Explicit Stress Update Algorithm
For Finite Element Analysis of Sheet Metal Forming Process

Kevin Z. Ding
B.E. (Northeastern University, P. R. China)
M.E. (The Australian National University, Australia)

September 2007

A thesis submitted for the degree of Doctor of Philosophy of Engineering
of The Australian National University

Department of Engineering
Faculty of Engineering and Information Technology
College of Engineering and Computer Science
The Australian National University
To my parents, my wife and my kids for their love and support
Declaration

This thesis contains no material which has been previously accepted for the award of any other degree or diploma in any university, institute or college, and contains no material previously published or written by another person, except where due reference is made.

Canberra, September 2007.

[Signature]

Kevin Z. Ding

Department of Engineering
Faculty of Engineering and Information Technology
College of Engineering and Computer Science
The Australian National University
Canberra ACT 0200, AUSTRALIA.
Papers:

Journal Papers:


Conference Paper:

Abstract

Sheet metal forming is a technologically important process in manufacturing industries that allows economical production of parts with complex shapes from flat sheet stock. In automotive industry, a great deal of effort is consumed in making stamping tooling adjustments faster, speeding the die tryout process and optimizing production rates.

However shortening the process time it takes to develop tooling requires a better understanding on the mechanism of the forming process and thus physical experiments is frequently performed and tools are repeatedly modified in response to the experimental results. Finite element analysis (FEA) can be advantageously used to minimize die tryout and in addition provides insights needed to guide the determination of optimum process parameters to minimize the cost of production. It is now well recognized that computer simulations makes it possible to design and optimise the total process to a level which can’t be reached by traditional theoretical and experimental methods.

Currently, the accuracy and reliability of numerical simulations of sheet metal forming processes do not yet satisfy the industrial requirements. The most severe limitation of computer simulations today is the constitutive models that are insufficient to describe material and frictional behaviour. A number of areas of industrial relevance can not be covered with the models presently available in most commercial finite element software, such as ABAQUS, LS-DYNA, etc. Thus more and more researchers and engineers are developing their own constitutive laws with the aims to predict key outcomes of the forming process more accurately. The problem arises when implementing such complex constitutive models into existing FE codes as the users have to develop their own stress update schemes correspondingly.

During the past decades, implicit algorithms have drawn growing attentions for their excellent convergent performance in nonlinear analysis. However, as more and
more complicated constitutive models, taking into account non-associated flow rule, complex yield criteria and hardening laws, are being developed for simulating the sheet metal forming accurately, application of the existing implicit algorithms to sheet metal forming analysis is quite often cumbersome or even infeasible and the convergence is not always guaranteed.

In this thesis, we present our development of new integration algorithms for elastic-plastic analysis of sheet metal forming. The method is based on so-called substepping scheme, combined with the stress correction process. The procedure outlined controls the error of computed stress by choosing each substep size automatically according to a prescribed tolerance. The results indicate that the combination of this substepping scheme and the stress correction which is applied at the end of integration process can increase both accuracy and efficiency significantly for the complex sheet metal deep drawing process. Most importantly, the proposed algorithm is applicable to any constitutive models, including those with non-associated flow.

We will also describe the implementation in further detail and give some examples of results obtained from both numerical runs and experimental comparisons. The results are designed to highlight the performance of the algorithm developed.
Acknowledgements

My first, and most earnest, acknowledgment must go to my supervisor and chair of my supervisory panel, Professor Qing Hua Qin. He has been instrumental in ensuring my academic, professional, and moral wellbeing. In every sense, none of this work would have been possible without him.

I also owe a huge debt of gratitude to Professor Michael Cardew-Hall, Head of Department of Engineering, for reprovision of the financial support when I resumed my full-time studies two years ago. Thanks also go to Dr. Zbigniew Stachurski who were instrumental in the success of my studies.

Far too many people to mention individually have assisted in so many ways during my work at the ANU. They all have my sincere gratitude. In particular, I would like to thank Josephine Farmer and Rob Gresham for their helps in every aspects of my studies at DE.

I would like to thank the staff at the Australian National University Supercomputer Facility, especially Ben Ians and Judy Jenkinson for their technical advices on running ABAQUS user subroutines and using new parallel systems.
## Contents

Declaration i

Abstract iii

Acknowledgements v

Notation xvi

1 Introduction 1

1.1 The Research Problem ................................................. 1
1.2 Literature Review .......................................................... 4
1.3 Aims ................................................................................. 7
1.4 Organisation of the thesis ............................................. 7

2 Finite Element Analysis of Sheet Metal Forming Process 10

2.1 Finite Element Method ..................................................... 10
2.1.1 Brief history of Finite Element Method ....................... 10
2.1.2 General Procedures of Finite Element Method ............. 11
2.1.3 Applications of Nonlinear Finite Element Analysis ......... 12
2.1.4 Finite element analysis of sheet metal forming process ... 13

2.2 Explicit finite element programs ..................................... 15
2.2.1 Central difference method ......................................... 16
2.2.2 Implementation ......................................................... 19
2.2.3 Accuracy ................................................................. 21
2.2.4 Mass scaling, subcycling and dynamic relaxation .......... 21
3 The Mathematical Formulation of Finite Element Elastic-Plastic Analysis

3.1 Fundamentals of Elastic-Plastic Problem .................................................... 23
  3.1.1 Hypoelastic-plastic constitutive models ........................................... 24
  3.1.2 The Yield Criterion .......................................................................... 27
  3.1.3 Strain Hardening ................................................................................ 32

3.2 Numerical expression of constitutive relations ............................................. 33
  3.2.1 Von Mises yield criterion ................................................................. 34
  3.2.2 Hill's yield criterion .......................................................................... 35
  3.2.3 Hardening rule ................................................................................... 36

4 Stress Update Algorithms ........................................................................ 38
  4.1 Implicit algorithms ......................................................................................... 38
    4.1.1 Return mapping algorithms ............................................................. 39
    4.1.2 Fully implicit backward Euler scheme ........................................... 40
    4.1.3 Application to $J_2$ flow theory - radial return algorithm ............... 44
  4.2 Explicit integration schemes .......................................................................... 47
    4.2.1 Intersection with the yield surface ................................................. 49
    4.2.2 Correction of the yield surface drift ................................................. 51
    4.2.3 Forward Euler method ....................................................................... 53
    4.2.4 New Runge-Kutta-Dormand-Prince integration scheme ............... 55
    4.2.5 Modified Euler method .................................................................... 60

5 Numerical Experiments ........................................................................ 64
  5.1 Material Models ............................................................................................ 64
    5.1.1 Von Mises Yield Criterion ................................................................. 65
    5.1.2 Hill’s yield criteria ............................................................................. 69
  5.2 Parallel performance of new Runge-Kutta-Dormand-Prince scheme .......... 71
    5.2.1 Cylindrical shell cap .......................................................................... 73
    5.2.2 Cantilever beam model .................................................................... 75
  5.3 Summary ........................................................................................................ 79
    5.3.1 Comparison of performance of different integration schemes .......... 79
    5.3.2 Parallel analysis ................................................................................ 80
6 Development of user material subroutine

6.1 ABAQUS Finite Element Analysis Package

6.1.1 Brief introduction to ABAQUS software

6.2 Implementation of user subroutine

6.2.1 ABAQUS interface

6.2.2 Syntax for calling the user subroutine

6.2.3 Initialization of state variables

6.3 Special considerations for shell element

6.3.1 Update of sheet metal thickness

6.3.2 Derivation of transverse shear stiffness

7 Numerical experiment

7.1 Numerical experiment

7.1.1 Numerical model

7.1.2 Von Mises Yield Criteria

7.1.3 Hill's yield criteria

7.2 Summary

8 Numerical and experiment verification

8.1 Deep drawing experiment

8.1.1 Experimental setup

8.1.2 Material properties

8.1.3 Comparison

8.2 Numisheet'93 benchmark problem

8.2.1 Experimental setup

8.2.2 Material Properties

8.2.3 Blank preparation and lubrication

8.2.4 Comparison of experiment and numerical results

9 Conclusion and future research

9.1 Concluding Remarks

9.2 Future Work

Bibliography
A Runge-Kutta Methods

B A example of ABAQUS input file

C User Material Subroutine for ABAQUS/Explicit
# List of Figures

3-1 Geometrical representation of the Tresca and von Mises yield surfaces in principal stress ................................................................. 29
3-2 2D representation of Tresca and von Mises yield surfaces ................... 30
3-3 Mathematical models for representation of strain hardening behaviour . 33

4-1 Closest point projection scheme for associative plasticity: \( r_{n+1} \sim \frac{\partial f}{\sigma_{n+1}} \) 42
4-2 Radial return method for \( J_2 \) plasticity - shown at the converged state. 46
4-3 Schematic illustration of the initial yielding state. .......................... 49
4-4 Schematic illustration of the stress correction process. ..................... 52
4-5 Coefficients for Runge-Kutta-Dormand-Prince scheme ................. 56

5-1 A typical three dimensional cantilever beam. ............................... 65
5-2 Effective strain - Effective stress curves for BE, ME, MEC, RKDP, RKDPC (von Mises model) .................................................. 67
5-3 Effective strain - Effective stress curves for BE, ME, MEC, RKDP, RKDPC (Hill’s model) .................................................... 70
5-4 A cylindrical shell under uniform pressure................................. 73
5-5 A vertical strip-wise partitioning on 4 processors ....................... 75
5-6 A horizontal strip-wise partitioning on 4 processors ................... 75

6-1 Illustration of a shell element under tractions, moments and transverse shear forces ................................................................. 89

7-1 Schematic flow chart of the stress integration process. ............... 93
7-2 Schematic illustration of the tools and blank for the deep-cup drawing process. ................................................................. 93
7-3 Deformed blank at the punch stroke 36 mm. ..................................................... 94
7-4 Contour of the effective stress from 'ideal run' (von Mises model). ............. 96
7-5 Contour of the effective stress from FE with stress correction (von Mises model). ................................................................. 96
7-6 Contour of the effective stress from ME($10^{-4}$) without stress correction (von Mises model). ................................................................. 96
7-7 Contour of the effective stress from RKDP ($10^{-4}$) without stress correction (von Mises model). ................................................................. 96
7-8 Contour of the effective stress from ME ($10^{-2}$) without stress correction (von Mises model). ................................................................. 96
7-9 Contour of the effective stress from FE without stress correction (von Mises model). ................................................................. 96
7-10 Contour of the blank thickness from 'ideal' run (von Mises model). ....... 97
7-11 Contour of the blank thickness from FE with stress correction (von Mises model). ................................................................. 97
7-12 Contour of the blank thickness from ME($10^{-4}$) without stress correction (von Mises model). ................................................................. 97
7-13 Contour of the blank thickness from RKDP ($10^{-4}$) without stress correction (von Mises model). ................................................................. 97
7-14 Contour of the blank thickness from ME ($10^{-2}$) without stress correction (von Mises model). ................................................................. 97
7-15 Contour of the blank thickness from FE without stress correction (von Mises model). ................................................................. 97
7-16 Effective strain - Effective stress curves for BE, FE, FEC, ME, MEC (von Mises model). ................................................................. 98
7-17 Effective strain - Effective stress curves for BE, FE, FEC, RKDP, RKDPC (von Mises model). ................................................................. 98
7-18 Plot of thickness strain for BE, FE, FEC, ME, MEC (von Mises model). .... 99
7-19 Plot of thickness strain for BE, FE, FEC, RKDP, RKDPC (von Mises model). ................................................................. 99
7-20 Contour of the effective stress from 'ideal run' (Hill's model). ................. 103
7-21 Contour of the effective stress from FE with stress correction (Hill's model).103
7-22 Contour of the effective stress from ME(10^{-4}) with stress correction (Hill's model) .................................................. 103
7-23 Contour of the effective stress from RKDP (10^{-2}) with stress correction (Hill's model) .................................................. 103
7-24 Contour of the effective stress from ME (10^{-4}) without stress correction (Hill's model) .................................................. 103
7-25 Contour of the effective stress from FE without stress correction (Hill's model) .......................................................... 103

7-26 Contour of the blank thickness from 'ideal' run (Hill's model). .................. 104
7-27 Contour of the blank thickness from FE with stress correction (Hill's model). .......................................................... 104
7-28 Contour of the blank thickness from ME(10^{-4}) with stress correction (Hill's model). ............................................. 104
7-29 Contour of the blank thickness from RKDP (10^{-2}) with stress correction (Hill's model). ............................................. 104
7-30 Contour of the blank thickness from ME (10^{-4}) without stress correction (Hill's model). ............................................. 104
7-31 Contour of the blank thickness from FE without stress correction (Hill's model). .......................................................... 104

7-32 Effective plastic strain - Effective stress curves for BE, FE, FEC, ME, MEC (Hill's model). ............................................. 105
7-33 Effective plastic strain - Effective stress curves for BE, FE, FEC, RKDP, RKDPC (Hill's model). ............................................. 105
7-34 Plot of thickness strain for BE, FE, FEC, ME, MEC (Hill's model) .......... 106
7-35 Plot of thickness strain for BE, FE, FEC, RKDP, RKDPC (Hill's model)106

8-1 ENERPAC 30 TON deep drawing machine. ............................................ 110
8-2 GOM optical strain measuring system. .................................................... 110
8-3 Discretized FEA model for rectangular cup drawing. ............................. 111
8-4 Discretized FEA model for square cup drawing. ........................................ 111
8-5 Contour of blank thickness from user subroutine (Hill's model with RKDP,BE).112
8-6 Contour of blank thickness from ABAQUS (von Mises model) ................. 112
8-7 Contour of blank thickness from user subroutine (Hill's model with RKDP,BE).112
8-8 Contour of blank thickness from ABAQUS (von Mises model) ........................................ 112
8-9 Plot of thickness strain for square cup drawing. ....................................................... 113
8-10 Plot of thickness strain for rectangular cup drawing. .......................................... 113
8-11 Schematic illustration of tooling geometry for Numisheet’93 benchmark square cup drawing. ................................................................. 116
8-12 Discretized FEA model for Numisheet’93 benchmark square cup drawing. 117
8-13 Contour of blank thickness from user subroutine (Hill’s model with RKDP,BE). 117
8-14 Contour of blank thickness from ABAQUS (von Mises model) .......................... 117
8-15 Plot of thickness strain for Numisheet’93 square cup drawing. ........................ 118
List of Tables

2.1 Flowchart for explicit time integration ....................................................... 19

3.1 Elastic-plastic constitutive model (Cauchy stress formulation) ............... 28

4.1 Backward Euler return mapping algorithm ................................................. 45
4.2 Radial return algorithm ................................................................................ 48
4.3 Forward Euler scheme with subincrements ................................................. 54
4.4 Runge-Kutta-Dormand-Prince scheme with subincrements ...................... 61
4.5 Modified Euler scheme with subincrements .............................................. 63

5.1 Results of errors for BE, ME and RKDP with different tolerance (von Mises model) .................................................................................................. 68
5.2 Total substeps needed in the overall solution for ME and RKDP schemes with different tolerance (von Mises model). ................................................... 69
5.3 CPU time (seconds) spent on the overall solution (von Mises model)....... 69
5.4 Results of errors for BE, FE, ME and RKDP with different tolerance (Hill’s model) .................................................................................................. 71
5.5 Total substeps needed in the overall solution for ME and RKDP schemes with different tolerance (Hill’s Model) .................................................... 71
5.6 CPU time (seconds) spent on the overall solution (Hill’s model) ............ 72
5.7 Performance of different problem sizes on different number of processors for cap model ................................................................. 74
5.8 Performance of vertical partitioning on fixed number of processors ........ 76
5.9 Performance of vertical partitioning on different number of processors . 77
5.10 Performance of horizontal partitioning on fixed number of processors ... 78
5.11 Performance of horizontal partitioning on different number of processors 79
7.1 Results of errors for BE, FE, ME and RKDP with different tolerance (von Mises model) ......................................................... 100
7.2 Total substeps needed in the overall solution for ME and RKDP schemes with different tolerance (von Mises model) ............. 101
7.3 CPU time (seconds) spent on the overall solution (von Mises model) ................................................................. 101
7.4 Results of errors for BE, FE, ME and RKDP with different tolerance (Hill’s model) .......................................................... 107
7.5 Total substeps needed in the overall solution for ME and RKDP schemes with different tolerance (Hill’s Model) ................ 107
7.6 CPU time (seconds) spent on the overall solution (Hill’s model) .......... 107
8.1 Material properties for square and rectangular cup drawing ............. 111
8.2 Material properties used in Numisheet’93 square cup drawing .......... 116
Notation

SYMBOLS

\( \dot{f} \) For a field, the superposed dot denotes the material time derivative, i.e. \( \dot{f}(X,t) = \frac{\partial f(X,t)}{\partial t} \); for a function of time only, it is the ordinary time derivative, i.e. \( \dot{f}(t) = \frac{df(t)}{dt} \)

\( f_{,x} \) Derivative with respect to the variable \( x \); when the comma is followed by an index, such as \( i,j,k \), to \( s \), it is the derivative with respect to the corresponding spatial coordinate, i.e. \( f_{,i} = \frac{\partial f}{\partial x_i} \)

\( \cdot \) as in \( a \cdot b \) Contraction of inner indices; for vectors, \( a \cdot b \) is the scalar product \( a_i b_i \); if one or more of the variables are tensors of second or higher, the contraction is on the inner indices, i.e. \( A \cdot B \) represents \( A_{ij} B_{jk} \), \( A \cdot a \) represents \( A_{ij} a_j \)

\( : \) as in \( A : B \) Double contraction of inner indices: \( A : B \) is given by \( A_{ij} B_{ij} \); \( C : D \) is \( C_{ijkl} D_{kl} \); Note that if \( A \) or \( B \) is symmetric \( A : B = A_{ij} B_{ji} \)

\( \otimes \) As in \( a \otimes b \) indicates vector product; in indicial notation, \( a \otimes b \rightarrow a_i b_j \); in matrix notation, \( a \otimes b \rightarrow \{a\}\{b\}^T \)

VARIABLES

\( C^{damp} \) Damping matrix

\( C_{eI}, C_e \) Elastic moduli

\( D \) Rate-of-deformation tensor

\( D^e \) Elastic part of rate-of-deformation tensor

\( D^p \) Plastic part of rate-of-deformation tensor

\( E \) Youngs elastic modulus

\( F \) Yield function in elastic-plastic constitutive models

\( I^{dev} \) Fourth-order symmetric deviatoric tensor
\( J, J_i \)  Invariants of the deviatoric stress
\( M, M_{IJ} \)  Mass matrix
\( R \)  Relative error in substep scheme
\( R_{ij} \)  Yield stress ratios for Hill's yield function
\( S, S_i \)  Deviatoric stress
\( T, \Delta T \)  Dimensionless time and time step
\( TOL \)  Convergence tolerance
\( W_p \)  Total plastic work
\( Y \)  Material constant for defining yield surface
\( a \)  Nodal acceleration
\( d \)  Nodal displacements
\( f, f_I, f_{II} \)  Nodal forces
\( c_e \)  Current wave speed in element \( e \)
\( f^{int}, f^{int}_I \)  Internal nodal forces
\( f^{int}, f^{ext}_I \)  External nodal forces
\( h \)  Scalar function for strain hardening
\( k \)  Material constant for defining yield surface
\( l_e \)  Characteristic length of element \( e \)
\( n, n_i, n^0_i \)  Unit normal in current (deformed) and initial (reference, undeformed) configurations
\( r \)  Plastic flow direction
\( r_0, r_{45}, r_{90} \)  Anisotropy coefficients for Hill's yield function
\( t, \Delta t \)  Time and time step of explicit time integration
\( \Delta t \)  Critical time step for explicit time integration
\( v \)  Nodal velocity
\( \sigma, \sigma_{ij}, \{\sigma\} \)  Cauchy stress tensor
\( \Delta\sigma, \Delta\sigma_{ij} \)  Stress increment
\( \sigma^{\nabla J} \)  Cauchy stress with elastic response specified in terms of the Jaumann rate
\( \sigma^{trial} \)  Trial stress
\( \sigma^{dev} \)  Deviatoric stress
\( \sigma_Y^o \)  Uniaxial yield stress
\( \bar{\sigma} \) Effective plastic stress

\( \sigma_i \) Principal stress

\( \varepsilon, \varepsilon_{ij}, \{ \varepsilon \} \) Strain tensor

\( \varepsilon \) Strain vector

\( \Delta \varepsilon \) Strain increment

\( \Delta \varepsilon_e \) Elastic strain increment

\( \Delta \varepsilon_p \) Plastic strain increment

\( \dot{\varepsilon}_p \) Effective plastic strain

\( \dot{\lambda} \) Scalar plastic flow rate

\( \nu \) Poisson’s ratio

\( \kappa \) Hardening parameter

\( \psi \) Plastic flow potential

\( \mathcal{G} \) Fourth-order unit tensor

\( \alpha \) Reduction factor that accounts for the destabilizing effects of non-linearities

\( \omega_{\text{max}} \) Maximum frequency of the linearized system
Chapter 1

Introduction

1.1 The Research Problem

The automotive industry is now facing world-wide serious challenges: fierce market competition and strict governmental regulations on environment protection. The strategies of the automakers to meet these challenges is sometimes called the 3R Strategy: Reduction in time-to-market, reduction in development costs to gain competitiveness, and reduction in the vehicle weight to improve fuel efficiency. The solutions to achieve this triple goal are essentially based on the implementation of CAD/CAE/CAM technologies in product development and process design. A very significant component of this endeavour is focused on the reduction of the tooling costs and the lead-time related to the stamping of autobody panels, even under increasing technological difficulties such as the use of aluminum alloys and high-strength steels, and requirements for higher geometrical accuracy of stamped parts.

The Finite Element Method (FEM) is now firmly accepted as a powerful general technique for predicting a variety of forming problems encountered in sheet metal forming process. Finite element analysis can be advantageously used to minimize die tryout and in addition to provide the insights needed to guide the determination of optimum process parameters to minimize the cost of production. With advances in finite element analyses and availability of increased computing power, sheet metal forming processes can be simulated more rapidly and with diminishing computational and human cost, which greatly shortens lead times and improves tool performance. However, because finite element analysis of realistic sheet metal forming operations always involves complex
3-D geometries, multiple forming steps, complex contact situations etc, the accuracy and reliability of numerical simulations of sheet metal forming processes do not yet satisfy most industrial requirements. The major reason is that the accuracy of simulation depends mainly on constitutive laws used and their material parameters identification and all the constitutive models provided with commercial FEA software are of general purpose.

For instance, the final shape of a product is strongly linked to both plastic material flow and springback phenomenon. Plastic anisotropy explains the undulated rims called ears in [1], which appear in a cup produced by cylindrical tools applied on a circular blank. The classical isotropic von Mises yield criterion predicts no ears at all. The simple quadratic anisotropic Hill yield criterion often simulates an inaccurate four-earing profile. More complex models relying on crystal plasticity and a homogenization approach provide results that are closer to the experimental observations. It is noted that, during past years, a number of numerical investigations was conducted on simulation of forming processes employing either the anisotropic (texture-based) strain-rate potential [2, 3, 4] or crystal plasticity [5, 6, 7, 8], in addition to the conventional macroscopic plasticity models. The former takes explicit account of initial texture by adopting an assumed analytical yield function, the coefficients of which are determined by fitting them to the polycrystalline yield surface obtained using crystal plasticity, and it has an obvious advantage over its conventional counterpart in its capability of describing plastic anisotropy. But further improvement on the model is required to be made so that it can describe texture evolution in the forming simulation.

More and more complex material models are being developed with the aim to predict accurately the final geometry of the sheet blank and the distribution of strain and stress and control various forming defects, such as thinning, wrinkling and springback, etc. However this introduces a somewhat new problem in terms of using these new material models because it is extremely time-consuming and expensive for the researchers in the world to develop their own FEA code, including pre/post processor, the stable explicit solver and suitable contact searching algorithm, to conduct the simulations. In this case the most effective way is to build their own material models into the existing finite element codes. However the use of such complex models raises a number of questions in relation to implementing an appropriate stress update algorithm.
1.1. THE RESEARCH PROBLEM

Most commonly used stress update schemes for integration of elastic-plastic constitutive laws fall within the categories of forward (explicit) and backward (implicit) algorithms. During the past decades, implicit algorithms have drawn growing attentions for their excellent convergent performance in nonlinear analysis. Return mapping schemes dominate integration algorithms used in the most commercial Finite Element software packages such as ABAQUS[9], ANSYS[10], LS-DYNA[11], etc. Obviously these implicit stress update algorithms work well with the relatively simple constitutive models. However, as more and more complicated constitutive models, taking into account non-associated flow rule, complex yield criteria and hardening laws, are being developed for simulating the sheet metal forming process, application of the existing implicit algorithms to sheet metal forming analysis is quite often cumbersome or even infeasible due to the second order derivative of plastic potential being needed and the convergence is not always guaranteed.

Compared with implicit algorithms, the explicit algorithm has the advantage of being more straightforward to implement. However, this technique requires that load increments be kept enough small so that the stresses computed at the end of integration procedure do not deviate too far from the yield surface. For the applications with relative large load increments, the computed stresses may not satisfy the yield criterion after integration process. Therefore, a correction of stresses has been used frequently. However, due to the lack of measures for controlling the error in the integration process, the explicit algorithm has following disadvantages:

1. If the correction-step is applied after each substep, the computational time will increase drastically. However, if it is done at the end of integration, it does not significantly affect the accuracy[12].

2. Since the number of substeps is usually determined by an empirical rule which is formulated by trial and error, the inappropriate choice of the number of the substeps usually leads to the lose of either accuracy or efficiency.

Therefore, it would be desirable to have an explicit integration scheme which can not only maintain as same accuracy as the implicit does but is easy to implement for simulating complex sheet metal forming process. The only possible solution is to use adaptive substep size to control the errors in the integration process.
1.2 Literature Review

In order to use FEA to simulate sheet metal forming process accurately, it is necessary to employ a constitutive models which can give a good description of the material behaviour. The most commonly-used material models are rigid-plastic or, rigid-viscoplastic model\[13, 14, 15\] and, elastic-plastic, or elastic-viscoplastic model\[16, 17, 18\]. Formulation and implementation of rigid-plastic model is rather simple, and fast calculation is achieved. However, due to the lack of elastic region in material model, it is not possible to calculate the springback after forming. So elastic-plastic model is the most widely used material model for simulating the sheet metal forming process and will be employed in this research.

When carrying out the elastic-plastic analysis of sheet metal forming process, it is well-known that the accuracy with which the constitutive models are integrated has a direct impact on the overall accuracy of the analysis. An accurate algorithm for integrating the stress-strain relations is therefore a key factor to make the analysis on extremely complex sheet metal forming process successful. The best algorithm, the one to be favored, is therefore the one which combines computational efficiency with accuracy.

The first implicit stress-point algorithm to be developed was the radial return algorithm proposed by Wilkins\[19\] for the elastic-perfectly plastic von Mises model. The algorithm was subsequently extended by Krieg and Key\[20\] to accommodate isotropic and kinematic hardening laws. The algorithms are analyzed in Krieg and Krieg\[21\]; Schreyer et al.\[22\]; Yoder and Wirley\[23\]; and Ortiz and Popov\[24\]. However, all these algorithms tend to be quite expensive and are not error-free. They are analyzed in Loret and Prevost\[25\]. Implicit integration algorithms for more complex models have also been developed, typically on a case-by-case basis (for example, Sandler and Rubin\[26\] for the cap model). However, no general framework for developing consistent, accurate and stable algorithms was available until recently, when the concept of an elastic predictor with a plastic return mapping was developed (Ortiz and Popov\[24\]).

The main component in return mapping algorithms is the enforcement of the plastic consistency condition defined by the yield function. Furthermore, the determination of the initial yielding state, i.e. the intersection with the yield surface, becomes unnecessary if the stress point changes from an elastic state to a plastic state. A common
feature of many of the techniques proposed to date is the use of an operator split strategy in the solution of this problem. Briefly, the final stresses and updated internal variables are obtained by imposing these conditions after a trial state has detected the activation of the plastic evolution equations. Common to many approaches is the use of an elastic trial state in this process, leading then to a plastic corrector step approximating the plastic evolution equations. This strategy involves an implicit approximation of the governing equations, leading to a nonlinear system of algebraic equations in the stresses and updated internal variables. The detailed introduction of the popular return mapping schemes can be found in [27, 28].

An iterative algorithm needs to be applied for the solution of the resulting nonlinear system of algebraic equations when return mapping method is used. A typical choice is the use of a Newton scheme. The asymptotic quadratic rate of convergence of this scheme makes this strategy very attractive. However, the well-known limited local convergence properties of Newton schemes, in the sense that actual convergence can only be assured for initial estimates close to the final solution, makes this approach difficult to motivate when considering the complex constitutive models used in many practical engineering applications of interest. These difficulties have been discussed in [29, 30]. Some researchers, de Souza Neto et al.[29], Bicanic and Pearce[31], Han and Reddy[32] have proposed several strategies to avoid these difficulties by considering alternative trial predictors initializing the iterative process. However, these alternative definitions are specific to particular models and thus difficult to generalize to other more complex yield surfaces or hardening laws.

Compared with implicit algorithms, the explicit algorithm has the advantage of being more straightforward to implement. The best-known explicit method for integrating elastic-plastic stress-strain relations is forward Euler scheme in which the strain increment that constitutes an elastic-plastic response is usually divided into a number of substeps. This subincrementation strategy with successive radial stress corrections was first proposed by Nayak and Zienkiewicz[33]. Other somewhat arbitrary stress corrections have also been attempted in Chen[34] and Mattsson[35] to correct for the inherent stress drift away from the yield surface. But the most comprehensive introduction of the stress correction procedures can be found in [36]. Since the number of substeps is usually determined by an empirical rule which is formulated by trial and
error, the inappropriate choice of the number of the substeps usually lead to the loss of either accuracy or efficiency.

A number of explicit integration algorithms therefore were developed with the aim to control the error in integration process. For example, Polat and Dokainish[37] used an automatic subincrementation scheme for accurate integration of elastic-plastic constitutive relations. Wissmann and Hauck[12] introduced an algorithm into integration process by using Richardson extrapolation to select a number of fixed size substeps. Pezeshk and Camp[38] indicated that an integration method based on a Modified Trapezoidal rule Method may be simple and effective. It is especially worthy noting that two explicit schemes with adaptive substepping developed by Sloan([39, 40]), the modified Euler scheme and the Runge-Kutta-England scheme, in which difference between the solutions obtained via two integration procedures with truncation errors of different order is used to determine the substep size. This idea has been further adopted and extended in a series of papers by Luccioni et al[41], Tamagnini et al[42], Jakobsen and Lade[43] and Fellin and Ostermann[44]. All the constitutive models employed in above researches are extremely complicated, meaning that it is extremely difficult or infeasible to use the popular implicit schemes in this case.

It is interesting to note that very few studies have been conducted in terms of comparing the substepping schemes with implicit schemes. The work of Tamagnini et al[42] and Potts and Ganendra[45] made some progress in this field. The numerical results in both works above indicated that the substepping schemes are more robust and efficient than the implicit schemes. However, it has been found that most of the existing work in this area focused on the plasticity in geomechanics. To our knowledge, the plasticity that occurs in geo-materials such as clay is very small due to their limitation to tolerate the plastic deformation. Also, as pointed out in [42], the use of "simple" numerical models, for example in [45], and reference to a given boundary value problem, rather than at the integration point level, made such comparisons not applicable for large plastic deformation problems. Gens and Potts[46] studied these two substepping schemes and suggested that, the deviation from the yield surface is directly related to the level of error in computed stresses and it is practically independent of the integration scheme adopted. Crisfield[47] also strongly advocates some form of stress correction when the model involves the hardening, as the drift is found to be generally
more pronounced. Since such discrepancies are usually cumulative, it is important to ensure that the stresses are corrected back to the yield surface at any time. As will be shown in Chapter 6 and 7, the substepping schemes with the error control are not satisfactory without correcting yield surface drift. However, they are superior to the implicit schemes when the convergence tolerance is adequate small and some form of the stress correction is applied at the end of step.

1.3 Aims

The major aim of this research project is to develop an effective integration scheme whose accuracy is comparable with the implicit scheme, but is easier to implement for complex constitutive models.

With this in mind, we will develop a substepping scheme for elastic-plastic stress integration process. The resulting algorithm can be applicable to a general type of constitutive law and controls the error in the integration process by adjusting the size of each substep automatically in accordance with the behaviour of the constitutive law. With this measure of error control, combined with the stress correction procedure, the new integration scheme will be capable to achieve the same level of the accuracy as the implicit scheme does, but is extremely easy to be employed in any constitutive model. The resulting algorithms will be built into the ABAQUS user material subroutine (VU-MAT) for analysing a typical 3D sheet metal deep drawing process.

1.4 Organisation of the thesis

This thesis chronicles our experience with non-linear elastic-plastic finite element simulation of sheet metal forming process over the past four years. An outline of this thesis is as follows.

Chapter 2 describes a general introduction of finite element method and solution procedures for elastic-plastic analysis of sheet metal forming process. A brief review of the finite element method and its application in sheet metal forming industry are presented. Both implicit and explicit time integration methods for this nonlinear problems are described but we focus on the description of explicit time integration which will be used in this research.
Chapter 3 is devoted to the basic theoretical formulations of the constitutive model for elastic-plastic problems are presented. We focus on the constitutive equations of hypoelastic-plasticity which often plays an important role in large deformation elastic-plastic constitutive relations. Other continuum mechanics concepts, i.e. the yield criterion, work or strain hardening, etc., are also discussed. To use these relations in ABAQUS user material subroutine, the numerical form of these relations are finally described.

Chapter 4 deals with the numerical algorithm for integrating the constitutive equations which is called constitutive integration algorithm or a stress update algorithm. In this Chapter, constitutive integration algorithms are presented for rate-independent plasticity. Both implicit and explicit stress update schemes are discussed in details. Finally a new explicit stress update algorithm for large deformation elastic-plastic materials is presented. The scheme is developed for use in finite element plasticity calculation and solve for the stress increments, assuming that the strain increments are known. To implement the new stress update algorithm, other techniques, such as determination of initial intersection with the yield surface, stress correction methods, etc., are also discussed. All the implicit and explicit stress update schemes mentioned will be implemented into an user material subroutine for analyzing a typical deep drawing process.

In Chapter 5, we use all the stress update algorithms introduced in Chapter 4 for a non-linear elastic-plastic analysis before building them into ABAQUS user subroutine VUMAT and conducting the analysis of sheet metal forming Process. The performances of all the algorithms are compared using more complex hardening and constitutive models. Some issues about the parallel analysis of such non-linear problems using new integration scheme are also addressed with the aim to demonstrate its performance in the parallel analysis of nonlinear engineering problems.

In Chapter 6, we present the development of the user material subroutine for ABAQUS/Explicit. A detailed discussion of the developing process will be presented with the aim for other user to build their constitutive model with this subroutine. Some important issues related to using shell elements in user subroutine, such as thickness update, computation of transverse shear stiffness, are also covered.

The numerical experiment of sheet metal forming is carried out in Chapter 7, us-
ing all the implicit and explicit stress update schemes introduced previously. Both the accuracy and efficiency of the proposed integration schemes are discussed through comparing the results computed by different integration schemes with the "ideal" run.

Chapter 8 describes the experiment verification of the proposed integration algorithms. To examine the performance of the integration schemes we carry out a real deep drawing experiment. Given that sheet metals exhibit a highly anisotropic material behaviour by cold rolling, the numerical model is modified to take the anisotropy into account by using the Hill's yield criterion. The deformation of the sheet metal is measured by the optical measuring equipment and will be compared with the numerical results.

Chapter 9 summaries the main conclusions of the study and provides some recommendations for the further research in this area.
Chapter 2

Finite Element Analysis of Sheet Metal Forming Process

This chapter describes a general introduction of finite element method and solution procedures for elastic-plastic analysis of sheet metal forming process. A brief review of the finite element method and its application in sheet metal forming industry are presented. Both implicit and explicit time integration methods for this nonlinear problems are described but we focus on the description of explicit time integration which will be used in this research. Some techniques for analyzing the sheet metal forming in ABAQUS/Explicit, such as mass scaling, etc. are discussed as well.

2.1 Finite Element Method

The finite element method is a numerical analysis technique for obtaining approximate solution to a wide variety of engineering problems. Although originally developed to study the stresses in complex airframe structures, it has since been extended and applied to the broad field of continuum mechanics. Because of its diversity and flexibility as an analysis tool, it is receiving much attention in engineering schools and in industry.

2.1.1 Brief history of Finite Element Method

Finite Element Analysis (FEA) was first developed in 1943 by R. Courant, who utilized the Ritz method of numerical analysis and minimization of variational calculus to obtain approximate solutions to vibration systems. Shortly thereafter, M. J. Turner, et al
published a paper which centered on the "stiffness and deflection of complex structures". This paper established a broader definition of numerical analysis.

During its early development for stress analysis problems FEA was limited to expensive mainframe computers generally owned by the aeronautics, automotive, defense, and nuclear industries and relied heavily on a physical interpretation in which the structure was assumed to be composed of elements physically connected only at a number of discrete nodal points. Later the application of the method to structural mechanics problems was developed through the use of the principle of virtual work and energy methods. The method was then generalised and its wider mathematical roots were recognised. More recently, finite element solutions have been developed which are based on the well known, classical techniques known as "weighted residual methods". Since the 1970's the rapid growth in engineering usage of computer technology has a significant effect upon the acceptance of the finite element method. With rapid decline in the cost of computers and the phenomenal increase in computing power, FEA has been developed to an incredible precision and could be applied to any engineering problem for which a variational functional existed. In fact the finite element method is now firmly established as an engineering tool of wide applicability. One of the principal advantage of the finite element method is the unifying approach it offers to the solution of diverse engineering problems.

