Constitutive Modelling of Soils under High Strain Rates

Theoretical, Numerical, and Experimental Results

Prepared By:
Kaiwen Xia (Ph.D.)
Mohammadamin Jafari
Patrick Paskalis Kanopoulos
Yao Wei
Department of Civil Engineering, University of Toronto, Toronto, Ontario

PWGSC Contract Number: W7701-135578/001/QCL
CSA: Grant McIntosh, Defence Scientist, 418-844-4000 ext. 4278

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of the Department of National Defence of Canada.

Contract Report
DRDC-RDDC-2015-C071
March 2015
CONSTITUTIVE MODELLING OF SOILS UNDER HIGH STRAIN RATES:
THEORETICAL, NUMERICAL, AND EXPERIMENTAL RESULTS

by

Kaiwen Xia (Ph.D.)
Mohammadamin Jafari
Patrick Paskalis Kanopoulos
Yao Wei

Department of Civil Engineering, University of Toronto
Abstract

The following report briefly outlines the concept of Perzyna-type viscoplasticity and its underlying constitutive equations that describe the nonlinear stress-strain relations of rate-depndant materials in the generalized tensor framework. Following the theoretical development, a numerical algorithm that computes the stress increment based on known a strain increment is presented. This algorithm is suitable for implementation into finite element codes (including hydrocodes such as LS-DYNA). Experimental stress-strain results from fine and coarse grained soils ranging from strain rates of 100 s\(^{-1}\) to 2100 s\(^{-1}\) are presented. The constitutive equations are calibrated against the experimental data using the Marquardt-Levenberg nonlinear optimisation algorithm. The material constants provided as a result of the calibration are suitable for implementation in hydrocodes, provided similar material types and conditions are being modelled over strain rates within the same orders of magnitude of the testing.

A three phase equation of state is presented both in the theoretical form and in a numerically appropriate pseudocode. The numerical code returns changes in pressure based on prescribed changes in volume, or computes an updated material bulk modulus. Parameters for the three phase model are derived from the literature for various sand types and degrees of saturation.

Finally, the constitutive model (both in the volumetric/deviator and deviator-only versions), and the three phase equation of state are implemented in FORTRAN77 source code and compiled as part of an LS-DYNA executable. Simulations are conducted in LS-DYNA and compared to readily available explosion experiments from the open literature. Simulated and experimental results show good agreement with each other.
Contents

Chapter 2: Numerical Implementation ............................................................................................. 7

Chapter 3: Experimental Procedures ............................................................................................... 11
  3.1 Split Hopkinson Pressure Bar Method ...................................................................................... 11
  3.2 Physical properties of the sand used in the experiments ......................................................... 12
  3.3 Dynamic experimental results on dry sand .............................................................................. 14
  3.4 Dynamic experimental results on saturated sand ..................................................................... 15
  3.5 Uniaxial strain tests in the quasi-static state ........................................................................... 16

Chapter 4: Calibration of the constitutive model ............................................................................... 17
  4.1 Calibration of fine grained dry sand ....................................................................................... 17
  4.2 Calibration of coarse grained dry sand ................................................................................... 17
  4.3 Calibration of coarse grained saturated sand ......................................................................... 20
  4.4 Calibration of fine grained saturated sand ............................................................................. 20

Chapter 5: Equation of State ............................................................................................................. 23
  5.1 Three-Phase Equation of State ............................................................................................... 23
  5.2 Numerical Implementation of the Equation of State ............................................................... 26
  5.3 Suitable parameters for the equation of state ....................................................................... 29

Chapter 6: Numerical simulation ..................................................................................................... 30
  6.1 Model Geometry and Parameters ......................................................................................... 30
  6.2 Model results ........................................................................................................................ 35

Conclusion ......................................................................................................................................... 39

References ......................................................................................................................................... 40

Appendix A: Subroutine implementation in LS-DYNA ................................................................. 41
Appendix B: User defined constitutive model ............................................................................... 43
  B.1 Deviatoric/Volumetric constitutive model source code and associated subroutines .......... 43
  B.2 Deviatoric-only constitutive model source code .................................................................. 60

Appendix C: User defined equation of state model source code ............................................... 68

Appendix D: SHPB Experimental Results ..................................................................................... 73

Appendix E: EOS Parameter Estimation ....................................................................................... 89
Chapter 1: Theoretical Framework

In this chapter, the underlying principles of Perzyna viscoplasticity are presented. Viscoplastic models are useful for describing the rate-dependent inelastic mechanical behaviour of materials, which is a characteristic often observed for many types of geomaterials. Specifically, increases in strain rate are often accompanied by increases in material stiffness. This is an important feature of materials that should be captured if an accurate representation of stress and blast waves is to be achieved in numerical models. Perzyna viscoplasticity is a relatively simple extension of classical plasticity wherein the stresses are allowed to temporarily occupy a state outside of the yield and cap surface; the so-called “overstress” state.

Using the standard vector notation for stresses and strains, the strain rate vector in Perzyna viscoplasticity can be decomposed into the elastic and viscoplastic parts:

\[ \dot{\varepsilon} = \dot{\varepsilon}^e + \dot{\varepsilon}^{vp} \]

where the elastic strains are given by the elastic constitutive relation

\[ \dot{\sigma} = C \dot{\varepsilon}^e . \]

The (associative) viscoplastic flow rule is given by the gradient to the yield surface \((f = 0)\)

\[ \dot{\varepsilon}^{vp} = \eta(\varphi(f)) \frac{\partial f}{\partial \sigma} \]

where \(\eta\) is a scalar parameter with units of inverse seconds, the triangular brackets indicate a ramp function \((\langle x \rangle = x + |x|/2)\), and \(\varphi(f)\) is a dimensionless viscous flow function given by

\[ \varphi(f) = \left( \frac{f}{f_0} \right)^N \]

where \(N\) is a material parameter and \(f_0\) is a normalizing constant with the same units as \(f\).

The yield surface is composed of three functions; the tension cut-off surface, the failure surface, and the cap surface. All can be conveniently represented in the \(I_1 - J_2\) stress invariant space. Recall

\[ I_1 = \sigma_{11} + \sigma_{22} + \sigma_{33} \]

and,

\[ J_2 = \frac{1}{2} (s_{11}^2 + s_{22}^2 + s_{33}^2) + s_{12}^2 + s_{23}^2 + s_{13}^2 \]
where the vector $\mathbf{s}$ is the stress deviator. Referring to Figure 1.1, we have the following three regions:

1. The cap surface region applies when $I_1 \geq L(k)$, and is defined by the ellipse

   $$ f = \sqrt{\frac{(I_1 - L(k))^2}{R^2}} - J_2 - \frac{L(k) - X(k)}{R} = 0 $$

   where $R$ is a material parameter (the ratio of the major to minor axis) and $k$ is the hardening parameter.

2. The failure surface region applies when $L(k) > I_1 > -T$, and is defined by

   $$ f = \sqrt{J_2 - F_e(I_1)} = \sqrt{J_2} - [\alpha - \gamma \exp(-\beta I_1) + \theta I_1] = 0 $$

   where $\alpha, \beta, \gamma$, and $\theta$ are material constants.

3. The tension cutoff region applies when $I_1 \leq -T$, and is defined by $f = I_1 - (-T) = 0$ where $T$ is the tensile cut-off stress.

The parameter $X(k)$ represents the intersection of the cap with the $I_1$ axis, and is given by:

$$ X(k) = k + R \cdot F_e(k). $$

The parameter $L(k)$ represents $I_1$ ordinate of the intersection between the cap and failure surfaces, and is given by the relation:

$$ L(k) = \begin{cases} k, & \text{if } k > 0 \\ 0, & \text{if } k \leq 0 \end{cases}. $$

Finally, the volumetric hardening function is given by the following relation:

$$ \varepsilon_v^{vp} = W(1 - \exp(-D(X(k) - X_0))) $$

where $\varepsilon_v^{vp} = \varepsilon_{11}^{vp} + \varepsilon_{22}^{vp} + \varepsilon_{33}^{vp}$, and the parameters $W$ and $D$ are the volumetric hardening parameters.
Figure 1.1: Schematic of the tension cut-off, failure surface and cap surfaces in the $I_1 - J_2$ stress invariant space. The viscoplastic solution, in contrast to the plastic solution, can occupy a region outside of the yield surface.
Chapter 2: Numerical Implementation

Displacement based, explicitly integrated finite element codes require the computation of stresses as a function of assumed displacements (strains) at Gaussian points. It is therefore necessary to design an algorithm which can accurately return incremental stresses for given incremental states of strains over a known time step. The most common method in computational inelasticity of accomplishing this is the return-mapping algorithm, which is effectively a Newton-Rhapson approximation from an elastic predictor stress onto the yield surface (or, in the case of viscoplasticity, onto the appropriate overstressed state).

The incremental stress and strain vectors are given by the following equations:

\[
\Delta \varepsilon = \Delta \varepsilon^e + \Delta \varepsilon^{vp}
\]

\[
\Delta \sigma = C \Delta \varepsilon^e = C(\Delta \varepsilon - \Delta \varepsilon^{vp})
\]

An Euler approximation of the total strain increment at time \( t + \Delta t \) yields

\[
\Delta \varepsilon = [(1 - \chi)\dot{\varepsilon}_t^{vp} + \chi \dot{\varepsilon}_{t+\Delta t}^{vp}]\Delta t, \quad 0 \leq \chi \leq 1
\]

where \( \chi \) is the integration parameter. In the forgoing, a fully implicit approximation is used (\( \chi = 1 \)), and in this case, the solution is unconditionally stable and the viscoplastic flow is determined by the gradient of the flow function only at time \( t + \Delta t \). Therefore, we have

\[
\Delta \varepsilon^{vp} = \Delta \dot{\varepsilon}^{vp} \Delta t = \eta(\phi(f))\Delta t \frac{\partial f}{\partial \sigma} = \Delta \lambda \frac{\partial f}{\partial \sigma}
\]

where \( \Delta \lambda \) is the plastic multiplier,

\[
\Delta \lambda = \eta(\phi(f))\Delta t.
\]

The problem is solved if the residual, \( \rho \), approaches zero during a local iteration:

\[
\rho = \frac{\Delta \lambda}{\eta \Delta t} - \phi(f) \to 0.
\]

The stress increment can be rewritten in terms of the plastic multiplier \( \Delta \lambda \),

\[
\Delta \sigma = C \left( \Delta \varepsilon - \Delta \lambda \frac{\partial f}{\partial \sigma} \right).
\]
In order to compute $\Delta \lambda$, a local Newton-Rhapson iteration is applied by taking the differential of the stress increment above for iteration $i$:

$$
\delta \sigma = C \left( \delta \varepsilon - \delta \lambda \frac{\partial f}{\partial \sigma} + \Delta \lambda^i \frac{\partial^2 f}{\partial \sigma^2} \delta \sigma - \Delta \lambda^i \frac{\partial^2 f}{\partial \sigma \partial \lambda} \delta \lambda \right).
$$

Using a pseudoelastic stiffness matrix, $H$, we can represent $\delta \sigma$ by

$$
\delta \sigma = H \left[ \delta \varepsilon - \left( \frac{\partial f}{\partial \sigma} - \Delta \lambda^i \frac{\partial^2 f}{\partial \sigma \partial \lambda} \right) \delta \lambda \right]
$$

with,

$$
H = \left[ C^{-1} + \Delta \lambda^i \frac{\partial^2 f}{\partial \sigma^2} \right]^{-1}.
$$

By differentiating the residual $\rho$, the Newton-Rhapson iteration $i$ is expressed as

$$
\rho^i = \left( \frac{1}{\eta \Delta t} - \frac{\partial \phi}{\partial \lambda} \right) \delta \lambda - \left( \frac{\partial \phi}{\partial \sigma} \right)^T \delta \sigma
$$

substituting $\delta \sigma$ into the equation for $\rho^i$ yields

$$
\delta \lambda = \frac{1}{\xi} \left[ \left( \frac{\partial \phi}{\partial \sigma} \right)^T H \delta \varepsilon + \rho^i \right]
$$

with

$$
\xi = \left( \frac{\partial \phi}{\partial \sigma} \right)^T H \left[ \frac{\partial f}{\partial \sigma} + \Delta \lambda^i \frac{\partial^2 f}{\partial \sigma \partial \lambda} \right] + \frac{1}{\eta \Delta t} \frac{\partial \phi}{\partial \lambda}.
$$

To obtain an accurate estimate of the stress increment $\Delta \sigma$, several local iterations for $\delta \sigma$ should be applied until suitable convergence is obtained. A detailed process in pseudocode is provided in the following list, which based, with some modification, on the method proposed by Tong and Tuan (2007).
1. Compute the trial stresses based on the elastic stiffness: \( \sigma_{t+\Delta t}^{trial} = \sigma_t + C\Delta \varepsilon \)

2. Check if stresses are outside the yield surface. If \( f > 0 \) and \( I_{1,t+\Delta t}^{trial} \geq -T \) go to step 3. If \( f > 0 \) and \( I_{1,t+\Delta t}^{trial} < -T \) go to step 5. If \( f \leq 0 \), return;

3. Assign the initial values before iteration:
   \[
   \Delta \lambda^{(0)} = 0 \\
   \sigma_{t+\Delta t}^{(0)} = \sigma_n \\
   \rho^{(0)} = \varphi(\sigma_{t+\Delta t}^{(0)} k_n) - \frac{\Delta \lambda^{(0)}}{\eta \Delta t}
   \]

4. Loop for local iteration \( i \),
   i. Calculate pseudoelastic matrix \( H \), and variable \( \xi \)
   ii. Update \( \Delta \lambda^{(i)} \), \( \sigma_{t+\Delta t}^{(i)} \), \( k^{(i)} \) according to
      \[
      \Delta \lambda^{(i+1)} = \Delta \lambda^{(i)} + \frac{\rho^{(i)}}{\xi} \\
      \sigma_{t+\Delta t}^{(i+1)} = \sigma_t + C \left[ \Delta \varepsilon - \Delta \lambda^{(i+1)} \frac{\partial f}{\partial \sigma} \right] \\
      \delta \varepsilon^{vp} = \Delta \lambda^{(i+1)} \left( \frac{\partial f}{\partial \sigma_{11}} + \frac{\partial f}{\partial \sigma_{22}} + \frac{\partial f}{\partial \sigma_{33}} \right) \\
      \varepsilon_{t+\Delta t}^{vp} = \varepsilon_t^{vp} + \delta \varepsilon^{vp} \\
      X^{i+1} = X_0 - \ln \left[ 1 - \frac{\varepsilon_{t+\Delta t}^{vp}}{W} \right] / D \\
      *or, return elastic loading stresses if \( \varepsilon_{t+\Delta t}^{vp} > 0.99W \) to ensure non-singularity of \( X^{i+1} \)
      \]
      \[
      k_{t+\Delta t}^{(i+1)} \quad \text{from implicit solution of} \quad X^{i+1}(k) = k + R \cdot F_\varepsilon(k)
      \]
iii. Check for convergence, and if $|\rho_{i+1}| \leq \text{tolerance}$, return;

$$\rho_{i+1} = \varphi \left( \sigma_{\text{t+}\Delta t}^{(i+1)}, k_{\text{t+}\Delta t}^{(i+1)} \right) - \frac{\Delta \lambda_{(i+1)}}{\eta \Delta t}$$

5. Tensile regime, return corrected tensile stresses as follows:

i. If $t_{1,t+\Delta t}^{\text{trial}} \leq -T$ and $\sqrt{f_{2,t+\Delta t}^{\text{trial}}} < F_e(-T)$, then

$$I_{1,t+\Delta t} = e^{-\eta \Delta t} t_{1,t+\Delta t}^{\text{trial}} + (1 - e^{-\eta \Delta t})(-T)$$

$$\sqrt{J_{2,t+\Delta t}} = \sqrt{f_{2,t+\Delta t}^{\text{trial}}}$$

ii. If $t_{1,t+\Delta t}^{\text{trial}} \leq -T$ and $\sqrt{f_{2,t+\Delta t}^{\text{trial}}} \geq F_e(-T)$, then

$$I_{1,t+\Delta t} = e^{-\eta \Delta t} t_{1,t+\Delta t}^{\text{trial}} + (1 - e^{-\eta \Delta t})(-T)$$

$$\sqrt{J_{2,t+\Delta t}} = e^{-\eta \Delta t} \sqrt{f_{2,t+\Delta t}^{\text{trial}}} + (1 - e^{-\eta \Delta t})F_e(-T)$$

iii. Return;

6. Return stresses and hardening parameter, end subroutine

It should be noted that in the user defined subroutines that have been implemented, an optional non-linear (exponential) elastic loading/unloading constitutive behaviour is provided. In this case, the tangential stiffness matrix is computed by means of the equations for bulk and shear moduli as:

$$K_{\text{tan}} = K_1 \exp(K_2 I_1)$$

$$G_{\text{tan}} = G_1 \exp(G_2 \sqrt{J_2})$$

where $K_1$ and $G_1$ have dimension of stress, $K_2$ has dimension of inverse stress, and $G_2$ has dimension of root inverse stress. Furthermore, $K_{\text{tan}}$ and $G_{\text{tan}}$ may be different in loading and unloading, where loading is defined as an increase in volumetric strain.
Chapter 3: Experimental Procedures

In this chapter, the experimental procedures are described in brief and the experimental results of the tests are summarised.

