AD FALCON API Manual

SANISAND Model Overview

Javad Ghorbani

March 26, 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 Parameters (input keywords)

The following UMAT parameters are required (pass them as `name=value`):

Keyword in input	Required	Meaning
G0	✓	Shear modulus constant.
K0	✓	Bulk modulus constant.
Mc	✓	Critical-state slope in triaxial compression.
Me	✓	Critical-state slope in triaxial extension.
Lambda	✓	Critical-state/packing slope parameter used in this implementation.
N_c	✓	Critical-state reference void ratio parameter.
alpha_c	✓	Critical-state curvature parameter.
n_b	✓	Bounding surface exponent parameter.

Keyword in input	Required	Meaning
ch	✓	Hardening parameter controlling modulus decay with void ratio.
n_d	✓	Dilatancy exponent parameter.
h0	✓	Hardening modulus scale.
A0	✓	Dilatancy scale parameter.
cz	✓	Fabric evolution parameter.
zmax	✓	Maximum fabric tensor magnitude.
Patm	✓	Atmospheric reference pressure used in normalization.
P_min	✓	Minimum pressure threshold used for numerical robustness.
STOL	✓	Integration tolerance for substepping.
LTOL	✓	Load/unload detection tolerance.

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`).

Name	Role
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.
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 Symbol glossary

The table below collects the mathematical symbols used on this page. Input keywords such as `G0`, `K0`, `Lambda`, and `AnalysisType` are documented separately in the parameter and mini-input sections.

Symbol	Meaning
e	Current void ratio.
e_c	Void ratio on the critical state line.
p'	Mean effective stress.
p_{net}	Mean net stress used by the unsaturated standalone driver.

Symbol	Meaning
p_{atm}	Atmospheric reference pressure.
$P_{\min}$	Minimum mean-stress threshold used for robustness.
p_{apex}	Apex pressure below which the smoothed response is elastic.
p_{join}	Mean stress where the apex cap joins the standard SANISAND branch.
p_{ratio}	Clamped denominator used to form the stress ratio in the apex-smoothing option.
q	Deviatoric stress.
q_{join}	Deviatoric stress at the apex-cap joining point.
K	Bulk modulus.
G	Shear modulus.
K_0	Bulk-modulus constant in the hypoelastic law.
G_0	Shear-modulus constant in the hypoelastic law.
M	Critical-state stress-ratio function in $q - p'$ space.
M_c	Critical-state slope in triaxial compression.
M_e	Critical-state slope in triaxial extension.
M^b	Peak stress ratio used in hardening.
M^d	Dilatancy stress ratio.
$g(\theta_l, \alpha')$	Lode-angle interpolation function for the critical-state slope.
θ_l	Lode angle.
N_c	Critical-state reference void-ratio parameter / reduced-model intercept slot used in this implementation.
λ	Critical-state slope parameter in the $e - \ln p'$ relation.
α_{CSL}	CSL curvature parameter.
α'	Ratio M_e/M_c used in the Lode-angle interpolation.
α_k	Kinematic hardening variable / backstress scalar in triaxial form.
a_k^{in}	Stored backstress reference at stress reversal.

Symbol	Meaning
$\boldsymbol{\alpha}$	Backstress tensor in deviatoric stress-ratio form.
α	Scalar measure derived from the backstress tensor in the apex-smoothing section.
η_σ	Stress ratio.
$\boldsymbol{\eta}$	Tensor stress ratio used in the apex-smoothing description.
ψ	State parameter, defined as $e - e_c$.
h	Hardening modulus.
h_0	Hardening modulus scale parameter.
c_h	Void-ratio-dependent hardening coefficient.
n^b	Bounding-surface exponent parameter.
n^d	Dilatancy exponent parameter.
A_d	Dilatancy parameter.
A_0	Base dilatancy scale parameter.
L	Loading-direction indicator used in the dilatancy/fabric law.
z	Fabric-dilatancy variable.
$z_{\max}$	Upper bound on the fabric variable magnitude.
c_z	Fabric evolution parameter.
ε_q^p	Plastic deviatoric strain.
ε_v^p	Plastic volumetric strain.
$\Delta\lambda$	Plastic multiplier increment.
$\langle\cdot\rangle$	Macaulay brackets.
p_c	Suction, defined here as $p_a - p_w$.
p_a	Pore-air pressure.
p_w	Pore-water pressure.
S_w	Degree of saturation used consistently in the constitutive and standalone unsaturated relations on this page.
χ	Effective-stress weighting factor in the unsaturated standalone driver.
ζ	Unsaturation state variable used in the CSL modifier.
$f_s(p_c)$	Suction-dependent scaling function used in ζ.
$c_\zeta(\zeta)$	Unsaturated multiplier applied to the CSL expression.

