

qmldivideNEWTON — left matrix division A\B for quaternion matrices

Solve $A^*X = B$ for X using a real embedding.

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('qmldivideNEWTON','file') ~= 2
    thisFile = mfilename('fullpath');
    if ~isempty(thisFile)
        rootGuess = fileparts(fileparts(fileparts(thisFile))); % .../docs/source ->
toolbox root
        if exist(fullfile(rootGuess,'qmldivideNEWTON.m'),'file')
            addpath(rootGuess);
            rehash toolboxcache
        end
    end
end
if exist('qmldivideNEWTON','file') ~= 2
    error('qmldivideNEWTON not found on the MATLAB path. Add the toolbox root
folder.');
end
```

Syntax

- $X = \text{qmldivideNEWTON}(A, B)$

Example

```
rng(1);
n = 3;
A = quaternion(randn(n),randn(n),randn(n),randn(n));
% Make A better conditioned for the demo:
A = A + quaternion(5,0,0,0)*eye(n);

B = quaternion(randn(n),randn(n),randn(n),randn(n));
```

```
X = qmldivideNEWTON(A,B);

disp('A:'); disp(A);
disp('B:'); disp(B);
disp('X = A\b\B (computed):'); disp(X);

% Check residual: ||A*X - B|| (componentwise max-norm)
R = qmtimesNEWTON(A,X) - B;
[w,x,y,z] = parts(R);
maxAbsRes = max(abs([w(:);x(:);y(:);z(:)]));
fprintf('Residual max-abs component: %.3e\n', maxAbsRes);
```

See also

[qmrdivideNEWTON](#), [qmtimesNEWTON](#)