3.1 Split Hopkinson Pressure Bar Method

The method used to determine the physical behaviour of sand at high strain rates (from $10^2$ to $10^4$ s$^{-1}$) was the split Hopkinson pressure bar (SHPB) method. In this method, the material sample is placed between an incident bar and a transmitted bar. A striker is launched towards the incident bar which initiates a stress wave that travels through the sample and into the transmitted bar. The impedance mismatch between the sample and steel bars results in a reflected and a transmitted wave. Comparisons of the incident, transmitted, and reflected waves, together with the simple theory of one dimensional wave propagation, allows the axial stress-strain history to be determined. Different maximum strains, and strain rates are achieved through variation of the striker velocity, and the type of pulse shaper used between the striker and the incident bar. Another important characteristic which should be observed during the course of the test is force balancing. Force balance is said to be achieved when the load history on each bar-material interface is equal at a given point in time. Inertial effects (the multiple internal propagation of stress waves in the sample) can be neglected if force balance is occurring, in other words, the material is deforming uniformly throughout during the test.

The typical SHPB setup was modified for the sand tests in order to achieve a uniaxial state of stress in the material. The sample strain histories can therefore be described as follows:

$$\varepsilon_{11}(t) = \varepsilon_{axial}(t), \quad \varepsilon_{22}(t) = \varepsilon_{33}(t) = 0.$$  

The uniaxial state of strain was achieved by means of a thick steel sample holder which completely constrained the expansion of the material during the course of the tests.

For all dynamic tests conducted in the SHPB system, the sample was 5 mm in length and 25.4 mm in diameter (which is the same diameter as the bars). All bars and strikers were maraging steel with a density of 8.1 g/cm$^3$, a Young’s modulus of 200 GPa, and a strength of 2.5 GPa. The striker bar was 30 cm long.
3.2 Physical properties of the sand used in the experiments

The sand used in the experiments was Ottawa sand (OS, shown in Figure 3.1). Both a fine grained (Figure 3.1 left) and coarse grained (Figure 3.1 right) sand were investigated over the course of the research program. The physical properties of each of these sands were measured and are reported in Table 3.1, and the particle size distributions for each sand is plotted in Figure 3.1 and summarized in Table 3.2.

![Figure 3.1](image)

**Figure 3.1** Ottawa Sand: fine grained sand OS1 (left) and coarse grain sand OS2 (right).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Dimension</th>
<th>OS1 (fine)</th>
<th>OS2 (coarse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particle Density</td>
<td>g/cm³</td>
<td>2.635</td>
<td>2.653</td>
</tr>
<tr>
<td>Bulk Density</td>
<td>g/cm³</td>
<td>1.596</td>
<td>1.664</td>
</tr>
<tr>
<td>Void Ratio</td>
<td></td>
<td>0.651</td>
<td>2.653</td>
</tr>
</tbody>
</table>

**Table 3.1**: Physical properties of the sand used in the experiments
### Table 3.2: Sieve analysis results for OS1 and OS2.

<table>
<thead>
<tr>
<th>Size (mm)</th>
<th>OS1 (fine)</th>
<th>OS2 (coarse)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mass (g)</td>
<td>Percent (%)</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.18</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.6</td>
<td>1.12</td>
<td>0.184</td>
</tr>
<tr>
<td>0.3</td>
<td>470.31</td>
<td>77.2545</td>
</tr>
<tr>
<td>0.18</td>
<td>129.13</td>
<td>21.2113</td>
</tr>
<tr>
<td>0.15</td>
<td>6.09</td>
<td>1.0004</td>
</tr>
<tr>
<td>0.106</td>
<td>1.62</td>
<td>0.266</td>
</tr>
<tr>
<td>0.075</td>
<td>0.31</td>
<td>0.0509</td>
</tr>
<tr>
<td>0.053</td>
<td>0.13</td>
<td>0.0214</td>
</tr>
<tr>
<td>&lt;0.053</td>
<td>0.07</td>
<td>0.0115</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>608.78</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

Figure 3.2: Sieve analysis results for OS1 and OS2.
3.3 Dynamic experimental results on dry sand

The dynamic stress-strain results for tests on fine and coarse sand in the dry condition are provided in this section. Average results for a given strain rate are summarized in the following figures. Each stress strain history is computed as an average of at least 11 tests.

Figure 3.3: SHPB uniaxial strain results on dry fine grained sand for various strain rates.

Figure 3.4: SHPB uniaxial strain results on dry coarse grained sand for various strain rates.
3.4 Dynamic experimental results on saturated sand

The dynamic stress-strain results for tests on fine and coarse sand in the saturated condition are provided in this section. Average results for a given strain rate are summarized in the following figures. Each stress strain history is computed as an average of at least 11 tests.

![Figure 3.5: SHPB uniaxial strain results on fully saturated fine grained sand for various strain rates.](image1)

![Figure 3.6: SHPB uniaxial strain results on fully saturated coarse grained sand for various strain rates.](image2)
3.5 Uniaxial strain tests in the quasi-static state

In order to gain some additional insight into the behaviour of the sand at high strains, a uniaxial strain test was conducted in an MTS machine with the same steel ring used to constrain the sample in the dynamic SHPB tests. These tests were conducted only in the dry condition due to the infeasibility of restricting the drainage of water during the course of slow strain rate test. The results of the tests are provided in the following figures for both the coarse and fine grained sand.

![Figure 3.7: Quasi-static uniaxial strain results for coarse and dry sand](image-url)
Chapter 4: Calibration of the constitutive model

Constitutive model parametrisation follows from the experimental data by recasting the constitutive problem where stresses are determined as a function of strain increments, to one of nonlinear optimisation, where both the stress and strain trajectories are specified (i.e. the experiment) under unknown model parameters (i.e. the elastic parameters, the surface and cap parameters, and the viscosity parameters). For example, given the objective function, $F$:

$$
\min(F) = \min\left\{ \sigma_{\text{exp}}(\varepsilon_{\text{exp}}) - \sigma_{\text{model}}(\varepsilon_{\text{exp}}; P) \right\}
$$

where $P$ is a vector containing the undetermined parameters we wish to solve for by the minimisation of the function $F$; $\varepsilon_{\text{exp}}$ are the set of known strains; $\sigma_{\text{exp}}$ are the stresses determined from the experiment; and $\sigma_{\text{model}}$ are the stresses computed from the constitutive model. The constitutive model was recast in this form (in MATLAB) and the Marquardt-Levenberg optimisation algorithm was used to solve for the vector of unknown parameters $P$. This method is similar to that used by Simo et al. (1988), to fit cap parameters to their model.

It should be noted that this problem, even with the availability of extensive experimental data, is non-unique, and therefore several combinations of parameters may possibly result in a suitable parametrised constitutive model. For this reason, several parameters were fixed (at values typical for sand), and other values were determined through the optimisation procedure.

4.1 Calibration of fine grained dry sand

The results of the calibration for the fine grained dry sand are provided in Table 4.1. In this model, a nonlinear exponential unloading elastic stiffness tensor is used (with coefficient $K_1$ and $G_1$, and exponent $K_2$ and $G_2$). The loading elastic stiffness tensor is linear. Model output for several experimental tests is provided in Figure 4.1.

4.2 Calibration of coarse grained dry sand

The results of the calibration for the coarse grained dry sand are provided in Table 4.2. In this model, a nonlinear exponential unloading elastic stiffness tensor is used (with coefficient $K_1$ and $G_1$, and exponents $K_2$ and $G_2$). The loading elastic stiffness tensor is linear. Model output for several experimental tests is provided in Figure 4.2.
Table 4.1: Results of the parametrisation on dry fine sand for the available SHPB data.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter</th>
<th>Value</th>
<th>Dimension</th>
<th>Fixed/Optimised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic</td>
<td>K1 – load</td>
<td>900</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>K2 – load</td>
<td>0</td>
<td>1/MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>G1 – load</td>
<td>415</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>G2 – load</td>
<td>0</td>
<td>1/MPa&lt;sup&gt;1/2&lt;/sup&gt;</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>K1 – unload</td>
<td>264</td>
<td>MPa</td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>K2 – unload</td>
<td>59.4</td>
<td>1/MPa</td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>G1 – unload</td>
<td>122</td>
<td>MPa</td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>G2 – unload</td>
<td>153</td>
<td>1/MPa&lt;sup&gt;1/2&lt;/sup&gt;</td>
<td>Optimised</td>
</tr>
<tr>
<td>Failure</td>
<td>α</td>
<td>3.654</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>β</td>
<td>0.003</td>
<td>1/MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>γ</td>
<td>3.5</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>θ</td>
<td>0.263</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>Hardening</td>
<td>W</td>
<td>0.1</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.01</td>
<td>1/MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td>Cap</td>
<td>R</td>
<td>3</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>X&lt;sub&gt;0&lt;/sub&gt;</td>
<td>(α-γ)*R</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>1</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td>Viscoplastic</td>
<td>f&lt;sub&gt;0&lt;/sub&gt;</td>
<td>1</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>1.055</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>η</td>
<td>217.5</td>
<td>1/s</td>
<td>Optimised</td>
</tr>
</tbody>
</table>

Figure 4.1: Comparison of experimental and constitutive model results for dry fine sand at different strain rates and maximum strains.
<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter</th>
<th>Value</th>
<th>Dimension</th>
<th>Fixed/Optimised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic</td>
<td>K1 – load</td>
<td>900 MPa</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>K2 – load</td>
<td>0 1/MPa</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>G1 – load</td>
<td>415 MPa</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>G2 – load</td>
<td>0 1/MPa$^{1/2}$</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>K1 – unload</td>
<td>264 MPa</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>K2 – unload</td>
<td>59.4 1/MPa</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>G1 – unload</td>
<td>122 MPa</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>G2 – unload</td>
<td>153 1/MPa$^{1/2}$</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td>Failure</td>
<td>α</td>
<td>3.654 MPa</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>β</td>
<td>0.003 1/MPa</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>γ</td>
<td>3.5 MPa</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>θ</td>
<td>0.263</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>Hardening</td>
<td>W</td>
<td>0.11</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.015 1/MPa</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>Cap</td>
<td>R</td>
<td>3</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>X_0</td>
<td>$(\alpha-\gamma)*R$  MPa</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>1 MPa</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>Viscoplastic</td>
<td>f_0</td>
<td>1 MPa</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>0.8890</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>η</td>
<td>190.0 1/s</td>
<td></td>
<td>Optimised</td>
</tr>
</tbody>
</table>

**Table 4.2:** Results of the parametrisation on dry coarse sand for the available SHPB data

**Figure 4.2:** Comparison of experimental and constitutive model results for dry coarse sand at different strain rates and maximum strains.
4.3 Calibration of coarse grained saturated sand

The results of the calibration for the coarse grained saturated sand are provided in Table 4.3. In this model, a nonlinear exponential unloading elastic stiffness tensor is used (with coefficient K1 and G1, and exponents K2 and G2). The loading elastic stiffness tensor is linear. Model output for several experimental tests is provided in Figure 4.3. It should be noted that only a portion of the tests are used to calibrate the model, since as peak strain is attained for each test, the loading rate drops and so too does the observed stiffness of the sample (c.f. Figures 3.5-3.6). Therefore, only where a calibration for the loading portion is performed only over the reliable loading region, and a separate unloading calibration is performed for the unloading elastic constants.

4.1 Calibration of fine grained saturated sand

The results of the calibration for the coarse grained saturated sand are provided in Table 4.4. In this model, a nonlinear exponential unloading elastic stiffness tensor is used (with coefficient K1 and G1, and exponents K2 and G2). The loading elastic stiffness tensor is linear. Model output for several experimental tests is provided in Figure 4.4. It should be noted that only a portion of the tests are used to calibrate the model, since as peak strain is attained for each test, the loading rate drops and so too does the observed stiffness of the sample (c.f. Figures 3.5-3.6). Therefore, only where a calibration for the loading portion is performed only over the reliable loading region, and a separate unloading calibration is performed for the unloading elastic constants.
<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter</th>
<th>Value</th>
<th>Dimension</th>
<th>Fixed/Optimised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic</td>
<td>K1 – load</td>
<td>2,900</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>K2 – load</td>
<td>0</td>
<td>1/MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>G1 – load</td>
<td>1,300</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>G2 – load</td>
<td>0</td>
<td>1/MPa(^{1/2})</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>K1 – unload</td>
<td>958</td>
<td>MPa</td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>K2 – unload</td>
<td>75.1</td>
<td>1/MPa</td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>G1 – unload</td>
<td>442</td>
<td>MPa</td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>G2 – unload</td>
<td>195</td>
<td>1/MPa(^{1/2})</td>
<td>Optimised</td>
</tr>
<tr>
<td>Failure</td>
<td>(\alpha)</td>
<td>3.654</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>(\beta)</td>
<td>0.003</td>
<td>1/MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>(\gamma)</td>
<td>3.5</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>(\theta)</td>
<td>0.263</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>Hardening</td>
<td>W</td>
<td>0.06</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.02</td>
<td>1/MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td>Cap</td>
<td>R</td>
<td>3</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>(X_0)</td>
<td>((\alpha\cdot\gamma)^*R)</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>1</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td>Viscoplastic</td>
<td>(f_0)</td>
<td>1</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>0.377</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>(\eta)</td>
<td>61.94</td>
<td>1/s</td>
<td>Optimised</td>
</tr>
</tbody>
</table>

Table 4.3: Results of the parametrisation on saturated coarse sand for the available SHPB data.

**Figure 4.3:** Comparison of experimental and constitutive model results for saturated coarse sand at different strain rates. Loading up to strain rate drop-off used to calibrate results.
<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter</th>
<th>Value</th>
<th>Dimension</th>
<th>Fixed/Optimised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic</td>
<td>K1 – load</td>
<td>2,900</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>K2 – load</td>
<td>0</td>
<td>1/MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>G1 – load</td>
<td>1,300</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>G2 – load</td>
<td>0</td>
<td>1/MPa(^{1/2})</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>K1 – unload</td>
<td>953</td>
<td>MPa</td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>K2 – unload</td>
<td>142</td>
<td>1/MPa</td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>G1 – unload</td>
<td>440</td>
<td>MPa</td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>G2 – unload</td>
<td>369</td>
<td>1/MPa(^{1/2})</td>
<td>Optimised</td>
</tr>
<tr>
<td>Failure</td>
<td>α</td>
<td>3.654</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>β</td>
<td>0.003</td>
<td>1/MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>γ</td>
<td>3.5</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>θ</td>
<td>0.263</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>Hardening</td>
<td>W</td>
<td>0.06</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.02</td>
<td>1/MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td>Cap</td>
<td>R</td>
<td>3</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>X_0</td>
<td>(α-γ)*R</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>1</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td>Viscoplastic</td>
<td>f_0</td>
<td>1</td>
<td>MPa</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>0.4029</td>
<td></td>
<td>Optimised</td>
</tr>
<tr>
<td></td>
<td>η</td>
<td>66.77</td>
<td>1/s</td>
<td>Optimised</td>
</tr>
</tbody>
</table>

Table 4.4: Results of the parametrisation on saturated fine sand for the available SHPB data.

Figure 4.4: Comparison of experimental and constitutive model results for saturated fine sand at different strain rates. Loading up to strain rate drop-off used to calibrate results.
Chapter 5: Equation of State

Accurate modelling of blast waves through solids requires the use of an equation of state (EOS) to predict the volumetric (hydrostatic) response of the material, especially when volumetric strains are extremely high. In some cases, the shear stresses of the material being modelled are neglected, since these are minimal in relation to the volumetrically induced pressures. Otherwise, a constitutive relation can be defined that models only the deviatoric portion of the stress-strain relation, whereas the hydrostatic portion is modelled through a suitable equation of state.

This chapter briefly outlines the theoretical framework of the three-phase EOS proposed by Wang et al. (2004) and developed by An et al. (2011) for a two or three phase material (such as unsaturated, partially saturated, or fully saturated sand). A numerical procedure for computing the material bulk modulus or pressure based on a change in the material volume (as is the case in an LS-DYNA hydrocode simulation) is presented. Finally, suitable parameters based on experiments conducted by Chapman et al. (2006) are presented. The FORTRAN source code for direct implementation into the LS-DYNA user defined subroutine files (dyn21b.F) is documented in Appendix C.

5.1 Three-Phase Equation of State

The three phase equation of state is an extension of the Mie-Gruneisen equation of state for modelling the pressure-volume-energy relations for media consisting of more than one phase. The known initial volume and mass ratios of each phase (silica sand, water, and air) are used to compute the relative pressure contribution to the total material pressure and specific energy for a given state.

