National Technical University of Athens

Computational Mechanics

Master's Thesis

Impact Problem Analysis using Node to Segment Algorithm

By

Theodoros Malmen, Mechanical Engineer

Supervisor:

Associate Professor, Vissarion Papadopoulos

Athens, June 2021

Blank page

Abstract

The current work addresses an impact - contact problem which simulates the impact of two bodies of different Young modulus. In particular, the case applied in this thesis refers to a body coming into contact with an approximately 1000 times less stiff body. Modelling, discretization and the solution are implemented in C# programming and, consequently, results are validated through the "Abaqus" software. More specifically, a circular ring and a rectangular body are being discretized by four-noded quadrilateral elements, whilst contact is enforced by the ring's initial velocity. "Node to Segment" contact elements will be employed as the "bridge elements" among locally separated but, potentially interacting, surfaces. In addition, the well-known "penalty method" will be applied to the aforementioned contact problem.

In order to reinforce our understanding of the present analysis approach, we briefly introduce the theoretical (i.e., physical and mathematical) framework of impact-contact mechanics that will be later applied to our application problem.

Contents

List of Figures

List of Tables

1 Introduction

Contact mechanics has its application in many engineering problems. No one can walk without frictional contact, and no car would move for the same reason. Hence contact mechanics has, from an engineering point of view, a long history, beginning in ancient Egypt with the movement of large stone blocks, over first experimental contributions from leading scientists like Leonardo da Vinci and Coulomb, to today's computational methods. In the past contact conditions were often modelled in engineering analysis by more simple boundary conditions since analytical solutions were not present for real world applications. In such cases, one investigated contact as a local problem using the stress and strain fields stemming from the analysis which was performed for the entire structure. With the rapidly increasing power of modern computers, more and more numerical simulations in engineering can include contact constraints directly, which make the problems nonlinear.

From a mechanical point of view, at macro-scale, contact is a notion for all types of interactions between separate bodies coming in touch. Direct contact between solids allows to transfer a load, a heat and an electric charge from one body to another. The physics of the contact interaction is particularly rich and complicated, due to the multi-scale and multi-physical nature of the phenomenon. The branch of mechanical engineering studying this interaction is called tribology - a science of relative motion of interacting surfaces in a comprehensive framework combining mechanical, physical and chemical effects at different scales. This dissertation presents the mathematical description and modeling of the mechanical aspects of this interaction. (Yastrebov, 2011)

Boundary value problems involving contact are of great importance in industry related to mechanical and civil engineering, but also in environmental and medical applications. Virtually all movements on this planet involve contact and friction, like simple walking or running, driving of cars, riding bicycles or steaming of trains. If friction were not present (see movement on ice), all these motions would not be possible. Also, the area in which a foot, a tyre or a wheel interacts with the soil, the road or the rail is not known *a priori*, leading to a nonlinear boundary value problem for these simple everyday tasks. (Wriggers, 2002)

The objective of this work is a study on contact-impact mechanics, and more specifically on contact detection, discretization and the ways that the contact constraints can be enforced to the solution algorithm. Also, since we have to do with a dynamic problem, an algorithm that solves the governing equation at each time step is developed and also the nonlinearity of the phenomenon is taken into account.

2 Impact-Transient Dynamics

2.1 Introduction

During the early use of the finite element method, equations of the order 10,000 were in many cases considered of large order. Currently, equations of the order 100,000 and more are solved without much difficulty. Depending on the kind and number of elements used in the assemblage and on the topology of the finite element mesh, in a linear static analysis the time required for solution of the equilibrium equations can be a considerable percentage of the total solution time, whereas in dynamic analysis or in nonlinear analysis, this percentage may be still higher. Therefore, if inappropriate techniques for the solution of the equilibrium equations are used, the total cost of analysis is affected a great deal, and indeed the cost may be many times, say 100 times, larger than is necessary.

In addition to considering the actual computer effort that is spent on the solution of the equilibrium equations, it is important to realize that an analysis may, in fact, not be possible if inappropriate numerical procedures are used. This may be the case because the analysis is simply too costly using the slow solution methods. But, more seriously, the analysis may not be possible because the solution procedures are unstable. We will observe that the stability of the solution procedures is particularly important in dynamic analysis.

Firstly, we are concerned with the solution of the simultaneous equations that arise in the static analysis of structures and solids, and we discuss first the solution of the equations that arise in linear analysis,

$$\mathbf{KU} = \mathbf{R} \tag{2.1}$$

where **K** is the stiffness matrix, **U** is the displacement vector, and **R** is the load vector of the finite element system. Since **R** and **U** may be functions of time *t*, we may also consider (2.1) as the dynamic equilibrium equations of a finite element system in which inertia and velocity-dependent damping forces have been neglected. It should be realized that since velocities and accelerations do not enter (2.1), we can evaluate the displacements at any time *t* independent of the displacement

history, which is not the case in dynamic analysis. However, these thoughts suggest that the algorithms used for the evaluation of **U** in (2.1) may also be employed as part of the solution algorithms used in dynamic analysis.

Essentially, there are two different classes of methods for the solution of the equations in (2.1): direct solution techniques and iterative solution methods. In a direct solution the equations in (2.1) are solved using a number of steps and operations that are predetermined in an exact manner, whereas iteration is used when an iterative solution method is employed. At present, direct techniques are employed in most cases, but for large systems iterative methods can be much more effective.

The most effective direct solution techniques currently used are basically applications of Gauss elimination, which C. F. Gauss proposed over a century ago. However, although the basic Gauss solution scheme can be applied to almost any set of simultaneous linear equations, the effectiveness in finite element analysis depends on the specific properties of the finite element stiffness matrix: symmetry, positive definiteness, and handedness.

In many analyses some form of direct solution based on Gauss elimination to solve the equilibrium equations **KU = R** is very efficient. It is interesting to note, however, that during the initial developments of the finite element method, iterative solution algorithms have been employed. A basic disadvantage of an iterative solution is that the time of solution can be estimated only very approximately because the number of iterations required for convergence depends on the condition number of the matrix **K** and whether the acceleration schemes used are effective for the particular case considered. It is primarily for this reason that the use of iterative methods in finite element analysis was largely abandoned during the 1960s and 1970s, while the direct methods of solution have been refined and rendered extremely effective.

However, when considering very large finite element systems, a direct method of solution can require a large amount of storage and computer time. The basic reason is that the required storage is proportional to nm_K, where n = number of equations, m_K = half-bandwidth, and a measure of the number of operations is $(1/2)nm_K^2$. Since the half-bandwidth is (roughly) proportional to $\sqrt{n}$, we recognize that as n increases, the demands on storage and computation time can become very large. In practice, the available storage on a computer frequently limits the size of finite element system that can be solved.

On the other hand, in an iterative solution the required storage is much less because we need to store only the actually nonzero matrix elements under the skyline of the matrix, a pointer array that indicates the location of each nonzero element, and some arrays, also of small size measured on the value of nm_K, for example, for the preconditioner and iteration vectors. The nonzero matrix elements under the skyline are only a small fraction of all the elements under the skyline.
The fact that considerable storage can be saved in an iterative solution has prompted a large amount of research effort to develop increasingly effective iterative schemes. The key to effectiveness is of course to reach convergence within a reasonable number of iterations. As we shall see, of major importance in an iterative scheme is therefore a procedure to accelerate convergence when slow convergence is observed. The fact that effective acceleration procedures have become available for many applications has rendered iterative methods very attractive.

2.2 Non Linear Boundary Conditions

Mechanical problems are classically formulated as boundary value problems, where a governing differential equations should be fulfilled within the domain Ω and ordinary boundary conditions are imposed on the domain's closure $\partial\Omega$. The balance of virtual work yields a weak (integral) form of this boundary value problem, which presents a basis on which the structural Finite Element Method is constructed. Contact constraints are formulated as sets of inequalities. Such a formulation is not usual for boundary value problems. The rigorous construction of a variational principle leads to a variational inequality instead of a classic variational equality. Such a new mathematical structure requires new solution approaches. The problem becomes even more complex when a frictional effect is assumed at the interface. Coulomb's friction law states that tangential resistance depends upon the normal contact pressure, but the latter is known only if the solution is known. Roughly speaking, the boundary conditions are solution dependent, which naturally leads to difficulties in the formulation of the frictional contact problem. Moreover, the nature of Coulomb's friction law yields a non-smooth energy functional resulting in even more difficulties from a numerical point of view. As pointed out in the book of Kikuchi and Oden (Kikuchi & Oden, 1988) "Frictional contact problem between continuous deformable solids involves formidable mathematical difficulties".

"Nonlinear" means at least one (and sometimes all) of the following complications:

- Changing boundary conditions (such as contact between parts that changes over time)
- Large deformations (for example the crumpling of sheet metal parts)
- Nonlinear materials that do not exhibit ideally elastic behavior (for example thermoplastic polymers)

As stated above, one source for nonlinearities is related to special boundary constraints. One major cause of nonlinear behavior are boundary constraints which change with the deformation state of a system (e.g. during the increase of a prescribed load). These occur when one body comes into contact with another one during a deformation process. Here a penetration of one body into the other is ruled out and the contact zone between the two bodies changes depending on the load level.

Due to the nonlinear nature of contact mechanics, such problems in the past were often approximated by special assumptions within the design process. Due to the rapid improvement of modern computer technology, one can today apply the tools of computational mechanics to simulate applications which include contact mechanisms numerically. This can be done to an accuracy which is sufficient for design purposes. However, even now most of the standard finite element software is not fully capable of solving contact problems, including friction, with robust algorithms. Hence there is still a challenge for the finite element society to design efficient and robust methods for computational contact mechanics.

Basically, geometrical constraints and equilibrium equations have to be considered in contact formulations which are not differentiable since the system can assume two different states of being in contact or being not in contact. This is reflected by the kink in the load deflection curve.

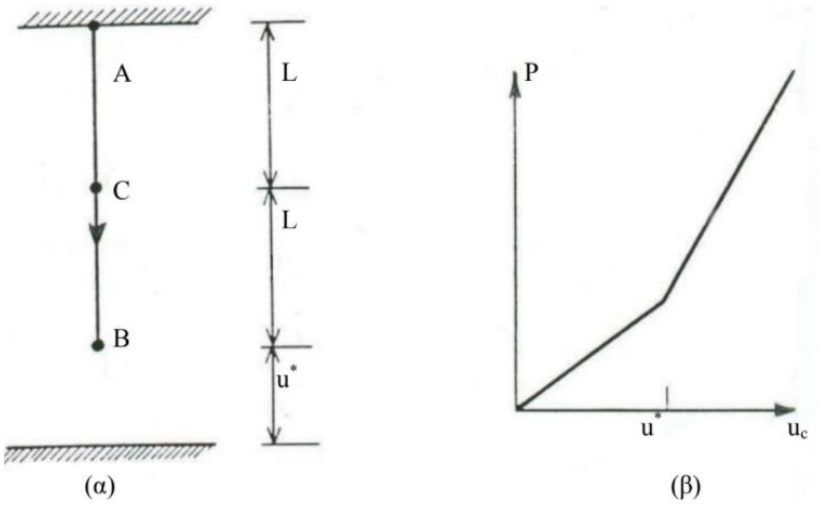


Figure 1. *(a) Boundary condition non-linearity (β) Force-displacement curve*

Since contact nonlinearities are linked in many technical applications to further nonlinearities, like finite or inelastic deformations, it is especially complicated to construct robust end efficient algorithms for contact problems.

All discussed examples show that the source of nonlinear behavior is quite different. Besides geometrical effects, material properties or changing boundary conditions yield nonlinear behavior. In the following chapter, the underlying theoretical basis is generalized for two- and three-dimensional solids and the necessary numerical solution schemes based on the finite element method will be developed.

For nonlinear problems in structural mechanics, there exist a large number of algorithms and solution procedures. The most common ones applied within finite element methods are

- fix-point methods,
- Newton-Raphson methods,
- quasi-Newton methods,
- dynamical relaxation and
- continuation or arc-length methods.

The efficiency of the different methods for the solution of nonlinear equation systems depends also upon the application and its size in terms of number of unknowns. As an example, the method of Newton-Raphson can be very efficient in combination with direct elimination methods for problems with low number of unknowns. For problems with large number of unknowns, quasi-Newton methods or the dynamical relaxation can be more efficient since, even with more iterations, they need less computation time. However, also a combination of the Newton-Raphson method together with iterative linear solvers can be very efficient and faster than quasi-Newton procedures. (Wriggers, 2008) or (Bathe, 1996).

NEWTON-RAPHSON METHOD

In solid mechanics and nonlinear structural mechanics, the range of problems is wide and areas are quite different (geometrical nonlinearity, physical nonlinearity, stability, etc.). Thus there exists up to now no iterative method which can be applied to all different problem areas in an efficient and robust way. Due to that, several methods will be presented which were developed for the solution of the nonlinear equation system $\boldsymbol{G}(\boldsymbol{v}) = 0.$

Finite element approximations with the interpolations lead to a system of nonlinear algebraic equations with *N* unknowns. This system of equations can be recast for the following considerations in the form

$$\boldsymbol{G}(\boldsymbol{v}, \lambda) = \boldsymbol{R}(\boldsymbol{v}) - \lambda \boldsymbol{P} = 0, \quad \boldsymbol{v} \in \mathbb{R}^{\boldsymbol{N}}. \tag{2.2}$$

The scaling factor λ in front of the load term $\boldsymbol{P}$ is called loading parameter. It is introduced to be able to change the load level with an iterative method. The parameter λ is usually determined by the problem at hand, e.g. as total given load. However, in special iterative methods, it makes sense to consider λ as an unknown variable.

The most frequently applied scheme for the iterative solution of systems of nonlinear algebraic equation is the Newton-Raphson algorithm. It is based on a Taylor series development of (2.2) at an already known state $\boldsymbol{v}_k$

$$\boldsymbol{G}(\boldsymbol{v}_k + \Delta\boldsymbol{v}, \bar{\lambda}) = \boldsymbol{G}(\boldsymbol{v}_k, \bar{\lambda}) + D\boldsymbol{G}(\boldsymbol{v}_k, \bar{\lambda})\Delta\boldsymbol{v} + \boldsymbol{r}(\boldsymbol{v}_k, \bar{\lambda}). \tag{2.3}$$

The loading parameter $\bar{\lambda}$ denotes the load level for which the solution has to be determined. In (2.3) $D\boldsymbol{G} \cdot \Delta\boldsymbol{v}$ characterizes the directional derivative of $\boldsymbol{G}$ at $\boldsymbol{v}_k$ also referred to as linearization. The linearization of the vector $\boldsymbol{G}$ yields a matrix, which is also known as Hesse-, Jacobi- or tangent matrix. This matrix will be abbreviated in the following by K_T. The vector $\boldsymbol{r}$ is the residuum of the Taylor series. By neglecting the residuum, the linear equation system $\boldsymbol{G}(\boldsymbol{v}_k + \Delta\boldsymbol{v}, \bar{\lambda}) = 0$ is obtained from (2.3) which is the basis of the following iterative algorithm for the solution of Eq. (2.2). This algorithm, so far, determines the solution for the load level defined by the load parameter $\bar{\lambda}$. The associated convergence behavior is depicted in Fig. 2 for a one-dimensional problem. For this purpose, the nonlinear equation $\boldsymbol{G}(\boldsymbol{v}, \bar{\lambda}) = \boldsymbol{R}(\boldsymbol{v}) - \bar{\lambda}\boldsymbol{P} = 0$ was normalized as: $\hat{\boldsymbol{R}}(\boldsymbol{v}) - \bar{\lambda} = 0$.

