Thèse de Doctorat

Asynchronous domain decomposition methods for massively parallel computing

Préparée et soutenue par

Guillaume Gbikpi-Benissan

Devant le Jury composé de

Raphaël Couturier	(Rapporteur)
Fabienne Jezequel	(Examinatrice)
Pierre Spiteri	(Rapporteur)
Stéphane Vialle	(Examinateur)
Frédéric Magoulès	(Directeur de thèse)

18 décembre 2017

Introduction

Asynchronous iterative methods

“Possibly the kind of methods which will allow the next generation of parallel machines to attain the expected potential.” *[Frommer and Szyld, 2000]*

INTRODUCTION

Asynchronous iterative methods

“Possibly the kind of methods which will allow the next generation of parallel machines to attain the expected potential.” *[Frommer and Szyld, 2000]*

Parallel processing

- Speedup limit *[Amdahl, 1967]*

$t(p)$: processing time using p processors
α: serial proportion of the processing (data management)

Introduction

Asynchronous iterative methods

“Possibly the kind of methods which will allow the next generation of parallel machines to attain the expected potential.” *[Frommer and Szyld, 2000]*

Parallel processing

- Speedup limit *[Amdahl, 1967]*

$t(p)$: processing time using p processors
α: serial proportion of the processing (data management)
$\Rightarrow$ Theoretical speedup:

$$s(p,\alpha) = \frac{t(1)}{t(p)} \leq \frac{1}{\alpha + \frac{1-\alpha}{p}}$$

Introduction

Asynchronous iterative methods

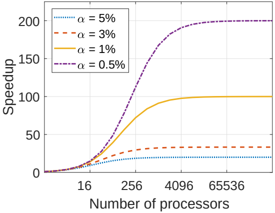
“Possibly the kind of methods which will allow the next generation of parallel machines to attain the expected potential.” *[Frommer and Szyld, 2000]*

Parallel processing

- Speedup limit *[Amdahl, 1967]*

$t(p)$: processing time using p processors
α: serial proportion of the processing (data management)
$\Rightarrow$ Theoretical speedup:

$$s(p,\alpha) = \frac{t(1)}{t(p)} \leq \frac{1}{\alpha + \frac{1-\alpha}{p}}$$



Introduction

Asynchronous iterative methods

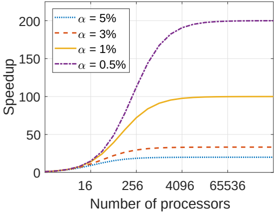
“Possibly the kind of methods which will allow the next generation of parallel machines to attain the expected potential.” *[Frommer and Szyld, 2000]*

Parallel processing

- Speedup limit *[Amdahl, 1967]*

$t(p)$: processing time using p processors
α: serial proportion of the processing (data management)
$\Rightarrow$ Theoretical speedup:

$$s(p,\alpha) = \frac{t(1)}{t(p)} \leq \frac{1}{\alpha + \frac{1-\alpha}{p}}$$



Minimize α: input/output, pre/post-processing, ..., and *inter-process communication*

INTRODUCTION

Asynchronous iterative methods

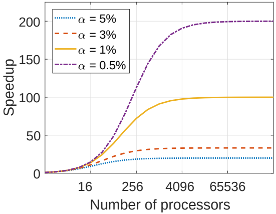
"Possibly the kind of methods which will allow the next generation of parallel machines to attain the expected potential." *[Frommer and Szyld, 2000]*

Parallel processing

- Speedup limit *[Amdahl, 1967]*

$t(p)$: processing time using p processors
α: serial proportion of the processing (data management)
$\Rightarrow$ Theoretical speedup:

$$s(p,\alpha) = \frac{t(1)}{t(p)} \leq \frac{1}{\alpha + \frac{1-\alpha}{p}}$$



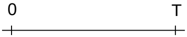
Minimize α: input/output, pre/post-processing, ..., and *inter-process communication*

- But also: load balancing and *fault-tolerance*

Introduction

Parallel processing

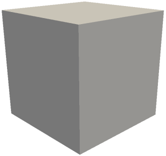
- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance



- Numerical simulation

Partial differential equations

$$\delta(u(s,t),s,t)=0, \quad t\in\mathbb{R}^+, \ \ s\in\Omega\subset\mathbb{R}^3$$



