} = j\Delta\vec{N} + j\Delta\vec{K}, \quad (4.4)$$

and the new node's position is calculated using:

$$j\vec{X} = j\vec{X} + j\Delta\vec{X}. \quad (4.5)$$

This procedure has been gathered together in Fig. 4.2, on which the green color represents the target placement and length of the beam (into which a beam would move, if it was not constrained by connections with other beams), the black color represents the current position of the beam, the blue color represents the current position of neighboring beams, and the red color represents the original position (angle) of the beam with respect to its all neighboring beams on both ends. The formulas in Fig. 4.2 were written for both ends of a beam for all types of geometric changes, with the exception of torsional deformation $i\vec{T}$, which is discussed in the next section. Attention to the sign must be paid, because for example the translation $i\Delta\vec{W}$ has the same sign for both ends of a beam, but the rotation $i\Delta\vec{R}$ has opposite signs. The components $i\Delta\vec{W}$ and $i\Delta\vec{R}$ repeat the movement that occurred in previous calculation step, thus their computation is straightforward (described in the next

Table 4.1: Main parameters used to describe the Lattice Geometrical Model

Stiffness parameters	
k_l	longitudinal stiffness [-]
k_b	bending stiffness [-]
k_t	torsional stiffness [-]
Fracture parameter	
ϵ_{\min}	critical tensile strain [-]

section). The component ${}_i\Delta\vec{D}$ is obtained by multiplying the beam's normal vector ${}_i\vec{n}$ by the beam's difference of length:

$${}_i\Delta\vec{D} = {}_i\vec{n}({}_id - {}_id_{init}), \quad (4.6)$$

where ${}_id$ and ${}_id_{init}$ are respectively beam's current and initial length. The component ${}_i\Delta\vec{B}$ is calculated by first calculating the angle (red lines on Fig. 4.2) at which the beam would restore its initial angle to its neighboring beams (on both ends), and then calculating an average of those two angles (green line on Fig. 4.2). In two dimensions, without using quaternions this can be written as:

$${}_i\Delta\phi = \sum_k \frac{\alpha^k - \alpha_{init}^k}{{}_ib_{sum}}, \quad (4.7)$$

$${}_i\Delta\vec{B} = {}_id_i\vec{n} \begin{bmatrix} \cos({}_i\Delta\phi) & -\sin({}_i\Delta\phi) \\ \sin({}_i\Delta\phi) & \cos({}_i\Delta\phi) \end{bmatrix} - {}_id_i\vec{n}, \quad (4.8)$$

where ${}_ib_{sum}$ is the amount of beams connected with beam i , the summation over index k refers to angles between this beam and all the beams it connects with (it is described with detail in the next section, the actual term used to describe an angle connecting a pair of beams is a flexural-torsional spring), α^k and α_{init}^k are the current and initial angles between this beam and each other beam. Thus the beam's directional vector ${}_id_i\vec{n}$ is rotated (Eq. 4.8) by the average angle ${}_i\Delta\phi$ (Eq. 4.7) and subtracted from its original position, which results in the displacement ${}_i\Delta\vec{B}$ of the beam's end. However the actual algorithm performs this operation in three-dimensions using quaternions for that purpose:

$${}_i\Delta\vec{B} = {}_i\Delta\vec{B}^{\circ} ({}_id_i\vec{n}) ({}_i\Delta\vec{B}^{\circ})^{-1} - {}_id_i\vec{n}, \quad (4.9)$$

where the method for obtaining quaternion ${}_i\Delta\vec{B}^{\circ}$ is explained in the next section. The operation of quaternion rotation is explained in Appendix C.

To summarize, the Lattice Geometrical Model developed by the author of this thesis, can be written using a single formula for the displacement of node j , which is calculated by averaging the displacements of the end of beams belonging to this node caused by translation, rotation, longitudinal deformation, bending deformation and torsional deformation:

$${}_j\Delta\vec{X} = \sum_i \frac{{}_i\Delta\vec{W} + {}_i\Delta\vec{R}}{{}_jn_{sum}} + \sum_i \frac{{}_i\Delta\vec{D}{}_ik_l + {}_i\Delta\vec{B}{}_ik_b + {}_i\Delta\vec{T}{}_ik_t}{{}_jk_{sum}}, \quad (4.10)$$

wherein the index i refers to all beams connected with node j . The node displacements are calculated successively during each calculation step. Next, a new position of the center of

the beam with index i is calculated as the average of positions of two end nodes belonging to this beam:

$${}_i\vec{X} = \frac{{}^A\vec{X} + {}^B\vec{X}}{2}, \quad (4.11)$$

next, a fracture criterion is checked, and each beam that exceeds the local critical tensile strain ϵ_{\min} (see Tab. 4.1) is removed:

$$\frac{{}_id - {}_id_{init}}{{}_id_{init}} > \epsilon_{\min}, \quad (4.12)$$

which closes the calculation loop. The Equations 4.10–4.12 are repeated infinitely until the calculation process is terminated.

4.3 Algorithm for the Lattice Geometrical Model in 3D

This section describes the algorithm for Lattice Geometrical Model in two stages, with the help of block diagrams. First in Sect. 4.3.1 and 4.3.2 a simplified 3D algorithm without bending and torsion is explained. The lattice mesh behaves as if it was made of rods which are not subject to rotation at joints. In Sect. 4.3.3 and 4.3.4, the algorithm is extended to support bending and torsion between beams (lattice lines), by introducing a new concept of flexural–torsional spring, which makes the model complete.

4.3.1 Beam and node elements used in the simplified algorithm

Figure 4.3 shows a single line element of the lattice that connects nodes A and B . This element is called a beam. However it must be clear that this beam does not behave like Bernoulli beams (Fig. 2.7 in Sect. 2.2). It is always a straight geometric line segment that connects two nodes. The actual bending capability of the Lattice Geometrical Model is achieved by means of preserving original angle between two beams. The beam is described by two unit vectors: a normal direction vector ${}_i\vec{n}$ (between points A and B), and a direction

Table 4.2: The variables stored in each node element

Variable	Description	Assigned	Used
Node			
${}_i\vec{X}$	node position $\{x, y, z\}$	① ⑨c*	⑥b ⑥a
${}_i\Delta\vec{X}$	overall node displacement $\{\Delta x, \Delta y, \Delta z\}$	⑨b	⑨c
${}_i\Delta\vec{N}$	displacement due to conservation of movement	① ⑧	⑨b
${}_i\Delta\vec{K}$	displacement due to element deformation	① ⑧a ⑧b	⑧a ⑧b ⑨b
${}_i n_{sum}$	number of beams connected with this node, used for averaging the movement	① ⑧	⑨a ⑨b
${}_i k_{sum}$	sum of stiffnesses of all beams connected with this node, used for wieghted average of the deformation: ${}_i k_{sum} = \sum_j j k_l + j k_b + j k_t$	① ⑧a ⑧b	⑧a ⑧b ⑨b

* ⑨ denotes the calculation step shown in Figs. 4.4–4.19

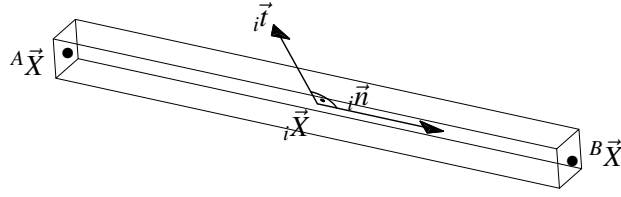


Figure 4.3: Vectors that describe a beam

Table 4.3: The variables stored in each beam element

Variable	Description	Assigned	Used
Beam			
iA, iB	indexes of nodes that this beam connects	①*	④ ⑥ ⑧
$i b_{sum}$	number of springs connected with this beam	①a ②	② ④ ⑤
$i d_{init}$	initial beam length	①	⑧a ⑦
$i d$	current beam length	① ⑥b	⑥ ⑦ ⑧a ⑧b
$i \vec{n}$	a unit vector of direction of the beam, from node iA to node iB	① ⑥b	③a ③b ⑤ ⑥c ⑥d ⑥e ⑧a ⑧b
$i \vec{t}$	the second direction of a beam used to track its rotation	① ⑥c	⑥c ⑥d ⑤
$i \vec{X}, i \vec{X}_{prev}$	position the center of the beam $\{x, y, z\}$	① ⑥a	⑥a
$i \Delta \vec{W}$	displacement of a beam due to conservation of movement	⑥a	⑧
$i \Delta \vec{R}$	displacement of beam's end due to conservation of rotation	⑥e	⑧
$i \Delta \vec{D}$	vector representing longitudinal deformation	⑧a	⑧a
$i \Delta \vec{B}$	vector representing bending deformation	⑧b	⑧b
$i \Delta \vec{T}$	vector representing torsional deformation	⑧b	⑧b
$i \vec{q}, i \vec{q}_{prev}$	orientation of the beam $\vec{q} = a + bi + cj + dk$	① ⑥d	⑥d
$i \Delta \vec{q}$	orientation increment of the beam	① ⑥d	⑥e
$i \omega$	torsional rotation of a beam	①a ⑤	⑤ ⑥c
$i \Delta \vec{B}$	rotation increment needed to restore original bending angle to all its neighboring beams	①a ④	④ ⑧b
$i \Delta \vec{T}$	rotation increment needed to restore original torsion angle to all its neighboring beams	①a ⑤	⑤ ⑧b
$i \epsilon_{min}$	critical tensile strain	①	⑦
$i k_l$	longitudinal stiffness parameter	①	⑧
$i k_b$	bending stiffness parameter	①	⑧b
$i k_t$	torsional stiffness parameter	①	⑧b

* ① denotes the calculation step shown in Figs. 4.4–4.19

vector ${}_i\vec{t}$ (always orthogonal to ${}_i\vec{n}$). The second direction vector is used to track the rotation of the beam around its axis ${}_i\vec{n}$. The center of the beam is ${}_i\vec{X}$. Other properties of the beam are listed with explanations in Tab. 4.3.

The beam connects two node elements with respective coordinates ${}_A\vec{X}$ and ${}_B\vec{X}$. The variables stored in each node along with their description are listed in Tab. 4.2. Those variables are used in the algorithm to perform the summation as expressed by Eq. 4.4 or Eq. 4.10.

4.3.2 Simplified 3D algorithm without bending and torsion

The overall block diagram for a simplified algorithm is depicted on Fig. 4.4. It is divided into 11 steps which are referred respectively as ①–⑪. The step ① is loading the initial configuration of the sample and performs preprocessing. After this step is finished, all the nodes and beams are created, and contain values that describe the sample. The mesh generation is an important part of preprocessing and is described in Sect. 4.5.1.

Next step ② is shown in Fig. 4.6. The temporary values in nodes are set to zero. This section skips handling of bending and torsion, therefore the next step is ⑥, which is shown in Fig. 4.5. It is a loop in which all the beams are successively calculated one after another. The loop starts by calculating the values of ${}_i\vec{W}$ and ${}_i\vec{R}$ (see Fig. 4.2) marked by bold boxes. The beam translation is calculated by subtracting old beam's position ${}_i\vec{X}_{prev}$ from the current value ${}_i\vec{X}$:

$${}_i\Delta\vec{W} = {}_i\vec{X} - {}_i\vec{X}_{prev}, \quad (4.13)$$

next, in step ⑥b a normal vector ${}_i\vec{n}$ of the beam and its length ${}_id$ are updated. Since the ${}_i\vec{n}$ was updated, the beam's second direction must be updated too, to keep it orthogonal. This is done in step ⑥c by doing orthonormalization. After this operation the dot product of ${}_i\vec{n}$ and ${}_i\vec{t}$ will be zero, which means that the vectors are orthogonal. Afterwards ${}_i\vec{t}$ is renormalized so it is again a unit vector. Step ⑥d calculates the orientation increment of the beam. It is a product of the current orientation of the beam (which is built from matrix $[{}_i\vec{n}, {}_i\vec{t}, {}_i\vec{n} \times {}_i\vec{t}]$, using an efficient method mentioned in Sect. C.5.2) and an inverse of previous orientation ${}_i\vec{q}_{prev}$. Finally in step ⑥e the current beam's rotation ${}_i\vec{R}$ is obtained:

$${}_i\Delta\vec{R} = {}_i\Delta\vec{q} {}_i\vec{h} ({}_i\Delta\vec{q})^{-1} - {}_i\vec{h}, \quad (4.14)$$

where ${}_i\vec{h}$ is a temporary half-length vector pointing from beam's center to beam's end.

Next step ⑦–⑧ checks the fracture criterion of a beam, and removes it if necessary as explained on Fig. 4.7.

Then in steps ⑧–⑨ in Fig. 4.8, each beam adds its contribution to the translation ${}^{A,B}\Delta\vec{N}$ and deformation ${}^{A,B}\Delta\vec{K}$ displacement vectors, by adding a respective value to the nodes A and B that this beam connects.

Finally in the last step ⑨–⑩ in Fig. 4.9, a new position for each node is calculated (Eq. 4.10). The variables ${}^i\Delta\vec{N}$ and ${}^i\Delta\vec{K}$ actually contain only the numerator from Eq. 4.1 and Eq. 4.2 and the division is finally done in step ⑨b. The boundary conditions are applied by controlling the position of constrained nodes as explained in the block diagram.

Then some results are saved (like average position of a group of nodes, or strain of a group of beams), and the loop ①–⑩ repeats until terminated.

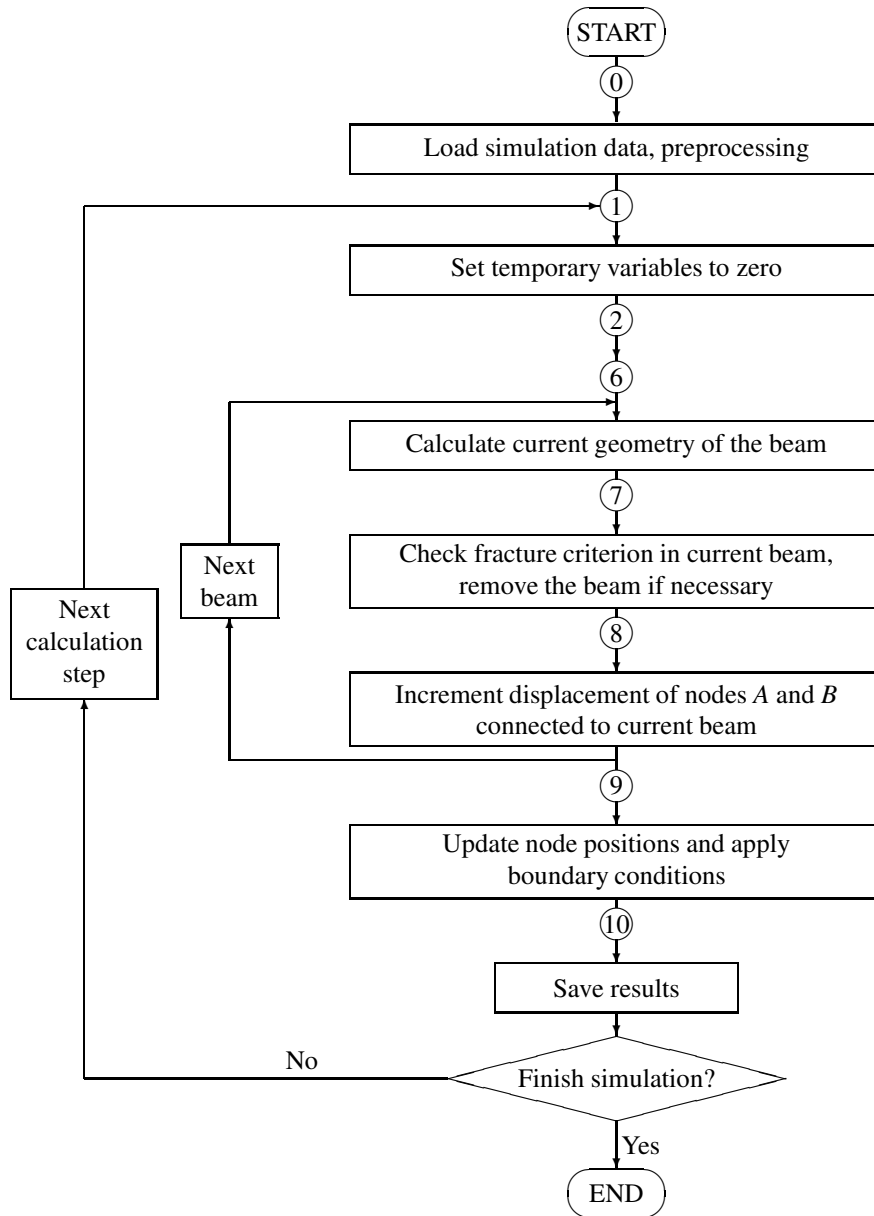


Figure 4.4: Simplified block diagram in calculation algorithm of Lattice Geometrical Model without bending and torsion (the beam elements behave as rods)

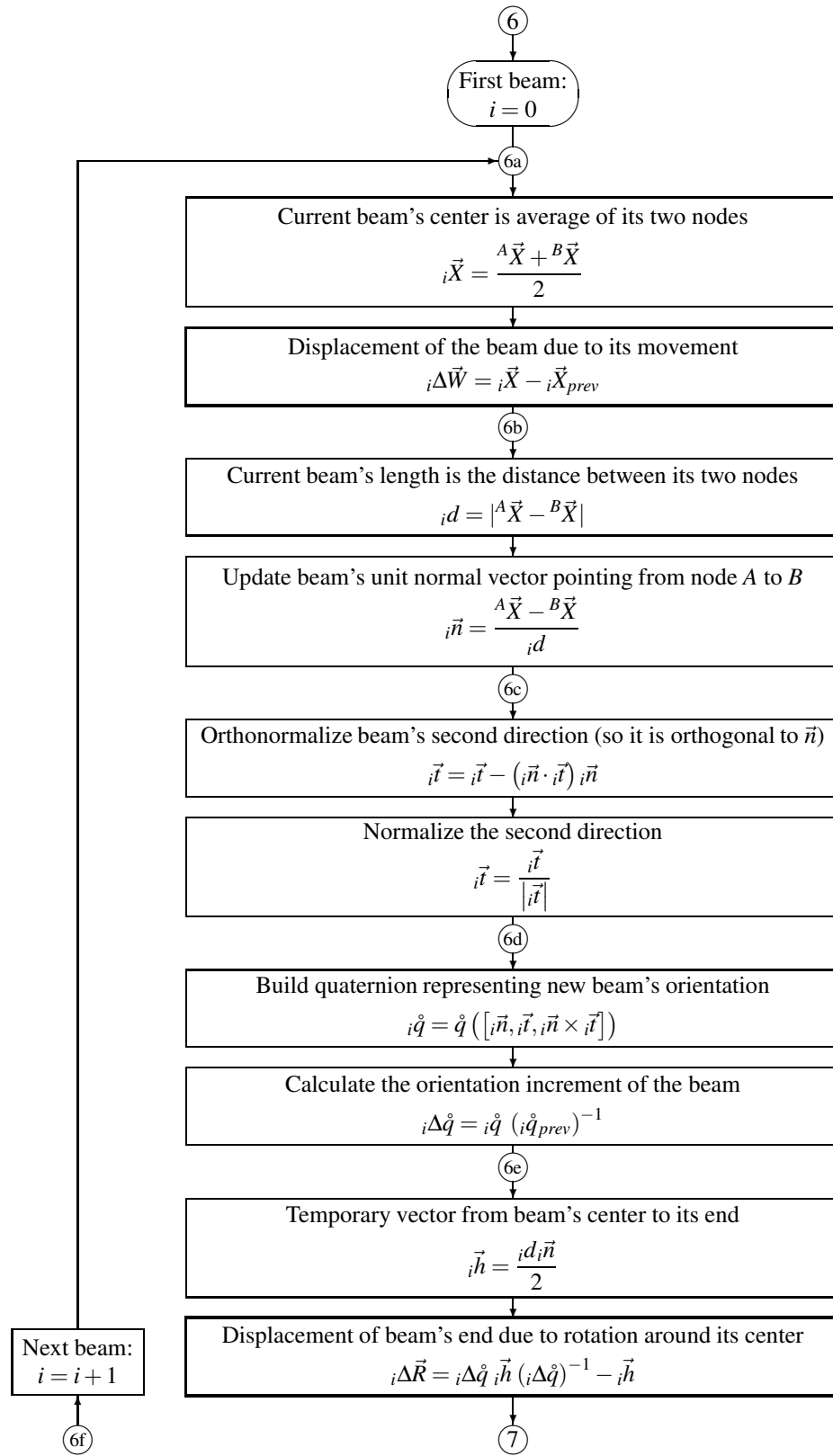


Figure 4.5: Simulation steps ⑥–⑦ (simplified algorithm): Calculate current geometry of the beam and update beam's translation and rotation

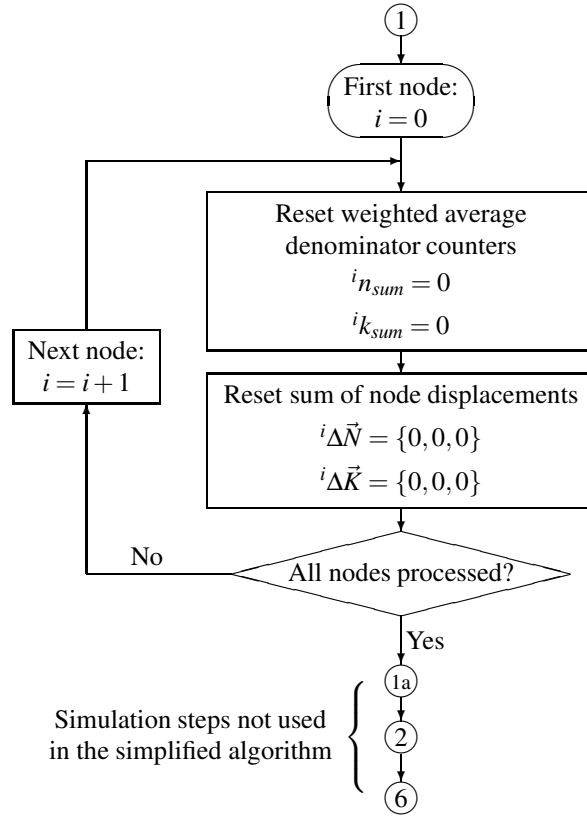


Figure 4.6: Simulation steps ①–②–⑥ (simplified algorithm): Set temporary variables to zero

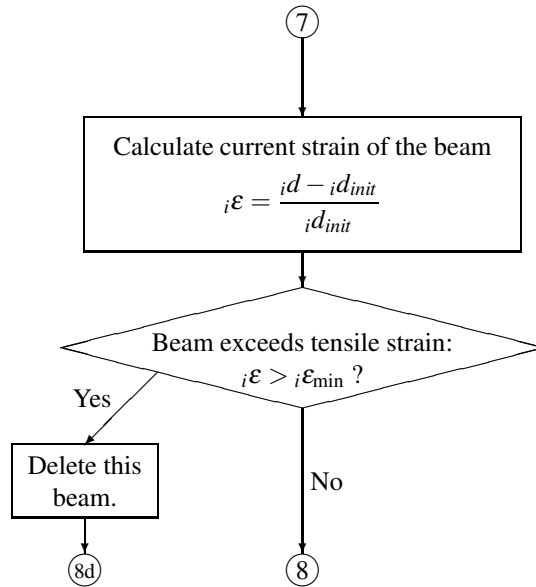


Figure 4.7: Simulation steps ⑦–⑧: Check fracture criterion in the current beam, remove the beam if necessary

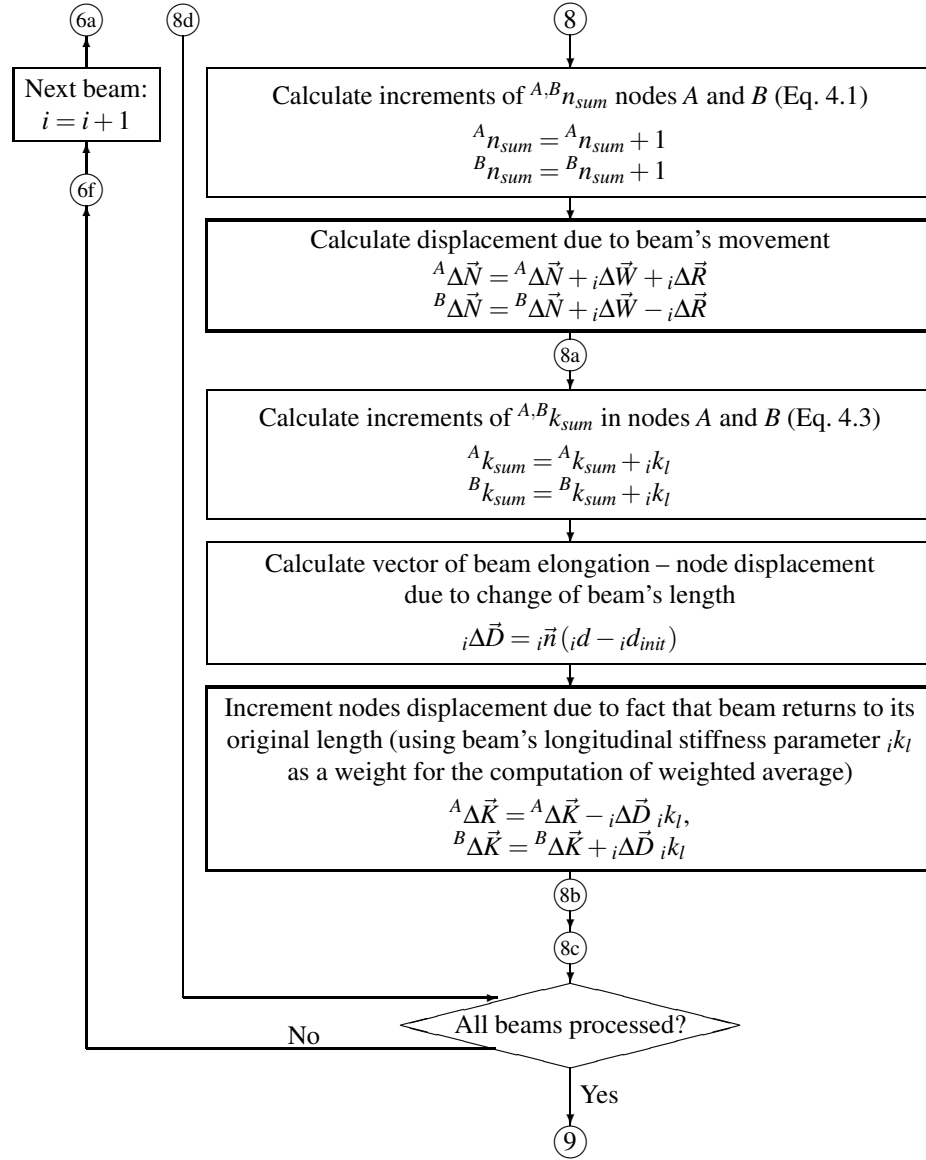


Figure 4.8: Simulation steps ⑧–⑨ (simplified algorithm): Increment displacement of nodes A and B connected to current beam

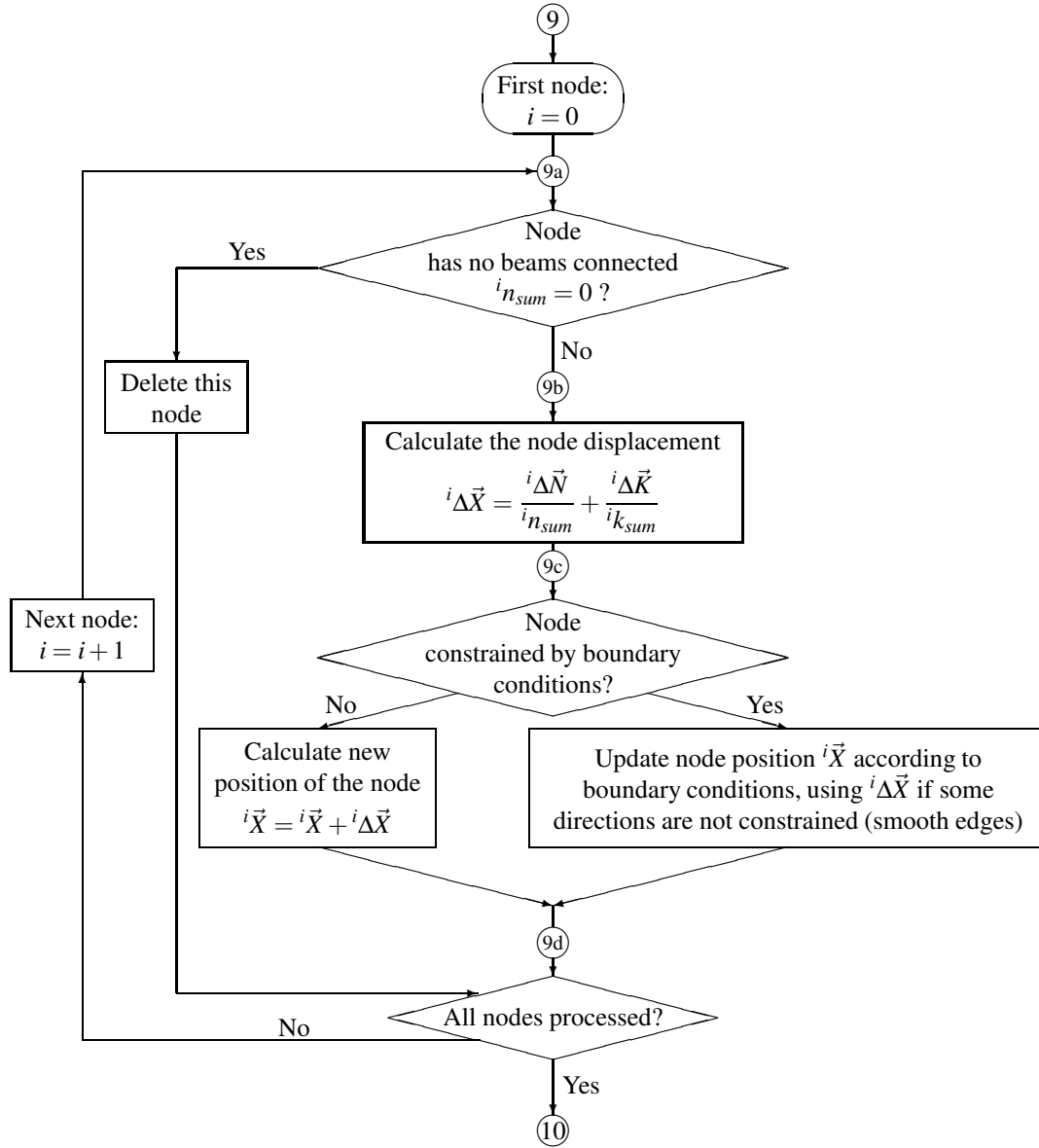


Figure 4.9: Simulation steps ⑨–⑩: Update node positions and apply boundary conditions

4.3.3 Flexural–torsional spring element used in full algorithm

Figure 4.10 shows a single flexural–torsional spring that spans between a pair of beams. The purpose of this element is to keep the original angles between the two beams. Just like a beam element serves the purpose of keeping the original distance between a pair of nodes.

The flexural–torsional spring is described by a unit vector \vec{p}^i which is a normal to the plane created by two beams, and stores three angles: the initial angle α_{init}^i between two beams (Fig. 4.10 a), and two angles $\beta_{1,2,init}^i$ between the initial second direction vector of each beam \vec{c}^i and \vec{d}^i , and the unit vector \vec{p}^i (Fig. 4.10 b). Also the element stores a sign s^i of the angle between two beams, it is necessary, because the $\arccos(\bullet)$ of a vector dot product returns only a value in range $\langle 0, \pi \rangle$. By using this sign, the full range $\langle -\pi, \pi \rangle$ is obtained. Similarly the sign f^i representing the order of nodes in connected beams is stored, it is used to correctly calculate the torsion angle ω^i (Fig. 4.10 d) between two beams. Figure 4.11 shows possible scenarios for the sign f^i . Also, the connected beams undergo a twist $\vartheta_{1,2}^i$ (Fig. 4.10 c) which is calculated from the difference between the initial and current values of angles $\beta_{1,2}^i$ projected on the axis $_{C,D}\vec{n}$ of the corresponding beam.

The calculation of the unit vector \vec{p}^i raises the need of dealing with the numerical precision of the computer. This happens due to the fact that the cross product $_{C,D}\vec{n} \times \vec{d}^i$ can sometimes be very close to zero, or equal to zero, when the connected beams are parallel. The vector \vec{p}^i must be normalized, and dividing it by a value very close to numerical zero results in error, and instability of the model. A constant value ξ depicting the numerical precision of the computer was used. All properties of the flexural–torsional spring element are listed in Tab. 4.4 with their explanations.

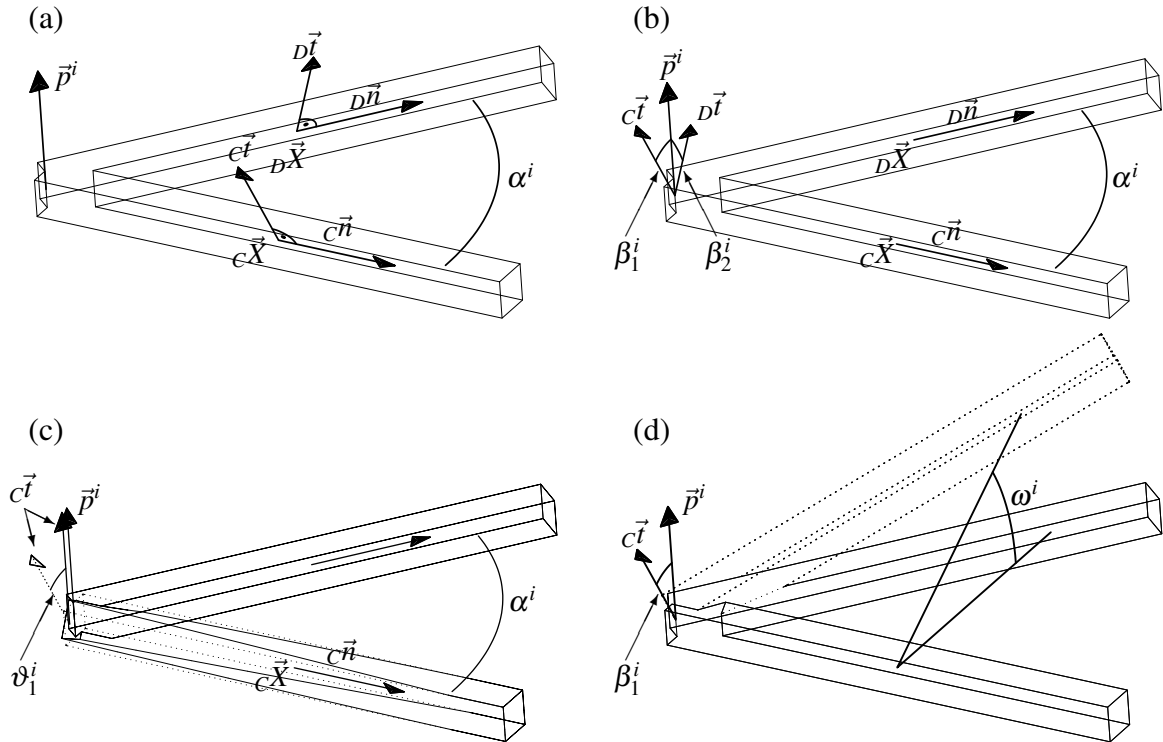


Figure 4.10: Vectors and angles that describe beams and a spring

Table 4.4: The variables stored in each flexural–torsional spring element

Variable	Description	Assigned	Used
Flexural–torsional spring			
C^i, D^i	indexes of beams that this spring connects	①*	④ ⑤
$\vec{p}^i, \vec{p}_{prev}^i$	normal of the plane formed by the two beams that this spring connects	③a	③b ③c ④ ⑤
s^i, s_{prev}^i	sign of the planar angle between two beams, either 1 or -1	① ③b	③b ④
f^i	sign representing the order of nodes in which beams are connected, 1 or -1 (see Fig. 4.11)	⑤	⑤
α_{init}^i	initial angle between two beams	①	④
α^i	current angle between two beams	③b	④
$\Delta\alpha^i$	delta between current and initial angle, used to calculate bending deformation	④	④
$\beta_{1,2,init}^i$	initial angle between $C, D\vec{t}$ and vector \vec{p}^i	①	⑤
$\beta_{1,2}^i$	current angle between $C, D\vec{t}$ and vector \vec{p}^i	⑤	⑤
$\Delta\beta_{1,2}^i$	delta between current and initial angles	⑤	⑤
\hat{B}^i	temporary quaternion representing bending		④
\hat{T}^i	temporary quaternion representing torsion		⑤
$\omega_{1,2}^i$	torsional angle between two beams	⑤	⑤
$\vartheta_{1,2}^i$	twist angle between two beams	⑤	⑤
ξ	numerical precision defined as smallest number that changes a numerical value: $1 + \xi \neq 1$. In the case of C++ double type on 32bit machine $\xi = 1 \cdot 10^{-15}$	constant	③c

* ① denotes the calculation step shown in Figs. 4.4–4.19

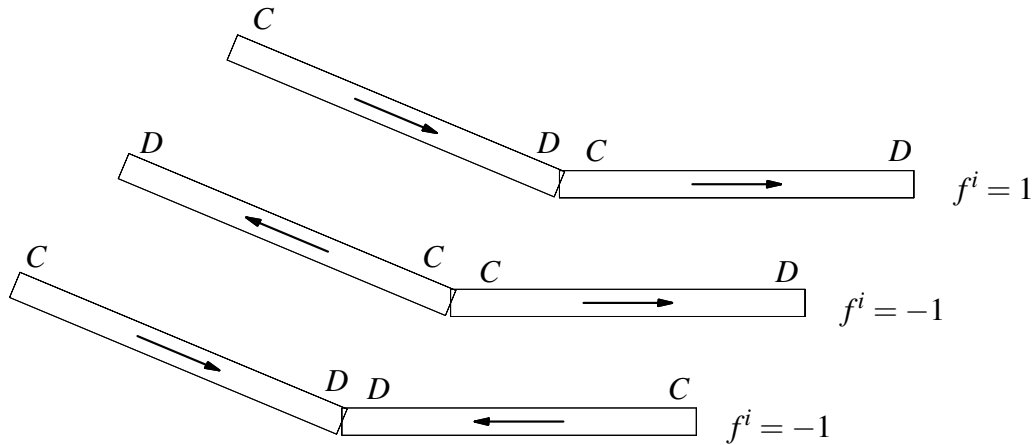


Figure 4.11: Possible scenarios of order of nodes between the connected beams which influence the f^i parameter

4.3.4 Full 3D algorithm

The overall block diagram for full algorithm is presented in Fig. 4.12. The simplified version was extended by adding the steps ①a–②–③–④–⑤–⑥, and ⑧b–⑧c and modifying the steps ⑥c–⑥d in the existing algorithm.

The extension of a simple algorithm with rods to a full version is done by analogy. The nodes store the displacement vectors due to conservation of movement ${}^i\vec{K}$ and deformation ${}^i\vec{N}$, the beams store displacement quaternions (rotation increment) due to bending ${}_i\Delta\vec{B}$ and torsion ${}_i\Delta\vec{T}$. Following the same analogy, beams store the number of springs connected with the beam ${}_ib_{sum}$. The calculation step ①a–② in Fig. 4.13 is analogous to the steps ①–②. The temporary variables are being reset. Next, in the steps ②–③ in Fig. 4.14, the amount of springs connected with each beam is counted.

The steps ③–④ in Figure 4.17 calculate the current geometry of the spring. The normal of the plane between two beams \vec{p}^i (step ③a) is used to track the changes of the sign s^i between 1 and -1 . If a dot product of $\vec{p}^i \cdot \vec{p}_{prev}^i$ is below zero, it means that \vec{p}^i „flipped” to the other side of the plane, and that the angle α^i between two beams had changed from π to $-\pi$ (step ③b). After that the \vec{p}^i is normalized if the numerical precision is high enough (step ③c, Sect. 4.3.3).

In the steps ④–⑤ in Fig. 4.18, small quaternion rotations are combined together to calculate the overall beam rotation ${}_i\Delta\vec{B}$. It is not possible to divide a combined rotation afterwards because the angle of rotation must be divided before combining it with previous rotations, which is done using a product of quaternions:

$${}_j\Delta\vec{B} = \prod_i \vec{B}^i \left(\vec{p}^i, \frac{-s^i \Delta\alpha^i}{{}_jb_{sum}} \right), \quad (4.15)$$

wherein the index i refers to all springs connected with beam j , a temporary quaternion \vec{B}^i is built from axis \vec{p}^i and angle $\Delta\alpha^i$ of each spring. The sign s^i makes sure that $\Delta\alpha^i \in \langle -\pi, \pi \rangle$, and ${}_jb_{sum}$ divides the angle by number of springs. This operation is fully analogous a performance of a mathematical average as in Eq. 4.1, except that it operates on rotations (quaternions), not on displacements (vectors).

The steps ⑤–⑥ in Fig. 4.18 calculate the quaternion rotation of a beam due to torsion between beams ${}_i\Delta\vec{T}$. First, the sign f^i is determined according to Fig. 4.11, then current values of angles $\beta_{1,2}^i$ between ${}_{C,D}\vec{t}$ and unit normal of the plane \vec{p}^i are calculated (Fig. 4.10). Later, the difference between the initial and the current angle $\Delta\beta_{1,2}^i$ is used to calculate the torsional angle between beams $\omega_{1,2}^i$ and angle of twist $\vartheta_{1,2}^i$.

The meaning of the torsion angle $\omega_{1,2}^i$ (Fig. 4.10) is that if a beam C is rotated around the axis of the beam D by ω_1^i it will restore its initial angle $\beta_{1,init}^i$ between ${}_{C}\vec{t}$ and \vec{p}^i . Conversely the twist angle has the meaning that if a beam C rotates around its own axis by ϑ_1^i it will also restore its initial angle $\beta_{1,init}^i$. Thus, the torsion and twist angle describe two possible ways in which a beam can restore the initial angle to the plane formed by two beams.

The angle α^i plays an important role in calculating the angle of twist. If two beams nearly overlap (α^i is close to zero) or are almost parallel ($|\alpha^i|$ is close to π), the twist cannot be induced between the beams — the beam will rotate instead (using ω_1^i), without twisting. In turn, if the beams are roughly orthogonal to each other ($|\alpha^i|$ is close to $\frac{\pi}{2}$), the induced twist is strong (Fig. 4.10). It is easier to twist a bar if the rotations on both ends are applied in the

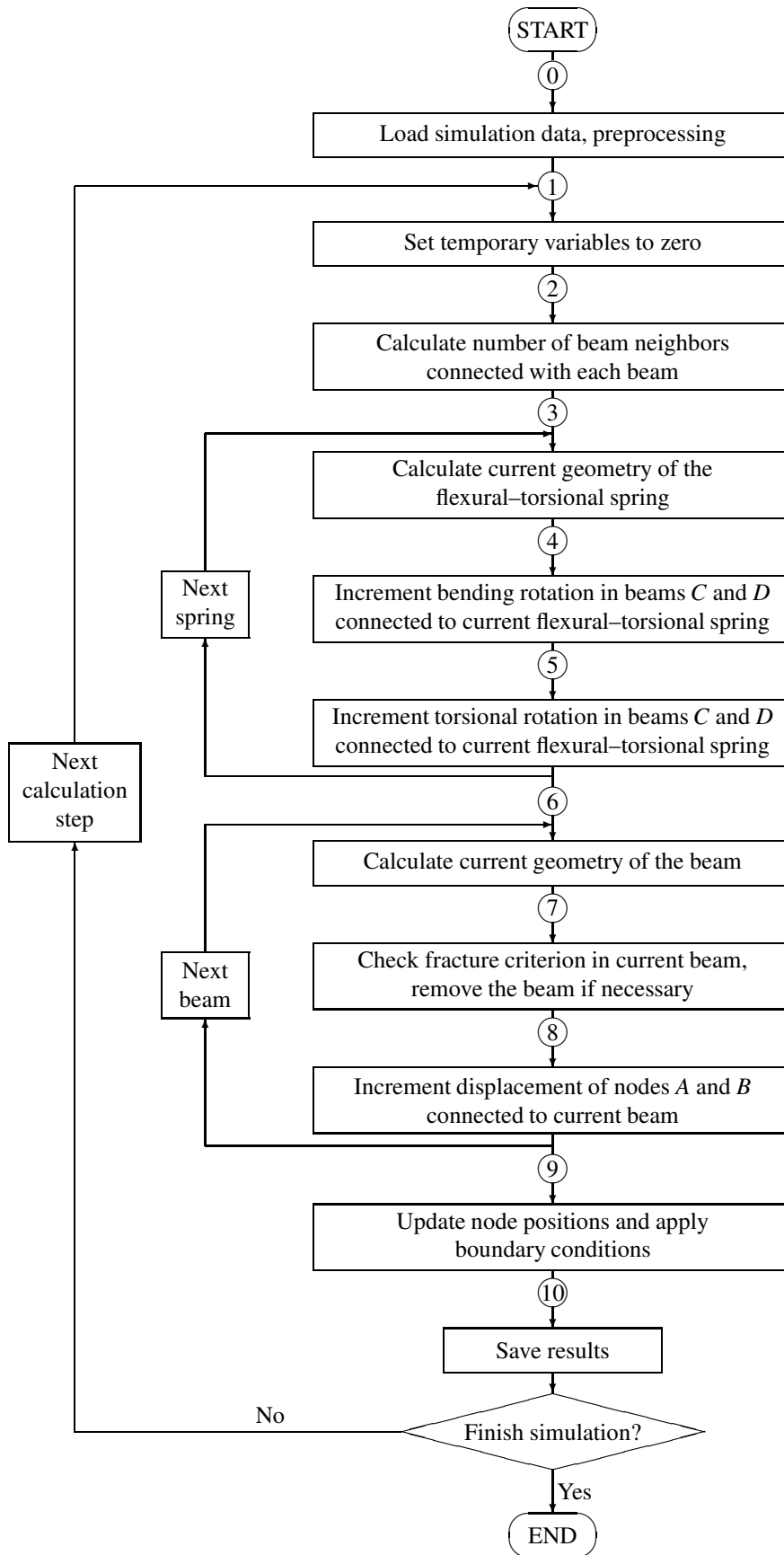


Figure 4.12: Block diagram of full calculation algorithm in Lattice Geometrical Model

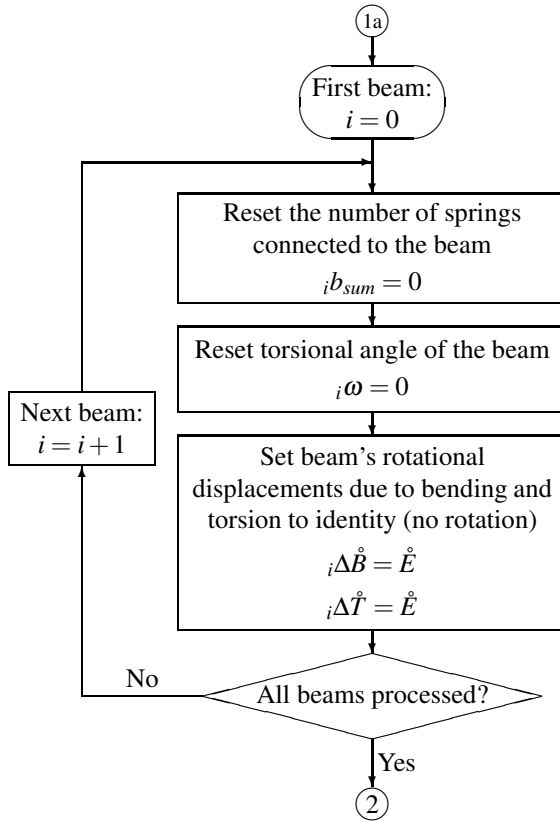


Figure 4.13: Simulation steps ①a–② (full algorithm with bending and torsion): Set temporary variables to zero

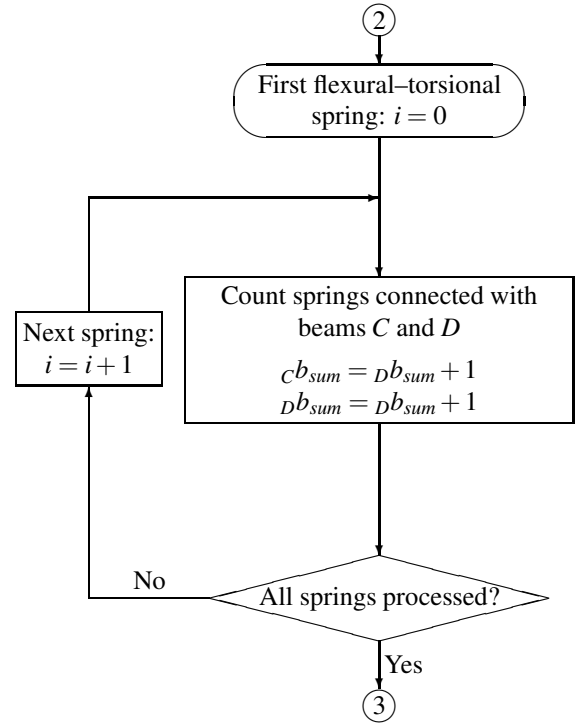


Figure 4.14: Simulation steps ②–③ (full algorithm): Calculate number of springs connected with each beam

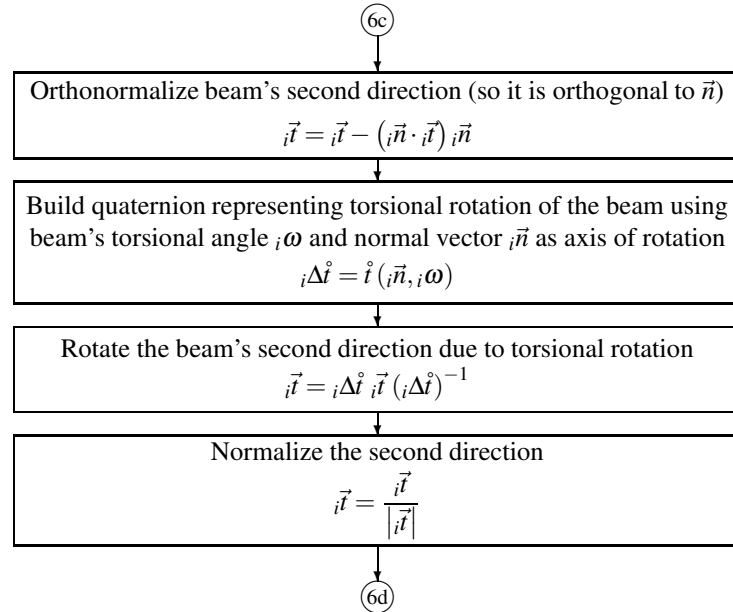


Figure 4.15: Simulation steps ⑥c–⑥d (full algorithm): Calculate the orientation of the beam, taking its torsion into account

plane orthogonal to the axis of the object. It turns out that the angle of twist is projected on the axis of beam, using $|\sin(\alpha^i)|$. However in this model a function $\sin^2(\alpha^i)$ has to be used, because the sharp peak (discontinuity) at $\alpha^i = 0$ leads to the instable behavior of the model (Fig. 4.16). Incidentally $\sin^2(\alpha^i)$ equals $|\vec{p}^i|^2$ before normalization, which allows to optimize the calculation. Also an integer counter $c_{1,2}^i$ is used for each beam which allows to store twist angles bigger than π (or smaller than $-\pi$), therefore $\vartheta_{1,2}^i$ is incremented by $2\pi c_{1,2}^i$. This was not included in the block diagram, because it would unnecessarily complicate it.