2.1.2 General Procedures of Finite Element Method

Regardless of the approach used to different applications, the solution of finite element method always follows an orderly step-by-step process. To summarize in general terms how the finite element method works we will succinctly list these steps.

1. **Discretize the continuum.** The first step is to divide the continuum or solution region into elements. A variety of element shapes is available for different analysis.

2. **Select interpolation functions.** The next step is to assign nodes to each element and then choose the type of interpolation function to represent the variation of the field variable over the element.

3. **Find the element properties.** Once the finite element model has been established, we are ready to determine the matrix equations expressing the properties
2.1. FINITE ELEMENT METHOD

of the individual element.

4. Assemble the element properties to obtain the system equations. To find the properties of the overall system modelled by the network of elements we must "assemble" all the element properties.

5. Solve the system equations. The assembly process of the preceding step gives a set of simultaneous equations that we can solve to obtain the unknown nodal values of the field variable. Technique to solve this equations system may generally be classified into direct and iterative methods.

6. Calculate the element characteristics. Normally we use the solution of the system equations to calculate other important parameters, such as strains, stresses, etc.

2.1.3 Applications of Nonlinear Finite Element Analysis

The finite element method is today one of the major tools of CAE (Computer Aided Engineering). The growth of computer power and the availability of software allows engineers to solve more and more complex linear and nonlinear problems rapidly and with diminishing computational and human cost. More precisely the availability of solid based CAD geometry, automatic meshing, associativity between geometry and finite elements, the development of efficient solvers for system of linear equations, etc. has drastically reduced the time needed to solve complex problems.

Nonlinear finite element analysis is an essential component of computer-aided design. In the past decades, the development of improved element characteristics and more efficient nonlinear solution algorithms and the experience gained in its application to engineering problems have ensured that nonlinear finite element analyses can now be performed with some confidence. Hence barriers to the common use of nonlinear finite element techniques are being rapidly removed and the process is already economically acceptable for selected industrial applications. Testing of prototypes is increasingly being replaced by simulation with nonlinear finite element methods because this provides a more rapid and less expensive way to evaluate design concepts and design details. In many fields of manufacturing, simulation is speeding the design process by allowing simulation of processes such as sheet metal forming, etc.
2.1. FINITE ELEMENT METHOD

However, nonlinear analyses are often sensitive to many factors that can make a single simulation quite misleading. Some of them which are of central importance in nonlinear analysis, include the following:

1. The selection of appropriate methods for the problem at hand

2. The selection of a suitable mesh description and kinematic and kinetic descriptions for a given problem

3. The examination of stability of the solution and the solution procedure

4. An awareness of the smoothness of the response of the model and its implication on the quality and difficulty of the solution

5. The role of major assumption and the likely sources of error

Thus the informed use of nonlinear software in both industry and research requires considerable understanding of nonlinear finite element methods. For example, while the explicit method is probably best suited for simulating sheet metal forming process, in the springback simulation implicit methods are more suitable. Without an understanding of the implication and meaning of the choices and difficulties mentioned above, an analyst is at a severe disadvantage.

2.1.4 Finite element analysis of sheet metal forming process

The aim of most sheet metal forming research is to minimize the time and cost for process development and production while minimizing scrap and optimizing the quality of the parts produced. Although a number of analyzing methods have been developed in the past and have been useful in qualitatively predicting forming loads, overall geometry changes of the deformed blank and material flow and in determining approximate optimum process conditions, a more accurate determination of the effects of various process parameters on the sheet metal forming process has became possible only recently, when the finite element method was developed for these analyses.

Rapid developments in computer hardware make the finite element analysis of complex deformation responses increasingly applicable. However, the accuracy and reliability of numerical simulations of sheet metal forming processes do not yet satisfy the industrial requirements. This is partly due to
2.1. FINITE ELEMENT METHOD

- During the last decade, there has been a tremendous development of sheet materials and sheet forming technology. A large number of new sheet qualities, e.g. HSLA-steel, pre-coated steel and high strength aluminum, have come into the market place. These new sheet materials have other properties, e.g. higher strength and more ductility, than conventional sheet steel, and have therefore to be worked differently. The existing constitutive models are not accurate enough to describe the behaviour of these new sheet steels in different forming processes.

- The development of new sheet metal forming processes, tooling etc. has up till now to a large extent been based on experience, rules of thumb and trial-error experiments without or with only little use of scientifically based engineering methods. There is therefore great need for the development of both theoretical and experimental engineering methods which enable the problems to be tackled effectively in finite element analysis.

- The demands required from the sheet metal processes are increasing both with regard to the tolerance requirements of the finished part and with regard to the complexity. To meet these requirements, a detailed knowledge about the material properties, the friction conditions and the forming process is needed.

In industry, the general methodology for using FEA to study various sheet metal forming techniques includes the following steps:

1. **Modeling or importing finite element models using a pre-processor**

   The pre-processors should be used to generate the geometry. The surface models can also either generated from scratch or imported in IGES format from CAD packages like Pro/Engineer and HyperMesh. The geometries are then meshed to generate the finite element models for conduction of analysis.

2. **Analysis using a processor**

   Normally an FEA solver, such as ABAQUS or LS-DYNA, etc., is used as the finite element analysis processor for the manufacturing simulations. The software package should provide a wide variety of options to define various input parameters such as those required for contacts, material properties, etc. Sheet metal forming analyses are non-linear, large-deformation, large-displacement problems,
it is necessary to solve the equilibrium equations at a number of time steps during the forming process. Currently the explicit solver is widely employed because the direct integration technique is capable of providing fast and accurate solutions for the dynamic equations of equilibrium that describe the behaviour of the sheet in sheet metal forming problems. The information about the dynamic state of the bodies and the applied forces at the first time is used to calculate the nodal deflections at the second time. Information at the second time is used to calculate the state at the third time. This process, called direct time integration, is continued until the end of the analysis. This process, though time consuming can result in accurate solutions.

3. Post-processing the results to study the forming process

Post-processing of the results was carried out using the post-processor or other software with the results processing capabilities, such as HYPERMESH. The main features of the results that are of interest include,

- Distributions of stress, strain and sheet thickness within the deformed sheet and the material flow during the forming process.
- Ratio of kinetic energy to internal energy of the sheet. The ratio of kinetic energy to the internal energy of the sheet is used to ensure that inertial effects in the problem due to velocity and/or mass scaling are negligible
- Other results help understand the response of the sheet in order to make modifications to improve the forming processes.

2.2 Explicit finite element programs

The conventional finite element codes are based on implicit time integration. This involves repeated solutions of large systems of equations. Furthermore, equilibrium must be fulfilled after each incremental step. As a result, implicit codes are computational time and memory consuming. Hence, a new class of finite element codes based on explicit time integration was developed, resulting in a drastic decrease of computational time. In an explicit code no system of equations needs to be solved and static equilibrium is not checked after each incremental step, as the algorithm assumes an
2.2. EXPLICIT FINITE ELEMENT PROGRAMS

inertia dominated process. The explicit procedure is conditionally stable with a critical time step, which is proportional to the smallest element in the mesh. To employ ABAQUS/Explicit for analysis of deep drawing process in the following chapters, a brief discussion of this scheme is given as follows.

2.2.1 Central difference method

The central difference method is among the most popular of the explicit methods in computational mechanics. It is developed from central difference formula for the velocity and acceleration. For the purpose of developing this and other time integrators we will use the following notation. Let the time of the simulation $0 \leq t \leq t_E$ be subdivided into time steps $\Delta t^n$, $n = 1$ to $n_{TS}$ where $n_{TS}$ is the number of time steps and $t_E$ is the end of the simulation. The superscript indicates the time step: $t^n$ and $d^n$ are the time and displacement, respectively, at time step $n$.

We consider here an algorithm with a variable time step. This is necessary in most practical calculations since the stable time step changes as the mesh deforms and the wave speed changes due to the stress. For this purpose, we define the time increments by

$$
\Delta t^{n+1/2} = t^{n+1} - t^n \\
t^{n+1/2} = \frac{1}{2} \left(t^{n+1} + t^n\right) \\
\Delta t^n = t^{n+1/2} - t^{n-1/2}
$$

(2.1)

The central difference formula for the velocity is

$$
d^{n+1/2} = v^{n+1/2} = \frac{d^{n+1} - d^n}{t^{n+1} - t^n} = \frac{1}{\Delta t^{n+1/2}} \left(d^{n+1} - d^n\right)
$$

(2.2)

where the definition of $\Delta t^{n+1/2}$ from (2.2) has been used in the last step. This difference formula can be converted to an integration formula by rearranging the terms as follows:

$$d^{n+1} = d^n + \Delta t^{n+1/2} v^{n+1/2}
$$

(2.3)

The acceleration and the corresponding integration formula are

$$
\ddot{a}^n = a^n = \begin{pmatrix} v^{n+1/2} - v^{n-1/2} \\ \frac{t^{n+1/2} - t^{n-1/2}}{2} \end{pmatrix} \\
v^{n+1/2} = v^{n-1/2} + \Delta t^n a^n
$$

(2.4)
2.2. EXPLICIT FINITE ELEMENT PROGRAMS

As can be seen from the above, the velocities are defined at the midpoint of the time intervals, which are called half-steps or midpoint steps. By substituting (2.2) and its counterpart for the previous time step into (2.4), the acceleration can be expressed directly in terms of the displacements:

\[ \ddot{d}^n \equiv a^n = \frac{\Delta t^{n-1/2}}{\Delta t^{n+1/2}} \left( \begin{array}{c} A^n - d^{n-1} \\ A^n \end{array} \right) \]

(2.5)

For the case of equal time steps the above reduces to

\[ \ddot{d}^n \equiv a^n = \frac{(d^{n+1} - 2d^n + d^{n-1})}{(\Delta t^n)^2} \]

(2.6)

This is the well-known difference formula for the second derivative of a function.

We now consider the time integration of the equations of motion:

\[ Ma + f^{int} = f^{ext} \]

(2.7)

Where \( M \) is the global mass matrix. \( f^{int} \) and \( f^{ext} \) are internal and external nodal forces, respectively. At time step \( n \) this equation becomes

\[ Ma^n = f^n = f^{ext}(d^n, t^n) - f^{int}(d^n, t^n) \]

(2.8)

subject to

\[ g_I(d^n) = 0, \quad I = 1 \text{ to } n_c \]

(2.9)

The Equation (2.8) are ordinary equations of second order in time. It is often called semidiscrete, since it has been discretized in space but not in time. Equation (2.9) is a generalized representation of \( n_c \) displacement boundary condition and the other constraints on the model. These constraints are linear or nonlinear algebraic functions of nodal displacements. If the constraint involves integral or differential relationships, it can be converted to the above form by using difference equations or a numerical approximation of the integral. The mass matrix is constant in this case.

The internal and external nodal forces are functions of nodal displacements and the time. The external loads are usually prescribed as functions of time, they may also be functions of the nodal displacements because they may depend on the configuration of the structure, as when pressure forces are applied to surfaces which undergo large deformations. The dependence of the internal nodal forces on displacements is obvious: the nodal internal forces depend on the stresses, which depend on the strain and strain
rates by the constitutive equations, which in turn depend on the displacements and their derivatives. The internal nodal forces can also depend directly on time, e.g. when the temperature is prescribed as a function of time, then the stresses and hence the internal nodal forces are directly functions of time.

The equations for updating the nodal velocities and displacements are obtained as follows. Substituting (2.8) into (2.4) gives

\[ v^{n+1/2} = v^{n-1/2} + \Delta t^n M^{-1} f^n \]  

(2.10)

It should be noted that, at any time step \( n \), the displacements \( d^n \) are known. The nodal forces \( f^n \) can be determined by sequentially evaluating the strain-displacement equations, the constitutive equation expressed in terms of \( D^{n-1/2} \) and the nodal external forces. Thus the entire right-hand side of (2.10) can be evaluated, and (2.10) can be used to obtain \( v^{n+1/2} \). The displacements \( d^{n+1} \) can then be determined by (2.3).

The update of the nodal velocities and nodal displacements can be accomplished without solving any equations provided that the mass matrix \( M \) is diagonal. This is the salient characteristic of an explicit method: \textit{in an explicit method, the time integration of the discrete momentum equations does not require the solution of any equations}. The avoidance of solution of equations of course depends critically on the use of a diagonal mass matrix.

In numerical analysis, integration methods are classified according to the structure of the time difference equation. The difference equations for first and second derivatives, respectively, can be written in following general expressions

\[ \sum_{k=0}^{n_S} \left( \alpha_k d^{n_S-k} - \Delta t \beta_k d^{n_S-k} \right) = 0 \]

\[ \sum_{k=0}^{n_S} \left( \bar{\alpha}_k d^{n_S-k} - \Delta t^2 \bar{\beta}_k d^{n_S-k} \right) = 0 \]  

(2.11)

where \( n_S \) is the number of steps in the difference equation. The difference formula for the first or second derivatives are called explicit if \( \beta_0 = 0 \) or \( \bar{\beta}_0 = 0 \), respectively. Thus a difference formula is called explicit if the equation for the function at time step \( n_S \) involves only the derivatives at previous time step. In the central difference formula for the second derivative, (2.6), \( \beta_0 = 0, \bar{\beta}_1 = 1, \bar{\beta}_2 = 0 \), so it is explicit. Difference equations which are explicit according to this classification generally lead to solution schemes which require no solution of equations.
2.2. EXPLICIT FINITE ELEMENT PROGRAMS

Table 2.1: Flowchart for explicit time integration

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Initial condition and initialization: set ( v^0, \sigma^0 ), and initial values of other material state variables; ( d^0 = 0, n = 1, t = 0 ); compute ( M )</td>
</tr>
<tr>
<td>2.</td>
<td>Calculate the nodal forces</td>
</tr>
<tr>
<td>3.</td>
<td>Compute accelerations ( a^n = M^{-1}(f^n - C^{damp}v^{n-1/2}) )</td>
</tr>
<tr>
<td>4.</td>
<td>Time update: ( t^{n+1} = t^n + \Delta t^{n+1/2}, T^{n+1/2} = \frac{1}{2}(t^n + t^{n+1}) )</td>
</tr>
<tr>
<td>5.</td>
<td>First partial update nodal velocities: ( v^{n+1/2} = v^n + (t^{n+1/2} - t^n)a^n )</td>
</tr>
<tr>
<td>6.</td>
<td>Enforce velocity boundary conditions: if nodal ( v^i ) on ( \Gamma_v ): ( v^i_{n+1/2} = \tilde{v}(x_I, t^{n+1/2}) )</td>
</tr>
<tr>
<td>7.</td>
<td>Update nodal displacements: ( d^{n+1} = d^n + \Delta t^{n+1/2}v^{n+1/2} )</td>
</tr>
<tr>
<td>8.</td>
<td>Calculate the nodal forces</td>
</tr>
<tr>
<td>9.</td>
<td>Compute ( a^{n+1} )</td>
</tr>
<tr>
<td>10.</td>
<td>Second partial update nodal velocities: ( v^{n+1} = v^{n+1/2} + (t^{n+1} - t^{n+1/2})a^{n+1} )</td>
</tr>
<tr>
<td>11.</td>
<td>Check energy balance at time step ( n + 1 )</td>
</tr>
<tr>
<td>12.</td>
<td>Update counter: ( n \leftarrow n + 1 )</td>
</tr>
<tr>
<td>13.</td>
<td>Output; if simulation not complete, go to 4.</td>
</tr>
</tbody>
</table>

Procedure for calculating nodal forces

1. Initialization: \( f^n = 0, \Delta t = \infty \) |
2. Calculate global external nodal forces \( f^n_{ext} \) |
3. Loop over elements \( e \)
   - i. GATHER element nodal displacements and velocities
   - ii. \( f^n_{int,e} = 0 \) |
   - iii. Loop over quadrature points \( \xi_Q \):
     1. if \( n = 0 \), go to 5 |
     2. compute measure of deformation: \( D^{n-1/2}(\xi_Q), F^n(\xi_Q), E^n(\xi_Q) \) |
     3. Compute stress \( \sigma^n(\xi_Q) \) by constitutive equation
     4. \( f^n_{int,e} \leftarrow f^n_{int,e} + B^T\sigma^n(\xi_Q)\sigma^n(\xi_Q) \) |
     END quadrature point loop
   - iv. Compute external nodal forces on element, \( f^n_{ext,e} \) |
   - v. \( f^n_e = f^n_{ext,e} - f^n_{int,e} \) |
   - vi. Compute \( \Delta t^{crit,e} \), if \( \Delta t^{crit,e} < \Delta t^{crit} \) then \( \Delta t^{crit} = \Delta t^{crit,e} \) |
   - vii. SCATTER \( f^n_e \) to global \( f^n \) |
4. END loop over elements |
5. \( \Delta t = \alpha \Delta t^{crit} \)

2.2.2 Implementation

Table 2.1 presents a flowchart for explicit time integration. The damping in this case is modeled by a linear viscous force \( f^{damp} = C^{damp}\nu \), so that the total force is \( f - C^{damp}\nu \).
The implementation of the velocity update is broken into two substeps by

\[
\begin{align*}
v^n & = v^{n-1/2} + \left( t^n - t^{n-1/2} \right) a^n \\
v^{n+1/2} & = v^n + \left( t^{n+1/2} - t^n \right) a^n
\end{align*}
\]  

(2.12)

This enables energy balance to be checked at integer time steps.

The cardinal dependent variables in this flowchart are the velocities and the Cauchy stress. Initial conditions must be given for the velocities, the Cauchy stress, and all material state variables. The initial displacements are assumed to vanish (initial displacements are meaningless in nonlinear analysis except for a hyperelastic material since the stress generally depends on the history of deformation).

The main part of the procedure is the calculation of the nodal forces, which is done in the second part. The major steps in this part are:

1. Extract the nodal displacements and velocities of the element from the global arrays by the GATHER operation.

2. The strain measures are computed at each quadrature point of the element.

3. The stresses are computed by the constitutive equation at each quadrature point. This is the part where our user subroutine will be inserted in.

4. Evaluate the internal nodal forces by integrating the product of the \( B \) matrix and the Cauchy stress over the domain of the element.

5. The nodal forces of the element are SCATTERED into the global array.

In the first time step, the strain measures and the stresses are not computed. Instead, as shown in Table 2.1, the internal nodal forces are computed directly for the initial stresses.

We also can see that an explicit method is easily implemented. Furthermore, explicit time integration is very robust, by which we mean that the explicit procedure seldom aborts due to failure of the numerical algorithm. The salient disadvantage of explicit integration, the price we have to pay for the simplicity of the method and its avoidance of the solution of equations, is the conditional stability of explicit methods. If the time step exceeds a critical value \( \Delta t_{\text{crit}} \), the solution will grow unboundedly.
The critical time step is also called the stable time step. A stable time step for a mesh of constant strain elements with rate-independent materials is given by

\[ \Delta t = \alpha \Delta t_{\text{crit}}, \quad \Delta t_{\text{crit}} = \frac{2}{\omega_{\text{max}}} \leq \min \frac{2}{\omega^{2}_e} = \min \frac{l_e}{c_e} \]  

(2.13)

where \( \omega_{\text{max}} \) is the maximum frequency of the linearized system, \( l_e \) is a characteristic length of element \( e \), \( c_e \) the current wavespeed in element \( e \), and \( \alpha \) is a reduction factor that accounts for the destabilizing effects of nonlinearities; a good choice for \( \alpha \) is \( 0.8 < \alpha < 0.98 \). The development of the above and additional discussion of time steps for ABAQUS/Explicit are given in [48].

2.2.3 Accuracy

The central difference method used in ABAQUS/Explicit is second order in time, i.e. the truncation error is of order \( \Delta t^2 \) of the displacements. We will see that the spatial error in the displacements in the \( L_2 \) norm for linear complete elements is of order \( h^2 \), where \( h \) is the element size. Although there are some technical differences between these two measures of error, the outcome is similar. Since the time step and the element size must be of the same order to meet the stability condition, (2.13), the time integration error and the spatial error are of the same order for central difference time integration. The accuracy of this method is sometimes inadequate for materials with rapidly varying stiffness, such as viscoplastic materials. However, the central difference method is accurate enough for elastic-plastic constitutive model if the time step is properly controlled.

2.2.4 Mass scaling, subcycling and dynamic relaxation

When a model contain a few very small or stiff elements, the efficiency of explicit integration is compromised severely, since the time step of the entire mesh is set by these very stiff elements. Several techniques are available for circumventing this difficulty:

1. Mass scaling: the masses of stiff elements is increased so that the time step is not decreased by these elements;

2. Subcycling: a smaller time step is used for the stiff elements.
Mass scaling should be used for problems where high frequency effects are not important. For example, in sheet metal forming process, which is essentially a static process, it causes no difficulties. On the other hand, if high frequency response is important, mass scaling is not recommended.

Subcycling is a technique which splits the model into subdomain and each is integrated with its own stable time step. This technique was first introduced by Belytschko, et al[49]. It has been shown that this method is only stable for first-order systems. Recent researchers have developed a stable subcycling methods for second-order systems. Interested users are suggested to refer to Smolinski and Daniel's works[50, 51].

Dynamic relaxation is often used in explicit codes to obtain static solutions. The basic idea is to apply the load very slowly and solve the dynamic system equations with enough damping so that oscillations are minimized. In path-dependent materials, dynamic relaxation often yields poor solutions. Furthermore, it is very slow. Newton methods combined with effective iterative solvers, such as preconditioned conjugate gradient or multigrid methods, are much faster and more accurate. Recently several researchers have made some important progress in this area, such as Don[52], Matthies[53], etc.

Among the several schemes available, only "mass scaling" is built into ABAQUS/Explicit. So this technique will be described further in Chapter 5.
Chapter 3

The Mathematical Formulation of Finite Element Elastic-Plastic Analysis

In the mathematical description of material behaviour, response of material is characterized by a constitutive equation which gives stresses as a function of the deformation history of a body. In this chapter, basic theoretical formulations of the constitutive model for elastic-plastic problems are presented. We focus on constitutive equations of hypoelastic-plasticity which is often plays an important role in large deformation elastic-plastic constitutive relations. Other continuum mechanics concepts, i.e. the yield criterion, work or strain hardening, etc., are also discussed. These relations can be introduced into ABAQUS by means of the interface of user material subroutine. Thus the numerical form of these relations are finally described.

3.1 Fundamentals of Elastic-Plastic Problem

In this section, we consider the elastic-plastic stress analysis of solid which conforms to three-dimensional conditions. The basic laws governing elastic-plastic material behaviour in a three-dimensional solid must be presented before the numerical aspects of the problem can be considered and to this end some concepts, such as the plastic potential, the flow rule and the hardening rule will be introduced. Only essential expressions will be provided in this thesis and a more complete theoretical treatment can
The object of the mathematical theory of plasticity is to provide a theoretical description of the relationship between stress and strain for a material which exhibits an elastic-plastic response. In essence, plastic behaviour is characterised by an irreversible straining which is not time dependent and which can only be sustained once a certain level of stress has been reached. In this thesis we outline the basic assumptions and associated theoretical expressions for a general continuum. In order to formulate a theory which models elastic-plastic material deformation three requirements have to be met:

- An explicit relationship between stress and strain must be formulated to describe material behaviour under elastic conditions, i.e. before the onset of plastic deformation.
- A yield criterion indicating the stress level at which plastic flow commences must be postulated.
- A relationship between stress and strain must be developed for post-yield behaviour, i.e. when the deformation is made up of both elastic and plastic components.

For large plastic strains involved in sheet metal forming process, the hypoelastic-plastic constitutive relations provide a general treatment. These formulations are typically based on an additive decomposition of the rate-of-deformation tensor into elastic and plastic parts and the elastic response is taken as hypoelastic.

### 3.1.1 Hypoelastic-plastic constitutive models

Hypoelastic-plastic models are typically used when elastic strains are small compared to the plastic strains. In these constitutive models, the additive decomposition of the rate-of-deformation tensor, $D$, into elastic and plastic parts is assumed:

$$ D = D^e + D^p $$  \hspace{1cm} (3.1)

The elastic response is hypoelastic: a suitable objective rate of stress is related to the elastic part of the rate-of-deformation tensor. The choice of objective stress rate in the constitutive response depends on several factors. Note that the choice of objective stress
3.1. FUNDAMENTALS OF ELASTIC-PLASTIC PROBLEM

rate should not be confused with the expression of a given constitutive relation in terms of different stress rates. The latter is accomplished by simply using the appropriate transformation between rates.

We first present a model based on the Cauchy stress with elastic response specified in terms of the Jaumann rate. The elastic response is specified by applying the hypoelastic law to the elastic part of the rate-of-deformation,

$$\sigma^J = C_{el}^J : D^e = C_{el}^J : (D - D^p)$$  \hspace{1cm} (3.2)

If the elastic moduli, $C_{el}^J$, are taken to be constant, they must be isotropic in order to satisfy the principle of material frame indifference.

The rate of plastic flow is given by

$$D^p = \dot{\lambda}r(\sigma, \kappa), \quad D^p_{ij} = \dot{\lambda}r_{ij}(\sigma, \kappa)$$  \hspace{1cm} (3.3)

where $\dot{\lambda}$ is a scalar plastic flow rate and $r(\sigma, \kappa)$ is the plastic flow direction. The plastic flow direction is often specified as $r = \partial \psi / \partial \sigma$ where $\psi$ is called the plastic flow potential. To avoid confusion of the plasticity parameter with the Lamé constant, the Lamé constant will subsequently be denoted as $\lambda^e$. The plastic flow direction depends on the Cauchy stress, $\sigma$, and on a set of internal variable denoted collectively as $\kappa$. Examples of scalar internal variables are the accumulated effective plastic strain and the void volume fraction. The backstress in kinematic hardening models is an example of an internal variable which is a second-order tensor.

Evolution equations for the internal variables are required and, for most plasticity models, can be specified as

$$\dot{\kappa} = \dot{\lambda}h(\sigma, \kappa), \quad \dot{\kappa}_\alpha = \dot{\lambda}h_\alpha(\sigma, \kappa)$$  \hspace{1cm} (3.4)

where $\alpha$ ranges over the number of internal variables and $h$ is a hardening parameter which will be discussed in details in the following. Here, the internal variables are a collection of scalars and the material time derivative is an objective rate. Note that the plastic parameter $\lambda$ or some function of it may be one of the internal variables. The evolution equation for the plastic parameter is obtained through the consistency condition below. The yield condition is

$$F(\sigma, \kappa) = 0$$  \hspace{1cm} (3.5)
The loading-unloading conditions are given by

\[ \dot{\lambda} \geq 0, \quad F \leq 0, \quad \dot{\lambda} F = 0 \]  

(3.6)

During plastic loading (\( \dot{\lambda} > 0 \)) the stress is required to remain on the yield surface \( F = 0 \). This can also be stated in terms of the *consistency* condition \( \dot{F} = 0 \) which can be expanded by the chain rule to give

\[
\dot{F} = F_\sigma : \dot{\sigma} + F_\kappa \cdot \dot{\kappa} = 0 \\
= (F_\sigma)_{ij} \dot{\sigma}_{jj} + (F_\kappa)_{\alpha} \dot{\kappa}_\alpha = 0
\]  

(3.7)

where we have adopted the notation \( F_\sigma = \partial F/\partial \sigma \) and \( F_\kappa = \partial F/\partial \kappa \).

The consistency condition involves the normal, \( F_\sigma \), to the yield surface. If the plastic flow direction is proportional to the normal to the yield surface, i.e., \( r \sim f_\sigma \), plastic flow is said to be *associative*, otherwise it is said to be *non-associative*. When the flow direction is given by the derivative of a plastic flow potential, the condition for associative plasticity is \( \psi_\sigma \sim F_\sigma \). For many materials, an approximate choice for the plastic potential is \( \psi = F \) which gives rise to an associative flow rule. Drucker[56] has shown that, when the yield surface is convex, associative plasticity models are stable for small strains if the strain hardening is positive.

From above results it follows that if \( F_\sigma \) and \( \sigma \) commute, i.e.,

\[ F_\sigma \cdot \sigma = \sigma \cdot F_\sigma \]  

(3.8)

then

\[ F_\sigma : \dot{\sigma} = F_\sigma : \sigma^{\nabla J} \]  

(3.9)

Substituting (3.9) into (3.7) gives

\[ \dot{F} = F_\sigma : \sigma^{\nabla J} + F_\kappa \cdot \dot{\kappa} = 0 \]  

(3.10)

Using the hypoelastic relation (3.2), the plastic flow relation (3.3) and the evolution equations (3.4) in (3.10) gives

\[ 0 = F_\sigma : C_{el}^{\sigma J} : (D - D^p) + F_\kappa \cdot \dot{\kappa} = F_\sigma : C_{el}^{\sigma J} : (D - \lambda r) + F_\kappa \cdot \dot{\lambda} h \]  

(3.11)

which can be solved for \( \dot{\lambda} \) to give

\[ \dot{\lambda} = \frac{F_\sigma : C_{el}^{\sigma J} : D}{-F_\kappa \cdot h + F_\sigma : C_{el}^{\sigma J} : r} \]  

(3.12)
Substituting (3.12) together with the plastic flow rule (3.3) into (3.2), we obtain a relation between the Jaumann rate of Cauchy stress and the total rate-of-deformation tensor:

\[
\dot{\sigma}^{\text{Jaumann}} = C_{\text{el}}^{\sigma} : (D - \lambda r) = C_{\text{el}}^{\sigma} : \left( D - \frac{F_{\sigma} : C_{\text{el}}^{\sigma} : D}{-F_{\sigma} \cdot h + F_{\sigma} : C_{\text{el}}^{\sigma} : r} \right) = C_{\text{pl}}^{\sigma} : D
\]  

(3.13)

The fourth-order tensor \( C_{\text{pl}}^{\sigma} \) is called the continuum elastic-plastic tangent modulus and is obtained by rearrangement of the expression in (3.13):

\[
C_{\sigma}^{\text{pl}} = C_{\text{el}}^{\sigma} - \frac{C_{\text{el}}^{\sigma} : r \otimes (F_{\sigma} : C_{\text{el}}^{\sigma})}{-F_{\sigma} \cdot h + F_{\sigma} : C_{\text{el}}^{\sigma} : r}
\]

\[
C_{ijkl}^{\sigma} = (C_{\text{el}}^{\sigma})_{ijkl} - \frac{(C_{\text{el}}^{\sigma})_{ijmn} r_{mn} (F_{\sigma})_{pq} (C_{\text{el}}^{\sigma})_{pqkl}}{-(F_{\sigma} \cdot h + (F_{\sigma})_{rs} (C_{\text{el}}^{\sigma})_{rstu} r_{tu})}
\]  

(3.14)

The symbol \( \otimes \) denotes the tensor or open product and is defined in the Notation. The elastic-plastic tangent modulus consists of the elastic tangent modulus and a term due to plastic flow. From symmetry of the stress rate and the rate-of-deformation, the elastic-plastic tangent modulus \( C_{\sigma}^{\text{pl}} \) has both minor symmetries. It has major symmetry, \( C_{ijkl}^{\sigma} = C_{klji}^{\sigma} \), when \( C_{\text{el}}^{\sigma} : r \sim F_{\sigma} : C_{\text{el}}^{\sigma} \) or, alternatively, when plastic flow is associative, i.e. \( r \sim F_{\sigma} \) (major symmetry of the elastic moduli is assumed). The above equations are summarized in Table 3.1

### 3.1.2 The Yield Criterion

In analysis of sheet metal forming process, it is vital to identify the various possible modes of failures of the sheet blank under service conditions. Excessive deformation is on such mode of failure in which the dimensional changes of the sheet metal under load may be too great to allow it to continue to fulfill its intended design function. A yield criterion provides a means of identifying those critical stress state for which yielding is initiated. In order to define the yield criterion, it is common to have the following two assumptions

1. Yield is independent of hydrostatic pressure;
2. The solid is isotropic.

The first assumption implies that the yield criterion can only depend on the deviatoric stress components

\[
S_{ij} = \sigma_{ij} - \frac{\sigma_{kk}}{3} \delta_{ij}
\]  

(3.15)
3.1. FUNDAMENTALS OF ELASTIC-PLASTIC PROBLEM

<table>
<thead>
<tr>
<th>Table 3.1: Elastic-plastic constitutive model (Cauchy stress formulation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Rate-of-deformation tensor:</td>
</tr>
<tr>
<td>[ D = D^e + D^p ]</td>
</tr>
<tr>
<td>2. Stress rate:</td>
</tr>
<tr>
<td>[ \sigma^\nabla J = C^\sigma J : D^e = C^\sigma J : (D - D^p) ]</td>
</tr>
<tr>
<td>3. Plastic flow rule and evolution equations:</td>
</tr>
<tr>
<td>[ D^p = \dot{\lambda} \tau(\sigma, \kappa), \quad D^p_{ij} = \dot{\lambda} \tau_{ij}(\sigma, \kappa) ]</td>
</tr>
<tr>
<td>4. Yield condition:</td>
</tr>
<tr>
<td>[ F(\sigma, \kappa) = 0 ]</td>
</tr>
<tr>
<td>5. Loading-unloading conditions:</td>
</tr>
<tr>
<td>[ \dot{\lambda} \geq 0, \quad \lambda F &lt; 0, \quad \lambda F = 0 ]</td>
</tr>
<tr>
<td>6. Plastic rate parameter (consistency condition):</td>
</tr>
<tr>
<td>[ \dot{\lambda} = \frac{F_\sigma : C^\sigma J : D}{-F_\kappa \cdot h + F_\sigma : C^\sigma J : \tau} ]</td>
</tr>
<tr>
<td>7. Stress rate - total rate-of-deformation relation:</td>
</tr>
<tr>
<td>[ \sigma^\nabla J = C^\sigma J D \quad \sigma^\nabla <em>{ij} = C^\sigma J</em>{ijkl} D_{kl} ]</td>
</tr>
</tbody>
</table>

During elastic loading or unloading, \( C^\sigma J = C^\sigma el \); for plastic loading \( C^\sigma J \) is given by the continuum elastic-plastic tangent modulus

\[ C^\sigma J = C^\sigma el - \frac{(C^\sigma el : \tau) \Theta (F_\sigma : C^\sigma el)}{-F_\kappa \cdot h + F_\sigma : C^\sigma el : \tau} \]
\[ C^\sigma J_{ijkl} = (C^\sigma el)_{ijkl} - \frac{(C^\sigma el)_{ijmn} \tau_{mn}(F_\sigma)_{pq} (C^\sigma el)_{pqkl}}{-(F_\sigma)_{h\alpha}(F_\sigma)_{rs}(C^\sigma el)_{rstu\tau}} \]

The second assumption implies that the onset of yield can only depend on the magnitudes of the principal stresses \( S_1, S_2, S_3 \), and cannot depend on the principal stress directions. Another way to state the same thing is to note that yield can only depend on the invariants of the deviatoric stress tensor

\[ J_1 = 0 \]
\[ J_2 = \frac{1}{2} S_{ij} S_{ij} \]  \( (3.16) \)
\[ J_3 = \frac{1}{3} S_{ik} S_{kj} S_{ji} \]
3.1. FUNDAMENTALS OF ELASTIC-PLASTIC PROBLEM

Where we are using $J$ to denote the invariants of the deviatoric stress. The conditions necessary to cause yield could be expressed as $F(\sigma_1, \sigma_2, \sigma_3, \text{material state variables})=0$ or $F(J_2, J_3, \text{material state variables})=0$ where \textit{material state variables} somehow characterizes the strength of the solid and could include a strain history dependent yield stress and temperature, for example.

Another way to visualize yield criteria is to represent a stress state in principal stress state. Since the material response can only depend on $\sigma_1, \sigma_2, \sigma_3$, we can represent all loading conditions as a point in 3D space with $\sigma_1, \sigma_2, \sigma_3$ as the axes (as shown in Figure 3-1).

A general stress state therefore consists of a hydrostatic component parallel to the line $\sigma_1 = \sigma_2 = \sigma_3$, and a deviatoric component, which must be perpendicular to the line (by definition, deviatoric stress has no hydrostatic component). The solid yields when the deviatoric stress reaches a critical magnitude. This condition defines the cylinder shown.

If we look down the axis of the cylinder, we would see a view like the one shown in Figure 3-2. We can see that for an isotropic solid, we can freely interchange $\sigma_1, \sigma_2, \sigma_3$ without affecting yield. This means that the yield surface must have 6 axes of symmetry as shown in the picture.
### 3.1. Fundamentals of Elastic-Plastic Problem

#### Tresca Yield Criterion

The Tresca yield criterion states that yielding begins when the maximum shear stress reaches a certain value. In Figure 3-2, Tresca yield criterion is shown as hexagonal surface which can be expressed as

\[
\max \left\{ \|\sigma_1 - \sigma_2\|, \|\sigma_1 - \sigma_3\|, \|\sigma_2 - \sigma_3\| \right\} - Y = 0
\]

\[
4(J_2 - k)(J_2 - 4k^2)^2 - 27J_3^2 = 0
\]

(3.17)

where \(Y\) and \(k\) are material constants (specifying the size of the yield locus) to be experimentally determined and which may be a function of the hardening parameter \(k\) and represent the yield stress of the solid in uniaxial tension and simple shear, respectively. For the Tresca surface they are related by \(Y = 2k\).

