A Space Decomposition Framework for Nonlinear Optimization

Zaikun Zhang

IRIT-ENSEEIHT, Toulouse, France

Joint work with Serge Gratton and Luis Nunes Vicente

Partially supported by IRT and Foundation STAE

13 January 2016

The problem

In this talk,

- to make things simple, consider an unconstrained problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

- suppose that minimizing f directly is out of reach

The problem

In this talk,

- to make things simple, consider an unconstrained problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

- suppose that minimizing f directly is out of reach
- aim for methods based on decomposition of the variable space for
  - large-scale optimization with derivatives
  - relatively large derivative-free optimization
- concerns: parallelism, scalability, convergence with guaranteed rate

The idea

- Divide the original problem into a series of ‘smaller’ problems, and solve each of them.

The idea

- Divide the original problem into a series of 'smaller' problems, and solve each of them.
- The division is done by decomposing the space of variables.

Space decomposition methods in optimization

- Block coordinate descent (deterministic/randomized)

Space decomposition methods in optimization

- Block coordinate descent (deterministic/randomized)
  - Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, *SIAM J. Optim.*, 22:341–362, 2012
  - S. J. Wright, Coordinate descent algorithms, *Marth. Program.*, 151:3–34, 2015

Space decomposition methods in optimization

- Block coordinate descent (deterministic/randomized)
  - Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, *SIAM J. Optim.*, 22:341–362, 2012
  - S. J. Wright, Coordinate descent algorithms, *Marth. Program.*, 151:3–34, 2015
- M. C. Ferris and O. L. Mangasarian, Parallel variable distribution, *SIAM J. Optim.*, 4:815–832, 1994
- M. Fukushima, Parallel variable transformation in unconstrained optimization, *SIAM J. Optim.*, 8:658–672, 1998
- C. Audet, J. E. Dennis Jr., and S. Le Digabel, Parallel space decomposition of the mesh adaptive direct search algorithm, *SIAM J. Optim.*, 19:1150–1170, 2008

Domain decomposition for PDEs and linear systems

- Successful applications to PDE-based optimization
- Parallel methods with fine scalability
- Many techniques worth learning:
  Restricted Additive Schwarz (RAS), coarse space, ...

Solve a linear system

Consider a linear system

$$A \begin{pmatrix} u \\ v \\ w \\ x \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}.$$

Solve a linear system

Consider a linear system

$$A\begin{pmatrix} u \\ v \\ w \\ x \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}.$$

Solve it iteratively:

$$\underbrace{(u_{k+1}, v_{k+1}, w_{k+1}, x_{k+1})}_{\text{Approximation}_{k+1}} = \underbrace{(u_k, v_k, w_k, x_k)}_{\text{Approximation}_k} + \underbrace{(\Delta u, \Delta v, \Delta w, \Delta x)}_{\text{Correction}}.$$

Solve a linear system

Consider a linear system

$$A \begin{pmatrix} u \\ v \\ w \\ x \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}.$$

Solve it iteratively:

$$\underbrace{(u_{k+1}, v_{k+1}, w_{k+1}, x_{k+1})}_{\text{Approximation}_{k+1}} = \underbrace{(u_k, v_k, w_k, x_k)}_{\text{Approximation}_k} + \underbrace{(\Delta u, \Delta v, \Delta w, \Delta x)}_{\text{Correction}}.$$

Ideally,

$$A \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \\ \Delta x \end{pmatrix} = \underbrace{\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} - A \begin{pmatrix} u_k \\ v_k \\ w_k \\ x_k \end{pmatrix}}_{\text{Residual}} \equiv \begin{pmatrix} a_k \\ b_k \\ c_k \\ d_k \end{pmatrix}.$$

Solve a linear system

Divide the unknowns into groups, and update them separately.

$$(u, v, w, x) \xrightarrow{\text{divide}} (u, v, w) \text{ and } (v, w, x)$$

Solve a linear system

Divide the unknowns into groups, and update them separately.

$$(u, v, w, x) \xrightarrow{\text{divide}} (u, {\color{red}v}, {\color{red}w}) \text{ and } ({\color{red}v}, {\color{red}w}, x)$$

Solve a linear system

Divide the unknowns into groups, and update them separately.

