PARALLEL PDE-CONSTRAINED OPTIMIZATION IN STAR-P®

1

Chad Lieberman

May 15, 2008

--- OUTLINE ---

- Introduction
- Background and motivation
- Formulation
- Computation
  - STIRNCG
- Parallelism
  - ppSTIRNCG
- Results
- Future work

--- INTRODUCTION ---

- Engineers require fast software for PDE-constrained optimization problems
  - Optimal control
  - Shape optimization
  - Inverse problems

- Derivative-based optimization routines utilize user-supplied objective function, gradient, and hessian calculations
  - In PDE-constrained optimization, these computations are expensive

- Derivative-free optimization methods are often not feasible for PDE-constrained problems because they do not scale well with the number of design variables

- Optimization routine must run on cluster to avoid frequent server/client communication

--- BACKGROUND / MOTIVATION ---

- Gradient-based optimization routines come with many adjectives
  - Newton
  - Conjugate gradient
  - Trust-region
  - Interior-reflective
  - Subspace
  - Inexact

- Engineers need faster turn-around time between design iterations
  - Aircraft wings
  - Circuit interconnect
  - Engines

- Goal: Implement PDE-constrained optimization routine in star-P

--- FORMULATION ---

- PDE-constrained optimization problem

$$u^* = \arg\min_{u \in \mathcal{U}} \mathcal{J} = \frac{1}{2}\|y - y_D\|^2 + \frac{1}{2}\beta\|u\|^2 \qquad (1)$$

$$\begin{aligned} \texttt{subject to} \quad R(u,x) &= 0 \\ y &= C(x) \end{aligned}$$