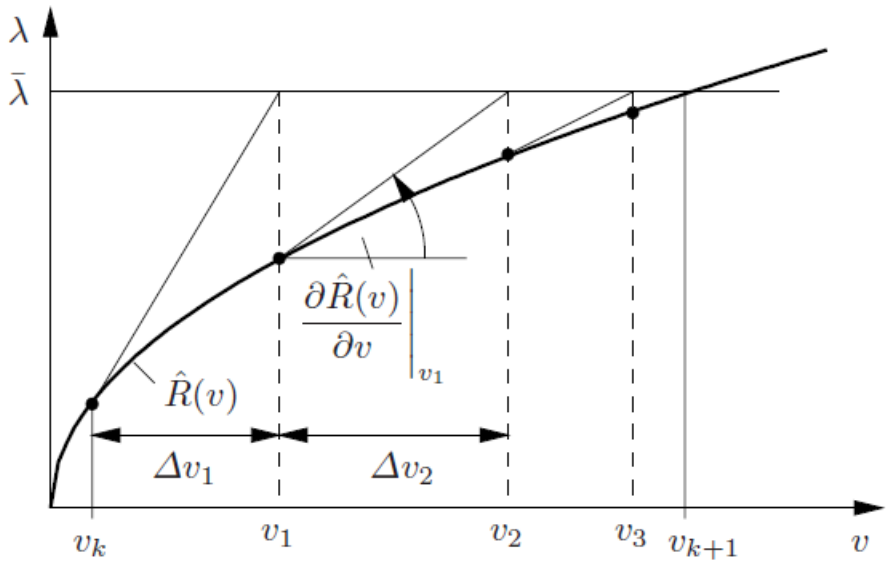


Figure 2. *Illustration of the Newton-Raphson scheme.*

The rate of convergence of the Newton-Raphson scheme is characterized by the inequality $\|\boldsymbol{v}_{k+1} - \boldsymbol{v}\| \leq C\|\boldsymbol{v}_k - \boldsymbol{v}\|^2$, where $\boldsymbol{v}$ is the solution of (2.2). The quadratic convergence of the Newton- Raphson scheme, which is apparent from the above inequality, has a local character since it is only valid near the solution point. This convergence behavior is advantageous since most of the time only a few iterations are needed to obtain the solution of (2.2). A drawback of the Newton-Raphson scheme stems from the fact that, in every iteration step, the tangent matrix K_T has to be computed and a linear equation system has to be solved, which can be quite time consuming and hence expensive. To shorten the notation, the term Newton scheme will be used instead of the historically more correct Newton-Raphson scheme. Since it is often very complicated to derive the tangent matrix analytically, different other approaches are possible. One is related to a combination of automatic and symbolic differentiation, and another computed the derivatives by difference quotients. In combination with Newton method, the latter approach is called a discrete Newton scheme. One possibility is to apply the forward difference quotient. Using its definition, the approximation

$$k_m \approx \frac{1}{h_m}\left[\boldsymbol{G}\left(\boldsymbol{v}_i + h_m e_m, \bar{\lambda}\right) - G\left(\boldsymbol{v}_i, \bar{\lambda}\right)\right] \tag{2.4}$$

is obtained for the *m-th* column of the tangent matrix. In this relation, h_m is the step size and e_m is a vector, which contains zeros everywhere besides at the position *m*, where it has the value 1. In case of *N* total unknowns, the tangent matrix can be obtained by *N* rows $\boldsymbol{k}_m$ as follows

$$K_T = [k_1 \;\; k_2 \;\; \ldots \; k_m \;\; \ldots \;\; k_N] \tag{2.5}$$

The step size in (2.4) has to be chosen such that the approximation of the tangent matrix is as good as possible. Optimal is a very small value for h_m. This choice is, however, not possible due to the limited computer accuracy. In case of a computer accuracy of η, the following estimate is valid

$$h_m = v(|(v_m)_i| + \tau) \qquad \textit{with} \;\; v = 10^{-3} \ldots 10^{-5} < \sqrt{\eta}, \tag{2.6}$$

where the number τ should be chosen as $\tau = 10^{-3}$ to prevent that h_m becomes zero for $(v_m)_i$= 0. Using such a step size leads even for the discrete Newton scheme to quadratic convergence near the solution point. A disadvantage of this simple scheme is the large number of evaluations of the residual ***G*** needed for the approximation of the tangent matrix in (2.5). In detail, when the equation system has *N* unknowns then also *N* evaluations are needed which makes the method inefficient for large equation systems. A more efficient way is to use the numerical differentiation procedure on element level; then only *n* evaluations related to the size of the element residual vector are needed. Furthermore, this methodology can be helpful during the development of nonlinear finite elements since the analytical derivation of the tangent matrix can be validated with the help of the numerical tangent obtained from (2.5). Also, for complicated constitutive equations, the incremental constitutive tensor can be determined at each Gauss point of a finite element using the finite difference scheme.

Table 1. Algorithm for the NEWTON RAPHSON scheme. (Wriggers, 2008)

Initial values: $\mathbf{v}_0 = \mathbf{v}_k$.

Iteration loop $i = 0, 1, \ldots$ *until convergence*

1. Compute $\boldsymbol{G}(\mathbf{v}_i, \bar{\lambda})$ and $\boldsymbol{K}_T(\mathbf{v}_i)$
2. Compute the displacement increments: $\boldsymbol{K}_T(\mathbf{v}_i)\,\Delta\mathbf{v}_{i+1} = -\boldsymbol{G}(\mathbf{v}_i, \bar{\lambda})$
3. Compute new displacement: $\mathbf{v}_{i+1} = \mathbf{v}_i + \Delta\mathbf{v}_{i+1}$
4. Test for convergence

$$\| \boldsymbol{G}(\mathbf{v}_{i+1}, \bar{\lambda}) \| \quad \begin{cases} \leq \text{TOL} \longrightarrow & \text{Set : } \mathbf{v}_{k+1} = \mathbf{v}_{i+1}\,, \quad \text{STOP} \\ > \text{TOL} \longrightarrow & \text{Set } i = i+1 \quad \text{go to 1)} \end{cases}$$

2.3 Equilibrium equations in Dynamic Analysis

When dealing with nonlinear partial differential equation which describe the deformation process of solids, then the change of state variables and deformations in time has to be considered. These problems are known as *initial boundary value problems* which additionally depend upon the time. Among engineering applications, related to initial value problems, are vibration analysis of structures or impact problems like car-crash simulations. In such cases, the inertia term in the linear momentum equation, cannot be neglected. Another class of problems is related to inelastic

constitutive behavior, such as elasto-plasticity, visco-plasticity or visco-elasticity. The inelastic response is governed by evolution equations, and thus in general a time dependent process.
Methods and algorithms are discussed in this chapter which can be applied to the above mentioned problem classes with special emphasis to dynamical systems in solid mechanics and nonlinear time dependent constitutive equations. Before starting with the specific application, some general remarks are made concerning the integration of algebraic differential systems or ordinary differential equations systems. These generally appear when the weak forms describing a solid are spatially discretized by finite elements which then results to a system of ordinary differential equations in time.

The discrete equations of motions were derived from the weak form of linear momentum and by including linear damping, the general form of the equations of motions follows as

$$\boldsymbol{M}\ddot{u} + \boldsymbol{C}\dot{u} + \boldsymbol{R}(u) = \boldsymbol{P} \tag{2.7}$$

where **M** stands for the mass matrix, the damping matrix $\boldsymbol{C}$ is assumed to be constant such that the damping force has the form $\boldsymbol{C}\dot{u}$, **R**(u) denotes the vector of the internal forces (stress divergence) and $\boldsymbol{P}$ contains the time dependent prescribed loads.

This equation can be transformed to a first order differential equation system by introducing the independent variables $\dot{\boldsymbol{u}} = \boldsymbol{v}$ and $\ddot{\boldsymbol{u}} = \dot{\boldsymbol{v}}$.

$$\dot{\boldsymbol{u}} = \boldsymbol{v}, \tag{2.8}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{M}^{-1}[\,\boldsymbol{P} - \boldsymbol{C}\boldsymbol{v} - \boldsymbol{R}(\boldsymbol{u})]. \tag{2.9}$$

For the description of the algorithms, the letter $\boldsymbol{a}$ is chosen to denote the accelerations $\ddot{\boldsymbol{u}}$ and the letter $\boldsymbol{v}$ denotes the velocities $\dot{\boldsymbol{u}}$. With this notation, the discretized equation of the linear momentum has at time t_{n+1} the form

$$\boldsymbol{M}a_{n+1} + \boldsymbol{C}v_{n+1} + \boldsymbol{R}(u_{n+1}) = \boldsymbol{P}_{n+1} \quad \textbf{(2.10)}$$

The index $(..)_{n+1}$ means that the relevant quantity has to be computed at time t_{n+1}.

Finally, for the definition of an initial value problem, the initial conditions have to be described. These are conditions for the displacements $\overline{\boldsymbol{u}}$ and the velocities $\overline{\boldsymbol{v}}$ at time $t = t_0$ (usually $t_0 = 0$ is selected):

$$\boldsymbol{u_0} = \overline{\boldsymbol{u}}, \quad \textbf{(2.11)}$$

$$\boldsymbol{v_0} = \overline{\boldsymbol{v}}. \quad \textbf{(2.12)}$$

2.3.1 Explicit integration scheme

When high frequencies (e.g. stemming from impact) or shock waves dominate the solution of a physical problem described by (2.10), then small time steps are required. In such case, the most efficient way to integrate the equations of motions is provided by an explicit method. The central difference scheme is one of the favorite methods applied to solve the equations of motions in case of solid mechanics or structural problems. Within this scheme, the velocities $\boldsymbol{v}$ and the accelerations $\boldsymbol{a}$ are approximated at time t_n by

$$\boldsymbol{v_n} = \frac{\boldsymbol{u_{n+1} - u_{n-1}}}{\boldsymbol{2\Delta t}}, \quad \textbf{(2.13)}$$

$$\boldsymbol{a_n} = \frac{\boldsymbol{u_{n+1} - 2u_n + u_{n-1}}}{\boldsymbol{\Delta t^2}}. \quad \textbf{(2.14)}$$

By inserting these relations into the discretized form of the linear momentum balance (2.7) at time t_n

$$\boldsymbol{M}(u_{n+1} - 2u_n + u_{n-1}) + \frac{\Delta t}{2}\boldsymbol{C}(u_{n+1} - u_{n-1}) + \Delta t^2 \boldsymbol{R}(u_n) = \Delta t^2 \boldsymbol{P_n} \qquad \textbf{(2.15)}$$

an equation system is obtained from which the unknown displacement $\boldsymbol{u_{n+1}}$ at time $\boldsymbol{t_{n+1}}$ can be computed

$$\left(\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C}\right) u_{n+1} = \Delta t^2[\boldsymbol{P_n} - \boldsymbol{R}(u_n)] + \frac{\Delta t}{2}\boldsymbol{C}u_{n-1} + \boldsymbol{M}(2u_n - u_{n-1}). \qquad \textbf{(2.16)}$$

Here the mass matrix $\boldsymbol{M}$ and the damping matrix $\boldsymbol{C}$ are constant. Hence, for the coefficient matrix $\boldsymbol{M} + (\Delta t/2)\,\boldsymbol{C}$, a triangular decomposition can be used which leads to an efficient algorithm for the solution of (2.16). Note that the term $\boldsymbol{R}(u_n)$ which contains all nonlinearities appears only on the right hand side.

In the case that $\boldsymbol{M}$ and $\boldsymbol{C}$ are given in diagonal form, (*lumping*), then the inversion of $\boldsymbol{M} + (\Delta t/2)\,\boldsymbol{C}$ is trivial and only the vectors on the right hand side of (2.16) have to be evaluated.

The initialization of the finite difference scheme needs some special considerations since in (2.15) the values for the displacements u_{-1} have to be determined in order to start the integration process in a consistent way. These displacements can be computed from the initial values at time $t0$ by using the initial conditions u_0 and v_0. Based on a second order accurate Taylor series expansion for the displacements at time t_{-1}, the relation

$$u_{-1} = u_0 + \Delta t v_0 + \frac{\Delta t^2}{2} a_0 \qquad \textbf{(2.17)}$$

is obtained where the accelerations at time t_0 can be computed from the balance of linear momentum (2.10)

$$a_0 = M^{-1}[-Cv_0 + R(u_0) + P_0] \tag{2.18}$$

A variant of the above stated central difference scheme for the solution of (2.7) can be found in Wood (1990). It is equivalent to the already described method but uses the approximations

$$\begin{aligned} u_{n+1} &= u_n + \Delta t v_n + \frac{(\Delta t)^2}{2} a_n, \\ v_{n+1} &= v_n + \frac{1}{2}\Delta t(a_n + a_{n+1}) \end{aligned} \tag{2.19}$$

for displacements and velocities. Equation (2.9) can be applied to determine the accelerations. This yields the algebraic equation system

$$\left(\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C}\right) a_{n+1} = \boldsymbol{P}_{n+1} - \boldsymbol{R}\left(u_n + \Delta t v_n + \frac{(\Delta t)^2}{2} a_n\right) - \frac{\Delta t}{2}\boldsymbol{C} a_n, \tag{2.20}$$

which has on the right hand side, besides the loading vector $\boldsymbol{P}_{n+1}$, only quantities which are measured at time t_n. Thus the initial conditions for displacements and velocities can be incorporated directly in this scheme. Displacements and velocities follow after the solution of (2.20) from (2.19). The coefficient matrix in (2.20) does not change when compared to (2.16); hence the same efficiency as in the first formulation is obtained, especially when lumping procedures are applied.

However, as already mentioned, explicit methods are not unconditionally stable. The critical time step is given for linear problems by

$$\Delta t \leq \frac{T_N}{\pi}. \tag{2.21}$$

In this criterion which is named after Courant, the time $T_N$denotes the smallest period for a given finite element discretization. It can be estimated based on the element size and the speed of a wave travelling through a solid by, see e.g. Bathe (1996),

$$\Delta t \approx \frac{h}{c_L}. \tag{2.22}$$

h is a characteristic dimension of the smallest element in the FE-mesh and c_L is the velocity of a compression wave in a linear solid ($c_L = 3K\,(1-\nu)\,/\,\rho(1+\nu)$ with the modulus of compression K, the Poisson ration ν and the density ρ).

A critical time step limit can be found for nonlinear problems in (Belytschko , et al., 1976)

$$\Delta t \leq \delta \frac{h}{c_L}. \tag{2.23}$$

The constant δ ($0.2 < \delta < 0.9$) is a reduction factor which has to be selected according to the nonlinear properties of the problem under consideration.

In other words, the time step may be no larger than the amount of time required for a sound wave to traverse the element in the mesh having the smallest transit time. This fact tells us immediately that explicit finite element methods are most appropriate for those problems featuring very high frequency response or wave-like phenomena. For problems featuring low frequency response, literally millions of time steps may be required to resolve even a single period of vibration due to the stringent stability limit posed by (2.22). For such problems, an unconditionally stable algorithm is highly desirable, albeit at the cost of explicit updates in each increment.

2.3.2 Implicit integration scheme

Implicit methods can be applied alternatively to solve the nonlinear discrete equations of motion (2.7) for many engineering applications in structural and solid mechanics. The most popular scheme is the Newmark method, see (Newmark, 1959). It is based on the following approximations for displacements and velocities at time t_{n+1}

$$u_{n+1} = u_n + \Delta t v_n + \frac{(\Delta t)^2}{2}[(1-2\beta)a_n + 2\beta a_{n+1}],$$
$$v_{n+1} = v_n + \Delta t[(1-\gamma)a_n + \gamma a_{n+1}]. \quad \textbf{(2.24)}$$

These relations not only depend upon quantities at time t_n but also on the accelerations at time t_{n+1}. The parameters β and γ are constants which determine the behavior of the integration method. They can be selected by the user. A mathematical analysis of Newmark method for the linear equations of motions leads to the following inequalities which limit the parameter values: $0 \leq \beta \leq 0.5$ and $0 \leq \gamma \leq 1$.

The still unknown accelerations a_{n+1} follow from the spatial discretized form of the linear momentum equation (2.10). By using the approximations for u_{n+1} and v_{n+1}, the accelerations a_{n+1} have to be determined from the nonlinear algebraic equation system

$$(\boldsymbol{M} + \gamma \Delta t \boldsymbol{C})a_{n+1} + \boldsymbol{R}(a_{n+1}, u_n, v_n, a_n) = \boldsymbol{P}_{n+1} - \bar{\boldsymbol{G}}(u_n, v_n, a_n). \quad \textbf{(2.25)}$$

All terms which depend linear upon the displacements, velocities and accelerations at time t_n, when (2.24) is inserted in (2.10), are assembled in the vector $\bar{\boldsymbol{G}}$. Equation (2.25) can be solved by using Newton method. It yields the accelerations a_{n+1}. When these are known, the displacements and velocities follow from (2.24).

Remark: For the parameter choice $\gamma = 0.5$ and $\beta = 0$, the approximations for displacements and velocities of the explicit central difference methods can be derived from (2.24), see also Eq. (2.15).

Often approximations (2.24) are rewritten in such a way that the velocities and accelerations depend upon the displacements

$$\begin{aligned} a_{n+1} &= a_1(u_{n+1} - u_n) - a_2 v_n - a_3 a_n, \\ v_{n+1} &= a_4(u_{n+1} - u_n) + a_5 v_n + a_6 a_n, \end{aligned} \tag{2.26}$$

Here the following constants were introduced

$$a_1 = \frac{1}{\beta \Delta t^2}, \quad a_2 = \frac{1}{\beta \Delta t}, \quad a_3 = \frac{1-2\beta}{2\beta},$$

$$a_4 = \frac{\gamma}{\beta \Delta t}, \quad a_5 = 1 - \frac{\gamma}{\beta}, \quad a_6 = (1 - \frac{\gamma}{2\beta})\Delta t.$$

The insertion of relations (2.26) in to the linear momentum equation (2.10) yields now a nonlinear algebraic equation system for the unknown displacements u_{n+1}:

$$\begin{aligned} G(u_{n+1}) = \boldsymbol{M}[a_1(u_{n+1} - u_n) - a_2 v_n - a_3 a_n] \\ + \boldsymbol{C}[a_4(u_{n+1} - u_n) + a_5 v_n + a_6 a_n] \\ + \boldsymbol{R}(u_{n+1}) - P_{n+1} = 0 \end{aligned} \tag{2.27}$$

Again Newton method can be applied to determine the unknown displacements at time t_{n+1}. This leads with the tangential stiffness matrix

$$K_T\left(u_{n+1}^i\right) = \frac{\partial R}{\partial u_{n+1}}\Big| u_{n+1}^i \tag{2.28}$$

to the iterative scheme (iteration index *i*)

$$\left[a_1\boldsymbol{M} + a_4\boldsymbol{C} + \boldsymbol{K_T}\left(u_{n+1}^i\right)\right] \Delta u_{n+1}^{i+1} = -\boldsymbol{G}\left(u_{n+1}^i\right),$$

$$u_{n+1}^{i+1} = u_{n+1}^i + \Delta u_{n+1}^{i+1} \tag{2.29}$$

which has to be carried out at each time step of the Newmark method. The starting value for this iteration is the last converged displacement vector from the last time step: $u_{n+1}^0 = u_n$. The iteration described in (2.29) is terminated when the convergence criterion of Newton Raphson is fulfilled. See table 1.

The accuracy and stability of the Newmark method can be analyzed for linear dynamical problems. The results stated in Table 2 are due to (Wood, 1990). It can be observed from this table that the Newmark method is optimal for the parameter choice γ = 0.5. In the case of damping free vibrations, the error in the amplitude is equal to zero. This is equivalent to the conversation of energy. Of equal order of accuracy is the method of central differences (γ = 0). However, as already mentioned, this method is not unconditional stable and has a critical time step of $\Delta t \leq T_N/\pi$, with T_N being the smallest period for a given finite element discretization.

