leigqNEWTON_cert_resPair — eigenpair residual

Compute the residual norm for a specific eigenpair candidate (lambda,v): resPair(A,lambda,v) = |A*v - lambda*v| (optionally normalized).

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

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

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

Syntax

- `r = leigqNEWTON_cert_resPair(A, lambda, v)`
- `r = leigqNEWTON_cert_resPair(A, lambda, v, 'ResidualNormalized',true/false)`

Example: compute resPair after polishing

Use a fixed 2×2 test (Huang–So Example 2.6).

```
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 ];

% Get one candidate pair.
[lambda, V] = leigqNEWTON(A, 'SolveProfile','default', 'Seed',1);

% Polish it to near machine precision.
[lamP, vP] = leigqNEWTON_refine_polish(A, lambda(1), V(:,1));

% Pair residual (typically ~1e-15 to 1e-16 after polish on small problems).
r = leigqNEWTON_cert_resPair(A, lamP, vP);
disp(r);
```

```
   6.6023e-12
```

See also

leigqNEWTON_cert_resMin, leigqNEWTON_refine_polish, leigqNEWTON_refine_batch