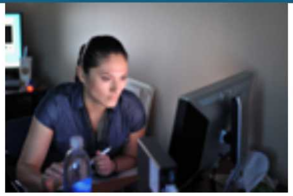
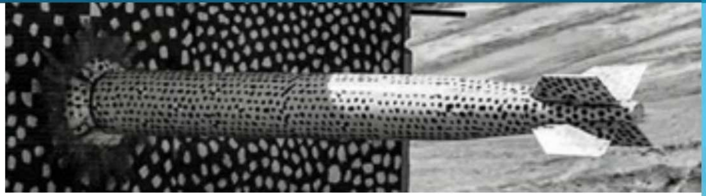
SAND2020-9328C

# Parameter Sensitivity Analysis of the SparTen High Performance Sparse Tensor Decomposition Software





Jeremy M. Myers[1,2]
Daniel M. Dunlavy[2]
Keita Teranishi[2]
D.S. Hollman[2]

[1]College of William and Mary, Williamsburg, VA
[2]Sandia National Laboratories, Albuquerque, NM & Livermore, CA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

IEEE HPEC 2020 - HIGH PERFORMANCE DATA ANALYSIS SESSION

# Summary of Work

**Problem:** Analyzing large, multiway count data

- Examples: Computer traffic data, term-document analysis, email analysis, link prediction, webpage analysis

**Solution:** Canonical Polyadic (CP) Tensor Decomposition

**Implementation:** SparTen: C++, Kokkos-based software for CP decomposition on sparse count data

**Challenge:** How to set all of the parameters in SparTen to …

- Converge to a solution
- Converge to a solution as fast as possible

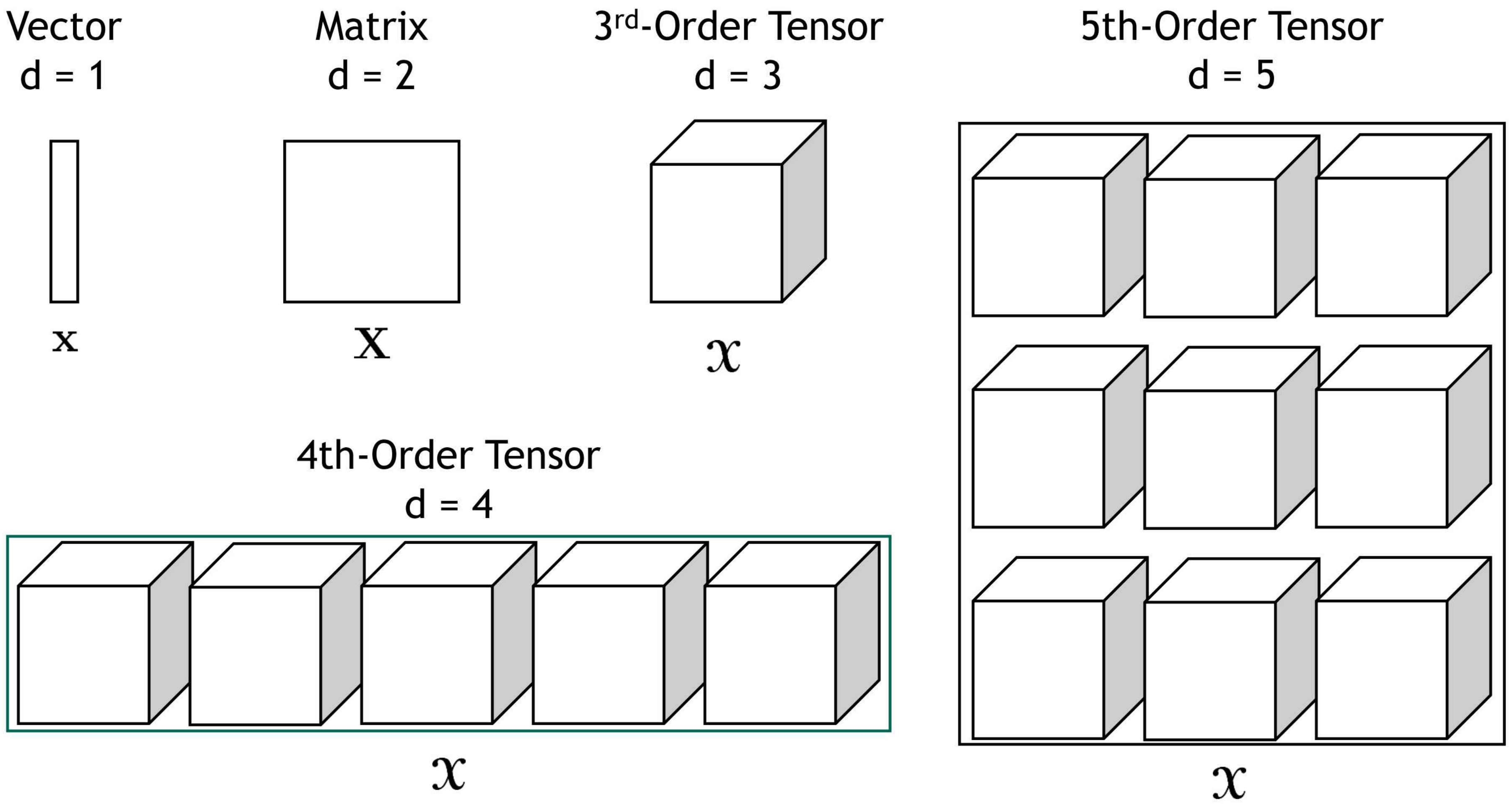
**Experiments:** 3 CPU architectures, 2 algorithms, 3 benchmark tensor data sets

