

# **Building a modular simulation platform for magnetic resonance force microscopy (MRFM) experiments (*mrfmsim* and *mmodel*)**

Peter Sun, John Marohn

Dept. Chemistry and Chemical Biology, Cornell University

July 29, 2024



# Table of contents

## 1 Motivation

---

- ▶ Motivation
- ▶ Magnetic Resonance Force Microscopy (MRFM)
- ▶ Challenges
- ▶ Backend — *mmodel*
- ▶ *mrfmsim*
- ▶ Conclusions

# We are graduate school experimentalists

## 1 Motivation

---

Code quality is not a priority as long as it works.

# We are graduate school experimentalists

## 1 Motivation

---

Code quality is not a priority as long as it works.

To simulate a new experiment:

1. copy a previous script that shares the most common parts  
(Jupyter Notebook script)
2. change/update parts that need to be changed
3. production

# We are graduate school experimentalists

## 1 Motivation

---

Code quality is not a priority as long as it works.

To simulate a new experiment:

1. copy a previous script that shares the most common parts  
(Jupyter Notebook script)
2. change/update parts that need to be changed
3. production

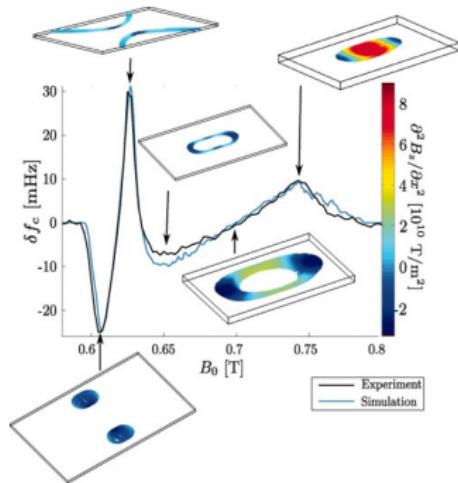
Missing:

✗ comments  
✗ tests

- error-prone: error propagation, lack of tests
- unreadable
- slow, difficult to optimize, time spent on debugging

# The lack of tests led to a publication error

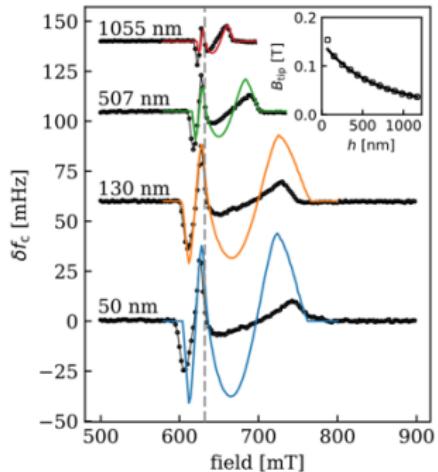
## 1 Motivation



“The numerical simulation agrees remarkably well with the observed signal.”

Moore, E.W. et al., *Proceedings of the National Academy of Sciences*, 2009, **106**, 22251–22256

The grid voxel size was too large.



Isaac, C.E. *Harnessing electron spin labels for single molecule magnetic resonance imaging* Ph.D. Dissertation, Cornell University, Ithaca, NY, 2018.

# Motivation

## 1 Motivation

---

Can we create a modular simulation platform

for a **new technique**  
in the **development phase**  
in a **graduate research group setting**

that helps reduce errors in production?

# Motivation

## 1 Motivation

---

Can we create a modular simulation platform

for a **new technique**  
in the **development phase**  
in a **graduate research group setting**

that helps reduce errors in production?

In this talk

- Overview of MRFM imaging experiments
- Challenges in modeling MRFM experiments
- Highlight features of *mrfmsim* and its backend *mmodel*

# Table of contents

## 2 Magnetic Resonance Force Microscopy (MRFM)

---

- ▶ Motivation
- ▶ Magnetic Resonance Force Microscopy (MRFM)
- ▶ Challenges
- ▶ Backend — *mmodel*
- ▶ *mrfmsim*
- ▶ Conclusions

# Goal

## 2 Magnetic Resonance Force Microscopy (MRFM)

**Goal:** High-resolution 3D imaging of individual biomolecules and protein assembly

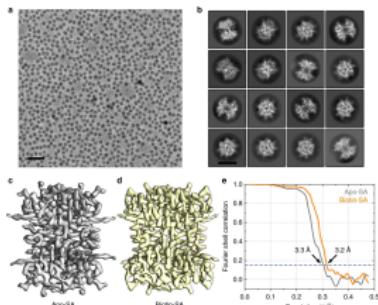
# Goal

## 2 Magnetic Resonance Force Microscopy (MRFM)

**Goal:** High-resolution 3D imaging of individual biomolecules and protein assembly

### CryoEM

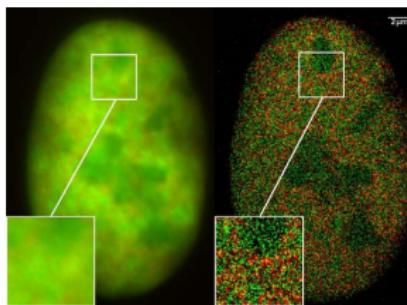
- requires a large number of identical copies
- sample damage



