How to Choose an Algorithm

Matthew Knepley

Computer Science and Engineering
University at Buffalo

Applied and Computational Mathematics
Virtual Seminar, University of Edinburgh
Edinburgh, UK   November 11, 2020

RELACS People

How do we choose an algorithm?

How do we choose an algorithm?

We choose the fastest one…

Timing is tricky. It’s sensitive to

Timing is tricky. It's sensitive to

machine characteristics

Timing is tricky. It’s sensitive to

machine characteristics

problem details

Proxy measures can simplify design:

Proxy measures can simplify design:

Computation (HPL)

Proxy measures can simplify design:

Computation	(HPL)
Bandwidth	(Roofline)

Proxy measures can simplify design:

Computation	(HPL)
Bandwidth	(Roofline)
Latency	(LogP)

Proxy measures can simplify design:

Computation	(HPL)
Bandwidth	(Roofline)
Latency	(LogP)
Concurrency	

These models can answer…

These models can answer…

Does this implementation scale weakly?

These models can answer…

Does this implementation scale weakly? strongly?

These models can answer…

Is one implementation more efficient than another on this machine?

What about questions like...

What about questions like...

Should I discretize this problem with CG or DG?

What about questions like...

Should I solve using the Picard or Newton Method?

The key notion we are missing is

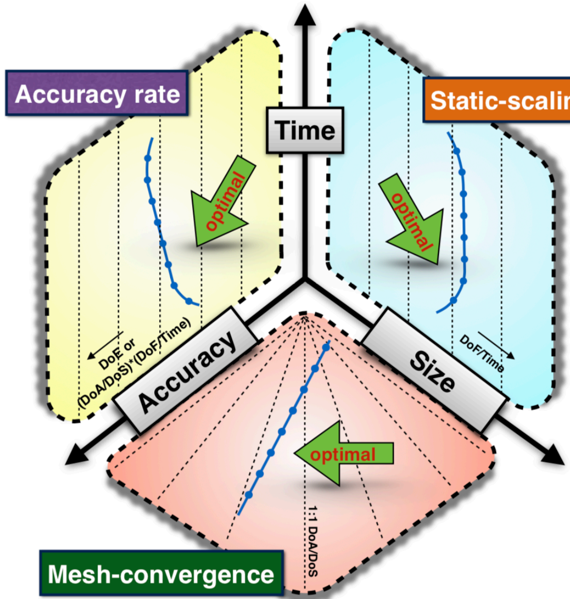
The key notion we are missing is

accuracy

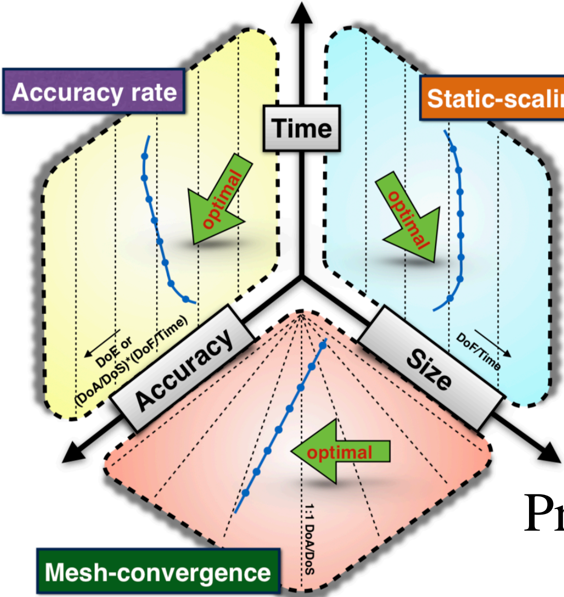
The key notion we are missing is

accuracy

It distinguishes algorithms with different convergence behavior (**ChangFabienKnepleyMills2018**)

