Class Notes: Introduction to the Finite Element Methods

Juan David Gómez Cataño
jgomezc1@eafit.edu.co

Grupo de Investigacion en Mecanica Aplicada
Civil Engineering Department
School of Engineering
Universidad EAFIT
Medellín, Colombia
2013

Table of Contents

Summary

Class Notes for the Course Introduction to the Finite Element Methods

Palabras clave: Computational Mechanics, Finite Element Methods.

Chapter 1

Introduction

In an introductory graduate course in the Finite Element Method (FEM), the purpose is to develop a basic understanding of the discrete or numerical strategy of Finite Element-(FE) Algorithms when a closed form solution to a Boundary Value Problem (BVP) is not possible due to complexities existing probably: in the boundary conditions, material behaviour or in the kinematic description used in the model. In such a course, in order to master and identify the main mathematical and algorithmic aspects of the method at the beginners level, the studied problem is kept lineal. In this sense 4 key aspects make the core of the introductory course (at least for the case of the linearized theory of elasticity boundary value problem):

Formulation and identification of the strong form of the Boundary Value Problem (BVP) where the continuum mechanics governing equations are revisited and particularized to the case of linear elastic material behavior. The BVP is completed by identifying the correct prescription of boundary conditions for a well posed mathematical problem.

Formulation and identification of the weak form of the BVP where the governing equations and natural boundary conditions representing equilibrium at the material point level are replaced by an equivalent but weaker form of equilibrium now valid at the global level. In the case of a Continuum Mechanics problem this so-called weak form can be shown to be equivalent to the principle of virtual power and lends itself for the partition of the problem into sub-domains or finite size elements.

Introduction of the idea of discretization dividing the problem into subdomains and using interpolation theory within each subdomain-In the context of the FE method this is the subject of the shape functions. Once the computational specimen has been divided into subdomains (or a mesh of finite elements) the selected primary variable is approximated within each element via interpolation of the known response at selected predefined points or nodes.

Computational aspects grouped also into 5 points: Formulation of elemental matrices in the physical space. Formulation of elemental matrices in the natural domain-Isoparametric transformation. Numerical integration-The Gauss quadrature. Assembly of system matrices and imposition of boundary conditions. Solution of the discrete equilibrium statement and calculation of elemental results.

These 4 key points are perfectly well documented in numerous nicely written textbooks and it could be argued that is not even worth to register for an introductory course since a moderately dedicated student can accomplish the task via self study . Unfortunately, linearity is scarce - although useful to grasp the basic understanding of the problem- and the real world is full of non-linear behavior and understanding the needed algorithms can be easily justified. In this Advanced Finite Element Methods course the goal is then to understand the basic aspects of non-linear finite element analysis. Like in the linear case there is also a vast amount of literature for the non-linear problem. However, in the non-linear case the kinematic problem itself may take different routes leading to a wide variety of FEM formulations that will difficult a self-study strategy. Considering the above this brief set of class notes is intended as a guide for self-study and more important represents a help towards the implementation of the algorithm for the consideration of Material and Geometric Non-linearities in Solids.

The basic reference is Professor Bathe's textbook but we will also follow closely ABAQUS Theory Manual. ABAQUS is a robust, multi-physics oriented commercially available finite element analysis tool that dominates the academic and research field. Its strength resides in the effective non-linear algorithms and on the capability of taking user subroutines written in good old Fortran or in C++. The possibility of implementing user subroutines makes it a very powerful research tool. In the particular case of a stress/displacement analysis problem with non-linearities these are considered through the kinematics contribution or through the material contribution. In the first case the non-linear behavior must be considered at the element level while in the second it corresponds to the response of a material point which in the context of the FEM algorithm corresponds to an integration point. In ABAQUS those two sources of non-linearity can be independently controlled by the user via user subroutines UEL and UMAT. In both cases the non-linearity is primarily solved by the classical Newton-Raphson scheme and that will be the approached followed herein. Although the notes are mainly written for an advanced course the specific problem of a linear solid usually studied in the introductory course can be derived like a particular case of the most general non-linear algorithm.

The current set of Class Notes is organized as follows. First and since we will be dealing with history dependent non-linear problems the most powerful (at least when it works) solution algorithm, namely the Newton-Raphson iteration is studied. The technique is first illustrated for the simple 1D-case and then generalized into the multi-degree of freedom system. In both cases pseudo-codes will be presented preparing the way for the Finite Element Algorithm. The presentation however is not exhaustive in mathematical terms and the reader is referred to excellent treatments like Burden() and Press et al() for the mathematical aspects of the Newton method.

In the next section the briefly introduced Newton-Raphson technique is contextualized to the case of a system of equations representing equilibrium between internal and external forces as typically found in a finite element model. Moreover, the non-linearities come into play through a dependence of the internal forces into displacements. At this stage the details of the formulation of the finite element equations via discretization into nodal variables is not presented but emphasis is laid down into the solution algorithm. Interest is then given to the particular form taken by the Newton-Raphson algorithm into the commercial finite element code ABAQUS. That code can be used as a powerful non-linear equation solver where the coefficient matrix and the excitation can be directly controlled by the user. Moreover the solver can be used into a multi-physics context in terms of generalized forces and fluxes. In the particular case of the stress analysis finite element method the user can control the elemental contribution to the coefficient matrix and the contribution of each material point to obtain that element contribution. This is achieved through the so-called user subroutines UEL for element and UMAT for material. Having introduced the Newton-Raphson method the notes concentrate next on general discretization aspects starting from the physical strong form of the equations in the deformed configuration and passing to an arbitrary weak form in the reference configuration. The resulting algorithm is therefore a Total Lagrangian (TL) method. Once the general equations are introduced a particular work conjugate stress-strain pair is chosen and the discrete equations, including kinematic interpolators, are described.

