



CRANFIELD UNIVERSITY

J. CAMPBELL

**Lagrangian hydrocode modelling of hypervelocity impact on
spacecraft**

COLLEGE OF AERONAUTICS

PhD THESIS

ProQuest Number: 10820899

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10820899

Published by ProQuest LLC (2019). Copyright of the Dissertation is held by Cranfield University.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346



CRANFIELD UNIVERSITY

COLLEGE OF AERONAUTICS

PhD THESIS

Academic Year 1997–8

J. CAMPBELL

**Lagrangian hydrocode modelling of hypervelocity impact on
spacecraft**

Supervisor: Dr. R. Vignjevic

May 1998

Abstract

This thesis addresses the problem of modelling hypervelocity impact on spacecraft structures using a Lagrangian hydrocode. Lagrangian hydrocodes offer the advantages of computational efficiency and structural elements, but traditionally have been unable to model large material deformations. Analyses of impact on a thin plate were performed using the finite element code DYNA3D. These analyses highlighted three areas where improvement was necessary: material modelling, element erosion criteria and modelling large deformations. To improve the material modelling the SESAME equation of state was implemented in the DYNA3D code. Two new element erosion criteria were then developed, one based on total element deformation, the second on element accuracy. The two criterion were then tested for modelling impacts onto semi-infinite and thin plate targets. The deformation criterion produced the best results for crater size and hole diameter, but can not be used to model the debris cloud. The element accuracy criterion allows a sufficient number of elements to survive to measure debris cloud velocities and spread angle. It was concluded that an alternative method for modelling the debris cloud is required. The Smoothed Particle Hydrodynamics (SPH) method was selected as it is a Lagrangian method, and allows modelling of large deformations as it does not use a computational mesh. The most recent SPH developments require the boundary conditions to be rigorously treated. A penalty contact algorithm for SPH was developed and tested in 1D and 2D. The tests revealed that for successful treatment of boundary conditions it was necessary to address the problem of zero-energy modes. A alternative discretisation method that calculates the velocity and stress at different points was proposed as a cure for the zero-energy mode problem. This method was tested in 1D, and was shown to be a solution to the zero-energy mode problem.

To my parents
Who have always supported me in the choices I have made

Acknowledgements

I am indebted to my supervisor, Dr. Rade Vignjevic, for all the help and advice he gave me throughout this work. Thanks Rade.

Thanks to Cranfield University for supporting me financially during this work.

Thanks to Dr. Larry Libersky for his help with the SPH method.

Sean McLellan and Chris Williams for reading and correcting this thesis.

Finally, all those in the College of Aeronautics who have helped me during my PhD.

Contents

1	Introduction	1
1.1	Structure of the thesis	1
1.2	Hypervelocity Impact	3
1.2.1	The impact process	4
1.2.2	The risk to spacecraft	4
1.2.3	Spacecraft protection	5
1.2.4	Hypervelocity impact research	6
1.3	Hydrocodes	6
1.4	Rationale for work on Lagrangian codes	8
2	The Structure of DYNA3D	9
2.1	Introduction	10
2.1.1	Basic Structure	10
2.1.2	Data handling	11
2.2	Input	12
2.3	Initialisation	13
2.4	Restart	14
2.5	Solution	15
2.5.1	The main solution subroutine	15
2.5.2	8 node hexahedron element subroutines	18
2.5.3	Contact algorithm routines	23
2.6	Output	25
3	Evaluation of DYNA3D	26
3.1	Modelling hypervelocity impact in DYNA3D	27
3.1.1	Contact	29
3.1.2	Boundary and Initial conditions	30
3.1.3	Material Models	30
3.2	Modelling hypervelocity impact without element erosion	31
3.3	Modelling hypervelocity impact with element erosion	32
3.3.1	Suitability of effective plastic strain erosion criterion	32
3.3.2	Contact reliability	33
3.3.3	Modelling multi-plate impact with element erosion	36
3.4	Discussion	39
3.4.1	Improving the material model	39
3.4.2	Improving the element erosion capability	40
3.4.3	Improving the modelling of the debris cloud	40

3.5	Summary	44
4	Implementation of the SESAME equation of state	45
4.1	The SESAME equation of state	46
4.2	Implementation in DYNA3D	47
4.3	Validation of implementation	50
4.4	Summary	53
5	Element Erosion Criteria	54
5.1	Element erosion algorithms	54
5.1.1	Extension of erosion to Johnson-Cook constitutive model . . .	55
5.2	Element strain criterion	56
5.2.1	Choice of criterion	56
5.2.2	Implementation in DYNA3D	56
5.3	Development of non-strain criterion	57
5.3.1	The DYNA3D solid element	58
5.3.2	Element accuracy	58
5.3.3	Implementation of criterion	59
6	Modelling hypervelocity impact in modified DYNA3D	61
6.1	Semi-infinite target tests	61
6.1.1	Effective plastic strain criterion	64
6.1.2	Deformation criterion	64
6.1.3	Accuracy criterion	65
6.1.4	Comparison with experimental data	68
6.1.5	Discussion of semi-infinite impact results.	70
6.2	Thin plate target tests	73
6.2.1	Effective plastic strain criterion	74
6.2.2	Deformation criterion	75
6.2.3	Accuracy criterion	76
6.2.4	Comparison with experimental data	78
6.2.5	Discussion	79
6.3	Discussion	82
7	Smoothed Particle Hydrodynamics	84
7.1	Basic Formulation	85
7.1.1	Kernel Approximation	85
7.1.2	Accuracy of Kernel Approximation	86
7.1.3	Kernels	86
7.1.4	Semi-discretisation of the Conservation Equations	87
7.1.5	Artificial viscosity for SPH	90
7.1.6	Variable Smoothing Length	91
7.1.7	Implementation	91
7.2	SPH Problems	92
7.2.1	Stability	92
7.2.2	Zero energy modes	94
7.3	Proposed Solutions	94

7.3.1	The Tensile Instability Treatment	95
7.3.2	Normalised Smoothing	96
7.3.3	Boundary Conditions	100
7.4	Summary	101
8	Treatment of contact boundary condition in SPH	102
8.1	The SPH code	103
8.2	Contact algorithm	105
8.2.1	Restoring force applied between particle centres.	109
8.2.2	Restoring force applied along average normal	109
8.3	Enforcing the contact condition	111
8.4	Implementation of the contact algorithm	113
8.5	Contact with two particle normals	114
8.6	Tests of contact algorithm	116
8.6.1	1D tests	116
8.6.2	2D tests	128
8.6.3	Discussion	138
8.7	Displaced stress point approach	138
8.7.1	1D algorithm.	139
8.7.2	Boundary conditions	143
8.8	Tests of displaced stress point method	145
8.9	Discussion	153
9	Conclusions	154
A	DYNA3D HEX8 element subroutines	165
B	Finite Element Results	174
B.1	Chapter 3	174
B.2	Chapter 4	179
B.3	Chapter 6	182
B.3.1	Tables	182
B.3.2	Figures	188
C	Derivation of SPH Equations	207
C.1	Conservation of Mass	207
C.2	Conservation of Momentum	208
C.3	Conservation of Energy	209
C.4	Rate-of-Deformation and Spin Tensors	210

List of Figures

2.1	Basic structure of DYNA3D	10
2.2	Structure of main solution subroutine.	17
2.3	Main routines of hexahedron element section, for strength model requiring an equation of state.	19
2.4	Main routines of penalty contact algorithm.	24
3.1	Finite element mesh for impact on thin plate	28
3.2	Effect of completely rectangular mesh on splash	28
3.3	Bumper plate mesh at $0.25 \mu s$. No erosion.	32
3.4	Comparison of shock wave propagation between the contact algorithm with no erosion (top) and with an erosion criterion of 2.5 (bottom). Both plots are at $0.1 \mu s$	34
3.5	Plate and projectile mesh for erosion criterion of 0.75. Time $0.3 \mu s$	35
3.6	Plate and projectile mesh for erosion criterion of 1.0. Time $0.3 \mu s$	35
3.7	Comparison of results using spall type 3(left) and spall type 2 (right), both analyses are for an erosion criterion of 0.5.). Time $0.5 \mu s$	37
3.8	Comparison of results using spall type 3(left) and spall type 2 (right), both analyses are for an erosion criterion of 0.5. Time $2.0 \mu s$	37
3.9	Comparison of results using spall type 3 with erosion criterion 1.0 (left) and spall type 2 with erosion criterion 0.5 (right). Time $0.5 \mu s$	38
3.10	Comparison of results using spall type 3 with erosion criterion 1.0 (left) and spall type 2 with erosion criterion 0.5 (right). Time $2.0 \mu s$	38
4.1	Graphical representation of a SESAME aluminium equation of state	46
4.2	Mesh used for Hugoniot calculation	51
4.3	Comparison of Hugoniot curve for SESAME 3717 with experiment	52
4.4	Comparison of Hugoniot curve for SESAME 3717 with DYNA results	52
6.1	Crater size definitions.	62
6.2	Mesh for semi-infinite impacts.	63
6.3	Results for semi-infinite impact with effective plastic strain criterion	64
6.4	Comparison between Johnson-Cook and Steinberg-Guinan strength models, deformation criterion	65
6.5	Results for deformation erosion criterion, non-equal erosion criteria.	66
6.6	Results for semi-infinite impact, element accuracy criterion, Johnson-Cook strength model	67
6.7	Results for semi-infinite impact, element accuracy criterion, showing effect of varying skew angle. Johnson-Cook strength model	67

6.8	Results for element accuracy criterion, non-equal erosion criteria. . .	68
6.9	Comparison of semi-infinite impact analyses with experimental results.	69
6.10	Comparison between final crater for the effective plastic strain criterion with value 1.5 (left) and the deformation criterion with value 2.5 (right). Both analyses used the Johnson-Cook strength model.	70
6.11	Comparison between final craters for deformation criterion. Steinberg-Guinan strength model (left) and Johnson-Cook strength model (right). Both analyses used the same value for erosion of 2.5.	71
6.12	Comparison between final crater for the element accuracy criterion with value 2.5 and skew angle 0 radians (left) and the deformation criterion with value 2.5 (right). Both analyses used the Johnson-Cook strength model.	71
6.13	Comparison between final crater for element accuracy criterion with skew angle 0 radians (left) and with skew angle 0.5 radians (right). Both analyses used the Johnson-Cook strength model.). Both analyses used the same value for erosion of 2.5.	72
6.14	Finite element mesh used for thin plate impact analyses.	73
6.15	Results for bumper plate impact, effective plastic strain erosion criterion with Johnson-Cook strength model.	74
6.16	Comparison between Johnson-Cook and Steinberg-Guinan strength models, deformation criterion	75
6.17	Results for deformation criterion, non-equal erosion criteria	76
6.18	Results for bumper plate impact, element accuracy criterion.	77
6.19	Results from variation in skew angle for erosion, element accuracy criterion	77
6.20	Comparison of bumper plate impact analyses with experimental results. T/d ration 4.97:1	78
6.21	Comparison of bumper plate impact analyses with experimental results. T/d ration 7.83:1	79
6.22	Comparison between debris cloud for effective plastic strain criterion (left) with accuracy criterion (right).	80
6.23	Comparison between debris cloud for deformation criterion (left) with accuracy criterion (right).	80
6.24	Comparison between debris cloud for deformation criterion and Johnson-Cook model (left) with deformation criterion and Steinberg-Guinan model(right).	81
6.25	Comparison between debris cloud for accuracy criterion and skew angle 0 radians(left) with accuracy criterion and skew angle 0.5 radians (right).	81
7.1	B-Spline and Quadratic kernel functions for 3 dimensions	87
7.2	B-Spline kernel stability	93
7.3	Velocity gradient field produced in 1D for alternating particle velocities	94
7.4	Normalised and non-normalised B-spline kernel estimates for the gradient of a linear function.	98

8.1	Comparison between normalised SPH approximations for a full neighbour set and a reduced neighbour set to the gradient of a function. . .	104
8.2	Particle materials for 2D test problem	106
8.3	Boundary particles for kernel contact.	106
8.4	z velocity at $0.4 \mu s$. Kernel contact.	107
8.5	z velocity at $1.4 \mu s$. Kernel contact.	107
8.6	a: Particle to surface contact. b: Particle to particle contact.	108
8.7	Interpenetration of two particles, restoring force applied between particle centres.	109
8.8	Vector diagrams showing the result of adding and subtracting the particle normal vectors.	110
8.9	Boundary particles for penalty contact.	113
8.10	z velocity at $0.4 \mu s$. Corner particles do not have stress free condition applied.	115
8.11	z velocity at $1.4 \mu s$. Corner particles have stress free condition applied.	115
8.12	Corner particle with two normals.	116
8.13	Stress profiles at $0.2 \mu s$ and $0.7 \mu s$ for kernel contact. $k = 0.5$	117
8.14	Stress profiles at $0.2 \mu s$ for penalty force types 1 and 2, using the lowest scale factor that did not give over penetration. $k = 0.5$	118
8.15	Stress profiles at $0.2 \mu s$ for penalty force types 1 and 2, using a higher scale factor. $k = 0.5$	118
8.16	Detail of stress oscillation at contact surface, penalty type 2.	119
8.17	Comparison of penetration time histories for penalty types 1 and 2.	120
8.18	Velocity profile at $0.2 \mu s$ for penalty type 3. $k = 0.5$	120
8.19	Velocity profile at $0.2 \mu s$ for modified penalty type 3. $k = 0.5$	121
8.20	Stress profile at $1.1 \mu s$ for penalty types 1 and 2. $k = 0.5$	122
8.21	Stress profile at $1.1 \mu s$ for penalty type 3. $k = 0.5$	123
8.22	Time histories for nodes 51 and 52, showing difference in behaviour caused by stress oscillation.	124
8.23	Velocity profile at $2.0 \mu s$. Tensile stresses have caused numerical fracture near contact point. Penalty type 2, $k = 0.5$, $K_p = 0.01$	125
8.24	Stress profiles at $0.2 \mu s$ for $k = 0.5$ and $k=0.1$. Showing the effect on the stress oscillation of reducing the time step.	125
8.25	Stress profiles at $0.2 \mu s$ for $k = 0.5$ and $k=0.1$. Showing the effect on the stress oscillation of reducing the contact force scale factor.	126
8.26	Stress profiles at $0.2 \mu s$ for $k = 0.5$ and $k=0.1$, for an impact velocity of $200 m/s$	127
8.27	Velocity contours at $0.4 \mu s$ (right) and $1.4 \mu s$ (left), for the different force vectors. Penalty force type 2, $K_p = 0.05$, $k = 0.1$	129
8.28	Velocity contours at $0.4 \mu s$ (right) and $1.4 \mu s$ (left), for the different force vectors. Penalty force type 3, $k = 0.1$	130
8.29	x momentum time histories for the different force vectors.	131
8.30	x velocity at $2.4 \mu s$ for offset impact tests. An instability can be seen with the average normal contact case. $k = 0.1$	133
8.31	σ_{xx} profile along contact surface at $1.0 \mu s$ for the average normal with penalty type 2 case, with initially offset particles.	134

8.32	σ_{xx} stress component (horizontal) and σ_{zz} stress component (vertical) at $1.0\mu s$ for average normal contact. $k = 0.1$	135
8.33	σ_{xx} profile along contact surface at $1.0\mu s$ for the average normal with penalty type 2 case. $k = 0.3$, $K_p = 0.05$	136
8.34	σ_{xx} profile along contact surface at $1.0\mu s$ for the average normal with penalty type 2 case. $k = 0.1$, $K_p = 0.01$	136
8.35	Comparison of particle positions at $2.0\mu s$ for 200 m/s impact.	137
8.36	Nearest stress point neighbour to boundary stress point.	144
8.37	Stress profiles for kernel contact.	146
8.38	Stress profiles for penalty equation 2.	147
8.39	Stress profiles for penalty equation 3.	148
8.40	Stress time histories for stress point closest to contact interface, for penalty equation 2.	149
8.41	Total kinetic energy time histories.	150
8.42	Total kinetic energy time histories, showing effect of increasing spatial resolution.	151
8.43	Comparison of stress time histories for velocity point on boundary and DYNA3D.	152
B.1	Pressure contour plot, no element erosion. Time $0.075\mu s$. Unmodified DYNA3D.	174
B.2	Pressure contour plot, no element erosion. Time $0.125\mu s$. Unmodified DYNA3D.	175
B.3	Pressure contour plot, no element erosion. Time $0.2\mu s$. Unmodified DYNA3D.	175
B.4	Pressure contour plot, no element erosion. Time $0.025\mu s$. Unmodified DYNA3D.	176
B.5	Pressure contour plot, erosion criterion 0.5. Time $0.025\mu s$. Unmodified DYNA3D.	176
B.6	Pressure contour plot, erosion criterion 1.0. Time $0.025\mu s$. Unmodified DYNA3D.	177
B.7	Pressure contour plot, erosion criterion 1.5. Time $0.025\mu s$. Unmodified DYNA3D.	177
B.8	Pressure contour plot, erosion criterion 2.5. Time $0.025\mu s$. Unmodified DYNA3D.	178
B.9	Displacement time history for node on the edge of the final hole Unmodified DYNA3D.	178
B.10	Time histories for initial velocity $0.1\text{ cm}/\mu s$. Modified DYNA3D. . .	179
B.11	Time histories for initial velocity $0.4\text{ cm}/\mu s$. Modified DYNA3D. . .	180
B.12	Time histories for initial velocity $0.8\text{ cm}/\mu s$. Modified DYNA3D. . .	181
B.13	Semi-infinite impact at $4\mu s$. Effective plastic strain criterion. Projectile criteria 1.5, Target criteria 1.5. Johnson-Cook strength model. . .	188
B.14	Semi-infinite impact at $24\mu s$. Effective plastic strain criterion. Projectile criteria 1.5, Target criteria 1.5. Johnson-Cook strength model. .	188
B.15	Semi-infinite impact at $4\mu s$. Deformation erosion criterion. Projectile criteria 1.5, Target criteria 1.5. Johnson-Cook strength model.	189

B.16	Semi-infinite impact at $24\ \mu s$. Deformation erosion criterion. Projectile criteria 1.5, Target criteria 1.5. Johnson-Cook strength model. . .	189
B.17	Semi-infinite impact at $4\ \mu s$. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5. Johnson-Cook strength model. . . .	190
B.18	Semi-infinite impact at $24\ \mu s$. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5. Johnson-Cook strength model. . .	190
B.19	Semi-infinite impact at $4\ \mu s$. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5. Steinberg-Guinan strength model . . .	191
B.20	Semi-infinite impact at $24\ \mu s$. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5 Steinberg-Guinan strength model .	191
B.21	Semi-infinite impact at $4\ \mu s$. Element accuracy erosion criterion. Projectile criteria 2.0, Target criteria 2.0. Johnson-Cook strength model.	192
B.22	Semi-infinite impact at $24\ \mu s$. Element accuracy criterion. Projectile criteria 2.0, Target criteria 2.0. Johnson-Cook strength model. . . .	192
B.23	Semi-infinite impact at $4\ \mu s$. Element accuracy erosion criterion. Projectile criteria 3.0, Target criteria 3.0. Johnson-Cook strength model.	193
B.24	Semi-infinite impact at $24\ \mu s$. Element accuracy criterion. Projectile criteria 3.0, Target criteria 3.0. Johnson-Cook strength model. . . .	193
B.25	Semi-infinite impact at $4\ \mu s$. 8 point element accuracy erosion criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0 radians. Johnson-Cook strength model.	194
B.26	Semi-infinite impact at $24\ \mu s$. 8 point element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0 radians. Johnson-Cook strength model.	194
B.27	Semi-infinite impact at $4\ \mu s$. 8 point element accuracy erosion criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0.5 radians. Johnson-Cook strength model.	195
B.28	Semi-infinite impact at $24\ \mu s$. 8 point element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0.5 radians. Johnson-Cook strength model.	195
B.29	Z displacement time history for node at centre of crater. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5	196
B.30	X displacement time history for node at edge of crater. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5	196
B.31	Mesh for semi-infinte impact tests with varying velocity.	197
B.32	X displacement time history for node at edge of hole. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5	197
B.33	Pressure contour plot, effective plastic strain erosion criterion, Johnson-Cook model. Time $0.075\ \mu s$. Erosion strain 1.5	198
B.34	Pressure contour plot, effective plastic strain erosion criterion, Johnson-Cook model. Time $0.2\ \mu s$. Erosion strain 1.5	198
B.35	Bumper shield impact at $0.5\ \mu s$. Effective plastic strain erosion criterion. Projectile criteria 1.0, Target criteria 1.0. Johnson-Cook strength model.	199
B.36	Bumper shield impact at $4\ \mu s$. Effective plastic strain erosion criterion. Projectile criteria 1.0, Target criteria 1.0. Johnson-Cook strength model.	199

B.37 Pressure contour plot, deformation erosion criterion.. Time 0.075 μs .	200
B.38 Pressure contour plot, deformation erosion criterion.. Time 0.2 μs .	
Erosion value 2.5.	200
B.39 Bumper shield impact at 0.5 μs . Deformation erosion criterion . Projectile criteria 1.5, Target criteria 1.5. Erosion value 2.5. Johnson-Cook strength model.	201
B.40 Bumper shield impact at 4 μs . Deformation erosion criterion . Projectile criteria 1.5, Target criteria 1.5. Johnson-Cook strength model. .	201
B.41 Bumper shield impact at 0.5 μs . Deformation erosion criterion . Projectile criteria 2.5, Target criteria 2.5. Johnson-Cook strength model. .	202
B.42 Bumper shield impact at 4 μs . Deformation erosion criterion . Projectile criteria 2.5, Target criteria 2.5. Johnson-Cook strength model. .	202
B.43 Bumper shield impact at 0.5 μs . Steinberg-Guinan strength model. Deformation erosion criterion . Projectile criteria 2.5, Target criteria 2.5	203
B.44 Bumper shield impact at 4 μs . Steinberg-Guinan strength model. Deformation erosion criterion . Projectile criteria 2.5, Target criteria 2.5	203
B.45 Pressure contour plot, accuracy erosion criterion.. Time 0.075 μs . . .	204
B.46 Pressure contour plot, accuracy erosion criterion.. Time 0.2 μs . . .	204
B.47 Bumper shield impact at 0.5 μs . Element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0.5 radians. Johnson-Cook strength model.	205
B.48 Bumper shield impact at 4 μs . Element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5 Skew angle 0.5 radians. Johnson-Cook strength model.	205
B.49 Bumper shield impact at 0.5 μs . Element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0 radians. Johnson-Cook strength model.	206
B.50 Bumper shield impact at 4 μs . Element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5 Skew angle 0 radians. Johnson-Cook strength model.	206

List of Tables

3.1	Initial conditions and experimental results of impact on thin plate . . .	27
4.1	CPU timings using Gruneisen and SESAME equations of state	49
4.2	DYNA results for points on aluminium Hugoniot curve	53
6.1	Published results for semi-infinite impact test case.	62
8.1	The variables required at each type of particle	142
B.1	Results for semi-infinite target. Effective plastic strain erosion criterion. Johnson-Cook strength model.	182
B.2	Results for semi-infinite target. Deformation erosion criterion. Johnson-Cook strength model.	182
B.3	Results for semi-infinite target. Deformation erosion criterion. Steinberg-Guinan strength model.	182
B.4	Results for semi-infinite target. Accuracy erosion criterion. Johnson-Cook strength model.	183
B.5	Results for semi-infinite target. Accuracy erosion criterion, varying only skew angle for erosion. Johnson-Cook strength model.	183
B.6	Experimental results for impact of 3.175mm aluminium sphere onto aluminium target. Data is from Denardo, Summers and Nysmith [1]. .	184
B.7	DYNA3D results for impact of 3.175mm aluminium sphere onto aluminium target. Deformation erosion criterion. Johnson-Cook strength model	184
B.8	Results for thin plate target. Effective plastic strain erosion criterion. Johnson-Cook strength model.	185
B.9	Results for thin plate target. Deformation erosion criterion. Johnson-Cook strength model.	185
B.10	Results for thin plate target. Deformation erosion criterion. Steinberg-Guinan strength model.	185
B.11	Results for thin plate target Accuracy erosion criterion, Jacobian determinant evaluated at 8 points. Johnson-Cook strength model.	186
B.12	Experimental results for impact of 7.95mm aluminium sphere onto 1.6mm aluminium plate. Data is from Schonberg, Bean and Darzi [2].	186
B.13	DYNA3D results for impact of 7.95mm aluminium sphere onto 1.6mm aluminium plate. Deformation erosion criterion. Johnson-Cook strength model.	187
B.14	Experimental results for impact of 6.35mm aluminium sphere onto 0.81mm aluminium plate. Data is from Schonberg, Bean and Darzi [2].	187

B.15 DYNA3D results for impact of 6.35mm aluminium sphere onto 0.81mm
aluminium plate. Deformation erosion criterion. Johnson-Cook strength
model 187

Notation

A_r	Atomic weight
$a^\alpha, \mathbf{a}$	Acceleration
$\mathbf{B}$	Strain displacement matrix
c	Speed of sound
D	Crater diameter, hole diameter
E	Specific internal energy
e	Internal energy
$\mathbf{F}$	Global nodal force vector
G	Shear modulus
h	SPH smoothing length
$\mathbf{J}$	Jacobian matrix
K	Bulk modulus
K_p	Penalty stiffness scale factor
k	Time step scale factor
$\mathbf{M}$	Lumped mass matrix
m	Mass
n^α	Unit normal vector to a surface
P	Pressure, Crater depth
p	Penetration
R	Universal gas constant
$\mathbf{r}$	Vector between two SPH particle positions
$s^{\alpha\beta}, \mathbf{s}$	Deviatoric stress tensor
Δt	Time step
u_p	Particle velocity behind shock wave
u_s	Shock wave velocity
V	Specific volume
v	SPH normalised inter-particle distance
$v^\alpha, \mathbf{v}$	Velocity
W	SPH kernel function
$x^\alpha, \mathbf{x}$	Position
γ_0	Gruneisen gamma
γ_t	Thermodynamic gamma
$\dot{\varepsilon}^{\alpha\beta}, \dot{\varepsilon}$	Strain rate, or Rate of deformation tensor
μ	Excess compression
$\xi^\alpha, \xi, \eta, \zeta$	Iso-parametric element co-ordinates
ρ	Density
ρ_0	Initial or reference density
$\sigma^{\alpha\beta}, \sigma$	Stress tensor
ϕ_i	Iso-parametric element shape functions
$\dot{\Omega}^{\alpha\beta}$	Spin tensor

Chapter 1

Introduction

This thesis addresses the problem of modelling hypervelocity impact in a Lagrangian hydrodynamics computer code (hydrocode). Lagrangian finite element and finite difference codes have well known problems when modelling large material deformations that prevent them from arriving at a solution. This has limited the applicability of Lagrangian codes to hypervelocity impact, as the impact process typically involves large material deformations. However the Lagrangian approach has certain advantages over the alternative approaches, most notably computational efficiency and the ability to use structural elements, such as shells, where modelling the hydrodynamic response of the material is not critical.

The aim of the work presented in this thesis was to produce a robust and accurate method for modelling hypervelocity impact on spacecraft structures. Computer simulation of hypervelocity impact is a potential tool for the design of shields for impact protection, and the assessment of potential impact damage to spacecraft. If computer simulation is to be a practical design tool for the spacecraft industry, the method used must be computationally efficient enough to produce results on a workstation class computer, and not require a super computer. This makes the Lagrangian approach attractive.

The starting point for this work was the explicit finite element code DYNA3D.

1.1 Structure of the thesis

Chapter 2 presents a summary of the DYNA3D program. A good understanding of the DYNA3D code was required in order to modify and improve the code. As information on the structure of the code was not available, a basic description is given here. The chapter concentrates on the principal routines used for all analyses and the routines most important for modelling hypervelocity impact: the solid element routines and the contact algorithm.

Chapter 3 discusses techniques for modelling hypervelocity impact in DYNA3D, and presents the results of analyses of hypervelocity impact on a thin plate made using the unmodified code. In order to model perforation of the plate the available element erosion algorithm was used, which removes highly strained elements from the calculation. The simulations did not provide accurate results, and the results identified three areas where improvement was required: material modelling, element erosion and modelling of debris cloud. No strength model or equation of state in DYNA was valid

for the whole range of conditions that occur during a hypervelocity impact. Available alternatives were considered, however no model was found that was valid for the whole range of strain rates encountered and for which suitable material constants were available. As shock wave propagation is dominant in hypervelocity impact it was concluded that a suitable equation of state had to be found and implemented. The element erosion algorithm was only implemented for one high strain rate strength model, and an incompatibility between this model and the element erosion algorithm caused melted elements to immediately be eroded. This interfered with the formation of the impact shock waves, leading to incorrect behaviour. It was concluded that the element erosion should be implemented for the other high strain rate strength model. The use of element erosion to model the perforation of the plate resulted in a significant proportion of the material that should form the debris cloud being eroded from the calculation. The eroded material meant that subsequent impact of the debris cloud could not be modelled accurately. Two potential solutions to this problem were proposed: develop an erosion criterion that erodes the minimum number of elements necessary, or implement an alternative to the Lagrangian finite element method that allows the modelling of large deformations without computational problems. The options were considered and it was concluded that a meshless Lagrangian method offered the best approach as they are capable of modelling large deformations.

Chapter 4 describes the implementation and validation of the SESAME equation of state in DYNA3D. The SESAME equation of state was selected as it is valid over the required range of conditions and was available with a wide range of material data. The implementation was validated by the calculation of points on the Hugoniot curve for aluminium from DYNA3D analyses. The implementation of this equation of state is a useful addition to the DYNA code.

Chapter 5 presents the development of two alternative erosion criteria. The development of element erosion algorithms has in the past concentrated on the problem of an efficient contact algorithm, and not on what makes an effective erosion criterion. The first criterion developed was based on the effective deformation, calculated from the rate-of-deformation tensor. This provides a strain based criterion that is independent of the material model used. The second criterion developed concentrated on eroding the minimum number of elements necessary. As element erosion is used for computational reasons, to remove elements that threaten the accurate continuation of the analysis, and not to model a physical process, a criterion based on element accuracy was developed. The solid element implemented in DYNA uses a 1 point spatial integration rule, the assumptions behind this rule mean that the integration of the element is not accurate when the element is not a parallelepiped. This condition was detected by calculating the determinant of the element Jacobian matrix at eight points within the element. If the ratio between the maximum and minimum values of the determinant exceeded a set value then the element was eroded.

Chapter 6 presents the results of simulations using the new erosion criteria. Two types of target were considered, semi-infinite and thin plate. For the semi-infinite tests the deformation criterion gave the most accurate results for the final crater size, and was computationally cheapest. Element erosion was unable to accurately predict the changes in crater size with impact velocity, which limits its use as a predictive tool. For the thin plate target tests the deformation criterion gave the best results for the final

hole diameter in the plate, however few element survived to form the debris cloud. The element accuracy criterion was best for the debris cloud, giving reasonable results for the debris cloud velocity and expansion angle. However still too many elements had been eroded to allow accurate simulation of subsequent impacts of the debris cloud. From this experience it was decided that further work on element erosion did not offer sufficient benefit to continue.

Chapter 7 is a review of the Smoothed Particle Hydrodynamics method (SPH). SPH is a Lagrangian method which does not use a structured computational mesh, and so is capable of modelling large deformations. The method is comparatively young, and there are several well known problems with the method when it is applied to solid mechanics. The chapter reviews proposed solutions to these problems, concentrating on conservative smoothing and the normalised smoothing function approach.

Chapter 8 covers the development and testing of a contact algorithm for SPH. The contact boundary condition in SPH has not been adequately addressed, and the development of the normalised smoothing function approach has highlighted the need for correct treatment of boundary conditions. A particle to particle contact algorithm was developed for 2D, which is simple to extend to 3D. As this approach does not use surfaces, it was not clear how best to apply the restoring force, and two possible approaches were proposed. The penalty formulation was used to enforce the contact condition, and several equations for the penalty force were considered. The contact algorithm was tested for one and two dimensional test problems to determine the best penalty force equation and the best approach for applying the restoring force. The tests showed that the zero-energy mode problem in the SPH method had to be addressed, as contact excited a zero-energy mode that caused non-physical motion of particles. The zero-energy modes are a characteristic of the SPH method where an alternating field variable produces a zero gradient at the particles and so the formation of an alternating field is not resisted. An alternative discretisation method was proposed that used two types of particle, particles where the velocity is known and particles where the stress is known, that are not co-located. This approach would prevent the zero-energy modes from occurring, and also is a probable solution to the tensile instability problem. A 1D algorithm was presented and tested, showing that it does solve the zero-energy mode problem. This approach also offered a method to satisfy the stress-free boundary condition without the need to explicitly apply it, as is required in conventional SPH. This was done by placing only points where the velocity is known at material boundaries.

1.2 Hypervelocity Impact

Hypervelocity impact can be defined as an impact where shock waves dominate the overall response of projectile and target. The lower velocity limit of hypervelocity impact is not fixed, but is dependent on the materials involved. For example hypervelocity impact occurs at a lower impact velocity in lead, speed of sound 2.03 km/s, than in aluminium, speed of sound 5.38 km/s. In practice the onset velocity for most metals is of the order of 3 km/s [3].

1.2.1 The impact process

In a hypervelocity impact shock waves are generated at the point of impact and propagate through the impacting bodies. The material behind the shock waves behaves effectively as an inviscid, compressible fluid [4]. Material strength is only important in the later stages of impact when the final configuration of the material is being determined.

Hypervelocity impacts can be split into three principal categories: impact into a semi-infinite target, impact into an intermediate thickness target and impact into a thin target. A good illustration of the differences between these three categories can be found in Hörz et al. [5].

A semi-infinite target is one in which the side and rear surfaces have no significant effect in the impact process. The front surface of the target is defined as the surface on which the impact occurs. The impact forms a crater in the target, the size and shape of which is strongly influenced by the velocity and shape of the projectile [6]. In general spherical projectiles or cylindrical projectiles with equal length and diameter will produce an approximately hemispherical crater in the target.

In an impact into an intermediate thickness target the rear surface of the target has an effect on the final result. When the shock wave in the target reaches the rear surface it is reflected as a tensile wave. When the strength of this wave exceeds the material spall strength fracture occurs along a plane parallel to the rear surface, a phenomenon known as spall. If the crater depth reaches approximately two-thirds of the target thickness then the crater floor can meet the spall plane, resulting in perforation of the target.

An impact into a thin target will result in the penetration of the target by the projectile. The impact generated shock wave in the target reflects from the rear of the target and travels back through the plate and projectile. The shock-expansion process fragments, melts or vaporises the projectile and material in the target near the impact. This material then forms a high velocity cloud of debris that will impact on subsequent bodies.

1.2.2 The risk to spacecraft

The risk to a spacecraft is that an impact will destroy or seriously damage the craft. The probability of impact with a body large enough to destroy the craft is small, however an impact with even a small particle could damage a critical system. For example a small impact could damage a fuel tank or fuel line, leading to an explosion. The risk to spacecraft comes from two sources, micrometeoroids and space debris.

Meteoroids are naturally occurring particles that in the near-Earth environment occur as either sporadic or stream meteoroids. Sporadic meteoroids are individual particles that have no known relation to other particles, and are randomly distributed. Stream meteoroids are generally thought to be cometary debris, and Earth's passage through individual streams occurs at predictable times each year. Nearly all meteoroids have a velocity relative to Earth of 11-72 km/s [7]. The lower limit is from earth orbital velocities, and the upper limit is from the sum of the earth's orbital velocity around the sun and the parabolic orbital velocity at one astronomical unit in the solar system.

In the late 1950's and early 1960's there was concern that the meteoroid flux in space might be high enough to pose a significant risk to space travel [8]. This concern drove the early interest in protective shields for spacecraft. By the mid to late 1960's, experience had shown that the risk from impact is small.

Space debris is defined as any man-made non-operational object in space. This debris ranges in size from non-functional satellites and launch vehicle upper stages down to minute chips of paint and aluminium oxide particles from solid rocket exhausts. The most probable impact velocities are dependent on the orbit of the spacecraft, for a space station orbit the most probable impact velocity is in the 11-13 km/s range [8].

Since the early 1980's space debris has been a subject of concern. There is considerable on going effort into improving our understanding of the debris environment, and in developing strategies to minimise the future growth of the population [9, 10, 11]. With the increase in the debris population has come the risk of 'cascading', where impacts between fragments in orbit create more fragments [8]. In the worst case this could render some orbits unusable.

In the space community the greatest interest is in particles ranging from 0.1 to 10 cm [8] in size. This range represents those particles large enough to potentially cause severe damage in an impact, yet are too small to be tracked from the ground.

The number of orbital impacts which have resulted in the loss or impairment of a spacecraft is hard to assess, as it is not usually possible to physically inspect a satellite after it has failed. One confirmed debris impact occurred in 1996 between a fragment of an Ariane rocket and the French Cerise satellite [12], this impact can be confirmed as both the fragment and satellite are large enough to be tracked from the ground. One hypothesis for the cause of the anomaly that ended the life of the OLYMPUS communications satellite is a meteoroid impact [13]. In addition there are 5 instances listed in [14] where the loss or impairment of a satellite is attributed to impact damage.

1.2.3 Spacecraft protection

There are two main approaches that can be taken to protect a spacecraft from impact. The first is to design the craft to minimise the risk that an impact will damage a critical component. This is most easily done by placing critical components inside the spacecraft, thus using the surrounding, less critical components, as shielding. For this to be effective requires a good understanding of the risk of impact, and the potential damage from an impact.

The second approach is to protect the spacecraft by using sacrificial shields. This approach has received considerable attention as part of the space station programme, in order to protect the pressurised modules from impact. The simplest shield design places a single sacrificial plate at a distance away from the body to be protected. The projectile impact on the shield fragments, melts or vaporised the projectile. The ensuing debris cloud expands in the space behind the shield. The impact of the debris cloud on the body being protected is then spread over an area of the body being protected, reducing the damage. This single shield is often called the 'Whipple shield' after its originator [15]. This concept was soon extended to the double bumper shield, that places a second shield in between the first shield and the body. The second shield further breaks up any fragments in the debris cloud, and is more effective at lower impact velocities.

More recently several shield designs have been proposed, all of which concentrate on giving equal or better protection for less mass. These are the Multi Shock Shield [16], the Mesh Double Bumper [17] and the Topographically Modified Bumper [18].

1.2.4 Hypervelocity impact research

Research into hypervelocity impact has been carried out for over four decades. Early research was driven by anti-ballistic missile technology, with the advent of the space programme research began into impact protection. Hypervelocity impact research relevant to spacecraft protection has concentrated on: understanding the impact process, shield design and characterisation of material behaviour at high pressures and strain rates.

For adequate laboratory testing of shielding designs, the capability to launch projectiles in the mass-velocity range of interest is required. The only laboratory launcher capable of launching projectiles with a complex shape, different materials and large masses is the light gas gun. However the light gas gun is currently only capable of launching projectiles at up to 8-9 km/s [19]. Techniques have been developed to allow impact testing at velocities in excess of 10 km/s, which include shaped charges [20] and explosive multi-stage launchers [21], but these limit the possible projectile shapes and materials. At the microscopic level higher velocities are possible, for example the electrostatic accelerator at the University of Kent at Canterbury is capable of accelerating dust particles to velocities between 0.4 and 87 km/s, but for masses in the 10^{-13} – 10^{-20} kg range [22]. Even with these limitations a large number of impact tests have been performed. A report by Schonberg and Bean [2] gives the results of over 500 impact tests, principally on bumper shield designs.

Laboratory facilities are not capable of adequately covering the mass-velocity range of interest for space debris impact protection. Hydrocodes, which do not have any inherent mass-velocity limitations, offer a useful complement to laboratory facilities for impact protection design and testing.

1.3 Hydrocodes

A hydrodynamics computer code, commonly known as a hydrocode, can be defined as a code for solving large deformation transient problems that occur on a short time scale. Typically these are problems where wave propagation is important. The term hydrocode covers a broad range of methods and approaches that have been used to solve transient problems.

This section is a brief introduction to hydrocodes, and their application to hypervelocity impact. It does not attempt to be a complete or comprehensive review. For more information of the theory of hydrocodes see the reviews by Anderson [23], Zukas [24] and Benson [25]. For more information on the history of hydrocode application to hypervelocity impact see Johnson and Anderson [26].

The first hydrocodes were developed in the late 1950's, in an attempt to solve problems that could not be solved with existing techniques. Hypervelocity impact was an early application, with one of the first problems considered being the Arizona meteorite crater [26]. Hydrocodes developed rapidly, in the early 1960's Lagrangian and continuous Eulerian schemes were formulated, and led to the development of codes capable of two and three dimensional calculations. Use and evolution of hydrocodes has grown along with computer power. Increases in computer speed and storage capacity have allowed increasingly complex models to be analysed, and have allowed more

sophisticated material models to be used. However large models still require significant amounts of computer time.

It is only in the past ten years that hydrocodes have begun to be used outside the hydrocode community. Increasingly hydrocodes are being used alongside experiments to increase our understanding of the impact process [27, 28, 29, 30], and also alongside experimental tests in the shield design process [18, 31, 32]. Though the codes used are commonly Eulerian or Arbitrary Lagrangian Eulerian.

Hydrocodes are based on the equations for the conservation of mass, momentum and energy. These differential equations relate density, velocity, internal energy, stress and external body forces. They are derived from the principles of continuum mechanics, see for example Malvern [33]. Hydrocodes differ in the approach they take to arrive at a solution using these equations.

A fundamental difference is in the use of Eulerian or Lagrangian spatial discretisation. The Eulerian description is a spatial description, the computational mesh is fixed in space and material moves through the mesh. The Lagrangian description is a material description, the computational mesh is embedded in the material and moves and deforms with the material. Both schemes have their advantages and disadvantages.

The advantages of the Eulerian approach

- Allows free surfaces to be created in a natural manner
- Allows modelling of turbulent flow and large deformations
- Contact between bodies handled naturally

The disadvantages of the Eulerian approach

- Requires sophisticated algorithm to track material interfaces accurately
- Requires computational mesh throughout domain of problem, not just where material exists
- Requires accurate advection algorithm to prevent mass loss through diffusion when modelling low density flows
- Required additional algorithm to track material history variables required for more sophisticated material models

The advantages of the Lagrangian approach

- Computationally straightforward
- Material boundaries sharply defined
- Material history variables easily tracked
- Have option to use structural elements where hydrostatic behaviour is not important
- Computational mesh only required where there is material

The disadvantages of the Lagrangian approach

- Requires separate contact algorithm to handle contact between bodies
- Large material deformations cause mesh problems and inaccuracy that prevents certain types of problems from being modelled

The comparative computational and mesh efficiency of the Lagrangian approach result in it being significantly computationally cheaper than the Eulerian approach. Although if the analysis involves a complex contact problem, the cost of the contact algorithm can offset the difference. However the problems that the Lagrangian approach have had modelling large material deformation have prevented it from being widely used for hypervelocity impact applications.

Due to the problems that both approaches have, there has been interest in developing methods that combine the advantages of both approaches. The Arbitrary Lagrangian-Eulerian (ALE) method is the main development of this approach. This is a method that allows the computational mesh to deform, and allows material to flow from cell to cell. There are two forms of ALE, single-material and multi-material. In single material ALE each cell can only contain one material, the result of this is that the nodes on the boundaries of the problem remain Lagrangian, and problems can still occur with large deformations. In multi-material ALE, like the Eulerian approach, each cell can contain more than one material. This approach is much closer to the Eulerian approach, but improves the mesh efficiency as the mesh can move in space, allowing areas of high resolution to move with the material.

More recently interest has grown in the Smoothed Particle Hydrodynamics (SPH) method [34]. This is a Lagrangian method that does not require a computational mesh, and so does not suffer problems when modelling large deformations. However the SPH approach is still comparatively young, and suffers from some known problems [35].

1.4 Rationale for work on Lagrangian codes

The application of Lagrangian codes to hypervelocity impact has been limited by the problems of modelling large deformations, though the SPH method does offer a solution to this. Eulerian and ALE codes are the most commonly used but do suffer from a high computational cost. For example Kerr [32] says that a 2D axisymmetric simulation of impact on a bumper shield using the Eulerian CTH code requires on the order of 50 CPU hours on a Cray YMP computer. CTH [36] is an Eulerian code that incorporates the second-order advection schemes and material interface tracking algorithms required for accurate simulation.

Hydrocodes have yet to become commonly used as a design tool. The Lagrangian approach has advantages as a design tool if it can be successfully used to model hypervelocity impact. It is computationally cheaper, it simplifies the implementation of sophisticated material models and the availability of structural elements results in a more versatile code that can also be used for non hydrodynamic problems.

Chapter 2

The Structure of DYNA3D

The purpose of this chapter is to provide an introduction to the structure and conventions of the DYNA3D program. The aim is to provide an aid to learning the structure of the program for anyone needing to debug or modify the code, and provide a supplement to the other documentation available. The version of the code described in this chapter is the 1995 release of public DYNA3D. Public DYNA3D is developed and maintained by the Methods Development Group at Lawrence Livermore National Laboratory.

The chapter is split into the following sections:

- 2.1 Introduction Describes basic structure and conventions of DYNA3D.
- 2.2 Input Describes input section.
- 2.3 Initialisation Describes initialisation section.
- 2.4 Restart Describes restart section.
- 2.5 Solution Describes main solution routine and HEX8 element routine.
- 2.6 Output Describes output subroutines.

The other documentation available for DYNA3D is the user manual [37], and the theoretical manual [38]. The user manual principally defines the structure of the input file and so covers, sometimes briefly, the capabilities available in the code. The theoretical manual covers the basic algorithms used in the code. However as it was written in 1983 it is no longer very comprehensive, but does cover the basic algorithms used. There is a more recent theoretical manual for the LS-DYNA3D code [39] dated 1994, but as LS-DYNA began diverging from public DYNA in 1988 not all of the manual is applicable.

In order to modify DYNA3D it is necessary to have an understanding of the structure of the code. This information was not in the available publications, and was necessary for the work presented in this thesis.

The DYNA3D code was first developed in the mid-70's by Hallquist at Lawrence Livermore National Laboratory, and in the two decades since has been very substantially modified and improved. It is an explicit three-dimensional finite element code intended for the analysis of problems involving large deformations and inelastic materials. The finite elements supported are linear elements: eight node bricks, four node shells and two node beams. This is due to the computational expense of higher order elements. It contains several contact-impact algorithms allowing different bodies to interact, along with material models and equations of state that cover a wide range of material behaviour.

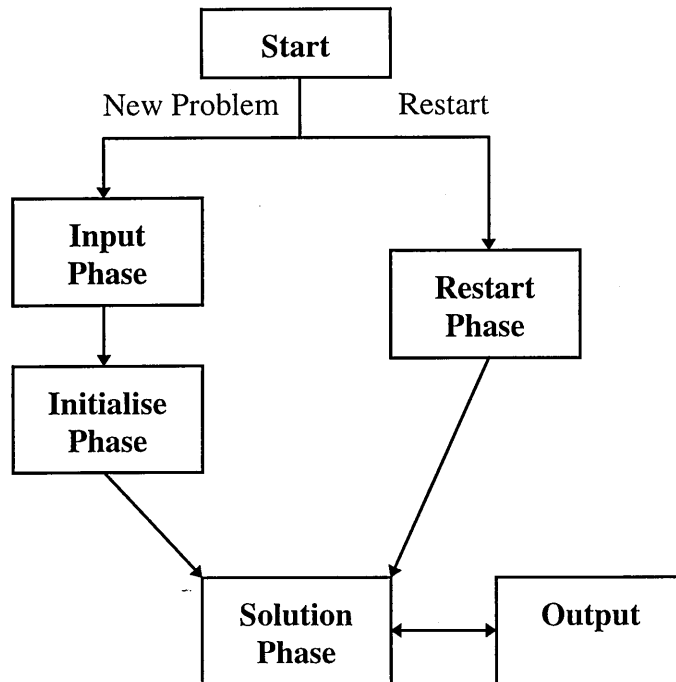


Figure 2.1: Basic structure of DYNA3D

DYNA3D is written using the Fortran77 language, a familiarity with the syntax and conventions of this language is assumed in this chapter. In particular a good understanding of the COMMON block, and the way that Fortran stores data in arrays is useful when learning the DYNA3D code.

The DYNA3D code consists of over 80,000 lines of FORTRAN, split into approximately 870 subroutines. The majority of the subroutines are specialised and are only used when a particular option or material model is being used. This chapter does not attempt to cover even the majority of these subroutines, but covers the basic structure of the code, sections that are common to all analyses, and those sections that are likely to be used when modelling hypervelocity impact.

2.1 Introduction

In this section the basic structure and conventions of DYNA3D will be described.

2.1.1 Basic Structure

The code can be split up into five main sections: Input, Initialise, Restart, Solution and Output. The relationship between these sections is shown in figure 2.1.

The input section is called when an entirely new analysis is started. This section reads in the text input file, the format of which is defined in the User Manual. From the information contained in the input file, the main database required by the program is set up. The main database stores all the element and nodal variables required by the program. The main subroutine of this section is `dynai`, which reads in the main

control cards and then calls the subroutines that load the specific data required for the analysis.

After the input section DYNA proceeds to the initialisation section. The main subroutine of this section is `initlz`. The principal tasks of this section is to initialise the contact algorithm logic, to calculate the lumped mass matrix and to calculate any other values required for the specific option used in the analysis. For example: material 11 (Steinberg-Guinan high-strain-rate elastic-plastic) requires the calculation of the cold-compression energy curve. At the end of this section all values required to begin the analysis have been calculated, and all the book-keeping data structures required have been set up.

The restart section is used if a problem is being restarted from a dump file. A dump file is a binary file which contains all the information necessary to restart the analysis from the point where the file was written. The dump files are named `d3dumpxx`, where `xx` is a number such as 04. This section first reads in a short text file which supplies the new termination time. Then the dump file is read, this file contains a copy of all the solution variables and data structures at the solution time that it was written.

The solution section is the main part of the program. The main subroutine is `fem3d`, this subroutine contains the main loop that performs the explicit time integration. Within the loop the individual subroutines that are required to calculate all the contributions to the global nodal force vectors are called.

The output section does not have a single main subroutine. Specific subroutines are called from the solution section at specified time intervals. DYNA produces three main output files, dump files, state plot files that contain all results for a specified solution time, and time history files.

2.1.2 Data handling

DYNA3D was first developed when computers contained less data storage capacity, especially memory. As a consequence the code is designed to be as efficient as possible in its use of memory. This has resulted in a programming style that can be hard to follow.

At the core of DYNA is the central database. This consists of a single large array, usually called `a` but in some subroutines called `b`. The size of this array places an upper limit on the size of the problem that the program can handle. The array is dimensioned at the start of the program, in the main routine. If there is insufficient space in the array for a problem to be analysed then the size of the array must be changed in the source code, and DYNA recompiled.

In the database all the solution variables are stored. This consists of data such as the nodal co-ordinates, velocities, positions, element connectivity, stress tensor and contact surface segments. The data stored is required or updated during each time step. In order to access the data DYNA calculates a number of pointers. Each pointer is an integer variable which contains the location in the database of each variable or array stored there. For example: `lc9` is the pointer to the array of nodal velocities. If $v(1, 1)$, the x velocity of node 1, is entry 1000 in `a` then the value of `lc9` will be 1000.

Variables are packed and unpacked from the database by using `CALL` statements. To continue with the previous example, subroutine `velocity` requires the nodal ve-

locities. The subroutine is called using:

```
call velocity(a(1c9))
```

and the subroutine starts with:

```
subroutine velocity(v)
```

Array `v` is then dimensioned later in the subroutine, and is used in the normal manner within the subroutine. When the subroutine is finished the values in the database are then automatically updated. The subroutine name `velocity` is used only as an example, and does not actually exist in the DYNA code.

The pointers are calculated during the input phase. Once DYNA knows how many nodes and elements are in the model and what options are active, it calculates the values of the pointers. The value of pointer `1c10`, which immediately follows `1c9`, is the value of `1c9` plus the number of degrees of freedom that each node has times the total number of nodes.

There are two ways that the variable associated with a pointer may be determined. The first is by looking at the subroutines where the pointers are calculated. Some of the pointers are documented within the code, and it is possible to identify some of the others by looking at the section of the input file that is being loaded at that point. The second way is by identifying a variable in other sections of the code and tracing it back to the point where it is passed in a call statement.

The other principal means by which variables are passed between subroutines is by the use of `COMMON` blocks. The 1995 version of DYNA3D uses 237 separate common blocks, and their use makes it hard to trace variables through the program. When learning DYNA it is worth running the `FTNCHEK` program, which is freely available on the internet. This program can produce useful information on a Fortran program. In particular it can produce a listing of which subroutines a `COMMON` block is referred to, and this information is invaluable when learning DYNA. To complicate matters further, variable names can differ from subroutine to subroutine, sometimes significantly, and can only be identified by their position within the particular `COMMON` block that they are passed in.

`CALL` statements are rarely used to pass variables between subroutine, unless variables are being packed or unpacked from the main database.

2.2 Input

The input phase reads the input file, sets up the main database and loads the input information into the database. This section also writes the input information into the ASCII outputfile `d3hsp`.

The main subroutine in the section is `dynai`. The structure of the subroutine is prescribed by the structure of the input file, which is defined in the User Manual. First the nine (optionally ten) main control cards are read in. These control cards define all the main parameters of the analysis, and define what options are active for the analysis.

The main parameters consist of basic information such as the number of nodes materials and elements, problem timing and output information such as the termination

time, how often to write restart or plot files, and the time step scale factor. The remaining information defines the options, which include the number of slide surfaces, the number of non reflecting boundary conditions, and how the initial conditions are defined. This information defines what data must be read from the input file, and what options must be active during the analysis.

The input file is read sequentially, so data is loaded in the order that it is presented in the input file. After the main control cards the material model data is loaded, then the nodal positions and boundary conditions and the element definitions.

The input section consists of a number of subroutines, each of which loads in a section of the input file and is called, if required, by `dynai`. The principal subroutines are:

<code>matin</code>	Load material model cards
<code>intrlb</code>	Load cross section properties for beam elements
<code>intrls</code>	Load cross section properties for shell elements
<code>nodein</code>	Load in nodal co-ordinates and displacement boundary conditions
<code>elemnh</code>	Load in element data for solid elements
<code>elemnb</code>	Load in element data for beam elements
<code>elemns</code>	Load in element data for shell elements
<code>elemnt</code>	Load in element data for thick shell elements
<code>timhis</code>	Load in load curve data
<code>sllin</code>	Load in slide surface control data

The second principal task of the input phase is to set up the main database. There is not one single section where all the pointers are calculated. They are calculated throughout the input section, generally immediately before they are required. Contact surfaces require a large amount of storage in the database. As, in addition to storing the contact segments, space is reserved for the bucket sorting required when searching for contact. When element erosion is used, contact type 11, space must be reserved for the master and slave volume databases. These are designed to allow DYNA to efficiently redefine the contact surface following the erosion of an element.

The initialisation section is called after the input section.

2.3 Initialisation

The initialisation section calculates values required during the solution section that are not separately defined in the input file. The tasks performed during this section are:

- Initialise slide surfaces (optional)
- Initialise non-reflecting boundaries (optional)
- Calculate nodal velocities for rigid body motions (optional)
- Set up temperatures (optional)
- Initialise material models
- Calculate lumped mass matrix
- Calculate gravity stress (optional)
- Initialise beam and shell elements (optional)
- Set up nodal constraints (optional)

Calculate maximum stable time step for contact

Those items marked optional are only done if that option is used in the analysis. The subroutines that perform these tasks are called from routine `initlz`

When slide surfaces are present, their initialisation is one of the principal tasks of this section. The input section has loaded the nodes or segments that make up the contact surfaces, or has specified that automatic contact is to be used.

The main task of the material model initialisation is to calculate the starting values of the auxiliary variables, to set up the initial conditions, and to calculate any additional values required by the model. Auxiliary variables are values required by a particular material model, and that have to be passed from time step to time step. An example of an auxiliary variable is effective plastic strain.

The lumped mass matrix is calculated from the element geometry and the element densities. Once this is calculated the total mass and inertial properties of each material are printed to the text output file.

Gravity stress is an option only available with solid elements. This option accounts for the increase in hydrostatic stress with depth in materials such as liquids. If this option is used the initialisation section calculates the initial element stress tensors due to gravity.

If beam or shell elements are used, geometrical properties such as element normals have to be known. The initialisation section calculates these values from the input data.

Nodal constraints allow sets of nodes to share a common degree of freedom.

Finally the maximum stable time step size for contact is calculated. It is well understood that using the penalty force algorithm to enforce contact affects the maximum stable time step of the problem. DYNA searches each slide surface that uses the penalty algorithm for the highest penalty stiffness, k_{max} , and the lowest node mass, m_{min} . From these values it calculate a maximum stable time step:

$$\Delta t_{max} = \frac{2}{\sqrt{k_{max}m_{min}}} \quad (2.1)$$

where the penalty stiffness, k , is:

$$k = K_p \frac{K A^2}{V} \quad (2.2)$$

K_p is the penalty stiffness scale factor, K is the bulk modulus, A is the segment area and V is the element volume.

2.4 Restart

The function of the restart section is to read the specified dump file, and to read the restart input file if one is supplied. The main subroutine for this section is called `rstart`, the second main routine is `rstari` which reads the restart input file, and is called from routine `rstart`.

DYNA3D dump files are binary files which contain a copy of the main database, and the principal COMMON blocks. This is all the information that is required to continue the analysis from the point that the dump file was written. The filename of a dump file

is `d3dumpxx`, where `xx` is a number. The number signifies the order in which the files were written. So the first dump file is `d3dump01`, the second `d3dump02`.

There is a maximum size set for a dump file, if the problem is large and so would produce a dump file larger than the maximum size then the data is split over more than one file. For example, if the problem would produce a dump file 2.5 times the maximum size then three files would be produced: `d3dump01`, `d3dump02` each of the maximum size and `d3dump03`, half the maximum size. The next dump file written would consist of files `d3dump04` to `d3dump06`. The maximum file size is stored in variable `maxsiz`, and is defined in the main routine of the program. The variable is passed between subroutines in COMMON block `bk06`.

Subroutine `rstart` first opens the dump file, then reads the principal COMMON blocks, including all the pointers to the database, and finally the main database. The subroutine `rstari` is then called if a restart input file is specified. The restart input file is a text file, the structure of which is defined in the User Manual. It allows the analyst to make a limited number of changes to the model. Three types of change can be made in the restart file. Termination time and output intervals may be changed. Elements, materials and slide surfaces may be deleted. Translational and rotational boundary conditions on nodes and rigid bodies may be changed.

The solution section is called after the restart section.

2.5 Solution

The solution section is the main part of the program. The main subroutine of this section is `fem3d`. This subroutine contains the loop that performs the time integration. As there are many options in DYNA only particular parts of the solution section will be looked at:

- The main subroutine `fem3d`
- The 8 node hexahedron element subroutines
- The sub-routines for type 3 and type 11 contact surfaces

These three parts have been selected as they are the principal components used when modelling hypervelocity impact. The main routine is considered as it is used in all analyses. As shock waves are dominant in hypervelocity impact it is necessary to use bulk elements, and the 8 node hexahedron is the only bulk element available in DYNA3D. The two types of contact surfaces are the types most applicable to hypervelocity impact. The type 3 contact surface is the 'sliding with separation and voids' surface. The type 11 contact surface is the element erosion surface, that allows elements on the contact surface to be deleted and redefines the contact surface when that occurs. However it still uses the same basic contact algorithm as type 3.

2.5.1 The main solution subroutine

Subroutine `fem3d` is a very large subroutine, consisting of approximately 1650 lines of Fortran, this section does not describe this routine exhaustively, but deals with the principal points.

Subroutine `fem3d` is called from subroutine `soltn`. This is in turn called by `overly` which is called by the top routine of DYN4. `overly` is the routine that calls the input, initialise, restart and solution sections in turn. `soltn` is used to unpack variables from the main database through the call statement to `fem3d`.

The basic structure of the subroutine is shown in figure 2.2. The time integration loop runs between label 10 and label 473. The core of the time integration loop are the equations:

$$\mathbf{a}^n = \mathbf{M}^{-1} \mathbf{F}^n \quad (2.3)$$

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \mathbf{a} \Delta t^n \quad (2.4)$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1/2} \Delta t^{n+1/2} \quad (2.5)$$

The same storage space in the main database is used for both the acceleration vector and the force vector, the pointer to this space in `lc10`. This is done to save storage space, and can be done because the nodal acceleration and force do not need to be known at the same time. Confusingly this array exists as two separate variable names in this routine, `a` and `ac`, and both are used at different points to denote acceleration. This duplication is achieved by placing `a(lc10)`, here `a` is the database, twice in the call statement of `fem3d`.

Equation 2.3 occurs at label 25:

```
do 25 n=1,numnp2
  ac(1,n)=ac(1,n)*xms(n)
  ac(2,n)=ac(2,n)*xms(n)
25  ac(3,n)=ac(3,n)*xms(n)
```

The lumped mass matrix, $\mathbf{M}$ is array `xms`, and the force and acceleration are array `ac`. Immediately following this the displacement boundary conditions are applied by setting the acceleration in the restrained direction to zero.

Equations 2.4 and 2.5 occur at label 170:

```
do 170 n=1,neq
  v(n)=v(n)+dtx*a(n)
170  x(n)=x(n)+dt2*v(n)
```

`neq` is the number of nodes times the number of degrees of freedom, the other variable names are self explanatory.

Between these two points in the routine the dynamic relaxation, if active, is carried out. Certain contact surfaces are processed, which are the types that use the nodal constraint method to enforce the contact condition, and require access to the nodal acceleration vector. The state plot, time history and optional output files are written.

Between labels 175 and 180, the solution time is checked, if it has reached the end time the run is terminated, and a dump file is written. Label 180 is the point where a new or restarted analysis starts. It is the point at which the new geometry has been calculated.

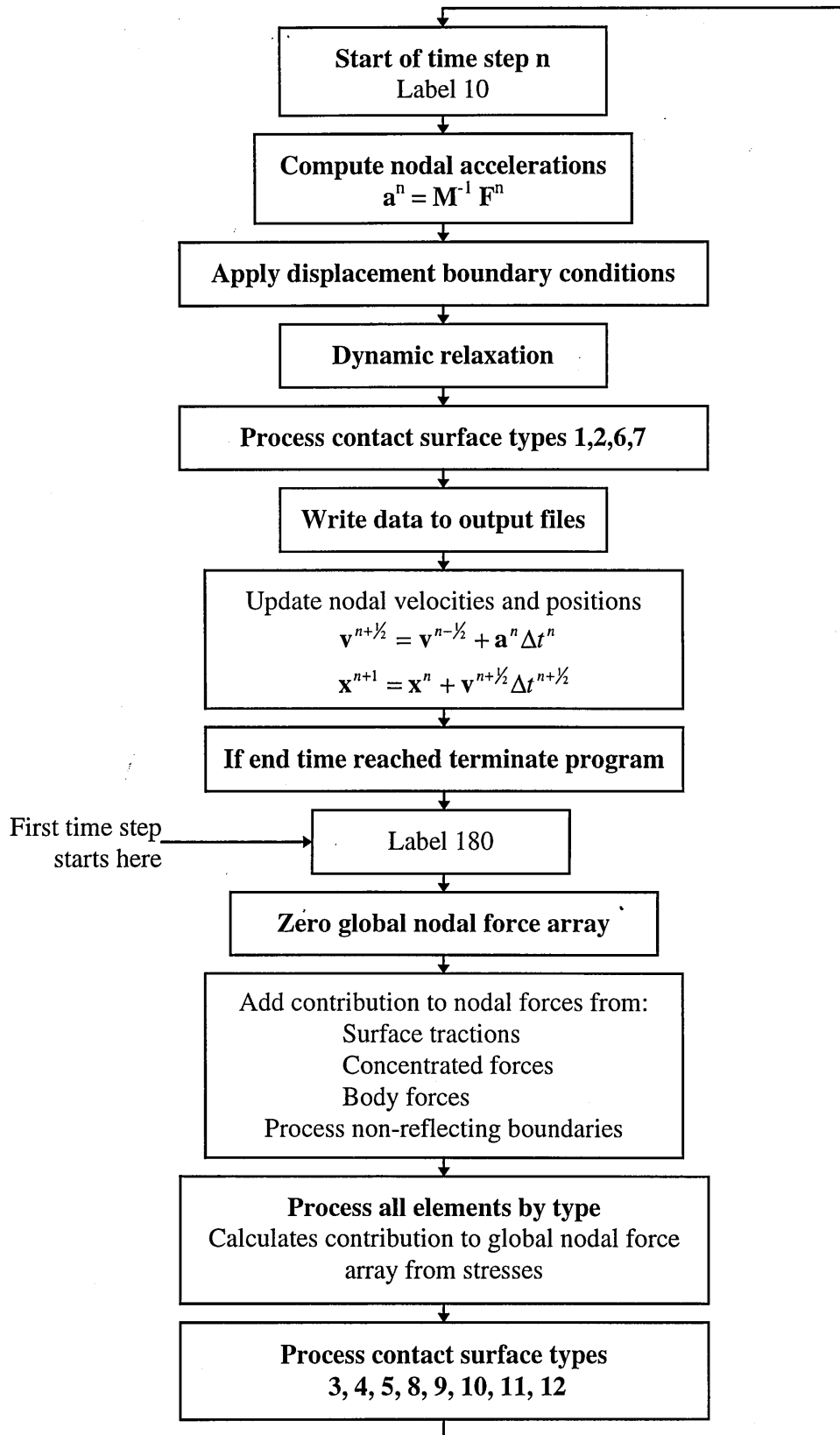


Figure 2.2: Structure of main solution subroutine.

Following label 180 the acceleration/force array is zeroed. This is done in subroutine `load`, which also calculates the contribution to the new force array from surface traction forces. From this point, and until equation 2.3 is reached again, this array is the global force array. For this section all the contributions to the nodal force array are calculated. First the contributions from surface tractions, subroutine `load`, concentrated loads, subroutine `loadcn`, body force loads, subroutine `loadbfe` and the effect of non-reflecting boundary segments, subroutine `nbscal`, are calculated. Then the elements are processed, and finally the remaining slide surfaces are processed. These are the types of surface that apply a nodal force to enforce the contact constraint, such as types 3 and 11 that use the penalty method.

A different subroutine is called for each basic type of element, these are:

<code>solide</code>	calls <code>solde</code>	processes solid elements
<code>beame1</code>	calls <code>elem1d</code>	processes beam elements
<code>ishle1</code>	calls <code>elem2d</code>	processes shell elements
<code>tshell</code>	calls <code>thicks</code>	processes thick shell elements
<code>addstf</code>		processes discrete elements

The reason that two routines are listed for all except discrete elements, is that the first routine has the same purpose as `soltn`. That is it only calls the subsequent routine, unpacking variables from the database in the call statement. Doing this makes this section of `fem3d` considerably neater.

Each routine calculates:

$$\sum_n \int_{v_n} \mathbf{B}^t \boldsymbol{\sigma} dv \quad (2.6)$$

where n is the number of elements of that type. This is the element stress contribution to the weak form of the equilibrium equation. Subroutine `solde` is covered in more detail in section 2.5.2.

The contact surfaces are processed between labels 335 and 440. In the 1997 version of `DYNA3D`, this section has been moved to a separate subroutine. The routines dealing with contact surface types 3 and 11 are described in section 2.5.3. Following this the routine loops back to label 10.

2.5.2 8 node hexahedron element subroutines

This section describes the routines that process the solid elements. Detailed notes on the subroutines in this section that are not material model specific are presented in Appendix A. Figure 2.3 shows the main subroutines called from `solde` when a strength model that requires a separate equation of state is used. As a separate equation of state is required for modelling hypervelocity impact, this section concentrates on these routines.

The `DYNA` code is vectorised in order to make efficient use of vector computers, such as Cray computers. A scalar computer can only perform an operation on one value at a time, a vector computer can perform the same operation on an array. An operation is an action such as add, subtract, multiply or set the value of a variable.

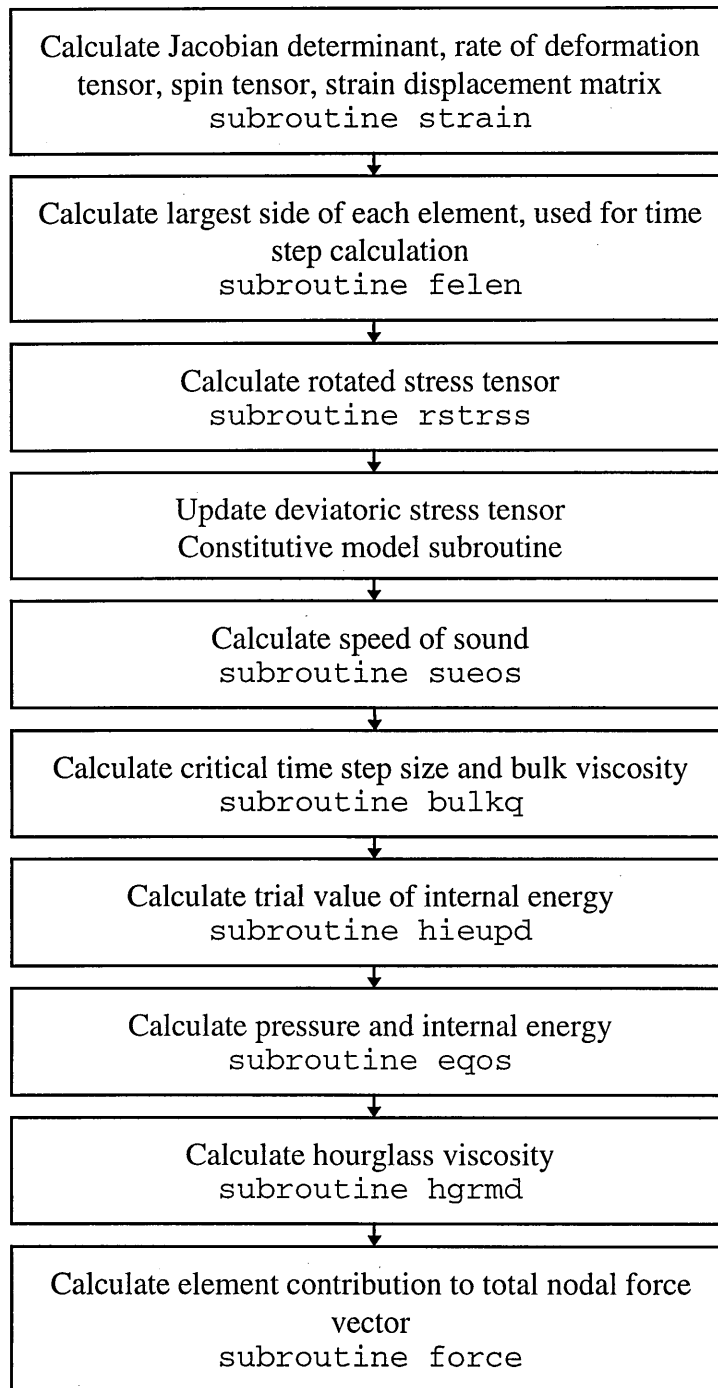


Figure 2.3: Main routines of hexahedron element section, for strength model requiring an equation of state.

The greatest gains come from vectorisation when the same operations are being carried out repeatedly, such as when processing the elements. Vectorisation is not a function carried out by the compiler, the code needs to be correctly structured to gain the benefits.

In a scalar implementation of these routines, there would be a main DO loop which would loop over each solid element in turn. For that one element each of the subroutines would be called in turn. In the vectorised code there is a main DO loop, but this loops over a number of element groups, then for each element group the subroutines are called in turn. The size of the element groups that DYNA uses varies between machines, but on a Cray computer, which is a vector computer, the groups are 128 elements long. This is a multiple of 64, which is the length of the vector register on these machines. Then a DO loop is used to repeat each operation for every element in the group being processed. So on a scalar computer a line would be:

```
a=b+c*x
```

in a vectorised code this is replaced by:

```
do 100 i=1ft,11t
    a(i)=b(i)+c(i)*x(i)
100 continue
```

where 1ft and 11t are the start and end numbers of the element group. A DO loop is not required for every line, this would result in a cumbersome code. Instead each DO loop will process a section of code.

In the initialisation phase DYNA sorts the elements into groups of the correct length. When elements in the group reference more than one material model then the group is split into subgroups, so ensuring that every element between 1ft and 11t require the same material model.

The start of the loop over all solid elements is line:

```
do 220 nn=1,nelg
```

where nelg is the number of groups. This is then followed by:

```
do 210 n=1,nsubg
```

where nsubg is the number of subgroups within the group being processed.

Many of the variables calculated in this section are not required outside of the solid element subroutines, such as the deviatoric stress tensor. Temporary arrays are used for these variables, the arrays are dimensioned to be the same length as the element group length. As a consequence the first action in the loop is to unpack data from the main database into temporary arrays. Subroutine unpki gets the element connectivity data for the elements in the group. The element connectivity data is the node numbers of the eight nodes that belong to the element. Then, if element erosion is being used, subroutine locflh loads the list of failed elements. The nodal coordinates and velocities are loaded in subroutine strain.

Subroutine strain also calculates the rate of deformation and spin tensors; and calls subroutine prtal which calculates the Jacobian matrix, and the strain displacement matrix.

The components of the Jacobian are:

$$\frac{\partial x^\alpha}{\partial \xi^\beta} = \sum_{i=1}^8 \frac{\partial \phi_i}{\partial \xi^\beta} x_i^\alpha \quad (2.7)$$

where ϕ_i are the element shape functions, ξ^β are the iso-parametric co-ordinates, and x_i^α are the nodal positions.

The Jacobian is used to calculate the element volume which, due to the one point integration rule used, can be approximated as: $V_e = 8|\mathbf{J}(0, 0, 0)|$.

The strain displacement matrix, $\mathbf{B}$, is:

$$\mathbf{B} = \begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} \\ \frac{\partial \phi_i}{\partial \eta} \\ \frac{\partial \phi_i}{\partial \zeta} \end{bmatrix} \quad (2.8)$$

The rate of deformation tensor, $\dot{\varepsilon}^{\alpha\beta}$, and the spin tensor, $\dot{\Omega}^{\alpha\beta}$, are.

$$\dot{\varepsilon}^{\alpha\beta} = \frac{1}{2} \left[\frac{\partial v^\alpha}{\partial x^\beta} + \frac{\partial v^\beta}{\partial x^\alpha} \right] \quad (2.9)$$

$$\dot{\Omega}^{\alpha\beta} = \frac{1}{2} \left[\frac{\partial v^\alpha}{\partial x^\beta} - \frac{\partial v^\beta}{\partial x^\alpha} \right] \quad (2.10)$$

where

$$\frac{\partial v^\alpha}{\partial x^\beta} = \sum_{k=1}^8 \frac{\partial \phi_k}{\partial x^\beta} v_k^\alpha \quad (2.11)$$

Next subroutine `fel en` calculates the element length used in the calculation of the critical time step. The length is calculated as:

$$\Delta x_{crit} = \frac{V_e}{A_{e_{max}}} \quad (2.12)$$

where $A_{e_{max}}$ is the area of the largest element side.

Depending on the particular material model used, different subroutines are called. It is from this point in `sol de` that the program splits to consider the different models.

Subroutine `hvpac1` loads the element stress tensor, and the element auxiliary variables. The auxiliary variables are variables that are specific to the material model, and need to be passed between time steps.

Subroutine `rstrss` calculates the rotated stress tensor, $R^{\alpha\beta}$:

$$R^{\alpha\beta} = \sigma^{\alpha\gamma} \dot{\Omega}^{\gamma\beta} \Delta t + \sigma^{\beta\gamma} \dot{\Omega}^{\gamma\alpha} \Delta t \quad (2.13)$$

To save storage space the element pressure is not stored in the main database, as it is now needed it is calculated from the stress tensor:

$$P = -1/3 \sigma^{\alpha\alpha} \quad (2.14)$$

Now the strength model specific subroutine is called. This routine calculates the new value of the deviatoric stress tensor. The subroutine name is `f3dmxx`, where `xx` is the material type number. For example the Johnson-Cook strength model is material type 15, the Johnson-Cook specific subroutine is `f3dm15`. Details of the material model algorithms are given in the theoretical manuals.

Next the element speed of sound is calculated. Subroutine `sueos` calls equation of state specific subroutines, which calculate $\rho_0 c^2$ using:

$$\rho_0 c^2 = \frac{4G}{3} + \rho_0 \left. \frac{\partial P}{\partial \rho} \right|_E + P V^2 \left. \frac{\partial P}{\partial E} \right|_\rho \quad (2.15)$$

Subroutine `bulkq` calculates the critical time step size and the bulk viscosity. The time step is:

$$\Delta t_{crit} = \frac{\Delta x_{crit}}{Q + \sqrt{Q^2 + c^2}} \quad (2.16)$$

where

$$Q = Q_2 c + Q_1 |\dot{\epsilon}^{\alpha\alpha}| \Delta x_{crit} \quad (2.17)$$

Q_1 and Q_2 are the bulk viscosity coefficients.

