AD FALCON API Manual

Infinite Elements

Javad Ghorbani

March 26, 2026

Contents

1 Infinite Elements

This guide explains the **Infinite Elements** section of the input file, including theory, syntax, supported types, required fields, and examples.

Infinite elements provide **far-field truncation**. In 2D, FALCON uses mapped infinite elements (geometric mapping extending to infinity). In 3D, the current `TET10` option generates a conforming **finite buffer layer** that pushes the truncation boundary outward.

1.1 Theory

1.1.1 Mapped Infinite Element Formulation

!!! note "3D support" The mapped-to-infinity formulation below applies to the 2D infinite elements (`TRI*`/`QUAD*`). For 3D (`@@Type: TET10`), FALCON currently generates a finite far-field buffer layer (standard `TET10` elements extruded along the selected side normal), not a mapped infinite element.

FALCON uses **mapped (geometric) infinite elements** where:

1. **Field Interpolation**: Standard element shape functions $N_i(\boldsymbol{\xi})$
   - Displacements: $\mathbf{u}(\boldsymbol{\xi}) = \sum_i N_i(\boldsymbol{\xi})\mathbf{u}_i$
   - Pressures: $p(\boldsymbol{\xi}) = \sum_i N_i(\boldsymbol{\xi})p_i$
2. **Geometry Mapping**: Modified mapping $\mathbf{x}(\boldsymbol{\xi})$ that extends to infinity
   - Jacobian: $\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}$ computed from mapped geometry
   - Derivatives: $\frac{\partial N_i}{\partial \mathbf{x}} = \mathbf{J}^{-1}\frac{\partial N_i}{\partial \boldsymbol{\xi}}$

1.1.2 Geometric Mappings

2D Quadrilaterals (QUAD4 / QUAD8) Node ordering: 0 (TR), 1 (TL), 2 (BL), 3 (BR) for QUAD4; corners 0,2,4,6 and mid-sides 1,3,5,7 for QUAD8.

Stretching function:

$$s(\eta) = \frac{1+\eta}{1-\eta}, \qquad s(-1) = 0, \quad s \to \infty \text{ as } \eta \to 1$$

Mapping (interface at $\eta = -1$, direction edge at $\eta = +1$):

$$\mathbf{x}(\xi,\eta) = \mathbf{x}_0(\xi) + s(\eta)\big(\mathbf{x}_1(\xi) - \mathbf{x}_0(\xi)\big)$$

2D Triangles (TRI3 / TRI6) Node ordering: 0 (direction node at $\xi = 1$), 1 and 2 (interface nodes at $\xi = 0$).

Stretching function:

$$s(\xi) = \frac{\xi}{1-\xi}, \qquad t = \frac{\eta}{1-\xi}, \qquad s \to \infty \text{ as } \xi \to 1$$

Mapping (interface at $\xi = 0$):

$$\mathbf{x}(\xi, \eta) = \mathbf{x}_0(t) + s(\xi)\big(\mathbf{x}_C - \mathbf{x}_0(t)\big)$$

where $\mathbf{x}_0(t)$ is a point on the interface edge and $\mathbf{x}_C$ is the direction node.

3D Far-Field Buffer Layer (TET10) For 3D analyses, the current `@@Type: TET10` option generates a **single conforming outer layer** of standard `TET10` elements:

- Boundary faces on the selected side (via `@@Sides`) are detected from the existing finite mesh.
- Those faces are extruded outward by `@@RayLength` along the outward normal to create an outer surface.
- The resulting prism region is split into `TET10` tetrahedra forming a finite buffer layer.

You can optionally apply displacement/pressure Dirichlet conditions on the far-plane nodes via the *Infinity BC directives* (see below).

1.1.3 Interface Consistency

For 2D mapped infinite elements, at the interface ($s = 0$) the mapping gives $\mathbf{x} = \mathbf{x}_0$, ensuring exact geometric continuity with the adjacent finite mesh. Field continuity is enforced by sharing nodes. For the 3D buffer layer, continuity is achieved by generating a conforming outer layer that shares the boundary-face nodes with the interior mesh.

