SpringerBriefs in Applied Sciences and Technology

Computational Mechanics

More information about this series at http://www.springer.com/series/8886

Michael Trapp · Andreas Öchsner

Computational Plasticity for Finite Elements

A Fortran-Based Introduction

Springer

Michael Trapp
TU Munich
Munich
Germany

Andreas Öchsner
Faculty of Mechanical Engineering
Esslingen University of Applied Sciences
Esslingen am Neckar
Germany

ISSN 2191-530X ISSN 2191-5318 (electronic)
SpringerBriefs in Applied Sciences and Technology
ISSN 2191-5342 ISSN 2191-5350 (electronic)
SpringerBriefs in Computational Mechanics
ISBN 978-3-319-77205-9 ISBN 978-3-319-77206-6 (eBook)
https://doi.org/10.1007/978-3-319-77206-6

Library of Congress Control Number: 2018933499

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

In this book, we want to demonstrate the use of the classic programming language FORTRAN for numerical computing in the context of the finite element method. For this purpose, we are focusing specifically on the subject of computational plasticity.

FORTRAN is a programming language, specifically designed for scientific and engineering applications. Since its initial release in the 1950s, many upgrade features have been added.

FORTRAN has asserted itself due to its practical use. Unlike earlier programs with machine or assembly code, FORTRAN uses higher language with abstract words and syntax, which automize computation instructions. This makes the code easy to read and write, and much less prone to bugs.

A decided advantage of FORTRAN is its flexibility. As it does not depend on a particular computing system, a transfer is very simple. In this context, it should be noted that all classical finite element packages are written in this language. Some even offer an interface to link one's own code to the commercial package. As a general-purpose tool, FORTRAN is a modern alternative for numerical computing and scientific programming. As opposed to many numerical software, FORTRAN is always transparent and easy to work with.

We look forward to receiving some comments and suggestions for the next edition of this textbook.

Gold Coast, Australia
January 2018

Michael Trapp
Andreas Öchsner
Griffith University

Acknowledgements

It is important to highlight the contribution of the students which helped to finalize the content of this book. Their questions, comments and struggle during different lectures, assignments and final examinations helped us to structure this book. Furthermore, we would like to express our sincere appreciation to the Springer Publisher, especially to Dr. Christoph Baumann, for giving us the opportunity to realize this book.

Contents

Symbols and Abbreviations

Latin Symbols (Capital Letters)

A	Area, cross-sectional area
C	Elasticity matrix
E	Young's modulus
E^{int}	Intermediate modulus
E^{pl}	Plastic modulus
E^{elpl}	Elasto-plastic modulus
$\tilde{E}$	(Approximated) tangent modulus
F	Yield condition
F_{tot}	(Applied (total)) force
F^{int}	Internal force
I	Identity matrix
J_1	First invariant
J_2	Second invariant
J_3	Third invariant
K	Tangent stiffness
$\tilde{K}$	Intermediate stiffness
K	(Global) tangent stiffness matrix
L	Length
L	Auxiliary matrix
$\tilde{L}$	Auxiliary matrix
Q	Plastic potential

Latin Symbols (Small Letters)

dim	Dimension
f	Yield criterion
h	Hardening function
k	Hardening function
k	Yield or flow stress

k^{init}	Initial flow stress
m	Vector of residuals
q	General hardening parameter
r	Function of flow direction
r	(Force) residual
r_σ	Stress residual
r_κ	Effective plastic strain residual
r_F	Yield condition residual
s	Deviatoric stress
t	Time
u	Displacement
u_{tot}	Total (prescribed) displacement
v	Solution vector, vector of unknowns
$\mathbf{1}$	Auxiliary vector $\{1, 1, 1, 0, 0, 0\}^T$
$\mathbf{2}$	Auxiliary vector $\{1, 1, 1, 2, 2, 2\}^T$

Greek Symbols (Capital Letters)

Δ	Incremental difference

Greek Symbols (Small Letters)

δ	Difference between iterations
κ	Effective plastic strain
ν	Poisson's ratio
$\Delta\lambda$	Effective plastic strain increment
σ	Stress
ε	(Total) Strain

Mathematical Symbols

$\times$	Multiplication sign (used where essential)
$\ldots^{\mathrm{T}}$	Transposed
$(\ldots)^{-1}$	Inverse
$\frac{\partial^n}{\partial \xi^n}$	*N*th partial derivative with respect to ξ
a	Vector, tensor or matrix
$\det = \lvert\ldots\rvert$	Determinant/absolute value
sgn	Signum function
$\mathbb{N}$	Natural numbers 1, 2, 3…

Indices, Superscripted

$\ldots^{\circ}$	Volumetric
$\ldots'$	Deviatoric

$\ldots^{el}$	Elastic
$\ldots^{elpl}$	Elasto-plastic
$\ldots^{pl}$	Plastic
$\ldots^{init}$	Initial, at the start of plastic regime
$\ldots^{j}$	Of iteration j
$\ldots^{m}$	Of element m
$\ldots^{i}$	Of node i
$\ldots^{t}$	Trial

Indices, Subscripted

$\ldots_{c}$	Compression
$\ldots_{eff}$	Effective
$\ldots_{eq}$	Equivalent, effective
$\ldots_{long}$	Longitudinal
$\ldots_{n}$	Of increment n
$\ldots_{t}$	Tension
$\ldots_{tvs}$	Transversal
$\ldots_{T}$	Tangent, tangential
$\ldots_{x}$	x-direction
$\ldots_{y}$	y-direction
$\ldots_{z}$	z-direction

Abbreviations

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
const	Constant
CPP	Closest point projection algorithm
FE	Finite element(s)
FEM	Finite element method
FIBE	Fully implicit backwards Euler method
lin	Linear
non-lin	Non-linear
NR	Newton–Raphson method
PR	Poisson's ratio ν
red	Reduced
SIBE	Semi-implicit backwards Euler method
tol	Tolerance
v.M.	von Mises