Symbol	Meaning
c_1, c_2	Parameters controlling the CSL unsaturation modifier.
$w(p')$	Smooth activation weight in the apex-smoothing option.
t	Normalized transition variable in the apex-smoothing weight.
Δp	Smoothing transition width near the apex.
k	Apex-cap parameter controlling the quadratic term in q.
c	Apex-cap coefficient for the quartic q^4 term.
$f_{\text{cap}}(\boldsymbol{\sigma})$	Apex-cap surface function.
$\boldsymbol{\sigma}$	Cauchy stress tensor in Voigt notation.
$\mathbf{s}$	Deviatoric stress tensor.
$J_2(\boldsymbol{\alpha})$	Second deviatoric invariant of the back-stress tensor.
$J_2(\mathbf{s})$	Second deviatoric invariant of the stress tensor.
$\mathbf{n}_{\text{SANISAND}}$	Standard SANISAND yield gradient.
$\mathbf{g}_{\text{SANISAND}}$	Standard SANISAND plastic-potential gradient.
$\mathbf{n}$	Blended yield gradient used with apex smoothing.
$\mathbf{g}$	Blended plastic-potential gradient used with apex smoothing.
$Epsilon_2$	Axial strain measure written by the standalone mini CSV output.
`deviatoric_strain`	Standalone exported deviatoric strain measure used in verification plots.
`volumetric_strain`	Standalone exported volumetric strain measure used in verification plots.
$dq/dp_{net} = 3$	Stress-path condition enforced in the unsaturated standalone verification branches.

1.6 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.

1.6.1 Bulk Modulus

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

1.6.2 Shear Modulus

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

1.7 Critical State

The critical state is defined in two primary spaces:

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

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

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:

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

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

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

1.7.2 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.

1.8 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:

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

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

1.9 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:

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

1.10 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.

1.11 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

1.11.1 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`).

1.11.2 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 $\boldsymbol{\eta}$ is clamped as:

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

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:** $K = K_0\, p_{\text{atm}} \frac{1+e}{e}\left(\frac{p'}{p_{\text{atm}}}\right)^{2/3}$	
Shear Modulus: $G = G_0\, p_{\text{atm}} \frac{(2.97-e)^2}{1+e}\left(\frac{p'}{p_{\text{atm}}}\right)^{1/2}$	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 |

1.12 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 $p' = 100,\ 1000,\ 2000,\ (3000)\,\text{kPa}$.

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

1.12.1 Model Parameters for Toyoura Sand

Parameter	Value	Parameter	Value

1.13 FALCON mini

The packaged mini tool id is `SANISAND`. It lives under `mini_tools/SANISAND`.

1.13.1 How to run

```
falcon --mini-root /path/to/UMATLIB_FALCON/falcon_minis --mini-tool SANISAND
--mini-input
/path/to/UMATLIB_FALCON/falcon_minis/SANISAND/cases/triaxial_drained
```

Packaged simulation families:

Packaged case	Path	Purpose
Drained triaxial	cases/triaxial_drained/input.txt	Monotonic drained Toyoura-like reference path.
Constant-p' triaxial	cases/triaxial_constant_p/input.txt	Monotonic triaxial path with the initial mean effective stress held constant.
Undrained triaxial	cases/triaxial_undrained/input.txt	Monotonic undrained Toyoura-like reference path.
Cyclic q-controlled triaxial	cases/cyclic_q_control/input.txt	Cyclic undrained stress-controlled example.
Cyclic strain-controlled drained	cases/cyclic_strain_controlled_drained/input.txt	Cyclic drained strain-history example.
Cyclic strain-controlled undrained	cases/cyclic_strain_controlled_undrained/input.txt	Cyclic undrained strain-history example.
Kurnell constant suction 10050D-D	cases/kurnell_const_suction_10050D/input.txt	Unsaturated drained $dq/dp_{net} = 3$ path at s=51 kPa, dense initial state.
Kurnell constant suction 5050L-D	cases/kurnell_const_suction_5050L/input.txt	Unsaturated drained $dq/dp_{net} = 3$ path at s=51 kPa, loose initial state.
Kurnell constant water content 100100D-D	cases/kurnell_const_wcontent_100100D/input.txt	Unsaturated drained $dq/dp_{net} = 3$ path with evolving suction, dense initial state.
Kurnell constant water content 50100L-D	cases/kurnell_const_wcontent_50100L/input.txt	Unsaturated drained $dq/dp_{net} = 3$ path with evolving suction, loose initial state.

