Use Adaptive Fast Function Approximator in Motor-Filament Binding Kinetics

Zihan Zhang (Steven)[1], Adam R Lamson[2], Robert Blackwell[3], Michael J Shelley[2,4]

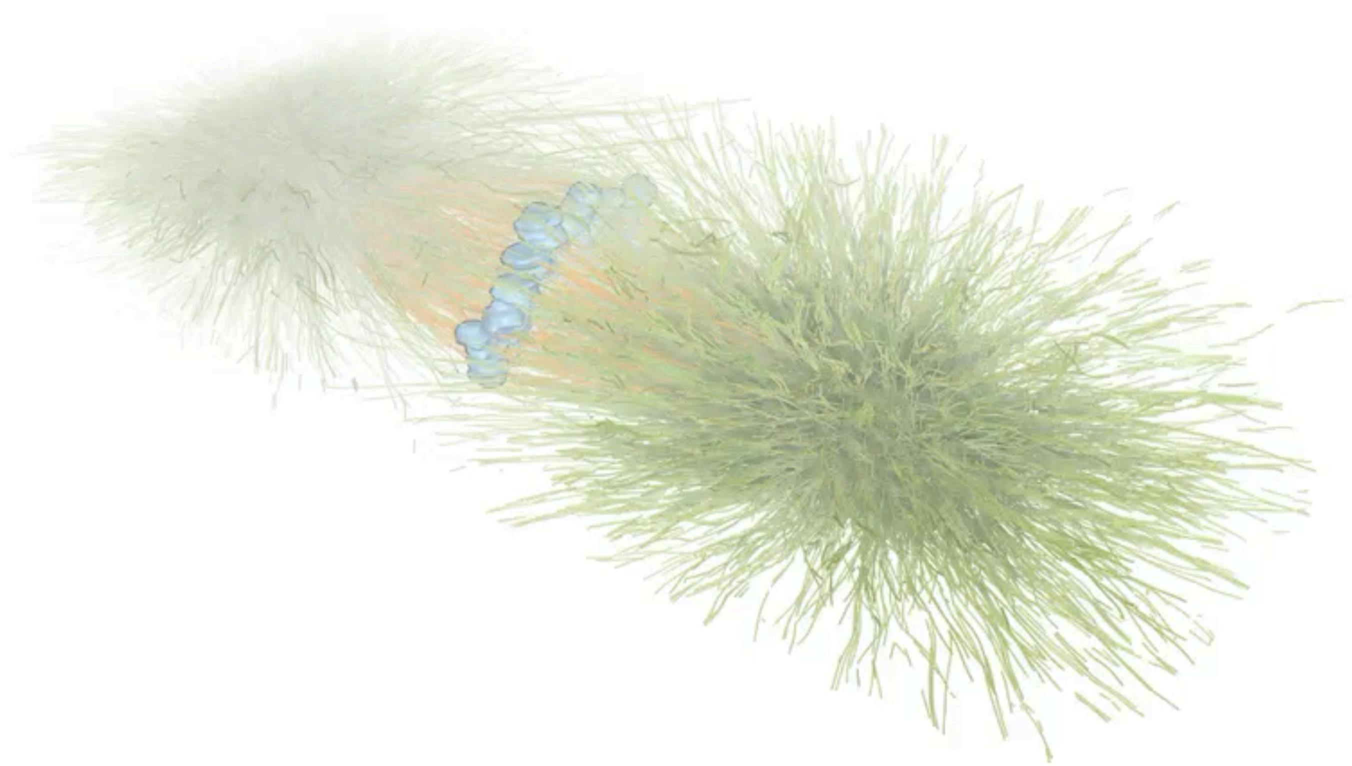
1. Department of Applied Mathematics, University of Washington
2. Center of Computational Biology, Flatiron Institute
3. Scientific Computing Core, Flatiron Institute
4. Courant Institute of Mathematical Sciences, New York University

Sep 26th 2023

CMB symposium

Motivation

- Cytoskeletal filament networks are responsible for cell movement, growth, and division.
- Motor activity is responsible for far-from-equilibrium phenomena, like active stress, self-organized flow, and spontaneous nematic defeat generation.
- Better understanding would allow us to predict how molecular perturbations change cell behavior and to design new complex and adaptive materials.

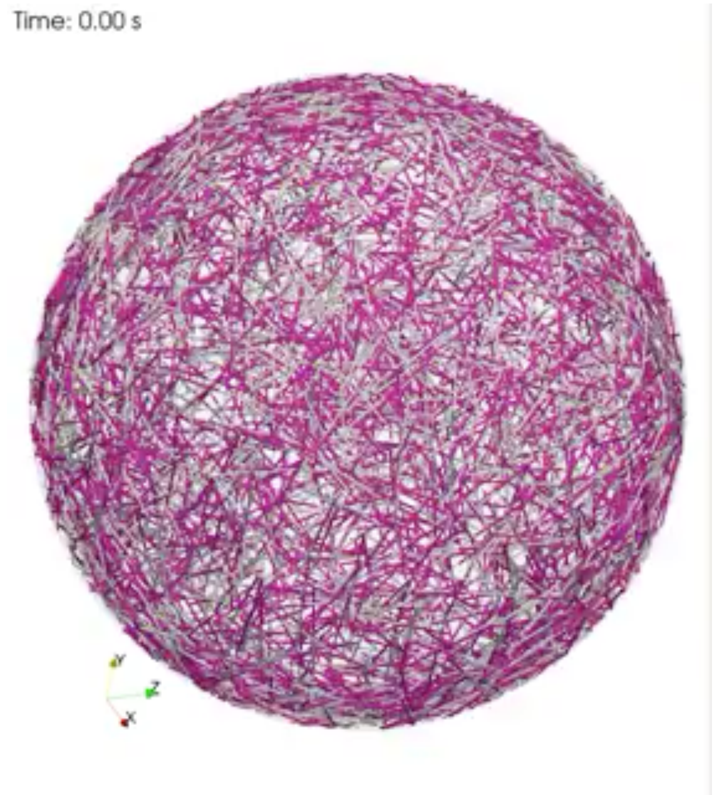


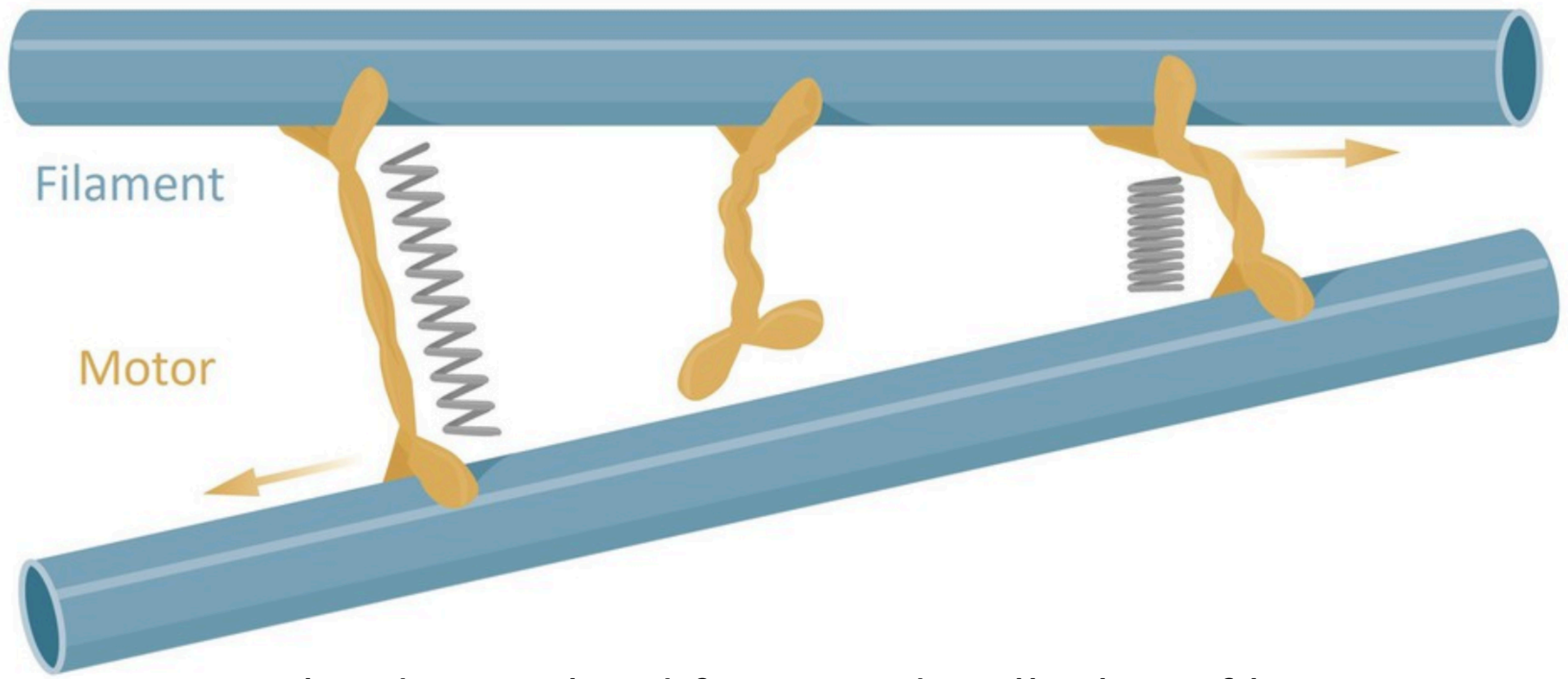
Visualization of the 3D reconstruction of a complete metaphase spindle in the early C. elegans embryo
[Redemann, S. et al. Nature Communications, 2017]

