plotDipoleMagnet

Plot dipole magnet which approximate a spherical magnet in its far field.

Syntax

```
plotDipoleMagnet()
```

Description

plotDipoleMagnet() load dipole constants from config.mat and construct magnet in its rest position in x and z layer for y = 0.

Examples

```
plotDipoleMagnet();
```

Input Arguments

None

Output Arguments

None

Requirements

- Other m-files: generateDipoleRotationMoments.m, computeDipoleH0Norm.m, computeDipoleHField
- Subfunctions: none
- MAT-files required: data/config.mat

See Also

- quiver
- imagesc
- streamslice


```
function plotDipoleMagnet()
    try
        % load dataset path and dataset content into function workspace
        load('config.mat', 'PathVariables', 'DipoleOptions');
%          close all;
    catch ME
        rethrow(ME)
    end

    % figure save path for different formats
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    figFilename = 'dipole_magnet';
    figPath = fullfile(PathVariables.saveImagesPath, figFilename);

    % load needed data from dataset in to local variables for better handling
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Radius in mm of magnetic sphere in which the magnetic dipole is centered.
    % So it can be seen as z-offset to the sensor array.
```

```
rsp = DipoleOptions.sphereRadius;

% H-field magnitude to multiply of generated and relative normed dipole
Hmag = DipoleOptions.H0mag;

% Distance in zero position of the spherical magnet in which is imprinted
z0 = DipoleOptions.z0;

% Magnetic moment magnitude attach rotation to the dipole field
m0 = DipoleOptions.M0mag;

% clear dataset all loaded
clear DipoleOptions;

% set construction dipole magnet, all length in mm and areas mm^2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% number of samples for good looking
nSamples = 501;
% slice in view for quiver, every 25th point
slice = 25:25:nSamples-25;
% grid edge of meshgrid, square grid
xz = 15;
% y layer in coordinate system
y = 0;
% orientat of magnet along z axes
pz = pi/2:0.01:3*pi/2;
% distances magnet surface to display in plot
zd = -rsp:-z0:-xz;
xd = zeros(1, length(zd));
% scale grid to simulate
x = linspace(-xz, xz, nSamples);
z = linspace(xz, -xz, nSamples);
[X, Z, Y] = meshgrid(x, z, y);

% compute dipole and fetch to far field to approximate a sperical magnet
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% generate dipole moment for 0°
m = generateDipoleRotationMoments(m0, 1);
% compute H-field norm factor imprieng H magnitude on dipole, rest position
H0norm = computeDipoleH0Norm(Hmag, m, [0; 0 ;-(z0 + rsp)]);
% compute dipole H-field for rest position in y = 0 layer
H = computeDipoleHField(X, Y, Z, m, H0norm);
% calculate magnitudes for each point in the grid
Habs = reshape(sqrt(sum(H.^2, 1)), nSamples, nSamples);
% split H-field in componets and reshape to meshgrid
Hx = reshape(H(1,:), nSamples, nSamples) ./ Habs;
Hy = reshape(H(2,:), nSamples, nSamples) ./ Habs;
Hz = reshape(H(3,:), nSamples, nSamples) ./ Habs;
% exculde value within the spherical magnet, < rsp
innerField = X.^2 + Z.^2 <= rsp.^2;
Habs(innerField) = NaN;
% find relevant magnitudes at anounced distances
Hd = interp2(X, Z, Habs, xd, zd, 'nearest', NaN);


% figure dipole magnet
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fig = figure('Name', 'Dipole Magnet');
```

```matlab
% plot magnitude as colormap
imagesc(x, z, log10(Habs), 'AlphaData', 1);
set(gca, 'YDir', 'normal');
colormap('jet');
shading flat;

% set colorbar to log10 scaling of map
cb = colorbar;
cb.Label.String = '$log_{10}(|H|)$ in kA/m';
cb.TickLabelInterpreter = 'latex';
cb.Label.Interpreter = 'latex';
cb.Label.FontSize = 24;

hold on;
grid on;

% plot field lines
st = streamslice(X, Z, Hx, Hz, 'arrows', 'cubic');
set(st, 'Color', 'k');


% plot magnet with north and south pole
rectangle('Position', [-rsp -rsp 2*rsp 2*rsp], ...
    'Curvature', [1 1], 'LineWidth', 3.5);
semicrc = rsp.*[cos(pz); sin(pz)];
patch(semicrc(1,:), semicrc(2,:),'r');
patch(-semicrc(1,:), -semicrc(2,:),'g');
text(-1.25, 0, '\textbf{N}');
text(0.5, 0, '\textbf{S}');

% additional figure text and lines
text(-(xz-1), -(xz-1), ...
    sprintf('$\\mathbf{Y = %.1f}$ \\textbf{mm}', y), 'Color', 'w');

% distance scale in -z direction for x =0, distance from magnet surface
line(xd, zd, 'Marker', '_', 'LineStyle', '-', 'LineWidth', 3.5, ...
    'Color', 'w');

% place text along marker
for i = 2:length(zd)-1
    markstr = "\\textbf{$\\mathbf{%d}$ mm, $\\mathbf{%.1f}$ kA/m}";
    mark = sprintf(markstr, abs(zd(i))-rsp, Hd(i));
    text(0.5, zd(i), mark, 'Color', 'w');
end

% limits ticks and labels
xlim([-xz xz]);
ylim([-xz xz]);
xticks(-xz:xz);
yticks(-xz:xz);
labels = string(xticks);
labels(1:2:end) = "";
xticklabels(labels)
yticklabels(labels)
xlabel('$X$ in mm');
ylabel('$Z$ in mm');

% axis shape set
axis equal;
axis tight;
grid off;
```

```
    % title and description
    disp('Title: Approximated Spherical Magnet with Dipole Far Field');
    disp("Description:");
    fprintf("Sphere whith imprinted H-field magnitude of %.1f kA/m\n", Hmag); ...
    fprintf("at distance d = %.1f mm with d_z = |z| - r\n", z0);
    fprintf("and sphere radius r = %.1f mm\n", rsp);

    % save results of figure
%     yesno = input('Save? [y/n]: ', 's');
%     if strcmp(yesno, 'y')
%         savefig(fig, figPath);
%         print(fig, figPath, '-dsvg');
%         print(fig, figPath, '-depsc', '-tiff', '-loose');
%         print(fig, figPath, '-dpdf', '-loose', '-fillpage');
%     end
%     close(fig)
end
```

Published with MATLAB® R2020b