Fan, X. et al., *Nature Communications*, 2019, 10, 2386

### Super-resolution optical microscopy

- $\sim 10$  nm resolution
- fluorescence labels



Gunkel, M. et al., *Biotechnology Journal*, 2009, 4, 927–938

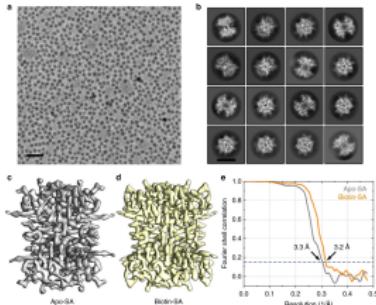
# Goal

## 2 Magnetic Resonance Force Microscopy (MRFM)

**Goal:** High-resolution 3D imaging of individual biomolecules and protein assembly

### CryoEM

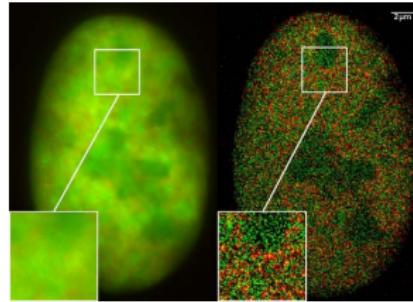
- requires a large number of identical copies
- sample damage



Fan, X. et al., *Nature Communications*, 2019, 10, 2386

### Super-resolution optical microscopy

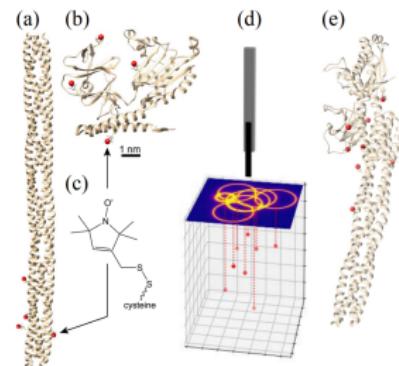
- $\sim 10$  nm resolution
- fluorescence labels



Gunkel, M. et al., *Biotechnology Journal*, 2009, 4, 927–938

### MRFM (proposed)

- sub-Å resolution
- electron radical spin labels (nitroxide)



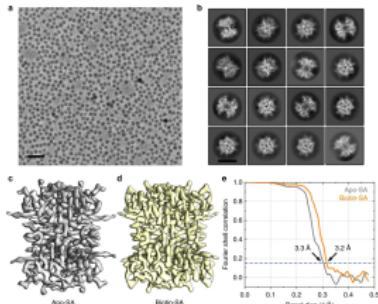
# Goal

## 2 Magnetic Resonance Force Microscopy (MRFM)

**Goal:** High-resolution 3D imaging of individual biomolecules and protein assembly

### CryoEM

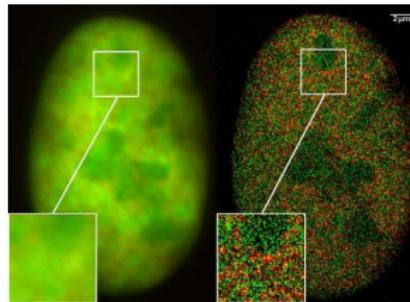
- requires a large number of identical copies
- sample damage



Fan, X. et al., *Nature Communications*, 2019, 10, 2386

### Super-resolution optical microscopy

- $\sim 10$  nm resolution
- fluorescence labels



Gunkel, M. et al., *Biotechnology Journal*, 2009, 4, 927–938

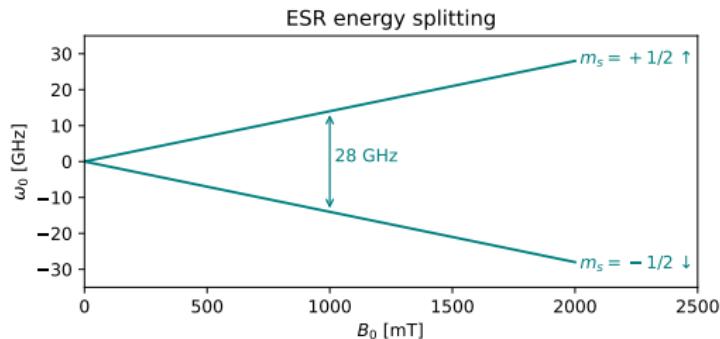
### MRFM (proposed)

- sub-Å resolution
- electron radical spin labels (nitroxide)

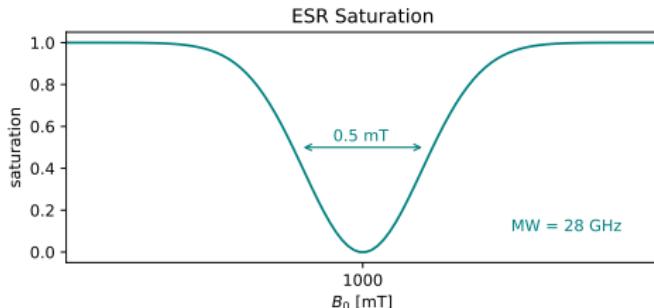
key proposal  
detect positions of single spins

