checkNEWTON — one-call diagnostic workflow

Print a compact report, compute missing lambdas/vectors, and optionally trigger sphere diagnostics.

Requirements

This toolbox relies on MATLAB's built-in quaternion class.

```
hasQuat = true;
try
    quaternion(0,0,0,0);
catch
    hasQuat = false;
end
if ~hasQuat
    disp('This toolbox requires MATLAB''s built-in quaternion class
(quaternion(w,x,y,z)).');
    disp('Examples in this page are skipped.');
    return;
end

% Ensure toolbox is on the path (add root only, if needed).
if exist('checkNEWTON','file') ~= 2
    thisFile = mfilename('fullpath');
    if ~isempty(thisFile)
        rootGuess = fileparts(fileparts(fileparts(thisFile))); % .../docs/source ->
toolbox root
        if exist(fullfile(rootGuess,'checkNEWTON.m'),'file')
            addpath(rootGuess);
        end
    end
end

if exist('checkNEWTON','file') ~= 2
    error('checkNEWTON not found on the MATLAB path. Add the toolbox root folder.');
end
```

Syntax

- `out = checkNEWTON(A)`
- `out = checkNEWTON(A, lambda)`
- `out = checkNEWTON(A, lambda, V)`
- `out = checkNEWTON(A, [], [], 'SolveArgs', {'SolveProfile','reliable','Seed',1}, 'SphereCheck','auto')`

Notes

- If `lambda` is omitted, CHECKNEWTON calls `leigqNEWTON`.

- If V is omitted, CHECKNEWTON computes best vectors by minimizing `resMin`.
- For presentation, it cleans and rounds lambdas (use 'CleanTol'/'PrintDigits').

Examples

These use fixed small matrices from Huang–So (LAA 323, 2001) for reproducibility.

```
q0 = quaternion(0,0,0,0);
q1 = quaternion(1,0,0,0);
qi = quaternion(0,1,0,0);
qj = quaternion(0,0,1,0);
qk = quaternion(0,0,0,1);
```

Example 1: simplest call (solver + report)

Huang–So Example 2.6

```
A = [ q0, qi;
      qj, q1 ];
out = checkNEWTON(A);
```

```
=== checkNEWTON ===
Matrix: 2x2 quaternion
Lambdas: 2 (computed).  Vectors: 2 columns (computed).
Certificate/residual summary (median | max):
  resMin abs: 1.34e-11 | 2.68e-11    resMin rel: 4.91e-12 | 9.81e-12
  resPair abs:1.34e-11 | 2.68e-11    resPair rel:4.91e-12 | 9.81e-12

Per-eigenvalue residuals:
  #  lambda (cleaned/rounded)                          resMin(abs/rel)        resPair(abs/rel)
   1   0.5 +0.5i +0.5j -0.5k                            2.68e-11/ 9.81e-12    2.68e-11/ 9.81e-12
   2   0.5 -0.5i -0.5j -0.5k                            6.31e-17/ 2.31e-17    3.16e-16/ 1.15e-16

Interestingness:
  Standard: Kdistinct = n = 2.
  Sphere check: not run. Use checkNEWTON(...,'SphereCheck','on') to force.

Outputs:
  out           : main struct (out.lambda, out.v, out.resMinAbs, out.resPairAbs, ...).
  cases{1}      : simple case struct with fields A, lamAll, lamSamples, sph, info (sphere tooling style).
  S             : summary struct (n, Ktot, Kdistinct, spheresFound, ...).
```