1.2 Syntax

1.2.1 Section Header

FALCON treats section names as case-insensitive and whitespace-insensitive:

```
% InfiniteElements
% Infinite Elements
% Infinite-Elements
```

1.2.2 Section Format

The section contains one or more boundary blocks. Each block defines infinite elements on a boundary:

2D Format:

```
@Boundary <Name>
  @@Type: TRI3 | TRI6 | QUAD4 | QUAD8
  @@Material: <MaterialID>
  @@Nodes: <node1> <node2> ... <nodeN>
  @@Edges: <n1> <n2> <n3> <n4> ...
  @@Topology: Auto | Wedge | Strip
  @@ExteriorPoint: <x> <y>
  @@InteriorPoint: <x> <y>
  @@RayLength: <length>
  @@ApplyInfinityBC: Yes | No
  @@ApplyInfinityBCPW: Yes | No
  @@DisX: <value>
  @@DisY: <value>
  @@PW: <value>
```

3D Format:

```
@Boundary <Name>
  @@Type: TET10
  @@Material: <MaterialID>
  @@Sides: XMin | XMax | YMin | YMax | ZMin | ZMax
  # NOTE: one side per @Boundary block
  @@Tolerance: <tolerance>
  @@RayLength: <length>
  @@ApplyInfinityBC: Yes | No
  @@ApplyInfinityBCPW: Yes | No
  @@ApplyInfinityBCPA: Yes | No
  @@DisX: <value>
  @@DisY: <value>
  @@DisZ: <value>
  @@PW: <value>
  @@PA: <value>
```

Notes:

- Lines starting with @ are parsed; others are ignored (allowing comments).
- Keys are case-insensitive.
- One or more leading @ characters accepted (e.g., `@Type:` and `@@Type:` are equivalent).
- Colon `:` after keys is optional.
- For backward compatibility, a single implicit block without `@Boundary` is supported.

1.3 Section Placement

Begin the section with:

```
% Infinite Elements
```

End with:

```
%%%
```

Place `% Infinite Elements` **before** `% BoundaryConditions` in the input file.

1.4 Input Format

1.4.1 Boundary Block Marker

`@Boundary <Name>` starts one generator block. `<Name>` is an arbitrary label (used only for readability in the input file).

1.4.2 Directive Summary

All directives below are written on lines that start with `@` (for example, `@@Type: ...`). Keys are case-insensitive and the trailing `:` is optional.

Directive	Applies to	Required?	Default	Meaning
`@Boundary <Name>`	2D / 3D	No	—	Starts a new boundary generator block. `<Name>` is a label only.
`@@Type`	2D / 3D	2D: No; 3D: Yes	`TRI3`	Element type to generate. Accepted: `TRI3/N3`, `TRI6/N6`, `QUAD4/N4`, `QUAD8/N8`, `TET10/ITET10/INF_TET10`.

Directive	Applies to	Required?	Default	Meaning
`@@Material`	2D / 3D	Yes	—	Material ID/name assigned to the generated elements (must exist in `%Materials`).
`@@Nodes`	2D	One of `@@Nodes`/`@@Edges`	—	Ordered boundary polyline (adjacent pairs define segments). If the first node is repeated at the end, the polyline is treated as closed (the repeated last node is ignored). For `TRI6`/`QUAD8`, list corner nodes only (mid-side nodes are inferred from adjacent interior elements).
`@@Edges`	2D	One of `@@Nodes`/`@@Edges`	—	Explicit boundary segments given as pairs: `n1 n2 n3 n4 ...` (even count). Each pair defines one segment.

Directive	Applies to	Required?	Default	Meaning
`@@Topology`	2D (`TRI3`/`TRI6`)	No	`Auto`	Topology rule for triangle-based generators: `Auto`, `Wedge`, `Strip`. `Strip` generates quads for coverage on (nearly) straight boundaries; `Wedge` uses a shared direction node; `Auto` selects based on boundary straightness.
`@@ExteriorPoint`	2D	No	—	A point known to lie outside the domain; helps determine the outward direction.
`@@InteriorPoint`	2D	No	—	A point known to lie inside the domain; helps determine the outward direction.

Directive	Applies to	Required?	Default	Meaning
`@@RayLength`	2D / 3D	No	auto	Distance from the interface to the far field (direction nodes in 2D; buffer-layer thickness in 3D). Default (2D): ≈ `2.5×` model extent normal to the boundary (fallback: average segment length). Default (3D): `2.5×` model extent along the selected side normal.
`@@Sides`	3D (`TET10`)	Yes	—	Which outer face to extend: `XMin`, `XMax`, `YMin`, `YMax`, `ZMin`, `ZMax`. Current support: exactly **one** side per block (token `ALL` is accepted by the parser but is invalid for `TET10`).

Directive	Applies to	Required?	Default	Meaning
`@@Tolerance`	3D (`TET10`)	No	`1e-6×max(extent, 1)`	Tolerance for selecting boundary faces near the requested side plane.
`@@FarFieldShape`	3D (`TET10`)	No	(ignored)	Accepted values include `Plane`, `Box`, `Cubic`, `Pole`, `Point`, `Apex`. For `TET10`, this is currently ignored (3D always generates a planar outer layer).
`@@ApplyInfinityBC`	2D / 3D	No	`Yes`	If `Yes`, restrains far-field displacement DOFs (either from explicit `@@Dis*` values or via default behavior when no `@@Dis*` are provided).

