AD FALCON API Manual

SANISAND Model Overview

Javad Ghorbani

March 14, 2026

Contents

1 SANISAND Model Overview

1.1 Syntax

This model is configured in `% Materials` as a user-defined mechanical material. Use `@UMAT:` with category `Mechanical` and pass the parameters as `name=value` pairs.

Example:

```
@UMAT: path/to/SANISANDModelUMAT.cpp path/to/SANISANDModelUMAT.hpp
Mechanical \
    G0=200 K0=400 Mc=1.2 Me=1.0 Lambda=0.02 N_c=0.8 alpha_c=0.0 \
    n_b=1.5 ch=0.5 n_d=1.0 h0=100 A0=1.0 cz=2.0 zmax=10 \
    Patm=100 P_min=0.1 STOL=1e-5 LTOL=1e-6 \
    CustomVariable=AlphaXX,AlphaYY,AlphaZZ,AlphaZY,AlphaZX,AlphaXY,ZXX,ZYY,
ZZZ,ZZY,ZZX,ZXY,AlphaInitialXX,AlphaInitialYY,AlphaInitialZZ,
AlphaInitialZY,AlphaInitialZX,AlphaInitialXY,eps_p_q
```

For readability, this example is wrapped across multiple lines; in input files you should write the full `@UMAT:` directive on a single line.

Use the parameter names listed in this page.

1.2 Material parameters

Table 1. Material parameters and their descriptions

Symbol	Keyword in input	Units	Required	Description
G_0	G0	–	✓	Shear modulus constant in the hypoelastic law.
K_0	K0	–	✓	Bulk modulus constant in the hypoelastic law.
M_c	Mc	–	✓	Critical-state slope in triaxial compression.
M_e	Me	–	✓	Critical-state slope in triaxial extension.

Symbol	Keyword in input	Units	Required	Description
λ	`Lambda`	–	✓	Critical-state / packing slope parameter used in this implementation.
N_c	`N_c`	–	✓	Critical-state reference void-ratio parameter.
α_{CSL}	`alpha_c`	stress	✓	Curvature parameter in the CSL expression.
n^b	`n_b`	–	✓	Bounding-surface exponent parameter.
c^h	`ch`	–	✓	Hardening parameter controlling modulus decay with void ratio.
n^d	`n_d`	–	✓	Dilatancy exponent parameter.
h_0	`h0`	–	✓	Hardening modulus scale.
A_0	`A0`	–	✓	Dilatancy scale parameter.
c_z	`cz`	–	✓	Fabric evolution parameter.
$z_{\max}$	`zmax`	–	✓	Maximum fabric tensor magnitude.
p_{atm}	`Patm`	stress	✓	Atmospheric reference pressure used in normalization.
$P_{\min}$	`P_min`	stress	✓	Minimum pressure threshold used for numerical robustness.
STOL	`STOL`	–	✓	Integration tolerance for substepping.

Symbol	Keyword in input	Units	Required	Description
LTOL	LTOL	–	✓	Load/unload detection tolerance.
c_1	c1	–	×	Optional unsaturation coupling coefficient in $c_\zeta(\zeta)$.
c_2	c2	–	×	Optional unsaturation coupling coefficient in $c_\zeta(\zeta)$.
p_{apex}	p_apex	stress	×	Apex pressure below which the smoothed response is elastic.
Δp_{apex}	apex_smooth_dp	stress	×	Transition width used by apex smoothing.

Use the parameter names shown in the table above and in the examples below. Optional UMAT parameters:

- `exp_clamp_min`, `exp_clamp_max` — clamps the exponential arguments in hardening/dilatancy terms to prevent overflow in extreme states.
- `consistency_drift_enable`, `consistency_drift_tol` — optional end-of-step "consistency drift" correction controls (if enabled).
- `overshoot_reposition_enable`, `overshoot_eps_qp_threshold`, `overshoot_mq_exponent` — optional memory repositioning controls for stress-overshooting during reversals/contact (see Ghorbani et al., 2023).
- `apex_smooth_enable`, `p_apex`, `apex_smooth_dp` — optional apex smoothing controls near small mean stress.
- `c1`, `c2` — optional unsaturation coupling modifiers used in $c_\zeta(\zeta)$ (set `c1=0` to disable).