# MRFM with spin resonance (ESR example)

## 2 Magnetic Resonance Force Microscopy (MRFM)

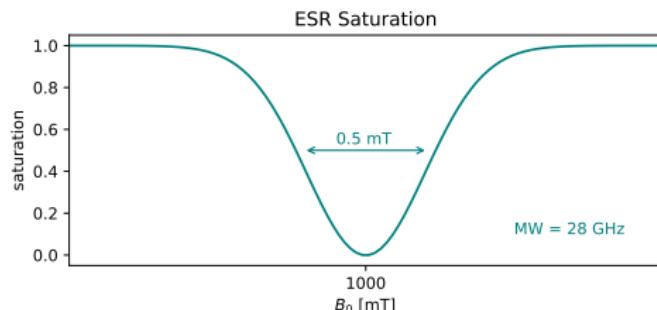
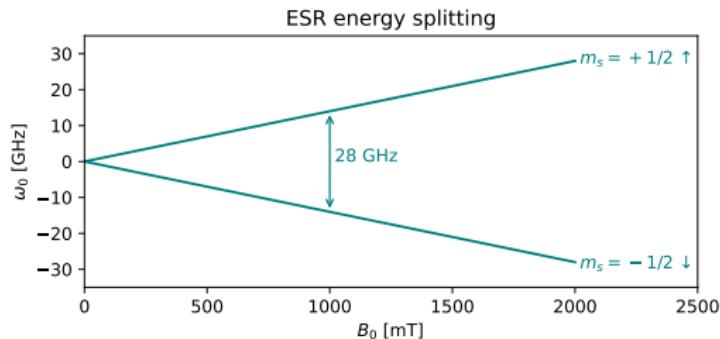


- The net spin magnetic moment flips when MWs match the Larmor frequency:  
$$\omega_0 = \gamma_e B_0$$
- $\gamma_e = 28 \text{ GHz/T}$ , electron gyromagnetic ratio



# MRFM with spin resonance (ESR example)

## 2 Magnetic Resonance Force Microscopy (MRFM)



- The net spin magnetic moment flips when MWs match the Larmor frequency:  
$$\omega_0 = \gamma_e B_0$$
- $\gamma_e = 28$  GHz/T, electron gyromagnetic ratio

- resolution:  
$$\frac{0.5\text{mT}}{5\text{mT/nm}} = 0.1 \text{ nm}$$
- good SNR in 1 second of signal averaging

# MRFM animation

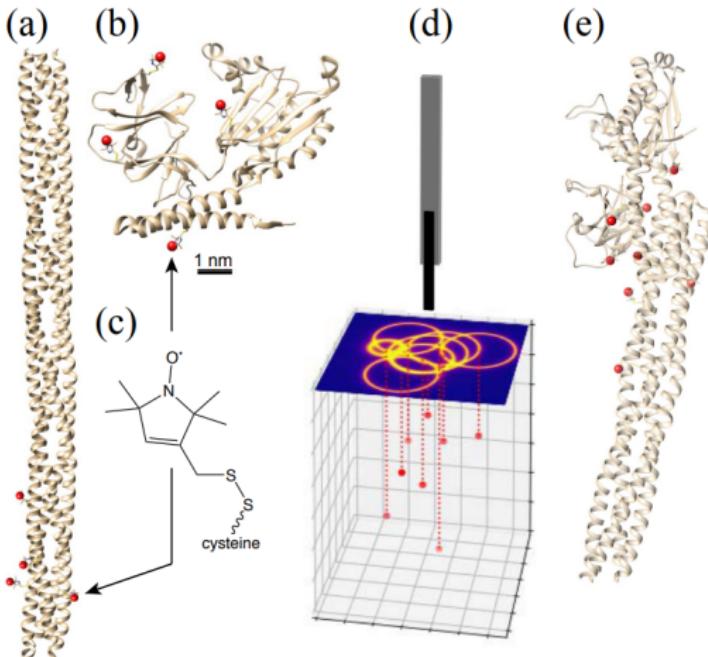
## 2 Magnetic Resonance Force Microscopy (MRFM)

---

- isolate electron spin with tip magnetic field
- couple magnetic moment changes with cantilever frequency change

# MRFM imaging: electron radical spin labels

## 2 Magnetic Resonance Force Microscopy (MRFM)



Chemotaxis proteins, nitroxide labels at cysteine residue,  
PDB: 3ja6

MRFM can determine protein structure and protein assembly:

- Each circle represents one electron position.
- Circle diameter  $\propto$  electron depth

Simulation is crucial!