aLENS

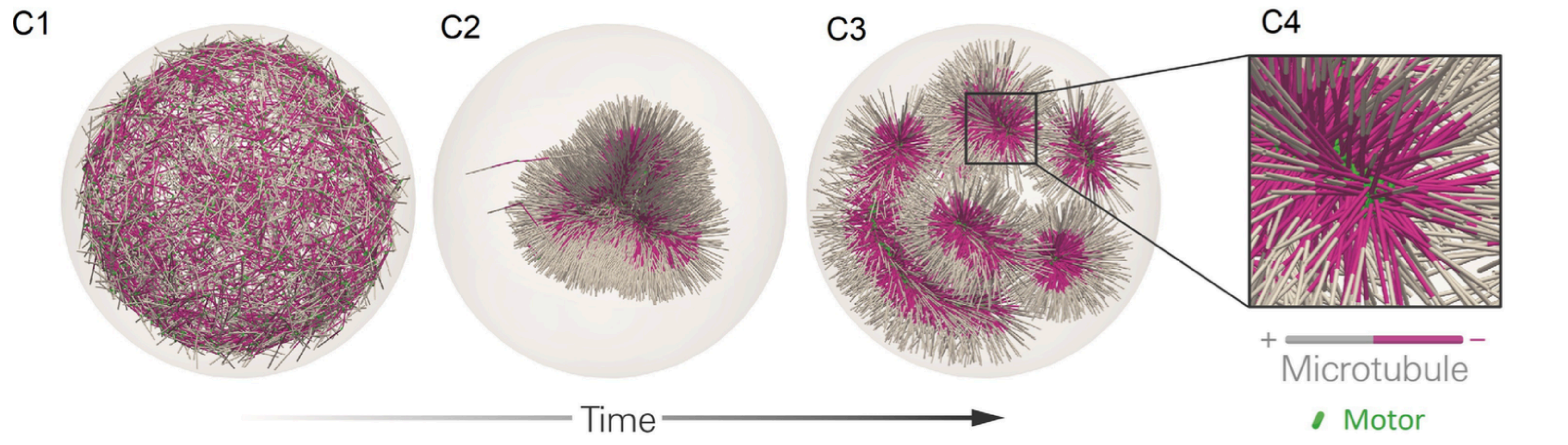
- **aLENS** (a Living ENsemble Simulator): high-performance software for simulating N rigid bodies interconnected by dynamic springs [Yan, W. et al. elife, 2022].
- Biopolymers are polar (microtubule: 25nm, stiff; actin: 7nm, soft).
- Motors are directed (Dynein: - ; Kinesin: + ; Myosin: ?)
- Sequential Pipeline:
  - **Motor diffusion and stepping**.
  - **Computing binding and unbinding** while maintaining realistic macroscopic statistics.
  - **Updating filament position** while overcoming stiffness constraints and maintaining steric exclusion.

Our Focus!





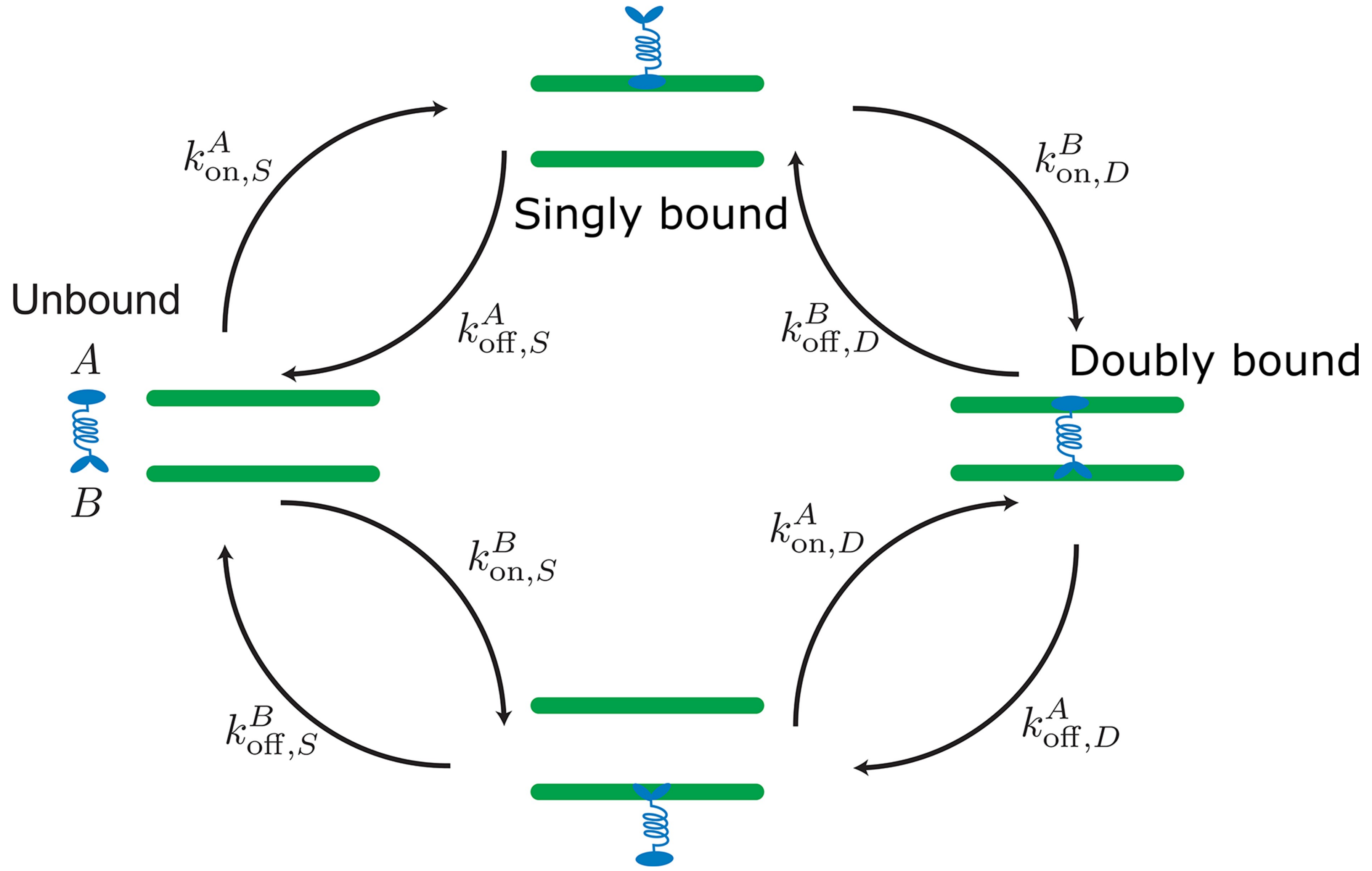
Motors bind to, unbind from, and walk along filaments.



Example simulation of microtubules organized into asters by minus-end-directed motors

Terminology

- **Kinetic Monte-Carlo** (KMC) [Gao, T. et al. Physical Review E, 2015; Lamson, A, et al. Eur. Phys. J. E, 2021.]:
  - Allow fluctuations in bound protein number and binding kinetics that recovers the equilibrium distribution of static crosslinking proteins.
  - Satisfy both local and global detailed balance (4-states transitions, $U \leftrightarrows (S_A, S_B) \leftrightarrows D$).