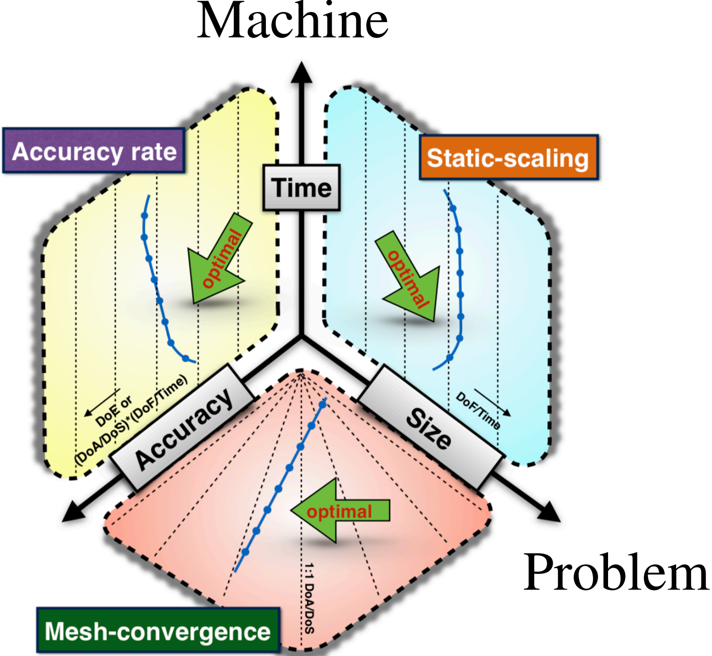


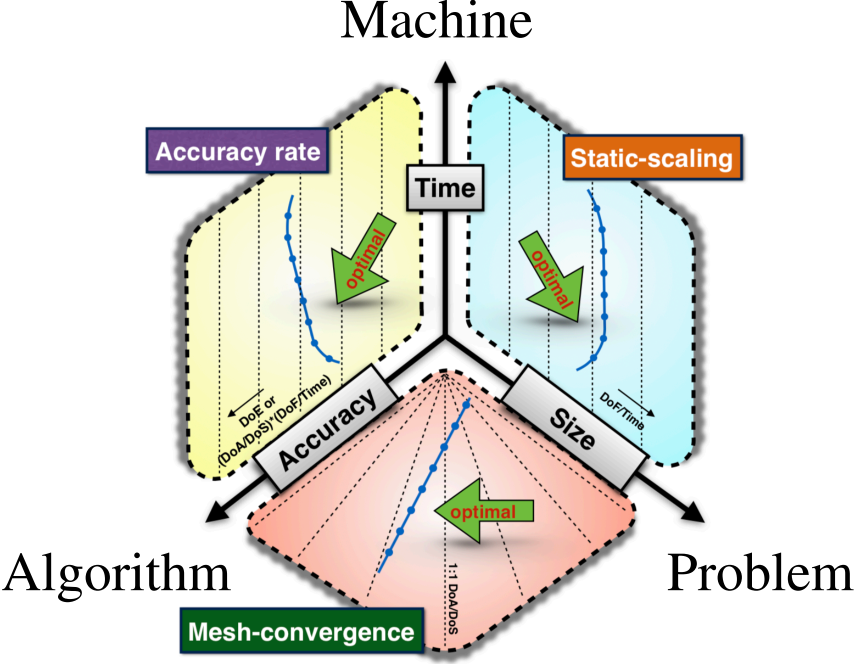
Time
Accuracy rate
Static-scaling
Mesh-convergence
Accuracy
Size
optimal
optimal
optimal
DoE or
(DoA/DoS)*(DoF/Time)
DoF/Time
1:1 DoA/DoS



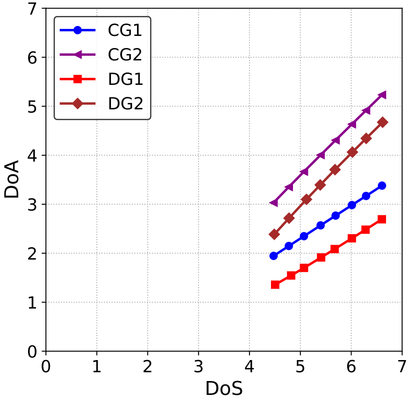
Time
Accuracy rate
Static-scaling
optimal
optimal
optimal
DoE or
(DoA/DoS)*(DoF/Time)
Accuracy
Size
DoF/Time
1:1 DoA/DoS
Mesh-convergence