Finally, the torsion angle ${}_j\omega$ for each beam can be calculated (the angle which the beam should rotate around its own axis):

$${}_j\omega = \sum_i \frac{{}_{this}\vartheta^i + {}_{this}\omega^i}{2{}_jb_{sum}}, \quad (4.16)$$

where index i refers to all springs connected with beam j and ${}_{this}\vartheta^i$, ${}_{this}\omega^i$ are respective angles calculated in spring i which refer to *this* beam j . This summation divided by $2{}_jb_{sum}$ is also a mathematical average, just like everything else in this model. The factor 2 used in the denominator may be caused by the $|\sin(\alpha^i)| \leftrightarrow \sin^2(\alpha^i)$ approximation (Fig. 4.16), and without it, the calculation would become unstable.

The torsional rotation ${}_j\Delta\hat{T}$ at the end of step ⑤ represents the rotation of the beam j around its neighboring beams, due to the torsion angle ω^i (Fig. 4.10). It is calculated by performing another quaternion product, also averaged over the amount ${}_jb_{sum}$ of beams involved:

$${}_j\Delta\hat{T} = \prod_i \hat{T}^i \left({}_{other}\vec{n}, \frac{-f^i {}_{this}\vartheta^i - {}_{other}\omega^i}{{}_jb_{sum}} \right), \quad (4.17)$$

where index i refers to all springs connected with beam j , the quaternion \hat{T}^i is build from the axis of rotation ${}_{other}\vec{n}$ of the *other* beam, and angle of rotation being a difference between twist and torsion.

The steps ⑥c–⑥d shown on Fig. 4.15 are a modification to the previous simplified algorithm. The beam's second direction ${}_i\vec{t}$ is rotated by the angle of torsion ${}_i\omega$ around the beam's normal axis ${}_i\vec{n}$. This effectively performs the rotation calculated in Eq. 4.16.

The addition of the steps ⑧b–⑧c (Fig. 4.19) makes the Eq. 4.10 now covered fully, by handling bending deformations ${}_i\Delta\vec{B}$ and torsional deformations ${}_i\Delta\vec{T}$. The operations in the block diagram are analogous to those in Fig. 4.8.

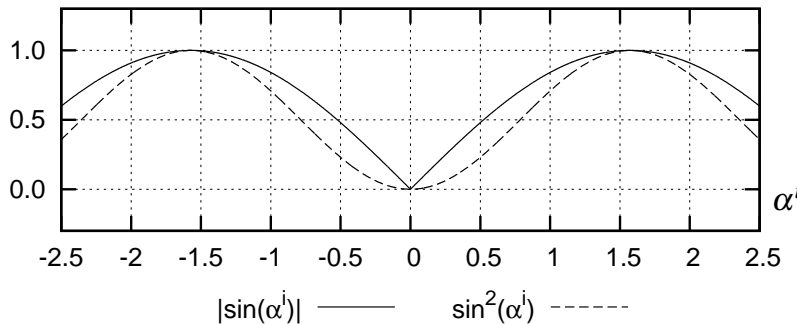


Figure 4.16: Plot showing the two possible functions for projecting twist angle ϑ^i : function $|\sin(\alpha^i)|$ is geometrically correct, but due to discontinuity has instable behavior at angle $\alpha^i = 0$; in turn function $\sin^2(\alpha^i)$ is continuous and results in a stable calculations.

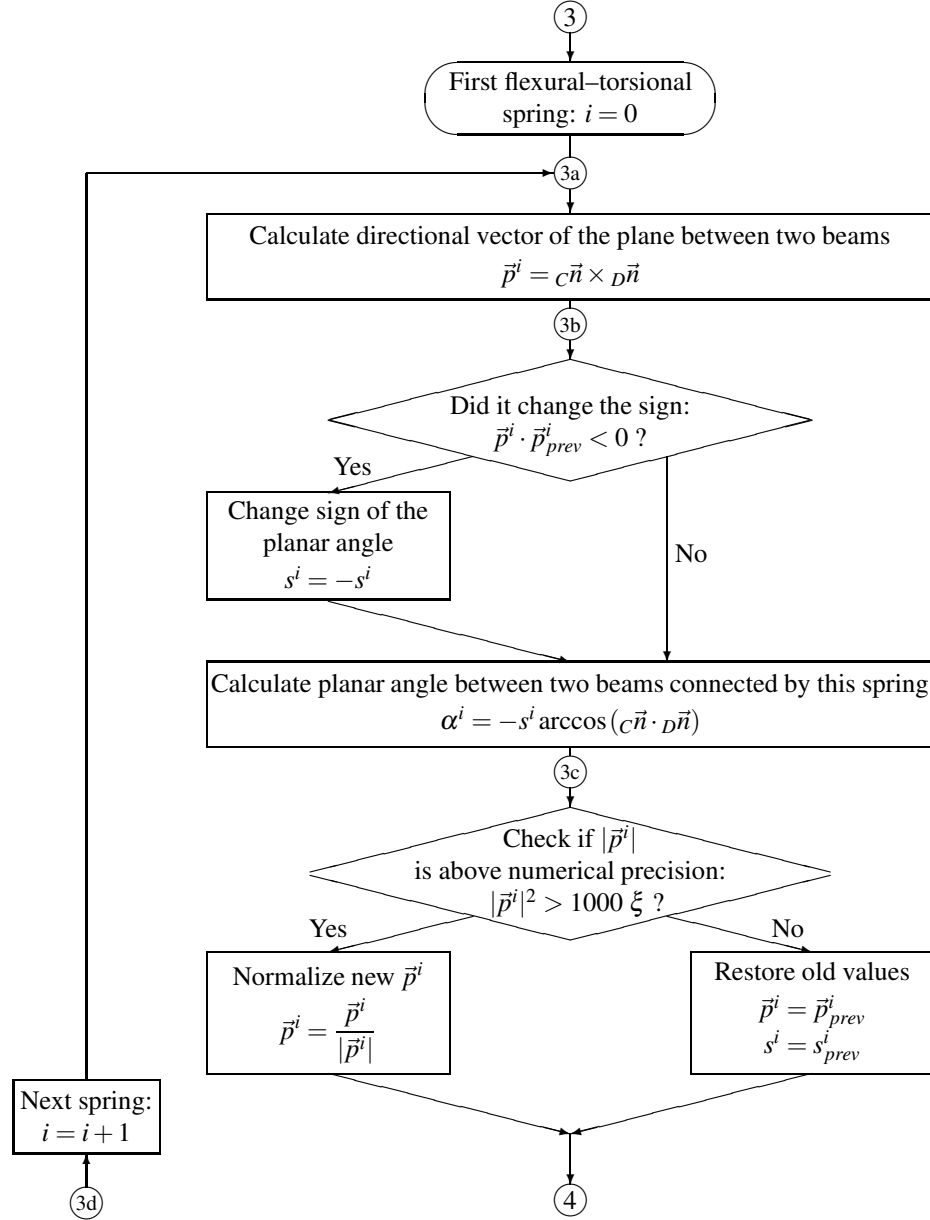


Figure 4.17: Simulation steps ③–④ (full algorithm): Calculate current geometry of the flexural-torsional spring

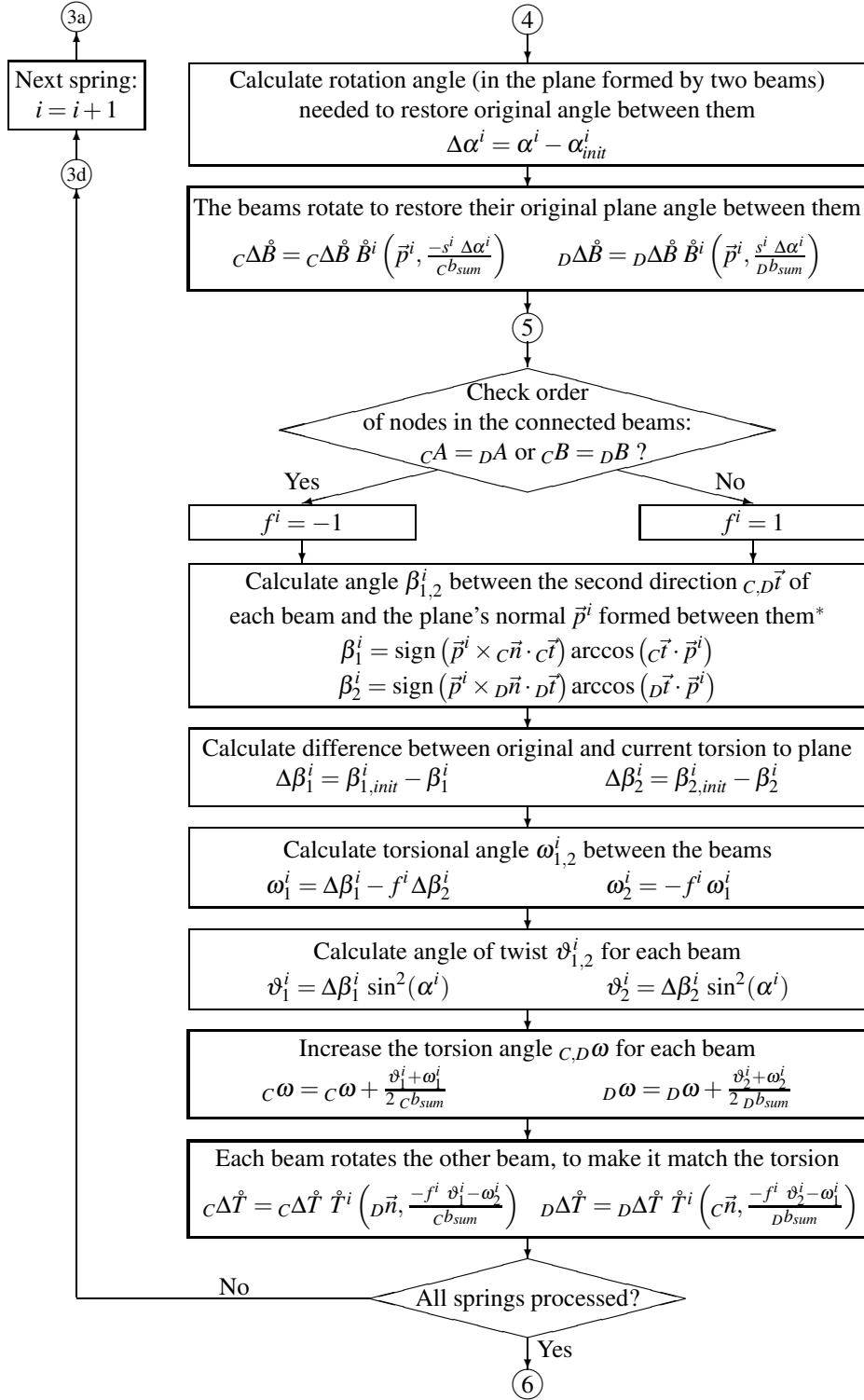


Figure 4.18: Simulation steps ④–⑤–⑥ (full algorithm): Increment bending and torsional rotation in beams A and B connected to current flexural–torsional spring

* the function „ $\text{sign}(\vec{p}^i \times {}_D\vec{n} \cdot {}_C\vec{t})$ ” is a simple method of ensuring a correct sign from the function „ $\arccos(\bullet)$ ”, thus the angle in the range $\langle -\pi, \pi \rangle$ is correctly assigned. This method however produces wrong results in the limit cases when the angle is very close to zero or π . The effect is not noticeable, because in such cases the sign is not important. The author used a little different method to check the sign, to be sure that it was always correct. This method involves several `if...then...else` statements checking $\Delta\beta_{1,2,prev}^i$, and eventually building a quaternion. Thus writing a block diagram for this operation would unnecessarily fill half of the page. The exact code can be found in YADE source, in file `LatticeLaw.cpp:184` on the webpage <http://yade.berlios.de>

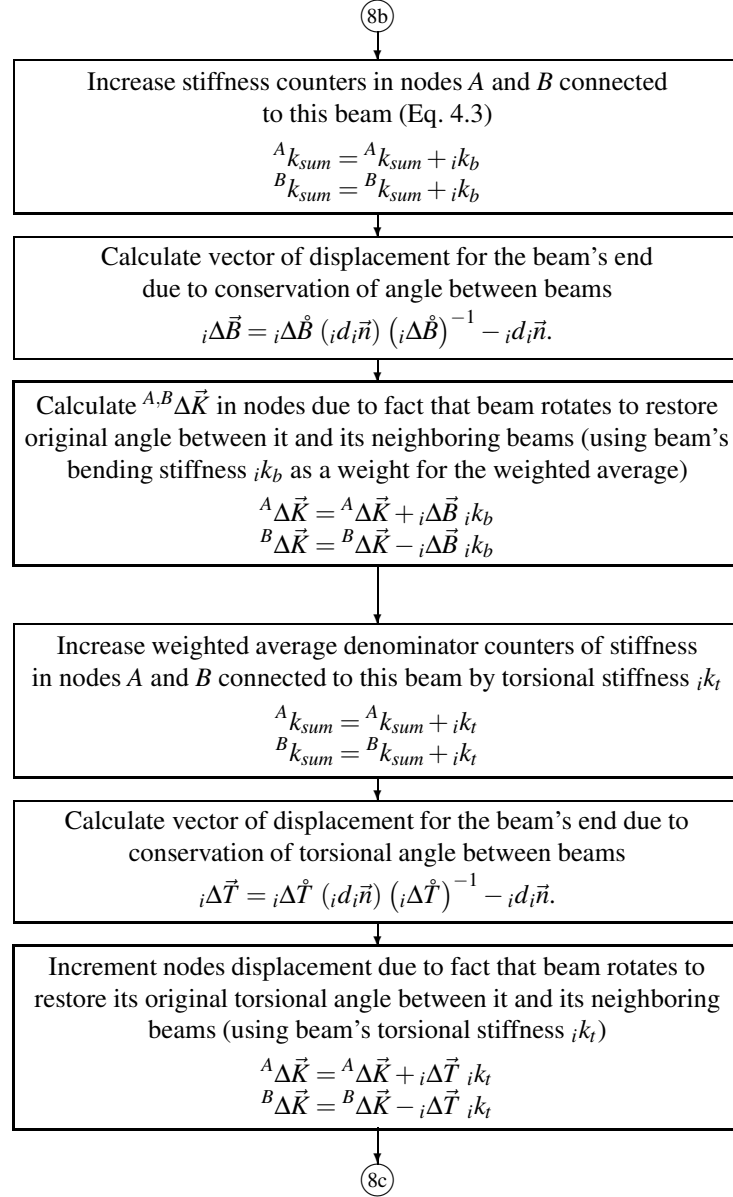


Figure 4.19: Simulation steps 8b–8c (full algorithm): Increment displacement of nodes A and B connected to current beam due to change of angle and torsion between beams

4.4 Example calculation

The calculation method of the resultant node displacement (Eq. 4.10) is shown below on the example of a simple frame composed of 4 beams during one prescribed displacement increment (Fig. 4.20). The nodes 3 and 5 are fixed and the node 1 is assumed to displace to the point with the new coordinates (0.58, 1.42). The displacement vectors of the node 2 in beams 1, 2 and 3 are: ${}^2_1\vec{W} = (0.09, -0.04)$, ${}^2_2\vec{W} = (0, 0)$, ${}^2_3\vec{W} = (0, 0)$, ${}^2_1\vec{R} = (-0.065, -0.012)$, ${}^2_2\vec{R} = (0, 0)$, ${}^2_3\vec{R} = (0, 0)$, ${}^2_1\vec{D} = (0.035, -0.131)$, ${}^2_2\vec{D} = (0, 0)$, ${}^2_3\vec{D} = (0, 0)$, ${}^2_1\vec{B} = (0.124, 0.044)$, ${}^2_2\vec{B} = (0.44, -0.064)$ and ${}^2_3\vec{B} = (-0.006, 0.041)$, respectively (with the rotation angle of the node 2: ${}^2_1\Delta\varphi = -8.9^\circ$ (beam 1), ${}^2_2\Delta\varphi = 4.45^\circ$ (beam 2), and ${}^2_3\Delta\varphi = -2.97^\circ$ (beam 3)). For the stiffness parameters $k_b = 0.6$ and $k_l = 1.0$, the resultant displacement vector of the node 2 is equal to (Eq. 4.10):

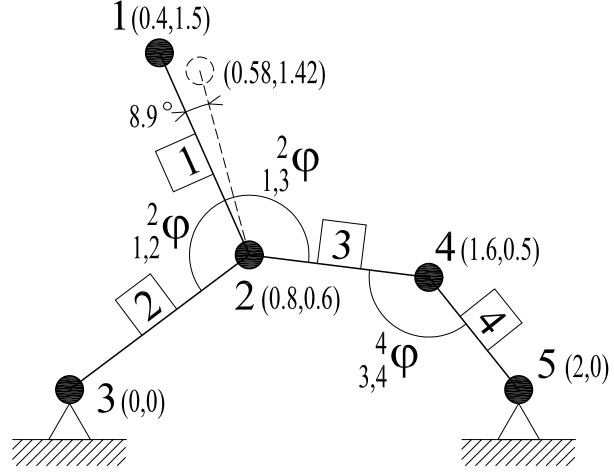


Figure 4.20: A simplified frame composed of 4 beams

$$\begin{aligned} \Delta\vec{X}^2 &= \frac{1}{3} \left({}^2_1\vec{W} + {}^2_2\vec{W} + {}^2_3\vec{W} + {}^2_1\vec{R} + {}^2_2\vec{R} + {}^2_3\vec{R} \right) + \\ &+ \left(\frac{{}^2_1k_l{}^2_1\vec{D} + {}^2_2k_l{}^2_2\vec{D} + {}^2_3k_l{}^2_3\vec{D} + {}^2_1k_b{}^2_1\vec{B} + {}^2_2k_b{}^2_2\vec{B} + {}^2_3k_b{}^2_3\vec{B}}{{}^2_1k_l + {}^2_2k_l + {}^2_3k_l + {}^2_1k_b + {}^2_2k_b + {}^2_3k_b} \right) = \\ &= \frac{1}{3} ((0.09, -0.04) + (0, 0) + (0, 0) + (-0.065, -0.012) + (0, 0) + (0, 0)) + \\ &+ \frac{((0.035, -0.131) + (0, 0) + (0, 0)) \cdot 1.0 + ((0.124, 0.044) + (0.44, -0.064) + (-0.006, 0.041)) \cdot 0.6}{(1.0 + 1.0 + 1.0 + 0.6 + 0.6 + 0.6)} = \\ &= (0.037, -0.042), \end{aligned} \quad (4.18)$$

and the new coordinates of node 2 are (0.837, 0.558). In turn, the displacement vectors of node 4 in beams 3 and 4 are: ${}^4_3\vec{W} = (0, 0)$, ${}^4_4\vec{W} = (0, 0)$, ${}^4_3\vec{R} = (0, 0)$, ${}^4_4\vec{R} = (0, 0)$, ${}^4_3\vec{D} = (0, 0)$, ${}^4_4\vec{D} = (0, 0)$, ${}^4_3\vec{B} = (-0.006, -0.041)$ and ${}^4_4\vec{B} = (0, 0)$, respectively (with the rotation angle of node 4: ${}^4_3\Delta\varphi = -2.97^\circ$ (beam 3) and ${}^4_4\Delta\varphi = 2.97^\circ$ (beam 4)). The resultant displacement vector of the node 4 is equal to (Eq. 4.10):

$$\begin{aligned} \Delta\vec{X}^4 &= \frac{1}{2} \left({}^4_3\vec{W} + {}^4_4\vec{W} + {}^4_3\vec{R} + {}^4_4\vec{R} \right) + \left(\frac{{}^4_3k_l{}^4_3\vec{D} + {}^4_4k_l{}^4_4\vec{D} + {}^4_3k_b{}^4_3\vec{B} + {}^4_4k_b{}^4_4\vec{B}}{{}^4_3k_l + {}^4_4k_l + {}^4_3k_b + {}^4_4k_b} \right) = \\ &= \frac{1}{2} ((0, 0) + (0, 0) + (0, 0) + (0, 0)) + \\ &+ \left(\frac{(0, 0) \cdot 1.0 + ((-0.006, -0.041) + (0, 0)) \cdot 0.6}{(1.0 + 1.0 + 0.6 + 0.6)} \right) = (-0.001, -0.007), \end{aligned} \quad (4.19)$$

and the new coordinates of node 4 are (1.599, 0.493).

4.5 Model discretization

The studies of Jirasek and Bazant [81] and Schlangen [134] showed the importance of the randomness of the lattice. A regular lattice of any kind can introduce severe bias for the crack propagation direction.

Two different generation methods were employed by the author. First a method described in the following section was used, later a conventional Delaunay triangulation was used. The results obtained with both methods are similar. However the Delaunay triangulation contributes to shorter computation time.

4.5.1 Mesh generation

Table 4.5 lists all parameters used for the mesh generation, and Fig. 4.21 shows the generation method. It is similar to method in Sect. 2.2 [99, 136], with the exception that a triangular grid is used, and not an orthogonal one. First, a triangular grid is created in the material with the side dimensions equal to g (Fig. 4.21). In each triangle of the grid, additional interior squares are assumed with an area of $s \times s$. Next, one point was selected at random within these interior squares. Later, all points inside of squares were connected with neighboring ones within a distance of r_{\max} to create a non-uniform mesh of beams, where the maximum beam length was r_{\max} (e.g. $r_{\max} = 2g$), the minimum beam length was determined by the random placement of points (e.g. $r_{\min} = 0.1g$ for $s = 0.6g$) and the minimum angle between beams was assumed to be as α (e.g. $\alpha = 20^\circ$). A uniform triangular mesh could be obtained with parameter $s = 0g$. Using this grid generation method, the beams could cross each other in two dimensional calculations (similarly as in [25]) but they did not intersect each other in three-dimensional analysis. To generate a mesh in three-dimensions an analogous method is used, with tetrahedral grid used instead of triangular one.

Alternative method of connecting points generated in this manner is to use a Delaunay triangulation. Then the parameters r_{\max} and α become irrelevant. The Delaunay triangulation for each point creates a triangle (or tetrahedron in 3D) which is circumscribed by the smallest possible circle (or sphere) containing this point and only two more points (or three more points in three-dimensions).

The generated lattice mesh has in fact nothing to do with concrete or any other brittle material. To make the model of concrete more realistic, the material structure must be overlaid on the lattice.

Table 4.5: Mesh generation parameters used in the Lattice Geometrical Model

Mesh generation parameters	
g	cell size [meter]
r_{\max}	maximum beam length [meter]*
α	min angle between beams [radians]*
s	mesh irregularity [meter]

*parameter used only in non-Delaunay generation method

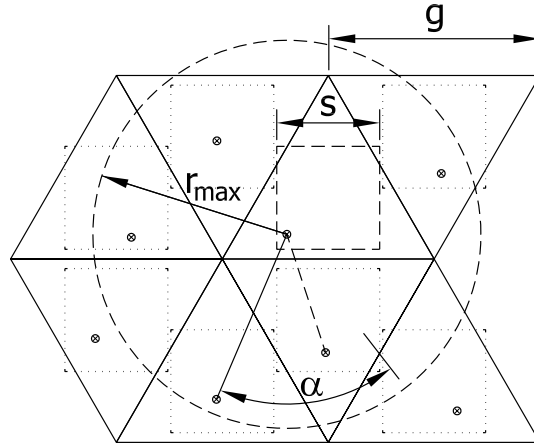


Figure 4.21: Scheme to assume a non-uniform distribution of beams in the lattice (s – size of interior squares, r_{\max} – maximum beam radius, α – minimum angle between two beams, g – size of triangular grid)

4.5.2 Distribution of aggregates

To project the aggregate structure onto the lattice, the aggregates are first generated using a selected grading curve. A certain amount of aggregates with diameters $d_1, d_2 \dots d_n$ is generated to fit the grading curve (Sect. 2.2, Fig. 2.10). Then, the aggregates are randomly placed inside the mesh with preserving a minimal mutual distance given by Eq. 2.4. The minimum aggregate diameter is 2 mm. The lattice beam length is assumed to be ≈ 1 mm which is longer than this assumed in [154] (equal to one third of the aggregate size). The beam is longer to reduce the computational effort.

When aggregates are placed inside the mesh, the lattice beam elements obtain different properties depending on whether they are a cement matrix, an aggregate or a bond (Fig. 2.11 in Sect. 2.2).

4.6 Non-local approach to calculate strain

To decrease the material brittleness in calculations (which is usually too large [156] in a 2D one-phase material), a non-local approach [10] can be used to calculate strains [97]. In the calculations, the normal strain in each beam can be assumed to be non-local (replacing the formula for strains in step ⑦ in Fig. 4.7):

$${}_k\bar{\epsilon} = \frac{\sum_i w({}_{k,i}r) {}_i\epsilon_i d \cos({}_{k,i}\alpha)}{\sum_i w({}_{k,i}r) d \cos({}_{k,i}\alpha)}, \quad (4.20)$$

where the index i refers to all beams within a certain radius (eg. $3l_c$), $w(\bullet)$ is the weighting function, ${}_{k,i}r$ is a distance between the mid-point of the beam k and the mid-points of other neighboring beams i and ${}_{k,i}\alpha$ is an angle between the beams. The beams do not have to be connected. In general, it is required that the weighting function should not alter an uniform field, which means that it must satisfy the normalizing condition [10]. Therefore,

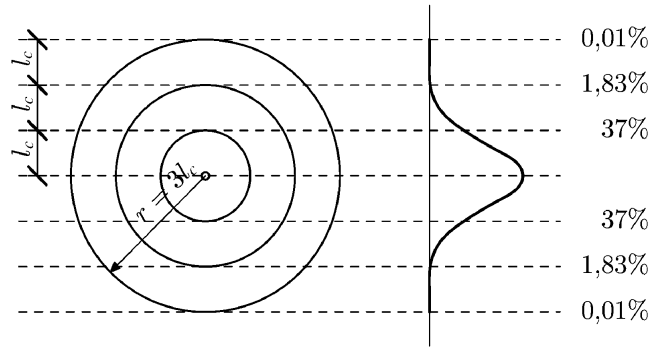


Figure 4.22: Range of averaging using gaussian distribution; [19]

as a weighting function $w(r)$ in Eq. 4.20, a Gauss distribution function for 2D problems was used:

$$w(r) = \frac{1}{l_c \sqrt{\pi}} e^{-\left(\frac{r}{l_c}\right)^2}, \quad (4.21)$$

where the parameter l_c is a characteristic length of micro-structure. The averaging in Eq. 4.20 is restricted to a small representative area around each material point (the influence of points at the distance of $r = 3l_c$ is only of 0.1%, see Fig. 4.22).

4.7 Other fracture parameters

It is possible to introduce more fracture parameters into the model. The parameters assumed by the author are listed in Tab. 4.6. The critical compressive strain is used to remove a beam when the beam strain in compression ${}_i\varepsilon$ (see Fig. 4.18) exceeds ε_{\max} . The shearing and bending angle is calculated using the angular difference between the current beam's orientation, and initial beam's orientation at which it would come back to the relaxed position with respect to neighboring beams. This angle is different on both ends of the beam. Therefore half of their difference and sum can be interpreted as geometrical equivalents of shearing:

$${}_i\tau = \frac{{}_i^A\alpha + {}_i^B\alpha}{2} \quad (4.22)$$

and bending:

$${}_i\chi = \frac{{}_i^A\alpha - {}_i^B\alpha}{2}, \quad (4.23)$$

where angles ${}^{A,B}{}_i\alpha$ are the original angular position of a beam with respect to its neighboring beams. Similarly a maximum torsion angle can be introduced.

Table 4.6: Additional fracture parameters

Additional fracture parameters	
ε_{\max}	critical compressive strain [-]
τ_{\max}	critical shearing [rad]
χ_{\max}	critical bending [rad]

4.8 Stress and force

The Lattice Geometrical Model, due to its geometrical nature does not include initially properties such as stresses, or forces. However it is possible to obtain stresses in a single beam prescribing the Young's modulus E . The resultant force F can be calculated in a selected specimen's cross-area A , with the aid of corresponding normal strains in each beam ${}_i\varepsilon$ (Fig. 4.7) projected on the unit normal to the cross-section plane, longitudinal stiffness parameters ${}_i k_l$, and moduli of elasticity ${}_i E$:

$$F = A \sum_i {}_i \varepsilon ({}_i \vec{n} \cdot \vec{n}_A) {}_i k_l {}_i E, \quad (4.24)$$

where the sum is made over all beams that intersect the selected specimen's cross-section A and \vec{n}_A is the unit normal of the cross-section A .

4.9 Model properties

The most important property of this model, distinguishing it from other lattice models, is that during bending the angle between elements changes. This property has following consequences: first, it is not possible to transfer shearing from one beam's end to another (although it's possible to calculate the shearing angle). Second, it is possible to achieve the Poisson's ratio in the full range ($\nu \in \langle -1, 0.5 \rangle$, Sect. 4.9.2).

Another property of this model is that it works without time, mass and beam area. It is possible to calculate forces (see Sect. 4.8) by introducing the modulus of elasticity.

4.9.1 Conservation of motion and shape

The model does not have kinetic energy or energy of deformation. However it exhibits the conservation of movement (aka kinetic energy) and, to some extent, conservation of deformation (aka energy of deformation).

The conservation of movement is achieved by component ${}_i \vec{N}$ (Eq. 4.1) – the beam always repeats its movement from the previous calculation step. Thus if the beam is not constrained (like other beams that move in different directions) the beam moves or rotates with constant velocity. This conservation of movement propagates from the single beam to the whole body. Figure 4.23 a shows a specimen without supports with an initial horizontal displacement in the first calculation step. Figure 4.23 b shows the increasing distance between points A and B . Thus movement with a constant velocity never stops. Figure 4.24 a shows a specimen connected to a single support and with an initial vertical displacement in the first calculation step. The oscillation of distance between points A and B can be observed, which means that the specimen preserves some of its rotational velocity. After large number of steps the rotation will cease.

Deformation is conserved by all components of ${}_i \vec{K}$ (Eq. 4.2). Figure 4.25 a shows a specimen with an initial tensile deformation. The components of ${}_i \vec{N}$, also play an important role because they contribute to a faster propagation of the deformation wave in the material. Figure 4.25 b demonstrates that the specimen loses its elastic behavior without those components.

The „mass“ is not conserved (if one would call it a mass, since the elements have no mass), due to the beam removal during the fracture precess.

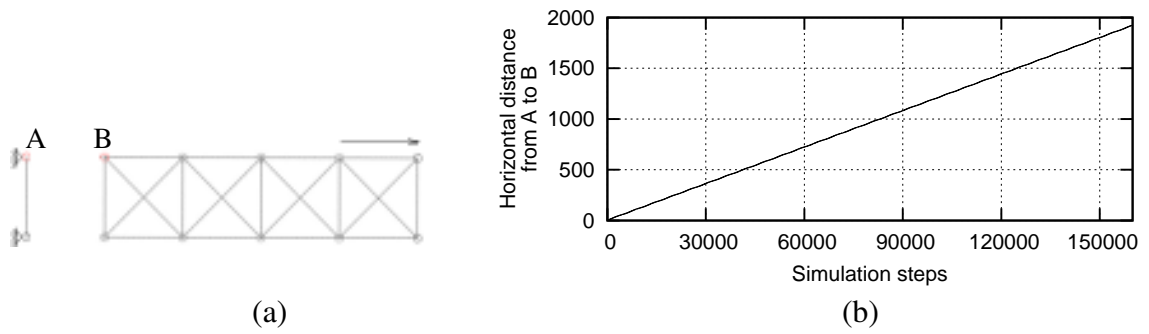


Figure 4.23: Conservation of movement: (a) induced horizontal displacement in the first step; (b) progressive forward movement of the specimen

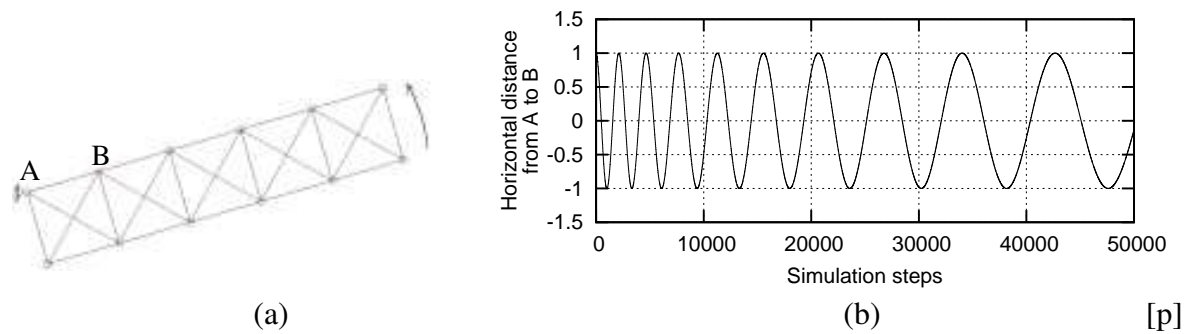


Figure 4.24: Conservation of rotational movement: (a) vertical displacement induced in the first step; (b) rotation around the support ceases after large number of steps

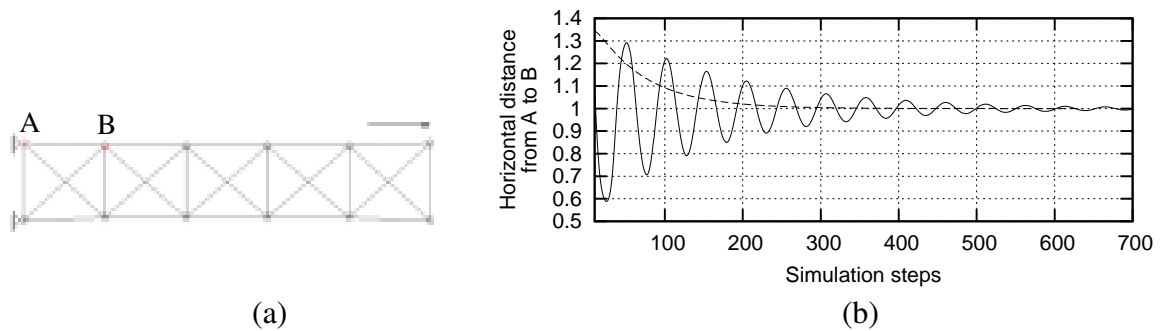


Figure 4.25: Shape conservation in tension: (a) induced horizontal displacement in the first step; (b) oscillation of the specimen until reaching its original shape (solid line – full version of the model, dashed line – without translation component $i\vec{N}$ (Eq. 4.1))

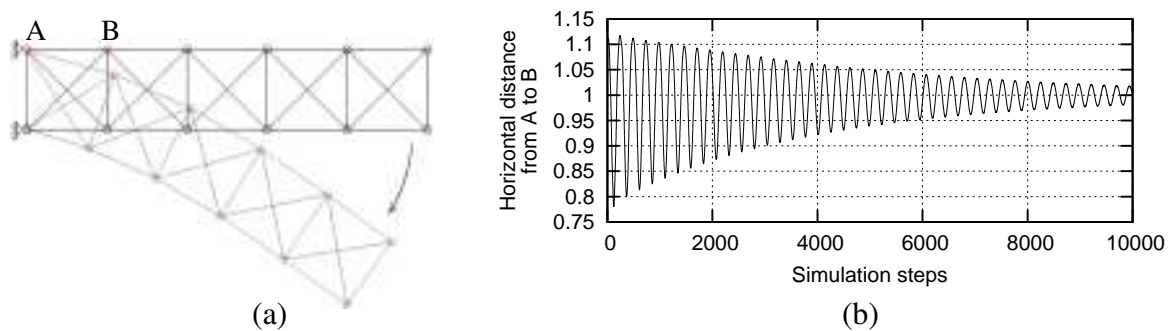


Figure 4.26: Shape conservation in bending: (a) vertical displacement induced in the first step; (b) slow convergence back to its original shape

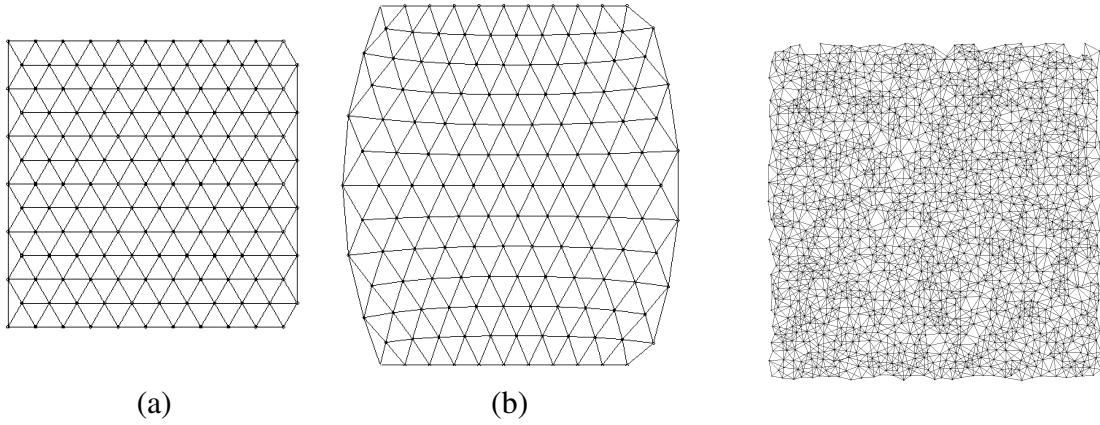


Figure 4.27: Sample specimen with regular mesh subject to uniaxial extension with rough edges, for ratio between the bending and longitudinal stiffness $p = 100$ ($k_b = 100$ and $k_l = 1$): (a) before extension; (b) after extension

Figure 4.28: Sample irregular mesh used for examination of Poisson's ratio

4.9.2 Poisson's ratio

If the lattice mesh used in this model is considered as a continuum material (which it aims to modelize), then the Poisson's ratio ν can be measured.

If the bending stiffness k_b is several times higher than the longitudinal stiffness k_l , there exists the absolute priority in the model to preserve the original angle between beams, while ignoring their need to preserve the original length. Thus the ratio p between the bending and longitudinal stiffness:

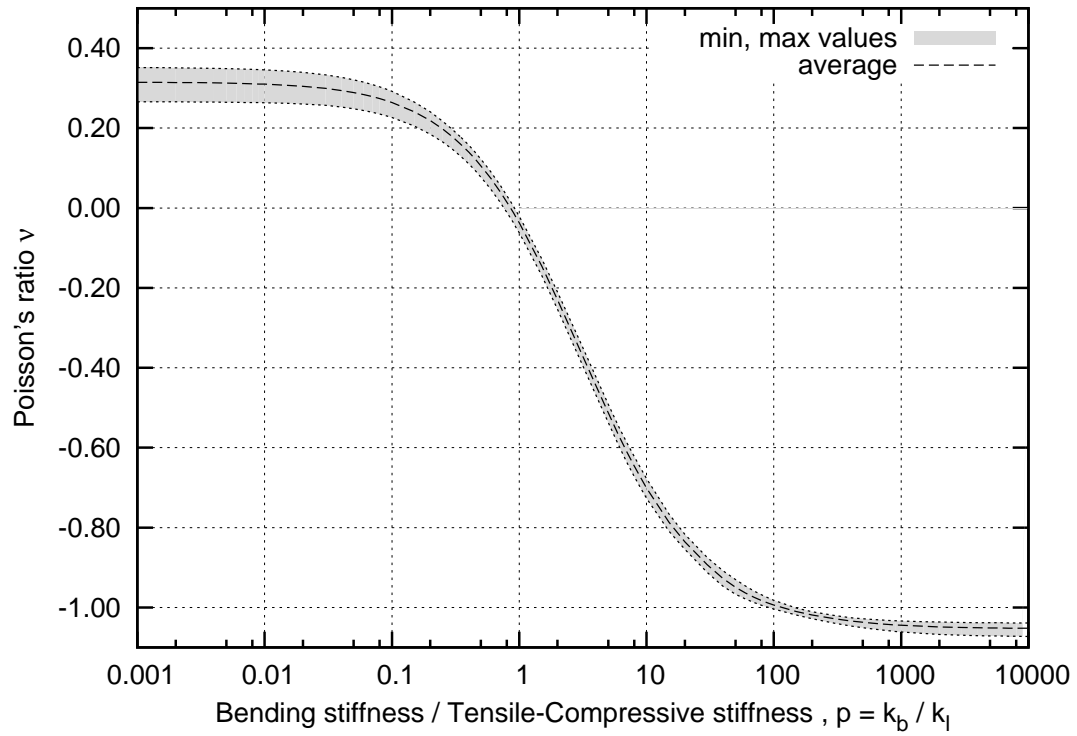
$$p = \frac{k_b}{k_l}, \quad (4.25)$$

has the major influence on the Poisson's ratio. Figure 4.27 shows a specimen subject to uniaxial extension with rough edges using $p = 100$.

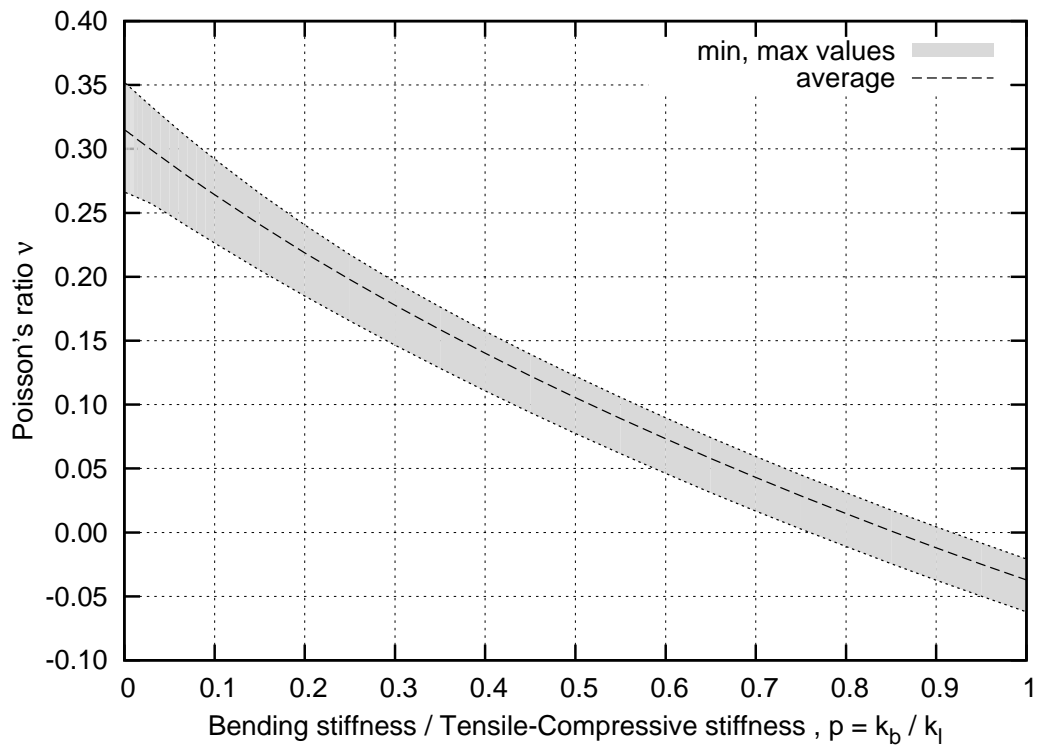
To measure the change of Poisson's ratio ν versus p , a set of 800 randomly generated specimens were used, using both mesh generation types (with $s = 0.6g$, $\alpha = 20^\circ$, $r_{\max} = 2g$, $g = 1$ mm, Sect. 4.5.1), one of which is shown in Fig. 4.28. Figure 4.29 shows the influence of p on Poisson's ratio ν . Tension and compression were considered with smooth horizontal edges at the beginning of the deformation process. The fracture criterion was not used (beams were not removed). If the stiffness parameter $p = 0.1$, the Poisson's ratio is 0.3. In turn, if the parameter $p > 1$, the Poisson's ratio became negative (with the smallest value approximately equal to $\nu \cong -1.0$ at $p = 10000$). The behavior of beams with values of p approaching zero corresponds obviously to that of bars [90, 96, 97].

4.9.3 Effect of mesh type and density

To demonstrate the effect of mesh generation type, eight specimens of a size 10×10 cm² with a small imperfection in the middle were generated using parameters listed in Tab. 4.7. The specimens were subject to uniaxial extension in a vertical direction applied to the top and bottom (displacement in each calculation step was $t = 2 \cdot 10^{-8}$ m), with smooth edges. Figure 4.30 shows the specimens after failure.