The conservation of mass, momentum, and energy in a soil in a shocked state (subscript H), in comparison to the reference state (subscript o), can be expressed as follows:

\[ \rho_o U_s = \rho (U_s - u_p) \]

\[ P_H = \rho_o U_s u_p \]

\[ E_H - E_o = \frac{P_H}{2} (V_o - V_H) \]

where \( U_s \) denotes the shock velocity and \( u_p \) denotes the particle velocity, \( V \) is the volume of the material, and \( \rho \) is the density. The Hugoniot of a material can be expressed as the relation between the shock velocity and the particle velocity as follows:

\[ U_s = C_o - s u_p \]
where $C_o$ is the sound speed at the reference pressure and temperature, and $s$ is the linear coefficient. From the conservation of mass and the Hugoniot relation above, we have:

$$U_s = \frac{C_o}{1 - s\Delta}$$

where $\Delta$ is the volumetric strain. Similarly,

$$u_p = U_s\Delta = \frac{C_o\Delta}{1 - s\Delta}.$$

Letting

$$\mu = \frac{\rho}{\rho_o} - 1 = \frac{V}{V_o} - 1$$

We have

$$\Delta = \frac{\mu}{1 + \mu}$$

Substituting $u_p$ and $U_s$ into the conservation of momentum equation yields,

$$P_H = \rho_o \left( \frac{C_o}{1 - s\Delta} \right) \left( \frac{C_o\Delta}{1 - s\Delta} \right) = \frac{\rho_oC_o^2\Delta}{(1 - s\Delta)^2}$$

and finally, substituting the expression for $\mu$ yields,

$$P_H = \frac{\rho_oC_o^2 \left( \frac{\mu}{1 + \mu} \right)}{(1 - s\frac{\mu}{1 + \mu})^2}.$$

A more accurate representation than the uniaxial strain case can be obtained by expressing the pressure in terms of the Gruneisen parameter $\gamma(V)$:

$$P - P_H = \rho\gamma(V)E_H = \frac{\gamma(V)}{V}(E - E_H)$$
where $E_H$ is the internal energy per unit mass for the Hugoniot (reference state), and $E$ is the energy per unit mass. From the conservation of energy, the above expression yields the Mie-Gruneisen equation:

$$P = P_H \left( 1 - \frac{Y}{Z} \mu \right) + \frac{Y}{V} \left( E - E_o \right)$$

If the Gruneisen parameter is expressed as follows

$$\gamma = \alpha + (\gamma_o - \alpha) \frac{V}{V_o}$$

where $\alpha$ is the first-order volume correction, we have

$$\gamma = \frac{1}{1 + \mu} (\gamma_o + \alpha \mu).$$

Substituting $\gamma$ and $P_H$ into the Mie-Gruneisen equation yields

$$P = \frac{\rho_o C_o^2 \mu \left[ 1 + \left( 1 - \frac{Y_o}{Z} \right) \mu - \frac{\alpha}{Z} \mu^2 \right]}{(1 + \mu - s \mu)^2} + (\gamma_o + \alpha \mu) E^v \quad (Eq. 5.1)$$

where $E^v = E/V_o$, the energy per unit initial volume. This equation can be used to compute the pressures of each of the three soil phases individually. The bulk modulus of the material is given as follows:

$$K = \frac{\rho_o C_o^2 \mu \left[ 1 + \left( 1 - \frac{Y_o}{Z} \right) \mu - \frac{\alpha}{Z} \mu^2 \right] \left[ 1 + \frac{2 \mu (s - 1)}{1 + \mu - s \mu} + \frac{\mu (\gamma_o + \alpha \mu)}{(1 + \mu)^2} \right] + \rho_o C_o^2 \mu \left( 1 - \frac{Y_o}{Z} - \alpha \mu \right)}{(1 + \mu - s \mu)^2}

... + \left[ \frac{(\gamma_o + \alpha \mu)^2}{(1 + \mu)^2} + \alpha \right] E^v \quad (Eq. 5.2).$$

The changes of volume fractions can be computed over the course of a pressure change in the material. Letting $A_s$, $A_w$, and $A_a$ be the initial volume fractions of the solid, water, and air phases respectively; $A^*_s$, $A^*_w$, and $A^*_a$ the new volume fractions under the change in pressure; and $\rho_s$, $\rho_w$, and $\rho_a$ the initial densities, we have the following relations:

$$A_s + A_w + A_a = 1$$

$$A_s \rho_s + A_w \rho_w + A_a \rho_a = \rho_o.$$
The changes in volume fraction can be computed as

\[
A_s^* = A_s \left( \frac{k_s (P_s - P_0)}{\rho_s C_0^2} + 1 \right)^{-1/k_s} \quad (Eq. 5.3a)
\]

\[
A_w^* = A_w \left( \frac{k_w (P_w - P_0)}{\rho_w C_0^2} + 1 \right)^{-1/k_w} \quad (Eq. 5.3b)
\]

\[
A_a^* = A_a \left( \frac{P_a}{P_0} \right)^{-1/k_a} \quad (Eq. 5.3c)
\]

where the \( k_s \), \( k_w \), and \( k_a \) are the exponents of the specific entropy for the solid water and air phases respectively. The soil density under pressure is

\[
\rho = \rho_0 (A_s^* + A_w^* + A_a^*)^{-1}
\]

Denoting the weight fractions of the solid, water and air phases \( R_s \), \( R_w \), and \( R_a \) respectively, we have

\[
E = R_s E_s + R_w E_w + R_a E_a
\]

and,

\[
V = R_s V_s + R_w V_w + R_a V_a .
\]

### 5.2 Numerical Implementation of the Equation of State

In this section, the simple numerical procedure for implementing the three-phase equation of state is outlined in pseudocode. In LS-DYNA, the program passes the user defined EOS the change in volumetric strain (along with several history variables which include, among others, energy, volume fractions, previous volumetric changes for each phase), and the corresponding pressure, or bulk modulus is computed (depending on which is requested by the program). For the full source code, see Appendix C.

1. If first = true (first call of subroutine at integration node), initialize history variables

   \[
   \text{hist}(1) = A_{s,0} \ , \ \text{hist}(2) = A_{w,0} \ , \ \text{hist}(3) = A_{a,0} \\]

   \[
   \text{hist}(4) = 0.0 \ , \ \text{hist}(5) = 0.0 \ , \ \text{hist}(6) = 0.0
   \]
2. Compute weight ratios for each phase
   \[ R_s = A_s \rho_{s,0}/\rho_o \], and similarly for water and air phases

3. Determine the specific energy for each phase
   \[ E_s = ER_s \], and similarly for water and air phases

4. Compute the volume ratios (df = V/Vₜ) for each phase
   \[ df_s = df \cdot A_s/V_o \], and similarly for water and air phases

5. Compute \( \mu \) for composite and each phase
   \[ \mu = \frac{1}{df} - 1 \], \( \mu_s = \frac{1}{df_s} - 1 \), and similarly for water and air phases

6. Compute the original volumes of solid, air and water
   \[ V_{s,prev} = (df \cdot V_o - 2 \cdot dvol) \cdot A_s \], and similarly for water and air phases

7. If iflag = 0, compute bulk modulus (Eq. 5.2), and return

8. If iflag = 1, compute the pressures of each phase using Eq. 5.1, and ensure all are above
   the cutoff pressure (pcut), continue

9. Compute the specific energy for each phase, and sum
   \[ E_s^\nu = E_s^\nu - 0.5 \cdot P_s \cdot dvols/(A_{s,o} \cdot V_o) \]
   
   \[ \ldots \]
   \[ E^\nu = E_s^\nu + E_w^\nu + E_a^\nu \]

10. If dvol \( \neq 0.0 \), compute new pressure by
    \[ P_{new} = \frac{P_s \cdot dvols + P_w \cdot dvolw + P_a \cdot dvola}{dvol} \]
    
    else,
    \[ P_{new} = P_s + P_w + P_a \]

11. Compute new volume ratios (Aₜ,ₚ, Aₜ,ₖ, and Aₜ,₇) by Equations 5.3a-c
12. Compute volumetric changes to each phase:

\[ d\text{vol}_s = df \cdot V_o \cdot A^*_s - V_{s,prev} \]

\[ d\text{vol}_a = df \cdot V_o \cdot A^*_w - V_{w,prev} \]

\[ d\text{vol}_l = df \cdot V_o \cdot A^*_a - V_{a,prev} \]

13. Update history variables:

\[ \text{hist}(1) = A^*_s \quad \text{hist}(2) = A^*_w \quad \text{hist}(3) = A^*_a \]

\[ \text{hist}(4) = d\text{vol}_s \quad \text{hist}(5) = d\text{vol}_a \quad \text{hist}(6) = d\text{vol}_l \]

14. Return, end.
### 5.3 Suitable parameters for the equation of state

The parameters reported here are reported by Chapman et al. (2006), and An et al. (2011) to model blasts in saturated and unsaturated sands. The various physical characteristics for each sand type with different levels of saturation are reported in Tables 5.1 and 5.2. The method by which these parameters are estimated is described in full in Appendix E.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol (dimension)</td>
<td>A (\text{--})</td>
<td>$\rho_0$ (kg/m$^3$)</td>
<td>$C_0$ (m/s)</td>
<td>s (\text{--})</td>
<td>k (\text{--})</td>
<td>$\gamma_0$ (\text{--})</td>
</tr>
<tr>
<td>Saturated soil</td>
<td>1.00</td>
<td>2,055</td>
<td>320</td>
<td>4.92</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>Solid</td>
<td>0.70</td>
<td>2,650</td>
<td>6,319</td>
<td>1.41</td>
<td>3</td>
<td>1.00</td>
</tr>
<tr>
<td>Water</td>
<td>0.20</td>
<td>1,000</td>
<td>1,460</td>
<td>2.00</td>
<td>7</td>
<td>0.60</td>
</tr>
<tr>
<td>Air</td>
<td>0.10</td>
<td>1.2</td>
<td>241</td>
<td>1.06</td>
<td>1.4</td>
<td>0.00</td>
</tr>
<tr>
<td>Dry soil</td>
<td>1.00</td>
<td>1,802</td>
<td>530</td>
<td>1.64</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>Solid</td>
<td>0.68</td>
<td>2,650</td>
<td>6,319</td>
<td>1.41</td>
<td>3</td>
<td>1.00</td>
</tr>
<tr>
<td>Water</td>
<td>0.00</td>
<td>1,000</td>
<td>1,460</td>
<td>2.00</td>
<td>7</td>
<td>0.60</td>
</tr>
<tr>
<td>Air</td>
<td>0.32</td>
<td>1.2</td>
<td>241</td>
<td>1.06</td>
<td>1.4</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Table 5.1:** Parameters for use in the three-phase EOS model for dry and saturated sand, used by An et al. (2011) for the so-called ARL sand with porosity 31.23%.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol (dimension)</td>
<td>A (\text{--})</td>
<td>$\rho_0$ (kg/m$^3$)</td>
<td>$C_0$ (m/s)</td>
<td>s (\text{--})</td>
<td>k (\text{--})</td>
<td>$\gamma_0$ (\text{--})</td>
</tr>
<tr>
<td>Saturated soil (22% by mass)</td>
<td>1.00</td>
<td>1,840 ± 50</td>
<td>320</td>
<td>4.92</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>Solid</td>
<td>0.57</td>
<td>2,650</td>
<td>6,319</td>
<td>1.41</td>
<td>3</td>
<td>1.00</td>
</tr>
<tr>
<td>Water</td>
<td>0.41</td>
<td>1,000</td>
<td>1,460</td>
<td>2.00</td>
<td>7</td>
<td>0.60</td>
</tr>
<tr>
<td>Air</td>
<td>0.02</td>
<td>1.2</td>
<td>241</td>
<td>1.06</td>
<td>1.4</td>
<td>0.00</td>
</tr>
<tr>
<td>Part. Sat. soil (20% by mass)</td>
<td>1.00</td>
<td>1,810 ± 50</td>
<td>710</td>
<td>2.90</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>Solid</td>
<td>0.57</td>
<td>2,650</td>
<td>6,319</td>
<td>1.41</td>
<td>3</td>
<td>1.00</td>
</tr>
<tr>
<td>Water</td>
<td>0.38</td>
<td>1,000</td>
<td>1,460</td>
<td>2.00</td>
<td>7</td>
<td>0.60</td>
</tr>
<tr>
<td>Air</td>
<td>0.05</td>
<td>1.2</td>
<td>241</td>
<td>1.06</td>
<td>1.4</td>
<td>0.00</td>
</tr>
<tr>
<td>Part. Sat. soil (10% by mass)</td>
<td>1.00</td>
<td>1,530 ± 50</td>
<td>230</td>
<td>2.26</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>Solid</td>
<td>0.57</td>
<td>2,650</td>
<td>6,319</td>
<td>1.41</td>
<td>3</td>
<td>1.00</td>
</tr>
<tr>
<td>Water</td>
<td>0.10</td>
<td>1,000</td>
<td>1,460</td>
<td>2.00</td>
<td>7</td>
<td>0.60</td>
</tr>
<tr>
<td>Air</td>
<td>0.33</td>
<td>1.2</td>
<td>241</td>
<td>1.06</td>
<td>1.4</td>
<td>0.00</td>
</tr>
<tr>
<td>Dry soil (0% by mass)</td>
<td>1.00</td>
<td>1,430 ± 50</td>
<td>530</td>
<td>1.64</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>Solid</td>
<td>0.57</td>
<td>2,650</td>
<td>6,319</td>
<td>1.41</td>
<td>3</td>
<td>1.00</td>
</tr>
<tr>
<td>Water</td>
<td>0.00</td>
<td>1,000</td>
<td>1,460</td>
<td>2.00</td>
<td>7</td>
<td>0.60</td>
</tr>
<tr>
<td>Air</td>
<td>0.43</td>
<td>1.2</td>
<td>241</td>
<td>1.06</td>
<td>1.4</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Table 5.2:** Parameters adapted for use in the three-phase EOS model for dry, partially saturated, and saturated sand based on quartz sand tested by Chapman et al. (2006) in plate impact experiments, with an average particle size of 230 μm and porosity of 43%.
Chapter 6: Numerical simulation

In order to determine the validity and the performance of the subroutines developed as part of this research project, several preliminary numerical simulations were conducted and compared to results readily available in the open literature for similar materials. In this chapter, the subroutines for the constitutive model and EOS are run with parameters reported by An et al. (2011) and compared to experimental results reported in the same paper.

6.1 Model Geometry and Parameters

The geometry used to model the experimental results reported by An et al. (2011) is an axisymmetric representation using the arbitrary Lagrange-Eulerian method in LS-DYNA. The explosive is modelled using the JWL high explosive with an equivalent weight of 100 grams buried at three centimeters below the air-sand interface. The parameters for the JWL high explosive are given in Table 6.1. The air above is modelled with a linear polynomial EOS with the parameters listed in Table 6.2.

The sand immediately surrounding the high explosive is modelled with the three-phase EOS (user defined u eos23) in addition to the deviatoric part of the Perzyna type constitutive model (umat47). Where lower maximum strains are expected, further afield of the explosive, the material is modelled with the full deviatoric/volumetric constitutive model (umat48).

Simulations were conducted for sand in the unsaturated and fully saturated conditions. The soil input parameters for both cases are available in Tables 6.3 and 6.4.

The model geometry, boundary conditions, and material locations are given in Figure 6.1. A mesh detail is provided in Figure 6.2. Mesh dimensions in the explosive and immediately adjacent are approximately 2.5 millimeters by 2.5 millimeters.

<table>
<thead>
<tr>
<th>A (MPa)</th>
<th>B (MPa)</th>
<th>R1</th>
<th>R2</th>
<th>ω</th>
<th>E0 (MPa)</th>
<th>V0</th>
</tr>
</thead>
<tbody>
<tr>
<td>609970</td>
<td>12950</td>
<td>4.5</td>
<td>1.4</td>
<td>0.25</td>
<td>9000</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.1: JWL Equation of state parameters for the C4 explosive simulated.

<table>
<thead>
<tr>
<th>C0</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>E0 (MPa)</th>
<th>V0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.4</td>
<td>0.0</td>
<td>0.257</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 6.2: Parameters for the linear-polynomial equation of state for air.
Figure 6.1: Geometry, materials, and boundary conditions of the numerical model used to simulate the explosive tests. All dimensions in millimeters. Not to scale.
Figure 6.2: Mesh detail of the explosive, air, and sand. Minimum mesh sizing is 2.5 x 2.5 mm in the explosive charge and in the adjacent sand elements.
Table 6.3: Parameters used in the constitutive model (umat47 and umat48) for the dry and saturated cases.