#### Von Mises Yield Criterion

Von-Mises yield criterion is the circular surface, which can be expressed in various ways

\[
\sqrt{\frac{1}{2} \left[ (\sigma_1 - \sigma_2)^2 + (\sigma_1 - \sigma_3)^2 + (\sigma_2 - \sigma_3)^2 \right]} - Y = 0
\]

\[
\sqrt{\frac{3}{2} S_{ij} S_{ij}} - Y = 0
\]

\[
\bar{\sigma} - Y = 0
\]

\[
\sqrt{J_2} - Y = 0
\]

(3.18)
in which $\bar{\sigma} = \sqrt{3S_{ij}S_{ij}}/2$ is the von-Mises effective stress. For most metals von Mises’ law fits the experimental date more closely than Tresca’s, but it frequently happens that the Tresca criterion is simpler to use in theoretical applications. As for other two important yield criteria, Mohr-Coulomb yield Criterion and Drucker-Prager yield Criterion, since they are normally used for concrete, rock and soil problems and will not used in this research, the user are suggested to refer Fenner’s book[57] for detailed introduction.

**Hill’s Yield Criterion**

Sheet metals exhibit a highly anisotropic material behavior by cold rolling. It is therefore of major importance to extend the plastic instability analysis to anisotropic materials. The yield criterion of Hill was adopted[58]. Hill’s potential function is an extension from the Mises function and can be expressed as

$$F(\sigma) = \sqrt{F(\sigma_y - \sigma_z)^2 + G(\sigma_z - \sigma_x)^2 + H(\sigma_x - \sigma_y)^2 + 2L\sigma_{yz}^2 + 2M\sigma_{zz}^2 + 2N\sigma_{xy}^2}$$

(3.19)

where $\sigma_{ij}$ denotes the stress components and $F$, $G$, $H$, $L$, $M$ and $N$ are material constants obtained by tests of the material in different directions. These constants can be expressed in terms of six yield stress ratios $R_{11}$, $R_{22}$, $R_{33}$, $R_{12}$, $R_{13}$ and $R_{23}$ according to

$$F = \frac{1}{2} \left( \frac{1}{R_{22}^2} + \frac{1}{R_{33}^2} - \frac{1}{R_{11}^2} \right)$$

$$G = \frac{1}{2} \left( \frac{1}{R_{33}^2} + \frac{1}{R_{22}^2} - \frac{1}{R_{11}^2} \right)$$

$$H = \frac{1}{2} \left( \frac{1}{R_{11}^2} + \frac{1}{R_{22}^2} - \frac{1}{R_{33}^2} \right)$$

$$L = \frac{3}{2R_{23}^2}$$

$$M = \frac{3}{2R_{13}^2}$$

$$N = \frac{3}{2R_{12}^2}$$

In sheet metal forming applications, it is common to use anisotropic material data in terms of ratios of width strain to thickness strain. The stress ratios can then be defined as:

$$R_{11} = R_{13} = R_{23} = 1, \quad R_{22} = \sqrt{\frac{r_{90}(r_0 + 1)}{r_0(r_{90} + 1)}}$$
3.1.3 Strain Hardening

After initial yielding, the stress level at which further plastic deformation occurs may be dependent on the current degree of plastic straining. Such a phenomenon is termed strain hardening. Thus the yield surface will vary at each stage of the plastic deformation, with the subsequent yield surfaces being dependent on the plastic strains in some way. Some alternative models which describe strain hardening in a material are illustrated in Figure 3-3. If the subsequent yield surfaces preserve their shape and orientation but translate in the stress space as a rigid body as shown in Figure 3-3(c), kinematic hardening is said to take place. Such a hardening model is useful in situations where components are subjected to cyclic loading as it gives rise to the experimentally observed Bauschinger. In this research, the isotropic hardening rule is employed and will be introduced in details.

Isotropic hardening

If the subsequent yield surfaces are a uniform expansion of the original yield curve, without translation, as shown in Figure 3-3(b) the strain hardening model is said to be isotropic.

The progressive development of the yield surface can be defined by relating the yield stress $Y$ to the plastic deformation by mean of the hardening parameter $\kappa$. This can be done in two ways. Firstly the degree of work hardening can be postulated to be a function of the total plastic work, $W_p$, only. Then,

$$\kappa = F(W_p)$$  \hspace{1cm} (3.20)

Alternatively $\kappa$ can be related to a measure of the total plastic deformation termed the effective plastic strain. Then the hardening parameter is assumed to be defined as

$$\kappa = \bar{\varepsilon}_p$$  \hspace{1cm} (3.21)
3.2 Numerical expression of constitutive relations

Before the onset of plastic yielding the relationship between stress and strain is given by the standard linear elastic expression.

\[
\Delta \varepsilon = (C_{el}^{\sigma J})^{-1} : \Delta \sigma
\]

(3.22)

where \( \bar{\varepsilon}_p \) is the result of integrating the increment of the effective strain \( d\bar{\varepsilon}_p \) over the strain path. This behaviour is termed strain hardening. Only an isotropic strain hardening model will be considered in this research.

Figure 3-3: Mathematical models for representation of strain hardening behaviour
3.2. NUMERICAL EXPRESSION OF CONSTITUTIVE RELATIONS

For three-dimensional isotropic problems

\[ \{ \Delta \sigma \} = (\Delta \sigma_x, \Delta \sigma_y, \Delta \sigma_z, \Delta \tau_{xy}, \Delta \tau_{yz}, \Delta \tau_{zx})^T \]  
(3.23)

\[ \{ \Delta \varepsilon \} = (\Delta \varepsilon_x, \Delta \varepsilon_y, \Delta \varepsilon_z, \Delta \gamma_{xy}, \Delta \gamma_{yz}, \Delta \gamma_{zx})^T \]  
(3.24)

\[ C_{el}^J = \begin{bmatrix} 
A_1 & A_2 & A_2 & 0 & 0 & 0 \\
A_1 & A_2 & 0 & 0 & 0 \\
A_1 & 0 & 0 & 0 \\
A_3 & 0 & 0 \\
sym. & A_3 
\end{bmatrix} \]  
(3.25)

where

\[ A_1 = \frac{E(1 - \nu)}{(1 + \nu)(1 - 2\nu)}; \quad A_2 = \frac{E \nu}{(1 + \nu)(1 - 2\nu)}; \quad A_3 = \frac{E}{2(1 + \nu)} \]

in which \( E \) and \( \nu \) are respectively the elastic modulus and Poisson's ratio of the materials.

In order to calculate the elastic-plastic tangent modulus \( C_{el}^J \) in Equation 3.14, we require to express the plastic flow direction \( r = \partial \psi / \partial \sigma \), the normal to the yield surface \( F_{\sigma} = \partial F / \partial \sigma \) and the hardening parameters \( \kappa \) in a form suitable for numerical computation. For numerical computation it is convenient to rewrite the yield function in terms of alternative stress invariants. The main advantage of this formulation is that it permits the computer coding of the yield function and flow rule in a general form. The condition is simplified when an associative flow rule is considered, i.e., \( \psi_{\sigma} \sim F_{\sigma} \). In this situation only \( F_{\sigma} \) needs to be determined. Only von Mises yield criterion and Hill's yield criterion are used in this research and the numerical forms of these two yield criteria are presented as follows:

### 3.2.1 Von Mises yield criterion

For associative plasticity, the von Mises stress potential can be expressed as:

\[ F(\sigma) = q \]  
(3.26)

where

\[ q = \sqrt{\frac{3}{2} S : S} \]  
(3.27)
in which $S$ is the deviatoric stress:

$$ S = \sigma_{ij} - \frac{1}{3} \delta_{ij} \sigma_{kk} $$  \hspace{1cm} (3.28)

so $q$ can be explicitly written as

$$ q = \sqrt{2J_2} $$  \hspace{1cm} (3.29)

with

$$ J_2 = \frac{1}{2} S : S $$

$$ = \frac{1}{6} \left[ (\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 \right] $$  \hspace{1cm} (3.30)

$$ = \frac{1}{2} \left[ \sigma_x^2 + \sigma_y^2 + \sigma_z^2 \right] + \sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2 $$

As shown previously, the yield function of von Mises is a circle in the plane normal to the hydrostatic axis in principal stress space. For this function, the normal to the yield surface $r = F_\sigma$ can be expressed as

$$ r = \frac{\partial F}{\partial \sigma} = \frac{1}{q} S $$  \hspace{1cm} (3.31)

The second derivative of the yield function with respect to the stress needs to be determined since we will employ the backward Euler algorithm into the ABAQUS user material subroutine. Based on the first derivative computed in (3.31), we can determine $r_\sigma$ as

$$ r_\sigma = \frac{\partial^2 F}{\partial \sigma \partial \sigma} = \frac{1}{q} \left( \frac{3}{2} \mathfrak{S} - \frac{1}{2} II - \frac{\partial F}{\partial \sigma} \right) $$  \hspace{1cm} (3.32)

in which $\mathfrak{S}$ is the fourth-order unit tensor.

### 3.2.2 Hill’s yield criterion

Hill’s stress potential is a simple extension of the Mises function to allow anisotropic behavior. This function is

$$ F(\sigma) = \sqrt{F(\sigma_y - \sigma_z)^2 + G(\sigma_x - \sigma_y)^2 + H(\sigma_x - \sigma_z)^2 + 2L\sigma_{yz}^2 + 2M\sigma_{zx}^2 + 2N\sigma_{xy}^2} $$  \hspace{1cm} (3.33)

in terms of rectangular Cartesian stress components, where $F,G,H,L,M,N$ are material constants defined previously.
3.2. NUMERICAL EXPRESSION OF CONSTITUTIVE RELATIONS

For this function, \( r = f_\sigma \) can be expressed as

\[
\frac{\partial F}{\partial \sigma} = \frac{1}{F} b
\]

(3.34)

where

\[
b = \begin{bmatrix}
-G(\sigma_z - \sigma_x) + H(\sigma_x - \sigma_y) \\
E(\sigma_y - \sigma_z) - H(\sigma_x - \sigma_y) \\
-F(\sigma_y - \sigma_z) + G(\sigma_z - \sigma_x) \\
2N\sigma_{xy} \\
2M\sigma_{xz} \\
2L\sigma_{yz}
\end{bmatrix}
\]

(3.35)

The second derivative of the yield function with respect to the stress \( r_\sigma \)

\[
\frac{\partial^2 F}{\partial \sigma \partial \sigma} = \frac{1}{F} \left( \frac{\partial b}{\partial \sigma} - \frac{1}{F^2} bb \right)
\]

(3.36)

where

\[
\frac{\partial b}{\partial \sigma} = \begin{bmatrix}
G + H & -H & -G & 0 & 0 & 0 \\
-H & F + H & -F & 0 & 0 & 0 \\
-G & -F & F + G & 0 & 0 & 0 \\
0 & 0 & 0 & 2N & 0 & 0 \\
0 & 0 & 0 & 0 & 2M & 0 \\
0 & 0 & 0 & 0 & 0 & 2L
\end{bmatrix}
\]

(3.37)

3.2.3 Hardening rule

It is quite generally postulated that yielding can occur only if the stress \( \sigma \) satisfy the general yield criterion

\[
F(\sigma, \kappa) = 0
\]

(3.38)

where \( \kappa \) is a hardening parameter, which is dependent on the plastic loading history.

A typical hardening parameter commonly used in practice is the effective plastic strain \( \bar{\varepsilon}_p \) defined by

\[
\bar{\varepsilon}_p = \sqrt{d\varepsilon_p : d\varepsilon_p}
\]

(3.39)

The effective plastic strain is seen as the accumulation of plastic strain. Another commonly used hardening parameter is the plastic work \( W_p \) defined by

\[
W_p = \sigma : d\varepsilon_p
\]

(3.40)
The plastic work physically represents the energy dissipation associated with plastic deformation.

Equations (3.39) and (3.40) together with the flow rule of Equation (3.3) indicate that the increment of the hardening parameter $\Delta \kappa$ can be expressed in general as

$$\Delta \kappa = \Delta \lambda h$$  \hspace{1cm} (3.41)

where $h$ is a scalar function defined by:

$$h = \begin{cases} 
\| \frac{\partial F}{\partial \sigma} \| & \text{Strain hardening} \\
\sigma \frac{\partial F}{\partial \sigma} & \text{Work hardening} 
\end{cases}$$

In this thesis, an isotropic strain hardening model is employed. The hardening parameter $\kappa$ is explicitly given by:

$$\Delta \kappa = \Delta \lambda (\sigma, \kappa, \Delta \varepsilon) \| b \|$$  \hspace{1cm} (3.42)

with

$$b = \left\{ \sqrt{2} \frac{\partial F}{\partial \sigma_x}, \sqrt{2} \frac{\partial F}{\partial \sigma_y}, \sqrt{2} \frac{\partial F}{\partial \sigma_z}, \frac{\partial F}{\partial \sigma_{xy}}, \frac{\partial F}{\partial \sigma_{yz}}, \frac{\partial F}{\partial \sigma_{zx}} \right\}$$  \hspace{1cm} (3.43)
Chapter 4

Stress Update Algorithms

The numerical algorithm for integrating rate constitutive equations is called a constitutive integration algorithm or a stress update algorithm. In this Chapter, constitutive integration algorithms are presented for rate-independent plasticity. Both implicit and explicit stress update schemes are discussed in details. Finally a new explicit stress update algorithm for large deformation elastic-plastic materials is presented. The scheme is developed for use in finite element plasticity calculation and solve for the stress increments, assuming that the strain increments are known. To implement the new stress update algorithm, other techniques, such as determination of initial intersection with the yield surface, stress correction methods, etc., are also discussed. All the implicit and explicit stress update schemes mentioned below will be implemented into an user material subroutine for analyzing a typical sheet metal deep drawing process.

4.1 Implicit algorithms

Implicit methods are attractive because the consistency condition is automatically satisfied at the end of integration. Furthermore, they do not require to determine initial yielding point if plastic yielding occurs at the stress point. A comprehensive discussion of various implicit integration schemes for elastic-plastic models can be found in Crisfield[47], Ortiz and Simo[59], and Belytschko[27]. One of the most popular of these is the backward Euler algorithm which is based on the concept of an elastic predictor with a plastic return mapping. In its most general form, the final stresses and hardening parameters are found by solving a small system of non-linear equations at each
4.1. IMPLICIT ALGORITHMS

Gauss point iteratively. Although most recent work preferred to use the backward-Euler scheme, it is difficult to implement such procedure for complex constitutive relations since it requires second order derivatives of the yield function and plastic potential. Moreover, for yield surfaces with vertices or rapid changes in curvature, divergence may occur and it is advisable to use multi-vector return schemes (Crisfield[47]). This procedure can be greatly simplified if von Mises plasticity is considered. For the special case of $J_2$ flow theory, the general return mapping algorithm above reduces to the well-known radial return method ([27] and [21]). The key feature of the radial return method is that the solution of the set of nonlinear equations is simplified to a linear function of $\Delta \lambda$ due to the radial and unchanged norm of the plastic flow direction. A detailed introduction of various implicit schemes are given as follows.

4.1.1 Return mapping algorithms

Consider the constitutive equations for hypoelastic-plasticity given in Table 3.1:

\[
\begin{align*}
\sigma &= C_e : \dot{\varepsilon} = C_e : (\dot{\varepsilon} - \dot{\varepsilon}^p) \\
\dot{\varepsilon}^p &= \dot{\lambda} r \\
\dot{\kappa} &= \dot{\lambda} h \\
\bar{F} &= F_\sigma : \dot{\sigma} + F_\kappa : \dot{\kappa} = 0 \\
\dot{\lambda} &\geq 0, \quad F \leq 0, \quad \dot{\lambda} F \leq 0 
\end{align*}
\]

The purpose of a constitutive integration algorithm is, given the set $(\varepsilon_n, \varepsilon_n^p, \kappa_n)$ at time $n$ and the strain increment $\Delta \varepsilon = \Delta t \dot{\varepsilon}$, to compute $(\varepsilon_{n+1}, \varepsilon_{n+1}^p, \kappa_{n+1})$ and satisfy the loading-unloading conditions. Note that the stress at time $n+1$ is given by $\sigma_{n+1} = C_e : (\varepsilon_{n+1} - \varepsilon_{n+1}^p)$. The consistency condition is solved for $\dot{\lambda}$ to give

\[
\dot{\lambda} = \frac{F_\sigma : C_e : \dot{\sigma}}{-F_\kappa : h + F_\sigma : C_e : r} \tag{4.2}
\]

where the elastic moduli is expressed as $C_e$ for simplicity. It might be supposed that we could use this value of the plasticity parameter to provide rates of stress, plastic strain and internal variables for updating and write the simple forward Euler integration
4.1. IMPLICIT ALGORITHMS

4.1. IMPLICIT ALGORITHMS

scheme

\[ \varepsilon_{n+1} = \varepsilon_n + \Delta \varepsilon \]
\[ \varepsilon_{n+1}^p = \varepsilon_n^p + \Delta \lambda_n \tau_n \]
\[ \kappa_{n+1} = \kappa_n + \Delta \lambda_n h_n \quad (4.3) \]
\[ \sigma_{n+1} = C_e : (\varepsilon_{n+1} - \varepsilon_{n+1}^p) = \sigma_n + C_{ep} : \Delta \varepsilon \]

where \( \Delta \lambda_n = \Delta t \dot{\lambda}_n \). However, these updated values of stress and internal variables do not satisfy the yield condition at the next step so \( F_{n+1} = F(\sigma_{n+1}, \kappa_{n+1}) \neq 0 \) and the solution drifts from the yield surface, often resulting in inaccurate solutions. The integration algorithm (4.3) is sometimes called a tangent modulus update scheme. This approach formed the basis for early work in computational rate-independent plasticity but because of the inaccuracies of the method it is no longer favored.

This leads us to consider alternative methods for integrating the rate constitutive equations. One of the objectives of these alternative methods is to enforce consistency at the end of the time step, i.e., \( F_{n+1} = 0 \), to avoid drift from the yield surface. There are many different approaches to integrating the constitutive equations. A summary of the principal methods is given by Simo and Hughes[60]. Here we focus mainly on a class of methods called return mapping algorithms which are robust and accurate and which are widely used in practice. The popular radial return method for von Mises plasticity is a special case of the return mapping algorithms.

Return mapping schemes consist of an initial elastic-predictor step, involving an excursion (in stress space) away from the yield surface, and a plastic-corrector step which returns the stress to the updated yield surface. Two ingredients of the method are an integration scheme which transforms the set of constitutive equations into a set of nonlinear algebraic equations and a solution scheme for the nonlinear algebraic equations. The method can be based on different integration schemes such as the generalized trapezoidal rule, generalized mid-point rule or Runge-Kutta methods, for example. Here we consider a fully implicit method based on backward Euler scheme.

4.1.2 Fully implicit backward Euler scheme

In the fully implicit backward Euler method, the increments in plastic strain and internal variables are calculated at the end of the step and the yield condition is enforced
at the end of the step. Thus the integration scheme is written as

\[
\begin{align*}
\varepsilon_{n+1} &= \varepsilon_n + \Delta \varepsilon \\
\varepsilon_{n+1}^p &= \varepsilon_n^p + \Delta \lambda_{n+1} \gamma_{n+1} \\
\kappa_{n+1} &= \kappa_n + \Delta \lambda_{n+1} \gamma_{n+1} \\
\sigma_{n+1} &= C_e : (\varepsilon_{n+1} - \varepsilon_{n+1}^p) \\
F_{n+1} &= F(\sigma_{n+1}, \kappa_{n+1}) = 0
\end{align*}
\] (4.4)

Given the set \((\varepsilon_n, \varepsilon_n^p, \kappa_n)\) at time \(n\) and the strain increment \(\Delta \varepsilon\), (4.4) are a set of nonlinear algebraic equations for \((\varepsilon_{n+1}, \varepsilon_{n+1}^p, \kappa_{n+1})\). It is noted that the variables are updated from the converged values at the end of the previous time-step. This avoids nonphysical effects such as spurious unloading which can occur when path-dependent plasticity equations are driven by nonconverged values of the plastic strain and internal variables. The strain \(\varepsilon_{n+1}\) is the strain obtained from the solution of the system of equations at time \(n + 1\). If the solution procedure is implicit, it is understood that \(\varepsilon_{n+1}\) is the total strain after the last iteration of the implicit solution scheme.

A geometric interpretation of the algorithm is given as follows. First note that from (4.4), the plastic strain increment is given by

\[
\Delta \varepsilon_{n+1}^p \equiv \varepsilon_{n+1}^p - \varepsilon_n^p = \Delta \lambda_{n+1} \gamma_{n+1}
\] (4.5)

Substituting this expression into 4.4 gives

\[
\begin{align*}
\sigma_{n+1} &= C_e : (\varepsilon_{n+1} - \varepsilon_n^p - \Delta \varepsilon_{n+1}^p) \\
&= C_e : (\varepsilon_n + \Delta \varepsilon - \varepsilon_n^p - \Delta \varepsilon_{n+1}^p) \\
&= C_e : (\varepsilon_n - \varepsilon_n^p + C_e : \Delta \varepsilon - C_e : \Delta \varepsilon_{n+1}^p) \\
&= (\sigma_n + C_e : \Delta \varepsilon) - C_e : \Delta \varepsilon_{n+1}^p \\
&= \sigma_{n+1}^{\text{trial}} - C_e : \Delta \varepsilon_{n+1}^p \\
&= \sigma_{n+1}^{\text{trial}} - \Delta \lambda_{n+1} C_e : r_{n+1}
\end{align*}
\] (4.6)

where \(\sigma_{n+1}^{\text{trial}} = \sigma_n + C_e : \Delta \varepsilon\) is the trial stress of elastic predictor and the quantity 
\(-\Delta \lambda_{n+1} C_e : r_{n+1}\) is the plastic corrector which returns or projects the trial stress onto the suitable updated (accounting for hardening) yield surface along a direction specified by the plastic flow direction at the end-point (see Figure 4-1). The elastic-predictor
4.1. IMPLICIT ALGORITHMS

Figure 4-1: Closest point projection scheme for associative plasticity: \( r_{n+1} \sim \partial f / \sigma_{n+1} \)

phase is driven by the increment in total strain while the plastic-corrector phase is driven by the increment \( \Delta \lambda_{n+1} \) in the plasticity parameter. Thus, during the elastic-predictor stage, the plastic strain and internal variables remain fixed, and during the plastic-corrector stage, the total strain is fixed. A consequence of this is that, from (4.4) during the plastic-corrector phase,

\[
\Delta \sigma_{n+1} = -C_e : \Delta \varepsilon^p_{n+1} = -\Delta \lambda_{n+1} C_e : r_{n+1}
\]  

which is a result we will use below in the solution of (4.4).

The solution of the set of nonlinear algebraic equations (4.4) is typically obtained by a Newton procedure. As discussed by Simo and Hughes[60], a Newton procedure based on the systematic liberalization of equations (4.4) gives rise to a plastic-corrector return to the yield surface based on the concept of closest point projection. During the plastic-corrector stage of the algorithm, the total strain is constant and liberalization is with respect to the increment \( \Delta \lambda \) in the plasticity parameter. We use the following notation in the Newton procedure: for the linearization of an equation \( g(\Delta \lambda) = 0 \), with \( \Delta \lambda^{(0)} = 0 \), at the \( k \)th iteration we write

\[
g^{(k)} + \left( \frac{dg}{d\Delta \lambda} \right)^{(k)} \delta \lambda^{(k)} = 0, \quad \Delta \lambda^{(k+1)} = \Delta \lambda^{(k)} + \delta \lambda^{(k)}
\]  

where \( \delta \lambda^{(k)} \) is the increment in \( \Delta \lambda \) at the \( k \)th iteration. For the most part, we will omit the load or time increment subscript \( n+1 \) on quantities in the remainder of this chapter. Thus, unless indicated otherwise, all quantities are evaluated at time \( n+1 \).
4.1. IMPLICIT ALGORITHMS

We write the plastic updates and yield condition in 4.4 in the form (4.8), suitable for Newton iteration:

\[ a = \varepsilon^p + \varepsilon_n^p + \Delta \lambda r = 0 \]

\[ b = -\kappa + \kappa_n + \Delta \lambda h = 0 \]

\[ F = F(\sigma, \kappa) = 0 \]

(4.9)

Linearization of these equations gives (using (4.7) in the form \( \Delta \varepsilon^{p(k)} = -C_e^{-1} : \Delta \sigma^{(k)} \))

\[ a^{(k)} + C_\sigma \Delta \sigma^{(k)} + \Delta \lambda^{(k)} r^{(k)} + \delta \lambda^{(k)} r^{(k)} = 0 \]

\[ b^{(k)} - \Delta \kappa^{(k)} \Delta h^{(k)} + \delta \lambda^{(k)} h^{(k)} = 0 \]

\[ F^{(k)} = F^{(k)}(\sigma) = \Delta \sigma^{(k)} + F^{(k)}_\kappa \cdot \Delta \kappa^{(k)} = 0 \]

(4.10)

where

\[ \Delta r^{(k)} = r^{(k)}_{\sigma} : \Delta \sigma^{(k)} + r^{(k)}_{\kappa} \cdot \Delta \kappa^{(k)} \]

\[ \Delta h^{(k)} = h^{(k)}_{\sigma} : \Delta \sigma^{(k)} + h^{(k)}_{\kappa} \cdot \Delta \kappa^{(k)} \]

and where a subscript \( \sigma \) or \( \kappa \) denotes a partial derivative. Equations (4.10) are a set of three equations which can be solved for \( \Delta \sigma^{(k)} \), \( \Delta \kappa^{(k)} \) and \( \delta \lambda^{(k)} \). Substituting (A.2) into the first two of (4.10) and writing the resulting pair of equations in matrix form gives

\[ [A^{(k)}]^{-1} \begin{bmatrix} \Delta \sigma^{(k)} \\ \Delta \kappa^{(k)} \end{bmatrix} = - \begin{bmatrix} \delta \lambda^{(k)} \\ \delta \lambda^{(k)} \end{bmatrix} \begin{bmatrix} \tilde{a}^{(k)} \\ \tilde{b}^{(k)} \end{bmatrix} \]

(4.11)

where

\[ [A^{(k)}]^{-1} = \begin{bmatrix} C_e^{-1} + \Delta \lambda a_\sigma & \Delta \lambda a_\kappa \\ \Delta \lambda h_\sigma & -I + \Delta \lambda h_\kappa \end{bmatrix}^{(k)} \]

\[ \begin{bmatrix} \tilde{a}^{(k)} \\ \tilde{b}^{(k)} \end{bmatrix} = \begin{bmatrix} a^{(k)} \\ b^{(k)} \end{bmatrix} \]

\[ \begin{bmatrix} \tilde{a}^{(k)} \\ \tilde{b}^{(k)} \end{bmatrix} = \begin{bmatrix} r^{(k)} \\ h^{(k)} \end{bmatrix} \]

Solving (4.21) for the stress and internal variable increments gives

\[ \begin{bmatrix} \Delta \sigma^{(k)} \\ \Delta \kappa^{(k)} \end{bmatrix} = - [A^{(k)}] \begin{bmatrix} \tilde{a}^{(k)} \\ \tilde{b}^{(k)} \end{bmatrix} - \delta \lambda^{(k)} [A^{(k)}] \begin{bmatrix} r^{(k)} \\ h^{(k)} \end{bmatrix} \]

(4.12)
Substituting this result into (4.10) and solving for $\delta \lambda^{(k)}$ we get

$$\delta \lambda^{(k)} = \frac{F^{(k)} - \partial F^{(k)} A^{(k)} \tilde{a}^{(k)}}{\partial F^{(k)} A^{(k)} \tilde{r}^{(k)}}$$  (4.13)

where we have used the notation $\partial F = [F_y F_x]$.

Thus, the update of the plastic strain, internal variables and the plasticity parameter is

$$\varepsilon_p^{(k+1)} = \varepsilon_p^{(k)} + \Delta \varepsilon_p^{(k)} = \varepsilon_p^{(k)} - C_{\varepsilon}^{-1} : \Delta \sigma^{(k)}$$

$$\kappa^{(k+1)} = \kappa^{(k)} + \Delta \kappa^{(k)}$$

$$\Delta \lambda^{(k+1)} = \Delta \lambda^{(k)} + \delta \lambda^{(k)}$$

with the increments as given in (4.12) and (4.13). The Newton procedure is continued until convergence to the updated yield surface is achieved to within a sufficient tolerance. As noted by Simo and Hughes [60], this procedure is implicit and involves the solution of a local (at the level of the element integration points) system of equations (4.21). A complicating feature of the method is that the gradients $r_{\sigma}$, $r_{\kappa}$, $h_{\sigma}$ and $h_{\kappa}$ of the plastic flow direction and the plastic moduli are required. These expression may be difficult or even impossible to obtain for complex constitutive models, making it infeasible to use with some new constitutive models. The complete backward Euler algorithm is given in Table (4.1).

### 4.1.3 Application to $J_2$ flow theory - radial return algorithm

For the special case of $J_2$ flow theory plasticity, the general return mapping algorithm above reduces to the well-known radial return method ([20, 61]).

Recall from (4.7) that the trial stress, denoted here by $\sigma^{(0)}$, is given by the elastic predictor, i.e.,

$$\sigma^{(0)} = C_{\varepsilon} : (\varepsilon_{n+1} - \varepsilon_p^{(0)})$$  (4.14)

The stress at iteration $k$ is given by

$$\sigma^{(k)} = \sigma^{(0)} - \Delta \lambda^{(k)} c_{\varepsilon} : r^{(k)}$$  (4.15)

For $J_2$ flow theory, we note that the plastic flow direction is in the direction of the deviatoric stress and is given by $r = 3\sigma^{dev}/2\bar{\sigma}$ which is also the normal to the yield surface, i.e., $r = F_{\sigma}$. In deviatoric stress space, the von Mises yield surface is circular.
Table 4.1: Backward Euler return mapping algorithm

1. Initialization: set initial values of plastic strain and internal variables to converged values at end of previous time-step, zero the increment in plasticity parameter and evaluate the elastic trial stress:

\[
k = 0, \quad \varepsilon_p^{(0)} = \varepsilon_p^0, \quad \kappa^{(0)} = \kappa, \quad \Delta \lambda^{(0)} = 0, \quad \sigma^{(0)} = C_e : (\varepsilon_{n+1} - \varepsilon_p^{(0)})
\]

2. Check yield condition and convergence at \( k \)th iterations:

\[
F^{(k)} = F(\Delta \sigma^{(k)}, \Delta \kappa^{(k)}), \quad \left\{ \tilde{\alpha}^{(k)} \right\} = \left\{ \alpha^{(k)} \right\} \right. \]

If: \( F^{(k)} < TOL_1 \) and \( \| \tilde{\alpha}^{(k)} \| < TOL_2 \), converged.
Else: go to 3.

3. Compute increment in plasticity parameter:

\[
\delta \lambda^{(k)} = \frac{F^{(k)} - \partial F^{(k)} A^{(k)} \tilde{\alpha}^{(k)}}{\partial F^{(k)} A^{(k)} \tilde{\alpha}^{(k)}}
\]

4. Obtain increments in stress and internal variables:

\[
\left\{ \Delta \sigma^{(k)} \Delta \kappa^{(k)} \right\} = - \left[ A^{(k)} \right] \left\{ \tilde{\alpha}^{(k)} \right\} - \delta \lambda^{(k)} \left[ A^{(k)} \right] \left\{ \tilde{r}^{(k)} \right\}
\]

5. Update plastic strain and internal variables:

\[
\varepsilon_p^{(k+1)} = \varepsilon_p^{(k)} + \Delta \varepsilon_p^{(k)} = \varepsilon_p^{(k)} - C_e^{-1} : \Delta \sigma^{(k)}
\]

\[
\kappa^{(k+1)} = \kappa^{(k)} + \Delta \kappa^{(k)}
\]

\[
\Delta \lambda^{(k+1)} = \Delta \lambda^{(k)} + \delta \lambda^{(k)}
\]

\[
\sigma^{(k+1)} = \sigma^{(k)} + \Delta \sigma^{(k)} = C_e : (\varepsilon_{n+1} - \varepsilon_p^{(k+1)})
\]

\[
k < -k + 1, \quad go \ to \ 2
\]

and therefore the normal to the yield surface is radial (see Figure 4-2). We define a unit normal vector in the (radial) direction of plastic flow as

\[
\hat{n} = \frac{r^{(0)}}{\| r^{(0)} \|} = \frac{\sigma_{\text{dev}}^{(0)}}{\| \sigma_{\text{dev}}^{(0)} \|}
\]

\[
r^{(0)} = \sqrt{3/2} \hat{n}
\]
The key feature of the radial return method is that \( \hat{n} \) remains radial and unchanged throughout the plastic-corrector phase of the algorithm. Referring to (4.9), the update of the plastic strain is therefore a linear function of \( \Delta \lambda \) and the plastic flow residual is identically zero: \( a^{(k)} = 0 \). The only internal variable (isotropic hardening) is the accumulated plastic strain, given by \( \kappa \equiv \bar{\varepsilon} = \lambda \) with \( h = 1 \). Therefore the update of the internal variable is also a linear function of \( \Delta \lambda \) and the corresponding residual is zero, i.e., \( b^{(k)} = 0 \).

By differentiating the expression for the plastic flow direction with respect to stress we obtain the result

\[
\hat{r}_\sigma = \frac{3}{2\bar{\sigma}} \hat{I}, \quad \hat{I} = I^{\text{dev}} - \hat{n} \otimes \hat{n}, \quad I^{\text{dev}} = I - \frac{1}{3} I \otimes I
\]

(4.16)

where \( I^{\text{dev}} \) is the fourth-order symmetric deviatoric tensor and the projection tensor \( \hat{I} \) has the properties

\[
\hat{I} \hat{n} = \hat{I} \hat{n}, \quad \hat{I} : \hat{n} = 0, \quad \hat{I} : I = 0, \quad \hat{I} : I^{\text{dev}} = \hat{I}
\]

(4.17)

The plastic flow direction is independent of the accumulated plastic strain and therefore \( r_\kappa = 0 \). Also, since \( h = 1 \), \( h_\sigma = 0 \) and \( h_\kappa = 0 \). The yield condition is given in Chapter 2 by \( F = \bar{\sigma} - \sigma_Y(\bar{\varepsilon}) = 0 \) and the derivatives of \( F \) are \( F_\sigma = r \) and \( F_\kappa = -d\sigma_Y/d\bar{\varepsilon} = -H \). The matrix \( A \) is therefore written as

\[
[A^{(k)}] = \begin{bmatrix}
(C_e^{-1} + \Delta \lambda r_\sigma)^{-1} & 0 \\
0 & -I
\end{bmatrix}^{(k)}
\]

(4.18)
Now note that
\[(C_ε^{-1} + Δλr_ε = (C_ε^{-1} + αI), \quad α = 3Δλ/2\bar{σ})\] (4.19)

For isotropic elastic moduli and using (4.19), the inverse can be written as
\[(C_ε^{-1} + Δλr_ε)^{-1} = \left( C_ε - 2μb\bar{I} \right), \quad b = \frac{2μα}{1 + 2μα}\] (4.20)

and A is
\[
A^{(k)} = \begin{bmatrix}
(C_ε - 2μb\bar{I}) & 0 \\
0 & -I
\end{bmatrix}
\] (4.21)

For isotropic elastic moduli, we have the identities \(C_ε : r = 2μr = 2μ√3/2\hat{n}\), \(I : (C_ε : r) = 0\) and \(r : C_ε : r = 3μ\). Using these identities and the expression (4.21) for A, and recalling that \(\bar{a}^{(k)} = 0\) (because \(a^{(k)} = b^{(k)} = 0\)), the increment in the plasticity parameter (4.13) is given by
\[\delta λ^{(k)} = \frac{F^{(k)}}{3μ + H^{(k)}}\] (4.22)

To obtain an alternative expression, we note that the deviatoric stress can be written as \(σ^{dev} = \sqrt{3/2}\bar{σ}\hat{n}\) and therefore, from (4.15)
\[σ_0^{(k)} = σ_0^{(0)} - 2μΔλ^{(k)}\bar{r}^{(k)} = \left( \sqrt{2/3}\bar{σ}^{(0)} - 2μΔλ^{(k)}\sqrt{3/2} \right)\hat{n}\] (4.23)

Using this expression, the effective stress is
\[\bar{σ}^{(k)} = \bar{σ}^{(0)} - 3μΔλ^{(k)}\] (4.24)

which is substituted into the yield function \(F^{(k)}\) in (4.22) to give the following expression for the increment in plasticity parameter:
\[\delta λ^{(k)} = \frac{(\bar{σ}^{(0)} - 3μΔλ^{(k)}) - σ_Y (\bar{ε}^{(k)})}{3μ + H^{(k)}}\] (4.25)

The radial return method is summarized in Table 4.2.

### 4.2 Explicit integration schemes

Compared with implicit methods, explicit methods can be used to implement a general integration for any elastic-plastic models as it only needs first derivative of the yield
Table 4.2: Radial return algorithm

1. Initialization:
\[ k = 0, \quad \varepsilon^{p(0)} = \varepsilon_n, \quad \bar{\varepsilon}^{(0)} = \bar{\varepsilon}_n, \quad \Delta \lambda^{(0)} = 0, \quad \sigma^{(0)} = C_e : (\varepsilon_{n+1} - \varepsilon^{p(0)}) \]

2. Check yield condition at \( k \)th iterations:
\[ F^{(k)} = \sigma^{(k)} - \sigma_Y (\bar{\varepsilon}^{(k)}) = (\bar{\sigma}^{(0)} - 3\mu\Delta \lambda^{(k)}) - \sigma_Y (\bar{\varepsilon}^{(k)}) \]
   If: \( F^{(k)} < TOL \) then converged
   Else: go to 3.