**Conclusions:**

- Some parameters can accelerate software convergence, others can slow or even prevent convergence
- Other parameters have no effect on convergence

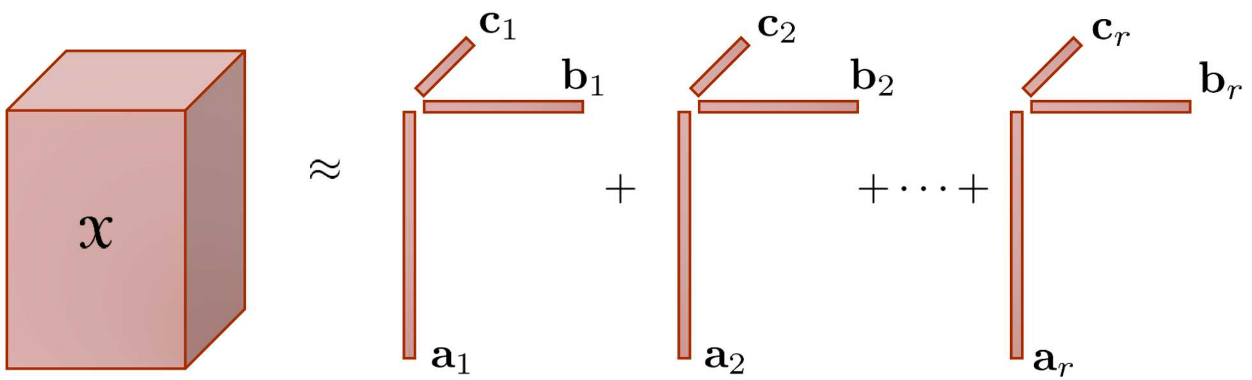
# Background
## CP Tensor Decompositions for Count Data

# A Tensor is an d-Way Array



# Solving the Poisson Regression Problem

**CP Model**
Sum of $d$-way
outer products




$$\min_{\mathcal{M}} \sum_i m_i - x_i \log m_i \text{ subject to } \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