<table>
<thead>
<tr>
<th>Model</th>
<th>Subroutine name</th>
<th>Parameter</th>
<th>Dimension</th>
<th>Dry Case</th>
<th>Saturated Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>umat 27/28</td>
<td>cm(1)</td>
<td>K1 - load</td>
<td>Pa</td>
<td>106.4e6</td>
<td>1e9</td>
</tr>
<tr>
<td></td>
<td>cm(2)</td>
<td>K2 - load</td>
<td>1/Pa</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>cm(3)</td>
<td>G1 - load</td>
<td>Pa</td>
<td>63.85e6</td>
<td>2e7</td>
</tr>
<tr>
<td></td>
<td>cm(4)</td>
<td>G2 - load</td>
<td>1/sqrt(Pa)</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>cm(5)</td>
<td>K1 - unload</td>
<td>Pa</td>
<td>106.4e6</td>
<td>1e9</td>
</tr>
<tr>
<td></td>
<td>cm(6)</td>
<td>K2 - unload</td>
<td>1/Pa</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>cm(7)</td>
<td>G1 - unload</td>
<td>Pa</td>
<td>63.85e6</td>
<td>2e7</td>
</tr>
<tr>
<td></td>
<td>cm(8)</td>
<td>G2 - unload</td>
<td>1/sqrt(Pa)</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>cm(9)</td>
<td>α</td>
<td>Pa</td>
<td>64,200</td>
<td>62,500</td>
</tr>
<tr>
<td></td>
<td>cm(10)</td>
<td>β</td>
<td>1/Pa</td>
<td>3.4283e-7</td>
<td>3.643e-7</td>
</tr>
<tr>
<td></td>
<td>cm(11)</td>
<td>γ</td>
<td>Pa</td>
<td>5890</td>
<td>3200</td>
</tr>
<tr>
<td></td>
<td>cm(12)</td>
<td>θ</td>
<td>1/Pa</td>
<td>0.18257</td>
<td>0.2490</td>
</tr>
<tr>
<td></td>
<td>cm(13)</td>
<td>W</td>
<td>0.2142</td>
<td>0.225</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cm(14)</td>
<td>D</td>
<td>1/Pa</td>
<td>9.52e-9</td>
<td>8.84e-9</td>
</tr>
<tr>
<td></td>
<td>cm(15)</td>
<td>R</td>
<td>5.0</td>
<td>5.320</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cm(16)</td>
<td>X0</td>
<td>Pa</td>
<td>10000</td>
<td>10000</td>
</tr>
<tr>
<td></td>
<td>cm(17)</td>
<td>T</td>
<td>Pa</td>
<td>6900</td>
<td>7200</td>
</tr>
<tr>
<td></td>
<td>cm(18)</td>
<td>η</td>
<td>1/s</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>cm(19)</td>
<td>N</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cm(20)</td>
<td>f0</td>
<td>Pa</td>
<td>1.0e11</td>
<td>1.2e11</td>
</tr>
<tr>
<td></td>
<td>cm(21)</td>
<td>ηT</td>
<td>Pa</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>cm(22)</td>
<td>itmax</td>
<td></td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>cm(23)</td>
<td>tol (ρ)</td>
<td></td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Density (ρ) kg/m³</td>
<td>1430</td>
<td>1840</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.3: Example input deck for the user defined material model (in this case umat48 for the dry case) in kg-m-s dimensions. For more information see the LS-DYNA manual, Appendix A.
<table>
<thead>
<tr>
<th>Model</th>
<th>Subroutine name</th>
<th>Parameter</th>
<th>Dimension</th>
<th>Dry Case</th>
<th>Saturated Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>ueos23s</td>
<td>eosp(1)</td>
<td>As0</td>
<td>0.68</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(2)</td>
<td>Aw0</td>
<td>0.00</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(3)</td>
<td>Aa0</td>
<td>0.32</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(4)</td>
<td>rhoS0</td>
<td>kg/m³</td>
<td>2,650</td>
<td>2,650</td>
</tr>
<tr>
<td></td>
<td>eosp(5)</td>
<td>rhoW0</td>
<td>kg/m³</td>
<td>1000</td>
<td>1,000</td>
</tr>
<tr>
<td></td>
<td>eosp(6)</td>
<td>rhoA0</td>
<td>kg/m³</td>
<td>1.2</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>eosp(7)</td>
<td>kS</td>
<td></td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>eosp(8)</td>
<td>kW</td>
<td></td>
<td>7.0</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>eosp(9)</td>
<td>kA</td>
<td>1.4</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(10)</td>
<td>C</td>
<td>m/s</td>
<td>320</td>
<td>320</td>
</tr>
<tr>
<td></td>
<td>eosp(11)</td>
<td>s1</td>
<td>4.92</td>
<td>4.92</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(12)</td>
<td>gama0</td>
<td>0.11</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(13)</td>
<td>alpha</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(14)</td>
<td>Cs</td>
<td>m/s</td>
<td>6319</td>
<td>6319</td>
</tr>
<tr>
<td></td>
<td>eosp(15)</td>
<td>s1s</td>
<td>1.41</td>
<td>1.41</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(16)</td>
<td>gama0w</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(17)</td>
<td>alphaS</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(18)</td>
<td>Cw</td>
<td>m/s</td>
<td>1460</td>
<td>1460</td>
</tr>
<tr>
<td></td>
<td>eosp(19)</td>
<td>s1w</td>
<td>2.0</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(20)</td>
<td>gama0w</td>
<td>0.6</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(21)</td>
<td>alphaW</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(22)</td>
<td>Ca</td>
<td>m/s</td>
<td>240.6</td>
<td>240.6</td>
</tr>
<tr>
<td></td>
<td>eosp(23)</td>
<td>s1a</td>
<td>1.0602</td>
<td>1.0602</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(24)</td>
<td>gama0a</td>
<td>0.4</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(25)</td>
<td>alphaA</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eosp(26)</td>
<td>p0</td>
<td>Pa</td>
<td>10000</td>
<td>10000</td>
</tr>
<tr>
<td></td>
<td>eosp(27)</td>
<td>Kcut</td>
<td>Pa</td>
<td>100e9</td>
<td>100e9</td>
</tr>
</tbody>
</table>

Table 6.4: Parameters used in the user defined three-phase equation of state (ueos23s).

```
*EOS_USER_DEFINED
$# eoid eost lmc nhv ivect eo vo bulk
 4 23 27 6 0 0.00 1.000 0.000
$# As0 Aw0 Aa0 rhoS0 rhoW0 rhoA0 kS kW
 0.68 0.0 0.32 2650 1000 1.2 3.0 7.0
$# kA C s1 gama0 alpha Cs s1s gama0s
 1.4 320 4.92 0.11 0.000 6319 1.41 1.0
$# alphaS Cw s1w gama0w alphaW Ca s1a gama0a
 0.000 1460 2.0 0.6 0.000 240.6 1.0602 0.4
$# alphaA p0 Kcut
 0.000 10000 100e9
```

Figure 6.4: Example input deck for user defined EOS (in this case ueos23s for the dry case) in kg-m-s dimensions. For more information see the LS-DYNA user manual, Appendix B.
6.2 Model results

As stated in the previous section, to evaluate the credibility of the subroutines in LS-DYNA, the explosion test results presented in the An et al. (2011) have been used. In general, six explosion tests have been reported: three for the dry sand; and three for the saturated sand. Figure 6.5 and 6.6 show the air shock pressures predicted by numerical simulation (pink square), the experimental results (dark blue circle), and average values of the experimental results (red star) at the distances of the 30, 70, 110 cm above the soil (the heights of the transducers). The difference between the averages of experimental results and predicted values shows that the predicted results have a quite good agreement with the experimental results (except for the results for saturated sand at 30 cm distance).

In addition, graphical expressions of the distribution of the air shock pressure and strain rate near the explosive material (C4) are shown in Figures 6.7 and 6.8. Based on the simulation results, the maximum strain rate in soil has been around $2.0 \times 10^5 \text{ s}^{-1}$ and occurred at the beginning of the explosion in the sand near of the explosive material. By moving the wave in the sand, the strain rate decreases as is expected.
**Figure 6.5:** Comparison between air shock pressures of experimental results and numerical simulation in dry sand for distance 30, 70, and 110 cm directly above the soil.

**Figure 6.6:** Comparison between air shock pressures of experimental results and numerical simulation in saturated sand for distance 30, 70, and 110 cm directly above the soil.
Figure 6.7: A graphical expression of air shock pressure for saturated sand in different elapsed time.
Figure 6.8: A graphical expression of strain rate for saturated sand in different elapsed time.
Conclusion

In this report, a comprehensive investigation into the dynamic constitutive behaviour of sand has been presented. Beginning with data from dynamic tests (both Split Hopkinson Pressure Bar tests and Plate Impact experiments), an understanding of the rate dependent material behaviour was derived. Building on the developments of other researchers, a Perzyna-type viscoplastic model and a three-phase equation of state were implemented numerically. The non-unique Perzyna viscoplastic model was calibrated using a Marquardt-Levenberg optimisation algorithm, and the EOS parameters were derived from known properties of sand and its constituent phases. Finally, the validity of the numerical models was tested against explosion tests available from the literature.

In addition to a comprehensive description of the constitutive behaviour of sand over several orders of magnitude of strain rate, this report can be used as framework for the future study of other types of soils such as clay, silt, and gravel.
References


An, J. Soil Behavior under Blast Loading, Ph.D. dissertation, University of Nebraska - Lincoln


Appendix A: Subroutine implementation in LS-DYNA

This appendix gives a brief overview of how to compile the source code written for the user-defined constitutive equations and equations of state into an LS-DYNA executable.

1. Download the required library from ftp.lstc.com. This library will contain source files (dyn21.F, dyn21b.f), an executable (nmake.exe for windows), a readme.txt and various other object files. For example, on a Windows machine with symmetric multiprocessing (SMP) for single precision at 64 bits, download:

   ls-dyna_smp_s_R711_winx64_ifort131_lib.zip

2. Consult the readme.txt for the necessary compilers that will be required to compile the source. For the example library in Step 1, the C++ compiler is “Microsoft Visual C++ 2010 x64 cross tools” and the FORTRAN compiler is “Intel Parallel Studio XE 2013”

3. Download and install the Microsoft C++ compiler. For the example, the compiler package containing the compiler is Microsoft Visual Studio 2010 (not the Express edition!).

4. Download and install the FORTRAN compiler. For the example, the compiler is included in Intel Parallel Studio XE 2013 (not XE 2013 Service Pack 1!). Be sure to integrate the intel compiler with MS Visual Studio during installation.

5. Once the exact compilers are installed correctly and in the order specified, open dyn21.f in a FORTRAN code editor of your choice. Find and highlight the lines of code beginning and ending with:

   subroutine umat48 (cm, eps, sig, epsp, hsv, dt1, capa, & etype, faille, crv, cma, qmat, elsiz, idele )
   ...
   ...
   return
   end

Delete this portion of the code. Replace with the code provided in Appendix B for the Perzyna type viscoplastic model. Do the same for umat47. Be sure all FORTRAN77 structured code conventions are followed properly (i.e. with respect to spacing and maximum number of characters per line). Note that there are some functions and subroutines defined that are outside of the new umat48 subroutine which are also required.
6. Open dyn21b.f. Find and highlight the lines of code beginning and ending with:

```fortran
subroutine ueos23s (iflag, cb, pnew, hist, rho0, eosp, specen, & df, dvol, v0, pc, dt, tt, crv, first)
  ...
  ...
  return
end
```

Delete this portion of the code. Replace with the code provided in Appendix C for the three-phase EOS model. Be sure all FORTRAN77 structured code conventions are followed properly.

7. Open the “Intel 64 Parallel Studio 2010 Mode” (or equivalent, as specified by the readme.txt file) command prompt and change directory to the appropriate directory where the source and nmake.exe files are stored. For example:

```bash
> cd C:\Users\Joe\Desktop\usermat
```

Then type (if on Windows):

```bash
> nmake.exe
```

or “make” if on Linux. The source code will then be compiled into an executable called lsdyna.exe.

8. Move the lsdyna.exe to the directory containing the ls-dyna program files. For example copy and paste into

```bash
C:\...\LSDYNA\program
```

9. From the LS-DYNA Program Manager, select the tab solver and select the solver lsdyna.exe that has been compiled by the user. The user defined EOS and Constitutive models are now be available for use from directly within the LS-DYNA prepost corresponding to EOS number 23, and user defined material model numbers 47 and 48.
Appendix B: User defined constitutive model

B.1 Deviatoric/Volumetric constitutive model source code and associated subroutines

```fortran
subroutine umat48 (cm,eps,sig,eps,p,hsv,dt1,capa,etype,tt
1 temper,failel,crv,cma,qmat,elsiz,idele)
* 
include 'nlqparm'
include 'bk06.inc'
include 'iounits.inc'
dimension crv(lq1,2,*),cma(*),qmat(3,3)
logical failel
character*5 etype

* * *
* A SUBROUTINE that computes the strain controlled stresses
* for an non-linear elastic-viscoplastic material based on the
* Perzyna model
* Patrick Kanopoulos and Mohammadamin Jafari, March 13, 2015
* University of Toronto, Department of Civil Engineering
* Update for CC-nonlinear, different formulation in loading and
* unloading
* * *
* INPUT:
* cm an array containing the material constants (see below)
* eps strain increment array (solid mech. sign convention)
* sigma current stress state array
* hsv material history variables (see below)
* dt time step
* OUTPUT:
* sigma return stress as a result of strain increment
* hsv updated history variables
* SUBROUTINES:
* matvcm matrix-vector product
* matinv (small) matrix inversion
* drvalt computes the spatial and material gradients related to
* the failure envelope
* FUNCTIONS
* FFe failure surface component [ f = sqrt(J2) - FFe(I1) ]
* LL I1 value indicating intersection of cap with failure
* surface
* fyield computes the value of the yield function
* phi computes the value of the viscoplastic function phi
* [ phi = ( f / f0 )^N ]
* DEFINITIONS
* cm(1) K1d, initial loading bulk modulus (MPa)
* cm(2) K2d, exponent of loading bulk modulus (-)
```
Initialize variable names and dimensions

```
* implicit none
* IO
real cm(23), eps(6), sig(6), hsv(1), dt1
* Stresses
real sigma(6,1), sigtri(6,1),
  & dsgitr(6,1), sstri(6,1), I1tri, J2tri, printri, signew(6,1),
  & I1new, J2new, ssnew(6,1), rj2tri, prsnew, Jratio
* Strains
real deps(6,1), depsv, epsvp, epsvp0, depsvp
* parameters
real K1d, K2d, G1d, G2d, K1un, K2un, G1un, G2un, alpha, beta,
  & gama, theta, W, D, R, T, hta, N, lame, f0,
  & CC(6,6), CCsup(6,6), CCinv(6,6), X0, htaT
* real K1, K2, G1, G2
* hardening parameters
real kappa0, kappaN, dkappa, Xitr
* time domain parameters
real dt
* Domain geometry variables
real Lk, FFeTT, ftri
integer domain
integer vpyield
* return mapping variables
real lamdaN, rhoN,
```
& conv, HH(6,6), HHinv(6,6), xi, xi1(6,1),
& x12(6,1), x13(1,1), x14, xxPL(6,1), yyPL(6,1)
* derivatives
real dfs(6,1), ddfs(6,6), ddfs1(6,1), dphis(6,1), dphi1, dkl,
& tdphis(1,6)
* function calls
real ll, FFe, fyield, phi, XX
* counting variables and iteration variables
integer i, ii, mxiter
* other variables
real phiN, lamoh}

* VARIABLE ASSIGNMENT

* assign the stresses, sigma // convert to g.mech convention
sigma(1,1) = -sig(6)
sigma(2,1) = -sig(6)
sigma(3,1) = -sig(6)
sigma(4,1) = -sig(6)
sigma(5,1) = -sig(6)
sigma(6,1) = -sig(6)
* assign the strain increment, deps // convert to g.mech convention
deps(1,1) = -eps(1)
deps(2,1) = -eps(2)
deps(3,1) = -eps(3)
deps(4,1) = -eps(4)
deps(5,1) = -eps(5)
deps(6,1) = -eps(6)
* Assign elastic constants
K1d = cm(1)
K2d = cm(2)
G1d = cm(3)
G2d = cm(4)
K1un = cm(5)
K2un = cm(6)
G1un = cm(7)
G2un = cm(8)
* assign plastic constants
alpha = cm(9)
beta = cm(10)
gama = cm(11)
theta = cm(12)
W = cm(13)
D = cm(14)
R = cm(15)
X0 = cm(16)
T = cm(17)
* assign viscosity constants
hta = cm(18)
N = cm(19)
f0 = cm(20)
htaT = cm(21)
* Set default maximum iteration values
mxiter = int(cm(22))
* Set the convergence criterion
conv = cm(23)
* assign initial value of hardening parameter
kappa0 = hsv(1)
kappaN = kappa0
* assign timestep
dt = dt1

* MAIN PROGRAM

* set elastic constants based on loading or unloading
depsv = deps(1,1) + deps(2,1) + deps(3,1)
if (depsv .GE. 0.0) then
  K1 = K1d
  K2 = K2d
  G1 = G1d
  G2 = G2d
else
  K1 = K1un
  K2 = K2un
  G1 = G1un
  G2 = G2un
endif
* assemble the elastic stiffness tensor
CC = 0
ccnlin( CC, sigma, deps, K1, K2, G1, G2 )
CCsup = CC
* compute the inverse stiffness tensor CCinv calling matinv
ccninv(CCsup,6,CCinv)
* compute the elastic estimator stress, sigma^est_i
dsigtr = CC * depsilon
matvcm(6,6,CC,deps,dsigtr)
sigtri = sigma + dsigtr
* determine I1trial
I1tri = sigtri(1,1)+sigtri(2,1)+sigtri(3,1)
* determine J2trial (from pressure and ss trial)
prstri = I1tri/3.0
sstri = sigtri
sstri(1,1) = sigtri(1,1) - prstri
sstri(2,1) = sigtri(2,1) - prstri
sstri(3,1) = sigtri(3,1) - prstri
J2tri = 0.5*(sstri(1,1)**2 + sstri(2,1)**2 + sstri(3,1)**2) +
& sstri(4,1)**2 + sstri(5,1)**2 + sstri(6,1)**2
* Determine case (i.e. Tension lower quadrant, Tension higher
quadrant, Yield, Cap)
* Tension: domain 1  I1trial <= -T, and, root(J2trial) < FF (-T)
Tension: domain 2  $I_{1trial} \leq -T$, and, $\sqrt{J_{2trial}} \geq FFe(-T)$

Yield: domain 3  $-T \leq I_{1trial} \leq L(k)$

Cap: domain 4  $I_{1trial} > L(k)$

$LL_k = LL(kappa_0)$
$FFeTT = FFe(T, alpha, gamma, beta, theta)$
$rJ2tri = \sqrt{J_{2trial}}$

if $(I_{1tri}.GT. LL_k)$ then
  domain = 4
elsif ($(I_{1tri}.LE. LL_k) .AND. (I_{1tri}.GT.-T))$ then
  domain = 3
elsif ($(I_{1tri}.LE.-T) .AND. (rJ2tri.GE.FFeTT))$ then
  domain = 2
elsif ($(I_{1tri}.LE.-T) .AND. (rJ2tri.LT.FFeTT))$ then
  domain = 1
endif

Compute the value of $fyield$
$ftri = fyield(I_{1tri}, J_{2trial}, kappa_0, domain, alpha, beta, gamma, & theta, R, T)$

Set logical yield variable ($vpyield$) to 1 or 0 if ($ftri.GT.0.) then
  $vpyield = 1$
else
  $vpyield = 0$
endif

-------------------------------------------------------------------------------------------------

 MAIN if STATEMENT -- ELASTIC, RET. MAPPING, OR TENSION C.O.