1.13.2 Input syntax

`input.txt` uses three explicit prefixes:

- `Property:` for constitutive parameters
- `StateVariable:` for the initial stress and hydraulic state
- `CustomVariable:` for internal variables such as backstress and fabric

The driver controls are then set with standalone keys such as `AnalysisType:`, `NumSteps:`, `EpsilonYY:`, `QUP:`, `QD:`, and, for the strain-controlled cyclic cases, `StrainPatternFile:`.

`AnalysisType` value	Meaning in the standalone mini
2	Drained triaxial monotonic loading.
3	Undrained triaxial monotonic loading.
4	Cyclic q-controlled triaxial loading.
5	$dq/dp_{net} = 3$ loading at constant suction.
6	$dq/dp_{net} = 3$ loading at constant water content.
7	Cyclic strain-controlled drained loading.
8	Cyclic strain-controlled undrained loading.
9	Monotonic triaxial loading at constant mean effective stress p'.

Mini inputs used by the packaged cases:
Constitutive records:

Record / key	Used by	Required / choices / defaults	Meaning
`Property:`	all cases	Required repeated record; key set depends on the constitutive calibration being used	SANISAND-type constitutive parameters such as `G0`, `Mc`, `Lambda`, fabric and dilatancy constants, and numerical safeguards.
`StateVariable:`	all cases	Required repeated record; packaged cases seed the initial state explicitly	Initial stress state, void ratio, pore pressures, and hydraulic state.
`CustomVariable:`	all cases	Required for packaged cases that seed backstress/fabric/history explicitly	Internal backstress, fabric, and other constitutive history variables.

Driver controls:

Input key	Used by	Required / choices / defaults	Meaning
`AnalysisType:`	all cases	Required; choices 2, 3, 4, 5, 6, 7, 8, 9 in the packaged mini	Selects the loading program.
`NumSteps:`	monotonic, constant-p', and q/p_{net} cases	Required for monotonic, constant-p', and q/p_{net} modes	Number of driver increments.
`EpsilonYY:`	monotonic, constant-p', and q/p_{net} cases	Required for monotonic, constant-p', and q/p_{net} modes	Axial strain increment used by the triaxial branches.
`QUP:`, `QD:`	cyclic q-controlled	Required for `AnalysisType = 4`	Upper and lower deviatoric stress reversal targets.
`PoreAirPressure`, `PoreWaterPressure`	unsaturated cases	Required for unsaturated packaged cases	Starting pore pressures used to define suction.
`StrainPattern File:`	cyclic strain-controlled cases	Required for `AnalysisType =` 7 and 8	External strain history file read from the case directory.

1.13.3 Hydromechanical assumptions

The packaged SANISAND-type mini is primarily an effective-stress sand model:

- monotonic drained, monotonic undrained, and cyclic triaxial examples are effective-stress mechanical paths
- the constant-p' branch solves the radial strain increment each step so the mean effective stress stays at its initial value; it does not hold σ_1 or σ_3 fixed
- the unsaturated q/p_{net} branches add suction, saturation, χ, and ζ output through the standalone driver
- the cyclic strain-controlled examples require both `input.txt` and `strain_pattern.txt` to be staged together

1.13.4 Sample input

Monotonic drained triaxial example Path: `mini_tools/SANISAND/cases/triaxial_drained/input.txt`

```
Property: G0 125.0
Property: K0 150.0
```

```
Property: Mc 1.25
Property: Lambda 0.37
Property: A0 0.4
Property: Patm 100.0
Property: P_min 0.1
StateVariable: VoidRatio 0.907
StateVariable: StressXX -100.0
StateVariable: StressYY -100.0
StateVariable: StressZZ -100.0
CustomVariable: AlphaXX 0.0
CustomVariable: AlphaYY 0.0
AnalysisType: 2
NumSteps: 300
EpsilonYY: -2.0e-5
```

This packaged case is the monotonic drained Toyoura-like reference used by the mini-result figures.

Monotonic undrained triaxial example Path: `mini_tools/SANISAND/cases/triaxial_undrained/input.txt`

This packaged case uses the same Toyoura-like loose initial state as the drained companion but switches the loading path to undrained triaxial compression.