Motors and crosslinkers may have different rates for separate binding heads.

- **Baobzi** [Github]:
  - N-ary tree structure. Faster than *chebfun*.
  - Leaves represent functions in small sub-boxes of domain with Chebyshev polynomials using Clenshaw algorithm.
  - Adaptive to different languages.
- ❖ Singularity.

```
# simple2d.py
from baobzi import Baobzi
import numpy as np

def py_test_func(x):
    return x[0] * x[1]

center = np.array([0.0, 0.0])
half_length = np.array([1.0, 1.0])
point = np.array([0.25, 0.25])
tol = 1E-8
minimum_leaf_fraction = 0.0 # optional/default
split_multi_eval = 1 # optional/default
max_depth = 50 # optional/default

test = Baobzi(py_test_func, 2, 6, center, half_length, 1E-8, minimum_leaf_fraction, split_multi_eva
test.save('test.baobzi')
print(test(point))
del test

test2 = Baobzi(filename='test.baobzi')
print(test2(point))
del test2
```

Python API

Transition Probabilities

- $(S_A, S_B) \leftrightarrows D$: enforce macroscopic thermodynamic statistics (correct equilibrium bound-unbound concentrations and distributions) and account tether deformation energy:

$$R_{on,D}(s_i, t) = \frac{\epsilon K_e k_{o,D}}{V_{bind}} \sum_j \int_{L_j} e^{-\beta U_{i,j}(s_i,s_j)} ds_j \,.$$

$$R_{off,D}(s_i, s_j, t) = k_{o,D} \,.$$

- Transition probabilities as inhomogeneous Poisson processes:

$$P(\Delta t) = 1 - \exp\left(- \int_0^{\Delta t} R(t) dt \right) = 1 - \exp(-R(0)\Delta t + O(\Delta t^2)) \,.$$

- Searching volume of unbounded head (not considering steric interactions with filaments):

$$V_{bind} = 4\pi \int_0^{r_{c,D}} e^{-\beta U_{i,j}(s_i,s_j)} r^2 dr \,.$$

Finite Lookup Table

- **Reduce Cumulative Distribution Function (CDF) dimensionality** by considering the lab position of each bound motor head and an **infinite carrier line** defined by the position and orientation of unbound filament.

$$CDF'(r_\perp, s) = sgn(s) \int_0^s e^{-\beta U(r_\perp, s')} ds'$$

Normal Lookup

Discretization & 2D Linear Interpolation

$$CDF(r_\perp, s) \approx \left(1 + m - \frac{r_\perp}{\Delta r}\right)\left(1 + n - \frac{s}{\Delta s}\right) CDF_{m,n} + \left(\frac{r_\perp}{\Delta r} - m\right)\left(1 + n - \frac{s}{\Delta s}\right) CDF_{m+1,n}$$
$$+ \left(1 + m - \frac{r_\perp}{\Delta r}\right)\left(\frac{s}{\Delta s} - n\right) CDF_{m,n+1} + \left(\frac{r_\perp}{\Delta r} - m\right)\left(\frac{s}{\Delta s} - n\right) CDF_{m+1,n+1} \,.$$

Reverse Lookup

Interpolation & binary search: $O(\log_2(\delta s_{\max}))$.

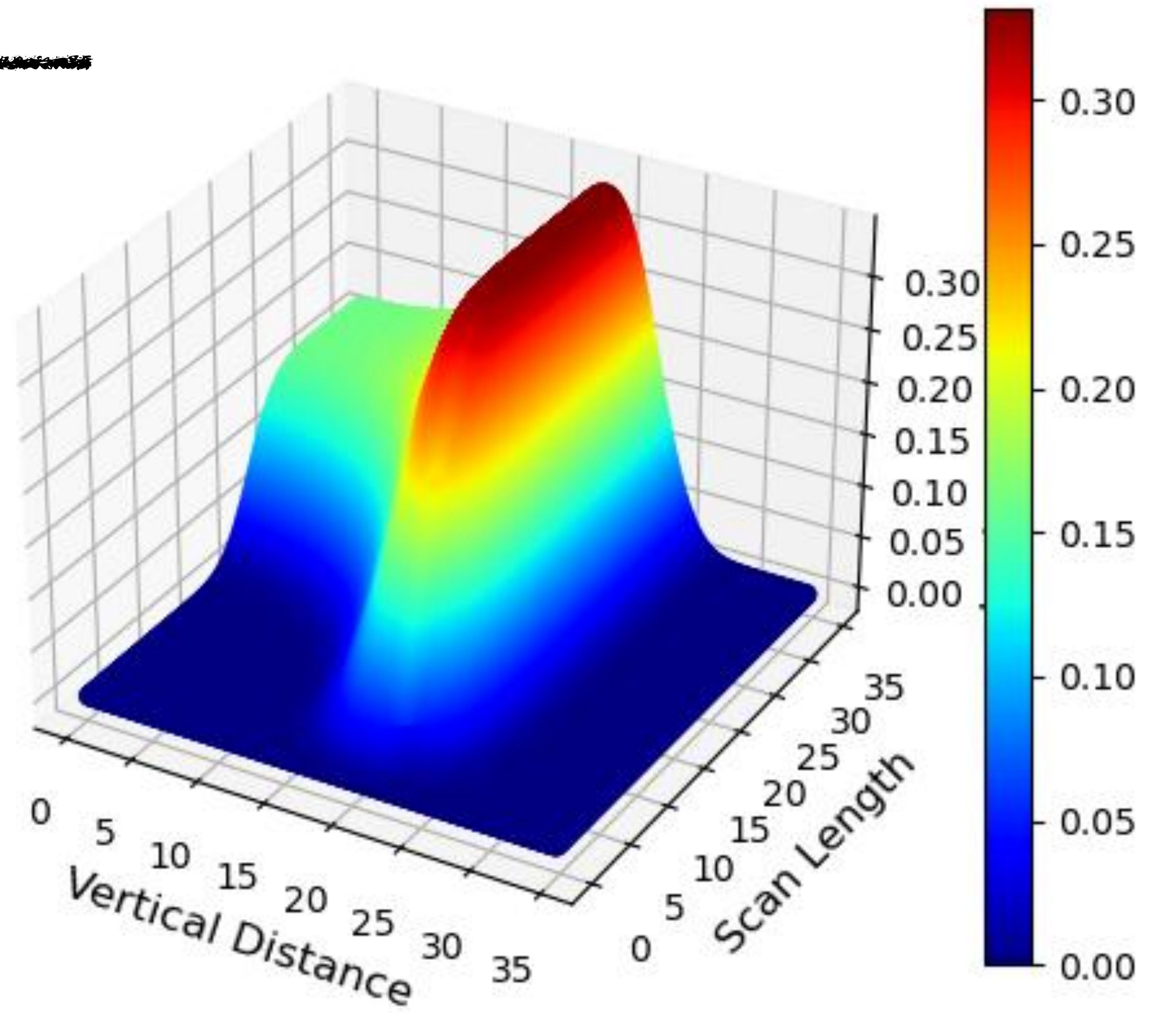
$$s_- = \Delta s \frac{X - CDF_{m,n_-}}{CDF_{m,n_-+1} - CDF_{m,n_-}} + \Delta s n_-,$$
$$s_- = \Delta s \frac{X - CDF_{m+1,n_+}}{CDF_{m+1,n_++1} - CDF_{m+1,n_+}} + \Delta s n_+,$$
$$s \approx (s_+ - s_-) \frac{r_\perp - r_-}{\Delta r} + s_- \,.$$

Baobzi in Normal Lookup

- Baobzi Family (BF): Normalize s and discretize $r_\perp$ dimension using multiple Baobzi objects (with grid size in same [or smaller] order of magnitude). Use binary search in point evaluation.

