



Yifei Wang

Expand of Finite Element Toolbox

Master's Thesis

30.05. 2016

Supervisor:

Johannes Rutzmoser

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Johannes Rutzmoser for the continuous support of my study and related research, for his patience, motivation, and immense knowledge I am deeply grateful of his help in the completion of this thesis. His guidance helped me in all the time of finite element research and get in with Finite- Element- Research-code. I could not have imagined having a better advisor and mentor for my study in Finite-Element- Method.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Structure of Study	2
2	Numerical Aspect for Finite Element Formulation	5
2.1	Shape Function	5
2.1.1	Quadrilateral Elements	5
2.1.2	Triangular and Tetrahedral Elements	6
2.2	Non- linear Element Formulation	6
2.2.1	Discretization of the Displacement Field	7
2.3	Non-linear Formulation of Strain and Stress	8
2.3.1	Gauss Integration	12
2.4	Extrapolation from Gauss Points	12
2.4.1	Quad4	12
2.4.2	Quadrangle with quadratic shape function and eight nodes: Quad8 . . .	16
2.4.3	Triangle with three nodes: Tri3	18
2.4.4	Second order triangle with six nodes: Tri6	19
2.4.5	Tetraeder element with four nodes: Tet4	20
2.5	Coordinate Transformations	20
3	Programming Implementation	25
3.1	Pre-Processing	25
4	Calculation of Element Strains and Stresses	29
4.1	Shape Function	29
4.1.1	Quadrilateral Elements	29
4.1.2	Triangular and Tetrahedral Elements	29
5	Bestimmung des optimalen Tunings k	31
6	Zusammenfassung und Ausblick	33
	Averaging-Methode	35

Rotationspendel	37
References	38

List of Figures

1.1	Stress-strain curve for steel	2
1.2	Flow chart of structure	3
1.3	Overview of AMfe structure	4
2.1	Linear shape function and quadratic shape function.	6
2.2	triangular and tetradral elemnets.	7
2.3	Quad4 in element coordinate and Gauss element coordinate.	14
2.4	Equation of side opposite corner 1 for Quad4.	15
2.5	Quad8 in element coordinate and Gauss element coordinate.	17
2.6	Equation of side opposite corner 1 for Quad8.	18
2.7	Tri3 in element coordinate and Gauss element coordinate.	19
2.8	Equation of side opposite corner 1 for Tri3.	21
2.9	Tri6 in element coordinate and Gauss element coordinate.	21
2.10	Equation of side opposite corner 1 for Tri6.	22
2.11	Tri4 in element coordinate and Gauss element coordinate.	22
3.1	A simple meshed geometry	26

List of Tables

2.1	Gauss-Legendre points and weights	13
2.2	Natural Coordinate of Quad4	13
2.3	Natural Coordinate of Tri3	20
2.4	Tetrahedral Coordinate of Tet4	23
3.1	Node list exported from ANSYS	26
3.2	Element list exported from ANSYS	26

Chapter 1

Introduction

1.1 Motivation

The finite element method (FEM) is a numerical technique for finding approximate solutions to boundary value problems for partial differential equations. FEM contains linear FEM and non-linear FEM. Linear analysis follows the equation $K \cdot D = F$, which K stands for stiffness matrix, D stands for displacement and F stands for force. It means that the correlation of force and displacement is linear. This relation between force and displacement is only valid for material that has elastic linear property. But in real situation, elastic material such as steel has also non-linear region. (see the Figure 1.1). The property is elastic linear, until at yield point, the steel is yielding and become non-linear. Non-linear cases include geometry non-linear and material non-linear. The example in Figure 1.1 just describes a non-linear case and it is also the main topic in this thesis. This thesis will only focus on nonlinear FEM. In engineering and science in general, FEM helps tremendously in producing strength visualization and it is also possible to lead engineer for minimizing weight, materials and costs. The accuracy of solution are only limited by the quality of model and by the available computational power. Since the computational power has been improved enormously, FEM software offers a wide range of simulation of complex model designs and analysis of a system.

AMfe is a nonlinear finite element code for structural application at chair of applied mechanics in Technical University of Munich. AMfe Toolbox is developed in Python and Fortran. Python is a high-level, interpreted and dynamically-typed programming language. There are many numerical packages built on Python, such as Numpy, Scipy, Pandas. These packages provide high-performance, easy to use data structures, which just match for a FEM developing work. And because Python is a high-level programming language, it makes less time to develop the code, but on the other hand, it is slow for repeated execution of low-level task. Each Python operation comes with a small type-checking overhead, and with many repeated small operation, this overhead becomes significant. For that reason, the part of code are rewritten in Fortran. Fortran is a general-purpose, imperative programming language that is especially suited to numeric computation and scientific computing. So this combination obtain the advantage from both of them- easy to develop in Python and fast numeric computation in Fortran. The aim of AMfe Toolbox is to solve and analyse FEM problems, especially structural mechanical problems. It contains several

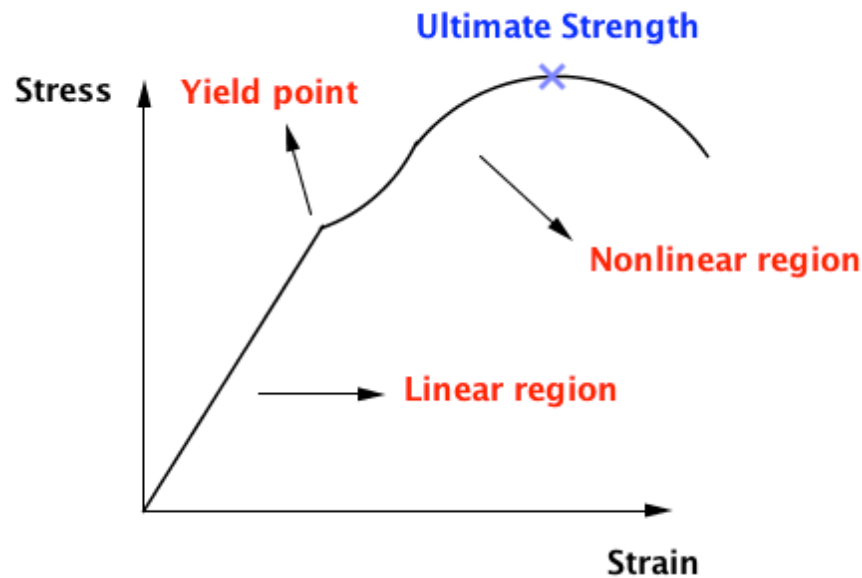


Figure 1.1: Stress-strain curve for steel

modules with different functions to solve problems step by step. A simple structure of AMfe are depicted in Figure 1.3. Displacement, stress and strain are three important factors for structural mechanical engineering. The function for calculation of nodal displacements is completed in AMfe Toolbox. Stress and strain calculation are still empty. Stress and strain calculations are of interest because in structural mechanical analysis and design for geometry the stresses are often very important to the engineer. For that reason, functions for solving of stress and strain will be added in AMfe Toolbox. When stress is able to export, a research to increase the accuracy of results is worthy of consideration. To improve the accuracy of the results, stress recovery is a approach to extrapolate the element solution to nodal solution. The goal is to get as much accuracy from the computed displacements while keeping the computational effort reasonable. When the most important functions are done and works well, it is time to check if AMfe Toolbox export the reasonable results. The results of stress and strain from AMfe Toolbox will be compared with a commercial FEM software - ANSYS. The compare of each element type will be recorded.

1.2 Structure of Study

This study can be divided into three parts. Figure 1.2 shows the flow chart of structure in this study. The first part is marked with green box. The main job for this part is to export data of meshed geometry by using ANSYS Parametric Design Language(APDL) code and import them into AMfe Toolbox. This procedure called Pre-Processing which is the first step in solving a