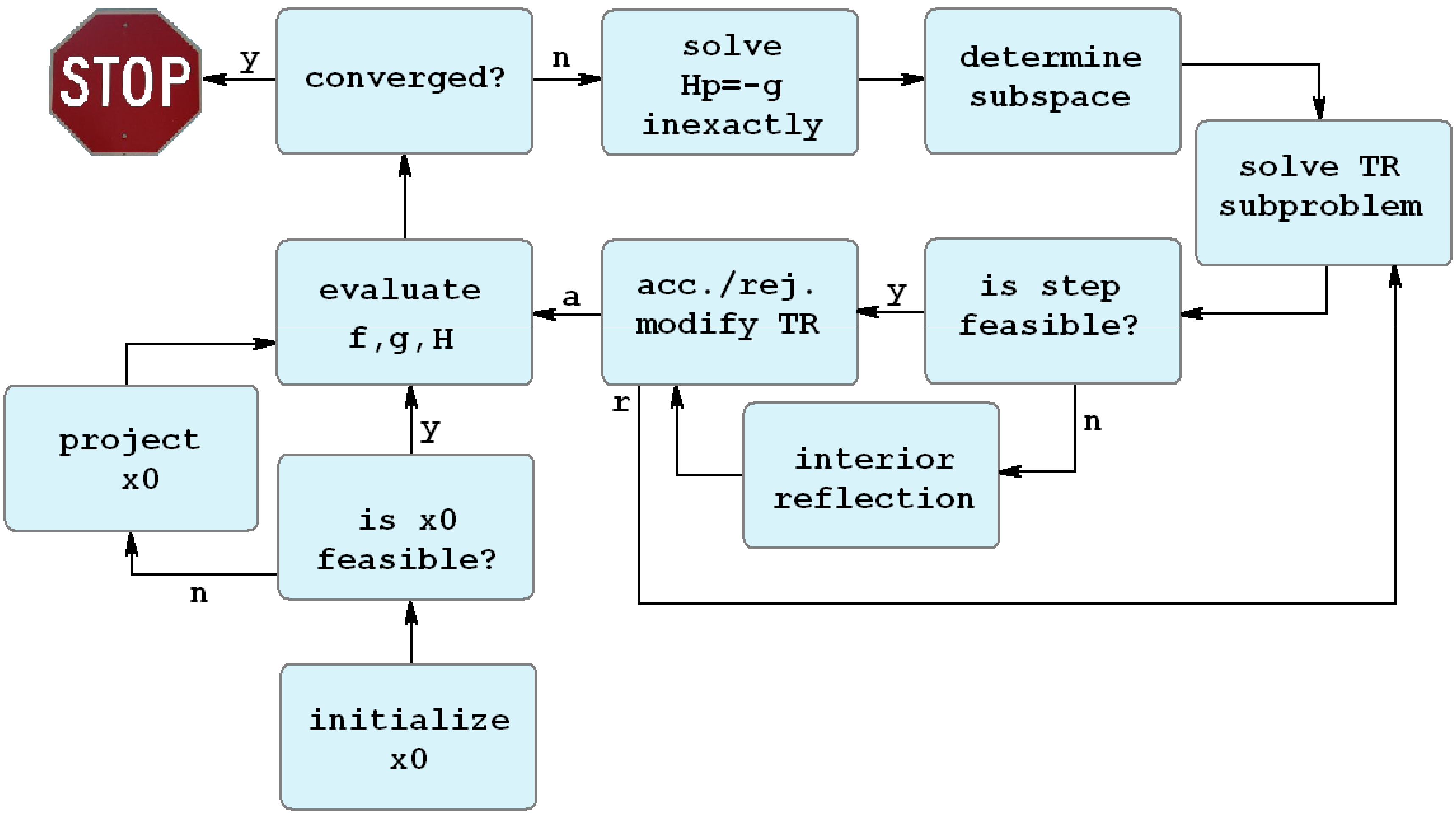
- Construct Lagrangian functional

$$\mathcal{L} = \mathcal{J} + \lambda^T R(u,x)$$

--- FORMULATION ---

- Calculate gradient and hessian of Lagrangian functional
  - Gradient $\nabla_u \mathcal{L}$
  - Hessian $\nabla_{uu} \mathcal{L}$
- Objective function evaluation depends on a PDE solve
- Gradient evaluation requires an adjoint solve
- Hessian is a combination of forward and adjoint solves