	Soft [Original]	Medium	Hard	Long + Soft	Long + Medium	Long + Hard
LT Test Acc	3.54855e-07	7.28005e-06	2.74903e-06	9.5964e-07	5.65323e-07	6.06751e-06
BF Test Acc	6.3079e-14	8.7667e-08	2.84853e-08	3.58351e-13	1.98738e-12	1.05103e-07
BF Global Test Acc	8.18472e-11	3.59992e-06	1.02089e-12	2.65102e-11	1.21617e-09	1.93482e-07
LT Build Time (s)	0.0221843	0.0231032	0.0227945	0.0218075	0.0280296	0.0791281
BF Build Time (s)	0.170291	0.0130241	0.406889	0.404616	0.279097	0.508119
Build Time Ratio	7.64412	0.56804	17.74	18.8279	10.0276	6.44766
LT Evaluation Time (s)	0.0147584	0.00514845	0.00214304	0.0219073	0.0123112	0.0093298
BF Evaluation Time (s)	0.131864	0.0455718	0.0573549	0.260138	0.107051	0.134007
Evaluation Time Ratio	8.93483	8.85155	26.7633	11.8745	8.69548	14.3633

Comparison of performances between Lookup Table (LT) and Baobzi Family (BF) under different stiffness and freelength of motor spring.

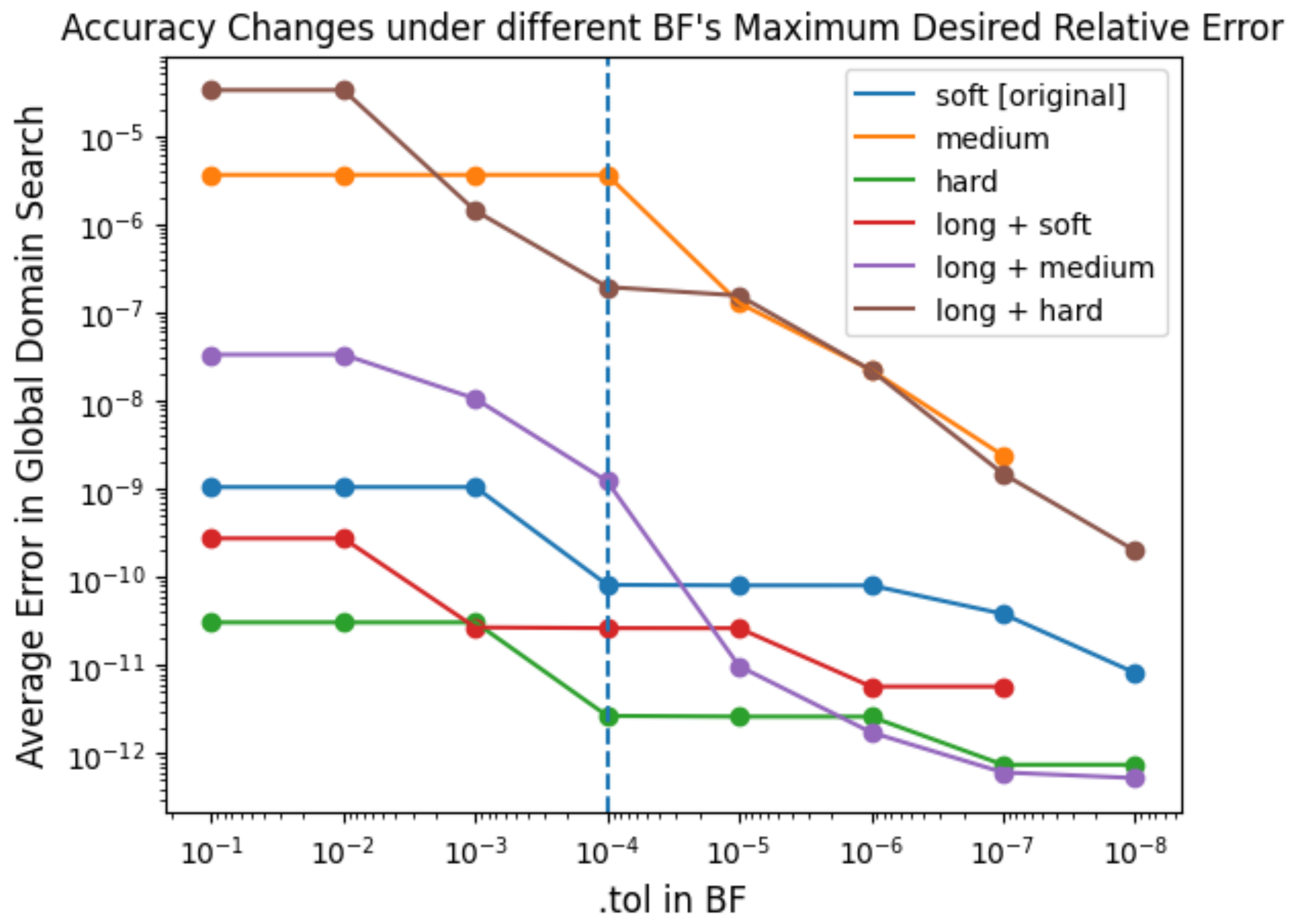


Reconstructed domain within $r_{c,D}$ limit in both dimensions.

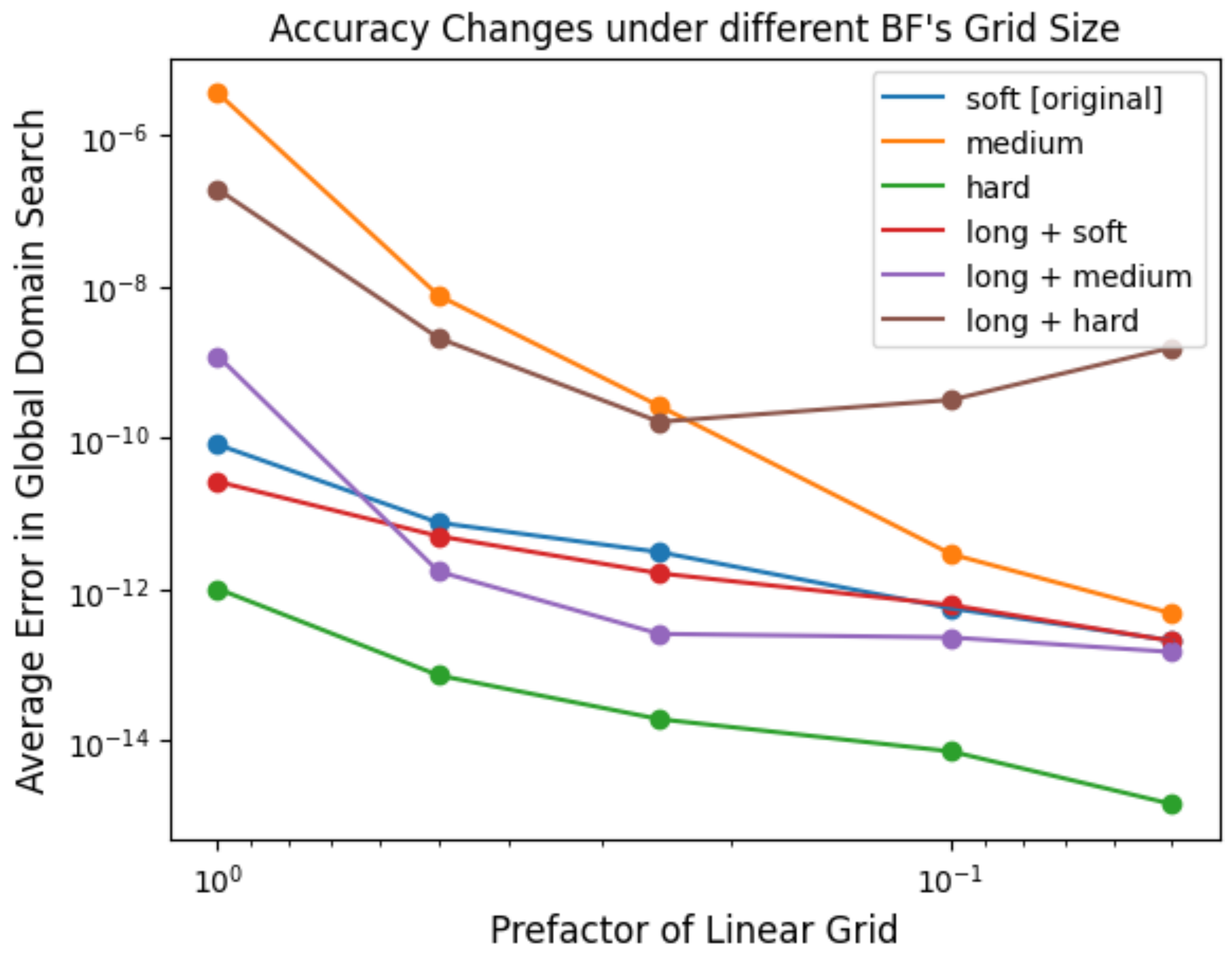
	Soft [Original]	Medium	Hard	Long + Soft	Long + Medium	Long + Hard
BF2 Global Test Acc	1.6132e-06	1.37935e-06	6.03662e-07	1.44775e-06	2.50996e-07	1.79677e-06
Build Time Ratio	0.0381889	0.038223	0.0393668	0.0591468	0.110794	0.0945438

Performance of BF with 2-dimensional normalization.