Constant-p' triaxial example Path: `mini_tools/SANISAND/cases/triaxial_constant_p/input.txt`

```
Property: G0 125.0
Property: K0 150.0
Property: Mc 1.25
Property: Lambda 0.37
Property: A0 0.4
Property: Patm 100.0
Property: P_min 0.1
StateVariable: VoidRatio 0.907
StateVariable: StressXX -100.0
StateVariable: StressYY -100.0
StateVariable: StressZZ -100.0
CustomVariable: AlphaXX 0.0
CustomVariable: AlphaYY 0.0
AnalysisType: 9
NumSteps: 300
EpsilonYY: -2.0e-5
```

This packaged case enforces the initial mean effective stress p' throughout the loading path by solving the radial strain increment each step. The principal stresses are free to evolve as long as their mean remains constant.

Cyclic q-controlled triaxial example Path: `mini_tools/SANISAND/cases/cyclic_q_control/input.txt`

```
Property: G0 125.0
Property: K0 150.0
Property: Mc 1.25
Property: Lambda 0.37
Property: A0 0.4
Property: Patm 100.0
Property: P_min 0.1
StateVariable: VoidRatio 0.75
StateVariable: StressXX -100.0
StateVariable: StressYY -100.0
StateVariable: StressZZ -100.0
CustomVariable: AlphaXX 0.0
CustomVariable: AlphaYY 0.0
AnalysisType: 4
NumSteps: 2000
EpsilonYY: -2.0e-5
QUP: 50.0
QD: -10.0
```

This packaged case is the stress-controlled undrained cyclic reference used for `images/mini_results/sanisand/cyclic_q_control.png`. The driver advances axial strain internally while reversing the path when `q` reaches the prescribed `QUP` and `QD` limits.

Cyclic strain-controlled examples Paths:

- `mini_tools/SANISAND/cases/cyclic_strain_controlled_drained/input.txt`
- `mini_tools/SANISAND/cases/cyclic_strain_controlled_undrained/input.txt`

Drained cyclic input:

```
Property: G0 125.0
Property: K0 150.0
Property: Mc 1.25
Property: Lambda 0.37
Property: A0 0.4
Property: Patm 100.0
```

```
Property: P_min 0.1
StateVariable: VoidRatio 0.75
StateVariable: StressXX -100.0
StateVariable: StressYY -100.0
StateVariable: StressZZ -100.0
CustomVariable: AlphaXX 0.0
CustomVariable: AlphaYY 0.0
AnalysisType: 7
NumSteps: 40
StrainPatternFile: strain_pattern.txt
```

Undrained cyclic input:

```
Property: G0 125.0
Property: K0 150.0
Property: Mc 1.25
Property: Lambda 0.37
Property: A0 0.4
Property: Patm 100.0
Property: P_min 0.1
StateVariable: VoidRatio 0.75
StateVariable: StressXX -100.0
StateVariable: StressYY -100.0
StateVariable: StressZZ -100.0
CustomVariable: AlphaXX 0.0
CustomVariable: AlphaYY 0.0
AnalysisType: 8
NumSteps: 40
StrainPatternFile: strain_pattern.txt
```

These packaged cases are driven by external `strain_pattern.txt` files staged in the same case directories. They are the source for `images/mini_results/sanisand/cyclic_strain_controlled_split.png`, with the drained response plotted separately from the undrained response.

Unsaturated $dq/dp_{net} = 3$ examples Paths:

- `mini_tools/SANISAND/cases/kurnell_const_suction_10050D/input.txt`
- `mini_tools/SANISAND/cases/kurnell_const_suction_5050L/input.txt`
- `mini_tools/SANISAND/cases/kurnell_const_wcontent_100100D/input.txt`
- `mini_tools/SANISAND/cases/kurnell_const_wcontent_50100L/input.txt`

These packaged cases exercise the standalone unsaturated $dq/dp_{net} = 3$ driver branches under either constant suction or constant water content. They use the Kurnell initial states reported in the paper, but the current SANISAND runs shown here are not presented as reproductions of the paper's unsaturated stress-strain curves.

The hydraulic input for the Kurnell cases now uses the paper SWCC form directly through `Property: SWRC_form 1`, together with the branch-specific parameters `VG_nd`, `VG_md`, `VG_nw`, `VG_mw`, `VG_Sra`, `VG_Srw`, `Pd`, `Pw`, and `VG_omega_prime` from the paper tables.