Directive	Applies to	Required?	Default	Meaning
`@@DisX`, `@@DisY`, `@@DisZ`	2D / 3D	No	`0` (if applied)	Prescribed displacement value(s) on the far field. If `@@ApplyInfinityBC=Yes` and no `@@Dis*` are provided, FALCON applies a roller-style restraint (dominant normal translation fixed to `0`).
`@@ApplyInfinityBCPW`	Coupled / Fully coupled	No	`Yes`	If `Yes`, restrains far-field `PW` (only when the analysis has a `PW` DOF).
`@@PW`	Coupled / Fully coupled	No	`0`	Value used when restraining `PW`.
`@@ApplyInfinityBCPA`	Fully coupled	No	`Yes`	If `Yes`, restrains far-field `PA` (only when the analysis has a `PA` DOF).
`@@PA`	Fully coupled	No	`0`	Value used when restraining `PA`.

1.4.3 Required Directives (2D)

Provide:

- `@@Material: <MaterialID>`
- A boundary definition using **either**:

  - `@@Nodes: <n1> <n2> ... <nN>` (ordered polyline, minimum 2 nodes), **or**
  - `@@Edges: <n1> <n2> <n3> <n4> ...` (explicit pairs; each pair defines one boundary segment)

`@@Type` is strongly recommended:

- If omitted, it defaults to `TRI3`.
- In coupled / fully coupled analyses, `TRI3` infinite elements are not supported, so you should explicitly choose `TRI6`, `QUAD4`, or `QUAD8`.

Notes:

- For `TRI6` / `QUAD8`: list **corner nodes only**. The generator finds mid-side nodes from the adjacent interior elements.
- If you write `@@Nodes: a b` (two nodes), the generator creates a **single** infinite element on that one edge.
- `@@Edges` is the "single-element / per-segment" form of the same generator. It lets you reproduce what a long `@@Nodes` polyline would do, but segment-by-segment (and without requiring a globally ordered polyline).

1.4.4 Required Directives (3D)

Provide:

- `@@Type: TET10`
- `@@Material: <MaterialID>`
- `@@Sides: XMin | XMax | YMin | YMax | ZMin | ZMax`

Notes:

- Only one `@@Sides` value is supported per `@Boundary` block (use multiple blocks for multiple faces).

1.4.5 Optional Directives

`@@Topology: Auto | Wedge | Strip` (2D only; relevant for `TRI3`/`TRI6`, default: `Auto`)

- `Strip`: generates mapped infinite **quads** with (approximately) parallel rays. This is the recommended choice for long, straight boundaries.
- `Wedge`: generates mapped infinite **triangles** that share a single "direction" node per block (sector-like far-field).
- `Auto`: uses `Strip` when the boundary is nearly straight (or when you use `@@Edges`), otherwise uses `Wedge`.

Note: when `@@Topology: Strip` is used with `@@Type: TRI3`/`TRI6`, the generator uses `QUAD4`/`QUAD8` internally to avoid exterior gaps.

`@@ExteriorPoint: <x> <y>` / **`@@InteriorPoint: <x> <y>`** (2D)

Hints to determine which side of the boundary is "outside". Use these when the outward direction cannot be inferred robustly.

`@@RayLength: <length>`

Distance from the interface to the far field (direction nodes in 2D; buffer-layer thickness in 3D).

`@@Tolerance: <tol>` (3D only)

Tolerance used to select boundary faces near the requested side plane.

Infinity BC directives

- `@@ApplyInfinityBC: Yes|No` with `@@DisX`, `@@DisY`, `@@DisZ`
- `@@ApplyInfinityBCPW: Yes|No` with `@@PW` (coupled/fully coupled)
- `@@ApplyInfinityBCPA: Yes|No` with `@@PA` (fully coupled)

Defaults and interpretation:

- `@@ApplyInfinityBC` defaults to `Yes`. If you set it to `Yes` but do not provide any of `@@DisX`/`@@DisY`/`@@DisZ`, FALCON applies a **roller-style** constraint at the far field (only the dominant normal translation is fixed to 0). If the outward normal cannot be inferred, all translations are fixed to 0.
- `@@ApplyInfinityBCPW` defaults to `Yes` (only relevant for coupled/fully coupled). If `@@PW` is omitted, the restrained value defaults to `0`.
- `@@ApplyInfinityBCPA` defaults to `Yes` (only relevant for fully coupled). If `@@PA` is omitted, the restrained value defaults to `0`.

