AD FALCON API Manual

Contact Mechanics in FALCON

Javad Ghorbani

March 26, 2026

Contents

1 Contact Mechanics in FALCON

FALCON's contact mechanics cover slave–master contact interfaces for soil–structure interaction in non-coupled, coupled, and fully coupled analyses. In coupled/fully coupled analyses, the master side is treated as single-phase (no PW/PA exchange across the contact interface), while drainage control can be applied on the slave boundary during contact and separation.

1.1 Overview

FALCON's contact mechanics uses a slave–master formulation that:

- Enforces contact constraints between deformable bodies through normal and tangential penalty stiffness
- Uses effective-stress reduction in the friction criterion for 2D/axisymmetric contact
- Allows user control over drainage physics at the slave boundary during contact and separation states
- Supports independent drainage policies for pore water (PW) and pore air (PA) in fully coupled analyses

The implementation is designed for coupled hydro-mechanical and fully coupled multi-phase analyses where pore pressures influence contact behavior.

1.2 Scope and Assumptions

- **Dimensionality**: 2D planar and axisymmetric analyses are supported. 3D contact is not currently available.
- **Algorithm**: Penalty or Augmented Lagrangian (normal direction).
- **Phase treatment**: In coupled/fully coupled analyses, master surfaces must be single-phase (no active PW/PA DOFs). Drainage control applies only to the slave boundary.

1.3 Effective Stress in Frictional Contact

1.3.1 Core Principle

Friction in porous media depends on effective stress, not total stress.

- This effective-stress reduction is applied in the 2D/axisymmetric contact friction criterion.

1.3.2 Computation Steps

At each contact Gauss point:

1. **Interpolate nodal pressures (2D/axisymmetric only)**: If a drainage policy for a phase is specified for the current contact state (see below), nodal PW and/or PA are interpolated to the contact point.
2. **Interpolate effective stress parameter**: Interpolate nodal `Xi` (state variable) to the contact point using the same shape functions. `Xi` is clamped to $[0, 1]$.
3. **Compute effective pore pressure (2D/axisymmetric only)**:
   - If both phases contribute: $p_{\text{eff}} = \Xi \cdot PW + (1 - \Xi) \cdot PA$
   - If only water is active: $p_{\text{eff}} = PW$
   - If only air is active: $p_{\text{eff}} = PA$
   - If no phases are active: $p_{\text{eff}} = 0$
4. **Compute effective normal traction**:

$$t_{n,\text{eff}} = \max(0, |t_n| - p_{\text{eff}})$$

   where $|t_n|$ is the magnitude of the penalty normal traction.
5. **Apply friction criterion**:
   - **Stick condition**: $|t_{t,\text{trial}}| \leq \mu \cdot t_{n,\text{eff}} + \text{tol}$
   - **Slip condition**: $|t_t| = \mu \cdot t_{n,\text{eff}}$

The slip-direction search also uses $t_{n,\text{eff}}$ to maintain consistency throughout the contact algorithm.

1.4 Drainage Policies on the Slave Boundary

FALCON allows you to specify how pore pressures behave at the slave boundary when contact is established and when contact separates. Policies can be set independently for:

- Contact state: "in contact" vs. "separation"
- Fluid phase: pore water (PW) and pore air (PA)

1.4.1 Available Drainage Modes

Mode	Description
Open	Dirichlet pressure set to 0 (atmospheric / open boundary).
Pressure v	Dirichlet pressure with explicit value v. Recommended for any known pressure, including $v = 0$.

Mode	Description
Closed	No Dirichlet; DOF is active (released). Leads to natural boundary behavior (zero normal discharge by default).
Discharge q	Parsed as an option, but not yet enforced as a boundary type by the contact drainage mechanism.