(a)



(b)

Figure 4.29: Influence of the ratio between the bending and longitudinal stiffness $p = k_b / k_l$ on Poisson's ratio ν during uniaxial compression and extension with smooth edges: (a) for $p \in \langle 0, 10000 \rangle$ (using semi-logarithmic scale); (b) for $p \in \langle 0, 1 \rangle$

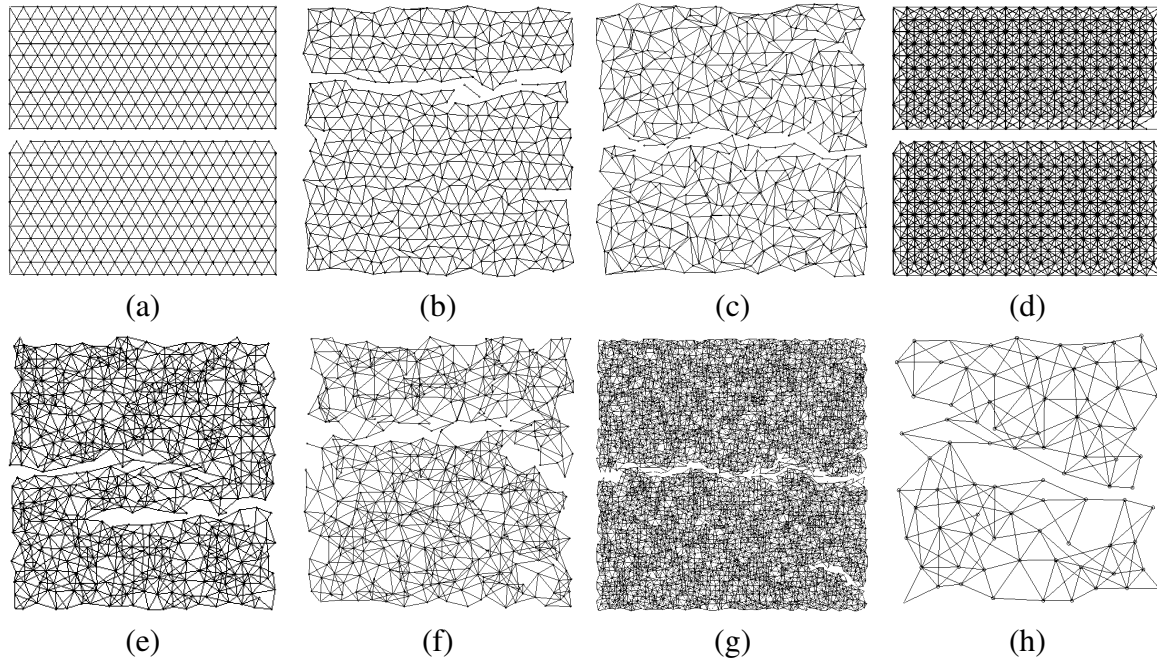


Figure 4.30: Meshes used to demonstrate the effect of mesh type and density (after failure), according to Table 4.7

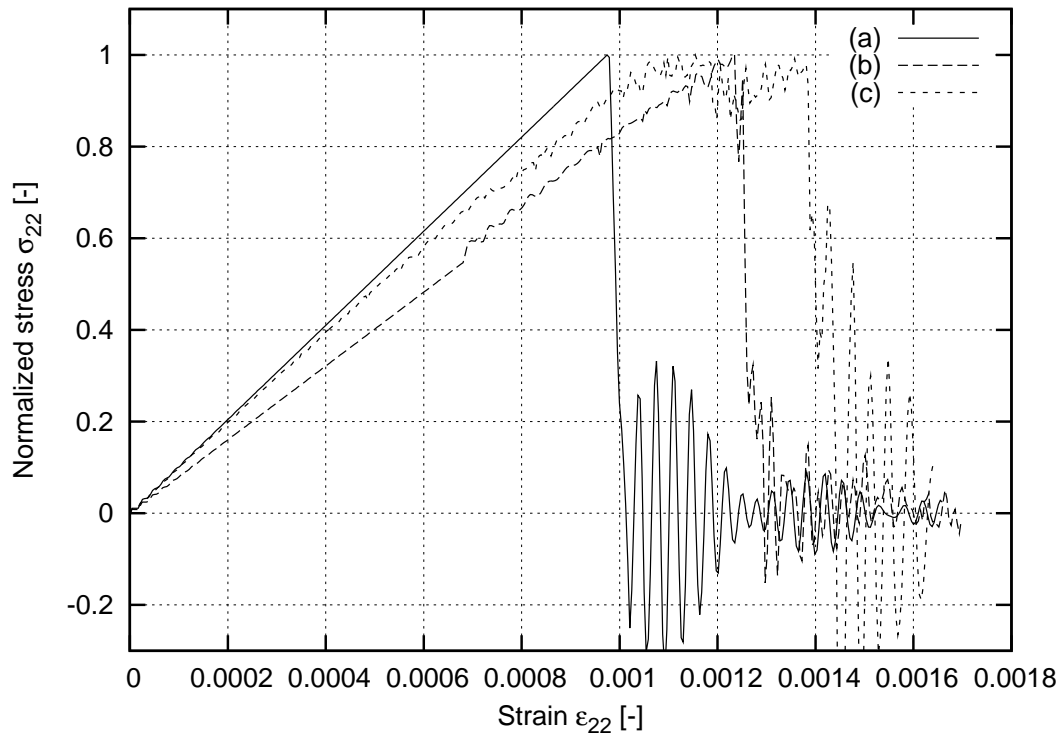
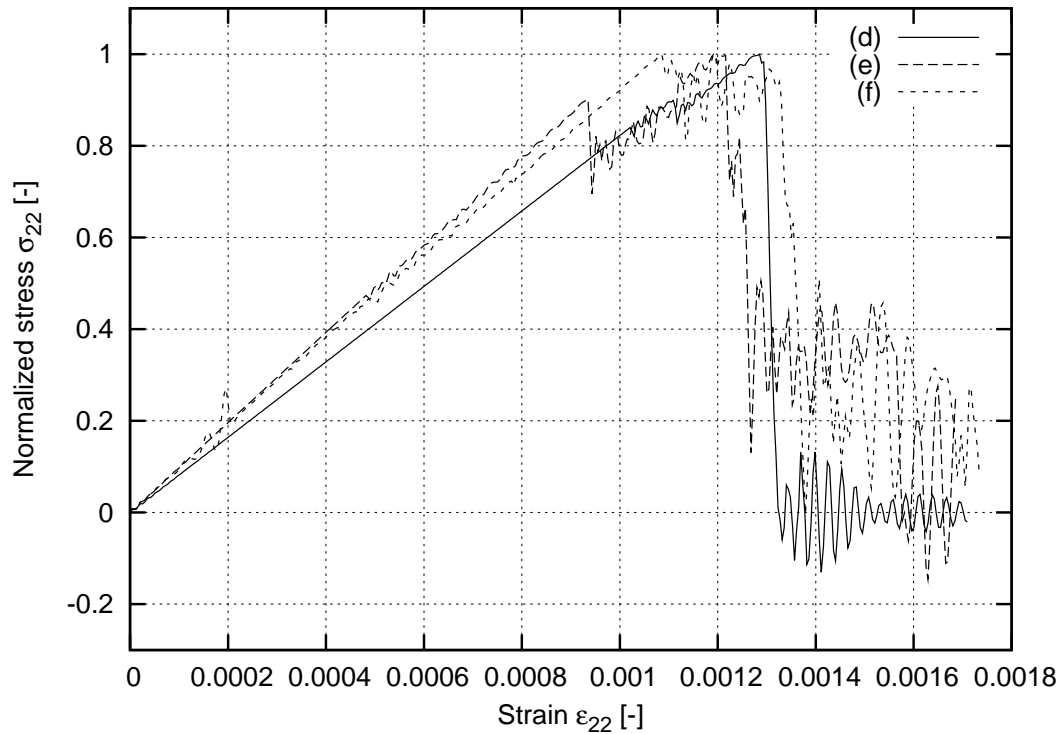


Figure 4.31: Effect of mesh randomness parameter s on the stress–strain curve during uniaxial extension, using Delaunay mesh generation (σ_{22} – normalized vertical stress, ϵ_{22} – vertical strain, parameters as in Table 4.7)

Table 4.7: Parameters used to demonstrate the effect of mesh type and density

No.	Stiffness		Fracture ϵ_{\min}	Mesh generation				
	k_l	k_b		Delaunay	g	r_{\max}	α	s
(a)	1	0.6	0.002	+	0.005 m	—	—	0.0g
(b)	1	0.6	0.002	+	0.005 m	—	—	0.6g
(c)	1	0.6	0.002	+	0.005 m	—	—	1.2g
(d)	1	0.6	0.002	—	0.005 m	1.9g	20°	0.0g
(e)	1	0.6	0.002	—	0.005 m	1.9g	20°	0.6g
(f)	1	0.6	0.002	—	0.005 m	1.9g	20°	1.2g
(g)	1	0.6	0.002	—	0.002 m	1.9g	20°	0.6g
(h)	1	0.6	0.002	—	0.012 m	1.9g	20°	0.6g

Figure 4.32: Effect of mesh randomness parameter s on the stress–strain curve during uniaxial extension, using non–Delaunay mesh generation (parameters as in Table 4.7)

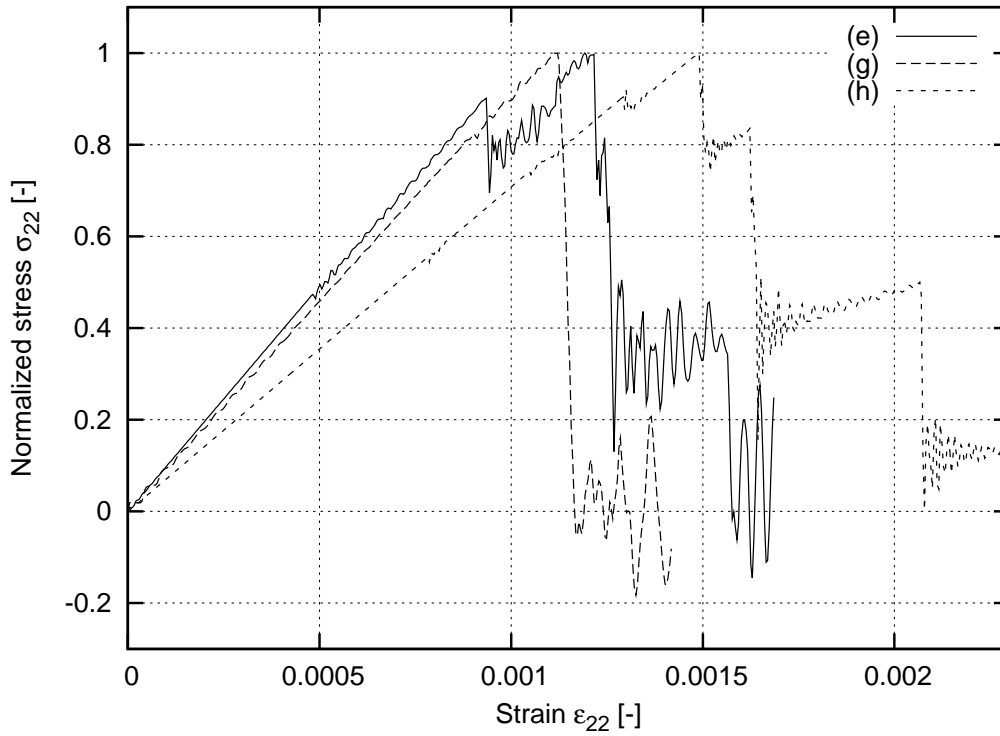


Figure 4.33: Effect of mesh density parameter g on the stress–strain curve during uniaxial extension, using non–Delaunay mesh generation (parameters as in Table 4.7)

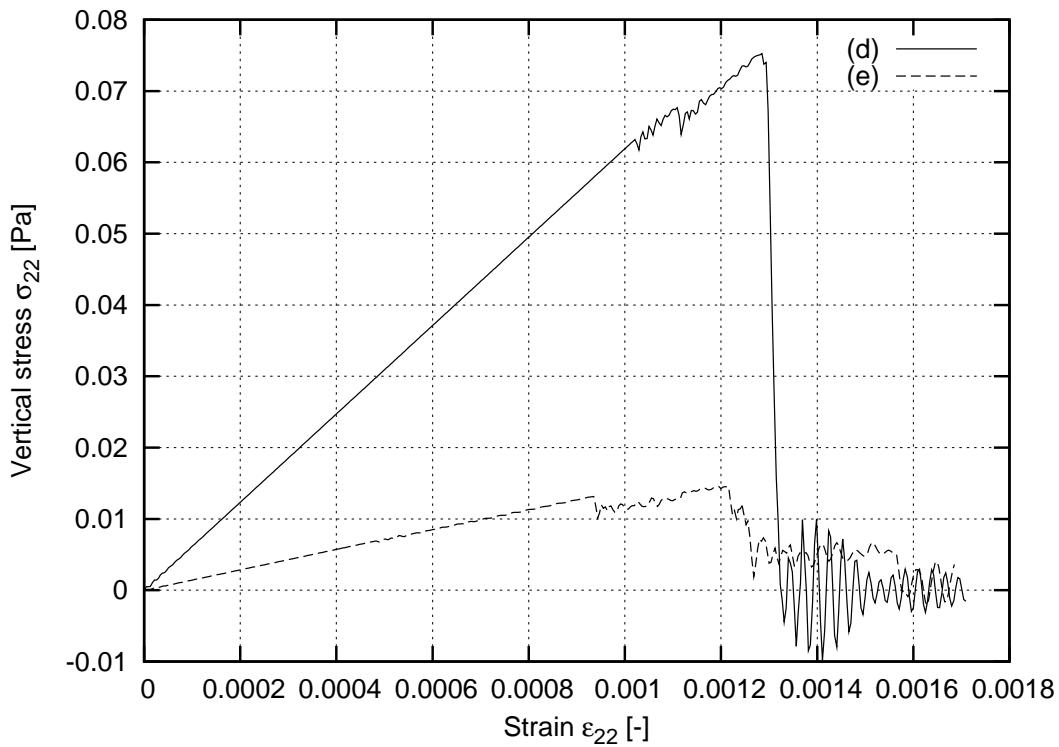


Figure 4.34: Effect of number of beams intersecting the cross section on normal vertical stress σ_{22} (parameters as in Table 4.7)

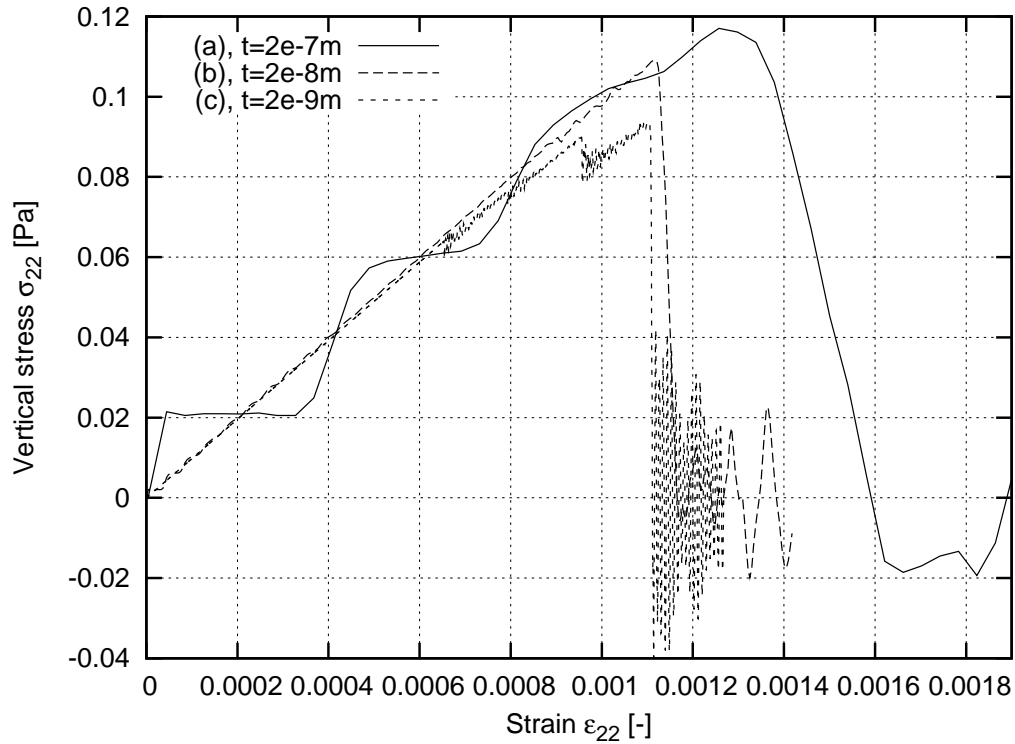


Figure 4.35: Effect of strain increment on normal vertical stress σ_{22} (parameters as in Table 4.7): (a) $t = 2 \cdot 10^{-7}$, $\Delta\epsilon = 0.000002\%$; (b) $t = 2 \cdot 10^{-8}$, $\Delta\epsilon = 0.0000002\%$; (c) $t = 2 \cdot 10^{-9}$, $\Delta\epsilon = 0.00000002\%$

Figure 4.31 shows the effect of the mesh randomness on the stress–strain curve (mesh was generated using the Delaunay method.) The regular mesh ($s = 0g$, Fig. 4.31 a, Figure 4.30 a) causes the most brittle behavior. The irregular mesh ($s = 1.2g$, Fig. 4.31 c, Figure 4.30 c) causes the less brittle behavior. The effect of the mesh irregularity is smaller, when the non–Delaunay generation method was used (Fig. 4.32), although it is still present. Figure 4.33 shows the effect of the mesh density on the stress–strain curve. The mesh with a big cell size ($g = 0.012$ m, Fig. 4.33 h, Fig. 4.30 h) during failure is ductile, and each jump on the softening curve corresponds to the removal of a single beam from the lattice. In turn, meshes with smaller cell sizes appear to be more brittle, and the jumps become less pronounced, because there are more beams in the lattice.

An increase of α and a decrease r_{\max} have the same effect as using Delaunay mesh generation method instead of non–Delaunay – they increase the material brittleness, and decrease the peak strain value. Apart from that, the mesh generation parameters have little impact on the brittle/ductile behavior of the material (it is always more or less brittle).

In Figures 4.31–4.33 the normalized vertical stress σ_{22} was calculated, so that the curves can be compared with respect to their shape and peak strain. However in each mesh, a different number of beams intersect the cross–section A , therefore the non–normalized stress will be different for each specimen. Figure 4.34 shows the stress σ_{22} carried by two specimens (d) and (e) (Tab. 4.7), with the assumed modulus of elasticity $E = 1$ Pa. Therefore the total strength carried by the material is largely influenced by the mesh generation type and parameters α , r_{\max} and g . This can be considered as a material „density” – a more dense mesh can carry a higher load.

4.9.4 Effect of strain increment

In the previous Section 4.9.3, all calculations were performed using the displacement in each increment step $t = 2 \cdot 10^{-8}$ m. Therefore, the strain increment $\Delta\epsilon = \frac{2 \cdot 10^{-8}}{0.1 \text{ m}} = 0.0000002\%$. Figure 4.35 shows the influence of different strain increments on the stress–strain curve (using specimen (g) from Tab. 4.7). There is a time/precision problem selecting too big strain increment (Figure 4.35 a), and selecting too small one (Figure 4.35 c). The difference between the stress–strain curves for the case (b) and (c) is small (however it exists). Thus there exists a maximum strain increment at which, the results are not affected. It turns out that it depends on how many calculation steps are needed to remove a single beam (due to the fracture criterion ϵ_{\min}), which is calculated as:

$$S_{critical} = \frac{g \epsilon_{\min}}{t}. \quad (4.26)$$

It has been found that the values: $S_{critical} > 500$ in tension and $S_{critical} > 500 \nu$ in compression (ν – Poisson’s ratio) produce results not influenced by the displacement value t .

4.10 CPU and memory consumption

The classical lattice model developed by van Mier, Schlangen, Lilliu and van Vliet [15, 99, 133–138, 153–158] uses the Finite Element Method for performing calculations [136, 156]. A global stiffness matrix is built and its inverse is required to obtain the response of the system. The size of a single lattice element has the same order of magnitude as the size of an aggregate (usually one third of an aggregate, $l_b < d_a^{\min}/3$), thus the number of elements for a small two–dimensional $10 \times 10 \text{ cm}^2$ specimen using Delaunay triangulation is 30 000 (for the element length of 1 mm), whereas for three–dimensions ($10 \times 10 \times 10 \text{ cm}^3$) it is around 8 000 000 elements. The effort to inverse the stiffness matrix has been recently reduced by using a conjugate gradient solver [136]. But the size of simulation still remains prohibitive for larger systems [136], due to calculation time and memory requirements. Moreover, the calculation is of the implicit type, which makes it difficult to distribute it using several computers and multiple processors. The calculations using the classical lattice model require from 7 hours to 158 days to be completed [156] (without using the optimizations by Schlangen [136]), however there is no comparison between the number of elements and time spent.

The computations performed in Sect. 4.9.3 with the newly developed Lattice Geometrical Model require from 5 minutes (for specimens (a)–(f)), with the beam length equal ≈ 5 mm (Tab. 4.7, parameter g) to 20 minutes for the specimen (g) with the beam length of ≈ 2 mm (specimen size $10 \times 10 \text{ cm}^2$). The calculations of a two–dimensional $10 \times 10 \text{ cm}^2$ specimen with the beam length ≈ 1 mm last about 2 up to 10 hours (depending on the strain increment) and consumes about 70 MB of RAM. The three–dimensional specimen $10 \times 10 \times 10 \text{ cm}^3$ calculated in Sect. 5.7 with the beam length ≈ 3.3 mm (subjected to tension) needs 14 hours to be completed (it included 380 000 beams and used 1.7 GB of the RAM memory). Calculating such specimen in the Nooru–Mohamed test (combined shear–extension) one needs about 4 days. However, the use of the beam length ≈ 1 mm in a 3D specimen $10 \times 10 \times 10 \text{ cm}^3$ would require 40 GB of the RAM memory (to store 8 000 000 beam elements) and the calculation would take 27 days.

It appears clear now that currently the limiting factor is not the processor speed (time of calculations) but the memory required. Fortunately, both problems can be solved (computation time and memory size) due to fact that the Lattice Geometrical Model is of explicit type, which means that the computation time and memory requirements will scale almost linearly with respect to the amount of processors and computers used in a parallelization method. AMD plans to release this year (2007) a quad-core processor on a motherboard with 4 processors, such a PC would reduce the calculations of 27 days into 2 days. Several computers together could sum up the available memory. The memory size requirements can be further reduced (eg. from 40 GB down to 10 GB), since in the current implementation of the model the author did not pay enough attention to this problem. This will be done next, along with the parallelization of the code. The Moore's Law implies that the size of the computer memory doubles every 2 years (and holds true since year 1980, the beginning of personal computers), which helps to predict when the newly developed model will be applicable for larger systems. The processor used for calculations of results presented in this thesis was AMD 4400 (dual-core), but currently only single processor was used.

Furthermore a multi-scale technique can be used by combining the model with coarse FEM meshes or by doing a coarse discretization (larger beam length) outside of the area with cracks.

With comparison to the classical lattice model [136, 156], which employs FEM, there is a major computation speed-up in the case of two-dimensional calculations. Most of the results presented in the Chapter 5 were obtained in time less than 10 hours.

Chapter 5

Verification of the model

This chapter presents the results obtained with a novel Lattice Geometrical Model developed by the author. First the parameters are calibrated and then comparison is done with own experimental results and experimental results from literature. The tests for tension, compression, combined extension–shear and three point bending are shown. The influence of aggregates on material response is investigated.

5.1 Calibration of model parameters

In this section the basic tests are simulated, to determine the effect of material parameters on the behavior of the simulated material. Finally, the material parameters are chosen that describe concrete in the most satisfactory way. In the calculation ϵ_{\min} is assumed 0.0002 because for this tensile strain the typical concrete specimen reaches its strength.

Uniaxial compression with smooth edges

The effect of the stiffness parameter $p = k_b/k_l$ on the evolution of the global stress–strain curve σ – ϵ (vertical normal stress versus the vertical strain) and crack propagation in a specimen during uniaxial compression with smooth edges is shown in Figs. 5.1 and 5.2 for parameters listed in Tab. 5.1 (specimen size $10 \times 10 \text{ cm}^2$, $\sigma = P/b$, $\epsilon = u_2/h$, P – global vertical force, u_2 – vertical displacement of the top edge).

The strength and ductility (ratio between the energy consumed during the fracture process after and before the peak) increase with increasing stiffness parameter p . The material becomes elastic for $p > 0.6$, quasi–brittle for $0.025 > p > 0.01$ and brittle for $p = 0.001$. In the last case, the vertical global strain corresponding to the material strength in compression is about 0.0003 (whereas critical tensile strain is $\epsilon_{\min} = 0.0002$). The cracks are predominantly vertical (parallel to the loading direction) if $p > 0.2$ (Fig. 5.2 a). In the case of $p < 0.1$, the predominant cracks are more inclined (Figures 5.2 b and 5.2 c).

First version of the model used both critical tensile strain ϵ_{\min} and critical compressive strain ϵ_{\max} , and the latter was assumed to be the ten times larger as the former, because concrete can carry approximately $10\times$ more stress in compression than in tension. Figures 5.3 and 5.4 show the results with $\epsilon_{\max} = 0.002$. The results are similar to those in previous case for $p \ll 0.3$. The strength increases with increasing p . The material is brittle for $p \leq 0.01$ and $p \geq 0.3$. For $p \ll 0.3$ the cracks are inclined (Fig. 5.4 j,l). In turn, for $p \geq 0.3$, the main crack is horizontal (perpendicular to the loading direction, Fig. 5.4 g).

Table 5.1: Assumed parameter values used to examine their effect on material behavior

No.	Stiffness		Fracture		Young's modulus	Mesh generation				
	k_l	k_b	ϵ_{\min}	ϵ_{\max}		Delaunay	g	r_{\max}	α	s
(a)	1	0.6	0.0002	—	20 GPa	—	0.001 m	1.9g	20°	0.6g
(b)	1	0.3	0.0002	—	20 GPa	—	0.001 m	1.9g	20°	0.6g
(c)	1	0.06	0.0002	—	20 GPa	—	0.001 m	1.9g	20°	0.6g
(d)	1	0.025	0.0002	—	20 GPa	—	0.001 m	1.9g	20°	0.6g
(e)	1	0.01	0.0002	—	20 GPa	—	0.001 m	1.9g	20°	0.6g
(f)	1	0.001	0.0002	—	20 GPa	—	0.001 m	1.9g	20°	0.6g
(g)	1	0.6	0.0002	0.002	20 GPa	—	0.001 m	1.9g	20°	0.6g
(h)	1	0.3	0.0002	0.002	20 GPa	—	0.001 m	1.9g	20°	0.6g
(i)	1	0.06	0.0002	0.002	20 GPa	—	0.001 m	1.9g	20°	0.6g
(j)	1	0.025	0.0002	0.002	20 GPa	—	0.001 m	1.9g	20°	0.6g
(k)	1	0.01	0.0002	0.002	20 GPa	—	0.001 m	1.9g	20°	0.6g
(l)	1	0.001	0.0002	0.002	20 GPa	—	0.001 m	1.9g	20°	0.6g

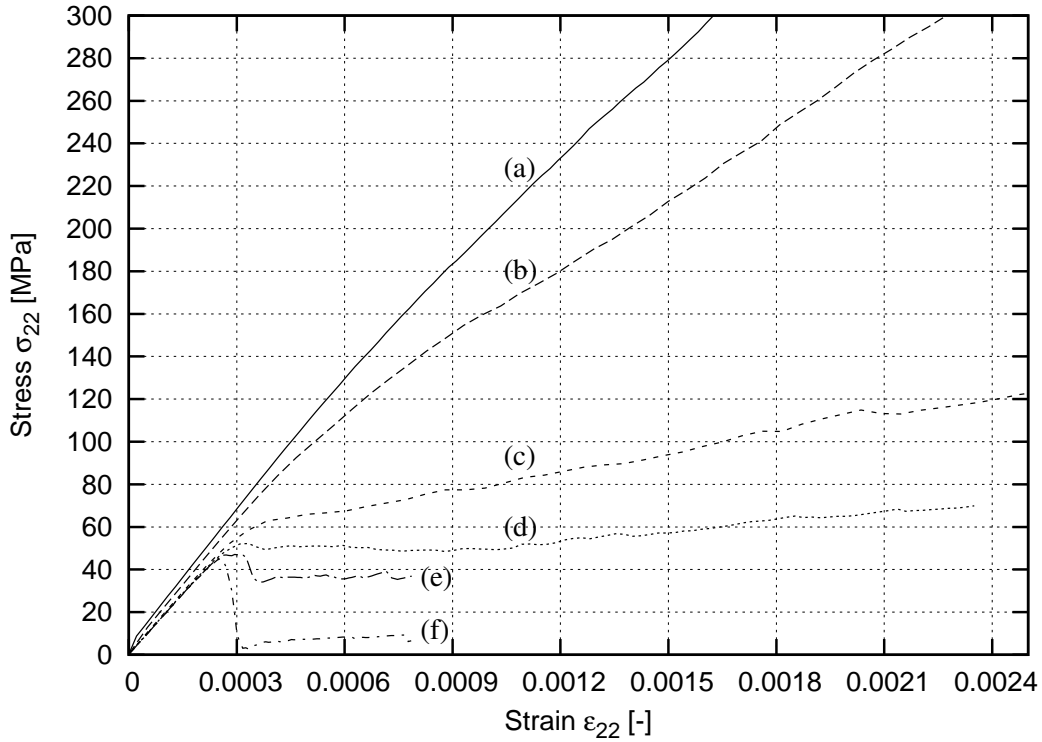


Figure 5.1: Effect of the stiffness ratio $p = k_b/k_l$ between the bending stiffness and longitudinal stiffness on the stress–strain curve during uniaxial compression with smooth edges (beams were removed when local $\epsilon_{\min} = 0.0002$), material parameters as in Table 5.1 (σ_{22} – vertical normal stress, ϵ_{22} – vertical normal strain)

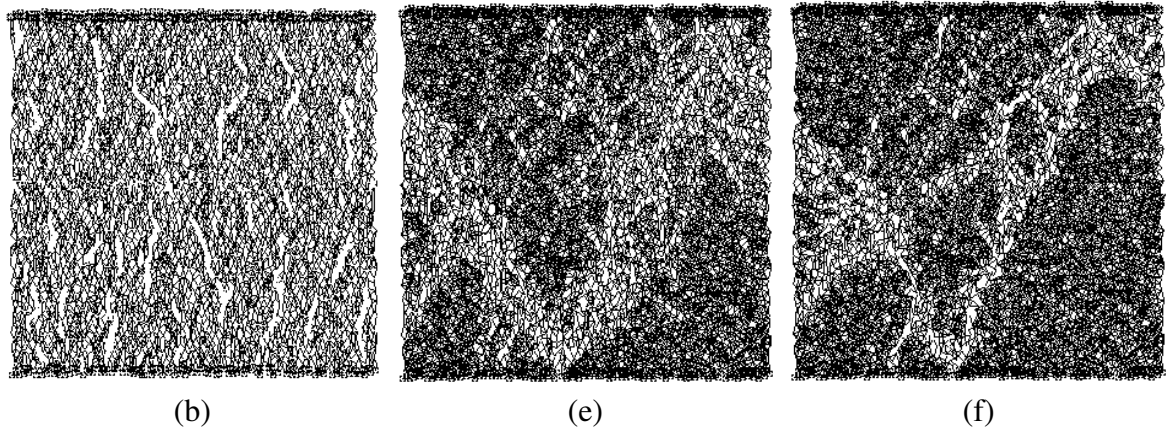


Figure 5.2: Effect of the stiffness ratio $p = k_b/k_l$ between the bending and longitudinal stiffness on the crack pattern during uniaxial compression with smooth edges (beams were removed when local $\epsilon_{\min} = 0.0002$), material parameters as in Table 5.1

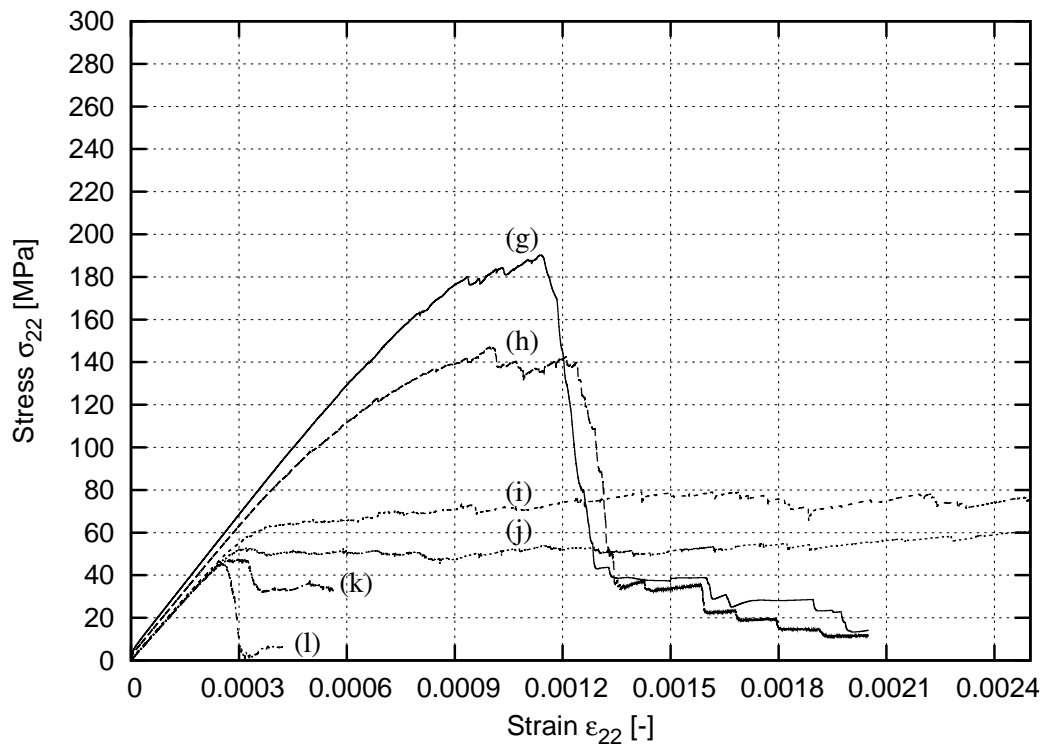


Figure 5.3: Effect of the stiffness ratio $p = k_b/k_l$ between the bending stiffness and longitudinal stiffness on the stress–strain curve during uniaxial compression with smooth edges (beams were removed when local $\epsilon_{\min} = 0.0002$ and $\epsilon_{\max} = 0.002$), material parameters as in Table 5.1 (σ_{22} – vertical normal stress, ϵ_{22} – vertical normal strain)

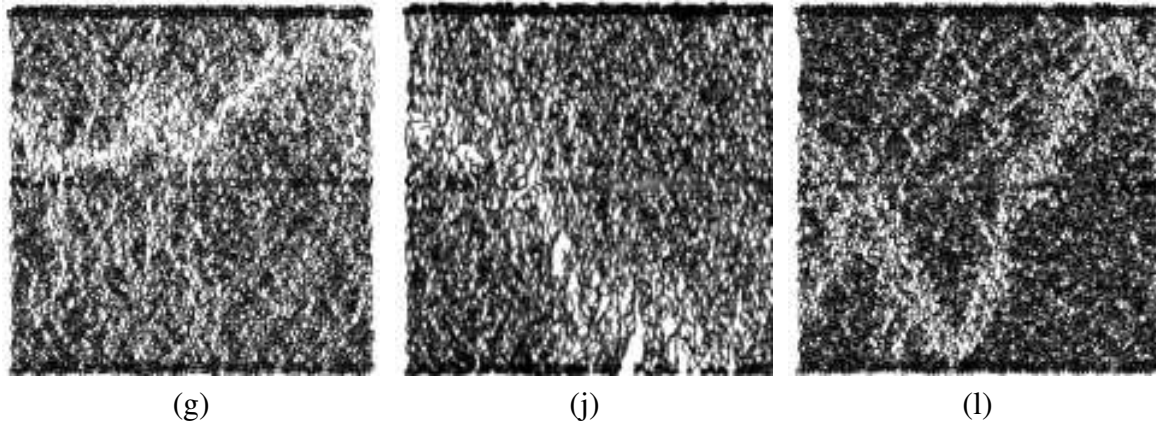


Figure 5.4: Effect of the stiffness ratio $p = k_b/k_l$ between the bending and longitudinal stiffness on the crack pattern during uniaxial compression with smooth edges (beams were removed when local $\epsilon_{\min} = 0.0002$ and $\epsilon_{\max} = 0.002$), material parameters as in Table 5.1

The effect of the roughness of both horizontal edges on the fracture process during uniaxial compression is shown on Fig. 5.5, for parameters (a) and (f) from Tab. 5.1. The results with very rough edges (horizontal displacements along both edges were assumed to be zero) indicate the appearance of diagonal intersecting cracks and stiff wedges in the specimen.

Uniaxial extension with smooth edges

The results for uniaxial tension with a small notch at mid-height of the left side and smooth horizontal edges are demonstrated in Fig. 5.6 (cases (b), (d) and (f) from Tab. 5.1). The material behaves in the elastic-purely brittle way for all values of p . The strength increases with increasing p , and the brittleness increases with decreasing p . The overall vertical strain corresponding to the peak stress values is about 0.00007–0.00009 (thus it is smaller than the local ϵ_{\min}). The crack pattern practically does not depend on the parameter p . The main crack is always initiated at the notch and then propagates almost horizontally through the specimen (Fig. 5.25 a).

Simple shear

Figure 5.7 shows the effect of $p = k_b/k_l$ on the fracture process in a specimen subject to simple shear (upper and lower surfaces move in opposite directions) for smooth and very rough edges. The main cracks are created in the direction perpendicular to the normal stress. The results show that the inclination angle of fracture due to shearing becomes more steep with increasing p (decreasing Poisson's ratio), and decreases in the case of smooth edges. In the case of smooth edges, more pronounced cracks are created.

Effect of other fracture parameters

The additional fracture parameters: critical shearing angle τ_{\max} and critical bending angle χ_{\max} were assumed (Tab. 4.6). Figure 5.8 shows the fracture process in the specimen subject to shearing. The results indicate that χ_{\max} controls the inclination angle of the fracture, and τ_{\max} controls the number of parallel cracks that occur. However with this additional rupture

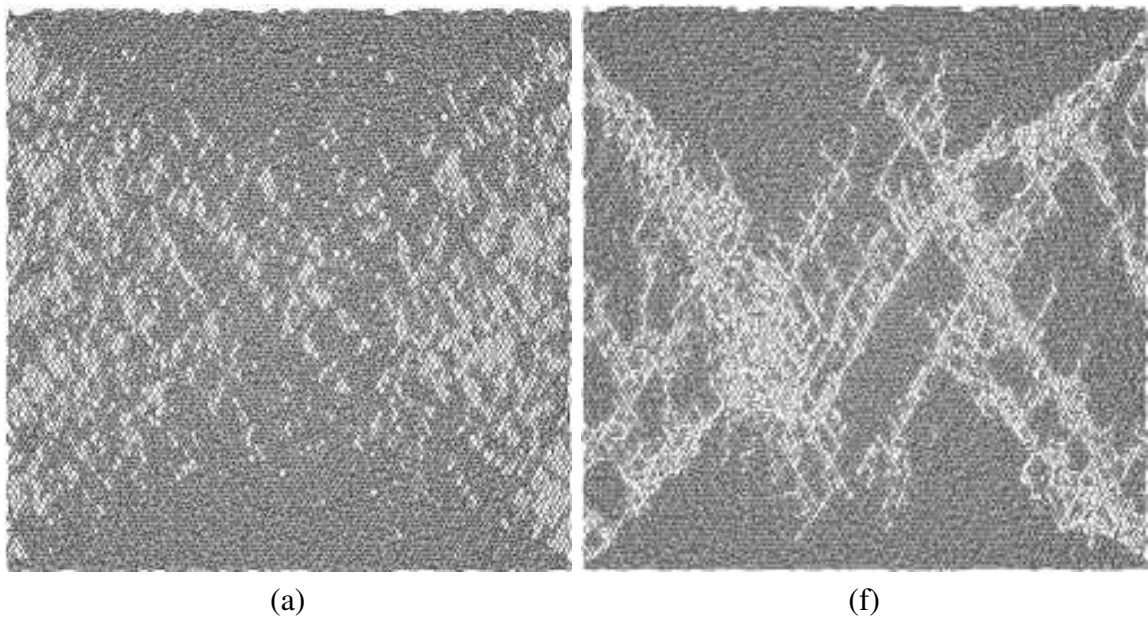


Figure 5.5: Effect of parameter $p = k_b/k_l$ on fracture process in compression with rough edges ($\epsilon_{\max} = 0.0002$), material parameters as in Table 5.1

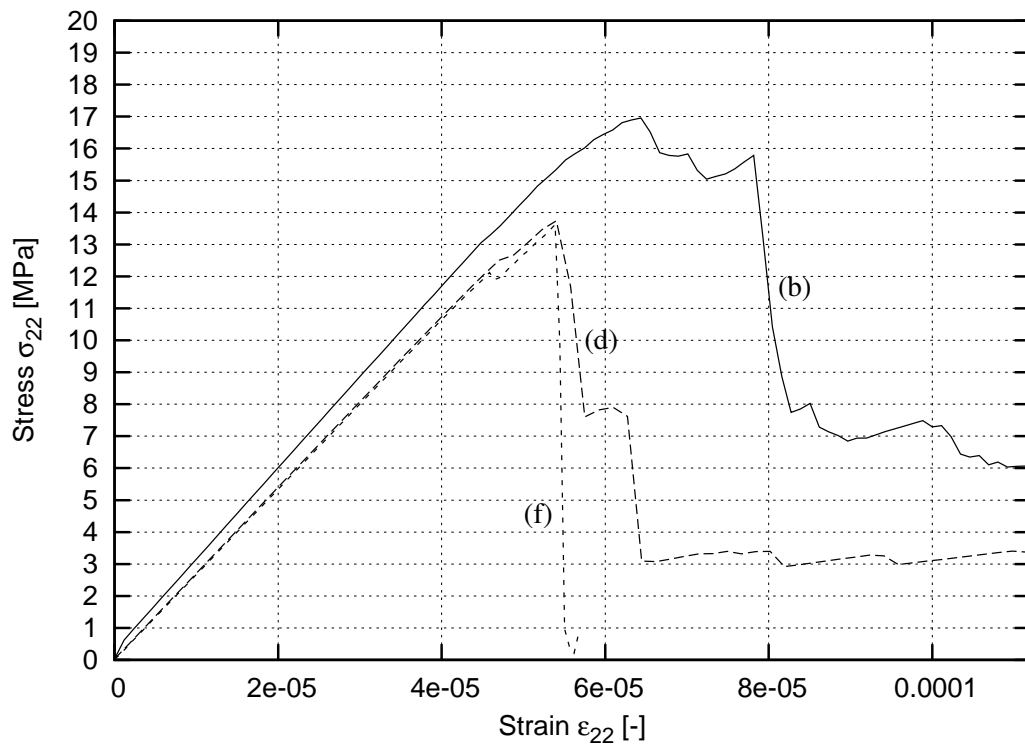


Figure 5.6: Effect of the stiffness ratio $p = k_b/k_l$ between the bending stiffness and longitudinal stiffness on the stress–strain curve during uniaxial extension (beams were removed when local $\epsilon_{\min} = 0.0002$), material parameters as in Table 5.1 (σ_{22} – vertical normal stress, ϵ_{22} – vertical normal strain)

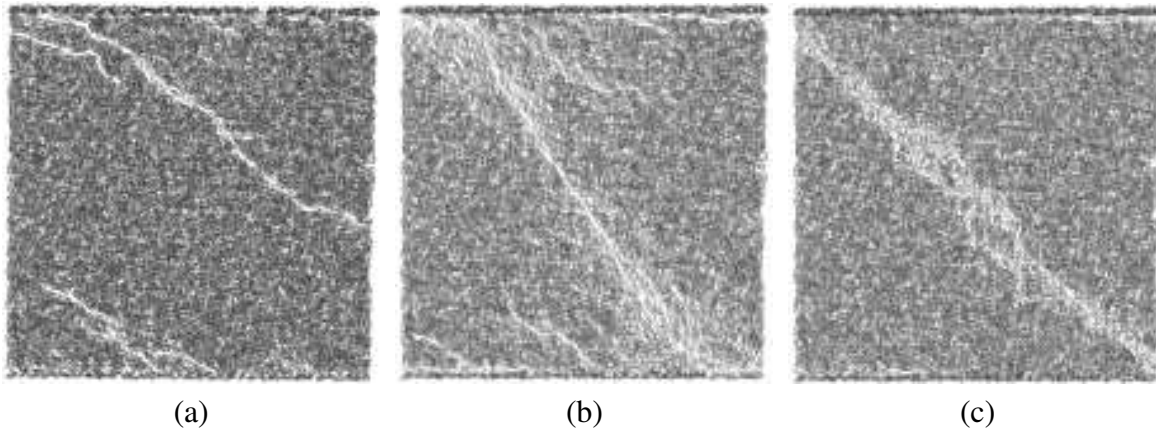


Figure 5.7: Effect of parameter $p = k_b/k_l$ on fracture process in simple shear ($\epsilon_{\min} = 0.0002$): (a) $p = 0.6$, smooth edges (b) $p = 0.6$, rough edges (c) $p = 0.01$

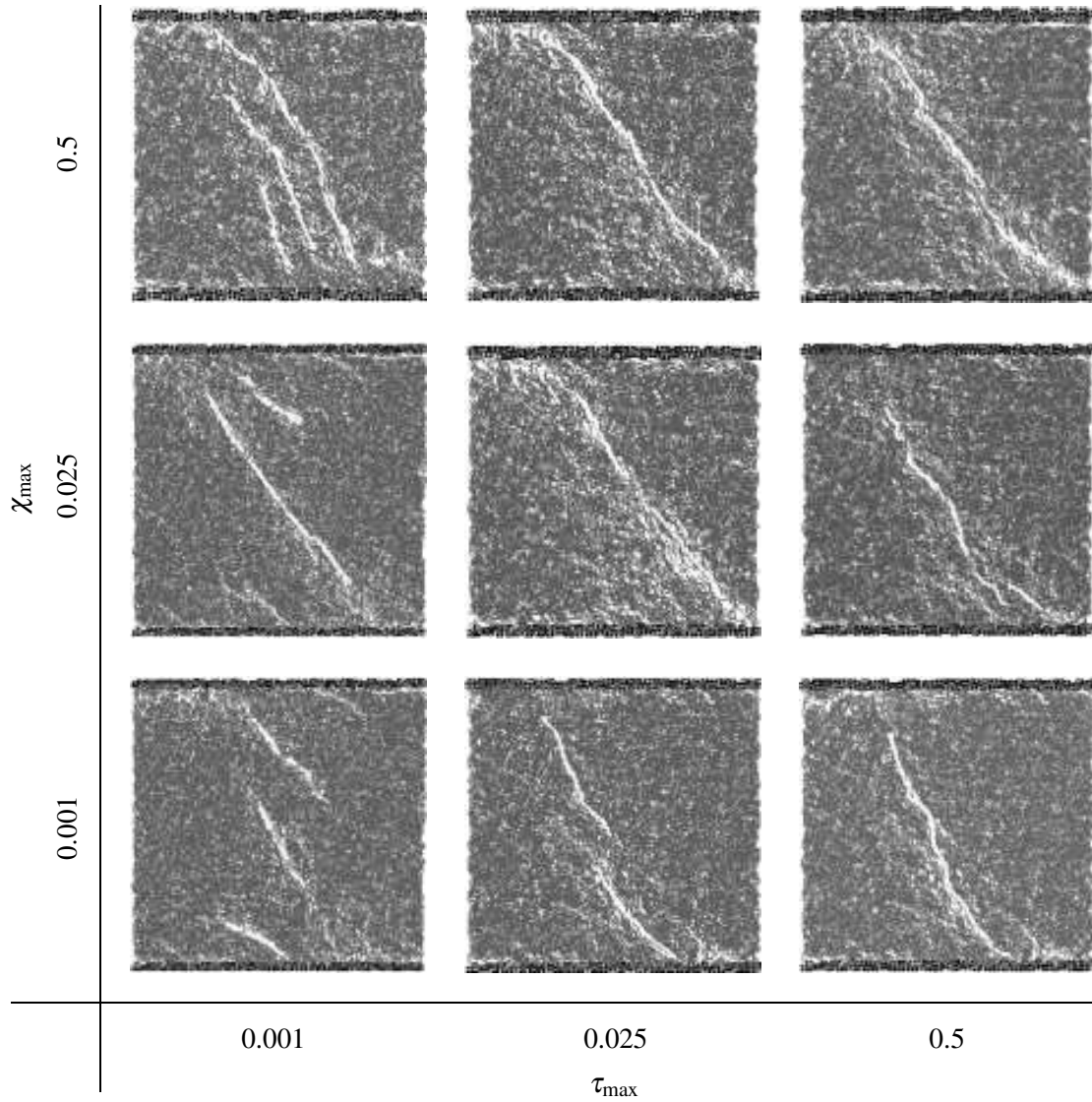


Figure 5.8: Fracture process in a specimen subject to shearing, depending of values of critical shearing angle τ_{\max} and critical bending angle χ_{\max} ($p = 0.01$)

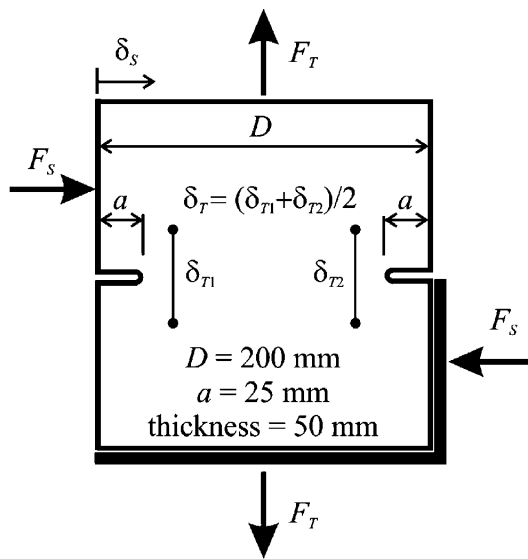


Figure 5.9: Experimental setup in Nooru–Mohamed test; [42]

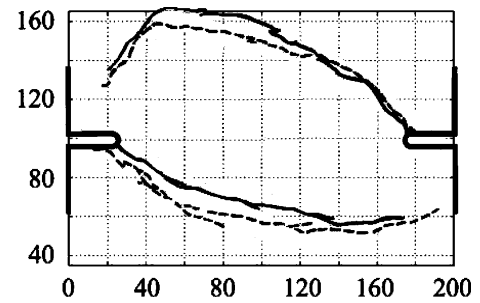


Figure 5.10: Experimental crack pattern in loading condition 4c; [42]

criterion, the material behaves more brittle. The gain of controlling the inclination angle of fracture does not justify the addition of these two parameters, especially that k_b can also be used to control this angle (Fig. 5.7).

Extension–shear test (Nooru–Mohamed test [111])

Figure 5.9 depicts the setup of the mixed–mode fracture test described in [111]. A double–notched prism (200 mm×200 mm×50 mm) with two 25 mm deep notches, was loaded in tension and shear. Various proportional and non–proportional loading paths were followed [111], with both displacement and load controls. This test is frequently used as a benchmark to check the ability of constitutive models to simulate complex crack paths. The loading path considered here is non–proportional, first the shear force F_s is increased up to a certain value, while tensile force F_T is kept zero. Afterwards, the shear force is kept constant while tensile force is applied under displacement control until the specimen fails completely. In these experiments, the shear force was kept constant at $5 \text{ kN} \approx F_{s,\max}/6$ (path 4a in [111]), $10 \text{ kN} \approx F_{s,\max}/3$ (path 4b in [111]) and $27.38 \text{ kN} = F_{s,\max}$ (path 4c in [111]), where $F_{s,\max}$ is the maximum force that the specimen could sustain in the absence of the tensile force.

Figure 5.10 shows the experimentally obtained crack path for the loading path 4c. Figure 5.11 shows results obtained with the the model. The parameter p has a major influence on the inclination angle of the resulting crack. For $p = 0.001$ (Figure 5.11 a), the crack is always horizontal, and the material behavior corresponds to that of rods. Increasing p up to 0.2 causes an appearance of the inclined crack, which is not curvilinear as it should be, and some beam elements still fail along the horizontal path (Figure 5.11 b). With $p = 0.6$, a correct fracture pattern is obtained for ϵ_{\min} 10× larger ($\epsilon_{\min} = 0.002$, Figure 5.11 c). In turn for $\epsilon_{\min} = 0.0002$, the obtained crack pattern with $p = 0.6$ is different than in the experiment (Figure 5.11 d) and the crack propagation does not start at the notch. This behavior is caused by the beam failure before the material could reach the skew–like shearing deformation responsible for the curvilinear crack path.

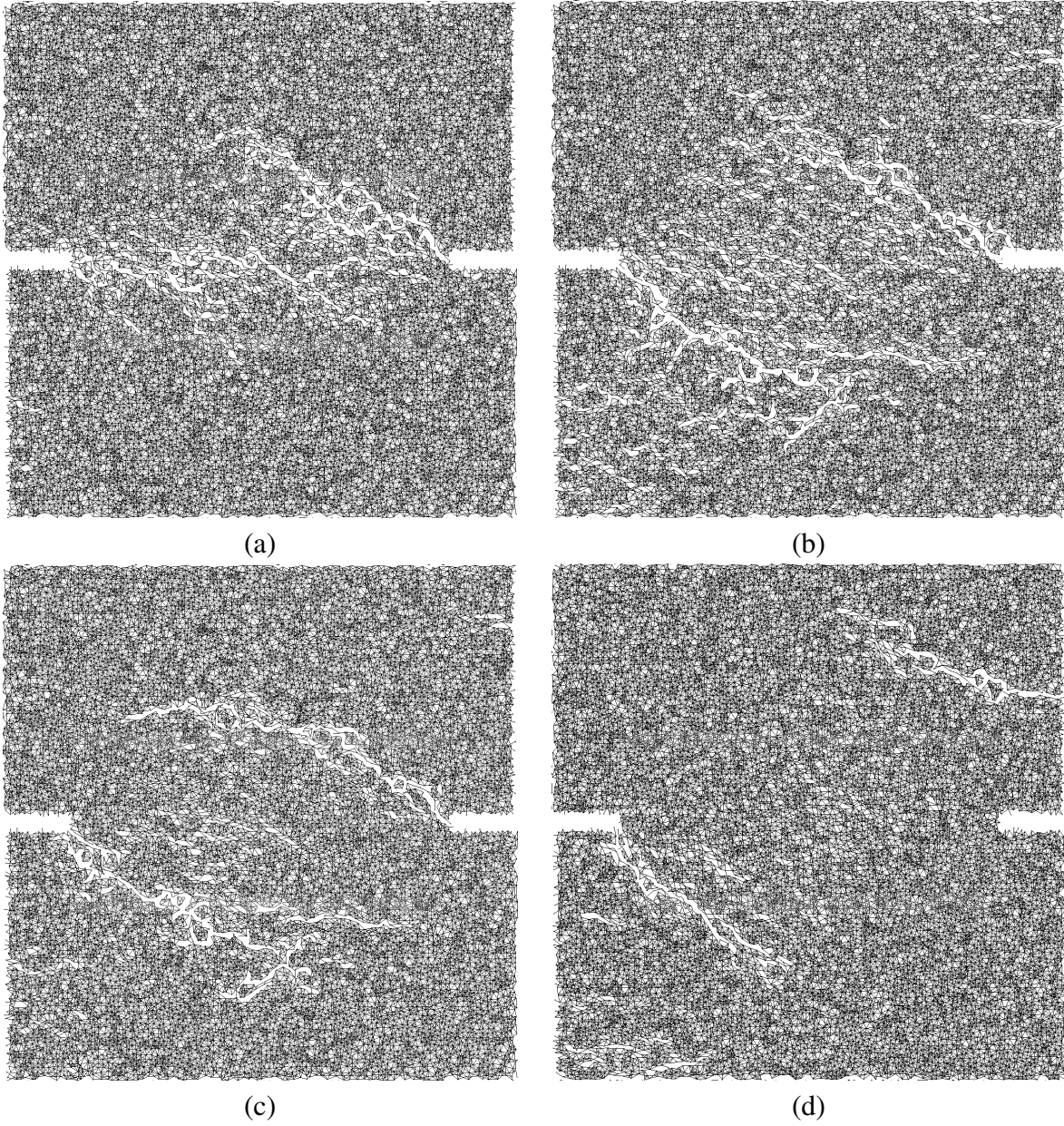


Figure 5.11: Effect of parameter $p = k_b/k_l$ on fracture process in Nooru–Mohamed test, with loading condition 4c: (a) $p = 0.001$ ($\epsilon_{\min} = 0.0002$), (b) $p = 0.2$ ($\epsilon_{\min} = 0.0002$), (c) $p = 0.6$, specimen fails correctly for $\epsilon_{\min} = 0.002$ (d) $p = 0.6$, specimen fails incorrectly for $\epsilon_{\min} = 0.0002$

Table 5.2: Parameters used to demonstrate the lack of effect of ϵ_{\min} on the stress–strain curve

Stiffness		Fracture ϵ_{\min}	Young's modulus	Mesh generation					
k_l	k_b			Delaunay	g	r_{\max}	α	s	
1	0.6	0.0002	20 GPa	—	0.001 m	1.9g	20°	0.6g	
0.1	0.06	0.002	20 GPa	—	0.001 m	1.9g	20°	0.6g	

The material parameters

Results presented in this section show that:

1. The peak compression strain in the model (if present) is around $10\times$ smaller than the peak compression strain of the concrete (Fig. 5.1 e,f and Fig. 5.3 k,l),
2. the peak extension strain is around $10\times$ smaller than the peak extension strain of concrete (Fig. 5.6),
3. the parameter ϵ_{\max} which influences beams subject to compression causes a non realistic fracture patterns (crack perpendicular to loading direction) in compression (Fig. 5.4 g) for values of parameter $p \geq 0.3$,
4. realistic fracture pattern in compression is obtained only for $p \leq 0.01$ (Fig. 5.4 f)
5. simulations of the Nooru–Mohamed test can produce correct results for $p \geq 0.6$ (Fig. 5.11 c) but only for ϵ_{\min} being $10\times$ larger,
6. crack pattern in uniaxial extension does not depend on p .

The ill fracture pattern when using ϵ_{\max} is the reason to abandon this parameter.