Parameter	Error in amplitude $(\boldsymbol{C}=\boldsymbol{0})$	Error in amplitude $(\boldsymbol{C}\neq\boldsymbol{0})$	stability
$\gamma = 0.5$	0	$O(\Delta t^2)$	$\beta \geq 0.25$
$\gamma \neq 0.5$	$O(\Delta t)$	$O(\Delta t)$	$2\beta \geq \gamma \geq 0.5$
$\gamma = 0$	0	$O(\Delta t^2)$	$\Delta t \leq \frac{T_N}{\pi}$

Table 2. Accuracy and stability of the Newmark and Central Difference method

Since the spatial finite element discretization approximates the lower eigenmodes a lot better than the higher one, it is sometimes advantageous to damp the higher modes during the numerical integration process. This often makes in engineering applications sense, since implicit time integration methods are used for problems where the response is governed by the low frequency modes. For the Newmark method, the parameter $\gamma > 0.5$ has to be selected. However, following the results in Table 2, this leads to a loss of accuracy. Due to that reason, modifications of the Newmark method have been proposed which preserve the order $O(\Delta t^2)$ but damp the high frequencies. The method due to Bossak, see (Wood, et al., 1981), is based on a changed discrete equation of motion (2.10)

$$(1-a)\boldsymbol{M}a_{n+1} + a\boldsymbol{M}a_n + \boldsymbol{C}v_{n+1} + \boldsymbol{R}(u_{n+1}) - \boldsymbol{P_{n+1}} = 0 \qquad \textbf{(2.30)}$$

but still uses the approximations (2.24) for displacements and velocities.

The method developed by Hilber (Hilber, et al., 1977) for problems of linear elasto-dynamics applies a different approximation to the equations of motion which weighs the displacements. Its nonlinear extension yields, instead of (2.30), to the system of equations

$$\boldsymbol{M}a_{n+1} + (1-a)[\boldsymbol{C}v_{n+1} - \boldsymbol{P_{n+1}}] + a[\boldsymbol{C}v_n - \boldsymbol{P_n}] + \boldsymbol{R}[(1-a)u_{n+1} + au_n)] = 0 \qquad \textbf{(2.31)}$$

Also, here the displacements and velocities are computed at time t_{n+1} as in the method of Wood et al. (1981) or (Newmark, 1959). The Hilber method damps high frequencies for a parameter choice $0.5 < \alpha < 1$. However, this method needs an evaluation of the vector of internal nodal forces $\boldsymbol{R}$ at time $t_{n+a} = (1-\alpha)\, t_{n+1} + \alpha t_n$, which is not trivial when complex nonlinear constitutive equations dictate the response of a system, and thus history variables have to be considered.

3 Contact Mechanics

3.1 Introduction

Boundary value problems involving contact are of great importance in industrial applications in mechanical and civil engineering. The range of application includes metal forming processes, drilling problems, bearings, crash analysis of cars, car tires or cooling of electronic devices. Other applications are related to biomechanics where human joints, implantats or teeth are of consideration. Due to this variety contact problems are today combined either with large elastic or inelastic deformations including time dependent responses. Thermal coupling might have to be considered, see the cooling of electronic devices, the heat removal within nuclear power plant vessels or thermal insulation of astronautic vehicles. Even stability behavior has to be linked to contact, like wrinkling arising in metal forming problems.

Due to this technical importance a great number of researchers have investigated contact problems. In the ancient Egypt people needed to move large stone blocks to build the pyramids and thus had to overcome the frictional force associated with it. Thus many known researchers in the past have investigated frictional contact problems, amongst them were Da Vinci, Amontons, Newton, Coulomb. Their investigations were based on the assumption of rigid bodies. Starting with the classical analytical work of Hertz (Hertz, 1882)on the elastic contact of two spheres the deformation of the bodies being in contact has been taken into account. However only very few problems involving contact can be solved analytically. Thus for most industrial applications numerical methods have to be applied when the contacting bodies have complex geometries. Due to that the solution of contact problems with finite element methods has a relatively long history. (Wriggers, 1995)

Locating the contact points between two surfaces is an important step in the numerical treatment of contact problems. Moreover, this is one of the major computational costs of contact algorithms both in explicit and implicit computations. Fast and accurate detection of contact is not an easy task and has to be considered in detail. (Yastrebov, 2011)

Contact elements are a kind of "bridge elements" between locally separated but potentially interacting surfaces. Each contact element contains components (nodes, edges, segments or their parts) of both surfaces; the composition of these components depends upon the choice of the contact discretization method. Each contact element has its own vector of unknowns, residual vector and tangential matrix, which are assembled with unknowns, residual vectors and matrices of ordinary structural elements. The set of unknowns and the structure of the residual vector and the tangential matrix are determined by the resolution method. For example, in addition to primal unknowns (e.g. displacement) contact elements may contain dual unknowns (Lagrange multipliers) representing contact stresses.

The discretization of the contact interface is one more challenge in computational contact mechanics. A simple and stable discretization for conforming meshes, i.e. each node on one contacting surface has a corresponding node on the other surface, can be established only in case of small deformations and infinitely small relative sliding. Such a discretization is called *Node to Node*. A less simple but multipurpose discretization implies the creation of contact pairs consisting of a node of one surface and a corresponding segment of the other surface. This approach is known as *Node to Segment* discretization. However, this discretization does not fulfill the so called Babuška - Brezzi conditions and leads to an unstable discretization. Recently, new techniques based on *Segment to Segment* discretization – Nitsche and mortar methods – have been successfully introduced in computational contact mechanics, however, the computer implementation of these methods for a general case presents a real challenge both from algorithmic and technical points of view. Seeking for a stable and relatively simple discretization of the contact interface is still in progress. (Yastrebov, 2011)

3.2 Two-Dimensional Contact Discretization

In contact analyses performed using the finite element method, due to the discretization of the continuum, the contact surfaces are usually divided into segments and a contact algorithm is adopted for the enforcement of the contact boundary conditions. As already mentioned, the contact discretization predetermines the structure of contact elements transferring efforts from one contacting surface to another. Three main types of discretization may be distinguished:

- *Node-to-Node*, NTN
- *Node-to-Segment*, NTS
- *Segment-to-Segment*, STS

3.2.1 *Node-to-Node* Elements

The simplest and the oldest *Node-to-Node* discretization does not allow any finite sliding or large deformations and introduces restrictions on mesh generation. On the other hand it passes the contact patch test – uniform pressure is transferred correctly through the conforming contact interface. The NTN discretization is applicable for linear and quadratic elements in two dimensional case and only to linear elements in three dimensional case. The NTN technique smooths the asymmetry between contacting surfaces. However, the normal vector for each pair of nodes is usually determined according to one of the surfaces.

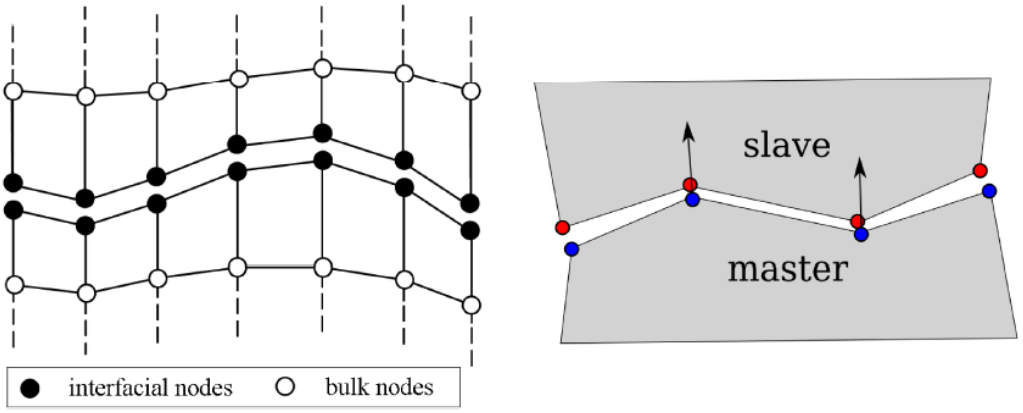


Figure 3. *Graphical representation of Node to Node discretization.*

3.2.2 *Node-to-Segment* Elements

In the most general case, nodes at the contact interface are located at different coordinates, hence *Node-to-Node* algorithms cannot be used even in geometrically linear cases. Among discretization techniques for large deformation contact between surfaces with non-matching meshes, the *Node to-Segment* (*NTS*) algorithm is the most widely used, due to its simplicity, clear physical meaning and flexibility. For these reasons it is currently incorporated in several finite element codes.

The *NTS* algorithm was originally introduced for the enforcement of purely geometric non penetration conditions between the bodies in contact. The non-penetration conditions are enforced by preventing the nodes on one contact surface (the ‘‘slave” one) from penetrating the contact segments on the counterpart contact surface (the ‘‘master” one). However, the same approach can be readily extended to incorporate micro-mechanical contact laws. It can be further extended both to model an interface under tension, incorporating a general cohesive law, and to the context of frictional contact, enforcing either a pure stick-slip condition or a general micromechanical frictional law.

Beside the aforementioned advantages of the *NTS* algorithm, which have determined its wide application in nonlinear finite element simulations, some intrinsic limitations of this approach have

also been pointed out. In the classical formulation, the *NTS* algorithm is unable to deal with some special cases, in which the identification of the master segment related to a slave-node is either ambiguous or impossible. These situations may either disturb the rate of convergence and/or the quality of the solution, or even prevent the achievement of convergence.

This discretization is not stable and does not pass the contact patch test proposed by Taylor and Papadopoulos (Taylor & Papadopoulos, 1991) for nonconforming meshes – a uniform contact pressure cannot be transferred correctly through the contact interface. However, this discretization technique passes this patch test in "double pass" for *Lagrange Multiplier Method*, which means that at each solution step the problem is solved twice: on the first step one assignment of master and slave surfaces is assumed and on the second step the master and slave surfaces are exchanged. A comprehensive discussion of the contact patch test for *NTS* discretization can be found in (Crisfield, 2000), where a new approach combining linear and quadratic shape functions is also suggested. Recently, a modification of the *NTS* discretization has been proposed (Zavarise & De Lorenzis, 2009), which passes the patch test if the *Penalty Method* is used. Besides the drawbacks of this discretization, it is quite simple and robust, that is why it is often implemented in commercial Finite Element Analysis packages.

Basic formulation of *NTS*:

The *NTS* algorithm applies to piecewise straight boundaries of the continuum discretization. Due to the consistent linearization, large displacements of the contact nodes are correctly taken into account, hence the contact element can be coupled with large-displacement and large-deformation formulations of the continuum. As mentioned earlier, the algorithm was originally introduced for the enforcement of purely geometric non-penetration conditions between the bodies in contact. One of the two contacting surfaces has to be selected as the ‘‘slave", and the other one is named the "master". Each "contact" element is composed by a node on the slave surface, called "slave-node", S, and the closest segment on the opposite (master) surface, i.e. the segment which contains its orthogonal projection (Fig. 4). This segment is called ‘‘master" and its (local) end nodes, 1 and 2, are called "master-nodes". Hence, for each contact element three nodes (S, 1, 2) are involved. The non-penetration conditions are enforced by preventing each of the nodes on the slave surface

from penetrating the corresponding master segment. For each slave-node, the normal distance (or gap) between the two surfaces, g_N, (Fig. 4) is defined as the length of the projection on the master segment, usually computed with the scalar product

$$g_N = (\boldsymbol{x_s} - \boldsymbol{x_1}) \cdot \boldsymbol{n}, \tag{3.1}$$

where $\boldsymbol{n}$ is the outward unit vector orthogonal to the master segment, while $\boldsymbol{x_s}$ and $\boldsymbol{x_1}$ are the vectors identifying the current positions, respectively, of nodes S and 1.

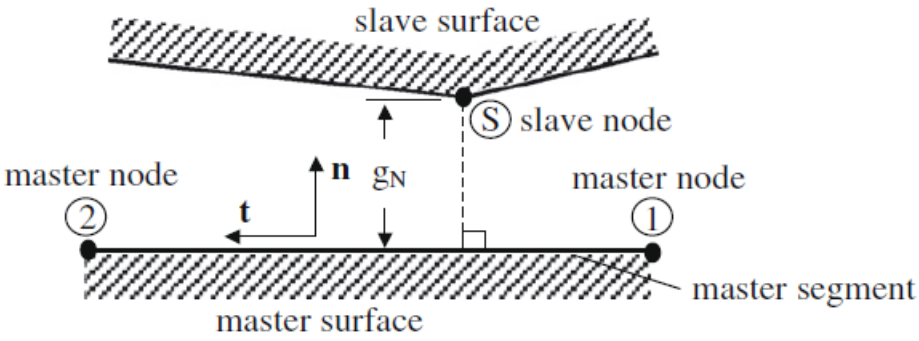


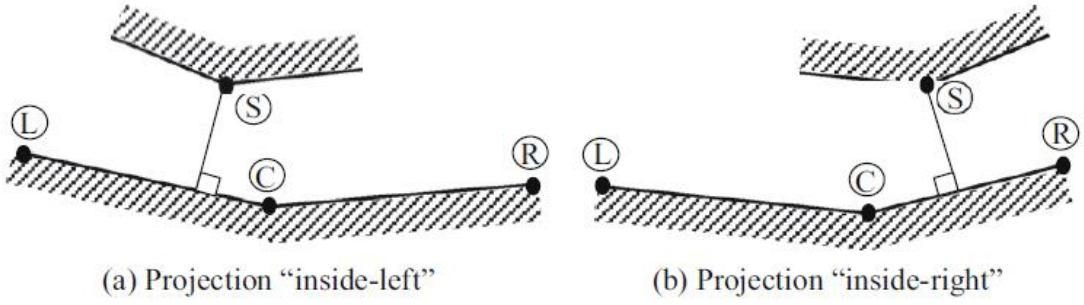
Figure 4. *Node-to-Segment contact element geometry.*

The *NTS* algorithm requires the following steps to be carried out for each slave-node:

- identification of the master segment;
- check of the gap status (open or closed);
- if the gap status is closed (i.e. if the contact element is active), computation of the contact element contribution to the stiffness matrix and to the residual vector.

The identification of the master segment can be carried out in several ways. A well-known one consists in finding first, for each slave-node, the closest node, **C**, on the master surface. Then, the related segments are considered as candidates to determine which of the two contains the normal projection of the node **S**. Typically, four nodes are involved: the slave one, **S**, on the slave surface, the closest node, **C**, on the master surface, and the two nodes adjacent to **C** on the right and left side, respectively **R** and **L** (Fig. 5). The master segment is identified by checking contemporarily the sign of the scalar products

$$g_R = \boldsymbol{CS} \cdot \boldsymbol{CR} \qquad g_L = \boldsymbol{CS} \cdot \boldsymbol{CL} \tag{3.2}$$



(a) Projection "inside-left" (b) Projection "inside-right"

Figure 5. *Regular projection cases.*

All the possible situations are collected in **Table 2.**

Table 2. *Nodal projection cases.*

Projection case	$g_L = \mathbf{CS} \cdot \mathbf{CL}$	$g_R = \mathbf{CS} \cdot \mathbf{CR}$
Inside-right	< 0	$\geqslant 0$
Inside-left	$\geqslant 0$	< 0
Out-of-both	< 0	< 0
In-of-both	$\geqslant 0$	$\geqslant 0$
Out-of-first	< 0	–
Out-of-last	–	< 0

The cases shown in Fig. 5, where the master segment corresponding to the slave-node S is unambiguously the left segment **CL** ("inside-left" case, Fig. 5a) or the right segment **CR** ("inside-right" case, Fig. 5b) are of course the most common ones. However, due to the C^0 continuity of the contact surfaces, other pathologic cases can take place when there are either no segments or more than one segment which contain the normal projection, as shown in Fig. 6 (corner projection cases). Also, it may happen that, due to numerical round-offs, the slave-node projection is just outside the first or the last master segment, as shown in Fig. 7 (out projection cases). These situations may either disturb the rate of convergence and/or the quality of the solution, or even prevent the achievement of convergence.

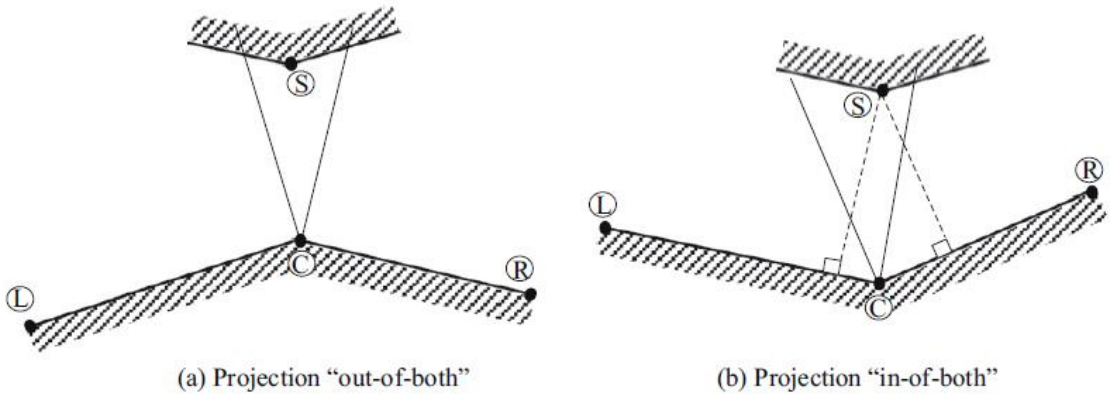


Figure 6. *Corner projection cases.*

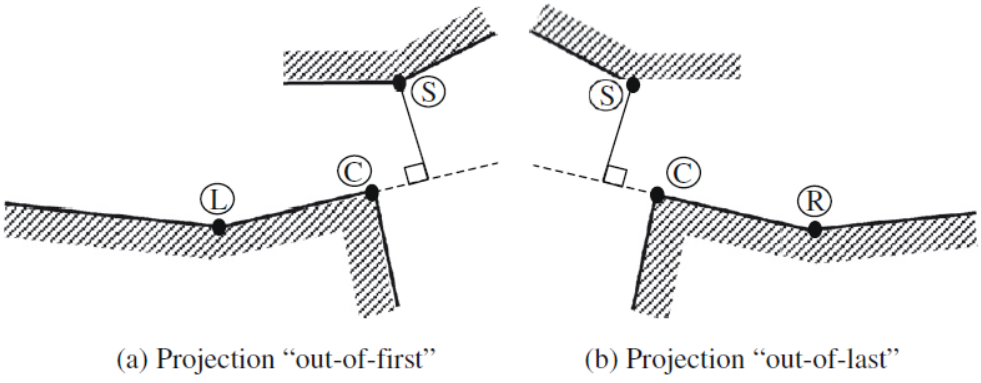


Figure 7. *Out projection cases.*

Once the master segment has been identified, the distance of the slave-node from it can be measured. The gap status (open or closed) is determined by checking the sign of the normal gap, g_N, given by Eq. (3.1), see also Fig. 8.