Notation

In a non-linear algorithm the bookkeeping is involved since we have to simultaneously record 4 different fields as follows; The physical fields in terms of tensorial descriptions. The time field since the problem is solved incrementally and time may appear as an artificial chronological variable or as a real quantity in a dynamic problem. The interpolation field. Since all the involved variables will be interpolated we will need to keep track of the way this interpolation is being performed. The iterations field needed in the solution of the non-linear problem.

In order to keep this bookkeeping simple we use the following indicial notation with subscripts and superscripts;

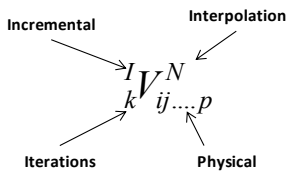

Figura 1.1. General notation to study non-linear finite element problems

Capital superscripts will be reserved for the incremental time description and for the interpolation scheme. For instance an expression like refers to the time instant while refers to . Similarly

a variable refers to interpolation over the node . Left and right subscripts will be used to make reference to the iteration being performed and the order of the tensorial variable. For instance refers to a second order tensor corresponding to the iteration .

Chapter 2

Equilibrium Equations

2.1 General Equilibrium Equations

In this section we develop the incremental equations needed in a general non-linear problem. First, geometric non-linearities valid for any stress-strain pair are considered. Second, focus is shifted towards the particular case of the Second Piola-Kirchoff-Green Lagrange Strain pair.

Let represent the current (deformed) configuration of a deformable medium with equilibrium equations and boundary conditions

$$\sigma_{ij,j} + f_i = 0 \tag{2.1}$$

$$t_i = \sigma_{ij}\hat{n}_j \tag{2.2}$$

where represents the Cauchy definition of stress (or force per unit of deformed surface) and the corresponding tractions vector at a surface with outward normal . In the equilibrium equations stated in (1) the domain is unknown which makes the problem inherently non-linear. On the other hand this equilibrium statement is mathematically indeterminate since 9 unknown stress components must be solved out of 6 field equations. The indeterminacy is destroyed after the problem is kinematically described and connected to the stress field via constitutive modeling which can involve additional sources of non-linearity. To summarize the problem is non-linear since the domain is unknown, this is what we typically call a Geometric Non-linearity and will be reflected in the mathematical description of changes in configuration.

2.1.1 Lagrangian description of the equilibrium equations

The conceptual definition of Cauchy stress is the only one useful for the engineer since it describes forces per unit deformed surface. This validity is clearly identified in the fact that the equilibrium equations stated in (1) have been formulated in the stressed deformed domain where the body is in fact in equilibrium. The difficulty associated with the unknown domain may be dealt with in alternative ways. In the case of a solid body it is convenient to refer everything to the undeformed or reference configuration (with known domain) and proceed from there using linearization which corresponds to a Total Lagrangian (TL) approach. It is then useful to understand the transformation of the problem to via pullback operations. This process implies the introduction or consideration of mathematically defined stress definitions and newly developed kinematic descriptions. In this section we consider such transformations first from a dynamic point of view and later using thermodynamic principles we address the problem of identifying the proper strain measures.

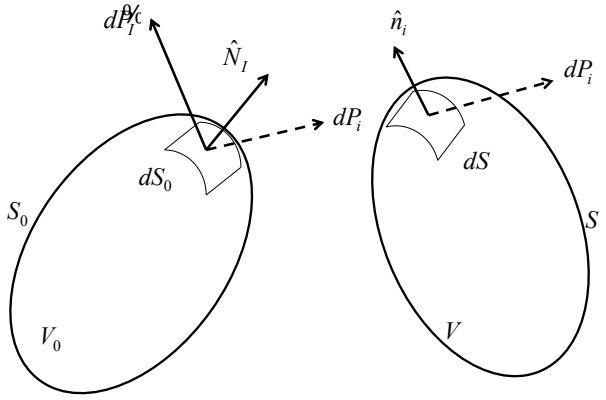

Figura 2.1. Definition of the natural domain

Nanson's formula

For the treatment that follows it will result useful to consider the relation between oriented differential surface elements in the reference and deformed configurations in the so-called Nanson's formula;

$$\hat{n}_i dS = \frac{\rho}{\rho_0} \hat{N}_I f_{Ii} dS_0 \tag{2.3}$$

Lagrangian stress definition

Let be the differential force associated to the Cauchy physical stress such

$$dP_i = t_i dS \equiv \sigma_{ji} n_j dS \tag{2.4}$$

(3)

Assume that can also be obtained in the undeformed configuration associated with a new traction definition;

$$dP_i = t_i^0 dS_0 \equiv T_{Ii}^0 N_I dS_0 \equiv \sigma_{ji} n_j dS \tag{2.5}$$

(4)

Using Nanson's formula it is possible to write;

$$T_{Ii}^0 N_I dS_0 \equiv \sigma_{ji} \frac{\rho_0}{\rho} N_I f_{Ij} dS_0 \tag{2.6}$$

(5)

Which simplifies into

$$T_{Ii}^0 = \frac{\rho_0}{\rho} f_{Ij} \sigma_{ji} \tag{2.7}$$

(6)

and this is the asymmetric First Piola-Kirchoff stress tensor.