3. Compute increment in plasticity scalar:
\[ \lambda^{(k)} = \frac{(\bar{\sigma}^{(0)} - 3\mu\Delta \lambda^{(k)}) - \sigma_Y (\bar{\varepsilon}^{(k)})}{3\mu + H^{(k)}} \]

4. Update plastic strain and internal variables:
\[ \bar{n} = \sigma^{(0)}_{\text{dev}} / \| \sigma^{(0)}_{\text{dev}} \|, \quad \Delta \varepsilon^{(k)} = -\delta^{(k)} \sqrt{\frac{3}{2}} \bar{n}, \quad \Delta \bar{\varepsilon}^{(k)} = \delta \lambda^{(k)} \]
\[ \varepsilon^{p(k+1)} = \varepsilon^{p(k)} + \Delta \varepsilon^{(k)} \]
\[ \sigma^{(k+1)} = C_e : (\varepsilon_{n+1} - \varepsilon^{p(0)}) = \sigma^{(k)} + \Delta \sigma^{(k)} = \sigma^{(k)} - 2\mu \delta \lambda^{(k)} \sqrt{\frac{3}{2}} \bar{n} \]
\[ \bar{\varepsilon}^{(k+1)} = \bar{\varepsilon}^{(k)} + \delta \bar{\varepsilon}^{(k)} \]
\[ \Delta \lambda^{(k+1)} = \Delta \lambda^{(k)} + \delta \lambda^{(k)} \]

\[ k \leftarrow k + 1, \text{ go to 2} \]

function and plastic potential. However, to achieve a better accuracy of the integration, the strain increment that constitutes an elastic-plastic response is usually divided into a number of substeps. Since the determination of the number of substeps are completely based on trial and error and there is no error control in the whole integration process, some forms of the stress correction are often used to return the stress state to the yield surface. For this purpose, Wissmann and Hauck[12] and Sloan[39] suggested a substepping scheme with error control to limit the error accumulation in the computed stresses. However, as shown in the next Section, such schemes only provide the desirable accuracy when used in conjunction with the stress correction. For all the explicit schemes, the initial yielding point is needed for the integration process to start with. In the following, the methods for restoring the stress states to the yield surface and the
4.2. EXPLICIT INTEGRATION SCHEMES

accuracy and determining the intersection point are introduced.

4.2.1 Intersection with the yield surface

As mentioned previously, for analysis of elastic-plastic materials, at any stage during the solution process the stress-strain behaviour at an integration point will either be 'elastic' or 'plastic'. If a point changes from an elastic to a plastic state, it is necessary to determine the portion of the stress increment that causes purely elastic deformation. Assume that the stress state changes from last elastic state \( \sigma^{r-1} \) (point A) to an elastic-plastic state (point B) shown in Figure 4-3. Note that point B represents the state of a trail stress, i.e.,

\[
\sigma_B = \sigma_A + \Delta\sigma^{trail} = \sigma_A + C_e : \Delta\varepsilon
\]

By definition, we have

\[
F(\sigma_A, \kappa) = F_A < 0 \quad (4.26)
\]

where the hardening parameter \( \kappa \) remains constant while deformation takes place within the yield surface. The plastic yielding must occur if

\[
F(\sigma_A + \Delta\sigma^{trail}, \kappa) = F(\sigma_B, \kappa) = F_B \leq 0 \quad (4.27)
\]

In order to determine the portion of the stress increment that lies within the yield surface, we need to find a scalar \( \alpha \) such that

\[
F(\sigma, \kappa) = 0 \quad (4.28)
\]

Figure 4-3: Schematic illustration of the initial yielding state.
where
\[ \sigma = \sigma_A + \alpha \Delta \sigma^{\text{trial}}, \quad 0 < \alpha < 1 \] (4.29)

A simple linear interpolation method is used in [54] for obtaining \( \alpha \), which gives
\[ \alpha_1 = \frac{F_A}{F_A - F_B} \] (4.30)

However, experience suggests that the \( \alpha \) values obtained from linear interpolation, if used without further refinement, are not sufficiently accurate for many plasticity calculations. In general, some form of iteration is necessary if the plastic deformations are large or the yield criterion is highly non-linear.

We may notice that equations (4.28) and (4.29) define a single non-linear equation of the form \( F(\alpha) = 0 \). In his earlier work[39], Sloan introduced two iterative schemes, i.e. the popular Newton-Raphson technique and the secant iteration algorithm, with an aim to obtain more accurate estimate for \( \alpha \). Using the popular Newton-Raphson technique, for example, the stresses and \( \alpha \) are updated using the relations
\[
\begin{align*}
\sigma_k &= \sigma_{k-1} + \alpha_k \Delta \sigma^{\text{trial}} \\
\alpha_{k+1} &= \frac{-F(\sigma_k, \kappa)}{r_k^T \Delta \sigma^{\text{trial}}} 
\end{align*}
\] (4.31)

where \( r_k = (F_\sigma)_k \) is evaluated at the stress \( \sigma_k \). The iterative procedure may be started by assuming \( \sigma_0 = \sigma_A \) and using \( \alpha_1 \) from (4.30). Equation (4.31) forms a standard update for Newton-Raphson iteration which is
\[ \alpha_{k+1} = \alpha_k + \Delta \alpha_{k+1} \] (4.32)

where
\[ \Delta \alpha_{k+1} = \frac{-F(\sigma_k, \kappa)}{F_\sigma} = \frac{-f(\sigma_k, \kappa)}{r_k^T \frac{\partial f}{\partial \sigma}} = \frac{-f(\sigma_k, \kappa)}{r_k^T \Delta \sigma} \] (4.33)

The iteration procedure for determine \( \alpha \) and \( \sigma \) may be terminated when the relative error in the norm of stresses, defined by \( \|\sigma_{k+1} - \sigma_k\|/\|\sigma_k\| \), is less than some specified tolerance.

Another scheme for computing \( \alpha \) is the secant iteration algorithm. This method does not require the evaluation of the gradient vector \( r \). Interested users are advised to refer to [62].

However in his later work[63], Sloan pointed out that the Newton-Raphson and secant methods offer rapid convergence but may diverge in some circumstances because
they do not constrain the solution for $\alpha$ to lie within specified bounds, and suggested
Pegasus method which provides unconditional and quick convergence.

In this research work we employ this new scheme for intersecting the yielding point.
If we set $F_0 = f(\sigma_0 + \alpha_0 \Delta \sigma_{\text{trial}}, \kappa_0)$ and $F_1 = f(\sigma_0 + \alpha_1 \Delta \sigma_{\text{trial}}, \kappa_0)$. The intersection point, i.e. $\alpha$ can be obtained by:

$$\alpha = \alpha_1 - F_1 \times (\alpha_1 - \alpha_0)/(F_1 - F_0) \quad (4.34)$$

The yield function can be updated by

$$F_{\text{new}} = f(\sigma_0 + \alpha \Delta \sigma_{\text{trial}}, \kappa_0) \quad (4.35)$$

This process can be solved iteratively with replacing $\alpha_0$ and $F_0$ with $\alpha$ and $F_{\text{new}}$ until $F_{\text{new}} \leq FTOL$. In the absence of better information, the algorithms above are started by specifying $\alpha_0 = 0$ and $\alpha_1 = 1$.

4.2.2 Correction of the yield surface drift

If the explicit integration schemes are used for updating the stress, the stress state
at the end of the increment may not fulfill the yield criterion. The error will essen­
tially depend on the size of the strain increment and number of subdivisions, but as
the error is cumulative it is important to ensure that the stresses are corrected back
to the yield surface during each increment. Sloan’s original work[39] suggested that
the stress correction process is not required for proposed substepping schemes. But
most researchers, such as Potts and Gens[64] and Crisfield[47] strongly advocate the
importance of correcting back for explicit schemes. Potts also proposed several meth­
ods of projecting back in [64]. In this thesis, an efficient and robust correction methods
proposed in Sloan[63] is used to restore the stresses to the yield surface.

The schematic illustration of the correcting process is shown in Figure 4-4. Assume
that the initial stress state (point A) locates on the yield surface, i.e. $F(\sigma_A, \kappa_A) = 0$
and the new stress state after integration (point B) drifts away from the updated
yield surface $F(\sigma_B, \kappa_B) = 0$. The point C represents the stress state after correction.
Ignoring second order terms and above, $F$ may be expanded in a Taylor series about
this stress point B to give:

$$F = F_B + a_B \delta \sigma + \frac{\partial F}{\partial \kappa} \delta \kappa \quad (4.36)$$
where $\delta\sigma$ is a small stress correction, $\delta\kappa$ is a small hardening parameter correction, $F_B = f(\sigma_B, \kappa_B)$, and $a_B$ is evaluated at $\sigma_B$. In returning the stress state to the yield surface, it is desirable that the total strain increment $\Delta\varepsilon$ remains unchanged, since this is consistent with the philosophy of the displacement finite element procedure. From equation (4.36), we see the imposed strain increments remain unchanged provided that stress correction obeys the relation:

$$
\delta\sigma = -\delta\lambda D_e b_B
$$

(4.37)

where $\delta\lambda$ is an unknown multiplier and $b_0$ is evaluated at $\sigma_B$. Using equation (4.37), the corresponding hardening correction is:

$$
\delta\kappa = \delta\lambda b_B = -\delta\lambda \frac{a_B}{dF/d\kappa}
$$

(4.38)

where $a_B$ and $b_B$ are evaluated at $\sigma_B$. Combining equations (4.36) - (4.38) and setting $F = 0$ gives the unknown multiplier as:

$$
\delta\lambda = F_b/(A_B + a_B^T D_e b_B)
$$

The corrections to the stresses and hardening parameter are thus given by:

$$
\delta\sigma = -F_B D_e b_B/(A_B + a_B^T D_e b_B)
$$

(4.39)

$$
\delta\kappa = F_B b_B/(A_B + a_B^T D_e b_B)
$$

and an improved stress state, which is closer to the yield surface, can be obtained from:

$$
\sigma = \sigma_B + \delta\sigma
$$

(4.40)

$$
\kappa = \kappa_B + \delta\kappa
$$
This type of scheme, which is known as a consistent correction, may be applied repeatedly until \( F(\sigma, \kappa) > FTOL \). Such a procedure is employed for the calculation presented in this thesis.

### 4.2.3 Forward Euler method

Once the stresses at the onset of initial yielding have been computed, the final stress state is needed to determine by integrating the the stress-strain relations over the strain path. The integration of stress-strain relations requires the solution of the initial value problem like

\[
\frac{d\sigma}{dT} = D_{ep}\Delta \varepsilon, \quad T \in [0,1]
\]

in which \( \sigma|_{T=0} \) defines the stress state which already satisfy the yield criterion, and \( \sigma|_{T=1} \) defines the stress at an end of load increment or iteration.

The explicit forward Euler scheme for solving above initial value problem is based on the intersection point in the stress and hardening parameter space for finding all the relevant derivatives and variables. To this end, the stresses are updated by replacing the infinitesimal elastic-plastic stress-strain relation by a finite incremental relation:

\[
\Delta \sigma = D_{ep}(\sigma_0, \kappa_0)\Delta \varepsilon
\]

where \( \sigma_0, \kappa_0 \) and the initial elastic-plastic constitutive matrix are evaluated at the intersection point. Obviously this linear approximation will not be accurate for very large strain increments. Thus the strain increment is usually divided into smaller subincrements ([65]):

\[
\delta \varepsilon = \frac{(1 - \alpha)\Delta \varepsilon}{m}
\]

where \( m \) is the number of the substeps. The changes of the stress and hardening parameter for each substep are determined as:

\[
\delta \sigma_i = D_{ep}(\sigma + \delta \sigma_{i-1}, \kappa + \delta \kappa_{i-1}) \delta \varepsilon
\]

\[
\delta \kappa_i = \delta \lambda(\sigma + \delta \sigma_{i-1}, \kappa + \delta \kappa_{i-1}, \delta \varepsilon) h
\]

where the stress and hardening parameter are updated as the sum of \( \delta \sigma_i \) as the substep
4.2. EXPLICIT INTEGRATION SCHEMES

Table 4.3: Forward Euler scheme with subincrements

1. Set the initial state to be the state on intersection point:
   \[ \sigma = \sigma_0, \quad \kappa = \kappa_0, \quad \Delta \varepsilon \]

2. Determine the size of substep:
   \[ \delta \varepsilon = \frac{\Delta \varepsilon}{m} \]

3. Do at each substep, i.e. \( i = 1, 2, \ldots, m \):
   \[
   \begin{align*}
   \delta \sigma_i &= D_{ep}(\sigma_{i-1}, \kappa_{i-1}) \delta \varepsilon \\
   \delta \kappa_i &= \delta \lambda(\sigma_{i-1}, \kappa_{i-1}, \delta \varepsilon) h \\
   \sigma_i &= \sigma_{i-1} + \delta \sigma_i \\
   \kappa_i &= \kappa_{i-1} + \delta \kappa_i
   \end{align*}
   \]

   Stop incrementation when \( i > m \)

4. Update the stress state
   \[ \sigma = \sigma_i, \quad \kappa = \kappa_i \]

The forward Euler scheme, as illustrated in Table 4.3, is the easiest to implement among all the explicit schemes. However, the stress states after integration tend to drift away from the yield surface if the number of the substeps is not sufficient small. Such a departure of stress from the yield surface is accumulative, and may lead to unacceptable solution for the whole load increment. Moreover, the procedure always uses subincrements of equal size, which turns out to be computationally inefficient as the number of substeps is always determined by trial and error. In sum, the forward Euler has the following disadvantages:

1. The number of substeps required for each integration point is usually determined by an empirical rule. Since these rules are formulated by trial and error, it is difficult to ensure that the elastic-plastic stress-strain relations are integrated with sufficient accuracy for a general type of constitutive law and strain path.

2. There appears to be no theoretical justification for preferring one type of stress...
correction over another. Indeed a number of different corrections may be formulated which lead to the computed stresses satisfying the yield condition approximately. These corrections, however, do not guarantee that the plastic deformation obey the prescribed flow rule.

3. No estimate of the error in the integration process is available.

It is then necessary to estimate the number of sub-increments that are required to produce a solution of specified accuracy. The key idea is to use two integration schemes, whose order of accuracy differs by one. The difference between the higher order solution and lower order solution provides an estimate of the local truncation error for the current step size [66]. These "so-called" substeping schemes will be introduced in details. Based on these algorithms, a new substeping scheme based on high-order Runge-Kutta-Dormand-Prince is developed.

4.2.4 New Runge-Kutta-Dormand-Prince integration scheme

The modified Euler scheme introduced in [39] is only of second-order accuracy. To use the explicit schemes for analysis of complex sheet metal forming process, the integration algorithms with high-order accuracy is always preferred in order to predict accurately various forming defects, such as thinning, wrinkling and springback, etc. Runge-Kutta type schemes[67, 68, 69] are widely used for stress integration purpose due to their high-order of accuracy. Some Runge-Kutta algorithms which have been successfully used in elastic-plastic computations are Runge-Kutta-England scheme[70], Runge-Kutta-Fehlberg scheme[40, 42] and Runge-Kutta-Merson scheme[71]. In this research we develop an integration scheme which is based on Runge-Kutta-Dormand-Prince (RKDP) method, introduced by Dormand and Prince[72]. Runge-Kutta-Dormand-Prince has been found to be superior to all existing Runge-Kutta type schemes in terms of being more efficient for high accuracy work[73, 74]. Runge-Kutta-Dormand-Prince is of fifth-order accuracy. One of its advantages is that high order estimate, \( \hat{\sigma} \), for the solution can be obtained without any extra function evaluations.

Different sets of coefficients for Runge-Kutta-Dormand-Prince scheme have been proposed. In practice, the most commonly-used coefficients for RKDP are shown in Coefficients Set I in Figure 4-5. In this research, we employ coefficient set II due to its
4.2. EXPLICIT INTEGRATION SCHEMES

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Coefficients Set I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/5</td>
<td>1/5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3/10</td>
<td>3/40</td>
<td>3/40</td>
<td></td>
</tr>
<tr>
<td>4/5</td>
<td>3/10</td>
<td>9/10</td>
<td>6/5</td>
</tr>
<tr>
<td>8/9</td>
<td>226</td>
<td>25</td>
<td>880</td>
</tr>
<tr>
<td>1</td>
<td>181</td>
<td>5</td>
<td>-266</td>
</tr>
<tr>
<td>σ</td>
<td>31</td>
<td>540</td>
<td>0</td>
</tr>
<tr>
<td>σ</td>
<td>19</td>
<td>216</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Coefficients Set II</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/5</td>
<td>1/5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3/10</td>
<td>3/40</td>
<td>3/40</td>
<td></td>
</tr>
<tr>
<td>4/5</td>
<td>44/45</td>
<td>9/15</td>
<td>32/9</td>
</tr>
<tr>
<td>8/9</td>
<td>15372</td>
<td>-25360</td>
<td>64448</td>
</tr>
<tr>
<td>1</td>
<td>9017</td>
<td>-355</td>
<td>46732</td>
</tr>
<tr>
<td>σ</td>
<td>55</td>
<td>384</td>
<td>0</td>
</tr>
<tr>
<td>σ</td>
<td>5178</td>
<td>1</td>
<td>7571</td>
</tr>
<tr>
<td>σ</td>
<td>57600</td>
<td>10</td>
<td>16695</td>
</tr>
</tbody>
</table>

Figure 4-5: Coefficients for Runge-Kutta-Dormand-Prince scheme

ability to minimize the local truncation error[75].

Firstly we define a dimensionless time $T$ in the time interval $[t_n, t_{n+1}]$ as

$$T := \frac{t - t_n}{t_{n+1} - t_n}$$  \hspace{1cm} (4.43)

with $T \in [0, 1]$. The strain increment for each substep is given by

$$\Delta e^{(k)} = \Delta T^{(k)} \Delta e$$ \hspace{1cm} (4.44)

where

$$\Delta T^{(k)} = (t^{(k+1)} - t_k)/(t_{n+1} - t_n) \leq 1 \quad \text{and} \quad \sum_{k=1}^{n} \Delta T^{(k)} = 1$$ \hspace{1cm} (4.45)

Extensive research [76, 77, 78] has been carried out with the aim to control the truncation error in each substep. The most effective algorithm is use of solutions obtained with two embedded algorithms of different order of accuracy to extrapolate the substep.
size which can achieve a prescribed accuracy of the solution. In this research, two different numerical solutions of the initial value problem (4.2.3) are obtained simultaneously according to

\[
\tilde{\sigma}^{(k+1)} = \sigma^{(k)} + \Phi_1(\sigma^{(k)}, \Delta \varepsilon, \Delta T^{(k)}) \Delta T^{(k)} \\
\tilde{\sigma}^{(k+1)} = \sigma^{(k)} + \Phi_2(\sigma^{(k)}, \Delta \varepsilon, \Delta T^{(k)}) \Delta T^{(k)}
\]  

(4.46)

The two functions \(\Phi_1\) and \(\Phi_2\) are constructed as follows:

\[
\Phi_1 := \sum_{j=0}^{m} \hat{c}_j F_j(\sigma^{(k)}, \Delta \varepsilon, \Delta T^{(k)}) \\
\Phi_2 := \sum_{j=0}^{m+1} \hat{c}_j F_j(\sigma^{(k)}, \Delta \varepsilon, \Delta T^{(k)})
\]  

(4.47)

where

\[
F_j(\sigma^{(k)}, \Delta \varepsilon, \Delta T^{(k)}) := F \left( \sigma^{(k)} + \Delta T^{(k)} \sum_{l=0}^{j-1} \beta_{jl} F_l \Delta \varepsilon \right)
\]  

(4.48)

with \(k = 0, 1, \ldots, m + 1\). The constants \(\hat{c}_i\) and \(\beta_{ij}\) are evaluated assuming that the functions in (4.46) generate one-step methods of order \(m\) and \(m + 1\), respectively.

Based on the two solutions available, adaptive step size control is accomplished as follows. Consider the following difference tensor:

\[
E^{(k)} = \tilde{\sigma}^{(k)} - \tilde{\sigma}^{(k)}
\]  

(4.49)

and the associated scalar measure of the relative error:

\[
R^{(k)} := \frac{\| P^{(k)} \|}{\| \tilde{\sigma}^{(k)} \|}
\]  

(4.50)

Due to the prescribed order of accuracy of the two embedded algorithms, the following expansion formula holds:

\[
R^{(k)} = (\Delta T^{(k)})^m C^{(k)}(T) + \text{terms of higher order}
\]  

(4.51)

where \(m = 5\) for Runge-Kutta-Dormand-Prince scheme.

The integration over the \(k\)th substep is assumed to be successful when, for a given error tolerance \(TOL\):

\[
R^{(k)} \leq TOL
\]  

(4.52)
4.2. EXPLICIT INTEGRATION SCHEMES

When this is the case, it follows (neglecting terms of higher order)

\[
\frac{\|C^{(k)}\|}{\|\dot{\sigma}^{(k)}\|} (\Delta T^{(k)})^m \leq TOL
\]

Now, in order to meet the prescribed accuracy requirements, the new step size must satisfy the same condition as well:

\[
\frac{\|C^{(k)}\|}{\|\dot{\sigma}^{(k+1)}\|} (\Delta T^{(k)})^m \leq TOL
\]

Combining (4.53) and (4.54), and by making the approximations:

\[
\|C^{(k+1)}\| \simeq \|C^{(k)}\|, \quad \|\dot{\sigma}^{(k+1)}\| \simeq \|\dot{\sigma}^{(k)}\|
\]

the following extrapolation formula is obtained for the new substep size \(\Delta T^{(k+1)}\):

\[
\Delta T^{(k+1)} = \Delta T^{(k)} \left[ \frac{TOL}{R^{(k+1)}} \right]^{1/m}
\]

In the practical implementation of the algorithms, (4.56) is replaced by

\[
\Delta T^{(k+1)} = 0.8 \Delta T^{(k)} \left[ \frac{TOL}{R^{(k+1)}} \right]^{1/m}
\]

which applies a suitable factor of safety to \(\Delta T^{(k+1)}\) in order to account for the approximations introduced in the extrapolation formula (4.56).

If condition (4.52) is not satisfied, i.e. the \(k\)th substep failed, (4.57) can be used to obtain a new, smaller value for the step size \(\Delta T^{*(k+1)}\)

\[
\Delta T^{*(k+1)} = 0.8 \Delta T^{(k)} \left[ \frac{TOL}{R^{(k+1)}} \right]^{1/m}
\]

An upper and a lower bound are also set for each new substep as follows, so that the extrapolation is not carried too far:

\[
\Delta T^{(k+1)}/\Delta T^{(k)} \leq 2, \quad \Delta T^{*(k+1)}/\Delta T^{(k)} \geq 0.1
\]

Following the procedures introduced above, we explicitly express the substepping Runge-Kutta-Dormand-Prince scheme in a form suitable for programming. In Runge-Kutta-Dormand-Prince scheme, the fourth and fifth order algorithms are used to estimate the new stresses and hardening parameter. Using the new coefficients mentioned
previously, we have:

\[
\begin{align*}
\tilde{\sigma}^{(k)} & = \sigma_0 + \frac{35}{384} \Delta\sigma_1 + \frac{500}{113} \Delta\sigma_3 + \frac{125}{192} \Delta\sigma_4 - \frac{2187}{6784} \Delta\sigma_5 + \frac{11}{84} \Delta\sigma_6 \\
\tilde{\kappa}^{(k)} & = \kappa_0 + \frac{35}{384} \Delta\kappa_1 + \frac{500}{113} \Delta\kappa_3 + \frac{125}{192} \Delta\kappa_4 - \frac{2187}{6784} \Delta\kappa_5 + \frac{11}{84} \Delta\kappa_6 \tag{4.60}
\end{align*}
\]

\[
\begin{align*}
\tilde{\sigma}^{(k)} & = \sigma_0 + \frac{5179}{57600} \Delta\sigma_1 + \frac{7571}{16695} \Delta\sigma_3 + \frac{393}{640} \Delta\sigma_4 - \frac{92097}{339200} \Delta\sigma_5 + \frac{187}{2100} \Delta\sigma_6 \\
\tilde{\kappa}^{(k)} & = \kappa_0 + \frac{5179}{57600} \Delta\kappa_1 + \frac{7571}{16695} \Delta\kappa_3 + \frac{393}{640} \Delta\kappa_4 - \frac{92097}{339200} \Delta\kappa_5 + \frac{187}{2100} \Delta\kappa_6
\end{align*}
\]

in which

\[
\begin{align*}
\Delta\sigma_i & = C_{ep}(\sigma_{i-1}, \kappa_{i-1}) \Delta\varepsilon \\
\Delta\kappa_i & = \Delta\lambda(\sigma_{i-1}, \kappa_{i-1}, \Delta\varepsilon) h
\end{align*}
\]

with

\[
\begin{align*}
\sigma_1 & = \sigma_0 + \frac{1}{5} \Delta\sigma_1 \\
\kappa_1 & = \kappa_0 + \frac{1}{5} \Delta\kappa_1 \\
\sigma_2 & = \sigma_0 + \frac{3}{40} \Delta\sigma_1 + \frac{9}{40} \Delta\sigma_2 \\
\kappa_2 & = \kappa_0 + \frac{3}{40} \Delta\kappa_1 + \frac{9}{40} \Delta\kappa_2 \\
\sigma_3 & = \sigma_0 + \frac{44}{45} \Delta\sigma_1 - \frac{56}{15} \Delta\sigma_2 + \frac{32}{9} \Delta\sigma_3 \\
\kappa_3 & = \kappa_0 + \frac{44}{45} \Delta\kappa_1 - \frac{56}{15} \Delta\kappa_2 + \frac{32}{9} \Delta\kappa_3 \\
\sigma_4 & = \sigma_0 + \frac{19372}{6561} \Delta\sigma_1 - \frac{25360}{2187} \Delta\sigma_2 + \frac{64448}{6561} \Delta\sigma_3 - \frac{212}{729} \Delta\sigma_4 \\
\kappa_4 & = \kappa_0 + \frac{19372}{6561} \Delta\kappa_1 - \frac{25360}{2187} \Delta\kappa_2 + \frac{64448}{6561} \Delta\kappa_3 - \frac{212}{729} \Delta\kappa_4 \\
\sigma_5 & = \sigma_0 + \frac{9017}{3168} \Delta\sigma_1 - \frac{355}{33} \Delta\sigma_2 - \frac{46732}{5247} \Delta\sigma_3 + \frac{49}{176} \Delta\sigma_4 - \frac{5103}{18656} \Delta\sigma_5 \\
\kappa_5 & = \kappa_0 + \frac{9017}{3168} \Delta\kappa_1 - \frac{355}{33} \Delta\kappa_2 - \frac{46732}{5247} \Delta\kappa_3 + \frac{49}{176} \Delta\kappa_4 - \frac{5103}{18656} \Delta\kappa_5
\end{align*}
\]

For simplicity, the elastic-plastic moduli is expressed as $C_{ep}$. Even though the integration process requires six evaluations of the elastic-plastic constitutive matrix, the scheme rapidly becomes competitive with modified Euler scheme as the error tolerance is tightened. As for the modified Euler scheme the estimated relative error can be expressed as:

\[
R^{(k)} = \frac{\|\hat{\sigma}^{(k)} - \tilde{\sigma}^{(k)}\|}{\|\hat{\sigma}^{(k)}\|} \tag{4.62}
\]
4.2. EXPLICIT INTEGRATION SCHEMES

As the local truncation error in the fourth order formula is $O(\Delta T^5)$ the factor that controls the size of the next dimensionless time step $\Delta T$ is given by:

$$\Delta T^{(k+1)} = \Delta T^{(k)} \left( \frac{TOL}{R^{(k)}} \right)^{1/5}$$

(4.63)

but the constrains still apply, which means we actually implemented

$$\Delta T^{(k+1)} = \Delta T^{(k)} \cdot min \left\{ 2, max \left\{ 0.1, 0.8 \sqrt[3]{TOL}/R^{(k)} \right\} \right\}$$

(4.64)

The Runge-Kutta-Dormand-Prince algorithm, which incorporates error control and a variable step size for each integration point, may be summarized as in Table 4.4. At the beginning, the substep size is initialized to $\Delta T^{(k)} = 1$. The two approximate solutions for $\sigma^{(k)}$ and $\kappa^{(k)}$ are then obtained according to (4.61), and the relative error measure is compared with the prescribed error tolerance. If $R^{(k)} \leq TOL$ the stress is updated and the size of the next substep is computed through the extrapolation formual (4.64). Before moving to a new substep, a check is performed in order to prevent that integration is carried over beyond $T = 1$. Instead, if $R^{(k)} \geq TOL$, (4.64) is used again to compute a new, reduced value of $\Delta T^{(k)}$, subject to the limitation, and the substep is repeated until the accuracy requirement is met. The process stops when $\Delta T^{(k)} = 1$.

4.2.5 Modified Euler method

The modified Euler scheme with active error control was introduced in Sloan[39] with the aims to reduce yield surface drift and computational costs of the forward Euler scheme. Since the new RKDP will be compared with both implicit Euler scheme and explicit low-order modified Euler scheme, a brief introduction of this method is given in this chapter.

Compared with the new Runge-Kutta-Dormand-Prince scheme developed above, this scheme uses the difference between the solutions obtained with two simple Euler algorithms. The first estimate of the updated stress and the hardening parameter at the end of the substep is given by:

$$\bar{\sigma} = \sigma_0 + \delta \sigma_1$$

$$\bar{\kappa} = \kappa_0 + \delta \kappa_1$$

(4.65)
4.2. EXPLICIT INTEGRATION SCHEMES

Table 4.4: Runge-Kutta-Dormand-Prince scheme with subincrements

1. Determine the initial state for $\sigma$, $\kappa$, substep counter $k$, time $T$ and its increment $\Delta T$:

   $\sigma = \sigma_0, \quad \kappa = \kappa_0, \quad k = 10, \quad T = 0, \quad \Delta T = 1$

2. While $T \leq 1$, perform 3 to 10. Otherwise go to 11.

3. Determine the substep size:

   $$\Delta \epsilon^{(k)} = \Delta T \Delta \epsilon$$

4. Calculate $\Delta \sigma_i$ and $\Delta \kappa_i$ for $i$ from 1 to 6 according to:

   $$\delta \sigma_i = C_{ep}(\sigma_{i-1}, \kappa_{i-1}) \Delta \epsilon^{(k)}$$
   $$\delta \kappa_i = \Delta \lambda(\sigma_{i-1}, \kappa_{i-1}) h$$

5. Compute approximate solutions for $\dot{\sigma}^{(k)}$ and $\dot{\kappa}^{(k)}$:

   $$\dot{\sigma}^{(k)} = \dot{\sigma}^{(k-1)} + \frac{35}{384} \Delta \sigma_1 + \frac{50}{113} \Delta \sigma_3 - \frac{125}{192} \Delta \sigma_4 + \frac{2187}{6784} \Delta \sigma_5 + \frac{11}{84} \Delta \sigma_6$$
   $$\dot{\kappa}^{(k)} = \dot{\kappa}^{(k-1)} + \frac{35}{384} \Delta \kappa_1 + \frac{50}{113} \Delta \kappa_3 - \frac{125}{192} \Delta \kappa_4 + \frac{2187}{6784} \Delta \kappa_5 + \frac{11}{84} \Delta \kappa_6$$

6. Compute the estimate of the relative error for the current substep:

   $$R^{(k)} = \frac{\|\dot{\sigma}^{(k)} - \sigma^{(k)}\|}{\|\dot{\sigma}^{(k)}\|}$$

7. Check if the current substep is accepted or rejected:

   $$\text{IF } R^{(k)} > TOL \text{ GO TO 10}$$

8. Update the stress, hardening parameter and integration time:

   $$\sigma^{(k)} = \dot{\sigma}, \quad \kappa^{(k)} = \dot{\kappa}, \quad T = T + \Delta T$$

9. Evaluate next substep size:

   $$\Delta T \leftarrow \Delta T^{(k+1)} = \Delta T^{(k)} \cdot \min \left\{ 0.9 \left[ \frac{TOL}{R^{(k)}} \right]^{\frac{1}{4}}, 2.0 \right\} \text{ GO TO 2}$$

10. If the step was rejected, compute a smaller substep:

    $$\Delta T \leftarrow \Delta T^{(k)} = \Delta T^{(k)} \cdot \max \left\{ 0.9 \left[ \frac{TOL}{R^{(k)}} \right]^{\frac{1}{4}}, 0.1 \right\} \text{ GO TO 2}$$

11. Return the final stress and hardening states:

    $$\sigma = \dot{\sigma}^{(k)}, \quad \kappa = \dot{\kappa}^{(k)}$$

where

$$\delta \sigma_1 = C_{ep}(\sigma_0, \kappa_0) \delta \epsilon^{(k)}$$
$$\delta \kappa_1 = \delta \lambda(\sigma_0, \kappa_0, \delta \epsilon^{(k)}) h$$
4.2. EXPLICIT INTEGRATION SCHEMES

The second estimate of the updated stress is given by using second-order Euler algorithm:

\[
\hat{\sigma} = \sigma_0 + \frac{1}{2}(\delta\sigma_1 + \delta\sigma_2) \\
\hat{\kappa} = \kappa_0 + \frac{1}{2}(\delta\kappa_1 + \delta\kappa_2)
\]

where

\[
\delta\sigma_2 = C_{ep}(\sigma_0 + \delta\sigma_1, \kappa_0 + \delta\kappa_1)\delta\varepsilon^{(k)} \\
\delta\kappa_2 = \delta\lambda(\sigma_0 + \delta\sigma_1, \kappa_0 + \delta\kappa_1, \delta\varepsilon^{(k)})h
\]

The relative error for this substep is defined by:

\[
R^{(k)} = \frac{\|\hat{\sigma} - \tilde{\sigma}\|}{\|\hat{\sigma}\|}
\]

Then \(R\) is compared with some prescribed tolerance \(TOL\) and the step is accepted if \(R \leq TOL\), and rejected otherwise. Then we use \(R\) to make an estimate for the asymptotically optimal substep size:

\[
\Delta T^{(k+1)} = \Delta T^{(k)}\sqrt{TOL/R}
\]

In case of rejection \(\Delta T^{(k+1)}\) is used instead of \(\Delta T^{(k)}\); in case of acceptance we use \(\Delta T^{(k+1)}\) to continue the integration. The constrains for limiting the size of next substep still apply, which serves to prevent an abrupt change in the substep size. In our subroutine, we actually implemented

\[
\Delta T^{(k+1)} = \Delta T^{(k)} \cdot \min \left\{ 2, \max \left\{ 0.1, 0.8\sqrt{TOL/R^{(k)}} \right\} \right\}
\]

The Modified Euler algorithm, which incorporates error control and a variable step size for each integration point, may be summarized as in Table 4.5. For a single substep, the modified Euler scheme and Runge-Kutta-Dormand-Prince scheme require two and six evaluations of \(C_{ep}\), respectively. Since its high-order, the new Runge-Kutta-Dormand-Prince scheme will be chosen for implementing the ‘ideal’ run.
4.2. EXPLICIT INTEGRATION SCHEMES

Table 4.5: Modified Euler scheme with subincrements

1. Determine the initial state for \( \sigma, \kappa \), substep counter \( k \), time \( T \) and its increment \( \Delta T \):
   \[
   \sigma = \sigma_0, \quad \kappa = \kappa_0, \quad k = 1, \quad T = 0, \quad \Delta T = 1
   \]

2. While \( T \leq 1 \), perform 3 to 10. Otherwise go to 11.

3. Determine the substep size:
   \[
   \Delta \varepsilon^{(k)} = \Delta T \Delta \varepsilon
   \]

4. Calculate \( \Delta \sigma_i \) and \( \Delta \kappa_i \) for \( i \) from 1 to 2 according to:
   \[
   \Delta \sigma_i = D_{ep}(\sigma_i, \kappa_i) \Delta \varepsilon^{(k)} \\
   \Delta \kappa_i = \Delta \lambda(\sigma_i, \kappa_i) B
   \]

5. Compute approximate solutions for \( \sigma^{(k)} \) and \( \kappa^{(k)} \):
   \[
   \sigma^{(k)} = \sigma^{(k-1)} + \Delta \sigma_1 \\
   \kappa^{(k)} = \sigma^{(k-1)} + \Delta \kappa_1 \\
   \hat{\sigma}^{(k)} = \sigma^{(k-1)} + \frac{1}{2}(\Delta \sigma_1 + \Delta \sigma_2) \\
   \hat{\kappa}^{(k)} = \kappa^{(k-1)} + \frac{1}{2}(\Delta \kappa_1 + \Delta \kappa_2)
   \]

6. Compute the estimate of the relative error for the current substep:
   \[
   R^{(k)} = \frac{||\hat{\sigma}^{(k)} - \sigma^{(k)}||}{||\hat{\sigma}^{(k)}||}
   \]

7. Check if the current substep is accepted or rejected:
   \[
   IF \quad R^{(k)} > TOL \quad GO \ TO \ 10
   \]

8. Update the stress, hardening parameter and integration time:
   \[
   \sigma^{(k)} = \hat{\sigma}, \quad \kappa^{(k)} = \hat{\kappa}, \quad T = T + \Delta T
   \]

9. Evaluate next substep size:
   \[
   \Delta T \leftarrow \Delta T_{k+1} = \Delta T^{(k)} \cdot \min \left\{ 0.8 \left[ \frac{TOL}{R^{(k)}} \right]^\frac{1}{2}, 2.0 \right\} \quad GO \ TO \ 2
   \]

10. If the step was rejected, compute a smaller substep:
    \[
    \Delta T \leftarrow \Delta T^{(k)} = \Delta T^{(k)} \cdot \max \left\{ 0.8 \left[ \frac{TOL}{R^{(k)}} \right]^\frac{1}{2}, 0.1 \right\} \quad GO \ TO \ 2
    \]

11. Return the final stress and hardening states:
    \[
    \sigma = \hat{\sigma}^{(k)}, \quad \kappa^{(k)} = \hat{\kappa}^{(k)}
    \]
Chapter 5

Numerical Experiments

Before using the Runge-Kutta-Dormand-Prince scheme to conduct the finite element analysis of sheet metal forming process, we tested the new substepping scheme with the non-linear elastic-plastic finite element analysis of a simple three dimensional cantilever beam. We use all the stress update algorithms introduced in Chapter 4 for a non-linear elastic-plastic analysis. Performance of all the algorithms are compared using more complex hardening and constitutive models. Some issues about the parallel analysis of such non-linear problems are also addressed in this chapter.