1.2.1 Examples (how to include optional UMAT parameters)

All optional UMAT parameters are included as extra `name=value` pairs on the same `@UMAT:` line.

Example A — enable exponential argument clamping

```
@UMAT: path/to/SANISANDModelUMAT.cpp path/to/SANISANDModelUMAT.hpp
Mechanical ... exp_clamp_min=-2 exp_clamp_max=2
```

Interpretation: any internal term of the form $\exp(\cdot)$ that uses the state parameter will clamp its argument to the provided bounds before exponentiation.

Example B — enable apex smoothing near small mean stress

```
@UMAT: path/to/SANISANDModelUMAT.cpp path/to/SANISANDModelUMAT.hpp
Mechanical ... apex_smooth_enable=1 p_apex=0.1 apex_smooth_dp=0.05
```

Interpretation: below `p_apex` the response is elastic; between `p_apex` and `p_apex + apex_smooth_dp` the model transitions smoothly to the standard SANISAND response.

Example C — enable memory repositioning (overshoot control)

```
@UMAT: path/to/SANISANDModelUMAT.cpp path/to/SANISANDModelUMAT.hpp
Mechanical ... overshoot_reposition_enable=1 overshoot_eps_qp_threshold=1e-4
overshoot_mq_exponent=1
```

Interpretation: activates the memory repositioning strategy for reversal/overshooting control. This is intended for situations where stress reversals (including contact problems) can lead to overshooting and poor convergence.

Brief theory (what “memory repositioning” does)

When the loading direction reverses, SANISAND-type models rely on an internal “memory” reference backstress (here, the `AlphaInitial*` custom variables) to determine whether the current increment is loading/unloading and how the plastic modulus should evolve. In difficult reversal situations (common in contact), that memory can lead to stress overshooting and excessive substepping.

Memory repositioning modifies the stored reference backstress during reversal/reloading so that the model transitions more robustly and avoids pathological overshoot, while still preserving the intended bounding-surface behavior.

Parameters and state (what each one is for)

UMAT parameters:

Name	Role
`overshoot_reposition_enable`	Enables/disables memory repositioning (`1`/`0`).
`overshoot_eps_qp_threshold`	Reference level for reversal classification (the algorithm compares the accumulated deviatoric plastic strain since reversal against this value). Smaller values trigger stronger/earlier repositioning effects.

Name	Role
overshoot_mq_exponent	Shapes how strongly the repositioning weight changes from "small reversal" to "true reversal". Larger values make the transition sharper.

Custom state variables involved (declare them via `CustomVariable=` if you want them stored and visible in outputs):

Name	Role
AlphaInitialXX...AlphaInitialXY	Reference (memory) backstress used by the reversal logic and plastic modulus evaluation. (These are part of the required SANISAND custom variables.)
Lstepcount	Reversal-step counter used to track reversal/reloading stages.
Ir	Stores the last positive reversal indicator value before a reversal (used as a scale in repositioning).
eps_qp_rev	Accumulated equivalent deviatoric plastic strain during the current reversal episode (used to compute the repositioning weight).

Reference: Ghorbani, J., Chen, L., Kodikara, J., Carter, J.P. and McCartney, J.S. (2023). *Memory repositioning in soil plasticity models used in contact problems.* Computational Mechanics, 71(3), 385–408.

Example D — enable unsaturation effects in the CSL modifier

```
@UMAT: path/to/SANISANDModelUMAT.cpp path/to/SANISANDModelUMAT.hpp
Mechanical ... c1=0.2 c2=1.5
```

Interpretation: enables $c_\zeta(\zeta) = 1 - c_1\,(1 - \exp(c_2\zeta))$ in the CSL-related terms. Set `c1=0` to disable (so $c_\zeta = 1$).