The bulk viscosity is

$$q = \Delta x_{crit} |\dot{\epsilon}^{\alpha\alpha}| (\rho Q_1 \Delta x_{crit} |\dot{\epsilon}^{\alpha\alpha}| + \rho Q_2 c) \times \frac{V_0}{V} \quad (2.18)$$

Next the pressure and internal energy are updated. As the new energy is a function of the new pressure, and the new pressure is a function of the new energy, the equations are implicit. To solve the equations, first a trial value of the internal energy, e^* , is calculated in subroutine `hieupd`:

$$e^* = e^n - \frac{1}{2} P^n \Delta V + V^{n+1/2} s^{n+1/2} \dot{\epsilon}'^{n+1/2} \Delta t^{n+1/2} \quad (2.19)$$

Then subroutine `eqos` calls an equation of state specific routine that calculates P^{n+1} and e^{n+1} .

With the calculation of the pressure the new value of the full stress tensor is known. Subroutine `hvpac2` stores the new stress tensor and auxiliary variables in the main database, and if element erosion is active subroutine `updf1h` updates the list of failed elements.

This is the point where the program stops differing between material models.

The hourglass viscosity is now calculated. DYNA3D supports two basic types of viscosity. The standard DYNA3D viscosity, and the Flanagan-Belytschko viscosity. Four subtypes of the the Flanagan-Belytschko viscosity are supported. Subroutine `hgrmd` is the standard DYNA3D viscosity subroutine.

Finally the element integration is performed, subroutine `force` calculates the contribution to the global nodal force vector from the element stresses and the hourglass forces, **H**:

$$\mathbf{F} = \mathbf{F} + \left[\mathbf{H} - \int_{V_e} \mathbf{B}^t \boldsymbol{\sigma} dV \right] \quad (2.20)$$

This is the end of subroutine `solde`.

2.5.3 Contact algorithm routines

This section describes the main routines used by contact surface types 3 and 11. Contact surface type 3 is called sliding with separation and friction. It is a general purpose contact algorithm that allows the two contact surfaces to be initially in contact or separate, the surfaces may separate and come together in an arbitrary manner during the analysis. It uses the penalty formulation to enforce the contact condition. Contact surface type 11 is called Slidesurfaces with adaptive new definitions (SAND). This is the contact portion of the element erosion algorithm, it uses the same algorithm as type 3 to detect and enforce contact, but allows the contact surface to be redefined as elements are eroded.

The main slide surface loop is between labels 335 and 440. This loops over all slide surfaces, and processes all except types 1,2,6 and 7. Figure 2.4 shows the main routines that are called to process types 3 and 11.

The first two routines are called if element erosion is active. These two routines redefine the contact surface if any changes have occurred. First subroutine `sandrn1`, which searches for new failed elements. This subroutine loops over all contact segments, and checks to see if the element belonging to that segment has been deleted, if so, the element is added to a list. For each deleted element, all the contact segments associated with that element are added to the list of failed elements. Next subroutine `dslnsg` deleted failed contact surfaces by setting their penalty stiffness to zero.

The penalty algorithm used is symmetric, which means that first contact between slave nodes and master segments is considered, then contact between master nodes and slave segments is considered. This technique makes the contact more reliable, as problems occur in a non-symmetric algorithm when the mesh density on the slave surface is similar or denser than that of the master surface. In the symmetric form the designations of master and slave surface are arbitrary, and are used only to differentiate the two surfaces.

Routines `dnasin`, `vctor2` and `oudnas` are each called twice, once for slave nodes onto master surface, and once for master nodes onto slave surface. Each of these routines will be described for the slave node onto master surface case.

Subroutine `dnasin` performs a bucket sort to find the nearest node on the master surface to each slave node. As the bucket sort is computationally expensive it is not performed every time step. `issflg` is a counter that is incremented time step, when it reaches 11 it is reset to 1. Subroutine `dnasin` is called only if `issflg` is 1. The counter is also reset to 1 in subroutine `sndrn1` if a new failed element is detected.

Subroutine `vctor2` calculates geometrical properties of each master surface segment, such as the unit normal vector to the surface, that are not dependent on the location

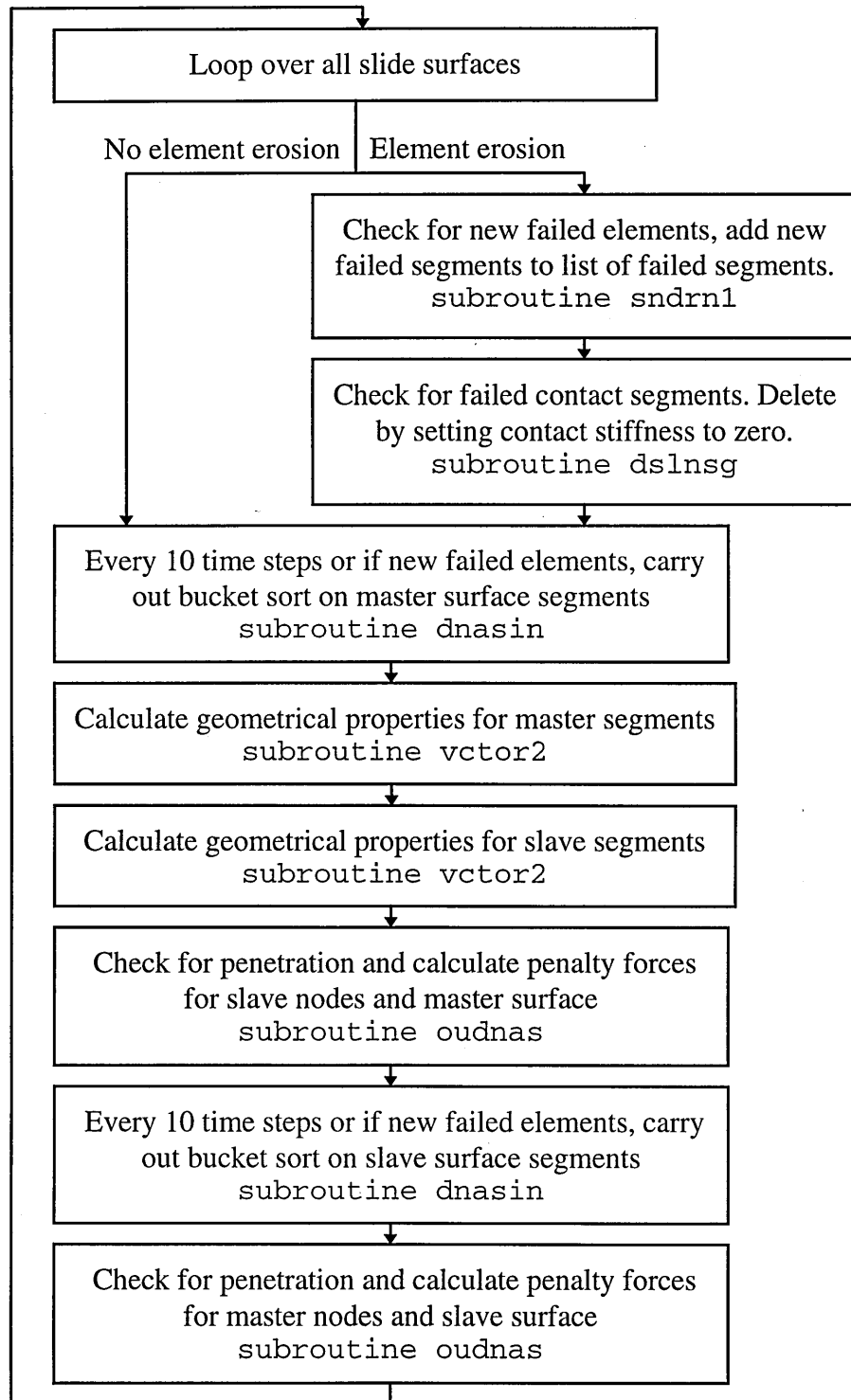


Figure 2.4: Main routines of penalty contact algorithm.

of any slave node. Finally subroutine `oudnas` checks for penetration and applies a restoring force to the slave node and the master segment nodes.

The other contact surface types processed in this section of `fem3d` are:

Type 4 single surface contact subroutine, `sscnc` is the principal subroutine.

Types 5 and 10 discrete nodes impacting surface and one pass sliding, these use a non-symmetric penalty formulation that uses the same routines as type 3.

Types 8 and 9 nodes spotwelded to surface and tied nodes with failure, these use routines `slave1`, `slave2`, `slav12` and `slavf2`.

Type 12 automatic contact, `fstglb` is the principal subroutine.

2.6 Output

Unlike the other section the output section is not a separate section. Individual routines in the section are called from other sections at appropriate intervals.

DYNA3D writes four main types of output file which are the printed output file, the dump file, the state plot file and the time history file. The principal output subroutines are:

<code>femdump</code>	Writes dump file
<code>prtdat</code>	Writes binary or text state plot file or dynamic relaxation plot file
<code>dtout</code>	Writes binary time history file

The binary plot file is the file that is read by the post-processor, and is the default output file written by DYNA3D. It is written at specified time intervals and contains data for all nodes and elements. The binary plot files are called `d3plotxx`, where `xx` is an integer. Like the dump files, the maximum size of each individual plot file is controlled by constant `maxsiz`. Optionally DYNA3D can create a text output file that writes data for the all elements and nodes in a format suitable for printing on a line printer. The time history file is called `d3thdt`. It is always created, but data is only written into it if time history blocks have been defined in the input file.

Chapter 3

Evaluation of DYNA3D

In this chapter work on modelling hypervelocity impact in unmodified DYNA3D is covered. The aim of this work was to gain experience with using the code, and to identify areas where improvements were required to allow accurate modelling of hypervelocity impact.

The chapter is split into the following sections:

- 3.1 Modelling hypervelocity impact in DYNA3D
- 3.2 Modelling hypervelocity impact without element erosion
- 3.3 Modelling hypervelocity impact with element erosion
- 3.4 Discussion
- 3.5 Summary

The problems that a Lagrangian finite element code, such as DYNA3D, has when modelling hypervelocity impact are well known [25, 24]. The large material deformation that occurs in the splash and the debris cloud cause mesh problems that force the calculation to terminate. The main problem is element inversion, in which the calculated volume of the element becomes negative. A second problem is that the time step becomes very small, due to the critical length of some distorted elements becoming small, which leads to the calculation becoming prohibitively expensive in CPU time. Element erosion algorithms, which remove these highly distorted elements from the calculation, were developed to counter these problems [40]. DYNA3D includes an element erosion algorithm, which offers the possibility of successfully modelling penetration.

In order to evaluate DYNA it was necessary to select a hypervelocity impact problem to be modelled. The problem chosen was an 8 km/s normal impact of a sphere onto a thin plate. This was selected as both experimental and Lagrangian hydrocode results have been published [41]. Choosing a normal impact allows multiple planes of symmetry to be used, so lowering the CPU time needed for each calculation. As both projectile and target are made of aluminium, isotropic constitutive models can be used. This simplifies the material modelling and allows attention to be focused on how to model hypervelocity impact in a Lagrangian code.

The problem chosen was the impact of an aluminium sphere onto a structure protected by a double bumper shield. Details of the initial conditions and the state after impact on the first bumper shield are given in table 3.1. Only the impact onto the first bumper shield was considered, as unless this impact could be successfully modelled it was not worth considering subsequent impacts of the debris cloud onto the second shield and the main structure. The main and total debris cloud expansion angles give

Projectile diameter	4.0 mm
Bumper plate thickness	0.8 mm
Projectile Velocity	8.0 km s ⁻¹
Hole diameter in bumper plate	7.80 mm
Debris cloud velocity	6.76 km s ⁻¹
Main debris cloud expansion angle	29°
Total debris cloud expansion angle	56°
Pressure behind initial shock wave	1.2 Mbar

Table 3.1: Initial conditions and experimental results of impact on thin plate

the expansion angle with respect to the centre line of the main portion of the debris cloud and the maximum angle of any material that forms part of the debris cloud.

The published hydrocode results are from the two dimensional Lagrangian hydrocode EFHYD. In order to model the perforation of the bumper shields and the expansion of the debris cloud, rezoning and remeshing were used to prevent elements becoming too distorted. The use of these tools requires the analyst to stop the calculation several times. Previous work [42] has shown that an experienced analyst is required to successfully model impact using this procedure, as the published results could not be duplicated using the EFHYD code, as well as the DYNA2D code.

3.1 Modelling hypervelocity impact in DYNA3D

In this section the design of the finite element model and the use of DYNA3D to analyse the model is discussed.

A basic error that can be made is with the units used in the model. The only assumption that DYNA makes about units is that the set of units used is *consistent*. In a consistent set of units, all derived units are based on the same set of basic units. For example: for all the analyses presented in this thesis the basic set of units used was: cm-g- μ s. So for the units to be consistent, velocity is measured in cm/ μ s (1 cm/ μ s = 10 km/s), pressure is measured in g/cm μ s² (1 g/cm μ s² = 1 Mbar = 100GPa). If any dimensioned input value is not in the consistent set of units used, then the results of the analysis will be in error.

The model was created using the TRUGRID pre-processor, which is a commercial development of the INGRID pre-processor developed for the Lawrence Livermore family of codes, which includes DYNA3D. This pre-processor was chosen due to the ease with which it allows the meshing of solid spheres and cylinders.

The finite element mesh used for this chapter is shown in figure 3.1. As the impact is normal the problem is axisymmetric, two symmetry planes at right angles to each other were used. This reduced the total number of elements, reducing the computer time required to solve the problem. The two symmetry planes were kept at right angles to each other as 8 node brick elements were used, and it would have unnecessarily complicated the mesh to have less than a right angle between the symmetry planes. If modelling an oblique impact only one plane of symmetry could be used.

The size of the elements in the region of the impact was chosen to allow sufficient

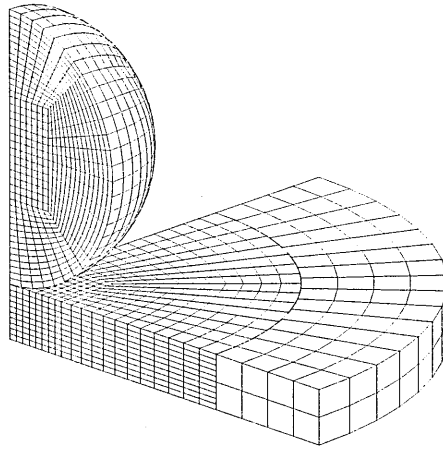


Figure 3.1: Finite element mesh for impact on thin plate

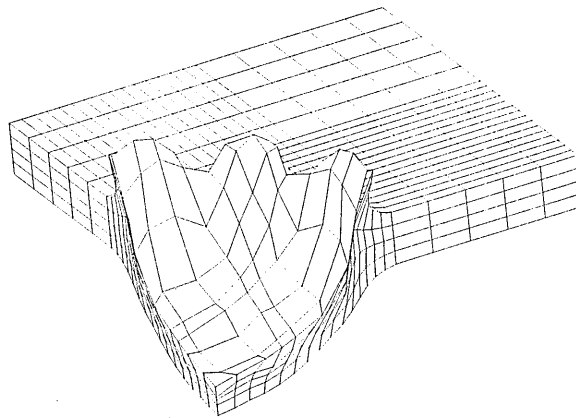


Figure 3.2: Effect of completely rectangular mesh on splash

resolution to model the shock wave propagation. As wave propagation is the dominant process during hypervelocity impact it is important to model it correctly. DYNA3D uses the shock viscosity method to model shock wave propagation, which results in each wave being spread over 3-4 elements. This gives a minimum limit on the number of elements to use. The upper limit is given by the computer time necessary to run the resulting model, as doubling the resolution in a 3D explicit code results in a 16 fold increase in run time. In this model ten elements were used through the plate thickness, as ten elements is sufficient to allow adequate resolution of the shock wave. The plate was meshed using a ‘butterfly’ mesh in order to get a cylindrical mesh away from the centre line of the impact. This was done as the orientation of the elements can have an effect on the geometry of the splash, see figure 3.2.

The outer boundary of the plate should be located at a sufficient distance from the impact, so that its location does not affect the results, and a non-reflective or ‘silent’ boundary condition can be applied to minimise wave reflection from the boundary.. This is not especially important when modelling a thin plate, as the wave propagation is naturally attenuated by the proximity of free surfaces. It is much more important

when modelling impact onto a semi-infinite target, where the assumption is that the target boundaries do not affect the impact in any way. An additional reason to place the boundary a distance away from the impact is for inertia reasons. In the real structure being modelled the plate extends out much further than is practical to mesh, and so resists any radial motion of the plate away from the impact. The boundaries should be placed far enough away from the impact that the lack of material does not affect the calculation, again this is more important when modelling semi-infinite impact. In order to reduce the number of elements in the model the outer portion of the plate has been modelled using coarse elements, attached to the finer mesh of the inner section using a tied slide-surface.

The spherical projectile was also meshed using a butterfly mesh. The mesh size was chosen to be similar to that of the plate. Due to the difficulty of changing the mesh resolution, the same basic element size was used throughout the sphere. In any case it would not be reasonable to change the mesh size too much, as a shock wave should pass through the whole of the projectile. The final mesh contains 8857 nodes and 7482 brick elements.

In hypervelocity impact the impact velocity often exceeds the speed of sound in the impacting materials. This means that when using a time step scale factor of close to one, even though the time step is theoretically small enough for stability, problems can occur. As the DYNA3D default (0.9, or 0.67 if using an explosive material) is too large, a lower scale factor must be specified. Following a suggestion by Benson [25] a scale factor of 0.1 was used.

3.1.1 Contact

A contact surface is used to model the interaction of projectile and plate. DYNA3D has two contact surfaces that can be used for hypervelocity impact. The first is 'sliding with separation and friction' (type 3), the second is 'SAND contact' (type 11). The difference between the two is that SAND allows for element erosion. Both algorithms are symmetric, so each surface is treated as master and slave each time step. Therefore one surface does not have to be meshed finer than the other.

When defining the two contact surfaces it is very important to get the orientation correct, even when using brick elements. DYNA3D calculates the outward normal vector of each segment from the order that the corner nodes are specified in the input file, using a 'right-hand rule'. Although the DYNA3D user manual [37] states that the outward normal is calculated automatically when using brick elements, and that the orientation is only important when using shell elements, experience has shown that this is not the case. Neither TRUGRID or INGRID will automatically get the orientation correct, but it can be explicitly defined when creating the slide surfaces in the pre-processor.

The default penalty scale factor in DYNA, 0.1, is too low for use with hypervelocity impact. With this value significant interpenetration occurs at the contact surface. From experience a scale factor of around 3 is reasonable for hypervelocity impact, though at the higher end of the debris impact velocity range it may be necessary to increase the penalty stiffness to the 5-10 range. Increasing the penalty stiffness higher does not bring any significant benefit with respect to enforcement of contact, and has an adverse

effect on the overall reliability of the calculation.

There are two other points to consider when setting up an analysis that improve the reliability of the contact algorithm. The first is to place a small gap between the two bodies, especially when nodes on a plane of symmetry will be in contact. In the case of the sphere-plate impact the first point where contact occurs is on the centre line. If the two nodes on the centreline start off co-located, contact is not always enforced properly. Placing a small gap, of the order of 1-10% of the element size was found to correct this. The second point that improves the reliability of the contact is to set a small initial time step size. An adequate value is about 10% of the time step size that DYNA initially calculates, this time step size can be found in the d3hsp output file. In the sphere-plate model presented, an initial time step of $1 \times 10^{-6} \mu s$ has been used. This does not have a significant effect on the overall run time as the time step grows until it reaches the critical time step calculated by DYNA. Setting a small initial time step increases the reliability of contact by keeping the magnitude of any early penetrations small.

In order to use element erosion the SAND (Slidesurface with Adaptive New Definitions) contact algorithm must be used. When setting up a model using the SAND algorithm, master and slave materials as well as contact surfaces, must be specified. This tells the code the regions of the model that contain elements that potentially can be eroded. At run time this information allows DYNA to redefine the contact surfaces when an element erodes. As the element erosion criterion is specified in the material model definition, erosion can only be used with the few material models that contain the erosion criterion.

3.1.2 Boundary and Initial conditions

As mentioned earlier, it may be necessary to put a non-reflecting boundary condition on the outer radial faces of the elements on the radial boundary of the plate. If modelling a semi-infinite impact then this boundary condition is even more important. The only initial condition required is an initial velocity for the projectile. Make sure that the sign of this initial velocity is correct, in the sphere-plate model the z-axis is normal to the plane of the plate, and is positive on the side that the projectile starts on. So the projectile has a negative initial z velocity.

In the model no translational constraints were placed on the outer nodes of the plate. If the aim was to find the final position of the plate in space, then it would be important to place constraints on the outermost nodes. As our interest was in the final hole diameter in the plate and the debris cloud produced by the impact, these constraints can be ignored as they do not affect the results of the calculation.

3.1.3 Material Models

DYNA3D offers a large range of material models, but few are suitable for modelling hypervelocity impact. As the hydrostatic behaviour is so important a separate equation of state is required. Only two of the available material models that can operate with an equation of state are designed to model behaviour at high strain rates. These two are the Steinberg-Guinan [43] and the Johnson-Cook [44] models. As the strain rates during the early stages of hypervelocity impact are of the order of $10^5 - 10^7 \text{ s}^{-1}$, strain rate effects should not be neglected.

The Steinberg-Guinan constitutive model is a strain rate independent model. Its basic assumption is that at stresses approaching 10 GPa and strain rates greater than 10^5 s^{-1} , the effects of strain rate can be ignored. These conditions occur during the early stages of hypervelocity impact. However the model is not valid at lower strain rates, which occur later in the impact.

The Johnson-Cook constitutive model is a strain rate dependent model. The strain rate range for which it is valid is not stated as explicitly, unlike the range for the Steinberg-Guinan model. In the original paper [44] test data at strain rates of up to 10^2 is used to generate the necessary constants. The strain rate dependent term in the constitutive model is $[1 + C \ln \dot{\epsilon}^*]$. At above strain rates of $5 \times 10^3 \text{ s}^{-1}$ the flow stress varies linearly rather than logarithmically with strain rate [45]. So this provides an upper limit to the validity of the Johnson-Cook model.

From this information the choice of which model to use is not clear, neither model is valid over the whole range of conditions that occur, and it is not possible to change the constitutive model used during a computation. For the analyses presented in this chapter the Steinberg-Guinan model was used as the element erosion algorithm was not available when using the Johnson-Cook model.

In addition to the strength model an equation of state is required to model the hydrostatic behaviour. DYNA contains ten equations of state, ranging from analytical to simple tabular, and covering high explosives as well as metals. The Gruneisen equation of state was selected as material input data was most easily available, and none of the available alternatives were superior for modelling hypervelocity impact.

3.2 Modelling hypervelocity impact without element erosion

The first analyses did not use element erosion. The aim of these analyses was threefold, to gain experience in modelling hypervelocity impact, to determine the point where the calculation stops due to element problems, and to show that the early stages of impact are modelled correctly.

If the early stages of impact are being modelled correctly, certain features will be seen: impact generated shockwaves in projectile and plate, reflection of shock wave in plate from free surface on other side of plate and the formation of the back-splash when shock waves propagate beyond the area of contact. In addition the magnitude of certain solution variables can be checked. From Hugoniot data [46] the pressure behind the shock wave and the shock velocity are known. For a pressure behind the shock of 1.2 Mbar, the shock velocity is approximately 10.5 km/s and the particle velocity is approximately 4.0 km/s. For a shock velocity of 10.5 km/s it will take the shock wave $0.076 \mu\text{s}$ to reach the far surface of the plate.

The DYNA3D results were in good agreement with the expected behaviour and values. The maximum pressure calculated is 1.17 Mbar, and at $0.075 \mu\text{s}$ the shock wave in the plate is just about to reach the rear of the plate. The expected wave propagation was seen, the shock wave in the plate reflected as an expansion wave from the rear of the plate. By $0.2 \mu\text{s}$ material from the plate was beginning to form the back-splash. Pressure contour plots of the analysis are shown in figures B.1–B.3.

The final state plot produced before the analysis crashed was at $0.25 \mu\text{s}$. As state plots were being taken every $0.025 \mu\text{s}$, DYNA3D crashed at a run time between 0.25

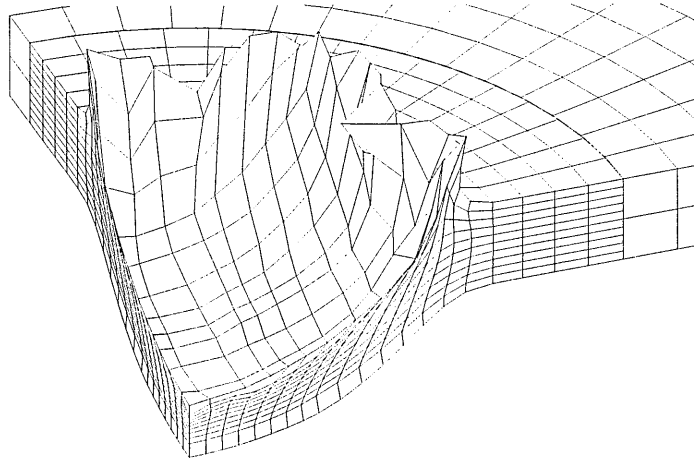


Figure 3.3: Bumper plate mesh at $0.25 \mu s$. No erosion.

and $0.275 \mu s$. Figure 3.3 shows the mesh at $0.25 \mu s$ with the projectile removed. The highly deformed elements that form the splash can be clearly seen, and these caused the end of the run.

These results showed that DYNA was capable of modelling the early stages of hypervelocity impact accurately. By showing that the shock wave behaviour can be modelled correctly, they provided a means of checking runs that used element erosion. As shock waves are the dominant process in hypervelocity impact, it is very important that they are modelled correctly. The same shock wave propagation should be seen whether or not element erosion is used.

3.3 Modelling hypervelocity impact with element erosion

The next step was to include element erosion in the model, to look at its effectiveness. Analyses were made using a range of values for the erosion criterion: 0.5, 0.75, 1.0, 1.5 and 2.5. These values bracket the range 1.0-2.0, which have been most commonly used in the literature [40].

The results of these analyses showed three main points:

- The effective plastic strain erosion criterion used with the Steinberg-Guinan constitutive model is not suitable for hypervelocity impact.
- The limit on the magnitude of the erosion criterion comes from the reliability of the contact surface.
- Element erosion is not suitable for modelling multi-plate impact, as it does not conserve mass and momentum.

3.3.1 Suitability of effective plastic strain erosion criterion

An area of concern that these analyses raised was the effect of element erosion on the formation and propagation of shock waves.

It was found that the use of element erosion had a major effect on the wave propagation during the early stages of impact, this can be clearly seen in figure 3.4. The expected wave propagation is seen in the case with no erosion. However with element erosion the shock wave has not yet reached the underside of the plate. On examination of the results from the very early stage of impact (0 to 0.03 μs) it was found that elements were being eroded before any significant deformation has taken place, and that changing the value of the erosion criterion had no effect, figures B.4–B.8. Even raising the criterion to very high values, 10–50, had no effect on the very early erosion.

When one of these undeformed elements was eroded, a significant gap was formed between the projectile and plate. The two surfaces of this gap acted as free surfaces, and so a release wave propagated away from the gap, interfering with the formation of the shock waves. It was these release waves interfering with the shock wave formation that causes the difference seen in figure 3.4.

The cause of this problem was traced to an incompatibility between the Steinberg-Guinan constitutive model, and the effective plastic strain erosion criterion. As implemented in DYNA3D the Steinberg-Guinan constitutive model allows the material in an element to melt [37]. This occurs when the element internal energy, E , exceeds the melting energy, E_m . The melting energy for each element is calculated each time step from the current conditions, and the material history. When the material in an element melts its shear strength, G , is set to zero, so that the material can behave like a fluid.

The equation used in DYNA to update the effective plastic strain is

$$\bar{\epsilon}_p^{n+1} = \bar{\epsilon}_p^n + \frac{(s^* - \sigma_y^n)}{(3G + 1 \times 10^{-10})} \quad (3.1)$$

where s^* is the effective trial stress and σ_y^n is the yield stress for that time step. The 1×10^{-10} term is present to prevent a division by zero error when the element melts. With the shear stress set to zero the increment of effective plastic strain becomes very large and consequently the calculated value of the effective plastic strain for the melted element becomes very large, orders of magnitude greater than 1. With element erosion the element is then immediately deleted for any practical value of the erosion strain.

This behaviour means that an effective plastic strain erosion criterion, used with the Steinberg-Guinan constitutive model, is not suitable for use in hypervelocity impact problems.

3.3.2 Contact reliability

Only with the lowest value of the erosion criterion, 0.5, were there no contact problems. The analyses using values of 0.75, 1.0, 1.5 and 2.5 all had problems with the reliability of the eroded contact surface. Figures 3.5 and 3.6 show the early stages of this problem for erosion criteria of 0.75 and 1.0. As the calculations progressed the results were obscured by the nodes and elements that had interpenetrated, making the results useless. At higher values the problem got correspondingly worse. The uneven contact surface resulting from element erosion was adversely affecting the reliability of the contact search algorithm, leading to the penetration of some nodes not being resisted.

The above results were found while using spall model 2. It was found that changing the spall model had a strong effect on these results. DYNA3D supports three spall

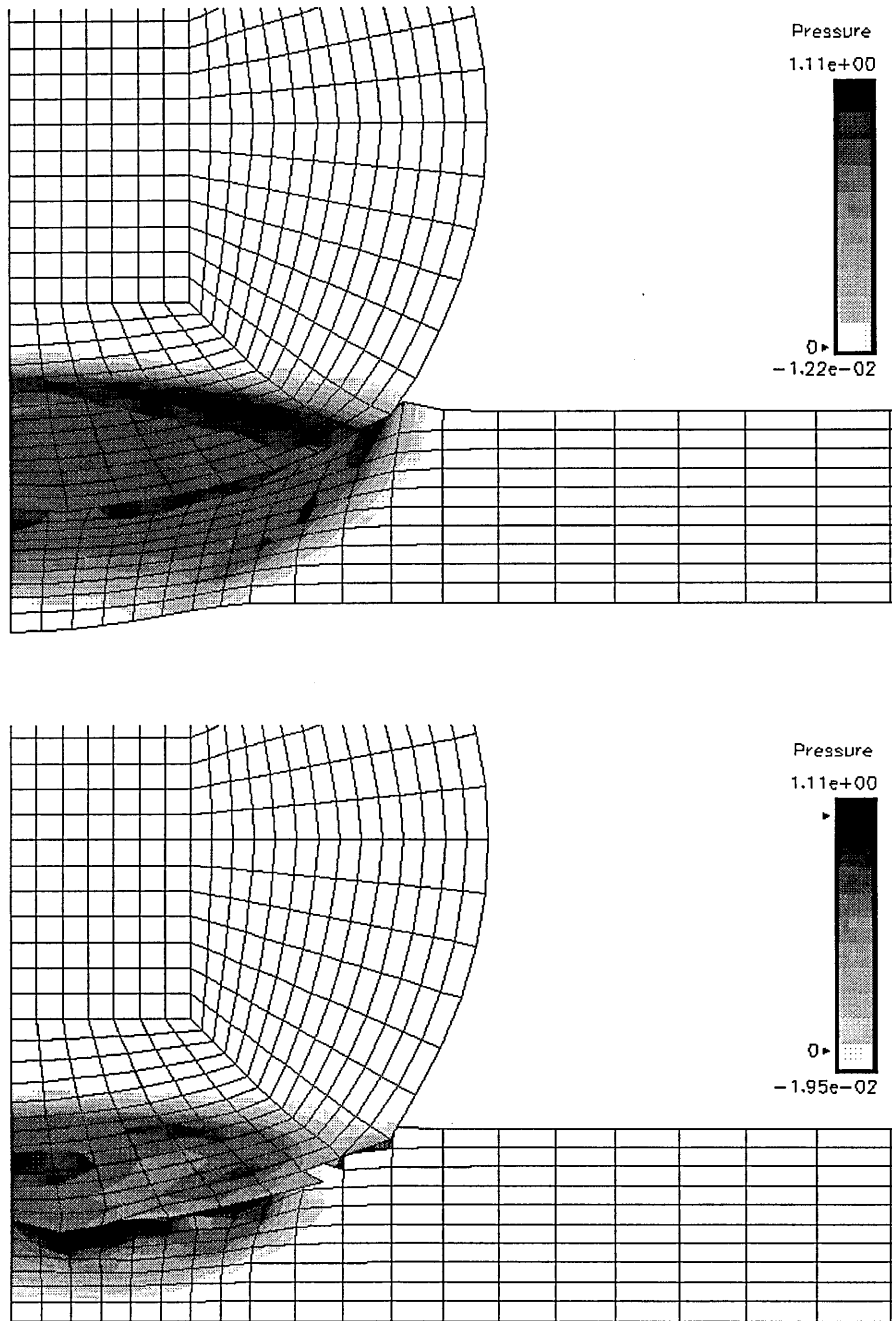


Figure 3.4: Comparison of shock wave propagation between the contact algorithm with no erosion (top) and with an erosion criterion of 2.5 (bottom). Both plots are at $0.1 \mu s$.

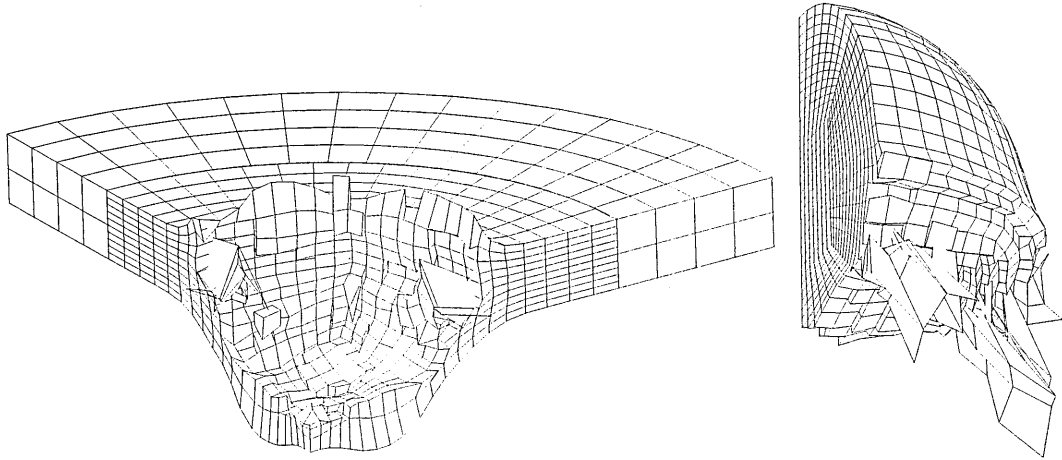


Figure 3.5: Plate and projectile mesh for erosion criterion of 0.75. Time $0.3 \mu s$

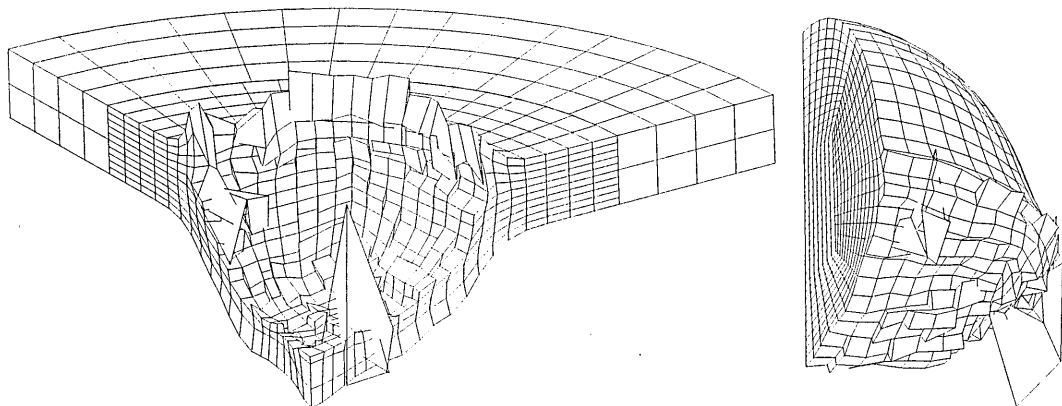


Figure 3.6: Plate and projectile mesh for erosion criterion of 1.0. Time $0.3 \mu s$

models with the Steinberg-Guinan strength model. For all three models there is a value of the cut-off pressure, P_{cut} , which is defined in the input file. For spall model 1, the pressure limit model, the hydrostatic tension is limited to P_{cut} , and the deviatoric stresses are not affected. For spall model 2, the maximum principal stress model, spall is detected if the maximum tensile principal stress exceeds P_{cut} . For spall model 3, the hydrostatic tension model, spall is detected if the pressure exceeds P_{cut} . For types 2 and 3, when spall is detected the deviatoric stresses are set to zero and the pressure is limited to compression only. Type 1 is not a proper spall model as the deviatoric stresses are not affected, and was not used.

A significant difference was seen between results using the type 2 and type 3 spall model. Figures 3.7 and 3.8 show the difference between results when using spall type 2 and type 3. With type 3 many more elements have been eroded from the calculation. Spall type 3 also suffered from contact problems at higher values of the element erosion, but the value of the erosion criterion where the results become useless is higher. Figures 3.9 and 3.10 show results for type 3 with an erosion strain of 1.0, compared with type 2 at an erosion strain of 0.5. Each of these results are for the highest value of erosion strain where reasonable results existed at $2.0\mu s$.

There are two causes for the difference between the two models. The first cause is that spall is detected differently. Type 2 detects spall if $P - P_{cut} < 0$. Type 3 detects spall if $P - s_{max} - P_{cut} < 0$, where s_{max} is the maximum principal tensile deviatoric stress. The second cause is that the effects of spall are applied at different points within the strength model subroutine. Equation 3.1 is used to calculate the new value of the effective plastic strain. The effective trial stress, s^* , is calculated from the trial value of the deviatoric stress tensor, $*s^{\alpha\beta}$, which is the elastic update of the deviatoric stress tensor:

$$s^* = \left(\frac{3}{2} *s^{\alpha\beta} *s^{\alpha\beta} \right)^{\frac{1}{2}} \quad (3.2)$$

The trial stress is then adjusted if yield has occurred. Spall type 3 is considered between the calculation of the trial stress, and the calculation of the effective trial stress. So if spall occurs the deviatoric stress is set to zero before the effective trial stress is calculated, and so it is zero. Spall type 2 is considered at the end of the routine, where if spall occurs the updated value of the deviatoric stress tensor is set to zero, so s^* can be non-zero. For an element in which spall has occurred the increment of effective plastic strain is different between spall types 2 and 3.

These effects combine to produce a lower effective plastic strain when using spall type 2 in elements where spall occurs. This results in fewer elements being eroded from the calculation, as can be seen in figures 3.7 to 3.10.

3.3.3 Modelling multi-plate impact with element erosion

When modelling hypervelocity impact on spacecraft structures the information that an engineer requires is probably not the result of the first impact. With impact on an unprotected structure the engineer wants to know if penetration occurs, and if so what internal systems are at risk from the debris cloud. With impact on a protected structure, the impact of the debris cloud on the structure being protected is of most interest.

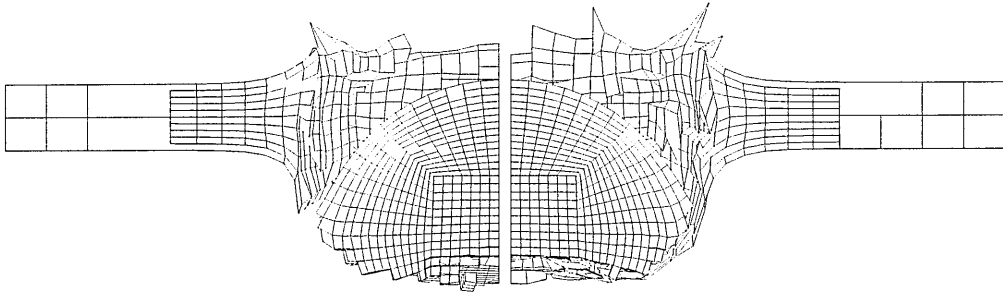


Figure 3.7: Comparison of results using spall type 3(left) and spall type 2 (right), both analyses are for an erosion criterion of 0.5.). Time $0.5\mu s$.

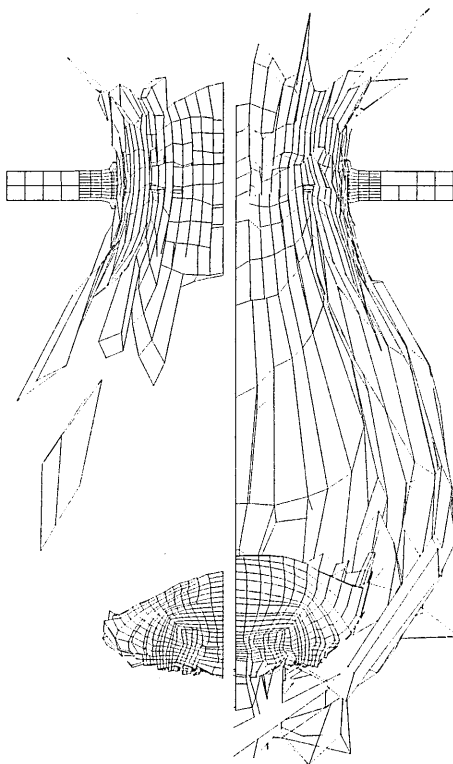


Figure 3.8: Comparison of results using spall type 3(left) and spall type 2 (right), both analyses are for an erosion criterion of 0.5. Time $2.0\mu s$.

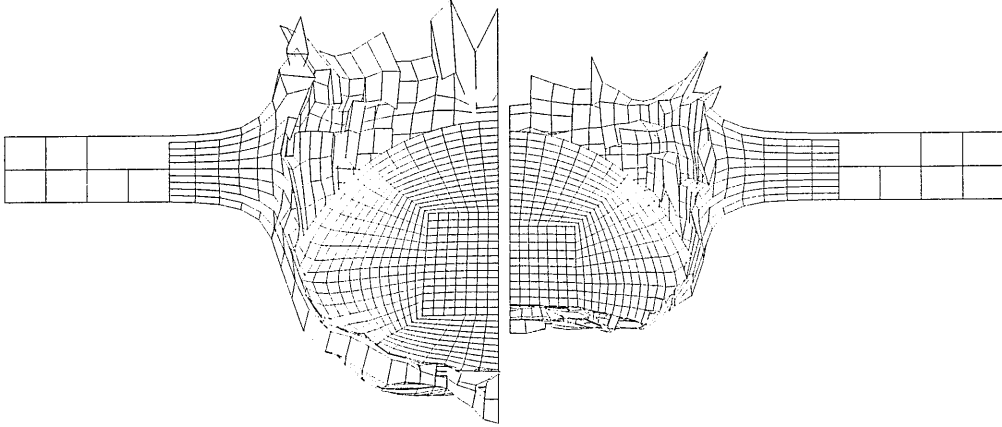


Figure 3.9: Comparison of results using spall type 3 with erosion criterion 1.0 (left) and spall type 2 with erosion criterion 0.5 (right). Time $0.5\mu s$.

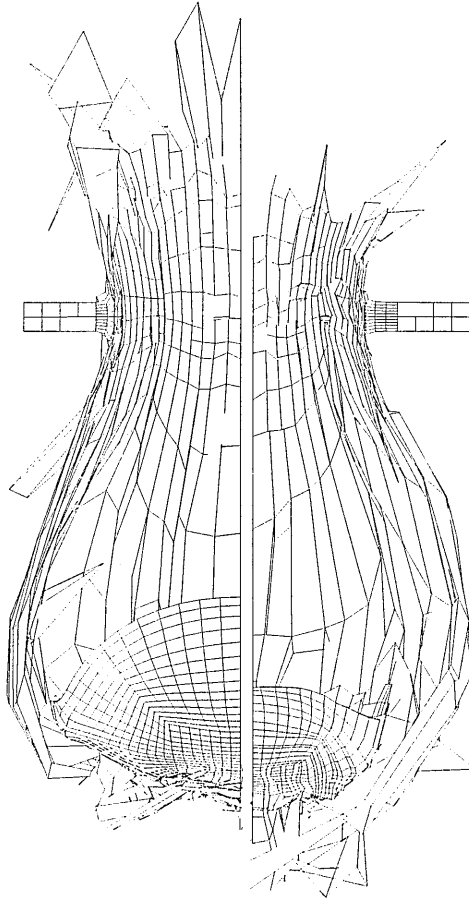


Figure 3.10: Comparison of results using spall type 3 with erosion criterion 1.0 (left) and spall type 2 with erosion criterion 0.5 (right). Time $2.0\mu s$.

In figure 3.10 the debris cloud can be clearly seen. In order to model the penetration of the plate 30-40% of the elements forming the projectile have been eroded, and approximately 80% of the plate elements that should form part of the debris cloud have been eroded. This means that a significant part of the material that forms the debris cloud has been discarded from the calculation.

The mass that has been lost should form the leading portion of the debris cloud. From experimental studies [47] this portion of the debris cloud has the highest velocity and density. So its loss has a significant effect on the loading experienced during a second impact and hence the results of this impact.

3.4 Discussion

From the experience gained with these analyses three areas were highlighted where improvements are required if the code is to be used to model hypervelocity impact on spacecraft structures. These were:

- Improve the material models available.
- Improve the element erosion capability.
- Improve the capability to model the debris cloud.

3.4.1 Improving the material model

During hypervelocity impact materials are subjected to a wide range of temperatures, pressures and strain rates. The strength model needs to be valid at the very high strain rates during the early stages of impact, as well as being valid later in the impact where material strength is important in determining the final hole diameter or crater size. In addition if impact of the debris cloud on subsequent plates is considered there are more failure mechanisms to consider than just spall. Impact on subsequent bumper shield can result in the failure of the shield by petalling and tearing. Impact of the debris cloud onto the main plate can result in the plate deforming by bending and shear. Modelling these processes requires a model valid for lower strain rates.

None of the constitutive models or equations of state available in DYNA3D are completely suitable for modelling hypervelocity impact. There are only two constitutive models available that are designed for high strain rates and the use of an equation of state. These are the Steinberg-Guinan model and the Johnson-Cook model. Of the two the Johnson-Cook has potentially the most validity, as it is designed for lower strain rates, and it is at these lower strain rates where material strength is most important during hypervelocity impact.

Where the hydrostatic behaviour is considered, the material can be multiply shocked when impact on more than one plate is considered. The material which forms the debris cloud can have a low density near the boundaries, as well as potentially being in the liquid or gas phases.

None of the equations of state in DYNA can reliably cover all these conditions. There are known problems with the Gruneisen type equation of state for the expanded liquid and vapour regions, like a debris cloud, and for multiply shocked states [48]. There

is no suitable alternative available in DYNA, the other equations of state available are either based on polynomials or are very simple tabulated (10 points defining a curve) equations, and offer no advantage over the Gruneisen equation for hypervelocity impact.

Developing a new constitutive model or equation of state is a major piece of work in itself, and is outside the scope of this thesis. So any improvements in this area would have to come from implementing already existing models into DYNA.

There are very few alternative strength models in the literature, and the available models are either improvements of or based on either the Steinberg-Guinan or Johnson-Cook models. Steinberg and Lund [49] have produced a modified version of the Steinberg-Guinan model, extending its validity to strain rates of 10^{-4} s^{-1} . For the Johnson-Cook model it has been suggested that replacing the term containing the natural logarithm of the strain rate with either a linear [45] or an exponential [50] results in a better representation of the strain rate effect. Zerilli and Armstrong [51] have developed a model, based on the Johnson-Cook model, that uses different equations for face-centred cubic and body-centred cubic metals. A major difficulty with all these models is the availability of input data for a range of materials. Even for the Steinberg-Guinan and Johnson-Cook models there is very little data to add to that given in the original papers. Without material constants for materials used on spacecraft, especially aluminium, there is no benefit in implementing any of these strength models.

For equations of state there is a wider range of options available, there are several alternatives that are widely used in hydrocodes. These range from other simple analytical equations, such as the Tillotsen EoS which includes a simple phase transition. Through more complex analytical equations of state like the ANEOS package used in the CTH code [52]. Finally to tabulated equations of state like SESAME [53]. As for the strength models, any choice will be strongly influenced by the availability of material data.

3.4.2 Improving the element erosion capability

The use of element erosion in DYNA for modelling hypervelocity impact is limited by its only being available for the Steinberg-Guinan model, and by the too early erosion of melted elements when used with this model. Element erosion is capable of producing useful results, especially for cratering [54, 55]. Improving the element erosion capability of DYNA would represent a useful improvement to the code.

Two areas need to be looked at. The first is allowing element erosion to be used with the Johnson-Cook model. The second is to look at alternative erosion criteria to determine if there are more suitable criteria than effective plastic strain.

3.4.3 Improving the modelling of the debris cloud

In the calculations presented in this chapter, the use of element erosion to allow modelling of the perforation of the thin plate has resulted in a significant portion of the material that should form the debris cloud being eroded, and thus removed from the calculation. This loss of mass and momentum, which is concentrated in the higher density forward area of the debris cloud, means that modelling the subsequent impact of the debris cloud can not be done accurately.

The obvious solution to this is don't throw any mass away, but, due to the problem of mesh tangling, this solution is not possible using the Lagrangian finite element method. The possible alternatives or additions to the Lagrangian finite element method are:

- Element erosion
- Remeshing or Adaptive meshing
- Eulerian
- Arbitrary Lagrangian Eulerian (ALE)
- Meshless Lagrangian method

All these approaches have their advantages and disadvantages, which one offers the best approach will depend on what the capabilities of the final code need to be.

The aim of this work is to produce a robust and accurate method for modelling hypervelocity impact on spacecraft structures. A method that does not require the intervention during the analysis of an experienced analyst to gain satisfactory results. The final code needs to be 3D, so that both normal and oblique impacts can be considered. In order to use more sophisticated material models, material history variables must be calculated. If the method is to be used for practical engineering then it must not require a super-computer to have reasonable run times. It should be capable of producing reasonable results on a workstation.

The method should be capable of being combined or coupled with the Lagrangian finite element method so that structural elements can still be used. Satellites are thin-walled structures, so using shell elements where the hydrodynamic behaviour of the material is not important offers significant savings in CPU time. Looking outside the application of hypervelocity impact, adding the capability to model arbitrarily large deformations to a code like DYNA3D would allow modelling of some problems that are currently very hard or impossible to model accurately. A good example of this class of problem is aircraft impact onto soft soil or water. The aircraft structure is modelled with shell elements. While the soil or water needs to be modelled with a method that is accurate for large deformations.

To summarise, the ideal is a 3D method which can model large deformations, allows material history variables to be easily calculated, can be linked or coupled with the existing Lagrangian finite element method, and is computationally efficient.

Element erosion

This approach has already been looked at. For calculating the results of the first impact only it appears a suitable approach. With a suitable criterion that only removes the minimum required number of elements it may allow reasonable results for the debris cloud to be gained, however mass and momentum will still have been lost from the calculation. For this reason element erosion does not offer all the capabilities that are required.

Remeshing or Adaptive meshing

Remeshing is a process where the calculation is halted, a new mesh is constructed within the current boundaries of the material, the solution variables are remapped from the old mesh onto the new mesh and the calculation is restarted. Successful remeshing will require the intervention of the analyst, this is because the new mesh must be designed to prevent elements becoming highly distorted again too fast. Intervention is also required to allow material to separate, for instance the debris cloud from the bumper shield. In addition there is difficulty in producing a good mesh in 3D for potentially very complex geometry. Remeshing is not a suitable solution.

The final approach possible while retaining the Lagrangian finite element method is adaptive meshing, or the Free Lagrange method. It was originally developed for fluid flow problems where there is significant relative motion, especially across interfaces, and where accurate tracking of these interfaces is important. The need for accurate handling of fluid interfaces led to a Lagrangian technique being developed, but due to large deformations, dependence on a fixed mesh had to be removed. At its simplest the Free Lagrange method is an approach where a new 'best' finite element mesh is generated every time step, or few time steps. However the codes are commonly more sophisticated, allowing the addition of nodes in order to enhance resolution where required, and the removal of nodes where the resolution is better than required.

The Free-Lagrange method appears suitable. It is a Lagrangian method so material history variables can be calculated, and boundaries are explicitly defined, so it can be easily combined with non-adaptive finite elements. It is more computationally expensive due to regularly generating a new mesh.

Eulerian

The Eulerian approach has been widely used in the hypervelocity impact field. As it does not have problems modelling large deformations or penetration it is capable of modelling impact on multi-plate structures. The principal disadvantage to the Eulerian approach is CPU time. Special algorithms need to be used to track material interfaces, and second order accurate advection schemes are required if errors when dealing with low density flow are to be avoided. In addition Eulerian cells must be placed everywhere where material might be at any time during the calculation, this requires a greater number of cells than the Lagrangian approach in order to achieve the same spatial resolution. The Eulerian method does not have an equivalent to structural elements, and so is not suitable for modelling problems involving thin-walled structures.

Due to its advantages there has been considerable interest in combining the Eulerian and Lagrangian methods to produce a code with the advantages of both. Two main approaches have been taken: to remap material between Eulerian and Lagrangian meshes using each approach where suitable, and to allow contact between Eulerian and Lagrangian elements. The first approach has been successfully carried out in 2D, with an interface between CTH and DYNA2D being developed [56]. This approach does require analyst intervention, with the calculation in one code being stopped and a conversion program being run before calculation continuing using the other code. The second approach requires a contact algorithm between Lagrangian and Eulerian mesh. This is a complex problem as the Lagrangian mesh will cut through Eulerian cells. This

approach has been used in codes like DYTRAN [57].

Both the approaches using the Eulerian method are possible. However both are complex to implement and are more computationally expensive than the Lagrangian approach.

Arbitrary Lagrangian Eulerian

The Arbitrary Lagrangian Eulerian (ALE) method is another attempt to combine the best features of the Lagrangian and Eulerian methods. It is a method that allows the mesh to distort, while also allowing material to advect between elements. Two main forms of ALE have been used, single-material and multi-material. In single-material ALE each element can only contain material of one kind, and so it behaves in a manner very similar to the Lagrangian method. It acts as a mesh smoothing method, and delays the onset of large deformation problems but does not cure them [42]. From this 2D experience single-material ALE does not offer a solution. Multi-material ALE is much closer to the Eulerian method, with each element able to contain more than one material. It offers a way of refining the Eulerian mesh, allowing the areas of high mesh density to move with material as required, and allows the mesh to be distorted to better track material boundaries. Multi-material ALE has been successfully used to model hypervelocity impact [29]. However it suffers from the same disadvantages as the Eulerian method, though it is computationally more efficient as the mesh can move with the material.

Meshless Lagrangian Methods

These are Lagrangian methods, so the computational nodes move with material, but they do not use a structured computational mesh. Instead a kernel interpolation is used, summing contributions from all nodes in the neighbourhood of the point being considered. The main methods that have been developed for solid mechanics applications are Smoothed Particle Hydrodynamics (SPH) [58], Reproducing Kernel Particle Method [59] and the Element Free Galerkin Method [60]. Of these SPH is the simplest and the only one currently applied to hypervelocity impact [34]. The advantages of SPH are: no computational mesh means that large deformations do not cause problems, it is Lagrangian so material histories are easily computed, with efficient neighbour searching it is not a computationally expensive method, and it can be linked to the finite element method using modifications of standard Lagrangian contact algorithms [61]. Its main disadvantage is that it is a comparatively young method and requires further development.

After considering the available options, the Meshless Lagrangian approach was selected as the most promising. The Eulerian and ALE approaches were comparatively computationally expensive, and linking them with the Lagrangian finite element approach is complex. Remeshing is a solution that requires the intervention of an experienced analyst. The Free Lagrangian method offers similar advantages to the meshless method, however the need to repeatedly generate a new mesh makes the Free Lagrangian method more complex, and therefore less attractive.

3.5 Summary

The capability of unmodified DYNA3D for modelling hypervelocity impact on a thin plate has been assessed. Element erosion has to be used in order to model perforation of the plate. An incompatibility between the effective plastic strain erosion criterion and the Steinberg-Guinan strength model resulted in the early stages of impact being modelled incorrectly. In addition the use of element erosion resulted in too much material that should form part of the debris cloud being deleted from the calculation. This would prevent accurate modelling of subsequent impacts.

Three principal conclusions were drawn:

- Improve the material models available.
- Improve the element erosion capability.
- Improve the capability to model the debris cloud.

For improving the material models it was decided to implement a new equation of state, more capable than the equations of state implemented in DYNA. To improve the element erosion capability alternative criteria needed to be investigated. To improve the capability to model the debris cloud it was decided to pursue the meshless Lagrangian approach.

Chapter 4

Implementation of the SESAME equation of state

In chapter 3 it was concluded that DYNA3D required a new equation of state if it is to be used in modelling hypervelocity impact. This chapter covers the choice of the new equation of state, its implementation into DYNA3D, and the validation of the implementation.

The chapter is split into the following sections:

- 4.1 The SESAME equation of state
- 4.2 Implementation in DYNA3D
- 4.3 Validation of implementation
- 4.4 Summary

A suitable equation of state should have the following characteristics. It must be accurate over the range of conditions met during hypervelocity impact, it is suitable for implementation into a hydrocode, there is data for a range of materials and finally the equation and material data must be available.

Equations of state that are used in hydrocodes can be split into three principal groups: simple analytical, complex analytical and tabulated.

The existing equations of state in DYNA3D were principally of the simple analytical form, consisting of one or two equations that use experimental Hugoniot data to provide the input data. The Tillotson equation of state was the only candidate looked at from this category, as it accounts for a solid-gas phase change. The Tillotson equation of state is [24]:

$$\begin{aligned} P &= P_{\pi} + A\mu + B\mu^2 \\ P_{\pi} &= E\rho \left[a + \frac{b}{E/E_0\eta^2 + 1} \right] \quad \rho \geq \rho_0 \end{aligned} \quad (4.1)$$

where A , B , a and b are material constants, $\eta = \mu + 1$, and P_{π} is valid only for increased density and internal energy less than the sublimation energy.

There are two other equations of state that are commonly used in hydrocodes when an accurate equation of state is required, these are ANEOS and SESAME. ANEOS, developed at Sandia National Laboratories, is an example of a complex analytical equation of state which uses a series of analytical formulae developed from experimental data and physics models. SESAME, developed at Los Alamos National Laboratory, is a tabulated equation of state which uses interpolation from a data table to calculate the required pressure.

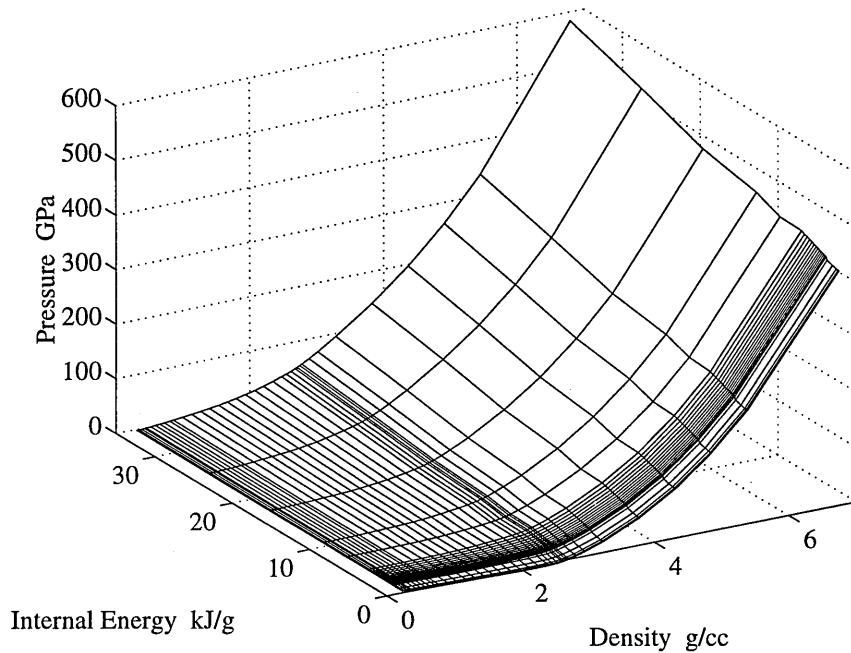


Figure 4.1: Graphical representation of a SESAME aluminium equation of state

The SESAME equation of state was available together with a wide range of material data. The ANEOS equation of state was not available. An additional advantage of SESAME is the ease of adding new materials. If a definition of the density-energy-pressure surface is available, it can be added simply by placing the data in the correct format.

4.1 The SESAME equation of state

The SESAME equation of state[53] consists of the actual database of materials and a set of Fortran subroutines which use the library. The actual database is a large text file containing data for about 150 equations of state that cover a wide range of materials. Each equation of state consists of a grid of density and temperature points, for each point the pressure and energy is specified. In addition to the equation of state surface, some materials have vaporisation, melt and shear tables. For many materials there is more than one equation of state in the database, as an example there are 6 entries for aluminium. These differ in the range of conditions covered, the resolution of different regions and the physics models used in the construction of the data table.

The Fortran routine library supplied with the data tables includes routines that support a binary format for the data tables that is more compact, and convert the data tables into a form that allows computation of pressure from density and internal energy, a form commonly used in hydrocodes. It also includes routines that calculate a point on the surface by interpolation from the surrounding data points. These supplied routines are suitable for inclusion in a hydrocode, so the principal task when adding SESAME to a hydrocode is to write subroutines which interface between the SESAME routines and the hydrocode.

An example of a SESAME equation of state is shown in figure 4.1. This shows part of the data for SESAME material number 3717, an aluminium equation of state. Each intersection of the gridlines represents a point in the data table, and variations in the resolution of the surface can be clearly seen. The section shown covers the range of conditions that would be found in orbital debris impact. The equation of state data is generated from a range of physics models, different models being used to generate different sections of the surface. A short description of the generation of a SESAME equation of state for lead is given by Holian and Burkett[62].

4.2 Implementation in DYNA3D

The basic steps to implementing a new equation of state into DYNA3D are:

- Modify input routines to recognise the new equation of state and load in material constants required.
- Modify restart routines, if necessary, to save and load the equation of state data to and from restart files.
- Write equation of state specific routines

In DYNA3D an equation of state is used at three points within the code. The first point is during the initialisation of the Steinburg-Guinan strength model and so is only required if this model is being used in an analysis. The second point it is used is in the calculation of the speed of sound in an element, which is calculated for every element, every time step. The third point is in the calculation of the new element pressure, also calculated for every element, every time step. Where possible the routines from the SESAME subroutine library were used.

Input and Restart

The SESAME equation of state was implemented in DYNA3D as equation of state 12. The information required from the input file is the identity number of the material to be used, for example 3717, the initial conditions and unit conversion factors. The conversion factors are required as SESAME data has units, and DYNA3D only assumes that a consistent set of units is used. So the code must know how to convert the SESAME data into the set of units used for the problem. The SESAME data tables use the following units:

Density	Mg/m ³
Temperature	K
Pressure	GPa
Energy	MJ/kg

For example, if the problem uses the cm-g- μ s set of units, the unit of pressure is cm/(g μ s²) which is equal to 100 GPa. This input data is stored in the `eosp` array, which is the standard location for storing equation of state input data.

The requested equations of state must be loaded into memory. As implemented, all required data tables must be contained in a binary file, filename `sesame.dat`,

located in the same directory as the input file. This file uses a modification of the standard SESAME binary format. The standard format uses end-of-file (EOF) markers to signify the end of one table and the beginning of another. This did not work reliably so it was necessary to add a table of contents at the start of the binary file to allow the correct data to be located within the file.

The only modification required to the DYNA restart subroutines was to load the correct data tables when a SESAME equation of state was used.

Solution phase

The first point in the main solution loop where an equation of state is used is to calculate the speed of sound in the element. Following the DYNA naming convention the new routine was called `sueos12`. DYNA uses the following equation to calculate the speed of sound:

$$c = \left[\frac{4G}{3\rho_0} + \frac{\partial P}{\partial \rho} \Big|_E + \frac{PV^2}{\rho_0} \frac{\partial P}{\partial E} \Big|_\rho \right]^{\frac{1}{2}} \quad (4.2)$$

The equation of state is required to calculate the pressure and the pressure derivatives. The SESAME library routine `s2eos1` was used to interpolate the pressure and its derivatives from the data table. This routine uses a rational function interpolation [53] to compute the pressure and the pressure derivatives.

The actual value calculated by a `sueos` subroutine is $\rho_0 c^2$, this is stored in array `cb()` which is passed as the second entry in common block `aux35`.

The second point where an equation of state is used is to update the pressure and internal energy. This was implemented as subroutine `eqos12`. The subroutine is required to update both pressure and internal energy due to the central difference integration rule used. The internal energy is updated using:

$$e^{n+1} = e^n - \sigma^{n+1/2} \dot{\epsilon}^{n+1/2} \quad (4.3)$$

As the pressure is evaluated at time n and $n+1$ this must be rewritten:

$$e^{n+1} = e^n - \frac{1}{2} \left(P^{n+1} + P^n \right) \Delta V + V^{n+1/2} s^{n+1/2} \dot{\epsilon}'^{n+1/2} \Delta t^{n+1/2} \quad (4.4)$$

Since P^{n+1} is a function of E^{n+1} this equation is implicit. This problem is solved by first calculating a trial value of the internal energy, e^* , using only the known terms on the right hand side of equation 4.4. If the equation of state is linear in the internal energy an explicit equation for P^{n+1} in terms of e^* can be found, see Benson [25]. Where the equation of state is not linear in internal energy, as for SESAME, DYNA uses a one step iteration.

$$P^* = P(V^{n+1}, e^*) \quad (4.5)$$

$$e^{n+1} = e^* - \frac{1}{2} \Delta V P^* \quad (4.6)$$

$$P^{n+1} = P(V^{n+1}, e^{n+1}) \quad (4.7)$$

In the DYNA code the trial value of the energy is calculated in subroutine `hieupd`. The calculation of the new energy and pressure using the SESAME tables was implemented as subroutine `eqos12`, again following DYNA naming conventions.

When using the SESAME equation of state it is necessary to perform three interpolations for every element, every time step. This causes SESAME to be a computationally expensive equation of state to use when compared to the existing DYNA equations of state. To investigate the actual effect that SESAME has on the CPU time two calculations were performed on a SPARC 20 workstation, one using the Gruneisen equation of state and one using SESAME. Both calculations used the same mesh, containing 2419 nodes and 1812 solid elements, and were run to the same solution time. The only difference between the two calculations was the equation of state used. The CPU times measured are shown in table 4.1. Using SESAME has resulted in a 35% increase in CPU time.

	Number of timesteps	CPU time (seconds)
Gruneisen equation of state	707	501.8
SESAME equation of state	714	681.8

Table 4.1: CPU timings using Gruneisen and SESAME equations of state

The advantage in using SESAME comes when an element has undergone significant volumetric deformation. In order to minimise the computational penalty SESAME was implemented in conjunction with the existing Gruneisen equation of state, which is accurate at lower pressures. The Gruneisen equation is used initially for all elements, an element changes to using SESAME once the element reaches a certain level of compression, arbitrarily chosen to be at an excess compression, μ , of 0.1.

Initialisation phase

The Steinberg-Guinan material model requires knowledge of the cold compression energy curve, or zero kelvin isotherm, to calculate the shear modulus and yield stress. The following equation is used to calculate the cold compression energy, E_c [37]:

$$E_c(x) = \int_0^x P dx - \frac{900 R' e^{ax}}{(1-x)^{\gamma_1 - a - 1/2}} \quad (4.8)$$

$$\text{where } x = 1 - \frac{\rho_0}{\rho} \quad \text{and} \quad R' = \frac{R\rho}{A_r}$$

R is the universal gas constant, A_r is the atomic weight and a is the coefficient of first-order volume correction.

As this equation is too computationally expensive to evaluate each step of the calculation, the curve is evaluated once, and a polynomial approximation is generated. The equation of state is required to evaluate the pressure integral term, where $P(x)$ is the pressure along the 300K adiabat. The integral is evaluated using the trapezoidal rule,

this requires the pressure on the adiabat to be found at multiple values. Each value of P can be found from the previous point by noting that along an adiabat:

$$dE = -PdV \quad (4.9)$$

Finding the value of the new pressure requires calculation of the new energy, resulting in an implicit equation. This is solved using the same technique used in the main calculation. For an equation of state which is not linear in internal energy a one step iteration is used, equations 4.5–4.7. The calculation of the new P term was implemented as subroutine `eos12`, following the DYNA naming convention.

4.3 Validation of implementation

The implementation was validated by calculating the Hugoniot curve for aluminium from a series of DYNA analyses, and comparing it to the curve calculated directly from the data tables. The aim of this was not to check the accuracy of the SESAME data, but to show that DYNA was correctly using the data. The Hugoniot curve was chosen as the check as it is dependent only on the equation of state, and it can be calculated from suitable DYNA analyses, as well as directly from the equation of state data.

The Hugoniot curve can be calculated from an equation of state by the solution of the Hugoniot equations:

$$\frac{\rho_1}{\rho_0} = \frac{u_s}{u_s - u_p} \quad (4.10)$$

$$P_1 - P_0 = \rho_0 u_s u_p \quad (4.11)$$

$$E_1 - E_0 = \frac{1}{2}(P_1 + P_0) \left(\frac{1}{\rho_0} - \frac{1}{\rho_1} \right) \quad (4.12)$$

These equations relate the density, pressure and energy across a plane normal shock wave in terms of the shock velocity, u_s , and the particle velocity, u_p . The subscript 0 refers to the state before the shock, the subscript 1 refers to the state behind the shock.

When calculating the principal Hugoniot curve the initial pressure and energy can be assumed to be zero, as it is assumed that the material is at room temperature and pressure. The equation of state and equation 4.12 give two relations between density, pressure and energy, any point which satisfies both lies on the Hugoniot curve. As SESAME consists of a data table each calculated point must be found by iteration. The supplied Fortran routines include a routine for finding a point on the Hugoniot as an example of the use of the SESAME routines.

Experimental Hugoniot data is gathered by the measurement of any pair of variables in equations 4.10–4.12. The simplest Hugoniot experiment consists of impacting two flat plates of identical material. If one is stationary then the particle velocity is one half the impact velocity, and the shock velocity can be found by measuring the time of arrival of the shock wave at a known point in one plate. The principal Hugoniot curve gives the state behind a plane normal shock wave, effectively a one dimensional

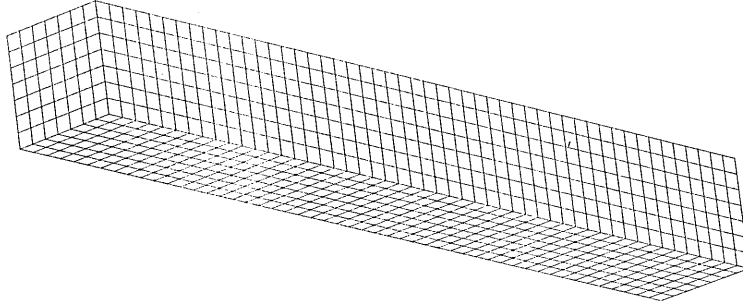


Figure 4.2: Mesh used for Hugoniot calculation

case, so the edges of the two plates must be far enough away from the measurement points to ensure that the release waves from the edges have not affected the shock wave propagation. Each of these experiments generates one point on the Hugoniot curve.

As in an experiment, a single DYNA analysis will produce one point on the Hugoniot curve. The mesh used is shown in figure 4.2. As it is a 1D problem four planes of symmetry were used, the contact with the second plate was represented by a rigid wall. The mesh represents a section of an infinite plate impacting a plate of identical material which has an equal but opposite velocity. The use of the symmetry planes and the rigid wall allowed a fine mesh to be used, while still retaining a fast run time. The mesh is finer than required for adequate shock wave resolution, the fine mesh was chosen in order to prevent mesh sensitivity effects. Time histories were taken from nodes and elements on the centreline of the mesh, allowing the necessary measurements to be taken.

In order to calculate a point on the Hugoniot curve two measurements are required. The simplest to make is the particle velocity, u_p , which is equal to the initial velocity of the mesh. Pressure is also easily measured from the element time histories. From the pressure and particle velocity the density and shock velocity can be easily calculated:

$$\rho = \frac{\rho_0 P}{P - \rho_0 u_p^2} \quad (4.13)$$

$$u_s = \frac{P}{\rho_0 u_p} \quad (4.14)$$

Twelve calculations were performed with the initial velocities in the range of 0.1–0.8 cm/ μ s (1–8 km/s). These velocities gave conditions equivalent to orbital debris impacts over the range 2–16 km/s, which covers the range of probable impacts on satellites in orbit. The material modelled was aluminium, and SESAME data table 3717 was used as the equation of state. Figure 4.3 shows a comparison between the Hugoniot curve calculated directly from the SESAME data, and experimentally determined Hugoniot points for aluminium taken from Marsh [46] and Mitchell et al. [63].

For each calculation, pressure and nodal velocity time histories were taken from five points within the mesh, with a distance of 0.1cm between these points. Examples of these time histories can be seen in figures B.10–B.12. In addition to pressure and

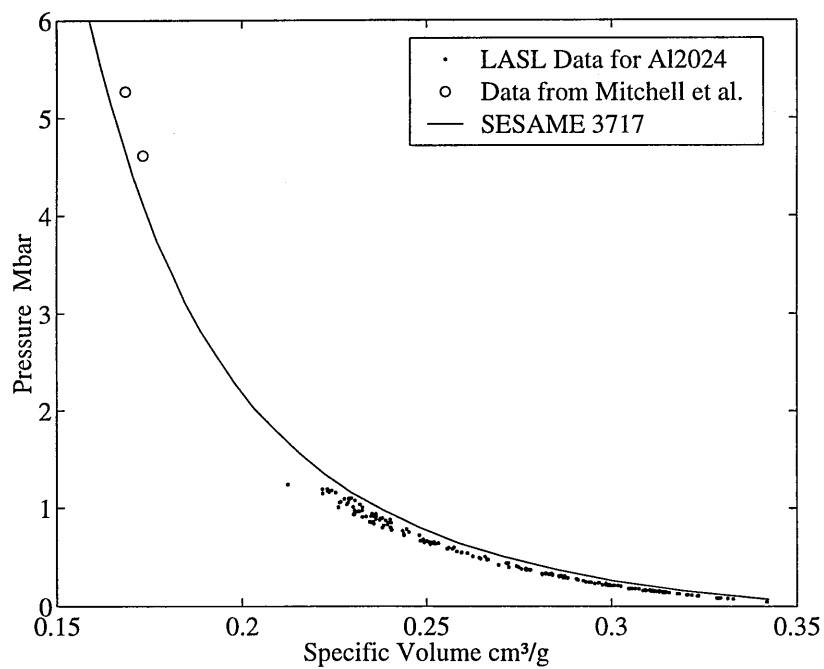


Figure 4.3: Comparison of Hugoniot curve for SESAME 3717 with experiment

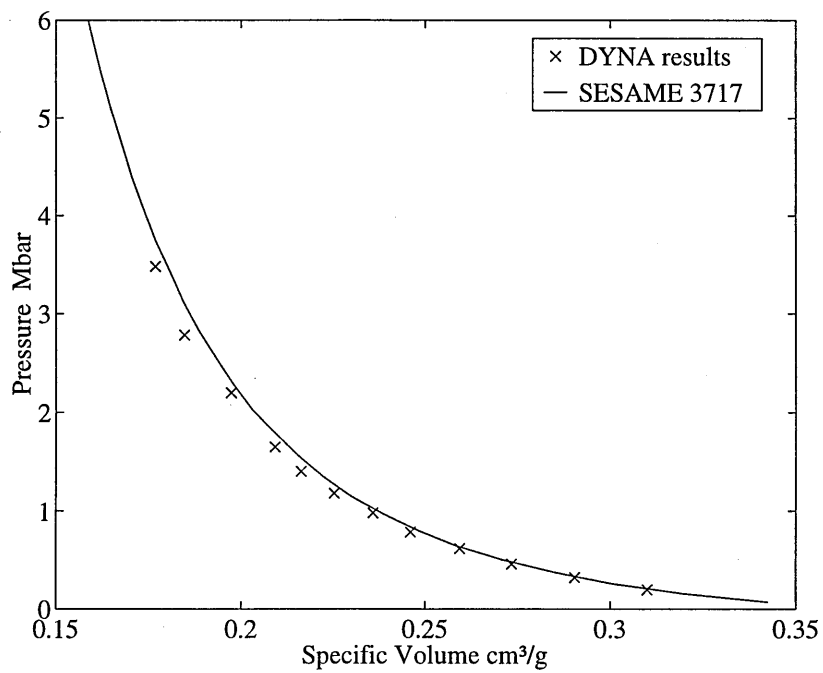


Figure 4.4: Comparison of Hugoniot curve for SESAME 3717 with DYNA results

u_p cm/ μ s	P Mbar	ρ g/cm ³	u_s calculated cm/ μ s	u_s measured cm/ μ s
0.10	0.196	3.226	0.708	0.704
0.15	0.319	3.443	0.768	0.752
0.20	0.457	3.657	0.825	0.811
0.25	0.615	3.855	0.888	0.870
0.30	0.783	4.064	0.942	0.952
0.35	0.979	4.239	1.010	1.010
0.40	1.179	4.438	1.064	1.071
0.45	1.400	4.622	1.123	1.124
0.50	1.647	4.780	1.189	1.177
0.60	2.200	5.067	1.324	1.304
0.70	2.780	5.413	1.434	1.429
0.80	3.475	5.655	1.568	1.539

Table 4.2: DYNA results for points on aluminium Hugoniot curve

particle velocity it was possible to get an estimate for the shock velocity from these time histories. The time interval between the arrival of the shock wave at the time history points was measured from the graphs, and knowing the distance between the points a value for the shock velocity was found. Comparing the calculated and measured values of the shock velocity provided an additional check on the calculation.