1.5 Supported Element Types

1.5.1 2D Elements

- **`TRI3`**: 3-node linear triangle (uncoupled only; wedge mapping)
- **`TRI6`**: 6-node quadratic triangle (uncoupled / coupled / fully coupled)
- **`QUAD4`**: 4-node bilinear quadrilateral (uncoupled / coupled / fully coupled)
- **`QUAD8`**: 8-node serendipity quadrilateral (uncoupled / coupled / fully coupled)

1.5.2 3D Elements

- **`TET10`**: 10-node quadratic tetrahedron buffer layer (uncoupled / coupled / fully coupled)

1.5.3 Current Limitations

- 3D generator targets axis-aligned exterior faces using `@@Sides` directive.
- 3D `TET10` currently generates a **finite buffer layer** (not a mapped-to-infinity element).
- TRI3 supported in uncoupled analysis only.

1.6 Topology Selection Guidelines

1.6.1 Strip Topology (2D)

- Generates QUAD4 or QUAD8 elements regardless of `@@Type`.
- Rays extend approximately parallel from interface.
- **Use for**: Straight boundaries representing semi-infinite half-spaces.
- **Advantage**: Provides complete coverage of exterior half-space.

1.6.2 Wedge Topology (2D)

- Generates TRI3 or TRI6 elements.
- Rays converge to direction nodes (sector geometry).
- **Use for**: Corners or intentional sector-type far-fields.
- **Warning**: On straight boundaries, wedge topology may leave uncovered exterior regions (voids).

1.6.3 Auto Topology (2D)

- Default behavior for TRI3/TRI6.
- If boundary polyline is nearly straight (within tolerance), automatically switches to Strip.
- **Recommendation**: Use `Auto` unless specific topology is required.

1.7 Ray Length Selection

The `@@RayLength` parameter controls the distance from the interface to the far field (direction nodes in 2D, buffer-layer thickness in 3D).

1.7.1 Default Behavior

When omitted, FALCON uses:

$$\texttt{RayLength} \approx 2.5 \times (\text{characteristic domain dimension normal to boundary})$$

1.7.2 Application-Specific Guidelines

Static Problems

- Element stiffness can be sensitive to ray length.
- Recommend testing with factors 1×, 2×, 5× of characteristic domain size.
- Longer rays generally reduce spurious boundary effects.

Dynamic Problems

- Ray length affects element size and wave propagation characteristics.
- Ensure adequate mesh resolution (elements per wavelength).

- Note: Infinite elements do not absorb waves; use PML for wave absorption.

Coupled Consolidation Problems

- Longer rays improve far-field drainage representation.
- Consider ray length $\geq 5\times$ typical consolidation domain size.

1.8 Examples

1.8.1 Example 1: 2D Strip Topology (Half-Space Truncation)

```
% Infinite Elements
@Boundary RightBoundary
  @@Type: QUAD8
  @@Material: Elastic1
  @@Nodes: 101 102 103 104 105
  @@Topology: Strip
  @@ExteriorPoint: 100.0 0.0
  @@RayLength: 50.0
  @@ApplyInfinityBC: Yes
  @@DisX: 0.0
  @@DisY: 0.0
  @@PW: 0.0
%%%
```

1.8.2 Example 2: 2D Wedge Topology (Corner)

```
% Infinite Elements
@Boundary Corner
  @@Type: TRI6
  @@Material: Soil1
  @@Nodes: 201 202 203
  @@Topology: Wedge
  @@RayLength: 30.0
  @@ApplyInfinityBC: Yes
  @@DisX: 0.0
  @@DisY: 0.0
%%%
```

1.8.3 Example 3: 3D Infinite Elements (Single Side)

```
% Infinite Elements
@Boundary TopSurface
  @@Type: TET10
  @@Material: Elastic1
  @@Sides: ZMax
  @@Tolerance: 0.01
  @@RayLength: 40.0
  @@ApplyInfinityBC: Yes
  @@DisX: 0.0
  @@DisY: 0.0
  @@DisZ: 0.0
  @@PW: 0.0
%%%
```