$$(u, v, w, x) \xrightarrow{\text{divide}} (u, v, w) \text{ and } (v, w, x)$$

Compute $(\Delta u, \Delta v, \Delta w)$, while x being fixed:

$$A \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \\ 0 \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \\ c_k \\ d_k \end{pmatrix}$$

Solve a linear system

Divide the unknowns into groups, and update them separately.

$$(u, v, w, x) \xrightarrow{\text{divide}} (u, v, w) \text{ and } (v, w, x)$$

Compute $(\Delta u, \Delta v, \Delta w)$, while x being fixed:

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \\ 0 \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \\ c_k \\ d_k \end{pmatrix}$$

Solve a linear system

Divide the unknowns into groups, and update them separately.

$$(u, v, w, x) \xrightarrow{\text{divide}} (u, v, w) \text{ and } (v, w, x)$$

Compute $(\Delta u, \Delta v, \Delta w)$, while x being fixed:

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & \\ A_{21} & A_{22} & A_{23} & \\ A_{31} & A_{32} & A_{33} & \\ A_{41} & A_{42} & A_{43} & \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \\ \\ \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \\ c_k \\ d_k \end{pmatrix}$$

Solve a linear system

Divide the unknowns into groups, and update them separately.

$$(u, v, w, x) \xrightarrow{\text{divide}} (u, v, w) \text{ and } (v, w, x)$$

Compute $(\Delta u, \Delta v, \Delta w)$, while x being fixed:

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \\ c_k \end{pmatrix}$$

Solve a linear system

Divide the unknowns into groups, and update them separately.

$$(u, v, w, x) \xrightarrow{\text{divide}} (u, v, w) \text{ and } (v, w, x)$$

Compute $(\Delta u, \Delta v, \Delta w)$, while x being fixed:

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & \\ A_{21} & A_{22} & A_{23} & \\ A_{31} & A_{32} & A_{33} & \\ & & & \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \\ \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \\ c_k \\ \end{pmatrix}$$

Compute $(\Delta v, \Delta w, \Delta x)$, while u being fixed:

$$\begin{pmatrix} & & & \\ & A_{22} & A_{23} & A_{24} \\ & A_{32} & A_{33} & A_{34} \\ & A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} \Delta v \\ \Delta w \\ \Delta x \end{pmatrix} = \begin{pmatrix} b_k \\ c_k \\ d_k \end{pmatrix}$$

Solve a linear system

Divide the unknowns into groups, and update them separately.

$$(u, v, w, x) \xrightarrow{\text{divide}} (u, v, w) \text{ and } (v, w, x)$$

Compute $(\Delta u, \Delta v, \Delta w)$, while x being fixed:

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & \\ A_{21} & A_{22} & A_{23} & \\ A_{31} & A_{32} & A_{33} & \\ & & & \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \\ \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \\ c_k \\ \end{pmatrix} \Rightarrow \begin{pmatrix} \Delta u^{(1)} \\ \Delta v^{(1)} \\ \Delta w^{(1)} \\ \end{pmatrix}$$

Compute $(\Delta v, \Delta w, \Delta x)$, while u being fixed:

$$\begin{pmatrix} & & & \\ & A_{22} & A_{23} & A_{24} \\ & A_{32} & A_{33} & A_{34} \\ & A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} \Delta v \\ \Delta w \\ \Delta x \end{pmatrix} = \begin{pmatrix} b_k \\ c_k \\ d_k \end{pmatrix} \Rightarrow \begin{pmatrix} \Delta v^{(2)} \\ \Delta w^{(2)} \\ \Delta x^{(2)} \end{pmatrix}$$

Solve a linear system

Divide the unknowns into groups, and update them separately.

$$(u, v, w, x) \xrightarrow{\text{divide}} (u, {\color{red}v}, {\color{red}w}) \text{ and } ({\color{red}v}, {\color{red}w}, x)$$

Compute $(\Delta u, \Delta v, \Delta w)$, while x being fixed:

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & \\ A_{21} & A_{22} & A_{23} & \\ A_{31} & A_{32} & A_{33} & \\ & & & \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \\ c_k \end{pmatrix} \Rightarrow \begin{pmatrix} \Delta u^{(1)} \\ {\color{red}\Delta v^{(1)}} \\ {\color{red}\Delta w^{(1)}} \end{pmatrix}$$