Example 2: lambda-only workflow

(you already have candidates; CHECKNEWTON will compute vectors and report residuals)

```
[lambda] = leigqNEWTON(A,'SolveProfile','fast','Seed',1);
out2 = checkNEWTON(A, lambda, []);
```

```
=== checkNEWTON ===
Matrix: 2x2 quaternion
Lambdas: 2 (provided).  Vectors: 2 columns (computed).
Certificate/residual summary (median | max):
  resMin abs: 1.34e-11 | 2.68e-11    resMin rel: 4.91e-12 | 9.81e-12
  resPair abs:1.34e-11 | 2.68e-11    resPair rel:4.91e-12 | 9.81e-12
```

```
Per-eigenvalue residuals:
  #  lambda (cleaned/rounded)                          resMin(abs/rel)        resPair(abs/rel)
   1   0.5 +0.5i +0.5j -0.5k                            2.68e-11/ 9.81e-12    2.68e-11/ 9.81e-12
   2   0.5 -0.5i -0.5j -0.5k                            6.31e-17/ 2.31e-17    3.16e-16/ 1.15e-16

Interestingness:
  Standard: Kdistinct = n = 2.
  Sphere check: not run. Use checkNEWTON(...,'SphereCheck','on') to force.

Outputs:
  out           : main struct (out.lambda, out.v, out.resMinAbs, out.resPairAbs, ...).
  cases{1}      : simple case struct with fields A, lamAll, lamSamples, sph, info (sphere tooling style).
  S             : summary struct (n, Ktot, Kdistinct, spheresFound, ...).
```

Example 3: sphere diagnostics (Huang–So Example 2.7)

```
A27 = [ quaternion(2,0,0,0),  qi;
        -qi,                  quaternion(2,0,0,0) ];

% With SphereCheck='on', checkNEWTON runs leigqNEWTON_sphere_sample internally.
out3 = checkNEWTON(A27, [], [], 'SphereCheck','on', 'SphereCollect',20,
'SphereSeed0',1);
```

```
=== checkNEWTON ===
Matrix: 2x2 quaternion
Lambdas: 2 (computed).  Vectors: 2 columns (computed).
Certificate/residual summary (median | max):
  resMin abs: 1.11e-11 | 1.89e-11    resMin rel: 2.25e-12 | 3.84e-12
  resPair abs:1.11e-11 | 1.89e-11    resPair rel:2.25e-12 | 3.84e-12

Per-eigenvalue residuals:
  #  lambda (cleaned/rounded)                          resMin(abs/rel)        resPair(abs/rel)
   1   1.526 +1.848e-11i -0.01666j +0.8802k             1.89e-11/ 3.84e-12    1.89e-11/ 3.84e-12
   2   1.672 +3.249e-12i -0.6181j +0.7145k              3.33e-12/ 6.55e-13    3.33e-12/ 6.55e-13

Interestingness:
  Standard: Kdistinct = n = 2.
  Sphere check: detected 1 sphere model(s).
    Sphere #1: radius=1  (basis4x3 present=1)
      p0   =  2.78 +6.726e-19i -0.1326j -0.611k
      center3 = [-0.466469 -0.860341 -0.205477]^T
      verdict = INCONCLUSIVE  reliability=LOW (0.20)
      Sphere #1: inliers = 20 distinct samples
        lamSamples(1) =  1.526 +1.848e-11i -0.01666j +0.8802k
        lamSamples(2) =  1.672 +3.249e-12i -0.6181j +0.7145k
        lamSamples(3) =  1.887 +0i +0.9889j -0.09615k
        lamSamples(4) =  2.146 +0i +0.6545j -0.7419k
        lamSamples(5) =  2.194 +0i -0.1478j -0.9698k
        lamSamples(6) =  1.614 +0i -0.3599j +0.8495k
    Your 2 provided/computed lambdas: 2 are within SphereTolRel=5e-3 of a detected sphere model.

Outputs:
  out           : main struct (out.lambda, out.v, out.resMinAbs, out.resPairAbs, ...).
  cases{1}      : simple case struct with fields A, lamAll, lamSamples, sph, info (sphere tooling style).
  S             : summary struct (n, Ktot, Kdistinct, spheresFound, ...).
```

See also

leigqNEWTON, leigqNEWTON_refine_batch, leigqNEWTON_sphere_sample