Assume now that there is a pseudo-force in the undeformed configuration which results from a pullback operation on the physical force . Recalling the kinematic connection between the current and undeformed configurations

$$dX_I = f_{Ii} dx_i \tag{2.8}$$

(7)

where is the inverse deformation gradient, we can write for the force and pseudo-force vectors;

$$d\tilde{P}_I = f_{Ii} dP_i. \tag{2.9}$$

(8)

Now we can define a tractions vector in the undeformed configuration and associated with the pseudo-force like

$$d\tilde{P}_I = \tilde{t}_I dS_0 \equiv T_{JI} N_J dS_0 \tag{2.10}$$

(9)

where we have at the same time introduced the associated stress tensor. It then directly follows that;

$$T_{JI} N_J dS_0 = f_{Ii} dP_i \equiv f_{Ii} \sigma_{ji} n_j dS \tag{2.11}$$

(10)

once again using Nanson's formula we can write

$$T_{JI} N_J dS_0 = f_{Ii} \sigma_{ji} \frac{\rho_0}{\rho} N_J f_{Jj} dS_0 \tag{2.12}$$

(11)

and

$$T_{JI} = \frac{\rho_0}{\rho} f_{Jj} \sigma_{ji} f_{Ii} \tag{2.13}$$

(12)

which corresponds to the symmetric Second Piola-Kirchoff stress tensor.

For the derivations that follow and in the actual computational implementation it will be convenient to have the inverse relationships expressing the Cauchy stress tensor in terms of the First and Second Piola-Kirchoff stress definitions.

Once again recall that and use the definition found for the first PK stress tensor to write

2.2 Equilibrium in Weak Form

Chapter 3

Nonlinear Problems

3.1 Nonlinear Scalar Equations

3.1.1 Newton-Raphson Scheme

3.2 Systems of Nonlinear Equations

Chapter 4

Elemental Matrices-The Continuum Mechanics Analogy

4.1 Statement of the Problem

In the incremental equilibrium equations (22) we need to perform integration over the reference element domain $V_0(\vec{x})$ corresponding to originally arbitrarily shaped sub-domains as created during the meshing process. In order to proceed with this integration it is useful to consider the following continuum mechanics analogy.

First assume that the actual physical domain $V_0(\vec{x})$ is the result of a deformation process imparted upon the natural domain as shown in Figure A1. In this analogy, the physical domain $V_0(\vec{x})$ is regarded like a "deformed" configuration at an imaginary time $t = t$, while the natural "un-deformed" domain $V(\vec{r})$ is treated like a reference un-deformed configurations at time $t = 0$. Both configurations are assumed to be connected through a deformation process;

$$\begin{aligned} \vec{X} &= \vec{X}(\vec{r}) \\ \vec{r} &= \vec{r}(\vec{X}) \end{aligned} \tag{4.1}$$

In (4.1) we can understand $\vec{r}$ like a material (Lagrangian) variable and $\vec{X}$ like a spatial (or Eulerian) variable. Using the continuum mechanics analogy it is clear that the "deformation" process at the continuum level is fully characterized by the "deformation" gradient or Jacobian of the transformation (4.1) and defined according to;