Compute $(\Delta v, \Delta w, \Delta x)$, while u being fixed:

$$\begin{pmatrix} & & & \\ & A_{22} & A_{23} & A_{24} \\ & A_{32} & A_{33} & A_{34} \\ & A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} \Delta v \\ \Delta w \\ \Delta x \end{pmatrix} = \begin{pmatrix} b_k \\ c_k \\ d_k \end{pmatrix} \Rightarrow \begin{pmatrix} {\color{red}\Delta v^{(2)}} \\ {\color{red}\Delta w^{(2)}} \\ \Delta x^{(2)} \end{pmatrix}$$

How to define $(\Delta u, {\color{red}\Delta v}, {\color{red}\Delta w}, \Delta x)$?

Solve a linear system

Strategies to define $(\Delta u, \Delta v, \Delta w, \Delta x)$:

- $(\Delta u^{(1)}, \Delta v^{(1)} + \Delta v^{(2)}, \Delta w^{(1)} + \Delta w^{(2)}, \Delta x^{(2)})$

  Corresponds to the Additive Schwarz (AS) preconditioner.

Solve a linear system

Strategies to define $(\Delta u, \Delta v, \Delta w, \Delta x)$:

- $(\Delta u^{(1)}, \Delta v^{(1)} + \Delta v^{(2)}, \Delta w^{(1)} + \Delta w^{(2)}, \Delta x^{(2)})$

  Corresponds to the Additive Schwarz (AS) preconditioner.

  The corrections on the overlap are superfluous.

Solve a linear system

Strategies to define $(\Delta u, \Delta v, \Delta w, \Delta x)$:

- $(\Delta u^{(1)}, \Delta v^{(1)} + \Delta v^{(2)}, \Delta w^{(1)} + \Delta w^{(2)}, \Delta x^{(2)})$

  Corresponds to the Additive Schwarz (AS) preconditioner.

  The corrections on the overlap are superfluous.

- $(\Delta u^{(1)}, \theta^{(1)} \Delta v^{(1)} + \theta^{(2)} \Delta v^{(2)}, \xi^{(1)} \Delta w^{(1)} + \xi^{(2)} \Delta w^{(2)}, \Delta x^{(2)})$

  with $\theta^{(1)} + \theta^{(2)} = \xi^{(1)} + \xi^{(2)} = 1$.

Solve a linear system

Strategies to define $(\Delta u, \Delta v, \Delta w, \Delta x)$:

- $(\Delta u^{(1)}, \Delta v^{(1)} + \Delta v^{(2)}, \Delta w^{(1)} + \Delta w^{(2)}, \Delta x^{(2)})$

  Corresponds to the Additive Schwarz (AS) preconditioner.

  The corrections on the overlap are superfluous.

- $(\Delta u^{(1)}, \theta^{(1)}\Delta v^{(1)} + \theta^{(2)}\Delta v^{(2)}, \xi^{(1)}\Delta w^{(1)} + \xi^{(2)}\Delta w^{(2)}, \Delta x^{(2)})$

  with $\theta^{(1)} + \theta^{(2)} = \xi^{(1)} + \xi^{(2)} = 1$.

  In specific, when $\theta^{(1)} = \xi^{(2)} = 1$ and $\theta^{(2)} = \xi^{(1)} = 0$, we have

- $(\Delta u^{(1)}, \Delta v^{(1)}, \Delta w^{(2)}, \Delta x^{(2)})$

  Corresponds to the Restricted Additive Schwarz (RAS) preconditioner.

Solve a linear system

Another approach: let

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} \Delta u^{(1)} \\ \Delta v^{(1)} \\ \Delta w^{(1)} \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} A_{22} & A_{23} & A_{24} \\ A_{32} & A_{33} & A_{34} \\ A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} \Delta v^{(2)} \\ \Delta w^{(2)} \\ \Delta x^{(2)} \end{pmatrix} = \begin{pmatrix} 0 \\ c_k \\ d_k \end{pmatrix}.$$

Then define the update to be

- $(\Delta u^{(1)}, \Delta v^{(1)} + \Delta v^{(2)}, \Delta w^{(1)} + \Delta w^{(2)}, \Delta x^{(2)})$
  - Additive Schwarz with Harmonic extension (ASH)