# Table of contents

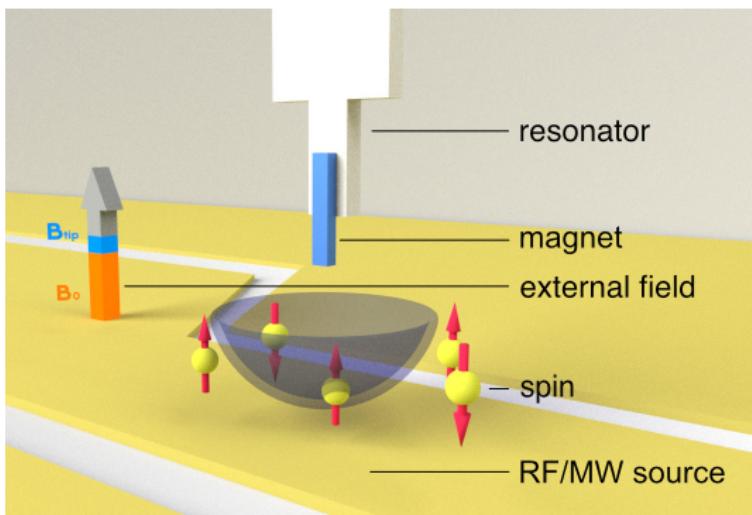
## 3 Challenges

---

- ▶ Motivation
- ▶ Magnetic Resonance Force Microscopy (MRFM)
- ▶ Challenges
- ▶ Backend — *mmodel*
- ▶ *mrfmsim*
- ▶ Conclusions

# New technique: common but differentiated components

## 3 Challenges



# New technique: common but differentiated components

## 3 Challenges

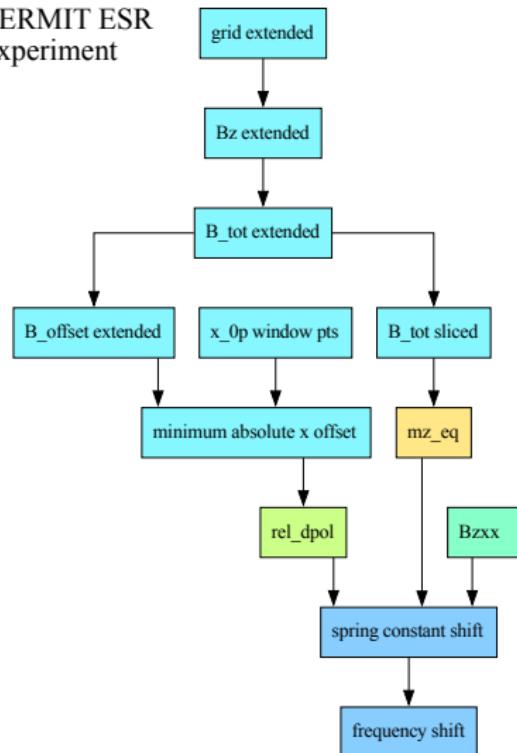
common but differentiated components

- resonator: cantilever, membrane, etc
  - magnet: different geometries and materials
  - sample: electron, nuclear
  - setup: magnet on cantilever, sample on cantilever
- ⇒  modular — can keep the code structure but have interchangeable parts

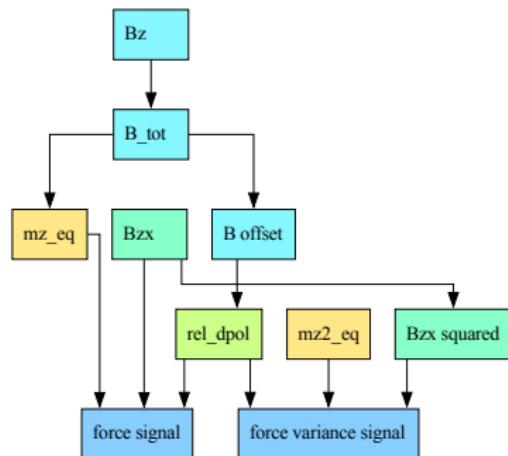
# Development phase: different protocols

## 3 Challenges

CERMIT ESR Experiment



Cyclic Inversion Experiment



field  
field gradient  
polarization

equilibrium magnetization  
signal

# Development phase: different protocols

## 3 Challenges

---

different detection protocols

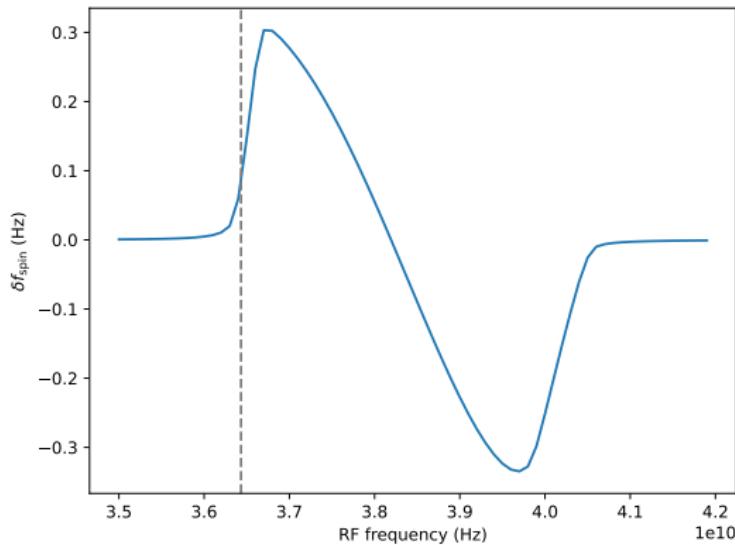
- CERMIT
- i-OSCAR
- Cyclic inversion
- Cyclic saturation
- ...