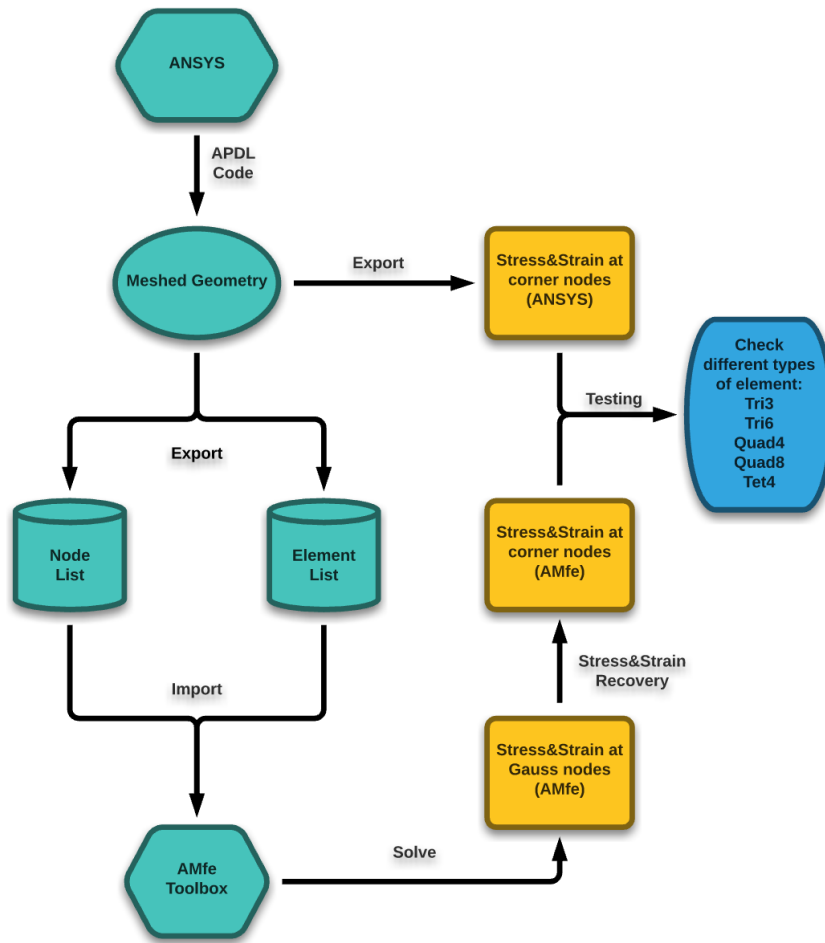


Figure 1.2: Flow chart of structure

problem in Finite Element Analysis. This step also ensures the both analysis with same set of elements and nodes. After that it is the processing step, which is marked with yellow box. AMfe Toolbox has several modules with different functions. These modules combine with each other and provides all the calculation function for processing. This study is focusing on calculating stress and strain. The last part marked with blue colour is to check the accuracy of stress and strain with using different types of elements.

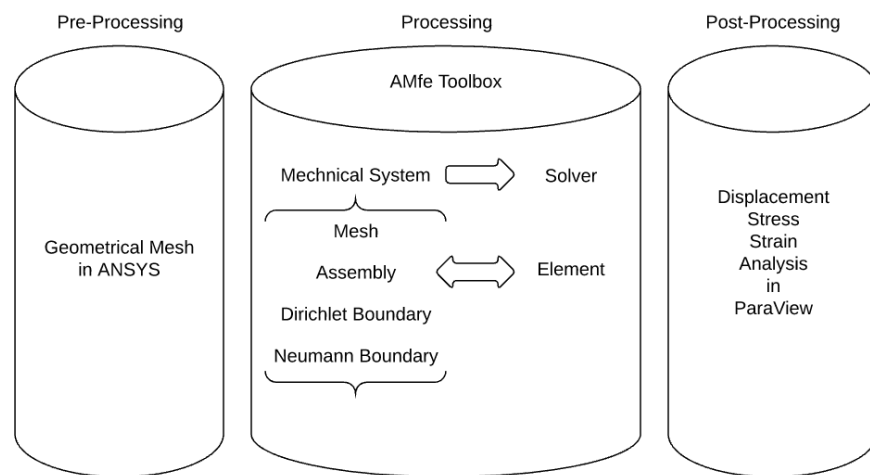


Figure 1.3: Overview of AMfe structure

Chapter 2

Numerical Aspect for Finite Element Formulation

2.1 Shape Function

2.1.1 Quadrilateral Elements

To get the approximate solution (position, displacement, stress and strain) in structural mechanical field is the goal of AMfe Code. The approximation is basically depended on shape function N . The standard approach for the definition of shape functions is to chose them as simple polynomials which are associated to nodes. One element includes n nodes and n shape functions, where the i -th shape function takes the value 1 at the i -th node in reference coordinates, then the j -th shape function can be written like this:

$$\begin{aligned} \text{if } j = i \text{ then } N_j(\hat{\xi}_i) &= 1 \\ \text{if } j \neq i \text{ then } N_j(\hat{\xi}_i) &= 0 \end{aligned}$$

for $i, j = 1, \dots, n$.

In Figure 2.1, linear, quadratic shape function are shown together with corresponding node positions. When it comes to a element with three nodes, as shown in the right of Figure 2.1. For each shape functions, there are three restrictions:

$$\begin{aligned} N_1(\hat{\xi}_1 = -1) &= 1, N_1(\hat{\xi}_2 = 0) = 0, N_1(\hat{\xi}_3 = 1) = 0; \\ N_2(\hat{\xi}_1 = -1) &= 0, N_2(\hat{\xi}_2 = 0) = 1, N_2(\hat{\xi}_3 = 1) = 0; \\ N_3(\hat{\xi}_1 = -1) &= 0, N_3(\hat{\xi}_2 = 0) = 0, N_3(\hat{\xi}_3 = 1) = 1 \end{aligned}$$

In general quadratic polynomial, the shape functions have three coefficients, and it can be constructed as follows:

$$p_{quad}(\xi) = c_0 + c_1\xi + c_2\xi^2$$

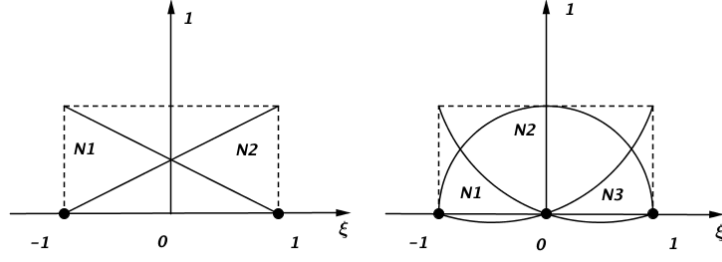


Figure 2.1: Linear shape function and quadratic shape function.

Lagrange polynomials are a general class of polynomials with the node association property: a Lagrange polynomial l_k^{n-1} (in one coordinate ξ) of order $n-1$ passes through n nodes with coordinates $\bar{\xi}^j$ ($j = 1, \dots, n$) of which the single node k evaluates unity ($l_k^{n-1}(\bar{\xi}^k) = 1$) and every other node results in zero ($l_k^{n-1}(\bar{\xi}^j) = 0$ for all $j \neq k$).

$$l_k^{n-1}(\xi) = \prod_{j=1, j \neq k}^n \frac{\xi - \bar{\xi}^j}{\bar{\xi}^k - \bar{\xi}^j} = \frac{(\xi - \bar{\xi}^1) \dots (\xi - \bar{\xi}^{k-1})(\xi - \bar{\xi}^{k+1}) \dots (\xi - \bar{\xi}^n)}{(\bar{\xi}^k - \bar{\xi}^1) \dots (\bar{\xi}^k - \bar{\xi}^{k-1})(\bar{\xi}^k - \bar{\xi}^{k+1}) \dots (\bar{\xi}^k - \bar{\xi}^n)}$$

$$k = 1, 2, \dots, n$$

For instance linear, i.e. order 1, Lagrange polynomials are achieved with $n-1 = 1 \Rightarrow n = 2$ nodes. Using $\bar{\xi}^1 = -1$ and $\bar{\xi}^2 = +1$ results in: $l_1^1(\xi) = -1/2(\xi - 1)$ and $l_2^1(\xi) = 1/2(\xi + 1)$. Then the linear shape functions can be simply constructed as $N_1 = l_1^1 = 1/2(1 - \xi)$ and $N_2 = l_2^1 = 1/2(1 + \xi)$.

2.1.2 Triangular and Tetrahedral Elements

Due to the higher flexibility of triangles and tetrahedra for meshing complex geometries, they are often preferred over quadrilaterals or hexahedra. The shape functions are given in an analogous method, but they are expressed in area and volume coordinates, respectively. Figure ?? shows the geometries and shape functions. These coordinates represent area and volume fractions, a visualization is given below:

$$L_1 = \frac{\text{area}_{P23}}{\text{area}_{123}}, \quad V_1 = \frac{\text{area}_{P234}}{\text{area}_{1234}}$$

$$\sum L_i = 1 \quad \sum V_i = 1$$

2.2 Non- linear Element Formulation

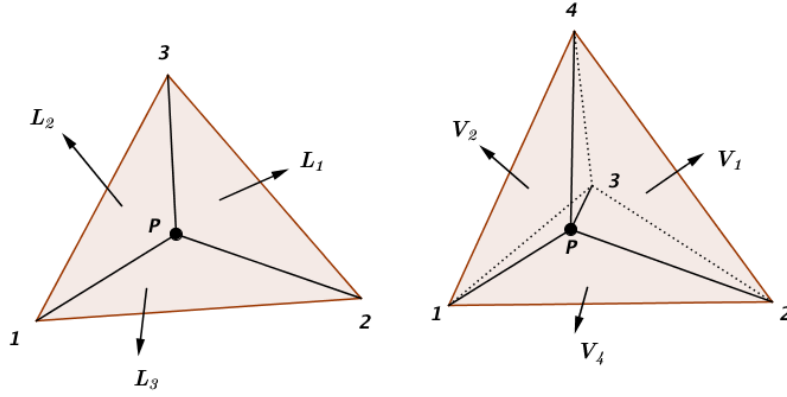


Figure 2.2: triangular and tetradral elemnets.