To solve the problem of the peak compression/extension strain being $10\times$ too small the local value of ϵ_{\min} must be assumed to be $10\times$ larger, and k_l and k_b parameters must be set $10\times$ smaller to retain the same modulus of elasticity 20 GPa. Figure 5.12 demonstrates the lack of the effect of ϵ_{\min} on the stress–strain curve in uniaxial extension for material parameters given in Tab. 5.2. This change, however, influences the produced crack patterns in the Nooru–Mohamed test so that a correct result is obtained (Fig. 5.11 c). Incidentally the

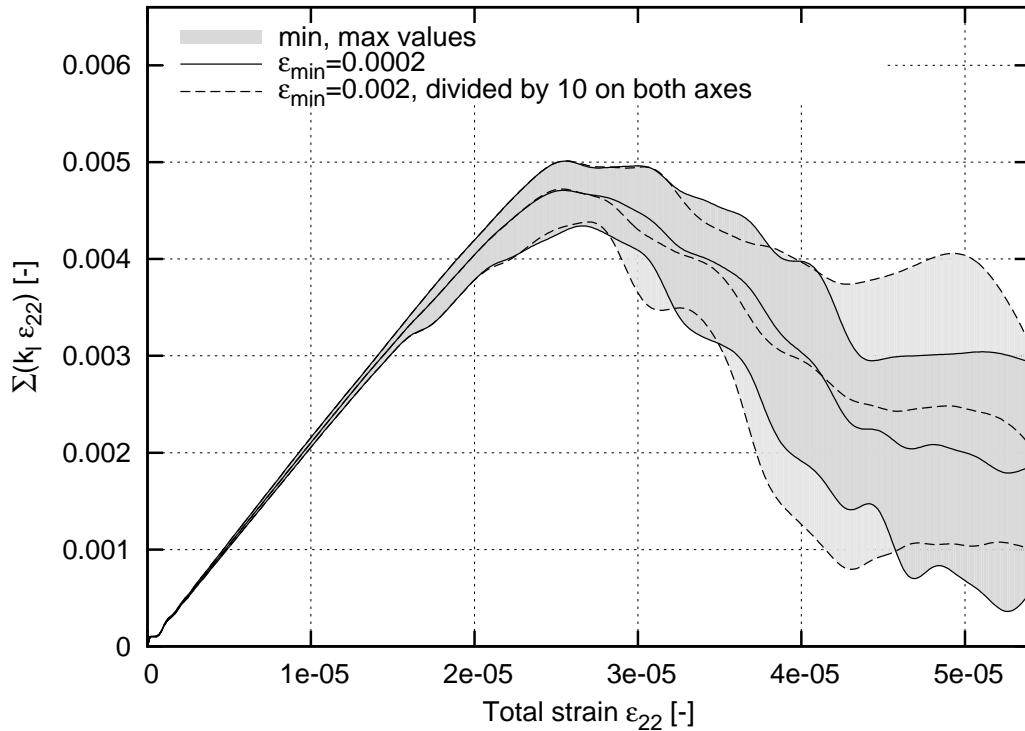


Figure 5.12: Effect of ϵ_{\min} on the shape of the stress–strain curve in uniaxial tension

$10\times$ higher value of $\epsilon_{\min} = 0.002$ corresponds well to the results obtained in experimental measurements of strain on the surface of concrete. Figure 3.23 b(B) shows the strain on the surface of concrete at the peak of the stress–strain curve, and the green color (value 0.002) dominates in the fracture zone. The red color (strain 0.005) corresponds to the crack that has already appeared (beam elements in the model were already removed).

A correct material failure in compression is obtained for $p \leq 0.01$ (Poisson’s ratio $\nu = 0.3$), while in combined–shear extension the value of parameter $p = 0.6$ is required. The proposed solution is to assume p smaller in tension and higher in compression. To choose p for compression one should not consider the stress–strain curves from Figure 5.1 because they change dramatically with the modified p . Figure 5.2 b shows a desired crack pattern for concrete for $p = 0.3$ (Poisson’s ratio $\nu = 0.17$), also Figure 5.2 e is not too far from correct ($p = 0.025$, Poisson’s ratio $\nu = 0.3$, Fig. 4.29). Therefore an arbitrary value in between was chosen $p = 0.2$. The change from 0.2 in compression to 0.6 in tension should be continuous to avoid instabilities in the calculations, therefore an arbitrary value of $\epsilon_{\min}/3$ was chosen as a limit for a linear transition between the two. Figure 5.13 a shows the assumed value of p as the function of the beam element strain, and the corresponding Poisson’s ratio is shown on Fig. 5.13 b.

The calibrated parameters for the Lattice Geometrical Model that should describe correctly the concrete material are listed in Tab. 5.3. Following sections demonstrate results with parameters based on those calibrated parameter values. In experiments with uniaxial tension the changes in parameter p (Fig. 5.13) do not affect the results.

Figure 5.14 shows the simulation results for a Nooru–Mohamed test for new calibrated parameters using a three–phase material from Tab. 5.6 (three–phase material is described in following sections). The results correspond well to results by Cusatis [42]. The results for uniaxial compression are presented in Sect. 5.6.

Table 5.3: The calibrated material parameters

Stiffness			Fracture	Young’s	Mesh generation				
k_l	$k_{b,tension}$	$k_{b,compr}$	ϵ_{\min}	modulus	Delaunay	g	r_{\max}	α	s
0.1	0.06	0.02	0.002	20 GPa	—	0.001 m	1.9g	20°	0.6g
0.1	0.06	0.02	0.002	20 GPa	✓	0.001 m	—	—	0.6g

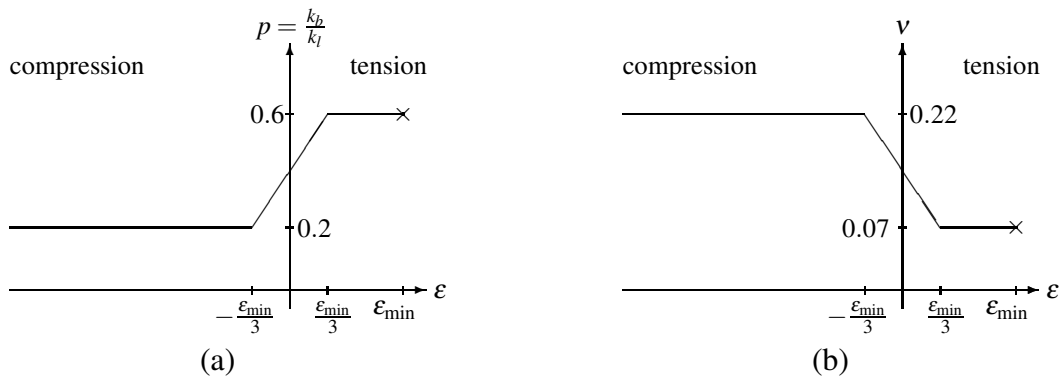


Figure 5.13: Change of parameter $p = k_b/k_l$ (a) and resulting change of Poisson’s ratio (b) in function of beam element strain ϵ

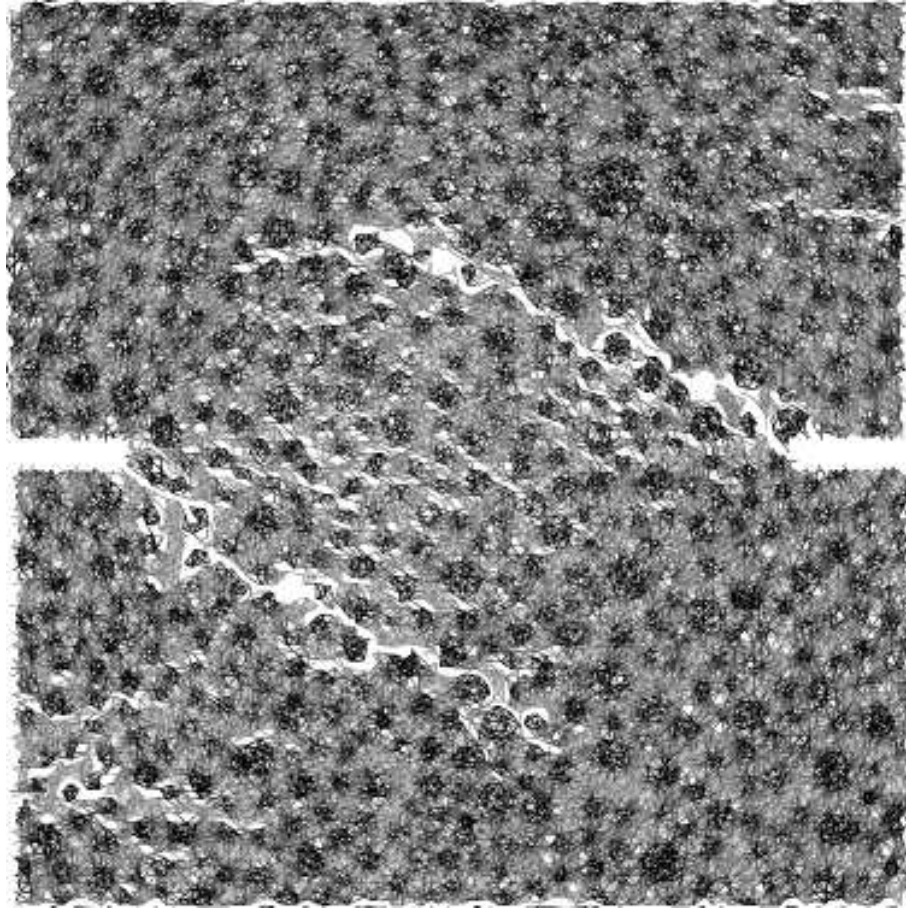


Figure 5.14: Fracture pattern in a Nooru–Mohamed test in three–phase material (gray denotes undamaged area, black color denotes aggregates, white color represents fracture damage)

5.2 Uniaxial tension (one and two–phase material)

Figures 5.15–5.16 show the effect of aggregates on the fracture behavior of 2D specimens under uniaxial extension (without interfacial transition zones).

The 2D calculations were carried out with a specimen size of $200 \times 200 \text{ mm}^2$ ($b \times h$) composed of 180 000 beam elements distributed non–uniformly ($\alpha = 20^\circ$, $s = 0.6g$, $g = 1 \text{ mm}$, $r_{\max} = 2g$). The minimum beam length was about 0.3 mm and the maximum one was about 2 mm.

The ratios between the parameters k_l and ϵ_{\min} for the cement matrix and aggregate were assumed on the basis of ratios between the elastic Young’s moduli and tensile strengths, respectively, assumed in [154] (see Sect. 2.2), as listed in Tab. 5.4.

Table 5.4: Paramaters used in calculation with two–phase material

Phase	Young’s modulus	$p = k_b/k_l$	k_l	local ϵ_{\min}
Cement matrix	$E_m = 20 \text{ GPa}$	$p_m = 0.6$	0.1	$\epsilon_m = 0.002$
Aggregate	$E_a = 60 \text{ GPa}$	$p_a = 0.6$	0.3	$\epsilon_a = 0.00133$

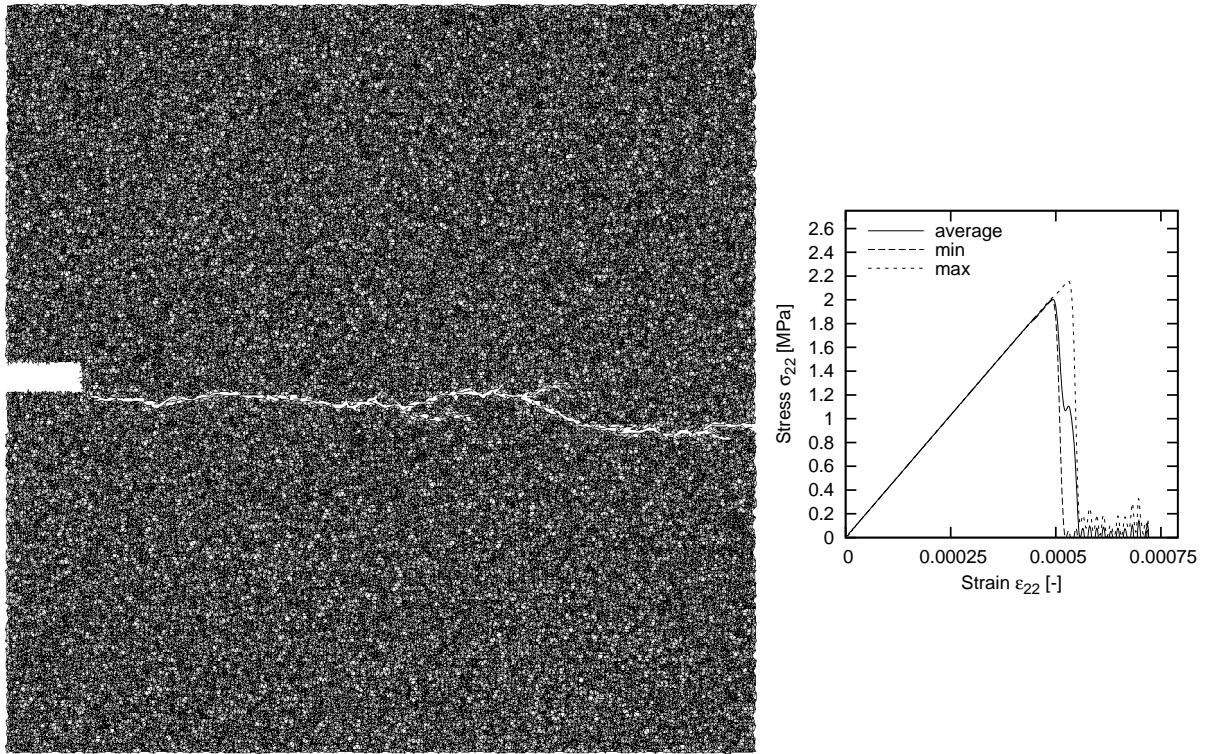


Figure 5.15: 2D specimen subject to uniaxial extension (one-phase material): $p = k_b/k_l = 0.7$ (with $k_l = 0.1$) and local $\varepsilon_{\min} = 0.0002$ (as in Tab. 5.4)

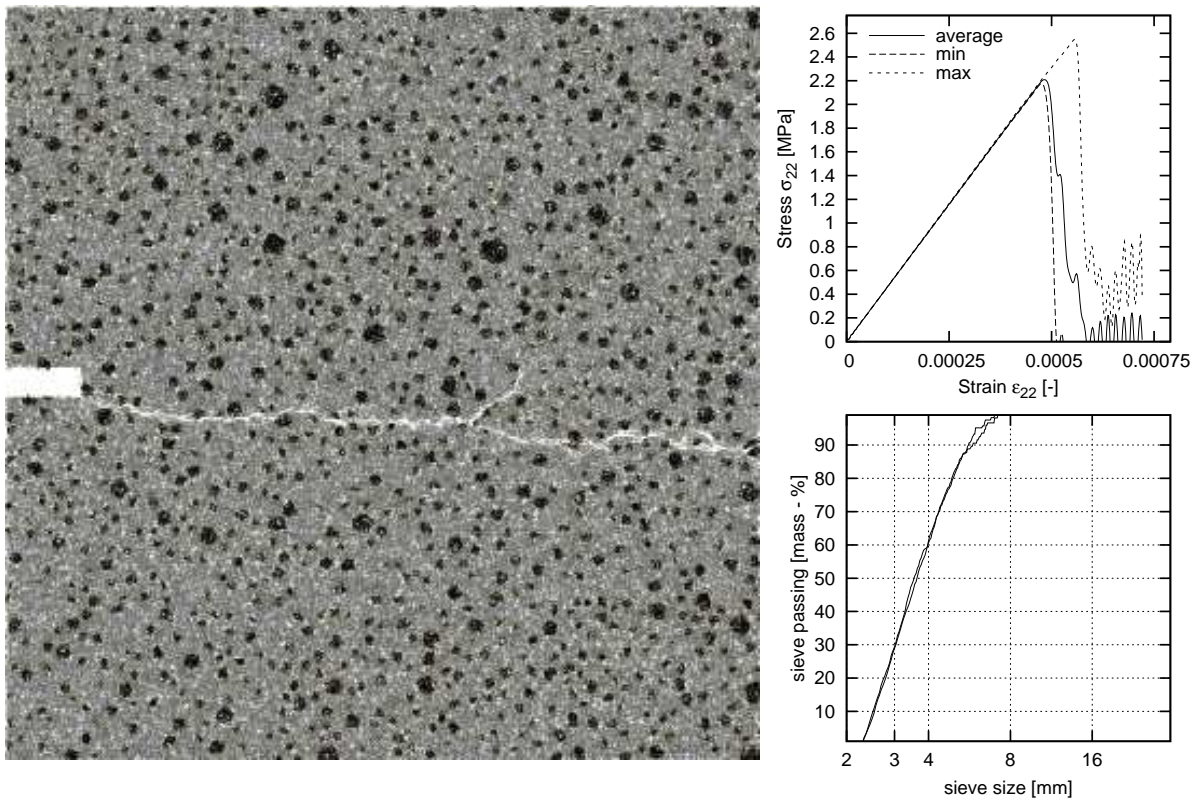


Figure 5.16: 2D specimen subject to uniaxial extension (two-phase material), material parameters as in Tab. 5.4

The particle distribution curve for aggregate is shown in Fig. 5.16 with the mean aggregate diameter of $d_{50} = 3.5$ mm. The aggregate volume density was taken as 50%. This 2D density is smaller than the particle density in real 3D concrete specimen usually equal to 70 – 75% [154]. The aggregate distribution was generated following the method given in [60] (Sect. 4.5.2).

The results of the uniaxial tensile test in Fig. 5.16 show evidently that the presence of only aggregates (without interfacial zone) does not significantly affect the load–displacement curve which remains still too brittle as compared to experiments with concrete specimens [156, 157], and is very similar to results with one–phase material on Fig. 5.15.

5.3 Uniaxial tension (three–phase material)

Figures 5.17–5.20 present the results with a square concrete specimen considered as a three–phase material. The 2D calculations were carried out mainly with a specimen size of 200×200 mm² ($b \times h$) composed of 180000 beam elements distributed non–uniformly ($\alpha = 20^\circ$, $s = 0.6g$, $g = 1$ mm, $r_{\max} = 2g$). The minimum beam length was about 0.3 mm and the maximum one was about 2 mm. The assumed material parameters are given in Table 5.5. The interface had the lowest strength. The aggregate density was assumed to be 25% or 50%, respectively. The mean aggregate diameter d_{50} was taken as 3.5 mm for the aggregate size of the range 2 – 8 mm and 12 mm for the aggregate size of the range 2–16 mm. Five simulations were performed for each case. The interfacial zones were added by assigning different properties to the beams which previously directly connected the aggregate with cement matrix (Fig. 2.11 on page 11).

The strength and pre–peak non–linearity decrease with increasing aggregate density and decreasing mean aggregate diameter. In turn, the material ductility increases when the density increases. The vertical strain corresponding to the peak increases with decreasing particle density. At the low particle content, de–bonding occurs extensively near the isolated aggregates (most of fractured elements are in bonds). This micro–cracking is responsible for the non–linear behavior in the pre–peak part of the stress–strain diagram. Next, after the peak, the fracture process progressively spreads through the entire specimen in the form of a macro–crack linking de–bonded aggregates in lines. With increasing number of aggregates, the fraction of bond elements increases and de–bonding prevails. At the high particle density, percolation of bond zones occurs, and the condition for macro–crack nucleation and growth occurs early in the loading history. The material becomes significantly weaker (since the interface strength is the weakest component of the system) and the pre–peak non–linearity does not appear. Since the amount of aggregates is large, the cracks cannot propagate in long lines. Instead of this, several discontinuous macro–cracks propagate in a tortuous manner.

Table 5.5: Paramaters used in calculation with three–phase material

Phase	Young's modulus	$p = k_b/k_l$	k_l	local ϵ_{\min}
Cement matrix	$E_m = 20$ GPa	$p_m = 0.6$	0.1	$\epsilon_m = 0.002$
Aggregate	$E_a = 60$ GPa	$p_a = 0.6$	0.3	$\epsilon_a = 0.00133$
Interface bond	$E_b = 14$ GPa	$p_b = 0.6$	0.07	$\epsilon_b = 0.0005$

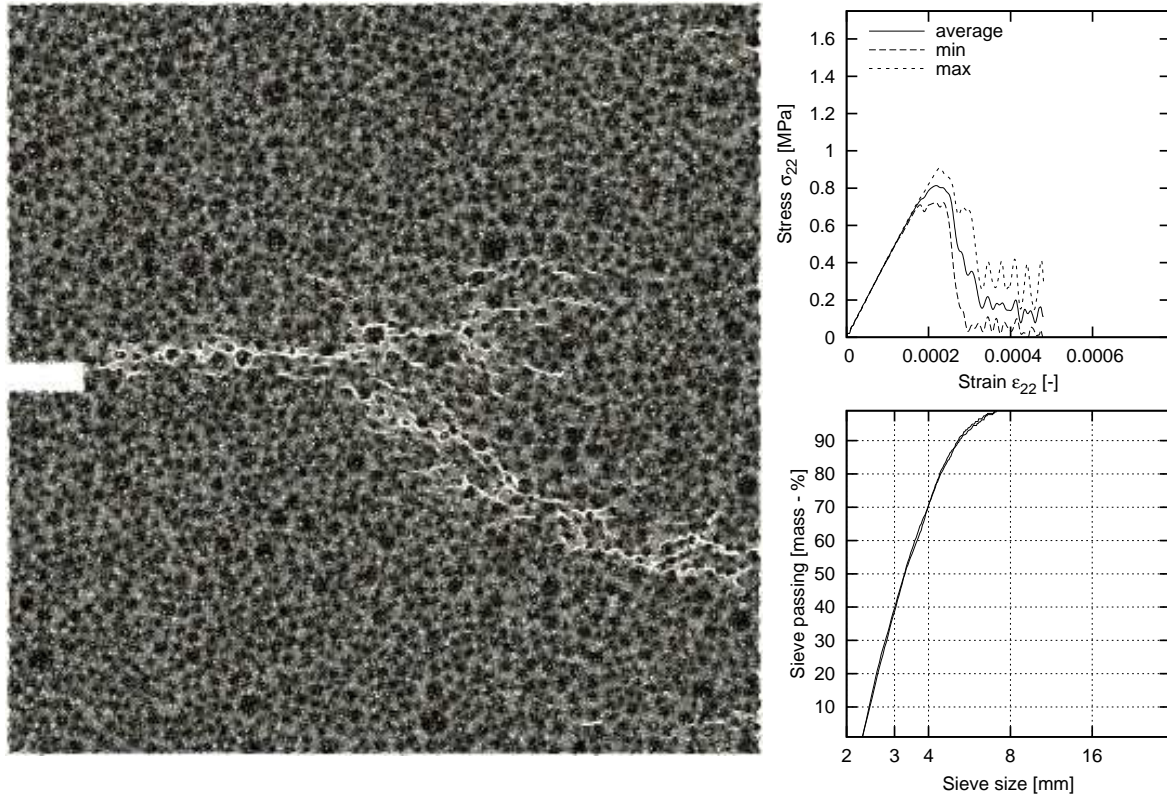


Figure 5.17: 2D concrete specimen subject to uniaxial extension (three-phase material), aggregate area percentage 50%, $d_{50} = 3.5$ mm, material parameters as in Table 5.5 (σ_{22} – vertical normal stress, ϵ_{22} – vertical normal strain)

The cracks overlap and form branches. As a result, the material ductility grows after the peak. All curves with consideration of interfacial transition zones resemble qualitatively the experimental curves for real concrete [156]. The uniaxial tensile strength changes between 0.7 – 1.7 MPa. The scatter of the material strength increases with decreasing particle density due to the larger possibility at the choice of the propagation way.

The vertical strain ϵ_{22} corresponding to the peak varies between 0.0002–0.0006, and equals to 0.0002 for grading curve assumed from experimental results presented in Chapter 3, for gravel concrete (Figures 5.17 and 3.14), where the diameters below 2 mm are not discretized into the mesh.

5.4 Size effect (uniaxial tension, three-phase material)

Figures 5.21–5.23 show a size effect during uniaxial tension (with the grading curve of Fig. 5.17). Several numerical simulations were carried out with two different rectangular concrete specimens: 10×10 cm² and 20×20 cm² using the same beam distribution. The results show that the material strength and ductility increase with decreasing specimen size (as in the experiments, Fig. 5.22) while the crack pattern remains similar (Fig. 5.23). In turn, the fracture energy decreases.

The obtained outcomes from numerical experiments for uniaxial tension are qualitatively in agreement with numerical solutions given in [99, 155, 156].

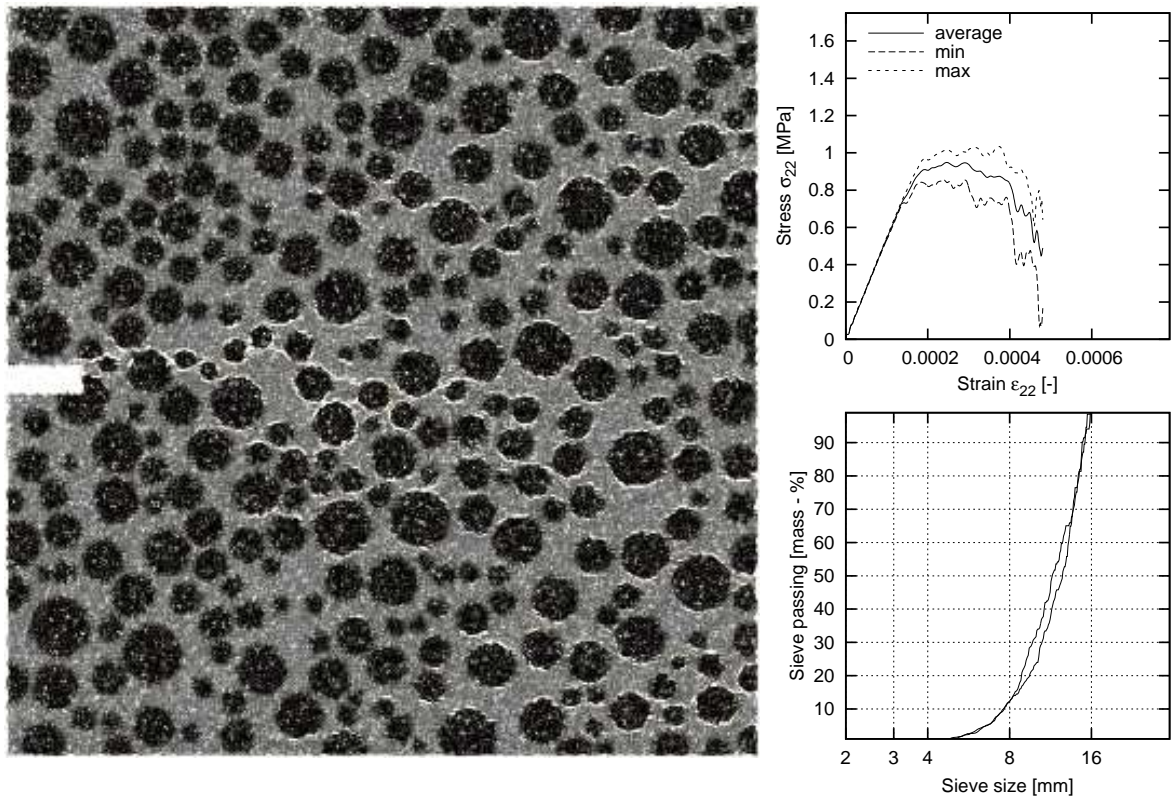


Figure 5.18: 2D concrete specimen subject to uniaxial extension (three-phase material) aggregate area percentage 50%, $d_{50} = 12$ mm, material parameters as in Table 5.5

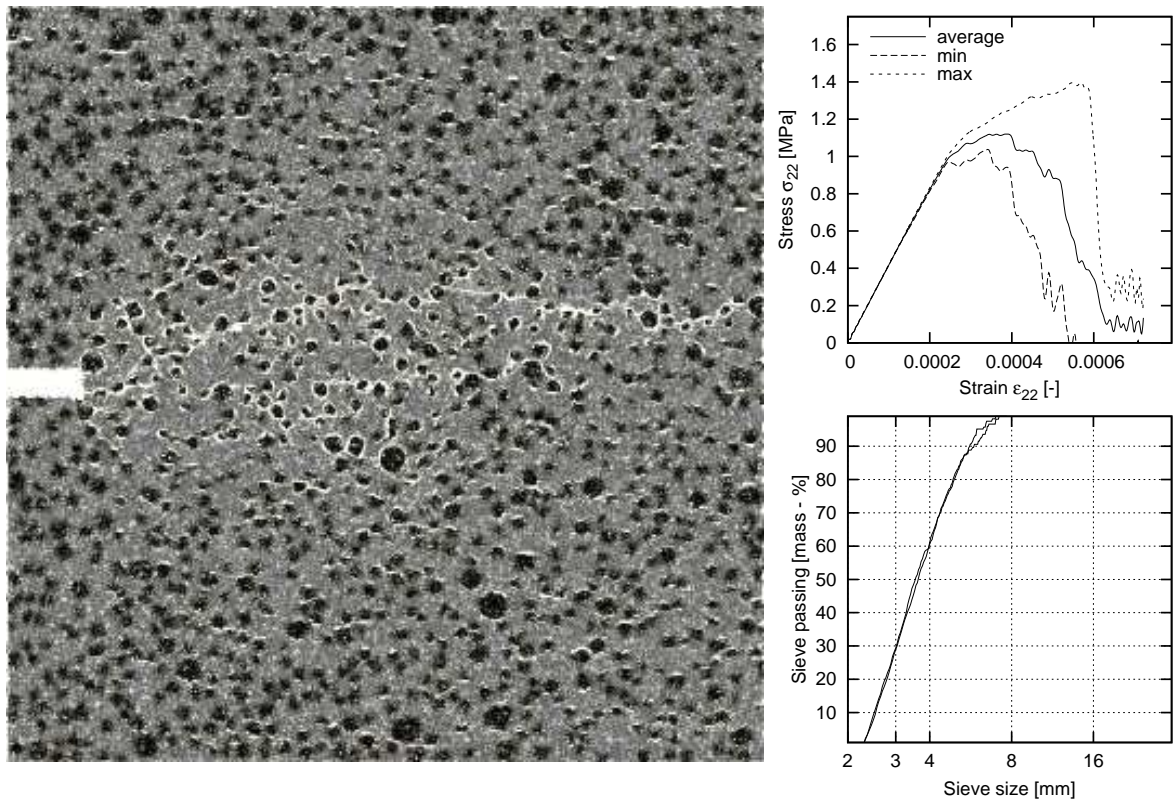


Figure 5.19: 2D concrete specimen subject to uniaxial extension (three-phase material) aggregate area percentage 25%, $d_{50} = 3.5$ mm, material parameters as in Table 5.5

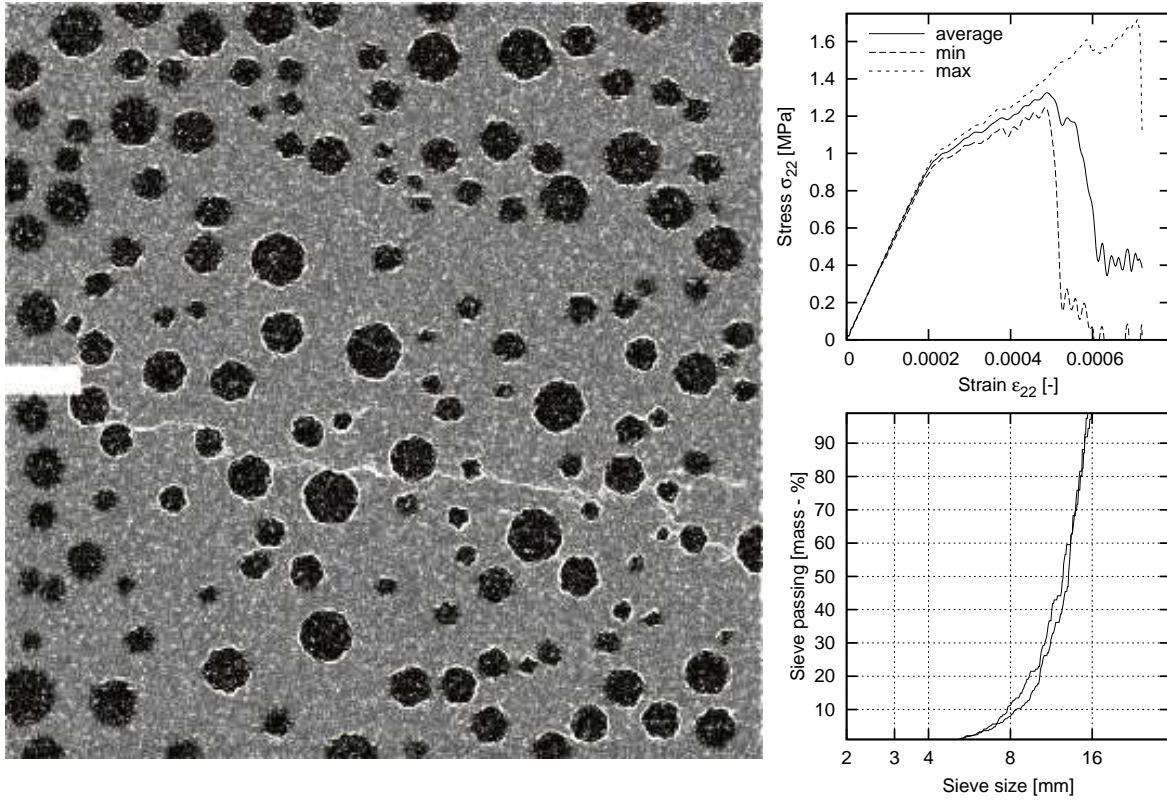


Figure 5.20: 2D concrete specimen subject to uniaxial extension (three-phase material) aggregate area percentage 25%, $d_{50} = 12$ mm, material parameters as in Table 5.5

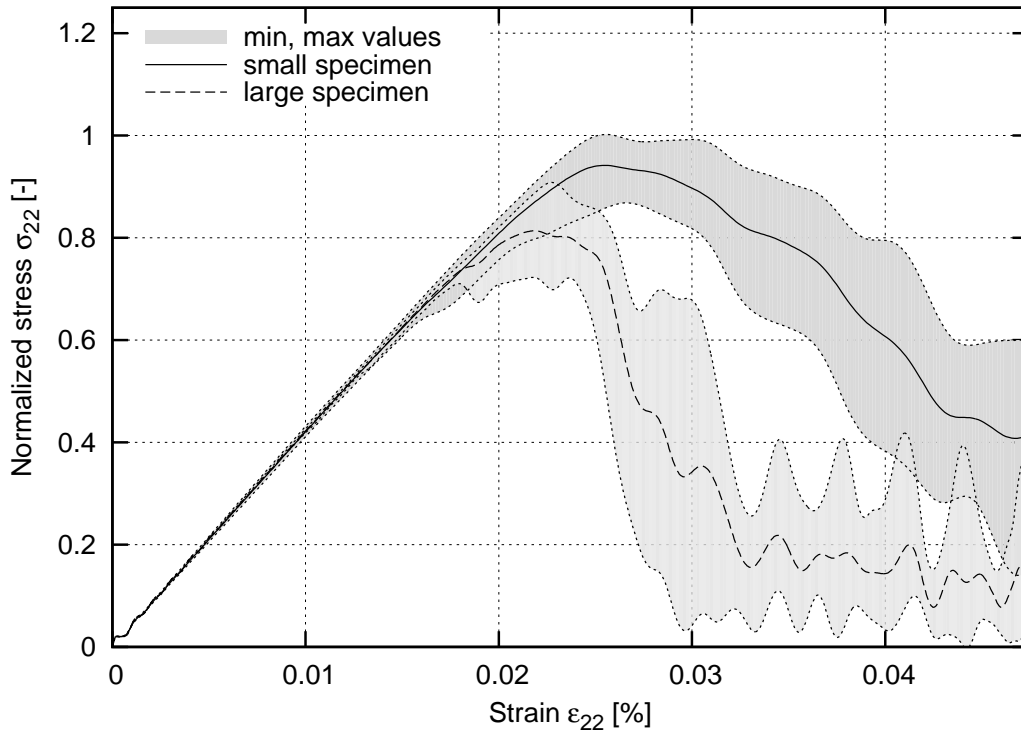


Figure 5.21: The size effect for 2D concrete specimens subject to uniaxial extension with sizes 10×10 cm² and 20×20 cm², aggregate area percentage 50%, $d_{50} = 3.5$ mm, material parameters as in Table 5.5, grading curve as in Fig. 5.17

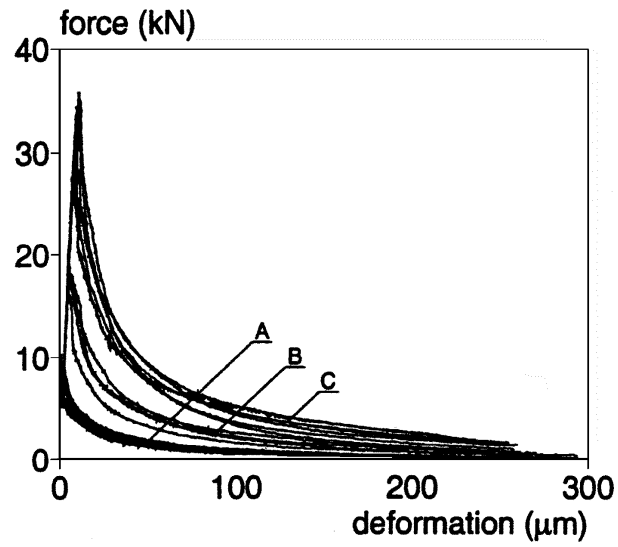


Figure 5.22: Experimental force-deformation diagram for 3 different dog-bone shaped specimens $h \times b$: (A) $75 \times 50 \text{ mm}^2$, (B) $150 \times 75 \text{ mm}^2$, (C) $300 \times 200 \text{ mm}^2$ [156] (h - height, b - width)

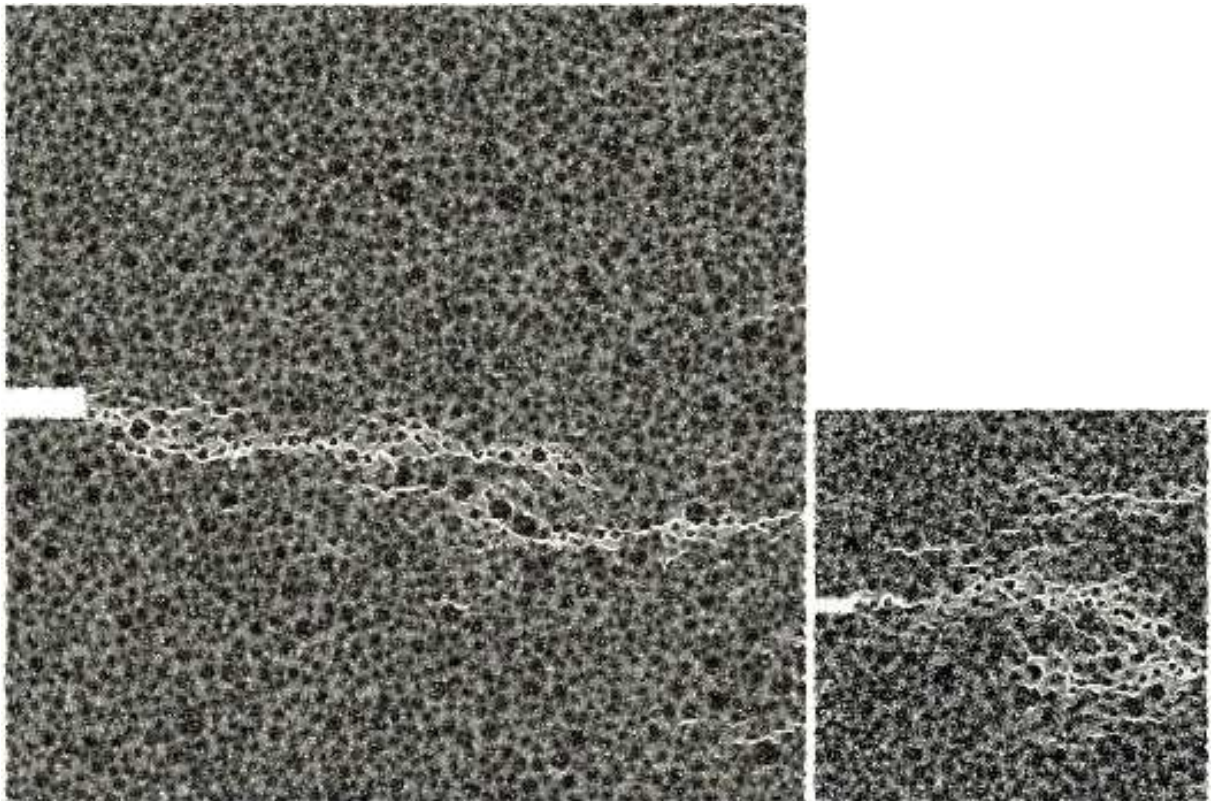


Figure 5.23: Fracture in 2 concrete specimens of different size ($20 \times 20 \text{ cm}^2$ and $10 \times 10 \text{ cm}^2$)

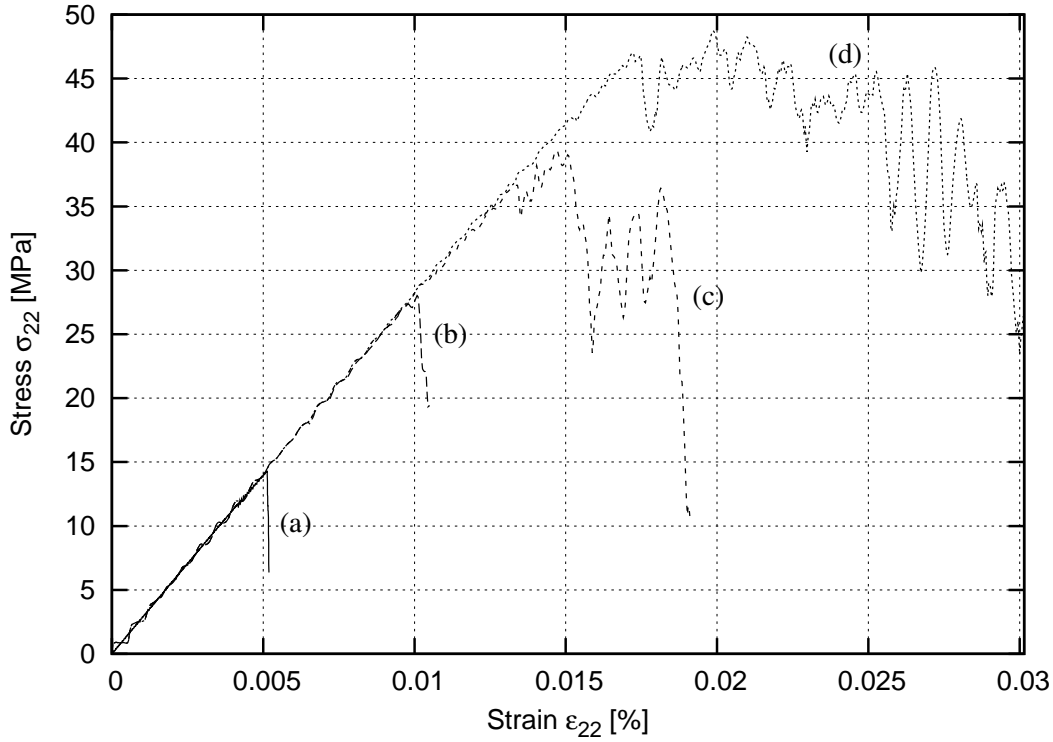


Figure 5.24: Effect of the characteristic length on the stress–strain curve ($p=0.6$, specimen $100 \times 100 \text{ mm}^2$): (a) $l_c=0$ (local approach), (b) $l_c = g$, (c) $l_c = 2g$, (d) $l_c = 3g$ (non–local approach, σ_{22} – vertical normal stress, ϵ_{22} – vertical normal strain)

5.5 Uniaxial tension with non–local approach

Section 4.6 describes the non–local approach to decrease the material brittleness in one–phase material. Figures 5.24 and 5.25 demonstrate the results for a non–local approach during uniaxial tension. The simulations were carried out with a specimen size $10 \times 10 \text{ cm}^2$. In the calculations, the different values of l_c ($l_c = 0, 1g, 2g$ and $3g$) were used. The results show that, the strength, normal strain corresponding to the peak and material ductility increase with increasing l_c .

A non–local approach contributes to a significant increase of the computation time. An increase of ductility can be also achieved using a three–phase material.

5.6 Uniaxial compression (three–phase material)

Figure 5.28 demonstrates the crack pattern in uniaxial compression with smooth edges using parameters listed in Tab. 5.6 using the three–phase material. The stress–strain curve is shown in Fig. 5.26 and the evolution of Poisson’s ratio during compression is shown in Fig. 5.27.

The material remains in the elastic regime for about one third of the total strain (marked with ‘A’ on Fig. 5.26), and afterwards the inclination of the stress–strain curve becomes smaller. Corresponding Poisson’s ratio prior to the point ‘A’ remains below 0.4, (starting from 0.2). After the material exceeds the elastic regime the Poisson’s ratio dramatically increases, up to value of 1.

The results correspond well with results by Caballero [29] and with laboratory experiments [157].

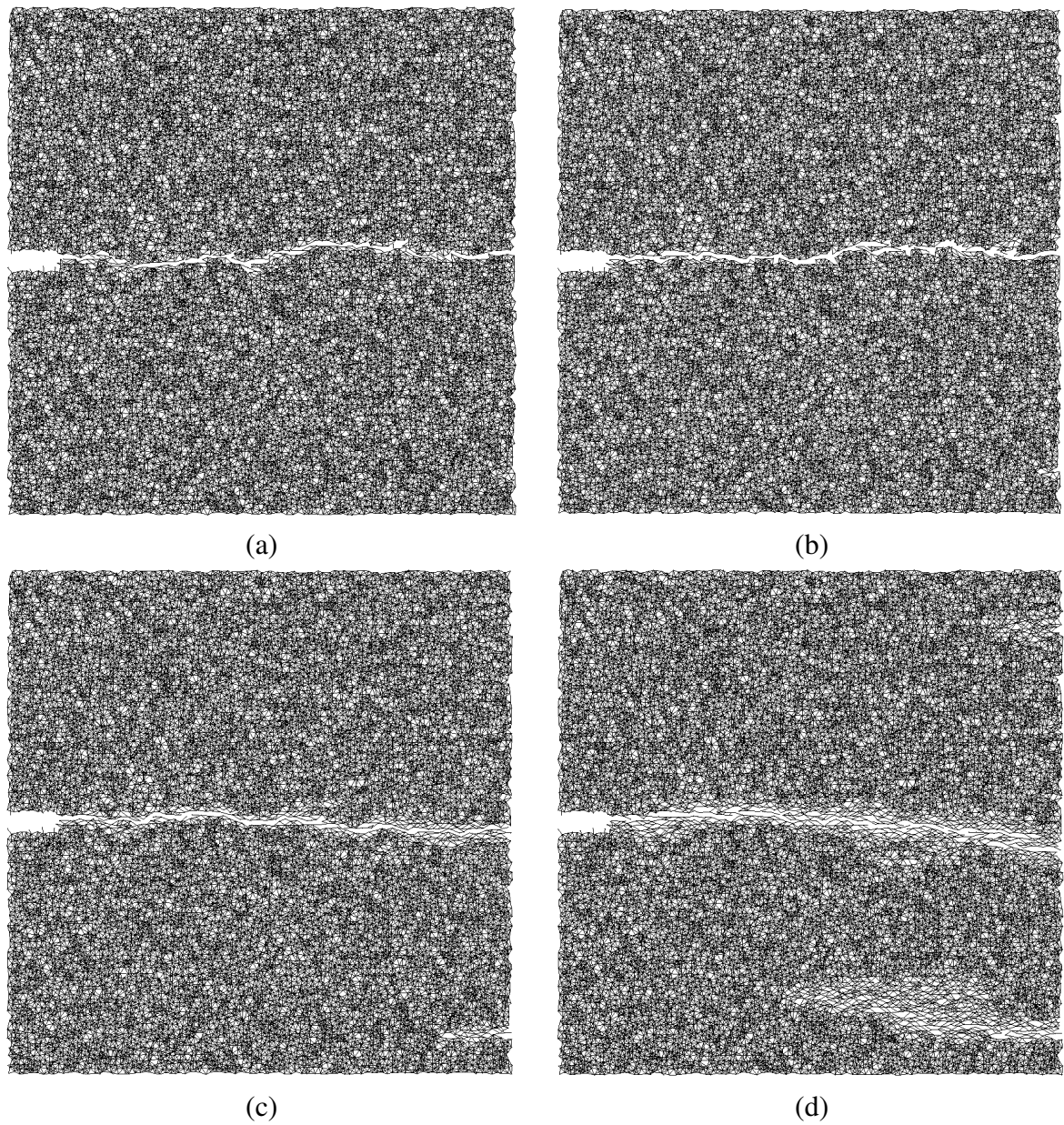


Figure 5.25: Effect of the characteristic length l_c on the fracture process (beams were removed when local $\epsilon_{\min} = 0.0002$, $p = 0.6$): (a) $l_c = 0$ (local approach), (b) $l_c = g$, (c) $l_c = 2g$, (d) $l_c = 3g$ (non-local approach)

Table 5.6: The calibrated material parameters (three-phase material)

Phase	Stiffness			Fracture local ϵ_{\min}	Young's modulus
	k_l	$p_{tension}$	p_{compr}		
Cement matrix	0.1	0.6	0.2	0.002	20 GPa
Aggregate	0.3	0.6	0.2	0.00133	60 GPa
Interface bond	0.07	0.6	0.2	0.0005	14 GPa

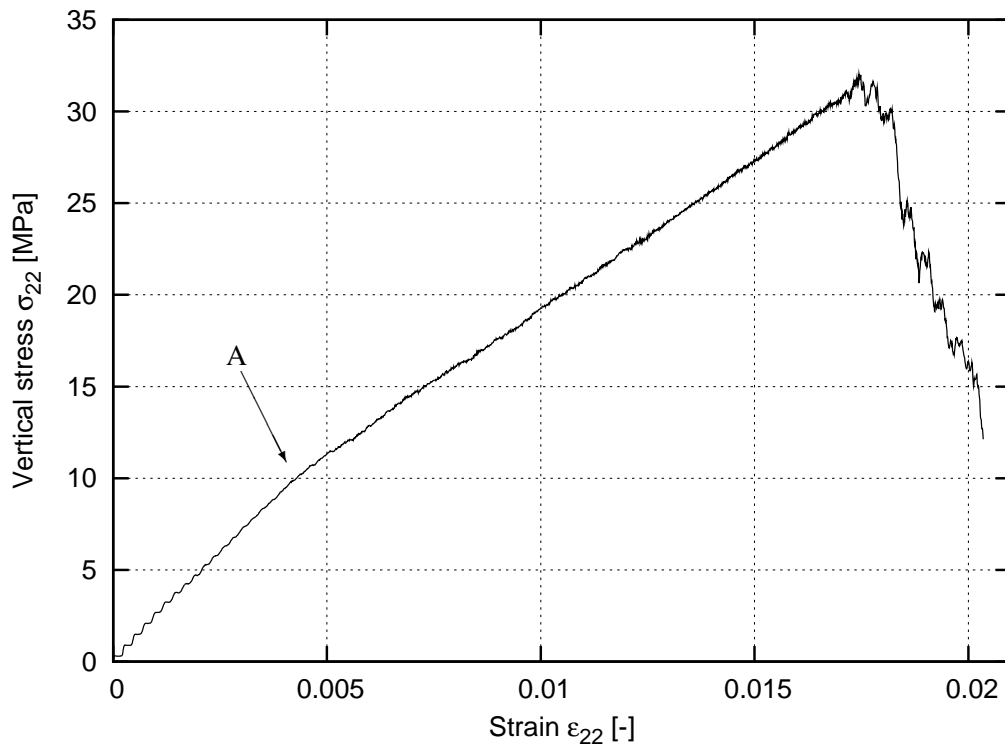


Figure 5.26: Stress–strain curve in uniaxial compression with smooth edges (three–phase material)

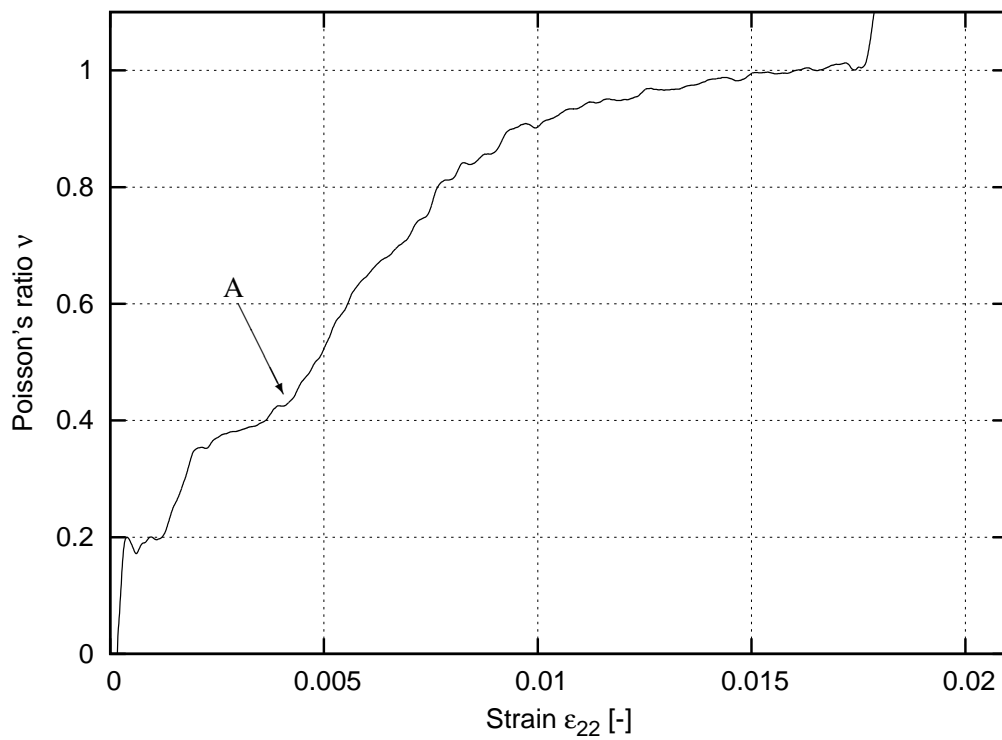


Figure 5.27: Evolution of Poisson's ratio in uniaxial compression with smooth edges (three–phase material)

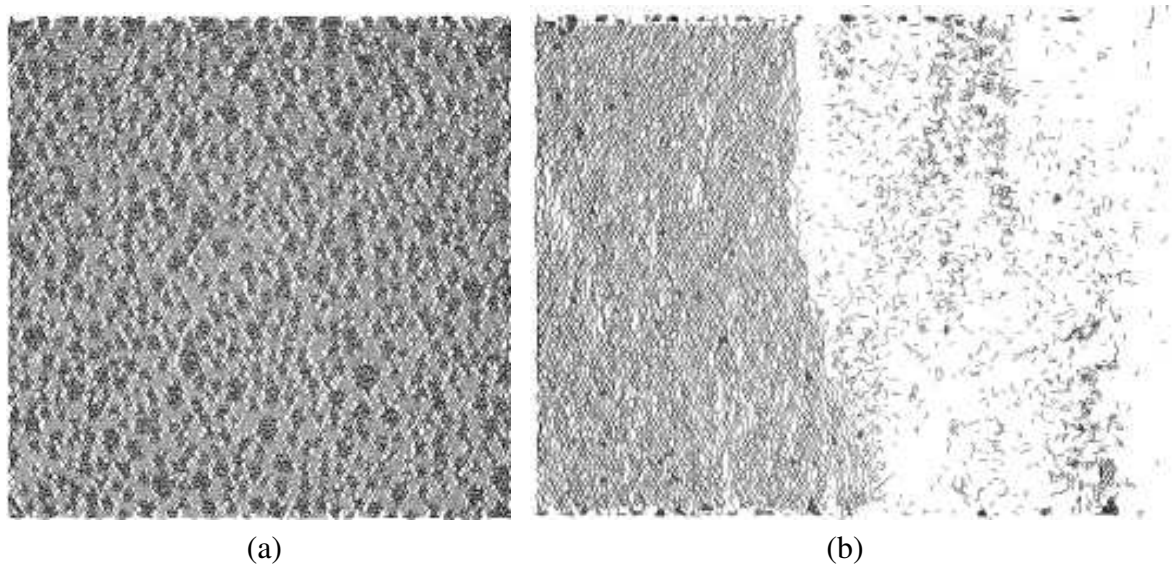


Figure 5.28: Fracture pattern in $10 \times 10 \text{ cm}^2$ specimen subject to uniaxial compression with smooth edges (using Delaunay mesh generation method Tab. 5.3) in three-phase material (gray denotes undamaged area, black color denotes aggregates, white color represents fracture damage): (a) in the course of loading; (b) after failure

5.7 Three-point bending (comparison with experiments)

The experiments performed in Chapter 3 were carried out for small and medium specimens. The model parameters were assumed from Tab. 5.6, with the grading curve from Figs. 5.17 and 3.14 (gravel). The resulting load displacement curve is shown in Fig. 5.29. The plot was smoothed to remove the oscillations induced by the elastic behavior (Sect. 4.9.1, Fig. 4.26), to increase readability. The oscillations indicate that the calculation step increment assumed was too big. The load–displacement curve is in good agreement with the experimental results of Fig. 3.17. The size effect was obtained (the peak force for the medium specimen is 4 kN, for the small specimen 2.5 kN, Fig. 5.29 a,b) and the larger specimen is more brittle than the medium specimen. For comparison, the simulation was performed (Fig. 5.29 c) with assumed the same value of $p = 0.6$ for tension and compression, it turns out that material is more brittle then (Fig. 5.29 a,c).