5.1 Material Models

Some research works have been done by Ding and Kalyanasundaram et al[79, 80] to test the performances of substepping schemes, such as the second order modified Euler scheme and Gill’s fourth-order Runge-Kutta method. These works provide a valuable insights in relation to the accuracy of these substepping schemes. However the use of "simple" numerical models, i.e. elastic-perfectly plastic or linear hardening models, made such comparisons not applicable for large complex plastic deformation problems. With this in mind, the nonlinear finite element code developed by them has been modified to test new substepping scheme with more complex constitutive model. In summary following modifications are made:

1. New seventh order Runge-Kutta-Dormand-Prince scheme and implicit backward Euler scheme are built into the original code;
2. The previous linear hardening model has been replaced by a complex Swift hardening law;

3. Hill’s anisotropic yield function has been added beside the previous von Mises yield function.

4. A new Pegasus method is used to replace the conventional linear interpolation method with the aim to obtain more accurate estimate of yielding point;

5. New efficient and robust correction scheme is used in new FE code;

6. The convergence criterion for the overall solution has been changed so that the program can converge even under large load increment and serious error caused by explicit schemes.

7. The non-linear elastic-plastic finite element analysis has been conducted with original three dimensional cantilever beam model shown in Fig. (5-1) but with large line pressure load in order to achieve large plastic deformation.

5.1.1 Von Mises Yield Criterion

Our first constitutive model used to test the RKDP scheme is von Mises yield surface and the yield function is set

\[ F(\sigma, \bar{\varepsilon}_p) = \sqrt{3}(J_2')^{1/2} - \bar{\sigma} \]  

(5.1)

Figure 5-1: A typical three dimensional cantilever beam.
5.1. MATERIAL MODELS

with

\[
J_2 = \frac{1}{2} S : S \\
= \frac{1}{6} \left[ (\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 \right] \\
= \frac{1}{2} \left[ \sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 \right] + \sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{xz}^2 
\]

(5.2)

in which \( S \) is the deviatoric stress:

\[
S = \sigma_{ij}' = \sigma_{ij} - \frac{1}{3} \delta_{ij} \sigma_{kk} 
\]

(5.3)

**Accuracy**

Difference of the results obtained from proposed integration schemes can be seen clearly from the effective strain - effective stress curves for the nonlinear analysis of three dimensional cantilever beam shown in Fig. (5-2), we present effective strain - effective stress curve for the nonlinear analysis of three dimensional cantilever beam. With reference to these figures, we can find that the results computed by all the substepping schemes without stress correction are almost the same no matter how tight we set the error tolerance. However, the results are completely different if the stress correction is employed. In fact, without stress correction, both ME and RKDP schemes can not give the satisfactory accuracy when a complex hardening model is used. We also can see that ME scheme gives the worst result without the stress correction, and the backward Euler scheme gives relatively accurate result. It is worth noting that even the Runge-Kutta-Dormand-Prince is nearly insensitive to the error tolerance when no stress correction is applied, although the accuracy is improved as the error tolerance gets tighter. This again proves that, because the approximation nature of the finite element method, yield surface drift may occur with the stresses moving away from the yield surface. This deviation is practically independent of the integration scheme adopted. When the model involves strain (work) hardening where the yield surface is moving with loading increment, the drift is more significant. Since such discrepancies are usually cumulative, it is important to ensure the stresses are corrected back to the current yield surface at each step of the integration. Since we apply the stress correction at the end of each time increment, it does not improve the accuracy significantly when the error caused by the integration scheme is too serious. This is demonstrated by the modified Euler
scheme with tolerance of $10^{-1}$. However, it can make the computed stress to fulfill the consistency condition at the end of load increment and avoid error accumulation and therefore instabilities in the following time increments.

Errors in the computed stresses from different algorithms with different tolerance are listed in Table (5.1). With reference to Table (5.1), it can be seen that, the Backward Euler scheme gives good result which is adequate for the engineering computation, compared with the explicit schemes. All the explicit schemes give incorrect results if no stress correction is applied, although the accuracy can be improved slightly as the tolerance decreases for both the modified Euler scheme and the Runge-Kutta-Dormand-Prince scheme. For the cases in which stress correction is applied for the explicit schemes, all the explicit schemes obtain better results than the return algorithm does if the integration error can be controlled to some extent, i.e. for the modified Euler scheme with tolerance of $10^{-5}$ and the Runge-Kutta-Dormand-Prince scheme with the tolerance of $10^{-3}$ or smaller. In [39], the measure of error which is approximately equal to tolerance was achieved. However, as mentioned previously, the results obtained from Modified Euler scheme and Runge-Kutta-Dormand-Prince scheme are not improved as the error tolerance becomes tighter. This again indicates that the stress correction applied at the end of each load increment does not improve the performance significantly unless the integration error is controlled to some extent.
Table 5.1: Results of errors for BE, ME and RKDP with different tolerance (von Mises model)

<table>
<thead>
<tr>
<th>Method</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>$1.34 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Error tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$10^{-1}$</td>
</tr>
<tr>
<td>ME</td>
<td>$9.54 \times 10^{-1}$</td>
</tr>
<tr>
<td>MEC</td>
<td>$6.64 \times 10^{-2}$</td>
</tr>
<tr>
<td>RKDP</td>
<td>$9.06 \times 10^{-1}$</td>
</tr>
<tr>
<td>RKDPC</td>
<td>$3.28 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

Efficiency

In Table (5.2), we list total number of substeps for overall solution of the Modified Euler scheme and Runge-Kutta-Dormand-Prince scheme. The corresponding CPU time spent on overall solution is listed in Table (5.3). We can see that, the Runge-Kutta-Dormand-Prince scheme generally requires less substeps than the Modified Euler scheme for the fixed value of tolerance. For the tolerance which is from $10^{-3}$ to $10^{-5}$, the Runge-Kutta-Dormand-Prince scheme uses approximately half of number of substeps consumed by the Modified Euler scheme. This confirms that the high-order Runge-Kutta-Dormand-Prince scheme does not need more substeps to obtain the prescribed accuracy. However, since the Runge-Kutta-Dormand-Prince scheme usually needs to compute elastic-plastic matrix six times for each substep, it still consumes more CPU time than the Modified Euler scheme does. It is interesting to see that the use of stress correction not only improves the accuracy, but saves the computational time significantly. It can be seen in Table (5.3) that for both modified Euler scheme and Runge-Kutta-Dormand-Prince scheme, the CPU time is greatly reduced when the stress correction is used. This is probably because the stress states are integrated more accurately when the yield drift is corrected back to yield surface at the end of each load increment, causing no more effort spent on trying to correct the accumulative errors in the following increments. When the stress correction is employed, the Runge-Kutta-Dormand-Prince scheme is better than the modified Euler scheme in terms of its ability to integrate more accurately with
5.1. MATERIAL MODELS

tight tolerance.

5.1.2 Hill's yield criteria

Hill's stress potential $Q$ is a simple extension of the Mises function to allow anisotropic behaviour. This function is

$$Q(\sigma) = \sqrt{F(\sigma_y - \sigma_z)^2 + G(\sigma_z - \sigma_x)^2 + H(\sigma_x - \sigma_y)^2 + 2L\sigma_{yz}^2 + 2M\sigma_{zx}^2 + 2N\sigma_{xy}^2}$$

where $F, G, H, L, M$ and $N$ are material constants obtained by tests of the material in different directions. These constants are given as follows:

$$F = 0.283, \quad G = 0.358, \quad H = 0.642, \quad L = M = N = 1.288$$

Table 5.2: Total substeps needed in the overall solution for ME and RKDP schemes with different tolerance (von Mises model).

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>$10^{-1}$</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>121784</td>
<td>128913</td>
<td>131069</td>
<td>169431</td>
<td>216540</td>
</tr>
<tr>
<td>MEC</td>
<td>118211</td>
<td>122861</td>
<td>126819</td>
<td>158114</td>
<td>197822</td>
</tr>
<tr>
<td>RKDP</td>
<td>115628</td>
<td>116218</td>
<td>116973</td>
<td>121874</td>
<td>128105</td>
</tr>
<tr>
<td>RKDPC</td>
<td>114312</td>
<td>115710</td>
<td>115934</td>
<td>118795</td>
<td>126541</td>
</tr>
</tbody>
</table>

Table 5.3: CPU time (seconds) spent on the overall solution (von Mises model).

<table>
<thead>
<tr>
<th>Method</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>336</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>$10^{-1}$</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>376</td>
<td>375</td>
<td>375</td>
<td>432</td>
<td>558</td>
</tr>
<tr>
<td>MEC</td>
<td>374</td>
<td>374</td>
<td>373</td>
<td>418</td>
<td>469</td>
</tr>
<tr>
<td>RKDP</td>
<td>596</td>
<td>597</td>
<td>614</td>
<td>636</td>
<td>657</td>
</tr>
<tr>
<td>RKDPC</td>
<td>513</td>
<td>508</td>
<td>514</td>
<td>514</td>
<td>526</td>
</tr>
</tbody>
</table>
5.1. MATERIAL MODELS

Accuracy

For demonstration, we present effective strain - effective stress curves in Figure (5-3). The results once more demonstrate that for Hill’s model, all the explicit schemes give even worse results when no stress correction is applied, no matter how tight the tolerance is. However, for such complex constitutive model, some of the models, i.e., ME with tolerance of $10^{-1}$ and $10^{-2}$ and RKDP with tolerance of $10^{-1}$, still produce unsatisfactory results even with the help of the stress correction.

With reference to the errors in the computed stresses from different algorithms shown in Table (7.4), we can see again that the Backward Euler scheme can provide accuracy which is superior to all the explicit schemes in the case that no stress correction is applied. However when the stress correction is employed, some of the models, such as ME ($10^{-4}$, $10^{-5}$) and RKDP ($10^{-4}$, $10^{-5}$) can produce same or even better level of accuracy than the backward Euler does.

Efficiency

We present the total number of substeps for overall solution of the Modified Euler scheme and Runge-Kutta-Dormand-Prince scheme in Table (7.5) and the corresponding CPU time spent on overall solution in Table (7.6). It can be seen that the results we obtained are similar to the ones from von Mises model, i.e., the Runge-Kutta-Dormand-Prince scheme generally requires less substeps than the Modified Euler scheme does for

![Figure 5-3: Effective strain - Effective stress curves for BE, ME, MEC, RKDP, RKDPC (Hill’s model).](image)
5.2 Parallel performance of new Runge-Kutta-Dormand-Prince scheme

In this section we modified the original parallel code to test Runge-Kutta-Dormand-Prince scheme in the parallel analysis of nonlinear problems. Following modifications are made to the parallel code developed for the previous research works:

1. New seventh order Runge-Kutta-Dormand-Prince scheme is used to replace Gill’s fourth order Runge-Kutta method for the new parallel three dimensional elastic-

---

Table 5.4: Results of errors for BE, FE, ME and RKDP with different tolerance (Hill’s model)

<table>
<thead>
<tr>
<th>Method</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>$9.03 \times 10^{-1}$</td>
</tr>
<tr>
<td>MEC</td>
<td>$3.36 \times 10^{-1}$</td>
</tr>
<tr>
<td>RKDP</td>
<td>$5.14 \times 10^{-1}$</td>
</tr>
<tr>
<td>RKDPC</td>
<td>$9.02 \times 10^{-1}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>$10^{-1}$</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>$9.03 \times 10^{-1}$</td>
<td>$5.81 \times 10^{-1}$</td>
<td>$5.81 \times 10^{-1}$</td>
<td>$5.82 \times 10^{-1}$</td>
<td>$5.64 \times 10^{-1}$</td>
</tr>
<tr>
<td>MEC</td>
<td>$3.36 \times 10^{-1}$</td>
<td>$5.34 \times 10^{-1}$</td>
<td>$5.08 \times 10^{-2}$</td>
<td>$5.06 \times 10^{-2}$</td>
<td>$4.34 \times 10^{-2}$</td>
</tr>
<tr>
<td>RKDP</td>
<td>$5.14 \times 10^{-1}$</td>
<td>$5.23 \times 10^{-1}$</td>
<td>$5.12 \times 10^{-1}$</td>
<td>$5.10 \times 10^{-1}$</td>
<td>$4.97 \times 10^{-1}$</td>
</tr>
<tr>
<td>RKDPC</td>
<td>$9.02 \times 10^{-1}$</td>
<td>$5.44 \times 10^{-2}$</td>
<td>$5.38 \times 10^{-2}$</td>
<td>$3.82 \times 10^{-2}$</td>
<td>$7.17 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 5.5: Total substeps needed in the overall solution for ME and RKDP schemes with different tolerance (Hill’s Model)

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>$10^{-1}$</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>127078</td>
<td>132186</td>
<td>135785</td>
<td>156971</td>
<td>227850</td>
</tr>
<tr>
<td>MEC</td>
<td>123147</td>
<td>128792</td>
<td>131079</td>
<td>150082</td>
<td>219174</td>
</tr>
<tr>
<td>RKDP</td>
<td>111453</td>
<td>117654</td>
<td>122198</td>
<td>127653</td>
<td>128431</td>
</tr>
<tr>
<td>RKDPC</td>
<td>112515</td>
<td>115417</td>
<td>119712</td>
<td>125876</td>
<td>127963</td>
</tr>
</tbody>
</table>

the fixed value of tolerance. However it uses more CPU time than the modified Euler does due to the six evaluations of the elastic-plastic matrix for each substep. Obviously this is the cost we have to pay when high level of accuracy is required.
5.2. PARALLEL PERFORMANCE OF NEW RUNGE-KUTTA-DORMAND-PRINCE SCHEME

Table 5.6: CPU time (seconds) spent on the overall solution (Hill's model)

<table>
<thead>
<tr>
<th>Method</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>347</td>
</tr>
<tr>
<td>FE</td>
<td>395</td>
</tr>
<tr>
<td>FEC</td>
<td>382</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>(10^{-1})</th>
<th>(10^{-2})</th>
<th>(10^{-3})</th>
<th>(10^{-4})</th>
<th>(10^{-5})</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>387</td>
<td>380</td>
<td>389</td>
<td>431</td>
<td>572</td>
</tr>
<tr>
<td>MEC</td>
<td>377</td>
<td>368</td>
<td>388</td>
<td>423</td>
<td>476</td>
</tr>
<tr>
<td>RKDP</td>
<td>635</td>
<td>639</td>
<td>648</td>
<td>674</td>
<td>692</td>
</tr>
<tr>
<td>RKDPC</td>
<td>556</td>
<td>553</td>
<td>563</td>
<td>580</td>
<td>618</td>
</tr>
</tbody>
</table>

plastic analysis;

2. The previous linear hardening model has been replaced by a complex Swift hardening law;

3. Hill's anisotropic yield function has been added in addition to the von Mises yield function.

4. A new Pegasus method is used to replace the previous linear interpolation method with the aim to obtain more accurate estimate of yielding point;

5. New efficient and robust correction scheme is used in new parallel FE code;

6. The parallel elastic-plastic finite element analysis has been conducted with both original three dimensional cantilever beam model shown in Fig. (5-1) and new cylindrical shell cap model shown in Fig. (5-4).

In addition to the enhancements mentioned previously, some techniques used in the old research works, i.e., parallel preconditioned conjugate gradient method with parallel MR smoothing, diagonal storage scheme, element numbering method and technique for optimizing interprocessor communication, etc., have been applied to the non-linear elastic-plastic finite element analysis of a cylindrical shell cap and a three dimensional
5.2. PARALLEL PERFORMANCE OF NEW RUNGE-KUTTA-DORMAND-PRINCE SCHEME

Figure 5-4: A cylindrical shell under uniform pressure.

cantilever beam using hill’s anisotropic yield function and von Mises yield function, respectively.

The parallel implementation of the code has been done via MPI ([81]) on APAC SGI Altix 3700 Bx2 cluster at The Australian National University. The cluster consists of 1928 1.6Ghz Itanium2 processors with totally 5.6 Tbytes of memory and 70 Tbytes of local scratch. As usual, the speedup and efficiency are tested using

\[
\text{Speedup} = \frac{\text{Time for solution on 1 processor}}{\text{Time for solution on } p \text{ processors}}
\]

and the efficiency by:

\[
\text{Efficiency} = \frac{\text{Speedup}}{\text{Number of processors}}
\]

5.2.1 Cylindrical shell cap

Our first numerical model is a cylindrical shell cap shown in Fig. 5-4 with the following dimensions: \(L=300\text{mm}, R=500\text{mm}, t=3\text{mm}\) and \(\alpha=45^\circ\). A uniform pressure of 3MPa is applied with the aim to achieve large plastic deformation. The material properties are taken as follows: Young’s modulus \(E = 200\text{GPa}\), Poisson’s ratio \(\nu = 0.3\), yield stress \(\sigma_y = 200\text{MPa}\), strain-hardening exponent \(n = 0.2637\) and \(K = 567.29\text{MPa}\). Hill’s yield function is considered and material constants are taken as follows: \(F = 0.283\), \(G = 0.358\), \(H = 0.642\) and \(L = M = N = 1.288\).

Due to uniform pressure applied on the whole surface, we partition the whole structure along \(x\) direction to achieve the better load balance. We increased the problem size by refining the finite element mesh and the models of different problem sizes were tested.
5.2. PARALLEL PERFORMANCE OF NEW RUNGE-KUTTA-DORMAND-PRINCE SCHEME

on different number of processors. Table 5.7 lists the performance of different problem sizes on different number of processors. It can be seen that the speed-up and the efficiency for different problem size are adequately good, even for the smaller problem size

<table>
<thead>
<tr>
<th>Problem size (elements)</th>
<th>Number of elements of interface substructure</th>
<th>Number of substructure</th>
<th>Total elapsed time (sec)</th>
<th>Speed-up (%)</th>
<th>Efficiency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5760</td>
<td>1</td>
<td>5760</td>
<td>49</td>
<td>389.12</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2880</td>
<td>49</td>
<td>200.58</td>
<td>1.94</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1920</td>
<td>49</td>
<td>136.53</td>
<td>2.85</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1440</td>
<td>49</td>
<td>103.49</td>
<td>3.76</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1152</td>
<td>49</td>
<td>85.52</td>
<td>4.55</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>960</td>
<td>49</td>
<td>73.30</td>
<td>5.28</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>720</td>
<td>49</td>
<td>56.23</td>
<td>6.92</td>
</tr>
<tr>
<td>11520</td>
<td>1</td>
<td>11520</td>
<td>97</td>
<td>3112.96</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5760</td>
<td>97</td>
<td>1588.24</td>
<td>1.96</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3840</td>
<td>97</td>
<td>1080.89</td>
<td>2.88</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2880</td>
<td>97</td>
<td>810.67</td>
<td>3.84</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2304</td>
<td>97</td>
<td>669.45</td>
<td>4.65</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1920</td>
<td>97</td>
<td>563.94</td>
<td>5.52</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1440</td>
<td>97</td>
<td>437.21</td>
<td>7.12</td>
</tr>
<tr>
<td>23040</td>
<td>1</td>
<td>23040</td>
<td>193</td>
<td>14728.19</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>11520</td>
<td>193</td>
<td>7438.48</td>
<td>1.98</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7680</td>
<td>193</td>
<td>4958.97</td>
<td>2.97</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5760</td>
<td>193</td>
<td>3757.19</td>
<td>3.92</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4608</td>
<td>193</td>
<td>3005.75</td>
<td>4.90</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3840</td>
<td>193</td>
<td>2530.62</td>
<td>5.82</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2880</td>
<td>193</td>
<td>1897.96</td>
<td>7.76</td>
</tr>
</tbody>
</table>
of 5760 elements. For the problem of 23040 elements we nearly achieved linear speed-up using no more than eight processors. This good performance can be attributed to: a) Substepping scheme can integrate the strain-stress relations more accurately and thus uses less iterations to converge, making the solution more efficient; b) Diagonal preconditioner increases the convergence speed of CG algorithm; c) The diagonal storage scheme saves the time wasted on zero fill-ins; d) More efficient interprocessor communication is implemented through using proposed communication technique. For this model it has been observed that the overall computational performance of all the parallel algorithms improves with increase in the problem size as expected.

5.2.2 Cantilever beam model

A typical three dimensional cantilever beam shown in Fig. (5-1) is considered as a second numerical example. The dimensions are L=300mm and a=40mm, respectively. A uniform line pressure P of 450N/mm is used. The material properties are taken as follows: Young’s modulus $E = 200\text{GPa}$, Poisson’s ratio $\nu = 0.3$, yield stress $\sigma_y = 250\text{MPa}$. Again the Swift hardening law is employed with strain-hardening exponent $n = 0.2637$ and constant $K = 567.29\text{MPa}$.

Figure 5-5: A vertical strip-wise partitioning on 4 processors

Figure 5-6: A horizontal strip-wise partitioning on 4 processors
We use the von Mises yield criteria and two partitioning schemes for this model, i.e. vertical and horizontal partitioning methods, shown in Figure 5-5 and 5-6. Tables 5.8 and 5.9 list the performances of different problem sizes on different number of processors by using vertical partitioning scheme. Tables 5.10 and 5.11 present the performances for the horizontal partitioning scheme. From the results shown in Tables 5.8 and 5.9, we can see that, although vertical partitioning scheme does not distribute the work load evenly into each processor, we still achieve good speedups and efficiencies. For this application, the speedup and efficiency are almost insensitive the change of the number of processors when the problem sizes become larger than 7680 elements. In fact, for the problem with 30720 elements, the speed-up and efficiency are almost linear. This good performance can be again attributed to the proposed substepping scheme and preconditioned conjugate gradient method with MR smoothing. Also other techniques, optimization of interprocessor communication and diagonal storage scheme, are making contributions to these. On the opposite, the horizontal partitioning scheme makes every process involved in the computation of the plastic regions, but it leads to

<table>
<thead>
<tr>
<th>Number of elements of CPUs</th>
<th>Number of interface substructure</th>
<th>Total time on 1 CPU (sec)</th>
<th>Total time on n CPUs (sec)</th>
<th>Speed up (%)</th>
<th>Efficiency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>45</td>
<td>332.68</td>
<td>88.48</td>
<td>3.76</td>
<td>94</td>
</tr>
<tr>
<td>960</td>
<td>81</td>
<td>692.06</td>
<td>178.36</td>
<td>3.88</td>
<td>97</td>
</tr>
<tr>
<td>1920</td>
<td>81</td>
<td>1689.52</td>
<td>435.44</td>
<td>3.88</td>
<td>97</td>
</tr>
<tr>
<td>3840</td>
<td>153</td>
<td>5728.50</td>
<td>1446.59</td>
<td>3.96</td>
<td>99</td>
</tr>
<tr>
<td>7680</td>
<td>289</td>
<td>28011.15</td>
<td>7073.52</td>
<td>3.96</td>
<td>99</td>
</tr>
<tr>
<td>6</td>
<td>45</td>
<td>332.68</td>
<td>61.60</td>
<td>5.40</td>
<td>90</td>
</tr>
<tr>
<td>640</td>
<td>81</td>
<td>692.06</td>
<td>122.71</td>
<td>5.64</td>
<td>94</td>
</tr>
<tr>
<td>1280</td>
<td>81</td>
<td>1689.52</td>
<td>296.41</td>
<td>5.70</td>
<td>95</td>
</tr>
<tr>
<td>2560</td>
<td>153</td>
<td>5728.50</td>
<td>984.28</td>
<td>5.82</td>
<td>97</td>
</tr>
<tr>
<td>5120</td>
<td>289</td>
<td>28011.15</td>
<td>4763.80</td>
<td>5.88</td>
<td>98</td>
</tr>
</tbody>
</table>
5.2. PARALLEL PERFORMANCE OF NEW RUNGE-KUTTA-DORMAND-PRINCE SCHEME

Table 5.9: Performance of vertical partitioning on different number of processors

<table>
<thead>
<tr>
<th>Problem size (elements)</th>
<th>No. of elements in each substructure</th>
<th>Number of nodes in each substructure</th>
<th>Total elapsed time (sec)</th>
<th>Speed</th>
<th>Efficiency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7680</td>
<td>1</td>
<td>81</td>
<td>1689.52</td>
<td>1.00</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>81</td>
<td>898.20</td>
<td>1.98</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>81</td>
<td>611.15</td>
<td>2.91</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>81</td>
<td>435.44</td>
<td>3.88</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>81</td>
<td>378.39</td>
<td>4.70</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>81</td>
<td>296.41</td>
<td>5.70</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>81</td>
<td>296.41</td>
<td>7.20</td>
<td>90</td>
</tr>
<tr>
<td>30720</td>
<td>1</td>
<td>289</td>
<td>28011.15</td>
<td>1.00</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>289</td>
<td>14147.05</td>
<td>1.98</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>289</td>
<td>9463.23</td>
<td>2.96</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>289</td>
<td>7073.52</td>
<td>3.96</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>289</td>
<td>5681.77</td>
<td>4.93</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>289</td>
<td>4763.80</td>
<td>5.88</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>289</td>
<td>3656.81</td>
<td>7.66</td>
<td>96</td>
</tr>
</tbody>
</table>

Worse results. It can be seen from Tables 5.10 and 5.11 that the performance could not be improved when increasing the problem size. This is due to more communication involved in the overall analysis of large problems. For example, for the problem which has 3840 elements, the shared degrees of freedom are 549 \times 3. However, for the problem which has 30720 elements, the shared degrees of freedom become 2057 \times 3. In this case, for large problem, although considerable amount of time is spent on computation, significant time is also spent on interprocessor communication. The solution of equations, known as the most time-consuming part in the finite element analysis, requires considerable communication. Thus significant time is spent on this phase as the number of shared nodes becomes larger. This is the reason that the speedup and efficiency generally deteriorate when increasing the problem size. The speedup and efficiency for
5.2. PARALLEL PERFORMANCE OF NEW RUNGE-KUTTA-DORMAND-PRINCE SCHEME

Table 5.10: Performance of horizontal partitioning on fixed number of processors

<table>
<thead>
<tr>
<th>No. of CPUs</th>
<th>Number of elements</th>
<th>Number of interface nodes in each substructure</th>
<th>Total time on 1 CPU (sec)</th>
<th>Total time on n CPUs (sec)</th>
<th>Speed up</th>
<th>Efficiency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>480</td>
<td>305</td>
<td>318.64</td>
<td>88.51</td>
<td>3.60</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>960</td>
<td>549</td>
<td>660.35</td>
<td>187.60</td>
<td>3.52</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>1920</td>
<td>1089</td>
<td>1457.73</td>
<td>428.74</td>
<td>3.40</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>3840</td>
<td>1089</td>
<td>4897.86</td>
<td>1302.62</td>
<td>3.76</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>7680</td>
<td>2057</td>
<td>21148.76</td>
<td>8392.37</td>
<td>2.52</td>
<td>63</td>
</tr>
<tr>
<td>8</td>
<td>240</td>
<td>305</td>
<td>318.64</td>
<td>53.11</td>
<td>6.00</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>480</td>
<td>549</td>
<td>660.35</td>
<td>107.20</td>
<td>6.16</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>960</td>
<td>1089</td>
<td>1457.73</td>
<td>253.08</td>
<td>5.76</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>1920</td>
<td>1089</td>
<td>4897.86</td>
<td>703.72</td>
<td>6.96</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>3840</td>
<td>2057</td>
<td>21148.76</td>
<td>4557.92</td>
<td>4.64</td>
<td>58</td>
</tr>
</tbody>
</table>

The problem with 15360 elements approve it. For this model, 7680-element problem size and 15360-element problem size involve same interface nodes and the speedup and efficiency of 15360-element problem size are much better than 7680-element problem. It again shows that, although idle time caused by poor load balancing degrades performance quickly, if the parallel analysis involves too much communication by using such ‘perfect’ partitioning scheme, the advantage of load balancing caused by such partitioning may be offset by involving too much communication.

It should be noted that, both speedup and efficiency are slightly oscillatory. This is because the cluster used in this research work is a multi-user system. So the elapsed clock time is dependent on the system loading which changes from time to time. When there are several parallel jobs running on the cluster, the speed of communication would certainly slow down.
5.3 Summary

5.3.1 Comparison of performance of different integration schemes

In this chapter, we present a substepping scheme, i.e. Runge-Kutta-Dormand-Prince method, with local error control, enhanced by the stress correction procedure, for analyzing the elastic-plastic problems using both von Mises yield criteria and Hill's Anisotropic yield criteria. Their performances have been compared with the commonly used implicit backward Euler algorithm.

Both explicit substepping schemes control the error in the integration process by permitting the size of each substep to vary in accordance with the behaviour of the constitutive law. The numerical examples indicated that the accuracy of all the substepping schemes can not be improved significantly as the tolerance decreases. Some forms of stress correction should thus be employed in order to ensure that the computed stresses remain on the yield surface at any time and maintain adequate accuracy of the overall solution. It is also has been found that, applying the stress correction at the end of integration does not increase the computational time significantly, but it can improve the accuracy if the errors during the integration can be reduced to some
extent. In summary, the experience suggests that

1. Neither of the explicit schemes with error control can give accurate result without the application of the stress correction.

2. High order Runge-Kutta-Dormand-Prince scheme is superior to both the backward Euler scheme and the modified forward Euler scheme when high level of accuracy is required.

3. For the two constitutive models used in this chapter, Runge-Kutta-Dormand-Prince scheme in combination with the stress correction can provide same or even better level of accuracy as the backward Euler scheme does.

4. To maintain adequate accuracy, the tolerance of $10^{-3}$ or under is required for the Runge-Kutta-Dormand-Prince scheme.

5.3.2 Parallel analysis

The new Runge-Kutta-Dormand-Prince, which controls the error in the integration process to within the vicinity of a specified tolerance, has also been presented for the parallel elastic-plastic analyses, in combination with a parallel preconditioned conjugate gradient algorithm with MR smoothing. This mechanism for controlling the integration process permits the size of each substep to vary in accordance with the behaviour of the constitutive law. In the resulting algorithm for solving the linear system of equations, a parallel substructure preconditioned conjugate gradient method is used. This solution algorithm does not require the assembly of the global system equations. Each processor in the parallel system is assigned a substructure and stores only the information relevant to the substructure that the processor represents. The combination of these two algorithms have been applied to two three dimensional elastic-plastic stress analyses and the good speedup and efficiency have been achieved if the partition scheme is chosen carefully, with the aid of other proposed techniques, such as optimizing interprocessor communication and diagonal storage scheme. The results indicated that, although load balance is important to the parallel finite element analysis, some load balance scheme can degrade the performance if it involves too much communication. In summary, the combination of these two algorithm provides a useful practical means for parallel finite element analysis of large elastic-plastic problems.
Chapter 6

Development of user material subroutine

In order to examine the performance of the algorithm developed in this research work, the stress update scheme has been implemented into ABAQUS user material subroutine VUMAT and used to simulate a typical deep cup drawing process. The use of user subroutine to define the material behaviour requires considerable expertise and the implementation of any realistic constitutive model requires extensive development and testing. Thus a detailed introduction of developing user subroutine for ABAQUS/Explicit is presented in order for other interested users to use the proposed stress update algorithms for their specific analysis. Some important issues for using user subroutine are also discussed.

6.1 ABAQUS Finite Element Analysis Package

6.1.1 Brief introduction to ABAQUS software

ABAQUS is a world’s leading software for advanced finite element analysis. When it was firstly marketed by HKS, Inc. in 1978, this program has had substantial impact in the area of computational mechanics because it was one of the first finite element programs to introduce gateways for researchers to add elements and material models. The ABAQUS software suite has an unsurpassed reputation for technology, quality and reliability and will be used in this research due to its superior capability to nonlinear
engineering problems.

ABAQUS/Standard and ABAQUS/Explicit are two core programs of ABAQUS. Each of these packages offers additional optional modules that address specialized capabilities some FEA professionals may need.

- ABAQUS/Standard provides an analysis technology to solve traditional implicit finite element analyses, such as static, dynamics, thermal, all powered with the widest range of contact and nonlinear material options. ABAQUS/Standard also has optional add-on and interface products which address design sensitivity analysis, offshore engineering, and integration with third party software, e.g., plastic injection molding analysis.

- ABAQUS/Explicit provides ABAQUS analysis technology focused on transient dynamics and quasi-static analyses using an explicit approach appropriate in many applications such as sheet metal forming, crushing and many manufacturing processes.

In the past decade, finite element analysis of sheet metal forming process has been thoroughly studied. Some researchers\cite{82, 83} has successfully applied implicit finite element method. However it has been found that finite element analysis of complex sheet metal forming process using implicit method is extremely time-consuming and the convergence is not always guaranteed. Thus most of the research works in this area was done based on explicit method due to its capacity to deal with transient dynamic problems, and the implicit method is mainly used for springback simulations\cite{84, 85, 86}. Thus the developed user material subroutine is implemented into ABAQUS/Explicit and used to simulate a deep drawing process.

6.2 Implementation of user subroutine

The finite element program ABAQUS provides an interface, whereby any constitutive relation can be added to the material library. To ensure proper functioning ABAQUS defines an interface containing a list of formal arguments for use with user defined material modules. The subroutine containing the constitutive relation must provide information about the material behaviour at the end of each strain increment. The
subroutine is called by ABAQUS/Explicit at each Gauss point for calculation or updating of:

1. Stresses

2. State variables

3. History information, i.e. values of path dependent parameters.

An introduction to the ABAQUS interface, the used syntax and some remarks on the call of user defined material modules are given in the following.

6.2.1 ABAQUS interface

The communication with ABAQUS is accomplished by use of a predefined interface written in FORTRAN code:

```fortran
SUBROUTINE VUMAT(
C Read only (unmodifiable) variables -
  1 nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
  2 stepTime, totalTime, dt, cmname, coordMp, charLength,
  3 props, density, strainInc, relSpinInc,
  4 tempOld, stretchOld, defgradOld, fieldOld,
  5 stressOld, stateOld, enerInternOld, enerInelasOld,
  6 tempNew, stretchNew, defgradNew, fieldNew,
C Write only (modifiable) variables -
  7 stressNew, stateNew, enerInternNew, enerInelasNew)
C
C include 'vaba_param.inc'
C
character*80 cmname
integer charLen
dimension props(nprops), density(nblock), coordMp(nblock,*),
  1 charLength(nblock), strainInc(nblock,ndir+nshr),
  2 relSpinInc(nblock,nshr), tempOld(nblock),
  3 stretchOld(nblock,ndir+nshr),
  4 defgradOld(nblock,ndir+nshr+nshr),
  5 fieldOld(nblock,nfieldv), stressOld(nblock,ndir+nshr),
  6 stateOld(nblock,nstatev), enerInternOld(nblock),
  7 enerInelasOld(nblock), tempNew(nblock),
  8 stretchNew(nblock,ndir+nshr),
  9 defgradNew(nblock,ndir+nshr+nshr),
 0 fieldNew(nblock,nfieldv),
 1 stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
 2 enerInternNew(nblock), enerInelasNew(nblock)
```


For temperature and time independent relations the user coding must calculate and update the variables stressNew and stateNew. A description of all the variables can be found in the ABAQUS manual[48], whereas only the variables used in the present subroutine are described.

Variables to be defined

The main output variables are:

stressNew | Stress tensor at each material point at the end of the increment.
stateNew | State variables at each material point at the end of the increment.

Variables that can be updated

The main output variables that can be updated are:

enerInternNew | Internal energy per unit mass at each material point at the end of the increment.
enerInelasNew | Dissipated inelastic energy per unit mass at each material point at the end of the increment.

Variables passed in for information

nblock | Number of material points to be processed in this call to VUMAT.
ndir | Number of direct components in a symmetric tensor.
nshr | Number of indirect components in a symmetric tensor.
nstatev | Number of user-defined state variables that are associated with this material type.
nfieldv | Number of user-defined external field variables.
nprops | Number of user-defined material properties.
lanneal | Flag indicating whether the routine is being called during an annealing process. lanneal=0 indicates that the routine is being called during a normal mechanics increment. lanneal=1 indicates that this is an annealing process and the user should re-initialize the internal state variables, stateNew, if necessary. ABAQUS/Explicit will automatically set the stresses, stretches, and state to a value of zero during the annealing process.
stepTime | Value of time since the step began.
totalTime | Value of total time. The time at the beginning of the step is given by totalTime - stepTime.
dt  Time increment size.