1.4.2 Behavior

- Drainage policies are evaluated and applied before assembling boundary conditions at each substep.
- The contact logic determines which slave segments are "in contact" or "separated" and applies the corresponding policy to the involved slave nodes.
- Use `Pressure v` to specify any Dirichlet value (e.g., atmospheric open boundary is `Pressure 0.0`).
- `Closed` means the pressure DOF is an unknown (released) and, absent an explicit flux boundary, the FE weak form imposes a natural no-flow condition.
- Contact policies overwrite the slave-node pressure boundary state while active. When contact does not assert a policy, user- or step-scheduled BCs apply.

1.5 Input Format

Contact pairs are defined in the input file between the `% ContactPairs` header and the terminating `%%%` line.

1.5.1 Multiple contact pairs

- Define multiple contact pairs by repeating `@ContactPair <ID> ...` blocks inside the same `% ContactPairs` section.
- Each block ends when the next `@ContactPair ...` begins or when the section ends (`%%%` or the next `% ...` section).

1.5.2 Basic Contact Pair Definition

```
@ContactPair <ID>
@@MasterNodes           <node_id_1> <node_id_2> ...
@@SlaveNodes            <node_id_1> <node_id_2> ...
@@OrderOfContact        <integer>
@@NumGaussPoints        <integer>
@@PenaltyCoefficientNormal   <double>
@@PenaltyCoefficientTraction <double>
@@Friction              <double>
```

```
[@@TLOPEN               <double>]  (optional, default = 0.0)
[@@TLOUTS               <double>]  (optional, default = 0.0)
@@InitiationStepId      <integer>
@@TerminationStepId     <integer>  (optional, default = -1, never ends)

@@Formulation           <Penalty|AugmentedLagrangian|AL>   (optional,
default = Penalty)
@@AlphaNormal           <double>   (AL only, default = 1.0)
@@ALMaxIterations       <integer>  (AL only, default = 0)
@@ALUpdateTolerance     <double>   (AL only, default = 0.0)
```

2D node ordering requirement: For planar/axisymmetric contact, list `@@SlaveNodes` in **clockwise** order along the interface, and list `@@MasterNodes` in **counterclockwise** order.

1.5.3 Drainage Policy Specification

Add drainage policies for contact and separation states using:

```
@@DrainageOnContact     <PW|PA> <Open|Closed|Pressure|Discharge> [value]
@@DrainageOnSeparation  <PW|PA> <Open|Closed|Pressure|Discharge> [value]
```

- **Phase**: `PW` (pore water) or `PA` (pore air)
- **Mode**: `Open`, `Closed`, `Pressure`, or `Discharge`
- **Value**: Optional. For `Pressure`, omitting the value defaults to `0.0`. (`Discharge` is accepted but currently not enforced.)

  Multiple lines can be used to specify policies for both phases.

1.5.4 Example

```
% ContactPairs
@ContactPair 1
@@MasterNodes           501 502 503 504 505 506
@@SlaveNodes            101 102 103 104 105 106
@@OrderOfContact        2
@@NumGaussPoints        30
@@PenaltyCoefficientNormal   1.0e6
@@PenaltyCoefficientTraction 1.0e6
@@Friction              0.3
@@InitiationStepId      1
@@TerminationStepId     -1

@@DrainageOnContact     PW Closed
```

```
@@DrainageOnContact     PA Open

@@DrainageOnSeparation  PW Pressure 0.0
@@DrainageOnSeparation  PA Open
%%%
```

Interpretation: - **During contact**: Pore water is closed (no flow), pore air is open (zero pressure). - **During separation**: Pore water pressure is enforced to 0.0, pore air is open (zero pressure).

1.6 Contact Formulation (Penalty vs Augmented Lagrangian)

FALCON supports two formulations for normal contact enforcement:

- **Penalty**: $t_n = k_n g_n$ (compressive-only), where g_n is the normal gap (penetration is typically negative).
- **Augmented Lagrangian (AL, normal direction)**: adds a normal multiplier λ_n and updates it after converged solves:

$$\lambda_n^{(i+1)} = \min\left(0,\ \lambda_n^{(i)} + \alpha_n\ k_n\ g_n\right)$$

This improves robustness compared to pure penalty in some problems, while still keeping a penalty stiffness k_n.