-------------------------------------------------------------------------------------------------

 If elastic step - return elastic estimator
 If viscoplastic step - do return mapping
 If tension step - return tension cutoff stresses

if ($(domain .EQ. 4 .OR. domain .EQ. 3) .AND. vpyield .EQ. 0) then
  output stress is simply the elastic trial stress,
  hardening parameter (kappa) remains unchanged.
  $signew = sigtri$
  $kappaN = kappa_0$

elseif ($(domain .EQ. 4 .OR. domain .EQ. 3) & .AND. vpyield .EQ. 1) then
  viscoplastic step! Do return mapping algorithm
  initialize initial variables (lambda, sigma, rho)
  $lambdaN = 0.0$
  $signew = sigtri$
  $rhoN = phi(signew,kappa_0,domain,N,f0, alpha, beta, gamma, theta, R, T) & - lambdaN / (hta*dt)$
  compute initial volumetric $vp$ strain, and initialize total and
  incremental strains
  $epsvp0 = W * (1 - exp(-D*}$
& (XX(kappa0,R, alpha, gama, beta, theta) - X0))

depsvp = 0.0
epsvp = 0.0

* loop through iterative improvements of signew and kappa
* do 100 ii = 1, mxiter
* find stress invariants for current stress state
I1new = signew(1,1) + signew(2,1) + signew(3,1)
prsnew = I1new/3.0
ssnew = signew
ssnew(1,1) = signew(1,1) - prsnew
ssnew(2,1) = signew(2,1) - prsnew
ssnew(3,1) = signew(3,1) - prsnew
J2new = 0.5*(ssnew(1,1)**2 + ssnew(2,1)**2 + ssnew(3,1)**2) +
& signew(4,1)**2 + signew(5,1)**2 + signew(6,1)**2

* compute stiffness tensor
call CCnlin(Cc, signew, deps, K1, K2, G1, G2)
CCsup = CC
call matinv(CCsup, 6, CCinv)
* call derivative subroutine -- returns necessary derivatives of
* cap, failure, and phi functions
call drvalt(I1new, J2new, ssnew, kappaN, domain, alpha, beta,
& gama, theta, W, D, R, X0, N, f0, dfs, ddfs, ddfl,
& dphis, dphil, dkl, lamdaN)

* compute the HH vector // calling function minv (matrix inverse)
HHinv = CCinv + lamdaN * ddfs
* invert HHinv matrix to find HH, calling matinv
call matinv(HHinv, 6, HH)
* take the transpose of dphis
do 235 i = 1, 6
Tdphis(1,i)=dphis(i,1)
235 continue

* compute xi calling the matrix-vector multiplication subroutine
* xi = Tdphis * HH * (dfs + lamdaN*ddfs1) + 1/(hta*dt) - dphi
xi1 = dfs + lamdaN*ddfs1
call matvcm(6,6,HH,xi1,xi2)
call matvcm(1,6,Tdphis,xi2,xi3)
xi4 = xi3(1,1)
xi = xi4 + ( 1.0 / (hta*dt) ) - dphil

* compute lamdaN
lamdaN = lamdaN + rhoN/ksi

* compute {signew} = {signew} + [CC]*{depsilon - lamdaN*dfs}
xxPL = deps - lamdaN * dfs
call matvcm(6, 6, CC, xxPL, yyPL)
signew = sigma + yyPL
* Compute kappaN from dkappa = delta_lamda_N * dkappa/dlamda
depsvp = lamdaN*( dfs(1,1) + dfs(2,1) + dfs(3,1) )
if (depsvp .LT. 0.0) depsvp = 0.0
epsvp = epsvp0 + depsvp
if ( epsvp .GE. (0.99*W) ) then
epsvp = 0.99*W
signew = sigtri
kappaN = kappa0
goto 300
endif
Xitr = X0 - log( 1 - epsvp/W ) / D
call ksolve(kappaN, Xitr, R, alpha, beta, gama, theta)
* Check the convergence by computing rho_N
phiN = phi(signew,kappaN,.domain,N,f0,alpha,beta,gama,theta,R,T)

* compute rho for current iteration
lamoht = lamdaN / (hta*dt)
 rhoN = phiN - lamoht

* Check for the convergence condition
if ( abs(rhoN) .LE. abs(conv) ) then
   goto 300
endif
end loop

100 continue

if ( abs(rhoN) .gt. abs(conv)) then
   write(*,*) 'Not Converged, ii = ', ii, 'rho_N = ', rhoN
endif

*---------------------------------------------------------------
elseif (domain .EQ. 2 .OR. domain .EQ. 1 ) then
* Tension cutoff stresses obtained - return tensile stresses from
* the tension formulas
if (domain .EQ. 1) then
* compute I1_new, pressure_new, J2_new, and 'Jratio' for domain 1
I1new = exp(-htaT*dt)*I1tri + (1-exp(-htaT*dt))*(-T)
prsnew = I1new/3.0
J2new = J2tri
Jratio = 1.0
else
* compute I1_new, pressure_new, J2_new, and 'Jratio' for domain 2
I1new = exp(-htaT*dt)*I1tri + (1-exp(-htaT*dt))*(-T)
prsnew = -I1new/3.0
& J2new = ( exp(-htaT*dt)*sqrt(J2tri) + (1-exp(-htaT*dt))**2
& Jratio = sqrt(J2new)/sqrt(J2tri)
endif

* Update the stresses for the new tensile regime
signew(1,1) = sstri(1,1)*Jratio + prsnew
signew(2,1) = sstri(2,1)*Jratio + prsnew
signew(3,1) = sstri(3,1)*Jratio + prsnew
signew(4,1) = sstri(4,1)*Jratio
signew(5,1) = sstri(5,1)*Jratio
signew(6,1) = sstri(6,1)*Jratio
endif

300 continue

*---------------------------------------------------------------
* STRESS OUTPUT
*
* update output variables sig // convert to solid mech. convention
sig(1) = -signew(1,1)
sig(2) = -signew(2,1)
sig(3) = -signew(3,1)
sig(4) = -signew(4,1)
sig(5) = -signew(5,1)
sig(6) = -signew(6,1)

* update history variables
hsv = kappaN

return
dend

---------------------------

FUNCTION DEFINITIONS
---------------------------

A FUNCTION that computes the value of the function F_e(I_1) which
is part of the yield surface function
real function FFe(I1, alpha, gama, beta, theta)
real I1, alpha, gama, beta, theta
FFe = alpha - gama*exp(-beta*I1) + theta*I1
return
dend

A FUNCTION that computes L(kappa)
real function LL(kappa)
real kappa
if (kappa .GT. 0.0) then
   LL = kappa
else
   LL = 0.0
endif
return
dend

A FUNCTION that computes X(k), i.e. intersection of the cap with
the I1-axis
real function XX(kappa, R, alpha, gama, beta, theta)
real kappa, R, alpha, gama, beta, theta, FFe
XX = kappa + R*FFe(kappa, alpha, gama, beta, theta)
return
dend

A FUNCTION that computes the value of the yield function f_yield
real function fyield(I1, J2, kappa, domain, alpha, beta, gama,
& theta, R, T)
real I1, J2, kappa, alpha, gama, beta, theta, R, T, FFe
real XX, LL, XXk, LLk
integer domain
if (domain .EQ. 4) then
   cap surface
   LLk = LL(kappa)
   XXk = XX(kappa, R, alpha, gama, beta, theta)
   fyield = sqrt((I1 - LLk)**2.0 / R**2.0 + J2**2.0) + (LLk - XXk)/R
elseif (domain .EQ. 3) then
   failure surface
   fyield = sqrt(J2) - FFe(I1, alpha, gama, beta, theta)
else
   fyield = I1 - T
endif
endif
return
end

* A FUNCTION to compute the value \( \phi(\sigma, \kappa) \)

real function phi(sigma, kappa, domain, N, f0, alpha, beta, gama, &
theta, R, T)
real sigma(6,1), ss(3,1), kappa, N, f0, alpha, beta, gama, R, prs,&
f, fyield, I1, J2
integer domain

compute I2 and J2
I1 = sigma(1,1) + sigma(2,1) + sigma(3,1)
prs = I1/3.0
ss(1,1) = sigma(1,1) - prs
ss(2,1) = sigma(2,1) - prs
ss(3,1) = sigma(3,1) - prs
J2 = 0.5*(ss(1,1)**2 + ss(2,1)**2 + ss(3,1)**2)
& + sigma(4,1)**2 + sigma(5,1)**2 + sigma(6,1)**2

Pass I2 and J2 to fyield to determine the value of \( f \)
f = fyield(I1, J2, kappa, domain, alpha, beta, gama, theta, R, T)

ensure the value of \( f \) does not fall below 0
if (f .LT. 0.0) f = 0.0

compute the value of \( \phi \)
phi = (f/f0)**N
return
end

* * *

SUBROUTINE SECTION

* * *

subroutine drvalt(I1, J2, ss, kappa, domain, alpha, beta, gama, &
theta, W, D, R, X0, N, f0, dfs, ddfss, ddfs1, &
dphis, dphil, dkl, lamda)

* * *

A SUBROUTINE that computes the necessary derivatives for the
return mapping algorithm.

INCLUDES ALTERNATE DEFINITION OF \( dkappa/\lambda \) AND USES
\( \lambda \) AS AN ADDITIONAL INPUT (FINAL ARGUMENT ABOVE)

* * *

INPUT:
* I1new first stress invariant
* J2new second deviatoric stress invariant
* ss deviatoric stress (6,1)
* kappa hardening parameter
* domain region in stress space, 3=failure region, 4=cap region

* * *
* alpha failure surface parameter
* beta failure surface parameter
* gama failure surface parameter
* theta failure surface parameter
* W cap parameter
* D cap parameter
* R cap parameter
* X0 cap parameter
* hta viscoplastic parameter
* N viscoplastic parameter
* f0 viscoplastic parameter

OUTPUT:
* dfs df_yield/dsigma, (6,1)
* ddfs ddf_yield/ddsigma (6,6)
* dfsl ddf_yield/dsigma.dlamda (1,1)
* dphis dphi/dsigma (6,1)
* dphidl dphi/dlamda (1,1)
* dk1 dkappa/dlamda (1,1)

INTERNAL VARIABLES:
* dfI1: df_yield/dI_1
* ddFeI1 ddf_yield/ddI_1
* dFeI1 dFe/dI1
* dfI2 df_yield/dI_2
* ddfJ2 ddf_yield/dJ2
* dphiiI1 dphi/dI1
* dphiiJ2 dphi/dJ2
* dphif dphi/df
* dI1sig dI1/dsigma (6,1)
* dJ2ss dJ2/dss (6,1)
* dmat Matrix to determine the ddfs (6,6)

FUNCTIONS:
* XX Returns value of XX(kappa) or XX(I1), see functiondefs
* LL Returns value of LL(kappa), see functiondefs

declaration, global variables
real I1, J2, ss(6,1), kappa, alpha, beta, gama, theta, W, D, R, &
  X0, N, f0, dfs(6,1), ddfs(6,6), ddfs1(6,1), &
  dphis(6,1), dphil, dkl, lamda

integer domain

description, internal variables
real dfI1, ddfI1, dfe, dfJ2, ddfJ2, ddfIJ, dphiiI1, dphiiJ2, dphif, &
  dI1sig(6,1), dJ2ssig(6,1), LLk, XXk, dmat(6,6)

functions
real XX, LL

* Compute dII/dsigma
dI1sig(1,1) = 1.0
dI1sig(2,1) = 1.0
dI1sig(3,1) = 1.0
dI1sig(4,1) = 0.0
dI1sig(5,1) = 0.0
dI1sig(6,1) = 0.0
* Compute $dJ_2/d\sigma$

\[
\begin{align*}
\text{dJ}_2\text{sig}(1,1) &= ss(1,1) \\
\text{dJ}_2\text{sig}(2,1) &= ss(2,1) \\
\text{dJ}_2\text{sig}(3,1) &= ss(3,1) \\
\text{dJ}_2\text{sig}(4,1) &= 2*ss(4,1) \\
\text{dJ}_2\text{sig}(5,1) &= 2*ss(5,1) \\
\text{dJ}_2\text{sig}(6,1) &= 2*ss(6,1)
\end{align*}
\]

* compute value of $LL(\kappa)$, $XX(\kappa)$, $XX(I_1)$ and $dF_{ek}$

\[
\begin{align*}
\text{LL}_{k} &= LL(\kappa) \\
\text{XX}_{k} &= XX(\kappa, R, \alpha, \gamma, \beta, \theta) \\
\text{dF}_{ek} &= \gamma \beta \exp(-\beta \kappa) + \theta \quad \text{if (domain .EQ. 3)}
\end{align*}
\]

* if (domain .EQ. 3) then

* -----------------------------------------------
* domain eq. 3, evaluate the failure surface derivatives
* -----------------------------------------------
* Value of $f$

\[
\text{fval} = \sqrt{J_2} - \text{FFE}(I_1, \alpha, \gamma, \beta, \theta)
\]

* df_failure/dI1 and second derivative and df/dI1 check the sign

\[
\begin{align*}
\text{dfI1} &= -\theta - \beta \gamma \exp(-I_1 \beta) \\
\text{ddfI1} &= \beta^2 \gamma \exp(-I_1 \beta)
\end{align*}
\]

* df_failure/dJ2 and second derivative

\[
\begin{align*}
\text{dfJ2} &= 1/(2*J_2^{0.5}) \\
\text{ddfJ2} &= -1/(4*J_2^{1.5})
\end{align*}
\]

* df_failure/dI1dJ2

\[
\text{ddfI1J} = 0.0
\]

* df/dI1dk, df/dJ2, based on $\phi = (f/f0)^N$ (Extra Derivatives)

\[
\begin{align*}
\text{dphiI1} &= (N^*(-\alpha + I_1 \theta - \gamma \exp(-I_1 \beta) - & J_2^{0.5})/f0)^*(N - 1.1)*(\theta + \beta \gamma \exp(-I_1 \beta))/f0 \\
\text{dphiJ2} &= (N^*(-\alpha + I_1 \theta - \gamma \exp(-I_1 \beta) & - J_2^{0.5})/f0)^*(N - 1)/(2*J_2^{0.5})*f0)
\end{align*}
\]

* dphi/df

\[
\begin{align*}
\text{dphif} &= (N*(fval/f0)^*(N - 1))/f0 \\
\text{else} & \quad \text{dphif} = 0
\end{align*}
\]

* dkl = 0.0

* else

* -----------------------------------------------
* domain eq. 4, Cap region derivatives apply

* dkappa/dlamda

\[
\text{dkl} = 0.0
\]

* dphi/dlamda

\[
\text{dphil} = 0.0
\]

* *
* Value of fc
fval = sqrt((I1 - LLk)**2/R**2 + J2) - (XXk - LLk)/R

* df_yield/dI1
dfI1 = ((I1 - LLk)**2 / R**2 + J2)**(-0.5) * (I1 - LLk) / R**2

ddfI1 = 1/(R**2*(J2 + (I1 - LLk)**2/R**2)**(0.5))
& - (2*I1 - 2*LLk)**2/(4*R**4*(J2 + (I1 - LLk)**2/R**2)**(1.5))

* df_yield/dJ2
dfJ2 = 0.5 * ((I1 - LLk)**2 / R**2 + J2)**(-0.5)
ddfJ2 = 0.5 * ((I1 - LLk)**2 / R**2 + J2)**(-1.5)