--- COMPUTATION ---

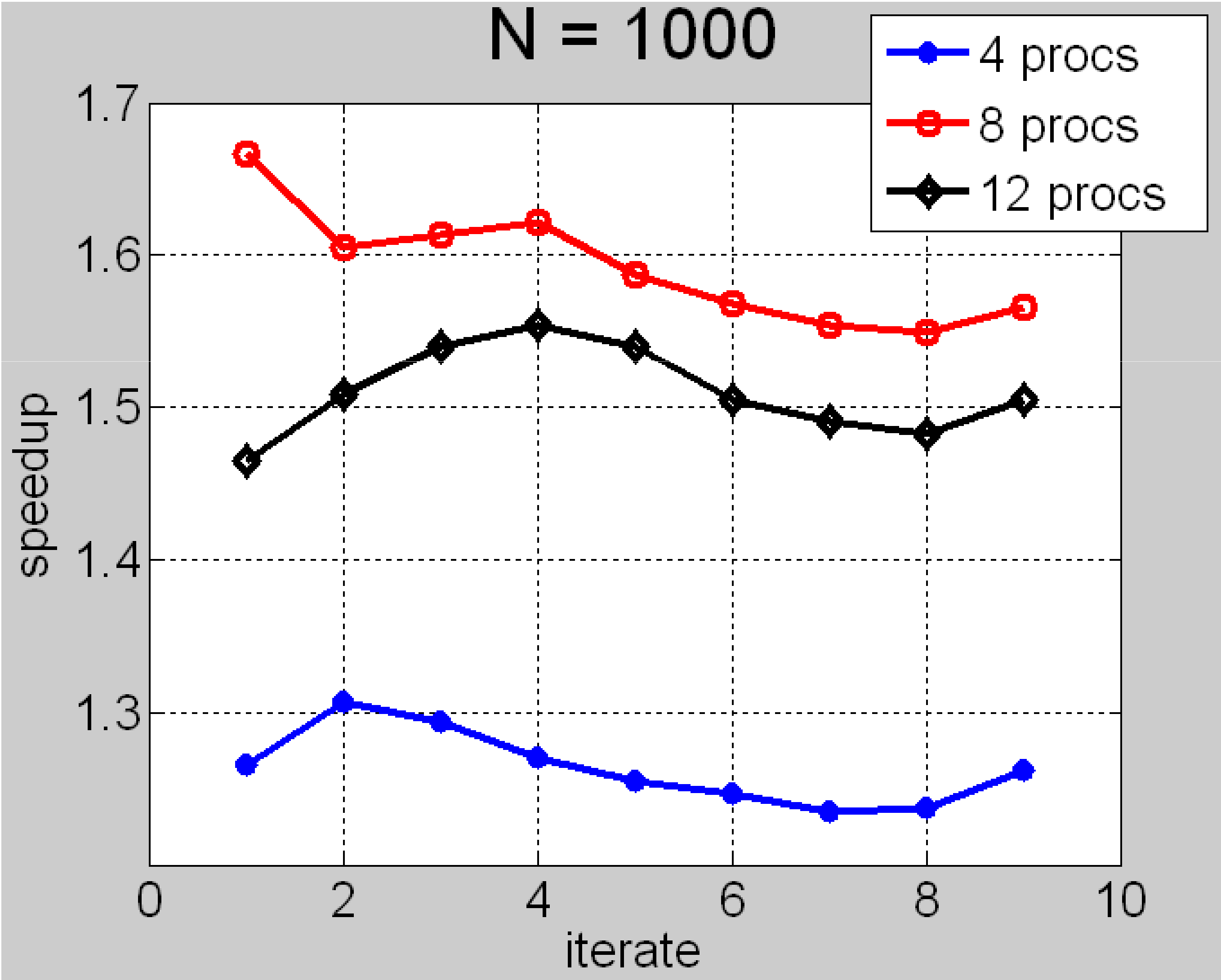


--- PARALLELISM ---

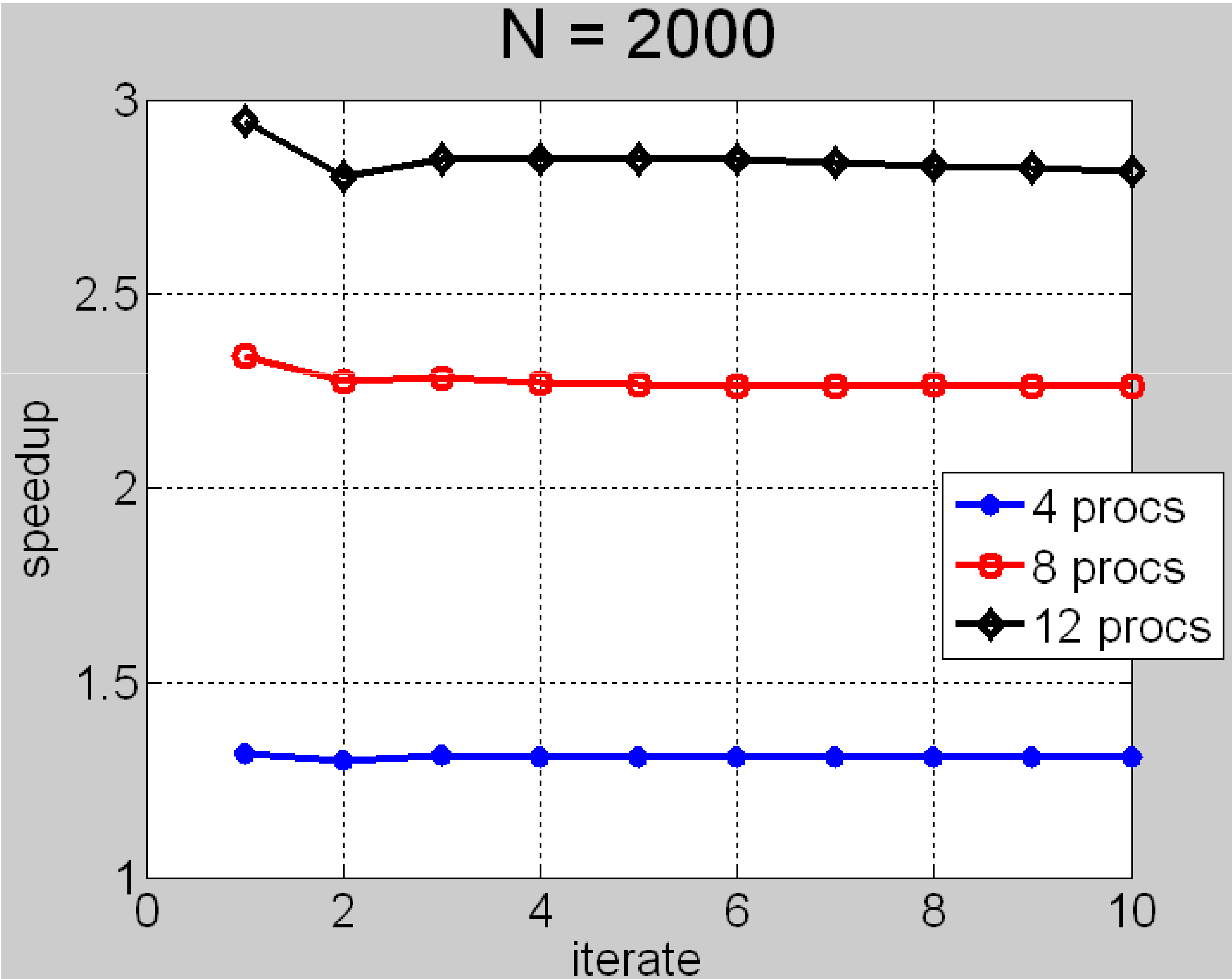
- Data parallelism
  - Objective function
  - Gradient
  - Hessian-vector product
  - Conjugate gradient solve

