Fiber Growth Simulation System: Technical Protocol and Performance Benchmarking Guide

11.03.2025

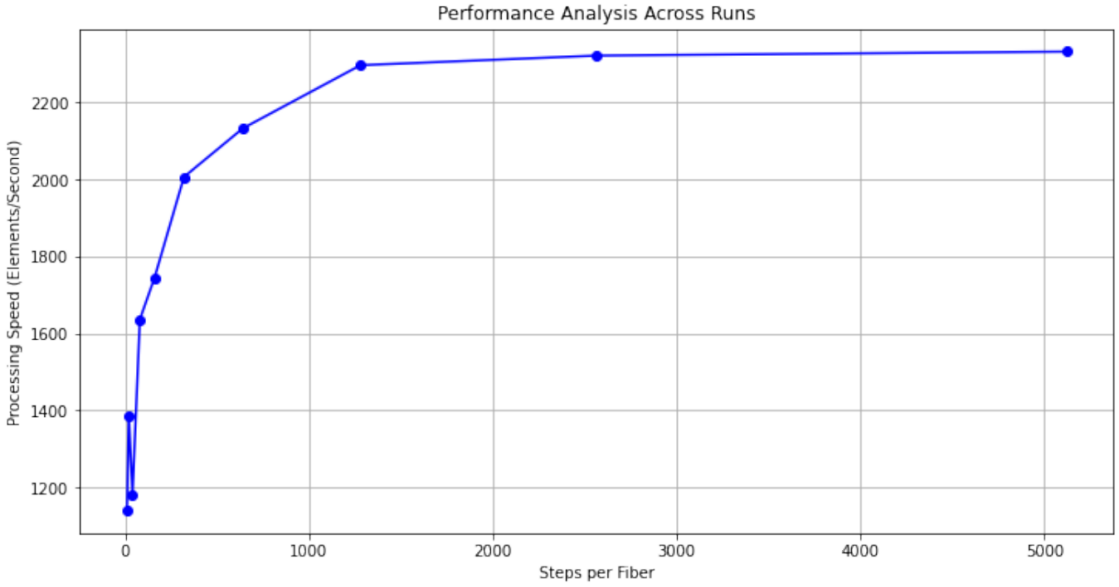
Dr. Yannic Kerkhoff[1,2]

[1]*Freie Universität Berlin, Department of Biology, Chemistry, Pharmacy, CRC1449 "Dynamic Hydrogels on biointerfaces"*

[2]*Zuse Institute Berlin, IT and Data Services Group*

1. Introduction and System Overview

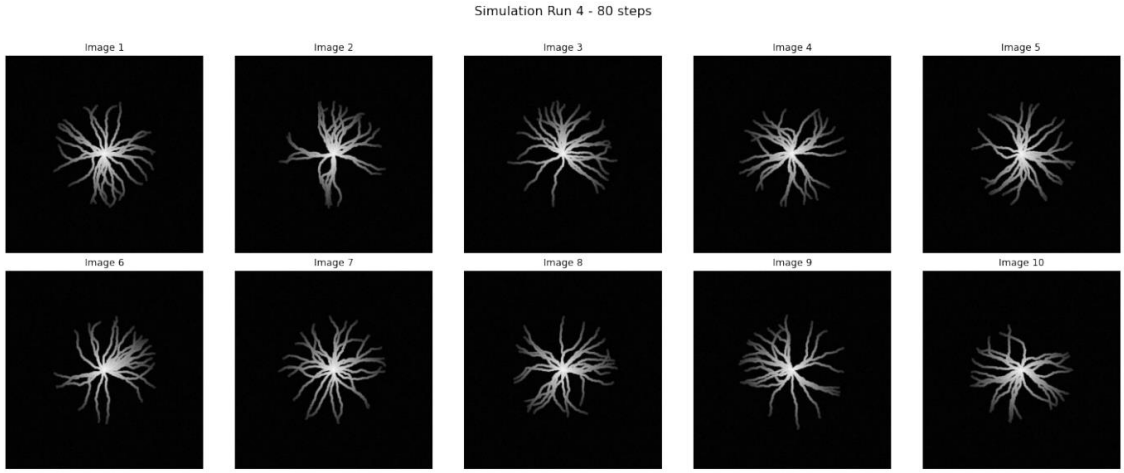
The fiber growth simulation system represents a computational tool designed to simulate complex growth patterns representative for biological and chemical systems. This simulation generates fiber structures that mirror natural growth phenomena, producing images that are similar to those obtained through advanced microscopy techniques. The system mainly serves as a performance benchmark for scientific computing systems as the random elements generated for the simulations can be scaled to test various hardware settings.