* df_dkappa
dfk = -((I1-LLk)/(R**2)) * (J2+(I1 - LLk)**2/R**2)**(-0.5)
& - beta*gama*exp(-LLk*beta) - theta

* ddf_yield/dI1dJ2
ddfI1J = -(2*I1 - 2*LLk)/(4*(R**2)*(J2 + (I1 - LLk)**2/R**2)**(1.5))

* df/dI1dK and df/dJ2dK
ddfI1K = (2*I1 - 2*LLk)**2/(4*R**4*(J2 + (I1 - LLk)**2/R**2)**(1.5))
& - 1/(R**2*(J2 + (I1 - LLk)**2/R**2)**(0.5))

ddfJ2K = (2*I1 - 2*LLk)/(4*R**2*(J2 + (I1 - LLk)**2/R**2)**(1.5))

* dkappa/dlamba
dlk = 3 * dfI1 / ( W*D*(1 + R*dFek)*exp(-D * (XXk - X0))
& - (3 * lambda * ddfI1K ))

* dphi/df and dphi/dlamba
if (fval .GT. 0.0) then
  dphif = (N*(fval/f0)**(N - 1.0))/f0
dphi1 = dphif * dfk * dlk
else
  dphif = 0.0
  dphi1 = 0.0
endif

endif

* compute the matrix which help to determine ddfs
  dmat = 0
  dmat(1,1) = 2.0/3.0
  dmat(1,2) = -2.0/6.0
  dmat(1,3) = -2.0/6.0
  dmat(2,1) = -2.0/6.0
  dmat(2,2) = 2.0/3.0
  dmat(2,3) = -2.0/6.0
  dmat(3,1) = -2.0/6.0
  dmat(3,2) = -2.0/6.0
  dmat(3,3) = 2.0/3.0
  dmat(4,4) = 2.0
dmat(5,5) = 2.0
dmat(6,6) = 2.0

* Assemble composite derivatives
dfs = dfl*dl1sig + dfj2*dj2sig
dphis = dphif * dfs
ddfsl = (ddfslk*dl1sig + ddfj2k*dj2sig) * dkl

do 185 i=1,6
do 185 j=1,6
ddfss(i,j) = ddfsl*dl1sig(i,1)*dl1sig(j,1) +
& ddfj*(dj1sig(i,1)*dj2sig(j,1) +
& dl1sig(j,1)*dj2sig(i,1)) +
& dfj2*j2sig(i,1)*dj2sig(j,1) + dfj2*dmat(i,j)
185 continue

return
end

* //////////////////////////////////////////////////////////////////

subroutine matvcm (m, n, A, x, y)
* A SUBROUTINE to compute the matrix-vector product
* \{y\} = [A]{x} with A 'm' rows by 'n' columns
*
integer m, n, i, j
real A(m,n), x(n,1), y(m,1)
*
y(:,:) = 0.0
*
do 25 i = 1,m
  do 25 j = 1,n
  y(i,1) = y(i,1) + A(i,j) * x(j,1)
25 continue
*
return
end

* //////////////////////////////////////////////////////////////////

subroutine matinv(A,N,X)
C Subroutine to invert matrix A(N,N) with the inverse stored
C in X(N,N) in the output.
C
DIMENSION A(N,N),X(N,N),INDX(N),B(N,N)
C
DO 20 I = 1, N  
  DO 10 J = 1, N  
    B(I,J) = 0.0  
10    CONTINUE  
20    CONTINUE  
DO 30 I = 1, N  
  B(I,I) = 1.0  
30    CONTINUE  
C CALL ELGS(A,N,INDX)  
C
DO 100 I = 1, N-1  
  DO 90 J = I+1, N  

DO 80 K = 1, N
   B(INDX(J),K) = B(INDX(J),K) *
         -A(INDX(J),I)*B(INDX(I),K)
80     CONTINUE
90    CONTINUE
100 CONTINUE

C
DO 200 I = 1, N
  X(N,I) = B(INDX(N),I)/A(INDX(N),N)
DO 190 J = N-1, 1, -1
  X(J,I) = B(INDX(J),I)
DO 180 K = J+1, N
  X(J,I) = X(J,I)-A(INDX(J),K)*X(K,I)
180     CONTINUE
190    CONTINUE
200    CONTINUE
C
RETURN
END

SUBROUTINE ELGS(A,N,INDX)
C
Subroutine to perform the partial-pivoting Gaussian elimination.
C
A(N,N) is the original matrix in the input and transformed
C
matrix plus the pivoting element ratios below the diagonal in
C
the output.  INDX(N) records the pivoting order.

C
INTEGER K
DIMENSION A(N,N),INDX(N),C(N)
C
C Initialize the index
C
DO 50 I = 1, N
   INDX(I) = I
50    CONTINUE
K = 0
C
Find the rescaling factors, one from each row
C
DO 100 I = 1, N
   C1= 0.0
   DO 90 J = 1, N
      C1 = AMAX1(C1,ABS(A(I,J)))
90    CONTINUE
   C(I) = C1
100   CONTINUE
C
Search the pivoting (largest) element from each column
C
DO 200 J = 1, N-1
   PI1 = 0.0
   DO 150 I = J, N
      PI = ABS(A(INDX(I),J))/C(INDX(I))
      IF (PI.GT.PI1) THEN
         PI1 = PI
         K = I
      ELSE
         ENDIF
150    CONTINUE

C
Interchange the rows via INDX(N) to record pivoting order

C

ITMP = INDX(J)
INDX(J) = INDX(K)
INDX(K) = ITMP
DO 170 I = J+1, N
   PJ = A(INDX(I),J)/A(INDX(J),J)
C

Record pivoting ratios below the diagonal
C

A(INDX(I),J) = PJ
C

Modify other elements accordingly
C

DO 160 K = J+1, N
   A(INDX(I),K) = A(INDX(I),K) - PJ*A(INDX(J),K)
C

RETURN
C

END

* //////////////////////////////////////////////////////////
* \\\\\\\\\\\\\\\\\\\\\\\\\\\\\
* subroutine CCnlin(CC, sigma, deps, K1, K2, G1, G2)
* A subroutine to compute the tangent nonlinear
* stiffness matrix CCnlin
* INPUT
* sigma stresses (MPa)
* deps strain increment
* K1, K2 bulk mod. intercept (MPa) and bulk mod. exponent
* G1, G2 shear mod. intercept (MPa) and shear mod. exponent
* OUTPUT
* CC Midpoint stiffness matrix estimate
* implicit none

real sigma(6,1), deps(6,1), K1, K2, G1, G2, CC(6,6), I1, J2, prs, &
   K, G, lame, dsig(6,1), sgn(6,1)

CC = 0.0
I1 = sigma(1,1) + sigma(2,1) + sigma(3,1)
prs = I1 / 3
J2 = 0.5 * ( ( sigma(1,1) - prs )**2 + ( sigma(2,1) - prs )**2 &
   + ( sigma(3,1) - prs )**2 ) + sigma(4,1)**2 + sigma(5,1)**2 &
   + sigma(6,1)**2

compute tangent moduli based on I1 and J2
K = K1 + K2*I1
G = G1 + G2*sqrt(J2)
    lame = K - 2.0*G/3.0

Assemble the stiffness matrix from the tangent moduli at state n
CC(1,1) = 2*G + lame
CC(1,2) = lame
CC(1,3) = lame
CC(2,1) = lame
CC(2,2) = 2*G + lame
CC(2,3) = lame
compute stress at state n+1

call matvcm(6, 6, CC, deps, dsig)

* Compute new invariants
  \[ I_1 = \text{sgn}(1,1) + \text{sgn}(2,1) + \text{sgn}(3,1) \]
  \[ \text{prs} = I_1 / 3 \]
  \[ J_2 = 0.5 \times (\text{sgn}(1,1) - \text{prs})^2 + (\text{sgn}(2,1) - \text{prs})^2 \]
  \[ \& + (\text{sgn}(3,1) - \text{prs})^2 + \text{sgn}(4,1)^2 + \text{sgn}(5,1)^2 + \text{sgn}(6,1)^2 \]

* Compute tangent moduli at (n + 1/2)
  \[ K = \frac{K + (K_1 + K_2 I_1)}{2} \]
  \[ G = \frac{G + (G_1 + G_2 \sqrt{J_2})}{2} \]
  \[ \text{lame} = K - 2.0 \times G / 3.0 \]

* Compute CC at (n + 1/2)
  \[ CC(1,1) = 2 \times G + \text{lame} \]
  \[ CC(1,2) = \text{lame} \]
  \[ CC(1,3) = \text{lame} \]
  \[ CC(2,1) = \text{lame} \]
  \[ CC(2,2) = 2 \times G + \text{lame} \]
  \[ CC(2,3) = \text{lame} \]
  \[ CC(3,1) = \text{lame} \]
  \[ CC(3,2) = \text{lame} \]
  \[ CC(3,3) = 2 \times G + \text{lame} \]
  \[ CC(4,4) = G \]
  \[ CC(5,5) = G \]
  \[ CC(6,6) = G \]

end subroutine CCnlin

subroutine ksolve(k, X, R, alpha, beta, gama, theta)
  This subroutine determine the value of kappa (k) based on X value
  by a Newton-Rhapson approximation of the root
real X, R, alpha, beta, gama, theta, crt, dfeK, f, f0
real*4 kn, k
integer itc

  crt = 1e-5 * (alpha-gama)
  itc = 0
  k = 0.0

do 40 itc = 1,60
  f0 = alpha - gama*exp(-beta*k) + theta*k
  dfeK = gama*beta*exp(-beta*k) + theta
  f = k + r*f0 - X
  if (abs(f).lt.crt) goto 60
  kn = k - f/(1.0+r*dfeK)
  k = kn
40 continue
return
end subroutine
B.2 Deviatoric-only constitutive model source code

```fortran
subroutine umat47 (cm,eps,sig,epsp,hsv,dt1,capa,etype,tt,
1 temper,failel,crv,cma,qmat,elsiz,idele)
*
include 'nlqparm'
include 'bk06.inc'
include 'iounits.inc'
dimension crv(lq1,2,*),cma(*),qmat(3,3)
logical failel
character*5 etype

*----------------------------------------
* A SUBROUTINE that computes the strain controlled stresses
* for an nonlinear elastic-viscoplastic material based on the
* Perzyna model. Stress deviator returned.
* FOR USE WITH AN EOS ONLY
* Patrick Kanopoulos and Mohammadamin Jafari, March 13, 2015
* University of Toronto, Department of Civil Engineering
* Update for CC-nonlinear, different formulation in loading and
* unloading
*----------------------------------------
*
* INPUT:
* cm   an array containing the material constants (see below)
* eps  strain increment array (solid mech. sign convention)
* sigma current stress state array
* hsv  material history variables (see below)
* dt   time step
* OUTPUT:
* sigma return stress as a result of strain increment
* hsv  updated history variables
* SUBROUTINES:
* matvcm matrix-vector product
* matinv (small) matrix inversion
* drvalt computes the spatial and material gradients related to
* the failure envelope
* FUNCTIONS
* FFe failure surface component [ f = sqrt(J2) - FFe(I1) ]
* LL   I1 value indicating intersection of cap with failure
*       surface
*fyield computes the value of the yield function
*phi  computes the value of the viscoplastic function phi
*     [ phi = ( f / f0 )^N ]
* DEFINITIONS
* cm(1)  K1d, initial loading bulk modulus (MPa)
* cm(2)  K2d, exponent of loading bulk modulus (---)
* cm(3)  G1d, initial loading shear modulus (GPa)
* cm(4)  G2d, exponent of loading shear modulus (---)
```
cm(5)  K1un, initial unloading bulk modulus (MPa)
cm(6)  K2un, exponent of unloading bulk modulus (-)
cm(7)  G1un, initial unloading shear modulus (GPa)
cm(8)  G2un, exponent of unloading shear modulus (-)
cm(9)  alpha, failure surface parameter (MPa)
cm(10) beta, failure surface parameter (MPa)
cm(11) gamma, failure surface parameter (MPa)
cm(12) theta, failure surface parameter (-)
cm(13) W, cap surface parameter (-)
cm(14) D, cap surface parameter (-)
cm(15) R, cap surface parameter (-)
cm(16) X0, volumetric viscoplastic strain hardening const. (-)
cm(17) T, tension cutoff (MPa)
cm(18) hta, viscoplastic fluidity parameter (inverse sec.)
cm(19) N, visplastic exponent
* cm(20) f0, viscoplastic denominator
* cm(21) htaT, tensile vp fluidity parameter (inv. sec.)
cm(22) maximum number of iterations (convert to INTEGER)
cm(23) return mapping tolerance (-)

hsv(1)  kappa, hardening parameter (-)

Initialize variable names and dimensions

implicit none
IO
real cm(23), eps(6), sig(6), hsv(1), dt1

real sigma(6,1), sigtri(6,1),
& dsigtri(6,1), sstri(6,1), I1tri, J2tri, prstri, signew(6,1),
& I1new, J2new, ssnew(6,1), rj2tri, prsnew, Jratio, prsfin

real deps(6,1), depsv, epsvp, epsvp0, depsvp

parameters
real K1d, K2d, G1d, G2d, K1un, K2un, G1un, G2un, alpha, beta,
& gamma, theta, W, D, R, T, hta, N, lame, f0,
& CC(6,6), CCsup(6,6), CCinv(6,6), X0, htaT

real K1, K2, G1, G2

hardening parameters
real kappa0, kappaN, dkappa, Xitr

time domain parameters
real dt

Domain geometry variables
real LLk, FFeTT, ftri
integer domain
integer vpyield

return mapping variables
real lamdaN, rhoN,
& conv, HH(6,6), HHinv(6,6), x1, xi1(6,1),
& xi2(6,1), xi3(1,1), xi4, xxPL(6,1), yyPL(6,1)
derivatives
real dfs(6,1), ddffs(6,6), ddfls(6,1), dphis(6,1), dphil, dkl, &
   tdphis(1,6)

function calls
real LL, FFe, fyield, phi, XX

counting variables and iteration variables
integer i, ii, mxiter

other variables
real phiN, lamohpt

VARIABLE ASSIGNMENT

assign the stresses, sigma // convert to g.mech convention
sigma(1,1) = -sig(6)
sigma(2,1) = -sig(6)
sigma(3,1) = -sig(6)
sigma(4,1) = -sig(6)
sigma(5,1) = -sig(6)
sigma(6,1) = -sig(6)

assign the strain increment, deps // convert to g.mech convention
deps(1,1) = -eps(1)
deps(2,1) = -eps(2)
deps(3,1) = -eps(3)
deps(4,1) = -eps(4)
deps(5,1) = -eps(5)
deps(6,1) = -eps(6)

Assign elastic constants
K1d = cm(1)
K2d = cm(2)
G1d = cm(3)
G2d = cm(4)
K1un = cm(5)
K2un = cm(6)
G1un = cm(7)
G2un = cm(8)

assign plastic constants
alpha = cm(9)
beta = cm(10)
gama = cm(11)
theta = cm(12)
W = cm(13)
D = cm(14)
R = cm(15)
X0 = cm(16)
T = cm(17)

assign viscosity constants
hta = cm(18)
N = cm(19)
f0 = cm(20)
htat = cm(21)
* Set default maximum iteration values
  mxiter = int(cm(22))
* Set the convergence criterion
  conv = cm(23)
* assign initial value of hardening parameter
  kappa0 = hsv(1)
  kappaN = kappa0
* assign timestep
  dt = dt1
*
* MAIN PROGRAM
*
* set elastic constants based on loading or unloading
  depsv = deps(1,1) + deps(2,1) + deps(3,1)
  if (depsv .GE. 0.0) then
    K1 = K1d
    K2 = K2d
    G1 = G1d
    G2 = G2d
  else
    K1 = K1un
    K2 = K2un
    G1 = G1un
    G2 = G2un
  endif
* assemble the elastic stiffness tensor
  CC = 0
  call CCnlin( CC, sigma, deps, K1, K2, G1, G2 )
  CCsup = CC
* compute the inverse stiffness tensor CCinv calling matinv
  call matinv(CCsup,6,CCinv)
* compute the elastic estimator stress, sigma^est_i
  dsigtr = CC * depsilon
  call matvcm(6,6,CCdeps,dsigtr)
  sigtri = sigma + dsigtr
* determine I1trial
  I1tri = sigtri(1,1) + sigtri(2,1) + sigtri(3,1)
* determine J2trial (from pressure and sstrial)
  prstri = I1tri/3.0
  sstrix = sigtri
  sstrix(1,1) = sigtri(1,1) - prstri
  sstrix(2,1) = sigtri(2,1) - prstri
  sstrix(3,1) = sigtri(3,1) - prstri
  J2tri = 0.5*(sstrix(1,1)**2 + sstrix(2,1)**2 + sstrix(3,1)**2) + &
        sstrix(4,1)**2 + sstrix(5,1)**2 + sstrix(6,1)**2
*
* Determine case (i.e. Tension lower quadrant, Tension higher quadrant, Yield, Cap)
* Tension: domain 1 I1trial <= -T, and, root(J2trial) < FFe(-T)
* Tension: domain 2 I1trial <= -T, and, root(J2trial) >= FFe(-T)
* Yield: domain 3 -T <= I1trial <= L(k)
Cap: domain 4  I1trial > L(k)

\[ LL_k = LL(\kappa_0) \]
\[ FFeT = FFe(T, \alpha, \gamma, \beta, \theta) \]
\[ rJ2tri = \sqrt{J2tri} \]

if (I1tri .GT. LLk) then
domain = 4
elseif ((I1tri .LE. LLk) .AND. (I1tri .GT. T)) then
domain = 3
elseif ((I1tri .LE. T) .AND. (rJ2tri .GT. FFeT)) then
domain = 2
elseif ((I1tri .LE. T) .AND. (rJ2tri .LT. FFeT)) then
domain = 1
endif

Compute the value of fyield
\[ ftri = fyield(I1tri, J2tri, \kappa_0, \alpha, \beta, \gamma, \theta, R, T) \]

Set logical yield variable (vpyield) to 1 or 0
if (ftri .GT. 0.) then
vpyield = 1
else
vpyield = 0
endif

MAIN if STATEMENT -- ELASTIC, RET. MAPPING, OR TENSION C.O.