- flexible (not a fixed sequence of processes)
- readable

# Development phase: post definition modification

## 3 Challenges



- additional modification to simulation —loop parameters



- flexible to modify (add optimized looping)

# Graduate students: different programming illiteracy

## 3 Challenges

---

- limited time
  - different levels of programming experience
  - collaboration
- ⇒
- readable
  - easy to test
  - easy to run
  - easy to optimize
  - easy to share

# Design a simulation platform for MRFM

## 3 Challenges

---

In summary

- modular
- flexible (structure and for modification)
- readable
- easy to test, run, optimize, and share
- can use existing codebase
- open source

# Table of contents

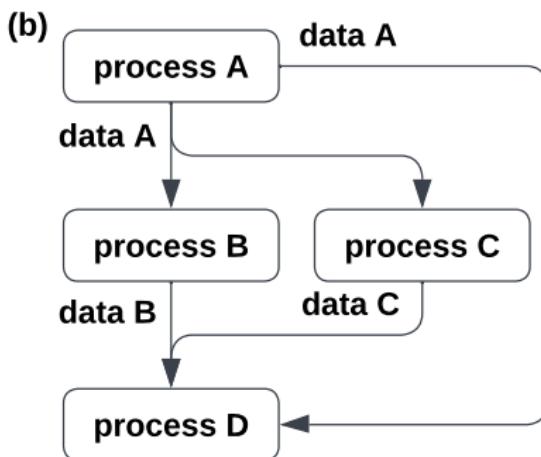
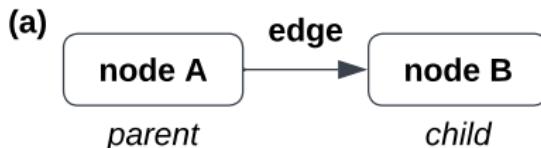
## 4 Backend — *mmodel*

---

- ▶ Motivation
- ▶ Magnetic Resonance Force Microscopy (MRFM)
- ▶ Challenges
- ▶ Backend — *mmodel*
- ▶ *mrfmsim*
- ▶ Conclusions

# Workflow with directed acyclic graph (DAG)

## 4 Backend — *mmodel*



- ✓ modular and interchangeable nodes
- ✓ flexible for different protocols
- ✓ readable with graph representation
- ✓ easy to test

# ***mmodel***: modular modeling for development phase

## 4 Backend — *mmodel*

---

Many workflow systems focus on well-defined processes and large data analysis

# ***mmodel*: modular modeling for development phase**

## 4 Backend — *mmodel*

---

Many workflow systems focus on well-defined processes and large data analysis

*mmodel* focuses on creating a workflow for techniques in the development phase, most notably

- lightweight
- flexible for modification
- can use existing Python functions

# **mmodel:** modular modeling for development phase

## 4 Backend — *mmodel*

Many workflow systems focus on well-defined processes and large data analysis

*mmodel* focuses on creating a workflow for techniques in the development phase, most notably

- lightweight
- flexible for modification
- can use existing Python functions

In *mmodel*'s DAG is based on *NetworkX*; is visualized using dot graph with *Graphviz*.

Hagberg, A.A., Schult, D.A. and Swart, P.J. In ed. Varoquaux, G., Vaught, T. and Millman, J., *Proceedings of the 7th python in science conference*, pp. 11 – 15, Pasadena, CA USA, 2008

Ellson, J. et al. in *Graph Drawing*, ed. Goos, G. et al., Vol. 2265; Springer Berlin Heidelberg, Berlin, Heidelberg, 2002;  
pp. 483–484

# mmodel: an example

## 4 Backend — mmodel

```
import math
import numpy as np
from mmodel import Graph, Model, Node, MemHandler

def mul(sum_xy, log_xy):
    """Multiple the inputs."""
    return sum_xy * log_xy

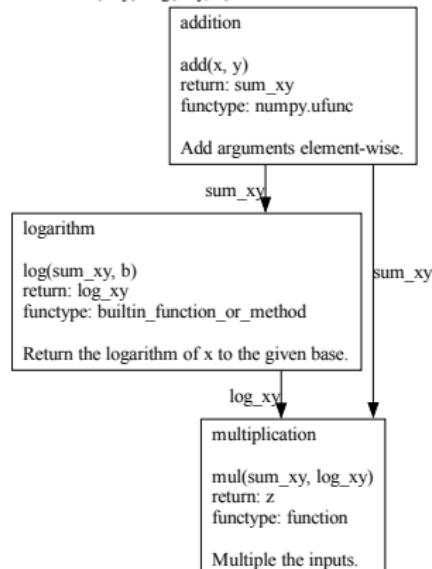
grouped_edges = [
    ("addition", ["logarithm", "multiplication"]),
    ("logarithm", "multiplication"),
]
node_objects = [
    Node("addition", np.add, ["x", "y"],
         output="sum_xy"),
    Node("logarithm", math.log, ["sum_xy", "b"],
         output="log_xy"),
    Node("multiplication", mul, output="z"),
]
G = Graph(name="example_graph")
G.add_grouped_edges_from(grouped_edges)
G.set_node_objects_from(node_objects)

example_model = Model(
    "example_model",
    G,
    handler=MemHandler,
    doc="Result of (x+y)*log(x+y, b).",
)
example_model.visualize()
```

*mmodel* representation of  
 $z = (x + y) \log_b(x + y)$

```
example_experiment(b, x, y)
returns: z
graph: example_graph
handler: MemHandler
```

Result of  $(x+y)*\log(x+y, b)$ .



