

## generateDipoleRotationMoments

Generate magnetic moments to perform a full rotation of a magnetic dipole in the z-axes with a certain tilt. The moments covers a rotation from 0° to 360° and are equal distributed between 0° and 360°. 0° and 360° are related to the first moment which is represented by the start vector of

$$\vec{m}_0 = |m_0| \cdot \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

Due to the start vector position the tilt of z-axes must be applied with a tilt angle in y-axes. So the rotated vector of the start moment is described by

$$\vec{m}_i = R_z(\theta_i)R_y(\phi)R_x(0^\circ)\vec{m}_0$$

The returning Moments matrix is 3 x N matrix where each moment vector

$$\vec{M} = [\vec{m}_1 \cdots \vec{m}_N]$$

corresponds to a i-th angle in 1 x N thetas vector.

$$\vec{\theta} = [\theta_1 \cdots \theta_N]$$

for

$$i = 1 \cdots N$$

The resolution of the angles can be modified additionally. At first the full angle vector theta is fully generated with given resolution and downsampled afterwards to the defined number of angles. On the resulting theta vector is base of magnetical moments.

### Contents

---

- [Syntax](#)
- [Description](#)
- [Examples](#)
- [Input Arguments](#)
- [Output Arguments](#)
- [Requirements](#)
- [See Also](#)

### Syntax

---

```
M = generateDipoleRotationMoments(m0, nTheta)
[M, thetas] = generateDipoleRotationMoments(m0, nTheta)
[M, thetas] = generateDipoleRotationMoments(m0, nTheta, phi)
[M, thetas] = generateDipoleRotationMoments(m0, nTheta, phi, resolution)
[M, thetas, index] = generateDipoleRotationMoments(m0, nTheta, phi, resolution, phaseIndex)
```

### Description

---

**M = generateDipoleRotationMoments(m0, nTheta)** generate magnetic moments for N numbers of rotation angles theta in 3 x N sized matrix. With a default angle resolution of 1° and a start angle of 0°.

**[M, theta] = generateDipoleRotationMoments(m0, nTheta)** returns so magnetic moments as before and related angles theta as 1 x N vector.

**[M, theta] = generateDipoleRotationMoments(m0, nTheta, phi)** generate magnetic moments for a rotation with a tilt angle phi.

**[M, theta] = generateDipoleRotationMoments(m0, nTheta, phi, resolution)** return moments and angles like described above but with given resolution in degree. The resolution is used in generation of full scale rotation angle base and sometime not visible in the output caused by the number of angles. So which angle are even picked from full scale rotation to

compute a down sampled set of angles.

**[M, theta, index] = generateDipoleRotationMoments(m0, nTheta, phi, resolution, phaseIndex)** returns the moments, the angles and index representation of down sampled angles in the full scale rotation vector.

## Examples

---

```
% choose a huge moment amplitude to withdraw numeric erros in later H-field
% strength calculations
m0 = 1e6;

% get a full scale (FS) rotation of with 0.5° resolution and no tilt
[MFS, thetaFS] = generateDipolRotationMoments(m0, 0, 0, 0.5);

% get down sampled (DS) rotation with equal distanced angles of the same full
% scale and refered index to the full scale. 8 angles.
[MDS, thetaDS, iFS] = generateDipolRotationMoments(m0, 8, 0, 0.5);

% check distribution to full scale must be true if distribution is correct
all(MFS(iFS) == MDS)
all(thetaFS(iFS) == thetaDS)

% now shift the sample pick by 22 samples (11° with resolution of 0.5°)
[MDSS, thetaDSS] = generateDipolRotationMoments(m0, 8, 0, 0.5, 22);

% check with index shift by 22 in iFS index
all(MFS(iFS + 22) == MDSS)
all(thetaFS(iFS + 22) == thetaDSS)
```

## Input Arguments

---

**m0** scalar value of magnetic moment magnitude. Choose huge value to prevent numeric failures in later field strength calculation. 1e6 is a proven value. The is later normated in the field calculation process. Can be any real number.

**nTheat** scalar value and number of angles which are even picked from the full rotation to produce smaller rotation datasets. Must be a positive integer or zero. If zero the full scale rotation is returned.

**phi** scalar angle in degree to tilt the z-axes of the rotation. Can be any real number. Default is 0°.

**resolution** scalar angle resolution must be real positive number and probably smaller than 360°. Default is 1°.

**phaseIndex** scalar integer number to shift the start index of down sampling the full scale rotation. Therefore nTheta must be greater than 0. Default is 0.

## Output Arguments

---

**M** matrix of magnetic moments related to vector theta. Matrix of size 3 x N.

**theta** related angles to calculated magnetic moments in a row vector of size 1 x N.

**index** reference to full scale angle vector. Empty if nTheta is zero and theta is the full scale vector.

## Requirements

---

- Other m-files required: rotate3DVector.m
- Subfunctions: length, downsample, ismember, find
- MAT-files required: None

## See Also

---

- [rotate3DVector](#)
- [downsample](#)
- [ismember](#)
- [find](#)

```

function [M, theta, index] = generateDipoleRotationMoments(m0, nTheta, ...
    phi, resolution, phaseIndex)
arguments
    % validate amplitude of magnetic moment as real scalar value
    m0 (1,1) double {mustBeReal}
    % validate number of used angulars as positive integer, for 0 return all
    nTheta (1,1) double {mustBeNonnegative, mustBeInteger}
    % validate tilt angle as real value with default 0°
    phi (1,1) double {mustBeReal} = 0
    % validate angle resolution as real positive value
    resolution (1,1) double {mustBePositive} = 1
    % validate downsample phase as positive integer with default 0, no shift
    phaseIndex (1,1) double {mustBeNonnegative, mustBeInteger} = 0
end

% scale full rotation angle vector with given resolution from 0° to 360°
% so run to 360°-resolution because 0° == 360°, its a circle
fullScale = 0:resolution:(360 - resolution);

% if nThetas is greater than 0 downsample to nTheta else use full scale
if nTheta
    % get equal distribute distance of samples in thetas for nThetas
    sampleDistance = length(downsample(fullScale, nTheta));

    % downsample with equal sample distance and passed sample phase to shift
    % first sample in downsample vector from 1 to phaseIndex
    theta = downsample(fullScale, sampleDistance, phaseIndex);

    % find index members of down sampled angles in full scale vector
    members = ismember(fullScale, theta);
    index = find(members);

else
    % 0 is given for number of theta so it returns the full scale rotation
    % no index relations if full scale is returned
    nTheta = length(fullScale);
    theta = fullScale;
    index = [];
end

% create start moment with given magnetic moment amplitude basic moment to
% produce rotate moments
m0 = m0 * [-1; 0; 0];

% allocate memory for the moments Matrix of rotated basic moments by i-th
% theta and fixed tilt of phi and rotate of theta angulars
M = zeros(3, nTheta);
for i = 1:nTheta
    M(:,i) = rotate3DVector(m0, 0, phi, theta(i));
end
end

```