If elastic step - return elastic estimator
If viscoplastic step - do return mapping
If tension step - return tension cutoff stresses

if ((domain .EQ. 4 .OR. domain .EQ. 3) .AND. vpyield .EQ. 0) then
output stress is simply the elastic trial stress,
hardening parameter (kappa) remains unchanged.
\[ \text{signew} = \text{sigtri} \]
\[ \kappaappaN = \kappaappa0 \]

elseif ((domain .EQ. 4 .OR. domain .EQ. 3) 
& .AND. vpyield .EQ. 1) then
viscoplastic step! Do return mapping algorithm
initialize initial variables (\lambda, \sigma, \rho)
\[ \text{lambdAN} = 0.0 \]
\[ \text{signew} = \text{sigtri} \]
\[ \text{rhON} = \phi(\text{signew}, \kappaappa0, \text{domain}, \text{N}, \theta, \alpha, \beta, \gamma, \theta, R, T) \]
& - \text{lambdAN} / (hta*dt)
compute initial volumetric \(\nu\) strain, and initialize total and
incremental strains
\[ \text{epsvp\theta} = W * (1 - \exp(-D*) \]
& (XX(\kappaappa0, R, \alpha, \gamma, \beta, \theta) - \theta))
\[ \text{depsvp} = 0.0 \]
epsvp = 0.0

* loop through iterative improvements of signew and kappa
do 100 ii = 1, mxiter
* find stress invariants for current stress state
    I1new = signew(1,1) + signew(2,1) + signew(3,1)
    prsnew = I1new/3.0
    ssnew = signew
    ssnew(1,1) = signew(1,1) - prsnew
    ssnew(2,1) = signew(2,1) - prsnew
    ssnew(3,1) = signew(3,1) - prsnew
    J2new = 0.5*(ssnew(1,1)**2 + ssnew(2,1)**2 + ssnew(3,1)**2) +
         & signew(4,1)**2 + signew(5,1)**2 + signew(6,1)**2
* compute stiffness tensor
    call CCnlin( CC, signew, deps, K1, K2, G1, G2 )
    CCsup = CC
    call matinv(CCsup,6,CCinv)
* call derivative subroutine -- returns necessary derivatives of cap, failure, and phi functions
    call drvlnt(I1new,J2new,ssnew,kappaN,domain,alpha,beta,
         & gama,theta, W, D, R, X0, N, f0, dfs, ddfss, ddfs1, &
         dphis, dphil, dkl, lamdaN )
* compute the HH vector // calling function minv (matrix inverse)
    HHinv = CCinv + lamdaN * ddfss
* invert HHinv matrix to find HH, calling matinv
    call matinv(HHinv,6,HH)
* take the transpose of dphis
    do 235 i=1,6
        Tdphis(1,i)=dphis(i,1)
    continue
* compute xi calling the matrix-vector multiplication subroutine
    xi = Tdphis * HH * (dfs+lamdaN*ddfs1) + 1/(hta*dt) - dphi
    xi1 = dfs + lamdaN*ddfs1
    call matvcm(6,6,HH,xi1,xi2)
    call matvcm(1,6,Tdphis,xi2,xi3)
    xi4 = xi3(1,1)
    xi = xi4 + ( 1.0 / (hta*dt) ) - dphil
* compute lamdaN
    lamdaN = lamdaN + rhoN/xi
* compute {signew} = {signew} + [CC]*{depsilon - lamdaN*dfs}
    xxPL = deps - lamdaN * dfs
    call matvcm(6, 6, CC, xxPL, yyPL)
    signew = sigma + yyPL
* Compute kappaN from dkappa = delta_lamda_N * dkappa/dlamda
depsvp = lamdaN*( dfs(1,1) + dfs(2,1) + dfs(3,1) )
if ( depsvp .LT. 0.0 ) depsvp = 0.0
epsvp = epsvp0 + depsvp
if ( epsvp .GE. (0.99*W) ) then
    epsvp = 0.99*W
    signew = sigtri
    kappaN = kappa0
    goto 300
endif
Xitr = X0 - Log( 1 - epsvp/W ) / D
    call ksolve(kappaN, Xitr, R, alpha, beta, gama, theta)
* Check the convergence by computing rho_N
    phiN = phi(signew,kappaN,domain,N,f0,alpaha,beta,gama,theta,R,T)
compute \( \rho \) for current iteration
\[ \text{lamoht} = \text{lamdaN} / (\text{hta} \times \text{dt}) \]
\[ \text{rhoN} = \phiN - \text{lamoht} \]

Check for the convergence condition
\[ \text{if} \ (\text{abs(rhoN)} \text{.LE. abs(conv)}) \text{ then} \]
goto 300
\[ \text{endif} \]
\[ \text{end loop} \]

Continue

if (\( \text{abs(rhoN)} \text{.GT. abs(conv)} \)) then
write(*,*) 'Not Converged, ii = ', ii, 'rho_N = ', rhoN
\[ \text{endif} \]

*---------------------------------------------------------------*
elseif (domain .EQ. 2 .OR. domain .EQ. 1) then
* Tension cutoff stresses obtained - return tensile stresses from
* the tension formulas
* if (domain .EQ. 1) then
* compute \( I_1_{\text{new}}, \text{pressure}_{\text{new}}, J_2_{\text{new}}, \) and 'Jratio' for domain 1
\[ \text{I1new} = \exp(-\text{htaT}*\text{dt}) \times \text{I1tri} + (1-\exp(-\text{htaT}*\text{dt})) \times (-T) \]
\[ \text{prsnew} = \text{I1new}/3.0 \]
\[ \text{J2new} = \text{J2tri} \]
\[ \text{Jratio} = 1.0 \]
else
* compute \( I_1_{\text{new}}, \text{pressure}_{\text{new}}, J_2_{\text{new}}, \) and 'Jratio' for domain 2
\[ \text{I1new} = \exp(-\text{htaT}*\text{dt}) \times \text{I1tri} + (1-\exp(-\text{htaT}*\text{dt})) \times (-T) \]
\[ \text{prsnew} = -\text{I1new}/3.0 \]
\[ \text{J2new} = (\exp(-\text{htaT}*\text{dt}) \times \sqrt{\text{J2tri}} + (1-\exp(-\text{htaT}*\text{dt})) \times \text{FFeTT} )**2 \]
\[ \text{Jratio} = \sqrt{\text{J2new}}/\sqrt{\text{J2tri}} \]
endif

* Update the stresses for the new tensile regime
\[ \text{signew}(1,1) = \text{sstri}(1,1) \times \text{Jratio} + \text{prsnew} \]
\[ \text{signew}(2,1) = \text{sstri}(2,1) \times \text{Jratio} + \text{prsnew} \]
\[ \text{signew}(3,1) = \text{sstri}(3,1) \times \text{Jratio} + \text{prsnew} \]
\[ \text{signew}(4,1) = \text{sstri}(4,1) \times \text{Jratio} \]
\[ \text{signew}(5,1) = \text{sstri}(5,1) \times \text{Jratio} \]
\[ \text{signew}(6,1) = \text{sstri}(6,1) \times \text{Jratio} \]
\[ \text{endif} \]

300 continue
* Decompose stresses and return only deviatoric portion
\[ \text{prsfin} = (\text{signew}(1,1) + \text{signew}(2,1) + \text{signew}(3,1)) / 3.0 \]
\[ \text{signew}(1,1) = \text{signew}(1,1) - \text{prsfin} \]
\[ \text{signew}(2,1) = \text{signew}(2,1) - \text{prsfin} \]
\[ \text{signew}(3,1) = \text{signew}(3,1) - \text{prsfin} \]

*---------------------------------------------------------------*
*---------------------------------------------------------------*
* STRESS OUTPUT
*---------------------------------------------------------------*
* update output variables sig // convert to solid mech. convention
\[ \text{sig}(1) = -\text{signew}(1,1) \]
\[ \text{sig}(2) = -\text{signew}(2,1) \]
\[ \text{sig}(3) = -\text{signew}(3,1) \]
\[ \text{sig}(4) = -\text{signew}(4,1) \]
\[ \text{sig}(5) = -\text{signew}(5,1) \]
\texttt{sig}(6) = \texttt{-signew(6,1)}

* update history variables
  \texttt{hsv} = \texttt{kappaN}

* \texttt{return}

\texttt{end}
Appendix C: User defined equation of state model source code

```fortran
subroutine ueos23s(iflag, cb, pnew, hist, rho0, eosp, specen,
    & df, dvol, v0, pc, dt, tt, crv, first)

* include 'nlqparm'
* include 'iounits.inc'
*
Scalar implementation of user defined EOS
This EOS is based on the 3-Phase model proposed by An, Tuan,
Cheeseman, and Gazonas (2011)
*
iflag ---- = 0 calculate bulk modulus
= 1 update pressure and energy
cb -------- bulk modulus
pnew ------ new pressure
hist ------ history variables
rho0 ------ reference variables
eosp ------ EOS constants
specen ---- energy
df --------- volume ratio, v/v0 = rho0/rho
dvol ------ change in volume over time step
v0 --------- reference volume
pc -------- pressure cutoff
dt -------- time step size
tt -------- current time
crv ------- curve array
first ------ logical .true. for tt,crv first time step (for
            initialization of history variables)
*
NOTE: soil refers to the entirety of solid + water + air
whereas solid refers to the solid component of the soil
*
EOSP DEFINITIONS
eosp(1) = As0 ------ Initial Volume Fraction of the Solid
eosp(2) = Aw0 ------ Initial Volume Fraction of the Water
eosp(3) = Aa0 ------ Initial Volume Fraction of the Air
eosp(4) = rhoS0 --- Initial density of the solid
eosp(5) = rhoW0 --- Initial density of the water
eosp(6) = rhoA0 --- Initial density of the air
eosp(7) = kS ------- entropy exponent of the solid
eosp(8) = kW ------- entropy exponent of the water
eosp(9) = kA ------- entropy exponent of the air
*
eosp(10) = C ------ Wavespeed of the soil
eosp(11) = s1 ------ Slope of Hugoniot for the soil
eosp(12) = gamma0 --- Initial gamma value for the soil
eosp(13) = alpha --- First order correction factor for soil
*
eosp(14) = Cs ------ Wavespeed of the solid
eosp(15) = s1s ------ Slope of the solid phase Hugoniot
eosp(16) = gamma0s -- Initial gama value of the solid
eosp(17) = alphaS -- First order correction factor for the solid
*
eosp(18) = Cw ------ Wavespeed of the water
eosp(19) = s1w ------ slope of the water phase Hugoniot
eosp(20) = gamma0w -- Initial gama value of the water
eosp(21) = alphaW -- First order correction factor for the water
*
eosp(22) = Ca ------ Wavespeed of the air
```
* eosp(23) = s1a ----- Slope of the air phase Hugoniot
* eosp(24) = gama0a -- Initial gama value of the air
* eosp(25) = alphaA -- First order correction factor for the air
* eosp(26) = p0 ------ Atmospheric pressure
* eosp(27) = Kcut ---- Bulk modulus cutoff for excessive pressures

HIST DEFINITIONS
* hist(1) = As
* hist(2) = Aw
* hist(3) = Aa
* hist(4) = dVs
* hist(5) = dVw
* hist(6) = dVa

Internal Variables
* Rs: Weight Fraction of the Solid
* Rw: Weight Fraction of the Water
* Ra: Weight Fraction of the Air
* Rho0: Initial Density of the Soil
* Pres: Pressure in the Soil
* SpEng: Internal Energy
* Vrat: Volume Ratio (df)

implicit none

* real cb, pnew, hist(6), rho0, eosp(*), specen, df, dvol, v0, pc,
   & dt, tt, crv(lq1,2,*)
* integer iflag
* logical first
* real As0, Aw0, Aa0, rhoS0, rhoW0, rhoA0, kS, kW, kA, C, s1,
   & gama0, Cs, alpha, s1s, gama0s, alphaS, Cw, s1w, gama0w
* real alphaW, Ca, s1a, gama0a, alphaA, p0, Kcut

* real V, As, Aw, Aa, dVs, dVw, dVa, RRs, RRw, RRa, sEngS, sEngW,
   & sEngA, ffs, ffw, ffa, mio, mios, miow, mioa
* real Ps, Pw, Pa, VSold, VWold, VAold
* Functions
* real Pres, BulkM

* MAIN CODE

* Assign EOS parameters
  As0   = eosp(1)
  Aw0   = eosp(2)
  Aa0   = eosp(3)
  rhoS0 = eosp(4)
  rhoW0 = eosp(5)
  rhoA0 = eosp(6)
  kS    = eosp(7)
  kW    = eosp(8)
  kA    = eosp(9)
  C     = eosp(10)
s1 = eosp(11)
gama0 = eosp(12)
alpha = eosp(13)
Cs = eosp(14)
s1s = eosp(15)
gama0s = eosp(16)
alphaS = eosp(17)
Cw = eosp(18)
s1w = eosp(19)
gama0w = eosp(20)
alphaW = eosp(21)
Ca = eosp(22)
s1a = eosp(23)
gama0a = eosp(24)
alphaA = eosp(25)
p0 = eosp(26)
Kcut = eosp(27)

* * Assign history parameters in case first = .true. (i.e. first 0th time step)
  if (first) then
    hist(1) = As0
    hist(2) = Aw0
    hist(3) = Aa0
    hist(4) = 0.0
    hist(5) = 0.0
    hist(6) = 0.0
    write (*,*) 'doing first subroutine'
  endif

* * Assign History variables to appropriate variables
  As = hist(1)
  Aw = hist(2)
  Aa = hist(3)
  dVs = hist(4)
  dVw = hist(5)
  dVa = hist(6)

* *
* Compute the weight ratios of each phase
  RRs = As * rhoS0 / rho0
  RRw = Aw * rhoW0 / rho0
  RRa = Aa * rhoA0 / rho0

* * Solid, Water & Air SpeEn (SpeEng is a history variable)
  sEngS = specen * RRs
  sEngW = specen * RRw
  sEngA = specen * RRa

* Update mio which is equal V0/V-1 (ff=V/V0)
  ffs = df*As/As0
  ffw = df*Aw/Aw0
  ffa = df*Aa/Aa0

  mio = (1/df) - 1
  mios = (1/ffs) - 1
  mioW = (1/ffw) - 1
  mioA = (1/ffa) - 1

* * Compute the original volumes of Solid, Air & Water (before step)
  Vsold = (df * v0 - 2.0 * dvol) * As
Vwold = (df * v0 - 2.0 * dvol) * Aw
Vaold = (df * v0 - 2.0 * dvol) * Aa

* if (iflag .EQ. 0) then
    * Bulk modulus calculation (cb)
      cb = BulkM (Rho0, C, gamma0, mio, alpha, s1, specen)
      * Check against cutoff bulk modulus
      if (cb .GT. Kcut) cb = Kcut
    else
      * Calculating the pressure for SOILD, WATER AND AIR
        Ps = Pres (rhos0, Cs, Gama0s, mios, alphaS, s1s, sEngs)
        Pw = Pres (rhoW0, Cw, Gamma0w, miow, alphaW, s1w, sEngw)
        Pa = Pres (rhoa0, Ca, Gamma0a, mioA, alphaA, s1a, sEnga)
      * Ensure newly computed pressures are above the cutoff pressure
        Ps = max(Ps,pc)
        Pw = max(Pw,pc)
        Pa = max(Pa,pc)
      * Calculating the internal Energy for SOILD, WATER AND AIR
        sEngs = sEngs - 0.5*Ps*dVs/(As0*V0)
        sEngw = sEngw - 0.5*Pw*dVw/(Aw0*V0)
        sEnga = sEnga - 0.5*Pa*dVa/(Aa0*V0)
      * specen = sEngs + sEngw + sEnga
      * Calculating the total pressure and SpecEng for soil
        if (dvol /= 0.0) then
            pnew = ( Ps*dVs + Pw*dVw + Pa*dVa ) / dvol
        else
            pnew = Ps + Pw + Pa
        endif
      * Updating (Calculating) the volumetric changes and fractions
        (History Variables)
        As = As*(Ks*(Ps-P0)/(RhoS0*Cs**2)+1)**(-ks**(-1))
        Aw = Aw*(Kw*(Pw-P0)/(RhoW0*Cw**2)+1)**(-kw**(-1))
        Aa = Aa*(Pa/P0)**(-ka**(-1))
      * dVs = df*V0*As-Vsold
      * dVw = df*V0*Aw-Vwold
      * dVa = df*V0*Aa-Vaold
    * Assign History Variables
      hist(1) = As
      hist(2) = Aw
      hist(3) = Aa
      hist(4) = dVs
      hist(5) = dVw
      hist(6) = dVa
      * endif
    * return
  endif
  * This function calculate the bulk modulus in each time step
real function BulkM (Rho, C, Gam0, mio, alph, s1, SpEng)
  * Real Rho, C, Gam0, mio, alph, s1, SpEng, AAA, BBB, CCC, DDD, &
  * EEE