2.2.1 Discretization of the Displacement Field

After meshing of the geometry, displacement field u can be discretized by inserting shape function N . Firstly, we introduce the nodal displacement d^e to save displacements at each node of the considered element for each spatial direction. In elastic materials, through the elastic constitutive equations can be derived that stresses σ are directly related to strains ε at each point:

$$\sigma = E\varepsilon \quad (2.1)$$

It follows that the stress calculation procedure begins with strain calculation, and that the accuracy of stresses depends on that of strains. In the following sections, we focus our attention on two-dimensional isoparametric elements, as the calculation of strains, stresses and axial forces in bar elements is straightforward. Suppose that we have solved the master stiffness equations.

$$Ku = f \quad (2.2)$$

for the node displacements u . To calculate strains and stresses we perform a loop over all defined elements. Let ε be the element index of a specific two-dimensional isoparametric element encountered during this loop, and $u(\varepsilon)$ the vector of computed element node displacements. The strains at any point in the element can be related to these displacements as

$$\varepsilon = Bu(\varepsilon) \quad (2.3)$$

where B is the strain-displacement matrix assembled with the x and y derivatives of the element shape functions evaluated at the point where we are calculating strains. The corresponding stresses are given by

$$\sigma = E\varepsilon = EBu \quad (2.4)$$

In the applications it is of interest to evaluate and report these stresses at the element nodal points located on the corners and possibly midpoints of the element. These are called element nodal

point stresses. It is important to realize that the stresses computed at the same nodal point from adjacent elements will not generally be the same, since stresses are not required to be continuous in displacement- assumed finite elements. This suggests some form of stress averaging can be used to improve the stress accuracy, and indeed this is part of the stress recovery technique further. The results from this averaging procedure are called nodal point stresses.

2.3 Non-linear Formulation of Strain and Stress

This section provides an introduction to the most common model and simulation technique for both 2D and 3D solid bodies in Non-Finite-Element-Method. According to Rutzmoser **Johannes**, in the first part of this introduction, we will pick a fictional 3D element with five nodes as object of study. The coordinate system is based on three coordinate: ξ_1 , ξ_2 and ξ_3 . Therefore the shape function of this element can be ordered in Voigt notation as follows:

$$N(\xi) = \begin{pmatrix} N_1(\xi) \\ N_2(\xi) \\ N_3(\xi) \\ N_4(\xi) \\ N_5(\xi) \end{pmatrix} \quad (2.5)$$

We can denote the coordinate of element as vector \mathbf{X}^e . The columns of \mathbf{X}^e stands for axis direction and the rows of \mathbf{X}^e stands for index of nodes. The matrix of \mathbf{X}^e is shown as:

$$\mathbf{X}^e = \begin{pmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ X_3 & Y_3 & Z_3 \\ X_4 & Y_4 & Z_5 \\ X_5 & Y_5 & Z_6 \end{pmatrix} \quad (2.6)$$

The coordinates \mathbf{X} of element in the initial configuration relies on the local coordinate of element ξ and the shape function N . It can be described as:

$$\mathbf{X}(\xi) = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = (\mathbf{X}^e)^T N = \begin{pmatrix} X_1 & X_2 & X_3 & X_4 & X_5 \\ Y_1 & Y_2 & Y_3 & Y_4 & Y_5 \\ Z_1 & Z_2 & Z_3 & Z_4 & Z_5 \end{pmatrix} \begin{pmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \end{pmatrix} \quad (2.7)$$

The displacements \mathbf{u} of any points of the element are interpolated from the nodal coordinate, just like it was done previous for the coordinates \mathbf{X} . So the displacements field can be shown as:

$$\mathbf{u}(\xi) = \begin{pmatrix} u_x(\xi) \\ u_y(\xi) \\ u_z(\xi) \end{pmatrix} = (\mathbf{u}^e)^T N = \begin{pmatrix} u_{x1} & u_{x2} & u_{x3} & u_{x4} & u_{x5} \\ u_{y1} & u_{y2} & u_{y3} & u_{y4} & u_{y5} \\ u_{z1} & u_{z2} & u_{z3} & u_{z4} & u_{z5} \end{pmatrix} \begin{pmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \end{pmatrix} \quad (2.8)$$

This approach is called isoparametric concept, that is possible to present all the parameters in Element. The method to calculate the magnitude of parameter in element is using the known value of parameters at every nodes and the shape function. An example of displacements can be formed like this:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{X}} = (\mathbf{u}^e)^T \frac{\partial \mathbf{N}}{\partial \mathbf{X}} = (\mathbf{u}^e)^T \tilde{\mathbf{B}}_0 \quad (2.9)$$

The deviation of parameter can be passed to the shape function. By using this approach, we can define a new term $\tilde{\mathbf{B}}_0 = \frac{\partial \mathbf{u}}{\partial \mathbf{X}}$. The expand of $\tilde{\mathbf{B}}_0$ can be derived as:

$$\frac{\partial \mathbf{N}}{\partial \mathbf{X}} = \tilde{\mathbf{B}}_0 = \begin{pmatrix} \frac{\partial N_1}{\partial X} & \frac{\partial N_1}{\partial Y} & \frac{\partial N_1}{\partial Z} \\ \frac{\partial N_2}{\partial X} & \frac{\partial N_2}{\partial Y} & \frac{\partial N_2}{\partial Z} \\ \frac{\partial N_3}{\partial X} & \frac{\partial N_3}{\partial Y} & \frac{\partial N_3}{\partial Z} \\ \frac{\partial N_4}{\partial X} & \frac{\partial N_4}{\partial Y} & \frac{\partial N_4}{\partial Z} \\ \frac{\partial N_5}{\partial X} & \frac{\partial N_5}{\partial Y} & \frac{\partial N_5}{\partial Z} \end{pmatrix} = \frac{\partial \mathbf{N}}{\partial \xi} \frac{\partial \xi}{\partial \mathbf{X}} \quad (2.10)$$

The first term of equation 2.10 can be directly presented with shape function. It is shown as:

$$\frac{\partial \mathbf{N}}{\partial \xi} = \begin{pmatrix} \frac{\partial N_1}{\partial \xi_1} & \frac{\partial N_1}{\partial \xi_2} & \frac{\partial N_1}{\partial \xi_3} \\ \frac{\partial N_2}{\partial \xi_1} & \frac{\partial N_2}{\partial \xi_2} & \frac{\partial N_2}{\partial \xi_3} \\ \frac{\partial N_3}{\partial \xi_1} & \frac{\partial N_3}{\partial \xi_2} & \frac{\partial N_3}{\partial \xi_3} \\ \frac{\partial N_4}{\partial \xi_1} & \frac{\partial N_4}{\partial \xi_2} & \frac{\partial N_4}{\partial \xi_3} \\ \frac{\partial N_5}{\partial \xi_1} & \frac{\partial N_5}{\partial \xi_2} & \frac{\partial N_5}{\partial \xi_3} \end{pmatrix} \quad (2.11)$$

The second term $\frac{\partial \xi}{\partial \mathbf{X}}$ is not straightforward to obtain. On the other hand it is much more easier to get the matrix $\frac{\partial \mathbf{X}}{\partial \xi}$. This matrix is called Jacobian matrix \mathbf{J} , which is a matrix of ξ with respect to x . The inverse Jacobian are denoted as \mathbf{J}^{-1} , which is the second term.

The deformation gradient $\mathbf{F} = \mathbf{H} + \mathbf{I}$ can be defined through the help matrix $\mathbf{H} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = (\mathbf{u}^e)^T \tilde{\mathbf{B}}_0$, which describes the mapping of an infinitesimal fiber in the initial state to its new position in the current configuration. Then, the Green-Lagrange strain tensor can be also formed as:

$$\mathbf{E} = \frac{1}{2} (\mathbf{H} + \mathbf{H}^T + \mathbf{H}^T \mathbf{H}) \quad (2.12)$$

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}) \quad (2.13)$$

The help matrix \mathbf{H} can be expressed more detailed as:

$$\mathbf{H} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \frac{\partial \mathbf{u}}{\partial \xi} \frac{\partial \xi}{\partial \mathbf{X}} = (\mathbf{u}^e)^T \frac{\partial \mathbf{N}}{\partial \xi} \left[(\mathbf{X}^e)^T \frac{\partial \mathbf{N}}{\partial \xi} \right]^{-1} \quad (2.14)$$

It is possible to calculate stress \mathbf{S} while strain is presented. One simply constitutive equation between stress \mathbf{S} (2. Piola-Kirchhoff-Stress tensor) and \mathbf{E} (Green-Lagrange-Strain tensor) can be formulated as:

$$\mathbf{S} = \mathbf{C} : \mathbf{E} \quad (2.15)$$

The result of stress is subsequently transfer into the degree of freedom at each nodes. The principle for calculation of stress can be formulated as follows:

In the total Lagrange approach, the principle of virtual work represents that the internal stress work $\delta \mathbf{W}_{int} = \int \boldsymbol{\sigma} : \delta \boldsymbol{\epsilon} d\Omega_0 = \int \mathbf{S} : \delta \mathbf{E} d\Omega_0$ equals the external node force $\delta \mathbf{W}_{ext} = (\delta \mathbf{u}^{e,v})^T \mathbf{f}_{nl}^v$

$$\delta \mathbf{W} = \delta \mathbf{W}_{ext} = (\delta \mathbf{u}^{e,v})^T \mathbf{f}_{int}^v = \int \mathbf{S} : \delta \mathbf{E} d\Omega_0 = \int (\delta \mathbf{E}^v)^T \mathbf{S}^v d\Omega_0 \quad (2.16)$$

The internal deformation work can be computed by matrix-vector-product in voigt-notation or direct product of two matrix with notation $(:)$. Now, we evaluate the variation of Green-Lagrange strain tensor $\delta \mathbf{E}$. From equation 2.13, it is obvious that the variation of tensor $\delta \mathbf{E}$ is determined by the variation of deformation gradient. According to above equation 2.14, the variation of deformation gradient can be transformed as:

$$\delta \mathbf{F} = \delta \mathbf{H} = (\delta \mathbf{u}^e)^T \frac{\delta \mathbf{N}}{\delta \mathbf{X}} = (\delta \mathbf{u}^e)^T \tilde{\mathbf{B}}_0 \quad (2.17)$$

Consequently, the variation of tensor $\delta \mathbf{E}$ as:

$$\delta \mathbf{E} = \frac{1}{2} (\delta \mathbf{F}^T \mathbf{F} + \mathbf{F}^T \delta \mathbf{F}) \quad (2.18)$$

Green-Lagrange-Strain are often represented in voigt-notation because of the complex matrix form in computer programming. Then, $\delta \mathbf{E}^v$ can be coupled with \mathbf{B}_0 -matrix as:

$$\delta \mathbf{E}^v = \mathbf{B}_0 \delta \mathbf{u}^{e,v} \quad (2.19)$$

The entries of \mathbf{B}_0 will be packed as a black box and direct implemented in the FEM simulation. Gathering the equation 2.16 and equation 2.19, the non-linear force results in:

$$\delta \mathbf{W} = (\delta \mathbf{u}^{e,v})^T \mathbf{f}_{int}^v = \int (\delta \mathbf{E}^v)^T \mathbf{S}^v d\Omega_0 = (\delta \mathbf{u}^{e,v})^T \int \mathbf{B}_0^T \mathbf{S}^v d\Omega_0 \quad (2.20)$$

$$\mathbf{f}_{int}^v = \int \mathbf{B}_0^T \mathbf{S}^v d\Omega_0 \quad (2.21)$$

The non-linear internal force of element concerning to coordinate of node can be integrated by \mathbf{B}_0 -matrix with second Piola-Kirchhoff-Stress tensor in voigt-notation. To obtain the tangential stiffness matrix, it is necessary to calculate the partial derivative of internal force \mathbf{f}_{int}^v respect to the nodal degree of freedom $\mathbf{u}^{e,v}$:

$$\frac{\partial \mathbf{f}_{int}^v}{\partial \mathbf{u}^{e,v}} = \mathbf{K} = \int \frac{\partial \mathbf{B}_0^T}{\partial \mathbf{u}^{e,v}} \mathbf{S}^v d\Omega_0 + \int \mathbf{B}_0^T \frac{\partial \mathbf{S}^v}{\partial \mathbf{u}^{e,v}} d\Omega_0 = \mathbf{K}_{geo} + \mathbf{K}_{mat} \quad (2.22)$$

Stiffness can be divide into two term, one for material stiffness matrix \mathbf{K}_{geo} , which is shown as:

$$\mathbf{K}_{mat} = \int \mathbf{B}_0^T \frac{\partial \mathbf{S}^v}{\partial \mathbf{u}^{e,v}} d\Omega_0 = \int \mathbf{B}_0^T \frac{\partial \mathbf{S}^v}{\partial \mathbf{E}^v} \frac{\partial \mathbf{E}^v}{\partial \mathbf{u}^{e,v}} d\Omega_0 = \int \mathbf{B}_0^T \mathbf{C}^{SE} \mathbf{B}_0 d\Omega_0 \quad (2.23)$$

The derivative of stress respect to strain is considered as Tangent-Modulus $\mathbf{C}^{SE} = \frac{\partial \mathbf{S}^v}{\partial \mathbf{E}^v}$ by constitutive equation. The derivative of strain is \mathbf{B}_0 as determined in equation 2.13.

The geometrical stiffness matrix is much more complex to derive. In a word, the internal work from equation 2.16 is reformulated by concerning relation between the deformation gradient and continuum mechanics $\mathbf{P} = \mathbf{S}\mathbf{F}^T$:

$$\delta \mathbf{W} = \mathbf{u}^e : \mathbf{f}_{int} = \int \delta \mathbf{F}^T : \mathbf{P} d\Omega_0 = \int \delta \mathbf{F}_{ij} \mathbf{P}_{ji} d\Omega_0 \quad (2.24)$$

The variation of deformation gradient $\delta \mathbf{F} = \delta \mathbf{u}^{eT} \mathbf{B}_0$. In index-notation form it can be written as $\delta \mathbf{F}_{ij} = \delta \mathbf{u}_{ki}^e \tilde{\mathbf{B}}_{kj}^0$. Then, the internal force in matrix notation can be derived as:

$$\delta \mathbf{W} = \delta \mathbf{u}^e : \mathbf{f}_{int} = \mathbf{u}_{ki}^e \mathbf{f}_{ki}^{int} = \int \delta \mathbf{F}_{ji} \mathbf{P}_{ij} d\Omega_0 = \delta \mathbf{u}_{ki}^e \int \tilde{\mathbf{B}}_{kj}^0 \mathbf{P}_{ji} d\Omega_0 = \delta \mathbf{u}^e : \int \tilde{\mathbf{B}}_0 \mathbf{P} d\Omega_0 \quad (2.25)$$

$$\mathbf{f}_{int} = \int \tilde{\mathbf{B}}_0 \mathbf{P} d\Omega_0 \quad (2.26)$$

The tangential stiffness matrix is defined by changing the force vector, which is determined over time concerning with the velocity of nodal displacement \mathbf{u}^e . Because Jacobi-Matrix is relative complex to build. Therefore, the derivative of internal force $\dot{\mathbf{f}}_{int}$ in time as follows:

$$\dot{\mathbf{f}}_{int} = \int \tilde{\mathbf{B}}_0 \dot{\mathbf{S}} \mathbf{F}^T d\Omega_0 + \int \tilde{\mathbf{B}}_0 \mathbf{S} \dot{\mathbf{F}}^T d\Omega_0 \quad (2.27)$$

The first term corresponds to material stiffness, because the time derivative of second Piola-Kirchhoff-Stress tensor is related with time-dependent part. And the second term corresponds to geometrical stiffness as shown in equation 2.22. Because the time derivative $\dot{\mathbf{F}}$ equals $\dot{\mathbf{u}}^{eT} \tilde{\mathbf{B}}_0$, equation 2.27 can be transformed as:

$$\dot{\mathbf{f}}_{int,geo} = \int \tilde{\mathbf{B}}_0 \mathbf{S} \tilde{\mathbf{B}}_0 \Omega_0 \dot{\mathbf{u}}^e \quad (2.28)$$

The geometrical stiffness matrix is given, because the temporal variation of non-linear force is coupled with the displacements over the tangential stiffness matrix.

$$\dot{\mathbf{f}}_{int,geo} = \mathbf{K}_{geo} \dot{\mathbf{u}}^e = \int \tilde{\mathbf{B}}_0 \mathbf{S} \tilde{\mathbf{B}}_0 \Omega_0 \dot{\mathbf{u}}^e \quad (2.29)$$

The last step to determine the non-linear force and the tangential stiffness matrix is to integrate on domain $d\Omega_0$. There are two concept for this integration to mention: the first one is transformation from domain $d\Omega_0$ to reference coordinate system:

$$\int \mathbf{f}(\mathbf{x}) d\Omega = \int \mathbf{f}(\mathbf{x}) \frac{\partial \xi}{\partial \mathbf{x}} d\Omega = \int \mathbf{f}(\mathbf{x}) \frac{\partial \Omega}{\partial \xi} d\xi = \int \mathbf{f}(\mathbf{x}) \det \left(\frac{\partial \mathbf{X}}{\partial \xi} \right) d\xi \quad (2.30)$$

For the transformation of integration, it is necessary to adapt the bound of integration. It is in FEM a pleasant benefit, its advantage is the new bound of integration is the standard bound of reference element. So it is easier to determine the bound of integration. And the Jacobi-matrix is ready to use, because we have once the inverse of Jacobi-matrix calculated.