# mmodel: an example

## 4 Backend — mmodel

```
import math
import numpy as np
from mmodel import Graph, Model, Node, MemHandler

def mul(sum_xy, log_xy):
    """Multiple the inputs."""
    return sum_xy * log_xy

grouped_edges = [
    ("addition", ["logarithm", "multiplication"]),
    ("logarithm", "multiplication"),
]
node_objects = [
    Node("addition", np.add, ["x", "y"],
         output="sum_xy"),
    Node("logarithm", math.log, ["sum_xy", "b"],
         output="log_xy"),
    Node("multiplication", mul, output="z"),
]
G = Graph(name="example_graph")
G.add_grouped_edges_from(grouped_edges)
G.set_node_objects_from(node_objects)

example_model = Model(
    "example_model",
    G,
    handler=MemHandler,
    doc="Result of (x+y)*log(x+y, b).",
)
example_model.visualize()
```

```
>>> example_model(2, 5, 3)
>>> 24.0
```

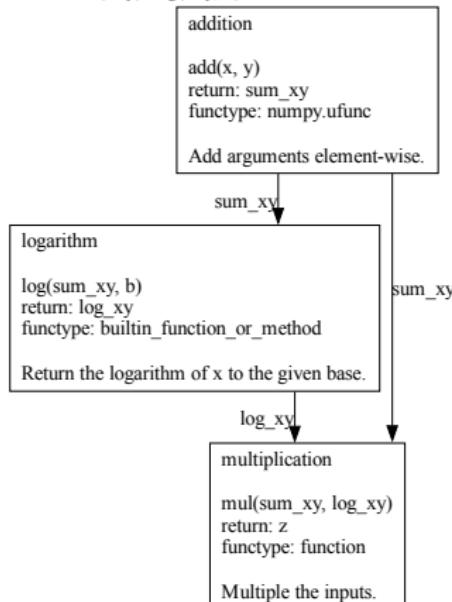
Sun, P. and Marohn, J.A., *The Journal of Chemical Physics*, 2023, 159, 044801

# Any function can be used as a node

## 4 Backend — *mmodel*

```
example_experiment(b, x, y)
returns: z
graph: example_graph
handler: MemHandler
```

Result of  $(x+y)*\log(x+y, b)$ .



Functions are converted into thinly wrapped Node objects:

```
node = Node(
    "addition",
    np.add,
    ["x", "y"],
    output="sum_xy",
)
```

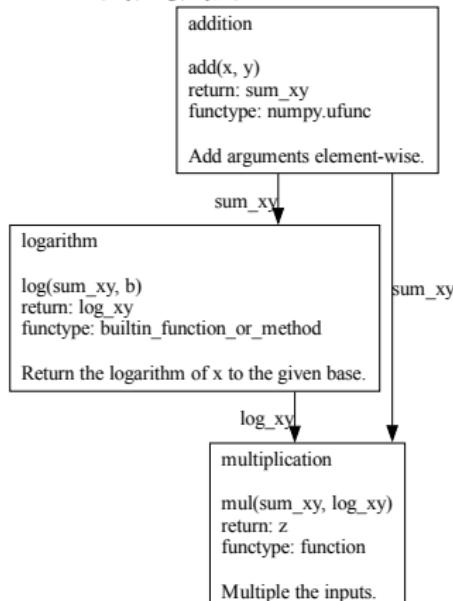
- allows all Python callable type
  - numpy.ufunc
  - Python builtin functions and methods
  - lambda function
  - `mmodel.Model` objects
- lightweight

# Handler: handles data flow

## 4 Backend — *mmodel*

```
example_experiment(b, x, y)
returns: z
graph: example_graph
handler: MemHandler
```

Result of  $(x+y) * \log(x+y, b)$ .



Available handlers:

- **BasicHandler**: store all values in a dictionary
- **MemHandler**: deletes intermediates values that are no longer in use to reduce peak memory
- **H5Handler**: stores intermediate values in HDF5 file format

```
example_model = Model(
    "example_model",
    G,
    handler=MemHandler,
    doc="Result of (x+y)*log(x+y, b).",
    returns=["sum_xy", "z"])
```

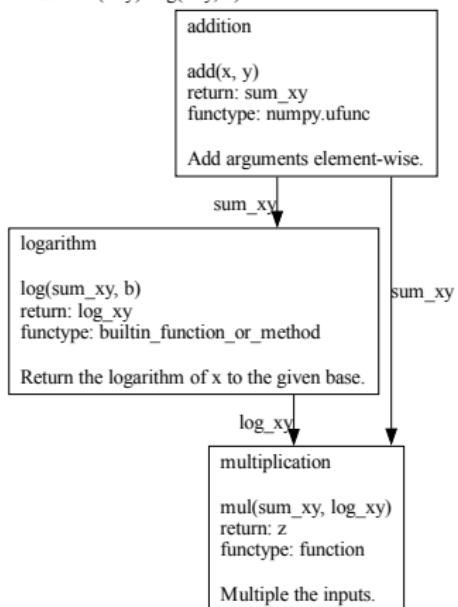
)