1.3 Custom state variables

This UMAT uses custom state variables for kinematic hardening, fabric, and internal integration state. Declare them using `CustomVariable=` in the `@UMAT:` line.

Required:

- `AlphaXX`, `AlphaYY`, `AlphaZZ`, `AlphaZY`, `AlphaZX`, `AlphaXY`

- ZXX, ZYY, ZZZ, ZZY, ZZX, ZXY
- AlphaInitialXX, AlphaInitialYY, AlphaInitialZZ, AlphaInitialZY, AlphaInitialZX, AlphaInitialXY
- eps_p_q

Optional diagnostics/tracking:

- PlasticStrainIncXX, PlasticStrainIncYY, PlasticStrainIncZZ, PlasticStrainIncZY, PlasticStrainIncZX, PlasticStrainIncXY (declare these via CustomVariable= if you want them tracked and available for output)
- Lstepcount, Ir, eps_qp_rev
- NumSubsteps, NumFailedSubsteps

1.4 Introduction

The model considered here shares many features with the SANISAND model by Dafalias and Manzari (2004), which has inspired a number of advanced models for granular soils (e.g., Chen et al., 2024; Petalas et al., 2020). The corresponding equations and parameters are summarized in Table 2.

The model's parameters are categorized into the following groups:

- **Elasticity**
- **Critical State**
- **Kinematic Hardening**
- **Dilatancy**

1.5 Elasticity

The model employs two hypoelastic laws to describe the elastic behavior through bulk and shear moduli, characterized by K_0 and G_0 as the primary elasticity parameters.

1.5.1 Bulk Modulus

$$K = K_0 p_{\text{atm}} \frac{(1+e)}{e} \left(\frac{p'}{p_{\text{atm}}} \right)^{\frac{2}{3}} \tag{1}$$

1.5.2 Shear Modulus

$$G = G_0 p_{\text{atm}} \frac{(2.97-e)^2}{1+e} \left(\frac{p'}{p_{\text{atm}}} \right)^{\frac{1}{2}} \tag{2}$$

1.6 Critical State

The critical state is defined in two primary spaces:

1.6.1 In the void ratio e - mean effective stress p' space

$$\ln e_c = \ln N_c - \lambda \ln\left(\max(P_{\min}, p') + \alpha_{\text{CSL}}\right) + \ln c_\zeta(\zeta) \tag{3}$$

where: - N_c denotes the intersection of the critical state line with the void ratio axis at unit mean effective stress. - λ is the slope of this line. - α_{CSL} (=`alpha_c`) controls its curvature. - $c_\zeta(\zeta)$ is an optional unsaturation-dependent multiplier. If not used, set it to 1.

The unsaturation coupling used by this implementation is:

$$\zeta = f_s(p_c)\,(1 - S_r)\,, \qquad f_s(p_c) = 1 + \frac{(p_c/p_{\text{atm}})}{10.7 + 2.4(p_c/p_{\text{atm}})}$$

$$c_\zeta(\zeta) = 1 - c_1\,(1 - \exp(c_2\zeta))$$

where $p_c = p_a - p_w$ is suction and S_r is the degree of saturation. If `c1=0` (the default), then $c_\zeta = 1$ and the CSL reduces to the saturation-independent form.

1.6.2 In the deviatoric stress q - mean effective stress p' space

$$q = Mp' \tag{4}$$

with:

$$M = M_c g(\theta_l, \alpha') \tag{5}$$

where the interpolation function is given by:

$$g(\theta_l, \alpha') = \left(\frac{2\alpha'^4}{1 + \alpha'^4 - (1 - \alpha'^4)\cos(3\theta_l)}\right)^{\frac{1}{4}} \tag{6}$$

$$\alpha' = \frac{M_e}{M_c} \tag{7}$$

where M_e and M_c correspond to the critical state slope in triaxial extension and compression, respectively.