Introduction

Parallel processing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance

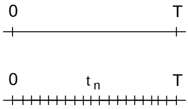
- Numerical simulation

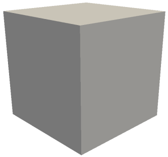
Partial differential equations

$$\delta(u(s,t),s,t) = 0, \quad t \in \mathbb{R}^+, \;\; s \in \Omega \subset \mathbb{R}^3$$

Time discretization

$$\alpha(u(s,t_{n+1})) = \beta(u(s,t_n))$$





Introduction

Parallel processing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance

- Numerical simulation

Partial differential equations

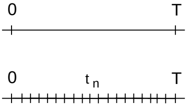
$$\delta(u(s,t),s,t) = 0, \quad t \in \mathbb{R}^+, \ \ s \in \Omega \subset \mathbb{R}^3$$

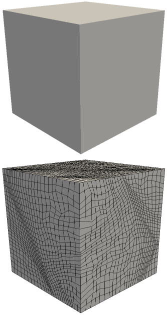
Time discretization

$$\alpha(u(s,t_{n+1})) = \beta(u(s,t_n))$$

Space discretization

$$AU_{n+1} = B_n, \quad U_{n+1} \in \mathbb{C}^m$$

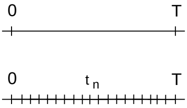




INTRODUCTION

Parallel processing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance



- Numerical simulation

Partial differential equations

$$\delta(u(s,t),s,t) = 0, \quad t \in \mathbb{R}^+, \;\; s \in \Omega \subset \mathbb{R}^3$$

Time discretization

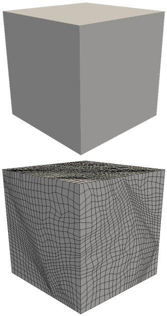
$$\alpha(u(s,t_{n+1})) = \beta(u(s,t_n))$$

Space discretization

$$AU_{n+1} = B_n, \quad U_{n+1} \in \mathbb{C}^m$$

Sparse system of equations

$$Ax = b, \quad x \in \mathbb{C}^m$$



Introduction

Parallel processing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance

- Numerical simulation

Partial differential equations

$$\delta(u(s,t),s,t)=0, \quad t\in\mathbb{R}^+, \ \ s\in\Omega\subset\mathbb{R}^3$$

Time discretization

$$\alpha(u(s,t_{n+1}))=\beta(u(s,t_n))$$

Space discretization

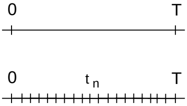
$$AU_{n+1}=B_n, \quad U_{n+1}\in\mathbb{C}^m$$

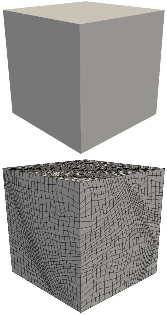
Sparse system of equations

$$Ax=b, \quad x\in\mathbb{C}^m$$

Iterative methods

$$x^{k+1}=f(x^k)$$

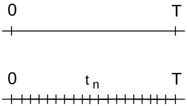




Introduction

Parallel processing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance



- Numerical simulation

Partial differential equations

$$\delta(u(s,t),s,t) = 0, \quad t \in \mathbb{R}^+, \; s \in \Omega \subset \mathbb{R}^3$$

Time discretization

$$\alpha(u(s,t_{n+1})) = \beta(u(s,t_n))$$

Space discretization

$$AU_{n+1} = B_n, \quad U_{n+1} \in \mathbb{C}^m$$

Sparse system of equations

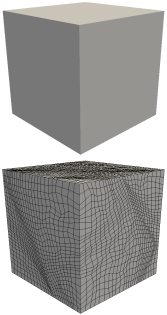
$$Ax = b, \quad x \in \mathbb{C}^m$$

Iterative methods

$$x^{k+1} = f(x^k)$$

$$x_i^{k+1} = f_i(x^k), \; \forall i \in \{1,\ldots,p\}$$

$$x := \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}, \quad p \le m$$



Parallel computing

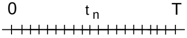
Introduction

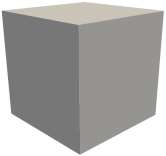
Parallel computing

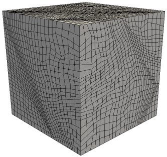
- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation

$$\delta(u(s,t),s,t) = 0, \quad t \in [0,T], \;\; s \in \Omega$$

- Domain decomposition methods







Introduction

Parallel computing

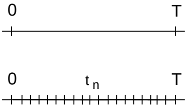
- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation

