leigqNEWTON_sphere_refine

Advanced engine behind `leigqNEWTON_sphere_validate`.

Both VALIDATE and REFINE expect a *case struct* that contains a sphere model in |cases.sph|. If no sphere model is supplied, the verdict will be "NO_SPHERE_MODEL".

This page is intentionally **fast** and runs only `leigqNEWTON_sphere_refine` on a small, pre-defined test case. We do **not** call detection/sampling here.The "previous step" (detect/sample) is emulated by providing: * a candidate sphere model `sph0`, * a small list of approximate candidates `lamAll` and DISTINCT samples `lamS`. Then `leigqNEWTON_sphere_refine` refines candidates, computes residual certificates, and (optionally) runs a modest grid test to decide sphere vs. artifact.

For more reliable (but slower) settings, see the "RELIABLE profile" section near the end of this page.

The examples require MATLAB built-in quaternion class.

Requirements and path

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

Test matrix (2x2 with a known eigen-sphere)

```
qi = quaternion(0,1,0,0);
A  = [ quaternion(2,0,0,0),  qi;
       -qi,                quaternion(2,0,0,0) ];
```

"Precomputed" inputs that would typically come from detection

Sphere model in R^4 for the 2x2 example: center4 = (2,0,0,0), radius = 1, subspace spanned by e2,e3,e4 (imaginary part).

```
B = [0 0 0;
     1 0 0;
     0 1 0;
     0 0 1];                    % 4x3 basis
```

```
center4 = [2;0;0;0];
sph0 = struct();
sph0.p0      = center4(:).';   % 1x4 row, numeric
sph0.center3 = [0;0;0];
sph0.center4 = center4(:).';
sph0.center  = quaternion(center4(1),center4(2),center4(3),center4(4));
sph0.radius  = 1;
sph0.basis4x3 = B;

% Candidate points on the sphere (then we perturb them slightly so refinement does
something)
U = [ 1  0  0;
     -1  0  0;
      0  1  0;
      0 -1  0;
      0  0  1;
      0  0 -1;
      1  1  0;
      1 -1  0]';
U(:,7:8) = U(:,7:8) / sqrt(2);  % normalize the last two
P4 = center4 + B*(U);           % 4xN

lamExact = quaternion(P4(1,:), P4(2,:), P4(3,:), P4(4,:)).';
sigma = 1e-3;  % doc-level perturbation (small)
rng(1);
dW = sigma*randn(size(P4,2),1);
dX = sigma*randn(size(P4,2),1);
dY = sigma*randn(size(P4,2),1);
dZ = sigma*randn(size(P4,2),1);
lam0 = quaternion(P4(1,:).'+dW, P4(2,:).'+dX, P4(3,:).'+dY, P4(4,:).'+dZ);

lamAll = lam0;
lamS   = lam0;
sph0.inliers = (1:numel(lamS)).';


C = struct('A',A,'lamAll',lamAll,'lamSamples',lamS,'sph',sph0);
```

Run refine/validate (doc profile)

We suppress near-singular warnings here because, in documentation runs, they are distracting and do not indicate a failure of the algorithm.

```
ws = warning;
warning('off','MATLAB:nearlySingularMatrix');
warning('off','MATLAB:singularMatrix');
cleanupW = onCleanup(@() warning(ws));

refineArgs =
{'Verbose',0,'DoPolish',false,'MaxIter',200,'TolFun',1e-14,'TolX',1e-14};
```

```
[A2, lamAll2, resAll, lamS2, resS, sph2, conf, out] = leigqNEWTON_sphere_refine( ...
    C, ...
    'RefineArgs', refineArgs, ...
    'TargetRes', 1e-14, ...
    'VerifySphere', true, ...
    'GridTheta', 8, 'GridPhi', 12, 'GridMaxPoints', 120, ...
    'RefineGrid', false, ...
    'RefitSphere', false, ...
    'Verbose', 1);
```

```
=== leigqNEWTON_sphere_refine: idx=NaN, seed=NaN, origin= ===
Ktot=8, Kdistinct=8, spheres=1
Initial residuals: lamAll  med=3.83e-01  max=1.41e+00 | lamSamples med=3.83e-01  max=1.41e+00
Refined residuals: lamAll  med=8.16e-15  max=1.82e-14 | lamSamples med=2.74e-15  max=9.20e-15
Inliers (n=8): res med=2.74e-15 max=9.20e-15
Inliers on-sphere (orig):  devRel med=7.48e-03 max=5.04e-01
Eigenvalue certification (TargetRes=1.0e-14): lamAll pass=75% | lamSamples pass=100%
Sphere decision: INCONCLUSIVE  reliability=LOW (0.20)
  Why: Mixed evidence; adjust tolerances or increase grid for a clearer decision.
Grid (no refine): N=96  passRes=25%  passSphere=100%  passBoth=25%
```

```
fprintf('Residuals after refine: samples med=%.2e  max=%.2e\n', median(resS),
max(resS));
```

```
Residuals after refine: samples med=2.74e-15  max=9.20e-15
```

```
disp(conf.sphere);
```

```
            verdict: 'INCONCLUSIVE'
                why: 'Mixed evidence; adjust tolerances or increase grid for a clearer decision.'
      reliabilityTag: 'LOW'
    reliabilityScore: 0.2000
```

RELIABLE profile (optional, slower)

```
RUN_SPHERE_REFINE_RELIABLE = false;
try
    RUN_SPHERE_REFINE_RELIABLE = evalin('base','RUN_SPHERE_REFINE_RELIABLE');
catch
end
if RUN_SPHERE_REFINE_RELIABLE
    disp('RELIABLE profile: increases the grid and allows polishing (can take
minutes).');
    refineArgs =
{'Verbose',0,'DoPolish',true,'MaxIter',400,'TolFun',1e-16,'TolX',1e-16,'MaxIterPolis
h',80,'TolResPolish',1e-15};
    [A2, lamAll2, resAll, lamS2, resS, sph2, conf, out] =
leigqNEWTON_sphere_refine( ...
        C, ...
        'RefineArgs', refineArgs, ...
        'TargetRes', 1e-14, ...
        'VerifySphere', true, ...
```

```
        'GridTheta', 16, 'GridPhi', 24, 'GridMaxPoints', 600, ...
        'RefineGrid', true, ...
        'RefitSphere', true, ...
        'Verbose', 1);
    disp(conf.sphere);
end
```

See also

leigqNEWTON_sphere_validate, leigqNEWTON_sphere_detect, leigqNEWTON_sphere_sample