The results from these DYNA3D calculations are presented in table 4.2. The values for the density and shock velocity were calculated using equations 4.13 and 4.14. The calculated and measured values of the shock velocity show reasonable agreement. Figure 4.4 shows the DYNA3D data points plotted with the Hugoniot curve calculated directly from SESAME. The points show good agreement with the expected Hugoniot curve, showing that the SESAME equation of state is working correctly in the DYNA code.

4.4 Summary

The SESAME tabulated equation of state was selected for implementation in DYNA3D as it is valid over the whole range of conditions found in hypervelocity impact, and a wide range of material data is available. SESAME was implemented as equation of state 12. SESAME is computationally expensive as an interpolation from the data table must be done three times per time step for every element. In order to minimise the increase in run time, SESAME was linked to the existing Gruneisen equation in DYNA. The Gruneisen equation is used at low compression, where it is valid, if the compression of an element exceeds a set value for the material, the code switches to using SESAME for that particular element. The implementation of the SESAME equation of state was validated by calculation of the Hugoniot curve for aluminium from DYNA results. This was compared with the Hugoniot curve calculated directly from the equation of state data tables. Good agreement was seen between the two curves, showing that SESAME was implemented correctly.

Chapter 5

Element Erosion Criteria

In chapter 3 it was concluded that the element erosion capability in the DYNA3D code should be improved. The use of erosion for modelling hypervelocity impact is limited by the few material models for which it is available, the Steinberg-Guinan constitutive model is the only high-strain rate model that it is implemented for. The Steinberg-Guinan constitutive model allows elements to melt, a weakness in the implementation of the erosion criterion results in any melted element immediately being eroded. This too early erosion of elements interferes with shock wave formation at the contact surface.

This chapter covers the extension of element erosion to work with the Johnson-Cook strength model, and the development of two alternative erosion criteria, one based on total element strain and the other on element accuracy.

The chapter is split into the following sections:

- 5.1 Element erosion algorithms
- 5.2 Element strain criterion
- 5.3 Development of non-strain criterion

5.1 Element erosion algorithms

Element erosion was developed to allow Lagrangian codes to deal with impact problems that involved large mesh distortions and erosion of material from projectile or target. Prior to its development, penetration problems required the use of rezoning and remeshing, or the use of Eulerian or Free-Lagrangian codes.

The basic assumption behind element erosion is that the material represented by the eroded element has ceased to play a significant role in the physics of the simulated event, so ignoring it will not affect the final result. Element erosion works by totally failing elements that meet the erosion criterion, and redefining the contact surface to remove the elements. This effectively deletes the element and the material related to the element from the calculation.

Element erosion was first developed by Stecher and Johnson [64] in order to solve problems where erosion is the primary mode of penetration. Erosion is a process that begins when the impact velocity relative to the contact surface exceeds the compressive plastic wave velocity [24]. A shock wave is formed, and the material behind the shock wave is failed and flows away from the contact surface. The result of this process can be most clearly seen in long rod impacts where the front of the rod has disintegrated, but the rear receives no permanent damage.

As implemented in a hydrocode, an element erosion algorithm consists of two sections: the test to decide when to erode an element, and the method of redefining the contact surface when an element has been eroded. In general the test is easy to implement, whereas a suitable contact surface redefinition algorithm, especially in three dimensions, is complex. The problem is complicated by the need for the algorithm to be computationally efficient. The result was that the principal effort went into designing an efficient contact algorithm, and the erosion test received little mention in publications [40, 64, 65, 66].

Element erosion was originally developed for the EPIC code, where effective plastic strain was used as the erosion criterion. The choice of effective plastic strain as the criterion is not explained [64]. Its choice as a measure of deformation probably owes a lot to its convenience, as it is a variable calculated as part of an elastic-plastic material model. Published results using EPIC cover impact onto semi-infinite targets and thin plates. For semi-infinite targets Stecher and Johnson [64] observe good agreement between element erosion and 1-D approximations for both rate of penetration and penetration depth. For thin plate impact Johnson and Cook [67] observe reasonable agreement between element erosion and experiment for oblique impact of long rods. However when modelling shaped charge jet impact onto armour, Raftenberg [68] was unable to get accurate results when using effective plastic strain. Much improved agreement was observed when more sophisticated failure models, tensile void and shear band, were combined with element erosion. The only alternative erosion criterion mentioned is a 'geometric strain criterion' in the AUTODYN code [55], but no further explanation is given.

The erosion algorithm in DYNA3D is known as SAND (Slidesurfaces with Adaptive New Definitions). The name refers to the contact portion of the erosion algorithm. The erosion criterion portion is implemented in individual material model subroutines, where elements are tested and failed elements flagged. When an element is failed its full stress tensor is set to zero, representing complete failure. The contact algorithm checks for newly failed elements, and adjusts the contact surface accordingly. In the contact algorithm there is no means for a node connected entirely to eroded elements to interact with any other part of the calculation. So the mass represented by the node is effectively lost from the calculation.

5.1.1 Extension of erosion to Johnson-Cook constitutive model

The Johnson-Cook strength model is widely used in impact calculations. Allowing it to work with element erosion represented a useful improvement to the code.

As no change was required in the contact algorithm, the necessary changes prove relatively simple. The principal changes were made to the material model subroutine, `f3dm15`. A check is required to find if the effective plastic strain, which is calculated as part of the material model, exceeds the erosion limit. If so, then the erosion flag is set, this is all the information that the contact algorithm requires, and the stress tensor is set to zero. The other changes required were to the input subroutine, to load and pass the effective plastic strain at failure to the material subroutine.

5.2 Element strain criterion

5.2.1 Choice of criterion

Due to the problems encountered when using the effective plastic strain criterion with the Steinberg-Guinan strength model, the possibility of using an alternative strain measure was investigated. In order to avoid difficulties with melted elements it was necessary to use a measure independent of the material model. A clear starting point was the rate of deformation tensor, $\dot{\epsilon}$.

Integrating the rate of deformation tensor in time would give a measure of the element strain. Here the term strain has to be used with caution as the rate of deformation tensor is not the same as the time derivative of the strain tensor, here denoted by $\dot{\epsilon}$.

$$\dot{\epsilon} = \frac{1}{2} \left[\frac{\partial v^\alpha}{\partial X^\beta} + \frac{\partial v^\beta}{\partial X^\alpha} \right] \quad (5.1)$$

whereas

$$\dot{\epsilon} = \frac{1}{2} \left[\frac{\partial v^\alpha}{\partial x^\beta} + \frac{\partial v^\beta}{\partial x^\alpha} \right] \quad (5.2)$$

x are the current spatial co-ordinates, while X are the original material co-ordinates. So knowing the time integral of the rate of deformation tensor and the current position does not allow the original position to be found.

As with plastic strain an effective value of the integrated rate of deformation tensor was used, allowing a single value to be used as the erosion criterion. For simplicity the effective rate of deformation was integrated in time, as opposed to the complete tensor, to provide the effective deformation value used. The effective rate of deformation, $\dot{\bar{\epsilon}}$, chosen was:

$$\dot{\bar{\epsilon}} = \left[\frac{2}{3} \dot{\epsilon}^{\alpha\beta} \dot{\epsilon}^{\alpha\beta} \right]^{\frac{1}{2}} \quad (5.3)$$

this definition of effective rate of deformation was chosen as it is the equivalent of the effective rate of plastic strain:

$$\dot{\bar{\epsilon}}_p = \left[\frac{2}{3} \dot{\epsilon}_p^{\alpha\beta} \dot{\epsilon}_p^{\alpha\beta} \right]^{\frac{1}{2}} \quad (5.4)$$

though equation 5.4 is not the actual method used to calculate the increment of effective plastic strain in the material subroutines.

5.2.2 Implementation in DYNA3D

In order to use this criterion in DYNA3D the value of the effective deformation for each element must be stored so that it is accessible during the following time step. It is possible to do this by using what DYNA calls auxiliary variables. These are material model variables that are not universal to all material models, but must be stored for use the following time step, for example: the effective plastic strain.

The number of auxiliary variables that the material model requires is defined in routine `blkdat`, the number of variables is loaded into array `nconst` by a DATA

statement. The array contains the number of variables required by all strength models implemented in the DYNA code, and arranged in order of the DYNA material type numbers. For example the number of variables required by the Steinberg-Guinan model, `nconst(11)`, is 4. Changing this value to 5 results in one extra auxiliary variable being stored. This value must be defined early in the code to allow the correct space to be reserved in the DYNA database. The structure of DYNA3D has been covered in chapter 2.

The auxiliary variables are passed between subroutines in COMMON block `aux14`. The first six variables in this common block are the components of the stress tensor, the first auxiliary variable is passed as the seventh variable in the COMMON block.

The effective deformation criterion was implemented for the Steinberg-Guinan and Johnson-Cook models. For each model a new input variable was defined, telling DYNA which erosion criterion to use. Effective plastic strain was set as type 1, effective deformation as type 2, with type 1 as the default. In both material subroutines, code was added to calculate the effective rate of deformation when this erosion criterion is used.

5.3 Development of non-strain criterion

The aim of this work is to allow modelling of impact on spacecraft structures. In order to successfully achieve this, it is necessary to be capable of modelling the formation and propagation of a debris cloud. Element erosion offers the only technique available, in a Lagrangian finite element code, to model the penetration process that results in the formation of the debris cloud, without requiring the intervention of the analyst as the remeshing technique requires. If the modelling of the debris cloud is to be accurate then the process of element erosion must not remove significant amounts of material from the main portion of the debris cloud. This requires an erosion criterion that removes the minimum number of elements necessary for the calculation to survive.

Using the name erosion can be misleading, as the physical penetration process that occurs at high velocity is also known as erosion. It must be noted that element erosion does not model the physical process of erosion. Element erosion is used to side step the computational problems that occur when a Lagrangian finite element code attempts to model the physical process of erosion. So it is being used for computational not physical reasons.

The large material deformations that occur in the erosion process lead to highly distorted elements that have a serious effect on the calculation and force it to terminate. Highly distorted elements give rise to three main problems:

- Hexahedral elements can invert, this means that the volume calculated for the element is negative. This forces the calculation to terminate.
- Highly distorted elements become inaccurate, introducing errors into the solution.
- The time step becomes small as the time step is proportional to the smallest element dimension. This leads to calculations taking an impractical amount of CPU time to solve, and can lead to significant round-off error.

Any erosion criterion must erode elements before they invert. It should remove elements that have become significantly inaccurate, otherwise the final result of the calculation cannot be used with confidence. Finally, a low time step is a practical consideration, round off errors can be reduced by using increased precision in storing numbers. The CPU time necessary to solve the problem is dependent on the particular computer available to the analyst. If it is found that an erosion criterion has not solved this problem, then it would be possible to add a criterion that erodes an element once its time step falls below a critical value.

In order to erode the minimum number of elements, a criterion is required that only erodes elements that threaten the accurate continuation of the calculation. These are elements that are about to invert and inaccurate elements. An inverted element has undergone significant deformation, and by definition has become inaccurate before inverting. So the need is for a criterion that detects inaccurate elements.

As any erosion criterion is evaluated for every element, every time step, or potentially every few time steps, it must not be too computationally expensive. In an explicit code anything that adds to the time needed to process one element can lead to significant increases in total solution time. This places a limit on the complexity of any possible erosion criterion. In developing a criterion to detect inaccurate elements, the starting point is the solid element used in DYNA, and the sources of inaccuracy in that element.

5.3.1 The DYNA3D solid element

The solid element used in the DYNA3D code is the 8-node hexahedron element. It is an isoparametric element integrated using one point Gaussian quadrature. So the element is a tri-linear constant stress element.

The element is under integrated as full integration would require a $2 \times 2 \times 2$ Gaussian integration rule. One point numerical integration is used for computational speed. As the stress must be found at each integration point, going from an eight point to a one point rule results in a very substantial saving in computer time [38]. The under integrated element does pass the patch test, and so does converge. The principal disadvantage of moving to one point integration is the need to control the zero energy modes, or hourglass modes. This is done by the use of a viscosity that only resists these specific deformation modes.

5.3.2 Element accuracy

To choose a suitable criterion to erode inaccurate elements it is necessary to start by looking at what makes an element inaccurate. Since the finite element method was developed a very considerable amount of effort has gone into quantifying the sources of error in the method. Assuming that the only errors are caused by deformed elements, a deformed element being one that is not a perfect cube, there are two possible sources of error. These are: error in evaluation of the strain-displacement relationship, equivalent to the strain rate-velocity relationship in the explicit finite element method; and error in the numerical integration of the element.

The strain rate-velocity relationship allows the strain rate, strictly the rate of deformation tensor, at the single integration point to be calculated from the nodal velocities.

The rate of deformation tensor is calculated from the spatial derivative of the velocity vector which is:

$$\frac{\partial v^\alpha}{\partial x^\beta} = \sum_{k=1}^8 \left[\frac{\partial \xi^\gamma}{\partial x^\beta} \frac{\partial \phi_k}{\partial \xi^\gamma} v_k^\alpha \right] \quad (5.5)$$

where $\frac{\partial \xi^\gamma}{\partial x^\beta}$ is the inverse of the element Jacobian, $\mathbf{J}$, which is evaluated at the integration point, which lies at $\xi^\gamma = 0$.

The element integration calculates the element contribution to the nodal force vector from the element strain. This is written:

$$\mathbf{F} = - \int_{V_e} \mathbf{B}^t \boldsymbol{\sigma} dV \quad (5.6)$$

For one point Gaussian integration this is written

$$\int_{V_e} \mathbf{B}^t \boldsymbol{\sigma} dV = 8 \mathbf{B}(0, 0, 0) \boldsymbol{\sigma}(0, 0, 0) |\mathbf{J}(0, 0, 0)| \quad (5.7)$$

where the strain displacement matrix, $\mathbf{B}$, is:

$$\mathbf{B} = \frac{\partial \phi_k}{\partial x^\beta} = \frac{\partial \xi^\gamma}{\partial x^\beta} \frac{\partial \phi_k}{\partial \xi^\gamma} \quad (5.8)$$

In both cases the only term that is dependent of the geometry of the element is the Jacobian. For the one point integration the Jacobian is evaluated at the integration point. This means that for the numerical integration to be exact the Jacobian must be constant over the whole element. A condition which is only true if the element geometry is a parallelepiped. If the element is not a parallelepiped the element integration is not exact and an error is introduced into the results.

The DYNA solid element is not accurate if the element Jacobian is not constant over the whole of the element. If the element is very distorted and inaccurate then there will be a significant variation of the Jacobian across the element. This is a condition which it is possible to test for. The simplest method of detecting the variation of the Jacobian is to evaluate the Jacobian determinant at several points within the element and compare them. This test is used in some finite element pre-processors as a check of mesh quality.

5.3.3 Implementation of criterion

The principal task to implement this criterion was to add the calculation of the Jacobian determinant at several points in the element. A new subroutine was written which evaluated the Jacobian determinant at eight points in the element, which were taken to be the same as the eight Gauss points of a $2 \times 2 \times 2$ rule. The maximum and minimum values of the determinant were found, and the ratio between the two was calculated. This subroutine is called from the strength model routine if the erosion criterion is being used. The element is eroded if the ratio exceeds a value set by the analyst in the input file.

As a practical consideration it was noted that this erosion criterion would not erode skewed elements that remained approximately a parallelepiped, but these elements could affect the calculation by reducing the time step. An additional check was added which would erode a skewed element with a ratio of maximum and minimum Jacobian determinant of close to 1.0. The skew is checked for by calculating the corner angles of the element faces. To save on CPU time only one angle is checked for each face. The largest value of the cosine of the corner angles is found for each element. If this exceeds a set value then the element is eroded. The analyst supplies a minimum skew angle in the input file, and the cosine of this angle is used for the check.

Chapter 6

Modelling hypervelocity impact in modified DYNA3D

This chapter presents the results of hypervelocity impact analyses using the modifications to the erosion algorithm presented in chapter 5.

The chapter is split into the following sections:

- 6.1 Semi-infinite target tests
- 6.2 Thin plate target tests
- 6.3 Discussion

Unlike the tests on the new equation of state in chapter 4, it is not possible to test the effect of the erosion algorithm in isolation. Element erosion is a tool that allows the code to model penetration, in itself a complex process. In order to test the changes to the erosion algorithm, a hypervelocity impact was modelled and the only parameters changed were the erosion criterion used, and the magnitude of the erosion criterion. Analyses of impact onto two types of target were considered, semi-infinite and thin plate.

Then for each type of target a second set of analyses was done, varying only the impact velocity. If element erosion is to be used as a predictive tool it must be capable of producing accurate results at velocities different to the velocities in the validation cases. For these analyses the element erosion algorithm that produced the most accurate results in the first set of tests was used.

In this chapter all results are presented graphically, tables containing the results in numerical form are presented in appendix B, section B.3.1. All the analyses were carried out on a Cray J916 vector computer. The tabulated results include the number of time steps that the calculation required to reach the termination time, and the CPU time in seconds used for the analysis. Both values were taken from the text output file d3hsp, so the CPU time is DYNA3D's own value.

6.1 Semi-infinite target tests

A semi-infinite target is one where the boundaries of the target, away from the point of impact, have no effect on the result of the impact. A semi-infinite impact was chosen as a test case for the modified DYNA3D as it is the type of problem that element erosion was originally developed to solve.

Hydrocode modelling of semi-infinite impacts is used as a tool in characterising impacts on surfaces returned from orbit [55]. Study of an impact crater on a surface can

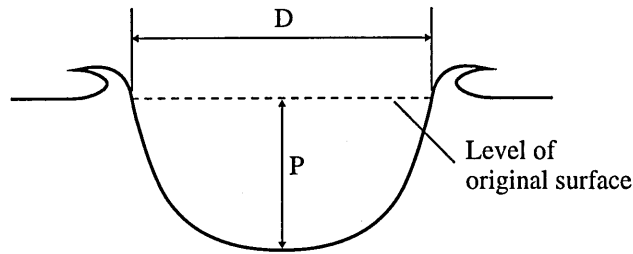


Figure 6.1: Crater size definitions.

allow estimates to be made of projectile size, density, velocity and the impact angle. This data can then be used to improve space debris environment models. A hydrocode can be used to characterise the effects that changing one of the impact variables has on the final crater morphology, especially for impact velocities higher than achievable in a laboratory.

The two basic measurements that define a crater’s shape are the depth and diameter. Only normal impact was considered, so the effects of oblique impact can be ignored. The definitions used for the results presented are:

- Crater depth, P , is the distance between the bottom surface of the crater and the plane of the original surface.
- Crater diameter, D , is the diameter of the crater at the level of the original surface plane.

A semi-infinite impact case was chosen for which published experimental and computational data is available [55]. The case is the normal impact of a 4.5mm diameter 4.5mm long cylinder at 5 km/s. Both the target and projectile material is aluminium 2024. Table 6.1 shows the published experimental and computational results available for this case. The 3D Lagrangian result used an element erosion algorithm to model the penetration. The experimental value for the crater diameter was not given. This type of impact, with a projectile length:diameter ratio of 1:1, would be expected to produce an approximately hemi-spherical crater.

Case	Crater Depth (mm)	Crater Diameter (mm)
Experiment	8.4	†
2D Eulerian computation	8.2	16.0
3D Lagrangian computation	8.3	15.0

†: value not given.

Table 6.1: Published results for semi-infinite impact test case.

If the experiment produced a perfectly hemi-spherical crater then the crater diameter would be 16.8mm. This suggests that the published results for the 2D Eulerian computation are the most accurate.

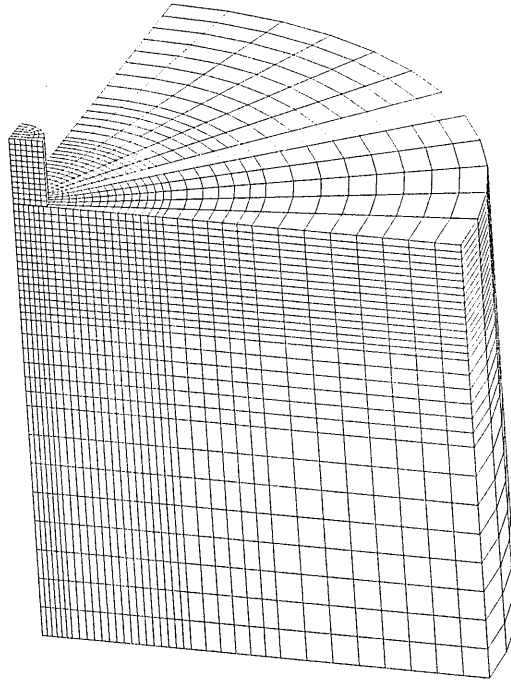


Figure 6.2: Mesh for semi-infinite impacts.

The finite element mesh used for these analyses is shown in figure 6.2. It consists of 14090 nodes and 12204 solid elements, of which 11880 elements make up the target. As it is a normal impact two planes of symmetry were used. The target is 30mm deep and 30 mm in radius. The depth and radius of the target were chosen to be equal, as the expected crater has approximately equal depth and radius. On the bottom and radial boundaries of the target a non-reflecting boundary condition was applied. This mesh is the third refinement of the mesh for the problem. In the first mesh the target was 12 mm deep, 12 mm radius and used a non-reflecting boundary condition. The boundary of the target was found to be too small, as the boundaries moved away from the point of impact, distorting the final shape of the crater. To prevent this behaviour a second mesh was developed by placing the 12mm by 12 mm block in a larger block of coarse elements, and a tied slide surface was used to link the two meshes, in the same way as for the thin plate mesh in chapter 3. This measure prevented the excessive motion of the fine mesh, however the impact shock waves were not perfectly transmitted across the tied slide surface which resulted in a partial reflection of the shock wave. Any wave reflection that results in the reflected wave reaching the area of impact before the impact process has completely finished violates the definition of a semi-infinite target. In order to correct the reflection the block of coarse mesh was removed, and the whole target was modelled as one block, figure 6.2.

All analyses used a termination time of $24\mu s$. By this point the final crater depth and diameter have been reached. This was checked by producing time history plots, which are shown in figures B.29 and B.30, of the displacement of the nodes where the depth and diameter were measured.

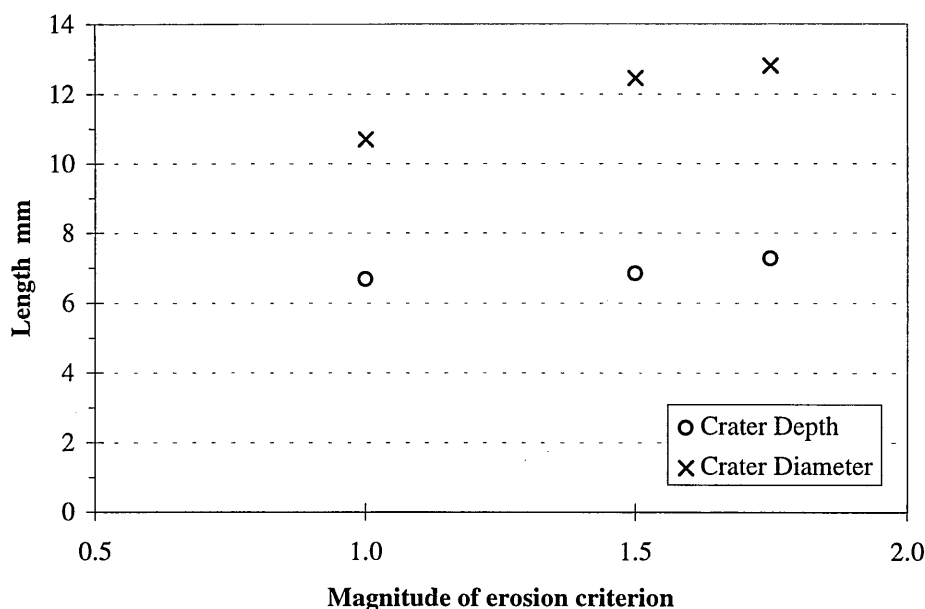


Figure 6.3: Results for semi-infinite impact with effective plastic strain criterion

6.1.1 Effective plastic strain criterion

Three analyses were made using the effective plastic strain erosion criterion with the Johnson-Cook material model. The highest value used for the erosion strain was 1.75. With values higher than this the contact became unreliable, and the solution was obscured by elements that penetrated the contact surface. This is the same problem that was observed in chapter 3 when using the effective plastic strain criterion to model thin plate impact.

Figure 6.3 shows the results for crater depth and diameter. Even at the highest value of erosion strain the final depth and diameter are too low, and the ratio between the two values is 1.76:1. It can be seen that the crater depth is not sensitive to change of the erosion strain. Intermediate and final crater shapes for the analysis with an erosion strain of 1.5 are shown in figures B.13–B.14.

6.1.2 Deformation criterion

For the deformation criterion, analyses with erosion values of between 1.0 and 3.0 were carried out. For each value of the erosion criterion two analyses were carried out, one using the Johnson-Cook material model for the projectile and target, the second using the Steinberg-Guinan model. This was to allow comparison of the two material models, which could not be done when using the effective plastic strain erosion criterion. An erosion value of 3.0 was found to be the highest practical value to use, as above 3.0 the mesh problems caused by excessive material deformation started to become significant and affect the reliability of the calculation.

Figure 6.4 shows the results for crater depth and diameter. Except for the lowest value of the criterion the depth and diameter from the Johnson-Cook results is higher. The best result for the depth was for an erosion criterion of 2.5 with the Johnson-Cook

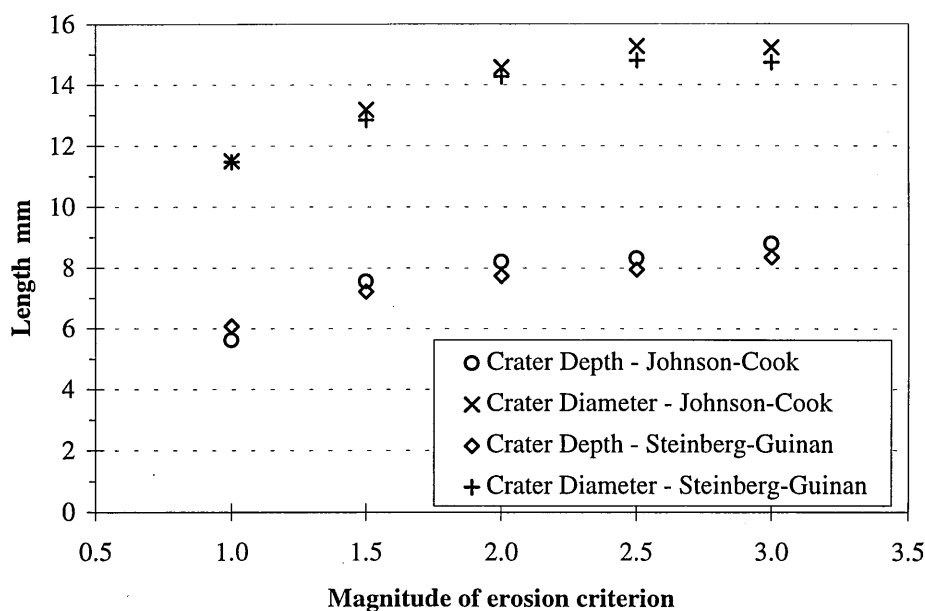


Figure 6.4: Comparison between Johnson-Cook and Steinberg-Guinan strength models, deformation criterion

strength model. This gave a crater depth of 8.31mm and a diameter of 15.27 mm, a ratio of 1.84:1, suggesting that the crater diameter is too small. Intermediate and final crater shapes for the analyses with an erosion value of 1.5 and 2.5 with the Johnson-Cook model, and erosion value 2.5 with the Steinberg-Guinan model, are shown in figures B.15–B.20.

The effect of using a different value for the erosion criterion between the projectile and target was investigated. Four analyses with the following values were carried out:

Projectile erosion	Target erosion
1.5	2.0
2.5	2.0
2.0	1.5
2.0	2.5

The results are shown in figure 6.5. Conclusions from these results are hard to draw. It was expected that either the results with a higher projectile criterion or those with a higher target criterion would give a consistently higher crater size. However the diameter for a target erosion of 1.5 is larger than the diameter for a plate erosion of 1.5, and the same is true for the erosion value of 2.5. The two cases where the target erosion was changed gave craters close to hemi-spherical, with the case for a target erosion of 2.5 giving a ration of 1.99:1.

6.1.3 Accuracy criterion

For the accuracy criterion analyses with erosion values of between 1.5 and 3.5 were carried out. As with the deformation criterion, it was found that using larger values for the erosion criterion resulted in problems due to excessive element deformation.

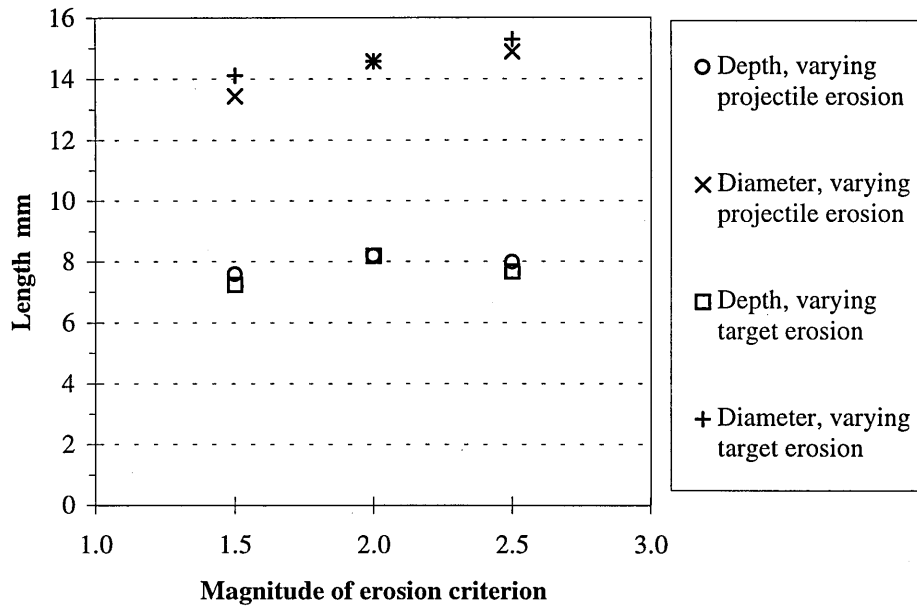


Figure 6.5: Results for deformation erosion criterion, non-equal erosion criteria.

These analyses all used a skew angle for erosion of 0.0 radians, effectively disabling this option. To test the effect of the skew angle analyses were carried out for skew angles of 0.1 to 0.5 radians with the element erosion ratio fixed at 2.5.

Figure 6.6 shows the results from varying the erosion ratio. The best estimate for crater depth came from the run with an erosion ratio of 3.5, which was 8.13mm. This value was measured at $20\mu s$, as DYNA crashed before the final plot dump at $24\mu s$. As the crater has finished growing by this point, this value should not have changed significantly by $24\mu s$. The crash occurred in the erosion algorithm subroutine, which uses the $\cos^{-1}$ function to calculate the element skew angle. If the element skew angle is approximately zero, round-off errors can cause the calculated value of the cosine of the skew angle to be greater than 1.0, causing an error in the $\cos^{-1}$ function. In all cases the diameter was under estimated, the crater from the run with an erosion ratio of 3.5 had a diameter:depth ratio of 1.62:1 which is representative of the results for the accuracy criterion. Intermediate and final crater shapes for the analyses with an erosion ratio of 2.0 and 3.0 are shown in figures B.21–B.24.

The dip in the results between erosion ratios of 2.5 and 3.0 occurred as an extra element survived at the base of the crater. The node at the base of the crater for the ratio 2.5 run was number 8, the node at the base of the crater for ratios 3.0 and 3.5 was number 7. The increase in erosion ratio allowed an extra element to survive on the centre line. This shows that a small change in the value of the erosion criterion can have a significant effect if it allows one extra element to survive or to be deleted at the point of measurement.

Figure 6.7 shows the results for the variation in skew angle. No trend can be seen in the crater depth results. The increase in skew angle has resulted in a small decrease in the final crater diameter. The most significant difference in the results was in the computer time required to solve the problem. To $24\mu s$ the run with an angle of 0.0

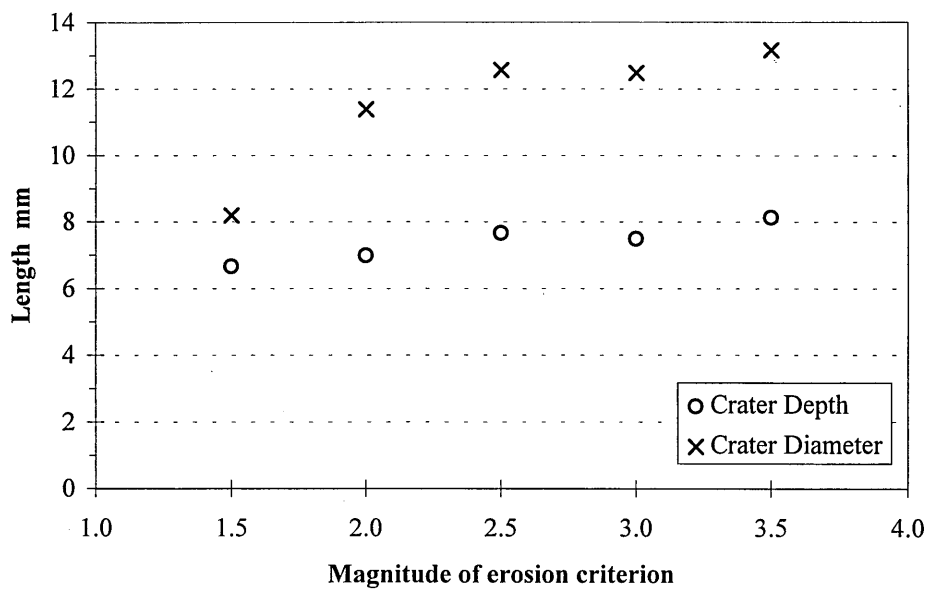


Figure 6.6: Results for semi-infinite impact, element accuracy criterion, Johnson-Cook strength model

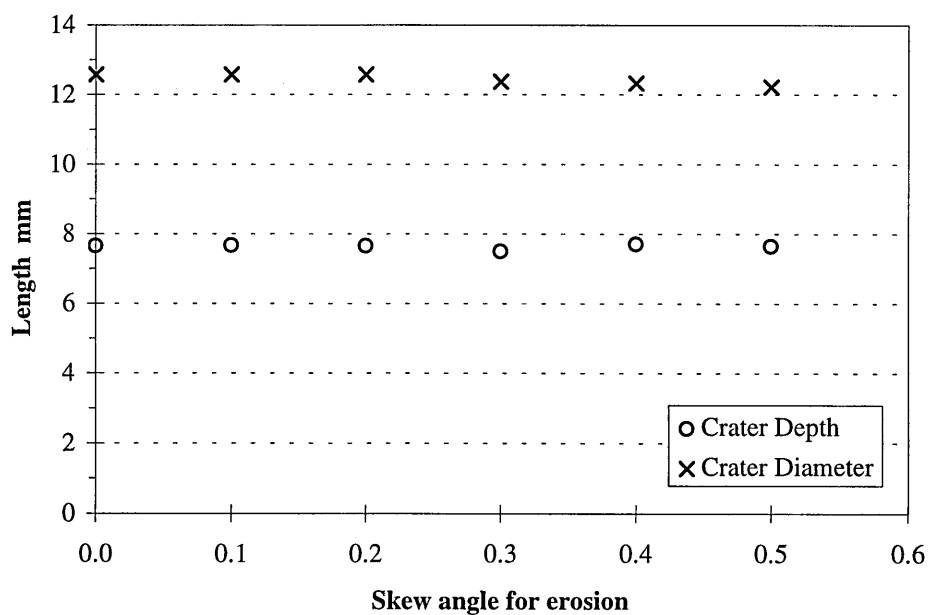


Figure 6.7: Results for semi-infinite impact, element accuracy criterion, showing effect of varying skew angle. Johnson-Cook strength model

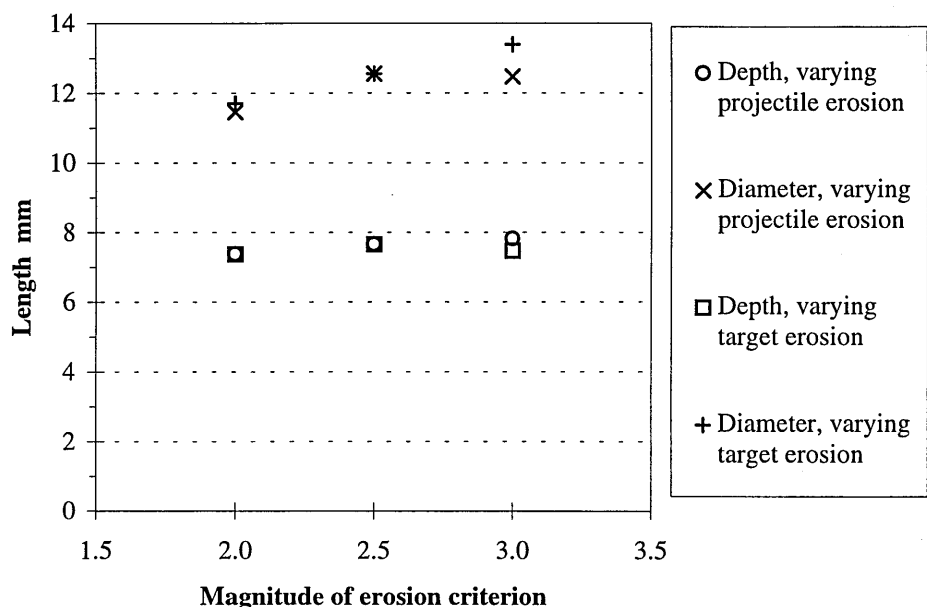


Figure 6.8: Results for element accuracy criterion, non-equal erosion criteria.

required 47953 s to solve, while the run for angle 0.5 radians required 34863 s. A difference of 13090 s, or about 3 hours 38 minutes, a significant saving in computer time. Intermediate and final crater shapes for the analysis with a skew angle of 0.0 and 0.5 are shown in figures B.25–B.28.

The effect of using a different value for the erosion criterion between the projectile and target was investigated. Either the projectile or target value was fixed at 2.5, and the other varied to 2.0 or 3.0. Figure 6.8 shows the results from these runs. As for the deformation criterion, varying the target erosion results in a larger diameter when compared with the equivalent run with a changed projectile erosion.

6.1.4 Comparison with experimental data

To test the ability of element erosion to be used as a predictive tool, analyses were carried out for a range of velocities, and compared with experimental data. Results for the crater size at different velocities was not available for the projectile used in the previous tests, so a different case had to be used. The experimental data used is from Denardo, Summers and Nysmith [1]. The problem modelled is the impact of a 3.175mm diameter aluminium (2017-T4) sphere onto an aluminium (2024-T4) target. The experimental data is for impact velocities between 3.28 and 8.52 km/s.

Analyses were made at impact velocities of between 4.0 and 8.0 km/s, using the deformation erosion criterion with a value for erosion of 2.5. The Johnson-Cook strength model was used. These values were chosen as they produced the best value for the crater depth when testing the different erosion algorithms. The mesh used, figure B.31, was based on the mesh used for the different erosion algorithm tests.

Denardo, Summers and Nysmith also present data for the same projectile impacting on to a soft aluminium (1100-0) target. Two obvious differences were seen between the

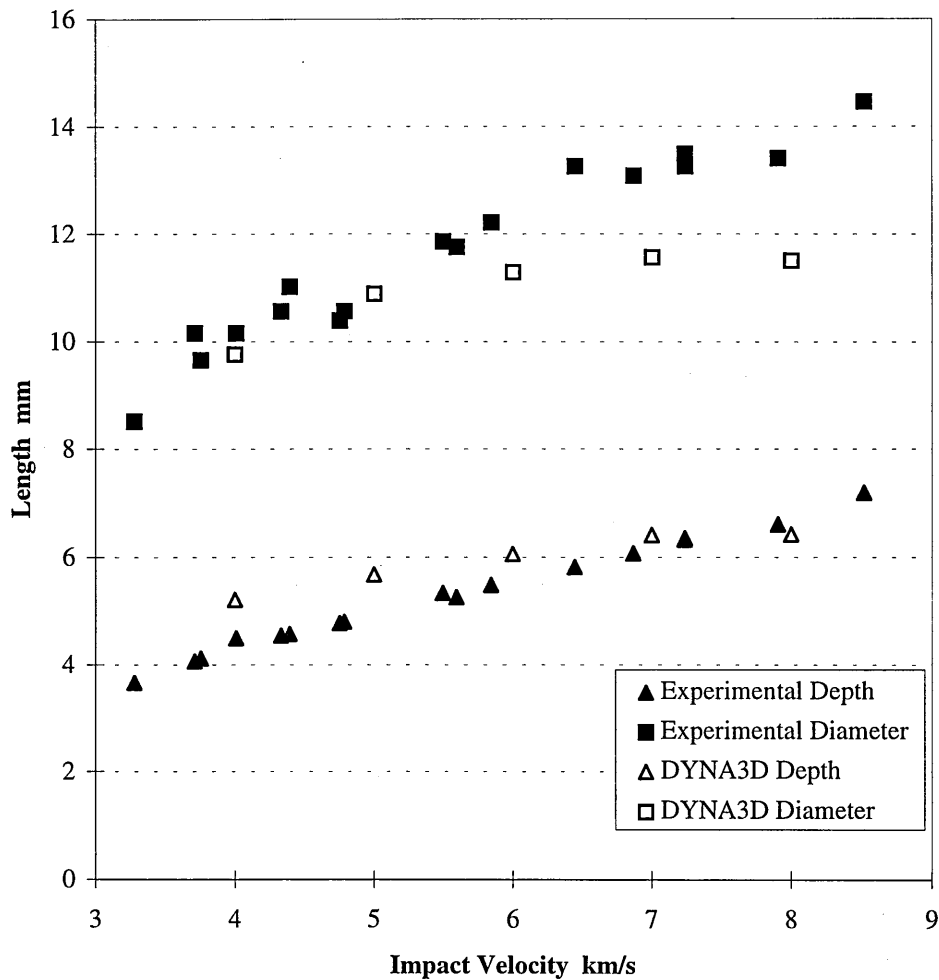


Figure 6.9: Comparison of semi-infinite impact analyses with experimental results.

results for the two target materials. The first difference was that the craters in the soft target were significantly larger, diameters increased by up to 25%, and depths by up to 40%. The second difference was that the crater lips in the hard targets were detached at impact velocities above of approximately 4.2 km/s, this occurs as 2024-T4 aluminium is more brittle than 1100-0. In order to measure the crater diameter Denardo, Summers and Nysmith used a plaster filler to estimate the original crater shape.

The experimental values for the hard aluminium targets used as the input constants for the Johnson-Cook model are for aluminium 2024-T351. No damage model is used, so detachment of the crater lips cannot be modelled. The experimental diameter values used are the estimates to the final crater shape.

Figure 6.9 shows the comparison between the experimental data and the DYNA3D results. It can be clearly seen that the DYNA3D results do not accurately predict the change in either depth or diameter. The largest error is seen in the diameter at the higher velocities. The predicted crater shapes are also in error, the experimental results have diameter:depth ratios ranging from 2.33:1 to 2.50:1 at velocities under 4 km/s, down to a ratio of 2.01:1 at 8.52 km/s. The DYNA3D results are 1.87:1 at 4km/s and 1.79:1 at

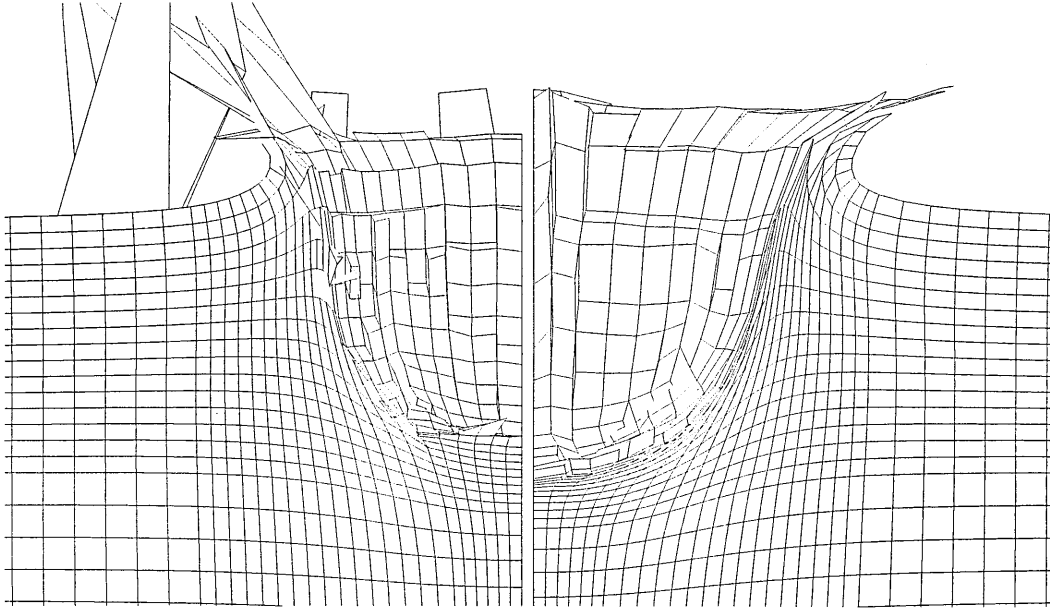


Figure 6.10: Comparison between final crater for the effective plastic strain criterion with value 1.5 (left) and the deformation criterion with value 2.5 (right). Both analyses used the Johnson-Cook strength model.

8 km/s.

6.1.5 Discussion of semi-infinite impact results.

The different erosion criteria have produced significantly different results for the final crater size and shape. The best estimate for the crater depth was produced by the deformation erosion criterion, with the other two criteria producing results that are too small.

Figure 6.10 shows a comparison between final crater shapes for the effective plastic strain criterion and the deformation criterion. The results for the effective plastic strain criterion give the smallest crater size. This is at least partly due to the low maximum value for the erosion strain that could be used. Even when used with the Johnson-Cook strength model it proved the least satisfactory as the upper limit on the erosion strain was imposed by the reliability of the contact, not by problems from highly deformed elements.

Figure 6.11 compares the final crater shapes for the Steinberg-Guinan and the Johnson-Cook strength models, for the deformation erosion criterion. The Johnson-Cook model gives a larger crater, in both depth and diameter, while the Steinberg-Guinan model gives a crater that is slightly closer to a hemisphere. A possible cause for the difference in the crater sizes is that the input parameters for the two models were calculated from tests of different alloys of aluminium. The Steinberg-Guinan parameters are for Al 6061-T6 [43], while the Johnson-Cook parameters are for Al 2024-T351 [44]. Published formulas relating crater size to material properties give an inverse relationship between crater size and hardness of the target [69] or crater size and the yield strength of the target [70]. Al 2024-T351 is both harder and has a higher yield strength

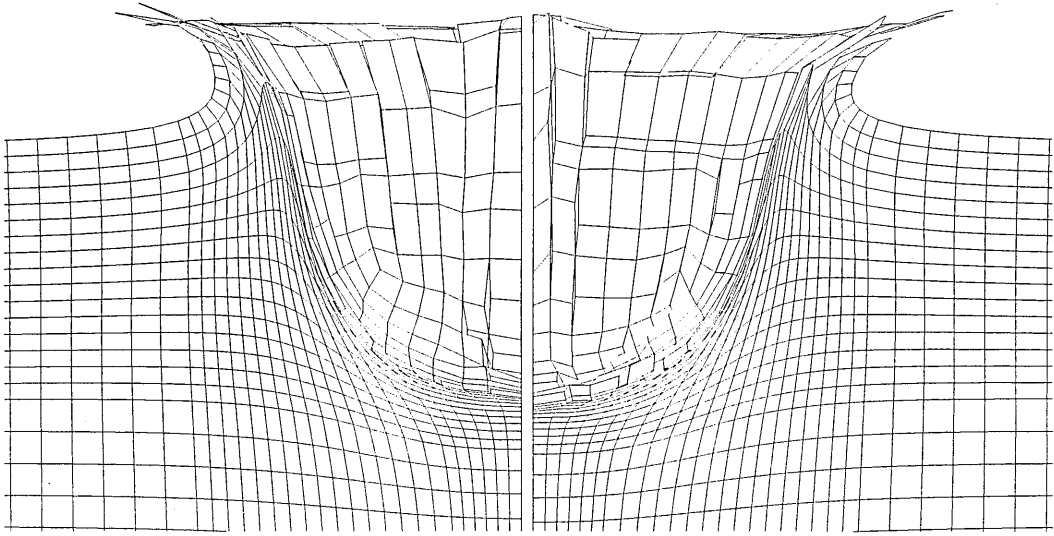


Figure 6.11: Comparison between final craters for deformation criterion. Steinberg-Guinan strength model (left) and Johnson-Cook strength model (right). Both analyses used the same value for erosion of 2.5.

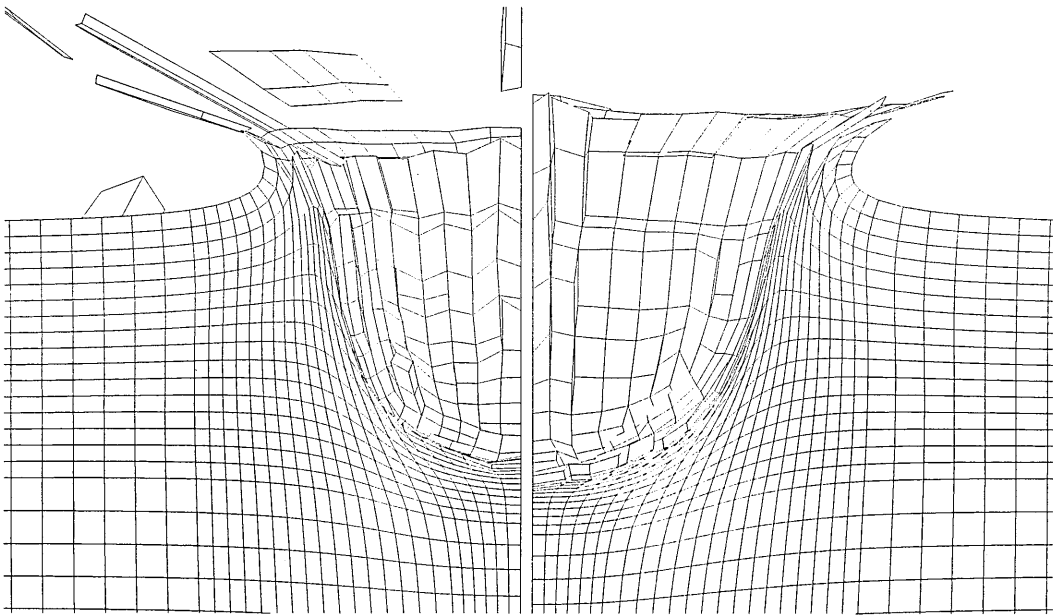


Figure 6.12: Comparison between final crater for the element accuracy criterion with value 2.5 and skew angle 0 radians (left) and the deformation criterion with value 2.5 (right). Both analyses used the Johnson-Cook strength model.

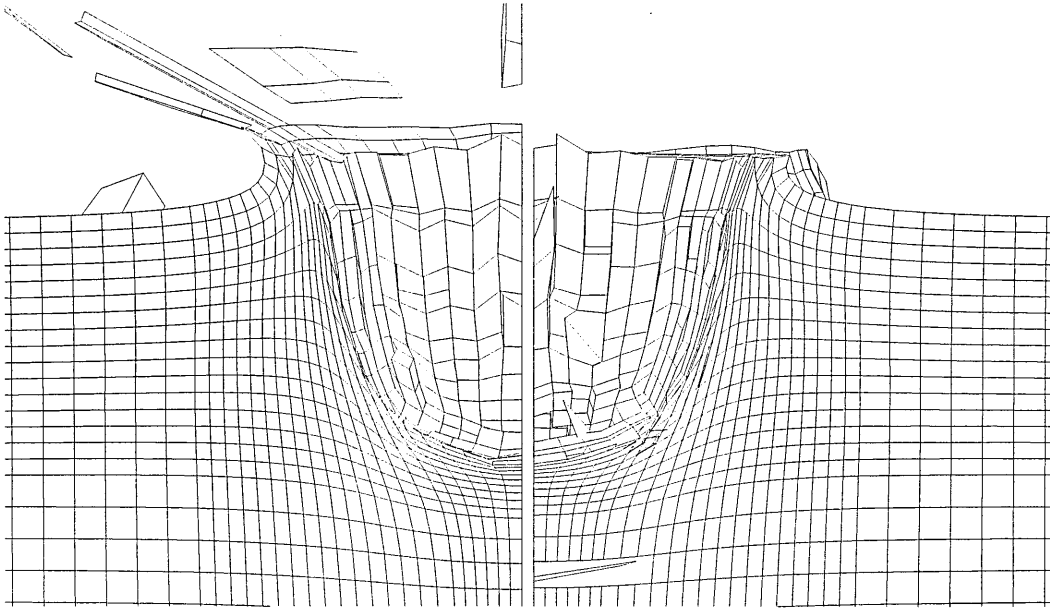


Figure 6.13: Comparison between final crater for element accuracy criterion with skew angle 0 radians (left) and with skew angle 0.5 radians (right). Both analyses used the Johnson-Cook strength model.). Both analyses used the same value for erosion of 2.5.

than Al 6061-T6 [71], which would indicate that for identical projectiles the crater in a Al 6061 target should be larger. The experimental results are for the 2024 alloy, this suggests that the Johnson-Cook model is giving the more accurate results as it can produce an accurate value for the crater depth. The Steinberg-Guinan model gives a small underestimate of the depth for the highest erosion value, but if the experimental results were for a Al 6061 target the crater would be expected to be larger.

Figure 6.12 compares the final crater shapes for the accuracy criterion and the deformation criterion. The accuracy criterion produces a smaller and less hemi-spherical crater, always underestimating the final crater depth. Figure 6.13 compares the final crater shapes when using the accuracy criterion with a skew angle of 0.0 radians and 0.5 radians. The final crater depth and diameter are very similar, the clearest difference is in the area of the crater lip where additional elements have been eroded for the higher skew angle.

The crater depth and diameter results show that the deformation criterion is the most suitable of the three criteria tested for modelling cratering. This conclusion is also supported by the CPU time data. The deformation criterion proved the most computationally efficient, with a CPU time of 18564s for an erosion value of 2.5. The accuracy criterion required 47953s for a skew angle of 0.0, reducing to 34863s for an angle of 0.5. The accuracy criterion requires a much greater time to solve and does not provide any benefit in terms of increased accuracy. The reduction in solution time seen when using the additional skew angle criterion occurs as elements near the lip of the crater have the smallest critical length, and so control the size of the time step.

The results for varying velocity show that element erosion does not accurately predict the effects of changing the impact velocity for constant erosion value. As an

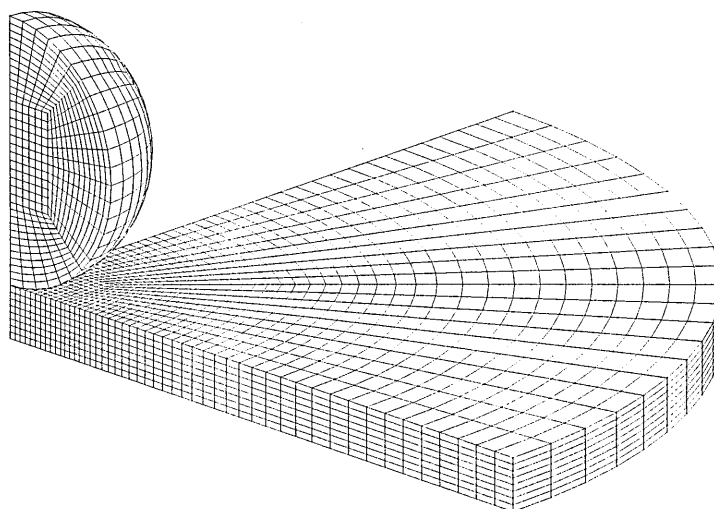


Figure 6.14: Finite element mesh used for thin plate impact analyses.

erosion value will only give accurate results over a small velocity range, a suitable erosion value would have to be determined at regular velocity intervals. Limiting its use at impact velocities higher than those achievable in the laboratory.

The analyses where the projectile and target have different erosion values suggest that improved results for the crater shape could be gained by adjusting the two values, this would require several more analyses to be performed. This complicates the validation process by increasing the number of analyses required to determine suitable values for a given impact velocity.

6.2 Thin plate target tests

For the thin plate impact tests the same impact case used in chapter 3 was taken. This is the normal impact of a 4mm aluminium sphere onto a 0.8mm thick aluminium plate at 8 km/s. The experimental value of the final hole diameter in the plate was 7.8mm.

The mesh used for the test is shown in figure 6.14. This mesh has been changed from the mesh used in chapter 3. The changes have been made to the plate mesh, leaving the projectile essentially unchanged. The outer section of coarse elements has been removed, and the outer boundary of the plate mesh has been extended further away from the impact point. The mesh consists of 11989 nodes and 10064 solid elements, of which 7040 elements make up the plate.

All analyses have a termination time of $4\mu\text{s}$. To check this termination time the displacement time history of a node at the edge of the bumper plate hole was checked, figure B.32. Before this point the hole diameter has reached its maximum value, at $2.5\text{--}3.0\mu\text{s}$, and has rebounded. The hole diameter is still oscillating. This oscillation was not considered important enough to increase the termination time, and thus increasing the computer time required to solve the problem.

The principal measurement made was the final hole diameter in the plate. This was calculated from the displacement of the edge of the hole. Where possible other

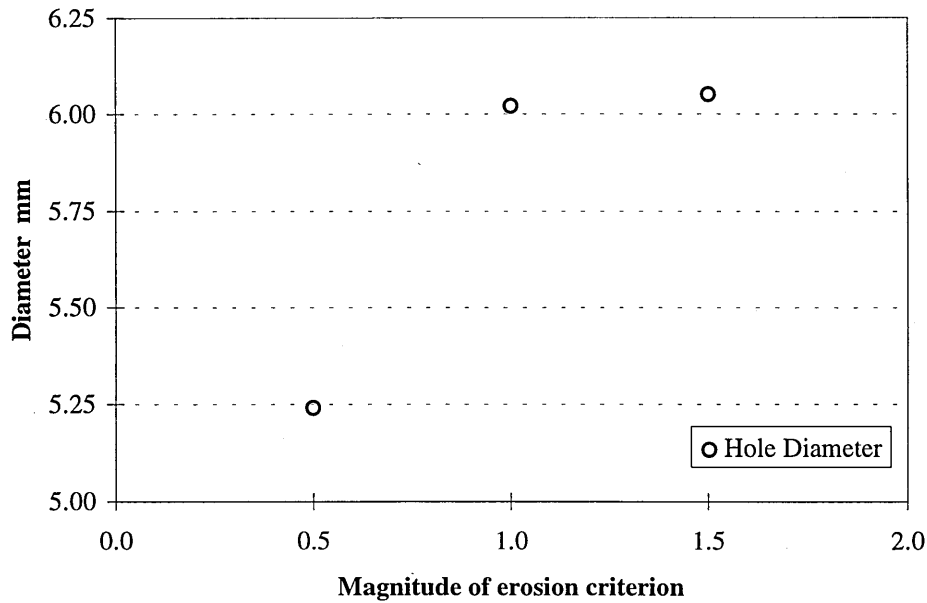


Figure 6.15: Results for bumper plate impact, effective plastic strain erosion criterion with Johnson-Cook strength model.

measurements were made, these were two debris cloud velocities and the main spray angle of the debris cloud. The published experimental results for this case [41] give the spray angle of the debris cloud. Estimates for the debris cloud velocities were calculated from published results by Piekutowski [47]. Piekutowski has measured the material velocity at several points in the debris cloud for varying impact velocity and projectile diameter to target thickness. From these results it is possible to calculate approximate values of the debris cloud velocity at certain points. The two velocities measured were the velocity at the leading edge of the main portion of the debris cloud and the trailing edge of the debris cloud, Piekutowski denotes these points as 2 and 4 respectively. Where they could be measured from the DYNA results they are denoted as v_2 and v_4 . The estimates to v_2 and v_4 for an impact velocity of 8 km/s and a diameter to thickness ratio of 0.2 are: $v_2 \approx 7.2$ km/s and $v_4 \approx 4.4$ km/s. The reason for measuring these two values is that they provide a measure of the accuracy of the wave propagation process after impact. If the wave propagation has been modelled accurately, and the equation of state is accurate, then the velocities measured from the analyses should be similar to these estimates.

6.2.1 Effective plastic strain criterion

Three analyses were made using the plastic strain erosion criterion with the Johnson-Cook material model. The highest value used for the erosion strain was 1.5. As for the semi-infinite impact case the upper limit was imposed by the reliability of the contact. At higher values of erosion strain the solution became obscured by nodes that have penetrated the contact surface.

Figure 6.15 shows the results for the hole diameter in the plate. The results for

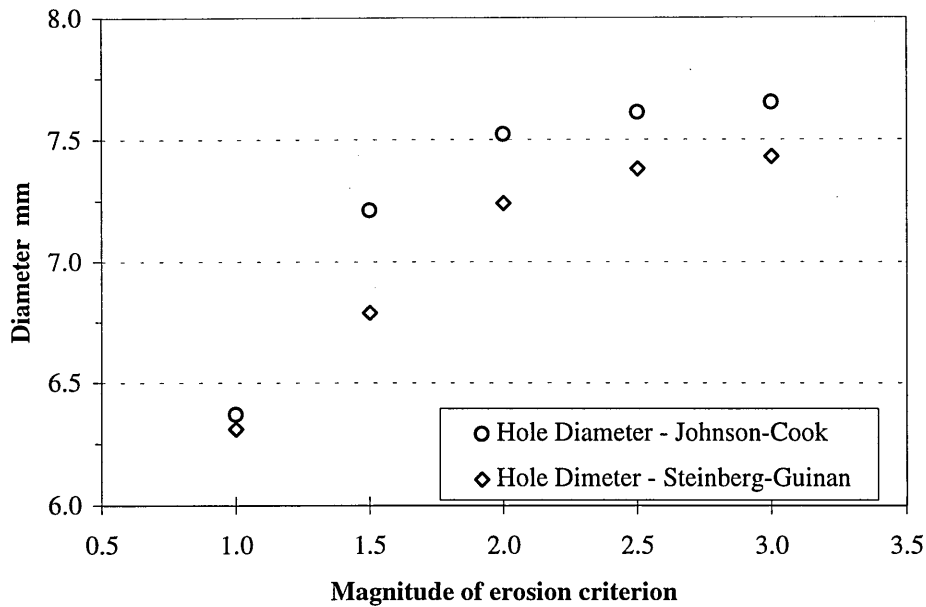


Figure 6.16: Comparison between Johnson-Cook and Steinberg-Guinan strength models, deformation criterion

the highest practical value of the erosion criterion are still significantly under estimate the final hole diameter. The debris cloud velocities could not be determined for an erosion strain of 1.5 as the results were obscured by nodes which had penetrated the contact surface. The debris cloud velocities for an erosion strain of 1.0 are $v_2 = 7.66$ km/s and $v_4 = 6.71$ km/s. These velocities are too high, which suggests that the shock wave formation and propagation is still not being accurately modelled, though pressure contour plots, figures B.33–B.34, show that the modelling is more accurate than when using the Steinberg-Guinan model. Intermediate and final hole shapes in the plate for an erosion strain of 1.0 are shown in figures B.35–B.36.

6.2.2 Deformation criterion

For the deformation criterion, analyses with erosion values of between 1.0 and 3.0 were carried out. For each value of the erosion criterion two analyses were carried out, one using the Johnson-Cook material model the second using the Steinberg-Guinan model. An erosion value of 3.0 was found to be the highest practical value of the erosion criterion.

Figure 6.16 shows the results for hole diameter for both material models. The Johnson-Cook material model consistently produces a larger final hole diameter, though the highest value of 7.61mm is still a small underestimate. Using this erosion criterion results in few elements that would form the debris cloud surviving the penetration process. Only at the higher values of the criterion did enough elements survive to allow measurement of the debris cloud velocities, and then only v_2 . Where the measurement could be made, v_2 lies in the range 7.38-7.51 km/s for the Johnson-Cook material model. This is higher than the estimated value of 7.2 km/s, but as few elements survive

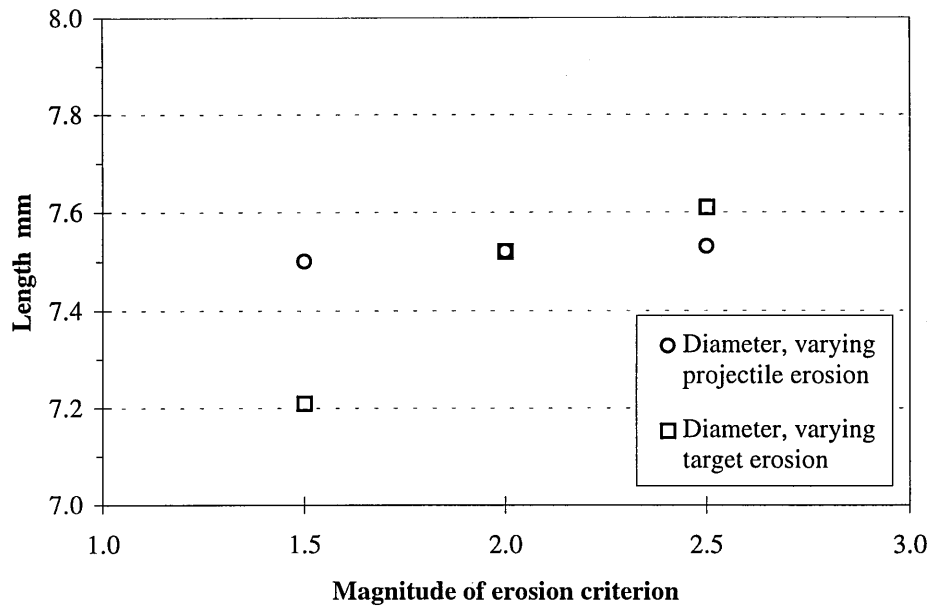


Figure 6.17: Results for deformation criterion, non-equal erosion criteria

to form the debris cloud it is hard to identify the leading edge of the main portion of the debris cloud. From these measurements it is not possible to estimate the accuracy of the wave propagation process, but pressure contour plots, figures B.37–B.38, suggest reasonable accuracy. Intermediate and final hole shapes for the Johnson-Cook and Steinberg-Guinan models are shown in figures B.39–B.44.

The effect of using a different value for the erosion criterion between the projectile and target was investigated. The results are shown in figure 6.17. These results show that the final hole diameter is sensitive to the erosion value in the target plate, but not sensitive to the value for the projectile.

6.2.3 Accuracy criterion

For the accuracy criterion analyses with erosion values of between 1.5 and 2.5 were carried out. Analyses for values of the criterion greater than 2.5 were not reliable due to problems with deformed elements. To test the effect of the skew angle, analyses were carried out for skew angles of 0.0 to 0.5 radians with the element erosion ratio fixed at 2.5.

Figure 6.18 shows the results for varying the erosion ratio. All the results underestimate the final hole diameter. Figure 6.19 shows the results for the variation in skew angle, which shows that the lower values of skew angle generally give a larger final hole diameter. The effect of changing the skew angle on the CPU time required to solve the problem is not clear as the run for a skew angle of 0.3 was significantly more expensive than the runs for an angle of 0.0 and 0.5. Intermediate and final hole shapes for skew angle 0.5 and 0.0 are shown in figures B.47–B.50.

Using this criterion a significant proportion of the debris cloud material survives the penetration process, allowing measurements of the debris cloud expansion angle and

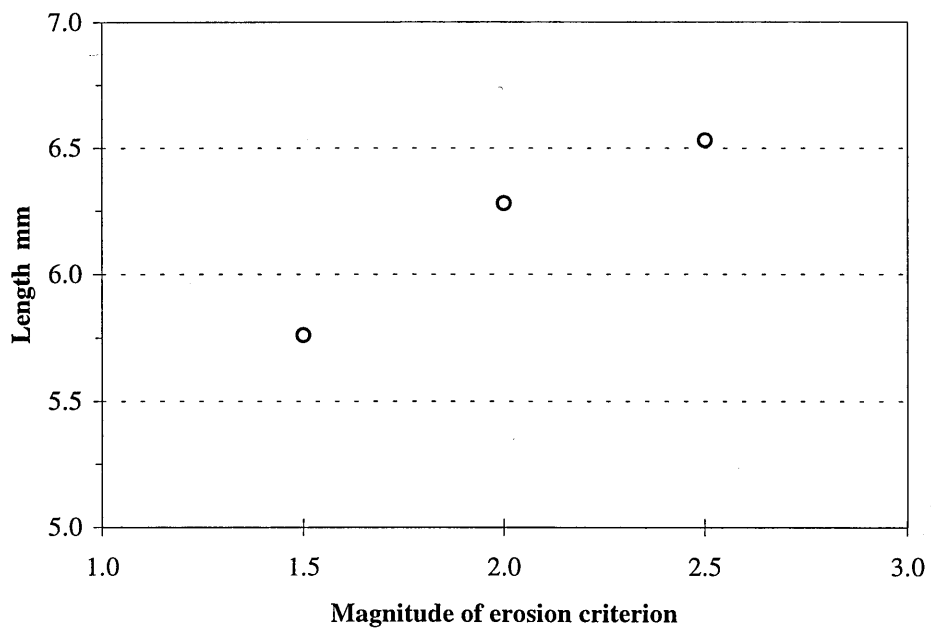


Figure 6.18: Results for bumper plate impact, element accuracy criterion.

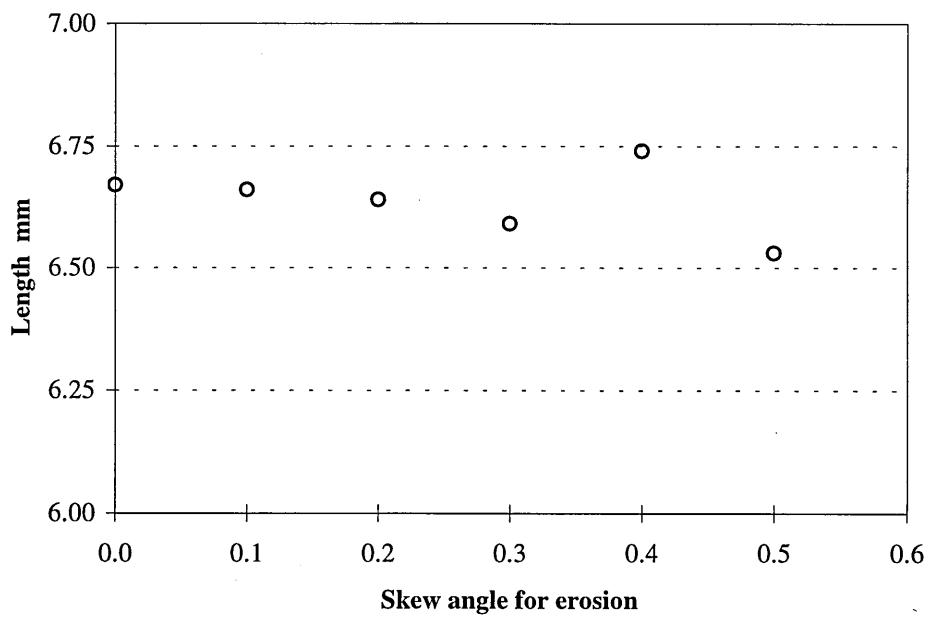


Figure 6.19: Results from variation in skew angle for erosion, element accuracy criterion

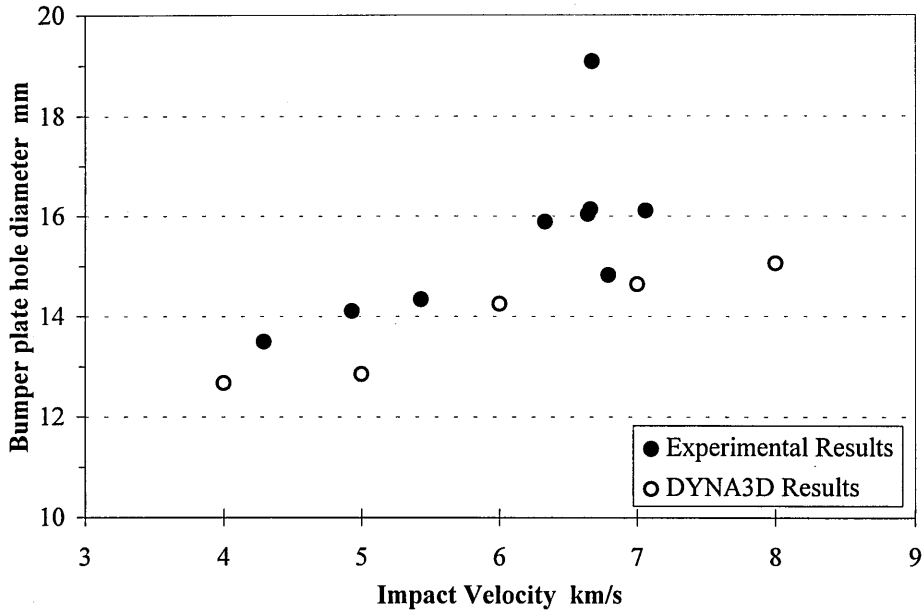


Figure 6.20: Comparison of bumper plate impact analyses with experimental results. T/d ratio 4.97:1

the debris cloud velocities to be made. The debris cloud velocities show reasonable agreement with the expected values, for the analyses with erosion value 2.5, the measured values were in the range $v_2 = 7.05\text{--}7.10$ km/s and $v_4 = 4.71\text{--}4.82$ km/s. These results show that the accuracy criterion allows accurate modelling of the shock wave propagation process. In the paper by Frey [41] the main debris cloud expansion angle is the angle between the centre line and the expansion angle of the main portion of the debris cloud. Enough material survived the penetration process to allow this to be measured. For the analyses with erosion value 2.5 the expansion angle was in the range $29.3\text{--}35.8^\circ$, which agrees with the experimental value of 30° .

6.2.4 Comparison with experimental data

To test the ability of element erosion to be used as a predictive tool, analyses were carried out for a range of velocities, and compared with experimental data. Experimental results were taken from Schonberg, Bean and Darzi [2]. As the ratio of projectile diameter to plate thickness affects the results of impact on to a thin plate, two sets of analyses were performed for different diameter to thickness ratios. In all the experimental results the projectile was a sphere of aluminium 1100 and the target plate was aluminium 6061-T6. The deformation erosion criterion was used with an erosion value of 2.5. The deformation criterion was used as it produced the best results for the hole diameter.

