SANDIA REPORT

SAND2015-7448
Unlimited Release
Printed August 2015

Evaluating the Material Point Method in CTH Using the Method of Manufactured Solutions

Vinesh V. Nishawala, Kevin P. Ruggirello

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2015-7448
Unlimited Release
Printed August 2015

Evaluating the Material Point Method in CTH Using the Method of Manufactured Solutions

Vinesh V. Nishawala and Kevin P. Ruggirello
Sandia National Laboratories
P.O. Box 5800 MS-0840
Albuquerque, NM 87185

Abstract

The Method of Manufactured Solutions (MMS) is used to evaluate the Material Point Method (MPM) implemented in CTH, i.e. Markers. MMS is a verification approach in which a desired deformation field is prescribed and the required forcing function to achieve the prescribed deformation is determined analytically. The calculated forcing function is applied within CTH markers determine if the correct displacement field is recovered. For the cases examined in this study, a ring is subjected to a finite, angular-independent, spatially varying body force, superposed with a rigid-body rotation. This test will assess the solid mechanics response of the MPM within CTH for large deformation problems.

This page intentionally left blank.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Material Point Method (MPM), as discussed below is implemented in the CTH hydroxide [8]. Within CTH it is commonly referred to as CTH Markers [9]. The Method of Manufactured Solutions (MMS) will subject the MPM to a prescribed load to determine if its response matches the original prescribed deformation. A very brief summary of the MPM and MMS are given below.

Material Point Method

There are two distinct frames of reference when referring to fields that vary in space and time. The Lagrangian frame is defined a reference frame which follows a specific mass of material through space and time. In contrast, the Eulerian frame describes a volume element which is fixed in space and material flows through the volume. Each approach has advantages and disadvantages when implemented in a numerical framework. Lagrangian based approaches are typically employed in finite element based methods, and are able to track material deformation and properties with minimal diffusion. Eulerian approaches, such as those typically used in computational fluid dynamics methods, excel at solving large deformation problems where a Lagrangian method would potentially fail due to mesh tangling. The MPM attempts to combine the advantages of each approach in a particle based method typically used to solve solid mechanics problems.

The MPM utilizes the fixed, Eulerian grid, and the deformed, Lagrangian grid. The Lagrangian grid points are called 'markers' and are free to move within the domain. The Eulerian grid points are fixed in their position throughout time. To map properties between markers and grid points, i.e. velocity, acceleration etc., the MPM utilizes linear shape functions typically used in Finite Elements. The discretized conservation equations are then solved on the grid points. The markers represent the integration points and integration over the materials domain is accomplished by a summation over markers. Steffen et. al. compare the MPM to a typical Finite Element formulation and present analysis of quadrature errors for the MPM applied to solid mechanics problems in [7]. More information on the MPM can be found in [4, 5], and information on its implementation within CTH can be found in [9].

Since the original MPM formulation only used local linear shape functions, the mass on the background grid would exhibit discrete jumps as a marker crossed a cell boundary. These discrete jumps were found to lead to spurious oscillations in the solution, and in some cases instabilities.

The linear shape functions also limited the markers to nearest neighbor communication on the background grid, which was found to result in numerical fracture for large deformation problems if the markers became separated by more than a cell width. To alleviate these issues, the Convected Particle Domain Interpolation (CPDI) method was recently devised [1]. The CPDI method introduces the concept of a particle/marker domain, uses the deformation gradient tensor ($\mathbf{F}$) to evolve the domain as a function of time to smoothly map the marker properties to the background grid.

Method of Manufactured Solutions

The Method of Manufactured Solutions (MMS) is used to help verify the implementation of the MPM. The MMS first takes a prescribed deformation field and to determine, analytically, what loading is required to achieve such a deformation. That loading is then applied to a numerical method, such as the MPM, in an effort to recover the original prescribed deformation and evaluate the fidelity of the method.

As given in [3, 6], the problem geometry chosen consists of a 2D ring. The deformation was chosen such that it is independent of the polar coordinate, Θ, and the only 'external' load is a body force.

The deformation combines a finite, shear deformation with a rigid body rotation. Therefore, this problem is a good test for the MPM in large deformation problems. Additional details on the problem are in the following chapter. Further information can be found in [3, 6].

Chapter 2

Problem Description

As stated above, the MMS takes a prescribed deformation and analytically determines the forcing function required to produce that deformation. The case below, as taken from [3, 6], requires a spatially varying body force to deform the model per it's prescribed deformation. Below we describe the geometry of the problem as well as provide some details on how the body force was derived.

Geometry

The problem consists of a ring with an inner radius of 0.75 m and an outer radius of 1.25 m. The deformation we want to recover is in the form of:

$$\alpha(R,t) = g(t)h(R) \tag{2.1}$$

where $\alpha(R,t)$ is the angular displacement, i.e. $\theta(R,t) = \alpha(R,t) + \Theta$ where $\theta(R,t)$ is the angular deformed coordinate and Θ is the undeformed angular coordinate. Note that the deformation, $\alpha(R,t)$ is independent of the angular coordinate, Θ.

In [6] the deformation is taken as:

$$g(t) = \frac{A}{2}(1-\cos(2\pi t))$$
$$h(R) = \left(1-32(R-1)^2+256(R-1)^4\right) = (16R^2-32R+15)^2$$

where A is the amplitude of deformation. For example, if we wanted a maximum angular displacement of 0.75 radians, we would chose $A = 0.75$. At the inner radius and outer radius of the ring, $h(R)$ was selected such that $h(R)$ and $h'(R)$ would both equal zero. As a result, as we will see below, the boundary conditions at the inner radius and outer radius are traction-free. Time varies from 0 to 1.0 seconds. Details on the derivation of the body force required to reproduce the deformation is given below.

A Neo-Hookean material model is used which is given as,

$$\boldsymbol{\sigma} = \frac{\lambda \ln(J)}{J}\mathbf{I} + \frac{\mu}{J}\left[\mathbf{F}\cdot\mathbf{F}^T - \mathbf{I}\right] \tag{2.2}$$

where the Cauchy stress is denoted by $\boldsymbol{\sigma}$, Lamé material constants are λ and μ, the deformation gradient is $\mathbf{F}$ and its determinant is J. λ and μ are taken to be 577 Pa and 385 Pa respectively. The initial density of the material is 1000 kg/m^3.

Body Force Derivation

References [3, 6] provide the details on deriving an expression for the body force. However, it seems that there may be typos in their final expression. As a result, beginning from an intermediate step, the derivation is reproduced below, hopefully without errors. If it is suspected that there are typos, there should be enough detail below to help guide the reader through the derivation process.

The deformation $\alpha(R,t)$ is a function of the radial coordinate, R and time, t. Both references agree with the form of the body forces which is given as,

$$b_r = -R\left(g'(t)h(R)\right)^2 - \frac{1}{\rho_o}\left[\left(\xi'(R)\frac{d\tau_{11}}{d\varepsilon} - h'(R)\tau_{21}\right)g(t) + \frac{1}{R}(\tau_{11}-\tau_{22})\right] \tag{2.3}$$

$$b_\theta = R\left(g''(t)h(R)\right) - \frac{1}{\rho_o}\left[\left(\xi'(R)\frac{d\tau_{21}}{d\varepsilon} + h'(R)\tau_{11}\right)g(t) + \frac{1}{R}(\tau_{12}+\tau_{21})\right] \tag{2.4}$$

where $\varepsilon(R,t) = g(t)\xi(R)$, $2\xi(R) = Rh'(R)$ and $\boldsymbol{\tau}$ is the first PK stress in the reference configuration.

As stated above, a Neo-Hookean constitutive relationship was used which is repeated here for convenience,

$$\boldsymbol{\sigma} = \frac{\lambda \ln(J)}{J}\mathbf{I} + \frac{\mu}{J}\left(\mathbf{F}\cdot\mathbf{F}^T - \mathbf{I}\right) \tag{2.5}$$

where $\boldsymbol{\sigma}$ is Cauchy Stress. The original derivation is not very clear on the relationship between Cauchy Stress and $\boldsymbol{\tau}$ used in equations 2.3 and 2.4. Therefore, with the help of [2], the relationship between the two parameters are derived as follows.

From [3, 6], the first PK stress (deformed coordinate system) is given as,

$$\mathbf{P} = \mathbf{r}\cdot\boldsymbol{\tau}\cdot\mathbf{q}^T \tag{2.6}$$

where $\mathbf{r}$ and $\mathbf{q}$ are orthogonal (rotation) tensors. From [2] the first PK stress is given by,

$$\mathbf{P} = J\boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \tag{2.7}$$

Equating equations 2.6 and 2.7 and solving for $\boldsymbol{\tau}$ results in,

$$\boldsymbol{\tau} = J\mathbf{r}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \cdot \mathbf{q}^{-T} = J\mathbf{r}^{T} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \cdot \mathbf{q} \tag{2.8}$$

Also given in the paper is a relationship between $\mathbf{F}$ and $\mathscr{F}$. Where $\mathbf{F}$ is the deformation gradient associated with $\mathbf{P}$ and $\mathscr{F}$ is the deformation gradient associated with $\boldsymbol{\tau}$ (different coordinate systems).

$$\mathbf{F} = \mathbf{r} \cdot \mathscr{F} \cdot \mathbf{q}^{T} \tag{2.9}$$

Using equation 2.9 and substituting equation 2.5 into 2.8 yields,

$$\boldsymbol{\tau} = \lambda \ln(J)\mathscr{F}^{-T} + \mu\left(\mathscr{F} - \mathscr{F}^{-T}\right) \tag{2.10}$$