Solve a linear system

Another approach: let

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} \Delta u^{(1)} \\ \Delta v^{(1)} \\ \Delta w^{(1)} \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} A_{22} & A_{23} & A_{24} \\ A_{32} & A_{33} & A_{34} \\ A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} \Delta v^{(2)} \\ \Delta w^{(2)} \\ \Delta x^{(2)} \end{pmatrix} = \begin{pmatrix} 0 \\ c_k \\ d_k \end{pmatrix}.$$

Then define the update to be

- $(\Delta u^{(1)}, \Delta v^{(1)} + \Delta v^{(2)}, \Delta w^{(1)} + \Delta w^{(2)}, \Delta x^{(2)})$
  - Additive Schwarz with Harmonic extension (ASH)
- $(\Delta u^{(1)}, \Delta v^{(1)}, \Delta w^{(2)}, \Delta x^{(2)})$
  - Symmetrized Additive Schwarz (RASH)

Solve a linear system

Another approach: let

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} \Delta u^{(1)} \\ \Delta v^{(1)} \\ \Delta w^{(1)} \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} A_{22} & A_{23} & A_{24} \\ A_{32} & A_{33} & A_{34} \\ A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} \Delta v^{(2)} \\ \Delta w^{(2)} \\ \Delta x^{(2)} \end{pmatrix} = \begin{pmatrix} 0 \\ c_k \\ d_k \end{pmatrix}.$$

Then define the update to be

- $(\Delta u^{(1)}, \Delta v^{(1)} + \Delta v^{(2)}, \Delta w^{(1)} + \Delta w^{(2)}, \Delta x^{(2)})$
  - Additive Schwarz with Harmonic extension (ASH)
- $(\Delta u^{(1)}, \Delta v^{(1)}, \Delta w^{(2)}, \Delta x^{(2)})$
  - Symmetrized Additive Schwarz (RASH)

X. C. Cai and M. Sarkis, "A restricted additive Schwarz preconditioner for general sparse linear systems", *SIAM J. Sci. Comput.*, 21:792–797, 1999

Solve a nonlinear optimization problem

How to extend the ideas to nonlinear optimization?

Solve a nonlinear optimization problem

How to extend the ideas to nonlinear optimization?

Consider a nonlinear optimization problem

$$\min f(u, v, w, x).$$

Given an approximate solution (u_k, v_k, w_k, x_k), look for an update.

Solve a nonlinear optimization problem

How to extend the ideas to nonlinear optimization?

Consider a nonlinear optimization problem

$$\min f(u, v, w, x).$$

Given an approximate solution (u_k, v_k, w_k, x_k), look for an update.

- Decomposition:
  - $(u, v, w, x) \xrightarrow{\text{divide}} (u, v, w)$ and (v, w, x).

Solve a nonlinear optimization problem

How to extend the ideas to nonlinear optimization?

Consider a nonlinear optimization problem

$$\min f(u, v, w, x).$$

Given an approximate solution (u_k, v_k, w_k, x_k), look for an update.

- Decomposition:
  - $(u, v, w, x) \xrightarrow{\text{divide}} (u, v, w)$ and (v, w, x).
  - Compute $(\Delta u, \Delta v, \Delta w)$, while x being fixed:

$$\min \ f(u_k + \Delta u, v_k + \Delta v, w_k + \Delta w, x_k) \Longrightarrow \left(\Delta u^{(1)}, \Delta v^{(1)}, \Delta w^{(1)}\right).$$

  - Compute $(\Delta v, \Delta w, \Delta x)$, while u being fixed:

$$\min \ f(u_k, v_k + \Delta v, w_k + \Delta w, x_k + \Delta x) \Longrightarrow \left(\Delta v^{(2)}, \Delta w^{(2)}, \Delta x^{(2)}\right).$$

Solve a nonlinear optimization problem

How to extend the ideas to nonlinear optimization?

Consider a nonlinear optimization problem

$$\min f(u, v, w, x).$$

Given an approximate solution (u_k, v_k, w_k, x_k), look for an update.