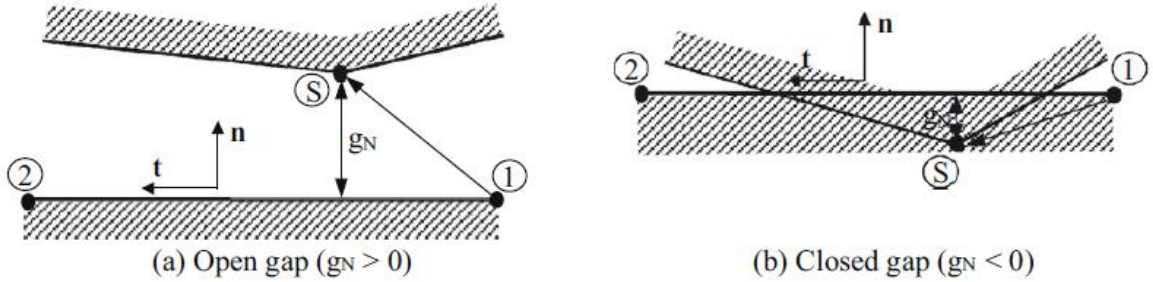


Figure 8. *Gap status.*

From the above description it is evident that the master–slave formulation is inherently unsymmetrical, and this fact does not agree with the physics of the problem. Also, due to the fact that only slave-nodes are checked it may occur that some master-nodes penetrate the slave body, as depicted in Fig. 9. A "two pass" approach, i.e. a double definition of the contact pair, exchanging master and slave surfaces, has been proposed to solve the above cited problem. This strategy can be profitably used for frictionless contact, however it creates problems when frictional contact forces are considered. (Zavarise & De Lorenzis, 2009)

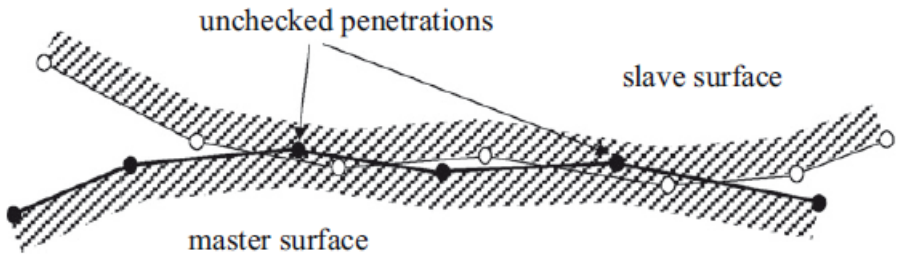


Figure 9. *Unchecked master-node penetrations.*

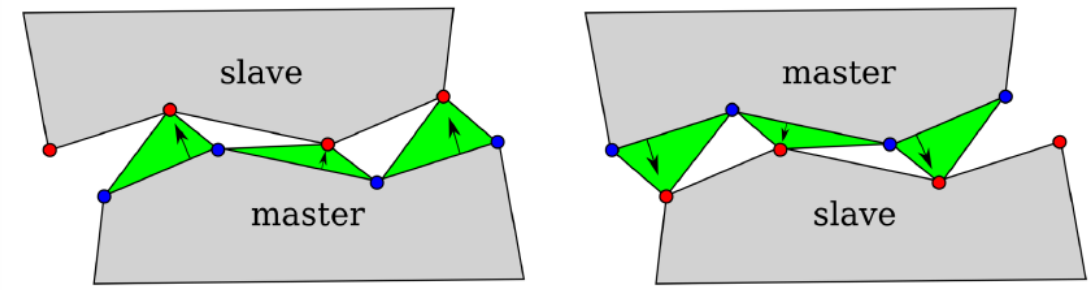


Figure 10. *Graphical representation of Node to Segment discretization for different choices of master and slave.*

Recently, another technique based on a symmetric *Node-to-Segment* discretization, the *Contact Domain Method*, has been proposed in (Oliver, et al., 2009), (Hartmann, et al., 2009). The discretization of the contact interface is based on a full triangulation of the zone between contacting surfaces based on surface nodes. This formulation seems to be rather stable and passes the patch test, but its three dimensional implementation reported in (Oliver, et al., 2010) is not applicable for arbitrary discretization of the contacting surfaces. (Zavarise & De Lorenzis, 2009)

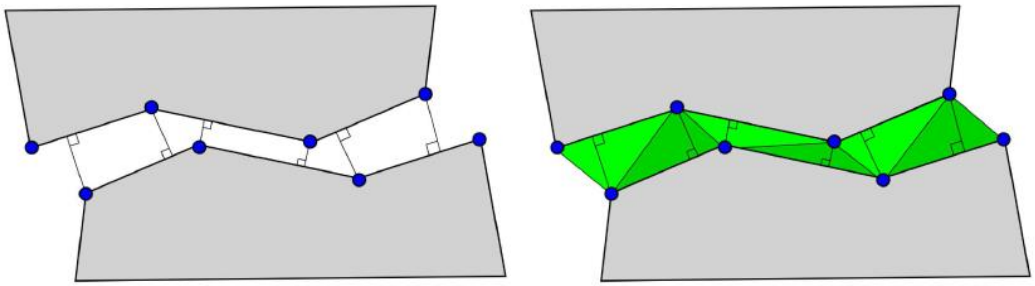


Figure 11. *Graphical representation of the discretization in the Contact Domain Method, including a triangulation of contact interface.*

3.2.3 *Segment-to-Segment* Elements

Segment-to-Segment discretization (Fig. 12) has been first proposed by Simo et al. (Simo, et al., 1985) for the two dimensional case. Recently such a discretization has been efficiently applied to two and three dimensional problems coupled with the mortar method for nonconforming meshes, inspired by the domain decomposition methods. This technique is stable and passes the patch test but its implementation for a general case presents a great challenge, “a nightmare”, according to Tod A. Laursen, one of the authors of the mortar method’s implementation for two and three dimensional both structural and contact problems.

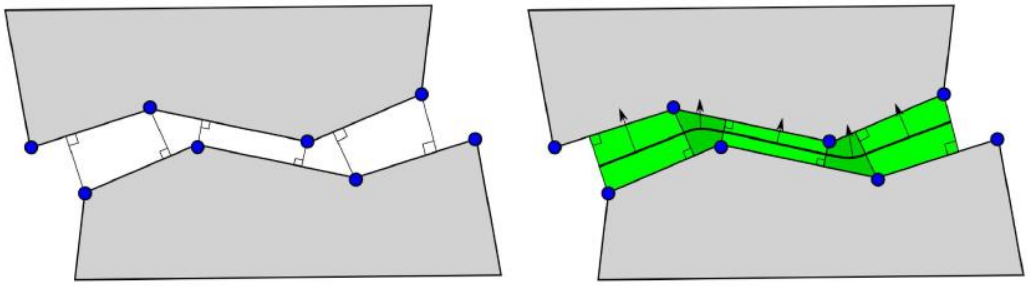


Figure 12. Graphical representation of the Segment-to-Segment discretization, contact elements and an intermediate surface. (Yastrebov, 2011)

3.3 Contact Detection

Roughly, two steps can be distinguished in the contact algorithm: contact detection and resolution. Resolution implies that penetration between contacting solids has to be eliminated by applying repulsive forces to penetrating elements. Consequently the detection phase must determine which elements of the discretized solids are going to penetrate. It is worth mentioning the key difference between contact detection in explicit and implicit resolutions:

- explicit – it is necessary to detect elements that have already penetrated and further apply contact forces;
- implicit – possible penetration has to be known at the beginning of each resolution step in order to include additional degrees of freedom in the problem and to change the residual vector and the stiffness matrix.

The Contact detection is a step preceding all others. The aim of this step is to create contact elements containing the proximal components of both surfaces which may contact on the current solution step. As a consequence the detection algorithm is based on a search for the closest components and presents a particular algorithmic task. The criterion of proximity is either provided by a user or is chosen automatically based on boundary conditions and/or discretization of

contacting surfaces. In order to incorporate contact elements in the resolution cycle, they should be created before a contact occurs and if needed should be removed and recreated at each solution step. Contrary to this scheme, in case of explicit integration, the searching step consists in the detection of penetration, which has already occurred. (Yastrebov, 2011)

In order to treat contact problems, from the programmer's point of view, a standard finite element code has to be complemented by

1. a class governing contact;
2. a contact detection algorithm;
3. a class of contact elements;
4. the corresponding residual vectors and tangential matrices.

The development of numerical methods and the increasing demands on complexity (large deformation, large sliding, self-contact) and size of problems in computational contact mechanics entailed the development of contact detection techniques. As previously mentioned the contact detection presents a purely algorithmic task and is strongly connected with the discretization of the contact interface. For example, in the case of *Node to Node* discretization, the contact detection consists simply in establishing close pairs of nodes: nodes from one surface form pairs with their closest opponents from another surface. Since the Node-to-Node discretization is limited to small deformation and infinitely small sliding, once created contact pairs do not change during the solution steps. *Node to Segment* discretization requires a more elaborated detection procedure: for nodes of one surface (slave) the closest point on the other surface (master) has to be found, the master segment possessing this point complemented by the slave node forms a *Node to Segment* contact element.

This simple detection procedure generates several difficulties. First, the detection of the closest point on the master segments may fail if the slave node is not sufficiently close to the master surface or if the latter is not smooth, which is the case in case of finite element discretization of the surface. The numerical scheme of the closest point detection is based on the seeking for a minimum of the distance function, but on the one hand this minimum does not always exist, and on the other hand there may be several or infinitely many equivalent minimum points. Second, the

detection has to be organized in a smart way. Large contact problems imply a large number of contacting nodes on both surfaces. That is why a simple detection technique, based on a comparison of distances from each slave node to all components of the master surface, leads to an excessively time-consuming algorithm, especially if contact elements must be frequently updated. *Segment to Segment* discretization requires totally different detection algorithms based on surface topologies. Since we confine ourselves to consideration of the *Node to Segment* contact discretization, the questions of detection for other discretization will be omitted.

3.4 Variation of the normal gap.

The variation of the geometrical contact quantities are needed in the weak forms for contact. In this section the variation of the gap in the normal contact direction and the variation of the tangential relative displacement in the contact interface are derived for the different cases.

The variation of the normal gap follows from the non-penetration condition:

$$g_N = (x^2 - \bar{x}^1) \cdot \bar{n}^1 \geq 0 \tag{3.3}$$

As

$$\delta g_N = \delta\{[x^2 - x^1(\bar{\xi}^1, \bar{\xi}^2)] \cdot \boldsymbol{n}^1 (\bar{\xi}^1, \bar{\xi}^2)\}. \tag{3.4}$$

To compute the variation explicitly, we have to consider all terms which depend upon the deformation. In the case of contact, we then have to take into account the projection of point x^2 onto the master surface parameterized by the convective coordinates ξ^1 and ξ^2. This leads to

$$\delta g_N = [\eta^2 - \bar{\eta}^1 - \bar{x}_{,a}{}^1 \delta\xi^a] \cdot \bar{n}^1 + [x^2 - \bar{x}^1] \cdot \delta\bar{n}^1, \tag{3.5}$$

where we have set the test function $\eta^a = \delta x^a$. Equation (3.5) simplifies due the fact that

$\bar{x}_{,a}{}^{1} \cdot \bar{n}^{1} = 0$. Furthermore we have $\bar{n}^{1} \cdot \delta\bar{n}^{1} = 0$. With the definition of the normal, this eliminates the last term in (3.5). Hence we obtain the result

$$\delta g_N = (\eta^2 - \bar{\eta}^1) \cdot \bar{\boldsymbol{n}}^{\mathbf{1}} . \qquad \textbf{(3.6)}$$

Note, however, that we have to start from (3.5) if we want to derive the linearization of the variation of the gap function (3.4), since a function can have a tangent at positions where it is zero.

3.5 Frictionless Case

The problem under analysis is more complicated when friction is present. In that case, not only are the inequality constraints in normal direction present, but there is also a special constitutive behavior in the tangent direction at the contact interface. This is governed by sudden changes of the solution states such that the solution jumps from a state of stick (in which the tangential contact stresses follow as reactions from the stick conditions) to a state of sliding (in which the tangential stresses are computed from a constitutive equation). This special behavior leads to even more mathematical difficulties when questions of existence and uniqueness of frictional contact problems are addressed.

3.6 Treatment of contact constraints.

In this section we shall discuss several different formulations that can be applied to incorporate the contact constraints into the variational formulation. Here frictionless as well as frictional contact formulations are derived. Most standard finite element codes which are able to handle contact problems use either the penalty or the LAGRANGE multiplier method. Each of the methods has its own advantages and disadvantages, which will be discussed in detail in the following. The methods are designed to fulfil the constrain equations in the normal direction in the contact interface. For the tangential part we need in general constitutive relations when stick/slip motion occurs.

Here we concentrate in general on different possibilities to formulate the contact conditions. For a more simple representation we assume that the contact interface is known. Once the contact interface is known we can write the weak form as following:

$$\sum_{\gamma=1}^{2}\left\{\int_{B\gamma}\tau^{\gamma}\cdot grad\,\boldsymbol{\eta}^{\gamma}\,dV-\int_{B\gamma}\bar{\boldsymbol{f}}^{\gamma}\cdot\boldsymbol{\eta}^{\gamma}\,dV-\int_{\Gamma_c^{\gamma}}\bar{\boldsymbol{t}}^{\boldsymbol{\gamma}}\cdot\boldsymbol{\eta}^{\gamma}\,dA\right\}+C_c=0 \tag{3.7}$$

Where $\bar{\boldsymbol{f}}^{\gamma}$ denotes the body force of body B^{γ} and $\bar{\boldsymbol{t}}^{\boldsymbol{\gamma}}$ is the surface traction applied on the boundary of B^{γ}. C_c are contact contributions associated with the active constraint set. $\boldsymbol{\eta}^{\gamma}\in V$ is the so called test function or virtual displacement, which is zero at the boundary $\Gamma_{\varphi}^{\gamma}$ where the deformations are prescribed.

In the case of hyperelastic materials, the starting point to derive equation (3.7) is the minimization of the total energy of the two bodies in contact.

$$\sum_{\gamma=1}^{2}\left\{\int_{B\gamma}W^{\gamma}(C)\,dV-\int_{B\gamma}\bar{\boldsymbol{f}}^{\gamma}\cdot\boldsymbol{\varphi}^{\gamma}\,dV-\int_{\Gamma_c^{\gamma}}t^{\gamma}\cdot\boldsymbol{\varphi}^{\gamma}\,dA\right\}+\Pi_c\rightarrow MIN \tag{3.8}$$

Where W^{γ} (**C**) is the strain energy related to body B^{γ} and φ^{γ} denotes the deformation of the two bodies. The contributions due to the contact constrains are enclosed in Π_c. Note that this formulation is only valid for contact problems which do not include frictional sliding, since the friction process is dissipative and hence the solution becomes path-dependent.

For two bodies in contact we obtain the weak form or the energy related to the interface by assuming that contact is active at the surface Γ_c. Several different variants for the formulation of Π_c and C_c are discussed below:

1. The LAGRANGE multiplier method.
2. The penalty method.

3. The method of direct elimination of the geometrical contact constraints.
4. The formulation of constitutive equations in the contact interface.
5. The NITSCHE method, which enforces geometrical constraints in a weak sense.
6. The perturbed LAGRANGE formulation which combines (1) and (2) in a mixed form.
7. The barrier method.
8. The augmented LAGRANGE method.
9. The cross-constraint method which combines the penalty and barrier methods.

This large variety of formulations also reflects the large number of different algorithms which have so far been developed to solve contact problems.