From [3, 6], we note that $\mathscr{F} = \mathbf{I} + 2\varepsilon(R,t)\mathbf{e}_2\mathbf{e}_1$ and $\mathscr{F}^{-T} = \mathbf{I} - 2\varepsilon(R,t)\mathbf{e}_1\mathbf{e}_2$. This information allows us to write,

$$\boldsymbol{\tau} = \lambda \ln(J)\left(\mathbf{I} - 2\varepsilon(R,t)\mathbf{e}_1\mathbf{e}_2\right) + 2\mu\varepsilon(R,t)\left(\mathbf{e}_2\mathbf{e}_1 + \mathbf{e}_1\mathbf{e}_2\right) \tag{2.11}$$

With the above relationship we can finally substitute back into our expressions for the body force but we should first determine,

$$\frac{d\tau_{11}}{d\varepsilon} = 0 \quad \tau_{21} = 2\mu\varepsilon(R,t) \quad \tau_{11} - \tau_{22} = 0$$
$$\frac{d\tau_{21}}{d\varepsilon} = 2\mu \quad \tau_{11} = \lambda \ln(J) \quad \tau_{12} + \tau_{21} = 4\mu\varepsilon(R,t) - 2\lambda\varepsilon(R,t)\ln(J)$$

Now substituting the above expressions into equations 2.3 and 2.4,

$$b_r = -R\left(g'(t)h(R)\right)^2 + \frac{2\mu}{\rho_o}\varepsilon(R,t)h'(R)g(t) \tag{2.12}$$

$$b_\theta = R\left(g''(t)h(R)\right) - \frac{1}{\rho_o}\left[\left(2\mu\xi'(R) + \lambda \ln(J)h'(R)\right)g(t) + \frac{2\varepsilon(R,t)}{R}(2\mu - \lambda \ln(J))\right] \tag{2.13}$$

In order to get an expression in terms of $g(t)$ and $h(R)$ we substitute the expressions for $\xi(R)$ and $\varepsilon(R,t)$, and note that $2\xi'(R) = h'(R) + Rh''(R)$,

$$b_r = -R\left(g'(t)h(R)\right)^2 + \frac{\mu}{\rho_o} R\left(h'(R)g(t)\right)^2 \tag{2.14}$$

$$b_\theta = R\left(g''(t)h(R)\right) - \frac{\mu}{\rho_o}\left[3h'(R) + Rh''(R)\right] g(t) \tag{2.15}$$

Bear in mind, the above relationships are independent of the material parameter λ. Substituting the given expressions for $g(t)$ and $h(t)$, and realizing that,

$$h'(R) = -64(R-1) + 1024(R-1)^3$$
$$h''(R) = -64 + 3072(R-1)^2$$

results in:

$$b_r = -R(A\pi\sin(2\pi t))^2(16R^2 - 32R + 15)^4 + \frac{\mu A^2}{4\rho_o} R(1-\cos(2\pi t))^2\left(-64(R-1) + 1024(R-1)^3\right)^2 \tag{2.16}$$

$$b_\theta = 2RA\pi^2\cos(2\pi t)(16R^2 - 32R + 15)^2 - \frac{32\mu A}{\rho_o}\left(-45 + 188R - 240R^2 + 96R^3\right)(1-\cos(2\pi t)) \tag{2.17}$$

Equations 2.16 and 2.17 are our final expressions for the components of body force. It should be noted that Reference [3] uses a different $g(t)$. The spatially varying body force, as derived here, is incorporated into the MPM via the MRK_MMS.F file which is provided in Appendix B.

Please note that in order to convert from polar to cartesian coordinates, [3, 6] suggest we use the following relationships:

$$b_x = b_r\cos(\theta) - b_\theta\sin(\theta)$$
$$b_y = b_r\sin(\theta) + b_\theta\cos(\theta)$$

However, since CTH uses an Eulerian grid, the deformed coordinates are directly input into the MRK_MMS.F file and additional steps as suggested in [3, 6] are not needed.

Material Model

The CTH MPM currently supports the Mooney-Rivlin hyper-elastic material model, which is given by,

$$\boldsymbol{\sigma} = \frac{1}{J}\left[\ln(J)\frac{2}{J^{2/3}}(C_1 + \bar{I}_1 C_2)\mathbf{B} - \frac{2}{J^{4/3}}C_2\mathbf{B}\cdot\mathbf{B} - \frac{2}{3}(C_1\bar{I}_1 + 2C_2\bar{I}_2)\mathbf{I}\right] \tag{2.18}$$

where $\bar{I}_1 = J^{-2/3}I_1$ and $\bar{I}_2 = J^{-4/3}I_2$ and I_1 and I_2 are the first and second invariants, respectively, of the deformation gradient $\mathbf{F}$. User inputs are denoted by C_1 and C_2. $\mathbf{B}$ is the left Cauchy-Green Deformation tensor and it is related to the deformation gradient by $\mathbf{B} = \mathbf{F}\cdot\mathbf{F}^T$.

Since the Mooney-Rivlin model contains expressions for J, and $\mathbf{B}$ it can be easily modified to accommodate the Neo-Hookean Model cited above.

The modified Mooney-Rivlin code, MOONR.F, see Appendix C, now interprets the first user input, C_1, as λ and the second user input, C_2, as μ for the cases examined in this report.

This page intentionally left blank.

Chapter 3

Results

Inputs

While the outer diameter of the ring is 2.5 m, the problem domain was chosen to be 3 m wide and 3 m tall to avoid the possibility of any boundary effects influencing the solution due to radial expansion. A symmetric boundary condition was chosen for the four boundaries since it is not expected to influence the solution. The 2D domain was discretized with varying mesh densities and varying marker densities to determine the effects on the accuracy of the solution in this study.

The MMP4 (multiple material pressures) option was selected which mixes the cell pressure, sound speed and partition of PdV work. See [9] for details on the various MMP options implemented into CTH. The SPATB card, which was added to CTH for this study, was used to turn on the implementation of the spatially varying body force. For the marker options, the strength card (MSTREN) was set to 3, which is currently the only supported option and gives the standard MPM formulation. All failure logic was disabled by setting the FAIL card to 10. The MPMONLY card was set to 1 so that the acceleration field was found by using the full stress tensor. Unless explicitly stated otherwise, the MPCDI card was set to 2 for all cases which use the CPDI shape functions.

The Mie-Gruneisen equation-of-state (EOS) model was selected. In order to match the properties in [3], the EOS values were chosen as shown in Table 3.1. The Gruneisen coefficient was set to 0 in order to disable any energy dependence of the EOS.

Table 3.1. Mie-Gruneisen EOS Properties

Parameter	Value
ρ_0 (R0)	1.0 g/cm^3
T_0	0.02578 eV
C_s	116 cm/s
S_1	0
S_2	0
λ (G0)	0
C_v	1

Table 3.2. End Time of Simulation for Linear MPM cases

Mesh Density	Markers	Simulation end time (sec)	
		$A = 1.0$	$A = 0.5$
300x300	2	0.47	0.77
300x300	4	0.43	0.711
300x300	8	0.46	1.0
600x600	2	0.34	0.48
600x600	4	0.34	0.52
600x600	8	0.38	1.0
1200x1200	2	0.27	0.39
1200x1200	4	0.29	0.43
1200x1200	8	0.33	0.47

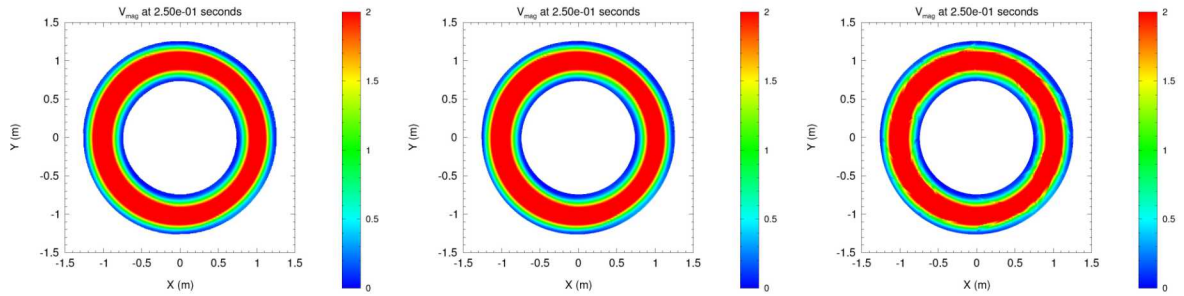
An example input deck using the above specified options is provided in Appendix A. The coding for computing the spatially varying body force, MRK_MMS.F, is provided in Appendix B and the modified material model file, MOONR.F, is provided in Appendix C.

Results

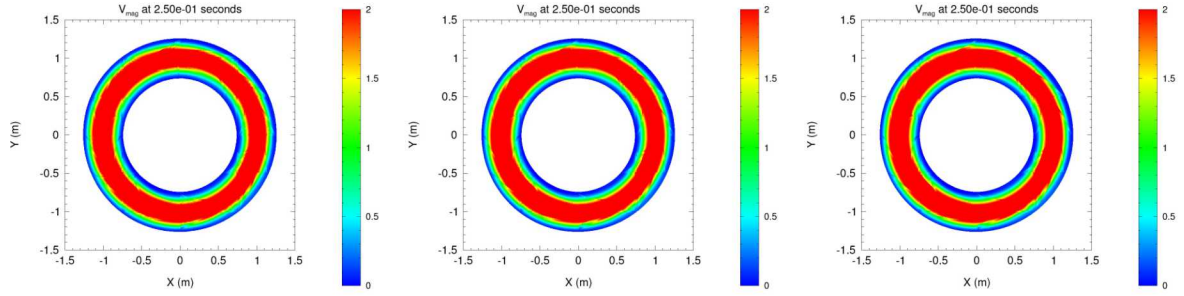
The maximum amplitude of angular deformation was initially chosen to be 1 radian, $A = 1.0$. While this is considered to be a very large deformation, the methods implemented in references [3, 6] were capable of supporting such a deformation, although the linear MPM produced clearly unphysical results. A total of nine runs were examined for this study. Three different mesh densities (300x300, 600x600 and 1200x1200) and three different marker per linear direction (2, 4 and 8) were selected.

