

qmtimesNEWTON — quaternion matrix multiply helper

Compute quaternion matrix products using a real embedding (toolbox-controlled helper).

Some MATLAB installations do not support the built-in operator $A*B$ for quaternion arrays. This helper provides a stand-alone, reproducible alternative used throughout leigqNEWTON.

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

end

Syntax

- $C = \text{qmtimesNEWTON}(A,B)$

Example

```
rng(1);
A = quaternion(randn(2),randn(2),randn(2),randn(2));
B = quaternion(randn(2),randn(2),randn(2),randn(2));

C = qmtimesNEWTON(A,B);
```

```
disp('A:'); disp(A);
```

```
A:
-0.64901 - 0.84555i - 0.19686j - 1.5094k    -0.75845 - 0.55868i - 0.85189j - 0.24279k
 1.1812 - 0.57266i + 0.58644j + 0.87587k    -1.1096 + 0.17838i + 0.80032j + 0.16681k
```

```
disp('B:'); disp(B);
```

```
B:
-1.9654 - 0.27516i - 1.8651j - 1.3677k    1.1752 + 1.7813i - 0.41738j + 1.2708k
-1.2701 + 0.60366i - 1.0511j - 0.29253k    2.0292 + 1.7737i + 1.4022j + 0.066009k
```

```
disp('C = qmtimesNEWTON(A,B):'); disp(C);
```

```
C = qmtimesNEWTON(A,B):
-1.0546 - 0.4598i + 2.4254j + 7.0089k    3.2419 - 5.2246i - 4.7604j - 1.7102k
 2.0043 + 0.67686i - 4.0772j - 2.6654k    -2.1614 + 0.75465i + 2.8363j + 0.82067k
```

```
% If your MATLAB supports |A*B| for quaternion arrays, compare results.
```

```
try
    Cbuiltin = A*B;
    D = C - Cbuiltin;
    [w,x,y,z] = parts(D);
    maxAbsDiff = max(abs([w(:);x(:);y(:);z(:)]));
    fprintf('Max abs difference vs built-in A*B: %.3e\n', maxAbsDiff);
catch
    disp('Note: built-in quaternion A*B is not available in this MATLAB
configuration. ');
    disp('    qmtimesNEWTON provides this functionality using a real embedding. ');
end
```

```
Note: built-in quaternion A*B is not available in this MATLAB configuration.
    qmtimesNEWTON provides this functionality using a real embedding.
```

See also

qmldivideNEWTON, qmrdivideNEWTON