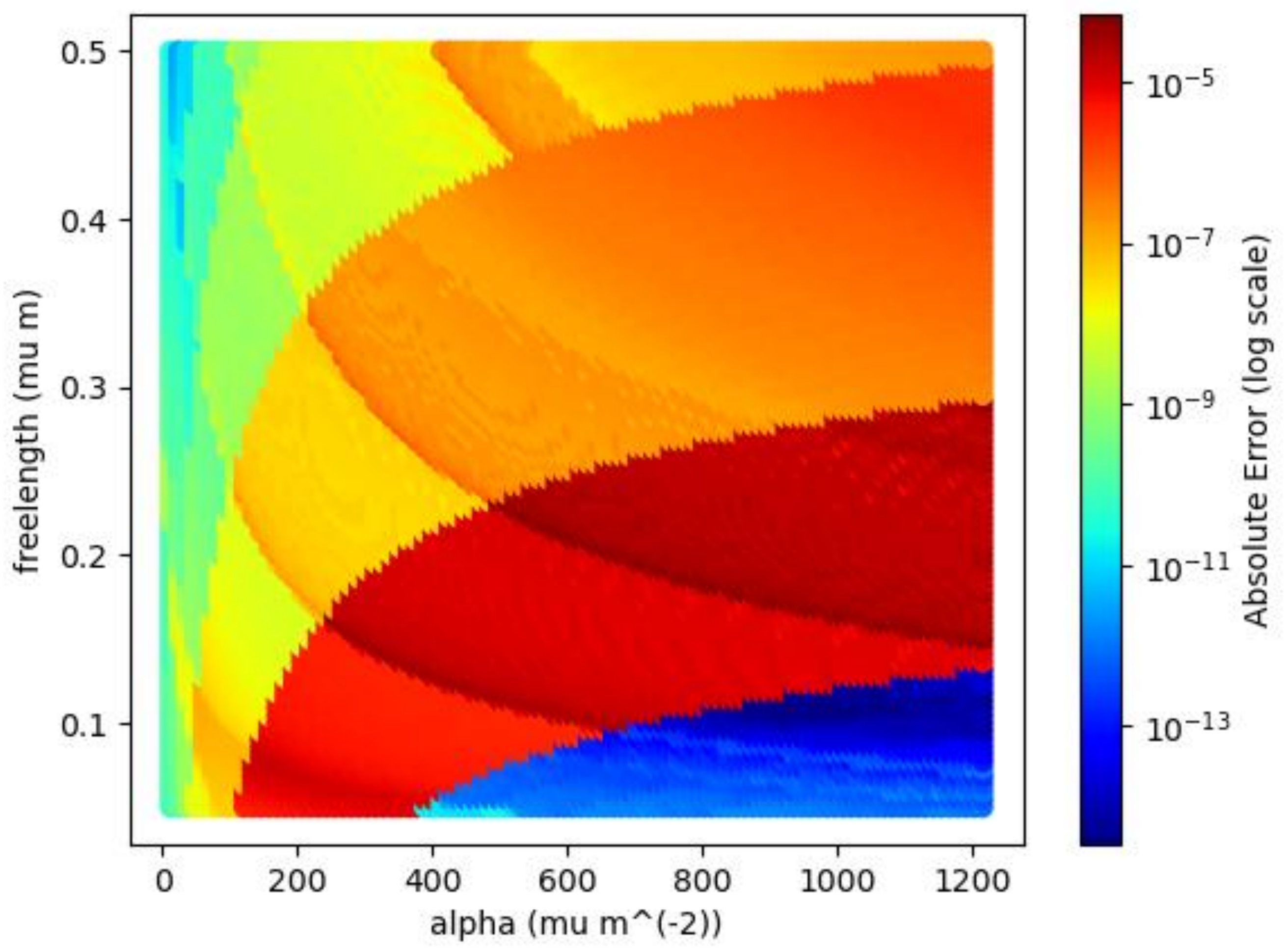
Hyperparameters Tuning



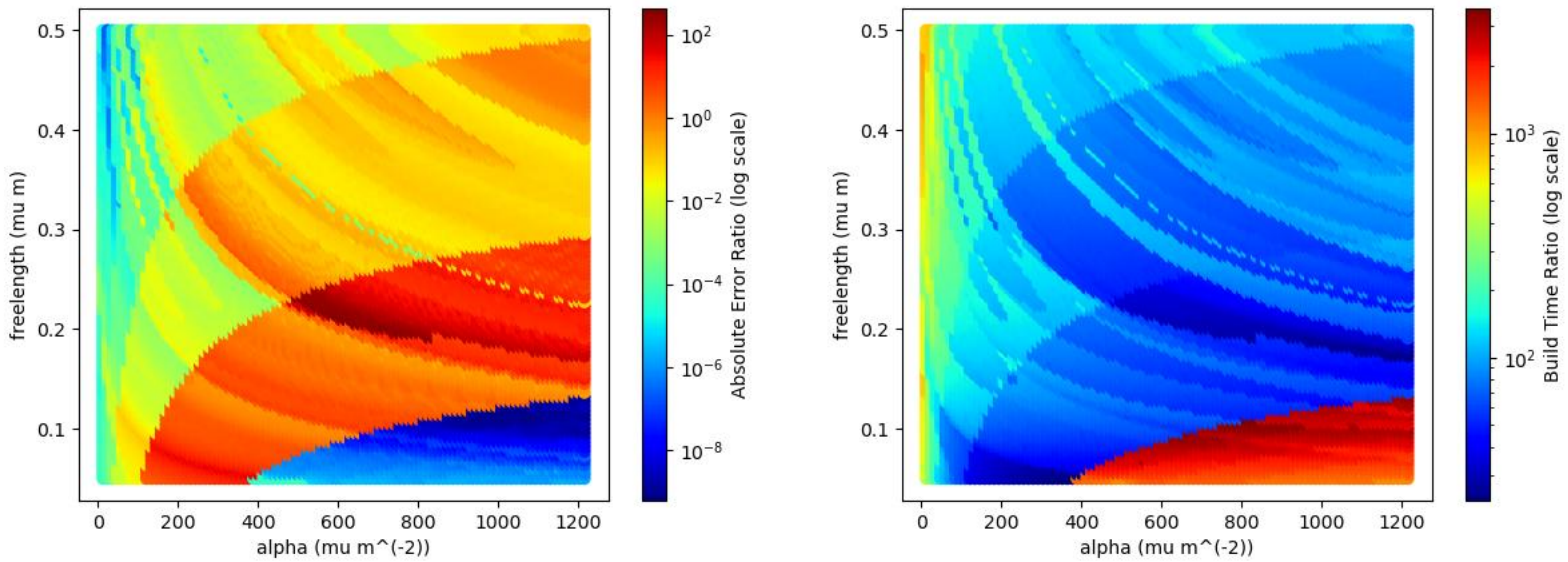
Choose maximum desired relative error (.tol in Baobzi) as 10^{-4} in implementation.



Improve all BF's accuracy to 10^{-10} level by decreasing coefficient α of Baobzi's linear grid.



BF's average error under different stiffness and freelength.



Comparison between BF and BF2 in accuracy and build time.

