CermitESR(B0, B1, cantilever, f_rf, grid, h, magnet, mw_x_0p, sample)
returns: df_spin
collection: CermitESR
graph: CermitESR_graph
handler: MemHandler
components:
- magnet: ['Bz_method', 'Bzx_method', 'Bzxx_method']
- sample: ['J', 'Gamma', 'spin_density', 'temperature', 'dB_sat', 'dB_hom']
- grid: ['grid_array', ['grid_shape', 'shape'], ['grid_step', 'step'], [...]
- cantilever: ['k2f_modulated']

CERMIT ESR experiment for a large tip.

grid extended

extend_grid(extend_grid_by_length, mw_x_0p)
return: ext_grid
functype: function

↓ ext_grid

Bz extended

field_func(Bz_method, ext_grid, h)
return: ext_Bz
functype: function

Calculate the field value at the given height and grid points.

↓ ext_Bz

B_tot extended

add(ext_Bz, B0)
return: ext_B_tot
functype: builtin_function_or_method

Calculate combined magnetic field.

→ ext_B_tot

ext_B_tot

B_offset extended

B_offset(ext_B_tot, f_rf, Gamma)
return: ext_B_offset
functype: numba.core.registry.CPUDispatcher

Calculate the resonance offset.

x_0p window pts

convert_grid_pts(mw_x_0p, grid_step)
return: ext_pts
functype: function

Convert distance to ext points.

B_tot sliced

slice_matrix(ext_B_tot, grid_shape)
return: B_tot
functype: function

Slice numpy matrix.

↓ ext_B_offset   ↓ ext_pts   ↓ B_tot

minimum absolute x offset

min_abs_offset(ext_B_offset, ext_pts)
return: B_offset
functype: function

Minimum absolute value of a matrix in x direction based on the window.

mz_eq

mz_eq(B_tot, Gamma, J, temperature)
return: mz_eq
functype: numba.core.registry.CPUDispatcher

Magnetization per spin at the thermal equilibrium using the Brillouin function.

↓ B_offset

rel_dpol

rel_dpol_sat_steadystate(B_offset, B1, dB_sat, dB_hom)
return: rel_dpol
functype: numba.core.registry.CPUDispatcher

Relative change in polarization for steady-state.

Bzxx

field_func(Bzxx_method, grid_array, h)
return: Bzxx
functype: function

Calculate the field value at the given height and grid points.

↓ rel_dpol   ↓ mz_eq   ↓ Bzxx

spring constant shift

neg_sum_of_product(Bzxx, rel_dpol, mz_eq, spin_density, grid_voxel)
return: dk_spin
functype: function

Calculate the negative sum of the product input values.

↓ dk_spin

frequency shift

mul(dk_spin, k2f_modulated)
return: df_spin
functype: builtin_function_or_method

Convert the spring constant shift to frequency shift.