Due to the large magnitude of deformation, none of the simulations were able to reach the termination time, 1.0 second. As seen in Table 3.2, only two simulations, the 300x300 mesh with 2 and 8 markers per linear direction cases, were able to run beyond 0.45 seconds. The 'negative residual volume' error eventually caused all of the simulations to fail. CTH Markers is considered to be a maturing capability and this error is a known issue that the CTH developers are investigating. The general trend from the end time of simulation seems to be that as the number of markers per linear direction is increased, the simulations run slightly longer, but this trend doesn't hold for the 300x300 mesh cases. No cases were run with 1 marker per cell because that configuration was found to be unstable for all cases.

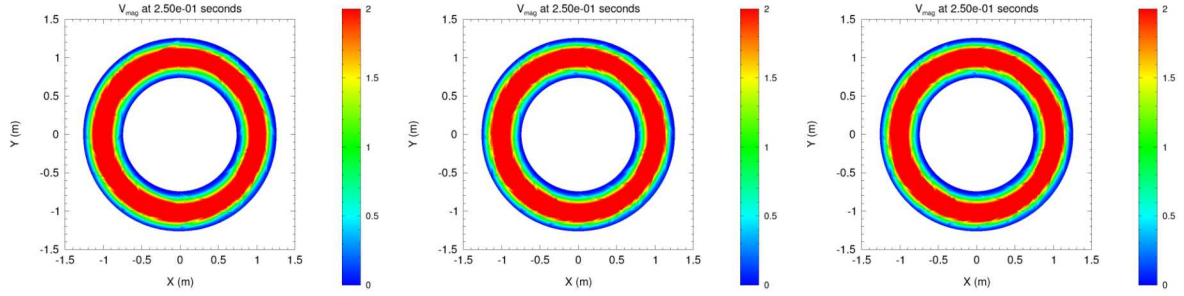
Figure 3.1 shows the response for each of the nine runs at 0.25 seconds. The values on the markers, and not the background grid are shown. It should be noted that edges between the red, green and blue areas are very smooth for the 300x300 mesh with 2 and 4 markers. The other mesh/marker densities show some roughness between the red/green and green/blue interfaces. This is due to aliasing due to the post-processing which occurs when there are several data points per



(a) Mesh: 300x300, Markers: 2 **(b)** Mesh: 300x300, Markers: 4 **(c)** Mesh: 300x300, Markers: 8



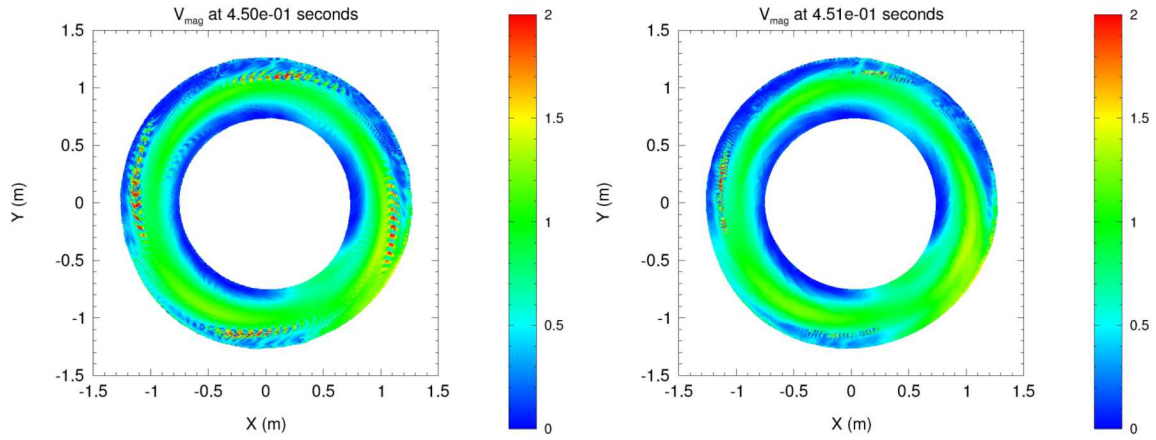
(d) Mesh: 600x600, Markers: 2 **(e)** Mesh: 600x600, Markers: 4 **(f)** Mesh: 600x600, Markers: 8



(g) Mesh: 1200x1200, Markers: 2 **(h)** Mesh: 1200x1200, Markers: 4 **(i)** Mesh: 1200x1200, Markers: 8

Figure 3.1. Response at 0.25 seconds, $A = 1$

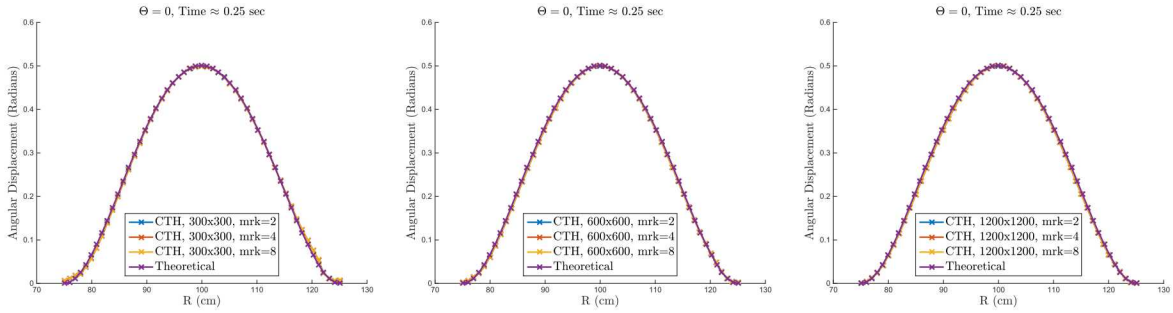
pixel in the images. The rough edges are not real. Other than the aliasing, there is very little discernible difference between the responses.



(a) Mesh: 300x300, Markers: 2 **(b)** Mesh: 300x300, Markers: 8

Figure 3.2. Response at 0.45 seconds, $A = 1$

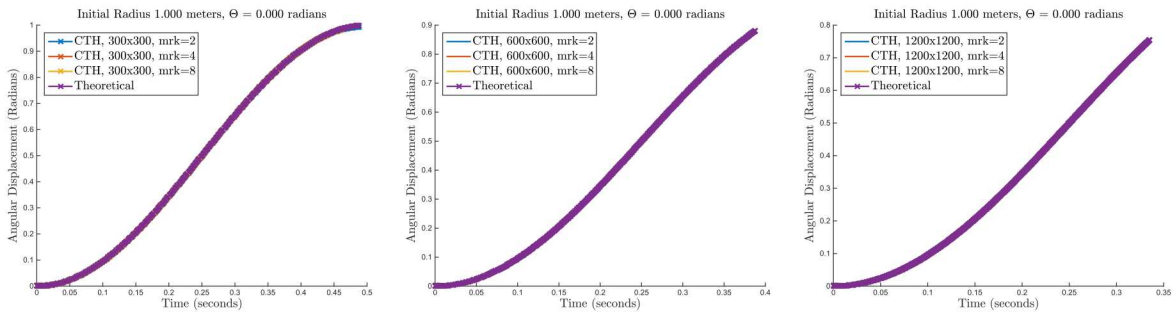
Figure 3.2 shows the marker response at 0.45 seconds of the only two runs that were able to make it that far, a mesh resolution of 300x300 with 2 and 8 markers. Between the two cases, there is a significant difference in the number of high velocity anomalies near the outer edge of the ring. All the cases which failed showed these anomalies before the 'negative residual volume' error terminated the simulations. Another concern is that the figures are not axisymmetric with respect to the angular polar coordinate as they are expected to be. Both figures show significant asymmetries at the bottom-right and bottom-left areas of the ring. This is an issue that should be investigated further as the implementation is matured.



(a) Mesh: 300x300 **(b)** Mesh: 600x600 **(c)** Mesh: 1200x1200

Figure 3.3. Response Profile at 0.25 seconds and $\Theta = 0$, $A = 1$

Figure 3.3 shows the deformation at $\Theta = 0$ at one snapshot in time compared to the prescribed deformation, $\alpha(R,T)$. From this figure, there is very little discernible difference between the exact



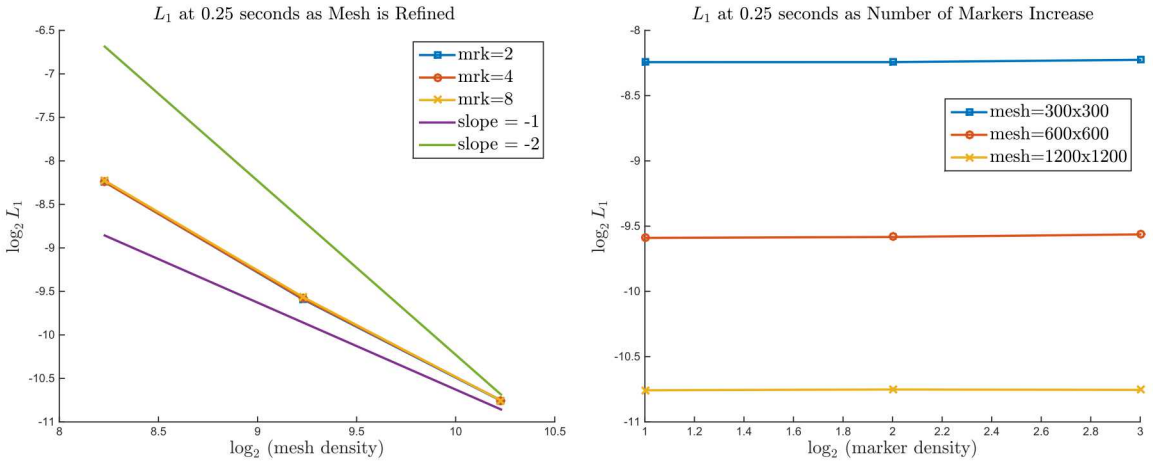
(a) Mesh: 300x300 **(b)** Mesh: 600x600 **(c)** Mesh: 1200x1200