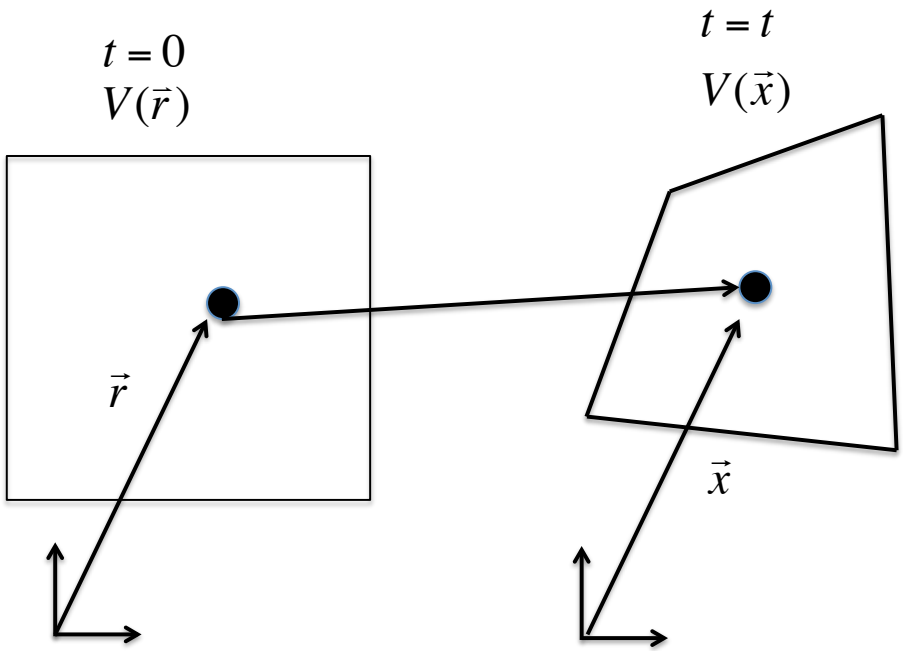

Figura 4.1. Definition of the natural domain

$$dX_i = \frac{\partial X_i}{\partial r_J} dr_J \equiv J_{iJ} dr_J \tag{4.2}$$

where dr_J and dX_i represent material vectors in the original and deformed configuration. From (4.2) it is evident that the Jacobian contains all the information describing the change of the physical sub-domain with respect to the natural element. For the element integration process we will assume that every element $V(\vec{r})$ in the natural domain deforms into the physical element $V_0(\vec{X})$ thus allowing us to write typical terms like the ones in the material stiffness matrix in (22) like;

$$\int_{V_0(\vec{X})} \hat{B}^K_{ij}(\vec{X}) C_{ijkl} \hat{B}^P_{kl}(\vec{X}) dV_0(\vec{X}) \equiv \int_{V_0(\vec{X})} \hat{B}^K_{ij}(\vec{r}) C_{ijkl} \hat{B}^P_{kl}(\vec{r}) J dV(\vec{r}) \tag{4.3}$$

where we have used $dV(\vec{X}) = J dV(\vec{r})$, with J being the determinant of the deformation gradient and in general we transform functions between the natural and physical space making use of (4.1) according to;

$$f(\vec{r}) = F[\vec{X}(\vec{r})] \tag{4.4}$$

Interpolation scheme

Having identified the fact that the integration process will take place in the natural domain we will approach the interpolation process directly in this natural space. In the case of the displacement based finite element method all the involved variables will then be obtained via interpolation of nodal displacements. For instance, assume that a given problem variable is defined in the physical space by the tensor $\Phi_{ik...p}(\vec{X})$. The interpolated variable is then obtained like;

$$\Phi_{ij...p}(\vec{X}) = H^K_{ij...p}(\vec{r})\hat{u}^K \tag{4.5}$$

where $\hat{u}^K$ represents a vector of nodal points displacements, see Figure 4.2, and $H^K_{ij...p}(\vec{r})$ is an interpolator which keeps the tensorial character of the original physical variable $\Phi_{ik...p}(\vec{X})$ and where the super-index makes reference to a nodal identifier and the summation convention applies.

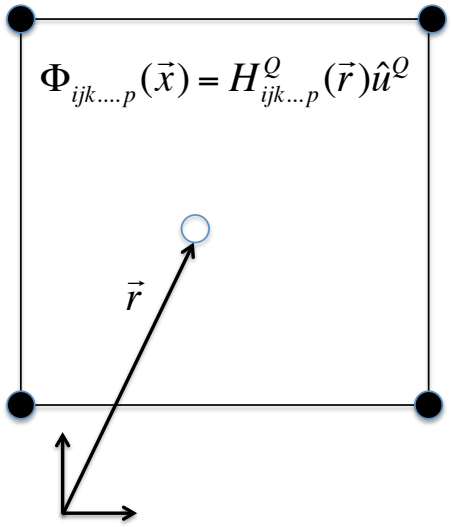

Figura 4.2. General interpolation strategy in the natural domain

Since the primary variable corresponds to displacements it must be kept in mind that $H^K_{ij...p}(\vec{r})$ corresponds to combinations of derivatives (or other arbitrary combinations) of the basic element shape functions defined in;

$$u_i(\vec{X}) = N^K_i(\vec{r})\hat{u}^K \tag{4.6}$$

For the general interpolation process we need two kinds of transformations. First we need to transform integrals over the physical space into integrals into the natural space which corresponds to;

$$\int_{V_0(\vec{X})} F(\vec{X})dV_0(\vec{X}) \equiv \int_{V_0(\vec{r})} f(\vec{r})JdV(\vec{r}) \tag{4.7}$$

Second we need to relate spatial differentiation in both, the physical and spatial domains. Let us define these operators like $\bigtriangledown_i^X$ and $\bigtriangledown_I^r$ respectively. It then follows from (4.4) and the chain rule of differentiation that;

$$\frac{\partial F}{\partial X_i} = \frac{\partial f}{\partial r_J}\frac{\partial r_J}{\partial X_i} \tag{4.8}$$

from where we can establish the connection between the two operators like

$$\bigtriangledown_i^X = J_{Ji}^{-1}\bigtriangledown_I^r \tag{4.9}$$

We further define the fundamental interpolator giving rise to gradients of the primary displacement variable in the physical space according to;

$$u_{i,j}(\vec{X}) = L_{ij}^K(\vec{r})\hat{u}^K \tag{4.10}$$

This fundamental interpolator $L_{ik}^K(\vec{r})$ is derived after using (4.6) and (4.9) in the physical displacement gradient definition as shown next;