1.7 Kinematic Hardening

The evolution of hardening in triaxial condition is expressed as:

$$\frac{\partial \alpha_k}{\partial \varepsilon_q^p} = h\left(M^b - \eta_\sigma\right) \tag{8}$$

where: - ε_q^p is the plastic deviatoric strain. - α_k is the kinematic hardening parameter. - η_σ is the stress ratio. - $M^b = M\langle\exp(-n^b\psi)\rangle$ is the peak stress ratio with Macaulay brackets $\langle\cdot\rangle$. - $\psi = e - e_c$ is the state parameter (Been and Jefferies, 1985).

The hardening modulus is defined as:

$$h = h_0 G_0 (1 - c_h e) \left(\frac{p'}{p_{\text{atm}}} \right)^{-\frac{1}{2}} \frac{1}{|a_k - a_k^{\text{in}}|} \tag{9}$$

where a_k^{in} stores the kinematic hardening parameter upon stress reversal.

1.8 Dilatancy

The dilatancy flow rule is given as:

$$LA_d \left(M^d - \eta_\sigma \right) \tag{10}$$

where: - $M^d = M \exp(n^d \psi)$ is the dilatancy stress ratio. - $L = 1$ if $\eta_\sigma - \alpha_k \geq 0$. - $L = -1$ if $\eta_\sigma - \alpha_k < 0$.

The fabric effect on dilatancy parameter is modeled as:

$$A_d = A_0 (1 + \langle zL \rangle) \tag{11}$$

The fabric evolution follows:

$$dz = -c_z \langle -d\varepsilon_v^p \rangle (z_{\max} L + z) \tag{12}$$

1.9 Integration Scheme

The integration scheme builds upon the foundational work by Sloan et al. (2001), extending its application into the domain of bounding surface plasticity as detailed by Ghorbani et al. (2021a) and Ghorbani and Airey (2021). This refined approach employs a comparative error analysis between:

- A second-order accurate modified Euler solution.
- A first-order accurate Euler solution.

This error comparison helps devise an adaptive and automatic substepping scheme, enhancing efficiency. The integration tolerance was set to 1×10^{-5} in all analyses.

1.10 Numerical Parameters

The model includes several numerical control parameters:

Keyword in input	Description	Default Value
STOL	Integration tolerance for substepping	1×10^{-5}
LTOL	Load/unload detection tolerance	1×10^{-6}
P_min	Minimum pressure threshold	0.1 kPa
Patm	Atmospheric reference pressure	100.0 kPa

1.10.1 Optional Exponential Argument Clamping

The exponential arguments in the hardening and dilatancy equations ($n^b\psi$ and $n^d\psi$) can optionally be clamped to prevent numerical overflow. Two optional parameters control these bounds:

Parameter	Description	Default Value
`exp_clamp_min`	Lower bound for exponential argument	-10^{30} (no bound)
`exp_clamp_max`	Upper bound for exponential argument	10^{30} (no bound)

By default, these parameters are set to very large values, effectively applying no clamping. For numerical stability in extreme conditions, users may set tighter bounds (e.g., `exp_clamp_min = -2.0` and `exp_clamp_max = 2.0`).

1.10.2 Optional Apex Smoothing Near $p' \to 0$

This option modifies only the **near-apex behavior** (small p') to improve robustness of stress-ratio quantities.

Step 1 — smooth activation weight

Define a C^1 smoothstep weight based on mean stress:

$$w(p') = \begin{cases} 0 & p' \le p_{\text{apex}}, \\ t^2(3-2t) & p_{\text{apex}} < p' < p_{\text{apex}} + \Delta p, \\ 1 & p' \ge p_{\text{apex}} + \Delta p, \end{cases} \qquad t = \frac{p' - p_{\text{apex}}}{\Delta p}.$$

This same weight is used to (i) deactivate plasticity below p_{apex} and (ii) blend cap vs. standard directions in the transition.