1.13.5 Output files and columns

Each packaged case writes `q_vs_epsilon2.csv`.

Output file	Produced by	Main use
`q_vs_epsilon2.csv`	all cases	Main SANISAND-type history file for monotonic, cyclic, and unsaturated packaged runs.

Primary columns:

Output column	Meaning
`Step`, `Epsilon_2`	Step index and axial strain measure used by the standalone driver.
`p`, `q`, `pnet`	Mean stress, deviatoric stress, and net mean stress.
`suction`, `saturation`, `zeta`, `chi`	Hydraulic/effective-stress state variables written by the standalone driver.
`void_ratio`, `pore_air`, `pore_water`	Volume and pore-pressure state variables.

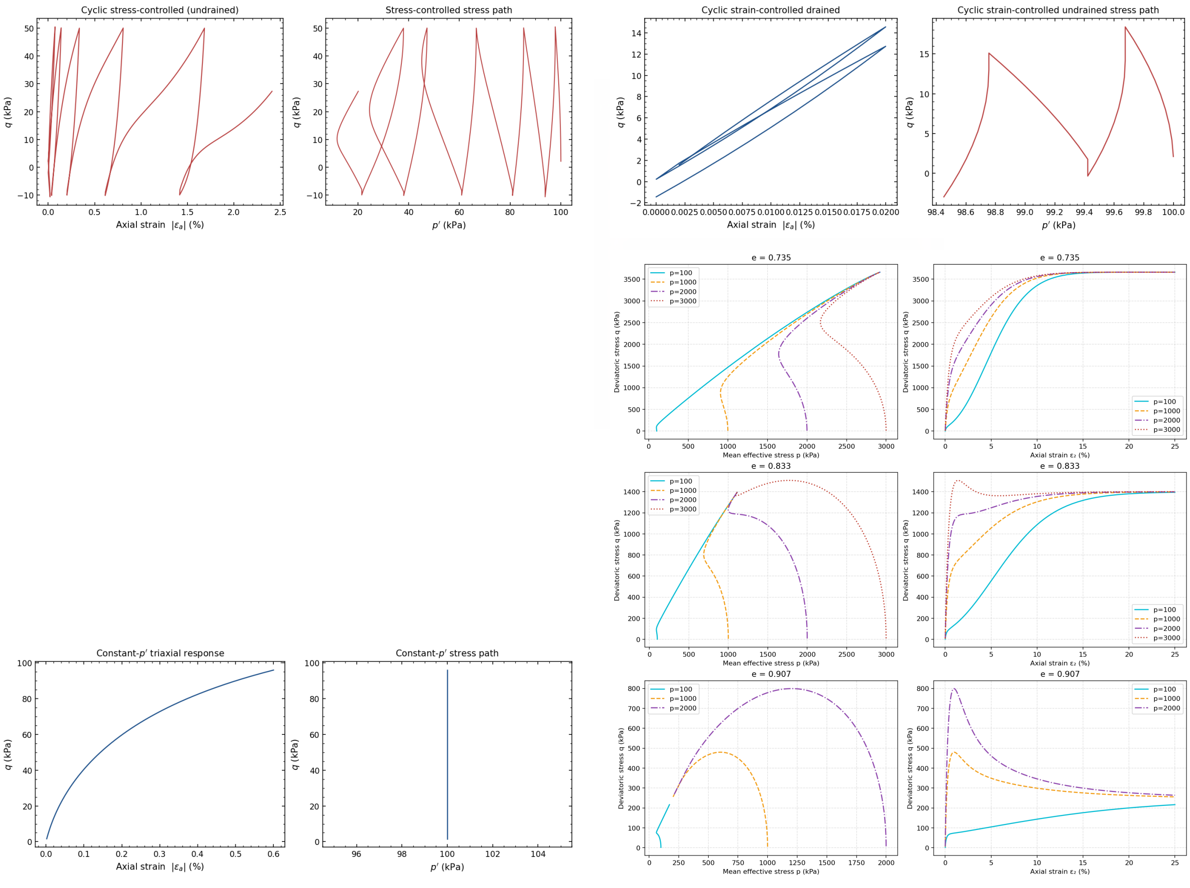
The strain-controlled cyclic cases stage both `input.txt` and `strain_pattern.txt` together. The plots in the next section are generated from these packaged case CSVs.

1.14 Results

The figures below combine the packaged cyclic FALCON mini cases under `mini_tools/SANISAND/cases` with an undrained Toyoura verification bundle (not shipped in the deployment package).