3.7 Penalty Method

In this formulation a penalty term due to the constraint condition (3.10) is added to Π_c in (3.8) as follows:

$$\Pi_c^P = \frac{1}{2}\int_{\Gamma_c}(\epsilon_N \overline{g_N}^{\,2} + \epsilon_T \boldsymbol{g_T}\cdot\boldsymbol{g_T})\,dA\,, \qquad \epsilon_N, \epsilon_T > 0 \tag{3.9}$$

ϵ_N and ϵ_T represent the penalty parameters. The penalty term Π_c^P is only added for active constraints which are defined by the following penetration function $\overline{g_N}$

$$\overline{g_N} = \begin{cases} (x^2 - \bar{x}^1)\cdot\bar{n}^1 \; if \;\; (x^2 - \bar{x}^1)\cdot\bar{n}^1 < 0 \\ 0 \qquad\qquad otherwise \end{cases} \tag{3.10}$$

and has to be formulated for normal and tangential contacts, the latter in the case of stick. The variation of (3.9) yields

$$C_c^P = \frac{1}{2}\int_{\Gamma_c}(\epsilon_N\,\overline{g_N}\,\delta\overline{g_N} + \epsilon_T\,\boldsymbol{g_T}\cdot\delta\boldsymbol{g_T})\,dA\,,\ \epsilon_N,\epsilon_T > 0 \qquad \textbf{(3.11)}$$

It can be shown, that the solution of the Lagrange multiplier method is recovered from this formulation for $\epsilon_N \to\infty$ and $\epsilon_T \to\infty$; however, large numbers for ϵ_N and ϵ_T will lead to an ill-conditioned numerical problem. As in the Lagrange multiplier method, we have to distinguish between pure stick in the contact interface which yields (3.11), and the slip condition which leads to

$$C_c^{slip} = \frac{1}{2}\int_{\Gamma_c}(\epsilon_N\,\overline{g_N}\,\delta\overline{g_N} + t_T\delta\boldsymbol{g_T})\,dA\,,\ \epsilon_N > 0 \qquad \textbf{(3.12)}$$

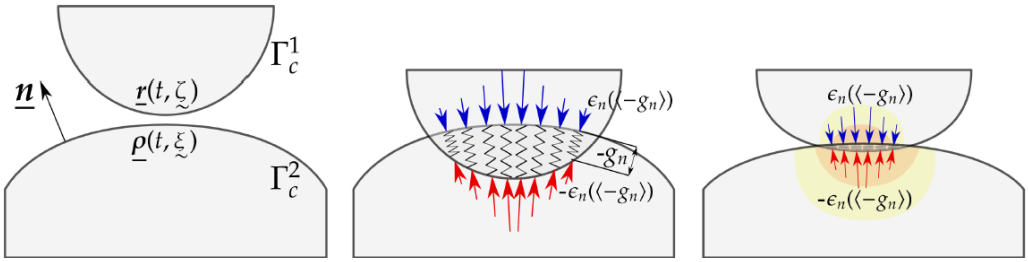


Figure 13. *Spring interpretation of the penalty method.*

On the above figure: **Left** - undeformed configuration; **Middle** - configuration after penetration $g_n\,(\xi)$, which in turn results in a contact pressure $\sigma_{n\,(\xi)} = \epsilon_n(-\,g_n(\xi))$ shown with red arrows on the master and in blue on the slave, contact pressure results in decreasing the penetration due to deformation and results in a **right** equilibrium state.

4 Finite Element Formulation

4.1 Introduction

The integral over a volume $\boldsymbol{V}$ is written as a sum of integrals over smaller volumes, which collectively constitute the whole volume. Such a small volume $\boldsymbol{V^e}$ is called an element. Subdividing the volume implies that also the surface with area $\boldsymbol{A}$ is subdivided in element surfaces (faces) with area $\boldsymbol{A^e}$.

The solid (or continuum) elements can be used for linear analysis and for complex nonlinear analyses involving contact, plasticity, and large deformations. They are available for stress, heat transfer, acoustic, coupled thermal-stress, coupled pore fluid-stress, piezoelectric, and coupled thermal-electrical analyses. First-order (linear) interpolation elements and second-order (quadratic) interpolation elements can be used in one, two, or three dimensions. Triangles and quadrilaterals are available in two dimensions; and tetrahedra, triangular prisms, and hexahedra (bricks) are provided in three dimensions. Modified second-order triangular and tetrahedral elements are also provided. In addition, reduced-integration, hybrid, and incompatible mode elements are available. Solid elements are more accurate if not distorted, particularly for quadrilaterals and hexahedra. The triangular and tetrahedral elements are less sensitive to distortion.

Second-order elements provide higher accuracy than first-order elements for smooth problems that do not involve complex contact conditions, impact, or severe element distortions. They capture stress concentrations more effectively and are better for modeling geometric features: they can model a curved surface with fewer elements. Finally, second-order elements are very effective in bending-dominated problems. First-order triangular and tetrahedral elements should be avoided as much as possible in stress analysis problems; the elements are overly stiff and exhibit slow convergence with mesh refinement, which is especially a problem with first-order tetrahedral elements. If they are required, an extremely fine mesh may be needed to obtain results of sufficient accuracy. The modified triangular and tetrahedral elements should be used in contact problems

with the default hard contact relationship because the contact forces are consistent with the direction of contact. These elements also perform better in analyses involving impact (because they have a lumped mass matrix), in analyses involving nearly incompressible material response, and in analyses requiring large element distortions, such as the simulation of certain manufacturing processes or the response of rubber components.

Reduced integration uses a lower-order integration to form the element stiffness. The mass matrix and distributed loadings use full integration. Reduced integration reduces running time, especially in three dimensions. The elements with reduced integration are also referred to as uniform strain or centroid strain elements with hourglass control. Second-order reduced-integration elements generally yield more accurate results than the corresponding fully integrated elements. However, for first-order elements the accuracy achieved with full versus reduced integration is largely dependent on the nature of the problem.

4.2 Isoparametric elements

Each point of a three-dimensional element can be identified with three local coordinates $\{\xi_1, \xi_2, \xi_3\}$. In two dimensions we need two and in one dimension only one local coordinate. The real geometry of the element can be considered to be the result of a deformation from the original cubic, square or line element with (side) length 2. The deformation can be described with a deformation matrix, which is called the Jacobian matrix $\boldsymbol{J}$. The determinant of this matrix relates two infinitesimal volumes, areas or lengths of both element representations.

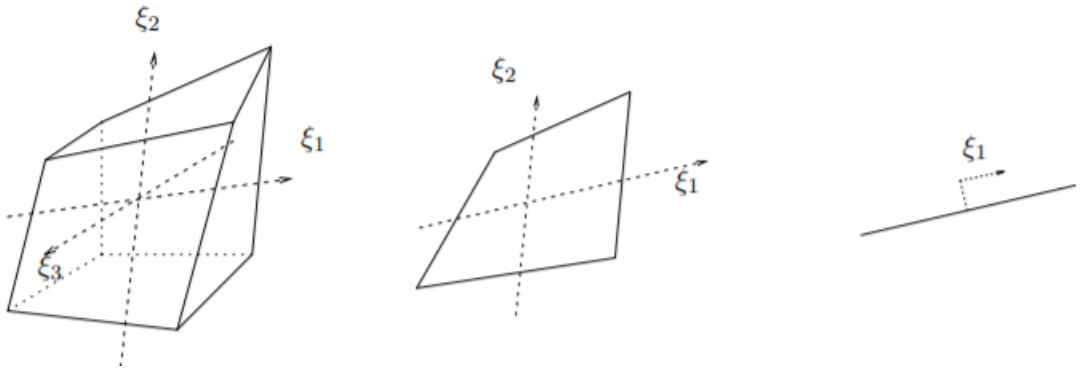


Figure 14. *Isoparametric elements.*

Isoparametric (local) coordinates $\{\xi_1, \xi_2, \xi_3\}$; $-1 \leq \xi_i \leq 1 \quad i = 1, 2, 3$

Jacobian matrix $J = (\nabla_\xi x^T)^T$; $dV^e = det(J)\, d\xi_1 d\xi_2 d\xi_3$

4.3 Four noded linear element

When the elements have a simple shape, e.g. a six-sided volume, the shape and thus volume is known when the position of a discrete number of edge points is known. These points are the element nodal points. For a cube with plane faces, eight corner points are needed. In two dimensions quadrilaterals with straight edges can be used, where four corner nodes describe the shape. The position of an internal point of the element can be expressed in the position of the nodal points. This interpolation is done with so-called shape- or interpolation functions, which are a function of local element coordinates ξ_i, i = 1, 2, 3, which assume values between -1 and +1. The value of the unknown quantity – here the displacement vector $\vec{u}$ – in an arbitrary point of the element, can also be interpolated between the values of that quantity in the element nodes.

Besides $\vec{x}$ and $\vec{u}$, the weighting function $\vec{w}$ also needs to be interpolated between nodal values. When this interpolation is the same as that for the displacement, the so-called Galerkin procedure is followed, which is generally the case for simple elements, considered here.

The gradient of $\vec{u}$, and $\vec{w}$ also has to be elaborated, and can be written as the product of a column which contains the derivatives of the interpolation functions, and the column with nodal components of $\vec{x}$ and $\vec{u}$.

Finally, everything is substituted in the weighted residual integral. The volume integral in the left hand side is the element stiffness matrix K^e. The integrals in the right hand side represent the external load and are summarized in the column f^e.

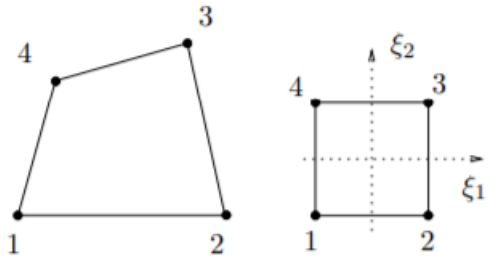


Figure 15. *Quadrilateral element with four corner node.*

Interpolation functions:

$$N^1 = \frac{1}{4}(\xi_1 - 1)(\xi_2 - 1) \quad ; \quad N^2 = -\frac{1}{4}(\xi_1 + 1)(\xi_2 - 1)$$

$$N^3 = \frac{1}{4}(\xi_1 + 1)(\xi_2 + 1) \quad ; \quad N^4 = -\frac{1}{4}(\xi_1 - 1)(\xi_2 + 1) \qquad \textbf{(4.1)}$$

Position: $\vec{x} = N^1\left(\underline{\xi}\right)\vec{x}^1 + N^2\left(\underline{\xi}\right)\vec{x}^2 + N^3\left(\underline{\xi}\right)\vec{x}^3 + N^4\left(\underline{\xi}\right)\vec{x}^4$ **(4.2)**

Displacement $\vec{\boldsymbol{u}} = \underline{N}^T\left(\underline{\xi}\right)\underline{\vec{u}}^e$ **(4.3)**

INTEGRATION

Calculating the element stiffness matrix K^e and the external loads f^e implies the evaluation of an integral over the element volume V^e and the element surface A^e. This integration is done numerically, using a fixed set of nip Gauss-points, which have s specific location in the element.

The value of the integrand is calculated in each Gauss-point and multiplied with a Gauss-point-specific weighting factor c^{ip} and added.

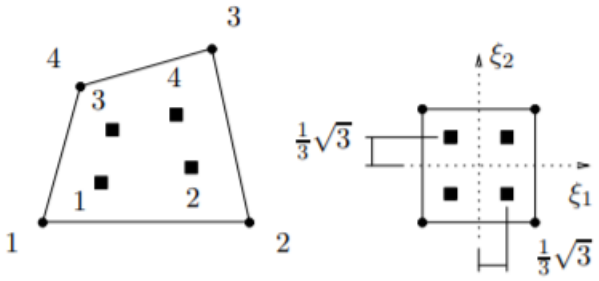


Figure 16. *Integration points in a 4 noded quadrilateral element.*

$$\int_{V^e} g(x_1, x_2)\, dV^e = \int_{\xi_1=-1}^{1} \int_{\xi_2=-1}^{1} f(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 = \sum_{ip=1}^{4} c^{ip}\, f(\xi_1^{ip}, \xi_2^{ip}) \qquad \textbf{(4.4)}$$

The weighted residual contribution of all elements have to be collected into the total weighted residual integral. This means that all elements are connected or assembled. This assembling is an administrative procedure. All the element matrices and columns are placed at appropriate locations into the structural or global stiffness matrix **K** and the load column f^e.

FINITE ELEMENT EQUATION

Relations (4.3) can be rewritten in a matrix form as follows:

$$\{u\} = [N]\{q\} \qquad \textbf{(4.5)}$$

where, q is the nodal displacement vector.

Strains can be determined through displacements at nodal points:

$$\{\varepsilon\} = [B]\{q\} \tag{4.6}$$

$$\{B\} = [D]\{N\} \tag{4.7}$$

$$[B_i] = \begin{bmatrix} \partial N_i/\partial x & 0 & 0 \\ 0 & \partial N_i/\partial y & 0 \\ 0 & 0 & \partial N_i/\partial z \\ \partial N_i/\partial y & \partial N_i/\partial x & 0 \\ 0 & \partial N_i/\partial z & \partial N_i/\partial y \\ \partial N_i/\partial z & 0 & \partial N_i/\partial x \end{bmatrix} \tag{4.8}$$

Weak Form - Principle of Virtual Displacement

$$\int_B \delta\{\varepsilon\}^T \cdot \{\boldsymbol{\sigma}\}\, dV = \int_B \delta \boldsymbol{u}^T \cdot \boldsymbol{f}\, dV + \int_{\partial B_t} \delta \boldsymbol{u}^T \cdot \boldsymbol{t}\, dA \tag{4.9}$$

From relations (4.5) and (4.6) the variation of displacement and strain yield:

$$\delta\{u\} = [N]\delta\{q\} \tag{4.10}$$

$$\delta\{\varepsilon\} = [B]\delta\{q\} \tag{4.11}$$

So, the equation (4.9) can be written as:

$$\int_B \delta \boldsymbol{q}^T \boldsymbol{B}^T \cdot \{\boldsymbol{\sigma}\}\, dV = \int_B \delta \boldsymbol{q}^T \boldsymbol{N}^T \cdot \boldsymbol{f}\, dV + \int_{\partial B_t} \delta \boldsymbol{q}^T \boldsymbol{N}^T \cdot \boldsymbol{t}\, dA \tag{4.12}$$

FE Matrix equation $F_{INT} = F_{EXT}$

$$F_{EXT} = \int_B N^T \boldsymbol{f}\, dV + \int_{\partial B_i} N^T \boldsymbol{t} dA$$

$$F_{INT} = \int_B B^T \{\boldsymbol{\sigma}\}\, dV$$

Grouping elements, Gauss points l, weights W_l :

$$F_{INT} = \bigcup_{e=1}^{n_{el}} f_{INT}^e \quad f_{INT}^e = \sum_{l=1}^{n_{INT}} (B_l^e)^T \boldsymbol{\sigma}_l W_l$$
$$F_{EXT} = \bigcup_{e=1}^{n_{el}} f_{EXT}^e \quad f_{EXT}^e = \sum_{l=1}^{n_{INT}} (N_l^e)^T \boldsymbol{f}_l W_l + \sum_{l=1}^{n_{INT}} (N_l^e)^T \boldsymbol{t}_l W_l \qquad \textbf{(4.13)}$$

Residual form: $\boldsymbol{R(r) = F_{EXT} - F_{INT}(r)}$ **(4.14)**

Solutions are found at discrete time points $t_n, t_{n+1}, \ldots$

Convergence occurs when: $\left\| \boldsymbol{R}(\boldsymbol{r}_{\boldsymbol{n+1}}^{(\boldsymbol{i})}) \right\| \leq \varepsilon \left\| \boldsymbol{R}(\boldsymbol{r}_{\boldsymbol{n+1}}^{(\boldsymbol{0})}) \right\}$ ε=*tol.par* **(4.15)**

5 Application

5.1 Contact – Impact Problem

In this chapter, a case of an impact between two deformable bodies will be analyzed. Two bodies of different geometry and material's properties are modelled using C# programming and various simulations will be run. Both implicit and explicit methods will be employed and the results of the two methods will be compared. Also, the same model will be created and run in "Abaqus" software and the results will be compared as well.

For simplicity reasons, we examine 2d applications, since a relevant 3d example would be very complex, with demanding coding and time consuming.

5.2 Pre Processing - Model discretization

5.2.1 Geometry

As having been referred above, two deformable bodies will be discretized.

The geometry of the two bodies is:

- Body 1: Ring of 0.5 meter external diameter and 0.3 meter internal diameter.
- Body 2: Rectangular of 3 meters width and 1 meter height.

The units that could be used for the consistency of the code and analysis are shown on the table 3.

Table 3. Consistent units

Quantity	SI	SI (mm)	US Unit (ft)	US Unit (inch)
Length	m	mm	ft	in
Force	N	N	lbf	lbf
Mass	kg	tonne (10^3 kg)	slug	lbf s^2/in
Time	s	s	s	s
Stress	Pa (N/m^2)	MPa (N/mm^2)	lbf/ft^2	psi (lbf/in^2)
Energy	J	mJ (10^{-3} J)	ft lbf	in lbf
Density	kg/m^3	tonne/mm^3	slug/ft^3	lbf s^2/in^4

5.2.2 Meshing

An algorithm that allocates the nodes of the elements on the desired geometry of the interacting bodies is developed. So, for "Body 1", 108 nodes will be created, forming 3 circles with diameters 0.5, 0.4 and 0.3 meters diameter. The relevant four noded quadrilateral elements that are being created by connecting the nodes are 72, as shown in Fig.17**.**

For "Body 2", 11 x 31 = 341 nodes will be created, forming a rectangular of 3 meters width and 1 meter height. The relevant four noded elements are 300, see Fig.17.

Linear 4 noded elements of first order and reduced integration will be used for the analysis.

CONTACT ELEMENTS

As having been mentioned in chapter 3.2, contact finite elements will be needed to run the analysis and simulate the contact of the bodies. A thorough description of the role of these elements has already preceded. *Node to Segment* elements will be used. The potential interactive nodes of "Body 1" will be the slave nodes and nodes from the upper surface of the rectangular "Body 2" will be the master nodes. See elements in red in Fig.17.

