AD FALCON API Manual

Linear Solvers

Javad Ghorbani

March 26, 2026

Contents

1 Linear Solvers

In **implicit** time integration, FALCON solves sparse linear systems that arise inside Newton iterations (and, for coupled problems, the implicit flow solve). Your linear solver choice mainly affects robustness, memory use, and runtime.

For **explicit** dynamics, FALCON does not run Newton iterations for the solid update; instead it advances the displacement DOFs using a mass solve (configured separately via the explicit/IMEX mass-solve keys in `% Step Definitions`).

1.1 Overview

FALCON currently provides two solver paths for implicit linear systems:

SolverType value	Path	Best For	Memory	Notes
`Direct`	Sparse LU (with QR fallback)	High-infinite systems		Most robust choice
`KSPBICG`	Iterative BiCGSTAB + ILUT	Large, well-conditioned problems	Lower	Iterative parameters are internal (not user-tunable)

!!! tip "Recommendation" For most nonlinear geomechanics problems (plasticity, softening, contact, coupled U–P), **`Direct`** is recommended for robustness.

1.2 Solver selection syntax

Specify the solver inside `% Step Definitions`:

```
% Step Definitions
@Step 1:
  @@SolverType: Direct
  ...
%%%
```

- Step keys are case-insensitive (e.g., `@@SolverType`, `@@solvertype`, `@@SOLVERTYPE` are equivalent).
- SolverType **values** are case-sensitive (use the spellings shown in the tables).

1.2.1 Parameters

Parameter	Description
`@@SolverType:`	Linear solver selection for implicit linear systems (`Direct`, `KSPBICG`).
`@@PreconditionerType:`	Accepted for compatibility; the current iterative backend uses ILUT internally and does not switch preconditioners based on this key.

1.3 Interaction with `TimeIntegration` (Implicit / Explicit / IMEX)

- `TimeIntegration: Implicit` uses the linear solver specified by `@@SolverType:` for the implicit Newton solves.
- `TimeIntegration: Explicit` does **not** use `@@SolverType:` for the solid update. Instead, the main linear-algebra choice is the **mass solve** (`@@MassSolve:` and related keys).
- `TimeIntegration: IMEX` uses:
  - an **explicit** solid update (mass solve, controlled by `@@MassSolve:`), and
  - an **implicit** flow solve (uses `@@SolverType:`).

See Steps for the explicit/IMEX mass-solve keys (`MassSolve`, `MassSolveTol`, `MassSolve MaxIters`).

1.4 Available solver paths (implicit linear systems)

1.4.1 1) Direct (`Direct`)

- Sparse LU factorization with partial pivoting.
- If LU fails (rank-deficient or highly non-normal matrices), FALCON falls back to a Sparse QR solve.

```
@@SolverType: Direct
```

1.4.2 2) Iterative (`KSPBICG`)

- BiCGSTAB with ILUT preconditioning.
- If the first attempt fails, a stricter retry is attempted.

```
@@SolverType: KSPBICG
```

1.5 Internal preprocessing (implicit linear systems)

Before solving, FALCON applies:

1.5.1 1) Ruiz equilibration

Row/column scaling to improve conditioning:

$$\hat{\mathbf{K}} = \mathbf{D}_r \, \mathbf{K} \, \mathbf{D}_c$$

1.5.2 2) Diagonal boost

A small perturbation is added to diagonal entries for numerical stability:

$$\hat{K}_{ii} \leftarrow \hat{K}_{ii} + \tau \cdot \max_j |\hat{K}_{ij}|$$

1.5.3 3) Back-scaling

The solution is back-scaled after solving:

$$\mathbf{x} = \mathbf{D}_c \, \hat{\mathbf{x}}$$

1.6 Troubleshooting

1.6.1 Iterative solver fails

Symptoms: ILUT factorization fails, or the iterative solve fails after retry.

Actions: 1. Switch to `@@SolverType: Direct` 2. Reduce the time step (or tighten automatic substepping) 3. Improve mesh quality / check boundary conditions 4. Check for material instabilities (softening/contact/near-singularity)

1.6.2 Direct solver runs out of memory

Actions: 1. Try `@@SolverType: KSPBICG` 2. Reduce mesh resolution 3. Prefer 2D/axisymmetric when possible

---

1.7 Advanced: implementation details

FALCON uses Eigen for sparse linear algebra. The current implicit linear-solver backends are:

- Direct: `Eigen::SparseLU` with fallback to `Eigen::SparseQR`
- Iterative: `Eigen::BiCGSTAB` with `Eigen::IncompleteLUT` (ILUT)

1.7.1 Default parameters (current)

Parameter	Value
Ruiz equilibration iterations	8
Ruiz epsilon	1e-15
Diagonal boost τ	1e-8
SparseLU pivot threshold	0.1
SparseQR pivot threshold	1e-12
BiCGSTAB drop tolerance	1e-5
BiCGSTAB fill factor	25
BiCGSTAB max iterations	8000
BiCGSTAB tolerance	1e-8
BiCGSTAB (retry) drop tolerance	5e-6
BiCGSTAB (retry) fill factor	30
BiCGSTAB (retry) max iterations	12000
BiCGSTAB (retry) tolerance	5e-9

1.8 References

- Eigen: https://eigen.tuxfamily.org/

1.9 See also

- Steps — Step configuration (including `TimeIntegration` and explicit/IMEX mass-solve keys)
- Automatic time incrementation — Automatic time stepping
- Error Dictionary — Error messages