- Decomposition:
  - $(u, v, w, x) \xrightarrow{\text{divide}} (u, v, w)$ and (v, w, x).
  - Compute $(\Delta u, \Delta v, \Delta w)$, while x being fixed:

$$\min \ f(u_k + \Delta u, v_k + \Delta v, w_k + \Delta w, x_k) \Longrightarrow \left(\Delta u^{(1)}, \Delta v^{(1)}, \Delta w^{(1)}\right).$$

  - Compute $(\Delta v, \Delta w, \Delta x)$, while u being fixed:

$$\min \ f(u_k, v_k + \Delta v, w_k + \Delta w, x_k + \Delta x) \Longrightarrow \left(\Delta v^{(2)}, \Delta w^{(2)}, \Delta x^{(2)}\right).$$

- Synchronization: Define the update according to AS or RAS.

Solve a nonlinear optimization problem

What about ASH and RASH?

- Decomposition:

$$\min f(u_k + \Delta u, v_k + \Delta v, w_k + \Delta w, x_k) - \frac{\partial f_k}{\partial w}\Delta w$$
$$\Longrightarrow \left(\Delta u^{(1)}, \Delta v^{(1)}, \Delta w^{(1)}\right)$$

$$\min f(u_k, v_k + \Delta v, w_k + \Delta w, x_k + \Delta x) - \frac{\partial f_k}{\partial v}\Delta v$$
$$\Longrightarrow \left(\Delta v^{(2)}, \Delta w^{(2)}, \Delta x^{(2)}\right)$$

- Synchronization:
  - ASH: $(\Delta u^{(1)}, \Delta v^{(1)} + \Delta v^{(2)}, \Delta w^{(1)} + \Delta w^{(2)}, \Delta x^{(2)})$
  - RASH: $(\Delta u^{(1)}, \Delta v^{(1)}, \Delta w^{(2)}, \Delta x^{(2)})$ (Christian Groß, 2009)

The main ingredients

- Decomposition
- Synchronization

The main ingredients

- Decomposition
  - Decompose the full space into (possibly overlapping) subspaces
  - Define subspace models of the objective function
  - Possibly modify the model gradients according to the overlaps
  - Minimize the (modified) models to obtain subspace steps
- Synchronization

The main ingredients

- Decomposition
  - Decompose the full space into (possibly overlapping) subspaces
  - Define subspace models of the objective function
  - Possibly modify the model gradients according to the overlaps
  - Minimize the (modified) models to obtain subspace steps
- Synchronization
  - Map the steps back to the full space
  - Possibly modify the subspace steps according to the overlaps
  - Take the sum of the (modified) subspace steps to obtain a step

The main ingredients

- Decomposition
  - Decompose the full space into (possibly overlapping) subspaces
  - Define subspace models of the objective function
  - Possibly modify the model gradients according to the overlaps
  - Minimize the (modified) models to obtain subspace steps
- Synchronization
  - Map the steps back to the full space
  - Possibly modify the subspace steps according to the overlaps
  - Take the sum of the (modified) subspace steps to obtain a step

In all the cases, the "modifications" are always linear transformations.

The space decomposition framework

Pick $x \in \mathbb{R}^n$. Then repeat the following.

The space decomposition framework

Pick $x \in \mathbb{R}^n$. Then repeat the following.

Step 1. Decomposition.

For $i \in \{1, \ldots, m\}$, pick $R^{(i)} \in \mathbb{R}^{n^{(i)} \times n}$, choose $h^{(i)} : \mathbb{R}^{n^{(i)}} \to \mathbb{R}$ satisfying

$$\nabla h^{(i)}(0) \; = \; R^{(i)} \nabla f(x),$$

and then obtain $d^{(i)}$ by solving

$$\min \; h^{(i)}(d).$$

The space decomposition framework

Pick $x \in \mathbb{R}^n$. Then repeat the following.

Step 1. Decomposition.

For $i \in \{1, \ldots, m\}$, pick $R^{(i)} \in \mathbb{R}^{n^{(i)} \times n}$, choose $h^{(i)} : \mathbb{R}^{n^{(i)}} \to \mathbb{R}$ satisfying

$$\nabla h^{(i)}(0) \; = \; R^{(i)} \nabla f(x),$$

and then obtain $d^{(i)}$ by solving

$$\min \; h^{(i)}(d).$$

Step 2. Synchronization.

Pick $T^{(i)} \in \mathbb{R}^{n \times n^{(i)}}$ $(i = 1, \ldots, m)$, and let

$$x \; := \; x + \sum_{i=1}^{m} T^{(i)} d^{(i)}.$$

Go to **Step 1**.