The first set of analyses were for a 7.95mm aluminium sphere impacting a 1.6mm aluminium plate, a ratio of 4.97:1. This is very similar to the ratio for the Frey test case of 5:1. Figure 6.20 compares the experimental results and the DYNA3D results for the hole diameter. There is one experimental data point of 19.08mm, this appears high and might be a typographical error in [2] as the value given is 0.751 inches but a

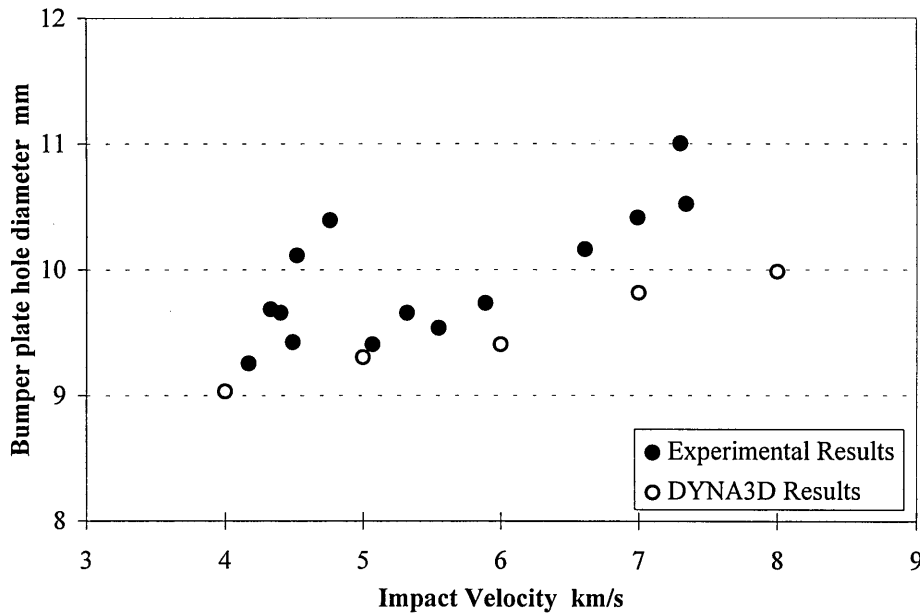


Figure 6.21: Comparison of bumper plate impact analyses with experimental results. T/d ratio 7.83:1

value of 0.651 inches (16.54mm) would be in better agreement with the other data. It is not easy to tell by inspection if the DYNA results are predicting the change of hole diameter with velocity reasonably. As a check, a linear fit to the computational and to the experimental data, not including the data point which may be in error, was done using the method of least squares. The gradient of the line for the experimental data was 0.93 while the gradient for the computational data was 0.65, suggesting that the DYNA results are not accurately predicting the change in hole diameter.

The second set of analyses were for a 6.35mm sphere impacting a 0.81mm aluminium plate, a ratio of 7.83:1. Figure 6.21 compares the experimental results and the DYNA3D results for the hole diameter. These results clearly show an increasing error in the DYNA results as the impact velocity increases.

6.2.5 Discussion

There is a significant difference between the results produced for the different erosion criteria. The deformation criterion produced the best results for the hole diameter in the plate, while the accuracy criterion produced the best results for the debris cloud.

Figure 6.22 shows a comparison between the results for the effective plastic strain criterion and the accuracy criterion. The differences in the debris cloud velocities can be clearly seen. The results for the effective plastic strain criterion gave the smallest hole diameter and did not accurately predict the debris cloud velocities.

Figure 6.23 compares the results for the deformation criterion and the accuracy criterion. Although the accuracy criterion gives the best results for hole diameter, it can be clearly seen that few elements survive in the debris cloud, and so little information about the debris cloud can be found. A comparison of the results for the Johnson-

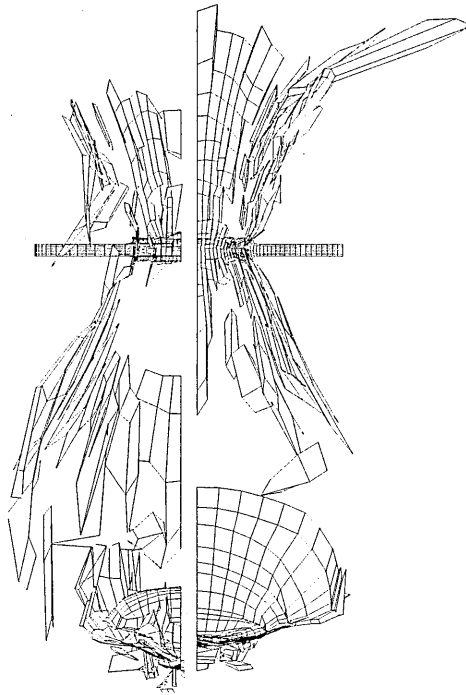


Figure 6.22: Comparison between debris cloud for effective plastic strain criterion (left) with accuracy criterion (right).

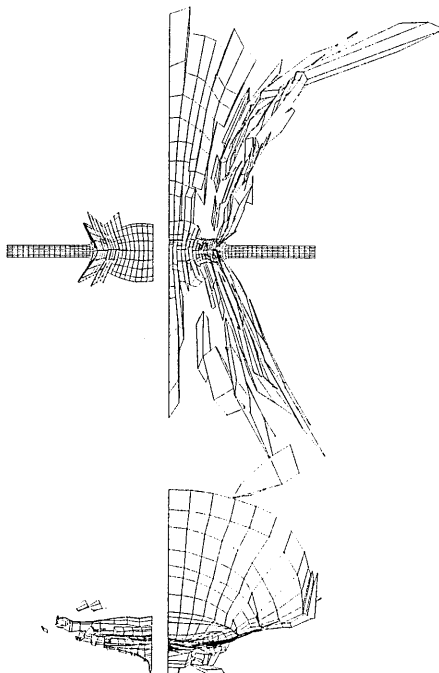


Figure 6.23: Comparison between debris cloud for deformation criterion (left) with accuracy criterion (right).

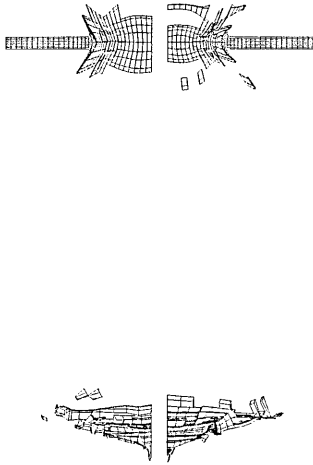


Figure 6.24: Comparison between debris cloud for deformation criterion and Johnson-Cook model (left) with deformation criterion and Steinberg-Guinan model(right).

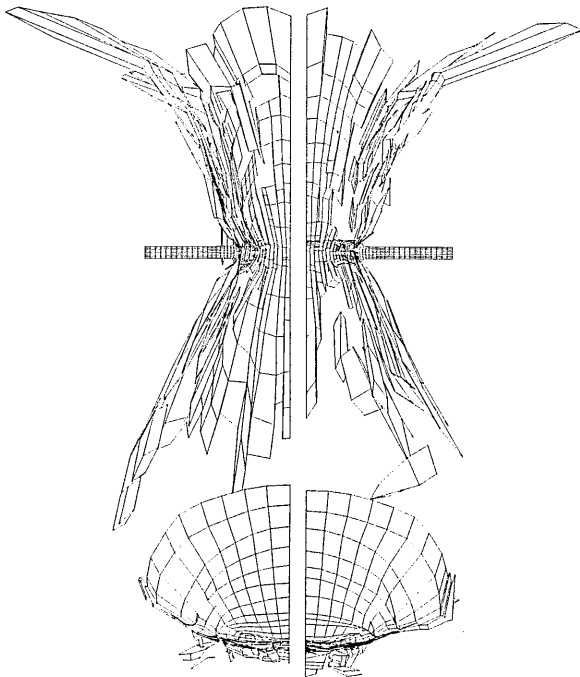


Figure 6.25: Comparison between debris cloud for accuracy criterion and skew angle 0 radians(left) with accuracy criterion and skew angle 0.5 radians (right).

Cook and the Steinberg-Guinan models, both for the deformation criterion, is shown in figure 6.24, overall there is little difference between the two results.

Figure 6.25 compares the results for the accuracy criterion between a skew angle of 0.0 and 0.5. The two results are very similar over all. For the skew angle of 0.5 radians there are fewer elements in the debris cloud and the splash. The erosion of these deformed elements has resulted in a reduction in the required CPU time, from 69793 s to 51833 s.

The most suitable erosion criterion to use when modelling impact on a thin plate depends on the information required. If the information is the hole diameter then the deformation criterion gives the best results and is computationally the cheapest. The apparent computational efficiency of the deformation criterion is partly because it erodes all the highly deformed elements, the result being that the debris cloud is not adequately modelled. If the information required is about the debris cloud then the accuracy criterion must be used, even though it is computationally expensive.

6.3 Discussion

The tests of the erosion criteria have shown that the erosion criterion used can have a strong effect on the final results. This makes the use of a suitable criterion important.

The results for the effective plastic strain criterion show that, even when extended to work with the Johnson-Cook model, it has not proved an effective criterion for hypervelocity impact. For both semi-infinite and thin plate targets it did not produce useful results, and the upper limit of erosion strain was set by the reliability of the contact unlike the other two algorithms tested.

The deformation criterion has proved useful. For the semi-infinite target case it produced the best estimate to the crater size, and for the thin plate target it produced the best estimate for hole diameter. It also proved to be the computationally cheapest, this was principally because all the highly deformed elements were eroded which allowed a larger average time step size.

The element accuracy criterion does not provide any benefit when modelling a semi-infinite target, but when modelling a thin plate target it allows information on the debris cloud to be measured. The aim when developing the accuracy criterion was to erode the minimum necessary number of elements from the calculation. By doing this it was hoped that sufficient elements would survive in the debris cloud to allow the modelling of subsequent impacts. As implemented, the accuracy criterion does erode fewer elements than the deformation criterion, which has allowed the measurement of some debris cloud velocities. With an erosion value of 2.5, approximately half of the projectile has been eroded, with the elements having been eroded from the higher density - higher velocity leading portions of the debris cloud. This missing material will result in a significant difference in the impulse applied to any subsequent plates impacted by the cloud, between the computational and real world results.

The development and testing of the erosion criteria presented here represents the first iteration of a development cycle. If element erosion is to be used for modelling the formation of a debris cloud it is necessary to refine the accuracy criterion, then implement and test the improvements. The computational results show that a criterion which retains deformed but still numerically accurate elements does offer benefits. A

refined criterion would need to erode fewer elements and ideally be computationally cheaper. However it is not clear that the improvement gained would justify the effort required. No-matter what the criterion used some elements will have to be eroded during the penetration process, and so material will be missing from the debris cloud. With eroded material the final results can never be completely accurate. It is not clear that a suitable criterion is possible as substantial changes would be required to the existing criterion.

At this point it was decided that it was more beneficial to pursue an approach which offers the ability to model large deformations without it being necessary to remove material from the calculation. The available alternatives were considered in chapter 3, and the SPH method was selected as the most suitable.

Chapter 7

Smoothed Particle Hydrodynamics

This chapter reviews the background and current state of the meshless method Smoothed Particle Hydrodynamics (SPH). First the background and the basic formulation of the method will be presented. Then the current state of development work will be discussed.

The chapter is split into the following sections:

- 7.1 Basic Formulation
- 7.2 SPH Problems
- 7.3 Proposed Solutions
- 7.4 Summary

In chapter 3 it was shown that the capability to model large material deformations was necessary in order to model impact on multi-plate structures. The problems that the conventional Lagrangian finite element approach have with large deformations are well known. The following available options to solve those problems were looked at:

- Element erosion
- Remeshing or Adaptive meshing
- Arbitrary Lagrangian Eulerian (ALE)
- Eulerian - Lagrangian coupling
- meshless Lagrangian method

The advantages of a meshless Lagrangian method are that it can model arbitrarily large deformations without the computational problems that occur with the Lagrangian finite element method. Coupling of the method with Lagrangian finite elements is considerably simpler than ALE or Eulerian. It is also possible to replace deformed finite elements with meshless nodes, effectively using an erosion algorithm.

Smoothed Particle Hydrodynamics (SPH) was initially developed in 1977 by Gingold and Monaghan [72] and by Lucy [73], who used the method to model astrophysics problems. Their breakthrough was a method for calculation of derivatives that did not require a structured computational mesh. Over the following ten years SPH continued to be developed for astrophysics problems, principally by Monaghan and his co-workers [74, 75, 76]. The state of SPH development at this time is covered in two review papers, one by Benz [58] and the other by Monaghan [77].

In 1990 Libersky and Petchek [78] extended SPH to work with the full stress tensor, developing a 2-D plane strain formulation. This addition allowed SPH to be used in problems where material strength is important, such as lower velocity impacts, and the later stages of hypervelocity impact. Work on developing SPH with strength of materials was continued with the development of 3D codes [34] and the linking of SPH with existing finite element codes [61, 79]. More recent work has concentrated on solving main problems that the current form of the SPH method suffers from: the tensile instability, zero-energy modes and treatment of boundary conditions [35, 80, 81, 82]. These problems will be covered in more detail in section 7.3.

SPH is not the only meshless method that is being currently developed. Principal alternatives for problems with strength of materials are Belytschko's Element Free Galerkin Method [60] and Liu's Reproducing Kernel Particle Method [59]. In general these methods offer greater accuracy than SPH, but at a greater computational cost.

7.1 Basic Formulation

7.1.1 Kernel Approximation

The basis of the SPH method is an interpolation technique that approximates a function in terms of its values at an arbitrary set of points.

The derivation of this method starts with the identity

$$f(x) = \int_V f(x') \delta(x - x') dx' \quad (7.1)$$

where f is a function of position vector x defined in the domain V , and δ is the Dirac delta function. Replacing $\delta(x - x')$ by kernel function $W(x - x', h)$, where h is known as the smoothing length, gives an approximation to f

$$\langle f(x) \rangle = \int_V f(x') W(x - x', h) dx' \quad (7.2)$$

The angle brackets $\langle \rangle$ denote a kernel approximation.

Following Lucy [73] the kernel function W must satisfy the following conditions:

$$\int_V W(x - x', h) dx' = 1 \quad (7.3)$$

and

$$W(x - x', h) = 0 \text{ for } x - x' > \sigma \quad (7.4)$$

Usually $\sigma = 2h$.

These two conditions ensure that as $h \rightarrow 0$, $W(x - x', h) \rightarrow \delta(x - x')$.

Now take the case where the function $f(x)$ is only known at N discrete points. In this case the integral in (7.2) can be approximated by a summation:

$$\langle f(x) \rangle = \sum_{j=1}^N \frac{m_j}{\rho_j} f_j W(x - x_j, h) \quad (7.5)$$

This equation is the basis of the SPH method, where the function f is known at a set of discrete points, usually referred to in SPH literature as particles. SPH equations take the general form of calculating a variable at a particle, denoted by subscript i , by summing the contributions from a set of neighbour particles, denoted by subscript j , for which $W \neq 0$. So (7.5) is written:

$$\langle f(x_i) \rangle = \sum_{j=1}^N \frac{m_j}{\rho_j} f_j W(x_i - x_j, h) \quad (7.6)$$

7.1.2 Accuracy of Kernel Approximation

Following Monaghan [74] and Benz [58] the accuracy of the kernel approximation can be estimated. This is done by expanding the function $f(x')$ in (7.2) in a Taylor series about the point $x' = x$, which gives

$$\langle f(x) \rangle = \int_V \{ f(x) + (x - x') f'(x) + \dots \} W(x - x', h) dx' \quad (7.7)$$

If W is an even function of $x - x'$, meaning $W(x - x', h) = W(|x - x'|, h)$, and satisfies equation 7.3 then the odd powers of $(x - x')$ vanish:

$$\int_V (x - x') f'(x) W(x - x', h) dx' = 0 \quad (7.8)$$

In this case the following relation applies:

$$\langle f(x) \rangle = f(x) + E \quad (7.9)$$

where E is the error term, which is second order. So the kernel approximation is at least equivalent to linear interpolation.

7.1.3 Kernels

There have been many kernel or smoothing functions proposed in SPH literature [58, 77]. The most common is the B-spline kernel proposed by Monaghan [83], which is:

$$W(v) = \frac{c}{h^n} \begin{cases} (1 - \frac{3}{2}v^2 + \frac{3}{4}|v|^3) & 0 \leq |v| \leq 1 \\ \frac{1}{4}(2 - |v|)^3 & 1 \leq |v| \leq 2 \\ 0 & \text{otherwise} \end{cases} \quad (7.10)$$

where $v = (x_i - x_j)/h$, n is the number of dimensions, and c is a normalisation constant with values

$$\frac{2}{3} \text{ for } n = 1 \quad \frac{10}{7\pi} \text{ for } n = 2 \quad \frac{1}{\pi} \text{ for } n = 3$$

This kernel has compact support, it is defined as 0 for $v > 2$, is always positive and has a continuous second derivative.

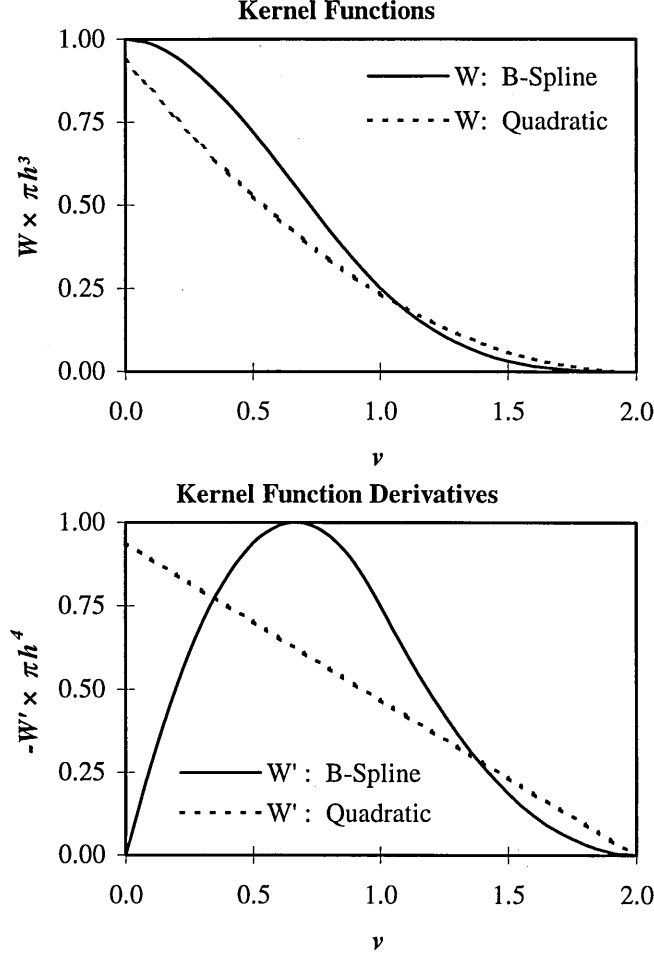


Figure 7.1: B-Spline and Quadratic kernel functions for 3 dimensions

An alternative kernel is the quadratic kernel, proposed by Johnson [84], which is:

$$W(v) = \frac{c}{\pi h^n} \left(\frac{3}{8}v^2 - \frac{3}{2}|v| + \frac{3}{2} \right) \quad 0 \leq |v| \leq 2 \quad (7.11)$$

and the values of c are:

$$\frac{1}{2} \text{ for } n = 1 \quad \frac{1}{\pi} \text{ for } n = 2 \quad \frac{5}{8\pi} \text{ for } n = 3$$

7.1.4 Semi-discretisation of the Conservation Equations

In this section the derivation of the basic SPH form of the conservation equations is summarised, and the assumptions made during the derivations are highlighted. For the full derivation of the equations see Appendix C.

The conservation equations are:

$$\frac{d\rho}{dt} = -\rho \frac{\partial v^\beta}{\partial x^\beta} \quad (7.12)$$

$$\frac{dv^\alpha}{dt} = \frac{1}{\rho} \frac{\partial \sigma^{\alpha\beta}}{\partial x^\beta} \quad (7.13)$$

$$\frac{dE}{dt} = \frac{\sigma^{\alpha\beta}}{\rho} \frac{dv^\alpha}{dx^\beta} \quad (7.14)$$

Following Monaghan [85] the momentum and energy equations are rewritten, and all three equations are written as kernel estimates.

$$\left\langle \frac{d\rho}{dt} \right\rangle = - \int_V W \rho' \frac{\partial v'^\beta}{\partial x'^\beta} dx' \quad (7.15)$$

$$\left\langle \frac{dv^\alpha}{dt} \right\rangle = \int_V W \frac{\partial}{\partial x'^\beta} \left(\frac{\sigma'^{\alpha\beta}}{\rho'} \right) dx' + \int_V W \frac{\sigma'^{\alpha\beta}}{\rho'^2} \frac{\partial \rho'}{\partial x'^\beta} dx' \quad (7.16)$$

$$\left\langle \frac{dE}{dt} \right\rangle = \int_V W \frac{\sigma'^{\alpha\beta}}{\rho'^2} \frac{\partial (\rho' v'^\alpha)}{\partial x'^\beta} dx' - \int_V W \frac{\sigma'^{\alpha\beta} v'^\alpha}{\rho'^2} \frac{\partial \rho'}{\partial x'^\beta} dx' \quad (7.17)$$

Now simplify the equations, this step is best covered by Benz [58]. Equations 7.15–7.17 all contain integrals of the form

$$\int_V f(x') \frac{\partial g(x')}{\partial x'} W dx' \quad (7.18)$$

Expanding $f(x') \frac{\partial g(x')}{\partial x'}$ in a Taylor series about point $x' = x$ gives

$$\int_V f(x') \frac{\partial g(x')}{\partial x'} W dx' = \int_V \left\{ f(x) \frac{\partial g(x)}{\partial x} + (x' - x) \frac{d}{dx} \left(f(x) \frac{\partial g(x)}{\partial x} \right) + \dots \right\} W dx' \quad (7.19)$$

As W is an even function of $x - x'$ the first order term will drop, see section 7.1.2. Neglecting second order and higher terms, which is consistent with the overall order of the method, again see section 7.1.2, gives the following approximation:

$$\int_V f(x') \frac{\partial g(x')}{\partial x'} W dx' = \left(f(x') \frac{\partial g(x')}{\partial x'} \right)_{x'=x} \quad (7.20)$$

Substituting $\frac{\partial}{\partial x'} \langle g(x) \rangle$ for $\frac{\partial}{\partial x'} g(x')$ gives

$$\left(f(x') \frac{\partial g(x')}{\partial x'} \right)_{x'=x} = f(x) \int_V \frac{\partial g(x')}{\partial x'} W dx' \quad (7.21)$$

Substituting the final term in (7.21) for (7.18) simplifies the SPH equations by only using the kernel interpolation for the calculation of the gradient terms.

Applying this to equations 7.15–7.17 gives

$$\left\langle \frac{d\rho}{dt} \right\rangle = -\rho \int_V W \frac{\partial v'^\beta}{\partial x'^\beta} dx' \quad (7.22)$$

$$\left\langle \frac{dv^\alpha}{dt} \right\rangle = \int_V W \frac{\partial}{\partial x'^\beta} \left(\frac{\sigma'^{\alpha\beta}}{\rho'} \right) dx' + \frac{\sigma^{\alpha\beta}}{\rho^2} \int_V W \frac{\partial \rho'}{\partial x'^\beta} dx' \quad (7.23)$$

$$\left\langle \frac{dE}{dt} \right\rangle = \frac{\sigma^{\alpha\beta}}{\rho^2} \int_V W \frac{\partial (\rho' v'^\alpha)}{\partial x'^\beta} dx' - \frac{\sigma^{\alpha\beta} v^\alpha}{\rho^2} \int_V W \frac{\partial \rho'}{\partial x'^\beta} dx' \quad (7.24)$$

The equations are now integrated by parts, and the resulting surface integrals are neglected. The assumption is that either W , or the variable, or both vanish on the boundary of the body. For astrophysical hydrodynamics the boundaries can be taken at infinity [74] and so this assumption can be justified. For more general problems, where the bodies have well defined boundaries, this assumption is no longer valid but remains as the resulting equations are considerably simplified. This step has the effect of moving the differentials of the field variables to the kernel.

$$\left\langle \frac{d\rho}{dt} \right\rangle = \rho \int_V v'^\beta \frac{\partial W}{\partial x'^\beta} dx' \quad (7.25)$$

$$\left\langle \frac{dv^\alpha}{dt} \right\rangle = - \int_V \frac{\sigma'^{\alpha\beta}}{\rho'} \frac{\partial W}{\partial x'^\beta} dx' - \frac{\sigma^{\alpha\beta}}{\rho^2} \int_V \rho' \frac{\partial W}{\partial x'^\beta} dx' \quad (7.26)$$

$$\left\langle \frac{dE}{dt} \right\rangle = - \frac{\sigma^{\alpha\beta}}{\rho^2} \int_V \rho' v'^\alpha \frac{\partial W}{\partial x'^\beta} dx' - \frac{\sigma^{\alpha\beta} v^\alpha}{\rho^2} \int_V \rho' \frac{\partial W}{\partial x'^\beta} dx' \quad (7.27)$$

Now the equations are evaluated by the particle method and rearranged to give

$$\frac{d\rho_i}{dt} = \rho_i \sum_j \frac{m_j}{\rho_j} (v_j^\beta - v_i^\beta) \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (7.28)$$

$$\frac{dv_i^\alpha}{dt} = - \sum_j m_j \left(\frac{\sigma_j^{\alpha\beta}}{\rho_j^2} + \frac{\sigma_i^{\alpha\beta}}{\rho_i^2} \right) \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (7.29)$$

$$\frac{dE_i}{dt} = - \frac{\sigma_i^{\alpha\beta}}{\rho_i^2} \sum_j m_j (v_j^\alpha - v_i^\alpha) \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (7.30)$$

These are the most common SPH forms of the conservation equations. In these equations $W_{ij} = W(x_i - x_j, h)$.

The step in the derivation between equations (7.22–7.24) and (7.25–7.27) was based on the assumption that surface integrals are zero. This assumption is one of the most significant assumptions made in the derivation of the basic SPH equations. However this assumption has never been seriously challenged because of good simulation results [34]. It is possible to ignore the boundaries as SPH still gives approximately zero stress components perpendicular to the free surface. This occurs because the particles near the boundary do not have a complete set of neighbour particles, and missing particles produce the same effect as particles with all components of the stress tensor equal to zero, so near a boundary the stress goes to zero anyway. Although allowing simple and apparently reasonable calculations to be performed, the assumption has caused some well known problems due to the kernel sum deficiency near the boundary. This is best illustrated by the kernel estimate of density, which is:

$$\langle \rho \rangle = \int_V \rho' W dx' = \sum_j \frac{m_j}{\rho_j} \rho_j W_{ij} = \sum_j m_j W_{ij} \quad (7.31)$$

This simple method leads to particles near a boundary having a lower density than a particle far from a boundary, this is the reason that the continuity equation is usually used to update the density for non-astrophysical problems.

A discussion of the effect of retaining the boundary terms in the equations can be found in Campbell [86]. He derives the momentum equation in one dimension as:

$$\begin{aligned} \frac{dv_i}{dt} = & - \left[\frac{P_B}{\rho_B^2} + \frac{P_i}{\rho_i^2} \right] \rho_B W(|r_i - r_B|, h) \Delta S_i \\ & - \sum_j m_j \left[\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right] \nabla W(|r_i - r_j|, h) \end{aligned} \quad (7.32)$$

where ρ_B is the density at the boundary, P_B is the boundary pressure and

$$\Delta S_i = \pi((2h_i)^2 - |r_i - r_B|^2) \quad (7.33)$$

which is an estimate of the area of intersection of the boundary with the domain of influence of particle i . Here pressure P is used rather than stress σ as Campbell was not considering the full stress tensor. It is clear that the boundary pressure is felt if the boundary is in the neighbourhood of particle i , and is present in a form very similar to that for any other neighbour particle j .

7.1.5 Artificial viscosity for SPH

In order to model shock waves SPH uses an artificial viscosity to smooth the shocks into continuous waves. A commonly used artificial viscosity was proposed by Monaghan and Gingold [85], and is

$$\Pi_{ij} = \begin{cases} \frac{-\alpha c_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\rho_{ij}} & \mu_{ij} < 0 \\ 0 & \mu_{ij} \geq 0 \end{cases} \quad (7.34)$$

where

$$\mu_{ij} = \frac{h(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|^2 + \epsilon h^2} \quad (7.35)$$

and

$$c_{ij} = \frac{1}{2}(c_i + c_j) \quad \rho_{ij} = \frac{1}{2}(\rho_i + \rho_j) \quad (7.36)$$

α and β are the linear and quadratic viscosity coefficients, and ϵ is usually taken as 0.1. This viscosity is also known as a bond viscosity as it is calculated for each particle pair. It is in effect a viscous force that resists a reduction in the distance between a particle pair (compression) but does not resist an increase in the distance between a particle pair (tension).

The effects of artificial viscosity on SPH calculations has been studied by Johnson [87, 81]. He observes that the inaccuracies introduced by the bond viscosity vary from problem to problem, and that caution is required when using this viscosity with some classes of problem.

7.1.6 Variable Smoothing Length

So far it has been assumed that the smoothing length, h , is constant and identical for all particles. It is easy to see that in a large deformation calculation a particle could become separated by more than $2h$ from all its neighbours, and so will no-longer interact. Varying the smoothing length can prevent this. It also allows a higher resolution in areas of interest if required. Introduction of a variable smoothing length raises two main questions: how should the smoothing length change? how does it affect the existing SPH equations?

In order to vary h , Benz [58] suggests

$$h = h_0 \left(\frac{\rho_0}{\rho} \right)^{\frac{1}{3}} \quad (7.37)$$

This equation is designed to keep the number of neighbour particles roughly constant. Other methods for updating h have been suggested [88, 58], with the same intention to keep the number of neighbour particles approximately constant.

The use of a varying h does require modification of the SPH equations, and this is covered by Swegle [35]. The main change required is to ensure conservation of momentum, as in the existing equation (7.29) with $h_i \neq h_j$ then $\dot{v}_{ij} \neq -\dot{v}_{ji}$. A common correction is to replace h by h_{ij} when calculating the kernel, where $h_{ij} = 0.5(h_i + h_j)$.

7.1.7 Implementation

The aspect of the implementation that can have the greatest effect on the computational speed of the code is the neighbour searching routine. As there is no fixed connectivity the list of neighbour particles for every particle has to be found every time step.

The simplest method is to calculate the distance between every particle pair, in order to check if the distance is less than $2h$. This is a very inefficient method, and the

computational cost is proportional to N^2 , where N is the total number of particles. So even for a small number of particles the computational cost of this method becomes prohibitive.

The most common SPH neighbour finding method used for solid mechanics is the linked list [89]. In this method an underlying regular grid is set up, with the length of the sides of each cell as $2h$. Then the particles are sorted so that each cell has a list of the particle contained within it. Then when checking for neighbouring particles only those in the same cell, or the surrounding cells, have to be checked. So for 2D nine cells have to be checked. This is a considerably more efficient algorithm, and the computational cost is proportional to $N \log N$.

A third alternative is the tree method [90], where a cell structure is set up and locally refined until there is only one particle per cell. This is more expensive than the linked list when all particles have similar smoothing lengths, but does offer advantages when there are widely varying smoothing lengths or long range forces, such as gravity in astrophysics calculations.

As SPH is a Lagrangian method the code will have a similar structure to a Lagrangian finite element code. Indeed if both options are present it would be possible for both approaches to use the same material model subroutines.

7.2 SPH Problems

There are well known problems with the basic SPH approach described in the previous section. The principal problems are: accuracy, tensile instability and zero energy modes.

The SPH interpolation is most accurate when using an ordered, uniform, arrangement of particles. A non-uniform arrangement of particles leads to inaccurate calculation of a gradient at a particle [84]. This represents a problem as when a problem involving large deformations is analysed the distribution of particles can become very non-uniform. A series of tests by Meglicki [91] show that node disorder results in a systematic error. His tests also show that low particle density results in error, and an inter particle distance of close to $h/2$ is required to cure this. This is a higher particle density than has been commonly used in SPH, typically an inter particle distance of h , and would result in an increased computational cost. The error that SPH shows near a boundary is a related problem, particles at the boundary of a body do not have a full neighbour set.

7.2.1 Stability

As SPH uses explicit time integration it is only conditionally stable. The stability criterion is the Courant condition:

$$\Delta t \leq \frac{\min \Delta x}{\max(c, u)} \quad (7.38)$$

where Δx is a characteristic length, c is the sound speed and u is the particle velocity.

Stability analyses of the SPH method with artificial viscosity have been carried out at Sandia by Swegle, Guenther, Wen and co-workers [35, 92, 80] and also by Randles and Libersky [82]. Both Guenther and Libersky point out that the sequence of operations does make a difference in the size of the stable time step. For example, the density

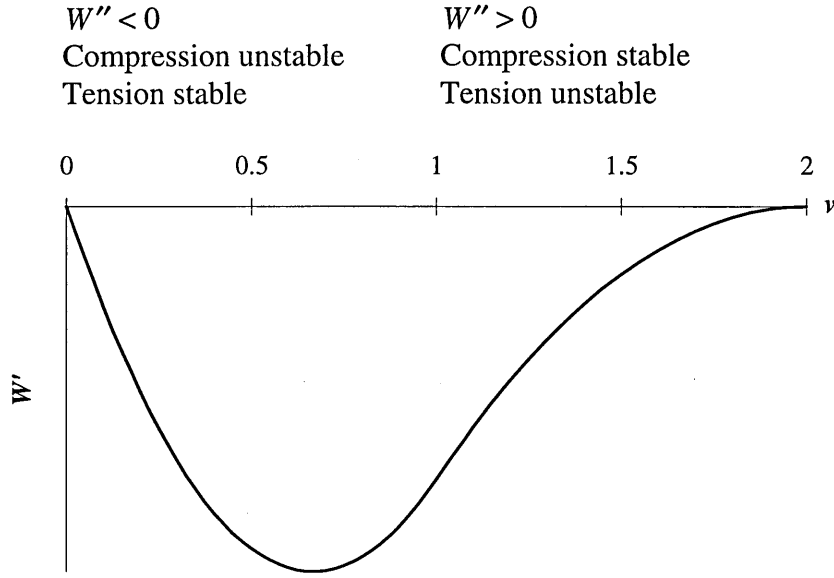


Figure 7.2: B-Spline kernel stability

can be advanced from t^n to t^{n+1} using the velocities at $t^{n-1/2}$ or $t^{n+1/2}$. Randles and Libersky show one sequence that is unstable. This sequence updates velocity, density and internal energy in the same loop, before updating the particle positions, and appears advantageous due to its computational efficiency.

The three reports produced by Sandia present a rigorous analysis of the SPH method. Of the three, the report by Swegle [35] is by far the easiest to read. They have shown that SPH suffers from an instability that is unaffected by the time integration algorithm used, and so is not sensitive to the size of the time step. As this instability is most often seen when high tensile stresses are seen in a material, it is known as the tensile instability. When it occurs it manifests itself as a clumping of particles, which results in fragmentation, but for numerical rather than physical reasons. In hypervelocity impact this is likely to be seen in the expanding debris cloud, and can prevent the calculation reaching a successful completion.

The analyses [35] showed that if the following criterion is true then the method is unconditionally unstable

$$W''\sigma > 0 \quad (7.39)$$

W'' is the second derivative of the kernel function with respect to its argument, and the stress σ is positive in tension.

Figure 7.2 shows the stability regimes for the B-spline kernel. As can be seen the name tension instability is not completely accurate, the instability can manifest itself in both tension or compression, depending on the sign of the second derivative. As a common choice for the distance to the nearest neighbour particle is $v = 1$ then this instability is usually seen in tension. Under high compression it is possible for the instability to develop, the point $W'' = 0$, where W'' changes sign, lies at $v = \frac{2}{3}$. So if the inter-particle distance falls below this the system is unstable in compression. The existence of this change in sign of W'' is a principal reason why Johnson proposed the

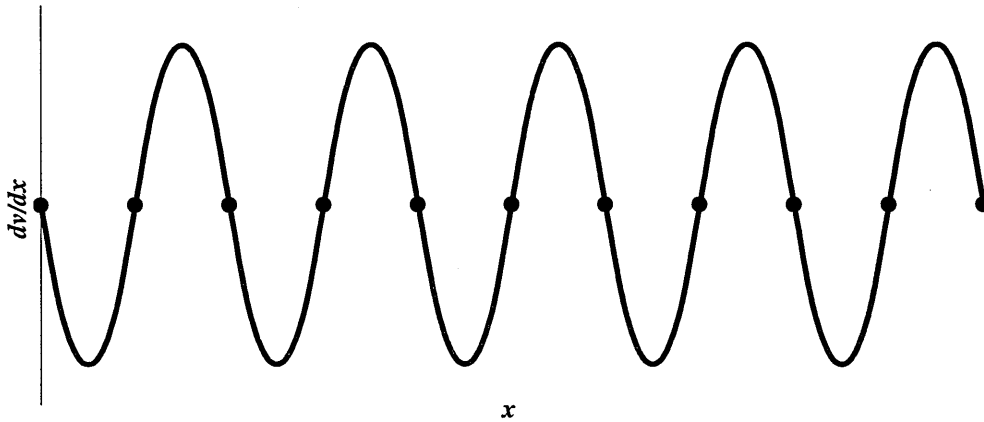


Figure 7.3: Velocity gradient field produced in 1D for alternating particle velocities

Quadratic kernel, whose second derivative is constant and positive.

Swegle also looked at the growth rate of the instability, and found that it was dependent on the magnitude of the stress and the artificial viscosity. So the instability may not always be observed if the conditions lead to a slow growth rate, and the system does not stay in the unstable state for too long.

In addition to showing the existence of the tensile instability, Guenther [92] and Wen [80] proposed a solution, known as Conservative Smoothing, which is covered in section 7.3.1.

7.2.2 Zero energy modes

In his analysis of SPH Swegle [35] showed that zero energy modes do exist in SPH. A zero-energy mode being a pattern of nodal displacement that is not a rigid body motion, and produces zero-strain energy. The zero energy modes are a consequence of the field variables and their derivatives being calculated at the same locations. This leads to the existence of a mode of deformation which is not resisted by stresses within the body. The zero-energy mode in SPH takes the form of an alternating velocity field with a wavelength of twice the inter-particle distance. This velocity field produces a stress field that appears constant at the nodes, but examining the stress gradient will show that the stress field is actually oscillating, see figure 7.3. As the stresses are constant at the nodes the oscillations produce no particle accelerations, and so this motion is not resisted. This behaviour can be explained by remembering that the SPH method is basically a way to fit a curve to a set of discrete data. It is well known that if the order of a curve fit is too high then a straight line can be fit with a sinusoidal curve.

7.3 Proposed Solutions

Considerable work has gone into developing solution to the problems of SPH. In this section proposed solutions to the tensile instability and the particle deficiency at the boundary are described.

7.3.1 The Tensile Instability Treatment

There have been two main proposals for solutions to the tensile instability: the stress point approach proposed by Dyka [93], and the conservative smoothing approach proposed by Guenther, Wen and co-workers [92, 80].

Dyka proposed that the stresses should be calculated at a point other than the SPH nodes. However his work was only in one dimension. This approach effectively uses two sets of particles. The first set carries velocity, mass and smoothing length information, and contains effectively the original SPH nodes. The second set carry stress, density and internal energy information. Two stress points were associated with each velocity point, placed either side of the particle (in 1D) at up to half the inter-particle distance. The SPH equations are solved at the appropriate point, either velocity or stress particle, by summing over its neighbour particles of the other kind. For example: the momentum equation is evaluated at each velocity point by summing over its neighbouring stress points. Dyka claims that this approach solves the tensile instability.

The difficulty with the stress point approach comes with its extension to higher dimensions. How many stress points to use, and how to place them around each velocity particle. It is also important to ensure that the stress points are moved in a Lagrangian manner if damage or constitutive models are used that require material history values to be known. This approach also offers a solution to the zero energy modes observed by Swegle, as by moving the stress points away from the velocity points an alternating stress field will cause accelerations at the velocity points.

Conservative smoothing offers a solution that works in 2 and 3 dimensions, and is easily incorporated into existing SPH codes. It was developed for one dimension by Guenther, Wen and co-workers at Sandia National Laboratories [92, 80].

In one dimension conservative smoothing takes the form:

$$\tilde{v}_i = v_i + \alpha_{cs}(v_{i-1} - 2v_i + v_{i+1}). \quad (7.40)$$

$\tilde{v}_i$ is the smoothed velocity for particle i , v_i is the velocity for particle i calculated from the momentum equation, α_{cs} is a coefficient that Wen takes as 0.25. The smoothed velocity is then used to update the particle position. Guenther [92] also shows that conservative smoothing can replace artificial viscosity in the modelling of shock waves. This equation essentially acts as a filter, smoothing $2h$ oscillations in the variable being smoothed. It is known as conservative smoothing as it has been shown to conserve linear momentum [80]. Conservative smoothing is not restricted just to smoothing the velocity field, and can be used to smooth other fields such as density and internal energy.

This approach has been extended to higher dimensions by Randles and Libersky [82]. Who use:

$$\tilde{v}_i = v_i + \alpha_{cs} \left[\frac{\sum_{j \neq i} \frac{m_j}{\rho_j} v_j W_{ij}}{\sum_{j \neq i} \frac{m_j}{\rho_j} W_{ij}} - v_i \right] \quad (7.41)$$

which is a slight modification of a filter proposed by Balsara [94] for removing small scale noise in an SPH simulation. The modification is the $j \neq i$ exclusion in the kernel sum.

When using this equation to smooth the velocities it is necessary to apply a correction to the internal energy [95]. This is necessary to enforce the conservation of energy as the method does not conserve kinetic energy.

As part of the derivation of the conservative smoothing approach Wen presents a stability analysis of SPH with this approach, and shows that the unconditionally unstable criterion, equation 7.39, has been removed. In addition Guenther states that replacing artificial viscosity by conservative smoothing gives a larger stable time step.

7.3.2 Normalised Smoothing

The idea behind the normalised smoothing function (NSF) algorithm developed by Johnson [84] is to improve the accuracy of the SPH method for non-uniform node distributions and near boundaries. In order to achieve this improvement in accuracy Johnson applies a correction term to the kernel estimate.

The example that Johnson uses in his derivation of the NSF algorithm is that of a radially stretching ring. Looking at a radial cross-section of the ring the material has a linearly varying velocity that produces a constant strain rate. Using ‘classic’ SPH with a regular particle arrangement a constant strain rate is calculated in the interior of the body, with the expected drop off near the boundaries. With a non-regular arrangement of nodes, the method no longer produces a constant strain rate in the interior of the body.

Johnson derives the NSF algorithm by calculating a correction term to make SPH accurately reproduce a linear strain field. In the ring example, the radial velocity is $v = 1.5x$, where x is the x co-ordinate of the material point. The SPH term for the strain rate is:

$$\dot{\epsilon}_x = - \sum_j \frac{m_j}{\rho_j} (v_j - v_i) \frac{\partial W_{ij}}{\partial x} \quad (7.42)$$

Add a correction term B

$$\dot{\epsilon}_x = -B_x \sum_j \frac{m_j}{\rho_j} (v_j - v_i) \frac{\partial W_{ij}}{\partial x} \quad (7.43)$$

For the strain rate to be constant the velocities are

$$v_j - v_i \equiv \dot{\epsilon}_x r_{ij} \quad (7.44)$$

where r_{ij} is the difference in the x co-ordinate of the two nodes. Substituting equation 7.44 into equation 7.43 gives

$$\dot{\epsilon}_x = -B_x \sum_j \frac{m_j}{\rho_j} \dot{\epsilon}_x r_{ij} \frac{\partial W_{ij}}{\partial x} \quad (7.45)$$

The strain rate cancels and so

$$B_x = \left[- \sum_j \frac{m_j}{\rho_j} r_{ij} \frac{\partial W_{ij}}{\partial x} \right]^{-1} \quad (7.46)$$

Johnson shows that using this equation to calculate the strain rates gives a constant and accurate value for the strain rate, even with a non-uniform particle arrangement, everywhere except at the corner of a body. At the corners there are 1–8% over estimations of the strain rate.

The effect of using kernel normalisation can be seen in figure 7.4. The figure shows normalised and non-normalised SPH interpolations of the gradient of a linear function in 1D using the B-spline kernel. Figure 7.4 a shows the interpolation for equally spaced nodes, in the centre of the body, good interpolation is seen, but near the edge the kernel deficiency is clear for the non-normalised case (black line). Figure 7.4 b shows the same case, but with unequally spaced nodes. The inner nodes have been moved by up to 10% of the original inter-particle distance from their original positions, and the outer two nodes have not moved. Here the inaccuracies of non-normalised SPH can be clearly seen, while normalised SPH still provides a good estimate of the gradient.

Randles and Libersky [82] have extended this method to get a normalisation procedure for general tensor fields. With this the normalisation can be used for every kernel sum carried out during the calculation. They derive their general normalisation in the same manner. Take a linear tensor of the form $c^{\alpha\beta} = a^{\alpha\beta} + b^{\alpha\beta\gamma} x^\gamma$ where a and b are constant second and third rank tensors, symmetric in the first two indices. Then derive an equation that can calculate an exact value for the gradient of c everywhere within the body being considered.

$$\frac{\partial c^{\alpha\beta}}{\partial x^\alpha} = \left[- \sum_j \frac{m_j}{\rho_j} (c_j^{\alpha\gamma} - c_i^{\alpha\gamma}) \frac{\partial W_{ij}}{\partial x^\gamma} \right] B^{\gamma\beta} \quad (7.47)$$

Evaluating equation 7.47 for the linear tensor field allows the calculation of $B^{\gamma\beta}$

$$B^{\gamma\beta} = \left[- \sum_j \frac{m_j}{\rho_j} (x_j^\gamma - x_i^\beta) \frac{\partial W_{ij}}{\partial x_i^\beta} \right]^{-1} \quad (7.48)$$

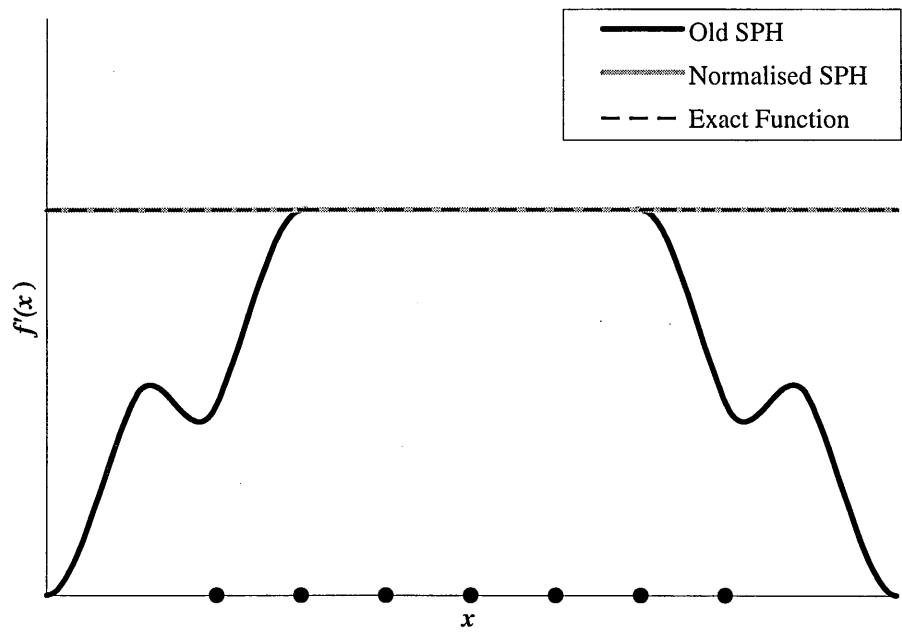
Using this method Randles and Libersky derived normalised forms of the conservation equations:

$$\frac{d\rho_i}{dt} = \rho_i \left[\left(- \sum_j \frac{m_j}{\rho_i \rho_j} (v_j^\alpha - v_i^\alpha) \frac{\partial W_{ij}}{\partial x^\alpha} \right) B^{\alpha\alpha} \right] \quad (7.49)$$

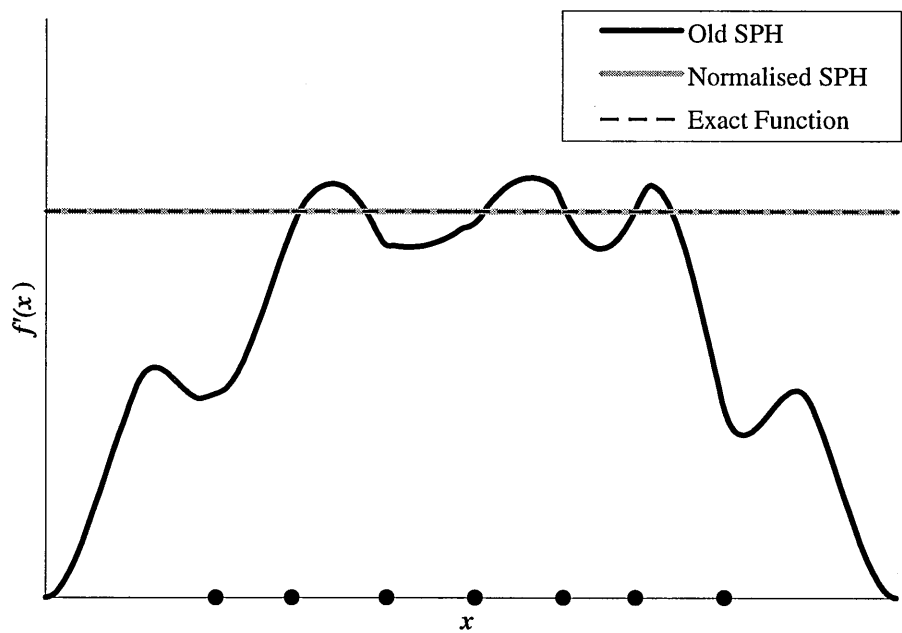
$$\frac{dv_i^\alpha}{dt} = \left(- \sum_j \frac{m_j}{\rho_i \rho_j} (\sigma_j^{\alpha\beta} - \sigma_i^{\alpha\beta}) \frac{\partial W_{ij}}{\partial x_i^\gamma} \right) B^{\gamma\beta} \quad (7.50)$$

$$\frac{dE_i}{dt} = \sigma^{\alpha\beta} \left[\left(- \sum_j \frac{m_j}{\rho_i \rho_j} (v_j^\alpha - v_i^\alpha) \frac{\partial W_{ij}}{\partial x^\gamma} \right) B^{\gamma\beta} \right] \quad (7.51)$$

In order to perform this normalisation a ‘stress-difference’ form of the momentum equation has to be used [82], as opposed to the ‘stress-sum’ form used in equation 7.29.



a: Equally spaced nodes



b: Unequally spaced nodes

Figure 7.4: Normalised and non-normalised B-spline kernel estimates for the gradient of a linear function.

It is also possible to derive the kernel normalisation as a correction to the kernel for the presence of a boundary within the support of the kernel function. For simplicity the following equations are derived for one dimension, but the results are easily generalised. For a kernel with compact support equation 7.3 may be written

$$\int_{-2h}^{2h} W(x - x', h) dx' = 1 \quad (7.52)$$

where $-2h$ to $2h$ are the limits of the kernel function. If a boundary lies at point m , where $0 \leq m \leq 2h$, the integral will no longer equal one, so a correction term, B , is required

$$B \int_{-2h}^m W(x - x', h) dx' = 1 \quad (7.53)$$

so

$$B = \left[\int_{-2h}^m W(x - x', h) dx' \right]^{-1} \quad (7.54)$$

The same procedure can be carried out for the kernel approximation of a function, equation 7.2 by enforcing

$$\int_{-2h}^{2h} f(x') W(x - x', h) dx' = B \int_{-2h}^m f(x') W(x - x', h) dx' \quad (7.55)$$

which results in the same correction term. Evaluating the integrals by the particle method gives

$$\langle f(x_i) \rangle = \frac{\sum \frac{m_j}{\rho_j} f(x_j) W_{ij}}{\sum \frac{m_j}{\rho_j} W_{ij}} \quad (7.56)$$

This equation is identical to the lowest order, order 0, form of the Moving Least Square approximation, often called Shepard functions [96]. To obtain the gradient of the estimate of f at point i , and knowing that:

$$\frac{\partial}{\partial x} \langle f(x) \rangle = \left\langle \frac{\partial f(x)}{\partial x} \right\rangle \quad (7.57)$$

differentiate (7.56):

$$\frac{d}{dx} \langle f(x_i) \rangle = \frac{\left(\sum \frac{m_j}{\rho_j} W_{ij} \right) \left(\sum \frac{m_j}{\rho_j} f(x_j) \frac{dW_{ij}}{dx} \right) - \left(\sum \frac{m_j}{\rho_j} f(x_j) W_{ij} \right) \left(\sum \frac{m_j}{\rho_j} \frac{dW_{ij}}{dx} \right)}{\left(\sum \frac{m_j}{\rho_j} W_{ij} \right)^2} \quad (7.58)$$

rearranging, and substituting $f(x_i)$ using equation 7.56 gives

$$\frac{d}{dx} \langle f(x_i) \rangle = \frac{\sum \frac{m_j}{\rho_j} (f(x_j) - f(x_i)) \frac{dW_{ij}}{dx}}{\sum \frac{m_j}{\rho_j} W_{ij}} \quad (7.59)$$

This equation does not give an accurate estimate for the gradient of a linear function. Correcting it in the same manner as for equation 7.47 results in

$$\frac{df_i}{dx} = \frac{\sum \frac{m_j}{\rho_j} (f(x_j) - f(x_i)) \frac{dW_{ij}}{dx}}{\sum \frac{m_j}{\rho_j} (x_j - x_i) \frac{dW_{ij}}{dx}} \quad (7.60)$$

which is the normalised SPH estimate to the gradient of a function.

The kernel normalisation approach offers a method to significantly improve the SPH method by improving its general accuracy and especially offering a solution to the kernel sum deficiency near boundaries. So as a result solution variables, such as stress, no longer tend to zero near boundaries.

Another approach to improving the accuracy of the SPH method is to use moving least square (MLS) interpolants. This approach offers similar benefits to the NSF approach [97].

7.3.3 Boundary Conditions

In the original form of SPH it was possible to ignore the traction boundary condition. The traction boundary condition is:

$$\sigma^{\alpha\beta} n^\alpha = t^\alpha \quad (7.61)$$

where n^α is the outward unit normal at a point on the boundary, and t^α is the surface traction vector at that point. The most important case of the traction boundary condition is the free-surface condition, when $t = 0$. It was possible to ignore the free-surface condition as the stress tensor naturally tended to zero at a point $2h$ from the boundary SPH particles, due to the lack of neighbour particles.

When using the normalised conservation equations, 7.49–7.51, the free-surface boundary condition is no longer satisfied at boundaries. The kernel normalisation acts as a correction for the lack of neighbours. With the stress free condition not applied un-physical behaviour is seen, with stress and shock waves not being reflected from a free surface.

Randles and Libersky [82] suggest explicitly applying the stress free condition at the boundaries of a body. As in SPH no particle lies exactly on a material boundary, unlike finite element nodes, they suggest arbitrarily choosing the boundary to lie a distance $\frac{h}{2}$ from a boundary particle along the outward surface normal. At this point they apply the free surface condition by setting $\sigma^{\alpha\beta} n^\alpha = 0$. Then interpolate for the value of the stress at the boundary particle from the stress at interior particles and the stress at the boundary.

Determining the boundary particles and the surface normals is not a trivial problem in SPH. Randles and Libersky have adapted an elegant method called the gradient of unity method. In this method a kernel estimate is made of the ‘kind’ of a body, denoted by Ψ . A boundary is found by the following inequality being large enough to indicate that the particle is next to a void or particles of a different ‘kind’.

$$\Psi_i \neq \sum_j \frac{m_j}{\rho_j} \Psi_j W_{ij} \quad (7.62)$$

Furthermore the method can be used to find an estimate for the surface normal by considering the gradient of the kernel estimate of 'kind':

$$\hat{n}_i^\alpha = \pm \left(\frac{\frac{\partial \Psi_i}{\partial x_i^\alpha}}{\left| \frac{\partial \Psi_i}{\partial x_i} \right|} \right) \quad (7.63)$$

where:

$$\frac{\partial \Psi_i}{\partial x^\alpha} = - \sum_j \frac{m_j}{\rho_j} \Psi_j \frac{\partial W_{ij}}{\partial x_i^\alpha} \quad \text{and} \quad \left| \frac{\partial \Psi_i}{\partial x_i} \right| = \left[\sum_\alpha \left(\frac{\partial \Psi_i}{\partial x_i^\alpha} \right)^2 \right]^{\frac{1}{2}} \quad (7.64)$$

This method allows boundaries and boundary normal vectors to be automatically found, and the free surface boundary condition to be satisfied on the boundary.

7.4 Summary

The past seven years have shown increasing interest in applying the SPH method to solid mechanics problems, such as hypervelocity impact. The basic SPH method has several problems when used to model a solid body.

- Tensile instability
- Kernel deficiency near boundaries
- Inaccuracy with disordered points
- Zero-energy modes

Much work has recently gone into finding solutions to these problems. This work has lead to the development of Conservative Smoothing to solve the tensile instability and the Normalised Smoothing Function approach to solve the consistency, accuracy and boundary problems.

With the normalised smoothing function based SPH method, it is no-longer possible to ignore boundary conditions. This was possible under old SPH as solution variables tended to zero at boundaries due to insufficient neighbour particles. More work is required before improved SPH is ready to be used as a general method.

Chapter 8

Treatment of contact boundary condition in SPH

In this chapter a contact algorithm for SPH is developed. The algorithm is tested in 1D and 2D. These tests show that the contact algorithm excites a zero energy mode, which can lead to significant non-physical particle motions. A 1D algorithm is then developed that does not suffer from the zero energy mode excited, and tests show that this algorithm is a solution to the problem.

The chapter is split into the following sections:

- 8.1 The SPH code
- 8.2 Contact algorithm
- 8.3 Enforcing the contact condition
- 8.4 Implementation of the contact algorithm
- 8.5 Contact with two particle normals
- 8.6 Tests of contact algorithm
- 8.7 Displaced stress point method
- 8.8 Tests of displaced stress point approach
- 8.9 Discussion

In chapter 7, the current state of progress in the SPH method has been presented. Major steps have been taken towards the production of a consistent and accurate meshless method. Conservative smoothing has provided a cure to the tensile instability. Normalised kernels and moving least square methods give first order consistency. These improvements have highlighted another important area that requires attention, boundary conditions. Randles and Libersky [82] have developed a method to apply the stress free condition to boundaries. This method is easily extended to the general traction boundary condition, when the surface traction vector is known.

This still leaves the contact boundary condition un-addressed. For two bodies in frictionless contact, with the sections of the boundary of each body that are in contact denoted as ΔB_1 and ΔB_2 , the contact boundary condition can be expressed as:

$$x_{\Delta B_1} = x_{\Delta B_2} \quad (8.1)$$

where x defines the position. At any point on this boundary:

$$\sigma_1^{\alpha\beta} n^\alpha = \sigma_2^{\alpha\beta} n^\alpha \quad (8.2)$$

so the position of the two boundaries is the same and the stress in both bodies is identical at the boundary. In Lagrangian FE codes this condition is usually enforced in one of

three methods: The penalty method, where a restoring force is applied proportional to the error in (8.1). The Lagrange multiplier method which adds additional constraint equations to be solved. Finally, directly enforcing (8.1) and then adjusting the velocities and stresses accordingly.

Like the stress free boundary condition, contact has been successfully ignored in SPH. Contact between two bodies could be reasonably handled through the conservation equations. When a particle from one body is within the smoothing distance of a particle of a second body, they each become neighbour particles of the other. They then interact through the sums over neighbour particles. This method can produce good results when modelling hypervelocity impact, see for example Libersky et al. [34].

However, when using this method a degree of penetration and mixing occurs at the contact surface, typically the first one to two particle rows of each body pass each other. This occurs as SPH does not require the velocity field to be single valued. Monaghan proposed a modification to prevent this penetration, which he called XSPH [76]. The modification was intended to make particles that are close together move with nearly identical velocities. The modification is:

$$\frac{dx_i^\alpha}{dt} = v_i^\alpha + \sum_j (v_j^\alpha - v_i^\alpha) W_{ij} \quad (8.3)$$

where v_i^α is the particle velocity calculated from the momentum equation, equation 7.29.

While Monaghan's modification does prevent penetration it does nothing to solve two other problems with modelling contact in this manner:

- Tensile forces, resisting separation of two bodies, can be generated.
- Shear stresses are generated, preventing friction-less or low friction sliding.

Separate contact algorithms have been used with SPH when combining the SPH method with the FE method [61, 79], but the only purpose of the algorithms were to allow SPH nodes to interact with an FE mesh.

8.1 The SPH code

This section will briefly describe the SPH code used to test the contact algorithm developed in this chapter. The code was developed at Cranfield with the help of Dr Libersky.

The code uses the normalised forms of the conservation equations, the form given by equations 7.49–7.51. It uses conservative smoothing, see equation 7.41, to cure the tensile instability. Boundary particles are found, and the free surface condition is applied using the method described in section 7.3.3. The material model used is a simple elastic-plastic strength model, coupled with a Mie-Gruneisen equation of state. The elastic-plastic strength model is not strain-rate or temperature dependent.

With the re-normalised SPH method, according to Libersky and Randles [95], a reduced neighbour set should be used. In 1D this is equivalent to using only the nearest particles, whilst in higher dimensions a search is done on the whole neighbour set of all particles within $2h$, and the 'best' are chosen. A 1D example of the effect of

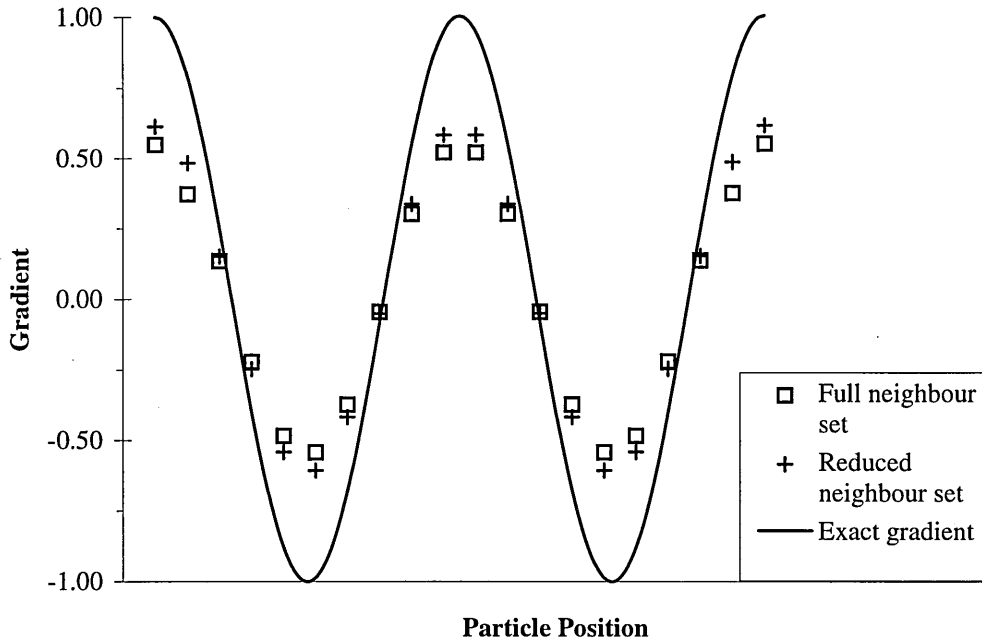


Figure 8.1: Comparison between normalised SPH approximations for a full neighbour set and a reduced neighbour set to the gradient of a function.

using a reduced neighbour set is shown in figure 8.1. The figure shows normalised SPH approximations to the gradient at the particles using a full neighbour set and a reduced neighbour set. The function is $\sin x$, scaled to produce two full wave-lengths, twenty particles are used and are spaced evenly. With the full neighbour set an h of 1.5 times the inter particle distance has been used, resulting in the interior particles having four neighbours. It can be seen that using the reduced neighbour set improves the approximation of the gradient. The error between the SPH approximation and the real value reduces as the number of particles increase. A non-linear function was chosen for this demonstration as both methods produce exact results for the gradient of a linear function.

The starting point for finding the reduced neighbour set in 2D is the full neighbour set returned by the linked list search. For each particle in the full set, the algorithm searches for a neighbour closer to the i particle and in a similar direction from the i particle. If one is not found then the particle is added to the reduced neighbour set.

The code can use one of two methods to find the full neighbour set. The first method is the standard linked list sort method. The second method is a neighbour of neighbours search, where each neighbour particle, and the neighbour particles of that particle, are checked to see if they lie within $2h$ of the i particle.

Both the B-spline kernel and the quadratic kernel are implemented, the analyses presented in this chapter all used the quadratic kernel. The choice of kernel is less important with the re-normalised method, as the kernel derivative appears in both the numerator and denominator of the re-normalised equations.

The critical time step calculated is:

$$\Delta t = k \left(\frac{|\mathbf{r}_{min}|}{c + |\mathbf{U}_{max}|} \right) \quad (8.4)$$

where $\mathbf{r}_{min}$ is the minimum inter particle spacing, $\mathbf{U}_{max}$ is the maximum relative velocity of approach, and k is a scale factor. In a stability analysis Randles and Libersky [82] found that with this time step and for the variable update order used, the maximum stable value of k is 2.0.

While the code does not contain any sophisticated material models, it contains all the most recent developments in the SPH field, but still does not treat contact correctly. Contact between two bodies is performed via the conservation equations, but the use of boundary particles allows for the bodies not to be in contact at the start of the calculation.

The initial particle positions of a test problem is shown in figure 8.2. The problem is modelling the normal impact of two steel blocks, the particles belonging to each block are shown with different levels of grey. Each block is 1.0 cm by 0.4 cm and consists of 50 particles by 20 particles, giving a total of 2000 particles in the whole model. The boundary particles are shown in figure 8.3, boundary particles have been detected around the body, but not along the contact surface.

This model was run with the two blocks initially having a relative velocity of 4.0 km/s. The z velocity, the z axis is vertical, at time $0.4 \mu s$ is shown in figure 8.4. At this point the shock waves generated by the impact can be clearly seen, and have not yet reached the top and bottom surfaces of the block. At either end of the contact surface an expansion wave has been generated, showing that the stress-free condition has been applied at the boundary particles. Figure 8.5 shows the z velocity at time $1.4 \mu s$. By this point the shock waves have reached the top and bottom surfaces, and the expansion waves have met at the contact surface. The z velocities at either side of the blocks are low due to release waves generated from the sides of the block. This analysis produces reasonable results until the point when the two release waves reach the contact surface. When this occurs the two blocks should begin to separate, but as kernel contact is being used this does not occur.

This problem is used as the test problem for the contact algorithm in 2D.

8.2 Contact algorithm

In this section the development of a contact algorithm for SPH will be described. The 2D form will be described, and its extension to 3D will be commented on.

First consider the attributes that an ideal contact algorithm would have.

- Should not be computationally expensive, as it is to be used in an explicit code.
- Extension from 2D to 3D should be simple, in keeping with the SPH method in general.
- Allow contact with separation
- Allow frictionless contact or contact with friction

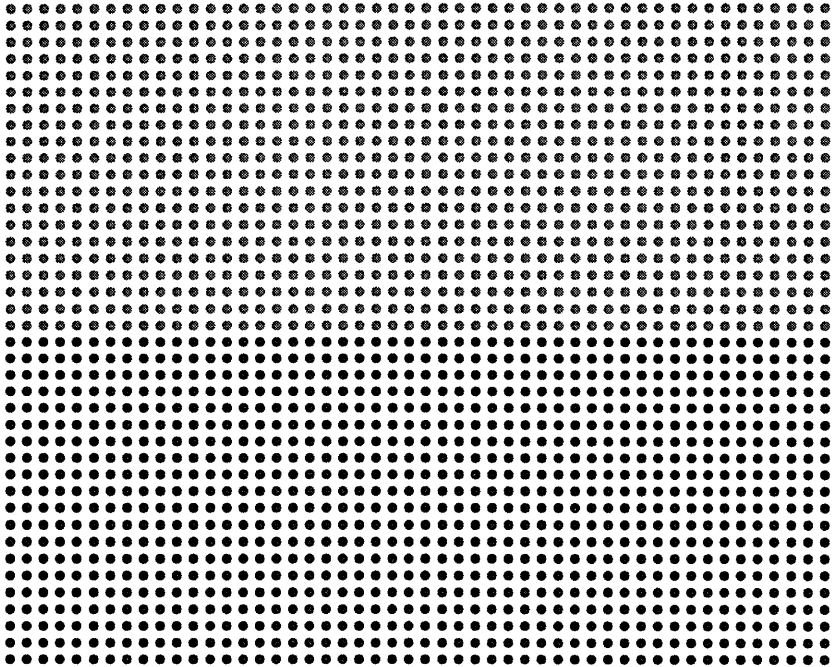


Figure 8.2: Particle materials for 2D test problem

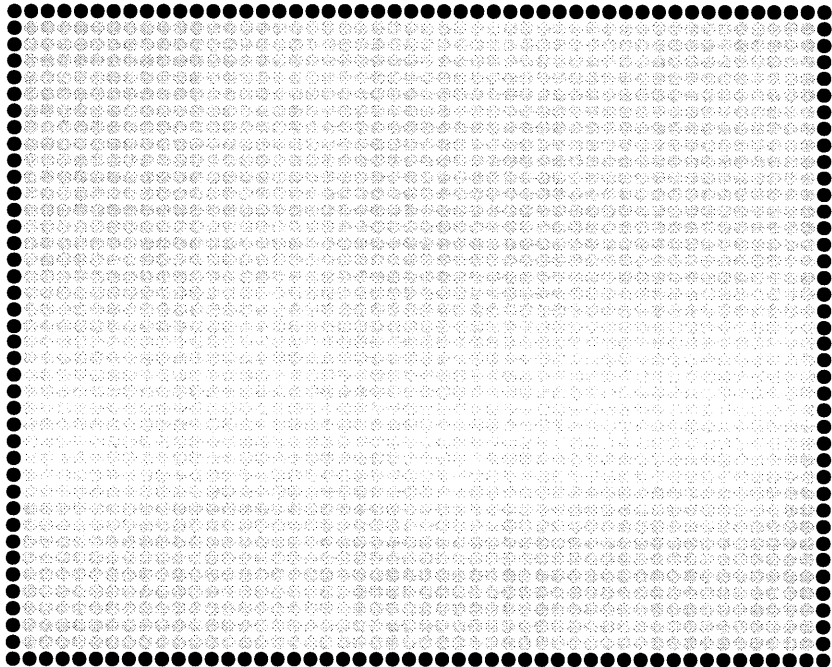


Figure 8.3: Boundary particles for kernel contact.

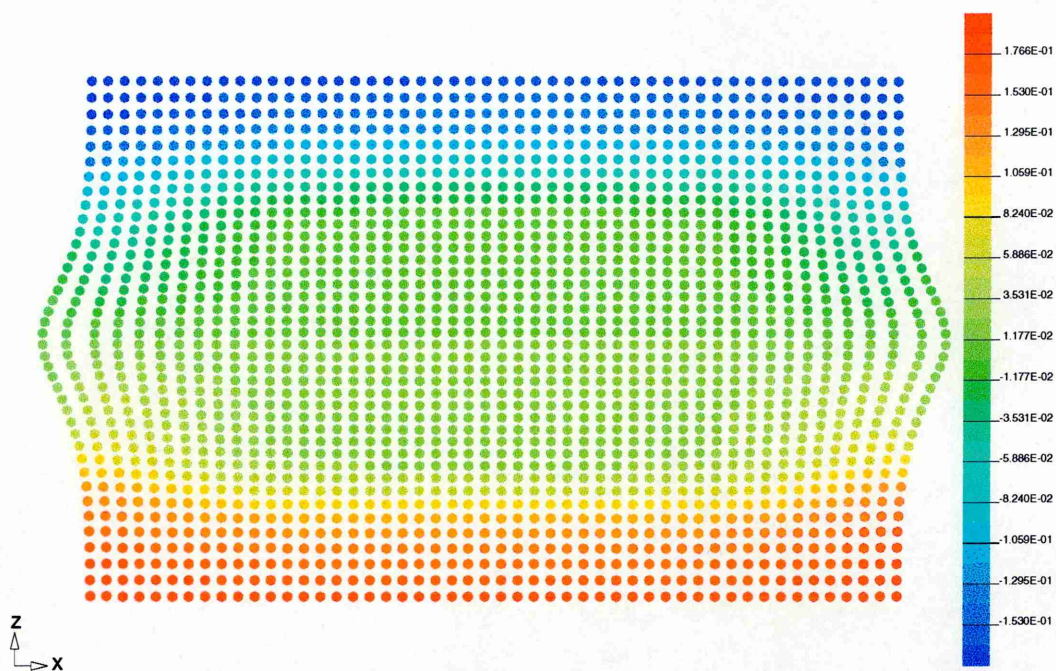


Figure 8.4: z velocity at $0.4 \mu s$. Kernel contact.

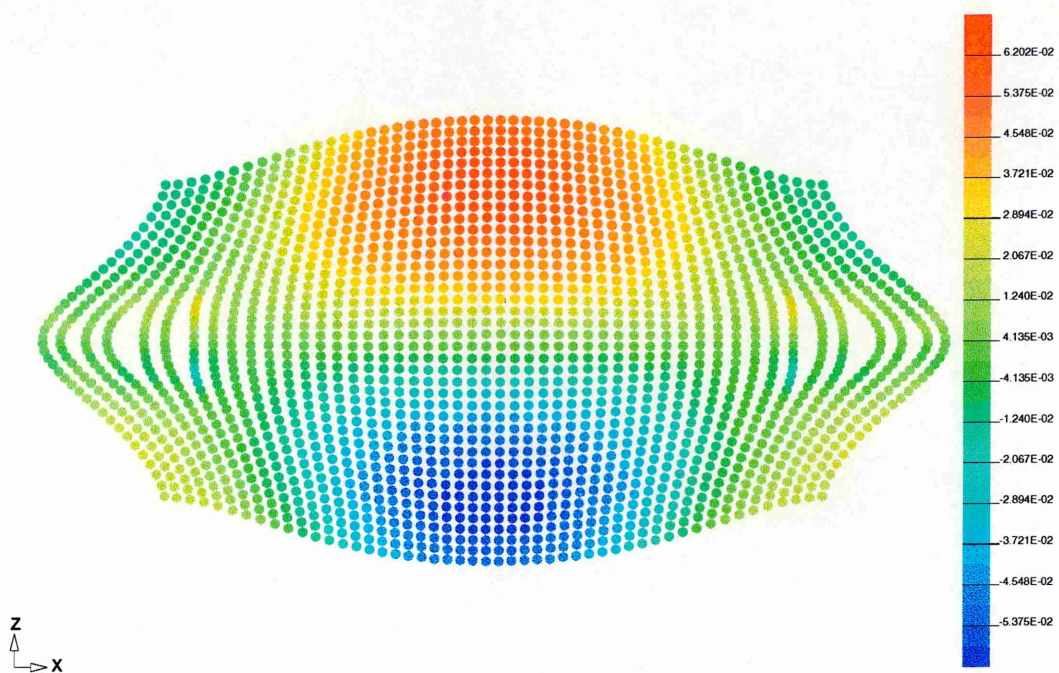


Figure 8.5: z velocity at $1.4 \mu s$. Kernel contact.

The first question to consider is the definition of the boundary. As the kernel estimate decays to zero over a distance of $2h$, the SPH method does not provide a sharp boundary. As the code already contains an arbitrary choice of the location of the boundary, that it lies $h/2$ from a boundary particle in the direction of the surface normal, this distance will be used for this contact algorithm.

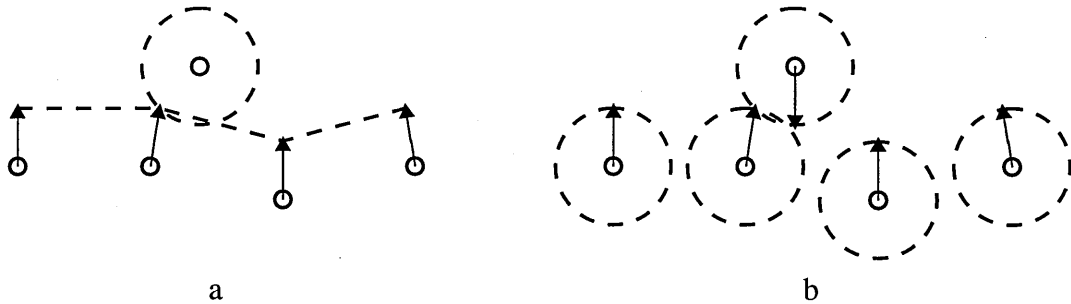


Figure 8.6: a: Particle to surface contact. b: Particle to particle contact.