# mmodel: post definition modification

## 4 Backend — mmodel

```
example_experiment(b, x, y)
returns: z
graph: example_graph
handler: MemHandler
```

Result of  $(x+y) * \log(x+y, b)$ .



```
node = Node(
    "addition",
    np.add,
    ["x", "y"],
    output="sum_xy",
    modifiers=[functools.cache],
)
```

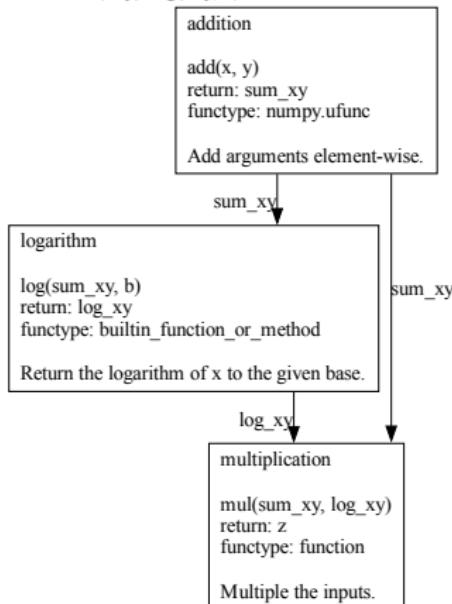
- **modifiers:** modify nodes
- **shortcuts:** modify models

# mmodel: post definition modification

## 4 Backend — mmodel

```
example_experiment(b, x, y)
returns: z
graph: example_graph
handler: MemHandler
```

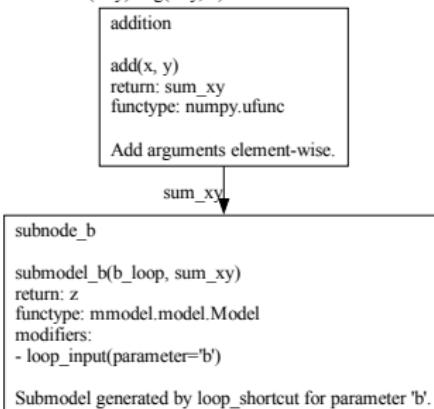
Result of  $(x+y) * \log(x+y, b)$ .



```
looped_model = loop_shortcut(
    example_model,
    "b",
    name="looped_model",
)
looped_model.visualize()
```

```
looped_model(b_loop, x, y)
returns: z
graph: example_graph
handler: MemHandler
```

Result of  $(x+y) * \log(x+y, b)$



# Table of contents

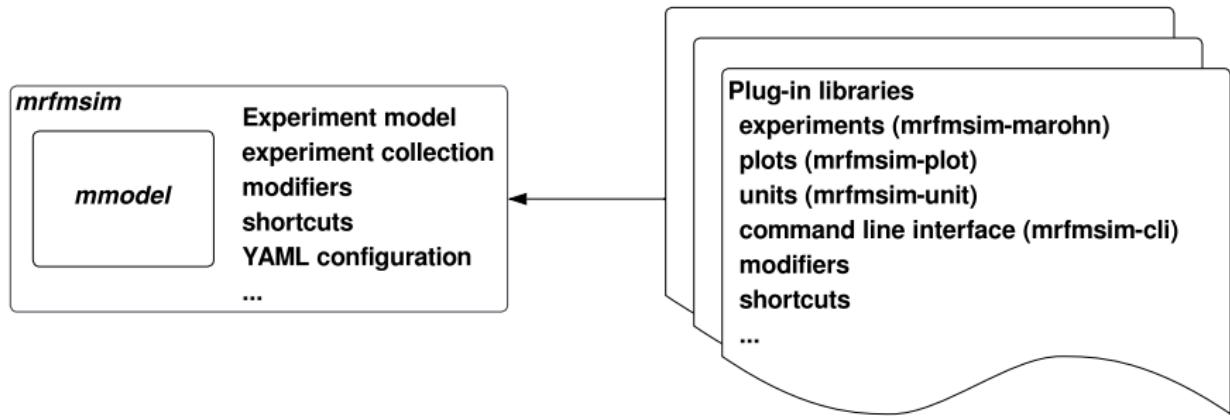
## 5 *mrfmsim*

---

- ▶ Motivation
- ▶ Magnetic Resonance Force Microscopy (MRFM)
- ▶ Challenges
- ▶ Backend — *mmodel*
- ▶ *mrfmsim*
- ▶ Conclusions

# *mrfmsim*: architecture

## 5 *mrfmsim*



Sun, P., *et al.*, in preparation, 2024.