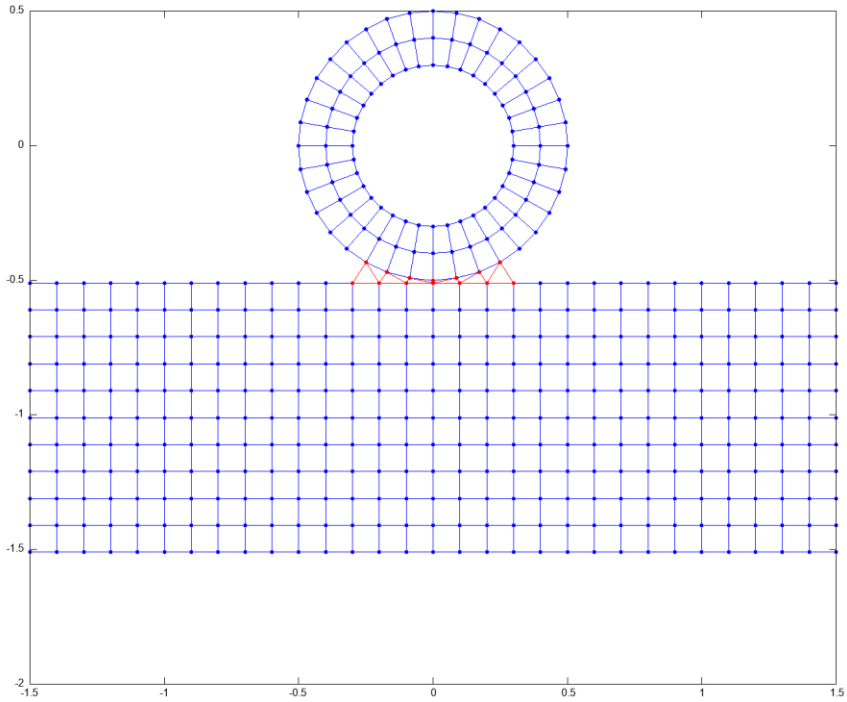


Figure 17. *Model discretization – Initial Geometry.*

5.2.3 Material Properties

The material properties of the two interacting bodies will be the following:

- Body 1 – Ring: Young's Modulus 1200 GPa, Density 3500 kg/m^3
- Body 2 – Rectangular : Young's Modulus 2 GPa, Density 1200 kg/m^3

The above properties have been assigned to the two bodies, as a case of a body of extremely high stiffness that impacts with a body of approximately the stiffness of a soft plastic material.

5.3 Processing – Solver

Before proceeding to solution, the appropriate boundary conditions and loads must be applied to the model. For the case under studying, nodes of the bottom boundary of the rectangular "Body 2" are being constrained in both directions ("x" and "y" axis). Also, nodes of the right and left boundaries are being constrained in "x" direction.

Initial velocity of different values are assigned to all the nodes of the ring – "Body 2" for the cases that are being studied. To be precise, three different cases with 20, 40 and 60 m/s velocity respectively are simulated, using both implicit and explicit methods as have been described in Chapter 2.

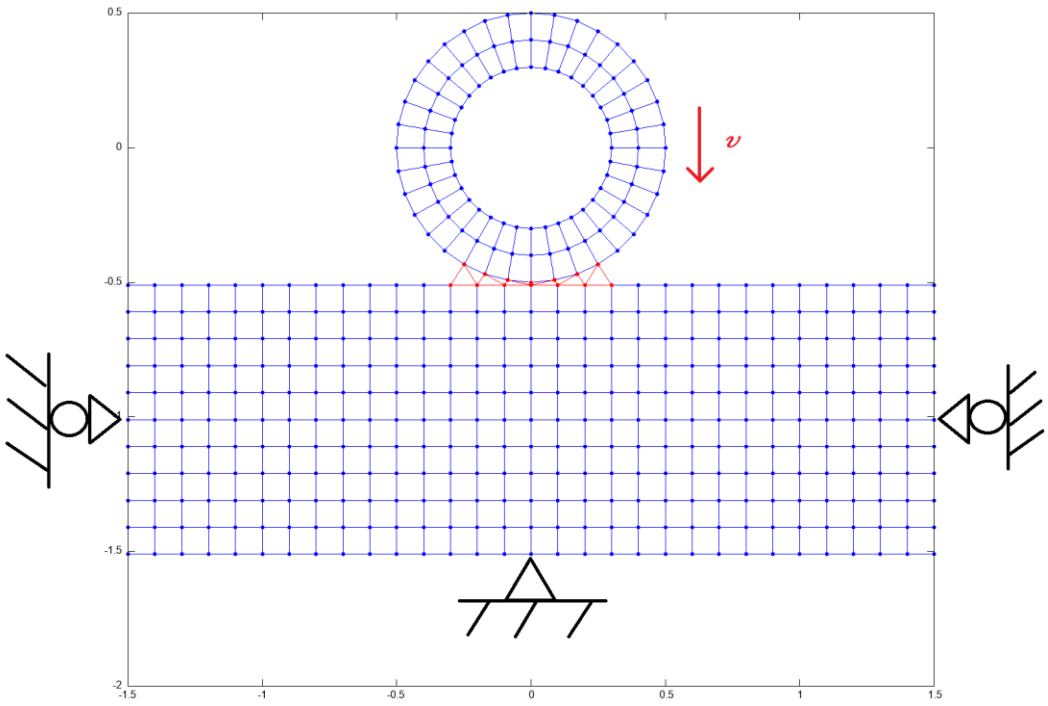


Figure 18. *Boundary conditions.*

5.4 Post Processing - Results

The results from the aforementioned analyses that have been run, are collected and shown in this section.

In figures 18 and 19, visualization of the positions of the nodes at the initial and final positions have been done using “Gnuplot”.

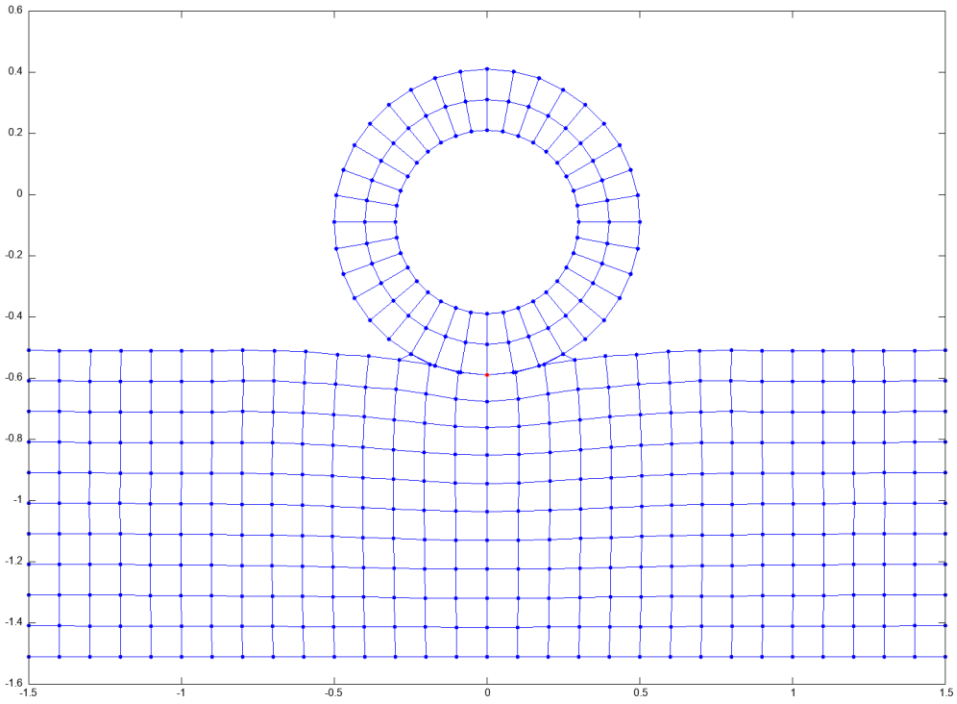


Figure 19. *Final geometry after 6 ms.*

For every simulation, equivalent "Mises" Stresses have been exported at the time of the maximum deformation of rectangular body ("Body 2"). The vertical displacement of the central node of the upper surface of the "Body 2" that comes into contact has been inspected for every time step of the analysis. The relevant values are shown in table 4.

Table 4. *Results – Displacement of node under inspection.*

Velocity of "Body 1"(m/s)	**Vertical Displacement of Node under Inspection (m)**	**Time of maximum Displacement (s - time step)**
Implicit Method		
20	-0.0513799	$0.00471 – 314^{th}$
40	-0.0952157	$0.00414 – 276^{th}$
60	-0.1340184	$0.00384 -256^{th}$
Explicit Method		
20	-0.0514367	$0.00466 – 2330^{th}$
40	-0.0954310	$0.00417 – 2085^{th}$
60	-0.1345468	$0.00385 – 1925^{th}$

Contours of "Mises" Stresses, Volumetric and Shear Strains for various cases using "Matlab" are shown below:

- Implicit scheme / Newmark method – 6 ms Total time/ 400 Time steps – Velocity 20 m/s

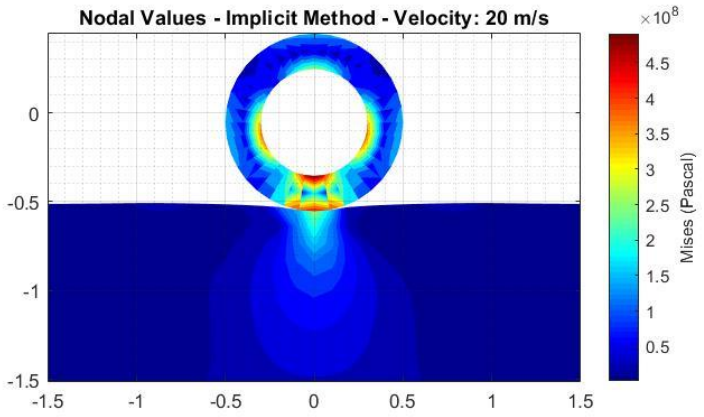


Figure 20. *"Mises" Stresses, Implicit Method - Velocity 20 m/s, Nodal Values, at 0.00471 s (314th time step).*

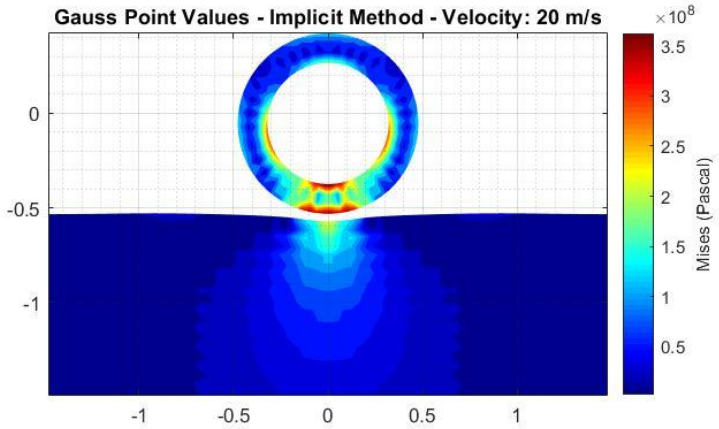


Figure 21. *"Mises" Stresses, Implicit Method - Velocity 20 m/s, Gauss Point Values, at 0.00471 s (314th time step).*

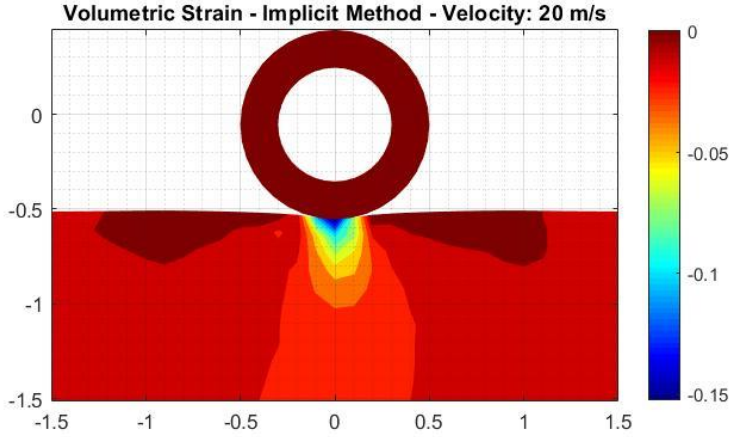


Figure 22. *Volumetric Strain, Implicit Method - Velocity 20 m/s, at 0.00471 s (314th time step).*

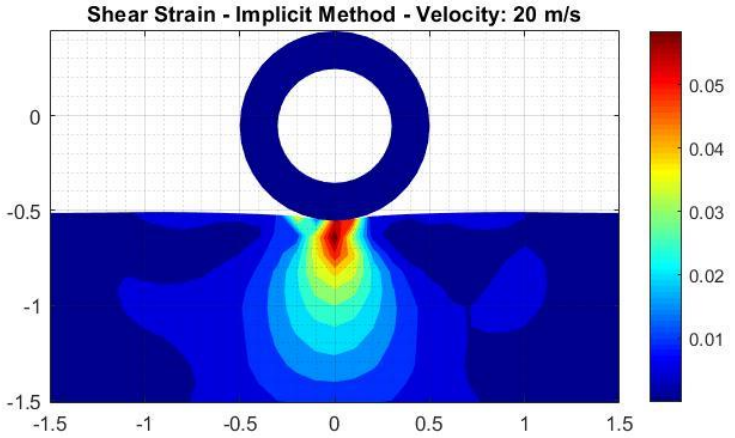


Figure 23. *Shear Strain, Implicit Method - Velocity 20 m/s, at 0.00471 s (314th time step).*

- Implicit scheme / Newmark method– 6 ms Total time/ 400 Time steps – Velocity 40 m/s

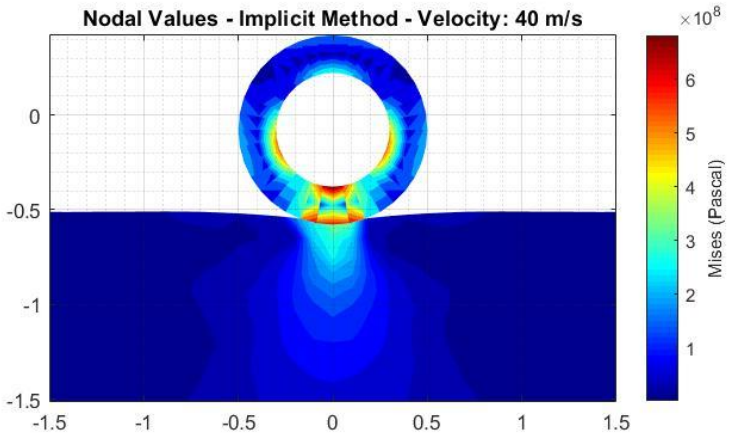


Figure 24. *"Mises" Stresses, Implicit Method - Velocity 40 m/s, Nodal Values, at 0.00414 s (276th time step).*

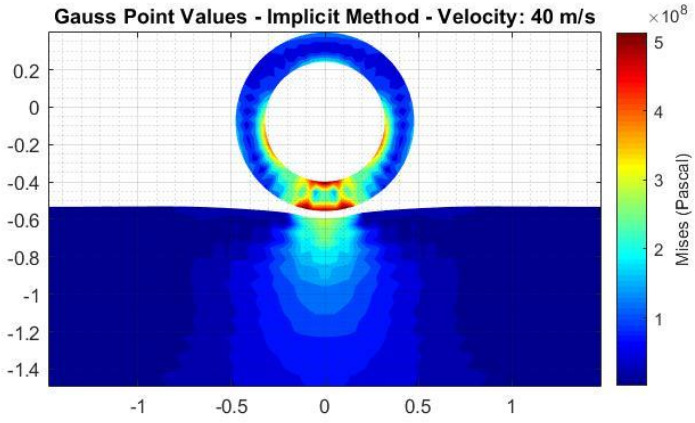


Figure 25. *"Mises" Stresses, Implicit Method - Velocity 40 m/s, Gauss Points Values, at 0.00414 s (276th time step).*

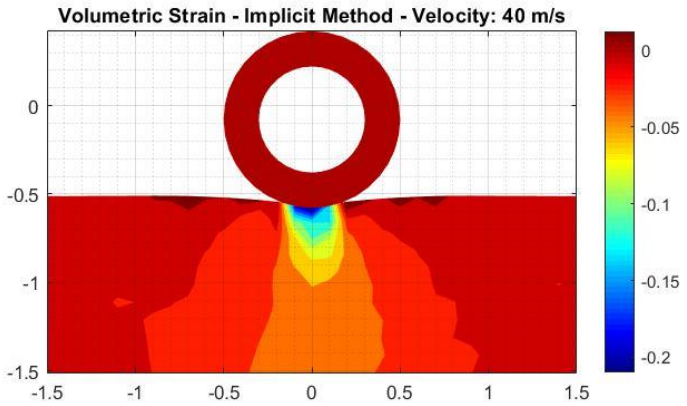


Figure 26. *Volumetric Strain, Implicit Method - Velocity 40 m/s, at 0.00414 s (276th time step).*

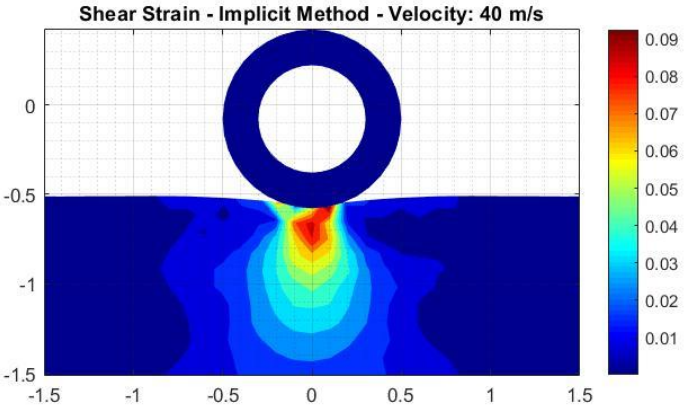


Figure 27. *Shear Strain, Implicit Method - Velocity 40 m/s, at 0.00414 s (276th time step).*

- Implicit scheme / Newmark method– 6 ms Total time/ 400 Time steps – Velocity 60 m/s