Having defined points that lie on the boundary, the next point to consider is how contact should be detected, and how the resulting contact force should be distributed. There are two principal choices: particle to surface contact, and particle to particle contact. Figure 8.6 shows an example of both cases. The arrows denote the normal vector to the particle calculated by the gradient of unity method, the dashed lines represent the surfaces used to determine if there is contact. In the particle to surface case a straight line is drawn between points that lie $h/2$ along the particle normals, if this line intersects the circle with radius $h/2$ from the oncoming particle then contact has occurred. The normal to the surface at the point of contact is defined by the normal to the line, and a restoring force is applied in the same direction as the surface normal. The force would have to be split between the two particles at either end of the contact line. In concept this is very similar to finite element contact algorithms, with a slave node impacting a master surface. The difference is that the nodes do not lie on the boundary, complicating the geometry. The finite element contact algorithm is more reliable when it is made symmetric, which means that each node is considered as a slave node, and a node on the master surface. It is thus probable that similar benefits would be seen in an SPH form. In the particle to particle form contact occurs when the circles around two particles intersect. This can be easily determined by checking the inter-particle distances, making the detection of contact considerably simpler than with the particle to surface version. Using this method, only the interaction of two particles is considered at any one time, removing the need to consider each particle twice in the algorithm. This approach has many similarities with Belytschko's pinball contact algorithm [98, 99]. Which is a contact algorithm for the finite element method that treats contact by embedding spheres within the elements, and enforcing contact on these spheres, rather than at the nodes.

The particle to particle approach was selected as using only particle to particle interaction is more in keeping with the meshless approach, and its extension to 3D is not complex. Extending the particle to surface algorithm to 3D is complicated by the need to define a boundary surface. It must be possible to redefine this surface regularly,

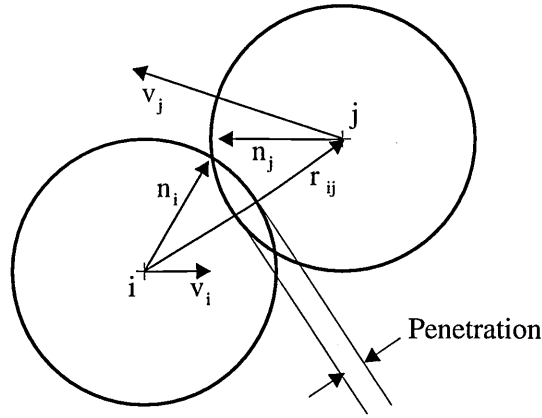


Figure 8.7: Interpenetration of two particles, restoring force applied between particle centres.

as using a boundary surface fixed to the initial boundary particles would sacrifice the principal advantage of the meshless method. In 2D it is not a major problem to determine neighbour boundary particles, in 3D it would probably be necessary to construct the surface from triangular segments, and the selection of the best nodes to make up the vertices of the segments adds considerable complexity to the algorithm. This problem could be solved by the adaptation of a mesh generation algorithm from the Free Lagrange method.

The choice of the particle to particle method presents the problem of determining the direction of the restoring force. The restoring force should be applied in a direction normal to the contact surface, however except in special cases, this will result in a force vector that does not pass through the centre of mass of both particles, generating a moment. Applying the restoring force along the vector connecting the particle centres solves this problem, but the restoring force will now have a component tangential to the contact surface. Algorithms for both cases were developed, and will now be described.

8.2.1 Restoring force applied between particle centres.

This is the simple case, penetration is detected if:

$$\frac{(h_i + h_j)}{2} - |\mathbf{r}_{ij}| \leq 0 \quad (8.5)$$

where

$$\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i \quad (8.6)$$

The actual penetration is the magnitude of equation 8.5. The restoring force is applied along vector $\mathbf{r}_{ij}$. This case is shown in figure 8.7.

8.2.2 Restoring force applied along average normal

In this case the restoring force is to be applied normal to the contact surface. This normal needs to be found without actually calculating the surface. For the stress-free

condition the surface normal at each boundary particle has been found, but it is not possible to assume that the two normals are equal and opposite, nor is it simple to design an algorithm to pick the normal that best represents the contact surface. It would be simplest to treat this problem by first applying a force along one normal, than along the other, but the most probable result from this is a force that is near parallel with the contact surface. Taking the difference between the two normal vectors results in a vector that is the average of the two vectors. Figure 8.8 shows these two cases for the particle normal shown in figure 8.7.

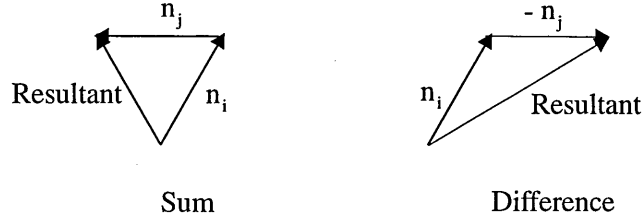
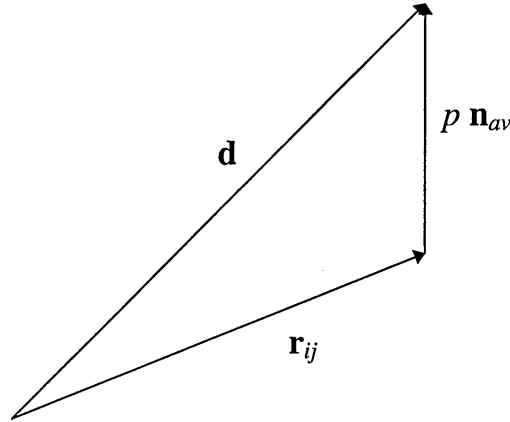


Figure 8.8: Vector diagrams showing the result of adding and subtracting the particle normal vectors.

This average normal vector is the vector along which the restoring force is applied. So the magnitude of the penetration can be defined as the relative displacement of the particle positions in the direction of the average normal that would result in zero penetration. This is the definition of penetration that is used in the pinball algorithm. The penetration, p , can be calculated using the following method:



$\mathbf{r}_{ij}$ is the vector from the centre of particle i to the centre of particle j .

$\mathbf{d}$ is the vector from the centre of particle i to the position of the centre of particle j that would give zero penetration. The vector is not known but:

$$|\mathbf{d}| = \frac{(h_i + h_j)}{2} \quad (8.7)$$

The unit average normal is:

$$\mathbf{n}_{av} = \frac{(\mathbf{n}_j - \mathbf{n}_i)}{|\mathbf{n}_j - \mathbf{n}_i|} \quad (8.8)$$

From these the penetration can be found:

$$p = \sqrt{|\mathbf{d}|^2 - |\mathbf{r}_{ij}|^2 + (\mathbf{n}_{av} \cdot \mathbf{r}_{ij})^2} - (\mathbf{n}_{av} \cdot \mathbf{r}_{ij}) \quad (8.9)$$

8.3 Enforcing the contact condition

Once contact is detected a method is required to enforce the contact condition. The penalty method was chosen to enforce contact as the simplest and most straight forward method. It does not add any additional equations to solve, it does not require the use of contact and release conditions, and has proved a reliable method for enforcing contact within finite element codes.

A consequence of allowing a degree of penetration to occur is the risk of over penetration. Penetration is detected only when the distance between the two particles is $0.5(h_i + h_j)$. If the two particles completely pass each other then penetration will no longer be detected, and no restoring force will be calculated. This will place a lower limit on the magnitude of the penalty force.

Three different expressions for the penalty force were investigated, and implemented into the code. This allowed the sensitivity of the method to the contact force calculation used to be tested.

The first expression was used by Belytschko in the pinball algorithm [99]. This defines the penalty force as:

$$F = K_p \min(F_1, F_2) \quad (8.10)$$

F_1 and F_2 are defined as:

$$F_1 = \begin{cases} \frac{\rho_i \rho_j R_i^3 R_j^3}{\rho_i R_i^3 + \rho_j R_j^3} \frac{\dot{p}}{\Delta t} & \dot{p} > 0 \\ 0 & \dot{p} < 0 \end{cases} \quad (8.11)$$

$$F_2 = \left[\frac{G_i G_j}{G_i + G_j} \sqrt{\frac{R_i R_j}{R_i + R_j}} \right] p^{3/2} \quad (8.12)$$

where ρ_i and ρ_j are the densities of the two particles, G_i and G_j are the shear moduli, R_i and R_j are the radii. K_p is a scale factor. Equation 8.11 is derived by considering the force required to bring the two particles to rest with respect to each other, while conserving momentum, an ideal plastic collision. The contact force in one time step applies the same impulse to both contact particles:

$$F \Delta t = m_i \Delta v_i \quad F \Delta t = m_j \Delta v_j \quad (8.13)$$

If the particles are to be brought to rest with respect to each other, then the rate of change of penetration, $\dot{p}$ is $\Delta v_i + \Delta v_j$. Then $\dot{p}$ is

$$\dot{p} = \frac{F_1 \Delta t}{m_i} + \frac{F_1 \Delta t}{m_j} \quad (8.14)$$

this can be rearranged to give 8.11 by substituting $m = \rho \frac{4}{3}\pi R^3$ and ignoring the constant $\frac{4}{3}\pi$.

Equation 8.12 is derived from the equations describing contact between two solid elastic spheres [100]. The equation for the penetration is:

$$p = 1.040 \sqrt[3]{\frac{F^2 C_E^2}{K_D}} \quad (8.15)$$

where

$$C_E = \left(\frac{1 - \nu_i^2}{E_i} \right) + \left(\frac{1 - \nu_j^2}{E_j} \right) \quad \text{and} \quad K_D = \frac{D_i D_j}{D_i + D_j}$$

The second penalty expression used was:

$$F = K_p \begin{cases} \max(F_1, F_2) & \dot{p} > 0 \\ 0 & \dot{p} < 0 \end{cases} \quad (8.16)$$

K_p is a scale factor. F_1 and F_2 are defined as:

$$F_1 = \frac{mp}{\Delta t^2} \quad (8.17)$$

$$F_2 = \frac{2mcp}{\Delta t \Delta x} \quad (8.18)$$

where $\Delta x = |\mathbf{x}_j - \mathbf{x}_i|$.

Equation 8.17 is used by Swegle [35] for contact between SPH nodes and finite elements. It is derived by assuming that the rate of penetration can be defined as $\frac{p}{\Delta t}$ and applying a force sufficient to bring the penetrating node to rest over a single time step.

Equation 8.18 can be derived by assuming that $m_i = m_j = m$ and applying a force sufficient to bring to rest a body of mass $2m$ travelling at the relative velocity of the 2 particles, v' :

$$F = \frac{2mv'}{\Delta t} \quad (8.19)$$

then substituting $v' = \frac{cp}{\Delta x}$, where $\frac{\Delta x}{c}$ is an approximation to the time over which the penetration has occurred, gives (8.18).

The equation for the stress behind a one dimensional elastic wave:

$$\sigma = \rho cv \quad (8.20)$$

is equivalent to (8.18) if the following substitutions are made: $\sigma = \frac{F}{A}$ and $v = \frac{p}{\Delta t}$. The third penalty expression was:

$$F = \begin{cases} \frac{\frac{|\mathbf{v}_j - \mathbf{v}_i|}{\Delta t}}{\frac{1}{m_i} + \frac{1}{m_j}} & \dot{p} > 0 \\ 0 & \dot{p} < 0 \end{cases} \quad (8.21)$$



Figure 8.9: Boundary particles for penalty contact.

This is derived from (8.14), by substituting the relative velocity $|\mathbf{v}_j - \mathbf{v}_i|$ for $\dot{p}$. No scale factor or other term is used

The two particles in contact must not have the stress free condition applied. So both particles must be flagged in the code as in contact. The stress free condition will only be applied to boundary particles that are not flagged as being in contact.

8.4 Implementation of the contact algorithm

This section will describe how the contact algorithm was implemented into the SPH code.

Implementing the contact algorithm required changes to the following sections: the search algorithm, the application of the stress free boundary conditions and the calculation of velocity. It required a new section to detect contact and calculate the opposing force.

With kernel contact, once the linked list sort has occurred, all particles within $2h$ of the particle being checked are added to its neighbour list. With contact this is modified so a particle is only added to the neighbour list if it is of the same body. In the same loop an additional check is made to detect potential contact particles. If the particle being checked is a boundary particle, and a particle of a different body is within $2h$, then this particle is added to a list of potential contact particles. This list is effectively a second neighbour list containing only particles from different bodies. It would be possible to do the actual contact check at this point and list only those particles that are in contact. Potential contact particles are detected as some contact algorithms look ahead to prevent penetration occurring in the next time step, and this allows such an algorithm to be implemented without requiring the existing contact routines to be rewritten. The

neighbours of neighbours search algorithm cannot be used for this contact search, as it only checks particles already in the neighbour lists and so would not detect new contact.

In a finite element code the contact can be a significant proportion of the processor time required for each time step. The principal cause of this cost is the search routine that is required to detect contact. As the SPH method already has a search routine as part of each time step, adding a contact algorithm need not represent a significant increase in the computer time needed to solve a problem.

As only particles from the same body are included in the neighbour lists, the boundary particle detection routine will now detect boundary particles along the contact surface without any modification, figure 8.9.

The contact subroutine is added next. For each boundary particle with one or more listed potential contact particles, it checks whether contact has occurred. If contact has occurred it flags both particles involved, and calculates the resulting penalty force.

The stress free boundary condition section is modified so that particles that have been flagged as in contact by the contact routine do not have the boundary condition applied. Finally the particle velocity calculation section must be modified to add the contribution from the contact force to the particles velocity.

8.5 Contact with two particle normals

The contact algorithm was tested in 2D using the same test problem presented earlier. Experience quickly showed that the contact algorithm required improvement in its treatment of corner particles. As implemented, every boundary particle in contact is flagged so that the stress free boundary condition is not applied, this caused incorrect behaviour at the corner particles at either end of the contact surface. Figure 8.10 shows the result of not applying the stress free condition at the corner particles when using the contact algorithm. As the stress free condition is not applied release waves are not generated correctly at the corner particle. The result is that the horizontal velocity of the corner particles is too low. The simplest alternative was not to flag the corner contact particles, so these particles would have a contact force applied but would also have the stress free condition applied. Figure 8.11 shows the results of applying the stress free condition at the corner particles in contact. The overall behaviour appears improved, however the corner particles themselves have significantly interpenetrated, unlike the other particles along the contact surface.

These two tests showed that the treatment of corner particles needs to be improved, as neither case is correct. The case with the stress free condition applied shows improved behaviour, but caused local problems. In order to treat the corner particles correctly two surface normal vectors are required, see figure 8.12. If the corner particle is not in contact then the stress free condition is applied for both particle normals. If the particle is in contact then the stress free condition is only applied for the particle normal that is not towards the other contact particle. This will allow the correct behaviour to be seen, while not generating accelerations normal to the contact surface.

The SPH code was modified to include this improvement. In an implementation of the two normal approach, it would unnecessarily complicate the bookkeeping section of the code to have some boundary particles with one normal vector, and to have some particles with two normal vectors. Instead, the code was modified to assign two normal

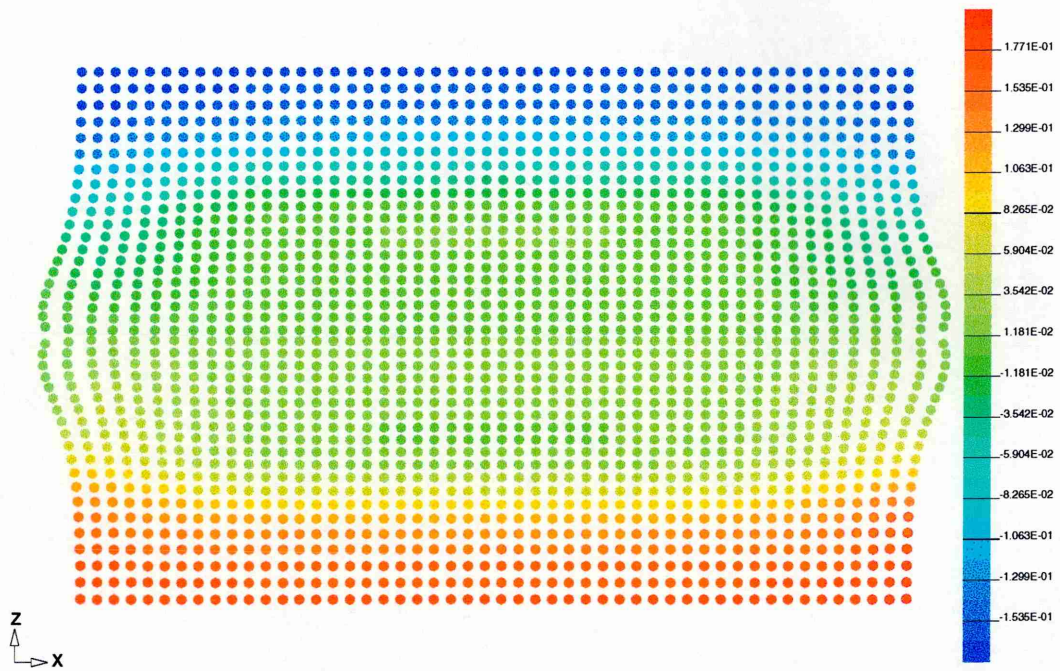


Figure 8.10: z velocity at $0.4 \mu s$. Corner particles do not have stress free condition applied.

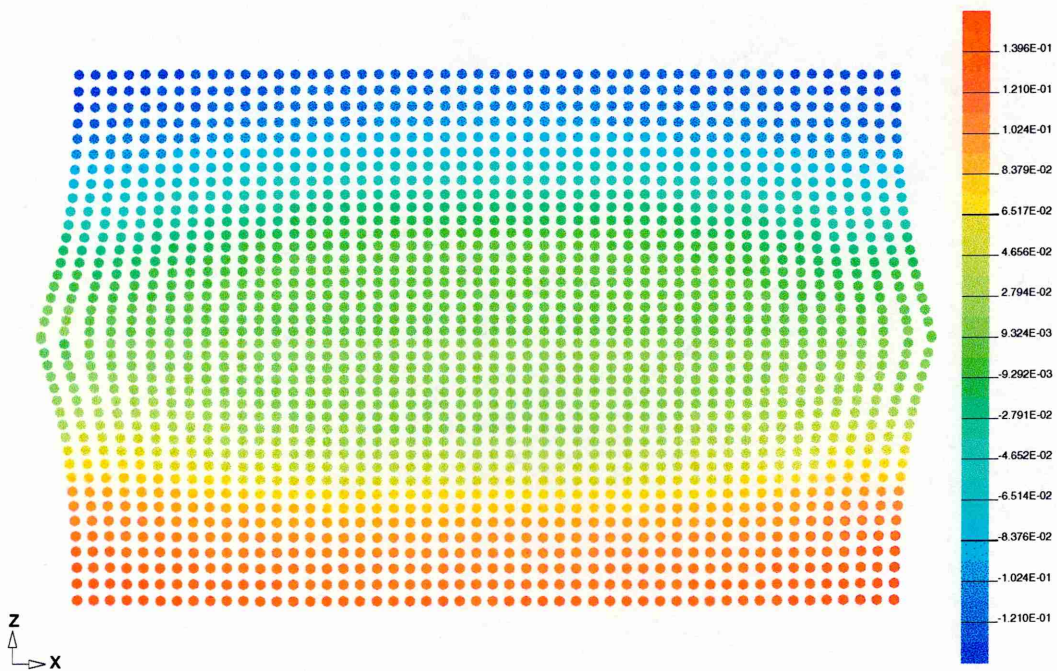


Figure 8.11: z velocity at $1.4 \mu s$. Corner particles have stress free condition applied.

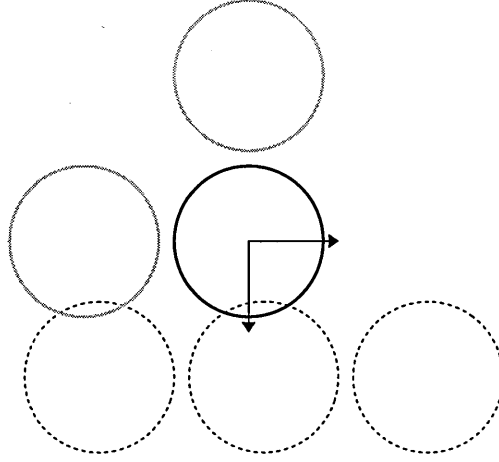


Figure 8.12: Corner particle with two normals.

vectors to every boundary particle.

To calculate the two vectors in 2D, the two neighbour boundary particles are identified for each boundary particle. Then the unit normal to the vector between the centre of the boundary particle and each of the neighbours is calculated, and these become the two normal vectors for that particle. So a corner particle will have two distinct normal vectors, while a boundary particle on a flat surface will have two coincident normal vectors.

In the contact subroutine it is necessary to find the best normal to use. The best normal is defined as the normal which gives the maximum value of:

$$\mathbf{n}_1 \cdot \mathbf{r}_{ij} \quad (8.22)$$

where $\mathbf{r}_{ij}$ is the vector between the centres of the two particles. The best normal is then used for the contact algorithm as if it were the single particle normal. The best normal is flagged, so that the stress free condition is not applied for that surface normal. For contact boundary particles that lie on a flat or near flat boundary it is necessary to flag the other normal as well, so that the stress free boundary is not applied at all to the particle. A test is necessary to differentiate between a corner particle and a surface particle. The angle between the two normal vectors is calculated, if this exceeds a set value then the particle is considered to be a corner particle and only the best normal is flagged. The set value implemented into the code was 45° .

8.6 Tests of contact algorithm

8.6.1 1D tests

The first set of tests were in 1D. The aim of these was to test the effects of the different penalty force algorithms, in exclusion from any effect that could occur in 2D from the two different methods of calculation of the vector along which the restoring force is applied.

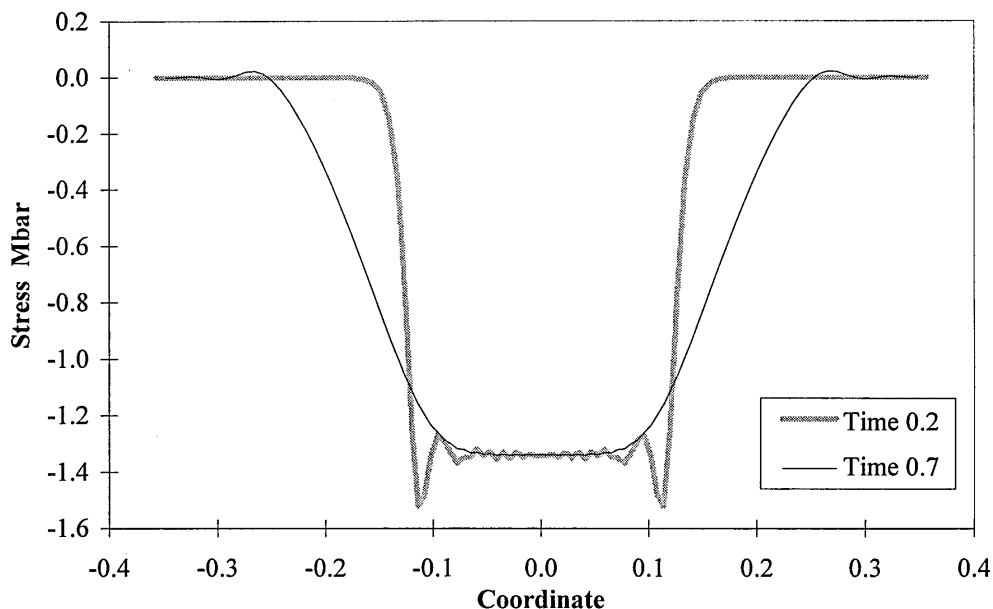


Figure 8.13: Stress profiles at $0.2\mu\text{s}$ and $0.7\mu\text{s}$ for kernel contact. $k = 0.5$.

The test problem used is the 1D equivalent of the 2D problem already presented, a 0.4 cm block of steel impacts an identical block. 100 particles were placed along the 1D line, 50 particles in each block. The smoothing length is 0.008 cm and the initial inter-particle space is equal to the smoothing length. The space between the two contact particles is 0.01, so there is no initial penetration. Each block is given an equal but opposite initial velocity of 2 km/s.

To provide a reference point the problem was first run with no contact, using kernel sums to allow interaction between the two bodies. With the kernel sums the two end particles are initially in contact as the distance between the two is $1.25h$. Figure 8.13 shows the resulting stress profiles. At time $0.2\mu\text{s}$ the shock waves are propagating away from the point of contact, at time $0.7\mu\text{s}$ the shock waves have reached the free ends, and have been reflected as release waves. The value of k used for the results shown was 0.5. The problem was also run with a value of 1.5 with no problems, the only difference observed was a higher amplitude, by a factor of 3–4, in the oscillations behind the shock wave.

With this reference point established, runs were made to compare the different penalty force equations. Figure 8.14 shows the stress profiles at $0.2\mu\text{s}$ for penalty force types one, equation 8.10, and two, equation 8.16. The two forms that use a scale factor, K_p , to scale the contact force. The scale factors used were the lowest values that did not result in $p > 2h$. The values were $K_p = 80,000$ for type 1, and $K_p = 0.008$ for type 2. Considerable oscillation is seen in the stress for both cases, and equivalent oscillation is also seen in the velocity profile. Figure 8.15 shows the stress profile for higher values of k , $K_p = 480,000$ for type 1 and $K_p = 0.05$ for type 2. Considerably improved behaviour is seen for both penalty forces, with the amplitude of the oscillations significantly reduced.

What is also seen in these two figures, especially with the higher penalty forces, is a

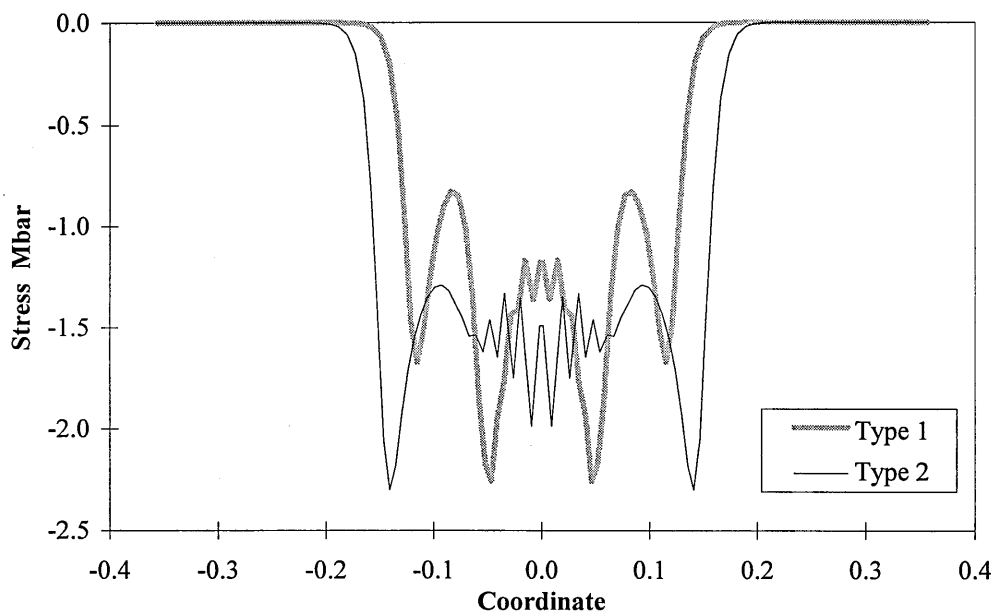


Figure 8.14: Stress profiles at $0.2\mu\text{s}$ for penalty force types 1 and 2, using the lowest scale factor that did not give over penetration. $k = 0.5$.

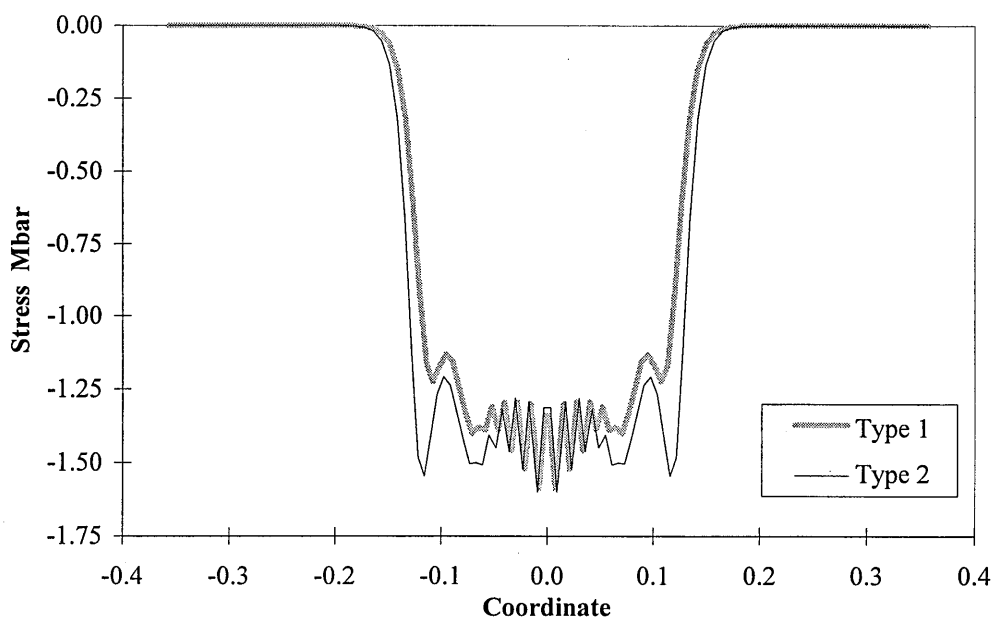


Figure 8.15: Stress profiles at $0.2\mu\text{s}$ for penalty force types 1 and 2, using a higher scale factor. $k = 0.5$.

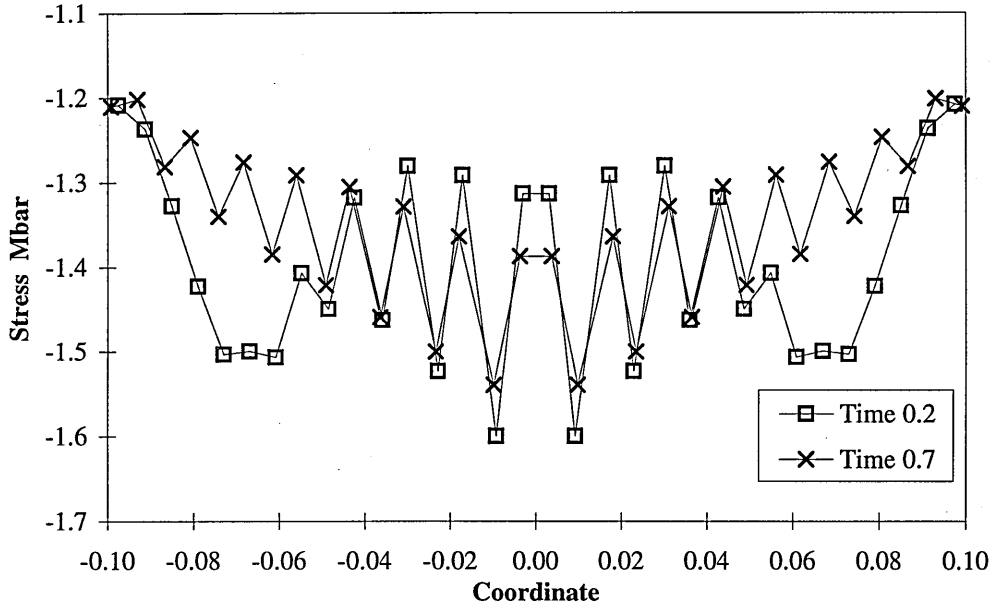


Figure 8.16: Detail of stress oscillation at contact surface, penalty type 2.

spurious stress oscillation at the interface. This oscillation is described by Swegle [35] who observed it when modelling the 1D impact of two initially separated bodies using kernel contact. This oscillation is a zero energy mode, as this stress field does not cause particle accelerations that would reduce the amplitude. Swegle shows that this is a characteristic of the SPH method, as the SPH estimate to the stress field has its local maxima and minima at the particles giving a gradient of zero. In the SPH method no acceleration will be produced at a particle if the stress at the two neighbour particles is equal, this is independent of the stress at the particle itself. This is the stress version of the SPH zero-energy mode where an alternating velocity field does not generate stresses in the material to resist the particle motion. Swegle attempted to control the oscillation by varying the time step, the artificial viscosity coefficient and the smoothing length, all without success.

The stress oscillation is the region of the contact does not change significantly during contact. Figure 8.16 compares the stress profile near the contact surface between time $0.2\mu\text{s}$ and $0.7\mu\text{s}$. This shows that only a small decrease in the amplitude has occurred.

Both the type 1 and the type 2 penalty forces exhibit similar overall behaviour. This has been seen in the stress profiles shown, and in figure 8.17. This shows a graph of the penetration between the two contact particles against time, for the same two analyses shown in figure 8.15. Both profiles show high initial changes in the penetration, before achieving a slow but steady decrease in penetration. Both lines are uneven in this phase, which is due to oscillation in the contact force. This force variation was the result of both penalty force algorithms returning a zero contact force if the two particles are separating, which is checked for by $\dot{p} < 0$. The result is a force that varies significantly from time step to time step.

For penalty force type 3, tests showed a steady increase in penetration, which should not occur. Figure 8.18 shows the velocity profile at $0.2\mu\text{s}$, the relative velocity between

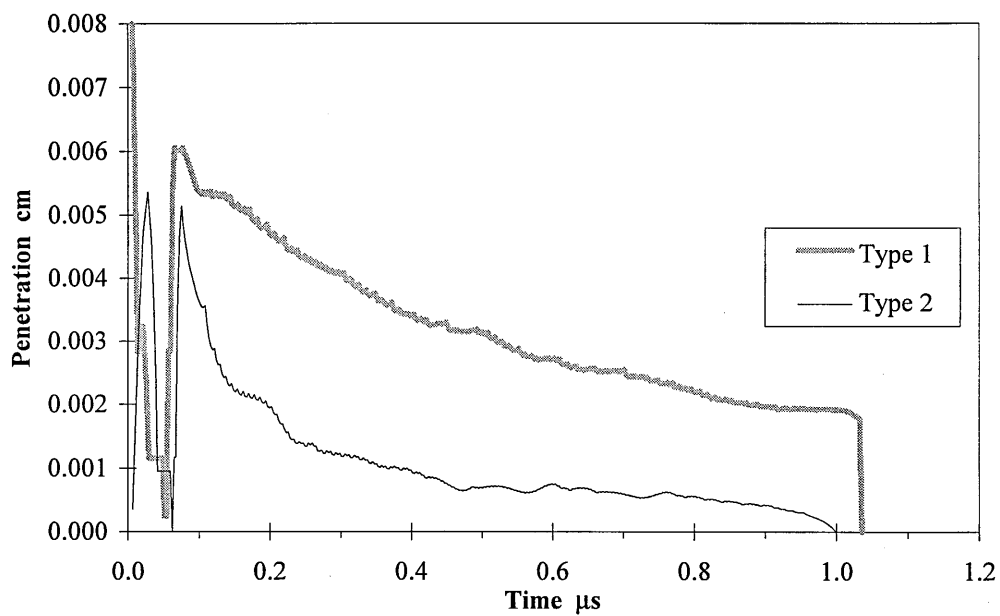


Figure 8.17: Comparison of penetration time histories for penalty types 1 and 2.

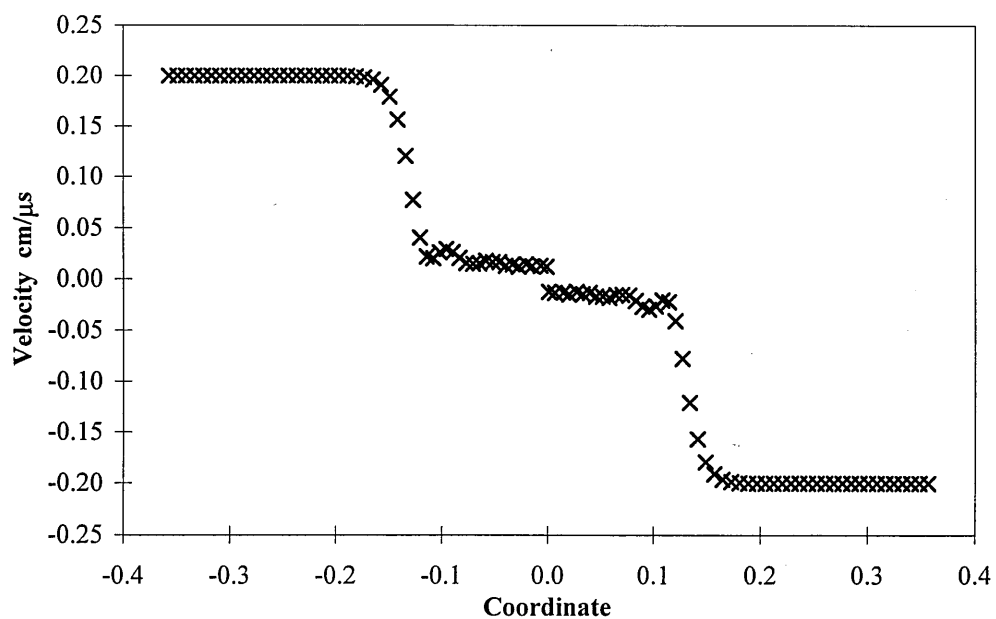


Figure 8.18: Velocity profile at $0.2\mu s$ for penalty type 3. $k = 0.5$

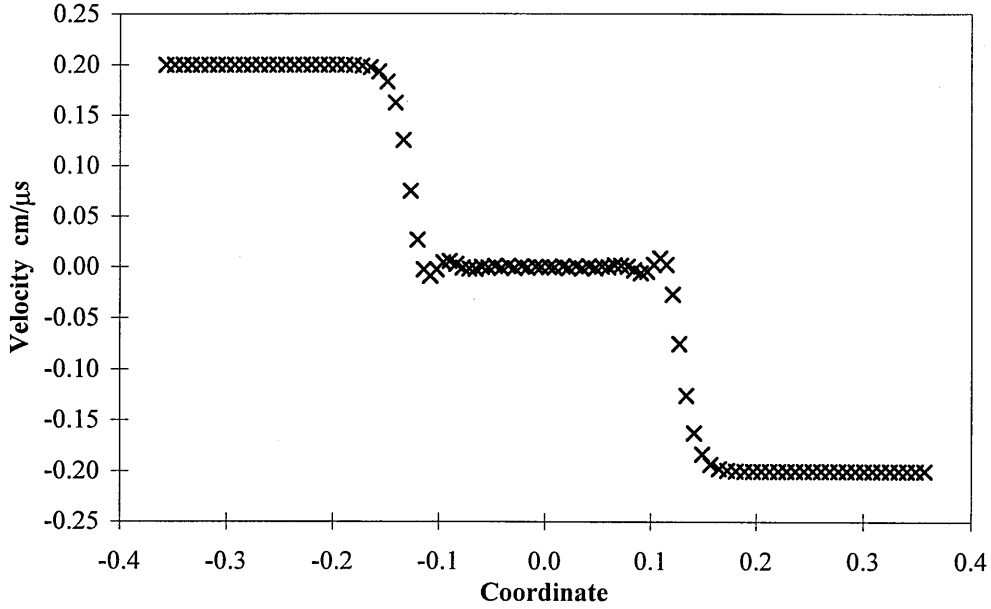


Figure 8.19: Velocity profile at $0.2\mu s$ for modified penalty type 3. $k = 0.5$

the two contact particle is $0.024 \text{ cm}/\mu s$. The cause of this velocity difference was the conservative smoothing algorithm. The acceleration due to the contact force is calculated in the same routine as the momentum equation is evaluated, the routine returns the new particle velocity. Before the particle positions are updated the conservative smoothing algorithm is called. For the boundary particles, which only have one neighbour, the conservative smoothing formula in 1D is:

$$v_i = v_i + \alpha_{cs}(v_j - v_i) \quad (8.23)$$

this is a simplified form of equation 7.41. For all the results presented in this chapter the value of the conservative smoothing scale factor, α_{cs} is 0.5. This acts to smooth local velocity oscillations, which the contact force generates.

In order to properly evaluate penalty force type, the code was modified to calculate the contact force after the conservative smoothing algorithm, but before the update of the particle positions. This modification was made only for penalty force type 3, as types 1 and 2 do not require this modification in order to work correctly. For the correct treatment of the stress free condition, contact is still detected before the stress free condition is applied, only the contact force and the new boundary particle velocities are calculated after the conservative smoothing algorithm. Figure 8.19 shows the velocity profile at $0.2\mu s$, which shows the expected behaviour. The penetration time history showed a constant value of the penetration, 7.46×10^{-4} , from time step 15 to the point when the leading edge of the release wave arrives at the interface. This modified version of contact, using the third type of penalty force, shows the best agreement with the velocity profile seen with kernel contact. However the stress profile shows that the stress oscillation is also present with this contact force type.

If the spurious stress oscillation did not have any affect on the results of the calculation then it would be possible to ignore it. During the early stages of the calculations,

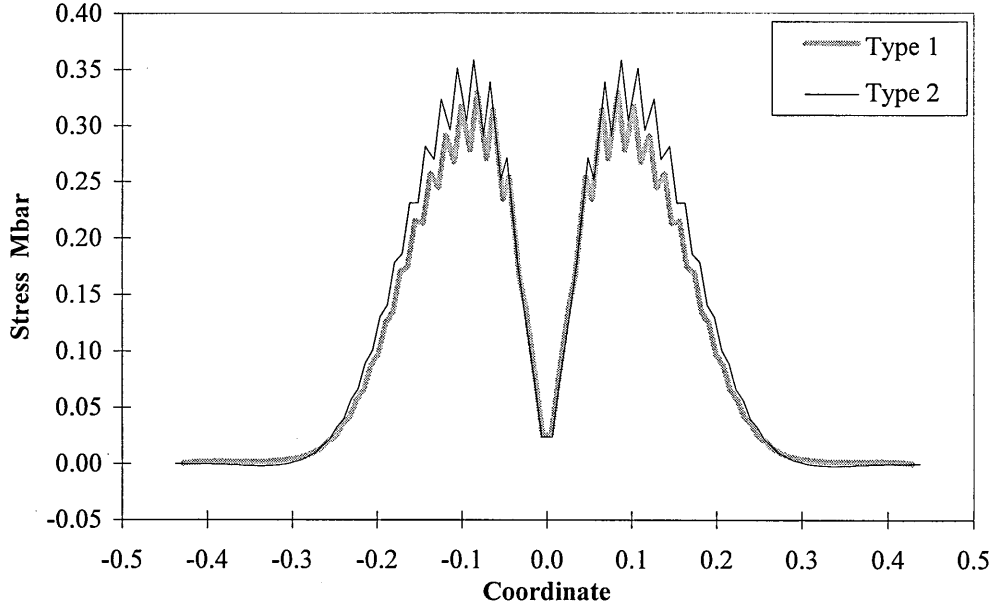


Figure 8.20: Stress profile at $1.1\mu\text{s}$ for penalty types 1 and 2. $k = 0.5$.

as have been discussed so far, it is not significant. It was found that it can have a significant effect once the release waves arrive at the contact surface. The spurious stress oscillation caused high tensile stresses in the material when the release waves reach the contact interface by affecting the behaviour of the nodes on the contact surface. Figure 8.20 shows the stress profile at time $1.1\mu\text{s}$ for penalty types 1 and 2, there is a high tensile stress near the contact interface, and the stress at the two contact particles is tensile. This effect is seen most clearly when using penalty type 3, figure 8.21, where the stress at the two contact particles is $0.17\text{ g / cm } \mu\text{s}^2$. The root cause of this tensile stress is the incorrect behaviour of the boundary particles due to the spurious stress oscillations. This can be explained by considering the momentum equation.

In 1D with a reduced neighbour set the momentum equation (7.50) for an internal particle reduces to:

$$\frac{dv_i}{dt} = \left[- \left(\frac{m_{j1}}{\rho_{j1}} (\sigma_{j1} - \sigma_i) \frac{dW_{ij1}}{dx} \right) - \left(\frac{m_{j2}}{\rho_{j2}} (\sigma_{j2} - \sigma_i) \frac{dW_{ij2}}{dx} \right) \right] B \quad (8.24)$$

where $j1$ and $j2$ are the two neighbour particles. So when $\sigma_{j1} = \sigma_{j2}$ the acceleration is zero. For a boundary this equation is:

$$\frac{dv_i}{dt} = \left[- \frac{m_j}{\rho_j} (\sigma_j - \sigma_i) \frac{dW_{ij}}{dx} \right] B \quad (8.25)$$

as the stress oscillation causes $\sigma_j \neq \sigma_i$ then the boundary particle has a non-zero acceleration. In all cases the stress oscillation caused by contact had $\sigma_i > \sigma_j$, which causes the boundary particle to have an acceleration away from its neighbour particle. This is clearly seen in figure 8.21, where although all the particles close to the contact interface have a tensile stress the contact particles have an acceleration that keeps them

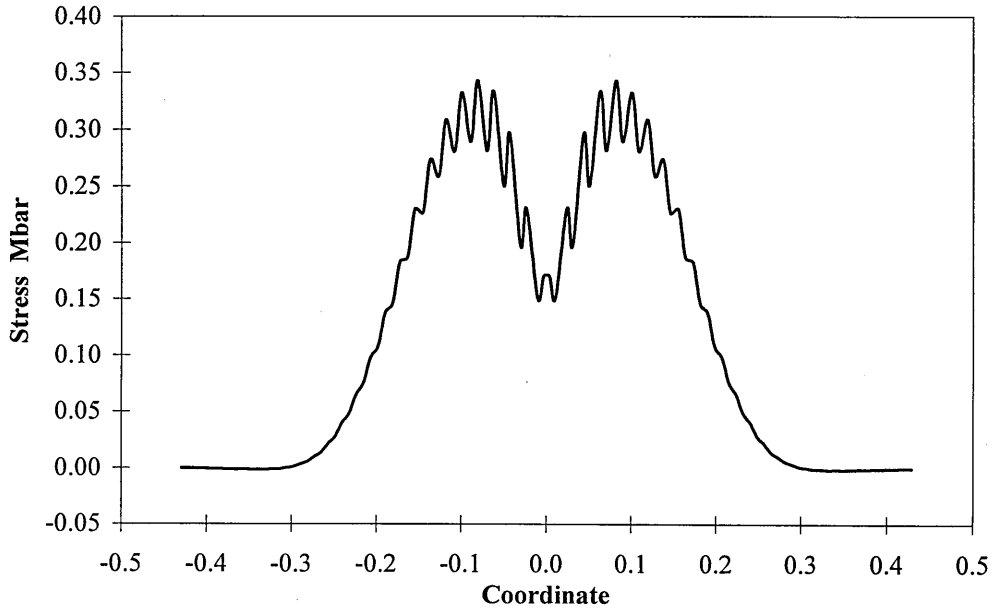


Figure 8.21: Stress profile at $1.1\mu\text{s}$ for penalty type 3. $k = 0.5$.

in contact as the stress at the contact particles is greater than the stress at their neighbour particles.

Figure 8.22 compares the velocity and stress time histories for penalty type 3 and kernel contact for particle 51, one of the two contact particles. It also compares the stress time history of particle 51 with particle 52, its neighbour particle, for penalty type 3. The initial velocity of node 51 is $-0.2\text{ cm}/\mu\text{s}$. The time histories for penalty type 3 are used as the effect is most pronounced in them, the same basic behaviour is seen with all of the penalty force types. Comparing the contact force case with the kernel contact case, the early behaviour is similar. It differs from a time of approximately $0.8\mu\text{s}$, when the leading edge of the release wave reaches the contact surface. In the kernel contact case, where there is no stress oscillation at the contact interface, particle 51 gains a positive velocity, which increases the distance between the contact particles. With the penalty contact the particle still has a zero velocity, it has a negative acceleration due to the stress gradient but the contact force prevents the penetration increasing. The lowest graph in figure 8.22 shows the stress time histories for particles 51 and 52. Particle 51 does not gain a positive velocity until approximately $1.2\mu\text{s}$, the time when $\sigma_{52} > \sigma_{51}$.

In most cases the effect that the tensile stress had on the final result was to delay the separation of the two bars, and to generate a tensile stress wave that propagated through the material. In one case the stress was high enough to cause numerical fracture, splitting the particles near the contact point from the remainder, figure 8.23. The velocity variation in the remaining material is due to the tensile stress wave.

The effect of the spurious stress oscillation has been established. As it has significant effect on the results it needs to be controlled. Tests were performed to find the effect of varying the time step, contact force and initial material velocity on the oscillation.

Figure 8.24 compares the stress profile at $0.2\mu\text{s}$ for a k of 0.5 and 0.1. Both use penalty type 2 with $K_p = 0.05$. Reducing the time step has only reduced the extent

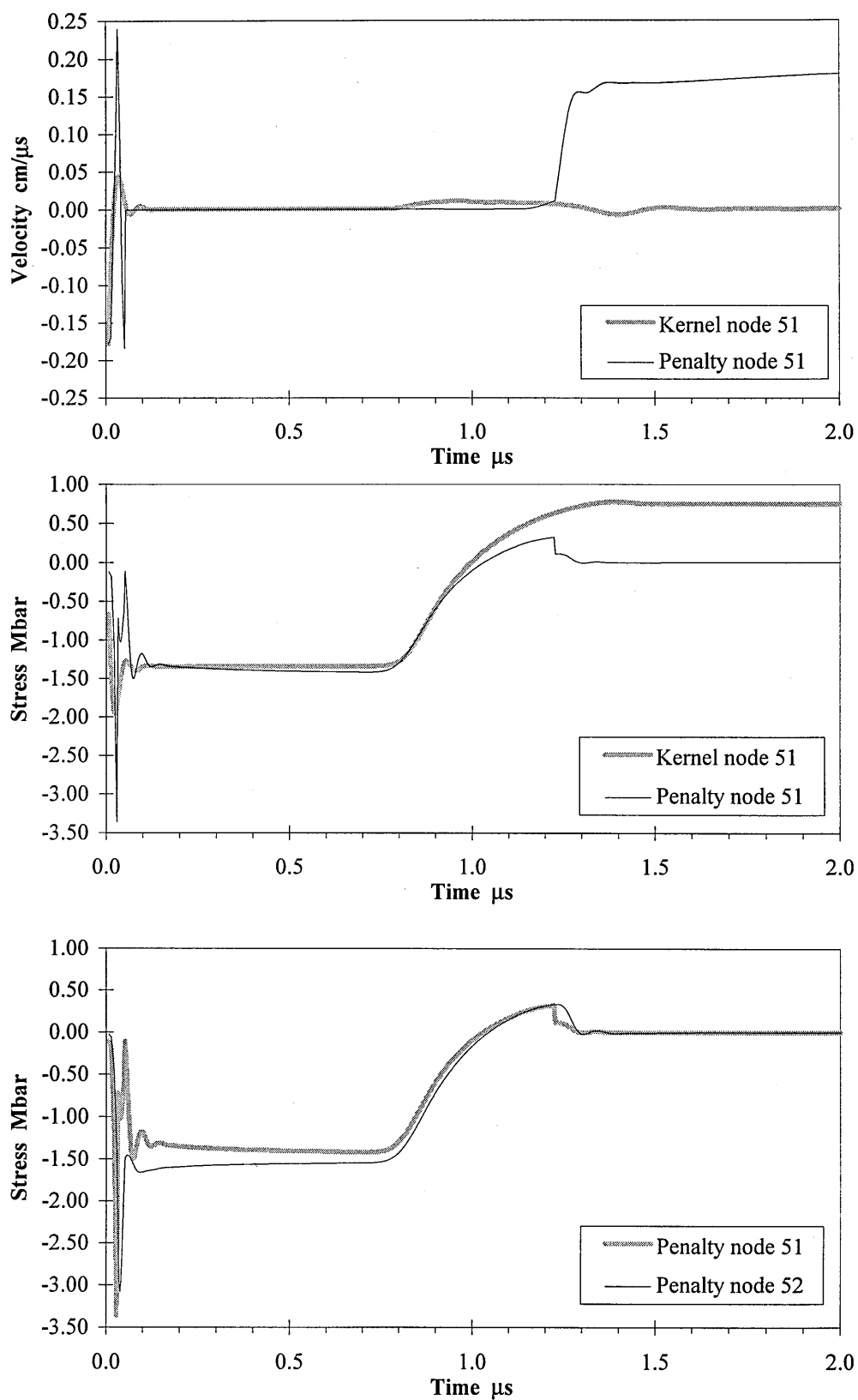


Figure 8.22: Time histories for nodes 51 and 52, showing difference in behaviour caused by stress oscillation.

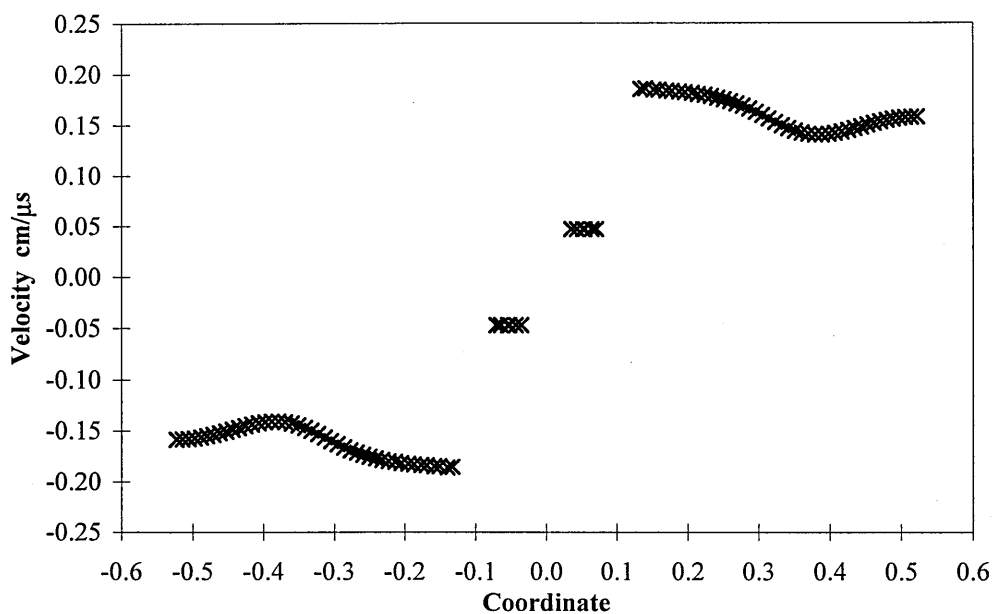


Figure 8.23: Velocity profile at $2.0\mu s$. Tensile stresses have caused numerical fracture near contact point. Penalty type 2, $k = 0.5$, $K_p = 0.01$.

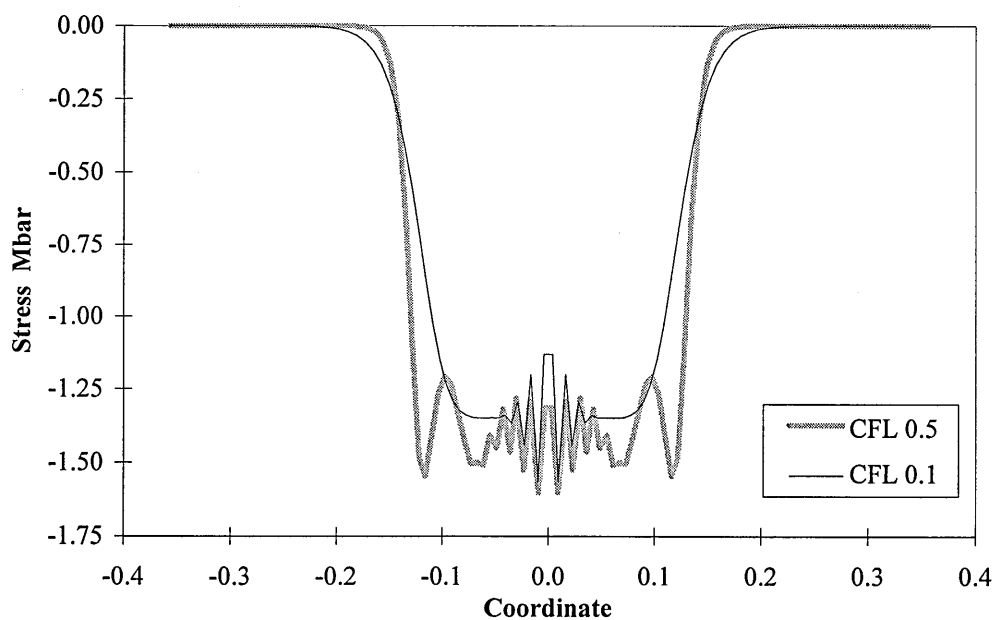


Figure 8.24: Stress profiles at $0.2\mu s$ for $k = 0.5$ and $k=0.1$. Showing the effect on the stress oscillation of reducing the time step.

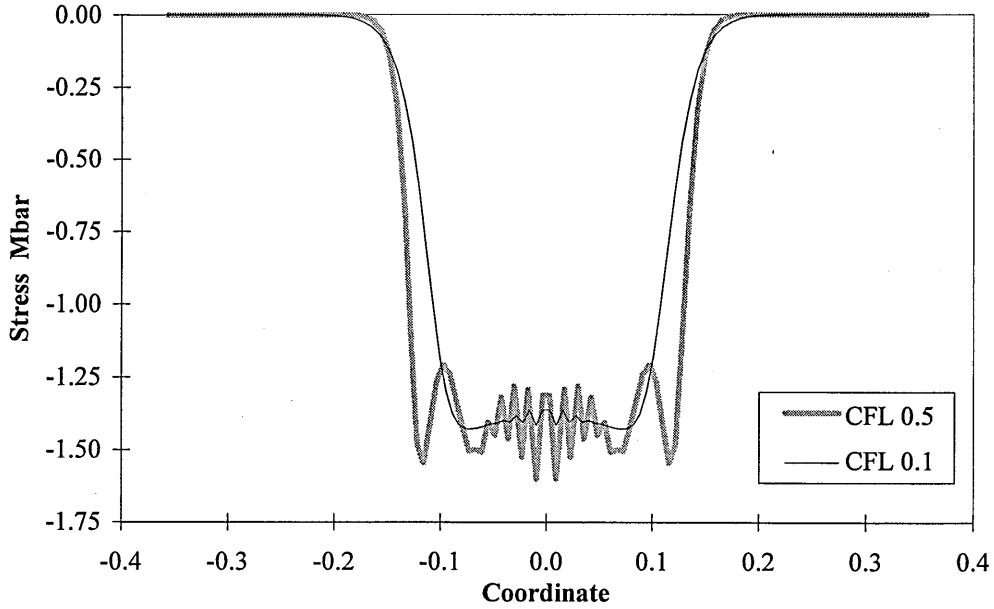


Figure 8.25: Stress profiles at $0.2 \mu\text{s}$ for $k = 0.5$ and $k=0.1$. Showing the effect on the stress oscillation of reducing the contact force scale factor.

of the oscillation, from 10 particles to 8 particles, it has not reduced the amplitude. Figure 8.25 compares the stress profile at $0.2 \mu\text{s}$ for a k of 0.5 and 0.1, but in this case $K_p = 0.01$ is used with $k = 0.1$. Reducing the contact force has reduced the amplitude of the oscillations.

To see the effect of varying the initial velocity, the initial velocity of each block was changed to 100 m/s. Figure 8.26 compares the stress profile at $0.2 \mu\text{s}$ for $k = 0.5$ and $K_p = 0.05$, with the profile for $k = 0.1$ and $K_p = 0.01$. Both using penalty force type 2. For the higher time step a large difference between the stress at the contact particles and the stress at their neighbours is seen, but the oscillation does not extend further away from the contact interface. For the reduced time step and contact force the spurious stress oscillation was not observed.

The amplitude of the oscillation is most sensitive to the magnitude of the contact force. Reducing the contact force gives a significant reduction in the amplitude. Reducing the impact velocity, which also has the effect of reducing the contact force, also reduces the oscillation. At the higher velocities the contact force can not be reduced to a level that controls the oscillation, as a low force leads to a large penetration which causes the contact algorithm to break down. Reducing the time step and the contact force gives a much reduced amplitude, and offers a way of controlling the amplitude. The disadvantage of this approach is that the tests with a reduced time step were using a time step of 5% of the maximum stable value. In higher dimensions with the consequent larger numbers of particles this size of time step will increase the computational cost of the calculation by an order of magnitude over the cost when using the kernel to provide contact.

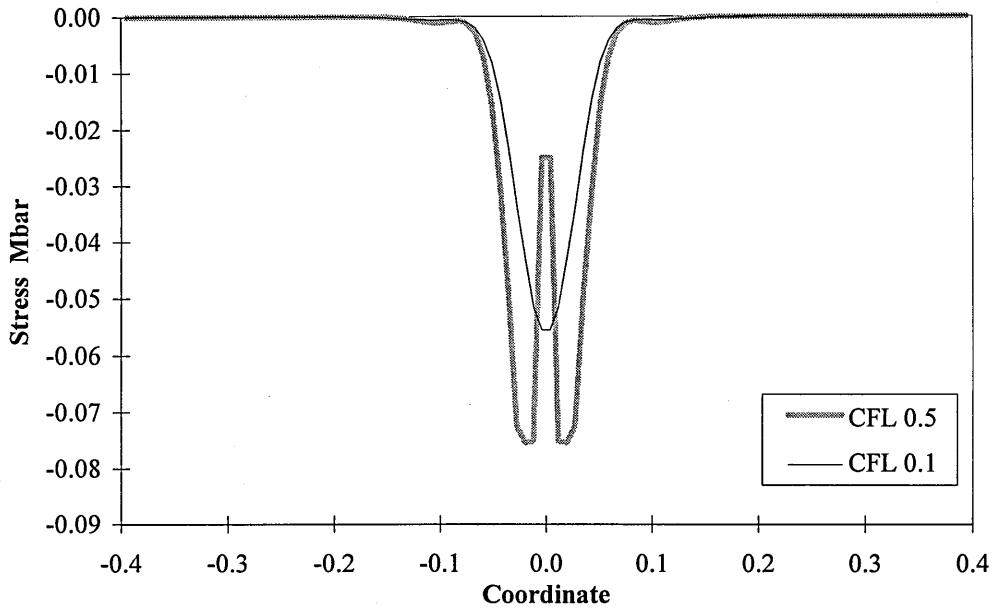


Figure 8.26: Stress profiles at $0.2 \mu\text{s}$ for $k = 0.5$ and $k=0.1$, for an impact velocity of 200m/s .

Summary of 1D tests

Tests for all three penalty force equations were run.

Penalty force 1 required a value of K_p of $80,000$ – $480,000$. The results were sensitive to change in K_p , with better results being gained for the higher value at higher impact velocities.

Penalty force 2 required a value of K_p in the range 0.008 – 0.05 . It gave very similar results to penalty type 1, and again gave improved results for higher K_p at higher velocities.

Penalty force 3 did not use a scale factor. Experience with the third penalty force type suggested that the contact force should be calculated after the conservative smoothing algorithm has been applied, because the conservative smoothing algorithm adjusts the velocity of the boundary particles. This is because conservative smoothing affects the velocities of the boundary particles.

In all cases an oscillation in the stress was observed in the area of the contact interface. As a consequence of the SPH form of the momentum equation, this spurious oscillation does not result in particle accelerations and so is not resisted. This stress oscillation causes non-physical behaviour to occur when the release wave arrives at the contact interface, leading to tensile stresses in the material.

Reducing the contact force and time step size significantly reduces the amplitude and extent of the spurious stress oscillation. However, the size of time step required would lead to computationally expensive solutions when the contact algorithm is applied to 2 and 3 dimensions.

8.6.2 2D tests

The second set of tests was in 2D. The aim of these test was to compare the effects of applying the restoring force along an average particle normal, or along the vector between the two particle centres. As the 1D tests showed that the behaviour of penalty force types 1 and 2 were similar, only penalty force types 2 and 3 were used in the 2D tests. The solid block impact problem shown in figures 8.4–8.5 was used. The contact boundary particles in each body are initially spaced h apart, so the initial penetration is exactly zero.

The following four basic cases were considered:

- Average normal contact with penalty force type 2.
- Average normal contact with penalty force type 3.
- Between centres contact with penalty force type 2.
- Between centres contact with penalty force type 3.

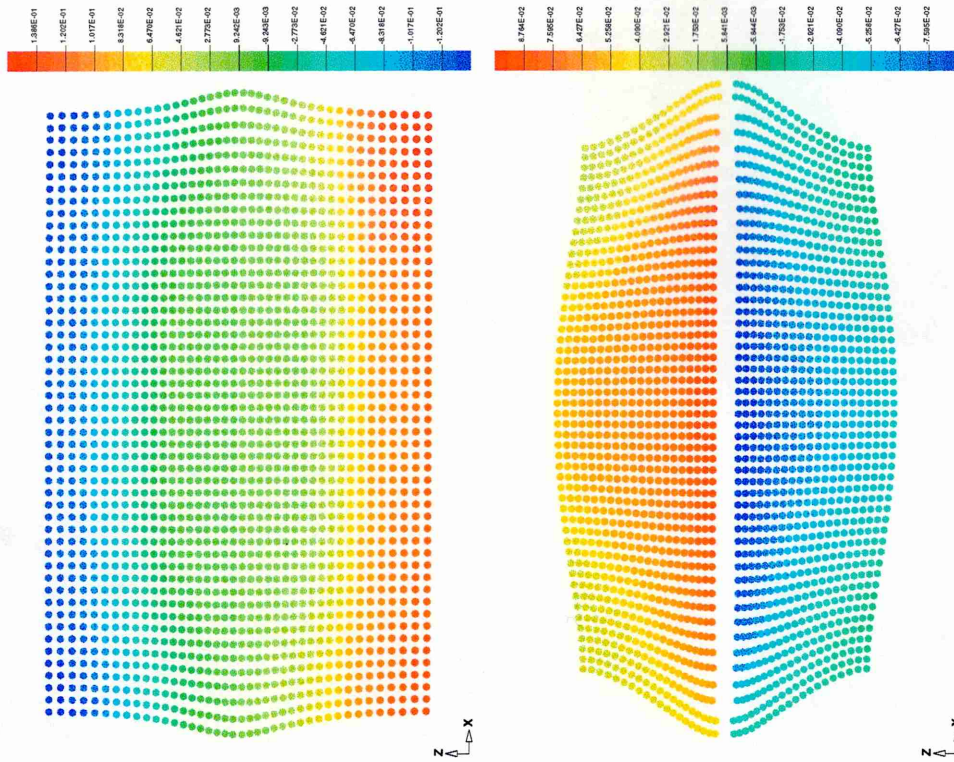
Figures 8.27–8.28 show the particle velocities for each of these cases at times $0.4\mu\text{s}$ and $1.4\mu\text{s}$, for an initial relative velocity of 4 km/s. For all four runs $k = 0.1$, and for penalty force type 2 K_p was 0.05. All four results are very similar, and the corner particles are behaving correctly. In all of these cases the particles in the two bodies are initially aligned, so that there is no difference between the average normal vector and the vector between the particle centres.

To test for the difference between the two methods, the lower block was given an initial horizontal, along the x axis, offset of $0.25h$. This offset means that the two vectors are no longer aligned, while not allowing a contact particle to be in contact with more than one other contact particle during the early stages of the calculation. To measure the difference between the cases, the total x momentum of each of the two blocks was recorded. To reduce the effect of the offset at the corners, a lower initial relative velocity of 200 m/s was used. Initially the x momentum in each body is zero and in the ideal case it would remain zero throughout the run. The x momentum time histories for the four cases are shown in figure 8.29.

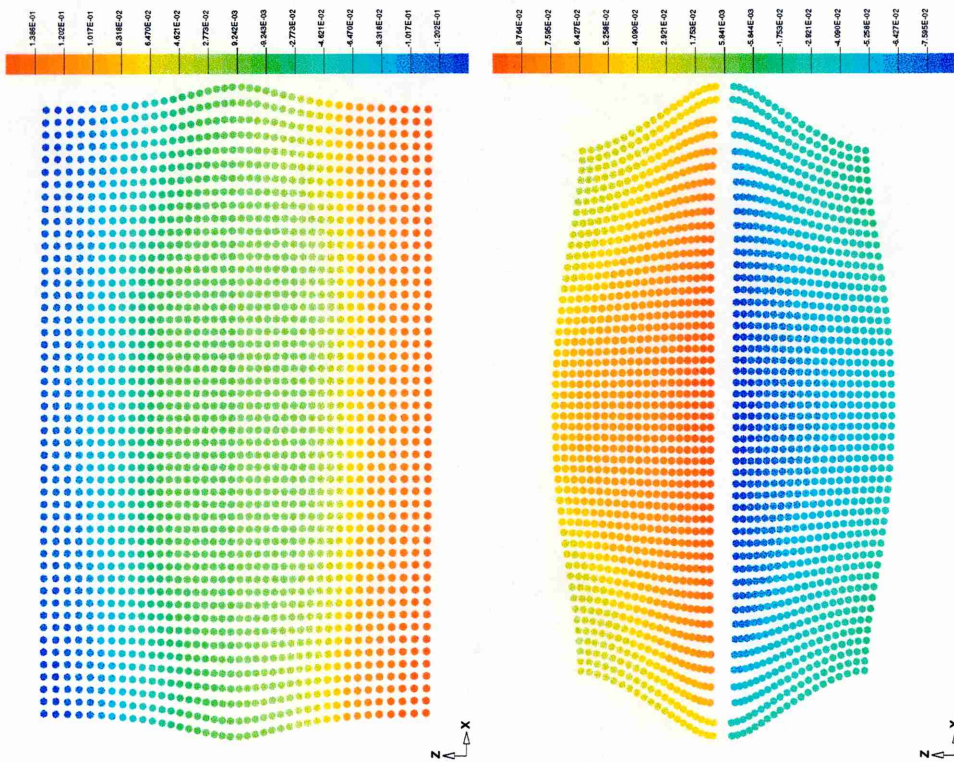
The difference between the two force vectors is clearly shown. With both penalty force types the between centres algorithm has resulted in a higher x momentum in the two bodies.

	Maximum x momentum
Average normal contact, penalty type 2	3.79×10^{-4}
Between centres contact, penalty type 2	4.30×10^{-3}
Average normal contact, penalty type 3	3.63×10^{-4}
Between centres contact, penalty type 3	5.04×10^{-3}

The maximum value for the average normal contact is for time $0.98\mu\text{s}$, the larger value that occurs later will be commented on shortly. There is an order of magnitude difference between the two cases, but is this difference significant? Each block is 1cm by 0.4 cm, and the initial material density is 7.89 g/cm^3 , giving each block a total mass of 3.156g. An x momentum of 5.04×10^{-3} means that the average horizontal velocity of the block is $0.0016 \text{ cm}/\mu\text{s}$, 16% of the initial vertical velocity of the block. Applying

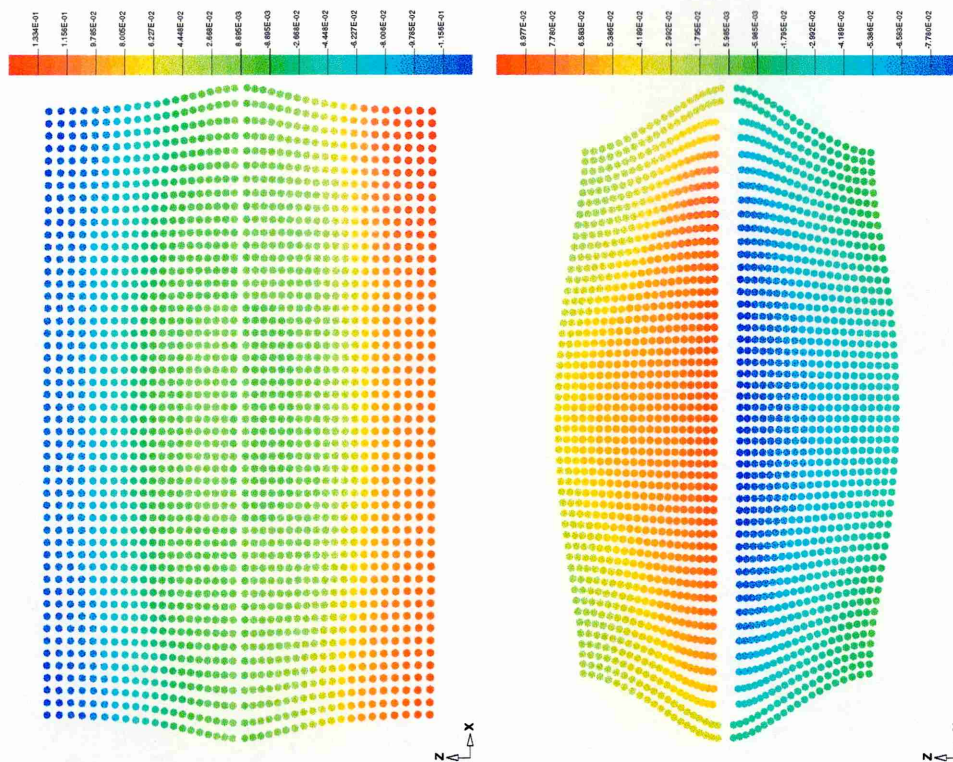


Average normal contact.

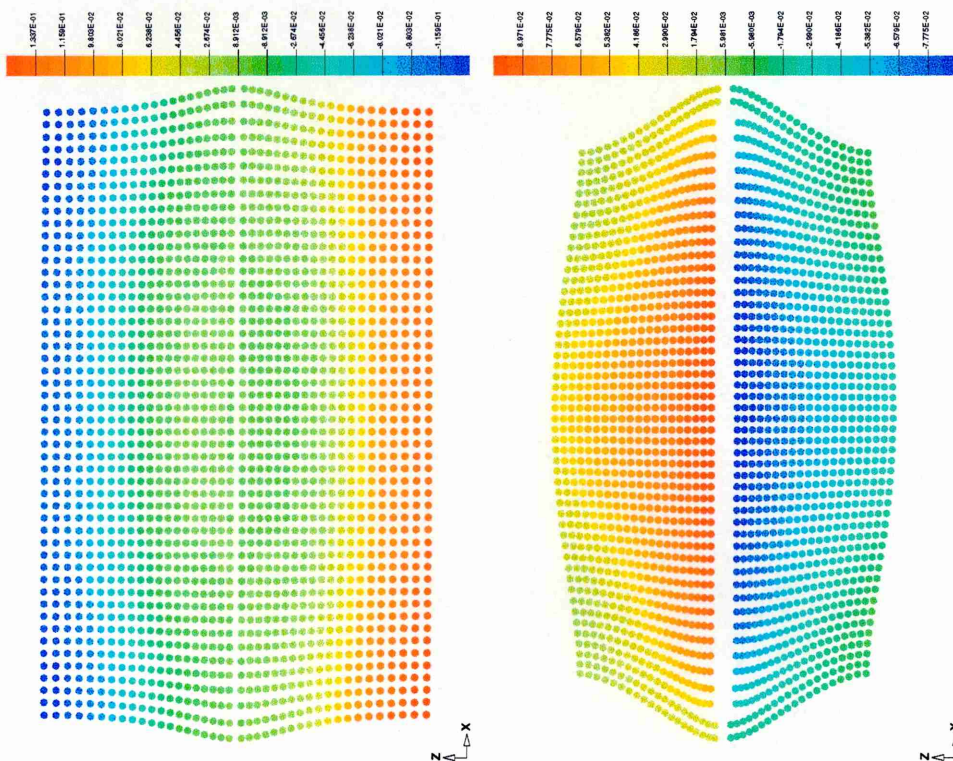


Between centres contact.

Figure 8.27: Velocity contours at $0.4\mu\text{s}$ (right) and $1.4\mu\text{s}$ (left), for the different force vectors. Penalty force type 2, $K_p = 0.05$, $k = 0.1$.



Average normal contact.



Between centres contact.

Figure 8.28: Velocity contours at $0.4 \mu\text{s}$ (right) and $1.4 \mu\text{s}$ (left), for the different force vectors. Penalty force type 3, $k = 0.1$.