- Communication
  - Run optimization routine entirely on server
  - ppfmincon
  - Node-to-node communication abstracted by star-P

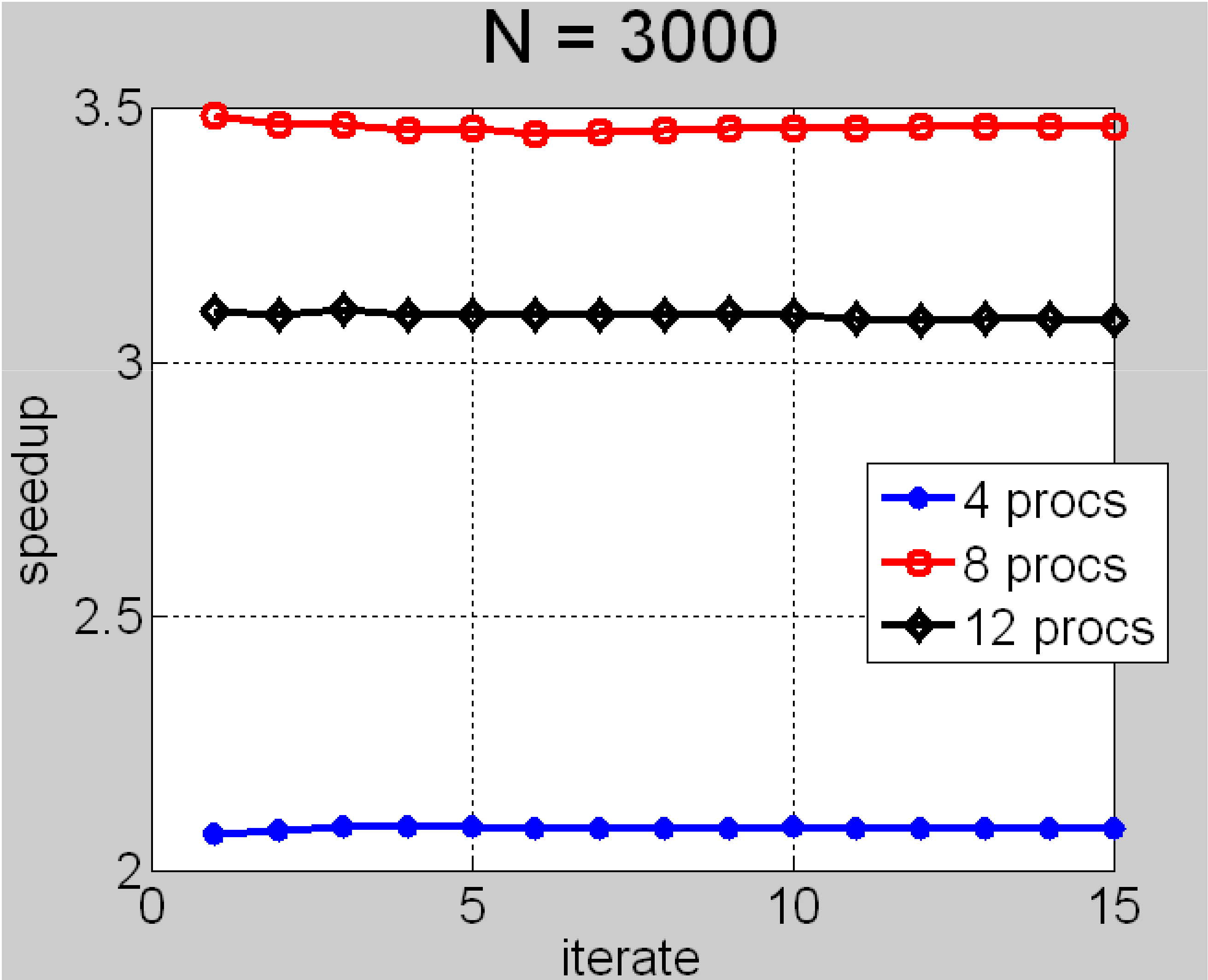
--- RESULTS ---


N = 1000
4 procs
8 procs
12 procs
speedup
1.7
1.6
1.5
1.4
1.3
0
2
4
6
8
10
iterate