2.3.1 Gauss Integration

Numerical integration plays a important role in finite element. Gauss integration, as an efficient approach compared to other numerical integration, integrate a function $f(\xi)$ on the spatial domain Ω is replaced by a summation of certain function values at the so-called Gauss points $\tilde{\xi}$ which are each multiplied by a scalar (weight) ω . In total $numgp_1 \times numgp_2 \times numgp_3$ Gauss points are used:

$$\int_E f(\xi) d\xi_1 d\xi_2 d\xi_3 = \sum_{i=1}^{numgp_1} \sum_{j=1}^{numgp_2} \sum_{k=1}^{numgp_3} f(\tilde{\xi}_1^i \tilde{\xi}_2^j \tilde{\xi}_3^k) \cdot w_1^i w_2^j w_3^k \quad (2.31)$$

In which we used corresponding weights for each of the three spatial directions. This allows to use Gauss points coordinates and weights tabulated for the one-dimensional case. The location and weights of these points can be given analytically in Table 2.1.

2.4 Extrapolation from Gauss Points

2.4.1 Quad4

This will be explained for the four-node bilinear quadrilateral. The normal Gauss integration rule for element stiffness evaluation is 2×2 , as illustrated in Figure 2.3. The stresses are calculated at the Gauss points, which are identified as k_1^G , k_2^G , k_3^G and k_4^G in Figure 2.3. Point k_i^G is closest to node k_i^E so it is seen that Gauss point numbering essentially follows element node numbering in the counterclockwise sense. The natural coordinates of these points are listed in Table 2.2. The stresses are evaluated at these Gauss points by passing these natural coordinates to the shape function subroutine. Then each stress component is “carried” to the corner nodes k_1^E through k_4^E through a bilinear extrapolation based on the computed values at k_1^G through k_4^G . To understand the extrapolation procedure more clearly it is convenient to consider the region bounded by the Gauss points as an “internal element” or “Gauss element”. This interpretation is depicted in Figure 2.3. The Gauss element, denoted by (G), is also a four-node quadrilateral. The coordinate of node for element and Gauss element can be represented as k_i^G and k_i^E , respectively. Its quadrilateral (natural) coordinates are denoted by ξ' and η' . These are linked to ξ' and η' by the simple relations from Gauss-Legendre quadrature in Table 2.2.

$$k_i^G = \frac{k_i^E}{\sqrt{3}}, \quad k_i^E = k_i^G \sqrt{3} \quad (2.32)$$

Table 2.1: Gauss-Legendre points and weights

$numgp$	i	$\tilde{\xi}^i$	ω^i
1	1	0	2
2	1	$-1/\sqrt{3}$	1
	2	$+1/\sqrt{3}$	1
3	1	$-\sqrt{3/5}$	5/9
	2	0	8/9
	3	$+\sqrt{3/5}$	5/9
4	1	$-\sqrt{(15 + \sqrt{120})/35}$	$(18 - \sqrt{30})/36$
	2	$-\sqrt{(15 - \sqrt{120})/35}$	$(18 + \sqrt{30})/36$
	3	$+\sqrt{(15 - \sqrt{120})/35}$	$(18 + \sqrt{30})/36$
	4	$+\sqrt{(15 + \sqrt{120})/35}$	$(18 - \sqrt{30})/36$
5	1	$-1/3\sqrt{5 + 2\sqrt{10/7}}$	$(332 - 13\sqrt{70})/900$
	2	$-1/3\sqrt{5 - 2\sqrt{10/7}}$	$(332 + 13\sqrt{70})/900$
	3	0	128/255
	4	$+1/3\sqrt{5 - 2\sqrt{10/7}}$	$(332 + 13\sqrt{70})/900$
	5	$+1/3\sqrt{5 + 2\sqrt{10/7}}$	$(332 - 13\sqrt{70})/900$

Table 2.2: Natural Coordinate of Quad4

Corner node	ξ	η	ξ'	η'	Gauss node	ξ	η	ξ'	η'
1	-1	-1	$-\sqrt{3}$	$-\sqrt{3}$	1'	$1/\sqrt{3}$	$-1/\sqrt{3}$	-1	-1
2	+1	-1	$+\sqrt{3}$	$-\sqrt{3}$	2'	$1/\sqrt{3}$	$1/\sqrt{3}$	+1	-1
3	+1	+1	$+\sqrt{3}$	$+\sqrt{3}$	3'	$1/\sqrt{3}$	$1/\sqrt{3}$	+1	+1
4	-1	+1	$-\sqrt{3}$	$+\sqrt{3}$	4'	$-1/\sqrt{3}$	$1/\sqrt{3}$	-1	+1

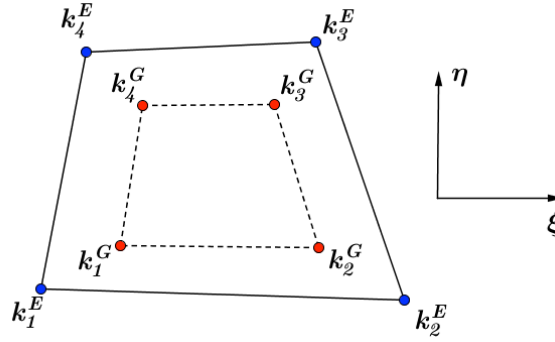


Figure 2.3: Quad4 in element coordinate and Gauss element coordinate.

The element geometry and natural coordinates are shown in Figure 2.3. Only one type of node (corner) and associated shape function is present. Consider node 1 as typical. Inspection of Figure 2.4 suggests trying

$$N_1^e = c_1 L_{2-3} L_{3-4} \quad (2.33)$$

This plainly vanishes over nodes 2, 3 and 4, and can be normalized to unity at node 1 by adjusting c_1 . By construction it vanishes over the sides 2-3 and 3-4 that do not belong to 1. The equation of side 2-3 is $\xi = 1$, or $\xi - 1 = 0$. The equation of side 3-4 is $\eta = 1$, or $\eta - 1 = 0$. Replacing in Equation (2.33) yields

$$N_1^e(\xi, \eta) = c_1 (\xi - 1)(\eta - 1) = c_1 (1 - \xi)(1 - \eta) \quad (2.34)$$

To find c_1 , evaluate at node 1, the natural coordinates of which are $\xi = \eta = -1$:

$$N_1^e(-1, -1) = c_1 \times 2 \times 2 = 4c_1 = 1 \quad (2.35)$$

Hence $c_1 = \frac{1}{4}$ and the shape functions is

$$N_1^e = \frac{1}{4} (1 - \xi)(1 - \eta) \quad (2.36)$$

For the other three nodes the procedure is the same, traversing the element cyclically. It can be verified that the general expression of the shape functions for this element is

$$N_i^e = \frac{1}{4} (1 + \xi_i \xi) (1 + \eta_i \eta) \quad (2.37)$$

Following this general expression, the shape functions of Node 2, 3 and 4 are demonstrated as

$$N_2^e = \frac{1}{4} (1 + \xi)(1 - \eta) \quad (2.38)$$

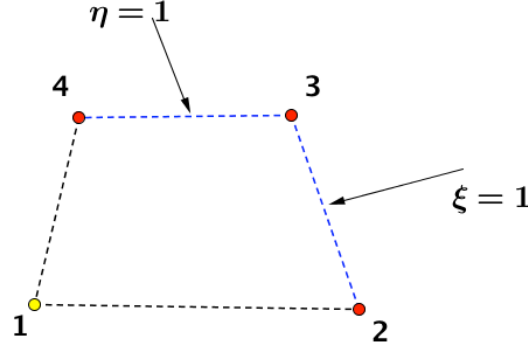


Figure 2.4: Equation of side opposite corner 1 for Quad4.

$$N_3^e = \frac{1}{4} (1 + \xi) (1 + \eta) \quad (2.39)$$

$$N_4^e = \frac{1}{4} (1 - \xi) (1 + \eta) \quad (2.40)$$

When we have all the shape function for Gauss element, it is able to extrapolate the component (stress, strain, etc) from Gauss points k_i^G to corner nodes k_i^E . According to 2.2 and 2.3, we have the corner nodes in Gauss coordinate: $k_1^E(-\sqrt{3}, -\sqrt{3})$, $k_2^E(\sqrt{3}, -\sqrt{3})$, $k_3^E(\sqrt{3}, \sqrt{3})$, $k_4^E(-\sqrt{3}, \sqrt{3})$. The extrapolation process is to replace all the corner nodes in Gauss coordinate into the shape function of each Gauss points. Then we obtain:

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} = \begin{pmatrix} 1 + \frac{1}{2}\sqrt{3} & -\frac{1}{2} & 1 - \frac{1}{2}\sqrt{3} & -\frac{1}{2} \\ -\frac{1}{2} & 1 + \frac{1}{2}\sqrt{3} & -\frac{1}{2} & 1 - \frac{1}{2}\sqrt{3} \\ 1 - \frac{1}{2}\sqrt{3} & -\frac{1}{2} & 1 + \frac{1}{2}\sqrt{3} & -\frac{1}{2} \\ -\frac{1}{2} & 1 - \frac{1}{2}\sqrt{3} & -\frac{1}{2} & 1 + \frac{1}{2}\sqrt{3} \end{pmatrix} \begin{pmatrix} w_1' \\ w_2' \\ w_3' \\ w_4' \end{pmatrix} \quad (2.41)$$

