Getting Started

A friendly introduction to the leigqNEWTON toolbox.

0) Setup (add toolbox to path)

If you are in the toolbox root folder, this is enough:

```
addpath(genpath(pwd));
```

1) Create a small demo quaternion matrix

```
rng(1);
n = 5;
A = quaternion(randn(n),randn(n),randn(n),randn(n));
display(A);
```

```
A = 5×5 quaternion array
     -0.64901 -    1.0511i -   0.55806j +  0.89873k     -0.57266 +    1.2708i +  0.19974j -  0.62368k     -0.85189 +  
       1.1812 -   0.41738i - 0.028453j +  0.37722k     -0.55868 + 0.066009i +  0.42586j -  0.59524k      0.80032 -  
     -0.75845 +    1.4022i -    1.4763j +   1.4524k      0.17838 +  0.45129i -     1.27j +   1.6113k      -1.5094 +  
      -1.1096 -    1.3677i +   0.2589j +  0.44695k     -0.19686 -  0.32221i -  0.48522j -    0.349k      0.87587 +  
     -0.84555 -   0.29253i -    2.0187j +  0.64582k      0.58644 +  0.78841i +  0.59431j +  0.16417k     -0.24279 -  
```

2) Solve: get candidate left eigenvalues

One-liner (recommended):

```
[lam,~,lamc] = leigqNEWTON(A,'SolveProfile','default','Seed',1);
[lambda, V, res, info, lambdaU, VU, resU] = leigqNEWTON(A,
'SolveProfile','default', 'Seed',1);

fprintf("\nAll lambda (lambda):\n");
```

```
All lambda (lambda):
```

```
display(lambda);
```

```
lambda = 5×1 quaternion array
      -2.3573 +    2.0169i -  0.40495j + 0.030024k
      -2.3573 +    2.0169i -  0.40495j + 0.030024k
      0.81292 -   0.83843i +   1.3555j +   2.3216k
      -2.3573 +    2.0169i -  0.40495j + 0.030024k
      -3.5932 -    2.0417i -    1.214j +   2.5199k
```

```


fprintf("\nDistinct lambda (lambdaU):\n");
```

```
Distinct lambda (lambdaU):
```

```
display(lambdaU);
```

```
lambdaU = 3×1 quaternion array
```

```
      -2.3573 +    2.0169i -   0.40495j + 0.030024k
      0.81292 -   0.83843i +    1.3555j +    2.3216k
      -3.5932 -    2.0417i -     1.214j +    2.5199k
```

```
% Quality summary (lower is better):
%median(resU), max(resU)
medRes = median(resU);
maxRes = max(resU);
fprintf("resMin: median = %.3e, max = %.3e\n", medRes, maxRes);
```

```
resMin: median = 3.325e-17, max = 1.004e-14
```

3) Refine and certify a batch of candidates

```
[lamR,VR,cert] = leigqNEWTON_refine_batch(A, lam,'Verbose',0);

fprintf("\nRefined lambda (lamR):\n");
```

```
Refined lambda (lamR):
```

```
display(lamR);
```

```
lamR = 5×1 quaternion array
      -2.3573 +    2.0169i -   0.40495j + 0.030024k
      -2.3573 +    2.0169i -   0.40495j + 0.030024k
      0.81292 -   0.83843i +    1.3555j +    2.3216k
      -2.3573 +    2.0169i -   0.40495j + 0.030024k
      -3.5932 -    2.0417i -     1.214j +    2.5199k
```

```
% Quality summary (lower is better):
%median(resU), max(resU)
medRes = median(cert.resMin);
maxRes = max(cert.resMin);
fprintf("resMin: median = %.3e, max = %.3e\n", medRes, maxRes);
```

```
resMin: median = 1.589e-16, max = 2.351e-16
```

4) Certificates post hoc computation (smaller is better):

```
fprintf('cert.resMin: median = %.3e, max = %.3e', median(cert.resMin),
max(cert.resMin));%% 4) Certificates only (eigenvalue-only vs eigenpair residual)
```

```
cert.resMin: median = 1.589e-16, max = 2.351e-16
```

```
rMin1  = leigqNEWTON_cert_resMin(A, lamR(1));
rPair1 = leigqNEWTON_cert_resPair(A, lamR(1), VR(:,1));
disp([rMin1, rPair1]);
```

```
   1.0e-14 *
    0.0222    0.2049
```

Spherical eigenvalues (advanced)

Some quaternion matrices have infinitely many left eigenvalues forming a sphere. Detecting/validating such spherical families is an advanced (and typically slower) workflow. For a dedicated, reproducible tutorial, see:

```
doc_SphereHunting
```

That page demonstrates the sphere-sampling/validation pipeline based on `leigqNEWTON_SPHERE_sample / _detect / _validate / _refine` and discusses the speed–reliability trade-offs (the run time can range from tens of seconds to minutes depending on settings and hardware).

Tip: from the Command Window you can run: `doc_SphereHunting`