- The cyclic stress-controlled plot uses `cases/cyclic_q_control/input.txt`.
- The cyclic strain-controlled split plot uses `cases/cyclic_strain_controlled_drained/input.txt` and `cases/cyclic_strain_controlled_undrained/input.txt`.
- The constant-p' plot uses `cases/triaxial_constant_p/input.txt`.

- The undrained Toyoura verification plot uses input families generated by `reproduce_toyoura_monotonic.py` (not shipped in the deployment package):
  - $e_0 = 0.735$: `cases/e_0.735/p_100/input.txt`, `cases/e_0.735/p_1000/input.txt`, `cases/e_0.735/p_2000/input.txt`, `cases/e_0.735/p_3000/input.txt`
  - $e_0 = 0.833$: `cases/e_0.833/p_100/input.txt`, `cases/e_0.833/p_1000/input.txt`, `cases/e_0.833/p_2000/input.txt`, `cases/e_0.833/p_3000/input.txt`
  - $e_0 = 0.907$: `cases/e_0.907/p_100/input.txt`, `cases/e_0.907/p_1000/input.txt`, `cases/e_0.907/p_2000/input.txt`



Reading order:

1. Top-left: cyclic q-controlled undrained response from the packaged mini.
2. Top-right: cyclic strain-controlled drained and undrained responses from the packaged strain-history drivers.
3. Bottom-left: packaged constant-p' triaxial response and stress path.
4. Bottom-right: combined undrained Toyoura verification responses from the standalone verification bundle.

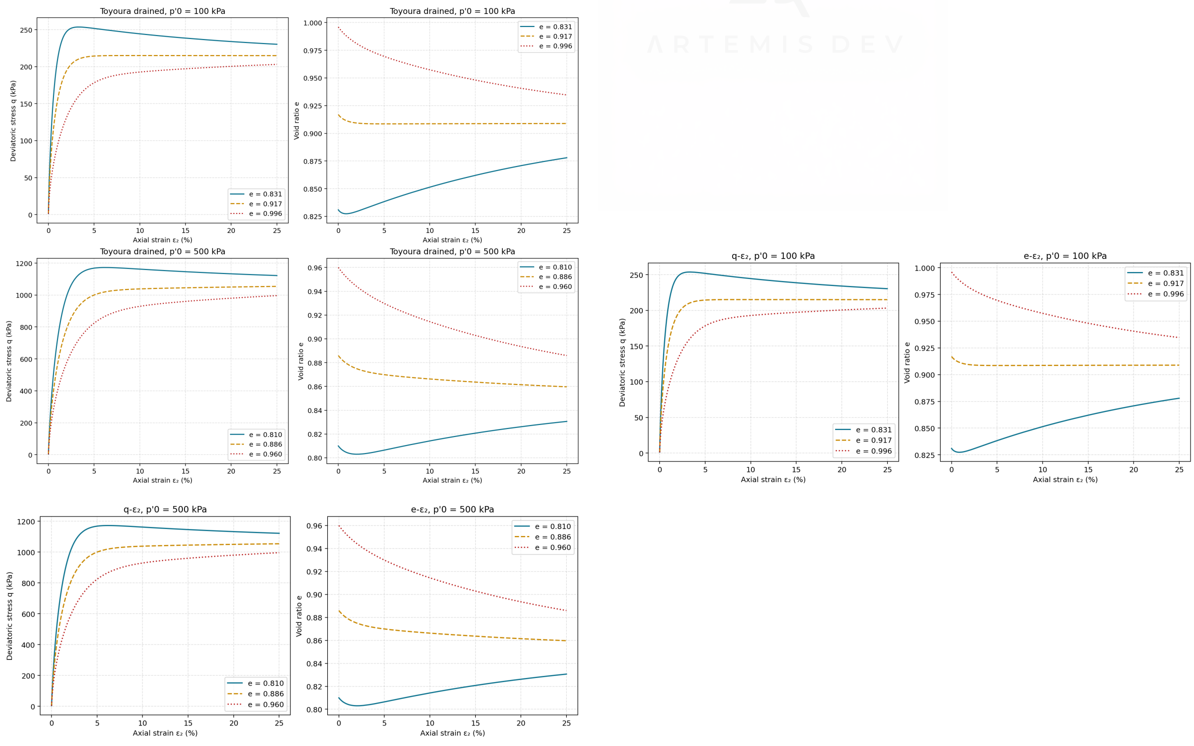
1.14.1 Verification: Toyoura drained suite

The repository also contains a separate drained verification bundle under `SANISAND/verification/toyoura_drained_s00466_020_01945_8` (not shipped in the deployment package). These plots are generated by `reproduce_toyoura_drained_s00466_020_01945_8.py`, which writes six monotonic drained `AnalysisType: 2` cases with `NumSteps: 12500` and `EpsilonYY: -2.0e-5`.