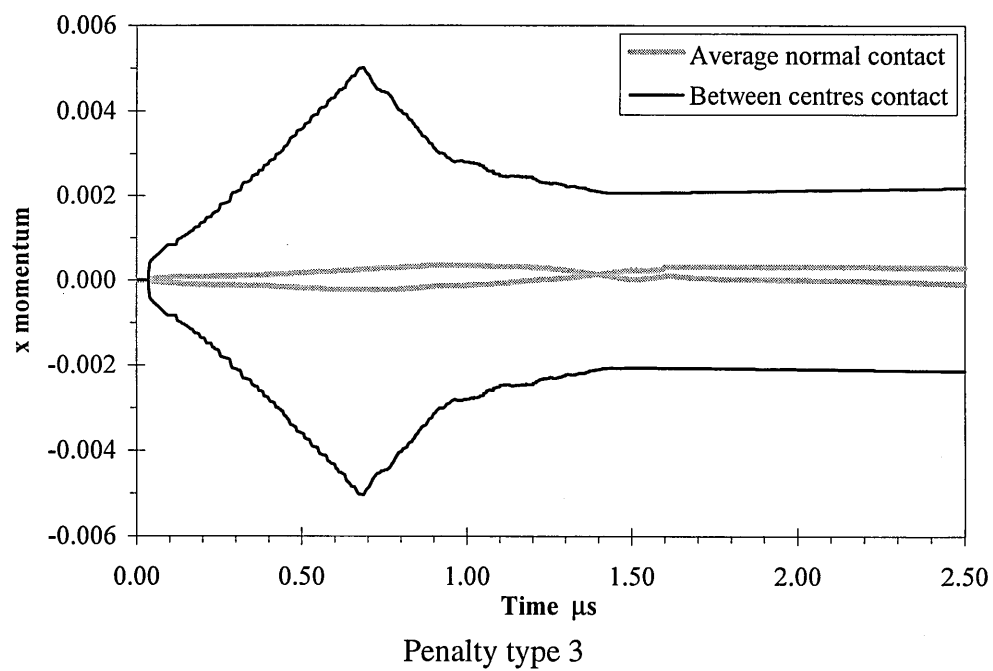
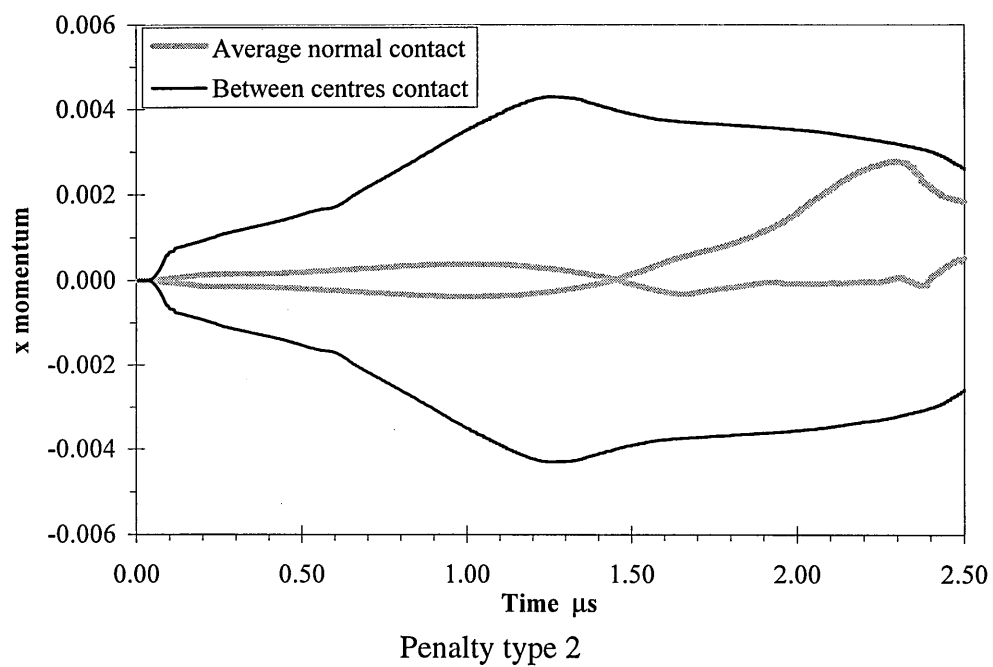


Figure 8.29: x momentum time histories for the different force vectors.

the penalty force along the vector between the two centres has resulted in each block gaining a significant horizontal momentum. These figures are the maximum values, in all cases the momentum drops after reaching its maximum value, this is because the horizontal motion has resulted in a given contact particle coming into contact with the neighbour of its initial contact particle, the resulting penalty force has an x component of opposite sign to the force from contact with the initial particle.

These tests represent a worst case, where the particle spacing of the two contact surface are the same, and all particles are offset in the same direction. In practice it is most likely that the particle spacing of both surfaces would not be identical, and so any forces tangent to the contact surface would not have such a significant global effect.

The momentum time history for the average normal contact with penalty force 2 case shows a large change in momentum after $1.5\mu\text{s}$. The cause of this change was an instability that developed near both ends of the contact surface. This instability caused visible, non-physical, particle motions. These motions were largest in this case, but were also visible in the other normal contact case and the between centres with penalty type 2 case. The only case where visible particle motion was not seen was for the between centres with penalty type 3 case. Figure 8.30 shows the particle x velocities at $2.4\mu\text{s}$ for the best and worst cases.

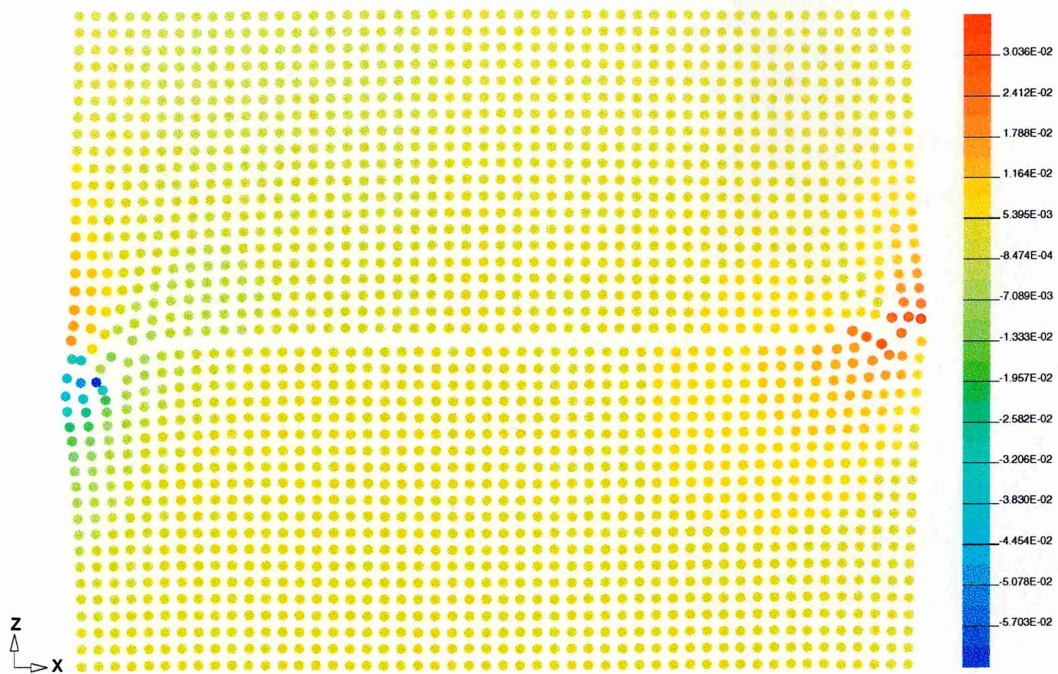
This instability is related to the saw tooth stress oscillation seen in the 1D tests, as the stress oscillation occurs near the corners before the particle motions become significant. Figure 8.31 shows the σ_{xx} stress profile at $1.0\mu\text{s}$ of the contact boundary particles of the upper block, for the average normal with penalty type2 case. The stress oscillation can be clearly seen at the two ends of the contact surface. No stress oscillation was seen in the σ_{zz} stress. The stress oscillation was also present in the two other cases where the instability caused visible particle motion.

Examining the particle stresses for the four runs without an initial particle offset showed that the stress oscillation was also present in those results. Like the offset results, the average normal contact with penalty type 2 showed the most extensive instability. Figure 8.32 shows plot of the σ_{xx} and σ_{zz} stress components at $1.0\mu\text{s}$ for this case. The stress oscillation can be seen at either end of the contact surface for both stress components.

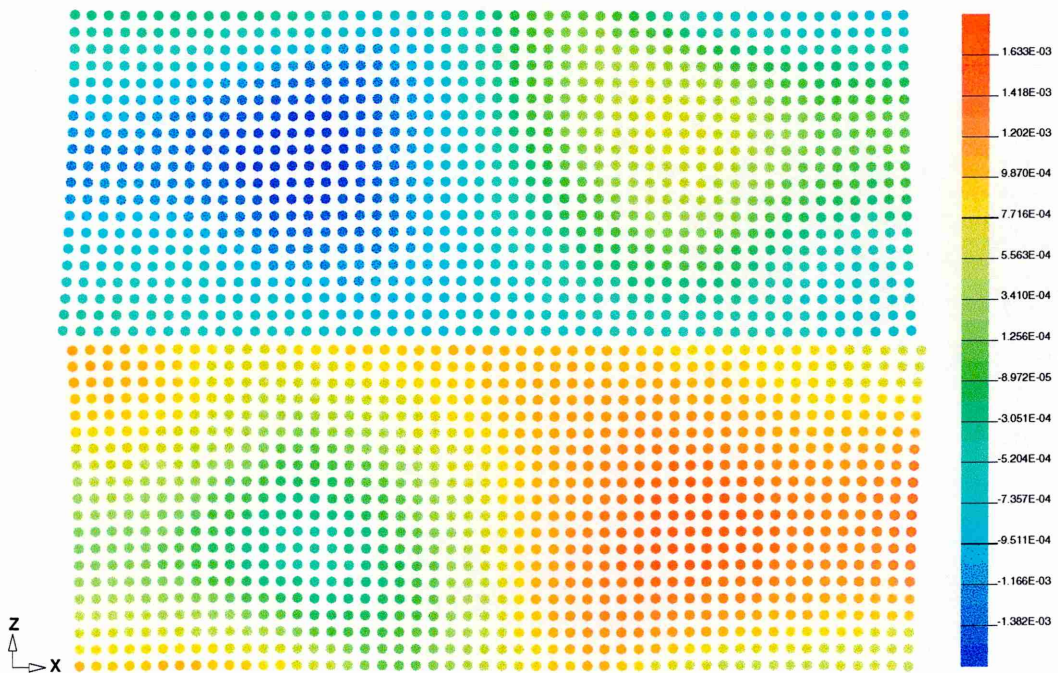
The presence of the instability appeared to be related to the stress oscillation. The 1D tests showed that reducing the time step scale factor and the penalty force reduced the amplitude and extent of the stress oscillation. Also reducing the impact velocity, with the corresponding decrease in the contact force, reduced the oscillation. Tests were carried out in 2D to find whether this instability sensitive to the same factors.

The first four tests presented in this section, for 4 km/s impact and $k = 0.1$ did not show any visible evidence of the instability. These four tests were repeated with $k = 0.3$, and all four calculations had to be stopped because the time step became very small. The decrease in the time step was caused by the minimum inter-particle distance becoming small in the region of the instability. The instability caused visible particle motions by $0.4\mu\text{s}$.

For the lower impact velocity, 200 m/s, reducing the time step and contact force also had an effect on the instability. Figure 8.33 shows the σ_{xx} stress profile along the contact surface for $k = 0.3$ and $K_p = 0.05$. Figure 8.34 shows the profile for $k = 0.1$ and a penalty scale factor on 0.01. Reducing the time step and contact force



Average normal contact, penalty force type 2.



Between centres contact, penalty force type 3.

Figure 8.30: x velocity at $2.4\mu\text{s}$ for offset impact tests. An instability can be seen with the average normal contact case. $k = 0.1$.

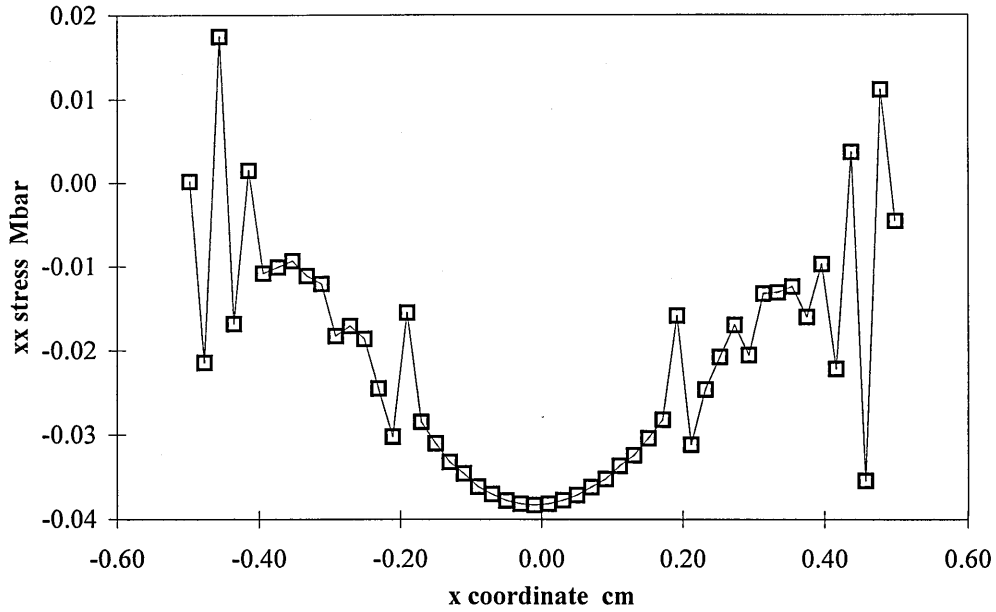


Figure 8.31: σ_{xx} profile along contact surface at $1.0 \mu s$ for the average normal with penalty type 2 case, with initially offset particles.

has reduced the amplitude and extent of the oscillation, but not removed it. Both runs showed visible evidence of the instability, and reducing the time step and penalty force has reduced the severity of the particle motions, figure 8.35.

The instability is closely related to the stress oscillation as it occurs in locations where the stress oscillation is already present. The severity of the instability is affected in the same way as the severity of the stress oscillation, by changing the impact velocity, time step and contact force. The contact algorithm used has a strong effect on the instability, with the instability being most apparent when applying the contact force along the average normal vector, and least apparent when applying the contact force between the particle centres. The average normal is calculated from the particle normals, which are in turn calculated from the surface geometry of the body, this makes this algorithm sensitive to small changes in the surface geometry.

Further tests are required before a solid conclusion can be drawn as to whether applying the restoring force between the particle centres or along an average normal is better. These tests are complicated by the presence of the instability, as over a large range of initial conditions this instability will grow to a point where it prevents the calculation from reaching the termination time or has a significant effect on the global behaviour of the problem. The instability problem must be addressed before further development of the contact algorithms can be carried out.

Summary of 2D tests

The average normal and between centres approaches were tested. The two approaches gave very similar results for a 4 km/s impact where the contact particles were initially aligned. In a test where the contact particles were initially offset, the between centres

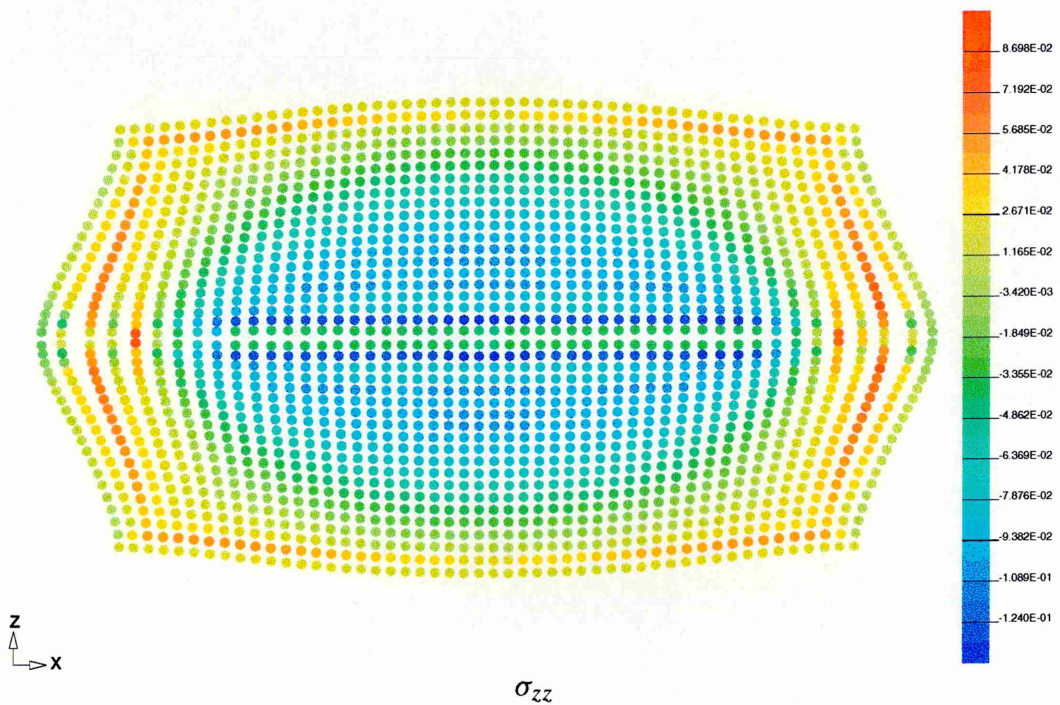
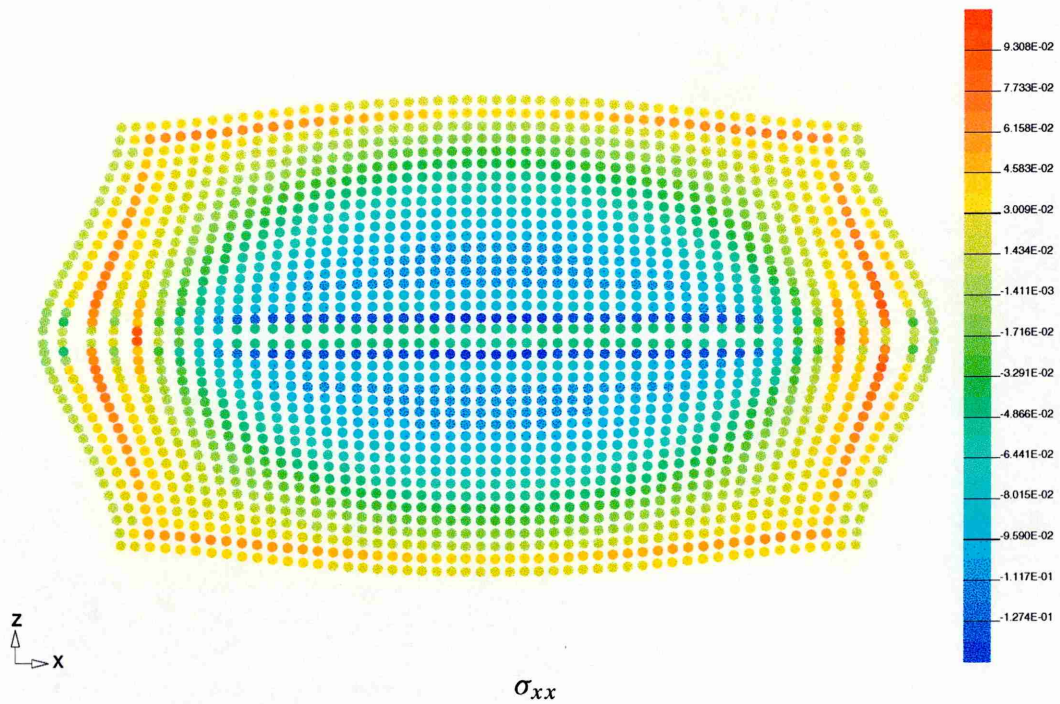


Figure 8.32: σ_{xx} stress component (horizontal) and σ_{zz} stress component (vertical) at $1.0\mu s$ for average normal contact. $k = 0.1$.

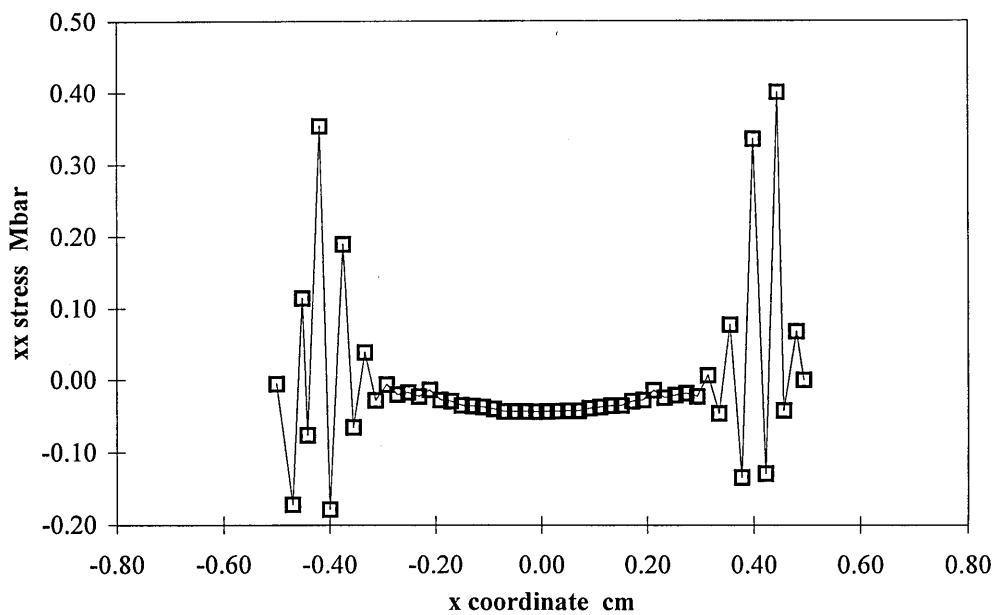


Figure 8.33: σ_{xx} profile along contact surface at $1.0 \mu s$ for the average normal with penalty type 2 case. $k = 0.3$, $K_p = 0.05$.

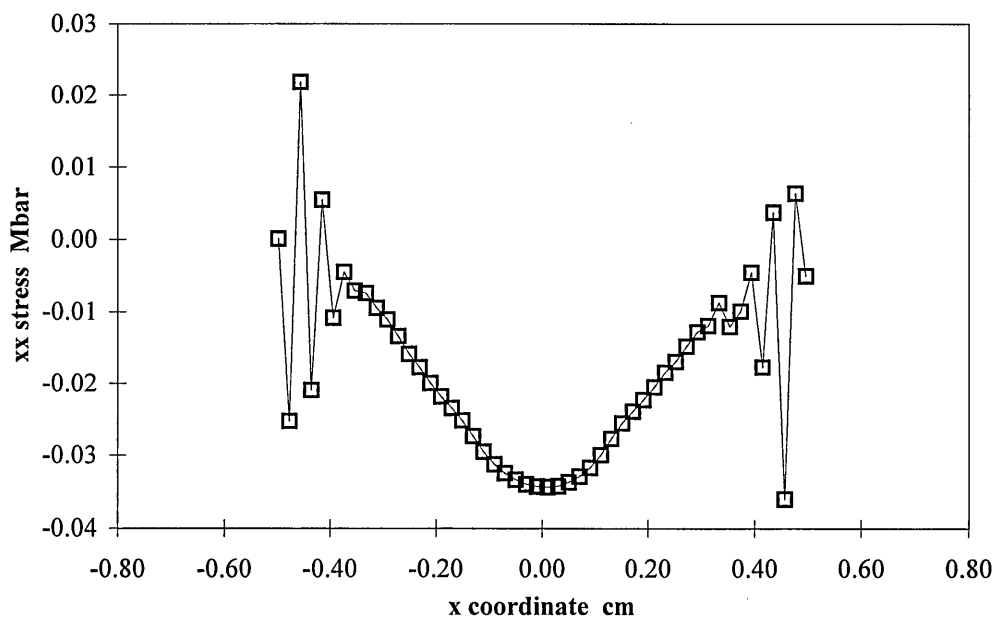


Figure 8.34: σ_{xx} profile along contact surface at $1.0 \mu s$ for the average normal with penalty type 2 case. $k = 0.1$, $K_p = 0.01$.

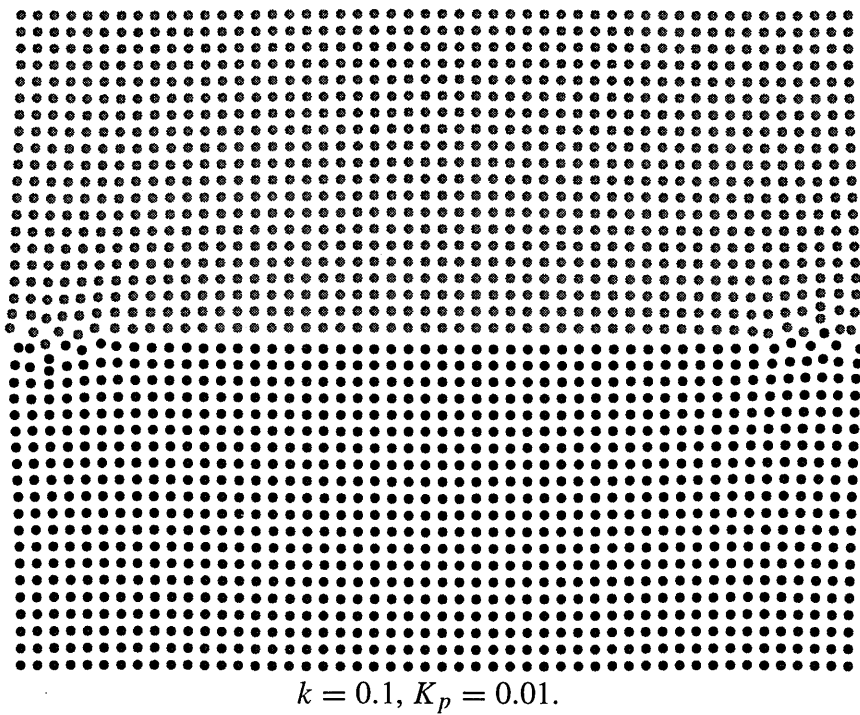
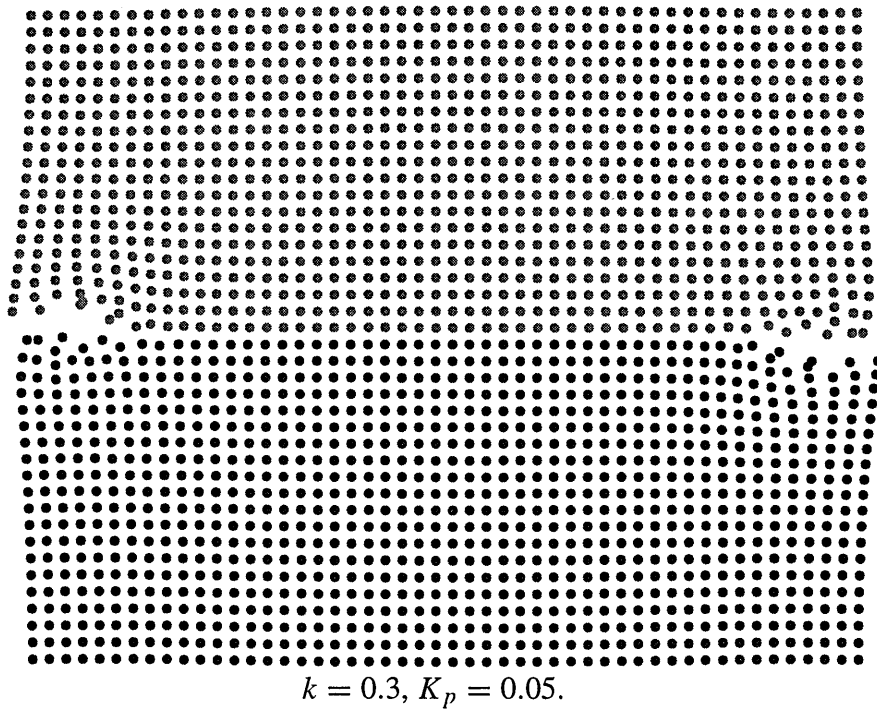


Figure 8.35: Comparison of particle positions at $2.0\mu s$ for 200 m/s impact.

approach generated significantly higher forces tangent to the contact surface.

These tests also showed an instability that occurred at either end of the contact surface, and resulted in non-physical motion of particles. The instability is related to the spurious stress oscillation observed in the 1D tests. Its severity can be reduced, but not eliminated, by reducing the contact force and time step. The presence of the instability prevents the full testing of the proposed contact algorithm.

This instability has to be addressed before the average normal and between centres approaches can be adequately tested, as the instability obscures any difference between the two.

8.6.3 Discussion

Both the 1D and 2D tests demonstrate that the penalty contact approach can be applied to the SPH method. Impact and separation were successfully modelled. The tests also showed that contact excites a stress oscillation in the region of the contact interface. In both 1D and 2D the oscillation affected the final results, in some cases rendering the final results completely inaccurate. Tests showed that reducing the time step and contact force significantly reduced the severity of the oscillation. Reducing the time step offers a method of controlling the instability, however the time step required is under 10% of the time step that is practical when using kernel contact. This large reduction in the time step represents a very significant computational penalty, and requiring the penalty would decrease the overall attractiveness of the SPH method.

As the stress oscillation is a characteristic of the SPH method, not the particular contact algorithm used, it is not possible to justify the view that changing the contact algorithm would prevent the oscillation occurring. The oscillation has also been observed in the standard SPH method when using kernel sums to simulate contact [35]. The stress oscillation requires addressing and controlling.

8.7 Displaced stress point approach

In both the 1D and 2D tests of the SPH contact algorithm, problems have occurred that can be traced to the spurious stress oscillation. If the contact algorithm is to be practical then a solution to the stress oscillation must be found. There are two approaches that could be taken:

- Find a way to resist the formation of, and dissipate, the spurious stress oscillation, perhaps similar in conception to the hourglass viscosity in finite elements.
- Use an alternative discretisation method that does not suffer from the saw-tooth stress oscillation.

The second approach is better, as the first approach will result in a code containing separate fixes to the pathologies in the basic SPH method: tensile instability and stress oscillation.

An ideal alternative discretisation method will not suffer from tensile instability or stress oscillation, but will retain the advantages of SPH, simplicity and computational efficiency. Other meshless methods, such as EFGM, are significantly more

computationally expensive to use. The oscillation is a consequence of evaluating all field variables and their derivatives at the same locations. A method that evaluates the stresses and velocities at separate points would not suffer from the oscillation. In addition, Dyka suggests that such a method would not suffer from the tensile instability [93]. This type of method can retain SPH kernel interpolation and so retain the computational efficiency. The sacrifice made is in simplicity.

The basic conception of such a method is to use two separate types of particle: velocity particles are the points at which the momentum equation is evaluated, and stress particles which are the points where the stress is evaluated. The velocity points are equivalent to the conventional SPH particles, and so are Lagrangian in nature. The stress points should also be Lagrangian, to allow the accurate calculation and tracking of material history variables. This adds the problem of how to move the stress points, as the momentum equation is not evaluated at these points. A second problem with this approach is how to initially distribute the particles, and ensure that there are always enough of both types of particle in a neighbourhood to allow accurate interpolation.

Neither of these two problems are trivial in 2D or 3D, but in 1D it is possible to easily solve both. In 1D a stress point would be initially placed between each pair of velocity particles. The velocity of the material at the stress point could be interpolated from its two neighbouring velocity points, and so the stress point will always remain between the two velocity points.

In order to test the displaced stress point approach with contact, a 1D code was developed that used this method. If this code solved the problems that appeared in the 1D SPH contact algorithm tests, section 8.6, then work on extending the method to higher dimensions would be worthwhile.

8.7.1 1D algorithm.

The only algorithm of this type that has been proposed is that of Dyka [93]. His algorithm associates an element with each velocity point. The boundaries of the element lie exactly half way between the point and its neighbours. Two stress integration points are placed within each element, at a distance R from the closest boundary. When $R = 0$ the points lie on the element boundaries, when $R = 0.5$ the points are coincident with the velocity point, which is conventional SPH.

The 1D algorithm presented here differs from Dyka's in that the stress points are not associated with the velocity points. It assumes that particles represent sampling points, where the corresponding field variable is known. The normalised SPH interpolation method is used to calculate the field, and the gradient of the field, at any point. In the initial distribution a stress point is located in between each pair of velocity points. Neighbour particles are located in the usual SPH manner, however one type of point can only have neighbours of the other type. The stress points must be moved in a Lagrangian manner, so it is necessary to know their velocities. An SPH interpolation is used to calculate the value of the velocity at the stress points, which are then moved with that velocity. In 1D this is equivalent to ensuring that the stress point always lies half way between the two velocity points. However moving stress points by keeping them between each particle pair is not a viable approach to moving stress points in higher dimensions.

Normalised forms of the kernel approximations were used. It was necessary to derive normalised forms of the approximation of the gradient at a point where the value of the function is not known, and of the approximation to the value of the function.

In 1D the SPH approximation to the value of a function, c , at point i is:

$$c(x_i) = \left[\sum_j \frac{m_j}{\rho_j} c(x_j) W_{ij} \right] \frac{1}{B} \quad (8.26)$$

where B is the factor used to correct the equation.

Assume $c = a + bx$, and substitute in (8.26):

$$a + bx_i = \left[\sum_j \frac{m_j}{\rho_j} (a + bx_j) W_{ij} \right] \frac{1}{B} \quad (8.27)$$

which can be rewritten

$$(a + bx_i)B = \sum_j \frac{m_j}{\rho_j} (a + bx_j) W_{ij} \quad (8.28)$$

so

$$B = \sum_j \frac{m_j}{\rho_j} W_{ij} \quad (8.29)$$

Now normalise the gradient approximation:

$$\frac{dc(x_i)}{dx} = \left[- \sum_j \frac{m_j}{\rho_j} c(x_j) \frac{dW_{ij}}{dx} \right] \frac{1}{B} \quad (8.30)$$

Assume $c = a + bx$, and substitute:

$$b = \left[- \sum_j \frac{m_j}{\rho_j} (a + x_j) \frac{dW_{ij}}{dx} \right] \frac{1}{B} \quad (8.31)$$

rearranging gives:

$$B = - \sum_j \frac{m_j}{\rho_j} \left(\frac{a}{b} + x_j \right) \frac{dW_{ij}}{dx} \quad (8.32)$$

Here there is a problem, as in the general case a and b are not known. This problem does not occur in standard SPH as the difference of the function, $(c_j - c_i)$ is used, and this results in the $\frac{a}{b}$ term cancelling. So the normalised approximation to the gradient of a function at a point where the value of that function is not known cannot be used directly in a computer code

For the purpose of the 1D test code, it was decided to approximate the gradient between two nodes as the gradient of the linear interpolation of the value. For a reduced neighbour set, where only the two immediate neighbours are considered, this is equivalent to the normalised gradient approximation, as will now be shown.

Consider a point i , where the gradient of function c must be calculated. It has two neighbour particles, $j1$ and $j2$ that lie on either side of point i . The linear approximation to the gradient at i is:

$$\left(\frac{dc}{dx}\right)_i = \frac{(c_{j2} - c_{j1})}{(x_{j2} - x_{j1})} = b \quad (8.33)$$

The normalised kernel approximation to the gradient at i , using (8.30) for 2 neighbour particles $j1$ and $j2$, is:

$$\left(\frac{dc}{dx}\right)_i = \frac{-\left[\frac{m_{j1}}{\rho_{j1}}c_{j1}\frac{dW_{ij1}}{dx_{j1}}\right] - \left[\frac{m_{j2}}{\rho_{j2}}c_{j2}\frac{dW_{ij2}}{dx_{j2}}\right]}{-\left[\frac{m_{j1}}{\rho_{j1}}\left(\frac{a_{j1}}{b_{j1}} + x_{j1}\right)\frac{dW_{ij1}}{dx_{j1}}\right] - \left[\frac{m_{j2}}{\rho_{j2}}\left(\frac{a_{j2}}{b_{j2}} + x_{j2}\right)\frac{dW_{ij2}}{dx_{j2}}\right]} \quad (8.34)$$

a and b are the coefficients of the local linear approximation. So $b_{j1} = b_{j2} = b$ and is the gradient calculated in equation 8.33, and a can be written as:

$$a_j = c_j - (bx_j) \quad (8.35)$$

substituting (8.35) into (8.34) and rearranging gives:

$$\left(\frac{dc}{dx}\right)_i = \frac{-\left[\frac{m_{j1}}{\rho_{j1}}c_{j1}\frac{dW_{ij1}}{dx_{j1}}\right] - \left[\frac{m_{j2}}{\rho_{j2}}c_{j2}\frac{dW_{ij2}}{dx_{j2}}\right]}{\frac{1}{b}\left(-\left[\frac{m_{j1}}{\rho_{j1}}c_{j1}\frac{dW_{ij1}}{dx_{j1}}\right] - \left[\frac{m_{j2}}{\rho_{j2}}c_{j2}\frac{dW_{ij2}}{dx_{j2}}\right]\right)} = b \quad (8.36)$$

So (8.34) is equal to (8.33).

The implementation of the algorithm is described below.

A consequence of using the SPH interpolation method is that every particle must have a mass and density. Ideally only the velocity points would have mass, and only the stress points, where the velocity gradient is evaluated, would store density. In order for the accelerations to be calculated correctly, the sum of all the velocity point masses must equal the total mass of the system, so it is necessary to set:

$$M = \sum m_{\text{velocity points}} = \sum m_{\text{stress points}} \quad (8.37)$$

where M is the total mass of the body. The density at the velocity points is calculated using a kernel interpolation, in the same manner as the velocity at the stress points. The variable known at each point are given in table 8.1.

Start of loop

Calculate velocity derivative at stress points:

$$\left(\frac{dv}{dx}\right)_i = \frac{v_{j2} - v_{j1}}{x_{j2} - x_{j1}} \quad (8.38)$$

where i is a stress point, $j1$ and $j2$ are the neighbour velocity points.

	Velocity points	Stress points
Variables calculated:	Acceleration Velocity Position	Density Strain rate Stress tensor Pressure Internal energy Position
Variables interpolated:	Density	Velocity

Table 8.1: The variables required at each type of particle

Update density

Use conservation of mass equation at stress points:

$$\frac{d\rho_i}{dt} = \rho_i \left(\frac{dv}{dx} \right)_i \quad (8.39)$$

$$\rho_i^{n+1} = \rho_i^n + \frac{d\rho_i}{dt} \Delta t \quad (8.40)$$

where i is a stress point.

Use SPH interpolation to find density at velocity points

$$\rho_i = \frac{\sum_j m_j W_{ij}}{\sum_j \frac{m_j}{\rho_j} W_{ij}} \quad (8.41)$$

where i is a velocity point, and j are stress points.

Calculate strain rate

$$\varepsilon_i = \left(\frac{dv}{dx} \right)_i \quad (8.42)$$

where i is a stress point.

Update stress tensor at stress points using material model.

Calculate acceleration of velocity points

Momentum equation:

$$\left(\frac{dv_i}{dt} \right)_i = \frac{1}{\rho_i} \left(\frac{\sigma_{j2} - \sigma_{j1}}{x_{j2} - x_{j1}} \right) \quad (8.43)$$

where i is a velocity point, $j1$ and $j2$ are the neighbour stress points.

Update velocities

For velocity points:

$$v_i^{n+1/2} = v_i^{n-1/2} + \left(\frac{dv_i}{dt} \right)_i \Delta t \quad (8.44)$$

where i is a velocity point

For stress points:

$$v_i = \frac{\sum_j \frac{m_j}{\rho_j} v_j W_{ij}}{\sum_j \frac{m_j}{\rho_j} W_{ij}} \quad (8.45)$$

where i is a stress point, and j are velocity points

Update positions of all particles.

$$x^{n+1} = x^n + v^{n+1/2} \Delta t \quad (8.46)$$

End of loop

In order to keep the test code simple, only a perfect elastic material model was implemented. In this model the stress increment is:

$$\Delta \sigma = (K \dot{\epsilon}_v + G(\dot{\epsilon} - \dot{\epsilon}_v)) \Delta t \quad (8.47)$$

where K and G are the elastic bulk and shear moduli, and $\dot{\epsilon}_v$ is the volumetric strain rate. This model is identical to the perfectly elastic material model in DYNA3D, and does not use an equation of state.

8.7.2 Boundary conditions

The algorithm presented above does not address boundary conditions. In the approach based on distinct velocity and stress points, it is possible to use either type as a boundary particle.

It was first assumed that at a boundary the sampling point closest to the boundary would be a stress point. As the stress points are just sampling points, the outermost stress point was assumed to lie exactly on the boundary. This simplifies the application of the stress free condition, as the boundary stress point is now the point where the stress component normal to the contact surface must be set to zero. As the stress points lie on the boundary contact is easily detected, and the penalty force is applied to the velocity point closest to the boundary. The use of kernel interpolation to calculate

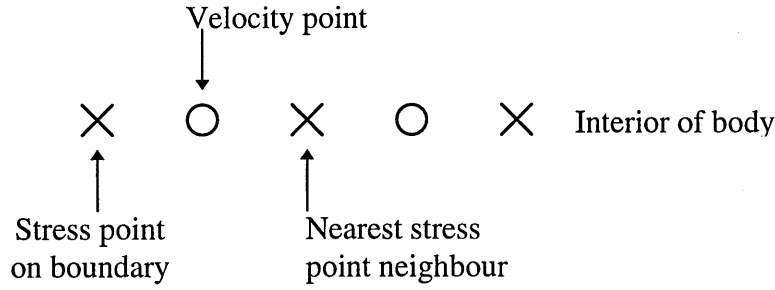


Figure 8.36: Nearest stress point neighbour to boundary stress point.

the velocity at stress points results in the velocity of the boundary stress point being identical to the velocity of its neighbour velocity point. With the stress point in contact, and so having a non-zero stress, there is the problem of how to calculate the stress at this point. Equation 8.33 cannot be used, as the boundary particle only has one velocity point neighbour. Initially the boundary stress point was assumed to have the same stress as its nearest stress point neighbour, see figure 8.36. However, this results in a zero stress gradient, and hence zero acceleration, at the outermost velocity point. The consequence of this is that once contact occurs, the points will never separate, even if the stress is tensile, due to the zero gradient. The method finally implemented was to linearly interpolate the stress at the boundary point from the stress inside the body. This results in a non-zero stress gradient at the velocity point. This method works, but is not ideal. Placing a stress point on the boundary has retained the need to have a switch which deactivates the stress free condition when contact occurs.

It would be more satisfactory to apply the stress free condition in a manner that does not require activation and deactivation. This can be achieved by placing a velocity point, rather than a stress point, on the boundary. This removes the need to calculate the stress at the boundary and so explicitly apply the stress free condition. With this arrangement all stress points have two neighbour velocity points, and so the velocity gradient can be found at all stress points. This leaves the problem of calculating the stress gradient at the boundary velocity point. This can be done by taking the stress outside the body as zero, and calculating the gradient accordingly, the equation implemented was:

$$\left(\frac{d\sigma}{dx}\right)_i = \frac{1}{2} \left(\frac{\sigma_j}{x_j - x_i}\right) \quad (8.48)$$

where i is the boundary velocity point, and j is its neighbour stress point. The result is that the force applied to the boundary point from the stresses act to move the boundary point in a manner that will reduce the stress at its neighbour stress point. The result is that the stress free boundary condition is satisfied for the surface. This is essentially the same method by which the Lagrangian finite element method satisfies the stress free condition. A second advantage of this approach is that the boundary condition no longer needs to be switched on and off, but is always applied. When the boundary is in contact the contact force acts to resist the motion of the boundary caused by the stress, resulting in a non zero stress at the boundary.

The penalty method was used to enforce the contact condition. Two penalty forces were used, equations 8.16 and 8.21.

8.8 Tests of displaced stress point method

The purpose of the tests of the displaced stress point algorithm were to show that the algorithm works correctly, and so show that it is a solution to the spurious stress oscillation problem in SPH. To do this a problem similar to that used for the 1D SPH tests in section 8.6 was used.

The case is the impact of two 1.0 cm thick blocks, each with an initial velocity of 2 km/s. The two blocks are initially spaced 0.2 cm apart, which means contact occurs $0.5\mu\text{s}$ after the start of the calculation. The number of each type of particle in each bar changes depending on whether a velocity point is on the boundary. If a stress point is on the boundary, each bar consists of 50 velocity points and 51 stress points. If a velocity point is on the boundary, each bar consists of 51 velocity points and 50 stress points. The material properties used for the aluminium bars were:

Density	2.70 g/cm^3
Youngs Modulus	0.70 Mbar
Poisson's ratio	0.33

The results from three tests are presented, each test is repeated for stress points on the boundary and for velocity points on the boundary.

The first test was to show that the method and the boundary conditions worked correctly. No contact algorithm was used, so the two bars were treated as one single piece of material with kernel sums allowing interaction between the two bars.. Figure 8.37 shows the stress profile at three times, for both boundary stress points and boundary velocity points. The earliest plot is for time $0.7\mu\text{s}$, and shows the stress waves propagating away from the point of contact, the middle plot is for time $2.5\mu\text{s}$ where the stress waves have reflected from the free ends, the final plot is at $4.0\mu\text{s}$ where the stress is now tensile, as kernel contact does not allow the blocks to separate. These plots show that the method and the boundary conditions are working correctly. No problems were seen when the bar went into tension.

The second test used equation 8.16 for the penalty force, the third test used equation 8.21 for the penalty force.. Figure 8.38 shows the stress profile at three times in the calculations using the second penalty equation. Figure 8.39 shows the stress profiles at the same three times for the calculations using the third penalty equation. The first plot is at $1.2\mu\text{s}$, which is $0.7\mu\text{s}$ after impact and so is equivalent to the first stress profile for the kernel contact. The second plot is at $3.0\mu\text{s}$, which is equivalent to time $2.5\mu\text{s}$ for the kernel contact. The third plot can not be compared to the kernel results as separation of the two blocks has occurred.

The calculations using the third penalty equation show very good agreement with the kernel results, and do not exhibit any oscillation at the contact interface. The calculations using the second penalty equation also show good agreement away from the contact interface, at the contact interface there is some stress oscillation with the velocity point boundary, and a large jump in stress for the stress point boundary. The reason for the difference between the two penalty forces is that using the second penalty equation results in a force that is sufficient to reduce the penetration to zero. When this occurs with a stress point boundary particle the stress is immediately set to zero. Figure 8.40 shows the stress time histories for the stress points closest to the contact

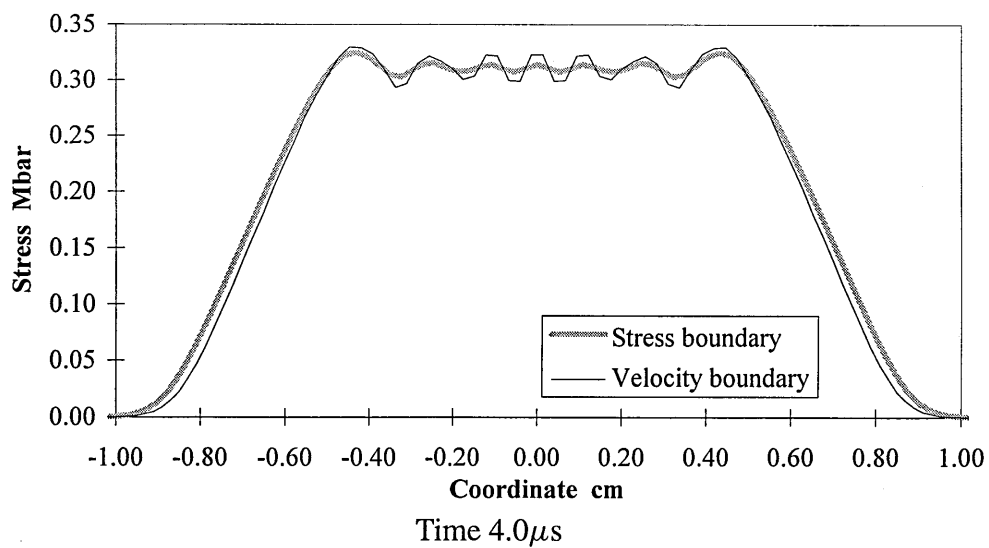
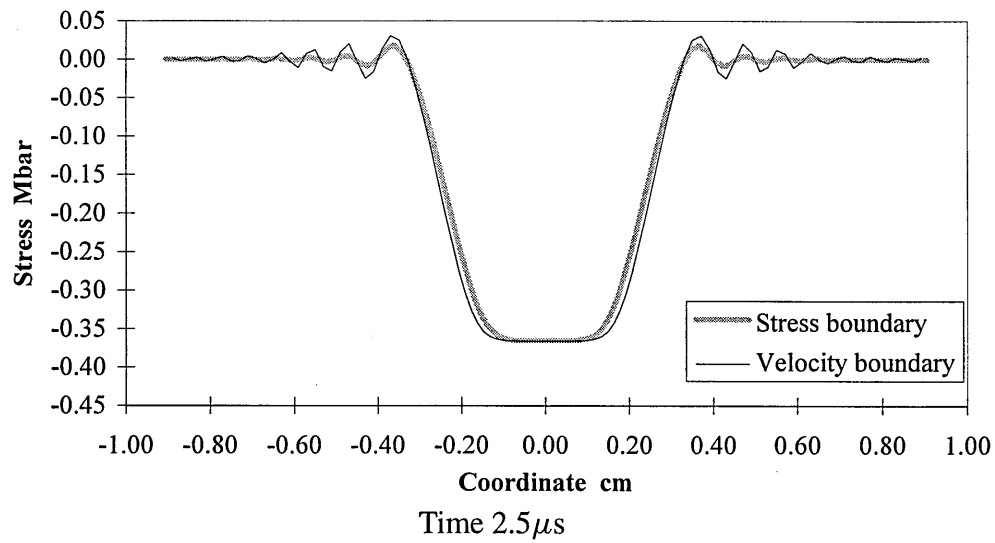
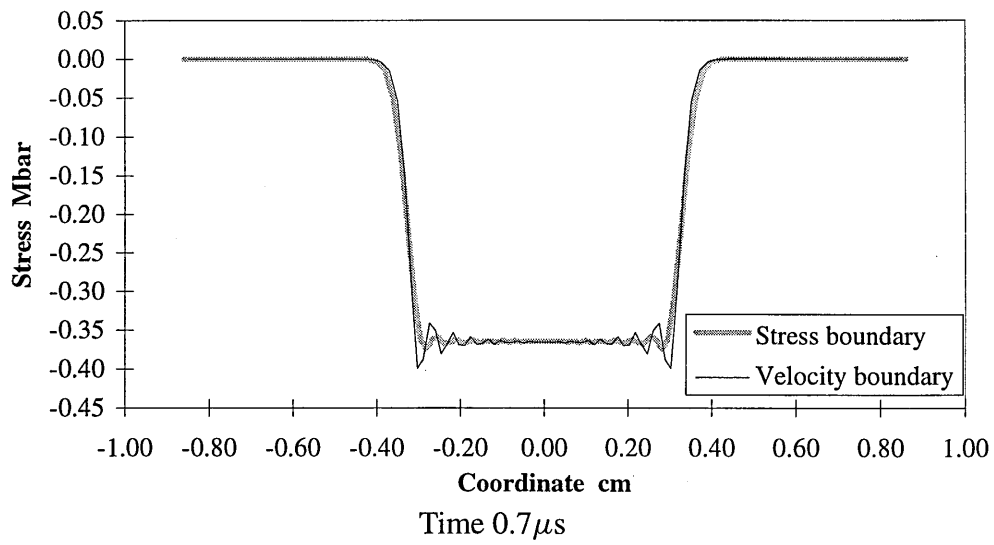


Figure 8.37: Stress profiles for kernel contact.

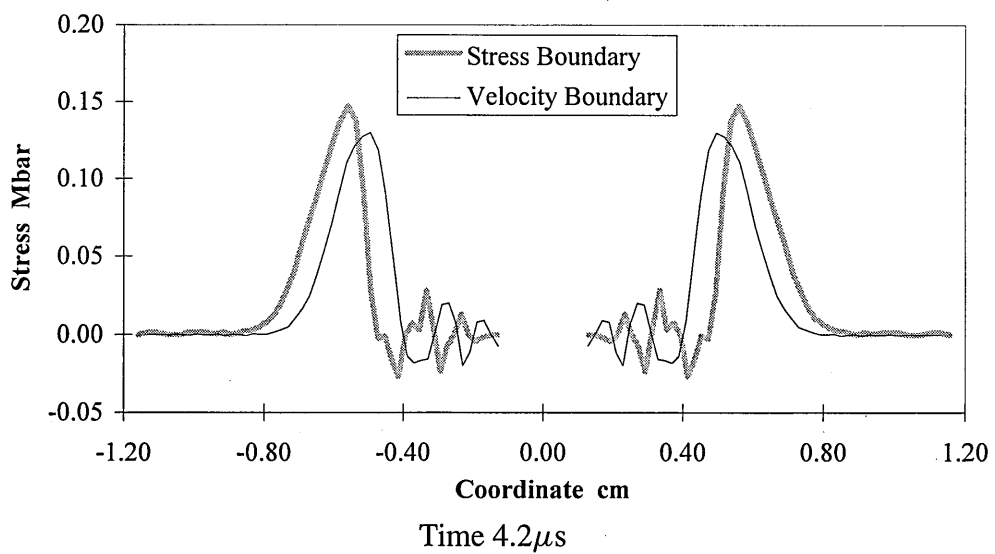
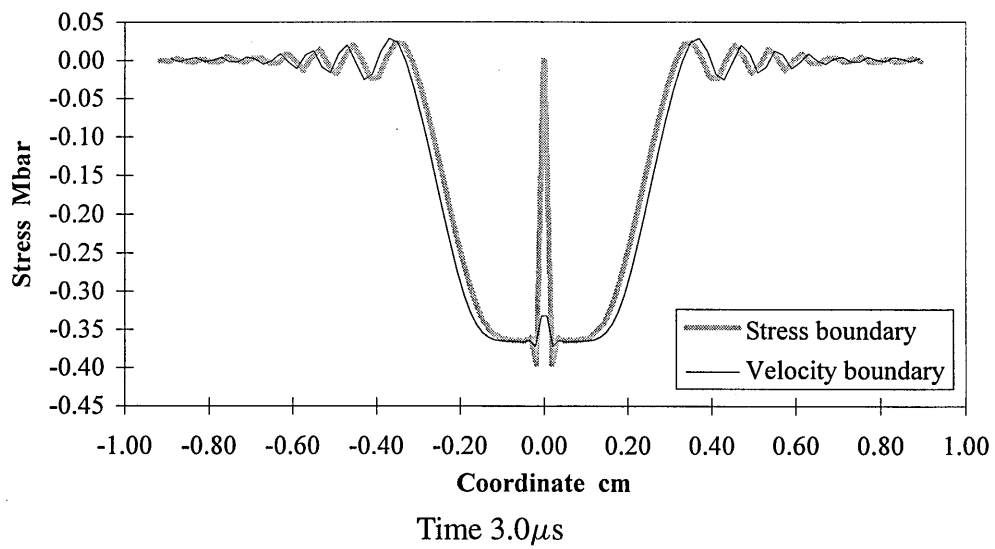
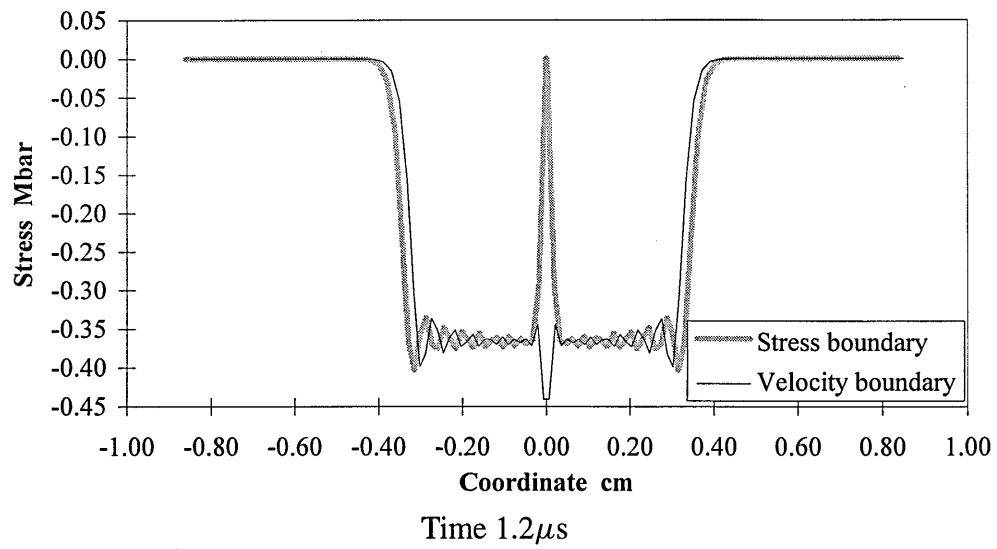


Figure 8.38: Stress profiles for penalty equation 2.

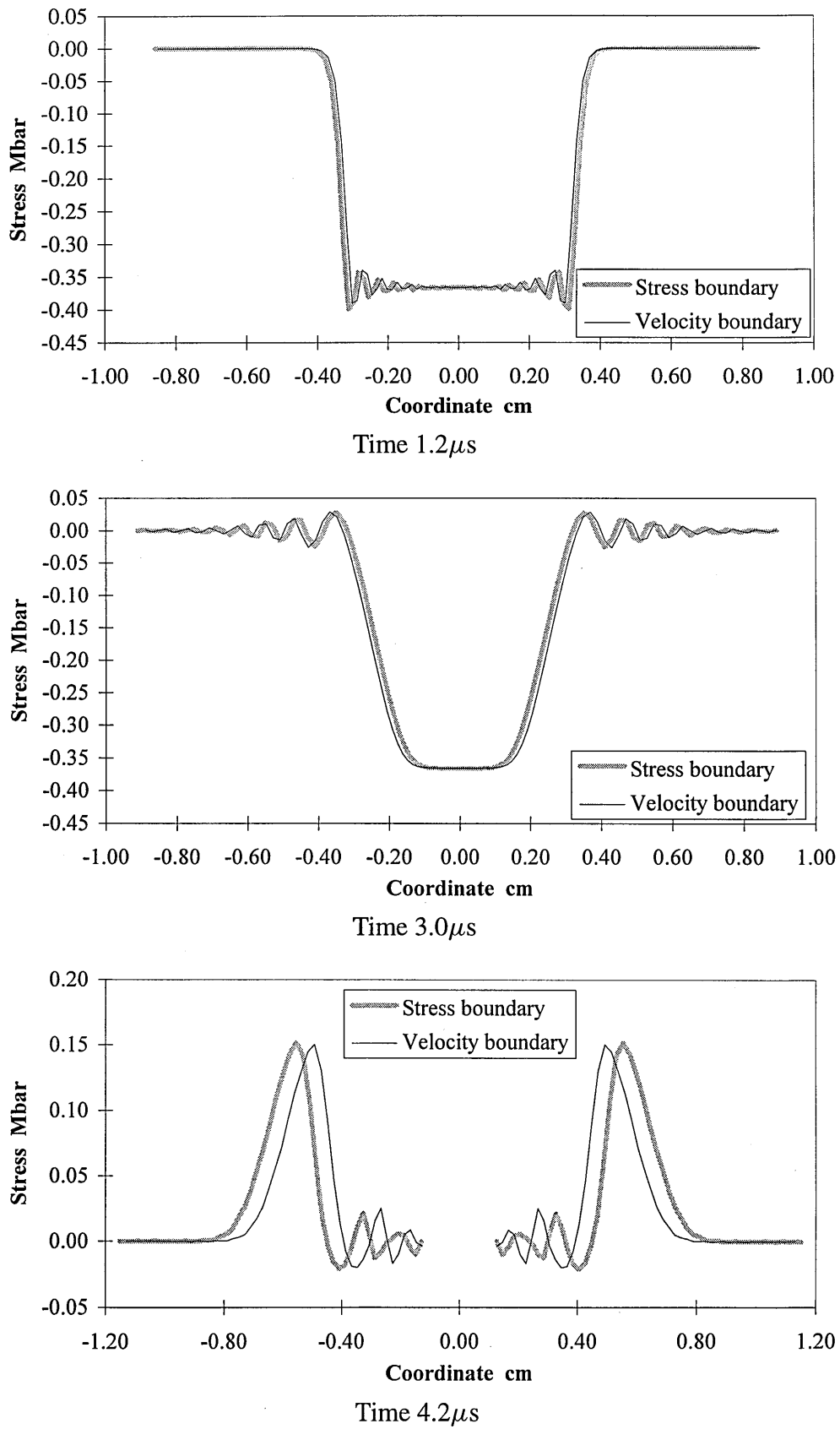


Figure 8.39: Stress profiles for penalty equation 3.

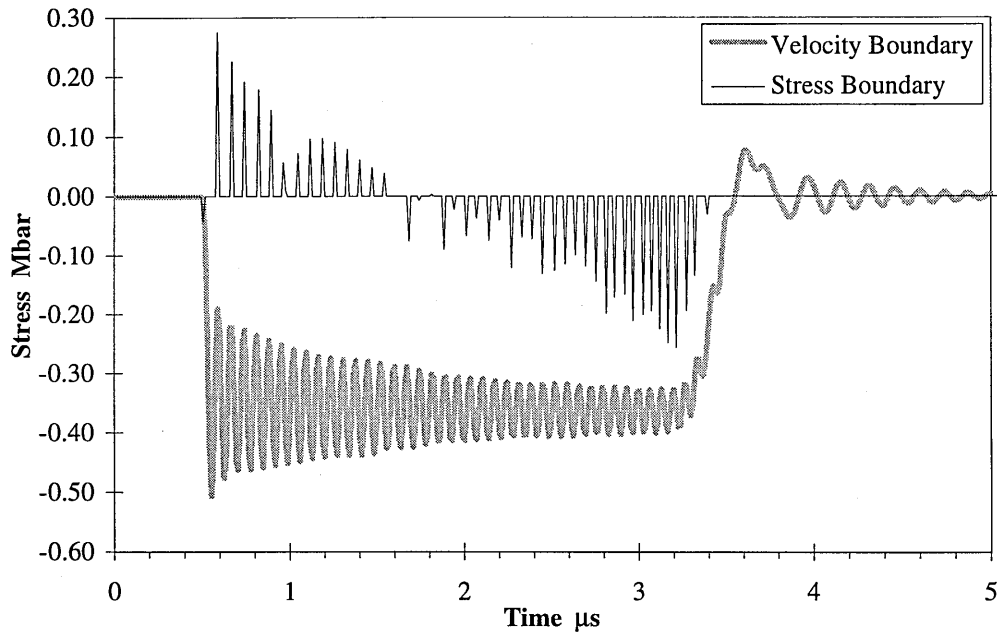


Figure 8.40: Stress time histories for stress point closest to contact interface, for penalty equation 2.

interface for penalty equation 2. This shows that with velocity boundary points the oscillation is tending towards a constant value. The oscillation for stress boundary points does not tend towards a constant value, and shows rapid changes between a zero and non-zero stress as the contact surfaces penetrate, separate and penetrate again. The tensile stresses are a consequence of the linear extrapolation that is used to calculate the stress at the boundary point.

The stress profiles for time $4.2\mu\text{s}$ show that a strong tensile stress wave has been generated on separation. This tensile stress can be seen in the time history for the velocity point boundary end stress in figure 8.40. The effects of the tensile stress can be seen in figure 8.41 which show the variation of the total kinetic energy with time. For a perfectly elastic impact, kinetic energy should be preserved, so the kinetic energy after the impact should be identical to the energy before the impact. The time histories show that the final kinetic energy is less than the initial. The values are:

	Energy $\text{g cm}^2 / \mu\text{s}^2$
Initial value	0.1080
Final value for stress point boundary, penalty equation 2	0.0965
Final value for stress point boundary, penalty equation 3	0.0967
Final value for velocity point boundary, penalty equation 2	0.0972
Final value for velocity point boundary, penalty equation 2	0.0966

There is approximately a 10% shortfall in the final value of the kinetic energy. This difference is due to the stress wave that was generated on separation.

This stress wave is a consequence of using a discrete system to model a continuous system. With both types of boundary particle the stress near the contact interface must go tensile to generate acceleration that results in separation. As the number of particles

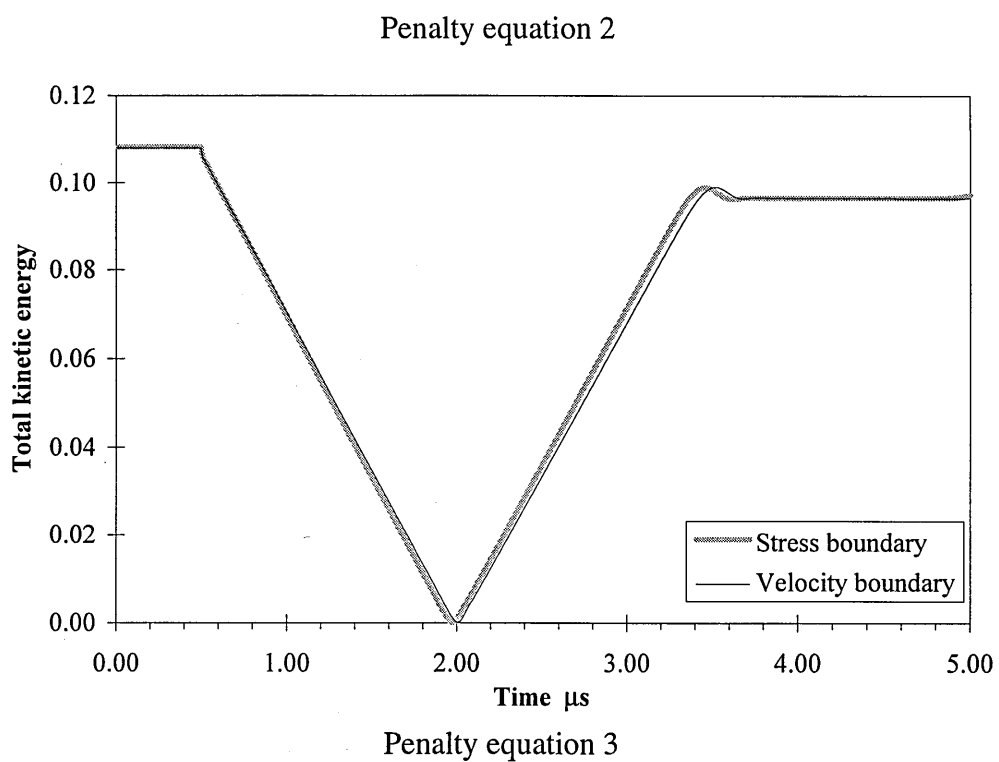
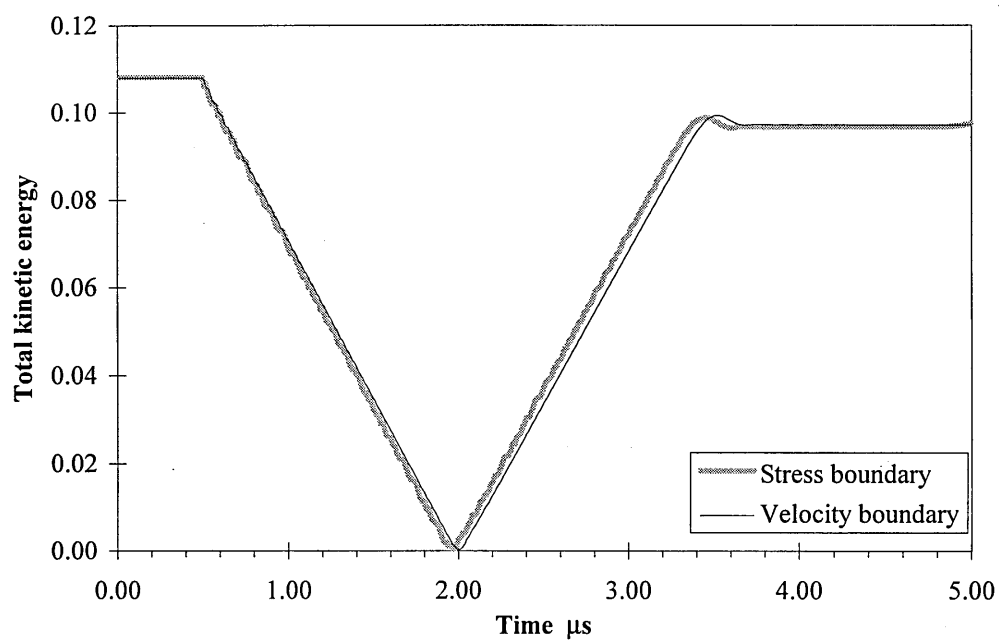


Figure 8.41: Total kinetic energy time histories.

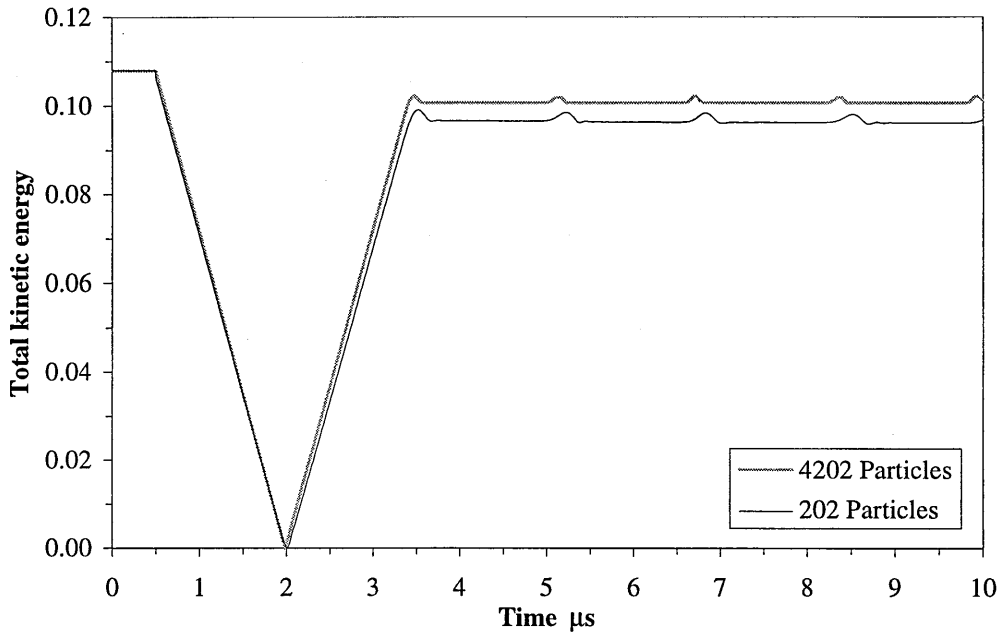


Figure 8.42: Total kinetic energy time histories, showing effect of increasing spatial resolution.

is increased, the error in the final value of the kinetic energy is reduced, figure 8.42. This shows the kinetic energy time histories for two problems, one using 202 total particles, the same number that has been used in the previous tests, and the other using 4202 particles. The final value of the kinetic energy with 4202 particles is $0.1006 \text{ g cm}^2 / \mu\text{s}^2$. The regular humps seen in both time histories after separation occur as the stress wave in the bars reflect off one of the free ends. As a final check the 1D SPH results were compared to DYNA3D results. A pseudo-1D mesh was constructed by using four symmetry planes to enclose a block of material, the same technique that was used in chapter 4 for the Hugoniot tests. One 1.0 cm block was modelled, with a stonewall being used to simulate symmetric contact. 50 elements were used through the thickness, giving the same node density as the SPH case. Figure 8.43 shows the stress time histories for the DYNA3D and SPH results at the contact surface, and at a mid-point in the bar. Good agreement is seen between the two. The amplitude of the oscillations in the SPH results is higher, and the rate of decay of the amplitude is lower, but the same basic features are seen in both results. The results also show a period difference, as both analyses use an identical material model and identical material properties, the probable cause is the difference in time integration algorithms used in the two codes. Both codes use central difference time integration, but DYNA3D uses the velocity centred form, with the velocity evaluated at time $n + 1/2$.

Summary of displaced stress point results

The tests of the 1D displaced stress point algorithm show that it works correctly, and provides a solution to the spurious stress oscillation that was shown up by the tests of the standard SPH method. Tests were carried out with both velocity points and stress

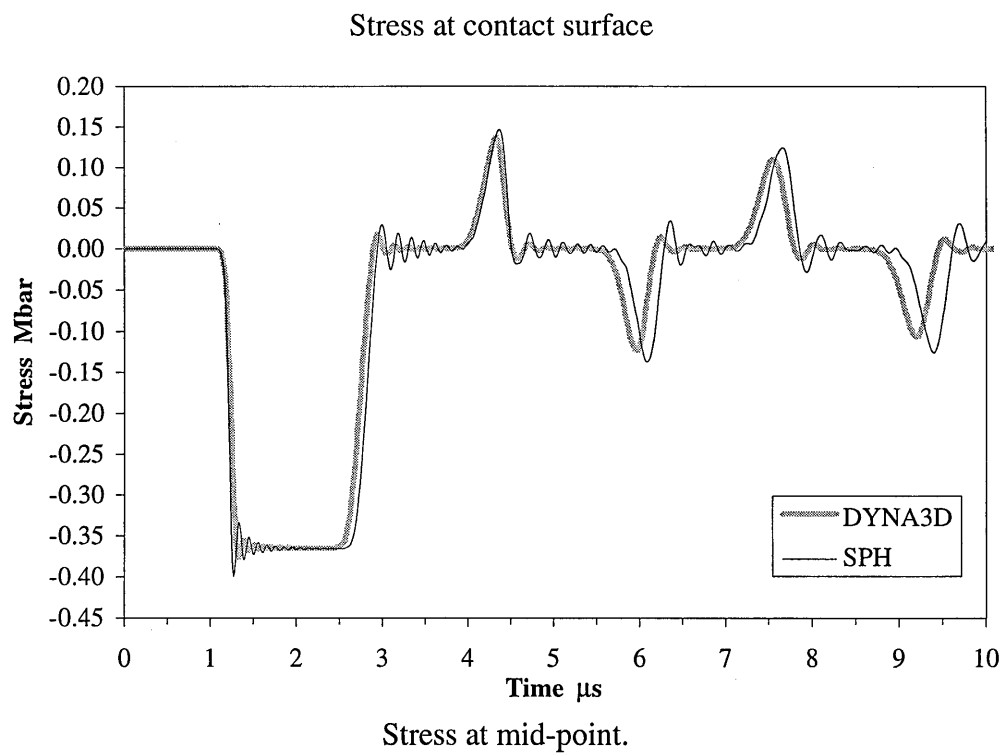
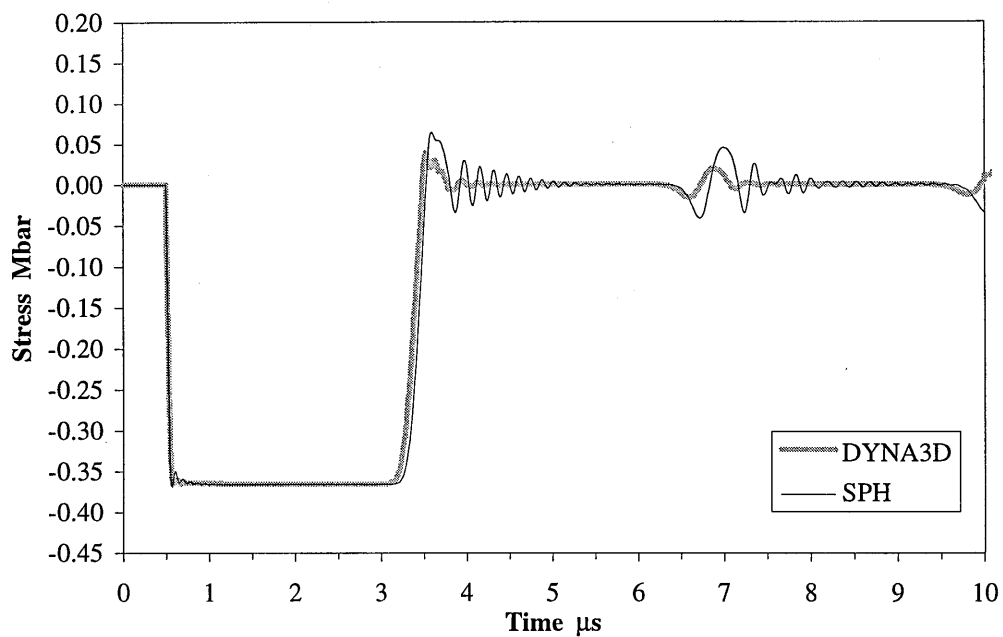


Figure 8.43: Comparison of stress time histories for velocity point on boundary and DYN3D.

points on the boundary. With stress points on the boundary the stress free condition has to be directly applied when not in contact, and this was the cause of oscillation at the contact surface because as soon as the penetration goes to zero the stress free condition is applied. The cases where a velocity point was the boundary particle gave better results in the contact tests, and this boundary condition was more satisfactory as it does not require an algorithm to switch the stress free condition off and on. The results were compared with results from DYNA3D, and good agreement was seen.

8.9 Discussion

This chapter addresses the treatment of contact in SPH. A penalty contact algorithm is proposed. This algorithm considers contact as a particle to particle interaction, not a particle to surface algorithm. This means that the 2D algorithm is easily extendible to 3D, unlike a particle to surface algorithm, where the contact surface would need to be generated each time step.

The 1D and 2D tests of the algorithm showed that that particle to particle contact using the penalty method to enforce contact can be successfully applied to the SPH method. The tests also showed that a spurious stress oscillation, which is inherent in the basic SPH method, can cause serious problems. In the 1D tests the oscillation occurred near to the contact surface and affected the behaviour at the contact surface by delaying separation of the two bodies. In the 2D tests the oscillation occurred near the ends of the contact surface, and was directly related to an instability that caused non-physical particle motions that could prevent successful completion of the calculations.

