leigqNEWTON_sphere_sample

Advanced engine: sample left eigenvalues and (optionally) fit sphere **candidates**.

This is a low-level sampling routine used by `leigqNEWTON_sphere_detect`. Typical users should call `leigqNEWTON_sphere_detect` (or `..._sphere_validate`) unless they need fine control over sampling settings.

Key Name-Value options (selected; see `help leigqNEWTON_sphere_sample`): 'Collect' : number of distinct samples to collect (default 20) 'RunsMax' : max number of Newton runs (default 120) 'Restarts' : restarts per run (default 1500) 'Seed0' : base seed (default 1) 'UniqueTol' : distinctness tolerance (default 1e-8) 'DetectSphere': true/false (default true) 'Report' : 'off'|'summary'|'progress'|'full'

Setup (requirements + 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)).');
    disp('Examples in this page are skipped.');
    return;
end

% Ensure toolbox root is on the path (add only the root, no genpath).
if exist('leigqNEWTON_sphere_sample','file') ~= 2
    thisFile = mfilename('fullpath');
    if ~isempty(thisFile)
        rootGuess = fileparts(fileparts(fileparts(thisFile))); % .../docs/source ->
toolbox root
        if exist(fullfile(rootGuess,'leigqNEWTON_sphere_sample.m'),'file')
            addpath(rootGuess);
            rehash toolboxcache
        end
    end
end
if exist('leigqNEWTON_sphere_sample','file') ~= 2
    error('Sphere functions not found on the MATLAB path. Add the toolbox root
folder.');
end
```

Example (small run): collect samples and inspect sphere candidates

A small 2×2 test matrix used in the sphere-hunting docs (Huang–So Example 2.7 style).

```
qi = quaternion(0,1,0,0);
```

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

% Keep parameters modest for a documentation run.
[lamAll, lamS, lam0, cls, sph, info] = leigqNEWTON_sphere_sample( ...
    A, 'Collect', 20, 'RunsMax', 60, 'Restarts', 200, 'Seed0', 1, 'Report', 'off');

fprintf('Collected: Ktot=%d  (lamAll),  Kdistinct=%d  (lamSamples)\n',
numel(lamAll), numel(lamS));
```

```
Collected: Ktot=20  (lamAll),  Kdistinct=20  (lamSamples)
```

```
fprintf('Sphere candidates detected: %d\n', numel(sph));
```

```
Sphere candidates detected: 1
```

```

% Inspect a few outputs
lam0
```

```
lam0 = struct with fields:
        certain: [0×1 quaternion]
    certainMult: 0
      newtonRaw: [0×1 quaternion]
        zeroTol: 0
```

```
if ~isempty(sph)
    disp('First detected sphere candidate (sph(1)):');
    disp(sph(1));
else
    disp('No sphere candidate detected in this quick run (this is common for many
matrices).');
end
```

```
First detected sphere candidate (sph(1)):
     center4: [2.0000 1.9383e-14 3.6560e-13 5.6577e-13]
      center: [1×1 quaternion]
      radius: 1.0000
    basis4x3: [4×3 double]
          p0: [1.8362 1.9414e-18 -0.1315 0.9777]
     center3: [3×1 double]
     inliers: [20×1 double]
    samples4: [4×1 quaternion]
     sampler: @(theta,phi)local_sphere_sampler(model,theta,phi)
```

See also

leigqNEWTON_sphere_detect, leigqNEWTON_sphere_validate, leigqNEWTON_sphere_refine