Alternating Poisson Regression

- Assume $(d-1)$ factor matrices are known and solve for the remaining one
- Multiplicative updates like Lee & Seung (2000) for NMF, but improved
- Typically assume data tensor $\boldsymbol{\mathcal{X}}$ is sparse and have special methods for this
- Newton or Quasi-Newton method

Chi & Kolda 2012; Hansen, Plantenga, & Kolda 2015

# 16,000 Experiments (5 months)

## Algorithms (best in class)

- PDNR – 2nd order Damped Newton Optimization
- PQNR – 1st order Quasi-Newton Optimization

## Data Sets (standard benchmarks – FROSTT.io*)

| Data set | Dimensions | Size | Density |
|---|---|---|---|
| Chicago Crime Comm | 4 | 6K x 20 x 70 x 30 | 1e-02 |
| LBNL-Network | 5 | 1.6K x 4K x 1.6K x 4K x 870K | 4e-14 |
| NELL-2 | 3 | 12K x 9K x 28K | 2e-05 |

## Architectures (performance-portability)

- Intel x86 (SandyBridge/Broadwell)
- IBM Power 9
- ARM Thunder X2

* S. Smith, J. W. Choi, J. Li, R. Vuduc, J. Park, X. Liu, and G. Karypis, "FROSTT: The formidable repository of open sparse tensors and tools," Available online, 2017. [Online]. Available: http://frostt.io/