Remarks

- $\{R^{(i)}\}$ defines the decomposition; $\{T^{(i)}\}$ defines the synchronization.
- To have 4-dimensional RAS as presented before, take

$$R^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad R^{(2)} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and

$$T^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad T^{(2)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Remarks

- $\{R^{(i)}\}$ defines the decomposition; $\{T^{(i)}\}$ defines the synchronization.
- To have 4-dimensional RAS as presented before, take

$$R^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad R^{(2)} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and

$$T^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad T^{(2)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

- The framework covers AS, RAS, ASH, RASH, and many more.
- The subspaces are not necessarily coordinate subspaces.

Questions

- Does this procedure always converge? — Of course not.

Questions

- Does this procedure always converge? — Of course not.

Is it possible to propose a general framework that

- guarantees global convergence with certified convergence rate,
- accepts inexact solutions of the subspace minimization,
- covers all the synchronization strategies (AS, RAS, ASH, RASH)?

Questions

- Does this procedure always converge? — Of course not.

Is it possible to propose a general framework that

- guarantees global convergence with certified convergence rate,
- accepts inexact solutions of the subspace minimization,
- covers all the synchronization strategies (AS, RAS, ASH, RASH)?

Yes of course!

A space transformation framework

Pick $x \in \mathbb{R}^n$. Then repeat the following.

A space transformation framework

Pick $x \in \mathbb{R}^n$. Then repeat the following.

Step 1. Transformation.

Pick $R \in \mathbb{R}^{N\times n}$, choose $h : \mathbb{R}^N \to \mathbb{R}$ satisfying

$$\nabla h(0) \ = \ R\nabla f(x),$$

and then obtain d by solving

$$\min \ h(d).$$

A space transformation framework

Pick $x \in \mathbb{R}^n$. Then repeat the following.

Step 1. Transformation.

Pick $R \in \mathbb{R}^{N \times n}$, choose $h : \mathbb{R}^N \to \mathbb{R}$ satisfying

$$\nabla h(0) \ = \ R \nabla f(x),$$

and then obtain d by solving

$$\min \ h(d).$$

Step 2. Reconstruction.

Pick $T \in \mathbb{R}^{n \times N}$, and let

$$x \ := \ x + Td.$$

Go to **Step 1**.

Space decomposition v.s. Space transformation

- Space transformation is a special case of space decomposition.
- Space decomposition is a special case of space transformation with

$$N = \sum_{i=1}^{m} n^{(i)}, \quad h = \sum_{i=1}^{m} h^{(i)},$$

$$R = \begin{pmatrix} R^{(1)} \\ \vdots \\ R^{(m)} \end{pmatrix}, \quad T = \begin{pmatrix} T^{(1)} & \dots & T^{(m)} \end{pmatrix},$$

and

$$d = \begin{pmatrix} d^{(1)} \\ \vdots \\ d^{(m)} \end{pmatrix}.$$

The space transformation framework: Globalized version

Pick $x \in \mathbb{R}^n$ and $\sigma > 0$. Then repeat the following.

Step 1. Transformation.
Pick $R \in \mathbb{R}^{N \times n}$, choose $h : \mathbb{R}^N \to \mathbb{R}$ satisfying

$$\nabla h(0) \; = \; R \nabla f(x),$$

and then obtain d by approximately solving

$$\min \; h(d) + \frac{\sigma}{2} \|d\|^2.$$

Step 2. Reconstruction.
Pick $T \in \mathbb{R}^{n \times N}$, and let

$$s \; = \; Td.$$

The space transformation framework: Globalized version

Pick $x \in \mathbb{R}^n$ and $\sigma > 0$. Then repeat the following.

Step 1. Transformation.

Pick $R \in \mathbb{R}^{N \times n}$, choose $h : \mathbb{R}^N \to \mathbb{R}$ satisfying

$$\nabla h(0) = R \nabla f(x),$$

and then obtain d by approximately solving

$$\min \; h(d) + \frac{\sigma}{2} \|d\|^2.$$

Step 2. Reconstruction.

Pick $T \in \mathbb{R}^{n \times N}$, and let

$$s = Td.$$

Step 3. Evaluation.

