

# leigqNEWTON\_refine\_polish — polish a single eigenpair

Local, fast refinement of a candidate eigenpair (lambda,v).

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

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

## Syntax

- [lamP, vP, resP] = leigqNewton\_refine\_polish(A, lambda, v)

## Example

Start from a solver hit and polish it.

```
q0 = quaternion(0,0,0,0);
q1 = quaternion(1,0,0,0);
qi = quaternion(0,1,0,0);
qj = quaternion(0,0,1,0);
A = [ q0, qi;
       qj, q1 ];
```

```
[lambda, V] = leigqNEWTON(A, 'SolveProfile', 'default', 'Seed', 1);

% Before polish:
r0 = leigqNEWTON_cert_resPair(A, lambda(1), V(:,1));

% Polish:
[lamP, vP, resP] = leigqNEWTON_refine_polish(A, lambda(1), V(:,1));

% After polish:
r1 = leigqNEWTON_cert_resPair(A, lamP, vP);
[r0, r1]
```

```
ans = 1x2
10^-10 x
0.2680    0.0660
```

## See also

[leigqNEWTON\\_refine\\_batch](#), [leigqNEWTON\\_cert\\_resPair](#), [leigqNEWTON\\_init\\_vec](#)