Figure 3.4. Response Profile at $\Theta = 0, R = 1.0$ over time, $A = 1$

solution and the CTH MPM result. Figure 3.4 shows the deformation of one point over time. Again, we see very little difference between runs. Therefore, to better quantify the differences between the runs, a L_1 error was calculated for all the cases. The L_1 error is defined as,

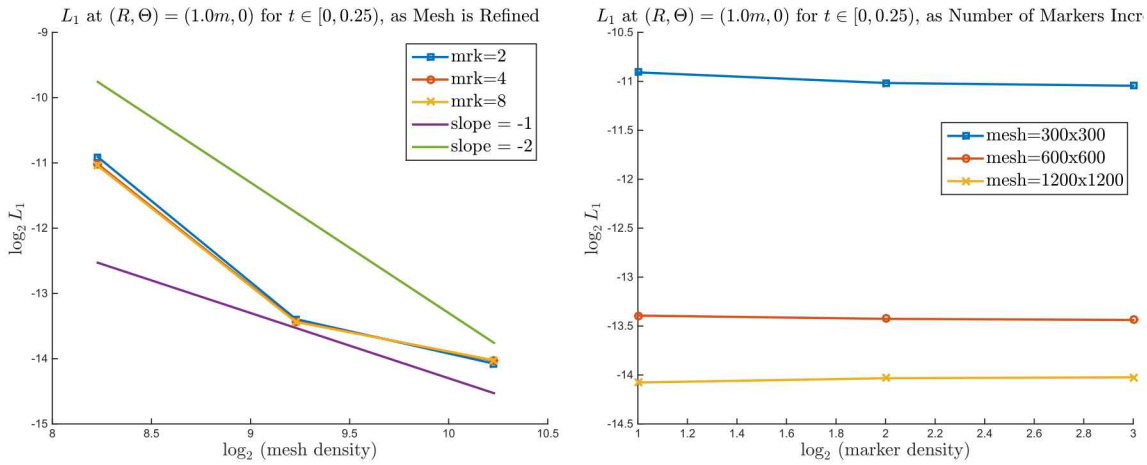
$$L_1 = \frac{\sum_{i=1}^{N} |x_{simulation} - x_{exact}|}{N} \tag{3.1}$$

where the vertical bars denote absolute value, N is the number of points, $x_{simulation}$ is a parameter output by the simulation and x_{exact} is the value of the parameter we expect using the previous section's $\alpha(R,t)$. Figure 3.3 shows the profiles used and Figure 3.5 is the corresponding L_1 plot. Figure 3.4 shows the profiles used and Figure 3.6 is the corresponding L_1 plot.



(a) L_1 as Mesh Density Increases **(b)** L_1 as Marker per Linear Direction Increases

Figure 3.5. L_1 at 0.25 seconds, $\Theta = 0$ and $A = 1$



(a) L_1 as Mesh Density Increases

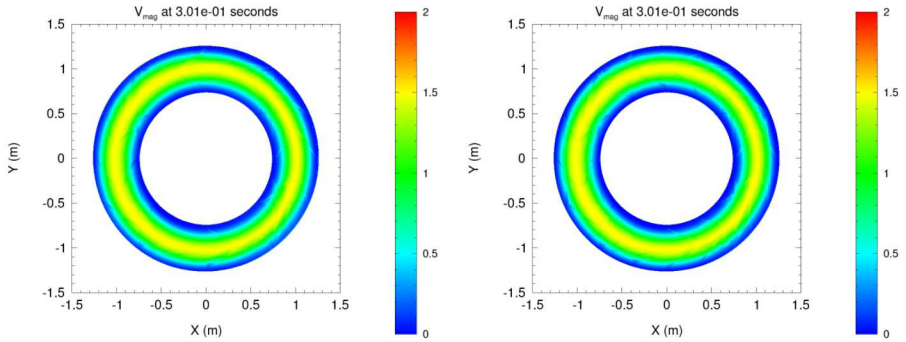
(b) L_1 as Marker per Linear Direction Increases

Figure 3.6. L_1 at $(R,\Theta) = (1.0, 0)$, for $t \in [0, 0.25)$ seconds, and $A = 1$

From Figure 3.5(a), the error decreases with a slope of about 1.25 as the mesh is refined. We see no convergence in the solution as we increase the number of markers per linear direction, as shown in Figure 3.6(b). However, from Table 3.2, we see that the number of markers does provide some stability as the simulations generally run for longer periods of time if there are more markers. The lack of additional convergence as the marker density is increased implies that the additional marker density does not provide any additional sub-cell resolution of the conservation equations.

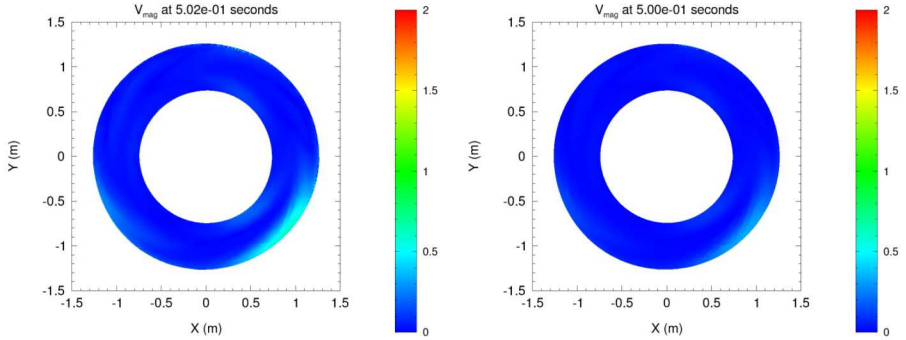
Since the simulations were not able to complete a full run, a lower maximum amplitude of 0.5 radians, $A = 0.5$, was evaluated in order to get most of the cases to run the full simulation time. Table 3.2 shows the time at which simulations terminated for the lower amplitude case. Two simulations were able to reach the final time, both with 8 markers, the 300x300 and 600x600 meshes. Surprisingly, the 1200x1200 mesh with 8 markers was not able to run the entire 1.0 seconds.

Figure 3.7 shows the velocity magnitude over time of the 300x300 and 600x600 meshes with 8 markers per linear direction. At 0.3 seconds, Figures 3.7(a) and 3.7(b), show that there is very little difference between the two runs. Figures 3.7(c) and 3.7(d) show the response at about 0.5 seconds. This time step is when the deformation is at the maximum amplitude and we expect a uniform zero velocity. However, the figures do not reflect that. There is a significant asymmetric nonzero velocity at the bottom-right of the figure for the 300x300 mesh. It seems that the magnitude of the velocity decreases as we refine the mesh to 600x600. At 0.7 seconds, Figures 3.7(e) and 3.7(f) show that there is a significant difference in the same bottom-right region of the ring. Finally at the end time of the simulation, 1.0 seconds, we expect a nonzero velocity throughout the ring, but Figures3.7(g) and 3.7(h) show that there is a significant non zero velocity around the ring for the 300x300 mesh which seems to be reduced for the 600x600 mesh.



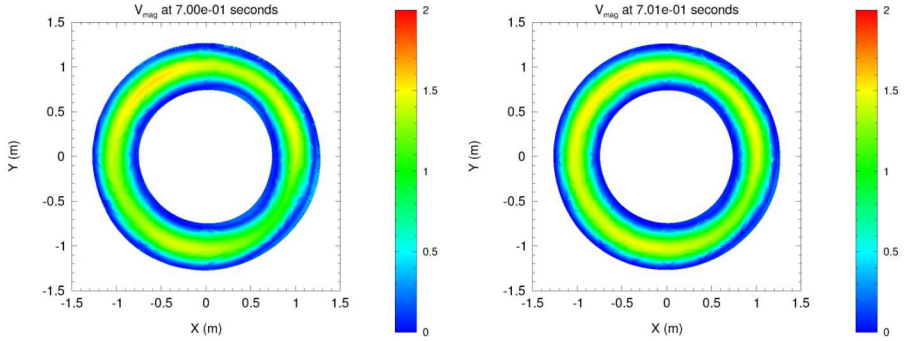
(a) Mesh: 300x300, t=0.3 seconds

(b) Mesh: 600x600 t=0.3 seconds



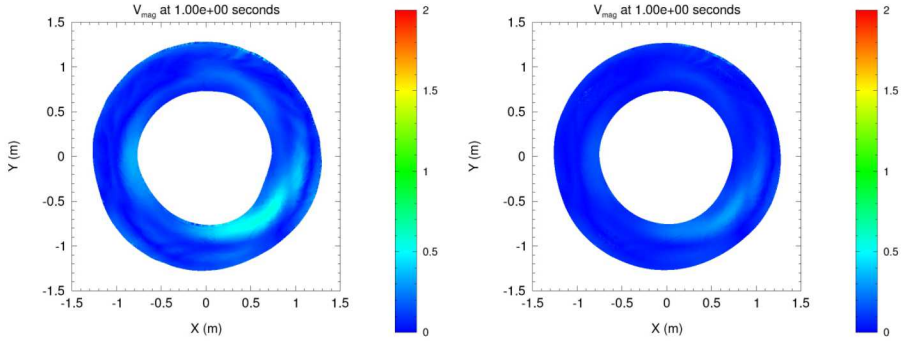
(c) Mesh: 300x300, t=0.5 seconds

(d) Mesh: 600x600, t=0.5 seconds



(e) Mesh: 300x300, t=0.7 seconds

(f) Mesh: 600x600, t=0.7 seconds



(g) Mesh: 300x300, t=1.0 seconds

(h) Mesh: 600x600, t=1.0 seconds

Figure 3.7. Response of 300x300 mesh and 600x600 mesh, both with 8 markers. $A = 1/2$