$$\begin{aligned}
u_{i,j}(\vec{X}) &= \bigtriangledown_j^X u_i(\vec{X}) \\
u_{i,j}(\vec{X}) &= \bigtriangledown_j^X N_i^K(\vec{r})\hat{u}^K \\
u_{i,j}(\vec{X}) &= J_{Qj}^{-1}\bigtriangledown_Q^r N_i^K(\vec{r})\hat{u}^K \\
u_{i,j}(\vec{X}) &= J_{Qj}^{-1}N_{i,Q}^K(\vec{r})\hat{u}^K
\end{aligned}$$

then

$$L_{ij}^K(\vec{r}) = J_{Qj}^{-1}N_{i,Q}^K(\vec{r}) \tag{4.11}$$

Elemental stiffness matrix

The elemental material stiffness matrix computed in the natural domain of Figure A3 reads;

$$K^{KP} = \int_{V_0(\vec{r})} \hat{B}^K_{ij}(\vec{r}) C_{ijkl} \hat{B}^P_{kl}(\vec{r}) J dV(\vec{r}) \equiv \int_{r=-1}^{r=+1} \int_{s=-1}^{s=+1} \hat{B}^K_{ij}(r,s) C_{ijkl} \hat{B}^P_{kl}(r,s) J(r,s) \mathrm{d}r\mathrm{d}s \tag{4.12}$$

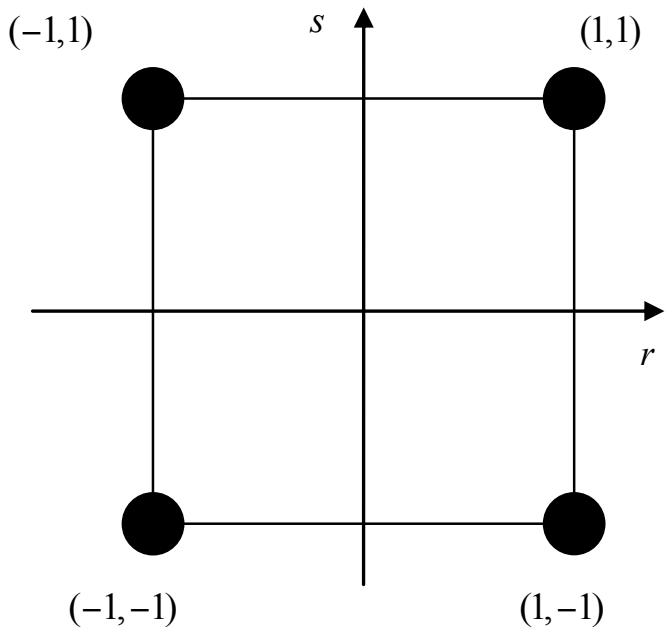

Figura 4.3. Natural domain of integration

Once the interpolator $\hat{B}^K_{ij}(\vec{r})$ has been identified the elemental stiffness matrix is obtained via numerical integration (quadrature) as described in (4.12);

$$\int_{r=-1}^{r=+1} \int_{s=-1}^{s=+1} \hat{B}^K_{ij}(r,s) C_{ijkl} \hat{B}^P_{kl}(r,s) J(r,s) \mathrm{d}r\mathrm{d}s \approx \sum_{i,j=1}^{NGPTS} \alpha_i \alpha_j \hat{B}^K_{kl}(r_i,s_j) C_{ijkl} \hat{B}^P_{kl}(r_i,s_j) J(r_i,s_j) \tag{4.13}$$

(A13)

and where NGPTS corresponds to the number of integration points, α_j is a weighting factor and r_i, s_j are the coordinates of a typical point $\vec{r}$ in the natural space of Figure A4 .

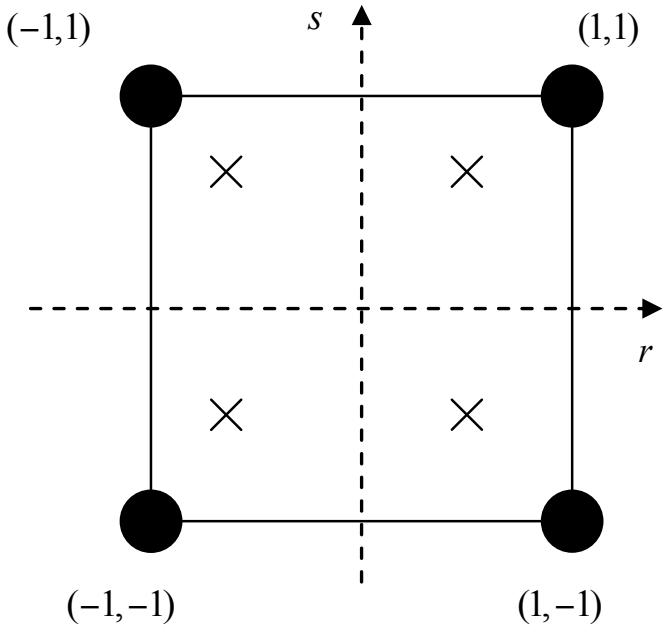

Figura 4.4. Natural integration domain showing quadrature evaluation nodes

One important aspect of the numerical integration that has to be kept in mind is accuracy. Depending on the particularly selected integration scheme, the number of introduced integration points fixes the maximum polynomial order of the considered functions that can be integrated accurately. In the case of the integrand in (4.13), it is clear that this order increases as the distortion of the physical element with respect to the natural element increases. One way of dealing with this dependency of accuracy with element distortion is to make use of adaptative integration techniques which are numerically expensive. What is actually done in standard FEM analysis is to choose the number of quadrature points beforehand and introduce distortion related error criteria inside the code in such a way that some sort of validation is performed before the numerical integration process is started.

