

leigqNEWTON — Newton solver for quaternion LEFT eigenpairs

This page documents the main solver leigqNEWTON. Convert to Live Script:

Requirements

These examples use MATLAB's built-in quaternion class (Aerospace Toolbox). If quaternion is not available, the text still publishes and all examples are skipped.

```
hasQuat = true;
try
    quaternion(0,0,0,0);
catch
    hasQuat = false;
end

if ~hasQuat
    disp('doc_leigqNewton: MATLAB class "quaternion" not available. Examples are
skipped.');
end
```

What it computes (left eigenpairs)

A **left** eigenpair (λ , v) satisfies

$$A v = \lambda v, \quad v \neq 0,$$

where the quaternion λ acts on the **left**.

Syntax

- $[\lambda, V, \text{res}] = \text{leigqNewton}(A)$
- $[\lambda, V, \text{res}, \text{info}] = \text{leigqNewton}(A, \text{Name}, \text{Value}, \dots)$
- $[\lambda, V, \text{res}, \text{info}, \lambda_U, V_U, \text{res}_U] = \text{leigqNewton}(A, \text{Name}, \text{Value}, \dots)$
- $[\dots] = \text{leigqNewton}(A, \text{Num}, \text{Name}, \text{Value}, \dots) \%$ convenience positional Num

Here A is an $n \times n$ quaternion matrix.

Output summary

- λ ($K \times 1$ quaternion): accepted eigenvalue hits (may contain duplicates).
- V ($n \times K$ quaternion): corresponding eigenvectors (columns).
- res ($K \times 1$ double): final residuals for each hit (normalized by default).
- info (cell): $\text{info}\{1\}$ is a summary struct; $\text{info}\{2:\text{end}\}$ are per-trial logs when requested (see '`InfoLevel`').
- $\lambda_U, V_U, \text{res}_U$: a **distinct representative set** extracted from λ using a tolerance (does not change λ ; it only groups near-duplicates).

Common one-liners

```

Default (balanced): [lam,V,res] = leigqNewton(A);

Reliable profile + distinct representatives: [lam,V,res,info,lambda,U,VU,resU] =
leigqNewton(A, 'SolveProfile', 'reliable', 'Seed', 1);

Request exactly 9 eigenpairs (two equivalent forms): [lam,V,res]
= leigqNewton(A, 9, 'SolveProfile', 'reliable'); [lam,V,res] =
leigqNewton(A, 'Num', 9, 'SolveProfile', 'reliable');

Reproducible run: [lam,V,res] = leigqNewton(A, 'Seed', 1, 'SolveProfile', 'default');

```

Minimal example (self-contained)

The solver is restart-based and therefore stochastic. For reproducible output, set a seed.

```

if hasQuat
    rng(1);
    n = 4;
    A = quaternion(randn(n),randn(n),randn(n),randn(n));

    [lam,V,res] = leigqNewton(A, 'Seed', 1, 'SolveProfile', 'default');

    fprintf('Returned %d hits. median(res)=%e, max(res)=%e\n', ...
        numel(lam), median(res), max(res));

    % Also return a distinct representative set (lambda)
    [~,~,~,info,lambda,U,~,resU] = leigqNewton(A, 'Seed', 1, 'SolveProfile', 'reliable');
    fprintf('Distinct representatives: %d (median(resU)=%e)\n', numel(lambda),
    median(resU));
    disp(info{1});
end

```

```

Returned 4 hits. median(res)=9.62e-16, max(res)=8.88e-13
Distinct representatives: 3 (median(resU)=4.51e-17)
    nRuns: 4
    nAccepted: 4
    nConverged: 4
        targetK: 4
    maxTrials: 200
        opt: [1x1 struct]
    rankA: 4
    df0: 0
    nAcceptedZero: 0
    trialsTotal: 4
        trialIter: [4x1 double]
    itersTotal: 35
    acceptedNewton: 4
    acceptedTrialsPerEig: [4x1 double]
        acceptedTrialIdx: [4x1 double]
        acceptedIter: [4x1 double]
    itersAcceptedTotal: 35
tailTrialsAfterLastAccept: 0
    restartsTotal: 0
        infoLevel: 'full'
        summary: [1x1 struct]
    distinct: [1x1 struct]

```

Reproducing paper / supplement examples (safe path handling)

Public examples shipped with the toolbox live in examples/. To avoid "path chaos", resolve the toolbox root using which:

```
root = fileparts(which('leigqNEWTON'));
run(fullfile(root,'examples','ExNewton_1_HuangSo.m'));
```

The snippet below is guarded by a flag so that publishing this page does not automatically run long scripts.

```
if hasQuat
    % Optional: run the longer paper/supplement example scripts.
    %
    % By default, this documentation page does NOT run the paper examples.
    % To enable them, run in the Command Window:
    % RUN_PAPER_EXAMPLES = true;
    % doc_leigqNEWTON
    if ~exist('RUN_PAPER_EXAMPLES','var')
        RUN_PAPER_EXAMPLES = false; % default
    end

    if RUN_PAPER_EXAMPLES
        root = fileparts(which('leigqNEWTON'));
        ex1 = fullfile(root,'examples','ExNewton_1_HuangSo.m');
        ex2 = fullfile(root,'examples','ExNewton_2_MVPS.m');

            if exist(ex1,'file'), run(ex1); else, error('Example not found:
%s', ex1); end
            if exist(ex2,'file'), run(ex2); else, error('Example not found:
%s', ex2); end
        else
            fprintf("Paper examples are disabled by default. Set
RUN_PAPER_EXAMPLES=true and rerun this page to execute them.\n");
        end
    end
```

Paper examples are disabled by default. Set RUN_PAPER_EXAMPLES=true and rerun this page to execute them.

Tips (reproducibility and clean output)

- Prefer 'Seed' for reproducibility.
- Use checkNEWTON as a smoke test after downloading/unzipping.
- If you want to benchmark random initialization on triangular matrices, set 'TriangularInit',false.

See also

checkNEWTON, leigqNEWTON_refine_batch, leigqNEWTON_cert_resMin, leigqNEWTON_sphere_sample