Baobzi in Reverse Lookup

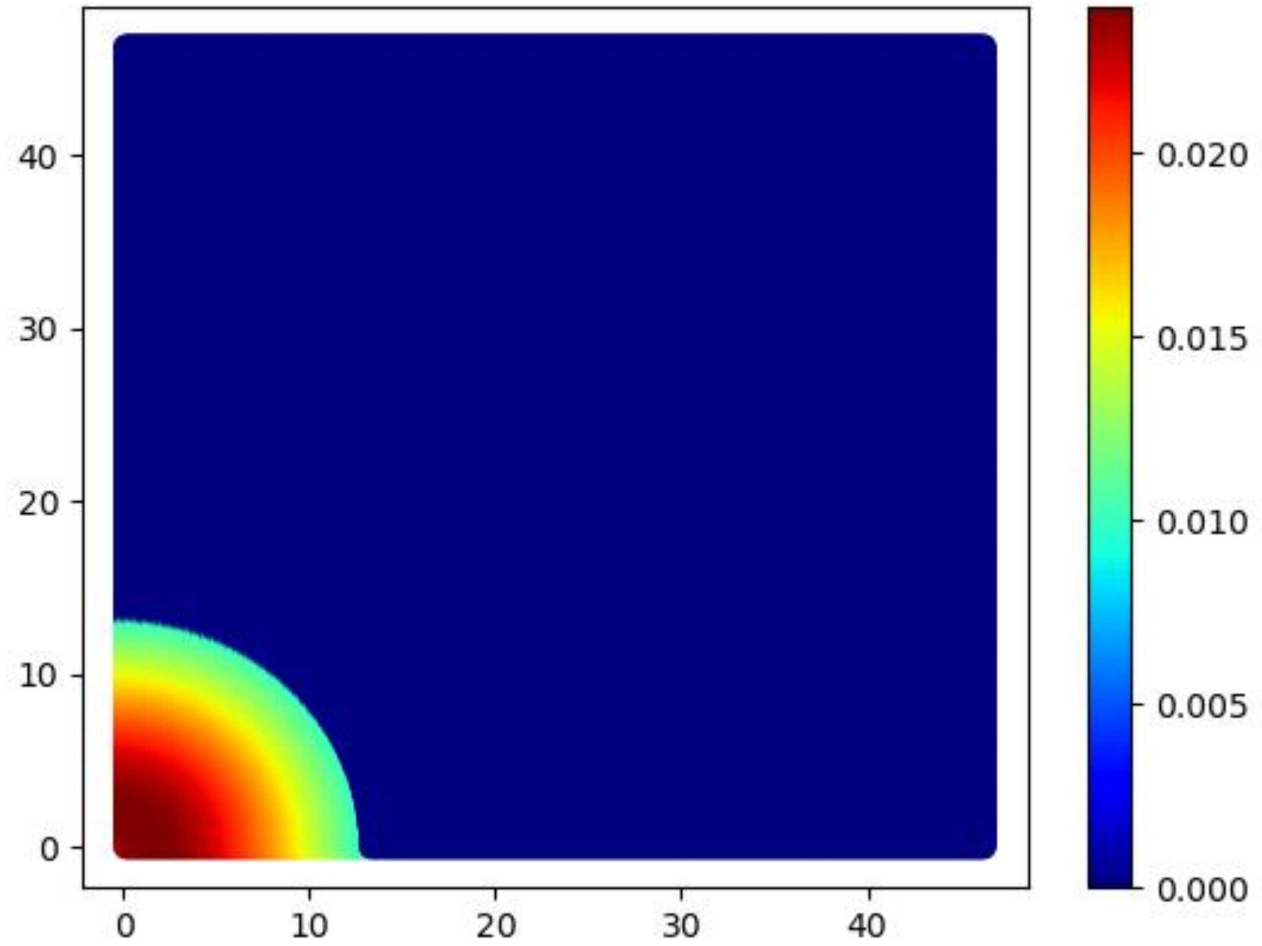
- Precalculate integral range of each Baobzi object and match its grid along $r_{\perp}$.
- Use Boost bisection method to find root without derivatives.
- Use OpenMP to speed up build process.
- Provide interface to save formulated BF object as external files and reloading for flexible time management.

	Soft [Original]	Soft/Medium	Medium	Medium/Hard	Hard
LT Test Acc $[2 \times D]$	2.00431e-05	5.82357e-06	7.34346e-06	3.49505e-06	3.15769e-06
BF Test Acc $[2 \times D]$	4.45106e-05	5.28389e-05	5.7634e-05	4.34567e-05	4.71208e-05
LT Test Acc $[4 \times D]$	4.05045e-05	3.01492e-05	3.96879e-05	1.47139e-04	2.31373e-04
BF Test Acc $[4 \times D]$	3.31202e-06	1.70791e-06	3.26094e-06	2.47522e-04	1.79769e-03
LT Global Test Accuracy	6.77140e-05	2.90116e-05	1.95725e-05	5.70124e-06	2.47043e-06
BF Global Test Accuracy	1.15098e-04	9.12639e-05	3.45134e-05	1.70362e-05	7.59949e-05
Relative Error (Dimensionless)	0.16466%	0.26871%	0.28861%	0.56919%	3.48151%
BF Build Time (s)	24.3361	9.19631	6.10951	3.20413	2.27102
BF Required Space (MB)	195.41	92.7586	68.9679	33.7672	24.6753

BF's reverse lookup result using parameters that optimize build time, not accuracy.

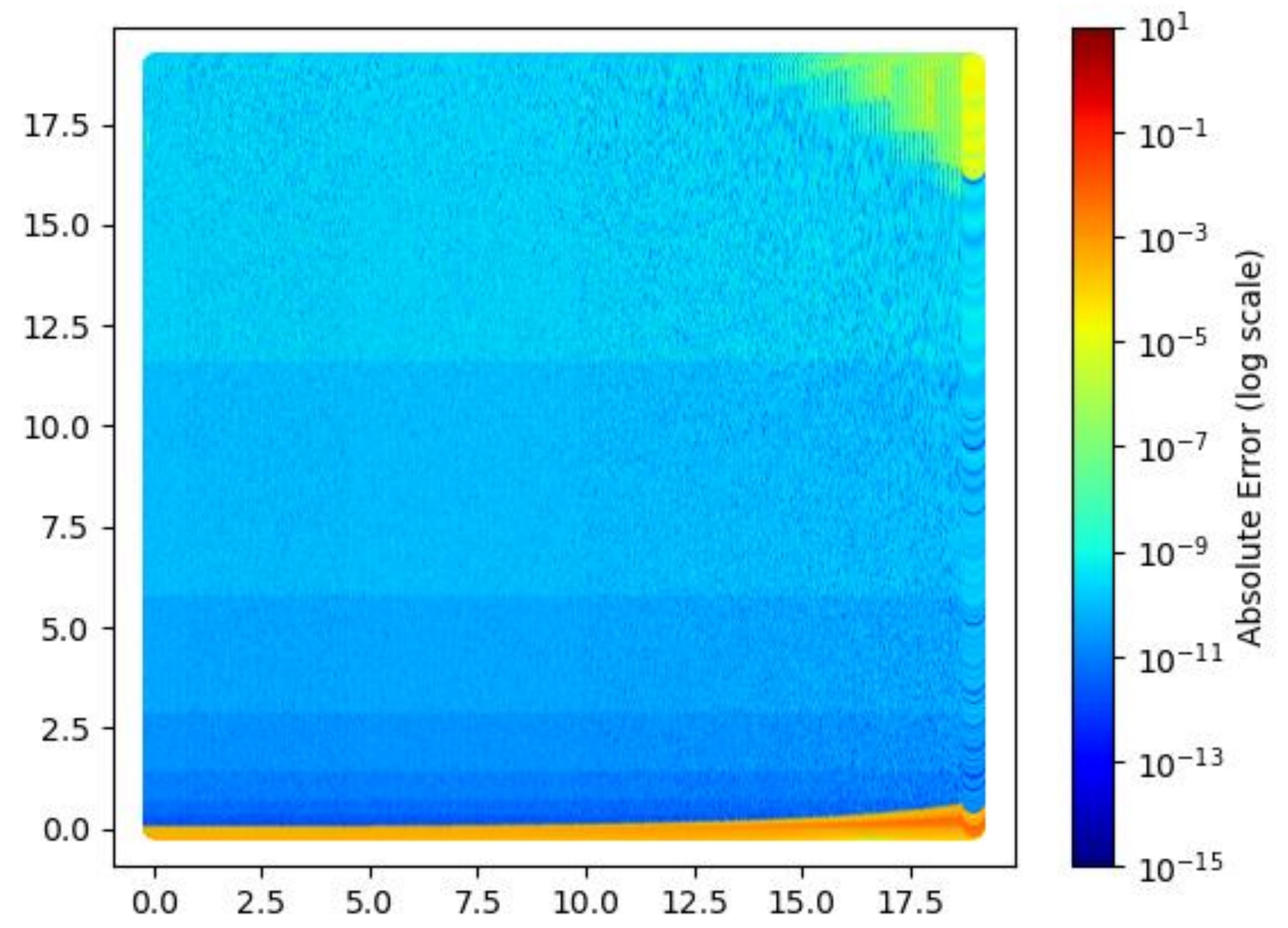
	Soft [Original]	Soft/Medium	Medium	Medium/Hard	Hard
Reconstruction Time (s)	4.29906	2.13036	0.890611	0.590745	0.301753
Required Space (MB)	171	81	60	29	21

