computeDipoleHField

Computes the magnetic field strength H of a dipole magnet dependent of position and magnetic moment and imprint a field strength magnitude on the resultating field by passing a norm factor which relates to the rest position of the dipole magnet. The resultating field strength has field components in x, y and z direction.

The magnetic dipole moment w must be a column vector or shape

$$\vec{m} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$$

so that the magnetic moment corresponds to a position vector

$$\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

with coordinates for x, y and z in 3D coordinate system which can be taken part of its unit vector and its magnitude.

$$\vec{r} = \hat{r} \cdot |\vec{r}|$$

It computes the field strenght at this position with the current magnetic moment for field compents in the same orientation.

$$\vec{H}(\vec{r}) = \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix}$$

The originally equation of the magnetic dipole is known as

$$\vec{H}(\vec{r}) = \frac{\vec{B}(\vec{r})}{\mu_0}$$

$$\vec{H}(\vec{r}) = \frac{1}{4\pi} \cdot \frac{3\vec{r} \cdot (\vec{m}^T \cdot \vec{r}) - \vec{m} \cdot |\vec{r}|^2}{|\vec{r}|^5}$$

which can be simplified by putting in the unit vector of the position in into the equation.

$$\vec{H}(\vec{r}) = \frac{1}{4\pi |\vec{r}|^3} \cdot \left(3\hat{r} \cdot (\vec{m}^T \cdot \hat{r}) - \vec{m}\right)$$

To imprint a certain field strength related to a rest position of the dipole the resulting field strength is multiplied with a norming factor. The factor must be computed with same magnitude of the magnetic dipole moments which is passed to this computation to get correct field strengths. To get fields without imprinting set the norming factor to 1.

$$\vec{H}(\vec{r}) \cdot H_{0norm}$$

Contents

- Syntax

- Description
- Examples
- Input Argurments
- Output Argurments
- Requirements
- See Also

Syntax

```
H = computeDipoleHField(x, y, z, m, H0norm)
```

Description

H = computeDipoleHField(x, y, z, m, H0norm)compute dipole field strength at passed position (x,y,z) with the magnetic dipole moment m. The resulting field strenght is a vector with components in x, y and z direction. A field strength norming is imprinted on a rest position computation and multiplied on the result by multiplying a norm factor to the field. The normfactor must be relate to the same magnitude of the magnetic dipole moment which is used here and correspond to the magnets rest position in defined distance of the magnets surface.

Examples

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

% compute a 3D grid of positions
x = linspace(-10, 10, 40);
y = linspace(10, -10, 40);
z = linspace(10, -10, 40);
[X, Y, Z] = meshgrid(x, y, z);

% allocate memory for field components in x,y,z
Hx = zeros(40, 40, 40);
Hy = zeros(40, 40, 40);
Hz = zeros(40, 40, 40);

% compute without norming for each z layer and reshape results into layer
% magnetic moments points in -x direction which implies north and south pole
% is in x direction and rotation axes in z
for i=1:40
    H = computeDipoleHField(X(:,:,i),Y(:,:,i),Z(:,:,i),[-1;0;0],1);
    Hx(:,:,i) = reshape(H(1,:),40,40);
    Hy(:,:,i) = reshape(H(2,:),40,40);
    Hz(:,:,i) = reshape(H(3,:),40,40);
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 = 1:4:40;

% 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
quiver3(Xds, Yds, Zds, Hxds, Hyds, Hzds);
axis equal;
```

Input Argurments

x coordinates of positions at the field strength is calculated can be scalar, vector or matrix of coordinates. Must be same size as y and z.

y coordinates of positions at the field strength is calculated can be scalar, vector or matrix of coordinates. Must be same size as x and z.

z coordinates of positions at the field strength is calculated can be scalar, vector or matrix of coordinates. Must be same size as x and y.

m magnetic dipole moment as 3 x 1 vector. The magnetic field strength is calculated with the same moment for all passed positions.

H0norm scalar factor to imprint a field strength to the dipole field. Must be computed with the same magnitude of passed magnetic moment vector. Set 1 to disable imprinting.

Output Argurments

H computed magnetic field strength at passed positions with related magnetic moment. If passed position is scalar H has size of 3 X 1 with its components in x, y and z dircetion. H(1) -> x, H(2) -> y and H(3) -> z. If passed positions are not scalar H has size of 3 x numel(x) with position relations in columns. So reshape rows to shapes of positions to keep orientation as origin.

Requirements

- Other m-files required: None
- Subfunctions: mustBeEqualSize
- MAT-files required: None

See Also

- generateDipoleRotationMoments
- generateSensorArraySquareGrid
- computeDipoleH0Norm


```
function [H] = computeDipoleHField(x, y, z, m, H0norm)
    arguments
        % validate position, can be any size but must be same size of
        x (:,:,:) double {mustBeReal}
        y (:,:,:) double {mustBeReal, mustBeEqualSize(x, y)}
        z (:,:,:) double {mustBeNumeric, mustBeReal, mustBeEqualSize(y, z)}
        % validate magnetic moment as 3 x 1 vector
        m (3,1) double {mustBeReal, mustBeVector}
        % validate norm factor as scalar
        H0norm (1,1) double {mustBeReal}
    end
```

```
    % unify positions to column vector or matrix of column vectors if positions
    % are not passed as column vectors or scalar, resulting size of position R
    % is 3 x length(X), a indication if is column vector is not needed because
    % x(:) is returning all content as column vector. Transpose to match shape.
    r = [x(:), y(:), z(:)]';

    % calculate the magnitude of all positions
    rabs = sqrt(sum(r.^2, 1));

    % calculate the the unit vector of all positions
    rhat = r ./ rabs;

    % calculate H-field of current magnetic moment for all passed positions
    % calculate constants in eqution once in the first bracket term, all vector
    % products in the second term and finially divide by related magnitude ^3
    H = (H0norm / 4 / pi) * (3 * rhat .* (m' * rhat) - m) ./ rabs.^3;
end

% Custom validation function
function mustBeEqualSize(a,b)
    % Test for equal size
    if ~isequal(size(a),size(b))
        eid = 'Size:notEqual';
        msg = 'X Y Z positions must be the same size and orientation.';
        throwAsCaller(MException(eid,msg))
    end
end
```

Published with MATLAB® R2020b