Figures 5.30 and 5.32 show the resulting crack pattern in the specimens. Figure 5.31 shows strains on the surface of concrete. The strains obtained are in well agreement with Digital Image Correlation results (Fig. 3.24, gravel concrete).

5.8 Uniaxial extension in 3D (one-phase material)

3D simulations were performed with the specimen size of $10 \times 10 \times 10 \text{ cm}^3$ with the cell size $g = 3.3 \text{ mm}$ (corresponds to average beam length), and with the smaller specimen $5 \times 5 \times 5 \text{ cm}^3$. A three-phase material was used (Tab. 5.6) with grading curve from Fig. 5.18 ($d_{50} = 12 \text{ mm}$, aggregate volume percentage 45%).

Figures 5.33 and 5.34 show the the specimens after failure, Fig. 5.35 shows normalized stress–strain curve. A pronounced size effect was obtained, and the smaller specimen has a larger strength and ductility after the peak. As compared with 2D simulations, the material becomes more ductile. It is in accordance with calculations by [99].

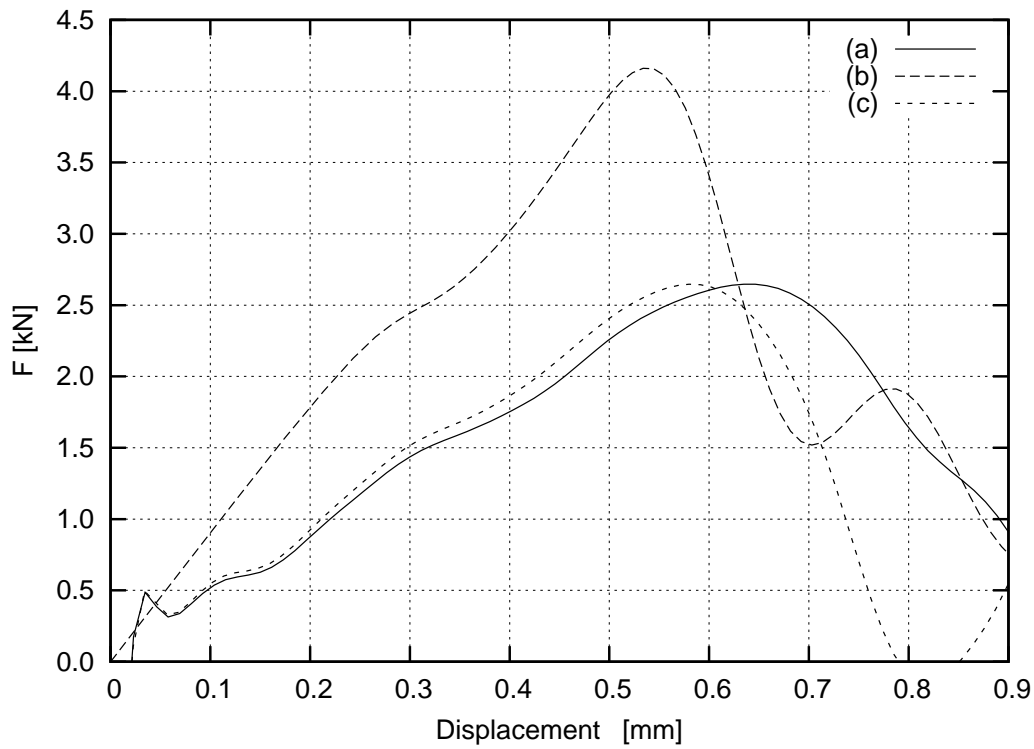


Figure 5.29: Load–displacement curves in three–point bending test; (a) small specimen ($p_{tension} = 0.6$, $p_{compr} = 0.2$); (b) medium specimen ($p_{tension} = 0.6$, $p_{compr} = 0.2$); (c) small specimen (the same value of $p = 0.6$)

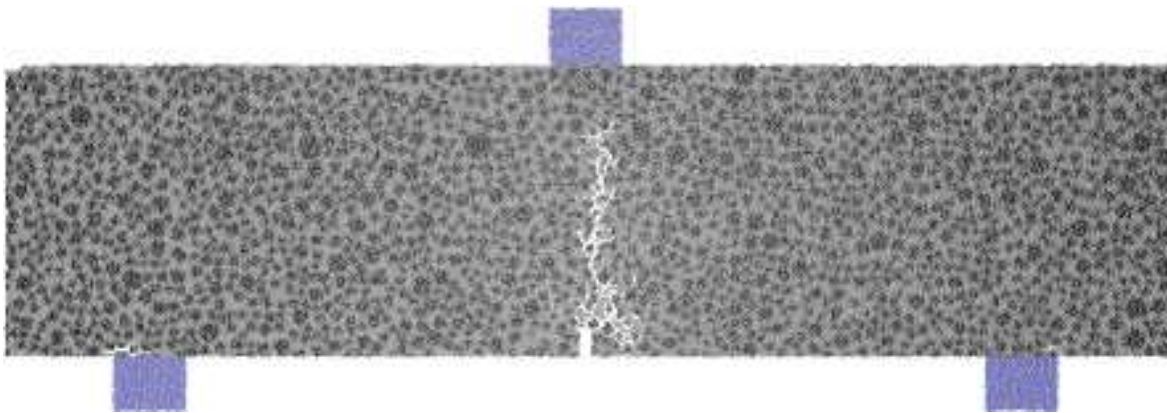


Figure 5.30: Crack pattern in three–point bending, small specimen

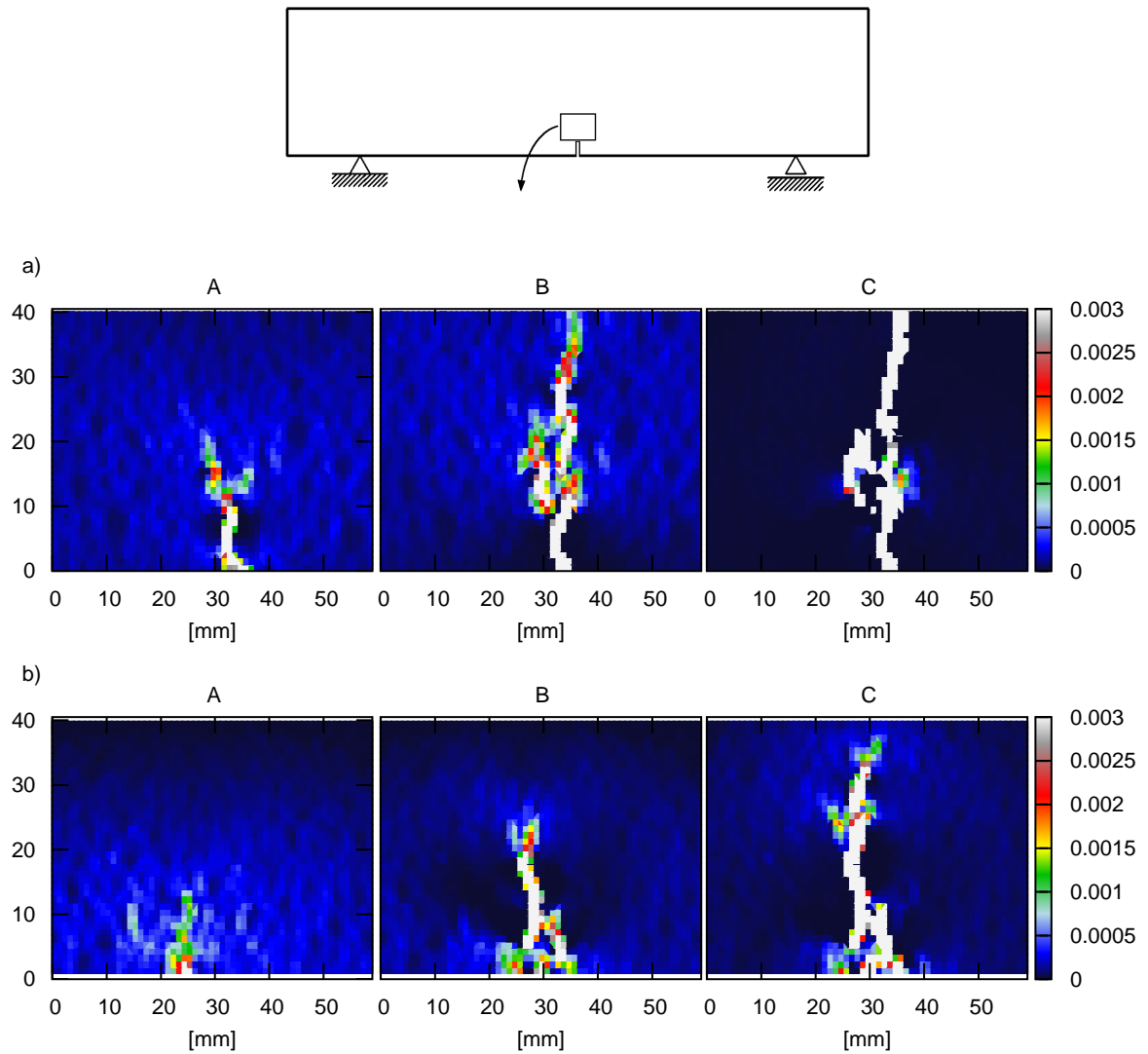


Figure 5.31: Evolution of numerically calculated strain field during three-point bending simulation: (a) small size beam; (b) medium size beam; (A) before the peak; (B) at peak; (C) after the peak

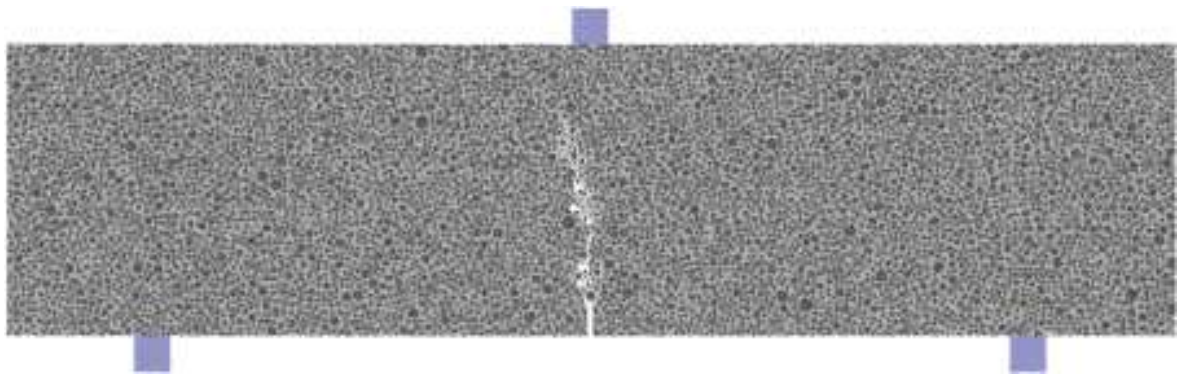


Figure 5.32: Crack pattern in three-point bending, medium specimen

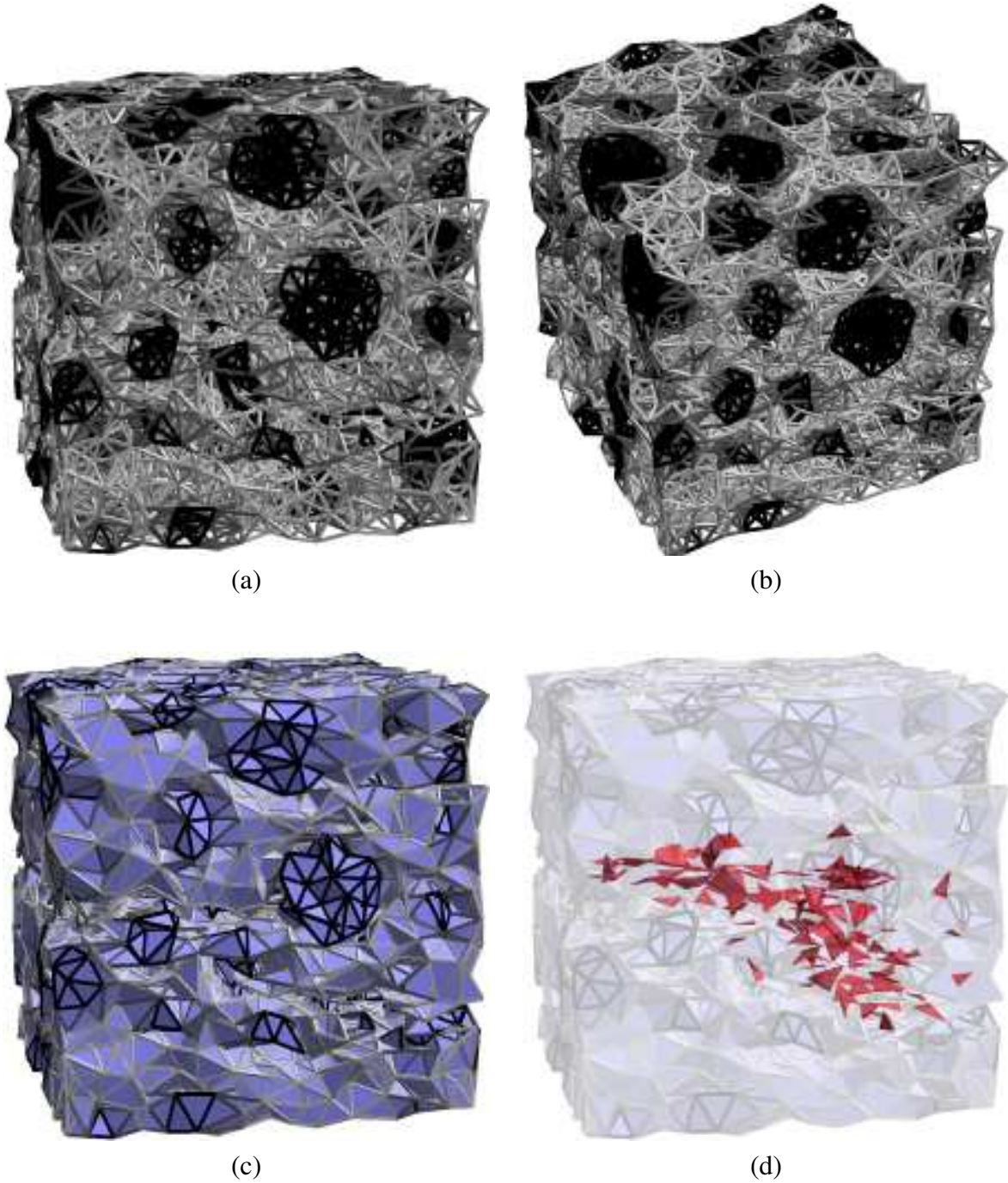


Figure 5.33: A three-dimensional $5 \times 5 \times 5 \text{ cm}^3$ specimen subject to uniaxial extension (parameters as in Tab. 5.6, $g = 3.3 \text{ mm}$, grading curve as in Fig. 5.18, aggregate volume percentage 45%): (a,b) aggregate structure (black – aggregates, dark gray – interface, light gray – cement matrix); (b) surface after failure; (c) fracture surface inside

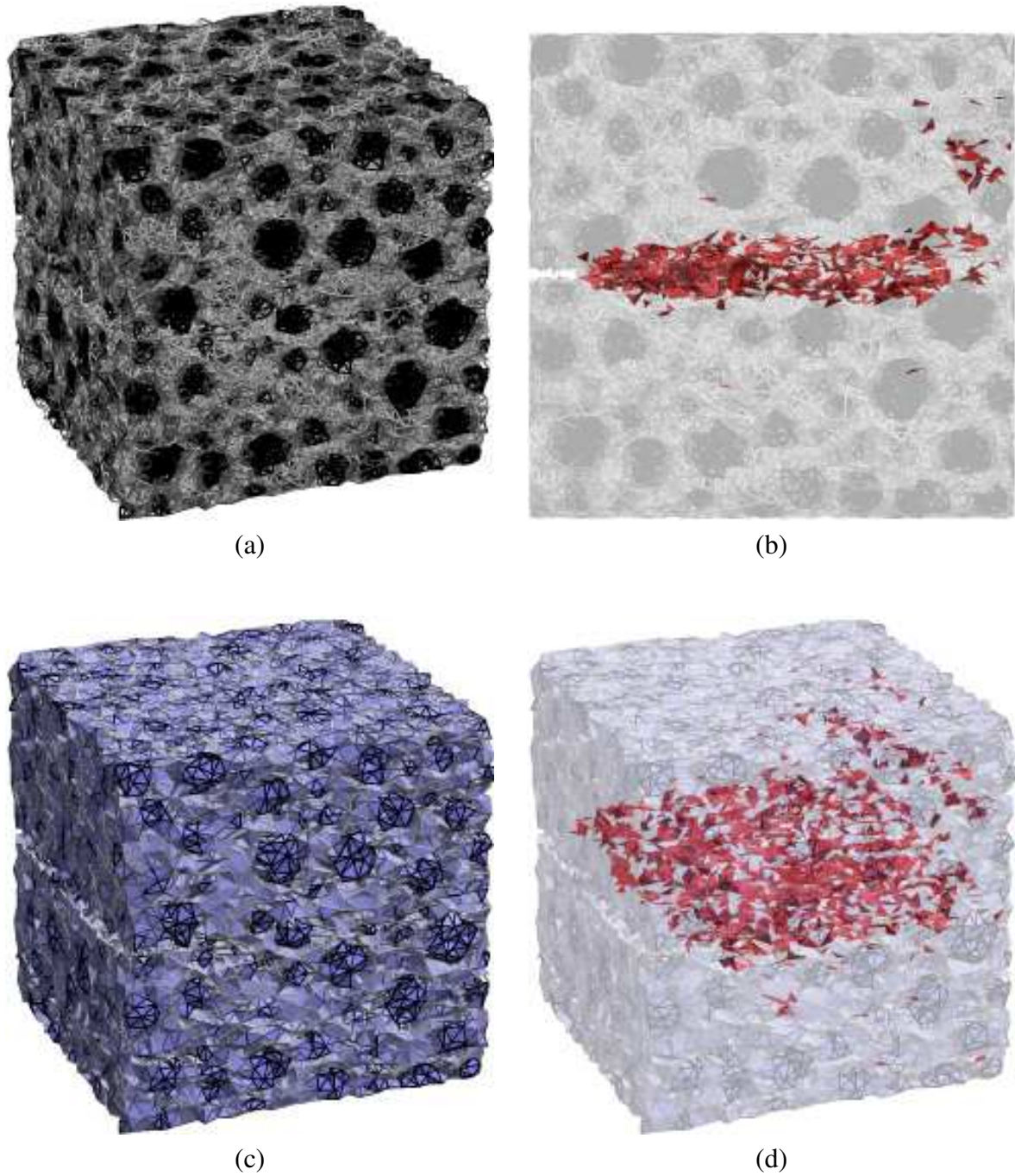


Figure 5.34: A three-dimensional $10 \times 10 \times 10 \text{ cm}^3$ specimen subject to uniaxial extension (parameters as in Tab. 5.6, $g = 3.3 \text{ mm}$, grading curve as in Fig. 5.18, aggregate volume percentage 45%): (a) aggregate structure (black – aggregates, dark gray – interface, light gray – cement matrix); (b) aggregate structure and fracture surface (red); (c) surface after failure; (d) fracture surface inside

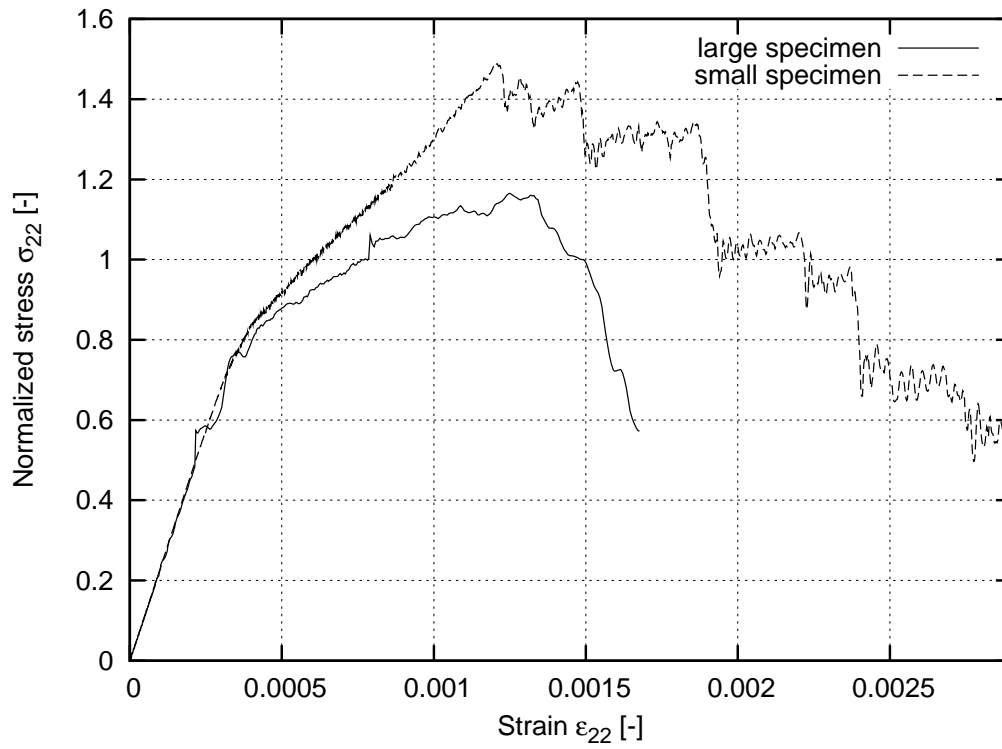


Figure 5.35: The size effect for 3D concrete specimens subject to uniaxial extension with sizes $10 \times 10 \times 10 \text{ cm}^3$ and $5 \times 5 \times 5 \text{ cm}^3$, aggregate area percentage 45%, $d_{50} = 12 \text{ mm}$, material parameters as in Table 5.6, grading curve as in Fig. 5.18

Chapter 6

Conclusions and future work

The lattice model is a simple approach to the fracture behavior in heterogeneous quasi-brittle materials. It is very useful in studying and understanding the phenomenon of the crack formation and crack propagation during uniaxial tension since it can reproduce fracture processes observed in real laboratory experiments. Owing to this, novel (stronger and better) engineering materials can be developed. By using an elastic–purely brittle local fracture law at the particle level of the material, global softening behavior is obtained. The heterogeneous 3D–lattice model for concrete presented in this thesis requires 5 material parameters (k_l , k_b , k_t , E , ε_{\min}) for each phase and 2 (or 4) grid parameters (g , s for Delaunay mesh generation method and additionally α and r_{\max} for non–Delaunay mesh generation method) related to the distribution, quantity and length of beams. The obtained results of crack patterns and stress–strain curves for a three–phase concrete material during uniaxial tension are qualitatively in agreement with experimental ones for concrete, and compare quite well with results published in [99, 155, 156]. As well as results in compression, three–point bending, four–point bending and combined shear–extension test [111].

The material composition has a significant effect on the material behavior, in particular the particle density and distribution of weak bond zones. The strength and pre–peak non–linearity decrease with increasing aggregate density and decreasing mean aggregate diameter during uniaxial tension. The material ductility increases when the aggregate density increases. The vertical strain corresponding to the peak increases with decreasing particle density. At the low particle content debonding occurs extensively near the isolated aggregates. At the high particle density, percolation of bond zones occurs, and the condition for macro–crack nucleation and growth occurs. The pre–peak non–linearity cannot be ignored at low particle density. The macro–crack process occurs before the maximum load. The 3D simulations increase the material ductility as well.

The simulations of a size effect show a decrease of nominal strength with increasing specimen size as well as an increase of fracture energy with size. The brittleness in one–phase material can be decreased by using a non–local approach when calculating strain. If no bond phase is included, the material strength and material ductility do not depend on the particle density.

The Poisson’s ratio for compression ($\nu \approx 0.22$) differs from the Poisson’s ratio in tension ($\nu \approx 0.1$).

The Digital Image Correlation is very effective and simple optical technique to determine the strain field on the surface of concrete with a large accuracy and without any physical contact. During the experiment, the digital camera must be fixed. Larger pixel subset

sizes increase accuracy and reduce noise, but also at the same time they blur the boundaries between regions with different strain.

A strong size effect occurs in experiments with notched concrete beams subject to bending. The beam ductility and nominal strength increase with decreasing specimen size.

The fracture process zone is created before the peak on the load–deflection diagram is met. It can occur as a single zone or a pattern of zones at the beginning of deformation. The width of the fracture process zone increases during deformation and is about 6 – 7 mm before the failure. The zone can be almost straight or strongly curved.

The calculations with a lattice model will be continued. The code will be parallelized. A multiscale model will be used linking the lattice model with the continuum elasto-plastic model with non-local softening [20, 23] wherein the first model will be only restricted to the damaged part of the structure.

Appendix A

Cellular automaton

Two cellular automaton models were developed by the author of this thesis to calculate the kinematics of non-cohesive granular materials.

The first model is a simplified cellular automaton which was used to calculate the kinematics of non-cohesive granular materials during confined flow in silos. In this model, granular flow was assumed to be an upward propagation of holes through a lattice composed of cells representing single particles. Calculations were carried out with different silo shapes and inserts, transition probabilities, migration rules, outflow schemes, grid types, wall roughness and cell numbers. To visualize the calculation process, horizontal layers of various shades were introduced. The simulation results were compared with laboratory tests in model silos.

The second model is an improved cellular automaton. This model is based on a gas model of hydrodynamics, where collisions and dissipation of particles were taken into account during granular flow. The model allows for investigations of dilatant zones in granular material during silo flow.

The advantages and disadvantages of both cellular automata were outlined.

A.1 Introduction

A cellular automaton is a powerful method to describe, understand and simulate the behavior of complex physical systems, which are difficult to describe using the more traditional approaches (by means of differential equations). It is viewed as an alternative form of the microscopic reality, which exhibits the expected macroscopic behavior [34, 165]. The concept dates back to the 1950's [100, 152]. During the next fifty years of applications, cellular automata have been considerably improved (e.g. lattice-gas automaton, lattice Boltzmann method [34, 165]) and used in many fields of physics, chemistry and biology dealing with the fluid flow of fluid and gas [34, 65]. They have also been used to simulate the behavior of granular materials during vibration, piling, toppling, segregation, displacement of retaining walls and rapid flow [4, 5, 31, 34, 51, 52, 64, 67, 70, 85, 88, 89, 91–94, 103, 110, 113, 119, 130–132]. In models, collisions, friction, rotations and particle size have been introduced. Cellular automata have been used to simulate rapid flow in silos [4, 5, 31, 51, 52, 88, 89, 91–94, 103, 113, 131, 132] as above the outlet silo fills behave more like fluids rather than solids (although the internal friction of the bulk material is still of major importance [150]).

The cellular automata have both advantages and disadvantages relative to other model-

ing techniques. The advantages include large numbers of particles, lack of restrictions for deformations, simplicity of implementation and small amounts of computer time needed to describe the flow. The behavior is described not in terms of differential equations but in discrete systems. The models provide insight into the main physical features of flow on microscopic and macroscopic level. The main disadvantage is that the models are purely kinematic and no flow dynamics is involved.

The objective of simulations presented in this paper was to determine flow patterns of non-cohesive granular bodies in model silos, with or without inserts, using two different cellular automaton models. A realistic prognosis of flow patterns of bulk solids is very important when designing silos [69, 120, 129, 139, 146] since loads in silos are directly related to the flow type. The flow of bulk material stored in silos is of two main patterns: funnel flow (core flow) and mass flow. In mass flow, the entire material is in motion during discharge. In funnel flow, movement occurs only in a channel within the stored material, and this channel is surrounded by non-flowing material.

Two different cellular models were applied using a two-dimensional array of hexagonal cells, a two-dimensional array of orthogonal cells and a three-dimensional array of cubic cells.

The first simplified model (Sect. A.2) presented in [88, 89, 131, 132], granular flow was assumed as an upward propagation of holes through a lattice composed of cells representing single particles (grains move downwards and voids move upwards).

In the second improved model (Sect. A.4) based on a gas model of hydrodynamics [2, 52, 119], inelastic collisions were taken into account. This model allows investigations of density zones in granular material during silo flow.

Cellular automata simulations have been carried out with a mass flow and funnel flow model silo, different silo shapes, migration rules, transport schemes, transition probabilities, outflow velocities, wall roughness and cell numbers. Comparative calculations have also been carried out with a three-dimensional grid of square cells. Four different types of inserts (often used in the silo industry) have mainly been taken into account: a wedge-shaped cone, an internal hopper, two inclined discharging tubes over the outlet, and a perforated vertical emptying tube. The theoretical flow patterns were compared with experimental ones.

Additionally, a sandpile and a flow in vertical channel were modelled.

A.2 Description of the simplified model

A simplified cellular automaton requires a lattice of adjacent cells covering a portion of a N -dimensional space, a set of variables attached to each site of the lattice and giving the local state of each cell at time $t = 0, 1, 2, \dots$ and a rule which specifies the variable evolution in time [34]. Thus, a cellular automaton is a system composed of adjacent cells (usually organized as a regular hexagonal, triangular or rectangular lattice), which evolves in a discrete time step. Each cell is characterized by an internal state whose value belongs to a finite set. These states are updated in parallel according to local rules involving the neighborhood of each cell.

In the simplified cellular automaton discussed here, granular flow of separate grains in silos is described as upward propagation and diffusion of holes through the lattice of cells (grains move downwards and voids move upwards). Each cell may be empty or filled by a particle of the material. The computation process involves searching through all lattice

cells for holes. At each time step, a particle flowing downwards due to gravity can move from one cell to an empty neighboring cell or remain at rest. The filling of one cell causes the formation of a new empty one, which is again filled with a randomly chosen particle (Fig. A.1). Thus the migration of holes inside the granular materials occurs. The transition of holes corresponds to the transport of the material in the opposite direction. The evolution rule describing the state change can be deterministic or probabilistic. For silo flow, it is usually assumed to be probabilistic, similarly as in a real bulk solid where the shapes and dimensions of particles, contact points and forces between particles, and particle roughness are completely random. The phenomena of collision, friction and rotation have not been taken into account in the model.

Baxter et al. [4, 5] have simulated a two-dimensional deterministic process of filling and flow of granular material in a hopper using a regular triangular array of cells. The direction of grain displacement corresponded to the direction in which mechanical energy was minimal. The energy consisted of two components: gravity and interaction between neighboring grains, including friction, dilatancy and relative orientation of stick-like grains.

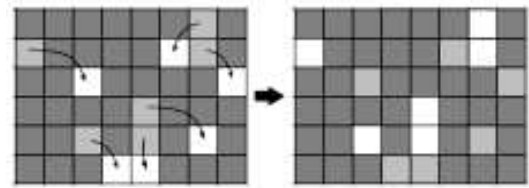


Figure A.1: Example propagation of cells in the simplified cellular automaton

Peng and Herrmann [119] investigated density waves during granular flow in a pipe with an improved cellular automaton. In their model, inelastic collisions of particles were taken into account (conserving mass and momentum), similarly as in a lattice-gas automaton [65, 68]. Particles could change their velocities during collisions and move during propagation in the direction of their velocities to neighboring sites. The four material parameters assumed were related to gravity, dissipation, density and wall roughness.

Savage [131, 132] used a simple probabilistic cellular automaton with regular rectangular cells to describe two-dimensional flow in a hopper. An empty cell could be filled by particles from 3 cells located in the horizontal layer above. The aspect ratio of cells was changed to model the internal friction angle of the silo fill. However, too large diffusion of particles was observed in the calculations. In a similar two-dimensional cellular model by Osinov [113] with a square grid of cells, an empty cell could be filled by particles from 3 cells (one above, the others at the sides). The transition probabilities and dimensions of cells were the governing parameters affecting the hopper flow process. In addition, the simulations took into account loosening of the material (flowing particles could generate their copies). In all the above mentioned simulations of silo flow [4, 5, 51, 52, 88, 89, 91–94, 103, 113, 131, 132], agreement with laboratory tests was surprisingly good in spite of the extreme simplicity of cellular automata models neglecting the dynamics of the granular flow process.

A.3 Simulation results for the simplified model

As compared with the calculations of silo flow with a similar simplified cellular automaton [113, 131, 132], other migration rules and transition probabilities were tested and different types of cell grids were compared. The calculations were also carried out with transition probabilities dependent on cell location to simulate walls of varied roughness. Moreover, various outflow velocities were simulated.

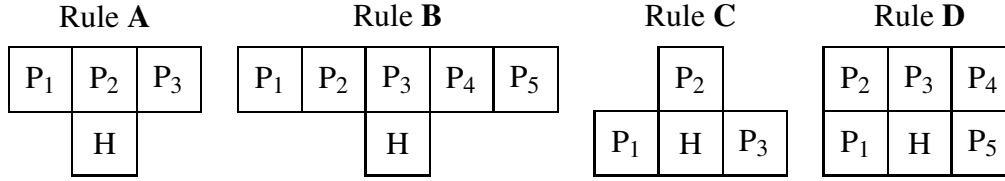


Figure A.2: Migration rules assumed for a two-dimensional array of quadratic cells (H - void)

The simulations were carried out with various:

- silo shapes,
- particle numbers,
- migration rules,
- transition probabilities,
- transport schemes,
- outlet velocities,
- wall roughness,
- lattice grids, and
- silo inserts.

Four different migration rules were investigated using a quadratic grid of cells (Fig. A.2). In the first and second case (rule **A** and **B**), each empty quadratic cell could be filled by particles from 3 or 5 cells located in the horizontal layer above. In the third case (rule **C**), each empty cell could be filled by particles from 3 cells located above and at the sides. In the fourth case (rule **D**), each empty cell was filled by particles from 5 cells located above and at the sides. The probability of hole transport from one cell to another was assumed to be p_i ($\sum p_i = 1$) and did not depend on coordinates.

To take into account the effect of the transport scheme, the so-called 'disturbed' flow was taken into account. In this case, the motion of empty cells (without particles) was also considered during flow. They propagated similarly to filled cells (containing particles). To capture material loosening (dilatancy), each particle passing through a certain number of cells (e.g. 100) was assumed to generate its copy in a hole to be filled [113]. Thus, the material's volume could grow during flow.

To simulate outflow with a constant outlet velocity, the filled cells were not emptied every calculation step but e.g. every 5 steps.

To model very rough walls, different transition probabilities were assumed in the wall region compared to the remaining region.

The calculations were performed with a square and hexagonal grid of cells. Hexagonal cells are usually used to simulate a Navier-Stokes equation of fluid flow [34]. They are preferable because of isotropy of the momentum advection tensor.

Effect of migration rules and transition probabilities

To find the most realistic migration rule for granular flow in silos, simulation results were compared with results of simple laboratory model tests carried out with a mass flow silo and a funnel flow silo (without inserts) made of perspex. The dimensions of the mass flow silo were: 0.28 m (height) by 0.09 m (width) by 0.07 m (length). The wall inclination of the

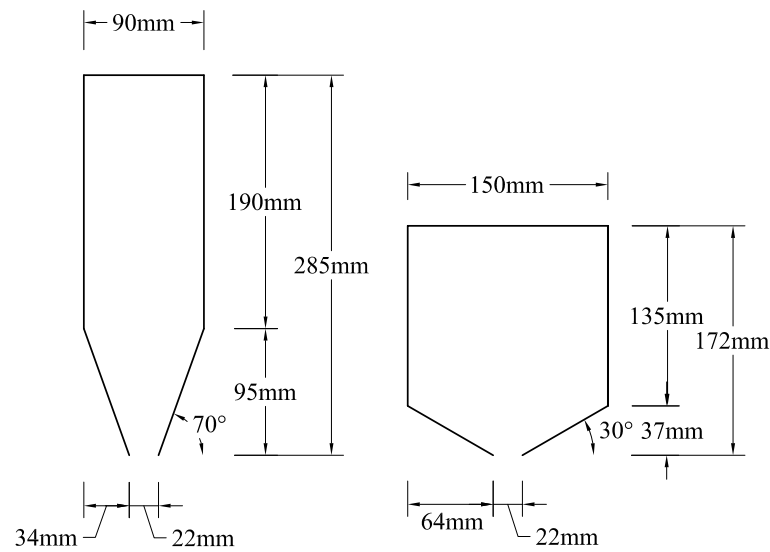


Figure A.3: The geometry of silo used in tests

hopper against the bottom was 70° . The dimensions of the funnel flow silo were: 0.17 m (height) by 0.15 m (width) by 0.07 m (length). The wall inclination of the hopper was 30° . The silos contained dry sand with a mean grain diameter of 0.5 mm (Fig. A.3). The outflow was due to gravitation. The experimental flow patterns during mass and funnel flow are shown in Fig. A.4.

The effects of the migration rules and transition probabilities are shown in Figs. A.5–A.12 for the mass flow silo and the funnel flow silo using 50 000 cells (Fig. A.4 b). The dimensions of the cells were $1 \times 1 \text{ mm}^2$. In addition, diagrams are attached indicating the amount of particles moving through each cell (darker regions denote greater numbers of flowing particles). The distribution of the transition probability values near each void was assumed to be symmetric. The maximum values of p_i were assumed in the middle or at sides of cells. The following random probabilistic values were used ($\sum p_i = 1$): $p_1 = 0.45$, $p_2 = 0.1$, $p_3 = 0.45$ and $p_1 = 0.15$, $p_2 = 0.7$, $p_3 = 0.15$ (rules **A** and **C** of Fig. A.2), $p_1 = 0.3$, $p_2 = 0.15$, $p_3 = 0.1$, $p_4 = 0.15$, $p_5 = 0.3$ and $p_1 = 0.05$, $p_2 = 0.2$, $p_3 = 0.5$, $p_4 = 0.2$, $p_5 = 0.05$ (rules **B** and **D** of Fig. A.2). To visualize the simulation process, horizontal layers of various shades were introduced.

The results show that the effect of the migration rule on flow patterns is pronounced (Figs. A.5–A.12). The angle of repose of the granular material is the smallest for the migration rule **C** when the greatest probability values are at the sides of the row of cells above the void (Figs. A.7 a and A.11 a), and the largest for the migration rule **D** when the greatest probability values are in the middle (Figs. A.8 b and A.12 b). In general, the angle of repose increases when the probability values increase towards the mid-point. A satisfactory agreement of flow patterns with experimental results (for both flow types) occurs in the simulations with migration rule **B** and transition probabilities decreasing towards the mid-point (Figs. A.6 a and A.10 a). To obtain an improved agreement, the transition probabilities should be calibrated better.

The flow rate evolution during silo discharge is shown in Fig. A.13 for the migration rule **B**. The mean flow rate is expressed by the overall number of cells going out of the silo outlet per iteration. At the beginning of discharge, a constant flow rate is observed throughout the run during both types of flow, in accordance with physical experiments [69, 139]. In

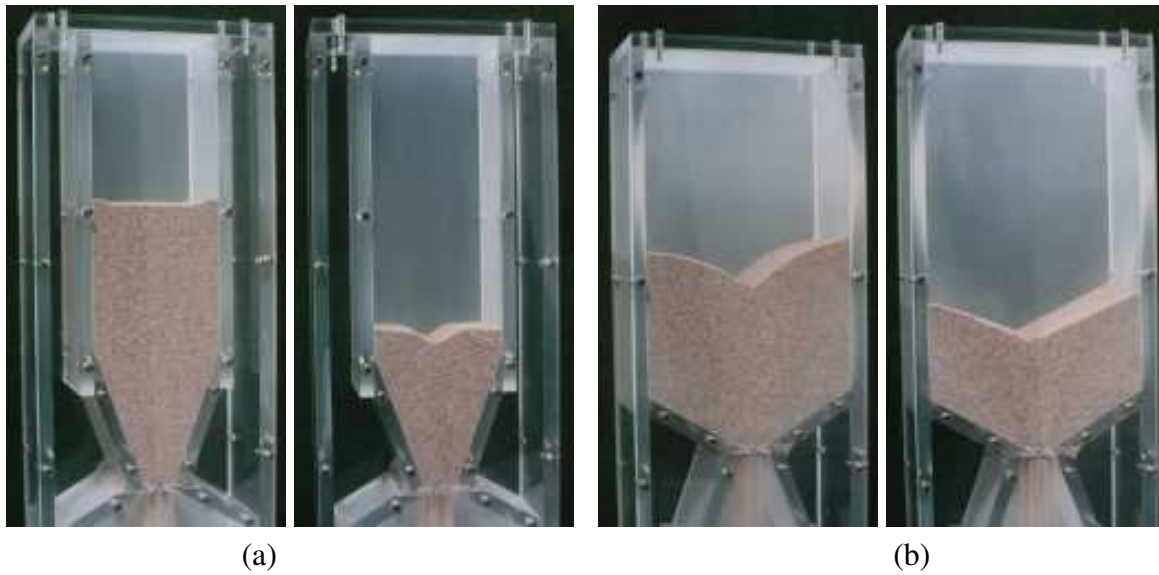


Figure A.4: Flow patterns observed in model silo tests: (a) mass flow; (b) funnel flow

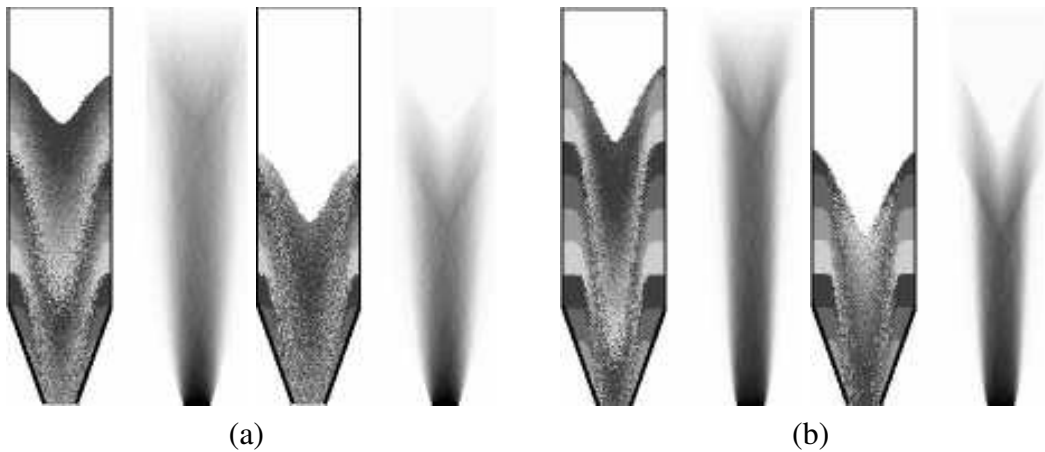


Figure A.5: Flow patterns and distribution of flow rate during granular flow in a mass flow silo (migration rule A of Fig. A.2, 50 000 cells): (a) $p_1 = p_3 = 0.45$, $p_2 = 0.1$; (b) $p_1 = p_3 = 0.15$, $p_2 = 0.7$

the final phase of funnel flow, a non-linear reduction of the flow rate is obtained due to a successive decrease of the material volume in the silo. In the case of mass flow, a drastic reduction of the flow rate takes place.

The calculation time for a silo flow with 50000 cells using a 2.0GHz PC was about 2 minutes.

Effect of the transport scheme

Figure A.14 shows the results during „disturbed” silo flow. Taking the flow of empty cells into account does not significantly influence the flow pattern. However, the material flows out more slowly.