1.8.4 Example 4: 3D Multiple Sides (Box Domain)

Use one block per face (current implementation supports exactly one `@@Sides` value per block):

```
% Infinite Elements
@Boundary XMin
  @@Type: TET10
  @@Material: Soil1
  @@Sides: XMin
  @@RayLength: 50.0
  @@ApplyInfinityBC: Yes
  @@DisX: 0.0
  @@DisY: 0.0
  @@DisZ: 0.0

@Boundary XMax
  @@Type: TET10
  @@Material: Soil1
  @@Sides: XMax
  @@RayLength: 50.0
  @@ApplyInfinityBC: Yes
  @@DisX: 0.0
  @@DisY: 0.0
  @@DisZ: 0.0

@Boundary YMin
  @@Type: TET10
```

```
  @@Material: Soil1
  @@Sides: YMin
  @@RayLength: 50.0
  @@ApplyInfinityBC: Yes
  @@DisX: 0.0
  @@DisY: 0.0
  @@DisZ: 0.0

@Boundary YMax
  @@Type: TET10
  @@Material: Soil1
  @@Sides: YMax
  @@RayLength: 50.0
  @@ApplyInfinityBC: Yes
  @@DisX: 0.0
  @@DisY: 0.0
  @@DisZ: 0.0

@Boundary ZMax
  @@Type: TET10
  @@Material: Soil1
  @@Sides: ZMax
  @@RayLength: 50.0
  @@ApplyInfinityBC: Yes
  @@DisX: 0.0
  @@DisY: 0.0
  @@DisZ: 0.0
%%%
```

1.8.5 Example 5: Multiple 2D Boundaries

```
% Infinite Elements
@Boundary Left
  @@Type: QUAD8
  @@Material: Elastic1
  @@Nodes: 1 2 3 4
  @@Topology: Strip
  @@ExteriorPoint: -50.0 0.0
  @@RayLength: 40.0

@Boundary Right
  @@Type: QUAD8
  @@Material: Elastic1
  @@Nodes: 10 11 12 13
```

```
  @@Topology: Strip
  @@ExteriorPoint: 50.0 0.0
  @@RayLength: 40.0

@Boundary Bottom
  @@Type: QUAD8
  @@Material: Elastic1
  @@Nodes: 20 21 22 23
  @@Topology: Strip
  @@ExteriorPoint: 0.0 -50.0
  @@RayLength: 40.0
%%%
```

1.8.6 Example 6: Coupled Analysis with Far-Field Drainage

```
% Infinite Elements
@Boundary PerimeterDrainage
  @@Type: QUAD8
  @@Material: SaturatedSoil
  @@Nodes: 301 302 303 304 305 306 307 308
  @@Topology: Strip
  @@RayLength: 100.0
  @@ApplyInfinityBC: Yes
  @@ApplyInfinityBCPW: Yes
  @@DisX: 0.0
  @@DisY: 0.0
  @@PW: 0.0
%%%
```

1.9 Implementation Notes

1.9.1 Generator Workflow

1. **Interface Detection**: Identifies interface nodes/edges from `@@Nodes` (2D) or `@@Sides` (3D).
2. **Direction Determination**: Computes outward normal automatically or using `@@ExteriorPoint`/`@@InteriorPoint` hints.
3. **Node Creation**: Creates new nodes at distance `@@RayLength` in direction determined.
4. **Element Creation**: Creates mapped infinite elements in 2D, or a conforming finite buffer layer in 3D, using the specified material.
5. **Boundary Condition Application**: If `@@ApplyInfinityBC: Yes`, applies Dirichlet conditions to direction nodes.

1.9.2 Automatic Mid-Node Detection (TRI6 / QUAD8)

For quadratic elements, list only corner nodes in `@@Nodes`. The generator:

1. Identifies adjacent finite element for each edge segment.
2. Extracts mid-side node from the adjacent element's node list.
3. Constructs infinite element with both corner and mid-side nodes.

This ensures geometric and field compatibility at the interface.

1.10 Comparison with Other Boundary Treatments

1.10.1 Infinite Elements vs Sponge Layer

- **Sponge**: Adds damping in a finite layer (domain remains finite).
- **Infinite Elements**: 2D uses mapped elements extending to infinity; 3D currently uses a finite buffer layer (no inherent damping).

See Sponge Layer.

1.10.2 Infinite Elements vs PML

- **PML**: Complex coordinate stretching for wave absorption (ideally reflection-free).
- **Infinite Elements**: Geometric truncation (reflections possible, especially for waves).

Recommendation: For wave absorption, use PML. For static far-field representation, use infinite elements.
See PML.

1.11 References

- O.C. Zienkiewicz, R.L. Taylor, *The Finite Element Method*, Vol. 1 (unbounded domain formulations).
- C. Bettess, "Infinite elements," *International Journal for Numerical Methods in Engineering*, 11(1):53-64, 1977.
- P. Bettess, *Infinite Elements*, Penshaw Press, 1992.