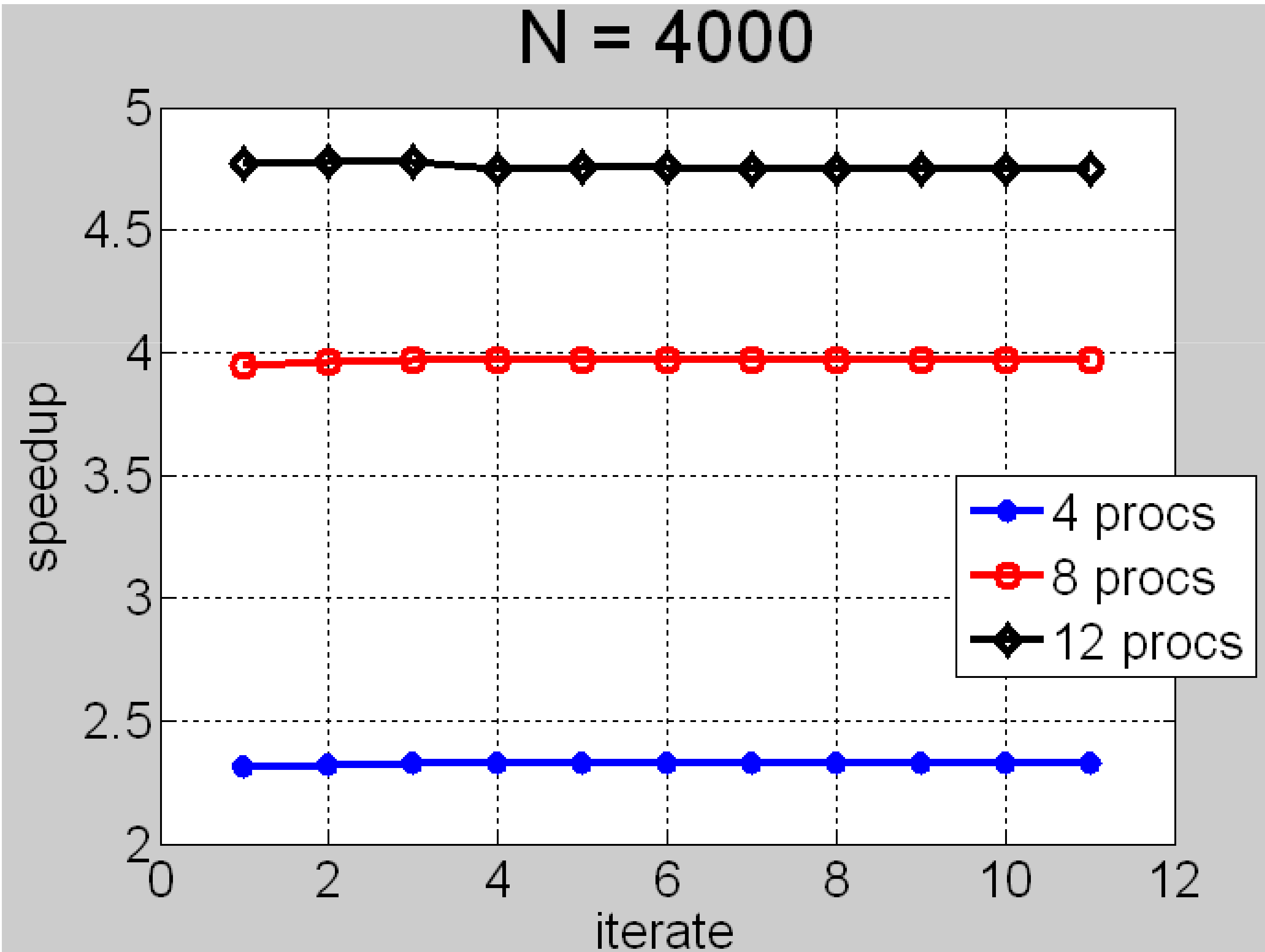
--- RESULTS ---


N = 2000
speedup
iterate
3
2.5
2
1.5
1
0
2
4
6
8
10
4 procs
8 procs
12 procs

--- RESULTS ---


N = 3000
3.5
3
2.5
2
0
5
10
15
speedup
iterate
4 procs
8 procs
12 procs

--- RESULTS ---


N = 4000
speedup
iterate
5
4.5
4
3.5
3
2.5
2
0
2
4
6
8
10
12
4 procs
8 procs
12 procs

--- FUTURE WORK ---

- Include additional optimization routines
- Find users
- Work with ISC to get ppeval with dsparse
- Refactor STIRNCG
- Better scalability

--- QUESTIONS ---

Advisor: Professor Karen Willcox (ACDL)

Thank you to Tan Bui-Thanh (MIT) for helpful discussion about STIRNCG.
Thanks also to Professor Alan Edelman (MIT), Sudarshan Raghunathan, and Viral Shah (ISC).