# SparTen Parameter Sensitivity Analysis

# Parameters

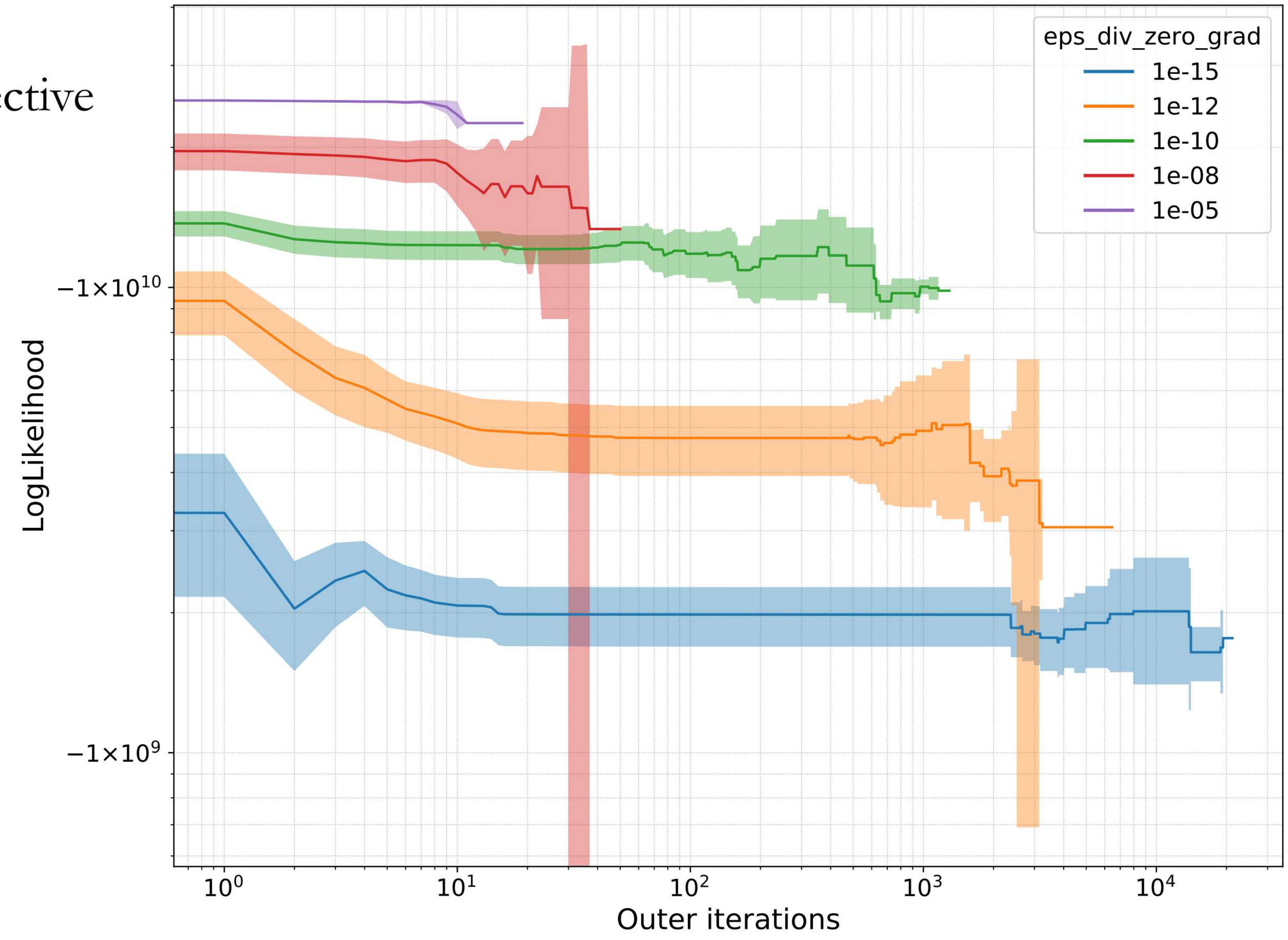
| Description | Default Value | No. of Values Tested* | Description | Default Value | No. of Values Tested* |
|---|---|---|---|---|---|
| Max. no. of outer iterations | 100000 | 10 | Tolerance defining active (nonzero) variables | 1e-03 (PDNR)<br>1e-08 (PQNR) | 4 |
| Max. no. of inner iterations | 20 | 4 | Initial value of damping parameter | 1e-05 | 3 |
| Max. no. of backtracking steps in line search | 10 | 7 | Scalar to increase damping parameter in next iterate | 7/2 | 5 |
| Tolerance for nonzero line search step length | 1e-07 | 4 | Scalar to decrease damping parameter in next iterate | 2/7 | 6 |
| Factor to reduce line search step size | 0.5 | 5 | Tolerance to increase damping parameter | 0.25 | 3 |
| Tolerance to ensure decreases in the objective function | 1e-04 | 4 | Tolerance to decrease damping parameter | 0.75 | 3 |
| Guard against divide-by-zero computing gradient & Hessian | 1e-10 | 5 | No. of limited-BFGS update pairs | 3 | 8 |
| Tolerance to avoid computing log(0) | 1e-16 | 6 | | | |

