

leigqNEWTON_refine_auto

Policy wrapper for refinement

Choose a refinement strategy automatically (e.g., lambda-only vs eigenpair polish).

Setup

```
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

if exist('leigqNEWTON_refine_auto','file') ~= 2
    thisFile = mfilename('fullpath');
    if ~isempty(thisFile)
        rootGuess = fileparts(fileparts(fileparts(thisFile))); % .../docs/source ->
toolbox root
        if exist(fullfile(rootGuess,'leigqNEWTON_refine_auto.m'),'file')
            addpath(rootGuess);
        end
    end
end

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

Syntax

- `[lamB, lamPol, rB, rPol, resInf, info] = leigqNEWTON_refine_auto(A, lam, Name, Value, ...)`

This function is a convenience wrapper that selects a refinement strategy per candidate.

Example: refine solver hits for Pan–Ng 4×4

```
q = @(w,x,y,z) quaternion(w,x,y,z);
a = q(-2, 1, 1, 4);
b = q( 2, 4, 1, 1);
```

```

c = q( 1, 3, 2, 2);
d = q(-1, 2, 2, 3);
A = [a b c d;
      d a b c;
      c d a b;
      b c d a];

lambda = leigqNewton(A, 'Seed', 1, 'SolveProfile', 'reliable', 'K', 2*size(A,1));
[lambda, lamPol, rB, rPol, resInf, info] = leigqNewton_refine_auto(A, lambda,
'Mode', 'auto', 'DoPolish', true); %#ok<ASGLU>
median(rB), max(rB)

```

See also

[leigqNewton_refine_batch](#), [leigqNewton_refine_lambda](#), [leigqNewton_refine_polish](#)