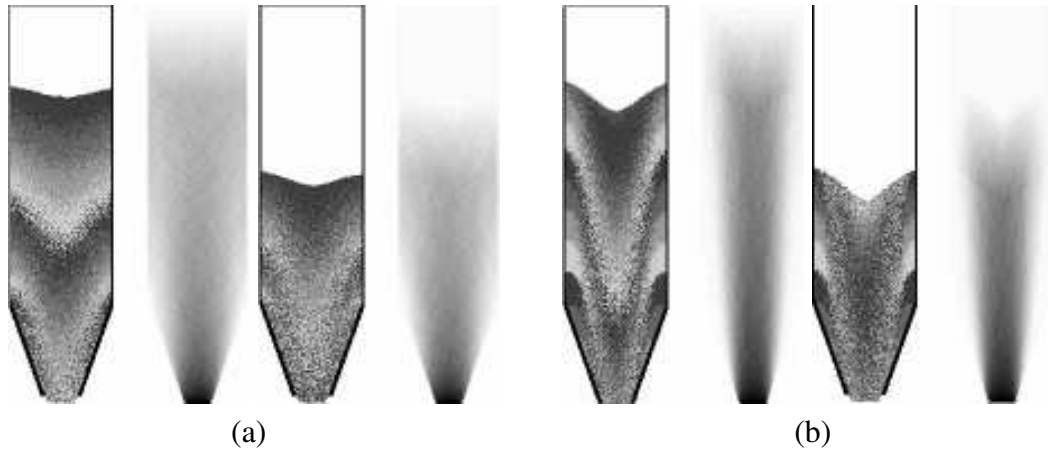


Figure A.6: Flow patterns and distribution of flow rate during granular flow in a mass flow silo (migration rule **B** of Fig. A.2, 50 000 cells): (a) $p_1 = p_5 = 0.3$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$; (b) $p_1 = p_5 = 0.05$, $p_2 = p_4 = 0.2$, $p_3 = 0.5$

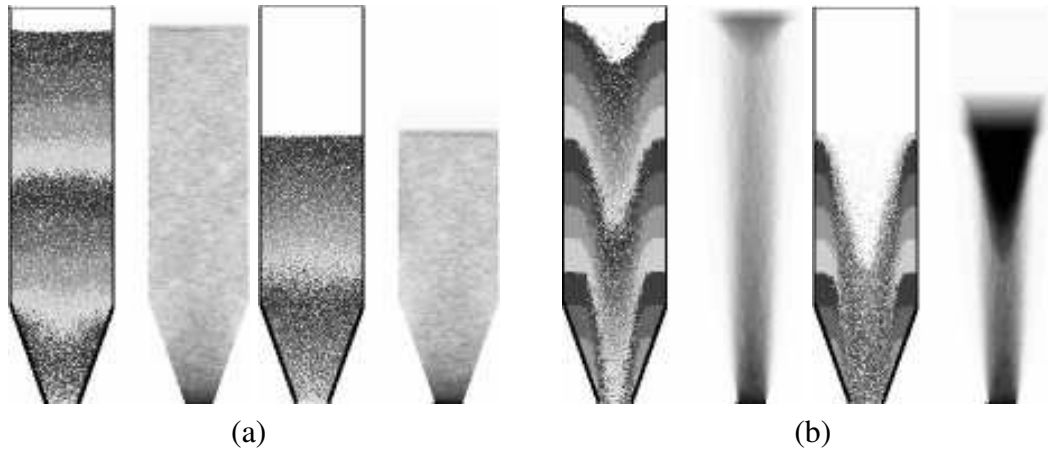


Figure A.7: Flow patterns and distribution of flow rate during granular flow in a mass flow silo (migration rule **C** of Fig. A.2, 50 000 cells): (a) $p_1 = p_3 = 0.45$, $p_2 = 0.1$; (b) $p_1 = p_3 = 0.15$, $p_2 = 0.7$

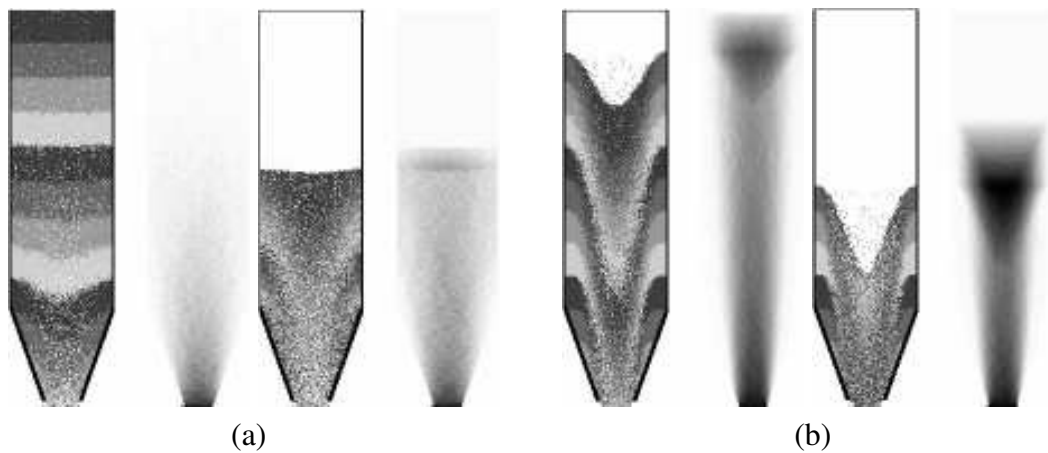


Figure A.8: Flow patterns and distribution of flow rate during granular flow in a mass flow silo (migration rule **D** of Fig. A.2, 50 000 cells): (a) $p_1 = p_5 = 0.3$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$; (b) $p_1 = p_5 = 0.05$, $p_2 = p_4 = 0.2$, $p_3 = 0.5$



Figure A.9: Flow patterns and distribution of flow rate during granular flow in a funnel flow silo (migration rule **A** of Fig. A.2, 50 000 cells): (a) $p_1 = p_3 = 0.45$, $p_2 = 0.1$; (b) $p_1 = p_3 = 0.15$, $p_2 = 0.7$

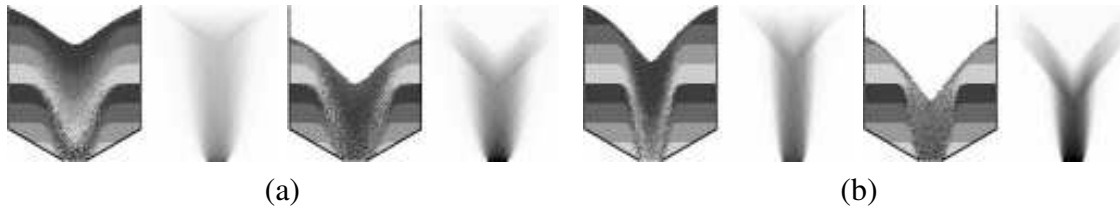


Figure A.10: Flow patterns and distribution of flow rate during granular flow in a funnel flow silo (migration rule **B** of Fig. A.2, 50 000 cells): (a) $p_1 = p_5 = 0.3$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$; (b) $p_1 = p_5 = 0.05$, $p_2 = p_4 = 0.2$, $p_3 = 0.5$

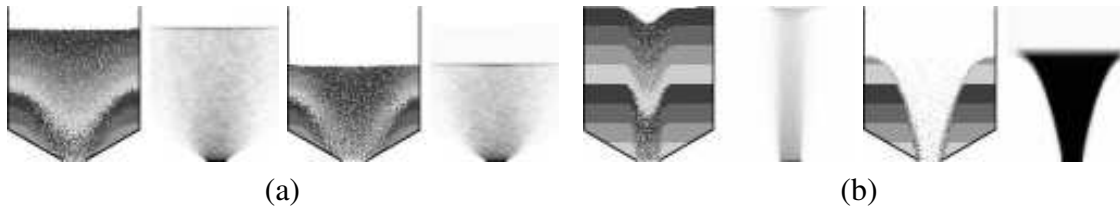


Figure A.11: Flow patterns and distribution of flow rate during granular flow in a funnel flow silo (migration rule **C** of Fig. A.2, 50 000 cells): (a) $p_1 = p_3 = 0.45$, $p_2 = 0.1$; (b) $p_1 = p_3 = 0.15$, $p_2 = 0.7$

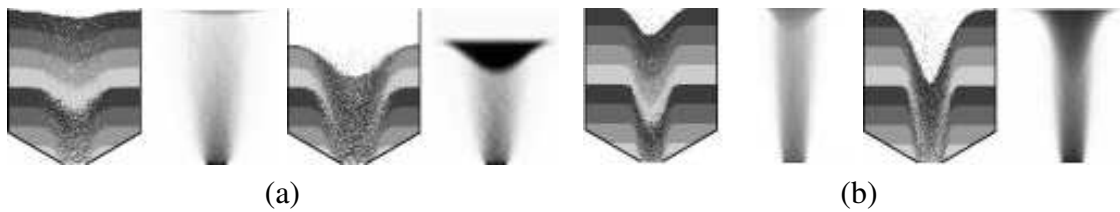


Figure A.12: Flow patterns and distribution of flow rate during granular flow in a funnel flow silo (migration rule **D** of Fig. A.2, 50 000 cells): (a) $p_1 = p_5 = 0.3$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$; (b) $p_1 = p_5 = 0.05$, $p_2 = p_4 = 0.2$, $p_3 = 0.5$

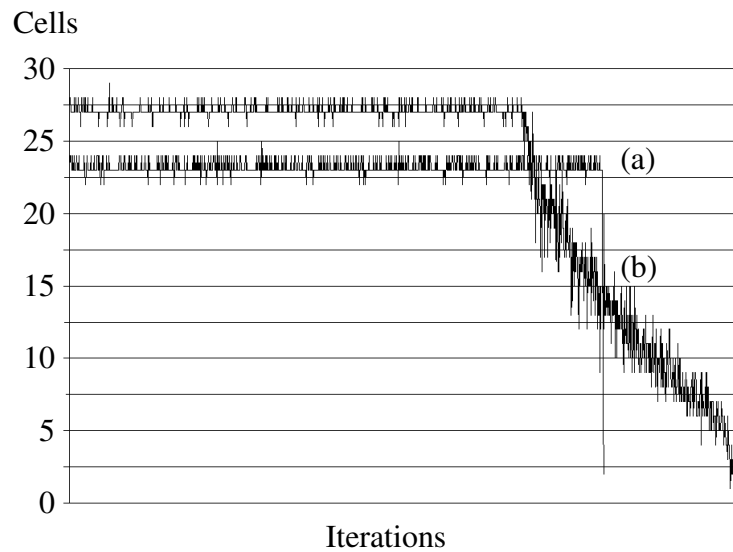


Figure A.13: Calculated flow rate during granular flow in (a) mass and (b) funnel flow silos (migration rule **B** of Fig. A.2, $p_1 = p_5 = 0.3$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$)

Effect of outlet velocity

In Fig. A.15 results of silo flow with a constant outlet velocity are shown. The reduction in the outflow velocity diminishes the flow rate in the entire silo. The flow rate becomes more uniform, while the angle of repose of the granular material increases during mass flow and decreases during funnel flow.

Effect of wall roughness

The effect of increased wall roughness is presented in Fig. A.16. In this case, a different migration scheme was assumed in the wall region at the distance of 10 mm from the silo walls. The transition probability for flow was assumed to decrease there in accordance with a cosine function. The bulk material flows out more slowly. At the walls, there appears a narrow wall shear zone.

Effect of particle numbers

The effect of the number of cells during funnel flow silo is shown in Fig. A.17. The calculations were performed with 200 000 cells. The dimensions of cells were $0.5 \times 0.5 \text{ mm}^2$. An increase in the number of cells causes an increase of the granulate's angle of repose.

Effect of three-dimensional simulation

Three-dimensional calculations of flow patterns in the mass and funnel flow silo of Fig. A.4 were carried out with 3 500 000 cells by assuming the migration rule **B** of Fig. A.2 in both horizontal directions along the cross-section (Fig. A.18 a), with $\sum p_i = 1$. Figures A.18 b and A.18 c demonstrate the flow pattern in different horizontal sections. The results are similar to those of two-dimensional calculations (due to the lack of wall friction). The calculation time using a 2.0GHz PC was about 2.5 hours.

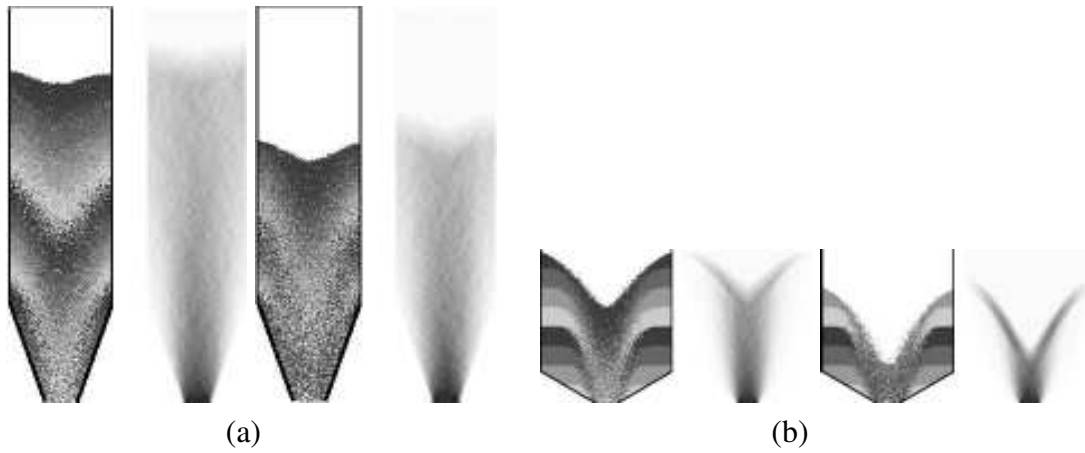


Figure A.14: Flow patterns and distribution of flow rate during granular disturbed flow in a mass (a) and funnel (b) flow silo (migration rule **B** of Fig. A.2, $p_1 = p_5 = 0.3$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$, 50 000 cells)

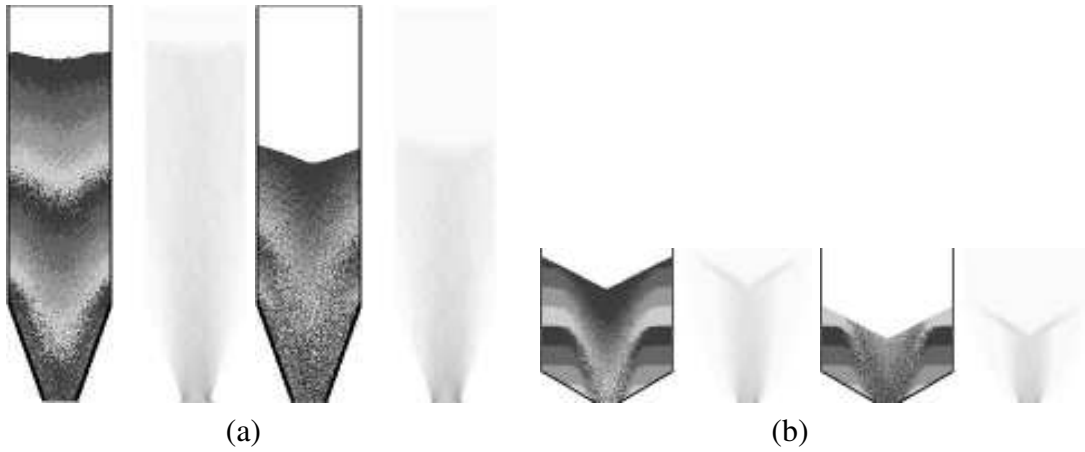


Figure A.15: Flow patterns and distribution of flow rate during granular flow with a constant (controlled) outflow velocity in a (a) mass and (b) funnel flow silo (migration rule **B** of Fig. A.2, $p_1 = p_5 = 0.3$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$, 50 000 cells)

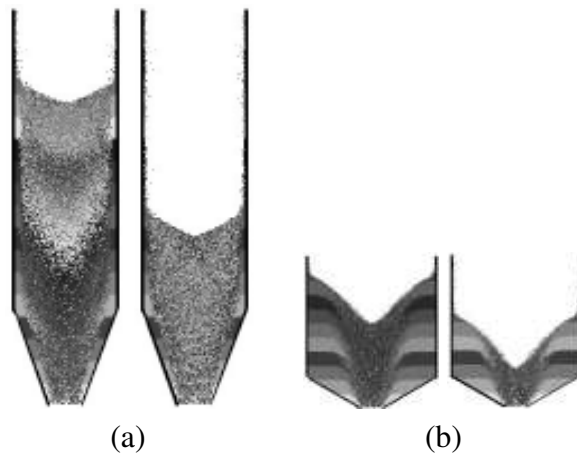


Figure A.16: Flow patterns during granular flow with rough walls in a (a) mass and (b) funnel flow silo (migration rule **B** of Fig. A.2, $p_1 = p_5 = 0.3$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$, 50 000 cells)

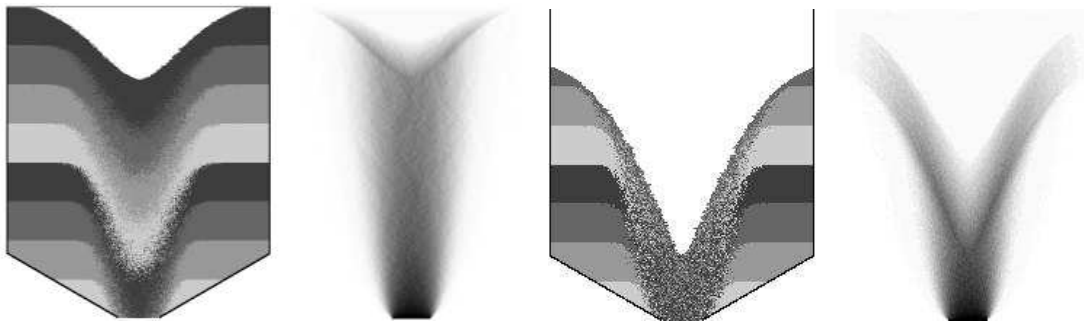


Figure A.17: Flow patterns and distribution of flow rate during granular flow in a funnel flow silo (migration rule **B** of Fig. A.2, $p_1 = p_5 = 0.3$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$, 200 000 cells)

Effect of the lattice type

The effect of the lattice type is shown in Figs. A.19 and A.20. The calculations were carried out with a hexagonal grid and the migration rule of Fig. A.19 a for two different distributions of transition probability values (smaller at the sides and greater above the void or inversely). The agreement of results of the flow pattern is also satisfactory when compared with experiments (Fig. A.4). In the case of a hexagonal grid, the angle of repose decreases when transition probabilities are the greatest at the sides and increases when they are the smallest there (with respect to a quadratic grid).

Effect of silo inserts

Silo inserts are used to improve the flow properties of bulk solids and to decrease silo wall pressures [69, 120, 129]. A wedge-shaped insert and an internal hopper change funnel flow into mass flow [61, 82, 147], while a perforated emptying tube [84, 128] or two inclined emptying tubes [83] change mass flow into funnel flow. When a wedge-shaped insert is located near the transition between the bin and the hopper, a significant reduction in pressures on the wall hopper is obtained [61]. An insert in the form of an internal hopper inside the main hopper (the cone-in-cone concept) is applied to obtain mass flow at considerably large hopper inclinations at which funnel flow occurs [61, 82]. The outlet of the internal hopper is usually equal to the outlet of the main hopper. The flow pattern is affected mainly by the wall inclination of the internal hopper and the horizontal distance between the main and the internal hopper [61]. The perforated emptying tube (also called a depression column or an anti-dynamic tube) is hanged in the middle of the silo or along the wall [84, 129]. It is usually made of steel and is connected to a silo's roof structure. It works only for non-cohesive materials. At the same time, two inclined discharging tubes are located symmetrically on the bin bottom [83]. Thanks to inserts of both of these types, a significant reduction of wall loads, flow rate and amplitude of dynamic pulsations is obtained due to the occurrence of funnel flow. In the case of too high pressures in silos, the method is cheaper than strengthening the silo structure with high-strength steel cables.

The calculations were carried out with a symmetric wedge-shaped insert located at two different heights of a funnel flow silo. During the analysis, 200 000 quadratic cells with the migration rule **B** of Fig. A.2 were used. The distribution of probability values was assumed to be symmetric with the largest values at both ends, diminishing towards the center of the layer above the void ($p_1 = p_5 = 0.3$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$). Figure A.21 demonstrates

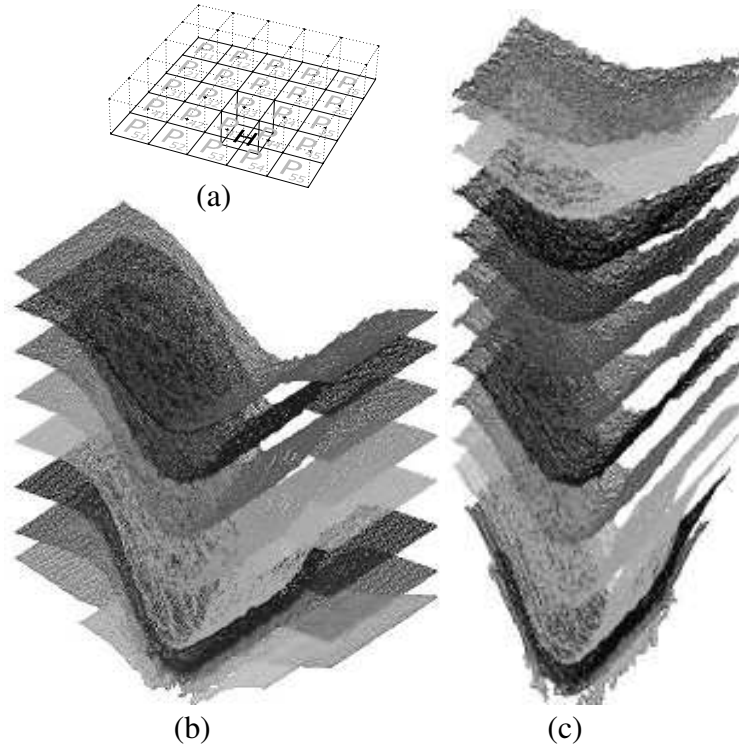


Figure A.18: Migration rule (a) and flow patterns during three-dimensional granular flow in a (b) funnel and (c) mass flow silo using quadratic cells ($p_{11} = p_{15} = p_{55} = p_{51} = 0.060$, $p_{12} = p_{21} = p_{14} = p_{25} = p_{41} = p_{45} = p_{54} = p_{52} = 0.053$, $p_{13} = p_{31} = p_{53} = p_{35} = 0.045$, $p_{22} = p_{42} = p_{44} = p_{24} = 0.023$, $p_{23} = p_{32} = p_{43} = p_{34} = 0.015$, $p_{33} = 0.003$, 3 500 000 cells)

the flow patterns and distribution of flow rate during granular flow in a silo with a wedge-shaped insert. The presence of the insert, its size and position influence significantly the flow pattern and rate. Mass flow is obtained for the insert's position of Fig. A.21 c.

The results for an internal hopper located at the transition between the bin and the main hopper are presented in Fig. A.22. During this analysis, 200 000 quadratic cells were used (migration rule **B** of Fig. A.2). Two different wall inclinations of the internal hopper were assumed (Figs. A.22 b and A.22 c). The flow in a silo without the internal hopper is of the funnel type (Fig. A.22 a). The application of an internal hopper induces mass flow (Figs. A.22 b and A.22 c), in accordance with experiments [61]. However, if the wall inclination of the internal hopper to the bottom is too small, the material flows too slowly in the center (Fig. A.22 c).

Figure A.23 presents the calculated flow patterns in silos including two inclined tubes above the outlet and a perforated tube in the middle of the silo (200 000 quadratic cells, migration rule **B** of Fig. A.2). As the simulation results indicate, these two inserts promote funnel flow.

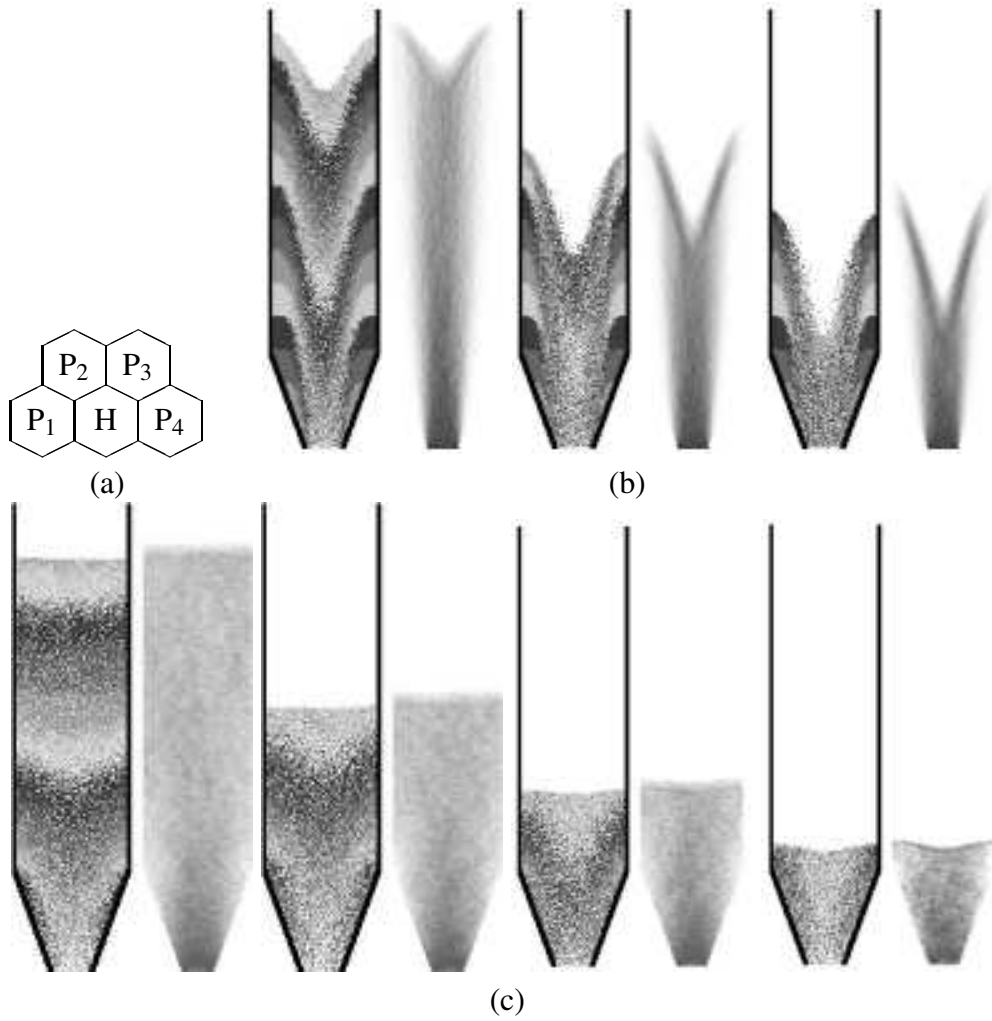


Figure A.19: Migration rule (a), flow patterns and distribution of flow rate during granular flow in a mass flow silo (two-dimensional hexagonal grid): (b) $p_1 = p_4 = 0.1$, $p_2 = p_3 = 0.4$ (c) $p_1 = p_4 = 0.4$, $p_2 = p_3 = 0.1$;

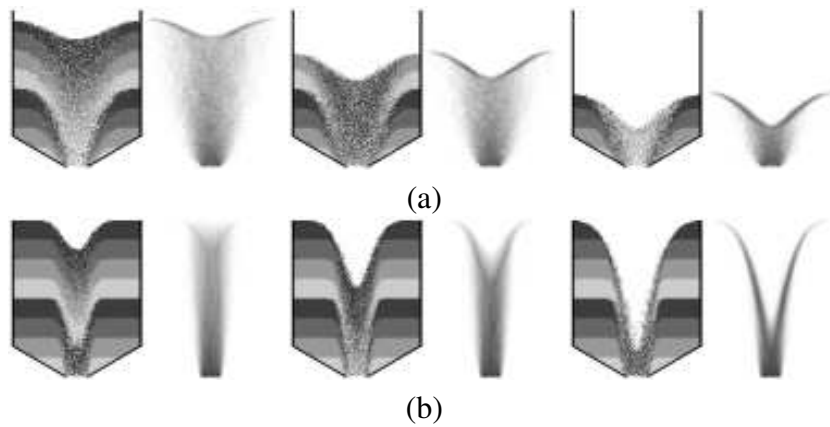


Figure A.20: Flow patterns and distribution of flow rate during granular flow in a funnel flow silo (two-dimensional hexagonal grid): (a) $p_1 = p_4 = 0.4$, $p_2 = p_3 = 0.1$; (b) $p_1 = p_4 = 0.1$, $p_2 = p_3 = 0.4$

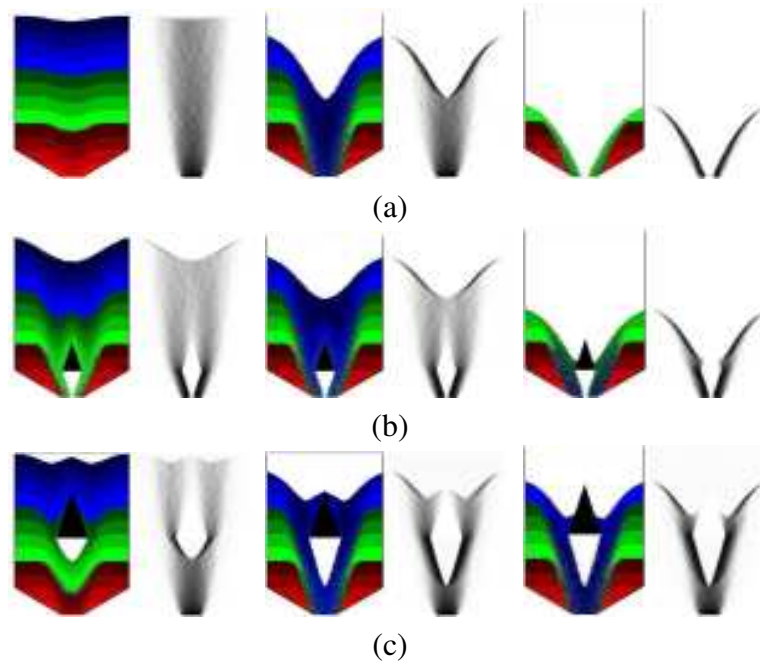


Figure A.21: Flow patterns and distribution of flow rate during granular flow in a silo without and with inserts (quadratic cells, migration rule **B** of Fig. A.2): (a) silo without insert, (b) and (c) silo with insert ($p_1 = p_5 = 0.30$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$)

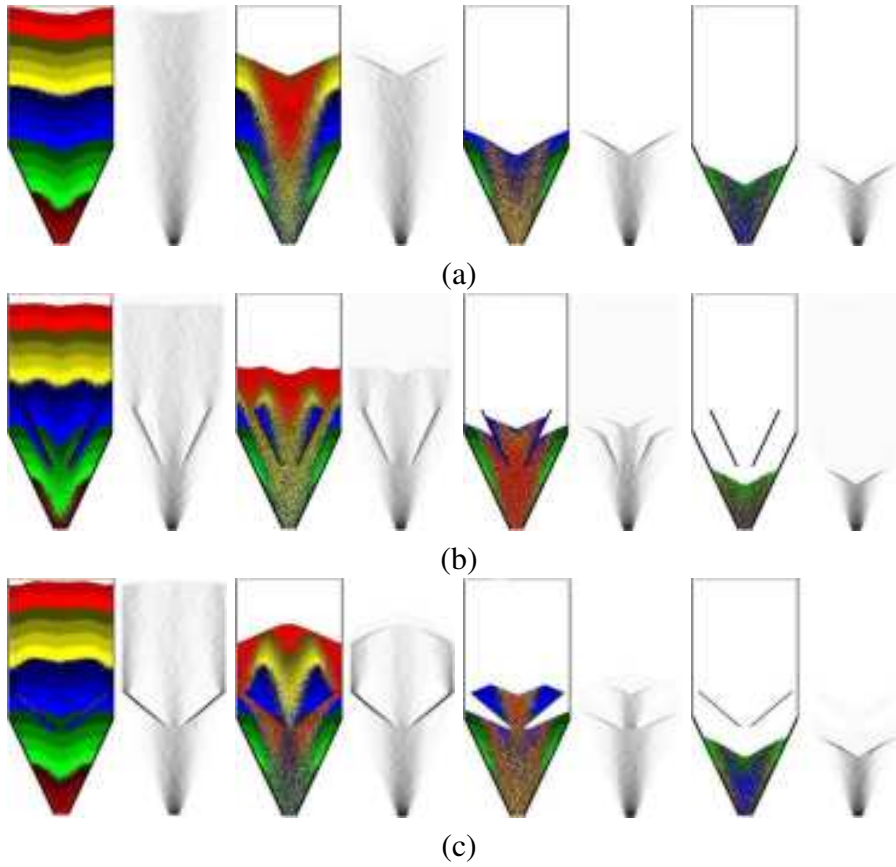


Figure A.22: Flow patterns and distribution of flow rate during granular flow in a silo without an internal hopper (a) and with an internal hopper (b), (c) (quadratic cells, migration rule **B** of Fig. A.2, $p_1 = p_5 = 0.30$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$)

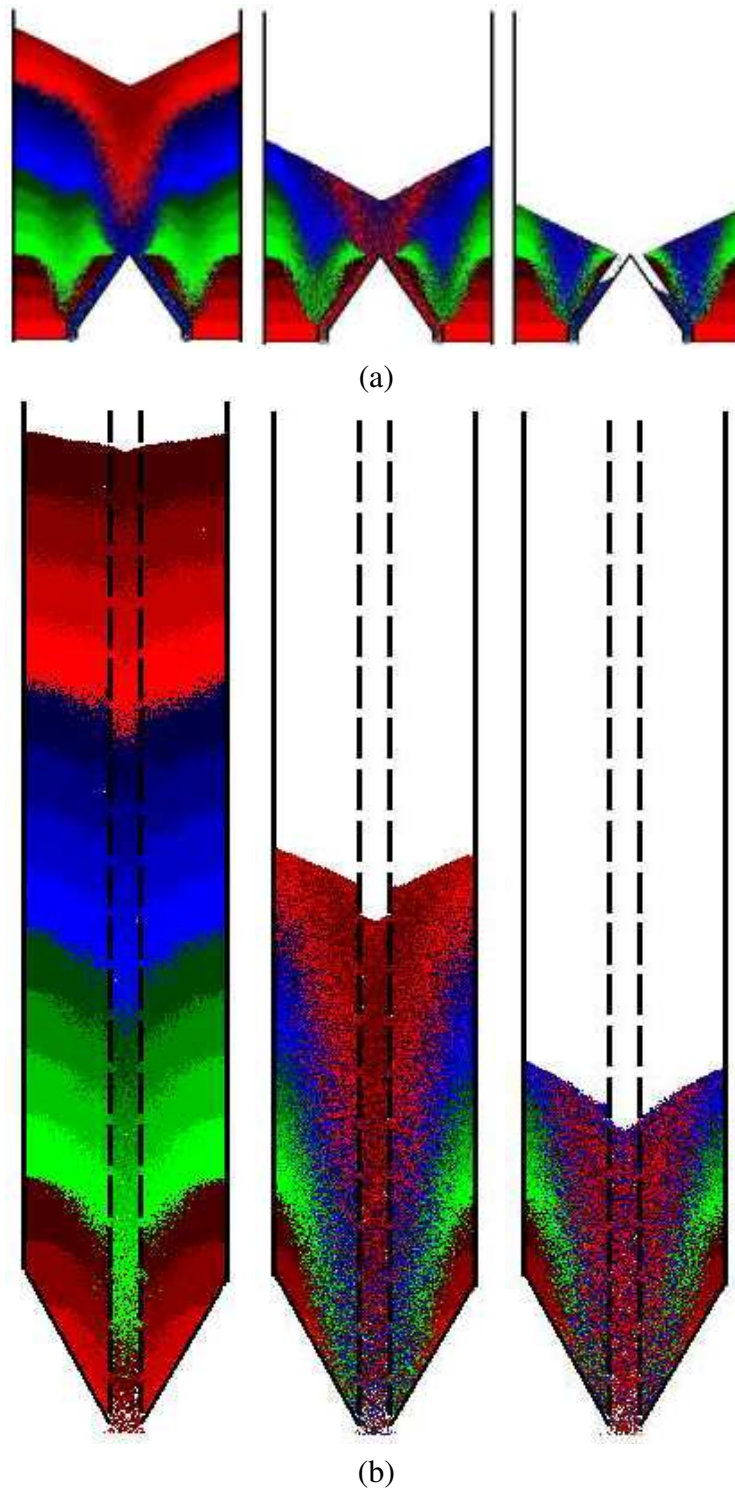


Figure A.23: Flow patterns and distribution of flow rate during granular flow in a silo with two inclined tubes (a) and with a perforated tube (b) (quadratic cells, migration rule **B** of Fig. A.2, $p_1 = p_5 = 0.30$, $p_2 = p_4 = 0.15$, $p_3 = 0.1$)

A.4 Description of the improved model

In a cellular automaton based on a lattice gas automaton [2, 119], some mechanical principles were taken into account. The model has enabled us to investigate propagating dilatant zones, which are an inherent characteristics of each discharge process in mass and funnel flow hoppers and parallel-converging bins during gravitational and controlled outflow of dense granular material composed of rough grains [4, 5, 24, 44, 56, 105, 106, 149]. In this mechanical model, each site of a hexagonal lattice could be occupied by one or more particles, a boundary wall or empty. The particles moved on the nearest-neighbor bonds (with unit velocity) and the second-nearest neighbors of the lattice (with unit velocity multiplied by the factor $\sqrt{3}$). Each particle at each site had 13 Boolean states which were related to velocity vectors v_i ($i = 0, 1, 2, \dots, 12$, see Fig. A.24 a). The direction of motion of a particle, v , could be toward any of its 12 nearest neighbors ($i = 1, 2, \dots, 12$) or the particle could be at rest ($v_0 = 0$). Thus, at the beginning of our calculations, the number of particles per site had a maximum value of 13 and a minimal value of 0 (an empty site). The time evolution consisted of one collision step and two propagation steps. In the collision step, both particles changed their velocities or remained at rest at the site by losing their total kinetic energy due to dissipation. In the first propagation step, particles (more than one) with no velocity were scattered randomly to the nearest empty sites with low numbers of particles. In the second propagation step, colliding particles (after changing their velocities) were transferred in the direction of their velocities to the nearest sites, where they collided again (see Fig. A.24 b,c). In general, collisions conserved mass and momentum. The simulations have shown that the calculation order of the lattice sites has no influence on the results. The following six parameters were assumed in the model: two collision parameters (p, q), two friction parameters (b, k) and two gravity parameters (g, h). All probabilistic parameters were in the range from 0 to 1.

The collision parameter p determined the number of particles, n , which remained at rest after a collision (due to energy dissipation). The n parameter was calculated from the following formula: $p^{(n+1)} < r < p^n$, where r is a random number. As some particles were stopped after collisions, other particles could leave the sites at higher velocities, due to the conservation of momentum. When more than one particle was stopped and momentum could not be conserved, particles moved in random directions.

The q parameter described the loss of energy during each collision. The number of

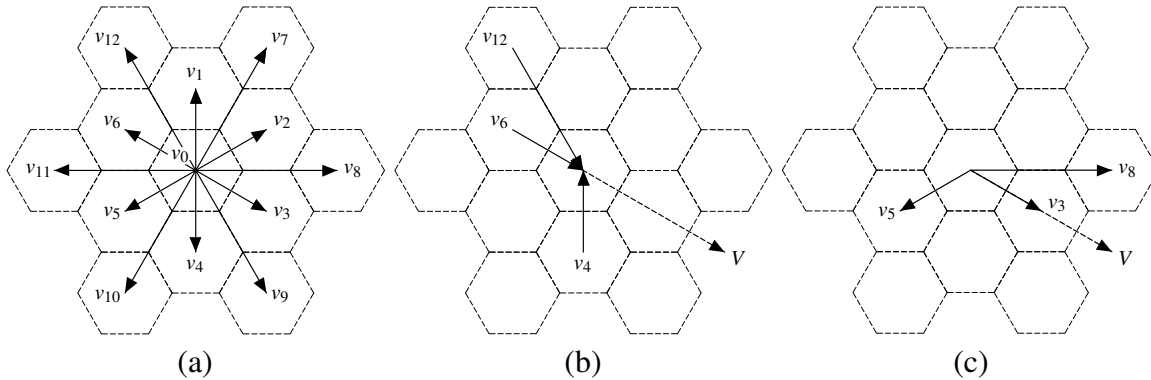


Figure A.24: Vectors of particle velocities at each site (a). Example collision: velocity vectors of particles (b) before, and (c) after collision (V – resultant velocity)

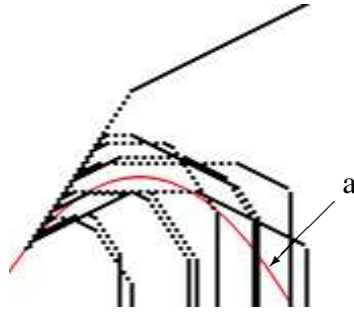


Figure A.25: Example traces of single particles in the gravity field thrown from the same point in the direction of v_7 of Fig. A.24 a compared to a parabolic trajectory (a — parabolic trajectory)

random particles, l , (which were not considered when calculating the resultant velocity) was obtained from the following formula: $q^{(l+1)} < r < q^l$. Due to the loss of energy, particles left sites with lower velocities.

The total number of possible combinations for collisions was equal to $2^{12} = 4096$ and the number of possible velocity directions was 108. To accelerate the calculation process, an appropriate table of collisions was generated (Fig. A.26). If parameters p and q were equal to zero, the sum of velocity vectors outgoing from each site was equal to the sum of incoming velocity vectors (elastic collisions took place).

The friction parameter b described wall roughness. It was introduced in two ways.

The first way was based on the model proposed in [119]. If parameter b was other than zero (rough walls), particles bounced back from the wall in random directions. If $b = 0$ (smooth walls), the angle at which particles hit the wall was equal to the angle at which they were reflected.

In the second, particles hitting the wall were stopped there with a probability of b (the increase in wall roughness corresponded to the increase in parameter b). Afterwards, they were scattered randomly to the nearest empty sites with low contents of particles.

The friction parameter k described the internal interaction among particles. When parameter k was other than zero, certain migration directions could become more dominant (to induce movement in one specific direction). In our simulations, the horizontal direction was assumed to be dominant. Thus, the resulting velocity vector of outgoing particles was replaced by the sum of random velocity vectors in the directions v_2, v_3, v_5, v_6, v_8 , and v_{11} (Fig. A.24 a) with a probability of k .

The gravity parameter g was taken into account during each collision by adding the vertical velocity, v_4 , (Fig. A.24 a) to the resultant velocity vector of outgoing particles with a probability of g . Thus, the particles were accelerated downwards after each collision. At the same time, the gravity parameter h was introduced to improve the direction of the resultant velocity vector so that the direction would become parabolic for single particles (Fig. A.25). The particle remaining at rest at a given site (v_0) could move downwards to the v_0 state in the neighboring empty site with no velocity only in one of the chosen directions v_3, v_4, v_5, v_9 and v_{10} (Fig. A.24 a). If $h = 1$, particles without velocities moved downwards (density became more uniform). However, if $h = 0$, particles without velocities did not move.

The initial density of the granulate at the onset of flow dependent on parameters p, q, g and k was generated during silo filling when the outlet was closed. Having settled in the silo, the granular material reached certain density, which increased with increasing p and g and decreasing q and k .

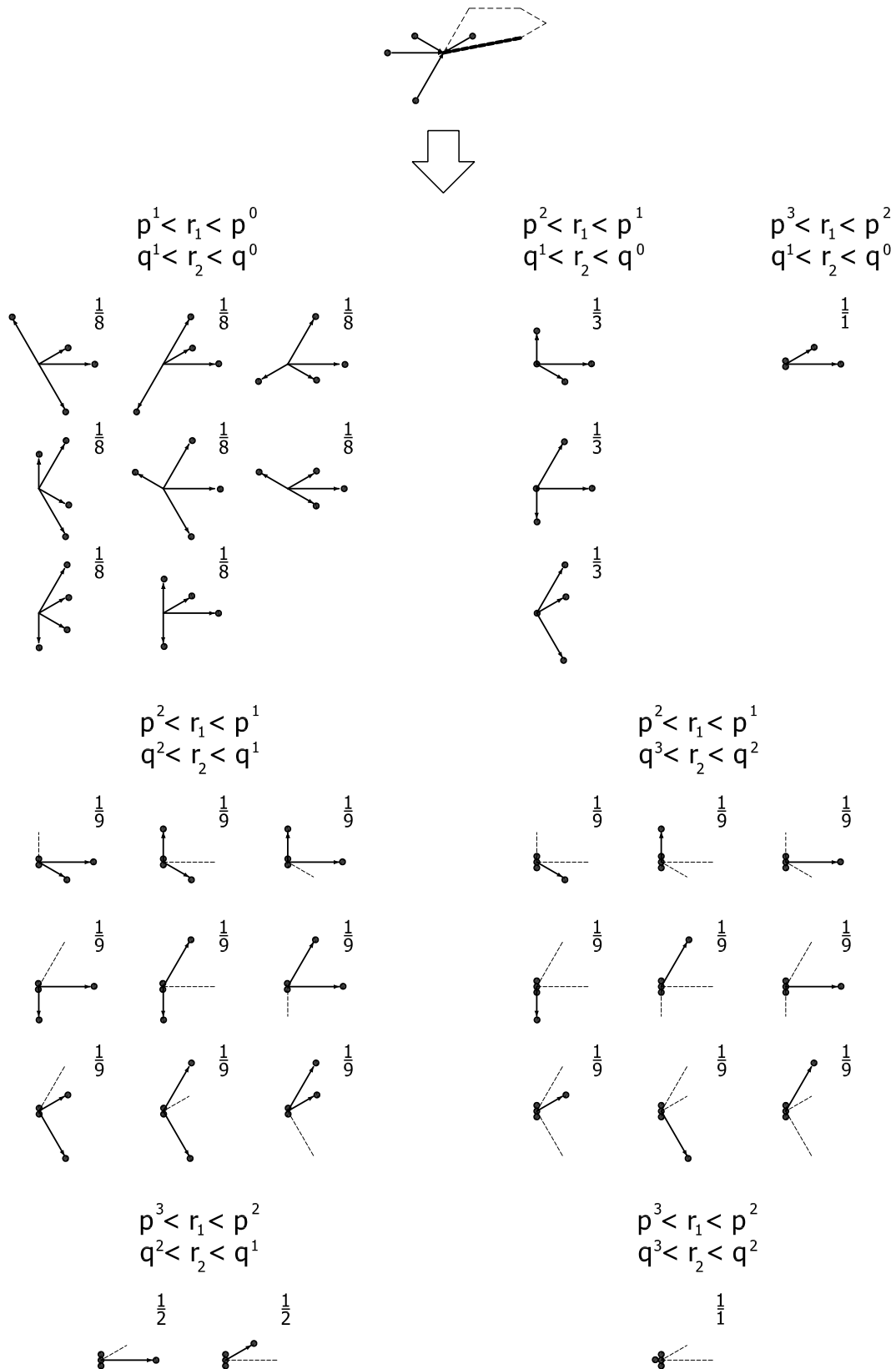


Figure A.26: Examples of collision rules for 4 particles flowing to one random site (arrows represent the directions of moving particles and full dots stand for particles)

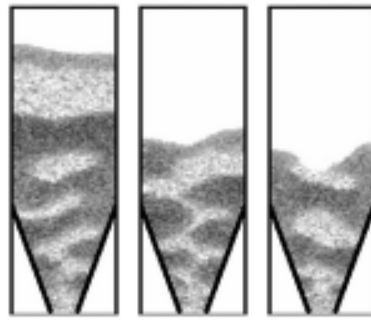


Figure A.27: Flow pattern in a mass flow silo: $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.05$

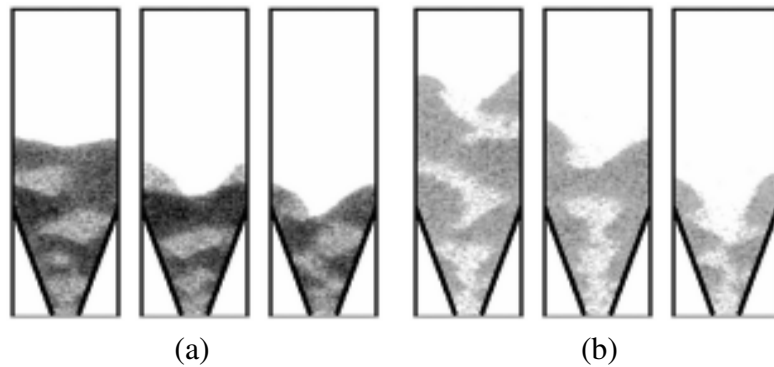


Figure A.28: Flow pattern in a mass flow silo: (a) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = \mathbf{0.95}$, $h = 0.05$ (b) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = \mathbf{0.05}$, $h = 0.05$

As mentioned above, the model is based on the approach proposed in [2, 119]. The following novelties were introduced: the number of possible vector directions was enhanced (13 instead of 7), density was not used as an additional parameter (density was determined by other parameters), gravitation was described by two parameters, rough walls were able to stop flowing particles, the method of describing collisions allowed to stop any number of particles, and a new parameter was introduced to describe the internal interaction among particles.

A.5 Simulation results for improved model

2D—results of simulations are shown in Figs. A.27–A.33 (mass flow silo) and Figs. A.34–A.40 (funnel flow silo). The shape and dimensions of the silos were similar to those of the laboratory tests described in Sect. A.2 (also in [88, 95]). The number of cells was 35 000. The effect of parameters g, h, k, b, p and q is shown in Figs. A.28 and A.35, Figs. A.29 and A.36, Figs. A.30 and A.37, Figs. A.31 and A.38, Figs. A.32 and A.39, and Figs. A.33 and A.40, respectively for the mass and the funnel flow silos. The darker regions are associated with higher densities, while the lighter regions are associated with lower densities. In the calculations with parameter b , describing wall roughness, the second method was applied (the first, in contrast to results in a vertical pipe [5], did not affect the results in a silo). Initially, the values of parameters (Figs. A.27 and A.34) were assumed to be random. Then, small and great parameter values were chosen for a comparison (Figs. A.28–A.33 and A.35–A.40).