Problem

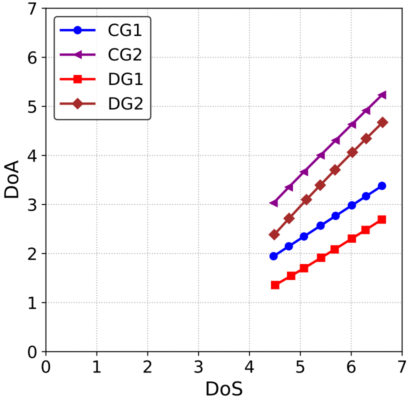

Machine
Accuracy rate
Time
Static-scaling
optimal
optimal
DoE or (DoA/DoS)*(DoF/Time)
Accuracy
Size
DoF/Time
optimal
1:1 DoA/DoS
Mesh-convergence
Problem


Machine
Time
Accuracy rate
Static-scaling
optimal
optimal
optimal
DoE or
(DoA/DoS)*(DoF/Time)
Accuracy
Size
DoF/Time
1:1 DoA/DoS
Algorithm
Problem
Mesh-convergence

Mesh Convergence Diagram

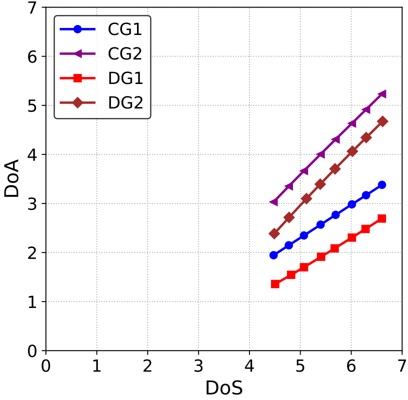


Mesh Convergence Diagram



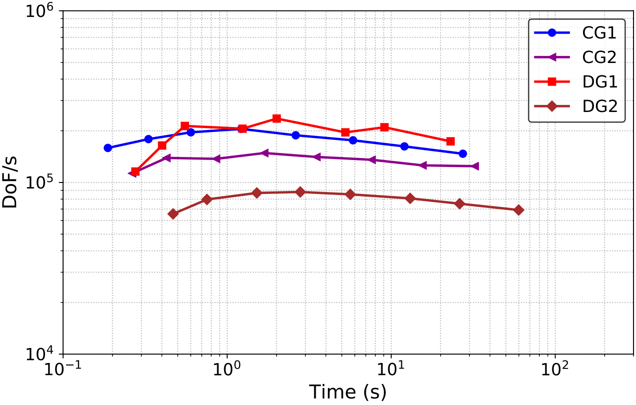
1/error vs. size

Mesh Convergence Diagram

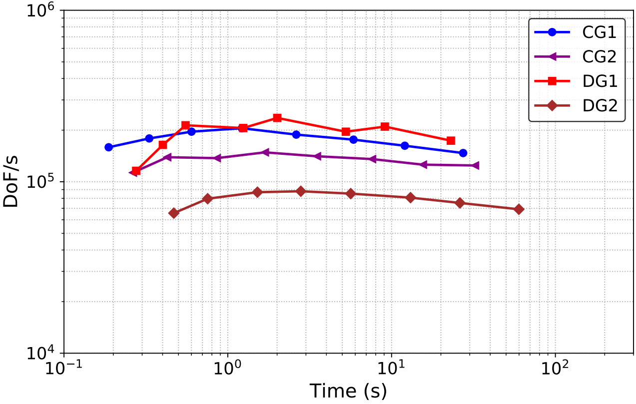


Does my Algorithm solve this Problem?