AL inputs (per contact pair):

- `@@Formulation`: `Penalty` (default) or `AugmentedLagrangian`/`AL`.
- `@@AlphaNormal`: $\alpha_n > 0$, multiplier update scale (default `1.0`).
- `@@ALMaxIterations`: outer restart count for AL (default `0`). If `> 0`, the solver may repeat equilibrium solves within a time step until multipliers stabilize (bounded by this value).
- `@@ALUpdateTolerance`: absolute threshold on $|\Delta\lambda_n|$ used to decide if multipliers have "changed" (default `0.0`).

Notes:

- Only the **normal** direction uses AL; tangential behavior is still controlled by tangential penalty + Coulomb friction.
- `@@ALAlphaTangential` is accepted as a keyword but tangential AL is not implemented.

1.7 Time Integration Support Matrix

Augmented-Lagrangian contact requires multiplier updates coupled to equilibrium iterations. As a result:

TimeIntegration	Penalty formulation	Augmented Lagrangian formulation
`Implicit`	Supported	Supported
`IMEX`	Supported	Not supported
`Explicit`	Supported	Not supported

1.8 Parameter Descriptions

Parameter	Type	Description	Default (if omitted)	Validation
`@@MasterNodes`	List	Node IDs forming the master surface	Required	Non-empty, valid node IDs
`@@SlaveNodes`	List	Node IDs forming the slave surface	Required	Non-empty, valid node IDs
`@@OrderOf Contact`	Integer	Polynomial order of contact shape functions	`2`	`> 0`
`@@NumGauss Points`	Integer	Gauss integration points per contact segment	`30`	`> 0`
`@@Penalty Coefficient Normal`	Double	Penalty stiffness for normal direction	`1e5`	`> 0`
`@@Penalty Coefficient Traction`	Double	Penalty stiffness for tangential direction	`1e5`	`> 0`
`@@Friction`	Double	Friction coefficient μ	`0.0`	`>= 0`
`@@TLOPEN`	Double	Contact “enter” threshold (penetration) used for contact hysteresis	`0.0`	(recommended) `<= 0`

Parameter	Type	Description	Default (if omitted)	Validation
@@TLOUTS	Double	Contact “exit” threshold (penetration) used for contact hysteresis	0.0	(recommended) <= 0
@@Initiation StepId	Integer	Simulation step when contact becomes active	0	>= 0
@@Termination StepId	Integer	Simulation step when contact ends (-1 means “never ends”)	-1	>= Initiation StepId or -1
@@Formulation	String	Contact enforcement formulation (Penalty or Augmented Lagrangian/AL)	Penalty	Recognized value
@@AlphaNormal	Double	AL normal multiplier update scale α_n (AL only)	1.0	> 0 (AL only)
@@ALMax Iterations	Integer	Outer AL restart limit (0 disables outer restarts) (AL only)	0	>= 0 (AL only)
@@ALUpdate Tolerance	Double	Threshold on multiplier change $\lvert\Delta\lambda$_n$\rvert$ (AL only)	0.0	>= 0 (AL only)
@@DrainageOn Contact	String	Drainage policy for slave boundary during contact (phase + mode + optional value)	None (for both PW/PA)	Valid phase/-mode/value

Parameter	Type	Description	Default (if omitted)	Validation
`@@DrainageOnSeparation`	String	Drainage policy for slave boundary during separation (phase + mode + optional value)	`None` (for both PW/PA)	Valid phase/-mode/value

