AD FALCON API Manual

Point & Line State Output

Javad Ghorbani

March 26, 2026

Contents

1 Point & Line State Output

This page documents the `% PointStateOutput` and `% LineStateOutput` sections, which write interpolated state variables to CSV files.

General notes:

- Directive keywords are case-insensitive and accept one or more leading @ characters (for example, `@Steps`, `@@Steps`, `@Steps:` are equivalent).
- Integer lists accept IDs separated by spaces/commas/semicolons, and ranges may be written as `lo-hi` (and also `lo:hi` where supported).
- Current support: interpolation is implemented for standard **2D** triangle/quad elements (3/6-node triangles and 4/8-node quads) and for **3D** 10-node tetrahedra (T10).

1.1 Point State Output (`% PointStateOutput`)

Interpolates selected state variables at user-defined points and writes them to a CSV file at selected simulation steps.

1.1.1 Syntax

```
% PointStateOutput
@Point  <x> <y> [z]
@Point  <x> <y> [z]
@StateVars  <varName...>
@Steps  <stepId...>
@OutputFile  <path>           (optional)
@Frequency   <integer>        (optional, default: 1)
%%%
```

1.1.2 Parameters

Directive	Required	Default	Description
`@Point x y [z]`	Yes	—	Query location(s) in global coordinates. In 2D/axisymmetric, z is optional (use `z = 0`). In 3D, provide z.

Directive	Required	Default	Description
`@StateVars ...` / `@StateVariables ...`	Yes	—	State variable names to interpolate (see State Variables; examples: `StressXX`, `VoidRatio`, `Pore WaterPressure`). UMAT-defined custom state variables are supported: list them by name (exact match) in `@StateVars` / `@StateVariables`.
`@Steps ...`	Yes	—	Simulation step IDs at which rows are written.
`@OutputFile path`	No	`point_state_output.csv`	Output CSV path.
`@Frequency n`	No	`1`	Write every *n*th converged substep.

1.1.3 Output format

- The output file is cleared once (the first time it is written) and then appended to for the rest of the run.
- The CSV header is written once and has the form:

```
StepID,Time,P1(x;y)_Var1,... (2D)  or  StepID,Time,P1(x;y;z)_Var1,... (3D)
```

If a point cannot be mapped inside an active element, the corresponding entry is written as `nan`.

1.1.4 Example

```
% PointStateOutput
@Point  0.50  0.25
@Point  1.00  0.25
@StateVars  StressXX  StressYY  VoidRatio  PoreWaterPressure
@Steps  1 3 5
@OutputFile  results/point_states.csv
@Frequency  1
%%%
```

1.2 Line State Output (`% LineStateOutput`)

Interpolates selected state variables along user-defined straight line segments and writes them to a CSV file.

1.2.1 Syntax (step-gated)

```
% LineStateOutput
@Line  <x0> <y0>  <x1> <y1>  <spacing>
@Line  <x0> <y0> <z0>  <x1> <y1> <z1>  <spacing>
@StateVars  <varName...>
@Steps  <stepId...>
@OutputFile  <path>           (optional)
@Frequency   <integer>        (optional, default: 1)
%%%
```

1.2.2 Syntax (time-triggered)

```
% LineStateOutput
@Line  <x0> <y0>  <x1> <y1>  <spacing>
@Line  <x0> <y0> <z0>  <x1> <y1> <z1>  <spacing>
@StateVars  <varName...>
@Times  <t1> <t2> ...
@OutputFile  <path>           (optional)
@Frequency   <integer>        (optional, default: 1)
%%%
```

1.2.3 Parameters

Directive	Required	Default	Description
`@Line x0 y0 x1 y1 spacing`	Yes (≥1 line)	—	2D line definition. `spacing` must be > 0.
`@Line x0 y0 z0 x1 y1 z1 spacing`	Yes (≥1 line)	—	3D line definition. `spacing` must be > 0.
`@StateVars ...` / `@StateVariables ...`	Yes	—	State variable names to interpolate (see State Variables). UMAT-defined custom state variables are supported: list them by name (exact match) in `@StateVars` / `@StateVariables`.
`@Steps ...`	Conditional	—	Step-gated output. Required if `@Times` is not provided.
`@Times ...`	Conditional	—	Time-triggered output. If present, it is used instead of `@Steps`.
`@OutputFile path`	No	`line_state_output.csv`	Output CSV path.
`@Frequency n`	No	`1`	Write every *n*th converged substep. With `@Times`, using `@Frequency 1` is recommended.

1.2.4 Output format

- The output file is cleared once (the first time it is written) and then appended to.
- Step-gated layout (`@Steps`): one row per output event, with columns `StepID,Time,...`.
- Time-triggered layout (`@Times`): one row per sample point at each output event, with columns `x,y,Time,...` in 2D and `x,y,z,Time,...` in 3D.

  If interpolation fails at a sample point, the corresponding variable value is written as `nan`.

1.2.5 Example

```
% LineStateOutput
@Line  0.0 0.0   0.0 2.0   0.02
@StateVars  StressXX  StressYY
@Times  0.1  0.5  1.0
@OutputFile  results/line_states.csv
@Frequency  1
%%%
```