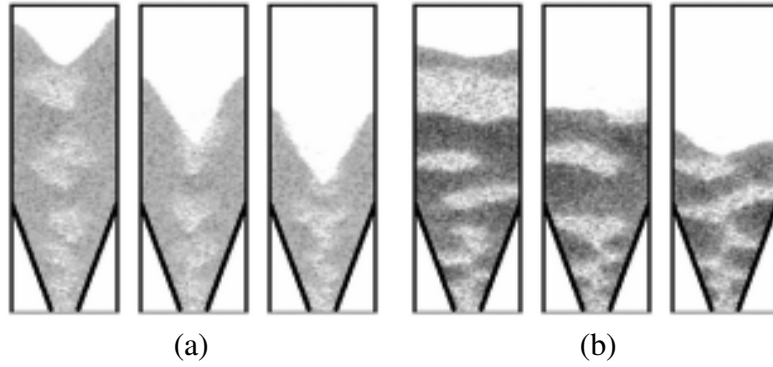


Figure A.29: Flow pattern in a mass flow silo: (a) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.8$ (b) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.0$

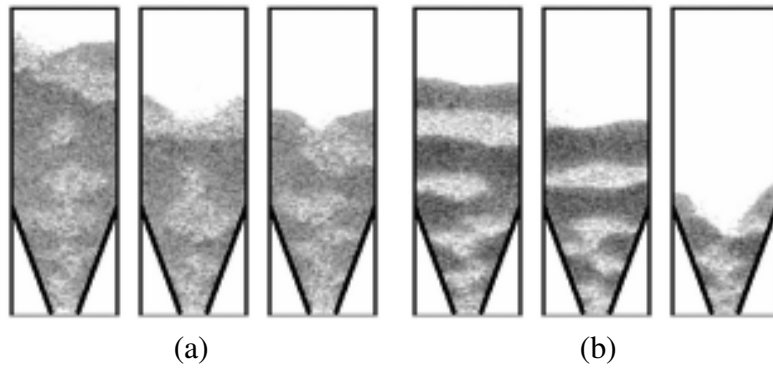


Figure A.30: Flow pattern in a mass flow silo: (a) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.8$, $g = 0.2$, $h = 0.05$ (b) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.0$, $g = 0.2$, $h = 0.05$

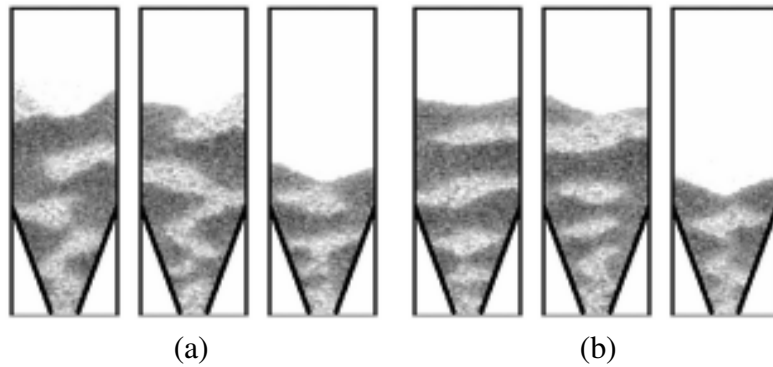


Figure A.31: Flow pattern in a mass flow silo: (a) $p = 0.2$, $q = 0.2$, $b = 0.8$, $k = 0.05$, $g = 0.2$, $h = 0.05$ (b) $p = 0.2$, $q = 0.2$, $b = 0.0$, $k = 0.05$, $g = 0.2$, $h = 0.05$

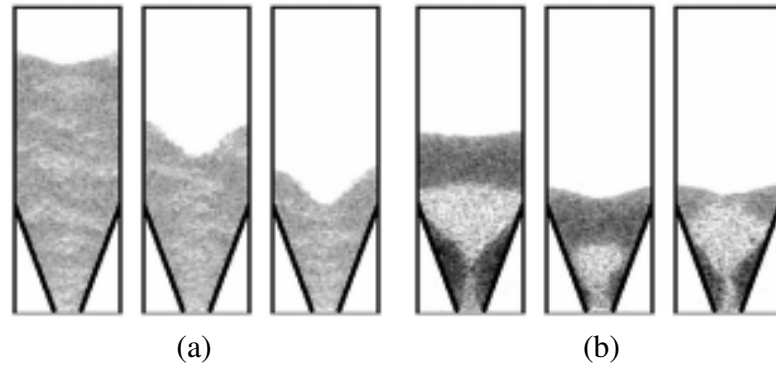


Figure A.32: Flow pattern in a mass flow silo: (a) $p = \mathbf{0.8}$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.05$ (b) $p = \mathbf{0.0}$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.05$

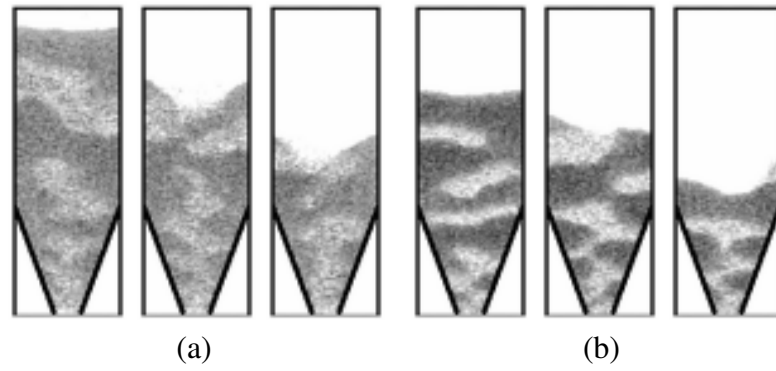


Figure A.33: Flow pattern in a mass flow silo: (a) $p = 0.2$, $q = \mathbf{0.8}$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.05$ (b) $p = 0.2$, $q = \mathbf{0.0}$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.05$

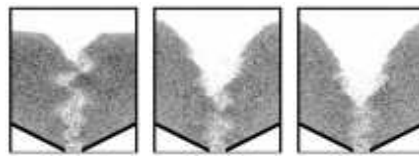


Figure A.34: Flow pattern in a funnel flow silo: $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.05$

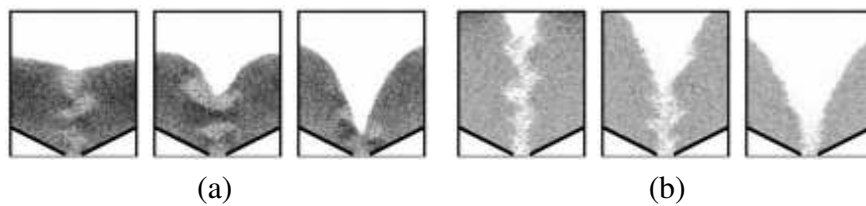


Figure A.35: Flow pattern in a funnel flow silo: (a) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = \mathbf{0.95}$, $h = 0.05$ (b) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = \mathbf{0.05}$, $h = 0.05$

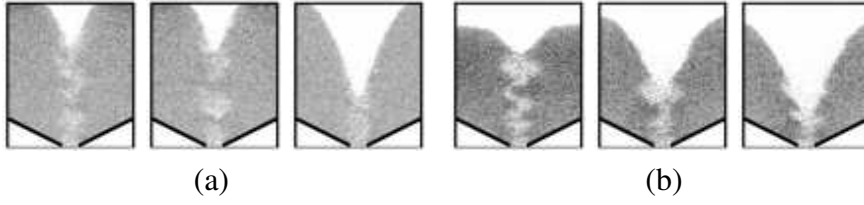


Figure A.36: Flow pattern in a funnel flow silo: (a) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.8$ (b) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.0$

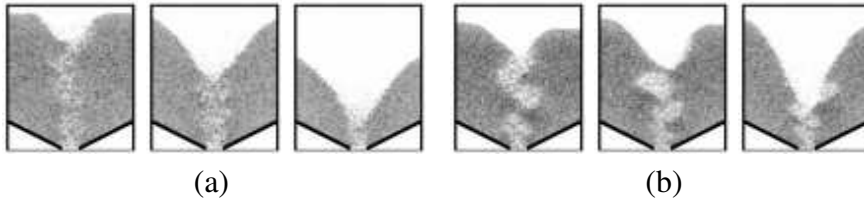


Figure A.37: Flow pattern in a funnel flow silo: (a) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.8$, $g = 0.2$, $h = 0.05$ (b) $p = 0.2$, $q = 0.2$, $b = 0.05$, $k = 0.0$, $g = 0.2$, $h = 0.05$

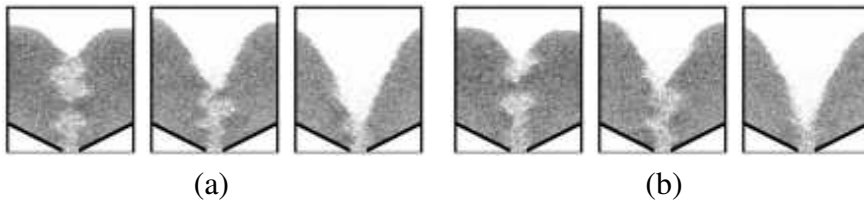


Figure A.38: Flow pattern in a funnel flow silo: (a) $p = 0.2$, $q = 0.2$, $b = 0.8$, $k = 0.05$, $g = 0.2$, $h = 0.05$ (b) $p = 0.2$, $q = 0.2$, $b = 0.0$, $k = 0.05$, $g = 0.2$, $h = 0.05$

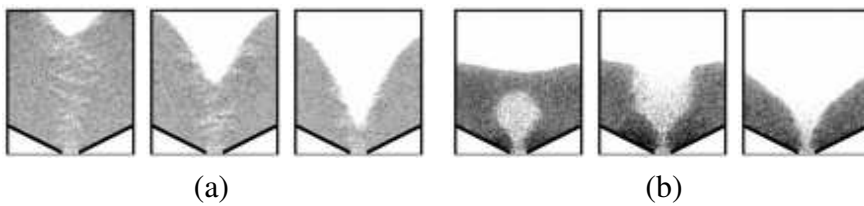


Figure A.39: Flow pattern in a funnel flow silo: (a) $p = 0.8$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.05$ (b) $p = 0.0$, $q = 0.2$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.05$

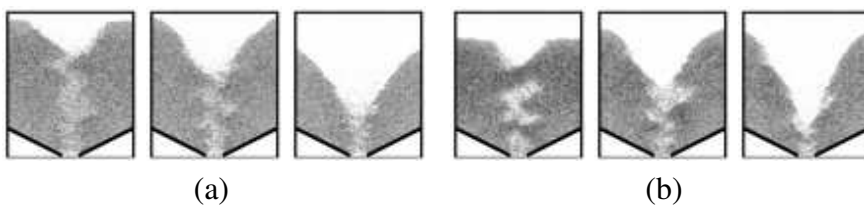


Figure A.40: Flow pattern in a funnel flow silo: (a) $p = 0.2$, $q = 0.8$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.05$ (b) $p = 0.2$, $q = 0.0$, $b = 0.05$, $k = 0.05$, $g = 0.2$, $h = 0.05$

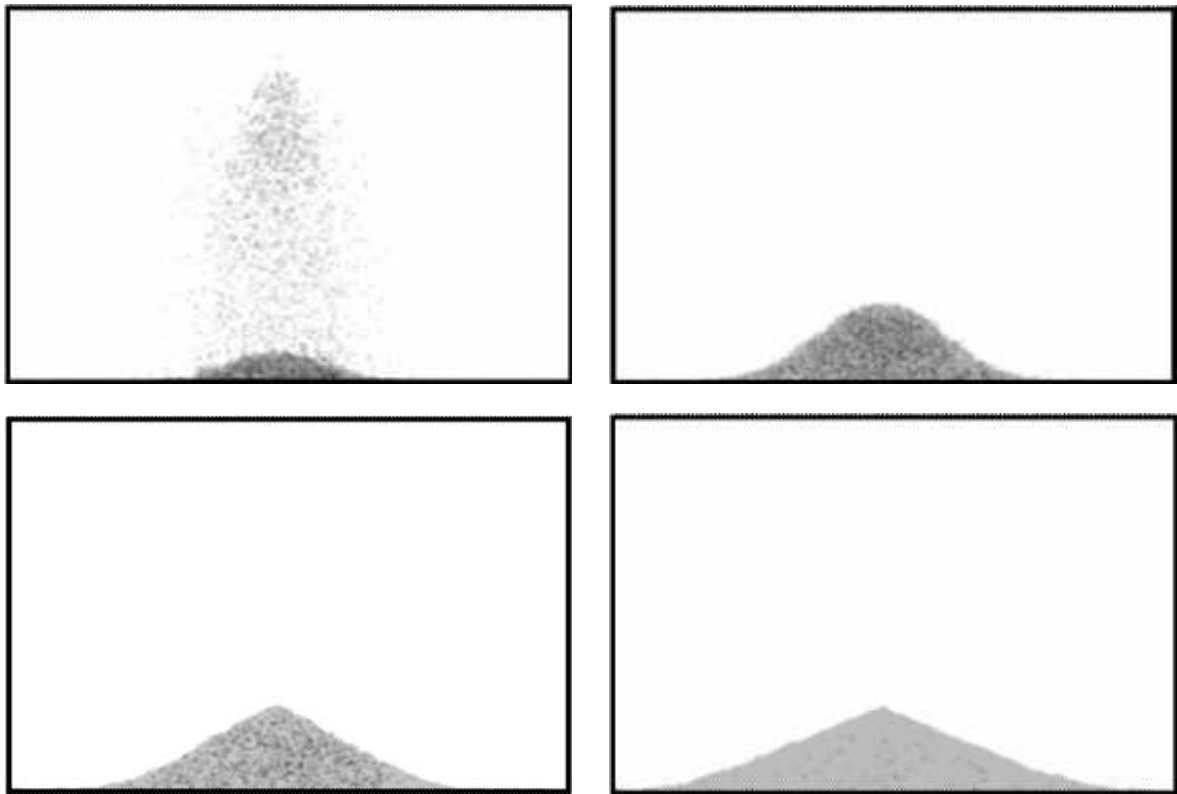


Figure A.41: Formation of a granular pile

In all cases, regions of various density occur during granular flow. The shape of dilatant zones is similar for both flow types and is strongly influenced by the collision parameter p and the gravity parameter h . In the mass flow silo, it is also dependent upon the wall roughness parameter b (more particles are stopped at the wall region with increasing wall roughness). With the decreasing collision parameter p , the height of dilatant zones increases significantly (the material behaves more like gas). If the gravity parameter h is great, the dilatant zones become narrower and more concentrated in the middle region of the silo due to more empty sites appearing in the middle of the silo than in the region close to its walls. For smaller wall roughness parameters b , the dilatant zones are more horizontal (in the mass flow silo) since the presence of walls disturbs less flow. The outflow rate obviously increases with the increasing gravity parameter g . With an increase of the gravity parameter g , the effect of other factors becomes insignificant. The inclination of the upper free surface of the material decreases with decreasing parameters p, q and h (the material then behaves more like gas). The non-uniformity of the material's density increases with increasing parameter g and decreasing parameters h, k, p and q .

Although the calculated flow patterns in silos are in satisfactory agreement with the experimental ones [95] (especially for the mass flow silo), the calculated shapes of the propagating dilatant zones in the flowing material differ from the experimental ones due to pure shearing of the granular material having been neglected. In the experiment with a mass flow silo [105, 106], a symmetrical pair of curvilinear dilatant rupture zones was created in the neighborhood of the outlet. The zones propagated upwards, crossed each other around the symmetry of the silo, reached the walls and were subsequently reflected from them. This process was repeated until the zones reached the free boundary in the converging hopper or

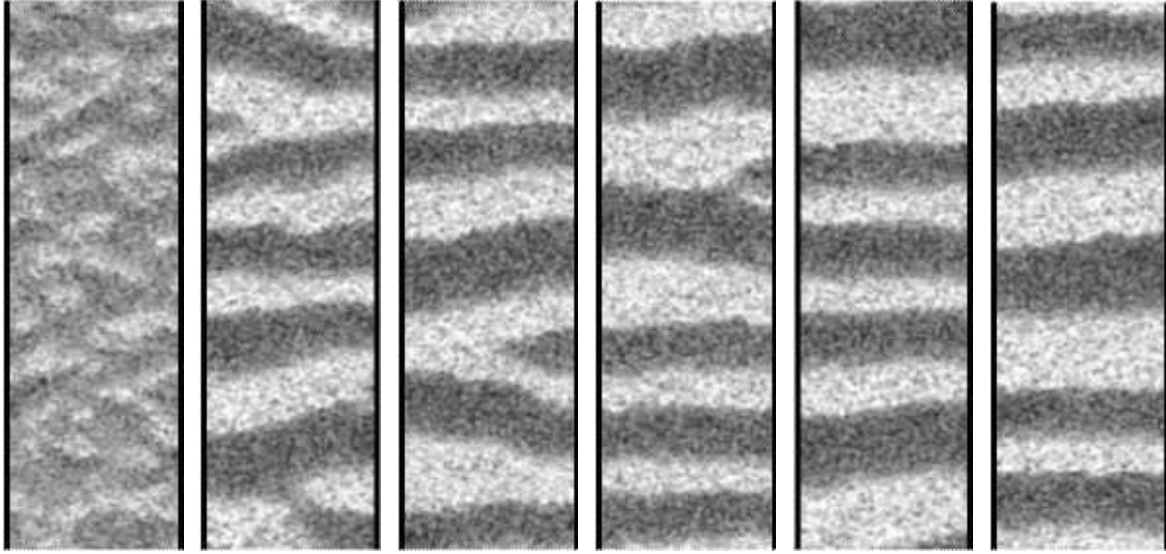


Figure A.42: Evolution of density waves during granular flow in a vertical channel

the transition zone in the parallel–converging silo. In the funnel flow silo [4,5], the curvilinear dilatant zones in the material core were symmetric about a vertical mid–line. Some of them crossed each other.

The 2D-results of simulations of a granular pile including 15000 cells (particles) are demonstrated in Fig. A.41. The material parameters were assumed as in Fig. A.27. During the simulation, the particles were slowly added at a fixed rate from the top of the system to the bottom. Gravity moved these particles down until they collided with a rigid wall at the bottom. The profile of the created heap is similar to a cone with a well defined angle of repose equal to 30° (determined by the geometry of the underlying lattice). As compared to a perfect cone, it shows the presence of small kinks along the surface and a slightly curved tail (with a slope inclination smaller than 30°) what is in accordance with experimental results [2].

The time evolution of the density in a vertical channel during 2D-results of simulations with 35000 cells is demonstrated in Fig. A.42. Periodic boundary conditions were used in the vertical direction. The darker regions are associated with the higher densities, and the lighter regions are associated with the lower densities of the silo fill. Initially, the density is overall non-homogeneous. During continuous flow, gradually layers of high and low density are being formed out of the system. The layers create permanent density waves. The shape of density waves is in accordance with experiments [37]. The density waves do not appear during granular flow if the collisions parameters and the wall roughness parameter are equal to zero.

A.6 Conclusions

Although cellular automata are purely kinematic models and simplify the behavior of granular materials, they can capture realistically flow patterns of granulates in silos on the basis of back analysis of laboratory experiments.

The improved cellular automaton, has been able to describe the propagation of dilatant zones in granular material during silo flow. Their shape was affected by gravity, collision and friction parameters. The flow dynamics was taken into account, and collision energy

was conserved. However, the results were different than that obtained in experiments.

The simplified cellular automata model is purely kinematic wherein flow dynamics is not taken into account. Although it is a gross oversimplification of real granular materials, it is capable of describing consistently the flow patterns of granulates in silos with and without inserts. The main parameters governing the motion of granular particles in silos in the simplified cellular automaton discussed here have been migration rules, transition probabilities, grid type and number of cells. The migration rules assuming a decrease of transition probability values towards the mid-point of cells above the void have been able to capture the angle of repose of the bulk solid. This automaton can be used to approximately and quickly determine the flow patterns of non-cohesive bulk materials in silos in the phase of the silo design.

Appendix B

Discrete Element Method in the simulation framework

The author of this thesis in cooperation with University Joseph–Fourier in Grenoble¹ has took part in a larger project which aims at developing a versatile framework for numerical simulations.

B.1 Introduction

When developing the simulation software the scientists have to often focus on marginal problems (not related to their scientific work), like: program interface, input/output of data, geometry handling, mesh generation or visualization of results. One solution is to use the existing common scientific frameworks with own calculation algorithms (eg. Abaqus, Dyna, Adina, Pfc3d). However those frameworks rarely give possibility of modelling together FEM and DEM. In such case the user has to overcome the obstacles presented by the flawed software [19]. A common solution here is to write the own software to perform simulations.

The proposed solution is to provide a stable and uniform environment for scientists to implement computational algorithms. Therefore they can focus on the work instead of reinventing the wheel of input/output or display. The YADE framework is divided into several layers shown on Fig. B.1. Each layer can depend on layers below it. Libraries in lower layers are not related to simulation itself, and can be utilized by other software.

The YADE common layer on Fig. B.1 contains components usually used by simulation: concept of force, momentum or displacement, Newton’s law, time integration algorithms (leapfrog [63], Newmark, Runge–Kutta 4, etc.), damping methods (Cundall non viscous damping), collision detection algorithms (eg. Sweep and Prune [35]), boundary conditions (imposing translation, applying gravity, etc.), classes that store information about bodies or interactions, and common OpenGL methods for drawing popular geometries.

Specialized layer is based on the common layer. Many specialized packages can exist: Discrete Element Method, Finite Element Method or Lattice Geometrical Model.

¹Le laboratoire Sols, Solides et Structures; F. Donzé research team

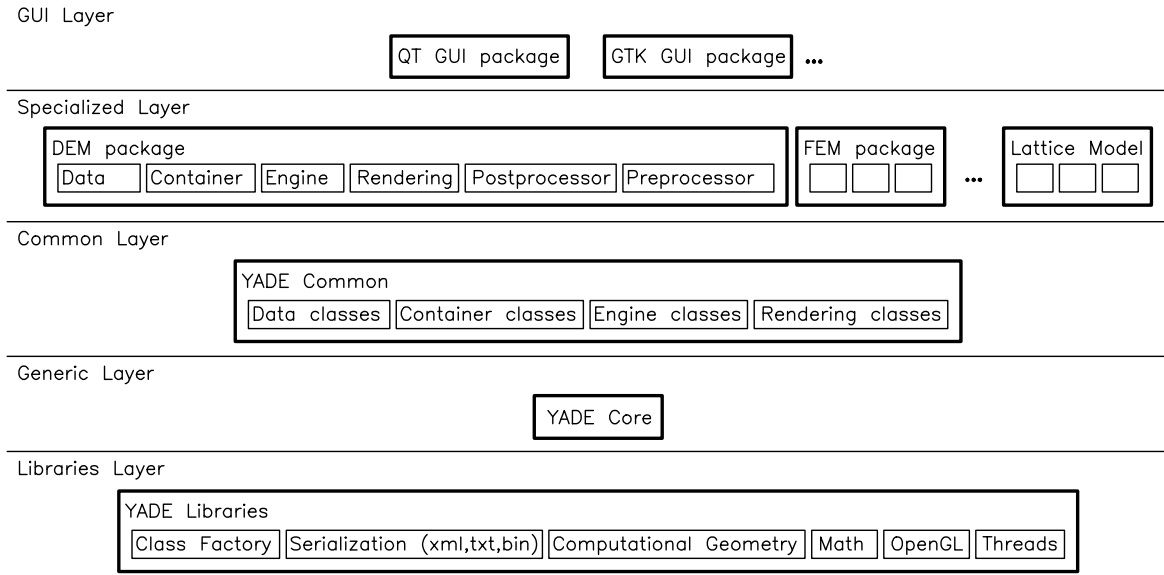


Figure B.1: Layered structure of YADE framework.

B.2 The YADE framework: DEM and other models

The Intention of the YADE design is to allow an easy adding of new simulation models and reuse already defined algorithms. Examples in subsections below describe what had to be implemented in specialized layers to perform simulations with DEM, FEM and Lattice Geometrical Model.

Discrete Element Method

To test the flexibility of YADE framework, the SDEC (Sect. 2.1) algorithms were first implemented. In DEM the contact is described by radiuses of two spheres: r_1 , r_2 , penetration depth d and the normal of the contact plane \vec{n} . To allow interaction between the sphere and a non-spherical object an imaginary mirror sphere of double radius is created (as proposed by Donzé [54]). Following this definition a new class *SpheresContactGeometry* was added (see Fig. B.11). Then two different *EngineUnit*-s with algorithms for building this contact description were added to *MetaEngine*: one to build from contact between two spheres another to build from contact between a sphere and a box (analogous engine units that build bounding volume are shown on Fig. B.15). This contact description can be used only when at least one object in contact is a sphere.

The class describing *InteractionPhysics* was added with the name *ElasticContactPhysics* which contains information about the contact: normal stiffness, shear stiffness and friction angle (see Fig. B.11). When contact occurs, this information is calculated by dedicated *EngineUnit* that calculates macro-micro relationship according to formula proposed by Donzé in [54].

The criterion for destroying interactions in DEM is physical, thus *persistentInteractions* were used, and a class *CoulombMohrCriterion* was added to control when the contact is lost (see Sect. 2.1, Fig. 2.4). Adding other criterions as plug-ins poses no difficulty for the user. To solve interactions a class *ElasticContactLaw* deriving from *Engine* was added, which contains DEM formulation inside (see Sect. 2.1, Eq. 2.2).

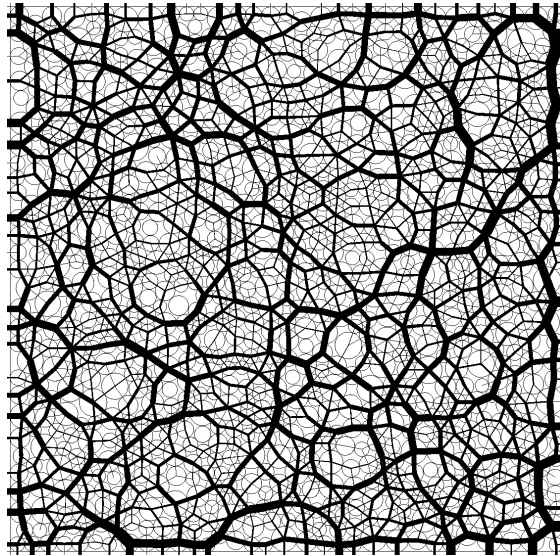


Figure B.2: Distribution of forces in a two dimensional DEM sample subject to biaxial compression, results obtained in YADE's DEM implementation.

Finally a simulation loop (see *simulationLoop* on Fig. B.13) for DEM calculation was built:

- Calculating time step with elastic criterion,
- building a bounding volume (the *BoundingVolume* is shown on Fig. B.8) using a *MetaEngine* as shown on Fig. B.15,
- performing collision detection with Sweep and Prune collider using the previously calculated bounding volumes,
- building interaction geometry and physics (the classes *SpheresContactGeometry* and *ElasticContactPhysics* shown on Fig. B.11 are built using a 2D *MetaEngine* analogously to Fig. B.15),
- solving interactions with DEM formulation (the class *ElasticContactLaw* contains the Eq. 2.2 from Sect. 2.1),
- applying the calculated response to the bodies by first calculating their new acceleration (using classes *NewtonsForceLaw* and *NewtonsMomentumLaw* which contain Eq. 2.1 for acceleration and angular acceleration respectively),
- and performing the time integration of bodies according to their new acceleration (eg. using a leapfrog or Runge–Kutta 4 integration method).

This loop repeats until the calculation is terminated, and it directly implements DEM as described in Sect. 2.1.

Figures B.2–B.4 show several example simulations done using the YADE framework. The biaxial compression in two dimensions shown on Fig. B.2 is in good agreement with results by Cundall [38, 49]. Figure B.3 shows a similar experiment performed in three dimensions. The „force paths” are clearly visible. Figure B.4 shows a concrete beam subject to three point bending, the concrete aggregates are using a cohesive law (see Fig. 2.4, Sect. 2.1) to transfer tensile force.

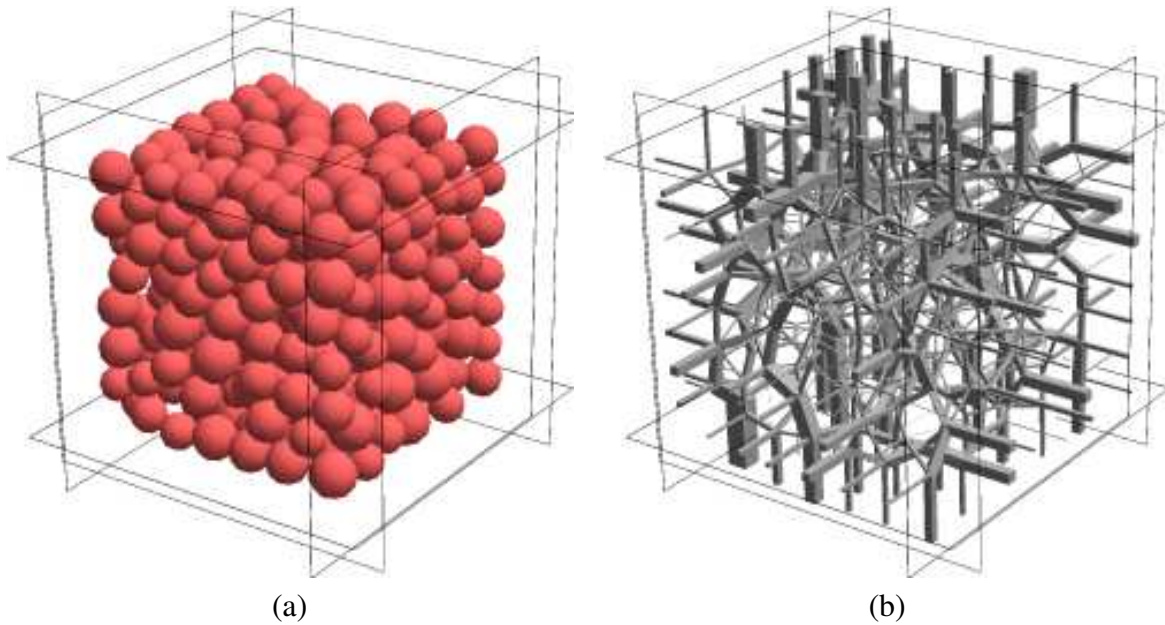


Figure B.3: A three dimensional DEM sample subject to triaxial compression, results obtained in YADE's DEM implementation: (a) view on aggregates; (b) distribution of forces

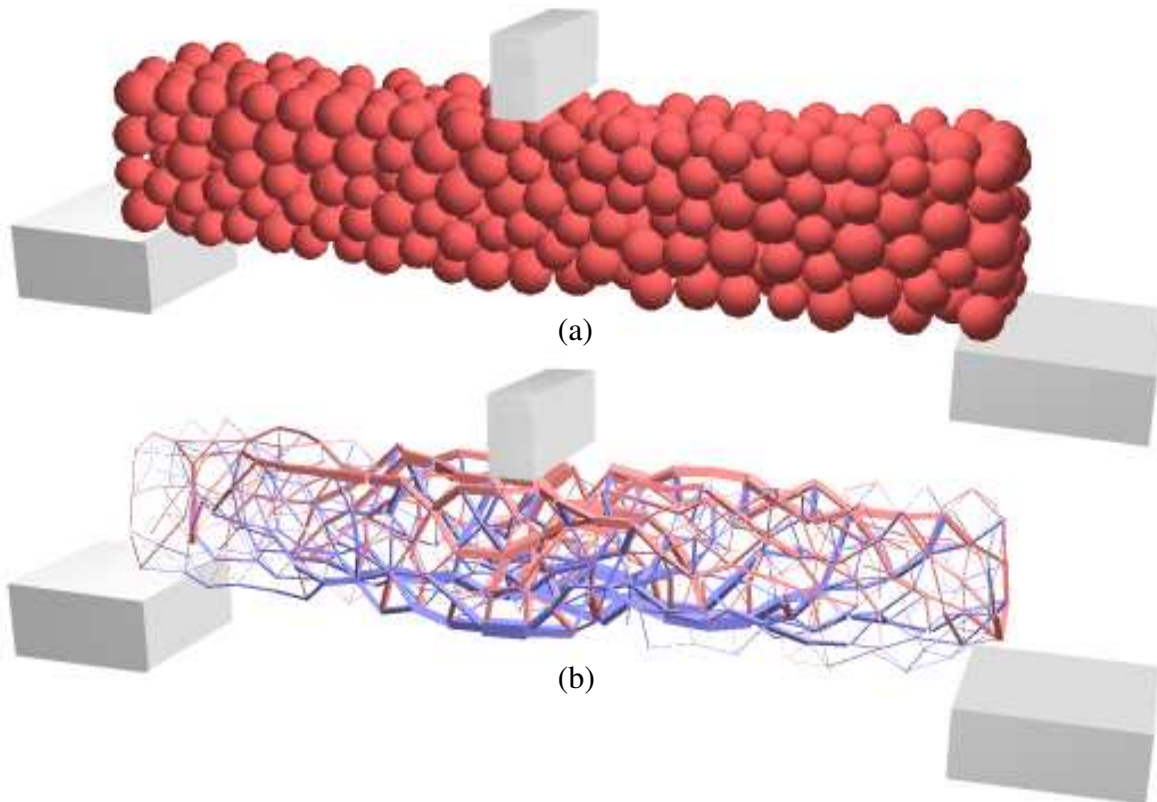


Figure B.4: A DEM specimen of concrete (aggregates connected using cohesive law, Fig. 2.4 from Sect. 2.1), subject to three point bending: (a) specimen's configuration; (b) forces in bonds between aggregates (red: compression, blue: tension)

Finite Element Method

To test capabilities of the YADE framework with respect to the Finite Element Method, a simple tetrahedron FEM element was implemented. Its results were compared with the same problem solved in matlab to verify the model. Further step is to couple FEM with DEM.

In current implementation one FEM entity is one *MetaBody*. Each node is a *Body* and each FEM element is a multi-body *Interaction*. In the case of tetrahedron element it is a four-body interaction, rather than an element by itself. One might argue that FEM tetrahedron is really an interaction, because it contains a K_e stiffness matrix just like a spring interaction or elastic contact interaction contain a stiffness property (compare with Fig. B.11). Majority however believes that it's not an interaction but an element. The author assertively believes that there is no definite answer in this subject aka chicken and egg problem. Consequently this interesting turn of events means that in future YADE refactoring a new kind of object will be added, which will be an element composed of several bodies. In this way a composite element (like four body FEM tetrahedron) will feel more naturally fitting into the framework.

A class *FEMTetrahedronInteraction* (compare with Fig. B.11) was added as an interaction between four nodes. An *EngineUnit* was added to calculate K_e stiffness matrix of tetrahedron. A *FEMLaw* engine was written with classical explicit FEM formulation.

Finally a simulation loop (see *simulationLoop* on Fig. B.13) for FEM calculation was built:

- Building a bounding volume as shown on Fig. B.15 for the whole FEM entity,
- calculation of the response on individual nodes using an explicit FEM formulation by class *FEMLaw* (produces a *PhysicalAction* with force for each *Body*, but does not apply the force yet),
- add gravity force to each *PhysicalAction* using a gravity engine,
- applying boundary conditions like translating selected nodes by a certain distance according to some velocity, or adding some external force to their *PhysicalAction*,
- using calculated forces to calculate acceleration, just like in the DEM simulation loop,
- and performing the time integration of bodies according to their new acceleration (eg. using a leapfrog or Runge–Kutta 4 integration method).

The loop repeats over and over. It should be noted that modular architecture of YADE framework allows to use the same simulation steps in both DEM and FEM and to easily replace one with another version that does the same in different manner, eg. replacing Runge–Kutta 4 integration with Newmark method.

Figure B.5 shows an example simulation performed with FEM implemented in YADE, a three dimensional beam discretized using tetrahedron element, subject to uniaxial extension.

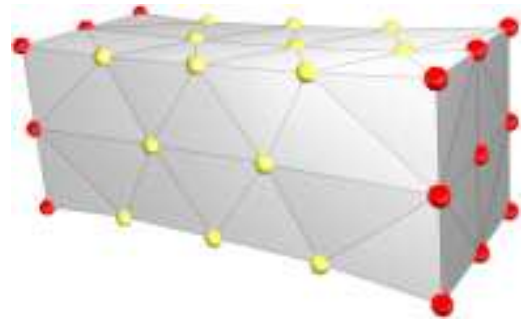


Figure B.5: A FEM beam, using a tetrahedron as an element, subject to uniaxial extension as calculated in YADE

Lattice Geometrical Model

Another model implemented in YADE was obviously the lattice model (Chapter 4). Just like in FEM an ambiguity arises whether a beam should be an interaction between two nodes, or a composite two-noded element. However since the angular springs are interactions between two beams the solution to refactor YADE by adding a new composite element type becomes even more appealing. Currently the beam is a *Body* in YADE, and it stores among other properties the *id* numbers of its two nodes. This means that in a single *BodyContainer* both beams and nodes are stored, to distinguish them a *group* number is used (a cheaper solution than doing `dynamic_cast` each time).

The necessary data classes were added for a node, a beam and an angular spring between two beams to store their respective information. Then a new engine *LatticeLaw* was written to perform the computations as described in Chapter 4. Finally a simulation loop for the calculation was built:

- Building a bounding volume for the whole lattice specimen,
- applying some boundary conditions like displacement of selected nodes (using a *Deux-ExMachina* displacement engine),
- calculating interactions between beams, namely the current angle and torsion of angular springs that connect them,
- solving interactions of lattice beams (the class *LatticeLaw*), by calculating a *PhysicalAction* displacement for each node in the model,
- removing the beams according to fracture criterion (using a *StandAloneEngine*),
- and finally applying the calculated response (stored in *PhysicalAction*) displacement to the nodes (notably skipping the Newton's Law and time integration),

This loop exposes similarities with previous two loops for DEM and FEM with the most important difference of not using a time integration. This is due to the fact that lattice model being of geometrical type has quasi static nature and does not depend on time.

Figure B.6 shows an example simulation with a lattice model, a three dimensional concrete specimen subject to uniaxial extension. The damage area inside the specimen is denoted with red surface. For further details about the lattice model and more results see the Chapters 4–6. Figure B.7 shows an overall view on YADE's graphical interface with a three dimensional specimen, for more details see Chapter 5.

The planned multi-scale modelling using lattice coupled with FEM (or DEM) should be relatively easy due to fact that all of the necessary models are already implemented in a single computational framework.

B.3 Software design applied to numerical simulations

Too much generality leads to abstractions where everything tends to fit, and no clear path is shown for the scientist to follow. On the contrary too strong concepts that structure the code can limit the diversity of simulation and framework flexibility. A good balance must be found between abstractions and concrete simulation concepts. To achieve this the data structures were separated from algorithms operating on them.

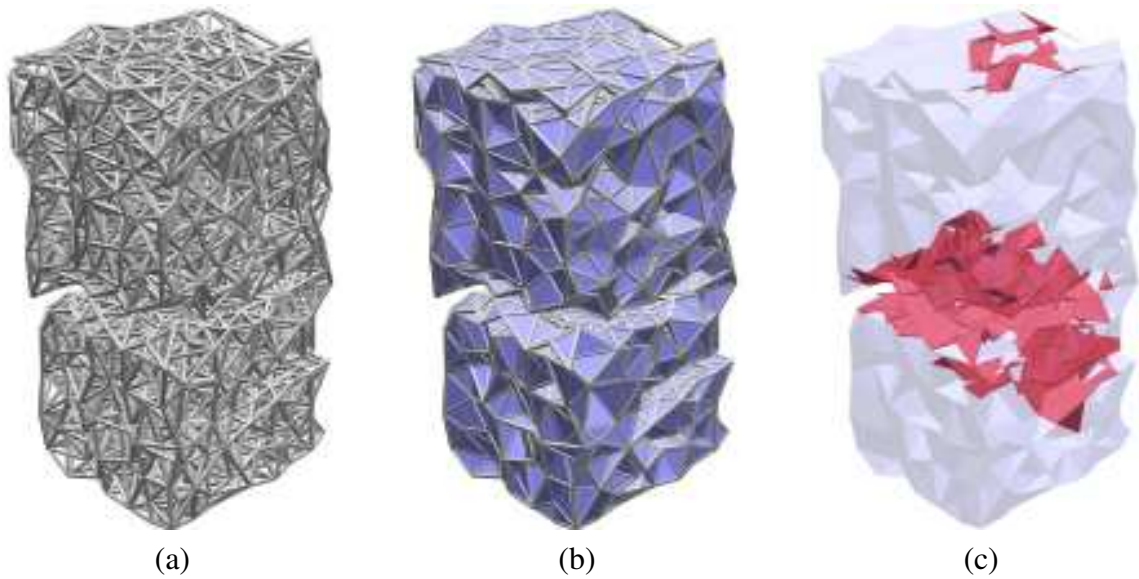


Figure B.6: A lattice beam model specimen subject to uniaxial extension as calculated in YADE: (a) the beams that build the specimen; (b) view on specimen's surface with visible fracture surfaces inside; (c) the damaged areas inside the specimen

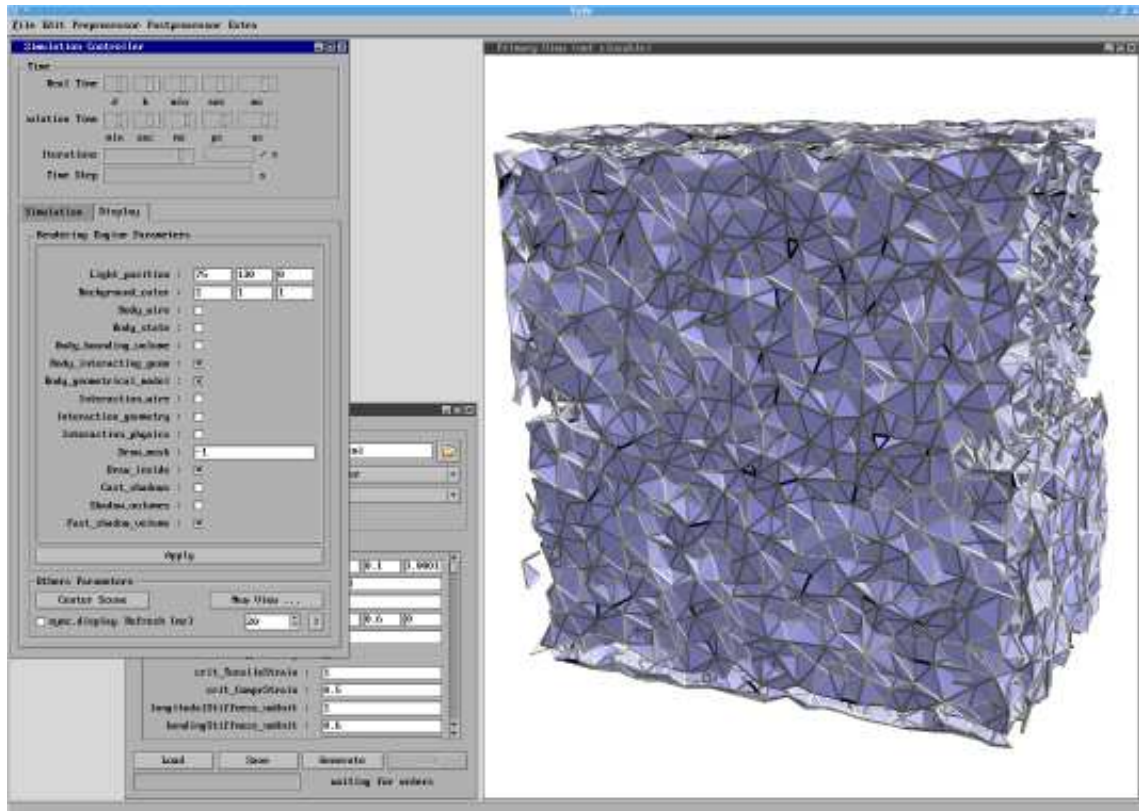


Figure B.7: A view on YADE's main graphical interface (with a 3D simulation)

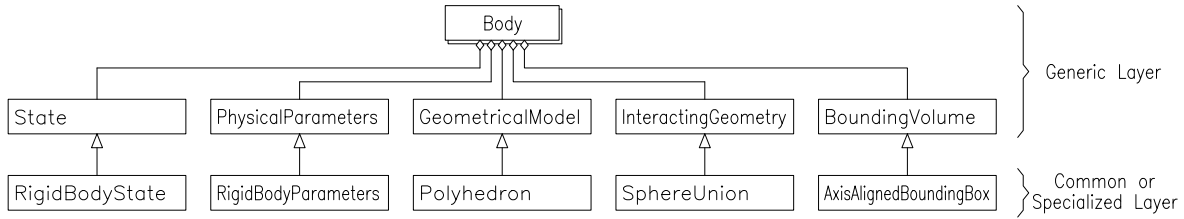
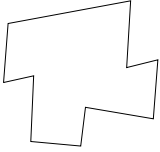
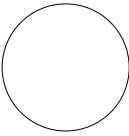
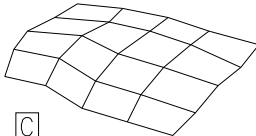
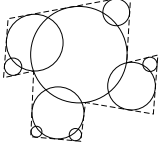
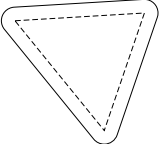
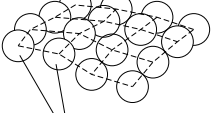
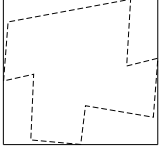
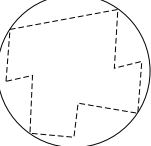
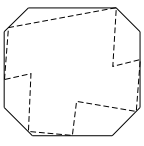


Figure B.8: Class *Body*, and abstract interface classes contained by it, with examples of concrete derived classes (from common or specialized layer).

State	ParticleState – position – velocity [C]	RigidBodyState – position – orientation – velocity – angular velocity [C]	LatticeBeamState – position – orientation – velocity – angular velocity – length [S]
Physical Parameters	ParticleParameters – mass [C]	RigidBodyParameters – mass – inertia [C]	ElasticBodyParameters – mass – inertia – young modulus – poisson's ratio [S]
Geometrical Model	Polyhedron  [C]	Sphere  [C]	Mesh2D  [C]
Interacting Geometry	SphereUnion  [S]	PolyhedralSweptSphere  [S]	MetalInteractingGeometry  [S]
Bounding Volume	AxisAlignedBoundingBox  [C]	BoundingSphere  [C]	KDop  [C]

[C] – Common Layer [S] – Specialized Layer

Figure B.9: Examples of concrete classes contained by *Body*

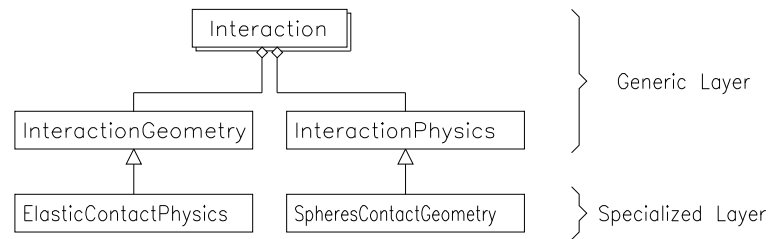
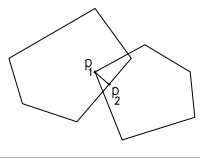
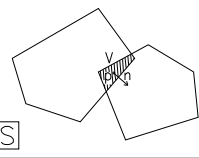
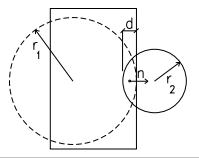





Figure B.10: Class *Interaction*, and abstract interface classes contained by it, with examples of concrete derived classes (from specialized layer).

Interaction Geometry	ClosestFeatures	VolumetricContact	SpheresContactGeometry
			
Interaction Physics	SpringPhysics	FEMInteraction	ElasticContactPhysics
	– spring stiffness 	– FEM element stiffness (K_e matrix) 	– normal stiffness – shear stiffness – friction angle 



 – Common Layer  – Specialized Layer

Figure B.11: Examples of concrete classes contained by *Interaction*

Data classes

Four data classes are distinguished: *Body*, *Interaction*, *PhysicalAction* and *MetaBody*. Objects of those classes cannot move or interact themselves, as they contain only data. Their movement, interaction, etc. are handled by *Engine* classes.

The class *Body* contains five abstract class interfaces (Figs. B.8 and B.9). The class *State* describes body's state which evolves during simulation. The class *PhysicalParameters* describe various body's material properties. The class *GeometricalModel* is the best possible representation of simulated body's geometry. The class *InteractingGeometry* is used to process interactions between bodies. The class *BoundingVolume* is used by fast interaction detection algorithms (eg. SAP [35] or grid based collision detection).

The class *Interaction* contains two class interfaces (Figs. B.10 and B.11). The class *InteractionGeometry* contains geometrical information describing the interaction. The class *InteractionPhysics* describes physical means of interaction (contact shearing angle, interaction stiffness, etc.).

The class *PhysicalAction* (Fig. B.12) conveys modification of body's state. Example physical actions are: force, momentum, acceleration, impulse, displacement, velocity.

The class *MetaBody* is the simulated world, it contains inside bodies, interactions, physical actions and engines (Fig. B.13). The *MetaBody's Engine* classes do the actual processing of its *data* (see Sect. B.3).

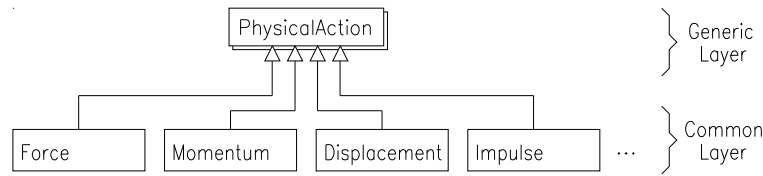


Figure B.12: Abstract class *PhysicalAction*, with examples of concrete derived classes.

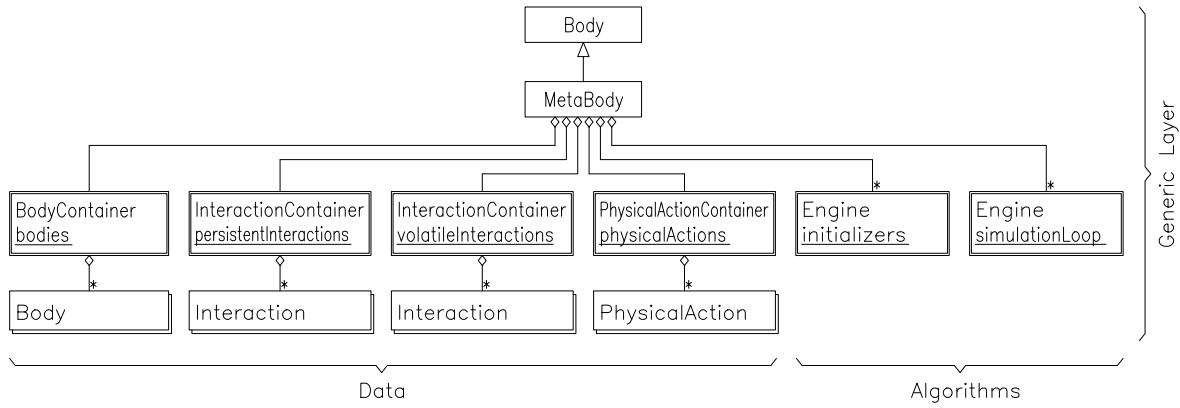


Figure B.13: Class *MetaBody* and container classes owned by it

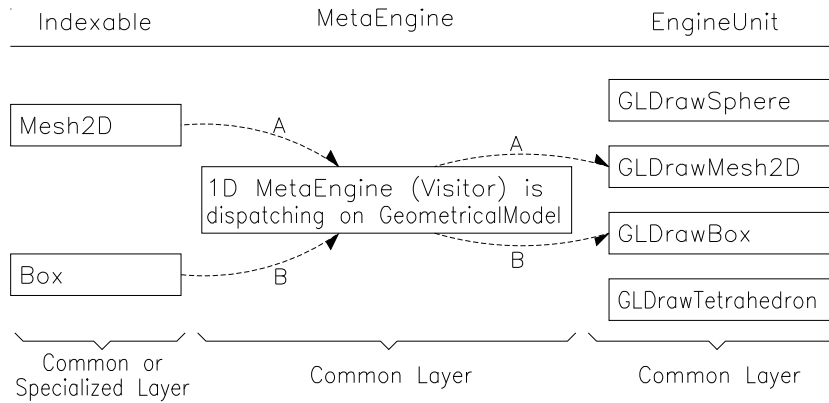


Figure B.14: One dimensional *MetaEngine*.

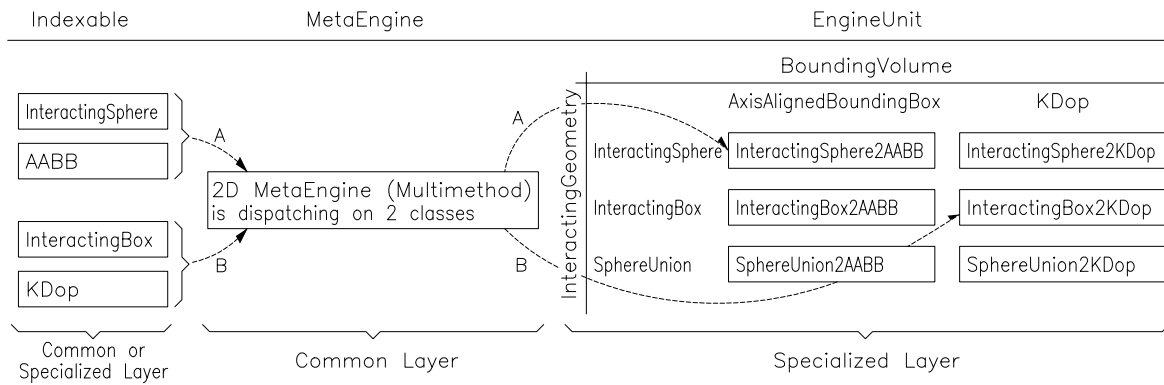


Figure B.15: Two dimensional *MetaEngine*.

Engine classes

Every operation concerning data is performed by dedicated to the task *Engine*. The class *StandAloneEngine* performs tasks that depend on any part of data. The class *DeuxEx-Machina* performs tasks that depend on conditions from outside.

Multimethod [1] dispatchers (called *MetaEngine*) perform tasks which are specific to some kind (or kinds) of data. Each *MetaEngine* contains *EngineUnits* that effectively become virtual methods for that specific *data*. Figure B.14 shows an example of one dimensional multimethod. *Indexable* classes *Mesh2D* and *Box* (which inherit from *GeometricalModel* class) are drawn on the screen with OpenGL using the corresponding classes *GLDrawMesh2D* and *GLDrawBox*. Figure B.15 shows how an *EngineUnit* is called depending on two classes: *InteractingGeometry* and *BoundingVolume*. Algorithm constructing axis aligned bounding box is different for box and for a sphere. Similarly *MetaEngine* can handle collision of bodies. Different algorithms are called when *Sphere* collides with *Box* or with another *Sphere*.