Chapter 5

Nonlineal Equilibrium Equations in a Finite Element Discretization

5.1 General Form

5.1.1 Newton-Raphson Scheme

5.2 Algorithm

Chapter 6

General Discrete Form for Any Given Stress-Strain Pair

6.1 General Equations

6.1.1

6.2

Chapter 7

Explicit Dynamic Analysis

7.1 Statement of the Problem

Consider the discrete dynamic equilibrium equations at time t

$$M\,{}^{t}A + C\,{}^{t}V + K\,{}^{t}U = {}^{t}F \tag{7.1}$$

where M, C, K are the assembled mass, damping and stiffness matrix respectively and similarly ${}^{t}A$, ${}^{t}V$, ${}^{t}U$, tF are the nodal accelerations, velocities, displacements and external loads vectors at time t. In terms of forces (7.1) can be written like;

$${}^{t}F^{I} + {}^{t}F^{D} + {}^{t}F^{s} = {}^{t}F \tag{7.2}$$

where ${}^{t}F^{I}$, ${}^{t}F^{D}$ and ${}^{t}F^{s}$ are inertial, damping and elastic components respectively.

Expanding the acceleration and velocity terms at time t in a consistent finite central differences scheme we have;

$$\begin{aligned} {}^{t}A &= \frac{1}{\Delta t^2}\left({}^{t-\Delta t}U - 2\,{}^{t}U + {}^{t+\Delta t}U\right) \\ {}^{t}V &= \frac{1}{2\Delta t}\left(-{}^{t-\Delta t}U + {}^{t+\Delta t}U\right) \end{aligned} \tag{7.3}$$

Using (7.3) in (7.2) yields;

$$\left(\frac{1}{\Delta t^2}M+\frac{1}{2\Delta t}C\right)^{t+\Delta t}U={}^{t}F-\left(K-\frac{2}{\Delta t^2}M\right)^{t}U-\left(\frac{1}{\Delta t^2}M-\frac{1}{2\Delta t^2}C\right)^{t-\Delta t}U \tag{7.4}$$

Redefine forces as follows

$$\begin{aligned} {}^{j}F^{I} &= \frac{1}{\Delta t^2}M^{j}U \\ {}^{j}F^{D} &= \frac{1}{2\Delta t}C^{j}U \\ {}^{j}F^{S} &= K^{j}U \end{aligned} \tag{7.5}$$

and write (7.4) as;

$${}^{t+\Delta t}F^{I}+{}^{t+\Delta t}F^{D}={}^{t}F-{}^{t}F^{s}+2{}^{t}F^{I}-{}^{t-\Delta t}F^{I}+{}^{t-\Delta t}F^{D} \tag{7.6}$$

- (7.6) is an equilibrium equation at time $t = t$ allowing to predict the displacements at time $t = t + \Delta t$ in terms of previously known values at times t and $t = t - \Delta t$.
- The equation is exact within the error introduced by the expansion used in (7.3).

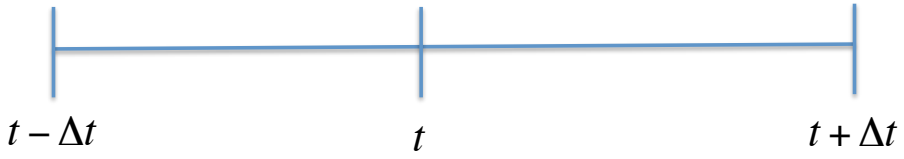

Figura 7.1. Definition of the general iteration

- The first predicted solution is at $t = \Delta t$ and we require data at $t = -\Delta t$ and at $t = 0$.

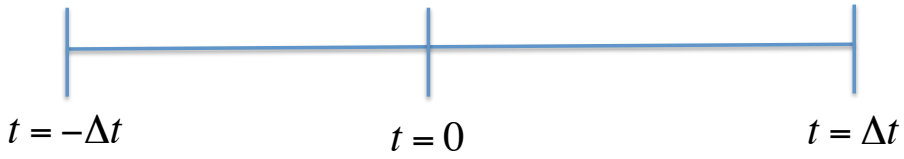

Figura 7.2. Definition of the general iteration

7.1.1 Damping Assumptions

1 Use Rayleigh Damping and retain the velocity expansion used in (7.3). That is;

$$C = \alpha M + \beta K \tag{7.7}$$

then we have (in terms of forces);

$$(1 + \beta \Delta t^2)^{t+\Delta t} F^I + \frac{\alpha}{2\Delta t}{}^{t+\Delta t} F^S = {}^t \hat{F} \tag{7.8}$$

where;

$${}^t \hat{F} = {}^t R - {}^t F^S + 2{}^t F^I - {}^{t-\Delta t} F^I + {}^{t-\Delta t} F^D$$

Solution in equation (7.8) requires the full assembly and factorization of an effective stiffness matrix.