Convexity guard (important)

The apex cap used by this implementation is constructed to be **convex** in the p'–q plane (so it does not create a non-convex yield cap near the transition). For the chosen q^4 cap (Step 4), convexity over $0 \le q \le q_{\text{join}}$ reduces to the condition:

$$\Delta p \le \frac{5}{3}\, p_{\text{apex}}.$$

If a user provides a larger `apex_smooth_dp`, the code **clamps** it to $(5/3)\, p_{\text{apex}}$ (and prints a warning) before constructing the cap. Note that the coefficient c is **not a user input**; it is computed internally from p_{apex}, Δp, and α.

Step 2 — stress-ratio denominator clamp (only when enabled)

When enabled, the stress-ratio denominator used to form η is clamped as:

$$p_{\text{ratio}} = \max\big(p',\, p_{\text{apex}}\big)\,, \qquad \eta = \mathbf{s}/p_{\text{ratio}}.$$

Step 3 — multiaxial scalar α from backstress tensor

The kinematic hardening/backstress is stored as a deviatoric stress-ratio tensor (Voigt order $[xx, yy, zz, zy, zx, xy]$):

$$\boldsymbol{\alpha} \equiv [\alpha_{xx},\ \alpha_{yy},\ \alpha_{zz},\ \alpha_{zy},\ \alpha_{zx},\ \alpha_{xy}]^T, \qquad \alpha \equiv \sqrt{3J_2(\boldsymbol{\alpha})}.$$

with:

$$J_2(\boldsymbol{\alpha}) = \frac{1}{2}\left(\alpha_{xx}^2 + \alpha_{yy}^2 + \alpha_{zz}^2 + 2\alpha_{zy}^2 + 2\alpha_{zx}^2 + 2\alpha_{xy}^2\right).$$

Step 4 — associated apex cap and its gradient

The cap surface is used only through its stress gradient (normal), defined implicitly by:

$$f_{\text{cap}}(\boldsymbol{\sigma}) = \left(p' - p_{\text{apex}}\right) - \frac{q^2}{k} - c\,q^4 = 0, \qquad q^2 = 3J_2(\mathbf{s}).$$

This implementation chooses:

$$p_{\text{join}} = p_{\text{apex}} + \Delta p, \qquad q_{\text{join}} = \alpha\,p_{\text{join}}, \qquad k = \frac{1}{\left(2\Delta p/q_{\text{join}}^2\right) - \left(1/(2\alpha q_{\text{join}})\right)}, \qquad c = \frac{1}{2\alpha q_{\text{join}}^3} - \frac{\Delta p}{q_{\text{join}}^4}.$$

Then:

$$\frac{\partial f_{\text{cap}}}{\partial \boldsymbol{\sigma}} = \frac{1}{3}[1, 1, 1, 0, 0, 0]^T - \left(\frac{1}{k} + 2cq^2\right)[3s_{xx},\ 3s_{yy},\ 3s_{zz},\ 6s_{zy},\ 6s_{zx},\ 6s_{xy}]^T.$$

Step 5 — blend yield and plastic-potential gradients

Let $\mathbf{n}_{\text{SANISAND}} = \partial f_{\text{SANISAND}}/\partial\boldsymbol{\sigma}$ be the standard yield gradient and $\mathbf{g}_{\text{SANISAND}}$ the standard plastic potential gradient. This implementation uses:

$$\mathbf{n} = (1-w)\,\frac{\partial f_{\text{cap}}}{\partial \boldsymbol{\sigma}} + w\,\mathbf{n}_{\text{SANISAND}}, \qquad \mathbf{g} = (1-w)\,\frac{\partial f_{\text{cap}}}{\partial \boldsymbol{\sigma}} + w\,\mathbf{g}_{\text{SANISAND}},$$

with a sign check to align the cap gradient with the SANISAND gradient.

Step 6 — plastic multiplier and tangents

Below p_{apex} (i.e. $w = 0$), the model returns a purely elastic response.