Compute

$$\rho = \frac{f(x) - f(x+s)}{h(0) - h(d)}.$$

If ρ is big enough, then $x := x + s$, $\sigma := \sigma/2$; else $\sigma := 2\sigma$. Go to **Step 1**.

Assumptions

Assumption (Objective function and models)

- *f is bounded from below and continuously differentiable; ∇f is Lipschitz continuous with a Lipschitz constant $L_f > 0$.*
- *For each $k \geq 0$, h_k is continuously differentiable; ∇h_k is Lipschitz continuous with a common Lipschitz constant $L_h > 0$.*

Assumptions

Assumption (Assumptions on R_k and T_k)

- $\{\|R_k\|\}$ *and* $\{\|T_k\|\}$ *are bounded.*
- $\{\|R_k g_k\|/\|g_k\|\}$ *and* $\{\|T_k^\top g_k\|/\|g_k\|\}$ *are bounded away from zero.*
- $\{g_k^\top T_k R_k g_k/\|T_k^\top g_k\|\|R_k g_k\|\}$ *is positive and bounded away from zero.*

- To satisfy the assumptions, we do not need to know $g_k \equiv \nabla f(x_k)$!
- Consider AS, RAS, ASH, and RASH . . .

Assumptions

Let

$$\alpha_k = \frac{h_k(0) - [h_k(d_k) + \sigma_k \|d_k\|^2/2]}{\|\nabla h_k\|^2/[2(\sigma_k + L_h)]}.$$

Assumption (Assumption on subproblem solutions)

There exist constants $\zeta > 1$, $\alpha > 0$, $\sigma > 0$ such that

$$\alpha_k \geq \max\left\{\zeta \sin^2 \varphi_k, \;\; \alpha\right\}$$

whenever $\sigma_k \geq \sigma$, where φ_k is the angle between $R_k g_k$ and $T_k^\top g_k$.

Convergence

Theorem

Under the assumptions on f, h_k, R_k, T_k, and d_k, the sequence $\{x_k\}$ generated by the framework satisfies

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0 \quad \textit{and} \quad \min_{0\le\ell\le k} \|\nabla f(x_\ell)\| = \mathcal{O}(1/\sqrt{k}).$$

Convergence

Theorem

Under the assumptions on f, h_k, R_k, T_k, and d_k, the sequence $\{x_k\}$ generated by the framework satisfies

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0 \quad \textit{and} \quad \min_{0\le\ell\le k} \|\nabla f(x_\ell)\| = \mathcal{O}(1/\sqrt{k}).$$

Assuming convexity or strong convexity of f, we have

$$\min_{0\le\ell\le k} \|\nabla f(x_\ell)\| = \mathcal{O}(1/k) \quad \textit{or} \quad \min_{0\le\ell\le k} \|\nabla f(x_\ell)\| = \mathcal{O}(\exp(-ck))$$

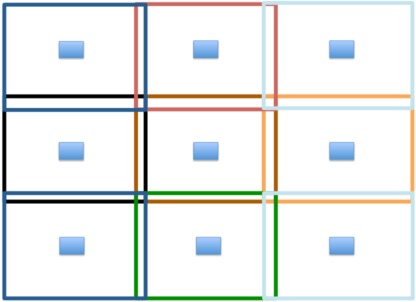
respectively.

2D Elliptic problem — quadratic

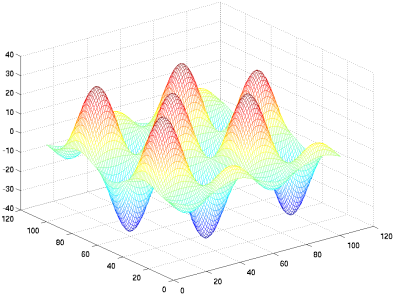
- Variational approach for $-\Delta u = f$ with zero boundary condition:

$$\min_v \frac{1}{2} a(v, v) - b(v)$$

- Finite differences are used on a cartesian grid
- The domain is split into $n_x \times n_y$ subdomains that may overlap
- A coarse grid correction may be used in addition



2D Elliptic problem — quadratic



$n_x \times n_y$	no overlap	overlap	overlap & coarse grid
2×2	223	80	55
3×3	304	107	65
4×4	375	128	64
5×5	445	151	66

1D Minimal surface problem — solved as a DFO

Observasion

The framework is applicable to optimization without derivatives:

Define $h_k^{(i)} = f$, *and solve the subproblems by DFO solvers.*

1D Minimal surface problem — solved as a DFO

Observasion

The framework is applicable to optimization without derivatives:

Define $h_k^{(i)} = f$*, and solve the subproblems by DFO solvers.*

Minimal surface problem (1-D):

$$\min_x \int_0^1 \left[\left(\frac{\mathrm{d}x}{\mathrm{d}t} \right)^2 + 1 \right]^{\frac{1}{2}} \mathrm{d}t + x^2(1/2).$$

Discretization:

$$\min_x \sum_{i=1}^n \left[(x_i - x_{i-1})^2 + \frac{1}{n^2} \right]^{\frac{1}{2}} + x_{n/2}^2.$$

1D Minimal surface problem — solved as a DFO

- Subspaces

$$\mathcal{S}_k^{(1)} = \text{span}\left\{e_0,\ \ldots,\ e_{(n+n_o)/2}\right\},$$
$$\mathcal{S}_k^{(2)} = \text{span}\left\{e_{(n-n_o)/2},\ \ldots,\ e_n\right\}.$$

  Test $n_o = 0$, $0.1n$, or $0.2n$.

- Subproblems are solved by NEWUOA (Powell, 2004).

- Synchronization strategy: RAS

1D Minimal surface problem — solved as a DFO

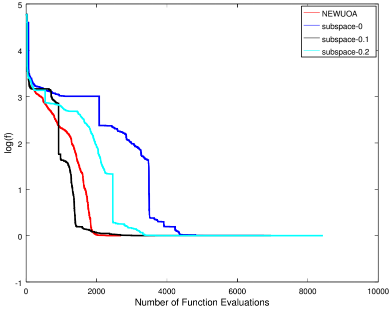


Figure: n = 20

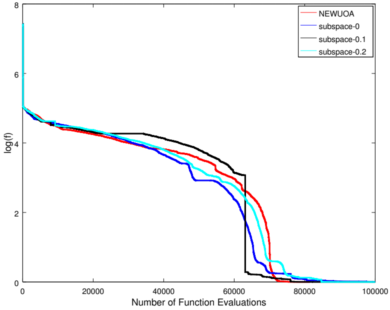


Figure: n = 80

1D Minimal surface problem — solved as a DFO

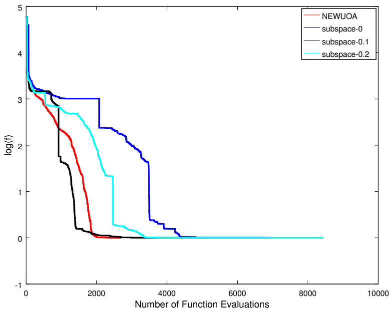


Figure: n = 20

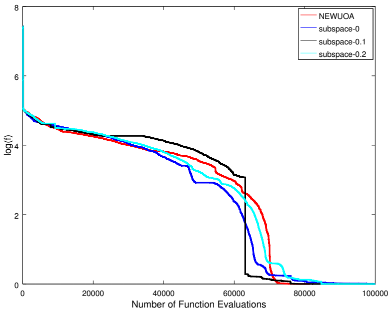


Figure: n = 80

Is this a fair play?

Concluding remarks

- A space decomposition framework that
  - extends classical techniques in Domain Decomposition to NLP
  - globally converges with guaranteed convergence rate
  - enables us to develop parallel methods with/without derivatives

Concluding remarks

- A space decomposition framework that
  - extends classical techniques in Domain Decomposition to NLP
  - globally converges with guaranteed convergence rate
  - enables us to develop parallel methods with/without derivatives
- Still much to do:
  - other globalization techniques: trust region, cubic regularization
  - more subspaces
    - take advantage of massively parallel computation
    - notice the cheap synchronization
  - choose subspaces according to problem structure, if known
  - coarse space technique

Concluding remarks

- A space decomposition framework that
  - extends classical techniques in Domain Decomposition to NLP
  - globally converges with guaranteed convergence rate
  - enables us to develop parallel methods with/without derivatives
- Still much to do:
  - other globalization techniques: trust region, cubic regularization
  - more subspaces
    - take advantage of massively parallel computation
    - notice the cheap synchronization
  - choose subspaces according to problem structure, if known
  - coarse space technique