Here w' means strains and stresses for our case. So the w' part can be written as:

$$\begin{pmatrix} w_1' \\ w_2' \\ w_3' \\ w_4' \end{pmatrix} = \begin{pmatrix} \epsilon_{1xx} & \epsilon_{1yy} & \epsilon_{1zz} & 2\epsilon_{1yz} & 2\epsilon_{1xz} & 2\epsilon_{1xy} \\ \epsilon_{2xx} & \epsilon_{2yy} & \epsilon_{2zz} & 2\epsilon_{2yz} & 2\epsilon_{2xz} & 2\epsilon_{2xy} \\ \epsilon_{3xx} & \epsilon_{3yy} & \epsilon_{3zz} & 2\epsilon_{3yz} & 2\epsilon_{3xz} & 2\epsilon_{3xy} \\ \epsilon_{4xx} & \epsilon_{4yy} & \epsilon_{4zz} & 2\epsilon_{4yz} & 2\epsilon_{4xz} & 2\epsilon_{4xy} \end{pmatrix} \quad (2.42)$$

$$\begin{pmatrix} w_1' \\ w_2' \\ w_3' \\ w_4' \end{pmatrix} = \begin{pmatrix} \sigma_{1xx} & \sigma_{1yy} & \sigma_{1zz} & \sigma_{1yz} & \sigma_{1xz} & \sigma_{1xy} \\ \sigma_{2xx} & \sigma_{2yy} & \sigma_{2zz} & \sigma_{2yz} & \sigma_{2xz} & \sigma_{2xy} \\ \sigma_{3xx} & \sigma_{3yy} & \sigma_{3zz} & \sigma_{3yz} & \sigma_{3xz} & \sigma_{3xy} \\ \sigma_{4xx} & \sigma_{4yy} & \sigma_{4zz} & \sigma_{4yz} & \sigma_{4xz} & \sigma_{4xy} \end{pmatrix} \quad (2.43)$$

2.4.2 Quadrangle with quadratic shape function and eight nodes: Quad8

For Quad8(Quadrangle with quadratic shape function and eight nodes) we have various choice for the type of Gauss element. We usually use four, eight and nine nodes for three quadrilateral elements. Here we introduce the Gauss element with nine nodes. The nine-nodes quadrilateral has three types of shape functions, which are associated with corner, midside nodes and center node, respectively. The element coordinate and Gauss element coordinate are illustrated in Figure 2.5

The lines whose product is used to construct three types of shape functions are illustrated in Figure for nodes 1, 5 and 9, respectively. The technique has been sufficiently illustrated in previous examples. Here the summary of calculation for nodes 1, 5 and 9, which are taken as representatives of the three types: The three types of shape function are represented in Figure 2.6.

$$N_1^e = c_1 L_{2-3} L_{3-4} L_{5-7} L_{6-8} = c_1 (\xi - 1)(\eta - 1) \xi \eta \quad (2.44)$$

$$N_5^e = c_5 L_{2-3} L_{1-4} L_{6-8} L_{3-4} = c_5 (\xi - 1)(\xi + 1) \eta (\eta - 1) = c_5 (1 - \xi^2) \eta (1 - \eta) \quad (2.45)$$

$$N_9^e = c_9 L_{1-2} L_{2-3} L_{3-4} L_{4-1} = c_9 (\xi - 1)(\eta - 1)(\xi + 1)(\eta + 1) = c_9 (1 - \xi^2)(1 - \eta^2) \quad (2.46)$$

Imposing the normalization conditions result in:

$$c_1 = \frac{1}{4}, \quad c_5 = -\frac{1}{2}, \quad c_9 = 1$$

By following this approach all the shape functions can be calculated

$$N_1 = \frac{1}{4} (\xi - 1)(\eta - 1) \cdot \xi \cdot \eta \quad (2.47)$$

$$N_2 = \frac{1}{4} (\xi + 1)(\eta - 1) \cdot \xi \cdot \eta \quad (2.48)$$

$$N_3 = \frac{1}{4} (\xi + 1)(\eta + 1) \cdot \xi \cdot \eta \quad (2.49)$$

$$N_4 = \frac{1}{4} (\xi - 1)(\eta + 1) \cdot \xi \cdot \eta \quad (2.50)$$

$$N_5 = \frac{1}{2} (1 + \xi)(1 + \xi)(\eta - 1) \cdot \eta \quad (2.51)$$

$$N_6 = \frac{1}{2} (1 + \eta)(1 - \eta)(\xi + 1) \cdot \xi \quad (2.52)$$

$$N_7 = \frac{1}{2} (1 + \xi)(1 + \xi)(\eta + 1) \cdot \eta \quad (2.53)$$

$$N_8 = \frac{1}{2} (1 + \eta)(1 - \eta)(\xi - 1) \cdot \xi \quad (2.54)$$

$$N_9 = (1 + \xi)(1 - \xi)(1 + \eta)(1 - \eta) \quad (2.55)$$

Same as Quad4, the extrapolation function can be expressed by replacing corner nodes in Gauss coordinate into the shape function of Gauss points:

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 & a_2 & a_4 & a_5 & a_5 & a_4 & a_6 \\ a_2 & a_1 & a_2 & a_3 & a_4 & a_4 & a_5 & a_5 & a_6 \\ a_3 & a_2 & a_1 & a_2 & a_5 & a_4 & a_4 & a_5 & a_6 \\ a_2 & a_3 & a_2 & a_1 & a_5 & a_5 & a_4 & a_4 & a_6 \\ 0 & 0 & 0 & 0 & a_7 & 0 & a_8 & 0 & a_9 \\ 0 & 0 & 0 & 0 & 0 & a_7 & 0 & a_8 & a_9 \\ 0 & 0 & 0 & 0 & a_8 & 0 & a_7 & 0 & a_9 \\ 0 & 0 & 0 & 0 & 0 & a_8 & 0 & a_7 & a_9 \end{pmatrix} \begin{pmatrix} w_1' \\ w_2' \\ w_3' \\ w_4' \\ w_5' \\ w_6' \\ w_7' \\ w_8' \\ w_9' \end{pmatrix} \quad (2.56)$$

$$a_1 = +2.1869398$$

$$a_2 = +0.2777778$$

$$a_3 = +0.0352824$$

$$a_4 = -0.9858870$$

$$a_5 = -0.1252241$$

$$a_6 = +0.4444444$$

$$a_7 = +1.4788331$$

$$a_8 = +0.1878361$$

$$a_9 = -0.6666666$$

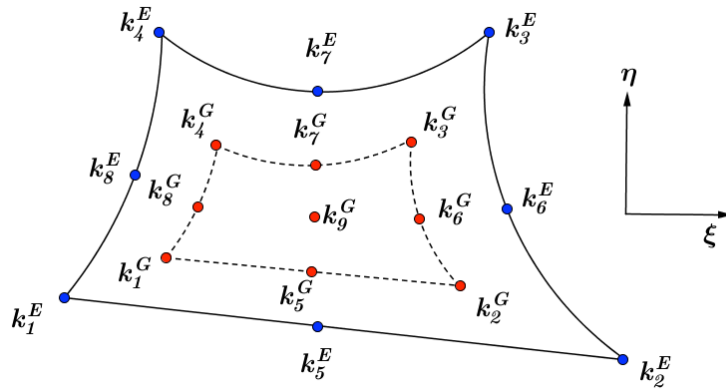


Figure 2.5: Quad8 in element coordinate and Gauss element coordinate.

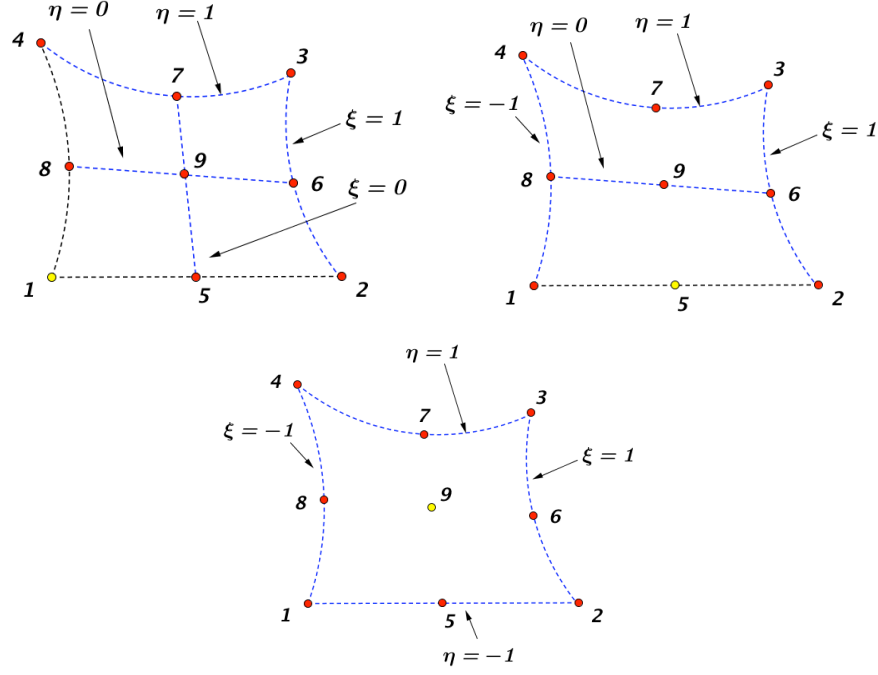


Figure 2.6: Equation of side opposite corner 1 for Quad8.