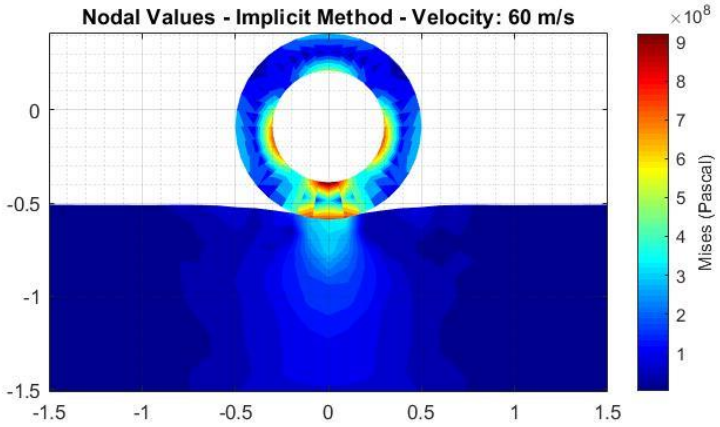


Figure 28. *"Mises" Stresses, Implicit Method - Velocity 60 m/s, Nodal Values, at 0.00384 s (256th time step).*

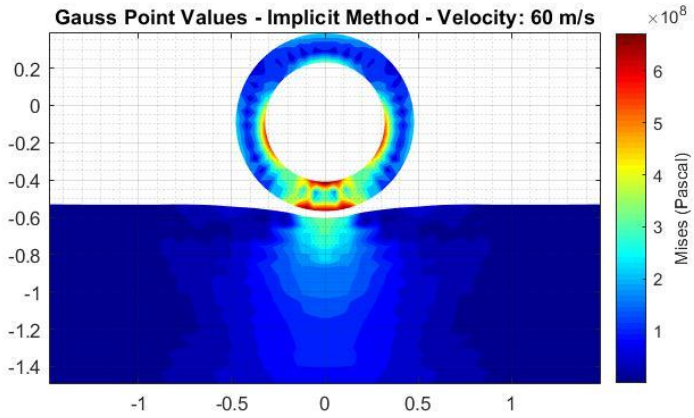


Figure 29. *Mises Stresses, Implicit Method - Velocity 60 m/s, Gauss Point Values, at 0.00384 s (256th time step).*

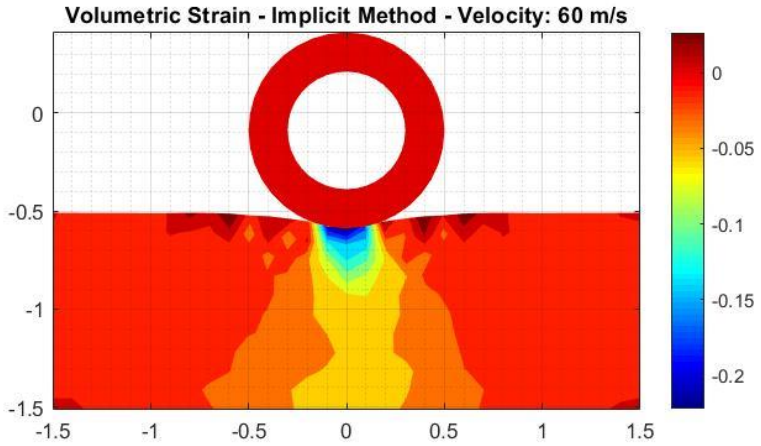


Figure 30. *Volumetric Strain, Implicit Method - Velocity 60 m/s, at 0.00384 s (256th time step).*

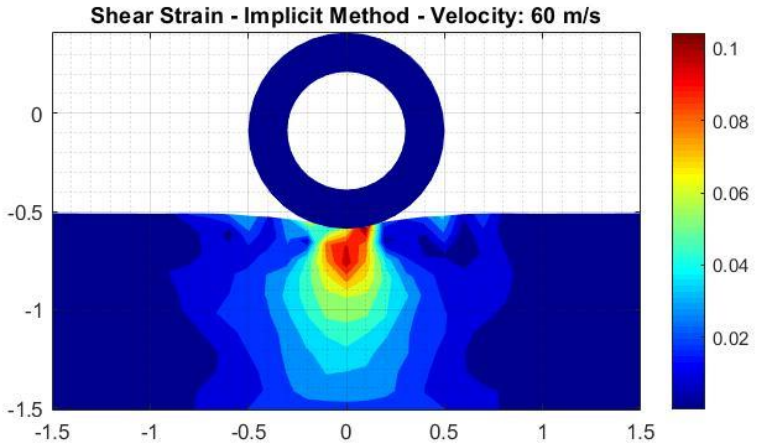


Figure 31. *Volumetric Strain, Implicit Method - Velocity 60 m/s, at 0.00384 s (256th time step).*

- Explicit scheme / Central Difference Method – 6 ms Total time/ 3000 Time steps – Velocity 20 m/s

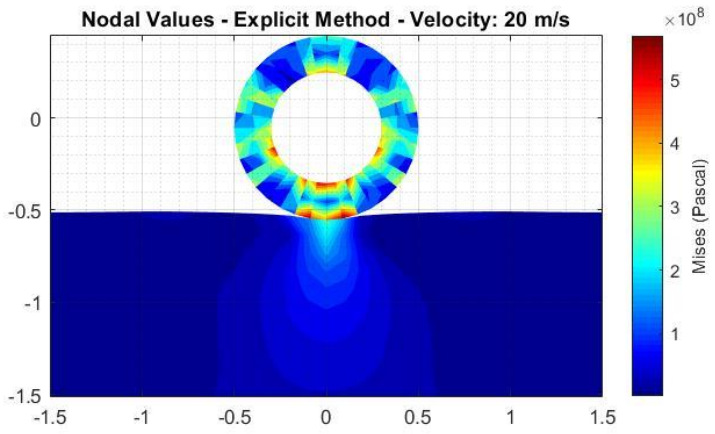


Figure 32. "Mises" Stresses, Explicit Method - Velocity 20 m/s, Nodal Values, at 0.00466 s (2330th time step).

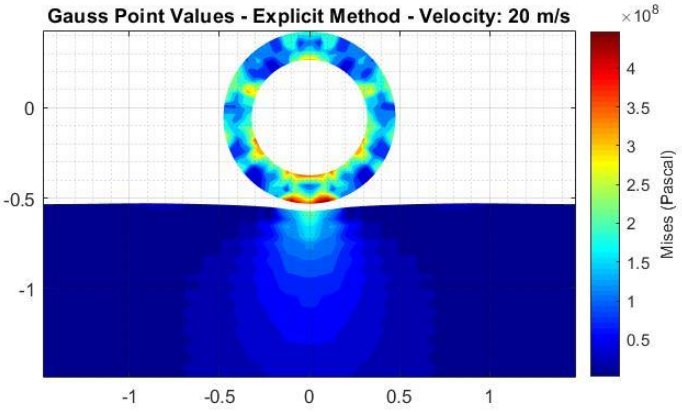


Figure 33. *"Mises" Stresses, Explicit Method - Velocity 20 m/s, Gauss Points Values,* at 0.00466 s (2330th time step).

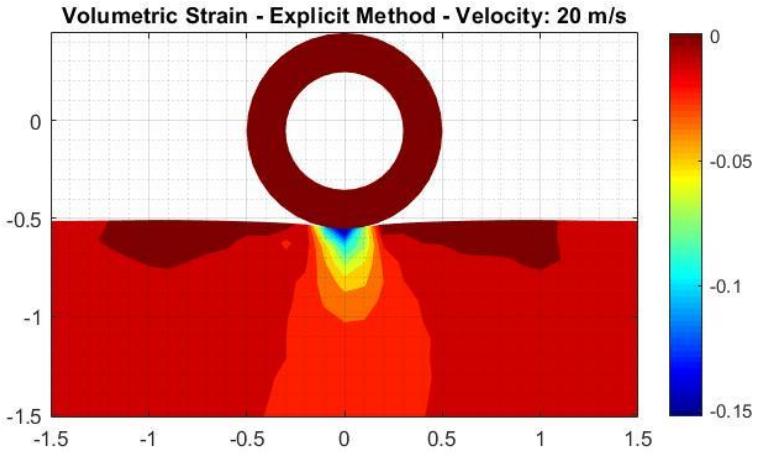


Figure 34. Volumetric Strain, Explicit Method - Velocity 20 m/s, at 0.00466 s (2330th time step).

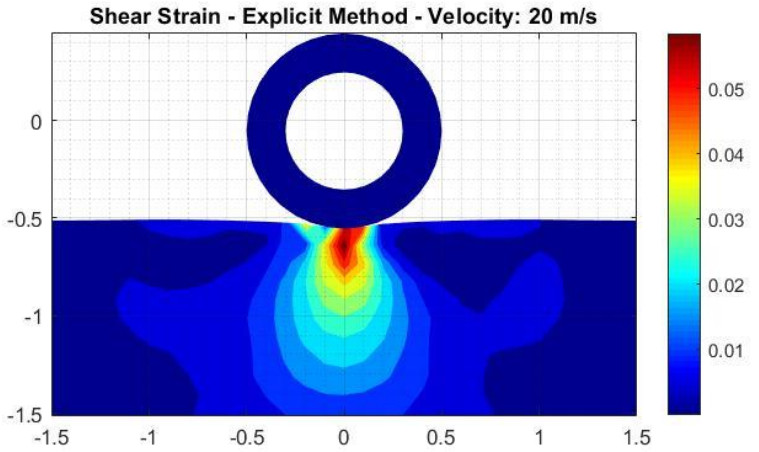


Figure 35. Shear Strain, Explicit Method - Velocity 20 m/s, at 0.00466 s (2330th time step).

- Explicit scheme / Central Difference Method – 6 ms Total time/ 3000 Time steps – Velocity 40 m/s

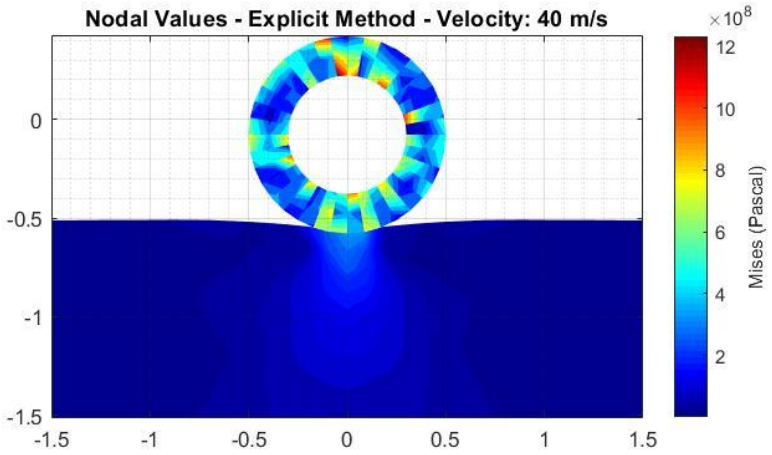


Figure 36. *"Mises" Stresses, Explicit Method - Velocity 40 m/s, Nodal Values, at 0.00417 s (2085th time step).*

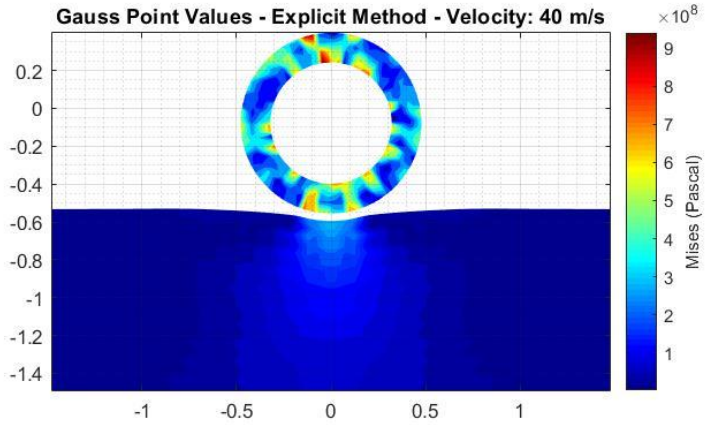


Figure 37. *"Mises" Stresses, Explicit Method - Velocity 40 m/s, Gauss Points Values, at 0.00417 s (2085th time step).*

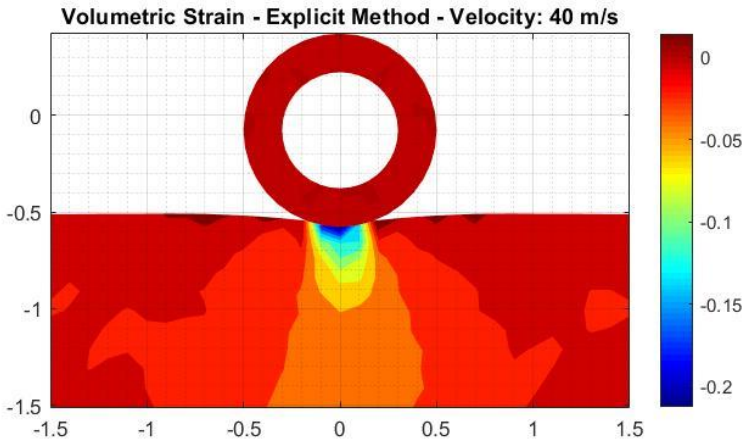


Figure 38. *Volumetric Strain, Explicit Method - Velocity 40 m/s, at 0.00417 s (2085th time step).*

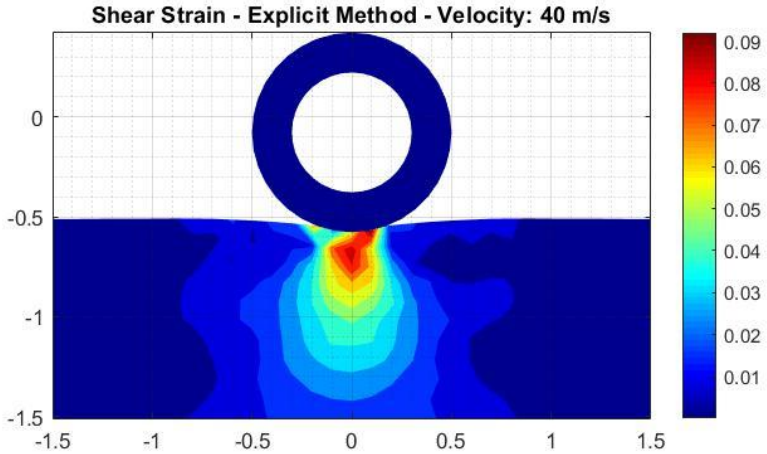


Figure 39. *Shear Strain, Explicit Method - Velocity 40 m/s, at 0.00417 s (2085th time step).*

- Explicit scheme / Central Difference Method – 6 ms Total time/ 3000 Time steps – Velocity 60 m/s

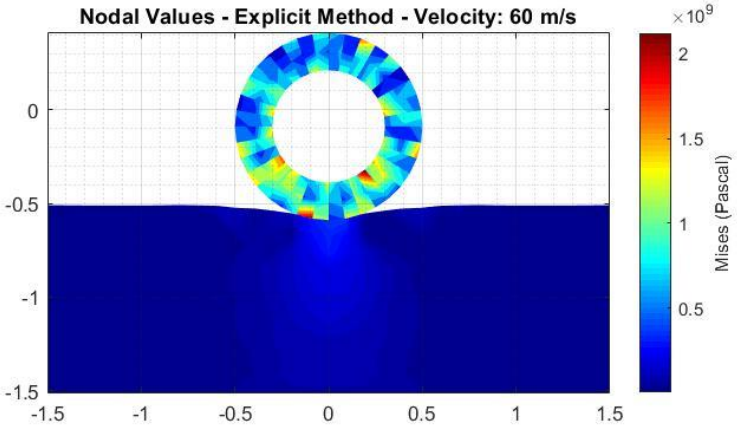


Figure 40. *"Mises" Stresses, Explicit Method - Velocity 60 m/s, Nodal Values, at 0.00385 s (1925th time step).*

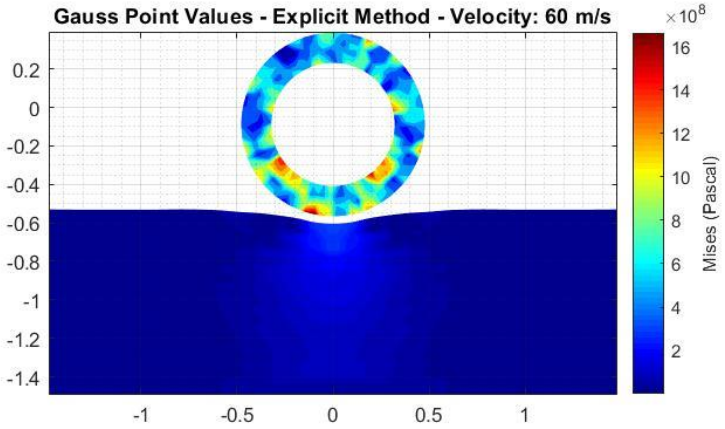


Figure 41. *"Mises" Stresses, Explicit Method - Velocity 60 m/s, Gauss Point Values, at 0.00385 s (1925th time step).*

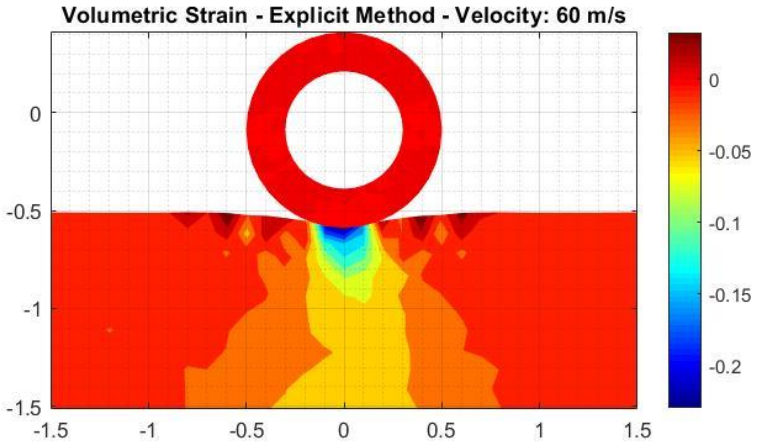


Figure 42. *Volumetric Strain, Explicit Method - Velocity 60 m/s, at 0.00385 s (1925th time step).*

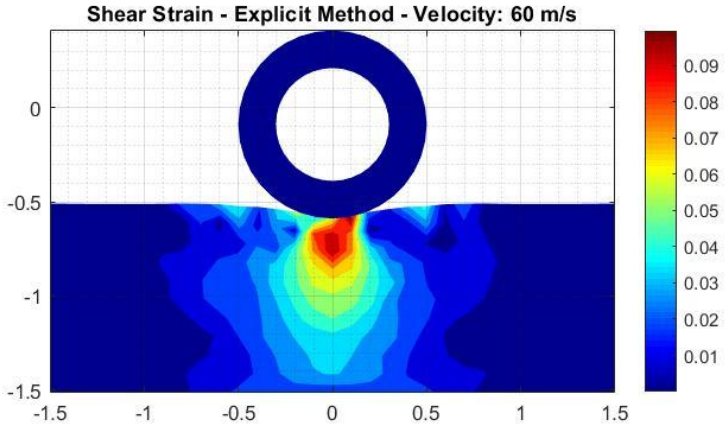


Figure 43. *Shear Strain, Explicit Method - Velocity 60 m/s, at 0.00385 s (1925th time step).*

5.5 Validation of results in "*Abaqus"* software.

Abaqus FEA (formerly *ABAQUS*) is a software suite for finite element analysis and computer-aided engineering, originally released in 1978. The name and logo of this software are based on the abacus calculation tool. The Abaqus product suite consists of five core software products:

1. *Abaqus/CAE*, or "**C**omplete **A**baqus **E**nvironment" (a backronym with a root in **C**omputer-**A**ided **E**ngineering). It is a software application used for both the modeling and analysis of mechanical components and assemblies (pre-processing) and visualizing the finite element analysis result. A subset of *Abaqus/CAE* including only the post-processing module can be launched independently in the Abaqus/Viewer product.
2. *Abaqus/Standard*, a general-purpose Finite-Element analyzer that employs implicit integration scheme (traditional).
3. *Abaqus/Explicit*, a special-purpose Finite-Element analyzer that employs explicit integration scheme to solve highly nonlinear systems with many complex contacts under transient loads.
4. *Abaqus/CFD*, a Computational Fluid Dynamics software application which provides advanced computational fluid dynamics capabilities with extensive support for preprocessing and post processing provided in *Abaqus/CAE*.
5. *Abaqus/Electromagnetic*, a Computational electromagnetics software application which solves advanced computational electromagnetic problems.