AAA = 1+(gam0/2)*mio-alph/2*mio**2
BBB = 1+mio-s1*mio
CCC = 1+2*mio*(s1-1)/BBB*mio*(gam0+alph*mio)/(1+mio)**2
DDD = mio*(1-gam0/2-alph*mio)
EEE = (gam0+alph*mio)**2 / ((1+mio)**2) + alph

BulkM = ( Rho*C**2*A**CCC + Rho*C**2*DDD ) / BBB**2 + EEE*SpEng

return
end

This function calculate the Pressure in each time step
( an Eq. 3.9)
real function Pres (Rho, C, Gam0, mio, alph, s1, SpEng)

Real Rho, C, Gam0, mio, alph, s1, SpEng, A, B

A = 1 + (gam0/2)*mio - alph/2*mio**2
B = ( 1 + mio - s1*mio )**2
Pres= Rho*C**2*mio*A/B+(gam0+alph*mio)*SpEng

return
end
Appendix D: SHPB Experimental Results

D.1 Dry coarse sand – SHPB Complete Results

![Graph showing strain vs compressive stress for two strain rates: 200 and 300 1/s.](image)
Strain-rate~400 1/s
- Exp. No.1
- Exp. No.2
- Exp. No.9
- Exp. No.10
- Exp. No.11
- Average

Strain-rate~550 1/s
- Exp. No.7
- Exp. No.8
- Exp. No.9
- Exp. No.10
- Exp. No.12
- Average
Test Averages

Compressive stress (MPa)

<table>
<thead>
<tr>
<th>Strain rate 200 1/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strain rate 300 1/s</td>
</tr>
<tr>
<td>Strain rate 400 1/s</td>
</tr>
<tr>
<td>Strain rate 550 1/s</td>
</tr>
<tr>
<td>Strain rate 830 1/s</td>
</tr>
<tr>
<td>Strain rate 980 1/s</td>
</tr>
<tr>
<td>Strain rate 1100 1/s</td>
</tr>
<tr>
<td>Strain rate 1700 1/s</td>
</tr>
<tr>
<td>Strain rate 2100 1/s</td>
</tr>
</tbody>
</table>

Strain

-0.02 0.00 0.02 0.04 0.06 0.08 0.10 0.12 0.14 0.16 0.18 0.20 0.22
D.2 Dry Fine Sand – SHPB Complete Results

Strain-rate~160 1/s

- Exp. No.3
- Exp. No.4
- Exp. No.7
- Exp. No.9
- Exp. No.11
- Average

Strain-rate~270 1/s

- Exp. No.2
- Exp. No.3
- Exp. No.4
- Exp. No.5
- Exp. No.6
- Average
Strain-rate ~410 1/s

- Exp. No.1
- Exp. No.3
- Exp. No.4
- Exp. No.5
- Exp. No.7
- Average

Strain-rate ~720 1/s

- Exp. No.1
- Exp. No.2
- Exp. No.3
- Exp. No.5
- Exp. No.9
- Average
Strain-rate ~820 1/s

- Exp. No. 2
- Exp. No. 4
- Exp. No. 5
- Exp. No. 9
- Exp. No. 10
- Average

Strain-rate ~970 1/s

- Exp. No. 1
- Exp. No. 2
- Exp. No. 3
- Exp. No. 5
- Exp. No. 7
- Average
Test Averages

Strain-rate~1250 1/s
- Exp. No.1
- Exp. No.2
- Exp. No.6
- Exp. No.8
- Exp. No.10
- Average

Compressive stress (MPa)

Strain rate 160 1/s
- Black

Strain rate 270 1/s
- Red

Strain rate 410 1/s
- Green

Strain rate 720 1/s
- Blue

Strain rate 820 1/s
- Cyan

Strain rate 970 1/s
- Magenta

Strain rate 1250 1/s
- Orange

Strain rate 1500 1/s
- Brown

Strain rate 2100 1/s
- Blue

Compressive stress (MPa)

Strain
D.3 Saturated coarse sand – SHPB Complete results

[Graph showing stress-strain curves for two different strain rates: 160 1/s and 220 1/s. Each curve represents the results of an experiment labeled with experiment number, and a red line represents the average of all experiments.]
Strain-rate ~490 \, 1/s

- Exp. No.1
- Exp. No.2
- Exp. No.3
- Exp. No.7
- Exp. No.8
- Average

Strain-rate ~660 \, 1/s

- Exp. No.6
- Exp. No.7
- Exp. No.9
- Exp. No.10
- Exp. No.11
- Average
Test Averages

- Black line: Strain rate 160 1/s
- Red line: Strain rate 220 1/s
- Green line: Strain rate 330 1/s
- Blue line: Strain rate 380 1/s
- Cyan line: Strain rate 490 1/s
- Pink line: Strain rate 660 1/s

Graph shows the relationship between compressive stress (MPa) and strain.
D.4 Saturated Fine Sand – SHPB Complete Results

Strain-rate ~140 1/s
- Exp. No.1
- Exp. No.2
- Exp. No.9
- Exp. No.11
- Exp. No.12
- Average

Exp. No.1
Exp. No.2
Exp. No.9
Exp. No.11
Exp. No.12
Average

Strain-rate ~260 1/s
- Exp. No.1
- Exp. No.3
- Exp. No.4
- Exp. No.8
- Exp. No.10
The graphs depict the compressive stress-strain behavior under different strain rates.

**Strain-rate ~500 1/s**
- **Exp. No.2**
- **Exp. No.6**
- **Exp. No.9**
- **Exp. No.10**
- **Exp. No.12**
- Average

**Strain-rate ~800 1/s**
- **Exp. No.1**
- **Exp. No.3**
- **Exp. No.4**
- **Exp. No.7**
- **Exp. No.10**
- Average
Test Averages

Compressive stress (MPa) vs. Strain

- Strain rate 140 1/s
- Strain rate 260 1/s
- Strain rate 300 1/s
- Strain rate 450 1/s
- Strain rate 500 1/s
- Strain rate 800 1/s
Appendix E: EOS Parameter Estimation

In this appendix, the standard methodology for determining the material Hugoniot is presented. The results provided by Chapman et al. (2006) are presented.

E.1 Plate Impact Experimental Methodology

Almost universally, the principal Hugoniot of highly porous granular materials has been developed from direct measurement of shock-velocity. A plate-impact reverberation technique is described here. Sensors are embedded in anvils surrounding a cavity containing the sample. The method has the disadvantage that no in-material stress data is obtained directly and must be inferred using the jump condition (eq. E1).

\[
\sigma_x = \rho_0 U_s u_p
\]  

(E1)

where \(\sigma_x\) is the longitudinal stress, \(\rho_0\) the initial density, \(U_s\) the shock-velocity, and \(u_p\) the particle velocity. The stress values obtained should be treated with caution as the method represents an approximation, neglecting material strength and assuming a steady state.

The Hugoniot data was determined through a plate-impact experiment technique. Plate-impact experiments were conducted using the 50 mm bore 5 m length single stage light gas gun. Impact velocities were measured to an accuracy of 0.5% using a sequential pin-shorting method and the target was aligned with the impactor (flyer) to less than 1 mrad by means of an adjustable specimen mount. Longitudinal stress was measured during impact using commercially produced manganin piezoresistive gauges. The output voltage was recorded on a fast (5 GS s\(^{-1}\)) digital storage oscilloscope. This voltage time data was then reduced to stress histories.

The plate impact configuration is shown in Figure E1. Projectiles were made from a lapped aluminum discs (called as ‘flyer’), and affixed to a polycarbonate sabot. A complete release was achieved by placing the rear of the flyer next to recess. The target structure is also shown in Figure E1. The sand is contained laterally by an aluminum annulus and longitudinally by anvils. These annuli and anvils were made from aluminum, and in each experiment, the flyer plate material is the same as that of the anvils and annulus in the target. Gauges were incorporated into the samples using a slow curing epoxy. In the front driving anvil, a gauge, G1, was glued at 1-2mm from the impact face and 1-2mm from the 3.2mm sand cavity. To the rear of the cavity a further gauge G2 was bonded 1mm into the back anvil. The thickness of gauge package was typically \(\sim\)100\(\mu\)m, and the rise-time was \(\sim\)30ns as only the manganin element was impedance mismatched. This resolution can detect the evidence of any precursor wave in the compaction process. The aluminum anvils had similar impedance to the sand providing a favourable environment where gauge hysteresis was kept to a minimum.
E.2 Hugoniot determination

Hugoniot measurements employed the impedance match method and an idealized gauge response and simplified X-T diagram is shown in Figure 5.4.2 and Figure 5.4.3.

The symmetric impact of the front anvil and flyer results in state A. Shocks move both forward into the front anvil and backwards into the flyer. The dimensions of the projectile were chosen so that longitudinal release from the rear of the projectile would not occur during the time of interest. The shock traverses the front anvil passing through the location of G1 where state A is recorded and is incident on the anvil sand cell boundary. The anvil will either be either reloaded or released depending upon the relative impedance of the anvil to that of the sand sample, resulting state B. Because the Al anvils were used in the experiment and the impedance of Al is higher than that of the sand, the released case is achieved in the experiment. The released case is represented in Figure 5.4.2 where state B is the intersection of the release isentrope of the state A and Hugoniot of the sand. For the purpose if the data analysis the release isentrope of the anvil was approximated by the Hugoniot. A release fan representative of state B passes back into the front anvil and is registered at G1. The shock/compaction front moves through the sand sample, eventually incident on the rear anvil. Again, depending upon the relative impedance of the anvil material to the sand, the sand is reloaded by the rear anvil, which is demonstrated in Figure 5.4.3 and is labelled state C. A shock moving into the rear anvil is measured by G2 and be used to ascertain off-Hugoniot points of the sand after the initial shock. The time of arrival at the gauges can be used to calculate a transit time for the shock through the sand if the shock velocity in the anvils is known. It can be shown from conservation of momentum that the line passing through points O and B has slope $\rho_0 U_s$. Consequently, the particle-velocity of state B can be determined from the intersection of the line with gradient $\rho_0 U_s$ and the anvil Hugoniot (release isentrope). The sand Hugoniot can be constructed by repeating this procedure at different impact velocities.
Figure E2: Impedance match method for Hugoniot measurement.

Figure E3: Idealized Gauge response and simplified X-T diagram using anvils of higher impedance than the sand.

The above analysis disregards any material strength, treating only a single shock moving through the sand bed. In practice, the stress profile will be complex, with compaction processes occurring and possibly low amplitude precursors moving through the skeletal bed of quartz. These precursors will reverberate between the boundaries and act upon the main shock front. However, these perturbations have a minor influence on the main transit time of the shock front in comparison to the general experimental uncertainty. The shock-velocities are measured as an average of the transit time through the entire sample. If a steady state is not achieved in a short
time compared with the transit time (certainly possible considering the granular bed compaction) this will add additional uncertainty in the measured shock-velocity. It has been demonstrated that the time taken to achieve a steady state is short compared with the transit time.

E.3 Results and discussions

Table E1 contains the details of the experiments performed by Chapman et al. (2006). The shock-velocities were calculated using the transit time measured between half-stress points on the rising stress profiles. This avoids complications associated with the ramping nature of the stress profiles in G2 and minimizes uncertainty. Particle velocities were obtained using the impedance matching method discussed above, whereas the longitudinal stress was inferred using the simple jump condition (eq. E1).

<table>
<thead>
<tr>
<th>Shot</th>
<th>Saturation type</th>
<th>Sand type</th>
<th>Anvil Material</th>
<th>Impact velocity (km/s)</th>
<th>Shock Velocity (km/s)</th>
<th>Particle Velocity (km/s)</th>
<th>Longitudinal Stress (GPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dry</td>
<td>OS1</td>
<td>Al</td>
<td>0.499</td>
<td>1.08</td>
<td>0.34</td>
<td>0.53</td>
</tr>
<tr>
<td>2</td>
<td>Dry</td>
<td>OS1</td>
<td>Al</td>
<td>0.919</td>
<td>1.56</td>
<td>0.58</td>
<td>1.28</td>
</tr>
<tr>
<td>3</td>
<td>Dry</td>
<td>OS1</td>
<td>Al</td>
<td>0.915</td>
<td>1.33</td>
<td>0.60</td>
<td>1.14</td>
</tr>
<tr>
<td>4</td>
<td>Dry</td>
<td>OS1</td>
<td>Al</td>
<td>0.915</td>
<td>1.56</td>
<td>0.57</td>
<td>1.28</td>
</tr>
<tr>
<td>5</td>
<td>Dry</td>
<td>OS1</td>
<td>Al</td>
<td>0.921</td>
<td>1.54</td>
<td>0.58</td>
<td>1.28</td>
</tr>
<tr>
<td>6</td>
<td>Dry</td>
<td>OS1</td>
<td>Al</td>
<td>0.494</td>
<td>1.20</td>
<td>0.47</td>
<td>0.81</td>
</tr>
<tr>
<td>7</td>
<td>Dry</td>
<td>OS1</td>
<td>Al</td>
<td>0.851</td>
<td>1.80</td>
<td>0.79</td>
<td>2.05</td>
</tr>
<tr>
<td>8</td>
<td>Full saturation</td>
<td>OS1</td>
<td>Al</td>
<td>0.499</td>
<td>1.74</td>
<td>0.26</td>
<td>0.84</td>
</tr>
<tr>
<td>9</td>
<td>Full saturation</td>
<td>OS1</td>
<td>Al</td>
<td>0.955</td>
<td>2.29</td>
<td>0.46</td>
<td>1.96</td>
</tr>
<tr>
<td>10</td>
<td>Full saturation</td>
<td>OS1</td>
<td>Al</td>
<td>0.502</td>
<td>2.57</td>
<td>0.44</td>
<td>2.01</td>
</tr>
<tr>
<td>11</td>
<td>Full saturation</td>
<td>OS1</td>
<td>Al</td>
<td>0.911</td>
<td>4.14</td>
<td>0.76</td>
<td>5.76</td>
</tr>
</tbody>
</table>

**Table E1:** Summary of the experiments performed by Chapman et al. (2006)

Figure E4 shows the measured Hugoniot data in shock-velocity particle-velocity space and Figure E5 in the stress particle-velocity space. Simple least square linear fits have been applied to the data. The values obtained for the linear coefficients are heavily dependent on the particle velocity points. We would expect that the shock-velocity particle-velocity dependence would be non-linear for such a porous system. However, we consider the simple linear fits presented to be adequately representative of the data over the investigated range of particle-velocities given the small number of points.

The presence of air in the voids will significantly influence the compaction process and should be included in any modelling undertaken. The water acts to homogenise the material, allowing
stress to be transmitted though the water filled voids. This is reflected by a significant increase in shock-velocity for full saturated sand when compared with the dry sand for a given input stress.

![Graph of Shock Velocity vs Particle Velocity]

**Figure E4:** Shock-velocity dependence on particle-velocity.

![Graph of Longitudinal Stress vs Particle Velocity]

**Figure E5:** Stress dependence on particle-velocity.
The Hugoniot curves for sand with dry and full saturation have been obtained. For dry sand, the shock response was analyzed. Moreover, for sand with full saturation, the saturation was found to have a significant effect on the measured shock-velocity.

According to the results of the plate-impact experiment, the parameters for linear polynomial EOS in LS-DYNA can be determined. The linear polynomial EOS can be written as

\[ P = C_0 + C_1 \mu \]

where \( P \) is the pressure, \( \mu = \frac{\rho}{\rho_0} - 1 \), \( C_0 \) and \( C_1 \) is constants.

Based on the value of initial bulk density of sand, shock velocity and particle velocity, the density under shock wave pressure can be calculated by equation

\[ \rho \rho_0 U_s = \rho (U_s - u_p) \]

Thereafter, \( \mu \) can be obtained, and the parameters of linear polynomial EOS can be determined through simple least square linear fitting and shown in Figure E6 and Table E2. The linear polynomial form can be used as a simple approximation, however the three phase model of Chapter 5 is likely more accurate.

![Figure E6: Stress dependence on particle-velocity.](image)
<table>
<thead>
<tr>
<th></th>
<th>$C_0$</th>
<th>$C_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dry</td>
<td>-0.29</td>
<td>27.04</td>
</tr>
<tr>
<td>Full saturation</td>
<td>-3.16</td>
<td>2.32</td>
</tr>
</tbody>
</table>

Table 5.4.2 The parameters of linear polynomial EOS.