2 Neglect damping (This is however inconvenient for finite domains)

$${}^{t+\Delta t} F^I = {}^t F - {}^t F^S + 2{}^t F^I - {}^{t-\Delta t} F^I \tag{7.9}$$

3 Use Rayleigh damping but modify the velocity expansion introduced in (7.3). Using

$${}^t V = \frac{1}{\Delta t}({}^t U - {}^{t-\Delta t} U) \tag{7.10}$$

yielding;

$${}^{t+\Delta t} F^I = {}^t F - {}^t F^S + 2{}^t F^I - {}^t F^D + {}^{t-\Delta t} F^D - {}^{t-\Delta t} F^I \tag{7.11}$$

Defining a set of forces associated to the initial conditions like;

$${}^{t-\Delta t} F^{IC} = {}^{t-\Delta t} F^I - {}^{t-\Delta t} F^D$$

we have

$${}^{t+\Delta t} F^I = {}^t F - {}^t F^S + 2{}^t F^I - {}^t F^D - {}^{t-\Delta t} F^{IC} \tag{7.12}$$

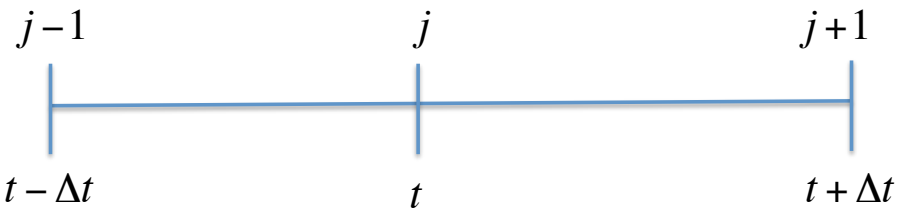

Figura 7.3. Definition of the general iteration

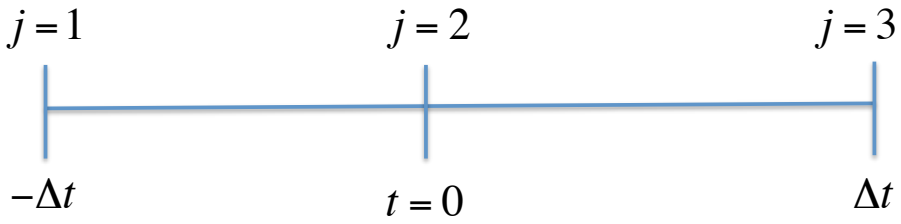

Figura 7.4. Definition of the initial iteration

7.1.2 Algorithm corresponding to the damping assumption 3

Let us write (7.12) like

$$^{j+1}F^I = ^j F - ^j F^S + 2^j F^I - ^j F^D - ^{j-1} F^{IC} \tag{7.13}$$

where the initialization process corresponds to;

Applying (7.13) for $t = 0$ we have;

$$^0F^I = ^0 F - ^0 F^S + 2^0 F^I - ^0 F^D - ^{-\Delta t} F^{IC}$$

from which it is clear that we require $-^{\Delta t}U$. Applying the central difference expansion at $t = 0$ and solving for $-^{\Delta t}U$ yields;

$$^{-\Delta t}U = ^0 F - \Delta t^0 V + \frac{\Delta t^2}{2}{}^0 A \tag{7.14}$$

Using (7.14) in (7.13) allows us to start up the algorithm.

Particulars

In what follows we concentrate on this last algorithm and in order to study some details we return to its standard displacements form. Writing (7.13) in terms of displacements and re-arranging yields;

$$\frac{1}{\Delta t^2}M^{t+\Delta t}U = {}^t F - (1+\frac{\beta}{\Delta t})K^t U + (\frac{2}{\Delta t^2} - \frac{\alpha}{\Delta t})M^t U - (\frac{1}{\Delta t^2} - \frac{\alpha}{\Delta t})M^{t-\Delta t}U + (\frac{\beta}{\Delta t})K^{t-\Delta t}U \quad (7.15)$$

Let;

$$\begin{aligned} a_1 &= 1 + \frac{\beta}{\Delta t} \\ a_2 &= \frac{2}{\Delta t^2} - \frac{\alpha}{\Delta t} \\ a_3 &= \frac{1}{\Delta t^2} - \frac{\alpha}{\Delta t} \\ a_4 &= \frac{\beta}{\Delta t} \end{aligned}$$

$${}^{t+\Delta t}F^I = {}^t F - a_1 K^t U + a_2 M^t U - a_3 M^{t-\Delta t}U + a_4 K^{t-\Delta t}U \quad (7.16)$$

7.1.3 Decoupling

Consider the equation for the i-th d.o.f;

$$\frac{1}{\Delta t^2}M_{ij}^{t+\Delta t}U_j = {}^t F_i - a_1 K_{ij}^t U_j + a_2 M_{ij}^t U_j - a_3 M_{ij}^{t-\Delta t}U_j + a_4 K_{ij}^{t-\Delta t}U_j \quad (7.17)$$

where we keep i fixed in (7.17). For a lumped mass matrix we can write;

$$M_{ij} = m_I \delta_{ij}$$

then (7.17) becomes;

$$\frac{1}{\Delta t^2} m_I^{t+\Delta t} U_i =^t F_I - a_1 K_{ij}^t U_j + a_2 m_I^t U_i - a_3 m_I^{t-\Delta t} U_i + a_4 K_{ij}^{t-\Delta t} U_j \tag{7.18}$$