Static Scaling Diagram

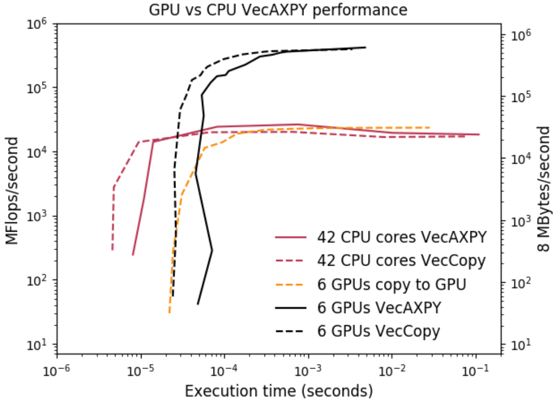


Static Scaling Diagram



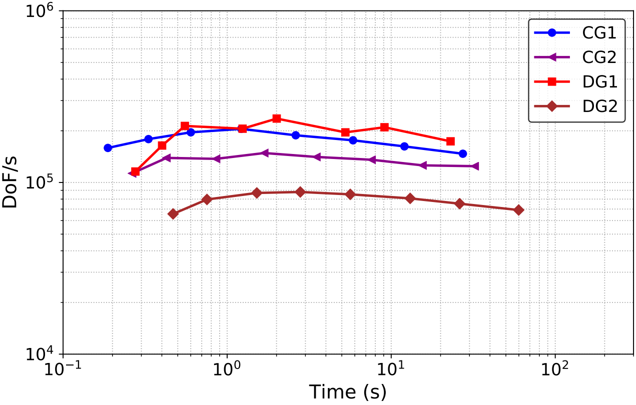
size/time vs. time

Static Scaling Diagram



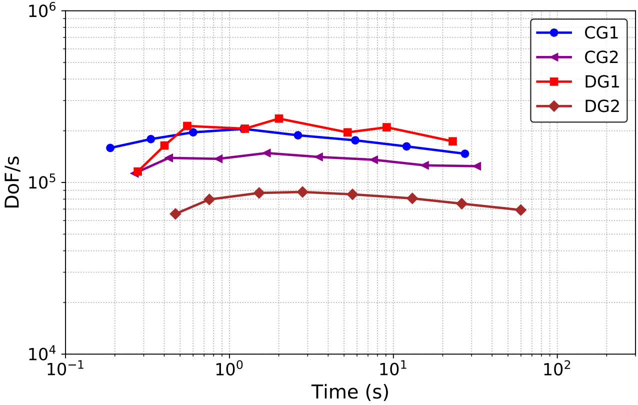
size/time vs. time

Static Scaling Diagram



size/time vs. time

Static Scaling Diagram



Is my Algorithm efficient on this Machine?

How should we measure accuracy?

How should we measure accuracy?

accuracy rate $\frac{e}{T}$

How should we measure accuracy?

accuracy rate $\frac{e}{T}$

Marginal accuracy rate falls off steeply with problem size

Consider an optimal PDE solver:

Consider an optimal PDE solver:

$$T = Wh^{-d} \text{ and } e = Ch^{\alpha}$$

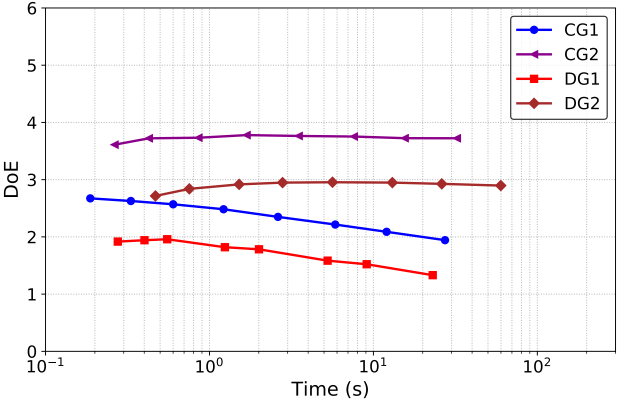
Consider an optimal PDE solver:

$$T = Wh^{-d} \text{ and } e = Ch^{\alpha}$$

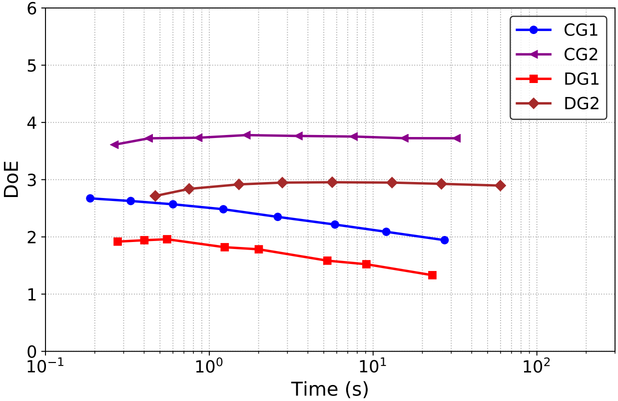
The error-time has a simple form

$$\begin{aligned} & -\log(e \cdot T) \\ = & -\log\left(Ch^{\alpha}Wh^{-d}\right) \\ = & (d-\alpha)\log(h) - \log(CW) \end{aligned}$$