Expected behavior   Unexpected behavior   No Effect

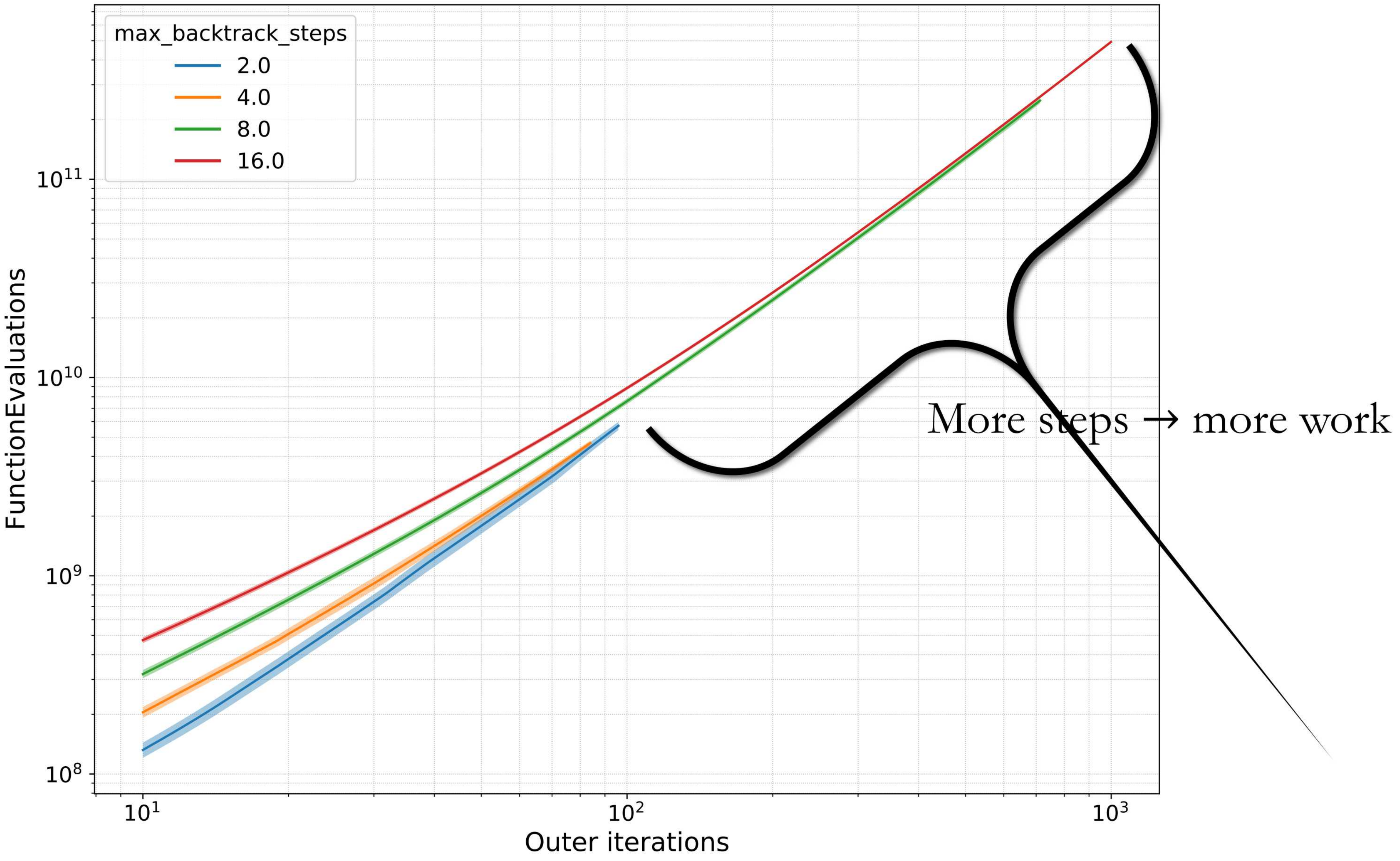
* Some parameters required more test values to determine general trends