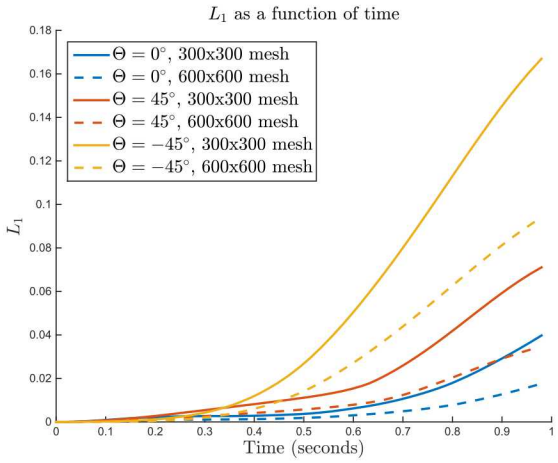
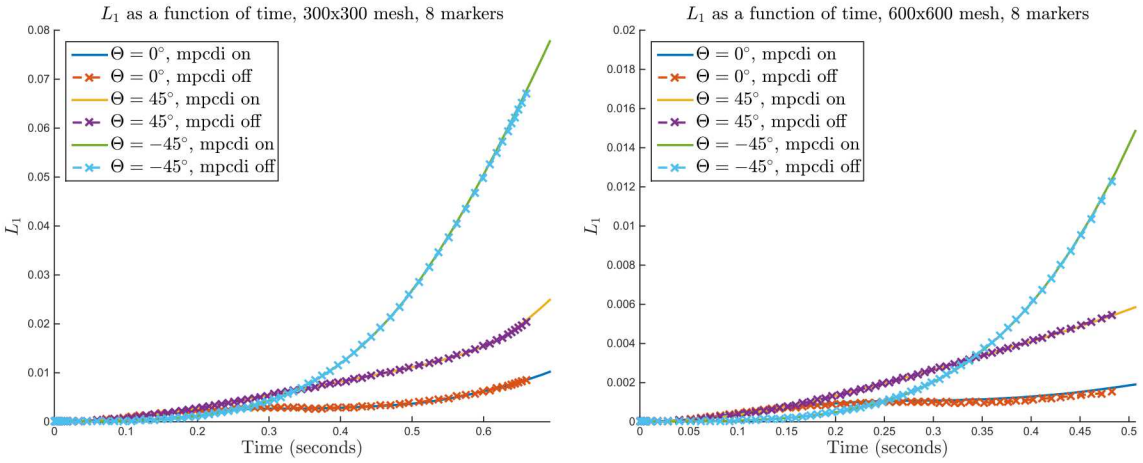


Figure 3.8. L_1 over time, comparing mesh densities, $A = 1/2$

Figure 3.8 shows the L_1 error calculated for a cross section deformation, at three different angles, for the 300x300 and 600x600 mesh with 8 markers as a function of time. The figure clearly shows that the refinement of the mesh reduces the error. Also, the error is different at $\Theta = 0, 45°$ and $-45°$. This asymmetry may be due to an error in the CTH Markers source code or an error in the implementation of the MMS methodology. Further investigation is required.



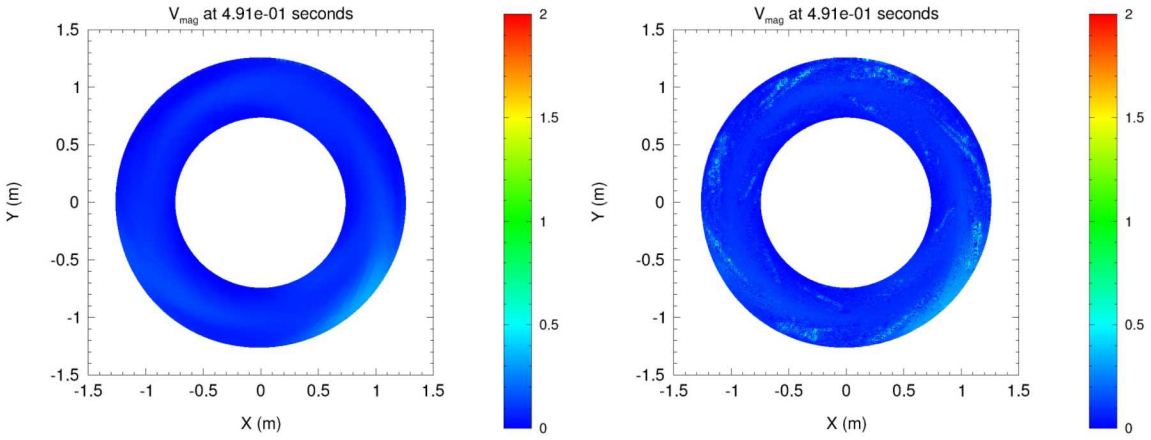
(a) L_1 for 300x300 mesh and 8 markers

(b) L_1 for 600x600 mesh and 8 markers

Figure 3.9. L_1 over time, testing mpcdi card, $A = 1/2$

Since two runs were able to complete the entire 1.0 second requirement, it makes it a good candidate to test the MCPDI card to examine the effects of the CPDI shape functions. Figure 3.9 shows the L_1 error calculated for a cross section deformation as a function of time comparing the MCPDI switch on and off. The figure shows that with the switch on or off the error values are

indistinguishable. However, with the MCPDI switch on, the simulations were not able to reach the 1.0 second mark. For 300x300 mesh with 8 markers, it was only able to reach 0.66 seconds and the 600x600 mesh was able to reach only 0.49 seconds. Figure 3.10 shows the final timestep of the 600x600 mesh with 8 markers with MCPDI off compared with MCPDI on. There is a stark contrast between the two figures. While the MCPDI on figure is much smoother, the MCPDI off figure shows significant velocity anomalies.



(a) MCPDI on

(b) MCPDI off

Figure 3.10. Comparing MCPDI card, Velocity Magnitude for 600x600 mesh, 8 markers, $A = 1/2$

This page intentionally left blank.

Chapter 4

Conclusion

In this study, the MMS was used to help in the verification process of the MPM implementation in CTH. In general, CTH Markers was able to reproduce the given deformation well. But, due to stability and robustness issues within CTH Markers, most of the simulations were unable to run to completion which made a rigorous verification impossible.

There was significant change in the calculated error as the mesh was refined, as expected. The L_1 error converged with an order of approximately 1.25. In contrast, there was no observed convergence as the marker density was increased. In most cases the increased marker density did seem to have a stabilizing effect on the simulations, allowing them to run for a longer period of time until the instability ended them. Regardless of the number of markers, the simulations fail due to a 'negative residual volume' error. The developers of CTH are aware of this issue and are investigating.

In order to better evaluate the system, the maximum amplitude was decreased. Two simulations were able to complete the entire 1.0 second with this decreased amplitude. Unfortunately, we expected an axisymmetric solution but CTH did not produce such a result. There are many potential sources of error as to the cause of the asymmetry. They include, but are not limited to an error in body force, an error in material model or an error in the CTH marker implementation.

In the future as the MPM capability in CTH is matured, this verification should be revisited to asses the fidelity of new methods/options as they are added to the implementation. The robustness and stability issues, along with the asymmetry should be further examined to determine the source of the issues.

It should also be noted that this report only examined the solid mechanics response of the MPM in CTH. The shock response, or the fracture behavior of the method were not assessed in anyway and extrapolation of the results presented in this report to those regimes is not valid. The use of the MPM for shock driven behavior is an ongoing area of research for the community and poses several challenges beyond just the solid mechanics response of the method.

This page intentionally left blank.

References

[1] R. M. Brannon A. Sadeghirad and J. Burghardt. A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. *International Journal for Numerical Methods in Engineering*, 86(12):1435–1456, 2011.

[2] Robert Asaro and Vlado Lubarda. *Mechanics of solids and materials*. Cambridge University Press, 2006.

[3] Rebecca M Brannon, Krishna Kamojjala, and Alireza Sadeghirad. Establishing credibility of particle methods through verification testing. In *II international conference on particle-based methods - fundamentals and applications*, 2011.

[4] Zhen Chen and Rebecca Brannon. An evaluation of the material point method. Technical report SAND2002-0482, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, February 2002.

[5] Zhen Chen Deborah Sulsky and Howard L. Schreyer. A particle method for history-dependent materials. Technical report SAND93-7044, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, June 1993.

[6] K Kamojjala, R Brannon, A Sadeghirad, and J Guilkey. Verification tests in solid mechanics. *Engineering with Computers*, 31(2):193–213, 2013.

[7] Robert M. Kirby Michael Steffen and Martin Berzins. Analysis and reduction of quadrature errors in the material point method (mpm). *International Journal for Numerical Methods in Engineering*, 76(6):922–948, 2008.

[8] R. G. Schmitt, A. L. Brundage, D. A. Crawford, E. N. Harstad, K. Ruggirello, S. C. Schumacher, and J. S. Simmons. Cth user's manual and input instructions, version 11.1. Technical report, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, May 2015.

[9] Shane C. Schumacher and Kevin P. Ruggirello. Cth reference manual: Marker technologies. Technical report SAND2013-1675, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, March 2013.

This page intentionally left blank.

Appendix A

CTH Input Deck

The input deck used for a 600x600 mesh with 8 markers per linear direction and mpcdi on is given below.