For $w > 0$, the plastic multiplier is computed in the standard return-mapping form using the (already blended) yield and plastic-potential gradients from Step 5, and is then clipped to be non-negative:

$$\Delta\lambda \leftarrow \max(0, \Delta\lambda)$$

No additional explicit w scaling of $\Delta\lambda$ is applied. Likewise, the tangent stiffness uses the standard elastoplastic tangent constructed from the same blended gradients (rather than a convex combination of elastic and elastoplastic tangents).

User controls

Parameter	Description	Default Value
`apex_smooth_enable`	Enable/disable apex smoothing (0/1)	`0`
`p_apex`	Apex pressure p_{apex} where plasticity is inactive	`P_min`
`apex_smooth_dp`	Transition width Δp above the apex	`0.1 * max(p_apex, P_min)`

Example

```
apex_smooth_enable=1 p_apex=0.1 apex_smooth_dp=0.1
```

Type	Default Equation	Variables
Elasticity	**Bulk Modulus:** $K = K_0\, p_{\text{atm}} \dfrac{1+e}{e}\left(\dfrac{p'}{p_{\text{atm}}}\right)^{2/3}$	
Shear Modulus: $G = G_0\, p_{\text{atm}} \dfrac{(2.97-e)^2}{1+e}\left(\dfrac{p'}{p_{\text{atm}}}\right)^{1/2}$	e: void ratio	

p': mean effective stress p_{atm}: atmospheric pressure | | **Critical State** | **CSL in e–p' space:** $\ln e_c = \ln N_c - \lambda \ln(\max(P_{\min}, p')) + \alpha_{\text{CSL}}) + \ln c_\zeta(\zeta)$ **CSL in q–p' space:** $q = M\,p'$ **With:** $M = M_c\, g(\theta_l, \alpha')$ $g(\theta_l, \alpha') = \left[\frac{2\,\alpha'^4}{1+\alpha'^4-(1-\alpha'^4)\cos 3\theta_l}\right]^{1/4}$ $\alpha' = \frac{M_e}{M_c}$ | e_c: void ratio on the critical state line p': mean effective stress $c_\zeta(\zeta)$: optional unsaturation multiplier (set to 1 if not used) q: deviatoric stress M: critical state slope (values M_e, M_c in extension/compression) θ_l: Lode angle α': ratio between extension/compression critical slopes | | **Kinematic hardening** | **Peak stress ratio:** $M^b = M\langle\exp(-n^b\,\psi)\rangle$ **State parameter:** $\psi = e - e_c$ **Hardening evolution:** $\frac{\partial \alpha_k}{\partial \varepsilon_q^p} = h\big(M^b - \eta_\sigma\big)$ **With:** $h = h_0 G_0 (1 - c_h e)\left(\frac{p'}{p_{\text{atm}}}\right)^{-1/2} \frac{1}{|a_k - a_k^{\text{in}}|}$ | M^b: peak stress ratio n^b: hardening exponent parameter ψ: state parameter α_k: kinematic hardening parameter ε_q^p: plastic deviatoric strain η_σ: stress ratio a_k^{in}: kinematic hardening parameter at stress reversal | | **Dilatancy** | **Dilatancy stress ratio:** $M^d = M\exp(n^d\,\psi)$ **Flow rule:** $LA_d(M^d - \eta_\sigma)$ **Fabric effect:** $A_d = A_0(1 + \langle zL\rangle)$ **Fabric evolution:** $dz = -c_z\langle -d\varepsilon_v^p\rangle(z_{\max} L + z)$ | M^d: dilatancy stress ratio n^d: dilatancy exponent parameter L: +1 if $\eta_\sigma - \alpha_k \geq 0$, −1 otherwise A_d: dilatancy parameter z: fabric–dilatancy parameter ε_v^p: plastic volumetric strain $\langle\cdot\rangle$: Macaulay brackets |

1.11 Toyoura Sand Test Setup

A suite of **undrained triaxial tests** on Toyoura Sand was simulated using a **single Gauss-point routine** (no FEM mesh) to verify the constitutive model implementation.
For each initial void ratio ($e = 0.735,\ 0.833,\ 0.907$), specimens were:

1. **Consolidated** drained to the target mean effective confining stress $p' = 100,\ 1000,\ 2000,\ (3000)\,\mathrm{kPa}$.