Case groups used by that verification script:

- $p_0' = 100$ kPa: `cases/p_100/e_0.831/input.txt`, `cases/p_100/e_0.917/input.txt`, `cases/p_100/e_0.996/input.txt`
- $p_0' = 500$ kPa: `cases/p_500/e_0.810/input.txt`, `cases/p_500/e_0.886/input.txt`, `cases/p_500/e_0.960/input.txt`

The first figure combines all six drained verification cases. The next two separate the $p_0' = 100$ kPa and 500 kPa groups into $q - \varepsilon_2$ and $e - \varepsilon_2$ panels.



1.14.2 Unsaturated $dq/dp_{net} = 3$ paths

The packaged SANISAND Kurnell examples under `kurnell_const_suction_10050D`, `kurnell_const_suction_5050L`, `kurnell_const_wcontent_100100D`, and `kurnell_const_wcontent_50100L` are kept as raw standalone examples. The plots shown below come from a separate verification bundle (not shipped in the deployment package).

These verification plots use the Kurnell Table 3 / Table 4 / Table 6 data with the standalone-mini mapping actually available in this driver. The wetting SWRC branch is kept parallel and below the drying branch by setting `VG_nw = VG_nd`, `VG_mw = VG_md`, and `Pw = 0.5 Pd`.

The verification bundle now includes the full 8 Table 6 tests: - 4 constant-suction runs with final suffix `-D` - 4 constant-water-content runs with final suffix `-U`

The response plots use the standalone unsaturated driver with $dq/dp_{net} = 3$, $p_{net} = p - \chi s$, and the exported `deviatoric_strain` / `volumetric_strain` measures from the SANISAND mini CSV output. This standalone mini exposes one density-intercept slot (`N_c`) rather than the paper's separate `N_l` and `N_c` pair, so the verification bundle uses a reduced-model fitted intercept in that slot. The initial stress state written into each mini case is the model effective mean stress $p' = p_{net} + \chi s$, so that the requested Table 6 p_{net} is satisfied at step 0. For these runs the net-stress weighting is enforced explicitly as $\chi = S_w$ through `chi_b1 = 1.0` and `chi_b2 = 0.0`; the unsaturated CSL modifier is carried separately by `c1` and `c2`.

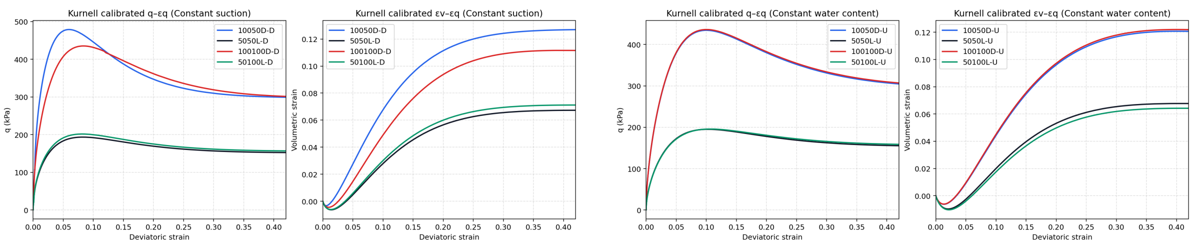
Material parameters actually used for the unsaturated verification plots below are:

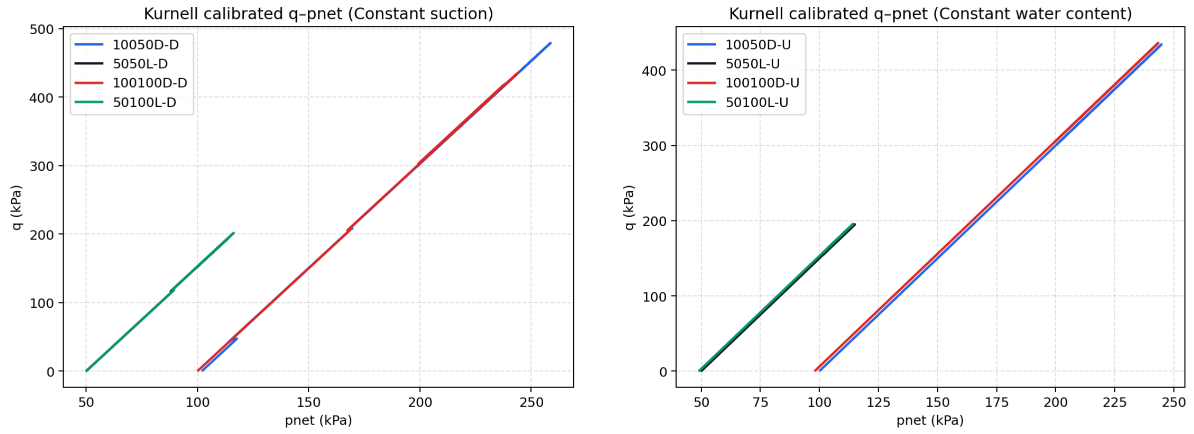
Group	Parameter	Value
Mechanical	`G0`	135.0
Mechanical	`K0`	160.0
Mechanical	`Mc`	1.475
Mechanical	`Me`	1.05
Mechanical	`Lambda`	0.35
Mechanical	`N_c`	14.7 (reduced-model fitted intercept slot)
Mechanical	`alpha_c`	2900.0
Mechanical	`n_b`	1.35
Mechanical	`ch`	0.97
Mechanical	`n_d`	1.25
Mechanical	`h0`	1.2
Mechanical	`A0`	1.15
Hydraulic / coupling	`beta_x1`	0.02 (paper label retained in generated inputs)
Hydraulic / coupling	`beta_x2`	0.05 (paper label retained in generated inputs)
Hydraulic / coupling	`c1`	0.02
Hydraulic / coupling	`c2`	0.05
Hydraulic / coupling	`chi_b1`	1.0 ($\chi = S_w$)
Hydraulic / coupling	`chi_b2`	0.0 ($\chi = S_w$)
Hydraulic / coupling	`SWRC_form`	1
Hydraulic / coupling	`VG_nd`	10.0
Hydraulic / coupling	`VG_md`	1.0
Hydraulic / coupling	`VG_nw`	10.0

Group	Parameter	Value
Hydraulic / coupling	`VG_mw`	1.0
Hydraulic / coupling	`Pd`	5.0
Hydraulic / coupling	`Pw`	2.5
Hydraulic / coupling	`VG_Sra`	1.0
Hydraulic / coupling	`VG_Srw`	0.009
Hydraulic / coupling	`VG_omega_prime`	2.1
Hydraulic / coupling	`b_d`	5.0
Hydraulic / coupling	`b_w`	-5.0
Hydraulic / coupling	`b_sc`	25.0

Initial conditions enforced from Table 6 are listed below; the initial saturation for each case is then obtained from the Table 4 drying main curve at that `e` and `s`.

Test	Initial `p_{net}` (kPa)	Initial `e`	Initial `s` (kPa)
`10050D-D`	102	0.658	51
`5050L-D`	50	0.770	51
`100100D-D`	100	0.687	100
`50100L-D`	50	0.763	100
`10050D-U`	100	0.676	50
`5050L-U`	50	0.771	50
`100100D-U`	98	0.674	100
`50100L-U`	49	0.777	98





1.15 Applications and limitations

- Best suited to monotonic and cyclic effective-stress response of sands, including fabric evolution and density-dependent dilatancy.
- Appropriate for uncoupled and effective-stress-based coupled analyses when the required hydraulic pieces are supplied externally.
- It is not an intrinsic unsaturated constitutive law and is not intended for clay anisotropy or rock-mass strength problems.

1.16 References

1. Been, K., and Jefferies, M. G. (1985). A state parameter for sands. *Géotechnique*, 35(2), 99-112.
2. Dafalias, Y. F., and Manzari, M. T. (2004). Simple plasticity sand model accounting for fabric change effects. *Journal of Engineering Mechanics*, 130(6), 622-634.
3. Ghorbani, J., and Airey, D. W. (2021). Modelling stress-induced anisotropy in multi-phase granular soils. *Computational Mechanics*, 67, 497-521.
4. 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.
5. Sloan, S. W., Abbo, A. J., and Sheng, D. (2001). Refined explicit integration of elastoplastic models with automatic error control. *Engineering Computations*, 18(1/2), 121-154.
6. Taiebat, M., and Dafalias, Y. F. (2008). SANISAND: Simple anisotropic sand plasticity model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32(8), 915-948.