Calculating the stresses at points away from the points where the stress gradient is evaluated offers a solution to the stress oscillation problem. An additional advantage of this approach is that it should not suffer from the tensile instability, and so not require the use of a conservative smoothing algorithm. A displaced stress point algorithm in 1D is proposed and tested. The tests show that this approach does cure the stress oscillation problem, and no evidence of a tensile instability was seen. This method also offers an approach to improve the treatment of the stress free boundary condition. In the normalised basic SPH method the stress free boundary condition must be explicitly applied, this requires the detection of the boundary particles where the condition must be applied. With the displaced stress point method, the boundary can be defined by points where only the velocity is known. This allows the stress free condition to be applied automatically, without the need to treat it separately and detect when to apply it.

The disadvantage of the stress free approach is that it is not easy to extend from 1D to 2 and 3D. Any 2 or 3D algorithm has to ensure that there will always be particles of each type in the neighbourhood of every particle to ensure correct interpolation of the solution variables. The advantages that this approach offers would make the necessary effort worthwhile.

Chapter 9

Conclusions

This thesis addresses the problem of modelling hypervelocity impact in a Lagrangian hydrocode. The aim of the thesis was to produce a robust and accurate method for modelling hypervelocity impact on spacecraft structures.

The principal work and conclusions of the thesis are:

- An unmodified DYNA3D code is not capable of accurately modelling hypervelocity impact on spacecraft structures. Three areas where improvement was required were: material modelling, element erosion and modelling of the debris cloud.
- The SESAME equation of state was implemented to improve the constitutive models needed in hypervelocity impact simulation.
- Two element erosion criteria were developed. The first based on total element deformation. The second on element accuracy.
- In tests the element deformation criterion gave the best results for crater size for a semi-infinite target and hole diameter in a thin target, but eroded too much material from the debris cloud for the thin plate targets. The accuracy criterion gave the best results for the debris cloud, allowing debris cloud velocities and expansion angle to be found.
- Element erosion does not offer a solution to the problem of modelling the formation of a debris cloud when modelling multiple impacts.
- The meshless Lagrangian method Smoothed Particle Hydrodynamics offers a way to model the debris cloud without eroding material. However it required development of contact algorithms.
- A particle to particle penalty contact algorithm was developed for 1 and 2D. 1D and 2D tests of the contact algorithm showed that the zero-energy mode problem in SPH had to be addressed.
- An alternative discretisation method was proposed that does not suffer from the zero-energy modes. This method uses two types of particles: velocity particles to discretise the velocity field and stress particles to discretise the stress field.

- Boundary condition treatment in SPH is simpler if velocity particles are used as boundary particles.

The work presented in this thesis has improved the DYNA3D code by implementing an equation of state that is more capable than any of the existing equations of state in the code. It has investigated the use of different element erosion criteria and the suitability of element erosion when applied to impact on thin plates. It has addressed the contact boundary condition in the Smoothed Particle Hydrodynamics method, and highlighted the need to provide a solution to the zero-energy modes of SPH. It has proposed an alternative discretisation method that provides a solution to the zero-energy modes.

Future work

Further work is required to achieve the goal of developing a robust and accurate method for modelling hypervelocity impact on spacecraft structures. The Lagrangian finite element method, coupled with the SPH method offers a suitable method. The SPH method requires further development to bring it to the point when it is mature enough to be used in engineering.

There are four areas where further work is required:

- Extend 1D velocity-stress point algorithm to higher dimensions.
- Finish development of contact algorithm for SPH once a 2 and 3D method is available that does not suffer from zero-energy modes.
- Link new SPH method with the Lagrangian finite element method.
- Address lack of suitable strength models. A strength model for metals is required which is valid over the whole strain rate range encountered and for which there is a reasonable range of material data. In addition satellites use advanced composite and non-metallic materials, suitable models for these materials are required.

References

- [1] B.P. Denardo, J.L. Summers, and C.R. Nysmith. Projectile size effects on hypervelocity impact craters in aluminium. Technical Report TN D-4067, NASA, July 1967.
- [2] W.P. Schonberg, A.J. Bean, and K. Darzi. Hypervelocity impact physics. Technical Report CP 4343, NASA, January 1991.
- [3] H. Fair. Hypervelocity then and now. *International Journal of Impact Engineering*, 5:1-11, 1987.
- [4] W.J. Rae. Analytical studies of impact-generated shock propagation. In R. Kinslow, editor, *High-Velocity Impact Phenomena*, pages 213-291. Academic Press, 1970.
- [5] F. Hörz, M.J. Cintala, R.P. Bernhard, and T.H. See. Dimensionally scaled penetration experiments to extract projectile sizes from space exposed surfaces. *International Journal of Impact Engineering*, 14:347-358, 1993.
- [6] B.P. Denardo. Projectile shape effects on hypervelocity impact craters in aluminium. Technical Report TN D-4953, NASA, December 1968.
- [7] P.B. Burbank, B.G. Cour-Palais, and W.E. McAllum. A meteoroid environment for near-earth, cislunar, and near-lunar operations. Technical Report TN D-2747, NASA, April 1965.
- [8] W.M. Isbell and W.J. Tedeschi. Hypervelocity research and the growing problem of space debris. *International Journal of Impact Engineering*, 14:359-372, 1993.
- [9] G.M. Levin and E.J. Phillips. The orbital debris program of the United States. In *Proceedings of the Second European Conference on Space Debris*, pages 3-7. ESA, 1997. SP-393.
- [10] V. Blagun, S. Kulik, and V. Lukyashenko. Activity of the Russian space agency to solve the space debris problem. In *Proceedings of the Second European Conference on Space Debris*, pages 21-23. ESA, 1997. SP-393.
- [11] W. Flury. European activities on space debris. In *Proceedings of the Second European Conference on Space Debris*, pages 35-41. ESA, 1997. SP-393.
- [12] F. Alby, E. Lansard, and T. Michal. Collision of Cerise with space debris. In *Proceedings of the Second European Conference on Space Debris*, pages 589-596. ESA, 1997. SP-393.

- [13] R.D. Caswell, N. McBride, and A. Taylor. OLYMPUS end of life anomaly - a Perseid meteoroid impact event? *International Journal of Impact Engineering*, 17:139–150, 1995.
- [14] K.L. Bedingfield, R.D. Leach, and M.B. Alexander. Spacecraft system failures and anomalies attributed to the natural space environment. Technical Report RP-1390, NASA, August 1996.
- [15] F.L. Whipple. On meteoroids and penetration. *Journal of Geophysical Research*, 68(17):4929–4939, 1963.
- [16] B.G. Cour-Palais and J.L. Crews. A multi-shock concept for spacecraft shielding. *International Journal of Impact Engineering*, 10:135–146, 1990.
- [17] E.L. Christiansen. Advanced meteoroid and debris shield concepts. AIAA 90-1336, 1990.
- [18] T.D. Maclay, R.D. Culp, L. Bareiss, T.G. Gillespie, and P.M. Kustas. Topographically modified bumper concepts for spacecraft shielding. *International Journal of Impact Engineering*, 14:479–489, 1993.
- [19] D.W. Bogdanoff. Optimisation study of the Ames 0.5" two-stage light gas gun. *International Journal of Impact Engineering*, 20:131–142, 1997.
- [20] J. Bol and W. Fücke. Shaped charge technique for hypervelocity impact tests at 11 km/s on space debris protection shields. In *Proceedings of the Second European Conference on Space Debris*, pages 405–410. ESA, 1997. SP-393.
- [21] A. Geille. Status of development of space-debris hypervelocity explosive multi-stage launcher. *International Journal of Impact Engineering*, 20:271–279, 1997.
- [22] P.R. Ratcliff, M.J. Cole, J.A.M. McDonnell, and H.A. Shaw. Experimental determination of energy partitioning in microparticle impacts at velocities from 1 to 100 kms⁻¹. IAF 95-I.5.03, 1995.
- [23] C.E. Anderson. An overview of the theory of hydrocodes. *International Journal of Impact Engineering*, 5:33–59, 1987.
- [24] J.A. Zukas, editor. *High Velocity Impact Dynamics*. John Wiley & Sons, New York, 1990.
- [25] D.J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99:235–394, 1992.
- [26] W.E. Johnson and C.E. Anderson. History and application of hydrocodes in hypervelocity impact. *International Journal of Impact Engineering*, 5:423–439, 1987.
- [27] J.C. Schultz, O.E.R. Heimdahl, and S.A. Finnegan. Computer characterisation of debris clouds resulting from hypervelocity impact. *International Journal of Impact Engineering*, 5:577–584, 1987.

- [28] M.E. Kipp, D.E. Grady, and J.W. Swegle. Numerical and experimental studies of high-velocity impact fragmentation. *International Journal of Impact Engineering*, 14:427–438, 1993.
- [29] M.L. Alme and C.E. Rhoades, Jr. A computational study of projectile melt in impact with typical Whipple shields. *International Journal of Impact Engineering*, 17:1–12, 1995.
- [30] S.A. Mullin, D.L. Littlefield, C.E. Anderson, and L.C. Chhabildas. An examination of velocity scaling. *International Journal of Impact Engineering*, 17:571–581, 1995.
- [31] E.L. Christiansen, J.L. Crews, J.E. Williamsen, J.H. Robinson, and A.M. Nolen. Enhanced meteoroid and orbital debris shielding. *International Journal of Impact Engineering*, 17:217–228, 1995.
- [32] J.H. Kerr and E.P. Fahrenthold. Three dimensional hypervelocity impact simulation for orbital debris shield design. *International Journal of Impact Engineering*, 20:479–489, 1997.
- [33] L.E. Malvern. *Introduction to the mechanics of a continuous medium*. Prentice-Hall, Inc., 1969.
- [34] L.D. Libersky, A.G. Petschek, T.C. Carney, J.R. Hipp, and F.A. Allahdadi. High strain Lagrangian hydrodynamics. *Journal of Computational Physics*, 109:67–75, 1993.
- [35] J.W. Swegle, S.W. Attaway, M.W. Heinstein, F.J. Mello, and D.L. Hicks. An analysis of Smoothed Particle Hydrodynamics. Technical Report SAND93-2513, Sandia National Laboratory, March 1994.
- [36] J.M. McGlaun, S.L. Thompson, and M.G. Elrick. CTH: A three-dimensional shock wave physics code. *International Journal of Impact Engineering*, 10:351–360, 1990.
- [37] R.G. Whirley and B.E. Engelmann. DYNA3D a nonlinear explicit, three-dimensional finite element code for solid and structural mechanics - user manual. Technical Report UCRL-MA-107254, Lawrence Livermore National Laboratory, November 1993.
- [38] J. Hallquist. Theoretical manual for DYNA3D. Technical Report UCID 19401, Lawrence Livermore National Laboratory, March 1983.
- [39] J. Hallquist. LS-DYNA3D theoretical manual. Technical report, Livermore Software Technology Corporation, 1994.
- [40] G.R. Johnson and R.A. Stryk. Eroding interface and improved tetrahedral element algorithms for high-velocity impact computations in three dimensions. *International Journal of Impact Engineering*, 5:411–421, 1987.

- [41] J.D. Frey, F. Janicot, X. Garaud, P. Groenenboom, and M. Lambert. The validation of hydrocodes for orbital debris impact simulation. *International Journal of Impact Engineering*, 14:255–265, 1993.
- [42] J.C.A. Campbell. Numerical simulation of hypervelocity impact on spacecraft. Master's thesis, Cranfield University, 1994.
- [43] D.J. Steinberg, S.G. Cochran, and M.W. Guinan. A constitutive model for metals applicable at high-strain rate. *Journal of Applied Physics*, 51(3):1498–1504, 1980.
- [44] G.R. Johnson and W.H. Cook. A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures. In *Proceedings of the Seventh International Symposium on Ballistics*, pages 541–547, 1983.
- [45] D. Townsend and N. Park. A tensile testing technique for obtaining materials properties at strain rates up to $5 \times 10^4 \text{ s}^{-1}$. Sowerby Research Centre, British Aerospace Plc, Filton, Bristol.
- [46] S.P. Marsh, editor. *LASL Shock Hugoniot Data*. University of California Press, 1980.
- [47] A.J. Piekutowski. Characteristics of debris clouds produced by hypervelocity impact of aluminium spheres with thin aluminium plates. *International Journal of Impact Engineering*, 14:573–586, 1993.
- [48] J.R. Asay and G.I. Kerley. The response of materials to dynamic loading. *International Journal of Impact Engineering*, 5:69–99, 1987.
- [49] D.J. Steinberg and C.M. Lund. A constitutive model for strain rates from 10^{-4} to 10^6 s^{-1} . *Journal of Applied Physics*, 65(4):1528–1533, 1989.
- [50] T.J. Holmquist and G.R. Johnson. Determination of constants and comparison of results for various constitutive models. *Journal de Physique IV*, 1:C3–853 – C3–860, 1991.
- [51] F.J. Zerilli and R.W. Armstrong. Dislocation-mechanics-based constitutive relations for material dynamics calculations. *Journal of Applied Physics*, 61(5):1816–1825, 1987.
- [52] T.G. Trucano and J.M. McGlaun. Hypervelocity impact calculations using CTH: Case studies. *International Journal of Impact Engineering*, 10:601–613, 1990.
- [53] S.P. Lyon and J.D. Johnson. SESAME: The Los Alamos national laboratory equation of state database. Technical Report LA-UR-92-3407, Los Alamos National Laboratory, 1992.
- [54] V. Hohler, H. Nahme, R. Tham, A. Stilp, K. Weber, Y.K. Yeh, J.P. Wang, and T.S. Lee. Experimental and numerical simulation of high velocity impact on steel targets. *International Journal of Impact Engineering*, 14:325–334, 1993.

- [55] C.J. Hayhurst, H.J. Ranson, D.J. Gardner, and N.K. Birnbaum. Modelling of microparticle hypervelocity oblique impacts on thick targets. *International Journal of Impact Engineering*, 17:375–386, 1995.
- [56] E.P. Fahrenthold. A Lagrangian model for debris cloud dynamics simulation. *International Journal of Impact Engineering*, 14:229–240, 1993.
- [57] MacNeal-Schwendler Corporation. *MSC/DYTRAN User's Manual. Version 2.3*, April 1995.
- [58] W. Benz. Smooth Particle Hydrodynamics: A review. In J.R. Buchler, editor, *The Numerical Modelling of Nonlinear Stellar Pulsations*, pages 269–288. Kluwer Academic Publishers, 1990.
- [59] W.K. Liu, S. Jun, S. Li, J. Adee, and T. Belytschko. Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering*, 38:1655–1679, 1995.
- [60] T. Belytschko, Y.Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical methods in Engineering*, 37:229–256, 1994.
- [61] G.R. Johnson. Linking of Lagrangian particle methods to standard finite element methods for high velocity impact computations. *Nuclear Engineering and Design*, 150:265–274, 1994.
- [62] K.S. Holian and M.W. Burkett. Sensitivity of hypervelocity impact simulations to equation of state. *International Journal of Impact Engineering*, 5:333–341, 1987.
- [63] A.C. Mitchell, W.J. Nellis, J.A. Moriarty, R.A. Heinle, N.C. Holmes, R.E. Tipton, and G.W. Repp. Equation of state of Al, Cu, Mo, and Pb at shock pressures up to 2.4 TPa (24 Mbar). *Journal of Applied Physics*, 69(5):2981–2986, 1991.
- [64] F.P. Stecher and G.R. Johnson. Lagrangian computations for projectile penetration into thick plates. In W.A. Gruver, editor, *Computers in Engineering*, volume 2. ASME, 1984.
- [65] T. Belytschko and J.I. Lin. A three-dimensional impact-penetration algorithm with erosion. *Computers and Structures*, 25(1):95–104, 1987.
- [66] J.J. Pyun, C.M. Kennedy, and D. Hruska. A new slideline/eroding algorithm for EPIC2. *International Journal of Impact Engineering*, 10:473–482, 1990.
- [67] G.R. Johnson and W.H. Cook. Lagrangian EPIC code computations for oblique, yawed-rod impacts onto thin-plate and spaced-plate targets at various velocities. *International Journal of Impact Engineering*, 14:373–383, 1993.
- [68] M.N. Raftenberg. Lagrangian hydrocode simulations of rolled-homogeneous-armor plate perforation by a shaped charge jet. *International Journal of Impact Engineering*, 15(5):619–643, 1994.

- [69] R.B. Pond and C.M. Glass. Metallurgical observations and energy partitioning. In R. Kinslow, editor, *High-Velocity Impact Phenomena*, pages 419–461. Academic Press, 1970.
- [70] A.J. Watts and D. Atkinson. Dimensional scaling for impact cratering and perforation. *International Journal of Impact Engineering*, 17:925–935, 1995.
- [71] *Aerospace Structural Metals Handbook Vol. 3*, 1978. Mechanical Properties Data Center, Department of Defence.
- [72] R.A. Gingold and J.J. Monaghan. Smoothed Particle Hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.
- [73] L.B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82(12):1013–1024, 1977.
- [74] J.J. Monaghan. Why particle methods work. *SIAM Journal on Scientific and Statistical Computing*, 3(4):422–433, 1982.
- [75] J.J. Monaghan and H. Pongracic. Artificial viscosity for particle methods. *Applied Numerical Mathematics*, 1:187–194, 1985.
- [76] J.J. Monaghan. On the problem of penetration in particle methods. *Journal of Computational Physics*, 82:1–15, 1989.
- [77] J.J. Monaghan. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574, 1992.
- [78] L.D. Libersky and A.G. Petschek. Smooth Particle Hydrodynamics with strength of materials. In H.E. Trease, M.J. Fritts, and W.P. Crowley, editors, *Advances in the Free-Lagrange Method*, pages 248–257. Springer-Verlag, 1990.
- [79] S.W. Attaway, M.W. Heinstein, and J.W. Swegle. Coupling of Smooth Particle Hydrodynamics with the finite element method. *Nuclear Engineering and Design*, 150:199–205, 1994.
- [80] Y. Wen, D.L. Hicks, and J.W. Swegle. Stabilising S.P.H. with conservative smoothing. Technical Report SAND94-1932, Sandia National Laboratory, August 1994.
- [81] G.R. Johnson, R.A. Stryk, and S.R. Beissel. SPH for high velocity impact computations. *Computer Methods in Applied Mechanics and Engineering*, 139:347–373, 1996.
- [82] P.W. Randles and L.D. Libersky. Smoothed Particle Hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:375–408, 1996.
- [83] J.J. Monaghan and J.C. Lattanzio. A refined particle method for astrophysical problems. *Astronomy and Astrophysics*, 149(1):135–143, 1985.

- [84] G.R. Johnson and S.R. Beissel. Normalised smoothing functions for SPH impact computations. *International Journal for Numerical Methods in Engineering*, 39:2725–2741, 1996.
- [85] J.J. Monaghan and R.A. Gingold. Shock simulation by the particle method SPH. *Journal of Computational Physics*, 52:374–389, 1983.
- [86] P.M. Campbell. Some new algorithms for boundary value problems in Smooth Particle Hydrodynamics. Technical Report DNA-TR-88-286, Mission Research Corporation, June 1989.
- [87] G.R. Johnson. Artificial viscosity effects for SPH impact computations. *International Journal of Impact Engineering*, 18(5):477–488, 1996.
- [88] L. Hernquist and N. Katz. TREESPH: A unification of SPH with the hierarchical tree method. *The Astrophysical Journal Supplement Series*, 70:419–446, 1989.
- [89] R.W. Hockney and J.W. Eastwood. *Computer Simulation using Particles*. McGraw-Hill, 1981.
- [90] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324:446–449, December 1986.
- [91] Z. Meglicki. *Analysis and Applications of Smoothed Particle Magnetohydrodynamics*. PhD thesis, Australian National University, July 1995.
- [92] C. Guenther, D.L. Hicks, and J.W. Swegle. Conservative smoothing versus artificial viscosity. Technical Report SAND94-1853, Sandia National Laboratory, August 1994.
- [93] C.T. Dyka and R.P. Ingel. An approach for tension instability in Smoothed Particle Hydrodynamics (SPH). *Computers and Structures*, 57(4):573–580, 1995.
- [94] D.S. Balsara. von Neumann stability analysis of Smoothed Particle Hydrodynamics - suggestions for optimal algorithms. *Journal of Computational Physics*, 121:357–372, 1995.
- [95] L.D. Libersky, 1997. Private Communication.
- [96] P. Lancaster and K. Salkauskas. *Curve and Surface Fitting An Introduction*. Academic Press, 1986.
- [97] G.A. Dilts. Moving-least squares particle hydrodynamics I. Consistency and stability. Technical Report LAUR 97-4168, Los Alamos National Laboratory, October 1997.
- [98] T. Belytscko and M. Neal. Contact-impact by the pinball algorithm with penalty and Lagrangian methods. *International Journal for Numerical Methods in Engineering*, 31:547–572, 1991.

- [99] T. Belytschko and I. Yeh. The splitting pinball method for contact-impact problems. *Computer Methods in Applied Mechanics and Engineering*, 105:375–393, 1993.
- [100] W.C. Young. *Roark's Formulas for Stress and Strain*. McGraw-Hill, 6th edition, 1989.

Appendix A

DYNA3D HEX8 element subroutines

In this appendix detailed notes are given for the 8 node brick element subroutines. The subroutine notes are arranged in the order that they are called by the DYNA program. As there are many available constitutive models, no strength model or equation of state specific subroutine is described in detail. The individual subroutines described are those required when using a constitutive model that requires a separate equation of state, and using the DYNA hourglass viscosity. Some different subroutines are called when using the Flanagan-Belytschko hourglass viscosity but these differ from the routine described only in detail, not in purpose.

Variables are denoted by the type font: `variable name`. As are common block names.

Subroutine SOLDE

This subroutine is the main subroutine for 8 node brick elements. It is called once each time step, and processes every 8 node brick element.

This subroutine is called by subroutine SOLIDE, which is in turn called by subroutine FEM3D, the main solution subroutine. The only operation carried out by subroutine SOLIDE is to call subroutine SOLDE.

The elements are processed in groups. The size of each group is set at the time the code is compiled, and varies between different computers. For example on a Sun workstation each group is 32 elements, on a Cray J916 each is 128 elements. This is done in order to vectorise the code, and signifies the length of the vectors used. The total number of HEX8 element groups is given by `nelg`, which is calculated during the initialisation phase.

The element connectivity data for the group of elements being processed is loaded in subroutine UNPKI.

If necessary the group of elements are processed in subgroups. This is done if elements requiring different material models are in the group being processed. For each group there are `nsubg` subgroups. All further subroutines are called once for each group

or subgroup, and all elements in the group or subgroup being processed belong to the same DYNA material number.

For the group being processed, basic material information is loaded, this includes the material model type, `mt_e`, the equation of state type, `nes`, and the hourglass stabilisation method, `ihg`.

Subroutine `LOCFLH` is called if element erosion is being used. This loads the array that lists the eroded elements.

Depending on the hourglass stabilisation method either subroutine `STRAIN` or subroutine `STRNFB` is called. Here the element Jacobian, volume, strain rate tensor and spin tensor are calculated.

Subroutine `FELEN` is called, which calculates the area of the largest side for each element. This is used in the time step calculation.

The next series of subroutine calls are dependent on the material model used. The routines called when using a strength model requiring a separate equation of state will be described.

Subroutine `HVPAC1` unpacks the element stress tensor and auxiliary variables from the database.

Subroutine `RSTRSS` calculates the rotated stress tensor.

The strength model specific subroutine is then called. The routines are called `F3DMxx`, where `xx` is the material type number. `F3DM11` is the Steinberg-Guinan strength model routine, `F3DM15` is the Johnson-Cook strength model routine.

Subroutine `SUEOS` calls an equation of state specific routine that calculates the element speed of sound.

Subroutine `BULKQ` calculates the critical time step size and the bulk viscosity.

Subroutine `ENGBRK` calculates the momentum and kinetic energy for book-keeping purposes.

Subroutine `HIEUPD` calculates the trial value of the internal energy.

Subroutine `EQOS` calls the equation of state specific subroutine that the new pressure and internal energy.

Subroutine `HVPAC2` packs the new element stress tensor, and auxiliary variables.

Subroutine `UPDFLH` is called if element erosion is being used. This updates the array listing the eroded elements.

The hourglass forces are calculated. Depending on the stabilisation method being used different subroutines are called. Subroutine HRGMD calculates the hourglass forces for the standard viscosity.

Subroutine FORCE carries out the element integration, adding the contributions from the element stress and hourglass forces to the global nodal force vector.

Subroutine UNPKI

Purpose: To unpack the element connectivity data

Variables: array ixp in the call statement contains the data to be unpacked. The connectivity data is passed as arrays $ix1(i)$ to $ix8(i)$ as the first eight entries in common block $aux33$. The material number, $mxt(i)$ is the ninth entry.

Notes: The connectivity data for element i consists of the numbers of the vertex nodes. The orientation of the element is defined by the order of the node numbers, see the user manual [37].

Subroutine STRAIN

Purpose: To calculate the rate of deformation and spin tensors, and the volumetric strain rate.

Calls: PRTAL

Variables: Arrays x and v in the call statement contain nodal positions and velocities. The rate of deformation tensor is passed as the first six entries in common block $aux2$, the spin tensor as the seventh to ninth entries. The volumetric strain rate is the first entry in common block $aux18$. The ratio of current to initial element volume is passed as the 2nd entry in common block $aux18$.

Notes: First the nodal co-ordinates are gathered into the local arrays $x1(i)$, $y1(i)$, $z1(i)$ to arrays $x8(i)$, $y8(i)$, $z8(i)$, and the nodal velocities are gathered into the local arrays $vx1(i)$, $vy1(i)$, $vz1(i)$ to arrays $vx8(i)$, $vy8(i)$, $vz8(i)$.

Then subroutine PRTAL is called, where the Jacobian determinant and the strain-displacement matrix $\mathbf{B}$ is calculated.

The rate of deformation tensor, $\dot{\epsilon}^{\alpha\beta}$, and the spin tensor, $\dot{\Omega}^{\alpha\beta}$, are then calculated.

$$\dot{\epsilon}^{\alpha\beta} = \frac{1}{2} \left[\frac{\partial v^\alpha}{\partial x^\beta} + \frac{\partial v^\beta}{\partial x^\alpha} \right]$$

$$\dot{\Omega}^{\alpha\beta} = \frac{1}{2} \left[\frac{\partial v^\alpha}{\partial x^\beta} - \frac{\partial v^\beta}{\partial x^\alpha} \right]$$

where

$$\frac{\partial v^\alpha}{\partial x^\beta} = \sum_{k=1}^8 \frac{\partial \phi_k}{\partial x^\beta} v_k^\alpha$$

$\frac{\partial \phi_k}{\partial x^\beta}$ are the components of **B** calculated in PRTAL, and are the variables px1 (i) to px8 (i).

The variables actually calculated are:

$$\begin{aligned} dxx &= \dot{\epsilon}^{xx} & dyy &= \dot{\epsilon}^{yy} & dzz &= \dot{\epsilon}^{zz} \\ d1 &= 2\dot{\epsilon}^{xy} & d2 &= 2\dot{\epsilon}^{yz} & d3 &= 2\dot{\epsilon}^{zx} \\ wxxdt &= \dot{\Omega}^{yz} dt & wyydt &= \dot{\Omega}^{zx} dt & wzzdt &= \dot{\Omega}^{xy} dt \end{aligned}$$

The volumetric strain rate is calculated as $d\bar{d} = -\dot{\epsilon}^{\alpha\alpha}$.

The ratio of the current to initial element volume is calculated as dfe

Subroutine PRTAL

Purpose: Calculate Jacobian, element volume and strain-displacement matrix

Variables: The Jacobian is aij1 (i) to aij9 (i), and is passed as the 4th through 12th entries in aux32. The element volume, det (i), is the second entry in aux9. The strain displacement matrix is px1 (i) through pz8 (i) and is passed as the first 24 entries in aux10.

Notes: The components of the Jacobian are:

$$\frac{\partial x^\alpha}{\partial \xi^\beta} = \sum_{i=1}^8 \frac{\partial \phi_i}{\partial \xi^\beta} x_i^\alpha$$

where ϕ_i are the element shape functions and ξ are the iso-parametric co-ordinates.

The element volume is approximated as $\det (i) = 8|J(0, 0, 0)|$.

The inverse of the Jacobian matrix is calculated as aji1 (i) to aji9i.

The twenty four components of the strain displacement matrix is then calculated:

$$\mathbf{B} = \begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} \\ \frac{\partial \phi_i}{\partial \eta} \\ \frac{\partial \phi_i}{\partial \zeta} \end{bmatrix} \quad (\text{A.1})$$

Due to the 1 point spatial integration **B** is anti-symmetric:

$$\frac{\partial \phi_1}{\partial x^\alpha} = -\frac{\partial \phi_7}{\partial x^\alpha} \quad \frac{\partial \phi_3}{\partial x^\alpha} = -\frac{\partial \phi_5}{\partial x^\alpha} \quad \frac{\partial \phi_2}{\partial x^\alpha} = -\frac{\partial \phi_8}{\partial x^\alpha} \quad \frac{\partial \phi_4}{\partial x^\alpha} = -\frac{\partial \phi_6}{\partial x^\alpha}$$

Only $px1(i)$ to $px4(i)$, $py1(i)$ to $py4(i)$ and $pz1(i)$ to $pz4(i)$ are directly calculated.

Subroutine FELEN

Purpose: Calculate an approximation of the element critical length.

Variables: The approximation of the critical length, $dx(i)$, is the second entry in common block `aux34`.

Notes: The area of each of the six element sides is calculated. The actual value calculated, `atest`, is $16A^2$.

The critical length is:

$$dx = \frac{V}{A_{MAX}}$$

where V is the element volume calculated in subroutine `PRTAL`,

Subroutine HVPAC1

Purpose: To unpack the stress tensor and the auxiliary variables for each element in the group or subgroup being processed.

Variables: array `aux` in the call statement contains the auxiliary variables to be unpacked. The unpacked variables are passed in common block `aux14`. The first six entries are the components of the stress tensor, the remaining entries are the auxiliary variables required for the particular material model.

Notes: The number of auxiliary variables required for a material model is defined in subroutine `blkdat`.

Subroutine RSTRSS

Purpose: Rotate stress tensor, calculate pressure from stress tensor.

Variables: The stress tensor is `t1` to `t6` in common `aux14`. These values are replaced by the components of the rotated tensor. Copy of the un-rotated tensor is passed as the second to seventh entries in common block `aux11`. The pressure is passed as the first entry in `aux11`.

Notes: RSTRSS calculates $\sigma^{\alpha\beta} R^{\alpha\beta}$ where:

$$R^{\alpha\beta} = \sigma^{\alpha\gamma} \dot{\Omega}^{\gamma\beta} \Delta t + \sigma^{\beta\gamma} \dot{\Omega}^{\gamma\alpha} \Delta t$$

$\dot{\Omega}^{\alpha\beta} \Delta t$ are calculated in subroutine STRAIN.

DYNA does not store the pressure separately from time step to time step. This requires the pressure to be recalculated from the stress tensor:

$$P = -1/3 \sigma^{\alpha\alpha}$$

Subroutine SUEOS

Purpose: Call correct equation of state subroutine, which calculates the element speed of sound, c .

Variables: The specific equation of state subroutine calculates $\rho_0 c^2$ which is passed as the second entry in common block aux35.

Notes: The equation of state specific subroutines are SUEOSx where x is the equation of state type, for example SUEOS4 is the Gruneisen subroutine. The speed of sound is calculated using:

$$\rho_0 c^2 = \frac{4G}{3} + \rho_0 \left. \frac{\partial P}{\partial \rho} \right|_E + P V^2 \left. \frac{\partial P}{\partial E} \right|_\rho$$

Subroutine BULKQ

Purpose: Calculate critical time step, and bulk viscosity

Variables: The element critical time step, dt_{crit} , is passed as the fourth entry in common block aux37. The smallest element time step is dt_2 , which is the second entry in common block bk02. The bulk viscosity, $qp(i)$, is passed as the first entry in common block aux15.

Notes: The element critical time step is:

$$\Delta t_{crit} = \frac{\Delta x_{crit}}{Q + \sqrt{Q^2 + c^2}}$$

where

$$Q = Q_2 c + Q_1 |\dot{\epsilon}^{\alpha\alpha}| \Delta x_{crit}$$

and Δx_{crit} is the critical length from routine FELEN. Q_1 and Q_2 are the bulk viscosity coefficients.

Then the bulk viscosity is calculated:

$$q = \Delta x_{crit} |\dot{\epsilon}^{\alpha\alpha}| (\rho Q_1 \Delta x_{crit} |\dot{\epsilon}^{\alpha\alpha}| + \rho Q_2 c) \times \frac{V_0}{V}$$

V_0/V is the ratio of current to initial element volume calculated in subroutine STRAIN.

Subroutine ENGBRK

Purpose: Calculate the momentum and kinetic energy for each element

Notes: These values are not needed for the solution, but are used as checks on conservation.

Subroutine HIEUPD

Purpose: Calculate the trial value of the element internal energy.

Variables: HIEUPD requires the new deviatoric stress tensor, s^{n+1} , which is sig1(i) to sig6(i), and the old full stress tensor, σ^n , which is sigv1(i) to sigv6(i). The trial value of the internal energy is ep_{x2}(i), the current values of the bulk viscosity, $q^{n+1/2}$, and element volume, V^{n+1} , are read as ep_{x3}(i) and ep_{x4}(i) respectively. In order to be available in the following time step. The superscripts here refers to the time at which the value is calculated.

Notes: The trial value of the internal energy is:

$$e^* = e^n - \frac{1}{2} \left(P^n + q^{n-1/2} + q^{n+1/2} \right) \Delta V \\ + \frac{1}{4} \Delta t^{n+1/2} \left(V^n + V^{n+1} \right) \left(s^n + s^{n+1} \right) \dot{\epsilon}'^{n+1/2}$$

e1 to e6 are the components of $(s^n + s^{n+1}) \dot{\epsilon}'^{n+1/2}$,
and einc is $(V^n + V^{n+1}) (s^n + s^{n+1}) \dot{\epsilon}'^{n+1/2}$.

Subroutine EQOS

Purpose: Call correct equation of state subroutine

Variables: The specific equation of state subroutine calculates the pressure, p_{new}, which is passed as the ninth entry in common block aux15, and the internal energy, ep_{x2}, which is the eighth entry in common block aux14.

Notes: The equation of state specific subroutines are EQOSx. The specific subroutine

calculates the new pressure and energy from the density and the trial value of the internal energy.

The subroutine also calculates the full stress tensor, from the pressure and the deviatoric stress tensor.

Subroutine HVPAC2

Purpose: To pack the stress tensor and the auxiliary variables for each element in the group or subgroup being processed.

Variables: The first six entries of `aux14` are the components of the stress tensor, the remaining entries are the auxiliary variables required for the particular material model. The packed variables are passed in the call statement.

Subroutine HGRMD

Purpose: To calculate hourglass viscosity forces

Variables: The hourglass viscosity forces, `p11` to `p38`, are passed as the 25th to 48th entries in common block `aux18`.

Subroutine FORCE

Purpose: Perform element integration for stress and hourglass forces.

Variables: The global nodal force vector, `e`, is passed in the call statement.

Notes: The equations of motion at time n can be written:

$$\mathbf{M}\mathbf{a}^n = \mathbf{P}^n - \mathbf{F}^n + \mathbf{H}^n$$

$\mathbf{M}$ is the diagonal mass matrix, $\mathbf{P}$ accounts for the body forces and external loads, $\mathbf{F}$ is the stress divergence vector and $\mathbf{H}$ is the hourglass resistance.

FORCE calculates $\mathbf{H}^n - \mathbf{F}^n$.

First `sgv1` to `sgv6` are calculated, these are the components of:

$$\int_{V_e} \sigma dV$$

Then `e11` to `e34`, the components of $\mathbf{F}$ are calculated:

$$\mathbf{F} = \int_{V_e} \mathbf{B}^t \sigma dV$$

where **B** is the strain-displacement matrix calculated in subroutine PRTAL.

ep11 to ep38 are then calculated, these are the components of the nodal force vector at each node due to the hourglass resistance and the element stress. These components are then added to the global nodal force array. The global nodal force array is zeroed every time step in subroutine LOAD, and all the contributions to the nodal forces, such as contact, body forces and stress, are added to it.

Appendix B

Finite Element Results

B.1 Chapter 3

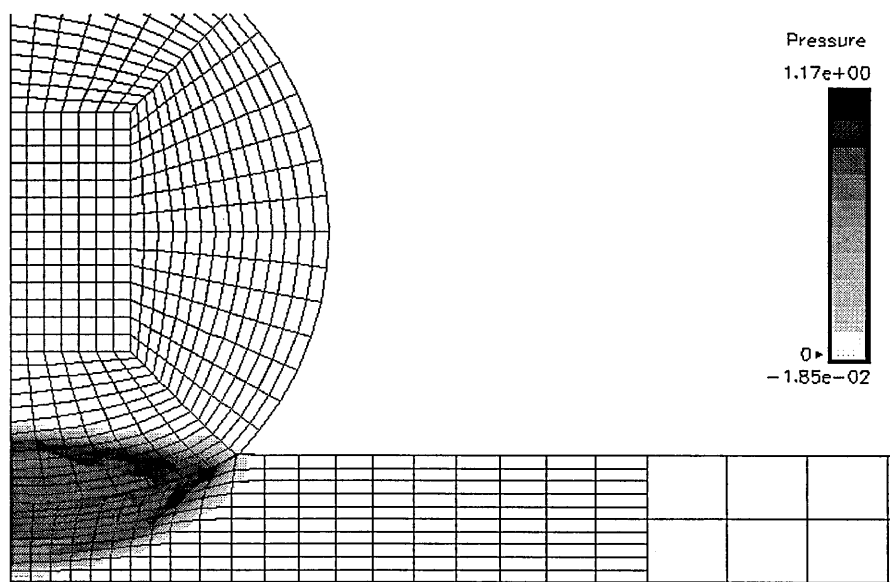


Figure B.1: Pressure contour plot, no element erosion. Time $0.075 \mu s$. Unmodified DYNA3D.

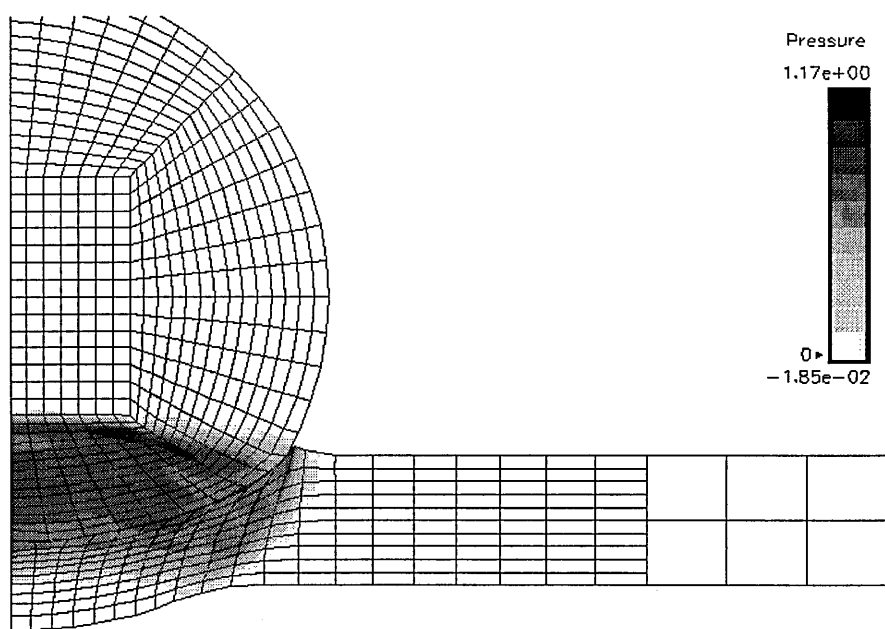


Figure B.2: Pressure contour plot, no element erosion. Time $0.125 \mu s$. Unmodified DYNA3D.

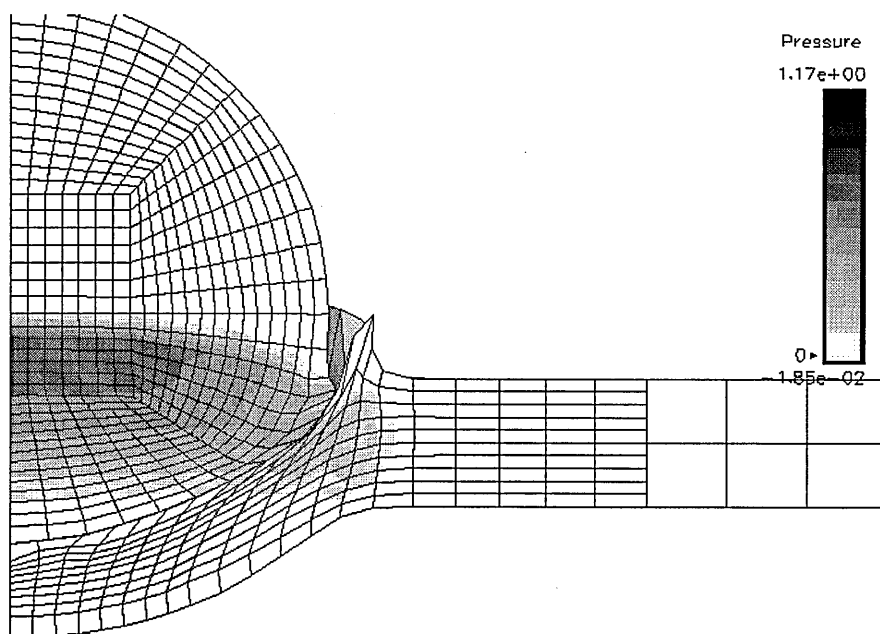


Figure B.3: Pressure contour plot, no element erosion. Time $0.2 \mu s$. Unmodified DYNA3D.

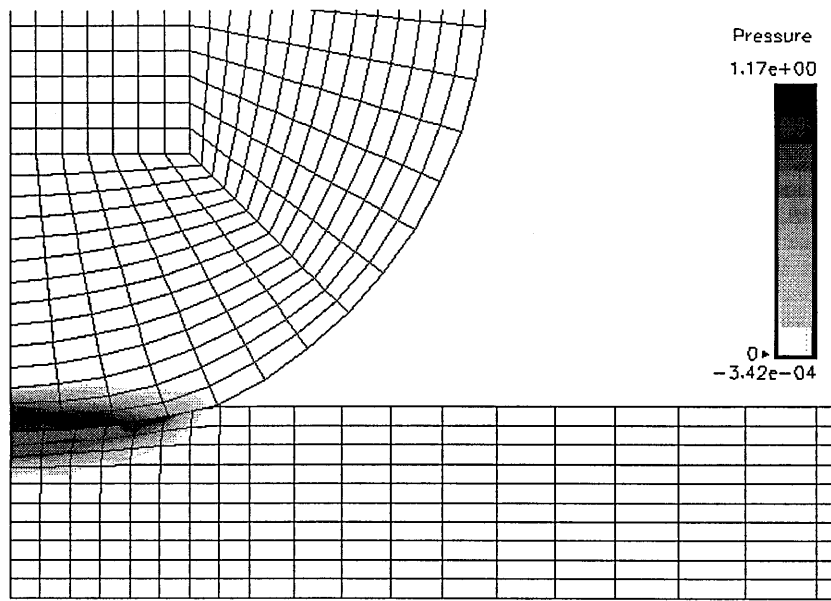


Figure B.4: Pressure contour plot, no element erosion. Time $0.025 \mu s$. Unmodified DYNA3D.

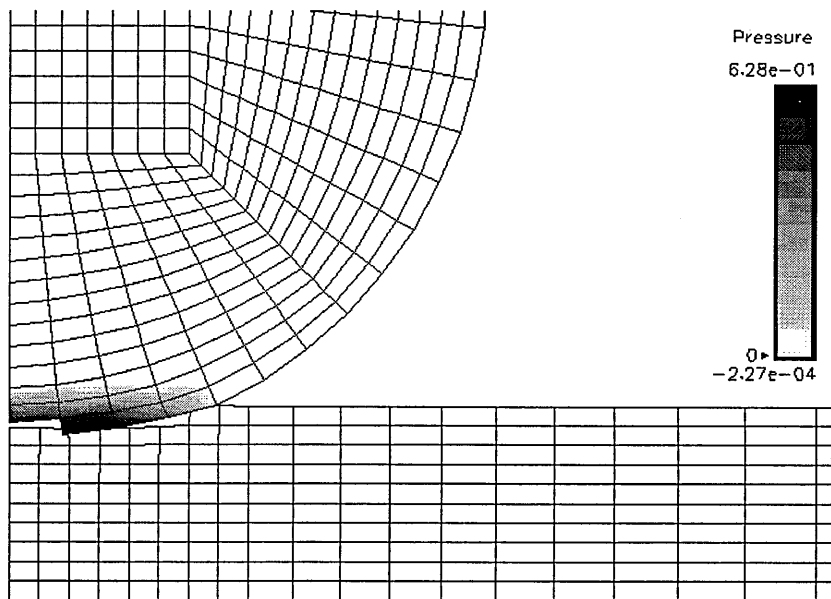


Figure B.5: Pressure contour plot, erosion criterion 0.5. Time $0.025 \mu s$. Unmodified DYNA3D.

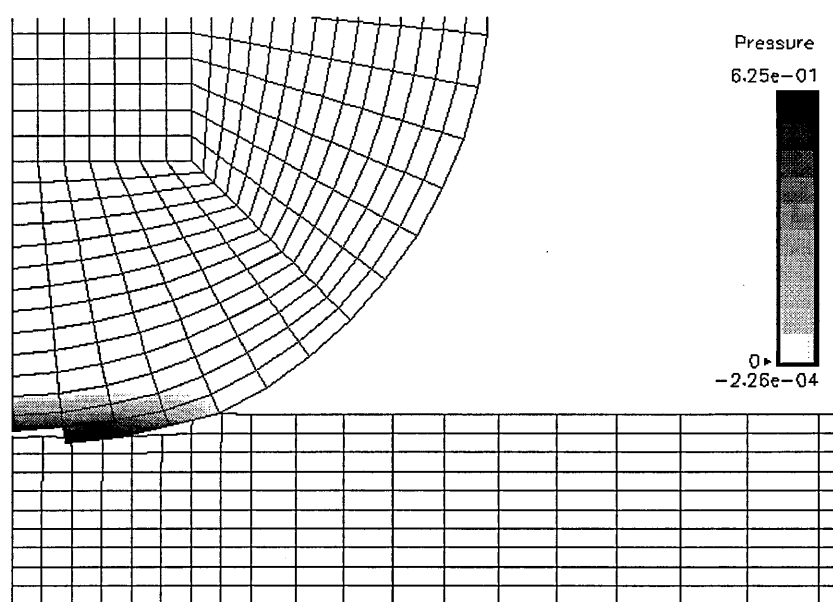


Figure B.6: Pressure contour plot, erosion criterion 1.0. Time $0.025 \mu s$. Unmodified DYNA3D.

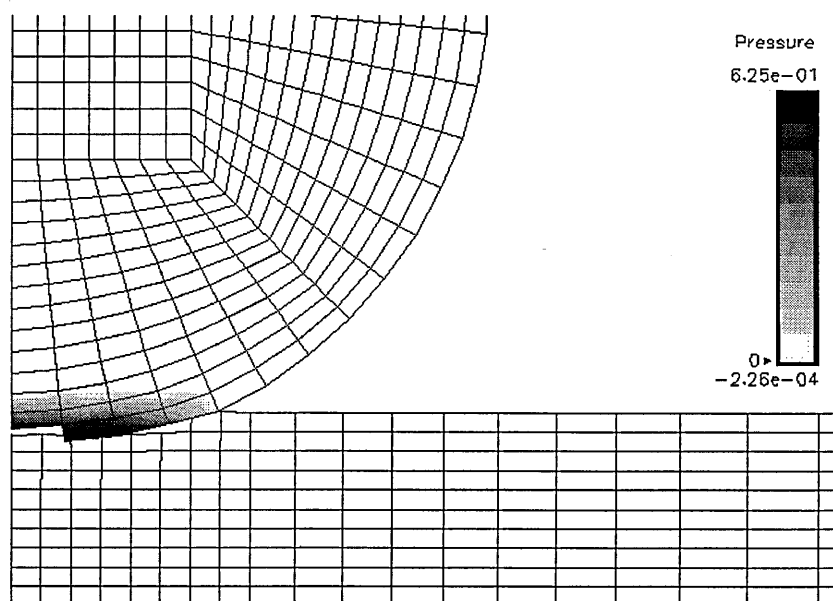


Figure B.7: Pressure contour plot, erosion criterion 1.5. Time $0.025 \mu s$. Unmodified DYNA3D.

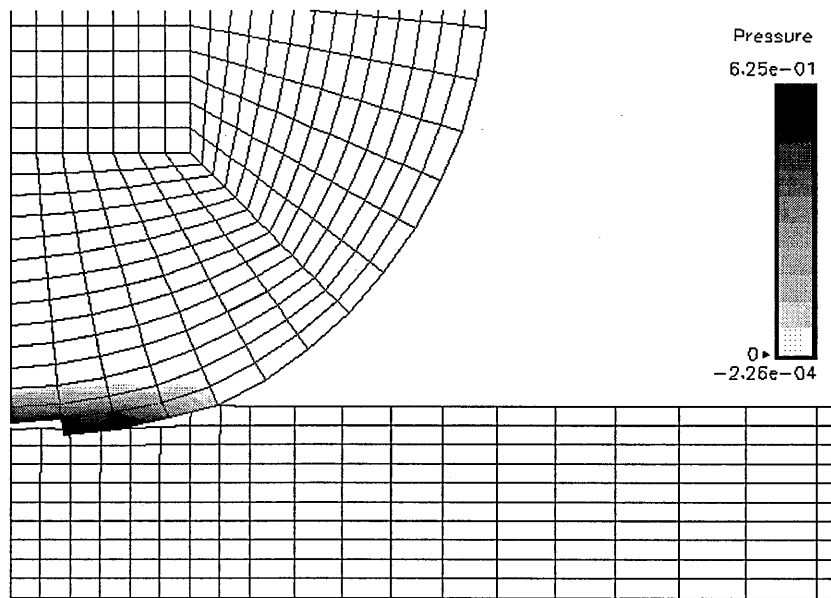


Figure B.8: Pressure contour plot, erosion criterion 2.5. Time $0.025 \mu s$. Unmodified DYNA3D.

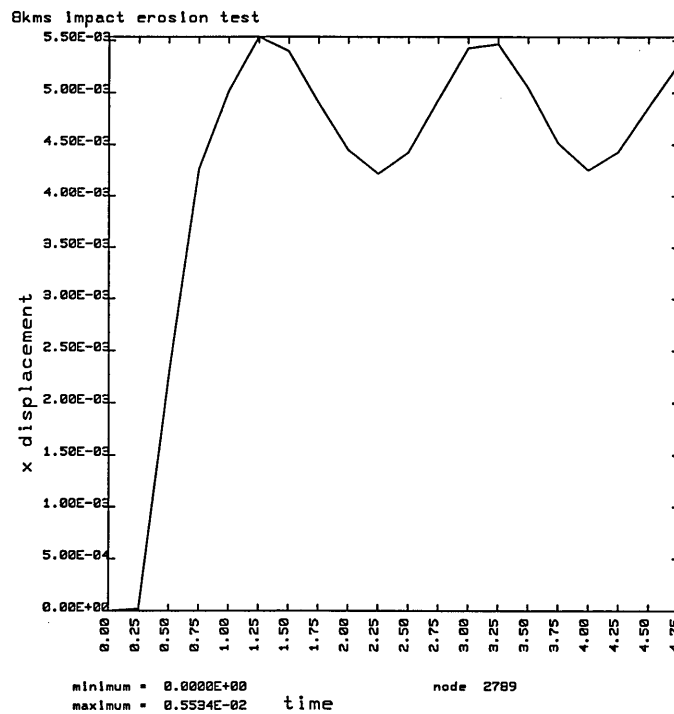
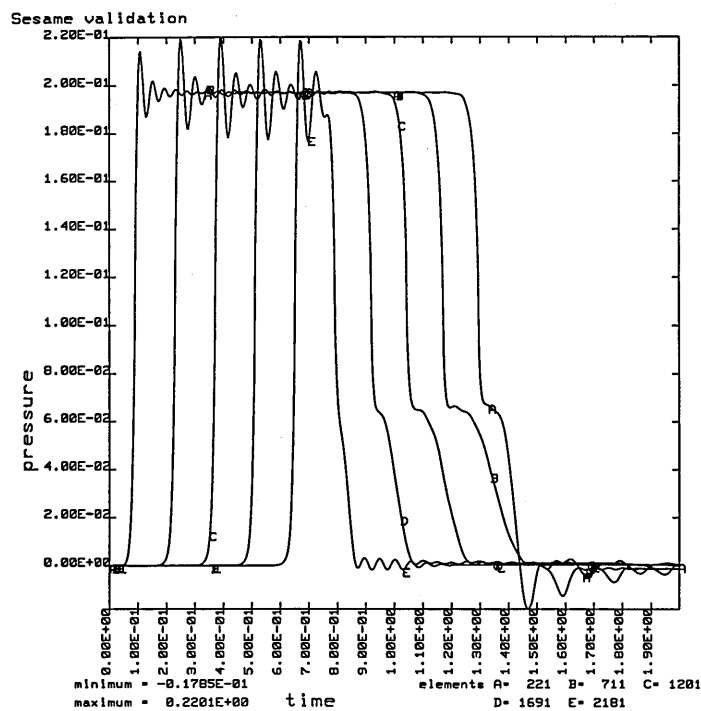
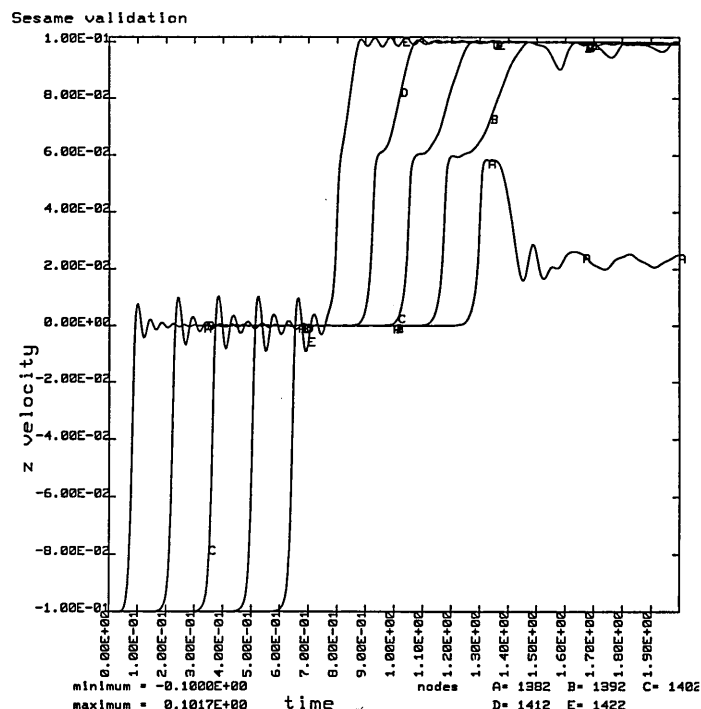


Figure B.9: Displacement time history for node on the edge of the final hole Unmodified DYNA3D.

B.2 Chapter 4



a: Pressure



b: Nodal Velocity

Figure B.10: Time histories for initial velocity 0.1 cm/ μ s . Modified DYNA3D.

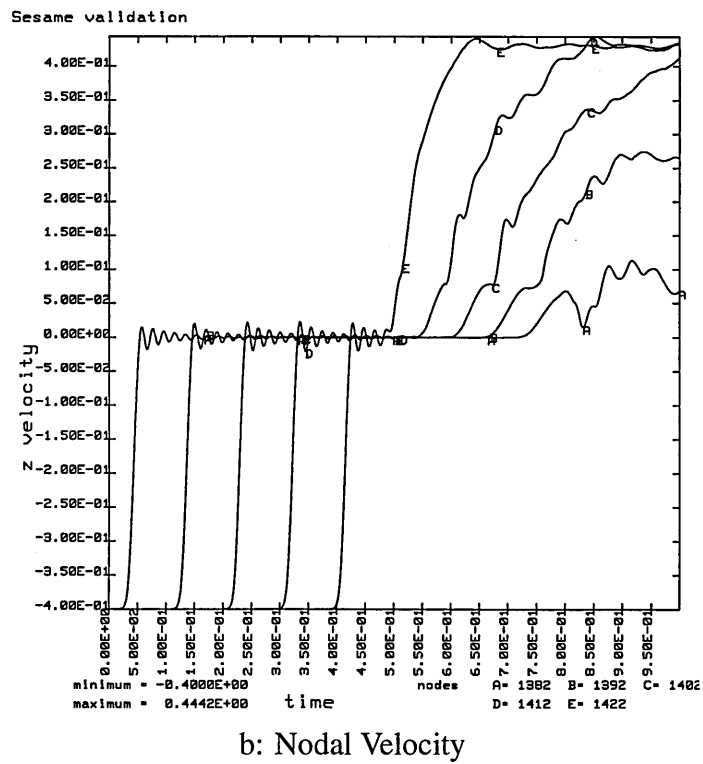
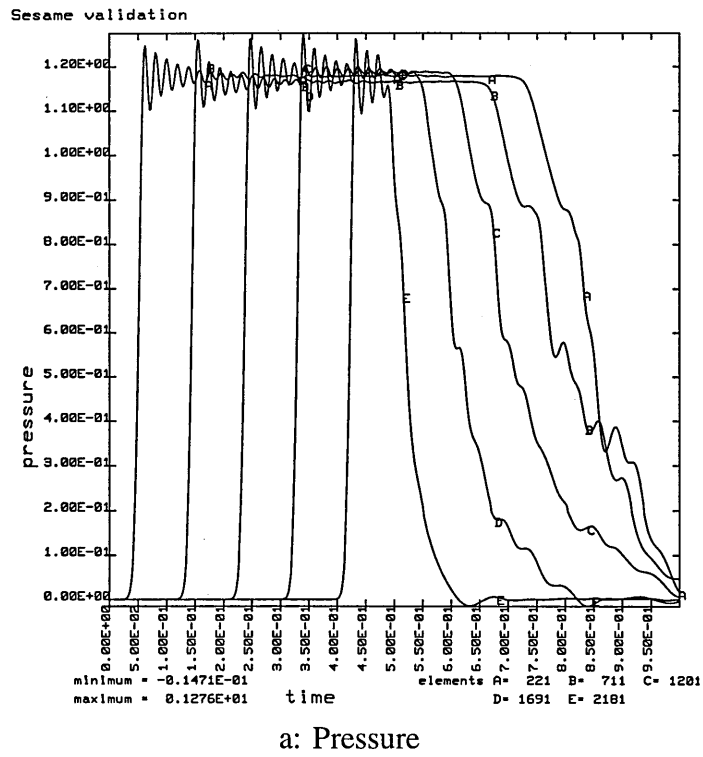


Figure B.11: Time histories for initial velocity $0.4 \text{ cm}/\mu\text{s}$. Modified DYNA3D.

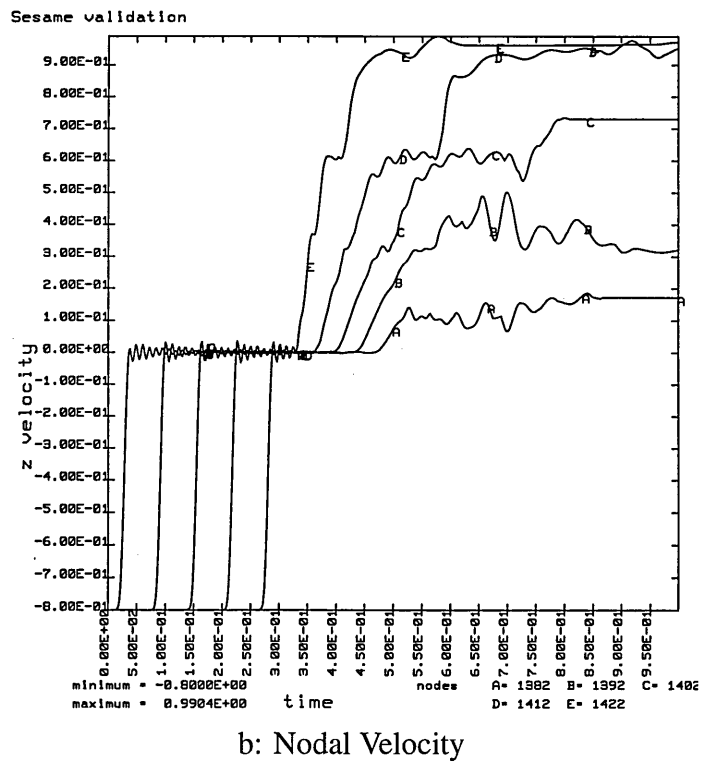
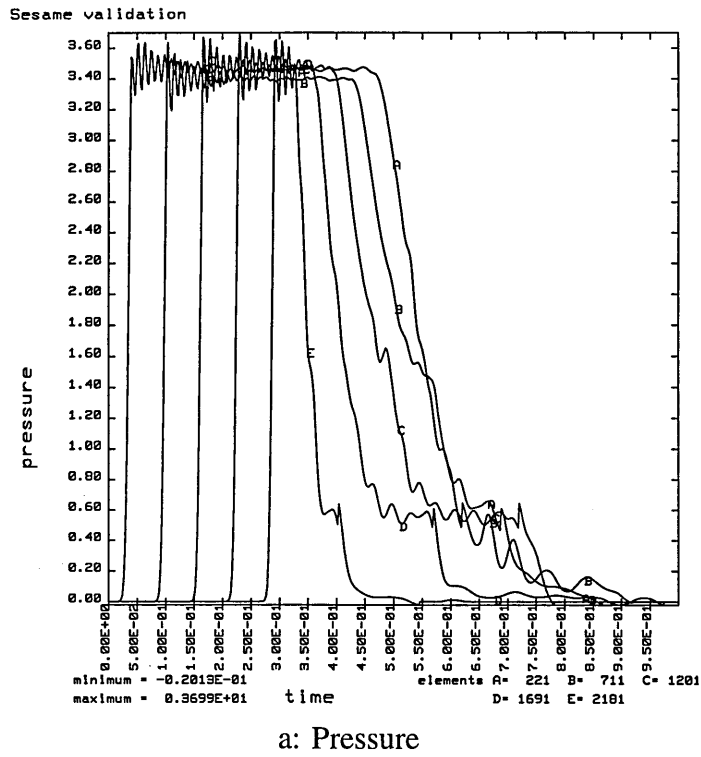


Figure B.12: Time histories for initial velocity $0.8 \text{ cm}/\mu\text{s}$. Modified DYNA3D.

B.3 Chapter 6

B.3.1 Tables

For the semi-infinte impact results, P is the crater depth, D is the crater diameter.

Projectile Erosion	Target Erosion	CPU to end (s)	Timesteps to end	P (mm)	D (mm)	$\frac{D}{P}$
1.0	1.0	5094	12997	6.67	10.70	1.61
1.5	1.5	8765	20406	6.84	12.46	1.82
1.75	1.75	17796	36638	7.27	12.82	1.76

Table B.1: Results for semi-infinite target. Effective plastic strain erosion criterion. Johnson-Cook strength model.

Projectile Erosion	Target Erosion	CPU to end (s)	Timesteps to end	P (mm)	D (mm)	$\frac{D}{P}$
1.0	1.0	4589	11400	5.62	11.51	2.05
1.5	1.5	6810	16474	7.54	13.18	1.75
2.0	2.0	11388	27883	8.19	14.57	1.78
2.5	2.5	18564	46676	8.31	15.27	1.84
3.0	3.0	28321	72039	8.79	15.22	1.73
1.5	2.0	11688	28839	7.58	13.44	1.77
2.5	2.0	11360	27521	7.98	14.89	1.87
2.0	1.5	6528	16070	7.25	14.11	1.95
2.0	2.5	17655	46543	7.68	15.29	1.99

Table B.2: Results for semi-infinite target. Deformation erosion criterion. Johnson-Cook strength model.

Projectile Erosion	Target Erosion	CPU to end (s)	Timesteps to end	P (mm)	D (mm)	$\frac{D}{P}$
1.0	1.0	4278	11288	6.08	11.47	1.89
1.5	1.5	7002	18711	7.22	12.85	1.78
2.0	2.0	10310	27365	7.73	14.27	1.85
2.5	2.5	16751	46181	7.94	14.80	1.86
3.0	3.0	25809	70576	8.35	14.73	1.76

Table B.3: Results for semi-infinite target. Deformation erosion criterion. Steinberg-Guinan strength model.

Projectile Erosion	Target Erosion	Skew Angle (radians)	CPU to end (s)	Timesteps to end	P (mm)	D (mm)	$\frac{D}{P}$
1.5	1.5	0.0	8884	16569	6.66	8.20	1.23
2.0	2.0	0.0	25827	48046	6.99	11.39	1.63
2.5	2.5	0.0	47953	92727	7.66	12.57	1.64
3.0	3.0	0.0	38266	68044	7.49	12.48	1.66
3.5	3.5	0.0	†	†	8.13	13.16	1.62
2.0	2.5	0.0	17240	31381	7.38	11.46	1.55
3.0	2.5	0.0	‡	‡	7.82	12.48	1.60
2.5	2.0	0.0	38783	72832	7.38	11.71	1.59
2.5	3.0	0.0	45540	80991	7.48	13.39	1.78

† Analysis crashed before end. Results are for 20 μ s .

‡ Analysis reached CPU time limit of 59000s. Results are for 16 μ s .

Table B.4: Results for semi-infinite target. Accuracy erosion criterion. Johnson-Cook strength model.

Projectile Erosion	Target Erosion	Skew Angle (radians)	CPU to end (s)	Timesteps to end	P (mm)	D (mm)	$\frac{D}{P}$
2.5	2.5	0.0	47953	92727	7.66	12.57	1.64
2.5	2.5	0.1	48584	92368	7.67	12.57	1.64
2.5	2.5	0.2	29506	55688	7.66	12.57	1.64
2.5	2.5	0.3	33497	56688	7.50	12.37	1.65
2.5	2.5	0.4	32933	54711	7.71	12.32	1.60
2.5	2.5	0.5	34863	57464	7.65	12.21	1.60

Table B.5: Results for semi-infinite target. Accuracy erosion criterion, varying only skew angle for erosion. Johnson-Cook strength model.

Shot ID	Impact Velocity (km/s)	Crater Depth (mm)	Crater Diameter (mm)	Diameter/ Depth
SP-700	3.28	3.66	8.51	2.33
SP-632	3.71	4.06	10.16	2.50
SP-637	3.76	4.11	9.65	2.35
SP-636	4.01	4.50	10.16	2.26
SP-699	4.33	4.55	10.57	2.32
SP-635	4.39	4.57	11.02	2.41
SP-534	4.75	4.78	10.39	2.18
SP-537	4.79	4.80	10.57	2.20
SP-532	5.50	5.33	11.86	2.22
SP-638	5.59	5.26	11.76	2.24
645	5.84	5.49	12.22	2.23
646	6.44	5.82	13.26	2.28
643	6.87	6.07	13.08	2.15
626	7.23	6.32	13.49	2.13
623	7.24	6.35	13.26	2.09
621	7.91	6.60	13.41	2.03
624	8.52	7.19	14.45	2.01

Table B.6: Experimental results for impact of 3.175mm aluminium sphere onto aluminium target. Data is from Denardo, Summers and Nysmith [1].

Impact Velocity (km/s)	Crater Depth (mm)	Crater Diameter (mm)	Diameter/ Depth
4.00	5.21	9.76	1.87
5.00	5.68	10.89	1.92
6.00	6.06	11.29	1.86
7.00	6.41	11.57	1.80
8.00	6.42	11.51	1.79

Table B.7: DYNA3D results for impact of 3.175mm aluminium sphere onto aluminium target. Deformation erosion criterion. Johnson-Cook strength model

Projectile Erosion	Target Erosion	CPU to end (s)	Timesteps to end	D (mm)	v_2 (km/s)	v_4 (km/s)
0.5	0.5	7074	9564	5.24	7.78	6.91
1.0	1.0	14382	17865	6.02	7.66	6.71
1.5	1.5	27259	21676	6.05	†	†

†: interpenetration at contact surface obscured results

Table B.8: Results for thin plate target. Effective plastic strain erosion criterion. Johnson-Cook strength model.

Projectile Erosion	Target Erosion	CPU to end (s)	Timesteps to end	D (mm)	v_2 (km/s)
1.0	1.0	5196	8690	6.37	‡
1.5	1.5	6397	10130	7.21	‡
2.0	2.0	11276	15994	7.52	‡
2.5	2.5	18825	25208	7.61	7.38
3.0	3.0	27714	38355	7.65	7.40
1.5	2.0	8636	12990	7.5	‡
2.5	2.0	16901	24745	7.53	7.51
2.0	1.5	10033	15599	7.21	‡
2.0	2.5	14248	19959	7.61	‡

‡: insufficient material in debris cloud to allow measurement

Table B.9: Results for thin plate target. Deformation erosion criterion. Johnson-Cook strength model.

Projectile Erosion	Target Erosion	CPU to end (s)	Timesteps to end	D (mm)	v_2 (km/s)
1.0	1.0	5454	9456	6.31	‡
1.5	1.5	6134	9922	6.79	‡
2.0	2.0	10509	14258	7.24	‡
2.5	2.5	18638	25191	7.38	7.37
3.0	3.0	23201	35981	7.43	7.65

‡: insufficient material in debris cloud to allow measurement

Table B.10: Results for thin plate target. Deformation erosion criterion. Steinberg-Guinan strength model.

Projectile Erosion	Target Erosion	Skew Angle (radians)	CPU to end (s)	Timesteps to end	D (mm)
1.5	1.5	0.5	52204	3032	5.76
2.0	2.0	0.5	58696	36792	6.28
2.5	2.5	0.5	51833	33394	6.53
2.5	2.5	0.4	76655	46297	6.74
2.5	2.5	0.3	86009	49870	6.59
2.5	2.5	0.2	72554	45843	6.64
2.5	2.5	0.1	71171	43318	6.66
2.5	2.5	0.0	69793	44775	6.67

Projectile Erosion	Target Erosion	Skew Angle (radians)	v_2 (km/s)	v_4 (km/s)	γ (°)
1.5	1.5	0.5	7.32	5.77	‡
2.0	2.0	0.5	7.11	4.96	‡
2.5	2.5	0.5	7.08	4.82	32.1
2.5	2.5	0.4	7.05	4.76	32.3
2.5	2.5	0.3	7.10	4.72	35.8
2.5	2.5	0.2	7.08	4.72	30.6
2.5	2.5	0.1	7.06	4.71	29.3
2.5	2.5	0.0	7.09	4.71	29.3

‡: insufficient material in debris cloud to allow measurement

Table B.11: Results for thin plate target Accuracy erosion criterion, Jacobian determinant evaluated at 8 points. Johnson-Cook strength model.

Shot ID	Impact Velocity (km/s)	Hole Diameter (mm)
SS-141B	4.29	13.49
SS-141C	4.93	14.10
SS-141A	5.43	14.33
SS-141D	6.33	15.88
EH2C	6.64	16.03
EH2D	6.66	16.13
EH2B	6.67	19.08 †
EH2E	6.79	14.81
EH2A	7.06	16.10

† This value appears too large and may be a typographical error in [2]. The printed value is 0.751 in, a value of 0.651 in (16.54 mm) appears more reasonable.

Table B.12: Experimental results for impact of 7.95mm aluminium sphere onto 1.6mm aluminium plate. Data is from Schonberg, Bean and Darzi [2].

Impact Velocity (km/s)	Hole Diameter (mm)
4.00	12.67
5.00	12.85
6.00	14.24
7.00	14.63
8.00	15.05

Table B.13: DYNA3D results for impact of 7.95mm aluminium sphere onto 1.6mm aluminium plate. Deformation erosion criterion. Johnson-Cook strength model.

Shot ID	Impact Velocity (km/s)	Hole Diameter (mm)
SS-117-2	4.17	9.25
SS-176D	4.33	9.68
SS-118-1	4.40	9.65
SS-118-2	4.49	9.42
SS-118-3	4.52	10.11
SS-119-1	4.76	10.39
SS-176C	5.07	9.40
SS-119-2	5.32	9.65
SS-119-3	5.55	9.53
SS-176B	5.89	9.73
SS-176A	6.61	10.16
SS-175A	6.99	10.41
SS-175C	7.30	11.00
SS-175B	7.34	10.52

Table B.14: Experimental results for impact of 6.35mm aluminium sphere onto 0.81mm aluminium plate. Data is from Schonberg, Bean and Darzi [2].

Impact Velocity (km/s)	Hole Diameter (mm)
4.00	9.03
5.00	9.30
6.00	9.40
7.00	9.81
8.00	9.98

Table B.15: DYNA3D results for impact of 6.35mm aluminium sphere onto 0.81mm aluminium plate. Deformation erosion criterion. Johnson-Cook strength model

B.3.2 Figures

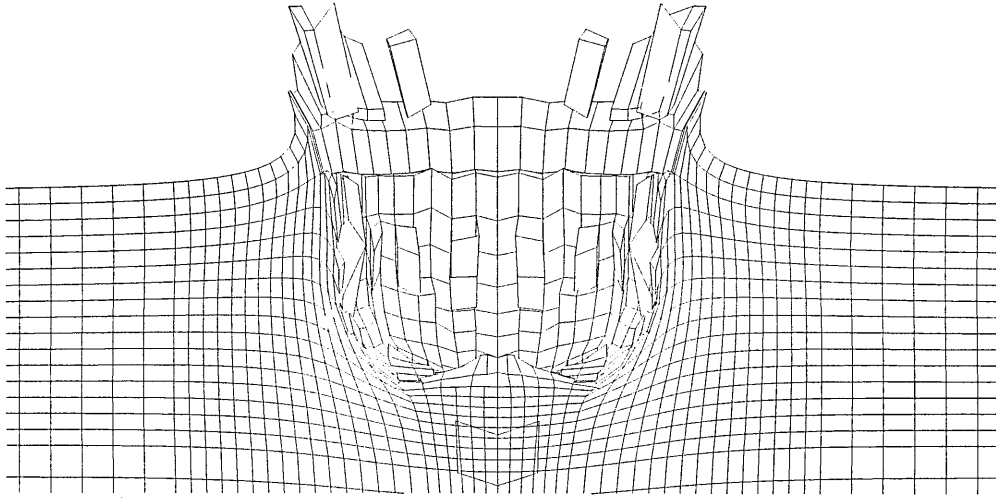


Figure B.13: Semi-infinite impact at $4 \mu s$. Effective plastic strain criterion. Projectile criteria 1.5, Target criteria 1.5. Johnson-Cook strength model.

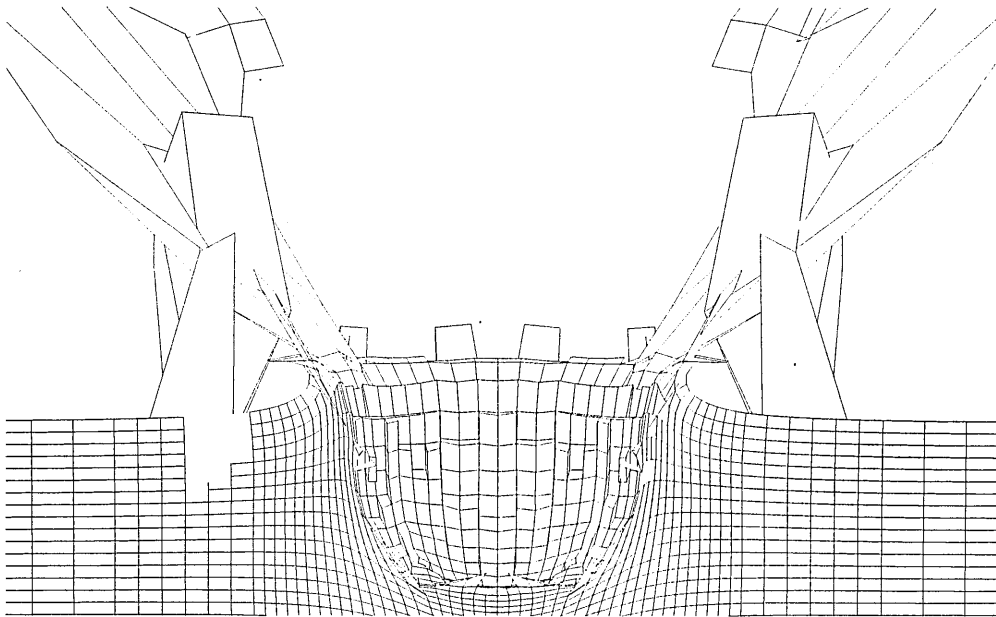


Figure B.14: Semi-infinite impact at $24 \mu s$. Effective plastic strain criterion. Projectile criteria 1.5, Target criteria 1.5. Johnson-Cook strength model.