*cmname*  Name given on the *MATERIAL* option, left justified. It is passed in as an upper case character string.

*coordMp*  Material point coordinates. It is the midplane material point for shell elements and the centroid for beam elements.

*charLength*  Characteristic element length. This is a typical length of a line across an element. For beams and trusses, it is a characteristic length along the element axis. For membranes and shells, it is a characteristic length in the reference surface.

*props*  User-supplied material properties defined using the *USER MATERIAL* option.

*density*  Current density at the material points in the midstep configuration. This value may be inaccurate in problems where the volumetric strain increment is very small. If an accurate value of the density is required in such cases, the analysis should be run in double precision. This value of the density is not affected by mass scaling.

*strainInc*  Strain increment tensor at each material point.

*relSpinInc*  Incremental relative rotation vector at each material point defined in the corotational system.

*tempOld*  Temperatures at each material point at the beginning of the increment.

*stretchOld*  Stretch tensor \( U \) at each material point at the beginning of the increment defined from the polar decomposition of the deformation gradient by \( F = U \dot{R} \).

*defgradOld*  Deformation gradient tensor at each material point at the beginning of the increment. Stored in 3-D as \((F_{11}, F_{22}, F_{33}, F_{12}, F_{23}, F_{31}, F_{21}, F_{32}, F_{13})\) and in 2-D as \((F_{11}, F_{22}, F_{33}, F_{12}, F_{21})\).

*fieldOld*  Values of the user-defined field variables at each material point at the beginning of the increment.

*stressOld*  Stress tensor at each material point at the beginning of the increment.

*stateOld*  State variables at each material point at the beginning of the increment.

*enerInternOld*  Internal energy per unit mass at each material point at the beginning of the increment.

*enerInelasOld*  Dissipated inelastic energy per unit mass at each material point at the beginning of the increment.

*tempNew*  Temperatures at each material point at the end of the increment.

*stretchNew*  Stretch tensor, \( U \), at each material point at the end of the increment defined from the polar decomposition of the deformation gradient by \( F = U \dot{R} \).

*defgradNew*  Deformation gradient tensor at each material point at the end of the increment.
6.2. IMPLEMENTATION OF USER SUBROUTINE

increment. Stored in 3-D as \((F_{11}, F_{22}, F_{33}, F_{12}, F_{23}, F_{31}, F_{21}, F_{32}, F_{13})\) and in 2-D as \((F_{11}, F_{22}, F_{33}, F_{12}, F_{21})\).

\(fieldNew\) Values of the user-defined field variables at each material point at the end of the increment.

It should be noticed that the sequence of stress and strain differ from the convention normally used in continuum mechanics as the shear components are interchanged. The stresses and the strain are given on vector form:

\[
\sigma^T = \{\sigma_x, \sigma_y, \sigma_z, \sigma_{xy}, \sigma_{xz}, \sigma_{yz}\}
\]

\[
\varepsilon^T = \{\varepsilon_x, \varepsilon_y, \varepsilon_z, \varepsilon_{xy}, \varepsilon_{xz}, \varepsilon_{yz}\}
\]

6.2.2 Syntax for calling the user subroutine

When using the user subroutine in ABAQUA/Explicit it is necessary to define an element set that uses the specific material definition. Furthermore, the number of material and model properties, \(NPROPS\), and the number of state variables, \(NSTATV\), must be defined together with the material and model properties. An example of a material definition in an ABAQUS input file is given below:

* MATERIAL,NAME=STEEL
  * User Material, constants=10
  211e+09, 0.3, 0.223, 513E+06, 91.3E+06, 1., 1.0E-02, 0., 1.0E-05, 200
  DEPVAR
  4,
  USER SUBROUTINE,INPUT=DeepDrawing.for

The material properties and various model control parameters are given by the property array, which is composed as follows:

\[
\begin{align*}
\text{PROPS}(1) & \quad \text{Parameter } E \quad \text{Young's Modulus} \\
\text{PROPS}(2) & \quad \text{Parameter } \nu \quad \text{Poisson's ratio} \\
\text{PROPS}(3) & \quad \text{Parameter } n \quad \text{strain hardening exponent in power law} \\
\text{PROPS}(4) & \quad \text{Parameter } K \quad \text{strength coefficient in power law} \\
\text{PROPS}(5) & \quad \text{Parameter } \sigma_Y \quad \text{initial yield stress} \\
\text{PROPS}(6) & \quad \text{User defined integration scheme:} \\
& \quad (1) \quad \text{Modified Euler scheme} \\
& \quad (2) \quad \text{Runge-Kutta-Dormand-Prince scheme} \\
& \quad (3) \quad \text{Forward Euler scheme}
\end{align*}
\]
6.2. IMPLEMENTATION OF USER SUBROUTINE

(4) Backward Euler scheme

PROPS(7) Parameter for integration schemes:
- Stress tolerance for RKDP scheme and ME scheme
- Number of subdivisions in the forward Euler scheme

PROPS(8) Correction for yield surface drift
- (0) Deactivated
- (1) Activated

PROPS(9) Tolerance for yield surface drift

PROPS(10) Maximum number of iterations

A number of state variables are used for storage of the materials local stress-strain history. The state variable array is composed as follows:

- STATEV(1) Current value of yield function
- STATEV(2) Current value of hardening function
- STATEV(3) Current value of effective plastic strain
- STATEV(4) Plastic indicator
  - (0) elastic state
  - (1) plastic state

6.2.3 Initialization of state variables

For use with user subroutine the ABAQUS input file is composed of at least six parts:

1. Initial geometry of the problem
2. Type of elements used to approximate the displacement field
3. Specification of material model properties
4. Boundary conditions, i.e. prescribed displacements and pore pressure
5. User defined geostatic stress field
6. Load steps, i.e. loads and prescribed displacements

In the data check phase of the analysis ABAQUS/Explicit calls user subroutine VUMAT with a set of fictitious strains and a totalTime and stepTime both equal to 0.0. This is done as a check on the user’s constitutive relation and to calculate the equivalent initial material properties, based upon which the initial elastic wave speeds are computed. For the first the time the user subroutine is called, all the state variables are initialized by VUMAT developed.
6.3 Special considerations for shell element

The use of user subroutine VUMAT requires special consideration for shell element type.

- The stresses and internal state variables must be defined by the user. In the case of shell elements, \( \text{strainInc}(*,3) \), the thickness strain increment, must be defined by the user.

- ABAQUS/Explicit cannot calculate a default value for the transverse shear stiffness of the element, when VUMAT is used to define the material response of shell elements. Hence, the ^*TRANSVERSE SHEAR STIFFNESS option must be used to define the element's transverse shear stiffness.

These two important topics form a major obstacle to the users who are interested in building their own VUMAT for sheet metal forming since ABAQUS analysis user’s manual does not provide a clear description about these issues. We discuss these issues here with the aim to ensure that our user subroutine can be employed easily by the interested users.

6.3.1 Update of sheet metal thickness

In sheet forming simulation, since the shell element is commonly utilized for a blank material under the plane stress condition, \( \Delta \varepsilon_z \) is not necessary for stress integration but \( \Delta \varepsilon_z \) needs to be updated for post-processing. Therefore, though four strain components

\[
\begin{bmatrix}
\Delta \varepsilon_x \\
\Delta \varepsilon_y \\
\Delta \varepsilon_z \\
\Delta \varepsilon_{xy}
\end{bmatrix}
= 
\begin{bmatrix}
\text{strainInc}(*,1) \\
\text{strainInc}(*,2) \\
\text{strainInc}(*,3) \\
\text{strainInc}(*,4)
\end{bmatrix}
\]

are transferred from VUMAT when the shell element is selected in ABQUS. Only three strain components

\[
\begin{bmatrix}
\Delta \varepsilon_x \\
\Delta \varepsilon_y \\
\Delta \varepsilon_{xy}
\end{bmatrix}
= 
\begin{bmatrix}
\text{strainInc}(*,1) \\
\text{strainInc}(*,2) \\
\text{strainInc}(*,4)
\end{bmatrix}
\]

are useful for the stress integration. Assuming previously the additive decoupling of the total strain increments into the elastic and incompressible plastic strain increments

\[
\Delta \varepsilon = \Delta \varepsilon^e + \Delta \varepsilon^p
\]  

(6.1)
where the elastic $\Delta \varepsilon^e = \{ \Delta \varepsilon^e_x, \Delta \varepsilon^e_y, \Delta \varepsilon^e_{xy} \}$ and plastic strain increments $\Delta \varepsilon^p = \{ \Delta \varepsilon^p_x, \Delta \varepsilon^p_y, \Delta \varepsilon^p_{xy} \}$ are obtained during the stress integration scheme. Then, the elastic thickness strain is obtained by the following equation, i.e.,

$$\Delta \varepsilon^e_z = -\frac{\nu}{1-\nu} (\Delta \varepsilon^e_x + \Delta \varepsilon^e_y) \tag{6.2}$$

Here, (6.2) can be derived by manipulating the following linear elasticity equations with the plane stress condition, $\Delta \sigma_z = 0$,

$$\Delta \varepsilon^e_x = \frac{\Delta \sigma_z}{E} - \frac{\nu}{E} (\Delta \sigma_y + \Delta \sigma_z)$$

$$\Delta \varepsilon^e_y = \frac{\Delta \sigma_z}{E} - \frac{\nu}{E} (\Delta \sigma_x + \Delta \sigma_z) \tag{6.3}$$

$$\Delta \varepsilon^e_z = \frac{\Delta \sigma_z}{E} - \frac{\nu}{E} (\Delta \sigma_x + \Delta \sigma_y)$$

Also, the plastic thickness strain is calculated with the incompressibility condition, i.e.,

$$\Delta \varepsilon^p_z = -\left( \Delta \varepsilon^p_x + \Delta \varepsilon^p_y \right) \tag{6.4}$$

Therefore, total thickness strain is obtained by (6.1), (6.2) and (6.4), i.e.,

$$\Delta \varepsilon_z = \Delta \varepsilon^e_z + \Delta \varepsilon^p_z = \Delta \varepsilon^e_z = -\frac{\nu}{1-\nu} (\Delta \varepsilon^e_x + \Delta \varepsilon^e_y) - \left( \Delta \varepsilon^p_x + \Delta \varepsilon^p_y \right) \tag{6.5}$$

Finally, $\Delta \varepsilon_z$ can be updated into strainInc(*,3).

### 6.3.2 Derivation of transverse shear stiffness

Consider a shell element of thickness $h$. Equilibrium equations are

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} = 0$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} = 0$$

$$\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} = 0$$
Resultants are defined as (tractions, moments and transverse shear forces)
\[
\begin{align*}
N_x &= \int \sigma_{xz} dz \\
N_y &= \int \sigma_{yz} dz \\
N_{xy} &= \int \sigma_{xy} dz \\
M_x &= \int \sigma_{xx} z dz \\
M_y &= \int \sigma_{yy} z dz \\
M_{xy} &= \int \sigma_{xy} z dz \\
Q_x &= \int \sigma_{xz} dz \\
Q_y &= \int \sigma_{yz} dz
\end{align*}
\]
Assume plane stress boundary condition \((\sigma_{xz} = \sigma_{yz} = \sigma_z = 0 \text{ at } z = \pm \frac{h}{2})\). Integration of equilibrium equations through thickness gives
\[
\frac{\partial N_x}{\partial x} + \frac{\partial N_{yx}}{\partial y} = 0
\]
\[
\frac{\partial N_{xy}}{\partial x} + \frac{\partial N_y}{\partial y} = 0
\]
\[
\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y} = 0
\]
Integration of equilibrium equations multiplied by \(z\) through thickness gives
\[
\frac{\partial M_x}{\partial x} + \frac{\partial M_{yx}}{\partial y} = Q_x
\]
\[
\frac{\partial M_{xy}}{\partial x} + \frac{\partial M_y}{\partial y} = Q_y
\]
Assume that membrane stresses \(\sigma_{xx}, \sigma_{yy} \text{ and } \sigma_{xy}\) are linear function of \(z\). Then, membrane stresses can be expressed in terms of resultants
\[
\begin{align*}
\sigma_{xx} &= \frac{N_x}{h} + \left(\frac{12M_x}{h^3}\right) z \\
\sigma_{yy} &= \frac{N_y}{h} + \left(\frac{12M_y}{h^3}\right) z \\
\sigma_{xy} &= \frac{N_{xy}}{h} + \left(\frac{12M_{xy}}{h^3}\right) z
\end{align*}
\]
Then, equilibrium equations become
\[
\begin{align*}
\frac{12Q_x}{h^3} z + \frac{\partial \sigma_{xx}}{\partial z} &= 0 \\
\frac{12Q_y}{h^3} z + \frac{\partial \sigma_{xy}}{\partial z} &= 0
\end{align*}
\]
Integration using plane stress boundary condition gives
\[
\begin{align*}
\sigma_{xx} &= \frac{3Q_x}{2h} \left[1 - \left(\frac{2z}{h}\right)^2\right] \\
\sigma_{yy} &= \frac{3Q_y}{2h} \left[1 - \left(\frac{2z}{h}\right)^2\right]
\end{align*}
\]
Thus transverse shear stiffness \(C_s\) is defined as
\[
Q_x = C_s \bar{q}_{zx}
\]
where $\gamma_{zz}$ is effective shear strain.

Strain energy per unit thickness can be expressed in resultants

$$\frac{1}{2} Q_x \gamma_{zz} = \frac{1}{2C_s} Q_x^2$$

(6.7)

Integration of strain energy through thickness gives ($\sigma_{xx} = G\gamma_{zz}$)

$$\int_{-h/2}^{h/2} \frac{1}{2} \sigma_{xx} \gamma_{zz} dz = \int_{-h/2}^{h/2} \frac{1}{2G} \sigma_{xx}^2 dz$$

$$= \frac{1}{2G} \left( \frac{3Q_x}{2h} \right)^2 \int_{-h/2}^{h/2} \left[ 1 - \left( \frac{2z}{h} \right)^2 \right]^2 dz$$

$$= \frac{1}{2G} \left( \frac{3Q_x}{2h} \right)^2 \frac{8}{15} h$$

(6.8)

Transverse shear stiffness is obtained by comparing (6.7) and (6.8)

$$C_s = \frac{5}{6} Gh$$

(6.9)

The detailed introduction of computing the transverse shear stiffness for shell elements can be found in [87, 88].
Chapter 7

Numerical experiment

In order to illustrate the performance of the proposed Runge-Kutta-Dormand-Prince algorithm in the finite element analysis of the sheet metal forming process, the proposed constitutive equations and the corresponding stress integration algorithms has been introduced into ABAQUS by way of user material subroutine to the analysis of a typical three dimensional deep cup drawing process, which is one of the benchmark problems in sheet metal forming industry.

7.1 Numerical experiment

For the implementation of the material model the finite element code ABAQUS/Explicit was selected due to its ability to deal with complex contact problems and large element distortions. The constitutive equations and the stress integration algorithms introduced in the previously chapters are formulated for the use in the materials subroutine VUMAT. Figure 7-1 shows a flow chart of stress integration process in user subroutine (VUMAT). For each time increment, ABAQUS/Explicit uses the central difference method introduced in Chapter 2 to update the strain tensor $\varepsilon$ at each integration point. Then it passes the updated strain tensors, together with the stress states already fulfilling the constitutive relations and other state variables, into the user subroutine. As shown in Figure 7-1, the user subroutine then uses these variables to compute the new stress states through the implicit and explicit stress update algorithms. All the stress update schemes are built into the user subroutine and the user can select which one to use for integrating the strain-stress relations by setting the parameter PROPS(6).

7.1.1 Numerical model

Figure 7-2 shows a schematic description of the tools. The blank is initially square, 200mm by 200mm. The die is a flat surface with a square hole 102.5mm by 102.5mm,
7.1. NUMERICAL EXPERIMENT

START OF USER SUBROUTINE VUMAT

ABACUS input: \( \sigma_{old}, \Delta \epsilon, \text{state}(\sigma_{old}, \kappa_{old}, \tilde{F}_{old}), nprop \)

Calculate elastic trial stress: \( \sigma_{trial} = \sigma_{old} + \Delta \epsilon \sigma_{old} + f \Delta \epsilon \)

\[ \sigma_{new} = \sigma_{trial} \]

Determine intersection point, i.e., \( F(\sigma_{trial}, \kappa_{old}) \leq 0 \)

Implicit integration

No

Explicit integration

Stress integration process

Determine the initial stress state \( \sigma = \sigma_{trial}, \kappa = \kappa_{old} \)

Modified Euler scheme.
Runge-Kutta-Dormand-Prince scheme.
Forward Euler scheme without error control.

Update Stress \( \sigma_{new} = \sigma_{old} + \Delta \sigma \)

Check if \( F(\sigma_{new}, \kappa_{new}) = 0 \)

Correcting yield surface drift

Update state variables

END OF USER SUBROUTINE VUMAT

Figure 7-1: Schematic flow chart of the stress integration process.

Figure 7-2: Schematic illustration of the tools and blank for the deep-cup drawing process.
rounded at the edges with a radius of 10mm. The square punch measures 100mm by 100mm and is rounded at the edges with the same 10mm radius. A load of 20000N is applied to the blank holder. The blank holder is then allowed to move only in the vertical direction to accommodate changes in the blank thickness. Owing to the symmetry of the material, only one-quarter of the structure is modelled. The FE mesh of the blank is composed of 1225 elements (4-node, reduced integration, doubly curved shell elements with five integration points through the shell section, called S4R in ABAQUS nomenclature [9]). The tools are meshed with 4-node, bilinear quadrilateral, rigid elements, called R3D4 in ABAQUS.

In a typical forming process the punch may move at speeds on the order of 1 m/sec, which is extremely slow compared to typical wave speeds in the materials to be formed. (The wave speed in steel is approximately 5000 m/sec.) In general, inertia forces will not play a dominant role for forming rates that are considerably higher than the nominal 1 m/sec rates found in the physical problem. In the results presented here, the drawing process is simulated by moving the punch downward through a total distance of 36 mm in 0.0036 seconds (as shown in Figure 7-3).

An associated flow rule is assumed. The Swift hardening law between effective stress $\bar{\sigma}$ and effective plastic strain $\bar{\varepsilon}_p$ is employed, i.e.,

$$\bar{\sigma} = K(\varepsilon_0 + \bar{\varepsilon}_p)^n$$

(7.1)

where $n$, $K$ and $\varepsilon_0$ are material constants as introduced previously. The material properties are taken as follows: mild steel, thickness 0.82mm, uniaxial true stress-true strain curve $\bar{\sigma} = 567.29(0.007127 + \bar{\varepsilon}_p)^{0.2637}$ MPa, Young's modulus $E = 206GPa$, Poisson's ratio $\nu = 0.33$, friction coefficient $\mu = 0.16$.

In order to test the influence of different tolerance on the accuracy, an "ideal" run is implemented with the same mesh, the same material properties and the same global

Figure 7-3: Deformed blank at the punch stroke 36 mm.
solution technique, but the Runge-Kutta-Dormand-Prince substepping scheme is used to integrate the constitutive law. The tolerance is set to be very small \((10^{-10})\). Even though the number of the substeps is sufficient high and the drift from the yield surface can be ignored, the stress correction are still used to ensure the accuracy of the 'ideal' run. These 'ideal' result is used to compare with the implicit return scheme, Runge-Kutta-Dormand-Prince method and Modified Euler scheme with different tolerances. The errors in the elastic-plastic stresses are computed using:

\[
Error = \frac{\left(\sum_{i=1}^{N} (\sigma_i - \sigma_i^{\text{ideal}})^2\right)^{1/2}}{\left(\sum_{i=1}^{N} \sigma_i^{\text{ideal}}^2\right)^{1/2}}
\]

where \(N\) is the total number of the integration points. For each explicit scheme, two different models are studied, i.e. with and without stress correction respectively.

Note that in all the following figures and tables, BE, FE, ME and RKDP denote the backward Euler scheme, forward Euler scheme, Modified Euler scheme and Runge-Kutta-Dormand-Prince scheme, respectively. FEC, MEC and RKDPC mean that stress correction is applied for these schemes.

### 7.1.2 Von Mises Yield Criteria

Our first constitutive model used to test the integration schemes is von Mises yield surface and the same yield function as the one used for the parallel analyses.

**Accuracy**

Difference of the results obtained from proposed integration schemes can be seen clearly from the effective stress and thickness contours of some models in Figures 7-4 - 7-11. In Fig. (7-16) and (7-17), we present strain-stress curve for the deep cup drawing process. The thickness strains are plotted in Fig. (7-18) and (7-19). With reference to these figures, we can find that, for the cases of large plastic strain problem, the results computed by the explicit schemes without stress correction are almost the same no matter how tight we set the error tolerance. However, the results are completely different if stress correction is employed. In fact, without stress correction, all the three explicit schemes can not give satisfactory accuracy for the case which involves strain hardening when the tolerance is set to be \(10^{-5}\) or smaller. We also can see that the forward Euler scheme gives the worst result without stress correction, and the backward Euler scheme gives relatively accurate result. It is worth noting that even the Runge-Kutta-Dormand-Prince is nearly insensitive to the error tolerance when no stress correction is applied for the deep drawing process, although the accuracy is improved.
7.1. NUMERICAL EXPERIMENT

Figure 7-4: Contour of the effective stress from 'ideal run' (von Mises model).

Figure 7-5: Contour of the effective stress from FE with stress correction (von Mises model).

Figure 7-6: Contour of the effective stress from ME(10^-4) without stress correction (von Mises model).

Figure 7-7: Contour of the effective stress from RKDP (10^-4) without stress correction (von Mises model).

Figure 7-8: Contour of the effective stress from ME (10^-2) without stress correction (von Mises model).

Figure 7-9: Contour of the effective stress from FE without stress correction (von Mises model).
7.1. NUMERICAL EXPERIMENT

Figure 7-10: Contour of the blank thickness from 'ideal' run (von Mises model).

Figure 7-11: Contour of the blank thickness from FE with stress correction (von Mises model).

Figure 7-12: Contour of the blank thickness from ME(10^-4) without stress correction (von Mises model).

Figure 7-13: Contour of the blank thickness from RKDP (10^-4) without stress correction (von Mises model).

Figure 7-14: Contour of the blank thickness from ME (10^-2) without stress correction (von Mises model).

Figure 7-15: Contour of the blank thickness from FE without stress correction (von Mises model).
as the error tolerance gets tighter. This again proves that, because the approximation nature of the finite element method, yield surface drift may occur with the stresses moving away from the yield surface. This deviation is practically independent of the integration scheme adopted. When the model involves strain (work) hardening where the yield surface is moving with loading increment, the drift is more significant. Since such discrepancies are usually cumulative, it is important to ensure that stresses are corrected back to the current yield surface at each step of the integration. Since we apply the stress correction at the end of each time increment, it does not improve the accuracy significantly when the error caused by the integration scheme is too serious. This is demonstrated by the Forward Euler scheme and the modified Euler scheme with tolerance of $10^{-1}$. However, it can make the computed stress to fulfill the consistency condition at the end of load increment and avoid error accumulation and therefore instabilities in the following time increments.
The errors in the computed stresses from different algorithms with different tolerance are listed in Table (7.1). With reference to Table (7.1), it can be seen that, the Backward Euler scheme gives good result which is adequate for the engineering computation, compared with the explicit schemes. All the explicit schemes give incorrect results if no stress correction is applied, although the accuracy can be improved slightly as the tolerance decreases for both the modified Euler scheme and the Runge-Kutta-Dormand-Prince scheme. For the cases in which stress correction is applied for the explicit schemes, all the explicit schemes give the better results than the return algorithm does if the integration error can be controlled to some extent, i.e. for the
Table 7.1: Results of errors for BE, FE, ME and RKDP with different tolerance (von Mises model)

<table>
<thead>
<tr>
<th>Method</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>1.34 x 10^{-2}</td>
</tr>
<tr>
<td>FE</td>
<td>8.26 x 10^{-1}</td>
</tr>
<tr>
<td>FEC</td>
<td>6.72 x 10^{-2}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>10^{-1}</th>
<th>10^{-2}</th>
<th>10^{-3}</th>
<th>10^{-4}</th>
<th>10^{-5}</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>9.54 x 10^{-1}</td>
<td>8.23 x 10^{-1}</td>
<td>5.84 x 10^{-1}</td>
<td>5.36 x 10^{-1}</td>
<td>5.37 x 10^{-1}</td>
</tr>
<tr>
<td>MEC</td>
<td>5.12 x 10^{-2}</td>
<td>2.56 x 10^{-2}</td>
<td>2.54 x 10^{-2}</td>
<td>1.58 x 10^{-2}</td>
<td>8.42 x 10^{-3}</td>
</tr>
<tr>
<td>RKDP</td>
<td>3.12 x 10^{-1}</td>
<td>2.09 x 10^{-1}</td>
<td>1.76 x 10^{-1}</td>
<td>1.78 x 10^{-1}</td>
<td>1.54 x 10^{-1}</td>
</tr>
<tr>
<td>RKDPC</td>
<td>2.38 x 10^{-2}</td>
<td>2.39 x 10^{-2}</td>
<td>6.13 x 10^{-3}</td>
<td>6.87 x 10^{-3}</td>
<td>6.35 x 10^{-3}</td>
</tr>
</tbody>
</table>

modified Euler scheme with tolerance of 10^{-5} and the Runge-Kutta-Dormand-Prince scheme with the tolerance of 10^{-3} or smaller. In [39], the measure of error which is approximately equal to the tolerance was achieved. However, as mentioned previously, the results obtained from Modified Euler scheme and Runge-Kutta-Dormand-Prince scheme are not improved as the error tolerance becomes tighter. This again indicates that the stress correction applied at the end of each load increment does not improve the performance significantly unless the integration error is controlled to some extent.

**Efficiency**

In Table (7.2), we list the total number of substeps for overall solution of the Modified Euler scheme and Runge-Kutta-Dormand-Prince scheme. The corresponding CPU time spent on overall solution is listed in Table (7.3). We can see that, the Runge-Kutta-Dormand-Prince scheme generally requires less substeps than the Modified Euler scheme for the fixed value of tolerance. For the tolerance which is from 10^{-3} to 10^{-4}, the Runge-Kutta-Dormand-Prince scheme uses approximately half of number of substeps consumed by the Modified Euler scheme. This confirms that the high-order Runge-Kutta-Dormand-Prince scheme does not need more substeps to obtain the prescribed accuracy. However, since the Runge-Kutta-Dormand-Prince scheme usually needs to compute elastic-plastic matrix six times for each substep, it still consumes more CPU time than the Modified Euler scheme does. It is interesting to see that the use of stress correction not only improves the accuracy, but saves the computational time significantly. It can be seen in Table (7.3) that for both modified Euler scheme and Runge-Kutta-Dormand-Prince scheme, the CPU time is greatly reduced when the
### 7.1. NUMERICAL EXPERIMENT

Table 7.2: Total substeps needed in the overall solution for ME and RKDP schemes with different tolerance (von Mises model).

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>(10^{-1})</th>
<th>(10^{-2})</th>
<th>(10^{-3})</th>
<th>(10^{-4})</th>
<th>(10^{-5})</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>46954562</td>
<td>46934813</td>
<td>46946482</td>
<td>55674653</td>
<td>87665434</td>
</tr>
<tr>
<td>MEC</td>
<td>46836448</td>
<td>46875346</td>
<td>46925948</td>
<td>52783446</td>
<td>81970435</td>
</tr>
<tr>
<td>RKDP</td>
<td>46816128</td>
<td>46785441</td>
<td>46786154</td>
<td>46786447</td>
<td>47343496</td>
</tr>
<tr>
<td>RKDPC</td>
<td>46807413</td>
<td>46768141</td>
<td>46747980</td>
<td>46747643</td>
<td>47302996</td>
</tr>
</tbody>
</table>

Table 7.3: CPU time (seconds) spent on the overall solution (von Mises model).

<table>
<thead>
<tr>
<th>Method</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>336</td>
</tr>
<tr>
<td>FE</td>
<td>396</td>
</tr>
<tr>
<td>FEC</td>
<td>392</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>(10^{-1})</th>
<th>(10^{-2})</th>
<th>(10^{-3})</th>
<th>(10^{-4})</th>
<th>(10^{-5})</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>376</td>
<td>375</td>
<td>375</td>
<td>432</td>
<td>558</td>
</tr>
<tr>
<td>MEC</td>
<td>374</td>
<td>374</td>
<td>373</td>
<td>418</td>
<td>469</td>
</tr>
<tr>
<td>RKDP</td>
<td>596</td>
<td>597</td>
<td>614</td>
<td>636</td>
<td>657</td>
</tr>
<tr>
<td>RKDPC</td>
<td>513</td>
<td>508</td>
<td>514</td>
<td>514</td>
<td>526</td>
</tr>
</tbody>
</table>

stress correction is used. This is probably because the stress states are integrated more accurately when the yield drift is corrected back to yield surface at the end of each load increment, causing no more effort spent on trying to correct the accumulative errors in the following increments. When the stress correction is employed, the Runge-Kutta-Dormand-Prince scheme is better than the modified Euler scheme in terms of its ability to integrate more accurately with tight tolerance.

#### 7.1.3 Hill's yield criteria

Cold-rolled sheet metal often exhibits anisotropic properties in the forming process, which are mostly caused by preferred orientations of grains developed during the severe plastic deformation such as cold rolling. The quadratic yield function of Hill[58] is still a commonly employed phenomenological approach to predict anisotropy in sheet metal forming operations despite its incapability to predict the so-called experimentally
observed anomalous behaviour.

Hill's stress potential is a simple extension of the Mises function to allow anisotropic behaviour. This function is

\[ F(\sigma) = \sqrt{F(\sigma_y - \sigma_z)^2 + G(\sigma_z - \sigma_x)^2 + H(\sigma_x - \sigma_y)^2 + 2L\sigma_{yx}^2 + 2M\sigma_{zx}^2 + 2N\sigma_{xy}^2} \]

where \( F, G, H, L, M, \) and \( N \) are material constants obtained by tests of the material in different directions. These constants are given as follows:

\[ F = 0.283, \quad G = 0.358, \quad H = 0.642, \quad L = M = N = 1.288 \]

Accuracy

For demonstration, we present the effective stress and thickness contours for some of the models with relatively large differences of the results in Figures (7-20) - (7-31). Difference of the results obtained from different integration schemes using Hill's yield criterion can be seen clearly in these figures. Again, we present effective plastic strain - effective stress curves in Figures (7-32) and (7-33) and the thickness strains are shown in Figure (8-9) and (8-10). The results once more demonstrate that for Hill's model, all the explicit schemes give inaccurate results when no stress correction is applied, no matter how tight the tolerance is. However, for such complex constitutive model, some of the models, i.e., ME with tolerance of \( 10^{-1} \) and \( 10^{-2} \) and RKDP with tolerance of \( 10^{-1} \), still produce unsatisfactory results even with the help of the stress correction.

With reference to errors in the computed stresses from different algorithms shown in Table (7.4), we can see again that the Backward Euler scheme can provide accuracy which is superior to all the explicit schemes in the case that no stress correction is applied. However when the stress correction is employed, some of the models, such as ME \( (10^{-4}, 10^{-5}) \) and RKDP \( (10^{-4}, 10^{-5}) \) can produce same or even better level of accuracy than the backward Euler does. Again, we did not obtain the measure of error which is approximately equal to tolerance, even with the high order RKDP scheme.

Efficiency

We present the total number of substeps for overall solution of Modified Euler scheme and Runge-Kutta-Dormand-Prince scheme in Table (7.5) and the corresponding CPU time spent on overall solution in Table (7.6). It can be seen that the results we obtained are similar to the ones from von Mises model, i.e., the Runge-Kutta-Dormand-Prince scheme generally requires less substeps than the Modified Euler scheme does for the fixed value of tolerance. However it uses more CPU time than the modified Euler does.
7.1. NUMERICAL EXPERIMENT

Figure 7-20: Contour of the effective stress from 'ideal run' (Hill’s model).

Figure 7-21: Contour of the effective stress from FE with stress correction (Hill’s model).

Figure 7-22: Contour of the effective stress from ME(10^{-4}) with stress correction (Hill’s model).

Figure 7-23: Contour of the effective stress from RKDP (10^{-2}) with stress correction (Hill’s model).

Figure 7-24: Contour of the effective stress from ME (10^{-4}) without stress correction (Hill’s model).

Figure 7-25: Contour of the effective stress from FE without stress correction (Hill’s model).
7.1. NUMERICAL EXPERIMENT

Figure 7-26: Contour of the blank thickness from 'ideal' run (Hill's model).

Figure 7-27: Contour of the blank thickness from FE with stress correction (Hill's model).

Figure 7-28: Contour of the blank thickness from ME(10^-4) with stress correction (Hill's model).

Figure 7-29: Contour of the blank thickness from RKDP (10^-2) with stress correction (Hill's model).

Figure 7-30: Contour of the blank thickness from ME (10^-4) without stress correction (Hill's model).

Figure 7-31: Contour of the blank thickness from FE without stress correction (Hill's model).
due to the six evaluations of the elastic-plastic matrix for each substep. Obviously this is the cost we have to pay when high level of accuracy is required.
Figure 7-34: Plot of thickness strain for BE, FE, FEC, ME, MEC (Hill’s model).

Figure 7-35: Plot of thickness strain for BE, FE, FEC, RKDP, RKDPC (Hill’s model).
### Table 7.4: Results of errors for BE, FE, ME and RKDP with different tolerance (Hill's model)

<table>
<thead>
<tr>
<th>Method</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>$3.65 \times 10^{-2}$</td>
</tr>
<tr>
<td>FE</td>
<td>$8.73 \times 10^{-1}$</td>
</tr>
<tr>
<td>FEC</td>
<td>$2.89 \times 10^{-1}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>$10^{-1}$</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>$9.03 \times 10^{-1}$</td>
<td>$6.74 \times 10^{-1}$</td>
<td>$6.75 \times 10^{-1}$</td>
<td>$6.74 \times 10^{-1}$</td>
<td>$6.52 \times 10^{-1}$</td>
</tr>
<tr>
<td>MEC</td>
<td>$2.85 \times 10^{-1}$</td>
<td>$2.86 \times 10^{-1}$</td>
<td>$4.12 \times 10^{-2}$</td>
<td>$3.76 \times 10^{-2}$</td>
<td>$3.68 \times 10^{-2}$</td>
</tr>
<tr>
<td>RKDP</td>
<td>$5.62 \times 10^{-1}$</td>
<td>$5.63 \times 10^{-1}$</td>
<td>$5.62 \times 10^{-1}$</td>
<td>$5.34 \times 10^{-1}$</td>
<td>$5.29 \times 10^{-1}$</td>
</tr>
<tr>
<td>RKDPC</td>
<td>$4.56 \times 10^{-1}$</td>
<td>$4.55 \times 10^{-2}$</td>
<td>$4.74 \times 10^{-2}$</td>
<td>$4.06 \times 10^{-2}$</td>
<td>$8.14 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

### Table 7.5: Total substeps needed in the overall solution for ME and RKDP schemes with different tolerance (Hill's Model)

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>$10^{-1}$</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>48256706</td>
<td>48258361</td>
<td>48264593</td>
<td>56247482</td>
<td>86196585</td>
</tr>
<tr>
<td>MEC</td>
<td>48247453</td>
<td>48256488</td>
<td>48325473</td>
<td>54886242</td>
<td>82924864</td>
</tr>
<tr>
<td>RKDP</td>
<td>47689724</td>
<td>47445372</td>
<td>47446585</td>
<td>47447150</td>
<td>48245686</td>
</tr>
<tr>
<td>RKDPC</td>
<td>47671342</td>
<td>47438546</td>
<td>47435692</td>
<td>47432238</td>
<td>47119227</td>
</tr>
</tbody>
</table>

### Table 7.6: CPU time (seconds) spent on the overall solution (Hill's model)

<table>
<thead>
<tr>
<th>Method</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>347</td>
</tr>
<tr>
<td>FE</td>
<td>395</td>
</tr>
<tr>
<td>FEC</td>
<td>382</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>$10^{-1}$</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>387</td>
<td>380</td>
<td>389</td>
<td>431</td>
<td>572</td>
</tr>
<tr>
<td>MEC</td>
<td>377</td>
<td>368</td>
<td>388</td>
<td>423</td>
<td>476</td>
</tr>
<tr>
<td>RKDP</td>
<td>635</td>
<td>639</td>
<td>648</td>
<td>674</td>
<td>692</td>
</tr>
<tr>
<td>RKDPC</td>
<td>556</td>
<td>553</td>
<td>563</td>
<td>580</td>
<td>618</td>
</tr>
</tbody>
</table>
7.2 Summary

In this chapter, we employed these stress update schemes for analysis of the three-dimensional sheet metal forming problem of which the strain hardening is considered. The performance of new Runge-Kutta-Dormand-Prince scheme has been compared with the commonly used implicit and explicit integration schemes, including backward Euler algorithm, forward Euler with subincrementation and the modified Euler scheme. This comparison is specifically focused on the sheet metal forming process. The numerical example indicated that

1. For large strain sheet metal forming problem, neither of the explicit substepping schemes with error control can give accurate result without the application of the stress correction. However, when the substepping schemes are used together with the stress correction, the accuracy can be improved significantly.