1.9 Coupled and Fully Coupled Analysis Workflow

1. **Contact detection**: At each iteration, the solver updates contact node coordinates and evaluates contact segments to determine which Gauss points are "in contact" or "separated."
2. **Drainage policy application**: Before assembling body and stress boundaries for the substep, drainage policies are applied to slave nodes based on their contact state.
3. **DOF management**: If a drainage policy changes a pressure DOF from free to Dirichlet (or vice versa), the system renumbers internal structures while preserving nodal state.
4. **Effective stress evaluation**: For 2D/axisymmetric contact, when a drainage policy specifies a phase for the current contact state, effective pore pressure p_{eff} is computed from nodal PW, PA, and `Xi` and used in the friction criterion.
5. **Force computation**: Normal and tangential contact forces are computed using effective normal traction $t_{n,\text{eff}}$ in the friction criterion.
6. **Assembly and solution**: Contact forces contribute to the residual and tangent stiffness, and the Newton iteration proceeds.

1.10 Limitations and Current Caveats

- **3D contact**: Not supported in this implementation.
- **Penalty tangent**: The slip/stick logic uses p_{eff} consistently, but the contact stiffness matrix does not yet include explicit derivatives of p_{eff}.
- **Master phase treatment**: Masters are treated as single-phase; drainage control applies only to the slave boundary.

1.11 Checklist Before Running a Contact Analysis

- ☐ Define `@@DrainageOnContact` and `@@DrainageOnSeparation` for each contact pair as needed.
- ☐ Verify that pore water and/or pore air DOFs exist and are active on slave nodes where drainage policies will be applied.
- ☐ Check that penalty coefficients are appropriately scaled for your problem (typically on the order of the material stiffness or higher).

1.12 Troubleshooting

1.12.1 User Dirichlet boundary condition not taking effect

- **Cause**: Contact drainage policy may be toggling the DOF unexpectedly, or user BC is not explicitly set.
- **Solution**: The contact helper respects explicit user Dirichlet conditions. Verify that the node's DOF has an explicit user condition. Check the contact state (in contact vs. separation) and corresponding drainage policy.

1.12.2 Contact forces seem too weak or too strong

- **Cause**: Penalty coefficients may be poorly scaled.
- **Solution**: Adjust `PenaltyCoefficientNormal` and `PenaltyCoefficientTraction` to match the stiffness scale of your materials. Start with values on the order of 10^3 to 10^6 times the material modulus.

1.13 MasterForceContact Output

1.13.1 Overview

The `% MasterForceContact` section enables monitoring and logging of contact forces, rigid body reactions, velocities, accelerations, and prescribed forces during simulation. This feature is particularly useful for:

- Validating contact algorithm behavior
- Debugging soil-structure interaction
- Tracking rigid body dynamics in coupled analyses
- Verifying force equilibrium in static and dynamic cases

1.13.2 Input Format

```
% MasterForceContact
@Id <unique_id>
@ContactID <contact_pair_id>
@Step <step_number>              (single step, OR use @Steps)
@Steps <step1> <step2> ...       (multiple steps, OR use @Step)
@OutputFile <path_to_csv>
@OutputFreq <frequency>
%%%
```

1.13.3 Parameters

Parameter	Type	Description	Required
`@Id`	Integer	Unique identifier for this output block	Yes
`@ContactID`	Integer	ID of the contact pair to monitor (must exist in `% ContactPairs`)	Yes
`@Step`	Integer	Single simulation step at which to sample output	No*
`@Steps`	List	Multiple simulation steps to sample (alternative to `@Step`)	No*
`@OutputFile`	String	Path to CSV output file (created/truncated at parse time)	Yes
`@OutputFreq`	Integer	Output frequency: write every Nth occurrence (default = 1)	Yes

*Either `@Step` or `@Steps` must be provided.