2. Core Simulation Mechanics

At the heart of the simulation lies a random walk algorithm that implements a balanced combination of directed and random movement patterns. The system maintains a 70% weighting toward directional persistence while incorporating 30% random movement, creating growth patterns that resemble those observed in natural systems. This balanced approach ensures that the generated structures maintain biological relevance while exhibiting appropriate levels of randomness, needed for scalable computing resources.

The fiber generation process incorporates gradient-based intensity modulation, which creates depth and structural variations in the output images. This feature is particularly important for researchers who need to compare simulation results with actual microscopy data, as it mimics the intensity variations commonly observed in electron microscopy and X-ray microscopy imaging.



3. Operating the System

The tool is provided in this entry as python code (.py) that can be executed e.g. with an IDE like Thonny, as well as an exported jupyter notebook (.ipynb). With default parameters there will be 10 images generated, each containing 25 fibers with 10 elements generated (base steps). The number of iterative runs is set to 10, so in the first run there will be a simulation of 2500 elements in total (10x25x10). The scaling factor is set to 2, so the number of base steps (elements per fiber) is doubled in each run. This leads to a total of 1280000 elements simulated in the tenth run. All these parameters can be tuned to test the performance of the system.

```
159 # Example usage:
160 stats = run_simulation(runs=10, scaling_factor=2, n_images=10, base_steps=10)
```

With the standard parameters, a “typically” equipped research computer with Intel i5 processors and 8GB of RAM will take some minutes to simulate the tenth step with these default settings (see example below).

```
Detailed Statistics per Run:
 Run  Steps  Total Elements  Execution Time  Elements/Second
   1     10            2500        2.193811      1139.569608
   2     20            5000        3.605950      1386.597015
   3     40           10000        8.478987      1179.386173
   4     80           20000       12.226321      1635.815061
   5    160           40000       22.927434      1744.634836
   6    320           80000       39.870718      2006.485036
   7    640          160000       75.017661      2132.831092
   8   1280          320000      139.303071      2297.149652
   9   2560          640000      275.617617      2322.057661
  10   5120         1280000      548.758659      2332.537225

Performance Analysis:
------------------------------------------------------------
Total execution time: 1128.00 seconds
Average elements/second: 1818
Maximum elements/second: 2333
Total elements processed: 2,557,500
```

4. Performance Benchmarking Methodology

4.1 System Requirements and Configurations

The simulation system demonstrates remarkable flexibility in terms of deployment options, ranging from standard laboratory computers to advanced high-performance computing environments. Understanding these options is crucial for both research planning and resource allocation.

For laboratory-based research, standard desktop and laptop configurations can be effectively utilized. Entry-level systems, typically equipped with Intel i5 processors and 8GB of RAM, provide a foundation for basic simulations and method development. Mid-range systems featuring Intel i7 processors with 8 cores and 16GB of RAM offer enhanced performance suitable for most research applications. For intensive research requirements, high-end systems with Intel i9 processors and 32GB or more of RAM enable complex simulations with larger datasets.

In high-throughput computing environments, the simulation can be scaled across various cluster configurations. Basic cluster setups, utilizing 4-8 CPU cores per node with 4GB RAM per core, provide an entry point into distributed computing applications. Advanced clusters, featuring 16-32 CPU cores per node and 8GB RAM per core, enable more sophisticated analyses. Enterprise-level clusters, with 64 or more CPU cores per node and 16GB+ RAM per core, support the most demanding research requirements.

4.2 Benchmarking Protocol and Implementation

The benchmarking process requires a systematic and well-documented approach to ensure reproducible results across different computing environments. Researchers should begin with a series of standardized tests that progressively increase in complexity to establish baseline performance metrics.

Initial testing should commence with minimal complexity configurations, utilizing small datasets to establish baseline performance metrics. These foundational tests help researchers understand the basic system response and identify any immediate constraints. As familiarity with the system grows, researchers can progress to medium-complexity simulations, which provide more realistic performance indicators for typical research applications. High-complexity runs serve to stress-test the system and identify performance boundaries.