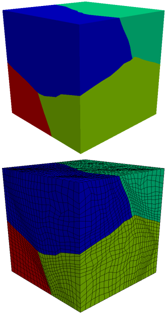
$$\delta(u(s,t),s,t)=0,\quad t\in[0,T],\ \ s\in\Omega$$

- Domain decomposition methods

Different subdomains

$$\Omega^{(1)},\dots,\Omega^{(p)}$$





Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation

$$\delta(u(s,t),s,t) = 0, \quad t \in [0,T], \;\; s \in \Omega$$

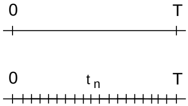
- Domain decomposition methods

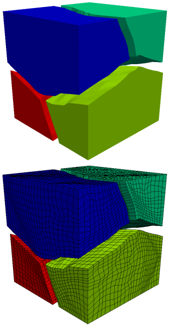
Different subdomains

$$\Omega^{(1)}, \ldots, \Omega^{(p)}$$

Parallel solutions

$$u^{(1)}(s,t_{n+1}), \ldots, u^{(p)}(s,t_{n+1})$$





Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation

$$\delta(u(s,t),s,t)=0, \quad t\in[0,T], \;\; s\in\Omega$$

- Domain decomposition methods

Different subdomains

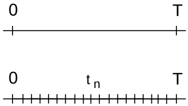
$$\Omega^{(1)},\ldots,\Omega^{(p)}$$

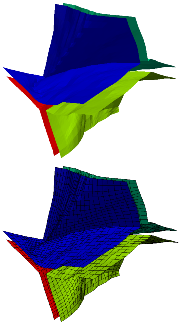
Parallel solutions

$$u^{(1)}(s,t_{n+1}),\ldots,u^{(p)}(s,t_{n+1})$$

Consistency across interfaces Γ^i_j, $i,j\in\{1,\ldots,p\}$

$$u^{(i)}(s_{\Gamma^i_j},t_{n+1})=u^{(j)}(s_{\Gamma^i_j},t_{n+1})$$





Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation

$\delta(u(s,t),s,t) = 0, \quad t \in [0,T], \ \ s \in \Omega$

- Domain decomposition methods

Different subdomains

$$\Omega^{(1)}, \dots, \Omega^{(p)}$$

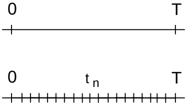
Parallel solutions

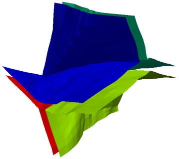
$$u^{(1)}(s,t_{n+1}), \dots, u^{(p)}(s,t_{n+1})$$

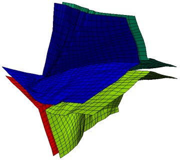
Consistency across interfaces Γ_j^i, $i,j \in \{1,\dots,p\}$

$$u^{(i)}(s_{\Gamma_j^i}, t_{n+1}) = u^{(j)}(s_{\Gamma_j^i}, t_{n+1})$$

+ parallel input/output
+ parallel pre/post-processing







Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation

$$\delta(u(s,t),s,t) = 0, \quad t \in [0,T], \;\; s \in \Omega$$

- Domain decomposition methods

Different subdomains

$$\Omega^{(1)}, \ldots, \Omega^{(p)}$$

Parallel solutions

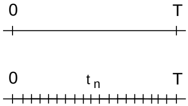
$$u^{(1)}(s,t_{n+1}), \ldots, u^{(p)}(s,t_{n+1})$$