1.13.4 CSV Output Columns

The generated CSV file contains the following columns:

Column	Description
`StepID`	Current simulation step ID
`SimulationTime`	Cumulative simulation time
`RbRx`	Rigid body reaction force in X direction
`RbRy`	Rigid body reaction force in Y direction
`RbRz`	Rigid body reaction force in Z direction
`RbVx`	Rigid body velocity in X direction (dynamic integration)
`RbVy`	Rigid body velocity in Y direction (dynamic integration)

Column	Description
`RbVz`	Rigid body velocity in Z direction (dynamic integration)
`RbAx`	Rigid body acceleration in X direction (dynamic integration)
`RbAy`	Rigid body acceleration in Y direction (dynamic integration)
`RbAz`	Rigid body acceleration in Z direction (dynamic integration)
`PrescribedFx`	Prescribed/applied force in X direction (time-scaled via Load Application)
`PrescribedFy`	Prescribed/applied force in Y direction (time-scaled via Load Application)
`PrescribedFz`	Prescribed/applied force in Z direction (time-scaled via Load Application)

1.13.5 Behavior

1. **Step gating**: Output is written only at the specified step(s). Use `@Steps` for multi-step monitoring.
2. **Frequency control**: `@OutputFreq` controls how often to write within active steps (e.g., `@OutputFreq 10` writes every 10th substep).
3. **Rigid body matching**: The writer identifies the rigid body by finding the first rigid body that shares at least one node with the contact pair's master nodes.
4. **Force calculation**:
   - Reactions (`RbRx/Ry/Rz`) are accumulated from `mesh.FR` for all nodes in the rigid body.
   - Prescribed forces (`PrescribedFx/Fy/Fz`) are computed from active `RigidMotionConstraint` blocks, scaled by their `LoadApplication` time variation (e.g., Ramp, Sinusoidal).
5. **Dynamic quantities**: Velocity and acceleration fields are populated when using `Dynamic` or `Consolidation` simulation modes with rigid body force control.

1.13.6 Example: Single-Step Output

```
% MasterForceContact
@Id 1
@ContactID 1
@Step 2
@OutputFile /path/to/contact_forces.csv
@OutputFreq 1
%%%
```

Interpretation: Monitor contact pair 1 during step 2, writing output every substep.

1.13.7 Example: Multi-Step Output

```
% MasterForceContact
@Id 1
@ContactID 1
@Steps 1 2 3 4 5
@OutputFile /path/to/contact_forces.csv
@OutputFreq 5
%%%
```

Interpretation: Monitor contact pair 1 during steps 1-5, writing output every 5th substep.

1.13.8 Validation

- `@ContactID` must reference an existing contact pair defined in `% ContactPairs`.
- All `@Step` or `@Steps` values must reference existing simulation steps.
- The output file path is validated and created at parse time; parent directories are created if needed.
- `@Id` must be unique across all `MasterForceContact` blocks.

1.13.9 Troubleshooting

No output written

- **Cause**: The specified contact pair may not be active at the given step, or no master nodes are in contact.
- **Solution**: With the updated implementation, output is always written (zeros if no contact). Verify step gating and contact pair initiation/termination steps.

Prescribed forces show zero

- **Cause**: No `RigidMotionConstraint` with force specification is active for the matched rigid body at the current step.
- **Solution**: Verify `@@StepIds` and `@@ForcePropagateStepsX/Y/Z` in your rigid motion constraint definition.

CSV file is empty

- **Cause**: Parse-time file creation succeeded but runtime step gating prevented any writes.
- **Solution**: Check that `@Step` or `@Steps` match actual simulation steps being executed.

1.14 Summary

FALCON's contact mechanics provides a robust framework for modeling contact and friction in coupled porous media analyses. Key features include:

- Effective stress-based friction that accounts for pore pressures
- User-controlled drainage policies for slave boundaries during contact and separation
- Independent treatment of pore water and pore air phases
- Consistent nodal interpolation of state variables at contact Gauss points
- Comprehensive force monitoring via `MasterForceContact` output with rigid body tracking

By carefully specifying drainage policies and ensuring proper initialization of state variables, you can model complex contact scenarios in saturated, unsaturated, and multiphase geotechnical problems.