Appendix C

Quaternion rotations

This appendix briefly explains the basics of quaternion rotations, along with some of the quaternion algebra. It serves as a short reference to Chapter 4.

There are several different ways to represent a 3D orientation, such as Euler angle and the rotation matrix. Those methods however suffer the problem with singularities and numerical instabilities near certain orientation values, such as a north and south pole on a sphere. This drawback is not present when quaternions are used to represent a 3D orientation. As an additional bonus, the computational cost of performing the rotation composition is smaller than a similar operation performed with a 3D rotation matrix.

In mathematics, quaternions are a non-commutative extension of complex numbers. They were first described by the Irish mathematician Sir William Rowan Hamilton in 1843, and applied to mechanics in a three-dimensional space. At first, quaternions were regarded as pathological, because they disobeyed the commutative law of multiplication $ab = ba$. However currently they are becoming widely popular for calculations involving three-dimensional rotations. The quaternion algebra holds a special place in analysis since it is one of only three finite-dimensional division rings containing the real numbers as a subring. The quaternion space is often denoted by double-barred letter \mathbb{H} after the initial of its inventor. In this thesis, all quaternions are marked using a math ring symbol above, thus: $\hat{q} \in \mathbb{H}$.

Quaternions' non-commutative property can be explained by the fact that three dimensional rotations are also not commutative. A simple exercise of applying two rotations to an asymmetrical object (eg. this thesis) can explain it. First, rotate this thesis 90 degrees clockwise around the z axis. Next, rotate it 180 degrees clockwise around the x axis. Then restore the original orientation, so that this text is again readable, and apply those rotations in the opposite order. This shows that, in general, the composition of two different rotations around two distinct spatial axes will not commute.

C.1 Definition

Quaternions are a generalization of complex numbers, obtained by adding the elements i , j , and k to the real numbers, where i , j , and k satisfy following equation:

$$i^2 = j^2 = k^2 = ijk = -1. \quad (\text{C.1})$$

Every quaternion is a real linear combination of the basis 1, i , j , and k . Thus every quaternion is uniquely expressible in the form $a + bi + cj + dk$ where a , b , c , and d are real numbers. The

equation $ijk = -1$ can be transformed by first right-multiplying both sides by k , then left-multiplying both sides by i then, left-multiplying both sides by j , which eventually leads to:

$$\begin{aligned} ij &= -ji = k, \\ jk &= -kj = i, \\ ki &= -ik = j. \end{aligned} \tag{C.2}$$

For the purpose of dealing with three-dimensional rotations it is also necessary that the quaternion is a **unit quaternion**, thus its components must satisfy the equation

$$a^2 + b^2 + c^2 + d^2 = 1. \tag{C.3}$$

A quaternion which does not satisfy this condition, although it is still a quaternion, is not useful for performing three-dimensional rotations. Due to this condition, to avoid numerical build-up errors, it is a common practice to re-normalize quaternions at frequent intervals, usually at each calculation step. In this thesis this was not necessary, because each iteration started using a new identity quaternion.

C.2 Properties

The base algebraic operations '+', '-' and '×' apply to quaternions, respecting Eq. C.2. The division is performed by multiplying by the inverse quaternion.

The quaternion multiplication is associative:

$$\hat{q}_1 \hat{q}_2 \hat{q}_3 = (\hat{q}_1 \hat{q}_2) \hat{q}_3 = \hat{q}_1 (\hat{q}_2 \hat{q}_3). \tag{C.4}$$

The quaternion multiplication is not commutative:

$$\hat{q}_1 \hat{q}_2 \neq \hat{q}_2 \hat{q}_1. \tag{C.5}$$

The **conjugate quaternion** of a quaternion $\hat{q} = a + bi + cj + dk$ is similarly as in algebra of complex numbers defined as:

$$\hat{q}^* = a - bi - cj - dk. \tag{C.6}$$

The **absolute value** of a quaternion is defined as:

$$|\hat{q}| = \sqrt{\hat{q} \hat{q}^*} = \sqrt{a^2 + b^2 + c^2 + d^2}. \tag{C.7}$$

An **identity quaternion** \hat{E} is a quaternion that multiplied by any other quaternion gives the quaternion unchanged

$$\hat{E} = 1 + 0i + 0j + 0k. \tag{C.8}$$

An **inverse of a quaternion** $\hat{q} = a + bi + cj + dk$ is similarly as in algebra of complex numbers defined as:

$$\hat{q}^{-1} = \frac{\hat{q}^*}{|\hat{q}|^2} = \frac{a - bi - cj - dk}{a^2 + b^2 + c^2 + d^2}. \tag{C.9}$$

This can be verified by an algebraic multiplication conforming to the rules in Eq. C.2:

$$\begin{aligned}
\hat{q}^{-1} \hat{q} &= (a - bi - cj - dk)(a + bi + cj + dk) |\hat{q}|^{-2} \\
&= (a^2 + abi + acj + adk \\
&\quad - bia - b^2 i^2 - bicj - bidk \\
&\quad - cja - cjb i - c^2 j^2 - cjdk \\
&\quad - dka - dkbi - dkcj - d^2 k^2) |\hat{q}|^{-2} \\
&= (a^2 + abi + acj + adk \\
&\quad - abi + b^2 - bck + bdj \\
&\quad - acj + bck + c^2 - cdi \\
&\quad - adk - bdj + cdi + d^2) |\hat{q}|^{-2} \\
&= (a^2 + b^2 + c^2 + d^2) |\hat{q}|^{-2} \\
&= 1 + 0i + 0j + 0k = \hat{E}.
\end{aligned} \tag{C.10}$$

Also with the unit quaternion restriction of Eq. C.3, the conjugate quaternion equals the inverse quaternion:

$$\hat{q}^* = \hat{q}^{-1}. \tag{C.11}$$

C.3 Operations concerning rotation

The quaternion addition and subtraction has no meaning in describing rotation, because such resulting quaternion would violate the unit quaternion requirement Eq. C.3.

The composition of two rotations is performed by multiplication of two quaternions \hat{q}_1 and \hat{q}_2 :

$$\hat{q}_3 = \hat{q}_1 \hat{q}_2. \tag{C.12}$$

The rotation difference $\Delta \hat{q}$ between two distinct rotations \hat{q}_1 and \hat{q}_2 is calculated as follows:

$$\Delta \hat{q} = \hat{q}_1 \hat{q}_2^{-1}. \tag{C.13}$$

To perform a rotation of 3D vector $\vec{v} = \{x, y, z\}$ by a quaternion, the vector is introduced as a quaternion with its imaginary coefficients equal respectively to x , y , and z :

$$\hat{v} = 0 + xi + yj + zk. \tag{C.14}$$

Then, the quaternion \hat{v} (which represents a vector) is rotated by performing a two-side multiplication, producing quaternion \hat{v}_{rot} (which represents a rotated vector):

$$\hat{v}_{rot} = \hat{q} \hat{v} \hat{q}^{-1}. \tag{C.15}$$

It can be verified that the resulting quaternion \hat{v}_{rot} has a real value equal to zero, because the $\hat{q} \hat{q}^{-1}$ multiplication cancels it. The three imaginary coefficients of \hat{v}_{rot} are the coordinates of the rotated vector $\vec{v}_{rot} = \{x, y, z\}$. For convenience, the conversion $\vec{v} \leftrightarrow \hat{v}$ is implied, which allows to write this formula in simplified form, which is not strictly correct:

$$\vec{v}_{rot} = \hat{q} \vec{v} \hat{q}^{-1}. \tag{C.16}$$

To reduce computational effort, this operation is usually done by converting first a quaternion into a rotation matrix \mathbf{R} (see Sect. C.5.2), then by performing multiplication $\vec{v}_{rot} = \mathbf{R} \vec{v}$.

C.4 Matrix representation

There are two ways to represent quaternions as matrices, in such a way that the quaternion addition and multiplication correspond to the matrix addition and matrix multiplication. First way is to represent it as a 4×4 matrix of real numbers \mathbf{Q}_r :

$$\mathbf{Q}_r = \begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix}. \quad (\text{C.17})$$

In this representation, the conjugate of a quaternion (Eq. C.6) corresponds to the transpose of the matrix. The fourth power of the absolute value of a quaternion is the determinant of the corresponding matrix.

The second way is to represent as a 2×2 matrix of complex numbers \mathbf{Q}_c :

$$\mathbf{Q}_c = \begin{bmatrix} (a+bi) & (c+di) \\ (-c+di) & (a-bi) \end{bmatrix}. \quad (\text{C.18})$$

This representation due to its isomorphism properties is especially useful in quantum mechanics for dealing with particle spins.

C.5 Conversions

C.5.1 Euler axis angle

The **Euler axis angle** representation of rotations uses a unit axis of rotation $\vec{v} = \{x, y, z\}$ (where $x^2 + y^2 + z^2 = 1$) and the angle of rotation α around this axis. The representation of a quaternion $\hat{q} = a + bi + cj + dk$ can be built using the formula:

$$\hat{q}(\vec{v}, \alpha) = \begin{cases} a = \cos(\alpha/2) \\ b = \sin(\alpha/2)x \\ c = \sin(\alpha/2)y \\ d = \sin(\alpha/2)z \end{cases}. \quad (\text{C.19})$$

Note that such quaternion satisfies a unit quaternion requirement Eq. C.3. If, during calculations the need arises for building a quaternion from rotations, this method was always used, because it is most efficient time-wise.

The construction of an Euler axis angle from a quaternion is achieved using the following formula:

$$\begin{aligned} \alpha &= 2 \arccos(a), \\ x &= \frac{b}{\sin(\alpha/2)}, \\ y &= \frac{c}{\sin(\alpha/2)}, \\ z &= \frac{d}{\sin(\alpha/2)}. \end{aligned} \quad (\text{C.20})$$

If the rotation angle $\alpha = 0$, then axis of the rotation can be anything since no rotation occurs. In this case to avoid the division by zero, the zeros are assigned to x , y and z .

C.5.2 Rotation matrix

The **rotation matrix**, also called **direction cosine matrix** is a 3×3 matrix that contains three unit vectors which are a set of new coordinate axes expressed in terms of original (non-rotated) coordinate axes. When computing a quaternion from the rotation matrix, there is a sign ambiguity since quaternions \hat{q} and $-\hat{q}$ represent the same rotation. One way of computing the quaternion from the rotation matrix $\mathbf{R} = [\vec{e}_1, \vec{e}_2, \vec{e}_3]$ is as follows:

$$\hat{q}(\mathbf{R}) = \begin{cases} a = \pm \frac{1}{2} \sqrt{1 + \mathbf{R}_{11} + \mathbf{R}_{22} + \mathbf{R}_{33}}, \\ b = \frac{1}{4a} (\mathbf{R}_{23} - \mathbf{R}_{32}), \\ c = \frac{1}{4a} (\mathbf{R}_{31} - \mathbf{R}_{13}), \\ d = \frac{1}{4a} (\mathbf{R}_{12} - \mathbf{R}_{21}). \end{cases} \quad (\text{C.21})$$

There are three other mathematically equivalent ways to compute \hat{q} . A numerical inaccuracy can be reduced by avoiding situations in which the denominator (in this case a) is close to zero. One of the other three methods looks as follows:

$$\hat{q}(\mathbf{R}) = \begin{cases} b = \pm \frac{1}{2} \sqrt{1 + \mathbf{R}_{11} - \mathbf{R}_{22} - \mathbf{R}_{33}}, \\ c = \frac{1}{4b} (\mathbf{R}_{12} + \mathbf{R}_{21}), \\ d = \frac{1}{4b} (\mathbf{R}_{13} + \mathbf{R}_{31}), \\ a = \frac{1}{4b} (\mathbf{R}_{23} - \mathbf{R}_{32}). \end{cases} \quad (\text{C.22})$$

Shoemake [140, 141] has published an efficient algorithm for a fast conversion of rotation matrix to a quaternion.

The rotation matrix \mathbf{R} can be computed from the quaternion \hat{q} with the formula:

$$\mathbf{R}(\hat{q}) = \begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2ac + 2bd \\ 2ad + 2bc & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2ab + 2cd & a^2 - b^2 - c^2 + d^2 \end{bmatrix}. \quad (\text{C.23})$$

To reduce the number of computer operations, it is possible to take advantage of the fact that \hat{q} is a unit quaternion:

$$\mathbf{R}(\hat{q}) = \begin{bmatrix} 1 - 2(c^2 + d^2) & 2(bc - ad) & 2(ac + bd) \\ 2(ad + bc) & 1 - 2(b^2 + d^2) & 2(cd - ab) \\ 2(bd - ac) & 2(ab + cd) & 1 - 2(b^2 + c^2) \end{bmatrix}. \quad (\text{C.24})$$

C.5.3 Euler angle

The **Euler angle** representation of rotation uses three angles ϕ , θ and ψ respectively named *roll*, *pitch* and *yaw* or *heading*, *attitude* and *bank*. A quaternion $\hat{q} = a + bi + cj + dk$ can be expressed in terms of Euler angles as:

$$\hat{q}(\phi, \theta, \psi) = \begin{cases} a = -\cos((\phi - \psi)/2) \sin(\theta/2), \\ b = \sin((\phi - \psi)/2) \sin(\theta/2), \\ c = -\sin((\phi + \psi)/2) \cos(\theta/2), \\ d = \cos((\phi + \psi)/2) \cos(\theta/2). \end{cases} \quad (\text{C.25})$$

A quaternion constructed in this way will satisfy the unit requirement of Eq. C.3.

The Euler angle representation can be calculated from quaternion as follows:

$$\begin{aligned} \phi &= \arctan\left(\frac{2(ab + cd)}{1 - 2(b^2 + c^2)}\right), \\ \theta &= \arcsin(2(ac - db)), \\ \psi &= \arctan\left(\frac{2(ad + bc)}{1 - 2(c^2 + d^2)}\right). \end{aligned} \quad (\text{C.26})$$

The necessity of special handling of a north/south pole due to arctan and arcsin discontinuities will not be discussed here. The reader is referred to [140] for a detailed discussion on this topic.

C.6 Efficiency

A straightforward implementation of the multiplication in Eq. C.15 requires 56 operations to be performed by a computer (32 multiplications and 24 additions). It turns out that it is more efficient to convert the quaternion first into the rotation matrix and then perform the multiplication. The rotation matrix \mathbf{R} in Eq. C.24 requires 12 multiplications and 12 additions to build. Then multiplying it by a vector \vec{v} (using formula $\vec{v}_{rot} = \mathbf{R}\vec{v}$) to perform rotation requires further 9 multiplications and 6 additions which results in 39 operations total. This is less than original 56 operations. However, it is more costly when compared to storing rotational operations directly in the rotation matrix (and not using quaternions at all) which would save 24 operations to perform the conversion.

The saving factor here is that the composition of rotations using quaternions costs 12 additions and 16 multiplications, giving only 28 operations in total. Whereas composing rotations expressed as rotational matrices is numerically less stable because a numerical precision error may introduce a matrix skew, especially when combining several rotations. Renormalizing the matrix afterwards is far more costly than renormalizing a quaternion. A composition of two rotation matrices costs 18 additions and 27 multiplications, giving totally 45 operations (much more than 28 for quaternions). Such comparison was done by Eberly in [59].

Concluding, the increased speed and reliability of composing two quaternion rotations (Eq. C.12) and calculating the rotation difference (Eq. C.13) justifies the cost (24 operations) of the conversion from the quaternion \hat{q} to the rotation matrix \mathbf{R} . Additionally, quaternions have a smaller memory footprint of only 4 values instead of 9.

Further generalizations are possible, like using a pair of quaternions to represent rotation in a 4D space. An interested reader is referred to literature, like [36, 57–59, 140, 141], for more information.

Bibliography

- [1] A. Alexandrescu. *Modern C++ Design, Generic Programming and Design Patterns Applied*. Addison-Wesley, 2001.
- [2] J. Alonso and H. Herrmann. Shape of the tail of a two-dimensional sandpile. *Physical Review Letters*, 76(26):4911–4914, 1996.
- [3] P. N. Asferg, P. N. Poulsen, and L. O. Nielsen. Modelling of crack propagation in concrete applying the xfem. *Computational Modelling of Concrete Structures, EURO-C (eds.: G. Meschke, R. de Borst, H. Mang and N. Bicanic)*, Taylor and Francis, pages 33–42, 2006.
- [4] G. Baxter and R. Behringer. Cellular automata models for the flow of granular materials. *Physica D*, 51:456–471, 1991.
- [5] G. Baxter, R. Behringer, T. Fagert, and G. Johnson. Pattern formation and time-dependence in flowing sand. *Two Phase Flows and Waves Springer Verlag, New York*, pages 1–29, 1990.
- [6] Z. P. Bazant. Size effect in blunt fracture: concrete, rock, metal. *ASCE Journal of Engineering Mechanics*, 110:518–535, 1984.
- [7] Z. P. Bazant. Mechanics of distributed cracking. *Appl. Mech. Rev.*, 26:675–705, 1986.
- [8] Z. P. Bazant. *Scaling of Structural Strength*. Hermes-Penton, London, 2003.
- [9] Z. P. Bazant and P. D. Bhat. Endochronic theory of inelasticity and failure of concrete. *ASCE Journal of Engineering Mechanics*, 102:701–722, 1976.
- [10] Z. P. Bazant and M. Jirasek. Nonlocal integral formulations of plasticity and damage: survey of progress. *J. Engng. Mech*, 128(11):1119–1149, 2002.
- [11] Z. P. Bazant and B. H. Oh. Crack band theory for fracture of concrete. *Material Structures, RILEM*, 16:155–177, 1983.
- [12] Z. P. Bazant and J. Ozbolt. Non-local microplane model for fracture, damage and size effect in structures. *ASCE Journal of Engineering Mechanics*, 116:2485–2505, 1990.
- [13] Z. P. Bazant and J. Planas. Fracture and size effect in concrete and other quasi-brittle materials. *CRC Press, Boca Raton*, 1998.
- [14] Z. P. Bazant and C. L. Shieh. Endochronic model for nonlinear triaxial behaviour of concrete. *Nucl. Engng. Des*, 47:305–315, 1978.

- [15] Z. P. Bazant, M. R. Tabarra, T. Kazemi, and G. Pijaudier-Cabot. Random particle model for fracture of aggregate or fiber composites. *J. Eng. Mech.*, 116(8):1686–1705, 1990.
- [16] W. J. Beranek and G. J. Hobbelman. Constitutive modelling of structural concrete as an assemblage of spheres. *Computer Modelling of Concrete Structures*, pages 37–51, 1994.
- [17] A. Berg and U. Svensson. Datorsimulering och analys av brottförlopp i en heterogen materialstruktur (simulation of fracture in heterogeneous materials). *Report no. TVSM-5050, Lund Institute of Technology, Sweden*, 1991.
- [18] A. R. Bhandari and J. Inoue. Experimental study of strain rates effects on strain localization characteristics of soft rocks. *Journal of the Mechanics and Physics of Solids*, 45(1):125–140, Feb. 2005.
- [19] J. Bobinski. Implementacja i przykłady zastosowań nieliniowych modeli betonu z nielokalnym osłabieniem. *PhD. thesis, Technical University of Gdansk*, 2006.
- [20] J. Bobinski and J. Tejchman. Numerical simulations of localization of deformation in quasi-brittle materials within non-local softening plasticity. *Computers and Concrete*, 4:433–455, 2004.
- [21] J. Bobinski and J. Tejchman. Modelling of concrete behaviour with a non-local continuum damage approach. *Archives of Hydro-Engineering and Environmental Mechanics*, 52(2):85–102, 2005.
- [22] J. Bobinski and J. Tejchman. Modeling of strain localization in quasi-brittle materials with a coupled elasto-plastic-damage model. *J. Theoretical and Applied Mechanics*, 4:44, 2006.
- [23] J. Bobinski and J. Tejchman. Modelling of size effects in concrete using elasto-plasticity with non-local softening. *Archives of Civil Engineering*, 52(1):7–35, 2006.
- [24] P. L. Bransby, P. M. Blair-Fish, and R. G. James. An investigation of the flow of granular materials. *Powder Technology*, 8:197–206, 1973.
- [25] N. J. Burt and J. W. Dougill. Progressive failure in a model heterogeneous medium. *ASCE Journal of Engineering Mechanics*, 103:365–376, 1977.
- [26] A. Caballero, I. Carol, and C. M. López. 3d fracture analysis of concrete under uniaxial tension and compression solicitation. *VIII International Conference on Computational Plasticity*, 2005.
- [27] A. Caballero, I. Carol, and C. M. López. 3d meso-structural fracture analysis of concrete under uniaxial tension and compression. *Anales de Mecánica de la fractura*, 22:581–586, 2005.
- [28] A. Caballero, I. Carol, and C. M. López. Advances in meso-structural analysis of concrete. *EUROMECH Colloquium 460, Numerical Modelling of Concrete Cracking, Innsbruck, Austria*, 2005.

- [29] A. Caballero, I. Carol, and C. M. López. New results in 3d meso-mechanical analysis of concrete specimens using interface elements. *Computational Modelling of Concrete Structures, EURO-C* (eds.: G. Meschke, R. de Borst, H. Mang and N. Bicanic), Taylor and Francis, pages 43–52, 2006.
- [30] F. Calvetti, G. Viggiani, and C. Tamagnini. Numerical investigation of the incremental non-linearity of granular soils. *Ital Geotech Journal*, 3(3):11–29, 2003.
- [31] H. Caram and D. C. Hong. Random-walk approach to granular flows. *Physical Reviews Letter*, 67:828–831, 1991.
- [32] I. Carol, C. López, and O. Roa. Micromechanical analysis of quasi-brittle materials using fracture based interface elements. *International Journal of Numerical Methods in Engineering*, 52:193–215, 2001.
- [33] J. Chen, H. Yuan, and D. Kalkhof. A nonlocal damage model for elastoplastic materials based on gradient plasticity theory. *Report Nr.01-13, Paul Scherrer Institut*, 1(13):1–130, 2001.
- [34] B. Chopard, B. Droz, and D. M. Cellular automata modeling of physical systems. *Cambridge University Press*, 1998.
- [35] J. D. Cohen, M. C. Lin, D. Manocha, and M. K. Ponamgi. I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments. In *Symposium on Interactive 3D Graphics*, pages 189–196, 218, 1995.
- [36] J. H. Conway and D. A. Smith. *On quaternions and octonions, their geometry, arithmetic and symmetry*. ISBN 1-56881-134-9 A.K. Peters Ltd.
- [37] S. C. Cowin and W. J. Comfort. Gravity-induced density discontinuity waves in sand columns. *Journal of Applied Mechanics*, 49:497–500, 1982.
- [38] P. Cundall and O. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29:47–65, 1979.
- [39] G. Cusatis and Z. Bazant. Size effect on compression fracture of concrete with or without V-notches: a numerical meso-mechanical study. *Computational Modelling of Concrete Structures, EURO-C* (eds.: G. Meschke, R. de Borst, H. Mang and N. Bicanic), Taylor and Francis, pages 71–76, 2006.
- [40] G. Cusatis, Z. P. Bazant, and L. Cedolin. Confinement-shear lattice model for concrete damage in tension and compression: I. theory. *ASCE Journal of Engineering Mechanics*, 129(12):1439–1448, 2003.
- [41] G. Cusatis, Z. P. Bazant, and L. Cedolin. Confinement-shear lattice model for concrete damage in tension and compression: II. computation and validation. *ASCE Journal of Engineering Mechanics*, 129(12):1449–1458, 2003.
- [42] G. Cusatis, Z. P. Bazant, and L. Cedolin. Confinement-shear lattice csl model for fracture propagation in concrete. *Comput. Methods Appl. Mech. Engrg*, 2005.

- [43] G. Cusatis and D. Pelessone. Mesolevel simulation of reinforced concrete structures under impact loadings. *Computational Modelling of Concrete Structures, EURO-C* (eds.: G. Meschke, R. de Borst, H. Mang and N. Bicanic), Taylor and Francis, pages 63–70, 2006.
- [44] J. Cutress and R. Pulfer. X-ray investigations of flowing powders. *Powder Technology*, pages 212–220, 1967.
- [45] G. A. D’Addetta, F. Kun, and E. Ramm. In the application of a discrete model to the fracture process of cohesive granular materials. *Granular Matter*, 4:77–90, 2002.
- [46] G. A. D’Addetta, F. Kun, E. Ramm, and H. Herrmann. From solids to granulates – discrete element simulations of fracture and fragmentation processes in geomaterials. *Lecture Note in Physics*, 568:231–258, 2001.
- [47] R. de Borst, H. B. Muhlhaus, J. Pamin, and L. Sluys. Computational modelling of localization of deformation. *Proc. of the 3rd Int. Conf. Comp. Plasticity, Swansea, Pineridge Press*, pages 483–508, 1992.
- [48] R. de Borst, J. Pamin, and M. Geers. On coupled gradient-dependent plasticity and damage theories with a view to localization analysis. *European Journal of Mechanics A/Solids*, 18(6):939–962, 1999.
- [49] D. J. de Jong and A. Verruijt. Etude photo-élastique d’un empilement de disques. *Cah. Grpe fr. Etud. Rhéol*, 2:73–86, 1969.
- [50] H. Dersch. Testing interpolator quality. 1999. <http://www.path.unimelb.edu.au/%7Edersch/interpolator/interpolator.html>.
- [51] D. Deserable and J. Martinez. Using a cellular automaton for the simulation of flow of granular materials. *Powder and Grains* (ed. Thornton), Balkema, pages 345–350, 1993.
- [52] D. Deserable, S. Masson, and J. Martinez. Influence of exclusion rules on flow pattern in a lattice-grain model. *Kishino* (ed.), *Powders and Grains Swets and Zeitlinger, Lisse*, pages 421–424, 2001.
- [53] M. di Prisco and J. Mazars. Crush-crack – a non-local damage model for concrete. *Mechanics of Cohesive-Frictional Materials*, pages 321–347, 1996.
- [54] F. V. Donze, S. A. Magnier, L. Daudeville, and C. Mariotti. Numerical study of compressive behaviour of concrete at high strain rates. *Journal for Engineering Mechanics*, pages 1154–1163, 1999.
- [55] A. Dragon and Z. Mróz. A continuum model for plastic-brittle behaviour of rock and concrete. *Int. Journ. Eng. Science*, 17:121–137, 1979.
- [56] A. Drescher, T. Cousens, and P. Bransby. Kinematics of the mass flow of granular material through a plane hopper. *Geotechnique*, 28(1):27–42, 1978.
- [57] D. Eberly. Quaternion algebra and calculus. *Magic Software, Inc.*, 1999. <http://www.geometrictools.com>.

- [58] D. Eberly. A linear algebraic approach to quaternions. *Magic Software, Inc.*, 2002. <http://www.geometrictools.com>.
- [59] D. Eberly. Rotation representations and performance issues. *Magic Software, Inc.*, 2002. <http://www.geometrictools.com>.
- [60] S. Eckardt and C. Könke. Simulation of damage in concrete structures using multi-scale models. *Computational Modelling of Concrete Structures, EURO-C (eds.: G. Meschke, R. de Borst, H. Mang and N. Bicanic)*, Taylor and Francis, pages 77–83, 2006.
- [61] O. E. Eiksa, J. Mosby, and E. E. Enstad. Experimental investigations on the use of binserts in order to obtain mass flow in silos. *Proc. Intern. Conference on Silos, Partec95, Nürnberg*, pages 417–426, 1995.
- [62] I. Ferrara and M. di Prisco. Mode i fracture behaviour in concrete: nonlocal damage modeling. *ASCE Journal of Engineering Mechanics*, 127(7):678–692, 2001.
- [63] D. Fincham. Leapfrog rotational algorithm. *Molecular Simulations*, 9:1165, 1992.
- [64] T. J. Fiske, S. B. Railkar, and D. M. Kalyon. Effects of segregation on the packing of spherical and non-spherical particles. *Powder Technology*, 81:57–64, 1994.
- [65] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical Reviews Letter*, 56:1505–1508, 1986.
- [66] M. Geers, T. Peijs, W. Brekelmans, and R. de Borst. Experimental monitoring of strain localization and failure behaviour of composite materials. *Compos. Sci. Technol.*, 56:1283–1290, 1996.
- [67] E. Goles, G. Gonzales, H. Herrmann, and S. Martinez. Simple lattice model with inertia for sand piles. *Granular Matter*, 1:137–140, 1998.
- [68] G. Gutt and P. Haff. An automaton model of granular materials. *Proc. 5th Distributed Memory computing conference, IEEE Computer society Press*, pages 550–556, 1990.
- [69] E. Hampe. *Silos*. VEB Verlag für Bauwesen, Berlin, 1987.
- [70] J. Hemmingson, H. J. Herrmann, and S. Roux. Vectorial cellular automaton for the stress in granular media. *Journal of Physics*, 1(7):291–302, 1997.
- [71] S. Hentz. Modélisation d’une structure en béton armé soumise à un choc par la méthode des éléments discrets. *PhD. thesis, Joseph Fourier University, Grenoble*, 2003.
- [72] S. Hentz, L. Daudeville, and F. Donzé. Discrete element modelling of concrete and identification of the constitutive behaviour. *15th ASCE Engineering Mechanics Conference*, 2002.
- [73] S. Hentz, F. Donzé, and L. Daudeville. Discrete element modelling of concrete submitted to dynamic loading at high strain rates. *Computers and Structures*, 82:2509–2524, 2004.

- [74] H. J. Herrmann, A. Hansen, and S. Roux. Fracture of disordered elastic lattices in two dimensions. *Physical Rev. B*, 39:637–647, 1989.
- [75] A. Hrennikoff. Solution of problems of elasticity by the framework method. *Journal of Applied Mechanics*, pages 169–175, 1941.
- [76] K. Hutter and N. Kuerchner. PIV for granular avalanches. *Dynamical Response of Granular and Powder Materials in Large and Catastrophic Deformations*, 2003.
- [77] A. Ibrahimbegovic, D. Markovic, and F. Gatuing. Constitutive model of coupled damage-plasticity and its finite element implementation. *Eur. J. Finite Elem.*, 12(4):381–405, 2003.
- [78] K. Iwashita and M. Hakuno. Modified distinct element method simulation of cliff collapse. *Structural Eng. / Earthquake Eng.*, 7(1):133–142, 1990.
- [79] K. Iwashita and M. Oda. Rolling resistance at contacts in simulation of shear band development by DEM. *Journal of Engineering Mechanics*, pages 285–292, 1998.
- [80] K. Iwashita and M. Oda. Micro-deformation mechanism of shear banding process based on modified distinct element method. *Powder Technology*, 109:192–205, 2000.
- [81] M. Jirasek and Z. P. Bazant. Macroscopic fracture characteristics of random particle systems. *International Journal of Fracture*, 69(3):201–228, 1995.
- [82] J. R. Johanson. Controlling flow patterns by bins by use of an insert. *Bulk Solid Handling*, 2(3):11, 1982.
- [83] M. Kaminski and R. Antonowicz. The flow of rape seed in silo equipped with a discharge device. *Task Quarterly*, 7(4):561–569, 2003.
- [84] M. Kaminski and M. Zubrzycki. Reduzieren des dynamischen horizontaldruckes in getreidesilos. *Bauingenieur*, pages 313–318, 1985.
- [85] N. Katsura, Shimosaka, A., Y. Shirakawa, and J. Hidaka. Simulation for flow behaviour of vibrating granular materials using cellular automata. *Powders and Grains (ed. Kishino), Swets and Zeitlinger, Lisse.*, pages 525–528, 2001.
- [86] M. Klisinski and Z. Mróz. Description of anelastic deformations and damage for concrete. *Technical University of Poznan (in polish)*, page 193, 1988.
- [87] T. A. Kompfner. Ein finites elementmodell für die geometrisch und physikalisch nicht-lineare berechnung von stahlbetonschalen. *PhD. thesis*, 1983.
- [88] J. Kozicki and J. Tejchman. Simulations of granular flow in silos with a cellular automata model. *The International Journal of Storing, Handling and Processing Powder (Powder Handling and Processing)*, 13(3):267–275, 2001.
- [89] J. Kozicki and J. Tejchman. Application of a cellular automata model to granular flow. *Task Quarterly, Gdansk University of Technology*, 6(3):429–436, 2002.

- [90] J. Kozicki and J. Tejchman. Discrete methods to describe the behaviour of quasi-brittle and granular materials. *Electronic Proceedings of 16th Engineering Mechanics Conference ASCE, University of Washington, Seattle, USA, 16.07–18.07.2003*, pages 1–10, 2003.
- [91] J. Kozicki and J. Tejchman. Application of a cellular automaton to simulations of granular flow in silos. *Granular Matter*, 7(1):45–54, 2005.
- [92] J. Kozicki and J. Tejchman. Simulations of behaviour of granular bodies using a lattice gas automaton. *Archives of Hydro-Engineering and Environmental Mechanics*, 52(1):21–37, 2005.
- [93] J. Kozicki and J. Tejchman. Simulations of flow pattern in silos with a cellular automaton — part I. *Task Quarterly, Gdansk University of Technology*, 9(1):81–102, 2005.
- [94] J. Kozicki and J. Tejchman. Simulations of flow pattern in silos with a cellular automaton — part II. *Task Quarterly, Gdansk University of Technology*, 9(1):103–114, 2005.
- [95] J. Kozicki and J. Tejchman. Simulations of fracture in concrete elements using a discrete lattice model. *Proc. Int. Conf. Computer Methods in Mechanics CMM*, 2005.
- [96] J. Kozicki and J. Tejchman. 2d lattice model for fracture in brittle materials. *Archives of Hydro-Engineering and Environmental Mechanics*, 53(2):71–88, 2006.
- [97] J. Kozicki and J. Tejchman. Modelling of fracture process in brittle materials using a lattice model. *Computational Modelling of Concrete Structures, EURO-C (eds.: G. Meschke, R. de Borst, H. Mang and N. Bicanic), Taylor and Francis*, pages 139–145, 2006.
- [98] C. Le Bellégo, J. F. Dube, G. Pijaudier-Cabot, and B. Gerard. Calibration of nonlocal damage model from size effect tests. *European Journal of Mechanics A/Solids*, 22:33–46, 2003.
- [99] G. Lilliu and J. G. M. van Mier. 3d lattice type fracture model for concrete. *Engineering Fracture Mechanics*, 70:927–941, 2003.
- [100] J. Litwiniszyn. Application of the equation of stochastic processes to mechanics of loose bodies. In *Archives of Applied Mechanics*, volume 8, pages 393–411, 1956.
- [101] T. C. Y. Liu, A. H. Nilson, and F. O. Slate. Biaxial stress-strain relationships for concrete. *ASCE Journal of Engineering Mechanics*, 103:423–439, 1996.
- [102] R. Mahnken and E. Kuhl. Parameter identification of gradient enhanced damage models. *European Journal of Mechanics A/Solids*, 18:819–835, 1999.
- [103] J. Martinez, S. Masson, and D. Deserable. Flow patterns and velocity profiles during silo discharge simulation with a lattice grain model. *Proc. Intern. Conference on Silos, Partec 95, Nürnberg*, pages 367–379, 1995.

- [104] P. Menetrey and K. J. Willam. Triaxial failure criterion for concrete and its generalization. *ACI Structural Journal*, pages 311–318, 1995.
- [105] R. L. Michalowski. Flow of granular material through a plane hopper. *Powder Technology*, 39:29–40, 1984.
- [106] R. L. Michalowski. Strain localization and periodic fluctuations in granular flow processes from hoppers. *Geotechnique*, 40(3):389–403, 1990.
- [107] R. R. Michalowski and L. Shi. Deformation patterns of reinforced foundation sand at failure. *Journal of Geotechnical and Geoenvironmental Engineering*, 129(6):439, June 2003.
- [108] Z. Mróz. Mathematical models of inelastic concrete behaviour. *University Waterloo Press*, pages 47–72, 1972.
- [109] H. Mühlhaus and E. C. Aifantis. A variational principle for gradient plasticity. *Int. J. Solids Structures*, 28:845–858, 1991.
- [110] A. Murakami, H. Sakaguchi, T. Takasuka, and H. Fuji. Study of microtopology and its evolution in granular materials using a cellular automata model. *Powders and Grains*, pages 33–36, 2001.
- [111] M. Nooru-Mohamed. Mixed-mode fracture of concrete: an experimental approach. *PhD. thesis, Delft University of Technology*, 1992.
- [112] K. Nübel. Experimental and numerical investigation of shear localization in granular materials. *Publication Series of the Institute of Soil and Rock Mechanics, University of Karlsruhe*, page 162, 2002.
- [113] V. Osinov. A model of a discrete stochastic medium for the problems of loose material flow. *Continuum Mechanics and Thermodynamics*, 6:51–60, 1994.
- [114] R. Palaniswamy and S. P. Shah. Fracture and stress-strain relationship of concrete under triaxial compression. *ASCE Journal of Engineering Mechanics*, 100:901–916, 1974.
- [115] J. Pamin. Gradient-enhanced continuum models: formulation, discretization and applications, habilitation monograph. *Cracow University of Technology*, 2004.
- [116] J. Pamin and R. de Borst. Simulation of crack spacing using a reinforced concrete model with an internal length parameter. *Arch. App. Mech.*, 68(9):613–625, 1998.
- [117] R. H. Peerlings, R. de Borst, W. A. M. Brekelmans, and M. G. D. Geers. Gradient enhanced damage modelling of concrete fracture. *Mechanics of Cohesive-Frictional Materials*, 3:323–342, 1998.
- [118] R. H. J. Peerlings, R. de Borst, W. A. M. Brekelmans, and M. G. D. Geers. Gradient enhanced damage modelling of concrete fracture. *Mechanics of Cohesive-Frictional Materials*, 3:323–342, 1998.

- [119] G. Peng and H. J. Herrmann. Density waves of granular flow in a pipe using lattice gas automata. *Physical Reviews Letter*, 49(3):1796–1799, 1994.
- [120] K. Pieper and K. Wagner. Der einfluß verschiedener auslaufarten auf die seitedrücke in silozellen. *Aufbereitungs technik*, 10:542–546, 1968.
- [121] S. Pietruszczak, J. Jiang, and F. A. Mirza. An elastoplastic constitutive model for concrete. *Int. J. Solids Structures*, 24(7):705–722, 1988.
- [122] G. Pijaudier-Cabot and Z. Bazant. Nonlocal damage theory. *ASCE Journal of Engineering Mechanics*, 113:1512–1533, 1987.
- [123] G. Pijaudier-Cabot, K. Haidar, and J. F. Dube. Non-local damage model with evolving internal length. *Int. J. Num. and Anal. Meths. in Geomech.*, 28:633–652, 2004.
- [124] D. Potyondi and P. Cundall. Modeling notch formation in the URL mine-by tunnel: Phase IV—enhancements to the PFC model of rock. *Tech. Rep. 06819-REP-01200-10002-R00. Itasca consulting group Inc.*, 1999.
- [125] D. Potyondy, P. Cundall, and L. C. Modelling rock using bonded assemblies of circular particles. *Rock Mech*, pages 1937–1944, 1996.
- [126] M. Raffel, C. Willert, and J. Kompenhaus. Particle image velocimetry. *Springer, Berlin Heidelberg*, 1998.
- [127] A. L. Rechenmacher. Grain-scale processes governing shear band initiation and evolution in sands. *Journal of the Mechanics and Physics of Solids*, 54:22–45, 2006.
- [128] M. Reimbert and A. Reimbert. *Silos — Theory and Practice*. Trans Tech Publishing, Clausthal, Germany, 1976.
- [129] S. Safarian and E. Harris. Design construction of silos and bunkers. *Von Nostrand Reinhold Company*, 1985.
- [130] H. Sakaguchi, A. Murakami, T. Hasegawa, and A. Shirai. Connected lattice cellular-automaton particles: model for pattern formation in vibrating granular media. *Solids and Foundations*, 36(1):105–110, 1996.
- [131] S. Savage. Some aspects of confined granular flows. *Proc. Int. Con. on Silos: Silos – Forschung and Praxis, University of Karlsruhe*, pages 111–121, 1992.
- [132] S. Savage. Disorder, diffusion and structure formation in granular flows. *Disorder and Granular Media* (eds. D. Bideau and A. Hansen), North-Holland, pages 255–185, 1993.
- [133] E. Schlangen. Experimental and numerical analysis of fracture processes in concrete. *PhD. thesis, Delft Univ. of Technology, Delft, The Netherlands.*, 1993.
- [134] E. Schlangen. Computational aspects of fracture simulations with lattice models. *F.H. Wittmann (Ed.), Fracture Mechanics of Concrete Structures*, pages 913–928, 1995.

- [135] E. Schlangen and E. Garboczi. New method for simulating fracture using an elastically uniform random geometry lattice. *International Journal of Engineering Science*, 34(10):1131–1144, 1996.
- [136] E. Schlangen and E. J. Garboczi. Fracture simulations of concrete using lattice models: computational aspects. *Engineering Fracture Mechanics*, 57:319–332, 1997.
- [137] E. Schlangen and J. van Mier. Shear fracture in cementitious composites, part II: Numerical simulations. Z.P. Bazant (Ed.), *Fracture Mechanics of Concrete Structures (Proc., FraMCoS-1)*, Elsevier, pages 671–676, 1992.
- [138] E. Schlangen and J. van Mier. Simple lattice model for numerical simulation of fracture of concrete materials and structures. *Materials and Structures*, 25:534–542, 1992.
- [139] J. Schwedes. Lagern und fließen von schüttgütern. hochschulkurs. *Internal Report, University of Braunschweig*, 1999.
- [140] K. Shoemake. Animating rotation with quaternion curves. *SIGGRAPH course notes*, 19(3):245–254, 1985.
- [141] K. Shoemake. Quaternion calculus and fast animation. *SIGGRAPH course notes*, 1987.
- [142] A. Simone and L. Sluys. Continuous-discontinuous modeling of mode-i and mode-ii failure. *Modelling of Cohesive-Frictional Materials (eds.: P.A. Vermeer, W. Ehlers, H. J. Herrmann and E. Ramm) Balkema*, pages 323–337, 2004.
- [143] C. Slominski, M. Niedostatkiwicz, and J. Tejchman. Deformation measurements in granular bodies using a particle image velocimetry technique. *Archives of Hydro-Engineering and Environmental Mechanics*, 53(1):71–94, 2006.
- [144] L. Sluys. Wave propagation, localisation and dispersion in softening solids. PhD. thesis. *Delft University of Technology*, 1992.
- [145] L. Sluys and R. de Borst. Dispersive properties of gradient and rate-dependent media. *Mech. Mater.*, 183:131–149, 1994.
- [146] S. Standard. *Lastannahmen für Bauten*. DIN 1055, Teil 6, 1987.
- [147] J. Strusch, J. Schwedes, and B. Hardow. Inserts loads and wall stress distributions in silos with inserts. *Proc. Inter. Conference on Silos, Partec95, Nürnberg*, pages 163–182, 1995.
- [148] K. Tajima and N. Shirai. Numerical prediction of crack width in reinforced concrete beams by particle model. *Computational Modelling of Concrete Structures, EURO-C (eds.: G. Meschke, R. de Borst, H. Mang and N. Bicanic)*, Taylor and Francis, pages 221–230, 2006.
- [149] J. Tejchman. *Shear localization and autogeneous dynamic effects in granular bodies*. Publication Series of the Institute for Rock and Soil Mechanics, Karlsruhe University, 1997.

- [150] J. Tejchman and M. Klisinski. Fe-studies on rapid flow of bulk solids in silos. *Granular Matter*, 3(4):215–231, 2001.
- [151] C. Thornton and L. Zhang. A DEM comparison of different shear testing devices (invited lecture). *Powders and Grains conference*, Kishino (ed.), 2001.
- [152] S. Ulm. Random processes and transformations. In *Proceedings of International Conference of Mathematics*, volume 2, pages 264–275, 1952.
- [153] J. van Mier, A. Vervuurt, and E. Schlangen. Boundary and size effects in uniaxial tensile tests: a numerical and experimental study. *Fracture and Damage in Quasibrittle Structures (Proc., NSF Workshop, Prague)*, EFN Spon, London, pages 289–302, 1994.
- [154] J. G. M. van Mier, E. Schlangen, and A. Vervuurt. Lattice type fracture models for concrete. *Continuum Models for Materials with Microstructure*, H.B. Mühlhaus, ed., John Wiley & Sons, pages 341–377, 1995.
- [155] J. G. M. van Mier and M. R. A. van Vliet. Influence of microstructure of concrete on size/scale effects in tensile fracture. *Engineering Fracture Mechanics*, 70:2281–2306, 2003.
- [156] M. R. A. van Vliet. Size effect in tensile fracture of concrete and rock. *PhD. thesis*, 2000.
- [157] M. R. A. van Vliet and J. G. M. van Mier. Experimental investigation of concrete fracture under uniaxial compression. *Mechanics of Cohesive-Frictional Materials*, 1:115–127, 1996.
- [158] A. Vervuurt, J. G. M. van Mier, and E. Schlangen. Lattice model for analyzing steel-concrete interactions. *Computer Methods and Advances in Geomechanics*, Siriwardane and Zaman, eds., Balkema, Rotterdam, pages 713–718, 1994.
- [159] R. Vidya Sagar. Size effect in tensile fracture of concrete – a study based on lattice model applied to ct-specimen, cd-rom. *Proc. 21th Intern. Congress on Theoretical and Applied Mechanics, ICTAM04, Warsaw*, pages 1–2, 2004.
- [160] J. Walraven. Aggregate interlock: a theoretical and experimental analysis. *PhD. thesis*, Delft University of Technology, 1980.
- [161] G. Wells and L. Sluys. A new method for modeling of cohesive cracks using finite elements. *Int. Journ. for Numerical Methods in Engineering*, 50(12):2667–2682, 2001.
- [162] D. J. White, W. A. Take, and M. D. Bolton. Soil deformation measurement using particle image velocimetry (PIV) and photogrammetry. *Géotechnique*, 53(7):619–631, 2003.
- [163] H. M. Zbib and C. E. Aifantis. A gradient dependent flow theory of plasticity: application to metal and soil instabilities. *Appl. Mech. Reviews*, 42(11):295–304, 1989.
- [164] L. Zhang and C. Thornton. Numerical simulations of biaxial compression. *International Congress for Particle Technology, PARTEC*, 2001.

- [165] J. G. Zhou. Lattice Boltzmann Methods for Shallow Water Flows. *Springer Verlag*, ISBN 3-540-40746-4, 2004.

List of publications

Publications in journals from ISI Master Journal List ¹

- [1] J. Kozicki, J. Tejchman, Application of a cellular automaton to simulations of granular flow in silos. *Granular Matter*, Volume 7, No 1, pages 45–54, 2005.
- [2] J. Kozicki, J. Tejchman, Effect of aggregate structure on fracture process in concrete using 2D lattice model. *Archives of Mechanics*, (accepted, in print), 2007.

Publications in peer-reviewed journals

- [3] J. Kozicki, J. Tejchman, Simulations of Behavior of Granular Bodies using a Lattice Gas Automaton. *Archives of Hydro-Engineering and Environmental Mechanics*, Volume 52, No 1, pages 21–37, 2005.
- [4] J. Kozicki, J. Tejchman, Simulations of granular flow in silos with a cellular automata model. *International Journal of Storing, Handling and Processing Powder*, pages 267–275, 2001.
- [5] J. Kozicki, J. Tejchman, 2D Lattice Model for Fracture in Brittle Materials. *Archives of Hydro-Engineering and Environmental Mechanics*, Volume 53, No 2, pages 71–88, 2006.
- [6] J. Kozicki, J. Tejchman, Application of a cellular automata model to granular flow. *Task Quarterly, Gdansk University of Technology*, Volume 6, No 3, pages 429–436, 2002.
- [7] J. Kozicki, J. Tejchman, Simulations of flow pattern in silos with a cellular automaton, part I. *Task Quarterly, Gdansk University of Technology*, No 1, 2005.
- [8] J. Kozicki, J. Tejchman, Simulations of flow pattern in silos with a cellular automaton, part II. *Task Quarterly, Gdansk University of Technology*, No 2, 2005.
- [9] J. Kozicki, Discrete lattice model used to describe the fracture process of concrete. *Discrete Element Group for Risk Mitigation Annual Report 1, Grenoble University of Joseph Fourier, France*, pages 95–101, 2005.

¹Institute for Scientific Information Master Journal List is also known under name „Philadelphian List”, see <http://scientific.thomson.com/mjl/>

- [10] J. Kozicki, J. Tejchman, Experimental investigations of strain localization in concrete using Digital Image Correlation (DIC) technique. *Archives of Hydro-Engineering and Environmental Mechanics*, (submitted for publication), 2007.

Conference materials

- [11] J. Kozicki, J. Tejchman, Modelling of fracture process in brittle materials using a lattice model. *Computational Modelling of Concrete Structures, EURO-C* (eds.: G. Meschke, R. de Borst, H. Mang and N. Bicanic), Taylor and Francis, pages 139–145, 2006.
- [12] J. Kozicki, J. Tejchman, Discrete methods to describe the behaviour of quasi-brittle and granular materials. *16th Engineering Mechanics Conference, University of Washington, Seattle*, CD-ROM, July 16–18, 2003.
- [13] J. Kozicki, J. Tejchman, Discrete methods to describe the behaviour of quasi-brittle and granular materials. *Proc. Conf. Computer Methods in Mechanics*, pages 195–197, Gliwice 6 July 2004.
- [14] J. Kozicki, J. Tejchman, Cellular automata model to describe granular flow in silos. *PTSK Symulacja w Badaniach i Rozwoju*, pages 214–219, 30 august — 1 september, 2003.
- [15] J. Kozicki, J. Tejchman, Lattice method to describe the behaviour of quasi-brittle materials. *CURE Workshop, Effective use of building materials*, pages 131–134, Sopot October 8-9, 2003.
- [16] J. Kozicki, J. Tejchman, Study of Fracture Process in Concrete using a Discrete Lattice Model. *CURE Workshop, Simulations in Urban Engineering*, Gdańsk September 20–22, 2004.
- [17] J. Kozicki, J. Tejchman, Application of an Improved Cellular Automaton for Granular Flow. *CURE Workshop, Simulations in Urban Engineering*, Gdańsk September 20–22, 2004.
- [18] J. Kozicki, J. Tejchman, Simulations of granular flow with a lattice-gas automaton. *Proc. Conf. Computer Methods in Mechanics*, Czestochowa, Poland, June 21–24 2005.
- [19] J. Kozicki, J. Tejchman, Simulations of fracture in concrete elements using a discrete lattice model. *Proc. Conf. Computer Methods in Mechanics*, Czestochowa, Poland, June 21–24 2005.
- [20] J. Kozicki, J. Tejchman, Simulation of the crack propagation in concrete with a discrete lattice model. *Proc. Conf. Analytical Models and New Concepts in Concrete and Masonry Structures*, Gliwice, Poland, June 12–14 2005.
- [21] J. Kozicki, J. Tejchman, Lattice type fracture model for brittle materials. *35th Solid Mechanics Conference*, pages 215–216, Krakow, 4–8 September 2006.

- [22] M. Niedostatkiewicz, J. Kozicki, J. Tejchman, Measurements of strain localization in granular and brittle specimens using a digital image correlation technique. *35th Solid Mechanics Conference*, pages 93–94, Krakow, 4–8 September 2006.

Conference materials in polish language

- [23] J. Kozicki, J. Tejchman, Zastosowanie automatów komorkowych do opisu przepływów materiałów sypkich w silosach. *Materialy XXXVII Zjazdu Fizyków Polskich*, pages 1116–1120, Gdansk 15–18 September 2003, 2003.
- [24] J. Kozicki, M. Niedostatkiewicz, Zastosowanie automatu komorkowego do opisu przepływu materiałów sypkich w silosach. *XII Konferencja Naukowo-Techniczna, Żelbetowe i sprezone zbiorniki na materiały sypkie i ciecze*, pages 115–122, Krakow 19–21 November 2003.