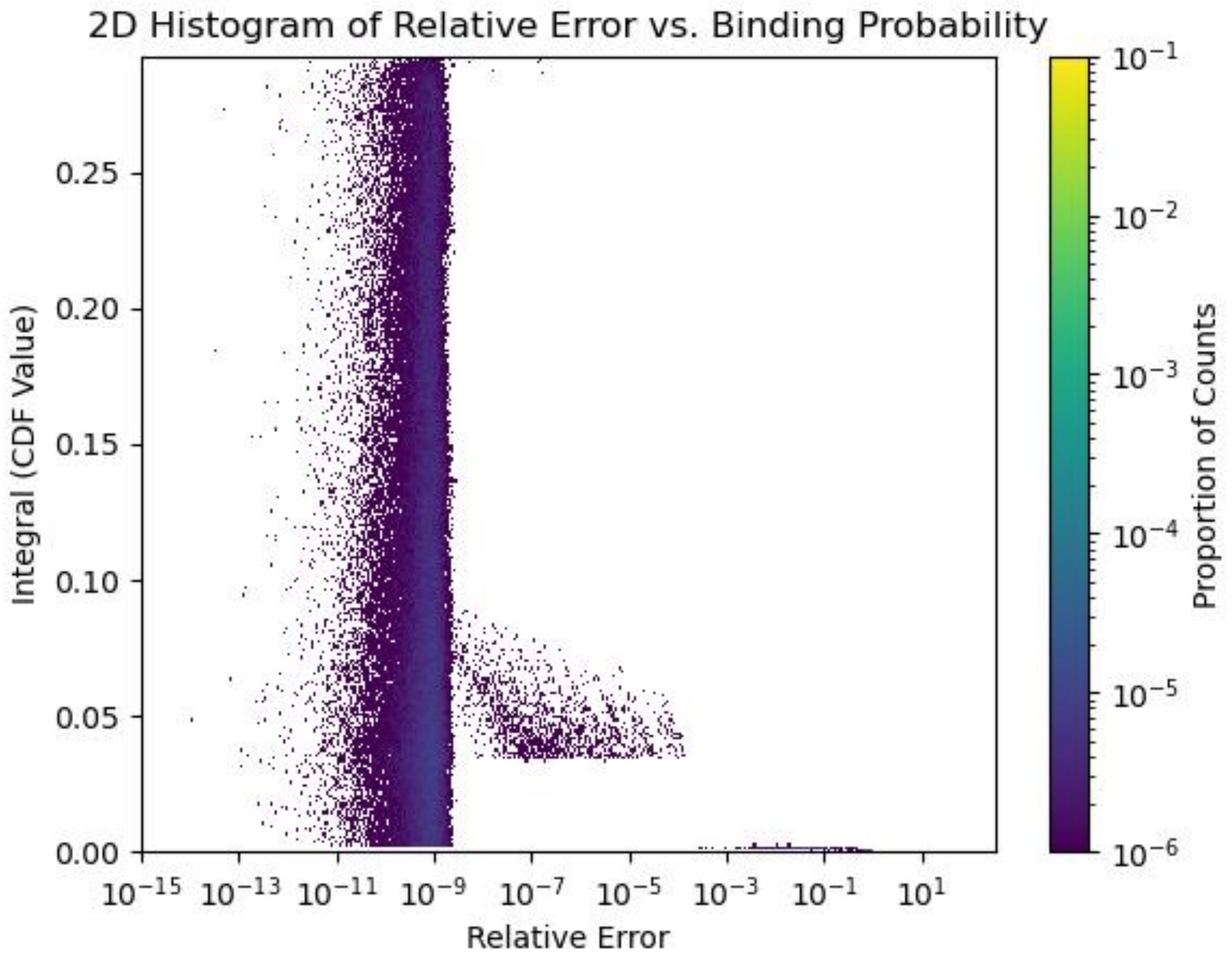
Performance of reconstruction interface.



Effective PDF with threshold filtering.
Use conformal mapping to speed domain search.