2. High order Runge-Kutta-Dormand-Prince in combination with tight tolerance is better than both implicit backward Euler scheme and explicit second order modified Euler scheme in terms of integrating strain-stress relations more accurately.
Chapter 8

Numerical and experiment verification

In this chapter the performance of the Runge-Kutta-Dormand-Prince algorithm is examined further through conducting different deep drawing tests and comparing the testing results with the FEA simulations. Due to being unable to use selected material to achieve larger draw-in before fracture, we also use RKDP scheme to conduct FEA on Numisheet’93 benchmark problem, i.e., square cup drawing, and compare our FEA simulation with the experimental results provided in Numisheet’93 proceedings.

8.1 Deep drawing experiment

Two deep drawing tests with different punch geometries were conducted in our sheet metal forming lab. Experiments were conducted in a ENERPAC Formability System (shown in Figure 8-1) with a punch capacity of 30 Tons, and a maximum punch stroke of approximately 200 mm. The GOM optical measuring system shown in Figure 8-2 is used to measure the strain on the deformed blank. Then measured strains at the critical regions of the blank are compared with the strains calculated by the simulations.

8.1.1 Experimental setup

The tooling used in drawing tests are similar to the one shown in Figure 7-2 in Chapter 7. Blanks of 0.8 mm thickness were originally sheared to a size of 200 x 200 mm and were cut further to the sizes of 100 x 100 mm and 150 x 200 mm for square cup drawing and rectangular cup drawing, respectively, to avoid serious wrinkling. The sizes of the dies are 45 x 45 mm and 60 x 105 mm for square cup and rectangular cup, respectively. The square punch measures 40 x 40 mm and the size of the rectangular punch is 100
8.1. DEEP DRAWING EXPERIMENT

Figure 8-1: ENERPAC 30 TON deep drawing machine.

Figure 8-2: Gom optical strain measuring system.

x 55 mm. A load of 19500N is applied to the blank holder. The blank holder is then allowed to move only in the vertical direction to accommodate changes in the blank thickness. Parts were then formed with a punch speed of 1 m/second. Care was taken to assure that the part was symmetrically aligned with respect to the punch and binder. The forming depth for the experiments was set at 25 mm. Only a quarter of the blank is analyzed on account of symmetry. Figure 8-3 and 8-4 show a schematic description of these two discretized FEA models. The blank is meshed by the 4-node, reduced integration, doubly curved shell elements with five integration points through the shell section (S4R) and the tools are meshed with 4-node, bilinear quadrilateral, rigid element (R3D4).
8.1. DEEP DRAWING EXPERIMENT

Figure 8-3: Discretized FEA model for rectangular cup drawing.

Figure 8-4: Discretized FEA model for square cup drawing.

Table 8.1: Material properties for square and rectangular cup drawing

<table>
<thead>
<tr>
<th>Offset from rolling direction (degree)</th>
<th>Yield stress (MPa)</th>
<th>Ultimate tensile stress (MPa)</th>
<th>n Value</th>
<th>K Value (MPa)</th>
<th>E (GPa)</th>
<th>ν</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mild steel, thickness 0.80 mm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>174.2</td>
<td>314.7</td>
<td>0.207</td>
<td>537.4</td>
<td>205</td>
<td>0.3</td>
</tr>
<tr>
<td>45</td>
<td>174.8</td>
<td>316.0</td>
<td>0.199</td>
<td>547.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>197.4</td>
<td>324.8</td>
<td>0.198</td>
<td>523.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean σ = 540.6εp^0.2037</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8.1.2 Material properties

The tensile test was not conducted due to the material properties being provided by the vendor. However when the material isotropy was taken into account, anisotropy coefficients \( r_0, r_{45} \text{ and } r_{90} \), were needed. In this case the standard values used wildly for mild steel were employed. This thus affects the accuracy of numerical results to some extent. The material properties are given in Table 8.1.

8.1.3 Comparison

In Figures 8-5 and 8-6, we give the contours of blank thickness for square cup drawing. Figures 8-7 and 8-8 show the contours of blank thickness for rectangle cup drawing. The differences of the results calculated by ABAQUS and ABAQUS user subroutine can be seen clearly from these figures. Figures 8-9 and 8-10 show the thickness strain
8.1. DEEP DRAWING EXPERIMENT

Figure 8-5: Contour of blank thickness from user subroutine (Hill's model with RKDP,BE).

Figure 8-6: Contour of blank thickness from ABAQUS (von Mises model).

Figure 8-7: Contour of blank thickness from user subroutine (Hill's model with RKDP,BE).

Figure 8-8: Contour of blank thickness from ABAQUS (von Mises model).
8.1. DEEP DRAWING EXPERIMENT

Figure 8-9: Plot of thickness strain for square cup drawing.

Figure 8-10: Plot of thickness strain for rectangular cup drawing.
versus diagonal distance curves for experimental result and two numerical simulations with and without user subroutine. With reference to these figures, we can see that the simulation calculated by ABAQUS is the worst one as it uses von Mises yield function. The simulation which was done via user subroutine gives better results due to Hill’s yield function being used. It again proves that the accuracy of simulation is mainly dependent on the constitutive models used. Also the numerical models again demonstrated that the RKDP scheme with tolerance of $10^{-4}$ can achieve the same level of the accuracy as the backward Euler scheme does.

It should be pointed out that the experimental results for both square cup drawing and rectangular cup drawing are not very accurate due to the black dots at some critical regions being rub out, making the strain at that area not measurable. In this case the optical measuring system uses the linear compensation method to interpret the strains at nearby areas or we used the strain measured from another deformed blank.
8.2 Numisheet'93 benchmark problem

The NUMISHEET Conferences have been established as a world-class forum through which new intellectual ideas and technology in the area of sheet metal simulations are exchanged. One of the distinguishing characteristics of the NUMISHEET Conference is the inclusion of the Benchmark Study. This study is designed to allow engineers and scientists to evaluate their ability to model accurately various sheet metal forming processes through careful examination of experimental and simulation data in a blind test.

This experimental investigation of the deep drawing of a square cup has been carried out as part of an international benchmark research program organised by the NUMISHEET'93 organising committee. The purpose of the benchmark research program was to have a number of different research institutes carry out the same deep drawing process with the same material and lubricant. The experimental results were compared at the NUMISHEET'93 conference, the same deep drawing process has been analysed by a number of different research institutes using simulation, at the conference the experimental results being compared to the simulation results with the purpose of seeing how accurate the different simulation tools used are. To test the RKDP scheme under large punch depth, our FEA model has been modified to reflect the experimental setup used in NUMISHEET'93.

8.2.1 Experimental setup

The deep drawing tool was made according to the schematic drawing shown in Figure 8-11. The drawing process was carried out on a double-action hydraulic press with the punch travel of exactly 15 and 40 mm. In our simulation, only punch travel of 40 mm is used for comparison and Discretized FEA model is shown in Figure 8-12.

8.2.2 Material Properties

The two materials employed in NUMISHEET'93 were mild steel and aluminum, both supplied cut to the dimension 150x150 mm by the NUMISHEET'93 organising committee. We only used mild steel in our research and its material data is given in Table 8.2

8.2.3 Blank preparation and lubrication

In order to be able to determine the strains on the surface after deformation, a grid consisting of squares with a grid pitch equal to 2.5 mm was electrochemically etched
Table 8.2: Material properties used in Numisheet’93 square cup drawing

<table>
<thead>
<tr>
<th>RD</th>
<th>YS (MPa)</th>
<th>TS (MPa)</th>
<th>El n (10-20%)</th>
<th>r (20%)</th>
<th>E (GPa)</th>
<th>ν</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>167.0</td>
<td>308.5</td>
<td>47.3</td>
<td>0.238</td>
<td>1.79</td>
<td>206</td>
</tr>
<tr>
<td>45</td>
<td>176.5</td>
<td>316.0</td>
<td>44.0</td>
<td>0.239</td>
<td>1.51</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>172.5</td>
<td>305.0</td>
<td>46.0</td>
<td>0.222</td>
<td>2.27</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>173.1</td>
<td>311.4</td>
<td>45.3</td>
<td>0.235</td>
<td>1.77</td>
<td></td>
</tr>
</tbody>
</table>

Mean true stress-true strain curve: \(565.32(0.007117 + \varepsilon_p)^{0.2589}\)

on to the surface of the undeformed blanks. The length of the blank diagonals were measured. Before each experiment each side of the blank were wiped with a paper towel dipped in the lubricant supplied by the NUMISHEET’93 organising committee and the blank was then kept in the vertical position for 30 minutes before it was position on blank holder. This lubrication procedure should yield a friction coefficient of \(\mu = 0.144\) for mild steel.

8.2.4 Comparison of experiment and numerical results

As mentioned previously the cup drawing simulations were conducted via ABAQUS/Explicit, one with the default von Mises material model without using user subroutine and the other with Hill’s anisotropic yield function in combination with the new RKDP scheme (tolerance of \(10^{-4}\)). In Figure 8-13 and 8-14, we give the contours of blank thickness.

Figure 8-11: Schematic illustration of tooling geometry for Numisheet’93 benchmark square cup drawing.
From these figures we can see the difference of these two numerical results. Figure 8-15 shows thickness strain versus diagonal distance curves for NUMISHEET'93 experimental result and two numerical simulations with and without user subroutine. Different from the comparison we conducted using our own experimental results, the numerical results calculated by user subroutine agree qualitatively well with the NUMISHEET'93 experimental results although the simulated results are overestimated compared to the experiments. The simulation calculated by ABAQUS/Explicit without user subroutine gives less accurate results, which again demonstrates that traditional von Mises constitutive model is not accurate enough for simulating complex large strain sheet metal forming process. It is interesting to see that the RKDP scheme again achieves same
level of accuracy as the backward Euler scheme does, even under large punch depth. In summary the comparison again demonstrates that new RKDP scheme is capable of controlling the integration error and achieving accurate results.

It should be noted that the experiments of square cup drawing were carried out by many research institutes and most of the experimental results obtained by different participants significantly disagreed with each other and thus provided very poor reference data for evaluation of the simulations. It is very important to establish a standard experimental data base which could be obtained by a well-defined experimental procedure. The experimental results chosen for comparison were recognised to be the better ones by NUMISHEET'93 organising committee.
Chapter 9

Conclusion and future research

9.1 Concluding Remarks

The automotive industry is now facing world-wide serious challenge: fierce market competition. A very significant component of this endeavour is focused on the reduction of the tooling costs and the lead-time related to the stamping of autobody panels. FEA is now firmly accepted as a powerful general technique for predicting a variety of forming problems encountered in sheet metal forming process.

As more and more complex material models, like anisotropic (texture-based) strain-rate potential or crystal plasticity, are being developed, the use of such complex models raises a number of questions in relation to implementing an appropriate stress update algorithm. It is well-known that application of the existing implicit algorithms to complex constitutive model is quite often cumbersome or even infeasible due to the second order derivative of plastic potential being needed and the convergence is not always guaranteed. However, the easy-to-use explicit scheme requires that load increments be kept small enough so that the stresses computed at the end of integration do not deviate too far from the yield surface and there is no error control in the integration process. Therefore, it would be desirable to have an explicit integration scheme which can not only maintain as same accuracy as the implicit does but is easy to implement for simulating complex sheet metal forming process.

Motivated by the background above, an advanced substepping scheme has thus been developed for the elastic-plastic nonlinear analysis in this work. The substepping scheme is based on the high order Runge-Kutta-Dormand-Prince method and controls the error in the integration process to the vicinity of a specified tolerance. This mechanism for controlling the integration process permits the size of each substep to vary in accordance with the behaviour of the constitutive law.

To test the performance of the Runge-Kutta-Dormand-Prince scheme, we firstly
begin with applying it to a non-linear elastic-plastic finite element analysis of a simple three dimensional cantilever beam with complex constitutive model and other enhancements of the old nonlinear code developed. The numerical examples indicated that high order RKDP scheme with tight tolerance in combination with advanced stress correction algorithm, is better than both implicit backward Euler scheme and explicit second order modified Euler scheme. It also has been found that under large plastic deformation, the accuracy of all the substepping schemes can not be improved significantly as the tolerance decreases. Some forms of stress correction should thus be employed in order to ensure that the computed stresses remain on the yield surface at any time and maintain adequate accuracy of the overall solution. The results also show that applying the stress correction at the end of integration does not increase the computational time significantly, but it can improve the accuracy if the errors during the integration can be reduced to some extent.

The new Runge-Kutta-Dormand-Prince scheme has also been presented for parallel elastic-plastic analyses, in combination with an advanced parallel preconditioned conjugate gradient algorithm with MR smoothing. In the resulting algorithm for solving the linear system of equations, a parallel substructure preconditioned conjugate gradient method is used. This solution algorithm does not require the assembly of the global system equations. Each processor in the parallel system is assigned a substructure and stores only the information relevant to the substructure that the processor represents. The RKDP scheme, the parallel substructure preconditioned conjugate gradient method and all the parallel techniques developed have been applied to two three dimensional elastic-plastic stress analyses with complex non-linear hardening using both von Mises and complex Hill's anisotropic model. The loads applied were set to be large in order to test the parallel performance of RKDP scheme under large plastic strain condition. The parallel numerical examples indicated that new RKDP scheme is superior to the old Gill's RK scheme with even better speedup and efficiency being achieved under both complex material model and large plastic strain condition. In summary, the combination of these two algorithm provides a useful practical means for parallel finite element analysis of large elastic-plastic problems.

After testing the new RKDP scheme initially with simple FEA model, we then employed all stress update schemes for analysis of the three-dimensional sheet metal forming problem of which the a extremely complex hardening model is employed. The performance of the new RKDP scheme has been assessed through comparing with the commonly used implicit and explicit integration schemes, including backward Euler algorithm, forward Euler with subincrementation and the modified Euler scheme. The numerical example indicated that high order RKDP scheme in combination with tight
tolerance is better than both implicit backward Euler scheme and explicit second order modified Euler scheme in terms of integrating strain-stress relations more accurately.

Finally the performance of the RKDP algorithm was examined further through conducting different deep drawing tests and comparing the experimental results with the FEA simulations. Due to being unable to use selected material to achieve larger draw-in before fracture, we also used RKDP scheme to conduct FEA on Numisheet'93 benchmark problem, i.e., square cup drawing, and compared our FEA simulation with the experimental results provided by Numisheet'93 organising committee. The numerical models again demonstrated that the RKDP scheme with tight tolerance in combination with stress correction can achieve the same level of the accuracy as the backward Euler scheme does.

In summary, our research results indicate that:

- Different from some research works done by other researchers, for large strain sheet metal forming problem, neither of the explicit substepping schemes with error control can give accurate result without the application of the stress correction. However, when the substepping schemes are used together with stress correction, the accuracy can be improved significantly.

- Applying the stress correction at the end of integration does not increase the computational time significantly, but it can improve the accuracy if the errors during the integration can be reduced to some extent.

- High order Runge-Kutta-Dormand-Prince in combination with tight tolerance is better than both implicit backward Euler scheme and explicit second order modified Euler scheme in terms of integrating strain-stress relations more accurately.

As more and more complex constitutive model is used for simulating the sheet metal forming process, implementation of the implicit stress update schemes on such model could be extremely difficult. In this case, the combination of the substepping scheme with the error control combined with the stress correction method proposed in this research work could offer an alternative way which is efficient and accurate. With the code of ABAQUS user subroutine provided with this thesis, other researcher can build their own constitutive models into this program and conduct more accurate simulations.

9.2 Future Work

The automotive industry is more and more relying on the numerical simulation to reduce the tooling costs and the lead-time related to the stamping of autobody panels.
The success of numerical simulation depends mainly on the advances in forming simulation codes. In order to fully understand the physics of the manufacturing processes and its effect on product performance and cost, the following areas need to be researched:

- **Material and Friction Modelling.** Numerical results are strongly affected by physical models such as the material models and friction models. It is, therefore, essential to improve these physical models in order to obtain satisfactory accuracy, especially for forming processes involving repeated bending of the sheet, large plastic deformations and/or strain-path changes.

- **Accurate calculation prediction of the springback.** Currently only static implicit incremental code can satisfactorily simulate the springback process. Some static explicit codes can be used for springback simulations, but sometimes it gives unsatisfactory results. There have been significant efforts to improve dynamic codes to obtain ability of springback prediction, but it has not reached a satisfactory level yet. Recently an important innovation was introduced in a direct sparse matrix solver which reduces the solution time of the stiffness equation as well as memory requirement by one order of magnitude compared with a band solver. Some researches already demonstrate that implementation of this sparse solver to a static FEM can improve the ability of the code. But more researches are still needed so that solving large-scale problems within a reasonable time using as many elements as used in a dynamic calculation can be achieved.

- **Simulation of Entire Cycle.** In most of the companies the simulated process is limited only to the deep drawing stage. However, the press stamping processes are comprised of a series of operations; deep drawing, trimming, flanging, and restriking. And, thereafter, the hemming and the spot welding are performed. It is expected that the simulation would perform calculations for the entire process to evaluate all the defects involved. Furthermore, sometimes the effects of forming process on the dent resistance and the fatigue resistance are also required.

- **High performance parallel computing systems.** Directly affect the computational time and the level of fine details that can be incorporated into the models, and hence the accuracy and benefits that can be obtained. Therefore, there is a need to develop an effective parallel computational strategy, including automatic mesh generator and powerful parallel solver, for performing an entire simulation with relevance to manufacturing applications.

The above features of future research work will lead to a better understanding of the physics involved in sheet metal modelling and when combined with a robust opti-
misation algorithm will provide the designer a powerful tool in designing high quality products at a faster rate and reduced cost.
Bibliography


1 Basic Topics. John Wiley and Sons Ltd, 1997.


algorithm for creep damage analysis. International Journal of Pressure Vessels 

integer time step ratios fo r linear structural dynamic systems. Comp. Mech., 
18:236–244, 1996.

[51] W.J.T. Daniel. Analysis and implementation of a new constant acceleration sub­ 
cycling algorithm. International Journal for Numerical Methods in Engineering, 

[52] D.R. Metzger. Adaptive damping for dynamic relaxation problems with non­ 
monotonic spectral response. International Journal for Numerical Methods in 


1950.

[59] M. Ortiz and J.C. Simo. An Analysis of a New Class of Integration Algorithms for 
Elastoplastic Constitutive Relations. International Journal of Numerical Methods 


Appendix A

Runge-Kutta Methods

A Runge-Kutta Method is one which employs a recurrence formula of the form

$$y_{i+1} = y_i + a_1 k_1 + a_2 k_2 + a_3 k_3 + \cdots + a_n k_n$$  \hspace{1cm} (A.1)

to calculate successive values of the dependent variable $y$ of the differential equation

$$\frac{dy}{dx} = y' = f(x, y)$$  \hspace{1cm} (A.2)

where

$$k_1 = (h)f(x_i, y_i)$$

$$k_2 = (h)f(x_i + p_1 h, y_i + q_1 k_1)$$

$$k_3 = (h)f(x_i + p_2 h, y_i + q_2 k_1 + q_2 k_2)$$

$$\cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots$$

$$k_n = (h)f(x_i + p_{n-1} h, y_i + q_{n-1,1} k_1 + q_{n-1,2} k_2 + \cdots + q_{n-1,n-1} k_{n-1})$$  \hspace{1cm} (A.3)

The above Runge-Kutta equations can be written more compactly as

$$y_{i+1} = y_i + \sum_{j=1}^{n} a_j k_j$$  \hspace{1cm} (A.4)

where

$$k_j = h f(x_i + p_j h, y_i + \sum_{l=1}^{j-1} q_{j-1,i} k_l) \quad (j = 1, 2, \cdots, n)$$  \hspace{1cm} (A.5)

in which, by definition,

$$p_0 = 0 \quad \text{and} \quad \sum_{l=1}^{j-1} q_{j-1,i} k_l = 0 \quad j = 1$$  \hspace{1cm} (A.6)

The $a's$, $p's$, and $q's$ must assume values such that Equation A.1 accurately yields successive values of $y$. These values are determined by making Equation A.1 equivalent to a certain specified number of terms of a Taylor-series expansion of $y$ about $x_i$. 

132
If we let $m$ be the order of the Runge-Kutta method, with $n$ the number of function evaluations per step (number of $k$ values), there is a particular maximum value of $m$ for each $n$. For $n$ values up to 7, we have

\begin{align*}
1 \leq n &\leq 4, \quad m_{\text{max}} = n \\
n & = 5, \quad m_{\text{max}} = 4 \\
n & = 6, \quad m_{\text{max}} = 5 \\
n & = 7, \quad m_{\text{max}} = 6
\end{align*}

Various Runge-Kutta methods are classified as $(m, n)$ methods. A $(5, 6)$ method, for example, would be a fifth-order method requiring six function evaluations per step.
Appendix B

A example of ABAQUS input file

**
** ABAQUS Input Deck Generated by HyperMesh
**

*HEADING
DEEP DRAWING OF A RECTANGULAR CUP
(1/4 SYMMETRY MODEL)
**

** Generate the mesh for the PUNCH
**

*NODE
52346, -17.54572873154, 34.49118021645, 39.830555018098
52347, -17.54572873154, 34.49118021645, 37.955555018098
52348, -17.54572873154, 34.49118021645, 36.080555018098

• • • in total, 835 NODES DEFINED

53179, -18.38833826768, 65.260410985681, 22.052767313647
53180, -24.42072873154, 84.49118021645, 19.830555018098
53181, -26.29572873154, 84.49118021645, 19.830555018098

*ELEMENT, TYPE=R3D4, ELSET=PUNCH
111358, 53074, 53170, 53138, 53073
111357, 53170, 53168, 53139, 53138
111356, 53075, 53164, 53170, 53074

• • • in total, 780 ELEMENTS DEFINED

111440, 53113, 53114, 53103, 53104
111441, 53112, 53113, 53104, 53105
111442, 53110, 53112, 53105, 53106

*RIGID BODY, REF NODE = 52523, ELSET = PUNCH
**
** Generate the mesh for the HOLDER
**
*NODE
53182, -28.17072873154, 84.49118021645 , 19.830555018098
53183, -30.04572873154, 84.49118021645 , 19.830555018098
53184, -31.92072873154, 84.49118021645 , 19.830555018098
... in total, 3362 NODES DEFINED
.
57185, 9.4542712684615, 29.08118021645 , -2.169444981902
57186, 9.4542712684615, 29.08118021645 , -1.169444981902
57187, 9.4542712684615, 29.08118021645 , -0.169444981902
*ELEMENT,TYPE=R3D4,ELSET=HOLDER
131163, 67640, 67639, 67644, 67646
131162, 67627, 67640, 67646, 67628
131161, 67646, 67644, 67574, 67575
... in total, 3248 ELEMENTS DEFINED
.
129171, 65611, 65444, 65445, 65613
129170, 65609, 65610, 65611, 65612
129169, 65610, 65443, 65444, 65611
*RIGID BODY, REF NODE = 67067, ELSET = HOLDER
**ELEMENT,TYPE=MASS,ELSET=emass
141240, 67067
**ELEMENT,TYPE=MASS,ELSET=emass
*MASS, ELSET=emass
6.3960E-04,
**
** Generate the mesh for the DIE
**
*NODE
57188, 9.4542712684615, 29.08118021645 , 0.8305550180976
57189, 10.454271268462, 29.08118021645 , 1.8305550180976
57190, -31.92072873154, 29.08118021645 , -2.169444981902
... in total, 2702 NODES DEFINED
.
65952, 29.49118021645 , 29.08118021645 , -8.799694803399
65953, 29.49118021645 , 29.08118021645 , -6.836731919029
65954, 29.49118021645 , 29.08118021645 , -6.836731919029
*ELEMENT,TYPE=R3D4,ELSET=DIE
129083, 65360, 65295, 65275, 65361
129082, 65359, 65298, 65295, 65360
129081, 65358, 65301, 65298, 65359
APPENDIX B. A EXAMPLE OF ABAQUS INPUT FILE

... in total, 2578 ELEMENTS DEFINED

355216, 313044, 315828, 315826, 313045  
355217, 315792, 315791, 315822, 315828  
355218, 315828, 315822, 315821, 315826

*RIGID BODY, REF NODE = 59617, ELSET = DIE

** Generate the mesh for the BLANK

**

*NODE

65955, -23.97806506714, 29.49118021645, -2.910806150333
65956, -25.9205695531, 29.49118021645, -0.947843266008
65957, -27.86166310431, 29.49118021645, 1.0151196183027

... in total, 5929 NODES DEFINED

67607, -22.07930017254, 35.49118021645, 10.830149818728
67607, -14.34596684155, 31.49118021645, 10.830149818728
67607, -14.34596684155, 33.49118021645, 10.830149818728

*ELEMENT,TYPE=S4R,ELSET=BLANK

117931, 59161, 59160, 59162, 59163
117930, 59156, 59161, 59163, 59157
117929, 59163, 59162, 59151, 59150

... in total, 5776 ELEMENTS DEFINED

131165, 67647, 53797, 53796, 55752
131168, 67647, 53749, 53499, 55753
131167, 67647, 53752, 53748, 55749

*SHELL SECTION,MATERIAL=STEEL,ELSET=BLANK

.0008 5

*TRANSVERSE SHEAR STIFFNESS

3.6E+05, 3.6E+05,

**

** Define material properties for BLANK

**

*MATERIAL, NAME=STEEL

*User Material, constants=13

21e+09, 0.3, 0.223, 513E+06, 91.3E+06, 1.0, 1.0E-02
0., 1.0E-05, 200, 0.7, 0.85, 0.6

*Depvar

5,

*DENSITY

7.8000E-09
**
** Define surfaces

**
*SURFACE, NAME = BLANK-TOP, TYPE = ELEMENT
131167, SPOS
131168, SPOS

... in total, 5776 ELEMENTS DEFINED

. 117930, SPOS
117931, SPOS
*SURFACE, NAME = BLANK-BOT, TYPE = ELEMENT
131167, SNEG
131168, SNEG

... in total, 5776 ELEMENTS DEFINED

. 117930, SNEG
117931, SNEG
*SURFACE, NAME = HOLDER-BOT, TYPE = ELEMENT
129169, SNEG
129170, SNEG

... in total, 3248 ELEMENTS DEFINED

. 111442, SNEG
111441, SNEG
*SURFACE, NAME = PUNCH-BOT, TYPE = ELEMENT
129169, SNEG
129170, SNEG

... in total, 780 ELEMENTS DEFINED

. 111357, SNEG
111358, SNEG
*SURFACE, NAME = DIE-TOP, TYPE = ELEMENT
127598, SNEG
127599, SNEG

... in total, 780 ELEMENTS DEFINED

. 129167, SNEG
129168, SNEG

**
** Apply symmetry boundary conditions

**
**BLANK SYMMETRY**

*BOUNDARY
55753, ZSYMM
53469, ZSYMM
.
... in total, 51 NODES DEFINED
.
53512, ZSYMM
53507, ZSYMM
*BOUNDARY
53444, XSYMM
53460, XSYMM
.
... in total, 51 NODES DEFINED
.
53459, XSYMM
53463, XSYMM

**

**HOLDER BOUNDARY**

*BOUNDARY
67067, 1,1
67067, 3,6
*BOUNDARY
67510, 3,3
67509, 3,3
.
... in total, 36 NODES DEFINED
.
65517, 3,3
65512, 3,3
*BOUNDARY
67604, 1,1
67603, 1,1
.
... in total, 36 NODES DEFINED
.
65461, 1,1
65456, 1,1

**

** PUNCH BOUNDARY**
*BOUNDARY
52523, 1,1
52523, 3,6

*BOUNDARY
53303, 3,3
53302, 3,3
.
... in total, 36 NODES DEFINED
.
52370, 3,3
52346, 3,3

*BOUNDARY
53199, 1,1
53198, 1,1
.
... in total, 36 NODES DEFINED
.
52450, 1,1
52427, 1,1

**
** DIE BOUNDARY
**

*BOUNDARY
59617, 1,6

*BOUNDARY
63904, 3,3
63905, 3,3
.
... in total, 36 NODES DEFINED
.
63902, 3,3
63903, 3,3

*BOUNDARY
63937, 1,1
63938, 1,1
.
... in total, 36 NODES DEFINED
.
63935, 1,1
63936, 1,1

**
** Simulate the deep drawing operation
**

*RESTART,WRITE,NUMBER INTERVAL=2,TIMEMARKS=NO

**
*STEP
*DYNAMIC,EXPLICIT
,.0025
*AMPLITUDE,DEFINITION=SMOOTH STEP,NAME=SMOOTH
0., 0., 0.0025, 1.
*BOUNDARY,AMPLITUDE=SMOOTH
52523.2, -25.0
*CLOAD
67067.2, -12050.0
**
*SURFACE INTERACTION,NAME=PUNCH-BLANK
*FRICITION
0.3,
*CONTACT PAIR, INTERACTION=PUNCH-BLANK
BLANK-TOP,PUNCH-BOT
*CONTACT PAIR
BLANK-TOP,HOLDER-BOT
*SURFACE INTERACTION,NAME=DIE-BLANK
*FRICITION
0.25
BLANK-TOP,HOLDER-BOT
*CONTACT PAIR, INTERACTION=DIE-BLANK
BLANK-BOT,DIE-TOP
**
** Output requests for postprocessing
**
*OUTPUT,FIELD,VAR=PRESELECT
*ELEMENT OUTPUT
STH, PEEQ, SDV
*OUTPUT,HISTORY,TIME INTERVAL=0.7E-4
*NODE OUTPUT,NSET=NPUCH
U3, RF3
*NODE OUTPUT,NSET=HOLDER
U3,
*ELEMENT OUTPUT,ELSET=ELHIST
MISES, STH
*END STEP
Appendix C

User Material Subroutine for ABAQUS/Explicit

SUBROUTINE VUMAT()
C Read only (unmodifiable) variables -
  1 nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
  2 stepTime, totalTime, dt, cmname, coordMp, charLength,
  3 props, density, strainInc, relSpinInc,
  4 tempOld, stretchOld, defgradOld, fieldOld,
  5 stressOld, stateOld, enerInternOld, enerInelasOld,
  6 tempNew, stretchNew, defgradNew, fieldNew,
C Write only (modifiable) variables -
  7 stressNew, stateNew, enerInternNew, enerInelasNew
C
  include 'vaba.param.inc'
C
  character*80 cmname
  integer charLen
  dimension props(nprops), density(nblock), coordMp(nblock,*),
  1 charLength(nblock), strainInc(nblock,ndir+nshr),
  2 relSpinInc(nblock,nshr), tempOld(nblock),
  3 stretchOld(nblock,ndir+nshr),
  4 defgradOld(nblock,ndir+nshr+nshr),
  5 fieldOld(nblock,nfieldv), stressOld(nblock,ndir+nshr),
  6 stateOld(nblock,nstatev), enerInternOld(nblock),
  7 enerInelasOld(nblock), tempNew(nblock),
  8 stretchNew(nblock,ndir+nshr),
  9 defgradNew(nblock,ndir+nshr+nshr),
  0 fieldNew(nblock,nfieldv),
  1 stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
  2 enerInternNew(nblock), enerInelasNew(nblock)
C ***************************************************************************
C PROPS: User defined properties (i)
C
C (1) Parameter E, Youngs Modulus
C (2) Parameter nu, Poisson’s ratio
C (3) Parameter n, strain hardening exponent in power law
C (4) Parameter K, strength coefficient in power law
C (5) Parameter SingmaY, yield stress
C (6) User defined integration scheme
C (6.1) Modified Euler (ME)
C (6.2) Runge-Kutta-Dormand-Prince (RKDP)
C (6.3) Forward Euler scheme (FE)
C (7) Integration scheme parameter

141
(7.1) Stress tolerance for ME and RKDP
(7.2) Number of subdivisions for FE
(8) Activate correction for yield surface drift
(8.0) No
(8.1) Yes
(9) Tolerance for yield surface drift
(10) Maximum number of iterations

STATEV: Initial and final state dependent variables (io)
(1) Current value of yield function
(2) Current value of hardening parameter
(3) Current value of effective plastic strain
(4) plastic indicator (0-elastic, 1-plastic)

Function declarations

REAL*8 NORM,YIELD,DHARD
DHARD: Calculate derivative of hardening function
NORM: Calculate vector norm
YIELD: Calculate value of yield function

Definition of variables defined within the subroutine

real*8 dfdw,dummy,diff,ef,es,ftol,g,ratio,stol,te,me
INTEGER ACTIVEDRIFT,EP,I,IMAX,INTSTEP,ITRD,ITRI,J,M,METHOD,K,
1 REDUCTION,count
REAL*8 CE(4,4),CP(4,4),DESTRESS(4),STATEV(NSTATEV),
1 DEPS(4),DF(4),DINVAR(4,3),DDSDDE(4,4),DSTRAIN(4),
2 ESINVAR(4),ESTRESS(4),SINVAR(3),ISTATEV(NSTATEV),
3 NSTRESS(4),temp(nprops),dees(4)

Initialise user variables
Do i=1,nprops
  temp(i)=props(i)
end Do
METHOD=IDNINT(temp(6))
IF (METHOD.LT.3) THEN
  STOL=temp(7)
ELSE
  M=IDNINT(temp(7))
END IF
ACTIVEDRIFT=IDNINT(temp(8))
FTOL=temp(9)
IMAX=IDNINT(temp(10))

Save stresses and strains into temp arrays
DO K = 1, nblock
  NSTRESS(I)=STRESSOLD(K,I)
  DSTRAIN(I)=STRAININC(K,I)
END DO

Check for zero strain increment. If zero return with elastic stiffness
DO I=1,NSTATEV
  STATEV(I)=STATEOLD(K,I)
END DO

Check whether state dependent variables have been initialised
IF (STATEV(3).EQ.0.D0) THEN
  CALL Initialise(STATEV,TEMP,NSTATEV,NPROPS)
END IF

Perform elastic shooting
DO I=1,4
  DESTRESS(I)=0
DO J=1,4

C Function declarations

REAL*8 NORM,YIELD,DHARD
DHARD: Calculate derivative of hardening function
NORM: Calculate vector norm
YIELD: Calculate value of yield function

Definition of variables defined within the subroutine

real*8 dfdw,dummy,diff,ef,es,ftol,g,ratio,stol,te,me
INTEGER ACTIVEDRIFT,EP,I,IMAX,INTSTEP,ITRD,ITRI,J,M,METHOD,K,
1 REDUCTION,count
REAL*8 CE(4,4),CP(4,4),DESTRESS(4),STATEV(NSTATEV),
1 DEPS(4),DF(4),DINVAR(4,3),DDSDDE(4,4),DSTRAIN(4),
2 ESINVAR(4),ESTRESS(4),SINVAR(3),ISTATEV(NSTATEV),
3 NSTRESS(4),temp(nprops),dees(4)

Initialise user variables
Do i=1,nprops
  temp(i)=props(i)
end Do
METHOD=IDNINT(temp(6))
IF (METHOD.LT.3) THEN
  STOL=temp(7)
ELSE
  M=IDNINT(temp(7))
END IF
ACTIVEDRIFT=IDNINT(temp(8))
FTOL=temp(9)
IMAX=IDNINT(temp(10))

Save stresses and strains into temp arrays
DO K = 1, nblock
  NSTRESS(I)=STRESSOLD(K,I)
  DSTRAIN(I)=STRAININC(K,I)
END DO

Check for zero strain increment. If zero return with elastic stiffness
DO I=1,NSTATEV
  STATEV(I)=STATEOLD(K,I)
END DO

Check whether state dependent variables have been initialised
IF (STATEV(3).EQ.0.D0) THEN
  CALL Initialise(STATEV,TEMP,NSTATEV,NPROPS)
END IF

Perform elastic shooting
DO I=1,4
  DESTRESS(I)=0
DO J=1,4
DESTRESS(I) = DESTRESS(I) + DDSDDE(I,J) * DSTRAIN(J)
END DO
END DO

C Calculate the total stresses after elastic shooting
DO I = 1, 4
ESTRESS(I) = NSTRESS(I) + DESTRESS(I)
END DO
CALL INVARIANTS(ESINVAR, ESTRESS)
EF = YIELD(ESINVAR, TEMP, NPROPS)
DIFF = EF - STATEV(2)
IF (DIFF.OT. . F TOL) THEN
DO I = 1, 4
NSTRESS(I) = ESTRESS(I)
END DO
STATEV(1) = EF
ELSE
C Check if the yield surface is crossed during loading
IF (METHOD.EQ.4) THEN
CALL RREULER(NSTRESS, ESTRESS, DESTRESS, STATEV, TEMP,
IMAX, ITRI, NSTATEV, NPROPS, DIFF, FTOL)
IF (ITRI.GE.imax) CALL DUMP(STATEV, NSTATEV, 3, K)
ELSE
ITRI = 0
RATIO = 0.0
C IF ((EF-STATEV(1)).GT.FTOL) THEN
IF (IDNINT(statev(4)).EQ.0) THEN
CALL INTERSECTION(NSTRESS, RATIO, STATEV, DESTRESS,
EF, TEMP, FTOL, IMAX, ITRI, NSTATEV, NPROPS)
IF (ITRI.GE.49) CALL DUMP(STATEV, NSTATEV, 1, K)
END IF
DO I = 1, 4
DEPS(I) = (1.0-RATIO) * DSTRAIN(I)
END DO
DO I = 1, NSTATEV
ISTATEV(I) = STATEV(I)
END DO
INTSTEP = 0
REDUCTION = 0
IF (METHOD.EQ.1) THEN
CALL MEULER(NSTRESS, STATEV, ISTATEV, DEPS,
TEMP, STOL, INTSTEP, REDUCTION, NSTATEV, NPROPS)
ELSE IF (METHOD.EQ.2) THEN
CALL RKDP(NSTRESS, STATEV, DEPS, TEMP,
STOL, INTSTEP, REDUCTION, NSTATEV, NPROPS)
ELSE IF (METHOD.EQ.3) THEN
CALL FEULER(NSTRESS, DEPS, TEMP, M,
NSTATEV, NPROPS, DIFF)
ENDIF
ENDIF
IF (ACTIVEDRIFT.EQ.1) THEN
C Correcting yield surface drift
IF ((ABS(STATEV(1)-STATEV(2))).GT.FTOL) THEN
CALL INVARIANTS(SINVAR, NSTATEV)
CALL ELASTIC(CE, TEMP, NPROPS)
ITRD = 0
CALL DRIFT(NSTRESS, STATEV, CE, TEMP, FTOL,
IMAX, ITRD, NSTATEV, NPROPS)
IF (ITRD.GE.IMAX) CALL DUMP(STATEV, NSTATEV, 2, K)
ENDIF
ENDIF
END IF
statev(4) = 1.0
END IF
statev(5) = statev(5) + intstep