# Example: Guard against divide-by-zero computing gradient & Hessian

Small step size minimizes objective function and accelerates convergence

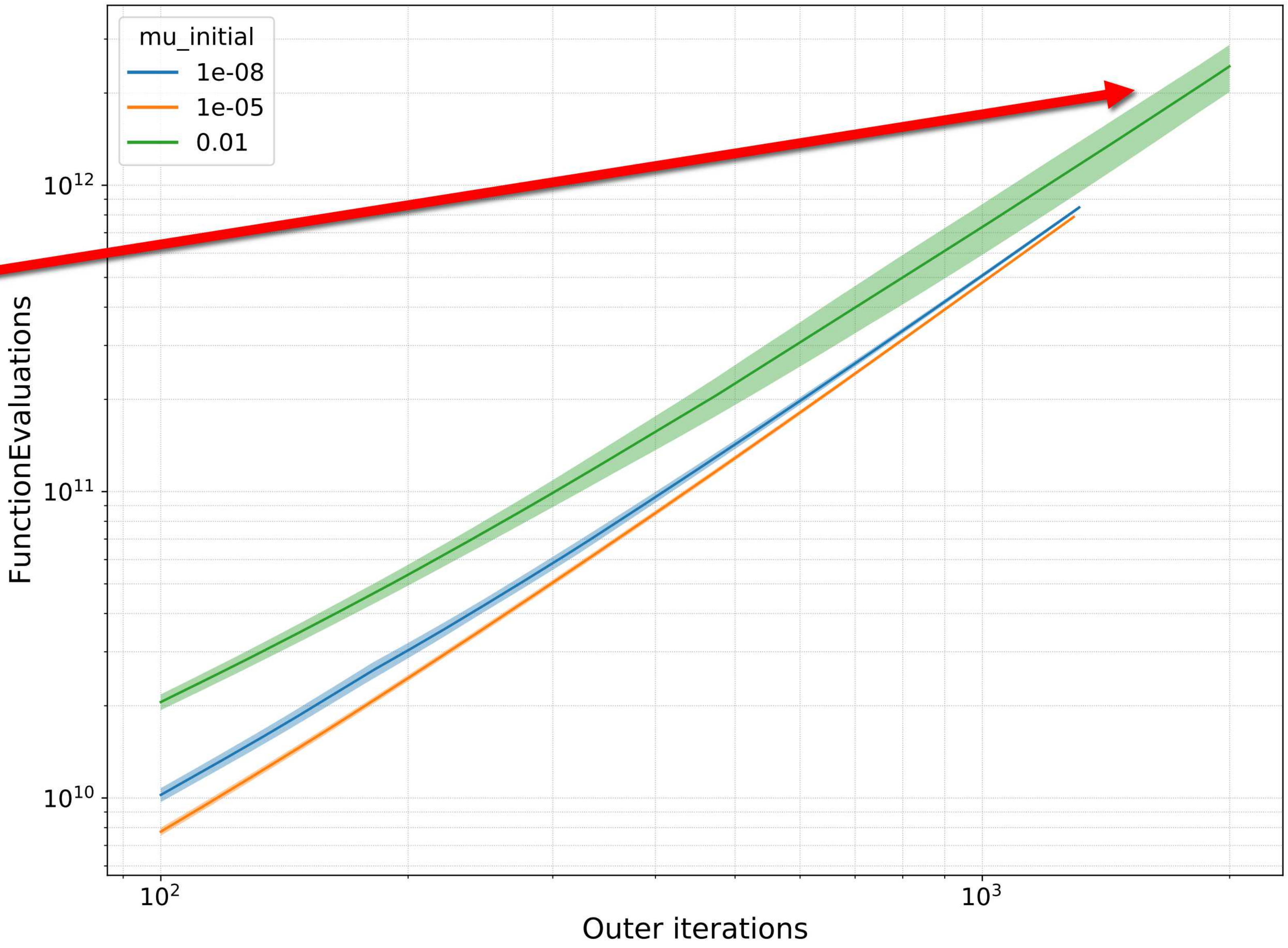


# Example: Max. no. of backtracking steps in line search



# Example: Initial value of damping parameter

Skewed by slowly converging experiments



# Example: Factor to reduce line search step size


Aggressive line search step size reductions may accelerate convergence

# Conclusions

Knowledge gained

- Some software parameters are sensitive to bounds on values
- Varying parameters can:
  - Impact algorithm performance
  - Result in qualitatively different solutions

Open questions

- How to avoid suboptimal local minimizers?

Future work

- Dynamic updates
- Coupled sensitivities among parameters

# Thank you!

# Questions?

SparTen https://gitlab.com/tensors/sparten

Jeremy M. Myers jmmyers@cs.wm.edu, jermyer@sandia.gov
Daniel M. Dunlavy dmdunla@sandia.gov
Keita Teranishi knteran@sandia.gov
D.S. Hollman dshollm@sandia.gov