computeDipoleHFieldTest

```
% compute a single point without norming
Hsingle = computeDipoleHField(1, 2, 3, [1; 0; 0], 1);

% compute a 3D grid of positions n+1 samples for even values
% in the grid and to
% include (0,0,0), in mm
x = linspace(-4, 4, 41);
y = linspace(4, -4, 41);
z = linspace(4, -4, 41);
[X, Y, Z] = meshgrid(x, y, z);

% magnetic dipole moment to define magnet orientation, no tilt
m = generateDipoleRotationMoments(-1e6, 1, 0);

% norm factor to imprint field strength in certain distance d = 1,
% r = 2 in mm,
% 200 kA/m, no tilt
r0 = rotate3DVector([0; 0; -3], 0, 0, 0);
H0norm = computeDipoleH0Norm(200, m, r0);

% allocate memory for field components in x,y,z
Hx = zeros(41, 41, 41);
Hy = zeros(41, 41, 41);
Hz = zeros(41, 41, 41);

% compute without norming for each z layer and reshape results into layer
for i=1:41
    Hgrid = computeDipoleHField(X(:,:,i),Y(:,:,i),Z(:,:,i),m,H0norm);
    Hx(:,:,i) = reshape(Hgrid(1,:),41,41);
    Hy(:,:,i) = reshape(Hgrid(2,:),41,41);
    Hz(:,:,i) = reshape(Hgrid(3,:),41,41);
end

% calculate magnitude in each point for better view the results
Habs = sqrt(Hx.^2+Hy.^2+Hz.^2);

% define a index to view only every 4th point for not overcrowded plot
idx = 1:4:41;

% downsample and norm
Xds = X(idx,idx,idx);
Yds = Y(idx,idx,idx);
Zds = Z(idx,idx,idx);
Hxds = Hx(idx,idx,idx) ./ Habs(idx,idx,idx);
Hyds = Hy(idx,idx,idx) ./ Habs(idx,idx,idx);
Hzds = Hz(idx,idx,idx) ./ Habs(idx,idx,idx);

% show results for test, comment out for regular unittest run, run suite
% quiver3(Xds, Yds, Zds, Hxds, Hyds, Hzds);
% xlabel('$X$ in mm', ...
%         'FontWeight', 'normal', ...
%         'FontSize', 16, ...
%         'FontName', 'Times', ...
%         'Interpreter', 'latex');
% ylabel('$Y$ in mm', ...
%         'FontWeight', 'normal', ...
```

```
%         'FontSize', 16, ...
%         'FontName', 'Times', ...
%         'Interpreter', 'latex');
% zlabel('$Z$ in mm', ...
%         'FontWeight', 'normal', ...
%         'FontSize', 16, ...
%         'FontName', 'Times', ...
%         'Interpreter', 'latex');
% title('Dipole H-Field - Equation Test', ...
%         'FontWeight', 'normal', ...
%         'FontSize', 18, ...
%         'FontName', 'Times', ...
%         'Interpreter', 'latex');
% subtitle('$X$-, $Y$-, $Z$-Meshgrid from $-4 \ldots 4$ mm', ...
%         'FontWeight', 'normal', ...
%         'FontSize', 14, ...
%         'FontName', 'Times', ...
%         'Interpreter', 'latex');
% axis equal;

% pattern for logical indexing the center or opposite
p0 = false(1, 41);
p0(21) = true;
pN0 = true(41, 41, 41);
pN0(21,21, 21) = false;

% pattern for symmetry investigation
plu = [true(1, 20), false, false(1, 20)];
prl = [false(1, 20), false, true(1,  20)];

% compare values to check if fits in unit pairs of m and A/m
% and mm and kA/m
r0Apm = rotate3DVector([0; 0; -3e-3], 0, 0, 0);
H0normApm = computeDipoleH0Norm(200e3, m, r0Apm);
Xm = X * 1e-3;
Ym = Y * 1e-3;
Zm = Z * 1e-3;
HxApm = zeros(41, 41, 41);
HyApm = zeros(41, 41, 41);
HzApm = zeros(41, 41, 41);
for i=1:41
    HApm = computeDipoleHField(Xm(:,:,i),Ym(:,:,i),Zm(:,:,i),m,H0normApm);
    HxApm(:,:,i) = reshape(HApm(1,:),41,41);
    HyApm(:,:,i) = reshape(HApm(2,:),41,41);
    HzApm(:,:,i) = reshape(HApm(3,:),41,41);
end
HabsApm = sqrt(HxApm.^2+HyApm.^2+HzApm.^2);
```

Test 1: output dimensions

```
assert(isequal(size(Hsingle), [3, 1]))
assert(isequal(size(Hgrid), [3, 1681]))
```

Test 2: center of field

```
assert(X(p0,p0,p0) == 0)
assert(Y(p0,p0,p0) == 0)
assert(Z(p0,p0,p0) == 0)
assert(isnan(Hx(p0,p0,p0)))
assert(isnan(Hy(p0,p0,p0)))
```

```
assert(isnan(Hz(p0,p0,p0)))
assert(all(Hx(~p0,p0,p0) ~= 0, 'all'))
assert(all(Hx(p0,~p0,p0) ~= 0, 'all'))
assert(all(Hy(~p0,p0,p0) == 0, 'all'))
assert(all(Hy(p0,~p0,p0) == 0, 'all'))
assert(all(Hz(~p0,~p0,p0) == 0, 'all'))
```

Test 3: magnetization

```
assert(all(Hx(~p0,p0,~p0) ~= 0, 'all'))
assert(all(Hx(p0,~p0,~p0) ~= 0, 'all'))
assert(all(Hx(p0,p0,~p0) ~= 0, 'all'))
assert(all(Hy(~p0,p0,~p0) == 0, 'all'))
assert(all(Hy(p0,~p0,~p0) == 0, 'all'))
assert(all(Hy(p0,p0,~p0) == 0, 'all'))
assert(all(Hz(~p0,p0,~p0) == 0, 'all'))
assert(all(Hz(p0,~p0,~p0) ~= 0, 'all'))
assert(all(Hz(p0,p0,~p0) == 0, 'all'))
```

Test 4: imprinting

index 6 is 3mm and 36 is -3mm from surface where 200 kA/m should be imprinted

```
assert(round(abs(Hx(p0,p0,6)),6) == 200)
assert(round(abs(Hx(p0,p0,36)),6) == 200)
```

Test 5: symmetry

```
assert(all((Hx(plu,:,:) - flip(Hx(prl,:,:),1))==0, 'all'))
assert(all((Hx(:,plu,:) - flip(Hx(:,prl,:),2))==0, 'all'))
assert(all((Hy(plu,:,:) + flip(Hy(prl,:,:),1))==0, 'all'))
assert(all((Hy(:,plu,:) + flip(Hy(:,prl,:),2))==0, 'all'))
assert(all((Hz(:,:,~p0) + flip(Hz(:,:,~p0),2))==0, 'all'))
```

Test 6: units milli kilo

```
assert(all(round(HxApm(pN0) * 1e-3, 6) == round(Hx(pN0), 6), 'all'))
assert(all(round(HyApm(pN0) * 1e-3, 6) == round(Hy(pN0), 6), 'all'))
assert(all(round(HzApm(pN0) * 1e-3, 6) == round(Hz(pN0), 6), 'all'))
assert(all(round(HabsApm(pN0) * 1e-3, 6) == round(Habs(pN0), 6), 'all'))
```

Published with MATLAB® R2020b