The Art of Scientific computation - A comparison of Python and Julia in the context of the Abelian Sandpile Model

# Summary

# Introduction

## Background

### Sandpile Models

* Criticality
* Self-organized criticality
* Abelian sandpile model
* Generalised toppling processes
* Applications ### Computational simulation
* Importance
* Challenges
* Programming languages
  + Python
  + Julia ## Aims
* Impliment the Abelian Sandpile Model in Python and Julia
* Compare the performance of the two languages
  + Speed
  + Memory usage
  + Scalability
  + Numerical stability

# Methodology

## Simulation Algorithms

* Naive algorithm
* Parallel algorithm
* Optimised algorithm ## Implementation Details ### Python Implementation
* naive
* parallel
* optimised
* total ### Julia Implementation
* naive
* parallel
* optimised
* total

## Computational Environment

* Hardware
* Software
* Libraries ## Performance Metrics
* Execution time
* Memory consumption
* Scalability
* Numerical stability # Results

## Performance Analysis

## Properties of the Sandpile

## Visualization of Simulations

## Comparison of Results

# Discussion

## Abelian model

## Language comparison

### Advantages and Limitations

## Implications for Computational Simulations

Reflect on what your findings mean for future computational work in similar fields.

# Conclusion

## Summary of Findings

## Recommendations

## Future Work

# References

# Appendices