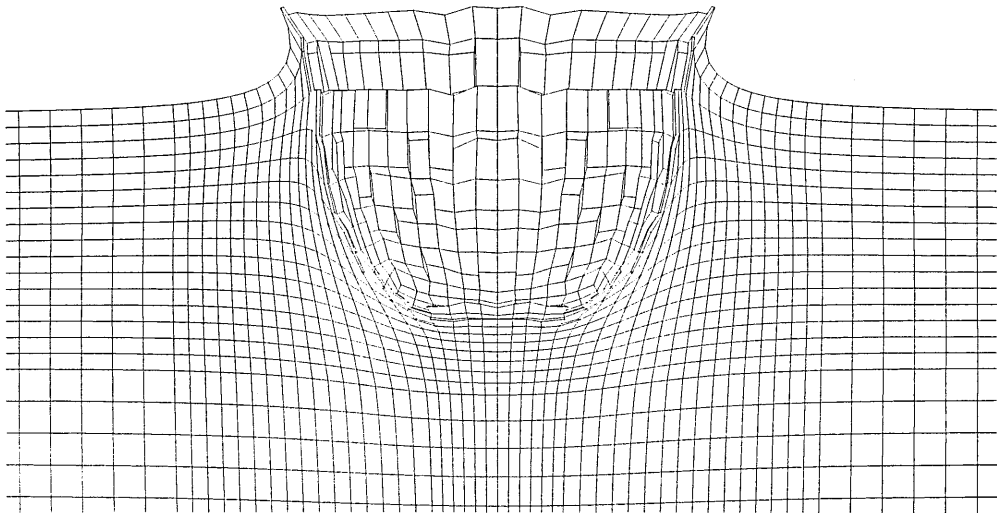


Figure B.15: Semi-infinite impact at $4\ \mu\text{s}$. Deformation erosion criterion. Projectile criteria 1.5, Target criteria 1.5. Johnson-Cook strength model.

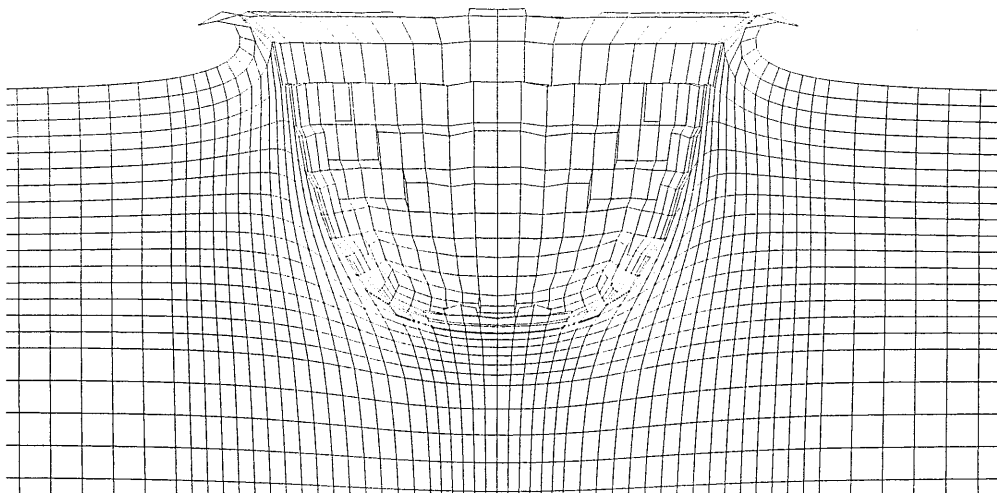


Figure B.16: Semi-infinite impact at $24\ \mu\text{s}$. Deformation erosion criterion. Projectile criteria 1.5, Target criteria 1.5. Johnson-Cook strength model.

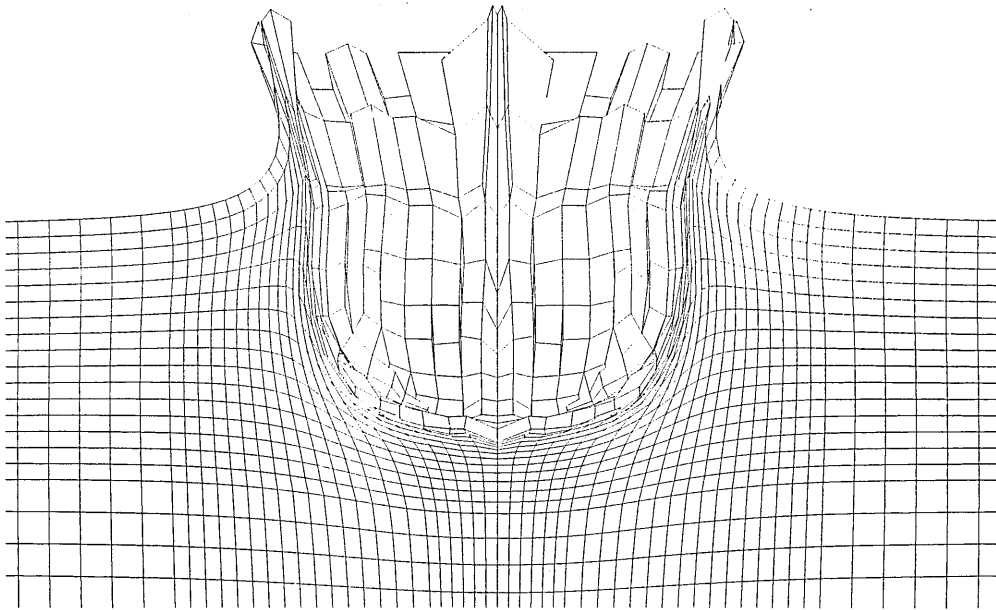


Figure B.17: Semi-infinite impact at $4 \mu s$. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5. Johnson-Cook strength model.

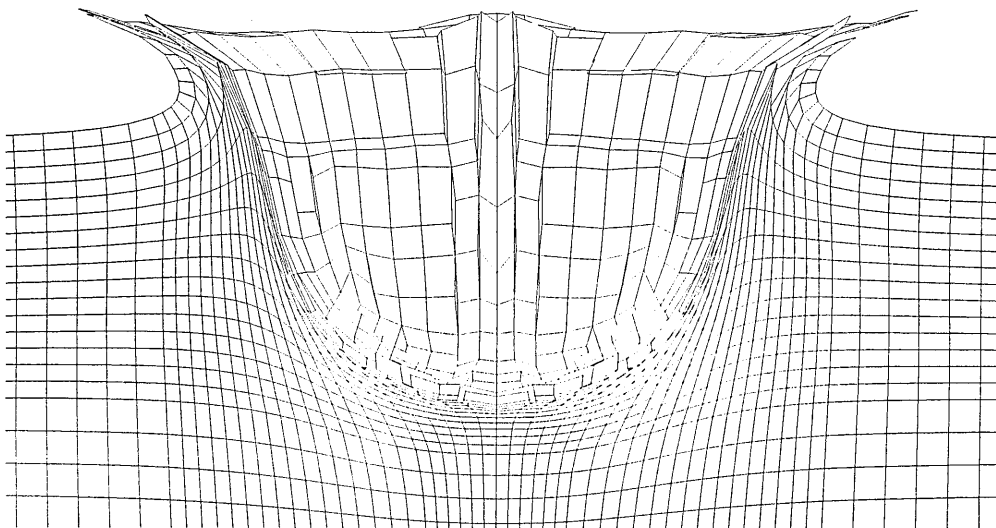


Figure B.18: Semi-infinite impact at $24 \mu s$. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5. Johnson-Cook strength model.

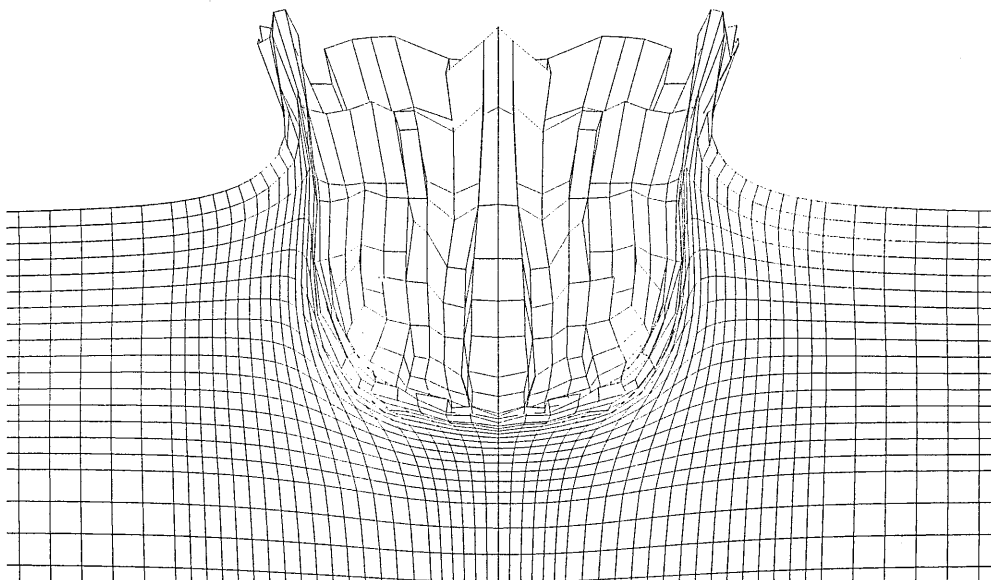


Figure B.19: Semi-infinite impact at $4 \mu s$. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5. Steinberg-Guinan strength model

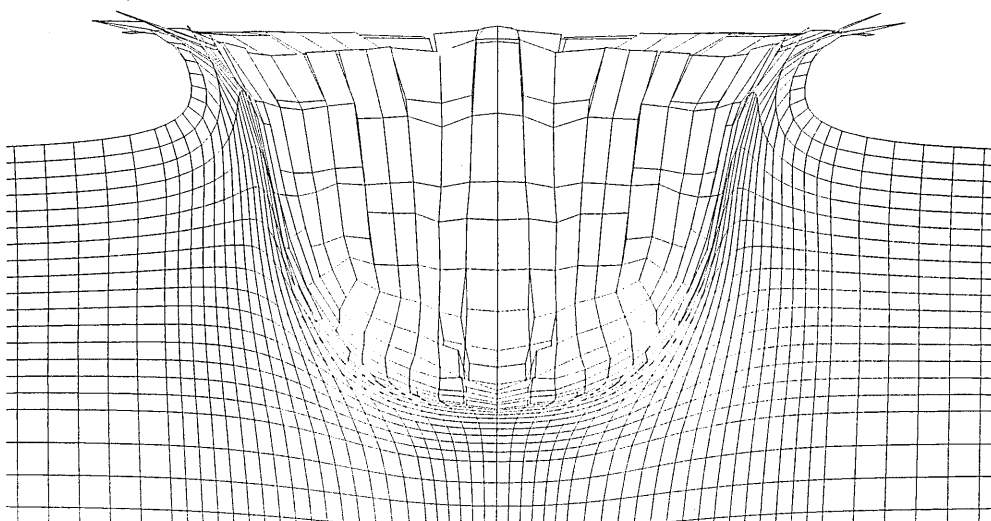


Figure B.20: Semi-infinite impact at $24 \mu s$. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5 Steinberg-Guinan strength model

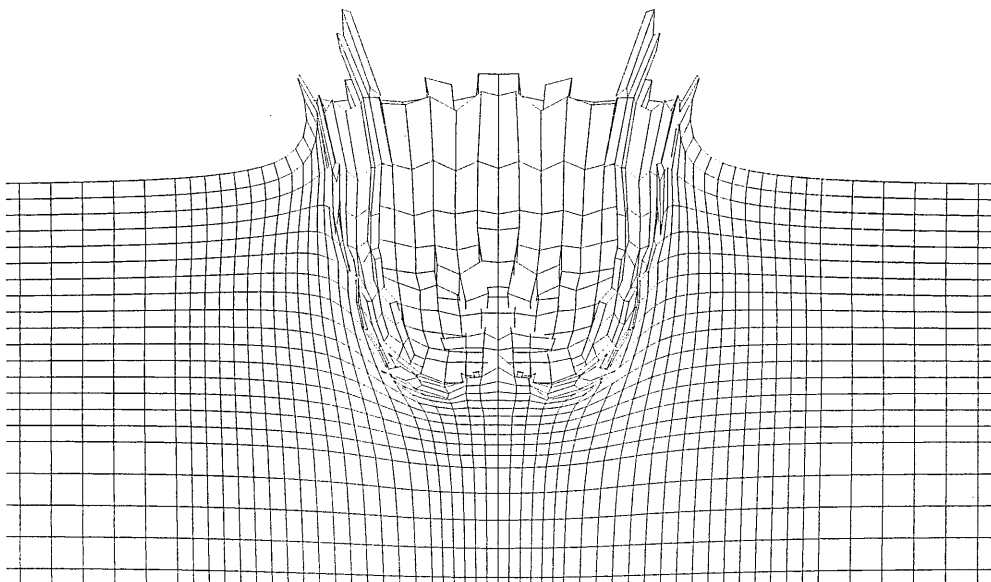


Figure B.21: Semi-infinite impact at 4 μs . Element accuracy erosion criterion. Projectile criteria 2.0, Target criteria 2.0. Johnson-Cook strength model.

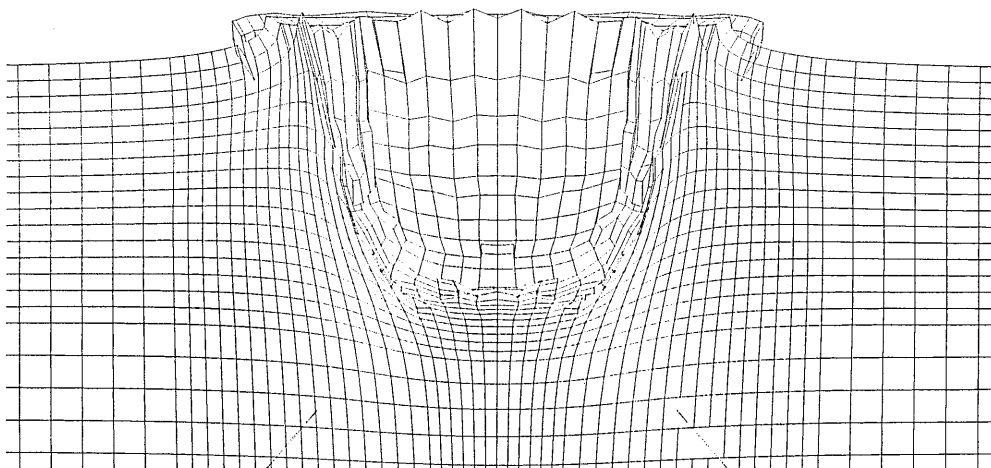


Figure B.22: Semi-infinite impact at 24 μs . Element accuracy criterion. Projectile criteria 2.0, Target criteria 2.0. Johnson-Cook strength model.

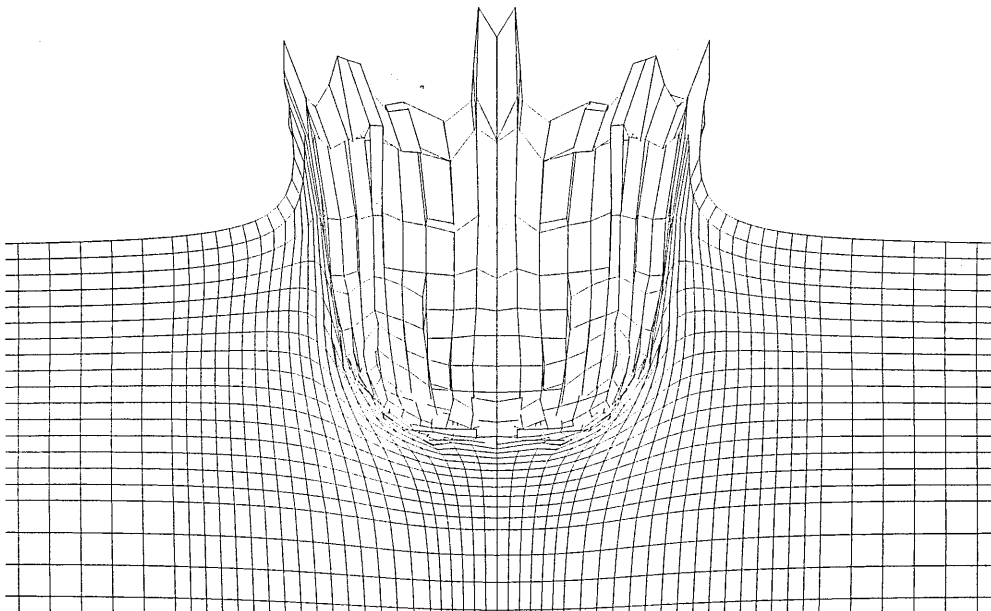


Figure B.23: Semi-infinite impact at $4 \mu s$. Element accuracy erosion criterion. Projectile criteria 3.0, Target criteria 3.0. Johnson-Cook strength model.

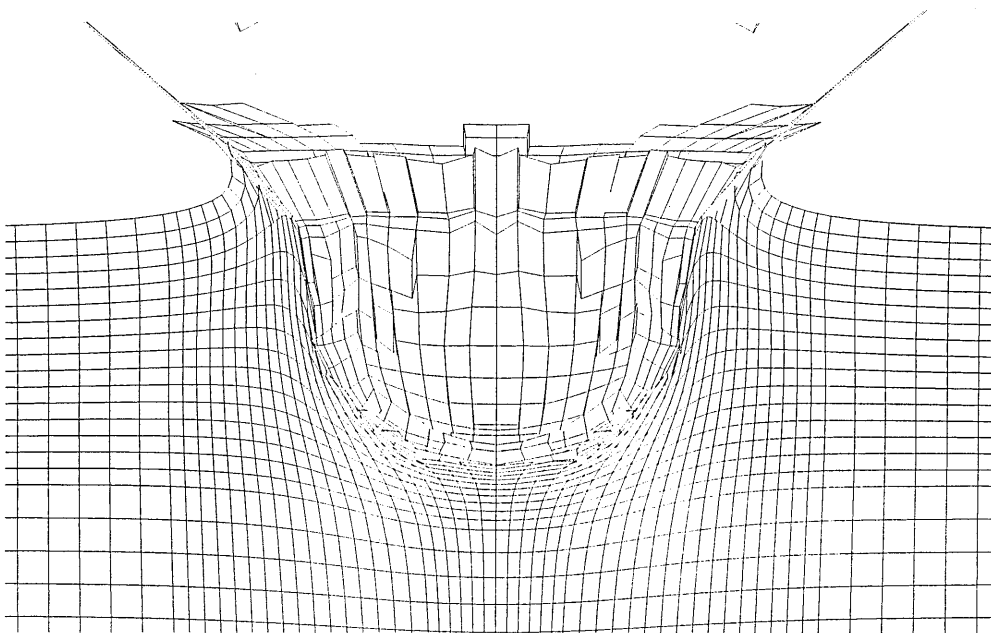


Figure B.24: Semi-infinite impact at $24 \mu s$. Element accuracy criterion. Projectile criteria 3.0, Target criteria 3.0. Johnson-Cook strength model.

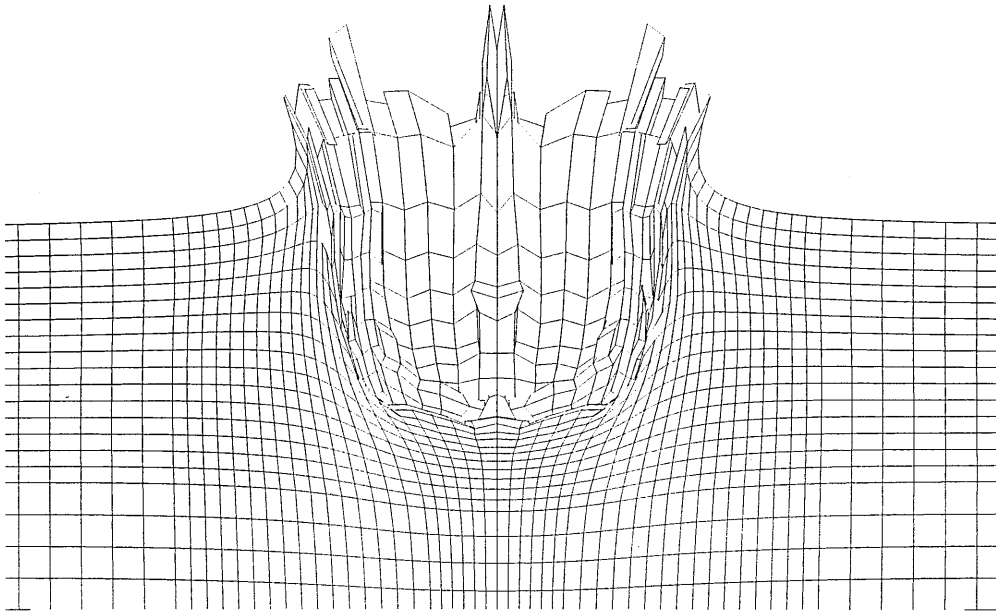


Figure B.25: Semi-infinite impact at $4 \mu s$. 8 point element accuracy erosion criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0 radians. Johnson-Cook strength model.

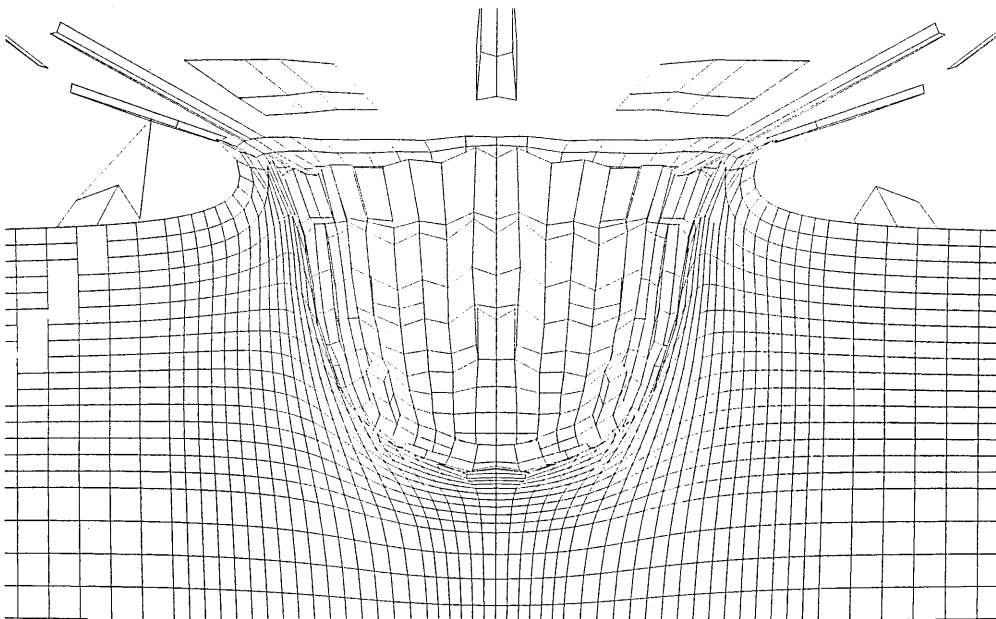


Figure B.26: Semi-infinite impact at $24 \mu s$. 8 point element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0 radians. Johnson-Cook strength model.

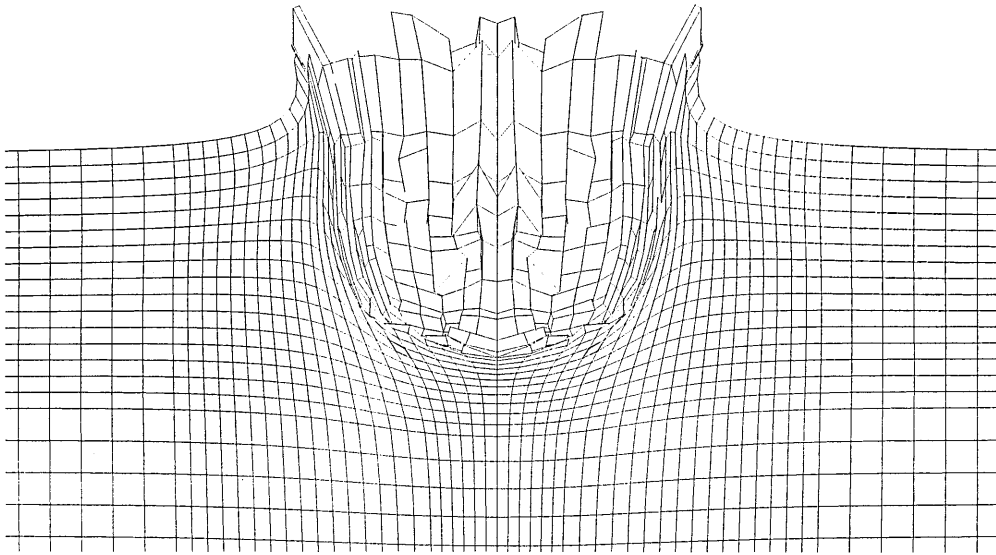


Figure B.27: Semi-infinite impact at $4 \mu s$. 8 point element accuracy erosion criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0.5 radians. Johnson-Cook strength model.

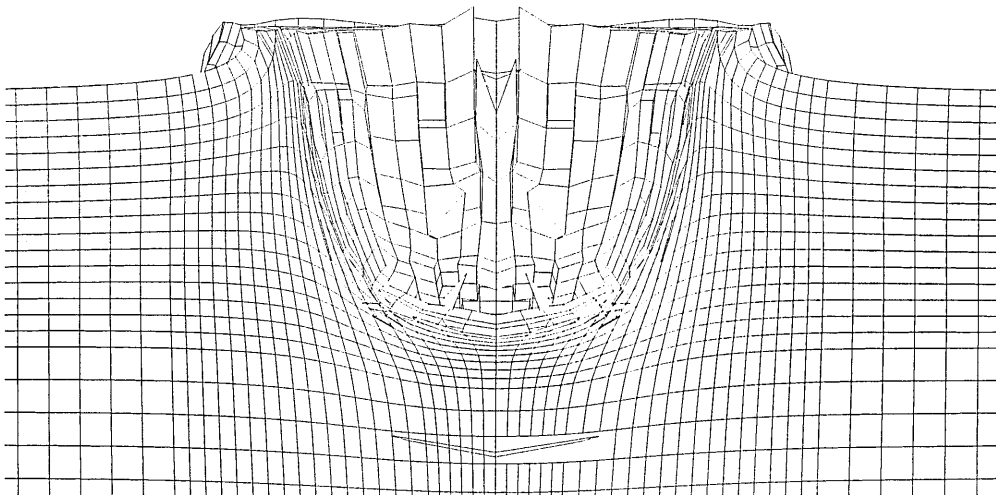


Figure B.28: Semi-infinite impact at $24 \mu s$. 8 point element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0.5 radians. Johnson-Cook strength model.

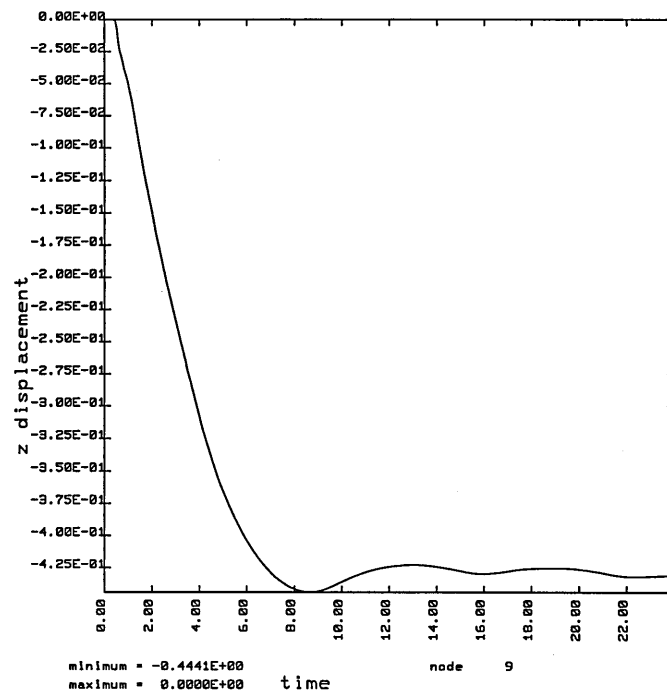


Figure B.29: Z displacement time history for node at centre of crater. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5

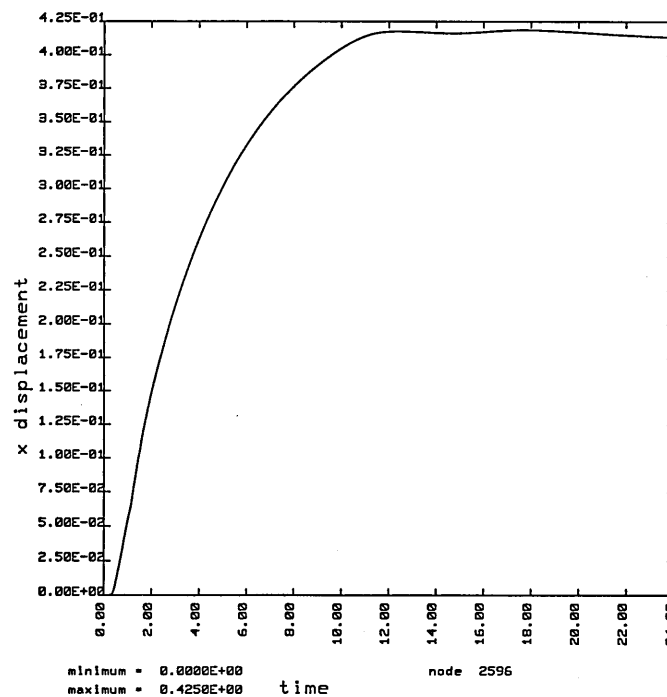


Figure B.30: X displacement time history for node at edge of crater. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5

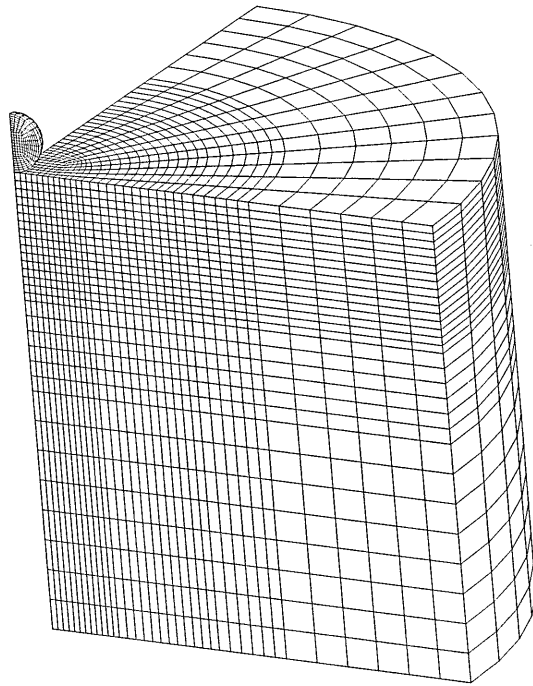


Figure B.31: Mesh for semi-infinite impact tests with varying velocity.

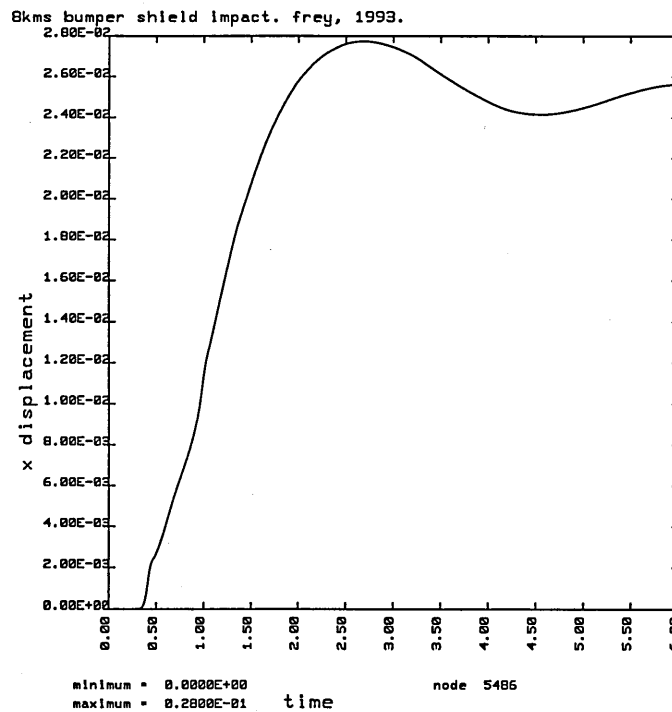


Figure B.32: X displacement time history for node at edge of hole. Deformation erosion criterion. Projectile criteria 2.5, Target criteria 2.5

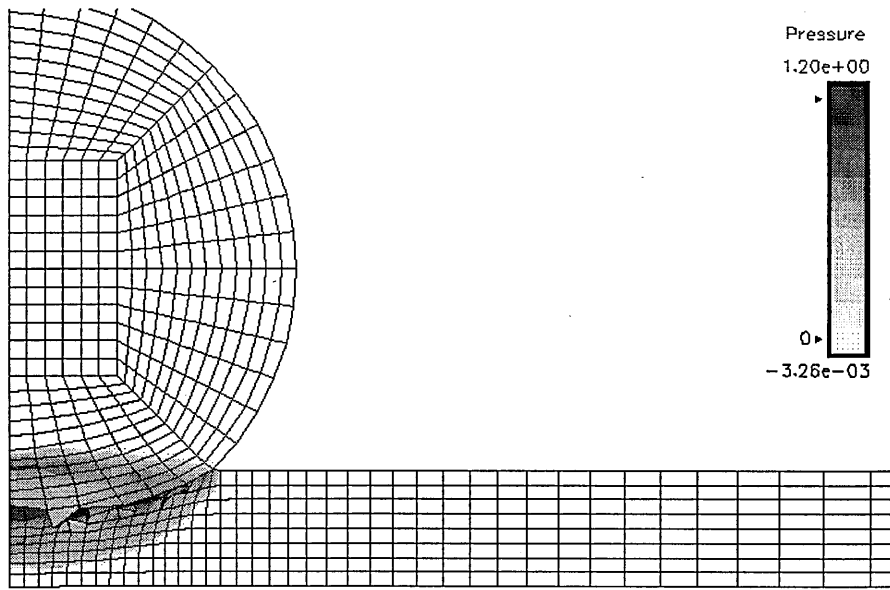


Figure B.33: Pressure contour plot, effective plastic strain erosion criterion, Johnson-Cook model. Time $0.075 \mu s$. Erosion strain 1.5

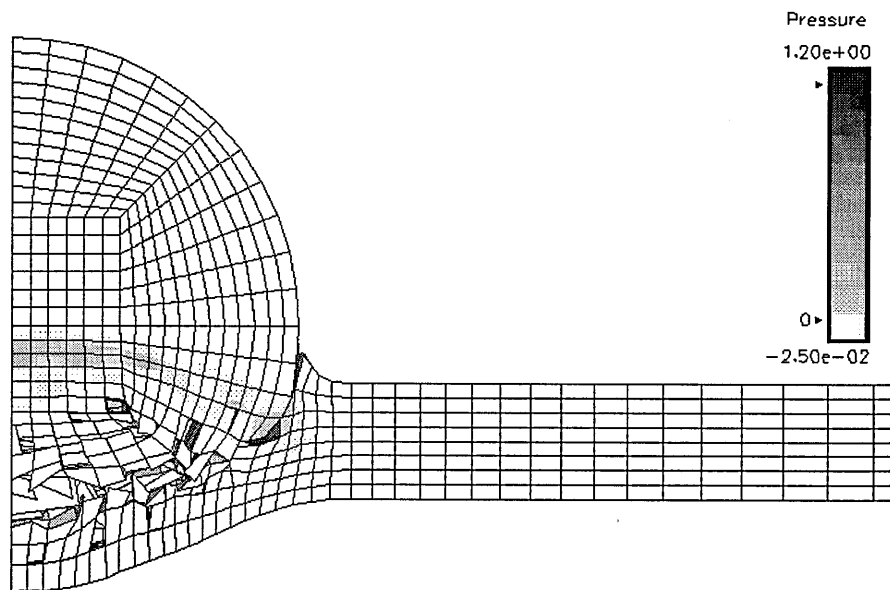


Figure B.34: Pressure contour plot, effective plastic strain erosion criterion, Johnson-Cook model. Time $0.2 \mu s$. Erosion strain 1.5

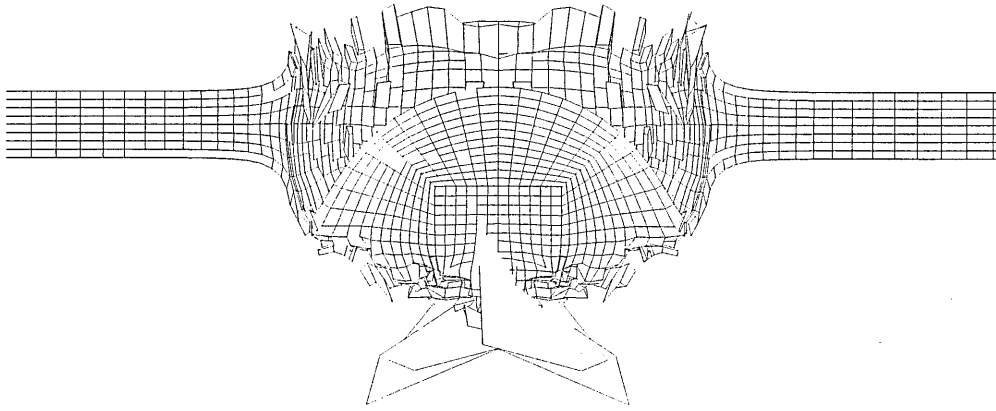


Figure B.35: Bumper shield impact at $0.5 \mu s$. Effective plastic strain erosion criterion. Projectile criteria 1.0, Target criteria 1.0. Johnson-Cook strength model.

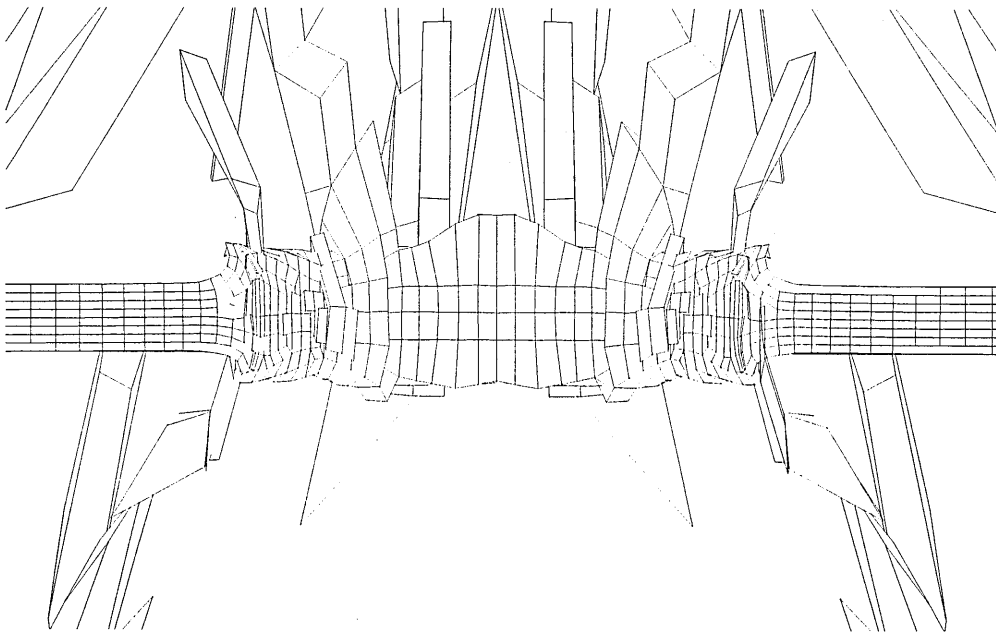


Figure B.36: Bumper shield impact at $4 \mu s$. Effective plastic strain erosion criterion. Projectile criteria 1.0, Target criteria 1.0. Johnson-Cook strength model.

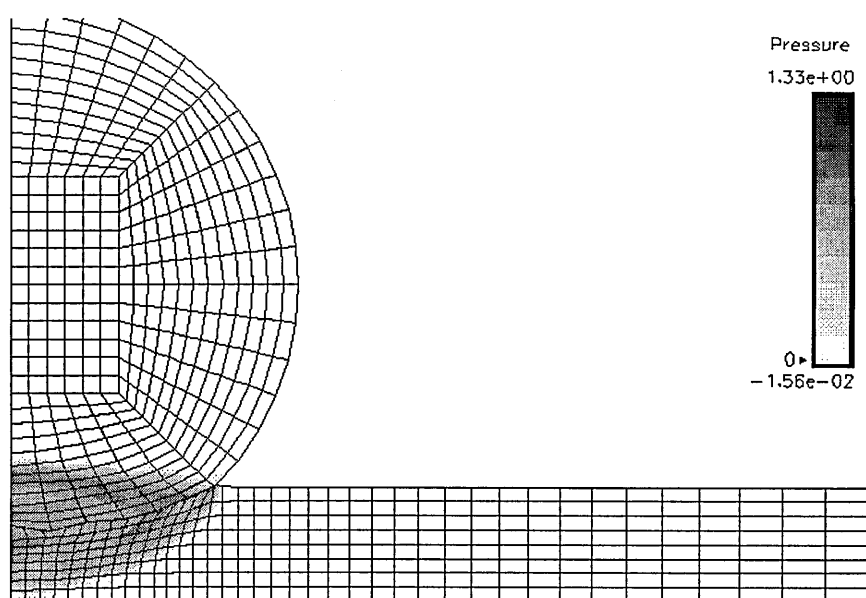


Figure B.37: Pressure contour plot, deformation erosion criterion.. Time $0.075 \mu s$.

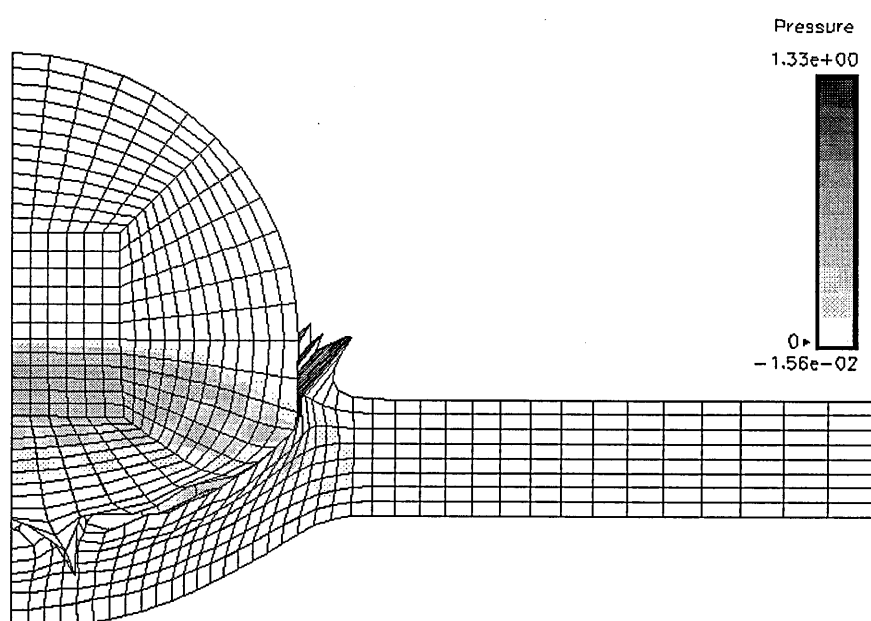


Figure B.38: Pressure contour plot, deformation erosion criterion.. Time $0.2 \mu s$.
Erosion value 2.5.

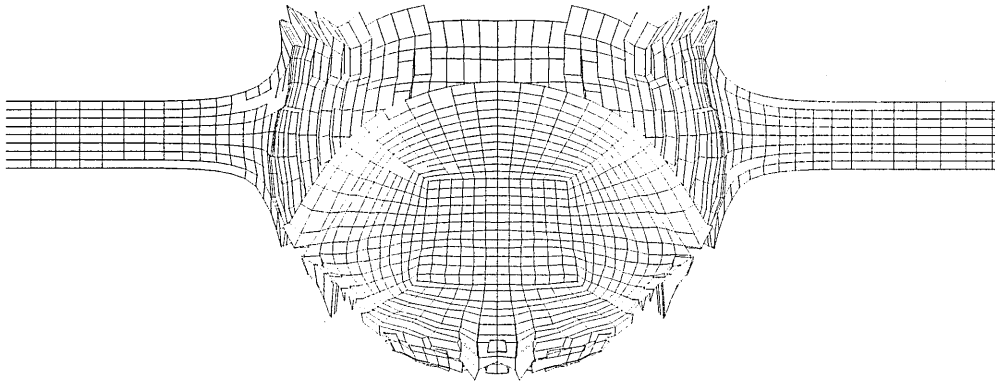


Figure B.39: Bumper shield impact at $0.5 \mu s$. Deformation erosion criterion . Projectile criteria 1.5, Target criteria 1.5. Erosion value 2.5. Johnson-Cook strength model.

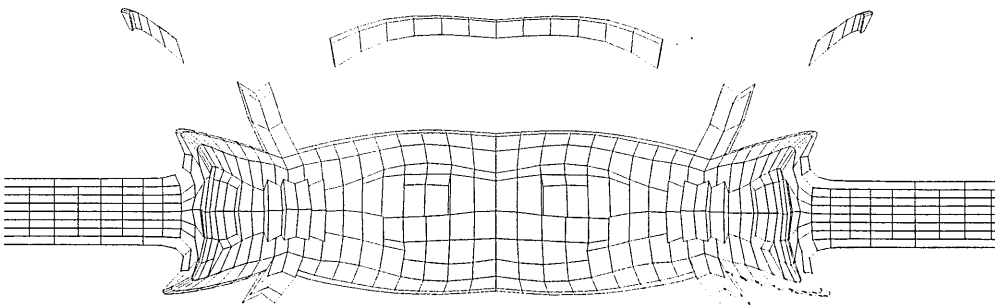


Figure B.40: Bumper shield impact at $4 \mu s$. Deformation erosion criterion . Projectile criteria 1.5, Target criteria 1.5. Johnson-Cook strength model.

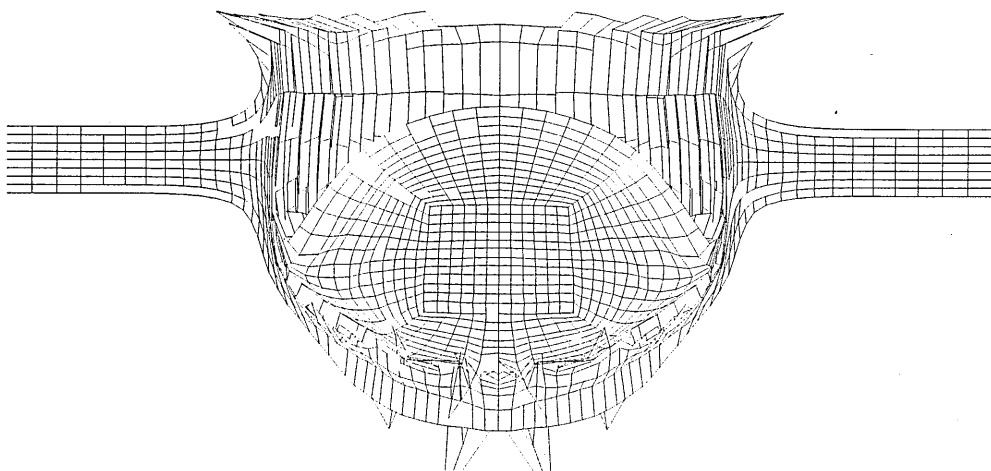


Figure B.41: Bumper shield impact at $0.5 \mu s$. Deformation erosion criterion . Projectile criteria 2.5, Target criteria 2.5. Johnson-Cook strength model.

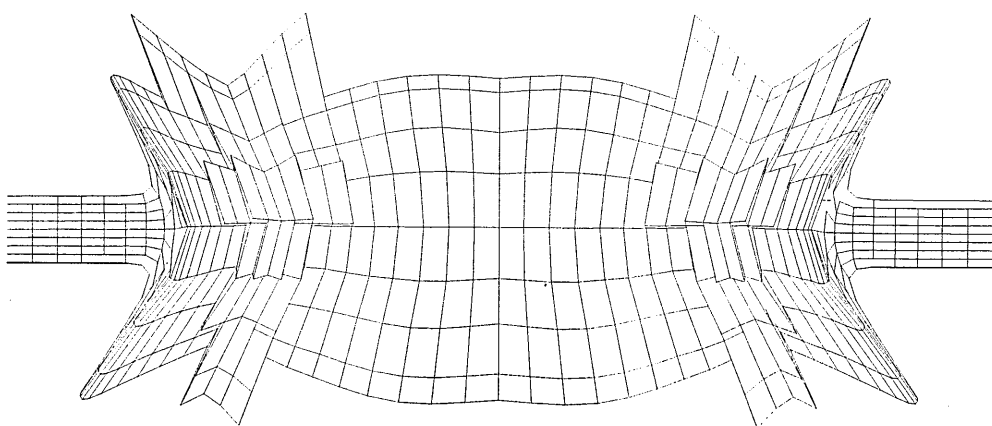


Figure B.42: Bumper shield impact at $4 \mu s$. Deformation erosion criterion . Projectile criteria 2.5, Target criteria 2.5. Johnson-Cook strength model.

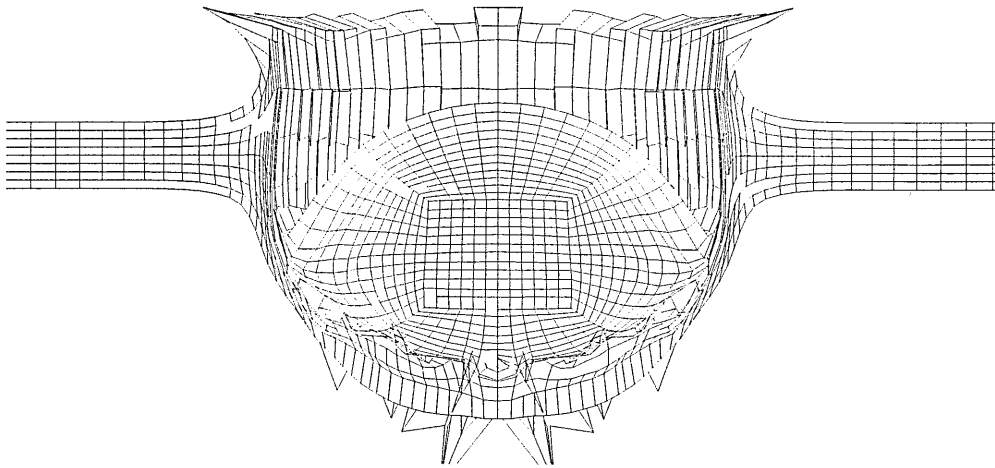


Figure B.43: Bumper shield impact at $0.5 \mu s$. Steinberg-Guinan strength model. Deformation erosion criterion . Projectile criteria 2.5, Target criteria 2.5

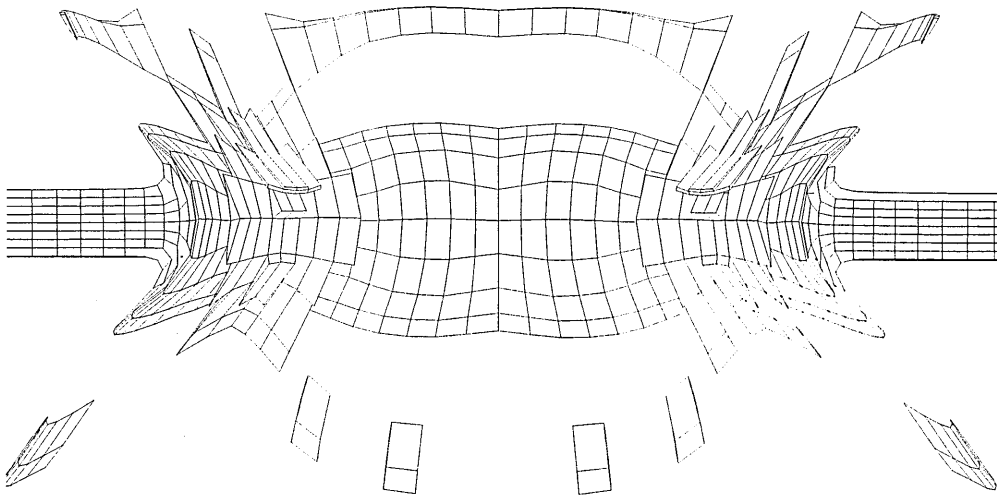


Figure B.44: Bumper shield impact at $4 \mu s$. Steinberg-Guinan strength model. Deformation erosion criterion . Projectile criteria 2.5, Target criteria 2.5

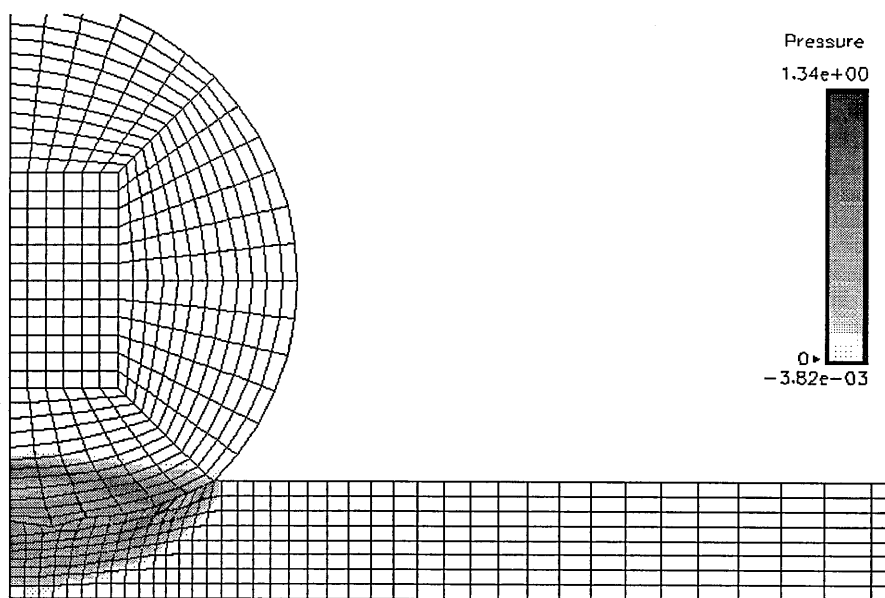


Figure B.45: Pressure contour plot, accuracy erosion criterion.. Time 0.075 μ s .

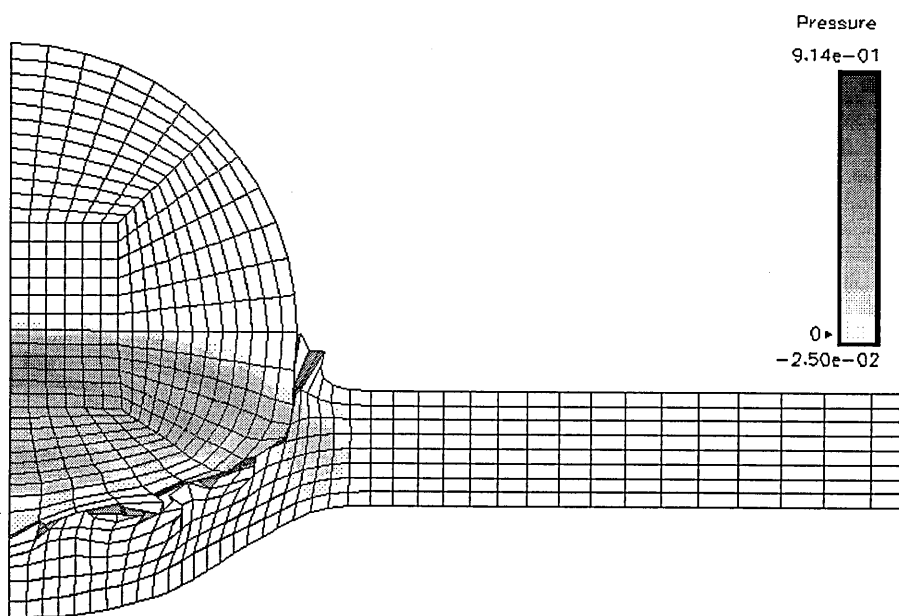


Figure B.46: Pressure contour plot, accuracy erosion criterion.. Time 0.2 μ s .

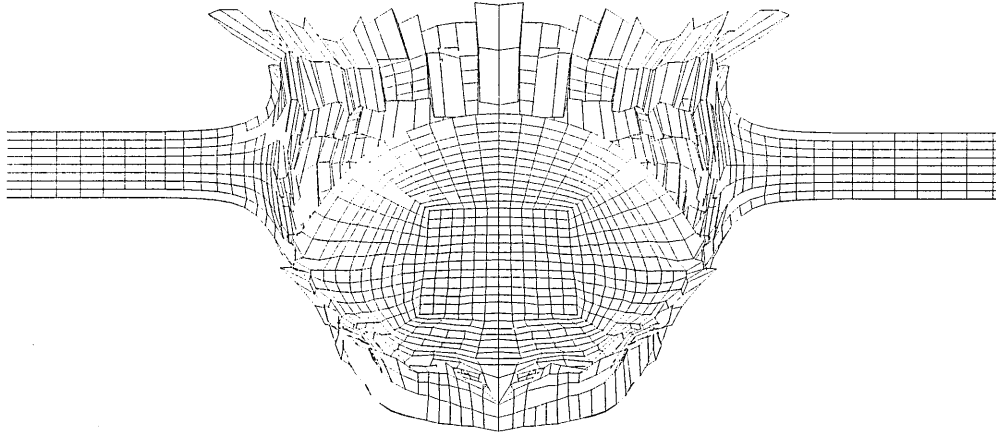


Figure B.47: Bumper shield impact at $0.5 \mu s$. Element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0.5 radians. Johnson-Cook strength model.

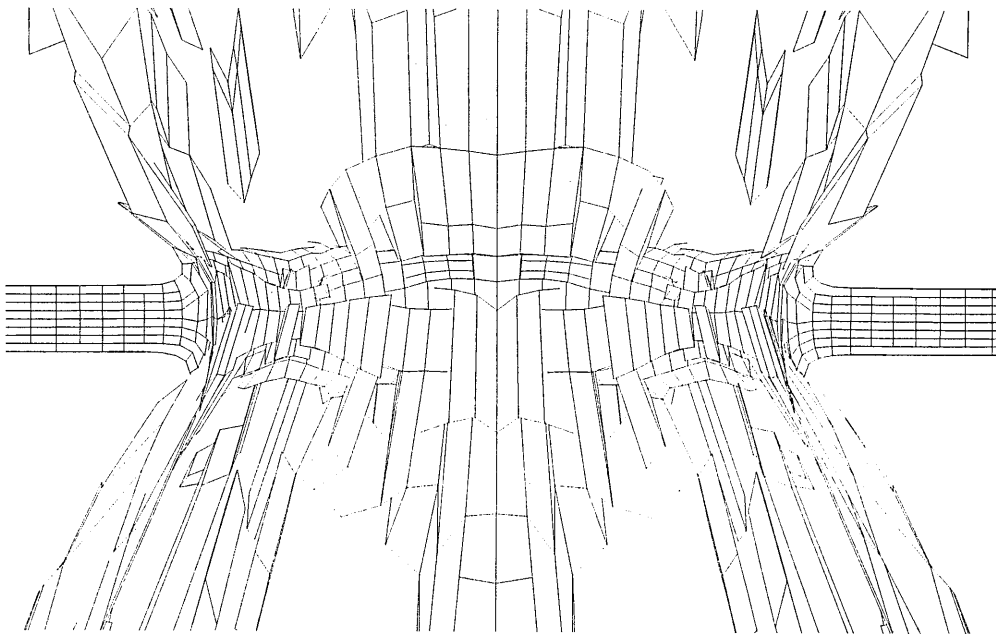


Figure B.48: Bumper shield impact at $4 \mu s$. Element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5 Skew angle 0.5 radians. Johnson-Cook strength model.

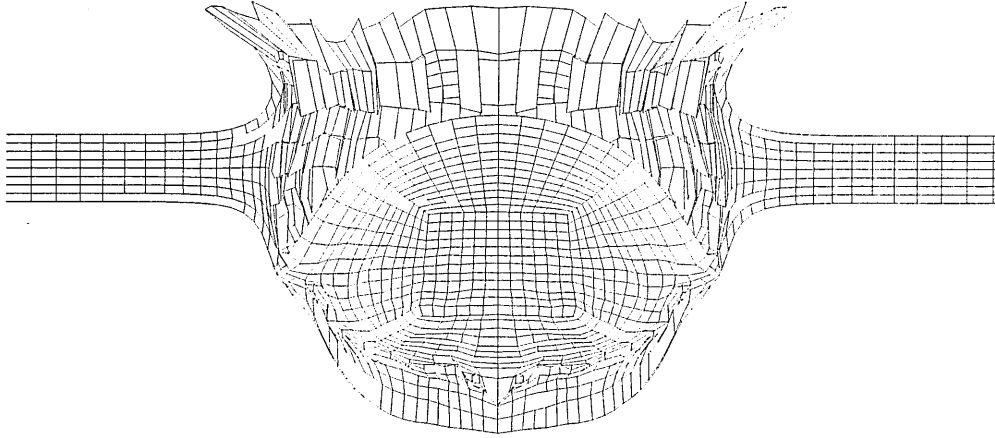


Figure B.49: Bumper shield impact at $0.5 \mu s$. Element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5. Skew angle 0 radians. Johnson-Cook strength model.

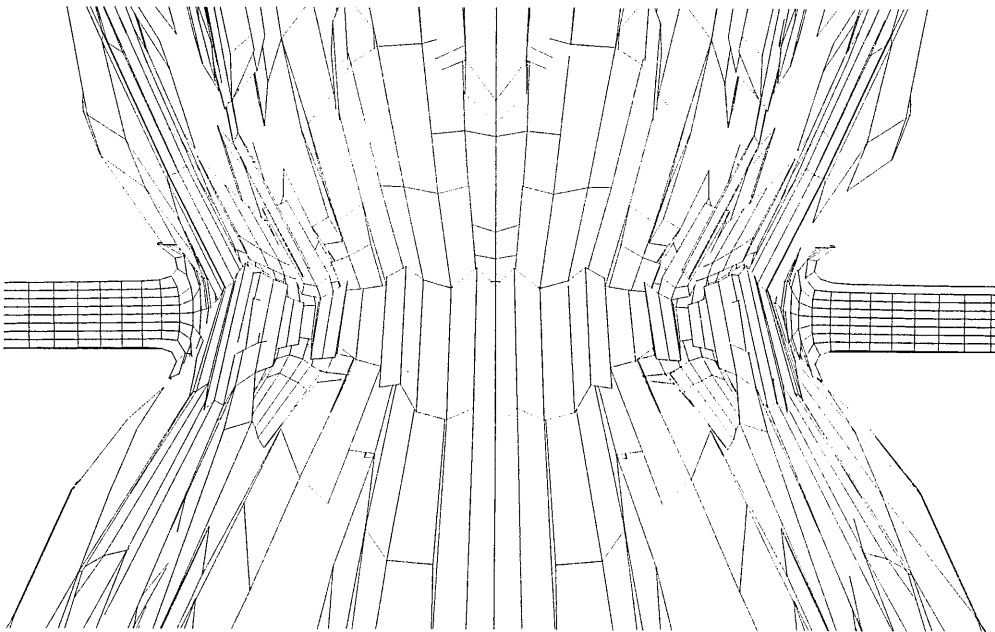


Figure B.50: Bumper shield impact at $4 \mu s$. Element accuracy criterion. Projectile criteria 2.5, Target criteria 2.5 Skew angle 0 radians. Johnson-Cook strength model.

Appendix C

Derivation of SPH Equations

In this section the full derivations of the basic SPH equations: the conservation of mass, momentum and energy, and the rate-of-deformation and spin tensors.

C.1 Conservation of Mass

The Continuity Equation

$$\frac{d\rho}{dt} = -\rho \frac{\partial v^\beta}{\partial x^\beta} \quad (\text{C.1})$$

Write as a kernel estimate

$$\left\langle \frac{d\rho}{dt} \right\rangle = - \int_V W \rho' \frac{\partial v'^\beta}{\partial x'^\beta} dx' \quad (\text{C.2})$$

the dash, ', represents values evaluated at point x' .

Simplify the equation, this step is explained in section 7.1.4.

$$\left\langle \frac{d\rho}{dt} \right\rangle = -\rho \int_V W \frac{\partial v'^\beta}{\partial x'^\beta} dx' \quad (\text{C.3})$$

Integrate by parts

$$\left\langle \frac{d\rho}{dt} \right\rangle = -\rho W v'^\beta + \rho \int_V v'^\beta \frac{\partial W}{\partial x'^\beta} dx' \quad (\text{C.4})$$

The first term on the RHS can be rewritten

$$\rho W v'^\beta = \rho \int_V \frac{d}{dx'} (W v'^\beta) dx' \quad (\text{C.5})$$

This can be expressed as a surface integral by Green's Theorem

$$\rho \int_V \frac{d}{dx'} (W v'^\beta) dx' = \rho \int_S W v'^\beta n_i dS \quad (\text{C.6})$$

Now assume that the solution space extends far enough that on its boundary, S , $W = 0$. The surface term is then identically zero. This assumption can be justified for

astrophysical situations, the application for which SPH was first developed, as very often the fluid is contained by self-gravity. For modelling solids this assumption is no longer valid, and boundary conditions are the subject of ongoing work.

By neglecting surface terms equation C.4 can be written

$$\left\langle \frac{d\rho}{dt} \right\rangle = \rho \int_V v'^\beta \frac{\partial W}{\partial x'^\beta} dx' \quad (\text{C.7})$$

Evaluating this integral by the particle method (section 7.1.1) gives

$$\frac{d\rho_i}{dt} = \rho_i \sum_j \frac{m_j}{\rho_j} v_j^\beta \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (\text{C.8})$$

It is desirable to introduce velocity differences in to this equation [34], as this is consistent with the energy and strain rate calculations, and maintains Galilean invariance. To do this the fact that the kernel vanishes at infinity is used:

$$\rho v^\beta \int_V \frac{\partial W}{\partial x'^\beta} dx' = 0 \quad (\text{C.9})$$

Evaluating this integral by the particle method gives

$$\rho_i v_i^\beta \sum_j \frac{m_j}{\rho_j} \frac{\partial W_{ij}}{\partial x_i^\beta} = 0 \quad (\text{C.10})$$

subtracting (C.10) from (C.8) gives

$$\frac{d\rho_i}{dt} = \rho_i \sum_j \frac{m_j}{\rho_j} (v_j^\beta - v_i^\beta) \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (\text{C.11})$$

C.2 Conservation of Momentum

$$\frac{dv^\alpha}{dt} = \frac{1}{\rho} \frac{\partial \sigma^{\alpha\beta}}{\partial x^\beta} \quad (\text{C.12})$$

Rewrite using Quotient rule

$$\frac{dv^\alpha}{dt} = \frac{\partial}{\partial x^\beta} \left(\frac{\sigma^{\alpha\beta}}{\rho} \right) + \frac{\sigma^{\alpha\beta}}{\rho^2} \frac{\partial \rho}{\partial x^\beta} \quad (\text{C.13})$$

Write as a kernel estimate

$$\left\langle \frac{dv^\alpha}{dt} \right\rangle = \int_V W \frac{\partial}{\partial x'^\beta} \left(\frac{\sigma'^{\alpha\beta}}{\rho'} \right) dx' + \int_V W \frac{\sigma'^{\alpha\beta}}{\rho'^2} \frac{\partial \rho'}{\partial x'^\beta} dx' \quad (\text{C.14})$$

Simplify the equation

$$\left\langle \frac{dv^\alpha}{dt} \right\rangle = \int_V W \frac{\partial}{\partial x'^\beta} \left(\frac{\sigma'^{\alpha\beta}}{\rho'} \right) dx' + \frac{\sigma^{\alpha\beta}}{\rho^2} \int_V W \frac{\partial \rho'}{\partial x'^\beta} dx' \quad (\text{C.15})$$

Integrate by parts and neglect surface integrals, see the derivation of the continuity equation for more detail.

$$\left\langle \frac{dv^\alpha}{dt} \right\rangle = - \int_V \frac{\sigma'^{\alpha\beta}}{\rho'} \frac{\partial W}{\partial x'^\beta} dx' - \frac{\sigma^{\alpha\beta}}{\rho^2} \int_V \rho' \frac{\partial W}{\partial x'^\beta} dx' \quad (\text{C.16})$$

Evaluating the integrals by the particle method (section 7.1.1) gives

$$\frac{dv_i^\alpha}{dt} = - \sum_j \frac{m_j}{\rho_j} \frac{\sigma_j^{\alpha\beta}}{\rho_j} \frac{\partial W_{ij}}{\partial x_i^\beta} - \frac{\sigma_i^{\alpha\beta}}{\rho_i^2} \sum_j \frac{m_j}{\rho_j} \rho_j \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (\text{C.17})$$

Which can be simplified to give

$$\frac{dv_i^\alpha}{dt} = - \sum_j m_j \left(\frac{\sigma_j^{\alpha\beta}}{\rho_j^2} + \frac{\sigma_i^{\alpha\beta}}{\rho_i^2} \right) \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (\text{C.18})$$

C.3 Conservation of Energy

$$\frac{dE}{dt} = \frac{\sigma^{\alpha\beta}}{\rho} \frac{dv^\alpha}{dx^\beta} \quad (\text{C.19})$$

where E is the specific internal energy This can be rewritten placing the density inside operators, Monaghan's 'second golden rule of SPH' [77]

$$\frac{dE}{dt} = \frac{\sigma^{\alpha\beta}}{\rho^2} \left(\rho^2 \frac{\partial}{\partial x^\beta} \left(\frac{v^\alpha}{\rho} \right) + v^\alpha \frac{\partial \rho}{\partial x^\beta} \right) \quad (\text{C.20})$$

The first term in the brackets can be rewritten as (using Quotient then Product rules)

$$\rho^2 \frac{\partial}{\partial x^\beta} \left(\frac{v^\alpha}{\rho} \right) = \frac{\partial}{\partial x^\beta} (\rho v^\alpha) - 2v^\alpha \frac{\partial \rho}{\partial x^\beta} \quad (\text{C.21})$$

Substituting back into C.20

$$\frac{dE}{dt} = \frac{\sigma^{\alpha\beta}}{\rho^2} \frac{\partial (\rho v^\alpha)}{\partial x^\beta} - \frac{\sigma^{\alpha\beta} v^\alpha}{\rho^2} \frac{\partial \rho}{\partial x^\beta} \quad (\text{C.22})$$

Write as a kernel estimate

$$\left\langle \frac{dE}{dt} \right\rangle = \int_V W \frac{\sigma'^{\alpha\beta}}{\rho'^2} \frac{\partial (\rho' v'^\alpha)}{\partial x'^\beta} dx' - \int_V W \frac{\sigma'^{\alpha\beta} v'^\alpha}{\rho'^2} \frac{\partial \rho'}{\partial x'^\beta} dx' \quad (\text{C.23})$$

Simplify the equation

$$\left\langle \frac{dE}{dt} \right\rangle = \frac{\sigma^{\alpha\beta}}{\rho^2} \int_V W \frac{\partial (\rho' v'^\alpha)}{\partial x'^\beta} dx' - \frac{\sigma^{\alpha\beta} v^\alpha}{\rho^2} \int_V W \frac{\partial \rho'}{\partial x'^\beta} dx' \quad (\text{C.24})$$

Integrate by parts and neglect surface terms, see the derivation of the continuity equation for more detail.

$$\left\langle \frac{dE}{dt} \right\rangle = -\frac{\sigma^{\alpha\beta}}{\rho^2} \int_V \rho' v'^\alpha \frac{\partial W}{\partial x'^\beta} dx' - \frac{\sigma^{\alpha\beta} v^\alpha}{\rho^2} \int_V \rho' \frac{\partial W}{\partial x'^\beta} dx' \quad (\text{C.25})$$

Evaluating the integrals by the particle method (section 7.1.1) gives

$$\frac{dE}{dt} = -\frac{\sigma_i^{\alpha\beta}}{\rho_i^2} \sum_j \frac{m_j}{\rho_j} \rho_j v_j^\alpha \frac{\partial W_{ij}}{\partial x_i^\beta} + \frac{\sigma_i^{\alpha\beta} v_i^\alpha}{\rho_i^2} \sum_j \frac{m_j}{\rho_j} \rho_j \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (\text{C.26})$$

Simplifying gives

$$\frac{dE}{dt} = -\frac{\sigma_i^{\alpha\beta}}{\rho_i^2} \sum_j m_j (v_j^\alpha - v_i^\alpha) \frac{\partial W_{ij}}{\partial x_i^\beta} \quad (\text{C.27})$$

C.4 Rate-of-Deformation and Spin Tensors

The rate-of-deformation tensor is defined as

$$\dot{\varepsilon}^{\alpha\beta} = \frac{1}{2} \left(\frac{\partial v^\alpha}{\partial x^\beta} + \frac{\partial v^\beta}{\partial x^\alpha} \right) \quad (\text{C.28})$$

Write as a kernel estimate

$$\langle \dot{\varepsilon}^{\alpha\beta} \rangle = \frac{1}{2} \int_V W \left(\frac{\partial v'^\alpha}{\partial x'^\beta} + \frac{\partial v'^\beta}{\partial x'^\alpha} \right) dx' \quad (\text{C.29})$$

Integrate by parts and neglect surface terms, see the derivation of the continuity equation for more detail.

$$\langle \dot{\varepsilon}^{\alpha\beta} \rangle = -\frac{1}{2} \int_V \left(v'^\alpha \frac{\partial W}{\partial x'^\beta} + v'^\beta \frac{\partial W}{\partial x'^\alpha} \right) dx' \quad (\text{C.30})$$

Evaluate integral by particle method

$$\dot{\varepsilon}_i^{\alpha\beta} = -\frac{1}{2} \sum_j \frac{m_j}{\rho_j} \left(v_j^\alpha \frac{\partial W_{ij}}{\partial x_i^\beta} + v_j^\beta \frac{\partial W_{ij}}{\partial x_i^\alpha} \right) \quad (\text{C.31})$$

Following the derivation of the Continuity equation, it is desirable to introduce velocity differences into this equation. This will ensure that

$$\dot{\varepsilon}^{\alpha\alpha} = \frac{\partial v^\alpha}{\partial x^\alpha} \quad (\text{C.32})$$

Using the fact that the kernel vanishes at infinity, the following expression is zero

$$\frac{1}{2} v_i^\alpha \sum_j \frac{m_j}{\rho_j} \frac{\partial W_{ij}}{\partial x_i^\beta} + \frac{1}{2} v_i^\beta \sum_j \frac{m_j}{\rho_j} \frac{\partial W_{ij}}{\partial x_i^\alpha} \quad (\text{C.33})$$

Adding (C.33) to (C.31) gives

$$\dot{\varepsilon}_i^{\alpha\beta} = -\frac{1}{2} \sum_j \frac{m_j}{\rho_j} (v_j^\alpha - v_i^\alpha) \frac{\partial W_{ij}}{\partial x_i^\beta} - \frac{1}{2} \sum_j \frac{m_j}{\rho_j} (v_j^\beta - v_i^\beta) \frac{\partial W_{ij}}{\partial x_i^\alpha} \quad (\text{C.34})$$

In the same manner the SPH form of the spin tensor can be derived

$$\dot{\Omega}^{\alpha\beta} = \frac{1}{2} \left(\frac{\partial v^\alpha}{\partial x^\beta} - \frac{\partial v^\beta}{\partial x^\alpha} \right) \quad (\text{C.35})$$

becomes

$$\dot{\Omega}_i^{\alpha\beta} = -\frac{1}{2} \sum_j \frac{m_j}{\rho_j} (v_j^\alpha - v_i^\alpha) \frac{\partial W_{ij}}{\partial x_i^\beta} + \frac{1}{2} \sum_j \frac{m_j}{\rho_j} (v_j^\beta - v_i^\beta) \frac{\partial W_{ij}}{\partial x_i^\alpha} \quad (\text{C.36})$$