2.4.3 Triangle with three nodes: Tri3

The geometry of the 3-node triangle shown in Figure 2.7 is specified by the location of its three corner nodes on the $\{x, y\}$ plane

The shape function for triangular element has a different form, which compares with quadrilateral elements. The three shape functions have the simply own coordinates - the triangular coordinates: $N_i = \xi_i$ for $i = 1, 2, 3$. The shape function can be derived from a method as follows:

The equation of the triangle side opposite to node i is $L_{j-k} = \xi_i = 0$, where j and k are the cyclic permutations of i . Here symbol L_{j-k} denotes the left hand side of the homogeneous equation of the natural coordinate line that passes through node points j and k . See Figure 2.8 for $i = 1, j = 2$ and $k = 3$. Hence the obvious suppose is:

$$N_i^e = c_i L_i \quad (2.57)$$

To find c_1 , evaluate $N_1^e(\xi_1, \xi_2, \xi_3)$ at node 1. The triangular coordinates of this node are $\xi_1 = 1, \xi_2 = \xi_3 = 0$. $N_1^e(1, 0, 0) = c_1 \times 1 = 1$. so $c_1 = 1$ and analogically $c_2 = 1, c_3 = 1$

$$N_1^e = L_1, \quad N_2^e = L_2, \quad N_3^e = L_3 \quad (2.58)$$

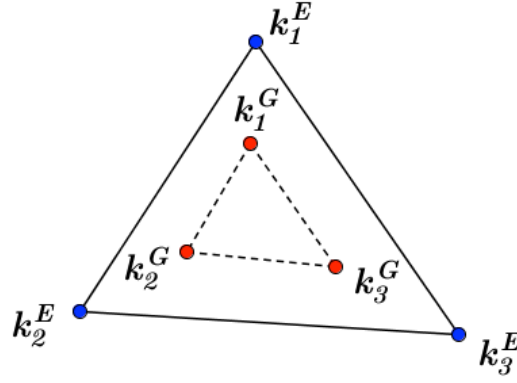


Figure 2.7: Tri3 in element coordinate and Gauss element coordinate.

2.4.4 Second order triangle with six nodes: Tri6

The geometry of the six-node quadratic is shown in Figure XXX. Inspection reveals two types of nodes: corners(1, 2 and 3) and midside nodes(4, 5 and 6). For both cases it is necessary to product the two linear functions in the triangular coordinates because the shape function should be quadratic. These functions are illustrated in Figures XXX for corner node 1 and midside node 4, respectively.

For corner node 1, inspection of Figure 2.10 at top left side suggests trying

$$N_1^e = c_1 L_{2-3} L_{4-6} \quad (2.59)$$

N_1^e will vanish over 2-5-3 and 4-6. This makes the function zero at node 2,3,4,5,6, as is obvious upon inspection of Figure XXX, while being nonzero at 1. The value can be adjusted to be unity if c_1 is appropriately chosen. The equations of the lines that appear (XXX equation upper) are

$$L_{2-3} : \xi_1 = 0, \quad L_{4-6} : \xi_1 - \frac{1}{2} = 0 \quad (2.60)$$

Replacing into (eq upper upper)

$$N_1^e = c_1 \xi_1 \left(\xi_1 - \frac{1}{2} \right) \quad (2.61)$$

Same as triangle with three nodes, $N_1^e(1, 0, 0) = c_1 \times 1 \times \frac{1}{2} = 1$. Then $c_1 = 2$ can be calculated and finally

$$N_1^e = 2\xi_1 \left(\xi_1 - \frac{1}{2} \right) = \xi_1 (2\xi_1 - 1) \quad (2.62)$$

For midside node 4, inspection of Figure XXX suggests trying

$$N_4^e = c_4 L_{2-3} L_{1-3} \quad (2.63)$$

Table 2.3: Natural Coordinate of Tri3

Corner node	ξ	η	ξ'	η'	Gauss node	ξ	η	ξ'	η'
1	-1	-1	-5/3	-5/3	1'	-2/3	-2/3	-1	-1
2	+1	-1	+7/3	-5/3	2'	+1/3	-2/3	+1	-1
3	-1	+1	-5/3	+7/3	3'	-2/3	+1/3	-1	+1

The equation of sides L_{2-3} and L_{1-3} are $\xi_1 = 0$ and $\xi_2 = 0$, respectively. Therefore $N_4^e(\xi_1, \xi_2, \xi_3) = c_4 \xi_1 \xi_2$. To find c_4 , evaluate this function at node 4, the triangular coordinates of which are $\xi_1 = \xi_2 = \frac{1}{2}, \xi_3 = 0$. Then $N_4^e\left(\frac{1}{2}, \frac{1}{2}, 0\right) = c_4 \times \frac{1}{2} \times \frac{1}{2} = 1$. Hence $c_4 = 4$, which the shape function gives

$$N_4^e = 4\xi_1\xi_2 \quad (2.64)$$

The rest shape function can be calculated by using same approach.

2.4.5 Tetraeder element with four nodes: Tet4

2.5 Coordinate Transformations

Quantities that are closely linked with the element geometry are best expressed in triangular coordinates. On the other hand, quantities such as displacements, strains and stresses are usually expressed in the Cartesian system. Thus we need transformation equations through which it is possible to pass from one coordinate system to the other. Cartesian and triangular coordinates are linked by the relation.

$$\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} \quad (2.65)$$

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} x_2y_3 - x_3y_2 & y_2 - y_3 & x_3 - x_2 \\ x_3y_1 - x_1y_3 & y_3 - y_1 & x_1 - x_3 \\ x_1y_2 - x_2y_1 & y_1 - y_2 & x_2 - x_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} 2A_{23} & y_{23} & x_{32} \\ 2A_{31} & y_{31} & x_{13} \\ 2A_{12} & y_{12} & x_{21} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \quad (2.66)$$

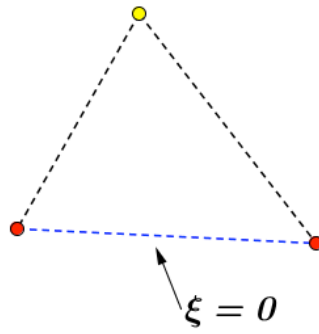


Figure 2.8: Equation of side opposite corner 1 for Tri3.

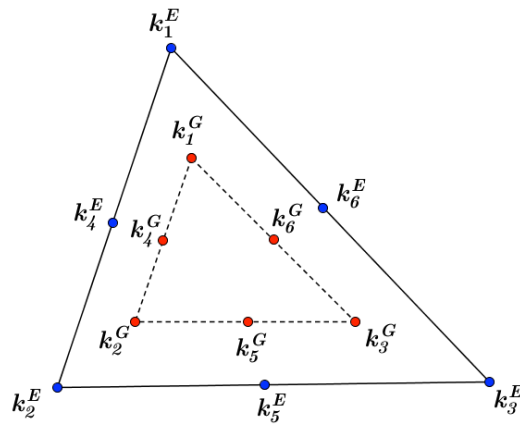


Figure 2.9: Tri6 in element coordinate and Gauss element coordinate.

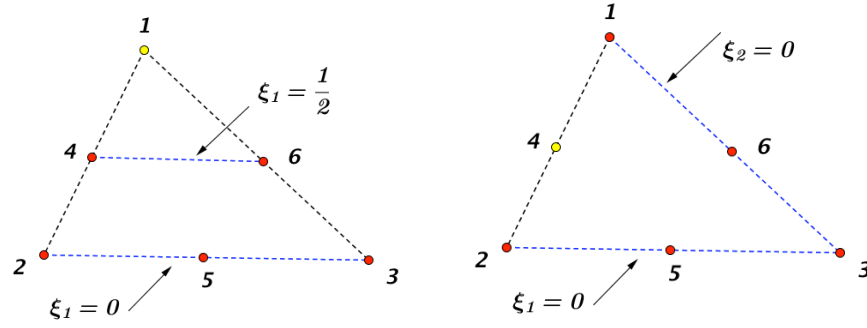


Figure 2.10: Equation of side opposite corner 1 for Tri6.

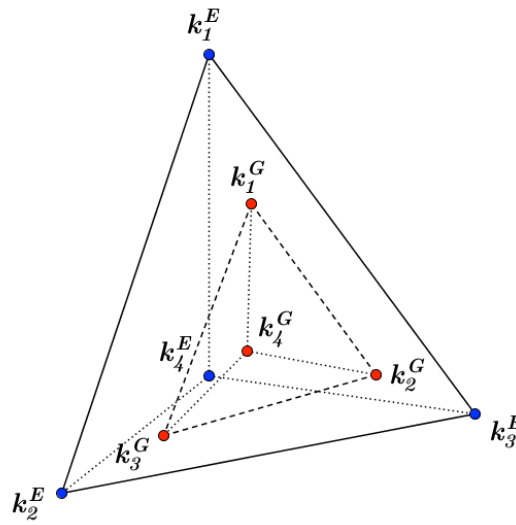


Figure 2.11: Tri4 in element coordinate and Gauss element coordinate.