A.1 disk_bodyforce.in

```
*CTHid SPYPLT SPYHIS
*eor*cthin
*
 2D MMS Problem *
*
************************ control block ******************************
control
        tstop = 1.0e0
        tbad 1E30
        print
        mmp4
        spatb=1 *turn on space-vary body force
        qfl 1
    rdumpf 1e30
endcontrol
*
mesh
        block geometry=2dr
         x0=-150
          x1 n=600 w=300 rat=1
         endx
         y0=-150
          y1 n=600 w=300 rat=1
         endy
        xact = -150 150
        yact = -150 150
        endb
        endmesh
*
```

```
*-------------------------------------------------------------------
*
spy
 Save("ALL");
 SaveTime(0, 1.0e-2);
 PlotTime(0, 1.0e-2);
 ImageFormat(1600,1200);

 define main()
 {
        pprintf(" PLOT: Cycle=%d, Time=%e\n",CYCLE,TIME);
        MatColors(GREEN);
        MatNames("Tungsten");

        XLimits(-150,150);
        YLimits(-150,150);

        Image("mrks",WHITE,BLACK);
         Window(0.,0.,0.85,1.);
         Label(sprintf("V`mag` at %6.2e seconds",TIME));
         HotMap;
         ColorMapRange(0.,2E2);
         PaintMarkers(0,"VMAG",5,0);
         DrawColorMap("V`MAG` (cm/s)",0.85,0.1,1.,1.);
        EndImage;

        Image("mrksVX",WHITE,BLACK);
         Window(0.,0.,0.85,1.);
         Label(sprintf("V`X` at %6.2e seconds",TIME));
         HotMap;
         ColorMapRange(-5.0E1,5.0E1);
         PaintMarkers(0,"VX",5,0);
         DrawColorMap("V`X` (cm/s)",0.85,0.1,1.,1.);
        EndImage;

        Image("mrksVY",WHITE,BLACK);
         Window(0.,0.,0.85,1.);
         Label(sprintf("V`Y` at %6.2e seconds",TIME));
         HotMap;
         ColorMapRange(-5.0E1,5.0E1);
         PaintMarkers(0,"VY",5,0);
         DrawColorMap("V`Y` (cm/s)",0.85,0.1,1.,1.);
        EndImage;

        Image("mrksDENS",WHITE,BLACK);
         Window(0.,0.,0.85,1.);
         Label(sprintf("Density at %6.2e seconds",TIME));
```

```
         HotMap;
         ColorMapRange(0.9,1.1);
         PaintMarkers(0,"DENS",4,0);
         DrawColorMap("Density (g/cm^3)",0.85,0.1,1.,1.);
        EndImage;

        Image("mrks_domains",WHITE,BLACK);
         Window(0.,0.,1.,1.);
         XLimits(115,125);
         YLimits(0,10);
         Label(sprintf("CPDI Domains at %6.2e seconds",TIME));
         DrawMarkerDomains(1);
        EndImage;

  }

  HisCycle(0,1);
  SaveHis("POSITION,VX,VY");
  SaveTracer(ALL);

  define spyhis_main()
  {
    %HisLoad(1,"hscth");
    %HisImageName("profile");
  }

endspy
*
*-------------------------------------------------------------------------
*
diatom
*
        package 'impactor'
                material 1
                dens=1.0
                insert circle
                        center 0. 0.
                        radius 125.0
                        rinner 75.0
            endinsert
        endpackage
enddiatom
*
mark * Marker section
        mmat 1 8 * Mat 1 is Marker mat, 8 Markers per lin dir
        mcpdi 2
        esmth 0
```

```
        vsmth 0
        ssmth 0
        stren 3 * Controls strength options, 3 is standard MPM
        fail 10 * turns off failure
        mpmonly 1
endm
*
eos
        mat1 mgr user R0=1 T0=0.02578 CS=116 S1=0 S2=0 G0=0 CV=1
endeos
*
epdata
    vpsave
        mix=3
        matep=1 moonr user MR1=577.0D1 MR2=385.0D1 MRU=0 MRG=0 MRT=0 MRF=0
endep
*
tracer
  add 75,0 to 125,0 num=51
  add 106.066,-106.066 to 176.777, -176.777 num=51
  add 106.066, 106.066 to 176.777, 176.777 num=51
endtracer
*
convct
        convect=1
        interface=smyra
endc
*
edit
*
    exact
        shortt
                tim=0. dt=1.0e30
        ends
        longt
                tim=0. dt=1.0e30
        endl
        plott
                time=0. dtfreq=10.0e30
        endp
        histc
                cycle=0
                dc=1000
                htracer all
        endh
        *
        endedit
```

```
  *
  restt
    time = 0.  dtfreq=1.0e30
  endrestt
  fracts
          stress
          pfrac1=-13e9
  endf
  *
  boundary
   bhydro
bxbot 0
bxtop 0
bybot 0
bytop 0
   endh
  endb
```

This page intentionally left blank.

Appendix B

Spatial Body Force Input

Below is the MRK_MMS.F file. This file applied the spatially varying body force within CTH Markers. The version below is for a maximum angular displacement of 1.0 radian, $A = 1$.

B.1 MRK_MMS.F

```
       SUBROUTINE MRK_MMS(KPLANE,SBF)
C
      use blank_module
      use kpoint_module
      use dbcxyz_module
C
#include "impdoubl.h"
#include "dbcsdm.h"
#include "dbcxyz.h"
#include "comint.h"
#include "parpie.h"
C parpie.h defines the pi, the mathematical constant
#include "comrel.h"
C comrel.h imports time
C Following Section 6.1 of Kamojjala, K., et. al.,
C "Verification␣Tests␣in␣solid␣mechanics",
C Engineering with Computers,(2015) 31:193-213,
C DOI 10.1007/s00366-013-0342-x
C
      DIMENSION SBF(IMAX,JMAX,3)
C
      PARAMETER(PONE=1.0D0, PTWO=2.0D0, PIPI=PPIE**2,
     1  PFOUR=4.0D0, PHALF=0.5D0)
C
      IJMAX = IMAX*JMAX

C RHO0=1000 KG/M^3
C XMU = 385 Pa
C g(t) = (A/2)*(1-cos(2*pi*t))
```

```
C h(R) = (1 - 32*(R-1)^2 + 256*(R-1)^4)
C \alpha(R,t)=g(t)*h(R)
C I CONVERTED ALL OF THE CTH UNITS INTO
C METERS AND Pa, AND PERFORMED THE CALCULATION
C THEN I CONVERTED THE FINAL BODY FORCE (ACCELERATION)
C RESULT BACK TO CTH UNITS
          RHO0 = 1000.0D0
          XMU = 385.0D0
          AA = PONE
       DO J=1,JMAX
         DO I=1,IMAX
            XLOC=(X(I)+X(I+1))/(PTWO*100D0)
            YLOC=(Y(J)+Y(J+1))/(PTWO*100D0)
C           Divide XLOC by 100 in order to get units right
C           Units: Paper - meters, CTH - cm, here - meter

            XRR = ((XLOC)**2 + (YLOC)**2)**0.5D0
            IF(XRR.LT.(1.25D0) .AND. XRR.GT.(0.75D0))THEN
            THETA_CAP = DATAN2(YLOC, XLOC)
C
C           TEMPORARY VARIABLES
C
            BR1 = -PONE*((AA*PPIE*DSIN(PTWO*PPIE*TIME))**2)
     1             *XRR*((15D0 - 32.0D0*XRR
     1                    + 16.0D0*XRR**2)**4)

            BR2 = ((AA*(PONE-DCOS(PTWO*PPIE*TIME))/PTWO)**2)
     1             *(XMU/RHO0)*XRR*(-64.0D0*(XRR-PONE)
     1             + 1024.0D0*(XRR-PONE)**3)**2
C

            BTHETA1 = XRR*((15.0D0-32.0D0*XRR+16.0D0*XRR**2)**2)
     1                 *(PTWO*AA*PIPI*DCOS(PTWO*PPIE*TIME))

            BTHETA2 = -64.0D0*(XMU/RHO0)*(-45.0D0 + 188.0D0*XRR
     1                - 240.0D0*XRR**2 + 96.0D0*XRR**3)
     1                *(AA/PTWO)*(PONE-DCOS(PTWO*PPIE*TIME))

C              CALCULATING POLAR BODY FORCES

            B_RADIAL = BR1 + BR2
            B_THETA = (BTHETA1 + BTHETA2)
            ALPHA = (AA/PTWO)*(PONE-DCOS(PTWO*PPIE*TIME))
     1               * ( PONE- 32.0D0*(XRR-PONE)**2
     1                + 256.0D0*(XRR-PONE)**4)
            THETA = THETA_CAP + ALPHA
```

```
C                CALCULATING CARTESIAN BODY FORCES

C                   BX = B_RADIAL*DCOS(THETA) - B_THETA*DSIN(THETA)
C                   BY = B_RADIAL*DSIN(THETA) + B_THETA*DCOS(THETA)

C                CALCULATING CARESTIAN BODY FORCES IN UNDEFORMED
C                COORDINATE SYSTEM

              BX = B_RADIAL*DCOS(THETA_CAP) - B_THETA*DSIN(THETA_CAP)
              BY = B_RADIAL*DSIN(THETA_CAP) + B_THETA*DCOS(THETA_CAP)

C Multiply by 100 below in order to convert units
C Units: paper - meters/sec^2, CTH - cm/sec^2, above - meters/sec^2
                BX2 = 100.0D0*BX
                BY2 = 100.0D0*BY
              ELSE
                BX2=0.0D0
                BY2=0.0D0
              ENDIF
                SBF(I,J,1)=BX2
                SBF(I,J,2)=BY2
                  ENDDO
      ENDDO

      RETURN
      END
```

This page intentionally left blank.

Appendix C

Modified Material Model Input

Below is the modified material file used to model the Neo-Hookean Material. Changes are identified by 'VVN'.

