

---

Institut für Festigkeitslehre

Christian C. Celigoj

---

# The Finite Element Method

SOOFEA MATLAB<sup>©</sup>

Software for Object Oriented Finite Element Analysis

Computer Exercises

*Michael Hammer* ©

*June 2011, Version 1.0 (alpha)*

---

Graz University of Technology  
Faculty for Mechanical Engineering  
Kopernikusgasse 24, A-8010 Graz

---

# Contents

<b>1</b>	<b>Preface</b>	<b>4</b>
1.1	Course Chronology . . . . .	4
<b>2</b>	<b>Interpolation</b>	<b>6</b>
2.1	Lagrange Polynomials . . . . .	6
2.1.1	First Derivative . . . . .	8
2.2	Multidimensional Interpolation - Shape Functions . . . . .	9
2.2.1	First Derivative . . . . .	12
2.3	Implementation . . . . .	13
2.3.1	<code>model.shape.Lagrange</code> . . . . .	13
2.3.2	<code>model.shape.Shape</code> and <code>model.shape.QuadShape</code> . . . . .	14
2.4	Examples . . . . .	16
<b>3</b>	<b>Numerical Integration</b>	<b>19</b>
3.1	Gauß-Legendre Quadrature . . . . .	19
3.2	Mapping of Integration Intervall . . . . .	20
3.3	Multidimensional Integration . . . . .	22
3.4	Implementation . . . . .	23
3.4.1	<code>model.IntegrationPoint</code> . . . . .	23
3.4.2	Input Handling of the Integration Points . . . . .	23
3.4.3	<code>math.NumInt</code> . . . . .	24
3.5	Examples . . . . .	25
<b>4</b>	<b>The Finite Element Method</b>	<b>26</b>
4.1	Elasticity Theory . . . . .	26
4.1.1	Strain . . . . .	27
4.1.2	Stress . . . . .	28
4.1.3	Strain-Stress Relation and Voigt's Notation . . . . .	29
4.1.4	Virtual Work Principle . . . . .	31
4.2	The Discretization . . . . .	32
4.2.1	Isoparametric Concept - The Jacobian Matrix . . . . .	34
4.2.2	Discrete Strains . . . . .	34
4.2.3	The System of Equations . . . . .	36
4.2.4	Assembling Algorithm . . . . .	37
4.3	Integration of Dirichlet Boundary Conditions . . . . .	40
4.4	Integration of Neumann Boundary Conditions . . . . .	41
4.5	Implementation . . . . .	41
4.5.1	<code>analyze.Jacobian</code> . . . . .	41

4.5.2	<code>analyze.Analysis</code> . . . . .	42
4.5.3	<code>analyze.ElementImpl</code> in particular <code>analyze.QuadImpl</code> . . . . .	44
4.6	Examples . . . . .	45
<b>5</b>	<b>Software Framework</b>	<b>46</b>
5.1	Code Structure and Design Rules . . . . .	46
5.1.1	Naming Standard . . . . .	46
5.2	The Model . . . . .	47
5.2.1	Node . . . . .	49
5.2.2	<code>ElementType</code> . . . . .	51
5.2.3	<code>Element</code> . . . . .	52
5.2.4	Degrees of Freedom . . . . .	53
5.2.5	Load . . . . .	54
5.2.6	Material . . . . .	54
5.3	The Math Module . . . . .	55
5.4	Preprocessing . . . . .	55
5.4.1	Input Handling . . . . .	56
5.5	The Analyze Module . . . . .	57
5.6	Postprocessing . . . . .	57
5.7	Examples . . . . .	58
	<b>List of Figures</b>	<b>63</b>
	<b>Bibliography</b>	<b>65</b>

# 1 Preface

This script contains notes and hints to accompany the course on finite element method computer exercises.

## 1.1 Course Chronology

1. Lecture:
  - Comprehensive introduction to MATLAB<sup>®</sup> (matrices, handles)
  - Comprehensive introduction to object oriented programming paradigms (classes, inheritance, interfaces)
2. Lecture: Chapter 2 especially section 2.1
  - Theory of interpolation
  - Start coding (explain mercurial repository, `set_env.m`, creating project structure)
  - Implement `model.shape.Lagrange`
3. Lecture: Section 2.2
  - Implement `model.shape.Shape` and `model.shape.QuadShape`
4. Lecture: Chapter 3
  - Theory of Numerical Integration
  - Implement `model.IntegrationPoint`
  - Explain `math.NumIntIO` and implement `math.NumInt`
5. Lecture: Chapter 4 especially section 4.1 and parts of section 4.2 (no coding)
6. Lecture: Chapter 5 starting with section 5.2
  - Explain the main classes of the model and their interactions
7. Lecture:
  - Implement `model.load.*` (see section 5.2.5)
  - Implement `model.dof.*` (see section 5.2.4)
  - Discuss the input handling (section 5.4)
8. Lecture: Section 5.5

## 1 Preface

- Implementing the base structure for the analysis
- Implement the Jacobian Matrix (see section 4.2.1)

9. Lecture: Section 4.2

- Implement `analyze.QuadImpl`, especially the method `calcElementStiffness()`
- Realize the assembling algorithm (section 4.2.4)
- Integration of the boundary conditions (section 4.3 and 4.4)

10. Lecture:

- Finalize the implementation
- Caculate some examples
- Do some post processing (section 5.6)

Preliminary