Efficacy Diagram

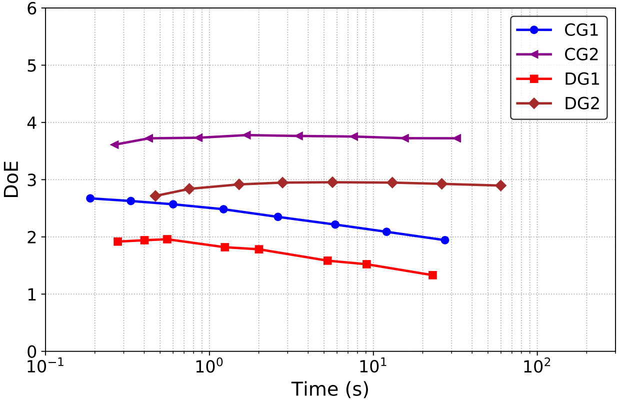


Efficacy Diagram



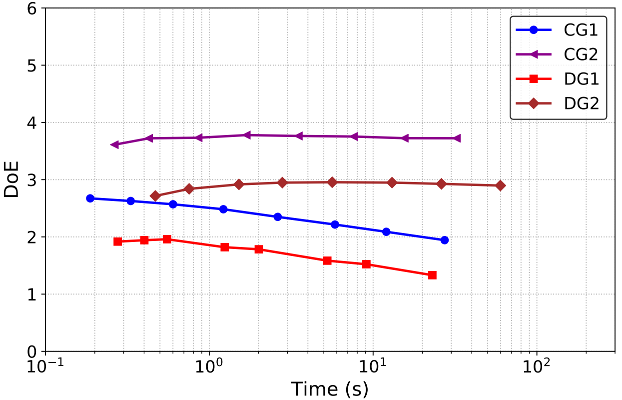
1/error-time vs. time

Efficacy Diagram



$$\frac{1/\text{error}}{\text{size}} \times \frac{\text{size}/\text{time}}{\text{time}} = \frac{1/(\text{error}\cdot\text{time})}{\text{time}}$$

Efficacy Diagram



Does my Algorithm solve this Problem efficiently on this Machine?

Efficacy vs. Static Scaling

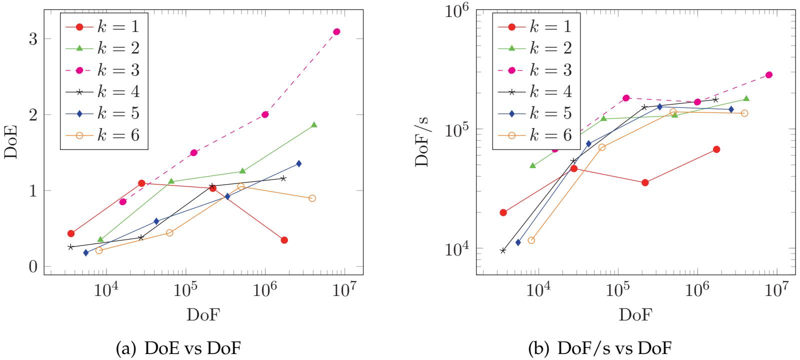


Figure 17: Time-accuracy performance analysis for the nearly incompressible problem ($\lambda = 10^6$).

(Fabien2019)

What else could we analyze?

What else could we analyze?

Communication-Avoiding (CA) algorithms have exciting lower bounds

(**BallardDemmelHoltzSchwartz2011**)

What else could we analyze?

CA TSQR is a great success

(DemmelGrigoriHoemmenLangou201

What else could we analyze?

CA Krylov not a success

What else could we analyze?

CA Krylov not a success

Accuracy depends on coarse grid communication in preconditioner

Future Questions:

Future Questions:

Is there a variational characterization of optimal algorithms?

Future Questions:

Can we think of error-time as an Algorithmic Action?

References I