2. **Sheared** undrained by imposing axial strain increments; responses in q vs. p' and q vs. axial strain were recorded.

1.11.1 Model Parameters for Toyoura Sand

Parameter	Value	Parameter	Value
G_0	125.0	c_z	600
K_0	150.0	z_{max}	4
M_c	1.25	p_{atm} kPa	100.0
λ	0.37	P_{min} kPa	1×10^{-1}
N_c	18.7	A_0	0.4
α_{CSL}	3370	n^b	1.25
c_h	0.968	n^d	2.3
h_0	12.0	M_e	0.89

Results

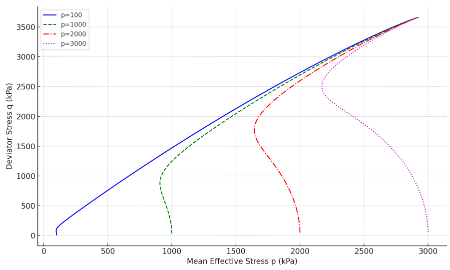
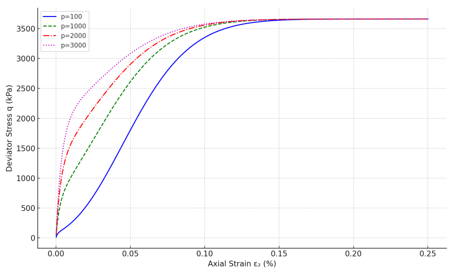
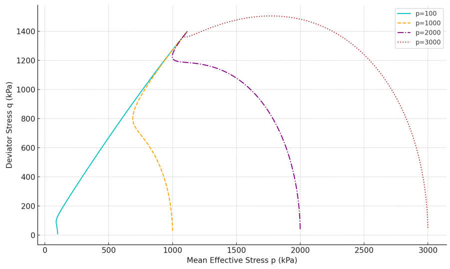
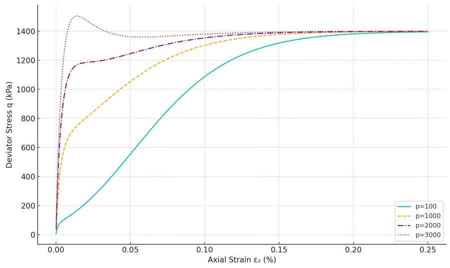
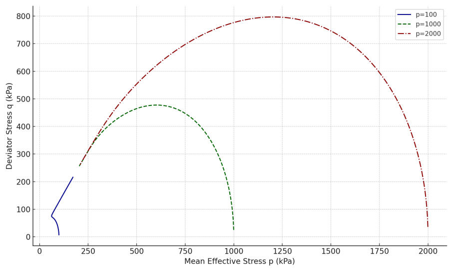
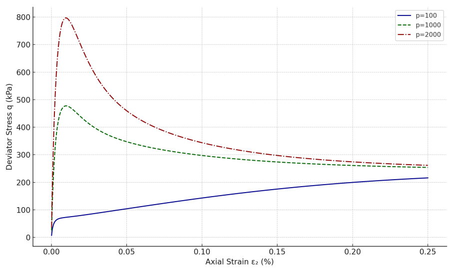
Void Ratio e	q vs p	q vs ϵ_2
0.735	 	

Figure: q vs p for e = 0.735

Void Ratio e	q vs p	q vs ϵ_2
Figure: q vs ϵ_2 for e = 0.735		
0.833	 	
Figure: q vs p for e = 0.833 Figure: q vs ϵ_2 for e = 0.833		
0.907	 	
Figure: q vs p for e = 0.907 Figure: q vs ϵ_2 for e = 0.907		