C.1 MOONR.F

```
C $Id: moonr.F,v 1.14 2014/11/06 16:08:13 scschum Exp $
C
      SUBROUTINE MOONR_INIT (IMAT,RNUSET,UI,VI)
C***********************************************************************
C
C     Initializes and checks validity of user inputs for Mooney-Rivlin
C     model.  Calculates and stores derived material constants.
C
C************************ abc mm/yy ************************************
C
C  written: mm/dd/yy
C  author:  Shane Schumacher
C
C  who mm/dd/yy                  m o d i f i c a t i o n
C  --- -------- -----------------------------------------------------
C
      use eosmig
      use marker
C
#include "impdoubl.h"
#include "iofils.h"
#include "maxmat.h"
#include "comint.h"
#include "elunep.h"
#include "elunvp.h"
#include "elunfr.h"
#include "elunav.h"
#include "psr.h"
#include "rotate.h"
```

```
#include "strain.h"
#include "enbdbc.h"
#include "matcona.h"
#include "elunfm.h"
#include "mclass.h"
#include "epeos.h"
#include "unitz.h"
#include "eldbsl.h"
#include "datpath.h"
#include "vpfile.h"
#include "migpar.h"
#include "mig.h"
#include "isvdat.h"
#include "adatf.h"
#include "sccchr.h"
#include "mpcthl.h"
#include "parpie.h"
C
      DIMENSION UI(*)
      DIMENSION VI(10)
C
      LOGICAL RNUSET(MAXMAT)
C
      PARAMETER ( PZERO  = 0.0D0  )
      PARAMETER ( PHALF  = 1.0D0/2.0D0 )
      PARAMETER ( PONE   = 1.0D0  )
      PARAMETER ( PTWO   = 2.0D0  )
      PARAMETER ( PTHREE = 3.0D0  )
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      IF(MARK_ON.EQ.0.OR.MARK_STREN.NE.3)THEN
       CALL FATERR('MOON_INIT','Marker␣and␣MPM␣only␣model')
      ENDIF
C
      YLDVM(IMAT) = PONE
C
      RNUSET(IMAT) = .TRUE.
C
      MARKMAT(IMAT)%MARK_FINT=1
C
C     Get initial density and bulk sound speed values from l data
C     to define an effective Poisson's␣ratio␣for␣timestep
C␣␣␣␣␣control␣(see␣coding␣and␣comments␣in␣FBCHK):
C
␣␣␣␣␣␣IF(NEQ(IMAT,1).EQ.2)THEN
␣␣␣␣␣␣␣ILOCATE=LCFW(IMAT,1)
```

```
␣␣␣␣␣␣␣RHO=UIEQ(3,IMAT,1)
␣␣␣␣␣␣␣CSB=DCEQ(ILOCATE+3)
␣␣␣␣␣␣ELSEIF(NEQ(IMAT,1).EQ.1)THEN
␣␣␣␣␣␣␣RHO=MAX(UIEQ(1,IMAT,1),UIEQ(15,IMAT,1))
␣␣␣␣␣␣␣CSB=UIEQ(3,IMAT,1)
␣␣␣␣␣␣ELSE
␣␣␣␣␣␣␣WRITE(KPT6,4330)
4330␣␣␣FORMAT('INVALID EOS SELECTION FOR MOONR MODEL')
␣␣␣␣␣␣CALL␣FATERR('MOONR_INIT','INVALID EOS SELECTION FOR MOONR MODEL')
␣␣␣␣␣␣ENDIF
C

C␣␣␣␣␣␣RSHEAR=PTWO*(UI(1)+UI(2))
C␣␣␣␣␣␣RBULK=RHO*CSB**PTWO
C␣VVN␣-␣NEO-HOOKEAN␣MODEL␣AS␣GIVEN␣IN␣PAPER
C␣VVN␣UI(1)␣=␣lambda,␣UI(2)␣=␣mu␣(lambda␣and␣mu
C␣VVN␣are␣lame␣parameters)
␣␣␣␣␣␣RSHEAR␣=␣UI(2)
␣␣␣␣␣␣RBULK␣=␣UI(1)␣+␣(PTWO/PTHREE)*UI(2)
C␣VVN
C
␣␣␣␣␣␣RNUZRO(IMAT)=(PTHREE*RBULK-PTWO*RSHEAR)/
␣␣␣␣␣&␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣(PTWO*(PTHREE*RBULK+RSHEAR))
␣␣␣␣␣␣IF(RNUZRO(IMAT).GT.PHALF.OR.RNUZRO(IMAT).LT.-PONE)THEN
␣␣␣␣␣␣␣CALL␣FATERR('MOONR_INIT',
␣␣␣␣␣&␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣'EOS & Strength parameters incompatible')
␣␣␣␣␣␣ELSE
␣␣␣␣␣␣␣IF(LPARNT)␣WRITE(KPT6,4321)␣IMAT,␣RNUZRO(IMAT)
4321␣␣␣␣FORMAT('Poissons Ratio for MOONR Mat',I2,
␣␣␣␣␣&␣␣␣␣␣␣␣␣␣' is ',E13.6)
␣␣␣␣␣␣ENDIF
␣␣␣␣␣␣VI(1)=RNUZRO(IMAT)
C
␣␣␣␣␣␣RETURN
␣␣␣␣␣␣END
C
␣␣␣␣␣␣SUBROUTINE␣MOONR_MAIN(MSF,␣NGS,␣MAT,␣UI,␣ICYCLE,␣DT,
␣␣␣␣␣*␣MGSMAP,␣DAM,␣SENRG,␣SCR)
C
*******************␣MOONEY␣RIVLIN␣MAT␣PACKAGE␣*****************
C
C␣␣␣␣␣This␣program␣determines␣the␣new␣stress␣for␣material␣(modlep=31)
C
************************␣SCS␣08/06␣***********************************
C
␣␣␣␣␣␣use␣marker
C
```

```
#include␣"impdoubl.h"
C
C␣*****␣parameter␣arrays␣*****
C
␣␣␣␣␣␣DIMENSION␣UI(*)
␣␣␣␣␣␣DIMENSION␣DAM(MSF)
␣␣␣␣␣␣DIMENSION␣SENRG(MSF)
␣␣␣␣␣␣DIMENSION␣SCR(MSF,9)
␣␣␣␣␣␣DIMENSION␣MGSMAP(MSF)
C
␣␣␣␣␣␣DIMENSION␣F(3,3)
␣␣␣␣␣␣DIMENSION␣FT(3,3)
␣␣␣␣␣␣DIMENSION␣FB(3,3)
␣␣␣␣␣␣DIMENSION␣FBT(3,3)
␣␣␣␣␣␣DIMENSION␣B(3,3)
␣␣␣␣␣␣DIMENSION␣BB(3,3)
C
␣␣␣␣␣␣CHARACTER*10␣IAM
␣␣␣␣␣␣CHARACTER␣STRN*80
C
C␣␣␣␣␣␣␣define␣parameters
␣␣␣␣␣␣PARAMETER␣(␣PZERO␣␣=␣␣0.0D0␣␣␣)
␣␣␣␣␣␣PARAMETER␣(␣PHALF␣␣=␣␣0.5D0␣␣␣)
␣␣␣␣␣␣PARAMETER␣(␣PONE␣␣␣=␣␣1.0D0␣␣␣)
␣␣␣␣␣␣PARAMETER␣(␣PTWO␣␣␣=␣␣2.0D0␣␣␣)
␣␣␣␣␣␣PARAMETER␣(␣PFOUR␣␣=␣␣4.0D0␣␣␣)
␣␣␣␣␣␣PARAMETER␣(␣PSIX␣␣␣=␣␣6.0D0␣␣␣)
␣␣␣␣␣␣PARAMETER␣(␣PTHREE␣=␣␣3.0D0␣␣␣)
␣␣␣␣␣␣PARAMETER␣(␣PFIFTY␣=␣␣50.0D0)
␣␣␣␣␣␣PARAMETER␣(␣BOLTZ␣=␣1.602180E-12)
␣␣␣␣␣␣PARAMETER␣(␣AVAG␣␣=␣6.0221367D23)
␣␣␣␣␣␣PARAMETER␣(␣TOU␣␣␣=␣1.0D-13␣)
␣␣␣␣␣␣PARAMETER␣(␣POFE␣␣␣␣=␣1.58D+0␣␣)
␣␣␣␣␣␣PARAMETER␣(␣LIMIT␣=␣50.0D+0)
␣␣␣␣␣␣PARAMETER␣(␣DFAIL␣=␣0.9999D0)
C
␣␣␣␣␣␣DATA␣IAM/'MOONR_MAIN'/
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
␣␣␣␣␣␣OVRFLWFLAG=PZERO
C
␣␣␣␣␣␣DO␣IGS=1,NGS
␣␣␣␣␣␣␣M=MGSMAP(IGS)
C
␣␣␣␣␣␣␣F(1,1)=MARKMAT(MAT)%F(M,1,1)
␣␣␣␣␣␣␣F(1,2)=MARKMAT(MAT)%F(M,1,2)
```

```
␣␣␣␣␣␣␣␣F(1,3)=MARKMAT(MAT)%F(M,1,3)
␣␣␣␣␣␣␣␣F(2,1)=MARKMAT(MAT)%F(M,2,1)
␣␣␣␣␣␣␣␣F(2,2)=MARKMAT(MAT)%F(M,2,2)
␣␣␣␣␣␣␣␣F(2,3)=MARKMAT(MAT)%F(M,2,3)