C Update the stresses
DO I = 1, 4
    STRESSNEW(K, I) = NSTRESS(I)
END DO
DO I = 1, NSTATEV
    STATENEW(K, I) = STATEV(I)
END DO

C Update the thickness strain
IF (IDNINT(STATEV(4)).EQ.0) THEN
    strainInc(K, 3) = -temp(2)*((STRAININC(K, 1) + STRAININC(K, 2))/
    1 *(1-temp(2)))
ELSE
    DEES(1) = STRESSNEW(K, 1)/TEMP(1)-STRESSOLD(K, 1)/TEMP(1)
    DEES(2) = STRESSNEW(K, 2)/TEMP(1)-STRESSOLD(K, 2)/TEMP(1)
    DEPS(1) = STRAININC(K, 1)-DEES(1)
    DEPS(2) = STRAININC(K, 2)-DEES(2)
    strainInc(K, 3) = -temp(2)*(DEES(1)+DEES(2))/(1-temp(2))
    1 -(DEPS(1)+DEPS(2))
END IF
END DO
RETURN
END

C SUBROUTINE INITIALISE(svar, mat, nsv, np)
C Initialise state dependent variables.
C
C INPUT
C invar: Stress invariants
C mat: Material properties
C np: Number of user defined properties
C nsv: Number of state dependent variables
C
C OUTPUT
C svar: Initial state dependent variables
C
SUBROUTINE INITIALISE(svar, mat, nsv, np)
C Define primary variables
IMPLICIT NONE
INTEGER nsv, np
REAL*8 svar(nsv), mat(np)
C Call statev(2) when statev(3) is zero
svar(2) = mat(5)
END

C SUBROUTINE INVARIANTS(inv,s)
C Calculate stress invariants
C
C INPUT
C s: Stresses [s11 s22 s33 s12]
C
C OUTPUT
C inv: Stress invariants [J1 J2 J3]
C
SUBROUTINE INVARIANTS(inv,s)
C Define primary variables
IMPLICIT NONE
REAL*8 s(4), inv(4), SX, SY, SZ
inv(1)=s(1)+s(2)+s(3)
SX=S(1)-inv(1)/3
SY=S(2)-inv(1)/3
SZ=S(3)-inv(1)/3
inv(2)=0.5D0*(SX**2.0+SY**2.0+SZ**2.0)+S(4)**2.0
inv(3)=SZ*(SZ**2.0-inv(2))
END

SUBROUTINE DINVARIANTS(dinv, inv, s)
C Calculate derivatives of stress invariants with respect to stresses.
C
C INPUT
C s: Stresses [s11 s22 s33 s12 s23 s31]
C
C OUTPUT
C dinv: derivative of stress invariants [a1 a2 a3]
C
C Define primary variables
IMPLICIT NONE
REAL*8 dinv(4,3), s(4), SX, SY, SZ, sj2, inv(4)
sj2=sqrt(inv(2))
SX=S(1)-inv(1)/3
SY=S(2)-inv(1)/3
SZ=S(3)-inv(1)/3
C Calculate derivative of the first stress invariant
dinv(1,1)=1.0D0
dinv(2,1)=1.0D0
dinv(3,1)=1.0D0
dinv(4,1)=0.0D0
C Calculate derivative of the second stress invariant
dinv(1,2)=SX/(2.0D0*sj2)
dinv(2,2)=SY/(2.0D0*sj2)
dinv(3,2)=SZ/(2.0D0*sj2)
dinv(4,2)=s(4)/sj2
C Calculate derivative of the third stress invariant
dinv(1,3)=SY*SZ+inv(2)/3.0D0
dinv(2,3)=SX*SZ+inv(2)/3.0D0
dinv(3,3)=SX*SY-s(4)**2.0D0+inv(2)/3.0D0
dinv(4,3)=-2.0D0*SZ*s(4)
END

FUNCTION HARD(svar, nsv, mat, np)
C Calculate the value of the hardening function
C
C OUTPUT
C HARD: Value of the hardening function
C
C Define primary variables
IMPLICIT NONE
INTEGER np, nsv
REAL*8 eps, mat(np), HARD, K, n, svar(nsv), sigmaY
sigmaY=mat(5)
eps=svar(3)
K=mat(4)
n=mat(3)
C Calculate value of hardening function
if (eps.eq.0.D0) then
  HARD=sigmaY
else
  HARD=K*(eps)**n
endif
END

Czeigen**************
C FUNCTION DHARD(eps,mat,np)
C Calculate the derivative of the hardening function wrt plastic strain
C at the current state of stress
C
C INPUT
C eps: Plastic work
C mat: Curvature parameter for failure criteria
C np: Number of user defined properties
C
C OUTPUT
C DHARD: Derivative of the hardening function
C
FUNCTION DHARD(eps,mat,np)
C Define primary variables
IMPLICIT NONE
INTEGER np
REAL*8 eps,mat(np),DHARD,K,n,p
C Define secondary variables C Calculate dependent parameters
K=mat(4)
n=mat(3)
C Calculate derivative of the hardening function
if (eps.eq.0.D0) then
  DHARD=0.D0
else
  p=eps**(n-1)
  DHARD=K*n*p
endif
END

Czeigen**************
C FUNCTION YIELD(inv,mat,np)
C Calculate the value of the yield function
C
C INPUT
C inv: Stress invariants [I1 I2 I3]
C s: Stress level relative to failure
C np: Number of user defined properties
C
C OUTPUT
C YIELD: Value of the yield function
C
FUNCTION YIELD(inv,mat,np)
C Define primary variables
IMPLICIT NONE
INTEGER np
REAL*8 inv(4),mat(np),YIELD
C Calculate dependent parameter C Calculate value of yield function
YIELD=sqrt(3*inv(2))

Czeigen**************
C SUBROUTINE DYIELD(df,inv,dinv,mat,s,f,np)
C Calculate the derivative of the yield function.
APPENDIX C. USER MATERIAL SUBROUTINE FOR ABAQUS/EXPLICIT

C INPUT
C inv: Stress invariants [I1 I2 I3]
C dinv: Derivatives of stress invariants [dI1 dI2 dI3]
C
C OUTPUT
C df: Derivative of the yield function [aT]
C
SUBROUTINE DYIELD(df,inv,dinv,mat,np)
C Define primary variables
IMPLICIT NONE
INTEGER np
REAL*8 inv(4),dinv(4,3),mat(np),df(4)
C Define secondary variables
INTEGER i
REAL*8 dfdIl,dfdI2,dfdI3
C Derivative of the exponent q with respect to the first stress invariant
dfdIl=0.D0
dfdI2=sqrt(3.D0)
dfdI3=0.D0
C The derivative of the yield function with respect to stress is obtained by use C of the chain rule
DO i=1,4
df(i)=dfdIl*dinv(i,1)+dfdI2*dinv(i,2)+dfdI3*dinv(i,3)
END DO
END

SUBROUTINE ELASTIC(C,inv,mat,np)
C Calculate the isotropic elastic stiffness matrix.
C
C INPUT
C inv: Stress invariants [I1 I2 I3]
C mat(1): Parameter E, Young's Modulus
C mat(2): Parameter nu, Poisson's ratio
C np: Number of user defined properties
C: Isotropic elastic stiffness matrix C
SUBROUTINE ELASTIC(C,mat,np)
C Define primary variables
IMPLICIT NONE
INTEGER np
REAL*8 mat(np),C(4,4)
C Define secondary variables
INTEGER i,j
REAL*8 E,v, terml, term2
C Calculate constants
E=mat(1)
v=mat(2)
term1=E/(1.D0-v*v)
C Assemble elastic stiffness matrix
DO i=1,4
DO j=1,4
C(i,j)=0.D0
END DO
END DO
C(1,1)=term1
C(2,2)=term1
C(4,4)=(1.D0-v)*term1/2.D0
C(1,2)=term1*v
C(2,1)=term1*v
END
SUBROUTINE PLASTIC(Cp,dl,Ce,df,mat,deps,np,dfsw,des)
C Calculate the plastic matrix and the plastic proportionality factor.
C
C INPUT
C Ce: Elastic constitutive matrix
C df: Derivative of yield function with respect to stresses
C deps: Strain increment
C
C OUTPUT
C Cp: Plastic constitutive matrix
C dl: Plastic proportionality factor
C
SUBROUTINE PLASTIC(Cp,dl,Ce,df,mat,deps,np,dfsw,des)
C Define primary variables
IMPLICIT NONE
INTEGER i,j,np
REAL*8 Cp(4,4),dl,Ce(4,4),df(4),mat(np),deps(4)
C Define secondary variables
REAL*8 H,Cedg(4),Cedf(4),dfCedg,dfCedeps,Cedeps(4),des,b,
1 NORM,dfsw,bv(4)
C Calculate hardening modulus
DO i=1,3
    bv(i)=DSQRT(2.D0)*df(i)
END DO
bv(4)=df(4)
b=NORM(bv,4,1.D0)
H=dfsw*b
C Calculate tensor product (Ce)(dg)
DO i=1,4
    Cedg(i)=0.D0
END DO
DO i=1,4
    DO j=1,4
        Cedg(i)=Cedg(i)+Ce(i,j)*df(j)
    END DO
END DO
C Calculate scalar (df)(Ce)(dg)
dfCedg=0.D0
DO i=1,4
    dfCedg=dfCedg+Cedg(i)*df(i)
END DO
C Calculate tensor product (Ce)(df)
DO i=1,4
    Cedf(i)=0.D0
END DO
DO i=1,4
    DO j=1,4
        Cedf(i)=Cedf(i)+Ce(i,j)*df(j)
    END DO
END DO
C Assemble the plastic constitutive matrix
DO i=1,4
    DO j=1,4
        Cp(i,j)=Cedg(i)*Cedf(j)/ (dfCedg+H)
    END DO
END DO
C Update the effective strain in second state variable
DO i=1,4

Cedeps(i)=0.D0
END DO
DO i=1,4
  DO j=1,4
    Cedeps(i)=Cedeps(i)+Ce(i,j)*deps(j)
  END DO
END DO

C Calculate scalar \((df)(Ce)(deps)\)

\[
dfCedeps=0.D0
\]
DO i=1,4
  dfCedeps=dfCedeps+Cedeps(i)*df(i)
END DO

C Calculate plastic proportionality factor

\[
dl=\text{DMAX1}(0.D0,\frac{dfCedeps}{(dfCedg+H)})
\]
\[
des=\text{dl}^b
\]
END

C SUBROUTINE INTERSECTION(s, alpha, svar, dse, fe, mat, tol, im, i, np, nsv)
C Determine ratio of strain increment causing elastic deformations.

C INPUT
C svar: State dependent variables
C (1) Current value of yield function
C (2) Current value of hardening or softening function
C s: Initial stress state
C dse: Elastic stress increment due to elastic shooting
C fe: Value of yield function after elastic shooting
C mat: Material properties
C tol: Tolerance
C im: Maximum iterations allowed
C np: Number of user defined properties
C nsv: Number of state dependent variables

C OUTPUT:
C s: Final stress state
C alpha: Ratio of strain increment causing purely elastic deformation
C svar: Updated values of state dependent variables
C (1): Final value of yield function (this should equal to svar(2))
C (7): Final stress level
C i: Iterations performed

SUBROUTINE INTERSECTION(s, alpha, svar, dse, fe, mat, tol, im, i, np, nsv)

C Define primary variables
IMPLICIT NONE
INTEGER nsv, np, im, i
REAL*8 s(4), alpha, svar(nsv), s0(4), dse(4), fe, mat(np), tol

C Define secondary variables
INTEGER j
REAL*8 inv(4), dinv(4,3), dfds(4), dalpha, error, fa, s1(4),
  1 dfdse, YIELD, HARD

C Calculate first estimate of elastic strain increment
fa=svar(1)-svar(2)
alpha=-fa/(fe-svar(1))

C Perform compatibility check for first estimate
DO j=1,4
  s0(j)=s(j)
s(j)=s0(j)+alpha*dse(j)
END DO
CALL INVARANTS(inv, s)
svar(1)=YIELD(inv,mat,np)
error=svar(1)-svar(2)
C Compatibility not obtained, perform iterations
i=0
DO WHILE ((abs(error).GT.tol).AND.(i.LT.im))
i=i+1
C Calculate derivatives of stress invariants
CALL DINVARIANTS(dinv,inv,s)
C Calculate derivatives of the yield function with respect to stresses
CALL DYIELD(dfdss,inv,dinv,mat,np)
C Calculate vector product dfdss dse
dfdse=0.D0
DO j=1,4
  dfdse=dfdse+dfdss(j)*dse(j)
END DO
dalpapa=-error/dfdse
alpha=alpha+dalpapa
C Calculate updated stresses and perform tolerance check
DO j=1,4
  s(j)=s0(j)+alpha*dse(j)
END DO
CALL INVARIANTS(inv,s)
svar(1)=YIELD(inv,mat,np)
error=svar(1)-svar(2)
END DO
END
C SUBROUTINE DRIFT(s,svar,C,mat,tol,im,i,nsv,np)
C Correction for yield surface drift.
C C INPUT
C s: Initial stress state
C svar: Initial value of state dependent variables
C (1) Initial value of yield function
C (2) Initial value of hardening function
C (3) Initial plastic work
C C: Elastic constitutive matrix
C mat: Material properties
C tol: Tolerance
C im: Maximum number of iterations
C np: Number of user defined properties
C nsv: Number of state dependent variables
C C C OUTPUT
C s: Final stress state
C svar: Updated state dependent variables
C (1) Final value of yield function
C (2) Final value of hardening function
C (3) Final plastic work
C (7) Final stress level
C i: Iterations performed
C SUBROUTINE DRIFT(s,svar,C,mat,tol,im,i,nsv,np)
C Define primary variables
IMPLICIT NONE
INTEGER nsv,np,i,im,j
REAL*8 s(4),svar(nsv),mat(np),tol, C(4,4)
C Define secondary variables
REAL*8 inv(4),dinv(4,3),ds(4), df(4), dfdw,DHARD,
APPENDIX C. USER MATERIAL SUBROUTINE FOR ABAQUS/EXPLICIT

1 alpha, error, DA(4), aDaT, b, deps, H, bv(4)
C Function declarations
REAL*8 YIELD, NORM, HARD
C Initialise local variables
i = 0
C check if f=0
CALL INVARIANTS(inv, s)
CALL DINVARIANTS(dinv, inv, s)
CALL DYIELD(df, inv, dinv, mat, np)
C Compute D*a
DO i = 1, 4
   DA(i) = 0.D0
END DO
DO i = 1, 4
   DO j = 1, 4
      DA(i) = DA(i) + C(ij)*df(j)
   END DO
END DO
C Compute a*D*aT
aDaT = 0.D0
DO i = 1, 4
   aDaT = aDaT + DA(i)*df(i)
END DO
DO i = 1, 3
   bv(i) = DSQRT(2.D0)*df(i)
END DO
bv(4) = df(4)
b = NORM(bv, 4, 1.D0)
C Check f
error = svar(1) - svar(2)
DO WHILE((abs(error).GT.tol).AND.(i.LE.im))
i = i + 1
C Calculate alpha
dfdw = DHARD(svar(3), mat, np)
H = dfdw * b
alpha = (error)/(aDaT + H)
deps = alpha * b
C Update effective plastic strain
svar(3) = svar(3) + deps
C Update the stress
DO i = 1, 4
   s(i) = s(i) - alpha * DA(i)
END DO
C Check if F=0 is met
CALL INVARIANTS(inv, s)
svar(1) = YIELD(inv, mat, np)
svar(2) = HARD(svar, nsv, mat, np)
error = svar(1) - svar(2)
END DO
END
C******************************************************************************
C SUBROUTINE MEULER(s, svar, isvar, deps, mat, tol, r, j, nsv, np)
C Update stresses by the modified Euler scheme
C
C INPUT
C s: Initial stress state
C isvar: Initial values of state dependent variables
C deps: Strain increment C mat: Material properties
C tol: Tolerance

C nsv: Number of state dependent variables
C np: Number of user defined properties
C
C OUTPUT
C s: Final stress state
C svar: Updated values of state dependent variables
C r: Number of substeps used for integration
C j: Number of corrections to the substep size
C
SUBROUTINE MEULER(s,svar,isvar.deps,mat,tol,r,j,nsv,np)
C Define primary variables
IMPLICIT NONE
INTEGER nsv,np,r,j
REAL*8 s(4),s0(4),svar(nsv),isvar(nsv),deps(4),mat(np),tol

C Define secondary variables
INTEGER p,t
REAL*8 q,dds(4,2),sumddeps(4),ddeps(4),error,ddes(2),
1 inv(4),dinv(4,3),df(4),dfdw,mindeps,resdeps(4),
2 dls1(4),ddepsp(4),ddds(4),es(2),DHARD,
3 destrain1,destrain2,Cp(4,4),Ce(4,4),HARD,des,temps0,sb

C Function declarations
REAL*8 NORM,YIELD

C Initialise local variables
r=0
j=0
q=1.0D0
DO p=1,4
  DO t=1,2
    dds(p,t)=0.D0
  END DO
  s0(p)=s(p)
  sumddeps(p)=0.D0
  ddeps(p)=deps(p)
  resdeps(p)=deps(p)
END DO
temps0=svar(3)
error=1
DO WHILE (NORM(resdeps,4,1.0D0).GT.tol)
  r=r+1
  C Determined size of next strain increment
  mindeps=NORM(ddeps,4,q)
  IF (NORM(resdeps,4,1.0D0).LT.mindeps) THEN
    DO p=1,4
      ddeps(p)=resdeps(p)
    END DO
  ELSE
    DO p=1,4
      ddeps(p)=q*ddeps(p)
    END DO
  END IF
  DO WHILE (error.GT.tol)
    DO p=1,nsv
      svar(p)=isvar(p)
    END DO
    CALL INVARIANTS(inv,s0)
    CALL DINVARIANTs(dinv,inv,s0)
    CALL DYIELD(df,inv,dinv,mat,np)
    dfdw=DHARD(svar(3),mat,np)
    C Calculate first estimate of stress increment
    CALL INVARNTS(inv,s0)
    CALL DINVARIANTS(dinv,inv,s0)
    CALL DYIELD(df,inv,dinv,mat,np)
    C Calculate derivative of hardening or softening law
    dfdw=DHARD(svar(3),mat,np)
CALL ELASTIC(Ce,mat,np)
CALL PLASTIC(Cp,dl,Ce,df,mat,ddeps,np,dfdw,des)

C Calculate first estimate of stress increment, plastic strain and plastic work
DO p=1,4
  dds(p,1)=0.D0
  DO t=1,4
    dds(p,1)=dds(p,1)+(Ce(p,t)-Cp(p,t))*ddeps(t)
  END DO
  sl(p)=s0(p)+dds(p,1)
END DO

detrainl=des

C Update state variables due to first estimate of stress increment
CALL INVARIANTS(inv, sl)
svar(3)=svar(3)+detrainl

C Calculate second estimate of stress increment
C First calculate derivatives of yield and plastic potential function
CALL DINVARINANTS(dinv, inv, sl)
CALL DYIELD(df, inv, dinv, mat, np)
dfdw=DHARD(svar(3), mat, np)
CALL ELASTIC(Ce, mat, np)
CALL PLASTIC(Cp, dl, Ce, df, mat, ddeps, np, dfdw, des)
DO p=1,4
  dds(p,2)=0.D0
  DO t=1,4
    dds(p,2)=dds(p,2) + (Ce(p,t)-Cp(p,t))*ddeps(t)
  END DO
  s(p)=s0(p)+0.5D0*(dds(p,1)+dds(p,2))
END DO

C Update the second effective strain estimate
detrain2=des
DO p=1,4
  dddels(p)=0.5D0*(dds(p,2)-dds(p,1))
END DO

error=NORM(dddels,4,1.0D0)/NORM(s,4,1.0D0)
IF (error.GT.tol) THEN
  q=DMAX1((0.9D0*DSQRT(tol/error)),0.01D0)
  DO p=1,4
    ddeps(p)=ddeps(p)*q
  END DO
  j=j+1
ELSE
  svar(3)=temps0+0.5D0*(detrain1+detrain2)
END IF
END DO

CALL INVARIANTS(inv, s)
svar(1)=YIELD(inv, mat, np)
svar(2)=HARD(svar, nsv, mat, np)
C subroutine RKDP(s,svar,isvar,deps,mat,tol,r,j,nsv,np)
C Update stresses by the Runge-Kutta-Dormand-Prince scheme.

C INPUT
C s: Initial stress state
C isvar: Initial values of state dependent variables
C deps: Strain increment
C mat: Material properties
C tol: Tolerance
C nsv: Number of state dependent variables
C np: Number of user defined properties

C OUTPUT
C s: Final stress state
C svar: Updated values of state dependent variables
C r: Number of substeps used for integration
C j: Number of corrections to the substep size

SUBROUTINE RKDP(s,svar,isvar,deps,mat,tol,r,j,nsv,np)

C Define primary variables
IMPLICIT NONE
INTEGER nsv,np,r,j
REAL*8 s(4),s0(4),svar(nsv),isvar(nsv),deps(4),mat(np),tol

C Define secondary variables
INTEGER i,p,t
REAL*8 ddeps(4),ddepsp(4),sumddeps(4),resdeps(4),minddeps,q,
1 si(4),s2(4),ds(4),dds(4),swei(4),inv(4),dinv(4,3),error,
2 ddwp(4),dss(4),des,eps,ieps(6),HARD,
3 df(4),dfdw,Cp(4,4),Ce(4,4),dl,coef(6),coefl(6),coef2(6),HARD,
4 drepr,tempO

C Function declarations
REAL*8 NORM,YIELD

C Initialise local variables
r=0
j=0
q=1.0D0
DO p=1,4
   DO t=1,6
      ds(p,t)=0.D0
      ieps(t)=0.D0
   END DO
s0(p)=s(p)
sumddeps(p)=0.D0
ddeps(p)=deps(p)
resdeps(p)=deps(p)
END DO
error=1
DO WHILE (NORM(resdeps,4,1.0D0).GT.0)
   r=r+1
END DO

C Determined size of next strain increment
minddeps=NORM(ddeps,4,q)
IF (NORM(resdeps,4,1.0D0).LT.minddeps) THEN
   DO p=1,4
      ddeps(p)=resdeps(p)
   END DO
ELSE
   DO p=1,4

APPENDIX C. USER MATERIAL SUBROUTINE FOR ABAQUS/EXPLICIT

END

C ******************************************************
C SUBROUTINE RKDP(s,svar,isvar,deps,mat,tol,r,j,nsv,np)
C Update stresses by the Runge-Kutta-Dormand-Prince scheme.
C
C INPUT
C s: Initial stress state
C isvar: Initial values of state dependent variables
C deps: Strain increment
C mat: Material properties
C tol: Tolerance
C nsv: Number of state dependent variables
C np: Number of user defined properties
C
C OUTPUT
C s: Final stress state
C svar: Updated values of state dependent variables
C r: Number of substeps used for integration
C j: Number of corrections to the substep size
C
C SUBROUTINE RKDP(s,svar,isvar,deps,mat,tol,r,j,nsv,np)

C Define primary variables
IMPLICIT NONE
INTEGER nsv,np,r,j
REAL*8 s(4),s0(4),svar(nsv),isvar(nsv),deps(4),mat(np),tol

C Define secondary variables
INTEGER i,p,t
REAL*8 ddeps(4),ddepsp(4),sumddeps(4),resdeps(4),minddeps,q,
1 si(4),s2(4),ds(4),dds(4),swei(4),inv(4),dinv(4,3),error,
2 ddwp(4),dss(4),des,eps,ieps(6),HARD,
3 df(4),dfdw,Cp(4,4),Ce(4,4),dl,coef(6),coefl(6),coef2(6),HARD,
4 drepr,tempO

C Function declarations
REAL*8 NORM,YIELD

C Initialise local variables
r=0
j=0
q=1.0D0
DO p=1,4
   DO t=1,6
      ds(p,t)=0.D0
      ieps(t)=0.D0
   END DO
s0(p)=s(p)
sumddeps(p)=0.D0
ddeps(p)=deps(p)
resdeps(p)=deps(p)
END DO
error=1
DO WHILE (NORM(resdeps,4,1.0D0).GT.0)
   r=r+1
END DO

C Determined size of next strain increment
minddeps=NORM(ddeps,4,q)
IF (NORM(resdeps,4,1.0D0).LT.minddeps) THEN
   DO p=1,4
      ddeps(p)=resdeps(p)
   END DO
ELSE
   DO p=1,4

APPENDIX C. USER MATERIAL SUBROUTINE FOR ABAQUS/EXPLICIT

\begin{verbatim}
  ddeps(p) = q * ddeps(p)
END DO
END IF
DO WHILE (error.GT.tol)
  p = 1, nsv
  svar(p) = isvar(p)
END DO

C Calculate estimates of stress increments 1-4
DO i = 1, 6
  C First establish coefficient vectors depending on the stress increment
  IF (i.EQ.1) THEN
    DO p = 1, 6
      coef(p) = 0.D0
    END DO
  ELSEIF (i.EQ.2) THEN
    coef(1) = 1.D0/5.D0
    coef(2) = 0.D0
    coef(3) = 0.D0
    coef(4) = 0.D0
    coef(5) = 0.D0
    coef(6) = 0.D0
  ELSEIF (i.EQ.3) THEN
    coef(1) = 3.D0/40.D0
    coef(2) = 9.D0/40.D0
    coef(3) = 0.D0
    coef(4) = 0.D0
    coef(5) = 0.D0
    coef(6) = 0.D0
  ELSEIF (i.EQ.4) THEN
    coef(1) = 44.D0/45.D0
    coef(2) = -56.D0/15.D0
    coef(3) = 32.D0/9.D0
    coef(4) = 0.D0
    coef(5) = 0.D0
    coef(6) = 0.D0
  ELSEIF (i.EQ.5) THEN
    coef(1) = 19372.D0/6561.D0
    coef(2) = -25360.D0/2187.D0
    coef(3) = 64448.D0/6561.D0
    coef(4) = -212.D0/729.D0
    coef(5) = 0.D0
    coef(6) = 0.D0
  ELSEIF (i.EQ.6) THEN
    coef(1) = 9017.D0/3168.D0
    coef(2) = -355.D0/33.D0
    coef(3) = 46732.D0/5247.D0
    coef(4) = 49.D0/176.D0
    coef(5) = -5103.D0/18656.D0
    coef(6) = 0.D0
  END IF
END IF

C Calculate stress increments and temporary weighted update of stresses and plastic strain
DO p = 1, 4
  swei(p) = s0(p)
END DO
DO t = 1, 6
  swei(p) = swei(p) + ds(p, t) * coef(t)
END DO
END DO

eps = svar(3)
\end{verbatim}
DO p=1,6
    eps=eps+ieps(p)*coef(p)
END DO

C Calculate stress invariants and derivatives of stress invariants
CALL INVARIANTS(inv,swei)
CALL DINVARIANTS(dinv,inv,swei)

C Calculate derivatives of yield, plastic potential and hardening functions
CALL DYIELD(df,inv,dinv,mat,np)
dfdw=DHARD(eps,mat,np)
CALL ELASTIC(Ce,mat,np)
CALL PLASTIC(Cp,dl,Ce,df,mat,ddeps,np,dfdw,des)
DO p=1,4
    ds(p,i)=0.D0
DO t=1,4
    ds(p,i)=ds(p,i)+(Ce(p,t)-Cp(p,t))*ddeps(t)
END DO
END DO
ieps(i)=des
END DO

C Weights for first and second estimate of stresses
coef1(1)=31.D0/540.D0
coef1(2)=0.D0
coef1(3)=190.D0/297.D0
coef1(4)=-145.D0/108.D0
coef1(5)=351.D0/220.D0
coef1(6)=1.D0/20.D0
coef2(1)=19.D0/216.D0
coef2(2)=0.D0
coef2(3)=1000.D0/2079.D0
coef2(4)=-125.D0/216.D0
coef2(5)=81.D0/88.D0
coef2(6)=5.D0/56.D0

C First and second estimate of stresses and plastic strain
DO p=1,4
    s1(p)=s0(p)
s2(p)=s0(p)
DO t=1,6
    s1(p)=s1(p)+ds(p,t)*coef1(t)
s2(p)=s2(p)+ds(p,t)*coef2(t)
END DO
dds(p)=s2(p)-s1(p)
END DO

error=NORM(dds,4,1.D0)/NORM(s2,4,1.D0)
IF (error.GT.tol) THEN
    q=DMAX1((0.9D0*(tol/error)**(1.D0/5.D0)),0.01D0)
    DO p=1,4
        ddeps(p)=ddeps(p)*q
    END DO
    j=j+1
ELSE
    dreps=0.D0
    DO p=1,6
        dreps=dreps+ieps(p)*coef2(p)
    END DO
    svar(3)=svar(3)+dreps
END IF
END DO
q=DMIN1(0.9D0*(tol/error)**(1.D0/5.D0),2.D0)
error = 1
DO p = 1, 4
    s0(p) = s2(p)
    resdeps(p) = deps(p) - sumddeps(p)
END DO
DO p = 1, nsv
    isvar(p) = svar(p)
END DO
END DO
DO p = 1, 4
    s(p) = s2(p)
END DO
CALL INVARIANTS(inv, s)
svar(1) = YIELD(inv, mat, np)
svar(2) = HARD(svar, nsv, mat, np)
END

C SUBROUTINE FEULER(s, svar, deps, mat, m, nsv, np)
C Update stresses by the forward Euler scheme.
C
C INPUT
C s: Initial stress state
C svar: Initial values of state dependent variables
C dl: Plastic multiplier
C deps: Strain increment
C mat: Material properties
C m: Number of subincrements
C nsv: Number of state dependent variables
C np: Number of user defined properties
C
C OUTPUT
C s: Final stress state
C svar: Updated values of state dependent variables
C
C SUBROUTINE FEULER(s, svar, deps, mat, m, nsv, np, diff)
C Define primary variables
IMPLICIT NONE
INTEGER nsv, np, m
REAL*8 s(4), svar(nsv), mat(np), deps(4), dl, diff
C Function declarations
REAL*8 YIELD
C Define secondary variables
INTEGER i, p, t
REAL*8 ddeps(4), ddepsp(4), s0(4), ds(4), inv(4), inv(4, 3),
       df(4), Ce(4, 4), Cp(4, 4), d, des, dfdw, D HARD, HARD
C Determine the number of subincrements
m = (diff*8)/svar(2)
m = int(m) + 1
C Calculate size of subincrements
DO p = 1, 4
    ddeps(p) = deps(p)/DFLOAT(m)
END DO
CALL INVARIANTS(inv, s)
CALL ELASTIC(Ce, mat, np)
C Perform m subincrements
DO i = 1, m
    C Define initial stress state for current substep
    DO p = 1, 4
APPENDIX C. USER MATERIAL SUBROUTINE FOR
ABAQUSS/EXPLICIT

sO(p)=s(p)
END DO
C Calculate stress invariants and derivative of stress invariants
CALL INVARIANTS(inv,sO)
CALL DINVARIANTS(dinv,inv,sO)
C Calculate derivatives of yield and plastic potential functions
CALL DYIELD(df,inv,dinv,mat,np)
dfdw=DHARD(svar(3),mat,np)
C CALL ELASTIC(Ce,mat,np)
CALL PLASTIC(Cp,dl,Ce,mat,ddeps,dp,dfdwdes)
DO p=1,4
ds(p)=0.D0
DO t=1,4
ds(p)=ds(p)+(Ce(p,t)-Cp(p,t))*ddeps(t)
END DO
s(p)=sO(p)+ds(p)
END DO
svar(3)=svar(3)+des
END DO
svar(2)=HARD(svar,nsv,mat,np)
END

C SUBROUTINE RREuler(olds,s,svar,deps,mat,m,nsv,np)
C Update stresses by the radial return scheme.
C
C INPUT
C s: Initial stress state
C svar: Initial values of state dependent variables
C dh: Plastic multiplier
C deps: Strain increment
C mat: Material properties
C m: Number of subincrements
C nsv: Number of state dependent variables
C np: Number of user defined properties
C
C OUTPUT
C s: Final stress state
C svar: Updated values of state dependent variables
C
SUBROUTINE RREULER(nstress,es,des,svar,mat,im,k,nsv,np,fpred,tol)

IMPLICIT NONE
INTEGER nsv,np,im
REAL*8 olds(4),es(4),svar(nsv),mat(np),dl,tol,s(4)

REAL*8 YIELD,NORM,DHARD,HARD

INTEGER i,J,K
REAL*8 nstress(4),ds(4),inv(4),dinv(4,3),
1 df(4),Ce(4,4),des(4),dfdwd,error,DA(4),y,
2 fpred,aDaT,dlda(4),lambda,Qinv(4,4),QA(4),bv(4),deps,
3 b,H,first

im=200
lambda=0.D0
DO i=1,4
 nstress(i)=es(i)
END DO
CALL ELASTIC(Ce,mat,np)
C check if \( f = 0 \)
CALL INVARINTS(inv,nstress)
CALL DINVARIANTS(dinv.inv.nstress)
CALL DYYIELD(df,inv,dinv,mat,np)
C Compute \( D^a \)
DO i=1,4
   DA(i)=0.D0
END DO
DO i=1,4
   DO j=1,4
      DA(i)=DA(i)+Ce(i,j)*df(j)
   END DO
END DO
C Compute \( a^T \cdot D^a \cdot a^T \)
   aDaT=0.D0
DO i=1,4
   aDaT=aDaT+DA(i)*df(i)
END DO
DO i=1,3
   bv(i)=DSQRT(2.D0)*df(i)
END DO
bv(4)=df(4)
b=NORM(bv,4,1.D0)
C Initialize \( f_0 \)
svar(1)=YIELD(inv,mat,np)
y=svar(1)
svar(2)=HARD(svar,nsv,mat,np)
k=0
error=svar(1)-svar(2)
DO WHILE ((abs(error).GT.tol).AND.(k.LE.im))
   k=k+1
   dfdw=DHARD(svar(3),mat,np)
   H=dfdw*b
   dl=(error)/(aDaT+H)
   deps=dl*b
   C Update effective plastic strain
   svar(3)=svar(3)+deps
   lamda=lamda+dl
C Calculate new ds
   DO i=1,4
      ds(i)=dl*DA(i)
   END DO
C Update the stress
   DO i=1,4
      nstress(i)=nstress(i)-ds(i)
   END DO
   first=y-lamda*aDaT
   svar(2)=HARD(svar,nsv,mat,np)
   error=first-svar(2)
END DO
C ******************************************************
C FUNCTION NORM(vec,n,q)
C Calculate the norm of a vector
C C INPUT
C vec: Vector
C n: Number of elements in vector
C q: Scalar multiplier C C OUTPUT C NORM: Norm of vector
FUNCTION NORM(vec,n,q)
C Define primary variables
IMPLICIT NONE
INTEGER n
REAL*8 vec(n),q,NORM
C Define secondary variables
INTEGER i
C Calculate length
NORM=0.0D0
DO i=1,n
   NORM=NORM+q**2.0D0*vec(i)**2.0D0
END DO
NORM=DSQRT(NORM)
END

SUBROUTINE DUMP(svar,nsv,problem,s,el,gp,n,i)
C Termination of ABAQUS in case of numerical problems
C
C INPUT
C svar: State dependent variables
C nsv: Number of state dependent variables
C problem: Problem identifier
C (1) Problem with intersection of yield surface
C (2) Problem with correction for yield surface drift
C s: Current stress state
C el: Element number
C gp: Gauss point
C n: Current load step
C i: Increment in load step
C
C SUBROUTINE DUMP(svar,nsv,problem,gp)
C Define primary variables
INTEGER nsv,problem,el,gp,n,i
REAL*8 svar(nsv),s(4)
C Define secondary variables
INTEGER j
WRITE(*,*)'******************** ABAQUS RUN IS TERMINATED ',
1 '******************************
   IF (problem.EQ.1) THEN
      WRITE(*,*)'Problem encountered during intersection of yield ',
   1 'surface'
   ELSE IF (problem.EQ.2) THEN
      WRITE(*,*)'Problems encountered during correction for yield ',
   1 'surface drift'
   ELSE
      WRITE(*,*)'Problems encountered during stress integration'
ENDIF
CALL XPLB_EXIT
END