Table 2.4: Tetrahedral Coordinate of Tet4

Corner node	L1	L2	L3	L4	L1'	L2'	L3'	L4'
1	1	0	0	0	α	β	β	β
2	0	1	0	0	β	α	β	β
3	0	0	1	0	β	β	α	β
4	0	0	0	1	β	β	β	α
Gauss node	L1	L2	L3	L4	L1'	L2'	L3'	L4'
1'	α	β	β	β	1	0	0	0
2'	β	α	β	β	0	1	0	0
3'	β	β	α	β	0	0	0	1
4'	β	β	β	α	0	0	0	1

$\alpha = 0.58541020$; $\beta = 0.13819660$

Chapter 3

Programming Implementation

3.1 Pre-Processing

The purpose of Pre-Processing is to build a numerical model and export as two lists- node list and element list. The data of model are ultimately imported into AMfe Toolbox. ANSYS Parametric Design Language (APDL) code is the tool to create a meshed geometry in ANSYS. A simple meshed geometry are shown in Figure 3.1. Task step for building model contains a series of related path. An example of a path step is:

Geometry \Rightarrow Element Type \Rightarrow Material \Rightarrow Mesh \Rightarrow Boundary Condition

After building the model, numerical data can be exported as node list and element list. Node list contains coordinate all the nodes of geometry. Element list are formed with the attributes of each elements and the index of node with belongs to element. The element attributes are assigned to different parts of model by 'pointing' to the appropriate entries in the element list. The pointers are simply a set of reference numbers that include the element index (ELEM), material number (MAT), element type number (TYP), real constant set number (REAL), a coordinate system number (ESY) and section ID number (SEC). Connecting with the example in Figure 3.1, both lists are depicted in Table 3.1 and Table 3.2. The numerical information of both lists are captured with regular expression approach and stored in AMfe Toolbox. For different data it is necessary to store in specific data type. Coordinate of each node and the index of nodes from elements are stored as Numpy array. NumPy is the fundamental package for scientific computing with Python. The advantage of Numpy for storage of node and element information depends on its efficient multi-dimensional container of generic data. It contains a powerful N-dimensional array object and sophisticated (broadcasting) functions. Moreover, it is possible for integrating Fortran code. General attributes of element can be stored as a efficient form-Pandas DataFrames. Pandas is a powerful data analysis toolkit, which provides fast, flexible, and expressive data structures designed to make working with relational or labeled data both easy and intuitive. Pandas DataFrames is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). Arithmetic operations align on both row and column labels. This format can be thought of as a dictionary-like container for Series objects.

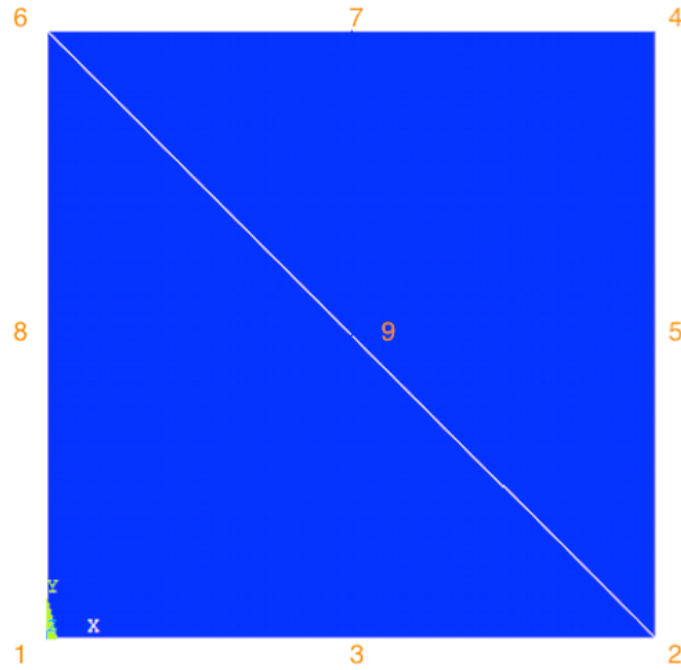


Figure 3.1: A simple meshed geometry

Table 3.1: Node list exported from ANSYS

NODE	X	Y	Z	THXY	THYZ	THZX
1	0.0000	0.0000	0.0000	0.00	0.00	0.00
2	50.000	0.0000	0.0000	0.00	0.00	0.00
3	25.000	0.0000	0.0000	0.00	0.00	0.00
4	50.000	50.000	0.0000	0.00	0.00	0.00
5	50.000	25.000	0.0000	0.00	0.00	0.00
6	0.0000	50.000	0.0000	0.00	0.00	0.00
7	25.000	50.000	0.0000	0.00	0.00	0.00
8	0.0000	25.000	0.0000	0.00	0.00	0.00
9	25.000	25.000	0.0000	0.00	0.00	0.00

Table 3.2: Element list exported from ANSYS

ELEM	MAT	TYP	REL	ESY	SEC	Node	Node	Node	Node	Node	Node
1	1	1	1	0	1	6	2	4	9	5	7
2	1	1	1	0	1	6	1	2	8	3	9

bibid Interelement Averaging can be as future research .. **Johannes**

Chapter 4

Calculation of Element Strains and Stresses

4.1 Shape Function

4.1.1 Quadrilateral Elements

4.1.2 Triangular and Tetrahedral Elements

Chapter 5

Bestimmung des optimalen Tunings k

Chapter 6

Zusammenfassung und Ausblick

Averaging-Methode

Die *Averaging-Methode* oder *Methode der Mittelwertbildung* (engl. *averaging method*) wird zur Analyse nichtlinearer Oszillationen eines dynamischen Systems verwendet **Mitropolsky:Averaging**. Die Grundidee der Methode ist die Approximation des Ausgangssystems durch ein gemitteltes System, welches dadurch deutlich einfacher zu untersuchen ist **Hagedorn:NichtlinSchwingungen1978**. Aus dem Verständnis der Dynamik des gemittelten Systems können Rückschlüsse auf die Dynamik des Ausgangssystems gezogen werden **Hagedorn:NichtlinSchwingungen1978**. Dabei wird durch Zeitintegration über eine Periode des Ausgangssystems der Einfluss der hochfrequenten Dynamiken herausgefiltert **Hagedorn:NichtlinSchwingungen1978**. Die wichtigsten Zusammenhänge zur praktischen Anwendung, wie sie im Rahmen der Herleitungen in dieser Arbeit benötigt werden, sind im Folgenden kurz dargelegt, wobei hier die einfachste Form der Mittelwertbildung, die *periodische Mittelwertbildung* **Sanders:AveragingMethods2007** betrachtet wird. Die angegebenen Zusammenhänge geben den Grundgedanken der Averaging-Methode wieder. Weitere Details, Beweise und Verallgemeinerungen sind in **Sanders:AveragingMethods2007** zu finden.

Ausgegangen wird von einem System in der Form

$$\dot{\vec{x}} = \epsilon \vec{f}(\vec{x}, t) + \mathcal{O}(\epsilon^2), \quad \vec{x}(0) = \vec{a}, \quad (6.1)$$

wobei ϵ ein kleiner Parameter und \vec{f} ein T -periodisches Vektorfeld in der Zeit t ist.

Vernachlässigung von Termen höherer Ordnung und Mittelwertbildung durch Integration über t führt auf die *gemittelte Gleichung*

$$\dot{\vec{z}} = \epsilon \vec{\bar{f}}(\vec{z}), \quad \vec{z}(0) = \vec{a}, \quad (6.2)$$

mit

$$\vec{\bar{f}}(\vec{z}) = \frac{1}{T} \int_0^T \vec{f}(\vec{z}, s) ds. \quad (6.3)$$

Das grundlegende Ergebnis dieser Mittelung ist, dass die Lösungen des gemittelten Systems nahe (von Ordnung ϵ) an den Lösungen des Ausgangssystems in einem Zeitintervall der Ordnung $1/\epsilon$ sind **Sanders:AveragingMethods2007** was sich als

$$\|\vec{x}(t) - \vec{z}(t)\| \leq c\epsilon \quad \text{für} \quad 0 \leq t \leq \frac{L}{\epsilon} \quad (6.4)$$

mit positiven Konstanten c und L schreiben lässt. Unter weiter spezifizierten Anforderungen, welche für reale Systeme in der Regel zutreffen, bleiben die Lösungen des gemittelten Systems

auch für das Zeitintervall $t \in [0, \infty)$ nahe (von Ordnung ϵ) an den Lösungen des Ausgangssystems **Sanders:AveragingMethods2007**

Rotationspendel

Disclaimer

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Garching, 30.05. 2016

(Signature)