␣␣␣␣␣␣␣␣F(3,1)=MARKMAT(MAT)%F(M,3,1)
␣␣␣␣␣␣␣␣F(3,2)=MARKMAT(MAT)%F(M,3,2)
␣␣␣␣␣␣␣␣F(3,3)=MARKMAT(MAT)%F(M,3,3)
C␣␣␣␣␣␣␣␣Det(F)=J
␣␣␣␣␣␣␣␣RJ=F(1,1)*F(2,2)*F(3,3)+
␣␣␣␣␣␣&␣␣␣␣␣␣F(1,2)*F(2,3)*F(3,1)+
␣␣␣␣␣␣&␣␣␣␣␣␣F(1,3)*F(2,1)*F(3,2)-
␣␣␣␣␣␣&␣␣␣␣␣␣F(1,1)*F(2,3)*F(3,2)-
␣␣␣␣␣␣&␣␣␣␣␣␣F(1,2)*F(2,1)*F(3,3)-
␣␣␣␣␣␣&␣␣␣␣␣␣F(1,3)*F(2,2)*F(3,1)
C
␣␣␣␣␣␣␣␣FT(1,1)=F(1,1)
␣␣␣␣␣␣␣␣FT(1,2)=F(2,1)
␣␣␣␣␣␣␣␣FT(1,3)=F(3,1)
␣␣␣␣␣␣␣␣FT(2,1)=F(1,2)
␣␣␣␣␣␣␣␣FT(2,2)=F(2,2)
␣␣␣␣␣␣␣␣FT(2,3)=F(3,2)
␣␣␣␣␣␣␣␣FT(3,1)=F(1,3)
␣␣␣␣␣␣␣␣FT(3,2)=F(2,3)
␣␣␣␣␣␣␣␣FT(3,3)=F(3,3)
C
␣␣␣␣␣␣␣␣B(1,1)=F(1,1)*FT(1,1)+F(1,2)*FT(2,1)+F(1,3)*FT(3,1)
␣␣␣␣␣␣␣␣B(1,2)=F(1,1)*FT(1,2)+F(1,2)*FT(2,2)+F(1,3)*FT(3,2)
␣␣␣␣␣␣␣␣B(1,3)=F(1,1)*FT(1,3)+F(1,2)*FT(2,3)+F(1,3)*FT(3,3)
␣␣␣␣␣␣␣␣B(2,1)=B(1,2)
␣␣␣␣␣␣␣␣B(2,2)=F(2,1)*FT(1,2)+F(2,2)*FT(2,2)+F(2,3)*FT(3,2)
␣␣␣␣␣␣␣␣B(2,3)=F(2,1)*FT(1,3)+F(2,2)*FT(2,3)+F(2,3)*FT(3,3)
␣␣␣␣␣␣␣␣B(3,1)=B(1,3)
␣␣␣␣␣␣␣␣B(3,2)=B(2,3)
␣␣␣␣␣␣␣␣B(3,3)=F(3,1)*FT(1,3)+F(3,2)*FT(2,3)+F(3,3)*FT(3,3)
C
C␣Overwrite␣old␣RBULK
C␣␣␣␣␣␣␣␣RBULK=MARKMAT(MAT)%DENS(M)*MARKMAT(MAT)%CS(M)**PTWO
␣␣␣␣␣␣␣␣RSHEAR␣=␣UI(2)
␣␣␣␣␣␣␣␣RBULK␣=␣UI(1)␣+␣(PTWO/PTHREE)*UI(2)
␣␣␣␣␣␣␣␣RP=RBULK*LOG(RJ)
C␣VVN␣-␣ADDED␣VARIABLE␣RLOGJ
␣␣␣␣␣␣␣␣RLOGJ␣=␣LOG(RJ)
␣␣␣␣␣␣␣␣RC1=UI(1)
␣␣␣␣␣␣␣␣RC2=UI(2)
C
␣␣␣␣␣␣␣␣BB(1,1)=B(1,1)*B(1,1)+B(1,2)*B(2,1)+B(1,3)*B(3,1)
␣␣␣␣␣␣␣␣BB(1,2)=B(1,1)*B(1,2)+B(1,2)*B(2,2)+B(1,3)*B(3,2)
```

```
␣␣␣␣␣␣␣BB(1,3)=B(1,1)*B(1,3)+B(1,2)*B(2,3)+B(1,3)*B(3,3)
␣␣␣␣␣␣␣BB(2,1)=BB(1,2)
␣␣␣␣␣␣␣BB(2,2)=B(2,1)*B(1,2)+B(2,2)*B(2,2)+B(2,3)*B(3,2)
␣␣␣␣␣␣␣BB(2,3)=B(2,1)*B(1,3)+B(2,2)*B(2,3)+B(2,3)*B(3,3)
␣␣␣␣␣␣␣BB(3,1)=BB(1,3)
␣␣␣␣␣␣␣BB(3,2)=BB(2,3)
␣␣␣␣␣␣␣BB(3,3)=B(3,1)*B(1,3)+B(3,2)*B(2,3)+B(3,3)*B(3,3)
C
C--->Nearly␣incompressible␣invariants
␣␣␣␣␣␣␣RI1=(PONE/(RJ**(PTWO/PTHREE)))*(B(1,1)+B(2,2)+B(3,3))
␣␣␣␣␣␣␣RI2=(PONE/(RJ**(PFOUR/PTHREE)))*
␣␣␣␣␣&␣␣␣␣␣␣␣(B(1,1)*B(2,2)+B(2,2)*B(3,3)+B(3,3)*B(1,1)
␣␣␣␣␣&␣␣␣␣␣␣␣-B(1,2)*B(2,1)-B(2,3)*B(3,2)-B(1,3)*B(3,1))
C␣VVN
C␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,1)=PONE/RJ*(
C␣␣␣␣␣&␣␣␣RP+PTWO/(RJ**(PTWO/PTHREE))*(RC1+RI1*RC2)*B(1,1)-
C␣␣␣␣␣&␣␣␣PTWO/(RJ**(PFOUR/PTHREE))*RC2*BB(1,1)-PTWO/PTHREE*
C␣␣␣␣␣&␣␣␣(RC1*RI1+PTWO*RC2*RI2))
C␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,2)=PONE/RJ*(
C␣␣␣␣␣&␣␣␣RP+PTWO/(RJ**(PTWO/PTHREE))*(RC1+RI1*RC2)*B(2,2)-
C␣␣␣␣␣&␣␣␣PTWO/(RJ**(PFOUR/PTHREE))*RC2*BB(2,2)-PTWO/PTHREE*
C␣␣␣␣␣&␣␣␣(RC1*RI1+PTWO*RC2*RI2))
c␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,3)=PONE/RJ*(
C␣␣␣␣␣&␣␣␣RP+PTWO/(RJ**(PTWO/PTHREE))*(RC1+RI1*RC2)*B(3,3)-
C␣␣␣␣␣&␣␣␣PTWO/(RJ**(PFOUR/PTHREE))*RC2*BB(3,3)-PTWO/PTHREE*
C␣␣␣␣␣&␣␣␣(RC1*RI1+PTWO*RC2*RI2))
C␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,4)=PONE/RJ*(
C␣␣␣␣␣&␣␣␣PTWO/(RJ**(PTWO/PTHREE))*(RC1+RI1*RC2)*B(1,2)-
C␣␣␣␣␣&␣␣␣PTWO/(RJ**(PFOUR/PTHREE))*RC2*BB(1,2))
C␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,5)=PONE/RJ*(
C␣␣␣␣␣&␣␣␣PTWO/(RJ**(PTWO/PTHREE))*(RC1+RI1*RC2)*B(1,3)-
C␣␣␣␣␣&␣␣␣PTWO/(RJ**(PFOUR/PTHREE))*RC2*BB(1,3))
C␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,6)=PONE/RJ*(
C␣␣␣␣␣&␣␣␣PTWO/(RJ**(PTWO/PTHREE))*(RC1+RI1*RC2)*B(2,3)-
C␣␣␣␣␣&␣␣␣PTWO/(RJ**(PFOUR/PTHREE))*RC2*BB(2,3))

C␣VVN␣-␣NEO-HOOKEAN␣MODEL␣AS␣GIVEN␣IN␣PAPER
C␣VVN␣UI(1)␣=␣lambda,␣UI(2)␣=␣mu␣(lambda␣and␣mu
C␣VVN␣are␣lame␣parameters)
␣␣␣␣␣␣␣␣␣␣␣XLAMBDA␣=␣UI(1)
␣␣␣␣␣␣␣␣␣␣␣XMU␣=␣UI(2)
␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,1)␣=␣(XLAMBDA␣*␣RLOGJ/RJ)
␣␣␣␣␣1␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣+␣(XMU/RJ)*(B(1,1)-PONE)
␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,2)␣=␣(XLAMBDA␣*␣RLOGJ/RJ)
␣␣␣␣␣1␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣+␣(XMU/RJ)*(B(2,2)-PONE)
␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,3)␣=␣(XLAMBDA␣*␣RLOGJ/RJ)
␣␣␣␣␣1␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣+␣(XMU/RJ)*(B(3,3)-PONE)
```

```
␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,4)␣=␣(XMU/RJ)*(B(1,2))
␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,5)␣=␣(XMU/RJ)*(B(1,3))
␣␣␣␣␣␣␣MARKMAT(MAT)%STRESS(M,6)␣=␣(XMU/RJ)*(B(2,3))
C␣VVN␣-␣END␣NEO-HOOKEAN␣MODEL
C
␣␣␣␣␣␣␣SENRG(IGS)=RC1*(RI1-PTHREE)+RC2*(RI2-PTHREE)
C
␣␣␣␣␣␣ENDDO
C
␣␣␣␣␣␣IF(INT(UI(6)).EQ.1)THEN
␣␣␣␣␣␣␣CALL␣KINETIC_FRACTURE_ISO_MAIN(MSF,NGS,MAT,UI,6,2,DT,MGSMAP,DAM)
␣␣␣␣␣␣ELSEIF(INT(UI(6)).EQ.2)THEN
␣␣␣␣␣␣␣DO␣IGS=1,NGS
␣␣␣␣␣␣␣␣M=MGSMAP(IGS)
␣␣␣␣␣␣␣␣RMAG=SQRT(MARKMAT(MAT)%STRAIN(M,1)**2+
␣␣␣␣␣&␣␣␣MARKMAT(MAT)%STRAIN(M,2)**2+MARKMAT(MAT)%STRAIN(M,3)**2+
␣␣␣␣␣&␣␣␣PTWO*(MARKMAT(MAT)%STRAIN(M,4)**2+
␣␣␣␣␣&␣␣␣MARKMAT(MAT)%STRAIN(M,5)**2+
␣␣␣␣␣&␣␣␣MARKMAT(MAT)%STRAIN(M,6)**2))
␣␣␣␣␣␣␣␣DAM(IGS)=MIN(RMAG/UI(3),PONE)
␣␣␣␣␣␣␣ENDDO
␣␣␣␣␣␣ENDIF
C
␣␣␣␣␣␣RETURN
␣␣␣␣␣␣END
```

DISTRIBUTION:

1 MS 0899 Technical Library, 9536 (electronic copy)


Sandia National Laboratories