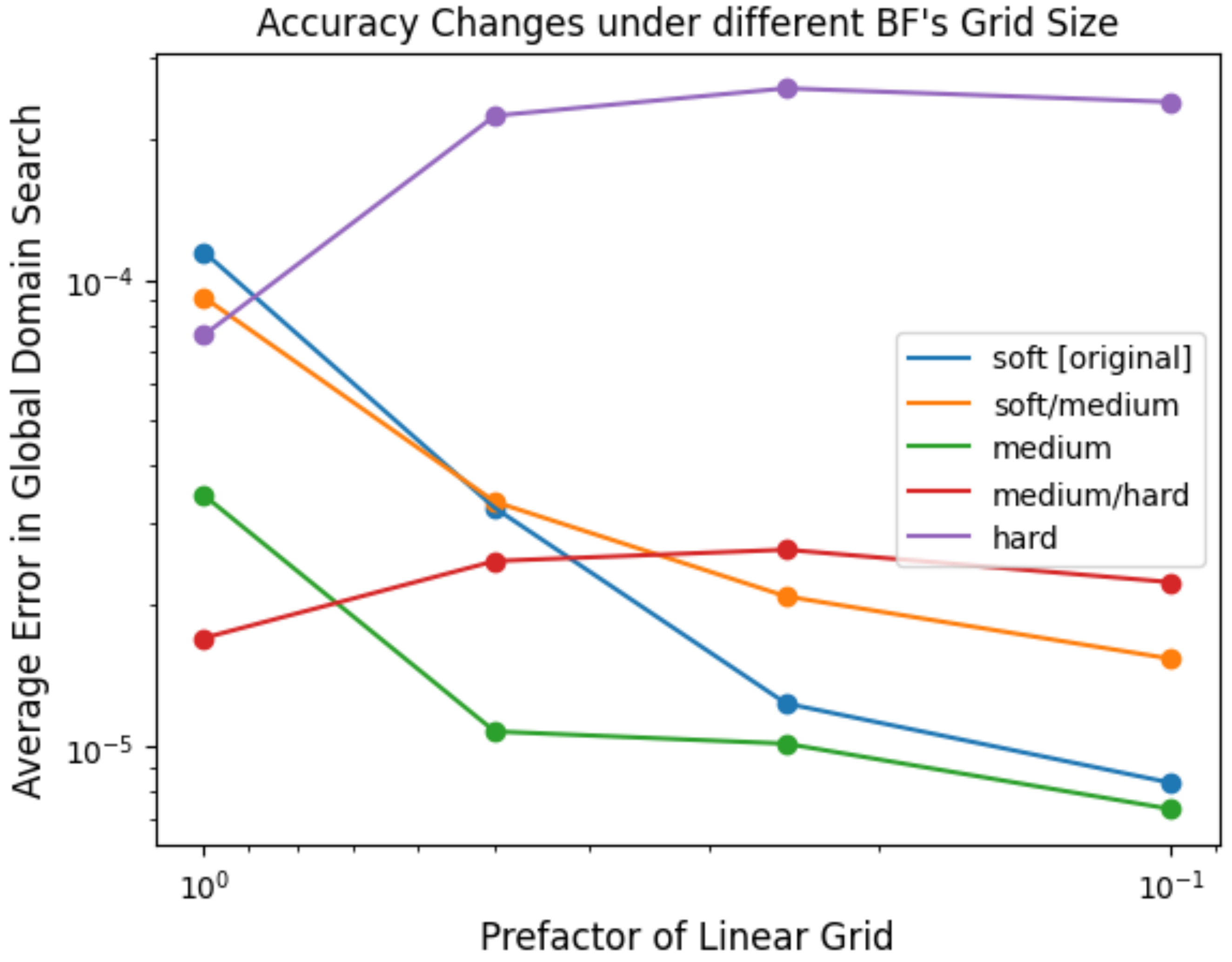
Hyperparameters Tuning

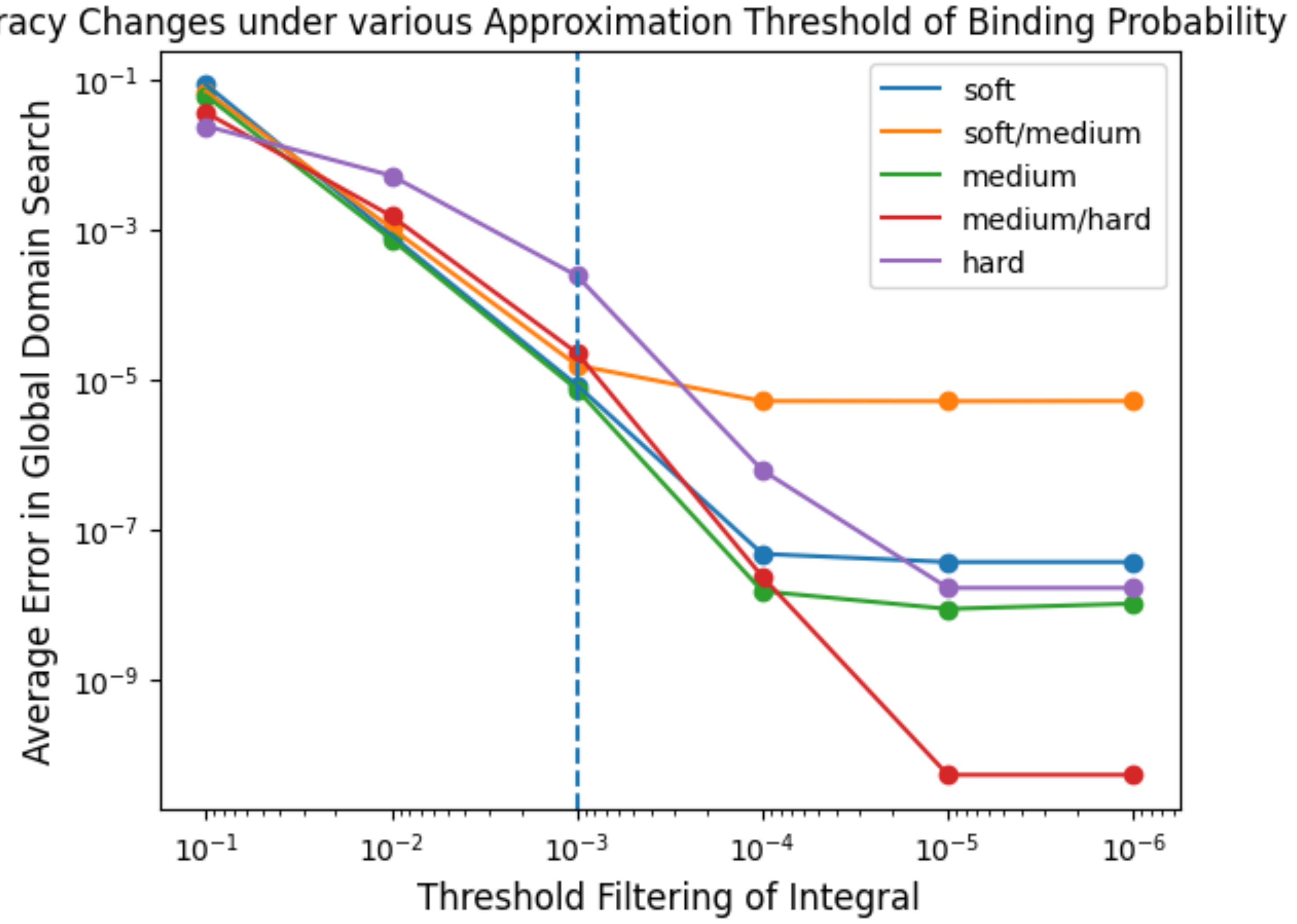




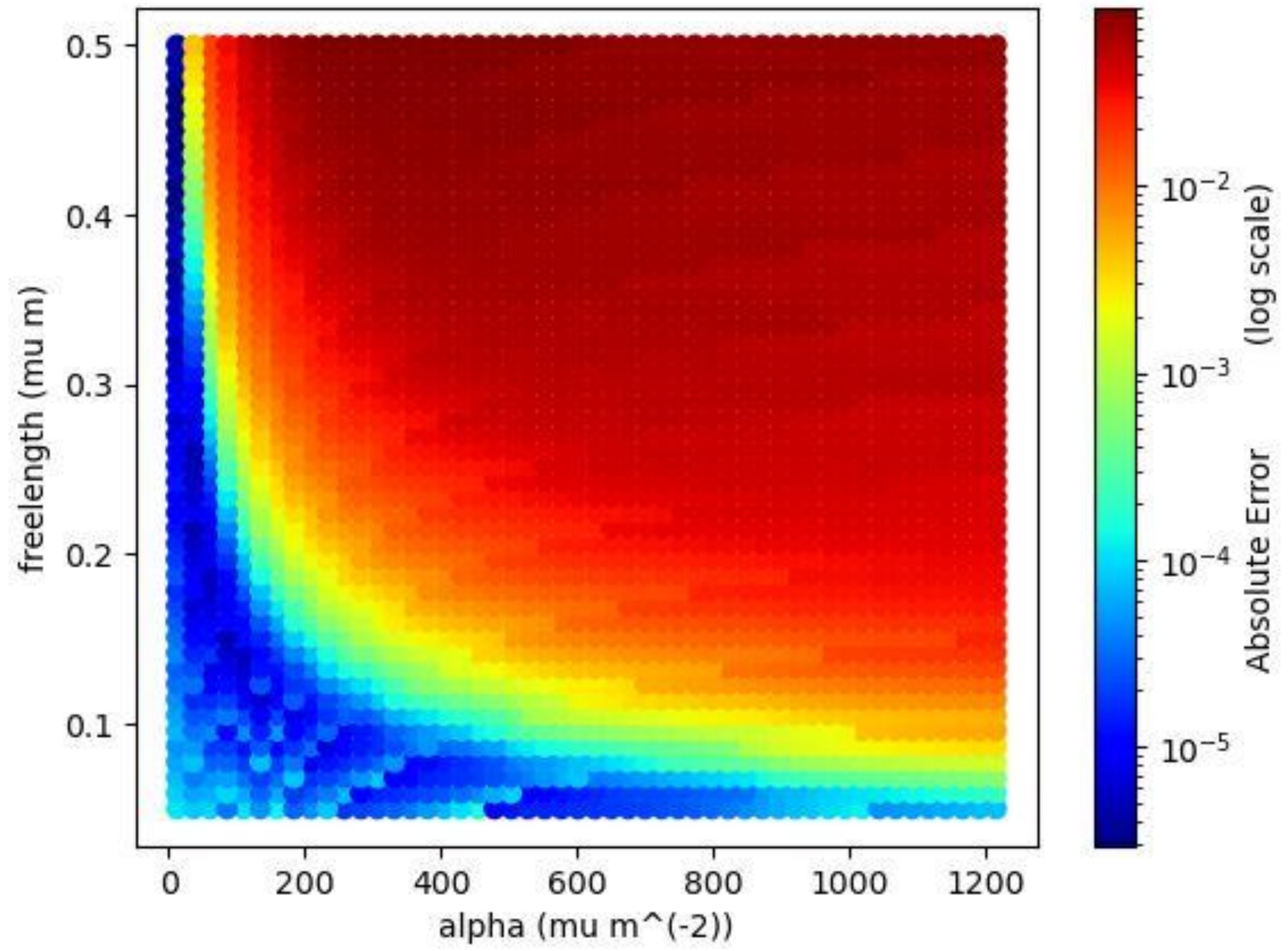
BF’s accuracy maintains 10^{-7} to 10^{-9} level except when s and CDF are small.

2D histogram between relative error and binding probability.





BF’s accuracy is significantly improved using finer grid and lower threshold filtering.



BF’s average error under different stiffness and freelength.

Conclusion & Future Works

- Apply adaptive Chebyshev approximation with parallel computing to simulate motors' binding rate on filaments with better accuracy and affordable costs in both directions searching.
- Together with lookup table, set up benchmarks after fine-tuning of parameters for further KMC tests.
- Provide extensible functionalities including pre-building and loading other formulations of integrand.

❖ Not scalable to more than 3-factors dependence.

❖ Exponential costs growth for better BF's accuracy.

❖ Potential application of rejection sampling or MCMC.

★ aLENS 2.0 is actively developed by Biophysical Modeling Group, CCB.