Let;

$$\begin{aligned} {}^{t+\Delta t}F_i^I &= \frac{1}{\Delta t^2} m_I^{t+\Delta t} U_i \\ {}^t\hat{F}_i^S &= a_1 K_{ij}^t U_j \\ {}^t\hat{F}_i^I &= a_2 m_I^t U_i \\ {}^{t-\Delta t}\hat{F}_i^I &= a_3 m_I^{t-\Delta t} U_i \\ {}^{t-\Delta t}\hat{F}_i^S &= a_4 K_{ij}^{t-\Delta t} U_j \end{aligned} \tag{7.19}$$

so the recursive equation takes the form;

$${}^{t+\Delta t}F_i^I =^t F_i -^t \hat{F}_i^S +^t \hat{F}_i^I -^{t-\Delta t} \hat{F}_i^I +^{t-\Delta t} \hat{F}_i^S \tag{7.20}$$

and the algorithm then reduces to;

Algoritmo 1: Summarized Algorithm

Data: Time span, Geometry, Material Paramters
Result: Displacements, Velocity and Acceleration time histories
Compute ${}^{t+\Delta t}F_i^I$
Solve for ${}^{t+\Delta t}U_i = \left(\frac{\Delta t^2}{m_I}\right)^{t+\Delta t} F_i^I$
Update ${}^tV_i$, ${}^tA_i$

To initialize the algorithm we apply the FD's equations at $t = 0$

$$\frac{1}{\Delta t^2} m_I^{\Delta t} U_i =^0 F_i - a_1 K_{ij}^0 U_j + a_2 m_I^0 U_i - a_3 m_I^{-\Delta t} U_i + a_4 K_{ij}^{-\Delta t} U_j$$

where ${}^{-\Delta t}U_i$ is obtained from (7.14)

$${}^{-\Delta t}U_i =^0 U_i - \Delta t^0 V_i + \frac{\Delta t^2}{2}^0 A_i \tag{7.21}$$

The initial acceleration is obtained after assuming homogeneous IC's;

$$m_I^0 A_i + C_{ij}^0 V_j + K_{ij}^0 U_j = {}^0 F_i$$

therefore

$$m_I^0 A_i = \frac{{}^0 F_i}{m_I}$$
$$-{}^{\Delta t} U_i = \frac{\Delta t^2}{2m_I}\, {}^0 F_i$$

Moreover, neglecting the damping effects on the prediction of ${}^{\Delta t} U_i$ yields;

$${}^{\Delta t} U_i = \frac{\Delta t^2}{2m_I}\, {}^0 F_i$$

Algoritmo 2: Full Algorithm

Data: Time span, Geometry, Material Paramters
Result: Displacements, Velocity and Acceleration time histories
Initialize solution vectors ($j = 1$);
${}^0 U_i \longrightarrow {}^1 U_i = 0$, ${}^0 V_i = 0$, ${}^1 A_i = \dfrac{{}^1 R_i}{m_I}$;
Select Δt and integration constants a_1,a_2, a_3, a_4;
Fix 1-st predicted value (let $j = 2$);

$${}^{\Delta t} U_i \longleftarrow \frac{\Delta t^2}{2m_I}\, {}^0 F_i \longleftrightarrow \left[{}^2 U_i \longleftarrow \frac{\Delta t^2}{2m_I}\, {}^1 F_i \right]$$

Time Integration Phase;
while $j \leq N$ **do**

$${}^{j+1} F_i^I \longleftarrow {}^j F_i - a_1 K_{ij}^j U_j + a_2 m_I^j U_j - a_3 m_I^{j-1} U_i + a_4 K_{ij}^{j-1} U_j$$
$${}^{j+1} U_i \longleftarrow \frac{\Delta t^2}{2m_I}\, {}^0 F_i^I$$
$${}^j A_i \longleftarrow \frac{1}{\Delta t^2} \left({}^{j-1} U_i - 2\, {}^j U_i + {}^{j+1} U_i \right)$$
$${}^j V_i \longleftarrow \frac{1}{2\Delta t} \left(-{}^{j-1} U_i + {}^{j+1} U_i \right)$$
$$j \longleftarrow j + 1$$

end

Nodal Assembler

In the uncoupled explicit finite element formulation the equation solving process proceeds one degree of freedom at a time. This implies a different assembly process to the one used in an implicit algorithm where a formal coefficient matrix is assembled and inverted. Now the mass, damping and stiffness elemental matrices are used to obtain effective nodal forces at each degree of freedom. In summary the mesh is not covered in an element by element basis, but in a node by node basis. In the following algorithm we discuss this nodal assembly process where in order to solve the displacement at a given degree of freedom prior knowledge of the element contributing to the given node is necessary. In the nodal assembler algorithm the following arrays are needed.

ILIST(): Stores the elements connected to the current node.
LPLIST():Stores the local position of the current node in each one of the elements of ILIST().
NIEL():Number of elements at the current node. This array is used to access ILIST().

Algoritmo 3: Nodal Assembler

Data: Number of nodal points, number of elements, Model
Result: Displacements, Velocity and Acceleration time histories
while $i \leq NUMNP$ **do**
| $K \leftarrow NIEL(i)$
end