The *Abaqus* products use the open-source scripting language Python for scripting and customization. *Abaqus/CAE* uses the fox-toolkit for GUI development. (ABAQUS FEA, n.d.)

5.5.1 Modelling

Appropriate modelling technique using "Abaqus" will be employed in order to have an equivalent model to the one constructed in C# and described in previous section,

First order, plane stress, solid, reduced integration, quadrilateral, four-noded elements are used to discretized the geometry of the two interacting bodies. Material properties will be assigned in the model and the interactions properties will be tuned so that we simulate the impact using the penalty method in the normal direction and zero reaction in tangential direction.

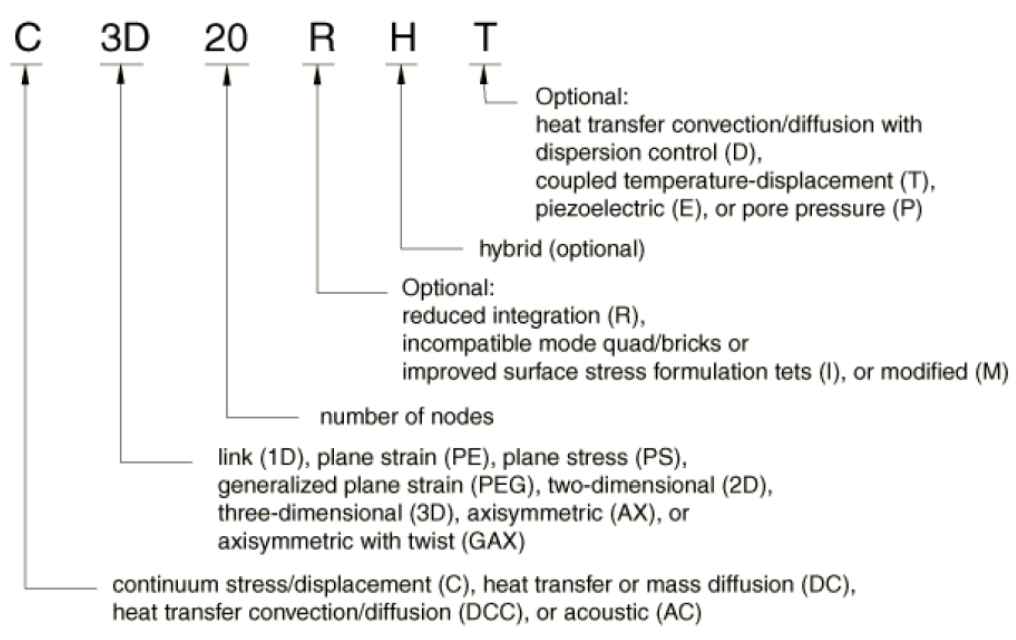


Figure 44. *Element naming convention.*

The boundary conditions are enforced and an initial velocity is applied on each node of the ring.

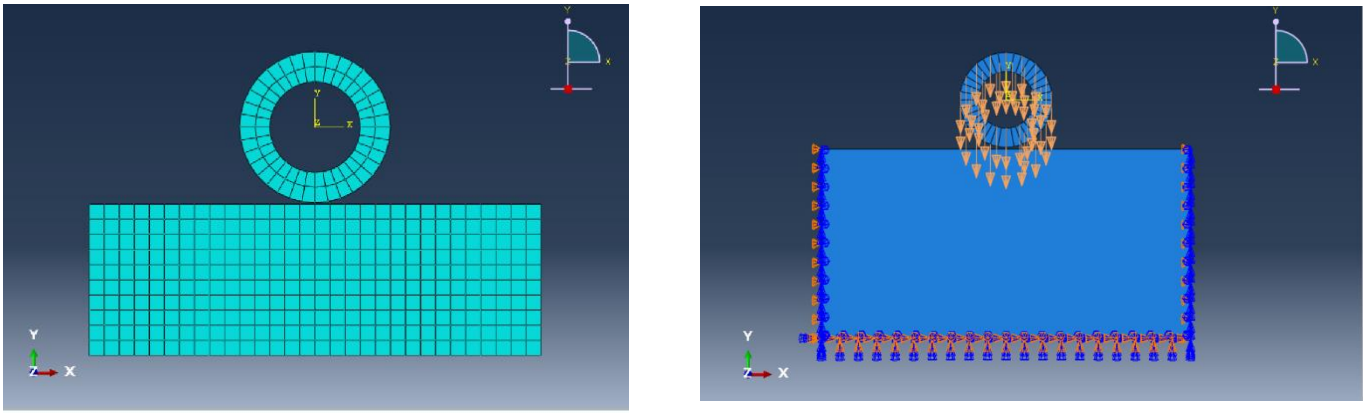


Figure 45. *Model discretization.*

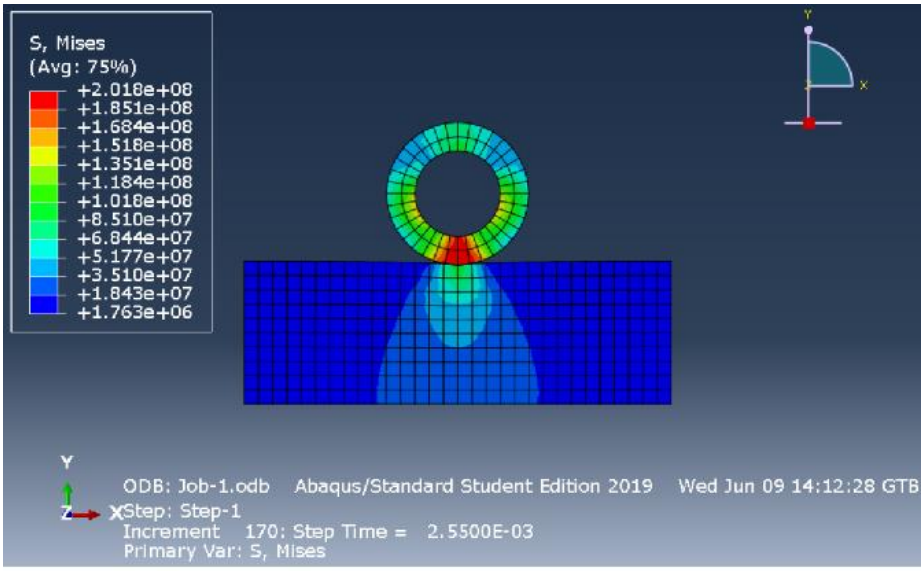


Figure 46. *"Mises" Stresses, Velocity 20 m/s.*

5.5.2 Comparison of results

Below, results from the two solutions (models constructed in C# coding and "Abaqus" software) have been collected and presented in table 5.

To be specific, the displacement at "y" axis of the central node of the upper surface of the rectangular body has been examined for the time that the maximum immersion occurs.

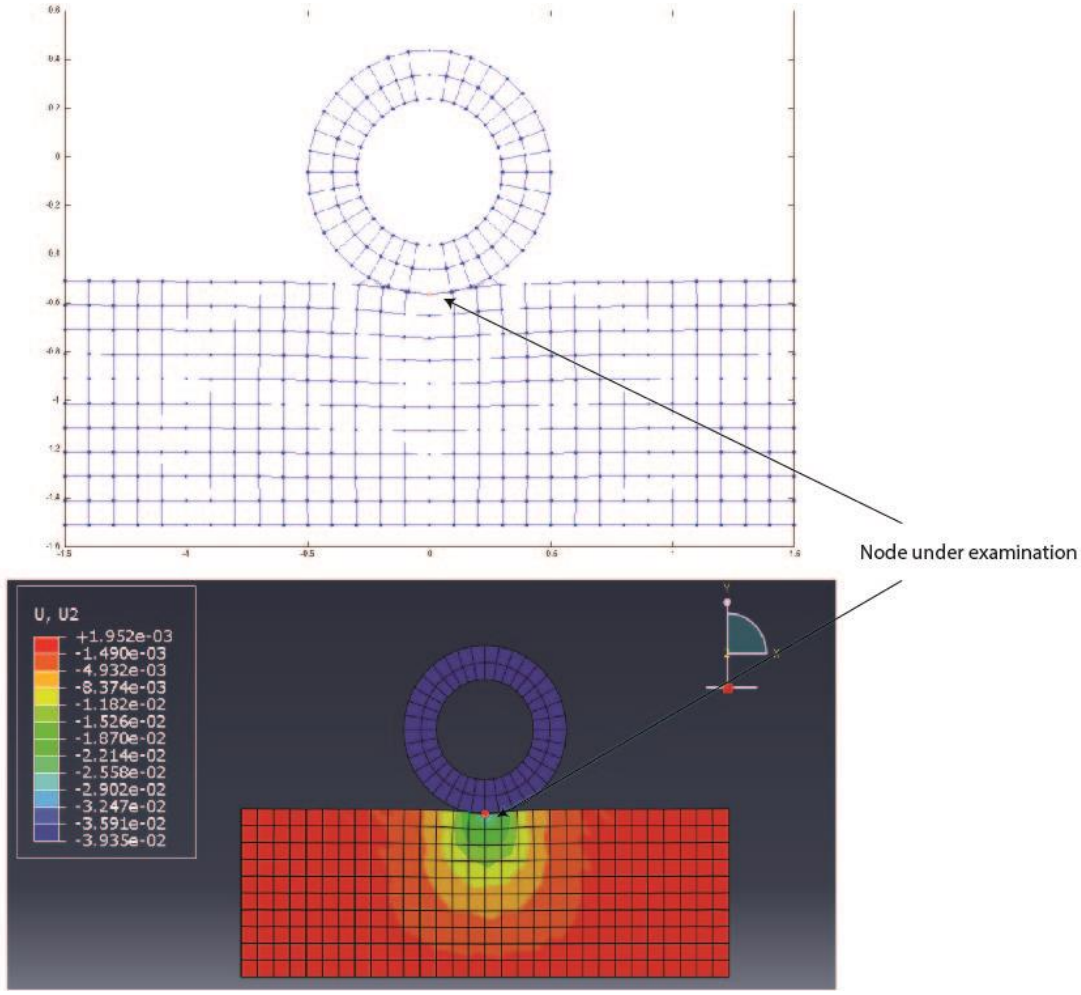


***Figure 47**. Node under inspection.*

For the aforementioned case, simulation is running for 6 ms, divided in 400 steps. In the below table, displacements at time steps of maximum deformation have been recorded. Additionally, in the right column of table, the deviation of the results between the two solutions has been included.

Table 5. *Results – Comparison.*

Velocity of "Body 1"(m/s)	**Vertical Displacement of Node under Inspection (m)**		**Deviation**
	C#	Abaqus	
Implicit Method			
20	-0.0513799	-0.0376743	0.0137056
40	-0.0952157	-0.0696694	0.0255463
60	-0.1340184	-0.1070990	0.0269194
Explicit Method			
20	-0.0514367	-0.0378642	0.0135725
40	-0.0954310	-0.0698743	0.0255567
60	-0.1345468	-0.1073642	0.0271826

6 Conclusion

This thesis has provided a thorough review of the theoretical framework of contact mechanics and more specific of classical Node to Segment formulation for 2D frictionless contact with the penalty method. Both implicit and explicit method, have been described for the solution of the equilibrium equations in dynamic problems.

Relevant applications, with different parameters have been developed. Rate of convergence, computational time, accuracy etc. are some difficulties that faced and needed to overcome. Suitable element discretization and time increment, were the 2 most critical parameters for the solution of the numerical examples that have been implemented in this project.

7 References

ABAQUS FEA, n.d. *SIMULIA.* [Online]
Available at: www.3ds.com

Bathe, K. J., 1996. *Finite Element Procedures.*

Belytschko , T., Chiapetta , T. & Bartel , R. L., 1976. Efficient large-scale non-linear transient analysis by finite elements. *International Journal for Numerical Methods in Engineering,* p. 10:579–596.

Crisfield, M., 2000. Re-visiting the contact patch test. *International Journal for Numerical Methods in Engineering,* Volume 48, p. 435–449.

Hartmann, S. et al., 2009. A contact domain method for large deformation frictional contact problems. Part 2: Numerical aspects. *Computer Methods in Applied Mechanics and Engineering,* Volume 198, p. 2607–2631.

Hertz, H., 1882. "Study on the Contact of Elastic Bodies",. *J. Reine Angew. Math.,,* Volume 29, p. 156–171.

Hilber, H., Hughes, T. R. & Taylor, R. L., 1977. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics,* p. 5:283–292.

Hughes, T. J. e. a., 1976. A Finite Element Method for a Class of Contact-Impact Problems. *Computer Methods in Applied Mechanics and Engineering*.

Kikuchi, N. & Oden, J. T., 1988. *Contact problems in elasticity: a study of variational inequalities and finite element..* SIAM, Philadelphia.

Konyukhov, A. & Schweizerhof, K., 2012. *Computational Contact Mechanics, Geometrically Exact Theory for Arbitrary Shaped Bodies.*

Laursen, T. A., 2002. *Computational Contact and Impact Mechanics.*

Newmark, N. M., 1959. A method of computation for structural dynamics. *Proceedings of ASCE, Journal of Engineering Mechanics,* p. 85:67–94.

Oliver, J., Hartmann,, S. & Cante, J. C., 2009. A contact domain method for large deformation frictional contact problems. Part 1: Theoretical basis. *Computer Methods in Applied Mechanics and Engineering,* Volume 198, p. 2591 – 2606.

Oliver, J., Hartmann, S. & Cante,, J. C., 2010. *On a new 3D contact domain method for large deformation contact problems.In Plenary lecture at IV European Conference on Computational Mechanics,Palais des Congrès.* Paris, France.

Simo, J. C., Wriggers, P. & Taylor, R. L., 1985. A perturbed Lagrangian formulation for the finite element solution of contact problems.. *Computer Methods in Applied Mechanics and Engineering.,* Volume 50, p. 163–180.

Taylor, R. L. & Papadopoulos, O., 1991. Ona patch test for contact problems in two dimensions. *In P.Wriggers &W.Wagner, Nonlinear Computational Mechanics, Springer*, p. 690–702.

Wood, W. L., 1990. Practical Time-stepping Schemes. *Clarendon Press.*

Wood, W. L., Bossak, M. & Zienkiewicz, O. C., 1981. An alpha modification of Newmark's method. *International Journal for Numerical Methods in Engineering*, p. 15:1562-1565.

Wriggers, P., 1995. Finite Element Algorithms for Contact Problems. *Archives of Computational Methods in Engineering,* Volume 2,4, p. 1-49.

Wriggers, P., 2002. *Computational Contact Mechanics.*

Wriggers, P., 2008. *Nonlinear Finite Element Methods.*

Wriggers, P. & Zavarise, G., 2007. A formulation for frictionless contact problems using a weak form introduced by Nitsche. *Springer*.

Yastrebov, V., 2011. *Computational Contact Mechanics: geometry, detection and numerical techniques..* École Nationale Supérieure des Mines de Paris.

Zavarise, G. & De Lorenzis, L., 2009. A modified node-to-segment algorithm passing the contact patch test. *International Journal for Numerical Methods in Engineering,* Volume 79, p. 379–416.

Zavarise, G. & De Lorenzis, L., 2009. The node-to-segment algorithm for 2D frictionless contact. Classical formulation and special cases. *Computer Methods in Applied Mechanics and Engineering*.