# ***mrfmsim***: plugin system

## 5 *mrfmsim*

---

```
[tool.poetry.plugins."mrfmsim_plugin"]
experiment = """
mrfmsim_marohn.experiment
[CermitARPCollection, CermitSingleSpinCollection,
CermitTDCollection, CermitESRCollection, IBMCylic]
"""
"""


```

- defined using the “entry points” of the setuptools
- separate development of non-core features
- we have added (available on GitHub)
  - Marohn group experiments (*mrfmsim-marohn*)
  - command-line interface (*mrfmsim-cli*)
  - units (*mrfmsim-unit*)
  - plotting (*mrfmsim-plot*)

# mrfmsim: configuration file using YAML

## 5 mrfmsim

```
!Experiment:example_experiment
graph: !Graph:example_graph
    grouped_edges:
        - [addition, logarithm]
        - [[addition, logarithm], multiplication]
node_objects: !nodes
    addition:
        func: !import numpy.add
        arglist: [x, y]
        output: sum_xy
    logarithm:
        func: !import math.log
        arglist: [sum_xy, b]
        output: log_xy
    multiplication:
        func: !import numpy.multiply
        arglist: [sum_xy, log_xy]
        output: z
doc: Result of (x+y)*log(x+y, b).
```

- more readable
- easier to share for collaboration

# Table of contents

## 6 Conclusions

---

- ▶ Motivation
- ▶ Magnetic Resonance Force Microscopy (MRFM)
- ▶ Challenges
- ▶ Backend — *mmodel*
- ▶ *mrfmsim*
- ▶ Conclusions

# Conclusions

## 6 Conclusions

A modular simulation platform for developing new experimental techniques: *mmodel* backend + *mrfmsim*

- ✓ Modular
- ✓ Flexible
- ✓ Readable (graph and rich metadata)
- ✓ Easy to test, run, optimize, and share
- ✓ Can use existing codebase
- ✓ Open source

# Conclusions

## 6 Conclusions

A modular simulation platform for developing new experimental techniques: *mmodel* backend + *mrfmsim*

- ✓ Modular
- ✓ Flexible
- ✓ Readable (graph and rich metadata)
- ✓ Easy to test, run, optimize, and share
- ✓ Can use existing codebase
- ✓ Open source

⇒ up to 20-fold speed up for simulation

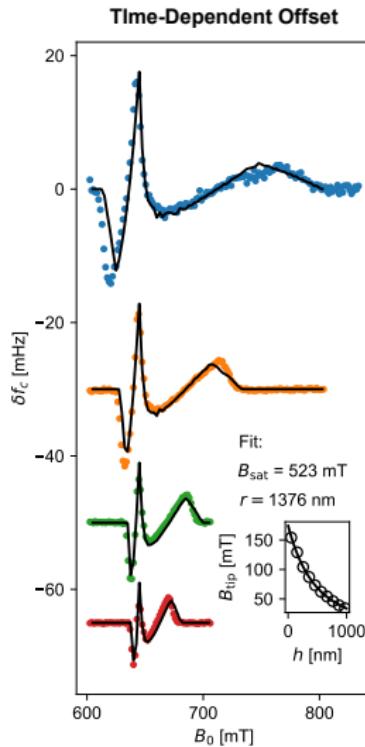
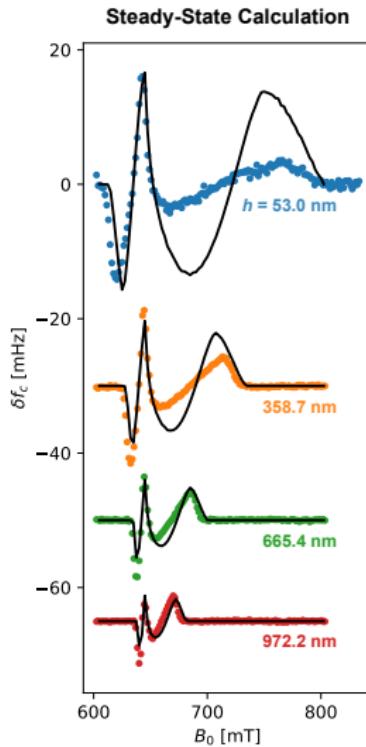
⇒ reduces error and development time

[github.com/marohn-group/mmodel](https://github.com/marohn-group/mmodel)

[github.com/marohn-group/mrfmsim](https://github.com/marohn-group/mrfmsim)

# Developed fast Time-dependent offset simulation

## 6 Conclusions



For large time dependence on the tip magnetic field, spins have insufficient time to saturate!

Time-dependent offset simulation developed using *mrfmsim*  
✓ Matches experimental peaks simultaneously

Boucher, M.C., Sun, P., et al.,  
*Journal of Magnetic Resonance*,  
2023, 354, 107523.

# Future directions

## 6 Conclusions

---

- apply the *mmodel* and the *mrfmsim* design philosophy to other developing techniques
- create conversion kit to convert *mrfmsim* and *mmodel* to other popular workflows for HPC and up-scaling

# Acknowledgements

## 6 Conclusions



### The Marohn Research Group:

George Du Laney, Michael Boucher, Emily Hiralal, Virginia McGhee, Rachael Cohn, **Prof. John Marohn**, Chris Petroff, and Russell Burgett;  
*Not pictured:* Beth Curley, Jacelyn Greenwald, Shanza Riaz, Corinne Isaac, Paméla Nasr, Hoang Nguyen

### Funding and facilities

- National Institutes of Health
- United States Army Research Office
- Cornell NanoScale Science and Technology Facility (NSF)
- Cornell Center for Materials Research (NSF)
- Cornell University



**Contributors:** Corinne Isaac, Michael Boucher, Eric Moore, Sarah Wright, and Lee Harrell