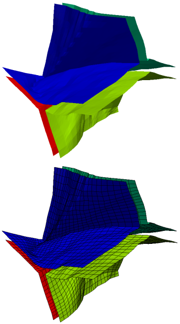
Consistency across interfaces Γ_j^i, $i,j \in \{1,\ldots,p\}$

$$u^{(i)}(s_{\Gamma_j^i}, t_{n+1}) = u^{(j)}(s_{\Gamma_j^i}, t_{n+1})$$

+ parallel input/output
+ parallel pre/post-processing

+ static load balancing





Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation

$$\delta(u(s,t),s,t)=0,\quad t\in[0,T],\;\; s\in\Omega$$

- Domain decomposition methods

Different subdomains

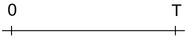
$$\Omega^{(1)},\ldots,\Omega^{(p)}$$

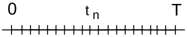
Parallel solutions

$$u^{(1)}(s,t_{n+1}),\ldots,u^{(p)}(s,t_{n+1})$$

Consistency across interfaces Γ_j^i, $i,j\in\{1,\ldots,p\}$

$$u^{(i)}(s_{\Gamma_j^i},t_{n+1})=u^{(j)}(s_{\Gamma_j^i},t_{n+1})$$

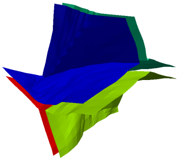


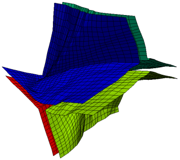


+ parallel input/output
+ parallel pre/post-processing

+ static load balancing

– serial parallel iterations
⇒ blocking inter-process synchronization





Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation

$$\delta(u(s,t),s,t)=0, \quad t\in[0,T], \; s\in\Omega$$

- Domain decomposition methods

Different subdomains

$$\Omega^{(1)},\dots,\Omega^{(p)}$$

Parallel solutions

$$u^{(1)}(s,t_{n+1}),\dots,u^{(p)}(s,t_{n+1})$$

Consistency across interfaces Γ_j^i,

$i,j\in\{1,\dots,p\}$

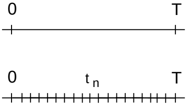
$$u^{(i)}(s_{\Gamma_j^i},t_{n+1})=u^{(j)}(s_{\Gamma_j^i},t_{n+1})$$

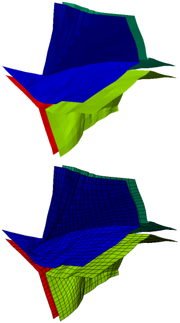
+ parallel input/output
+ parallel pre/post-processing

+ static load balancing

– serial parallel iterations
⇒ blocking inter-process synchronization

⇒ – non fault-tolerant



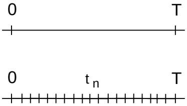


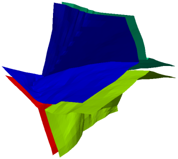
Introduction

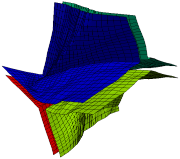
Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation
- Domain decomposition methods
- Asynchronous iterations *[Chazan and Miranker, 1969]*

+ static load balancing
– serial parallel iterations
– non fault-tolerant



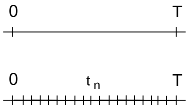




Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation
- Domain decomposition methods



+ static load balancing

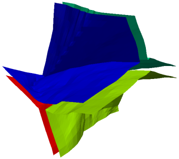
– serial parallel iterations

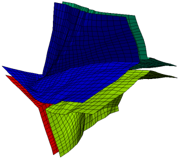
– non fault-tolerant

- Asynchronous iterations *[Chazan and Miranker, 1969]*

+ parallel serial iterations

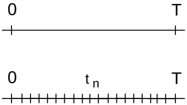
⇒ no speedup limit





Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation
- Domain decomposition methods

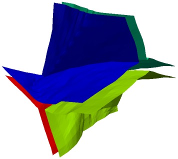


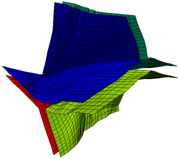
+ static load balancing
– serial parallel iterations
– non fault-tolerant

- Asynchronous iterations *[Chazan and Miranker, 1969]*

+ parallel serial iterations
⇒ no speedup limit

+ non-deterministic inter-process communication ⇒ fault-tolerance





Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation
- Domain decomposition methods

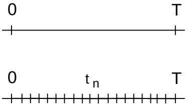
+ static load balancing
– serial parallel iterations
– non fault-tolerant

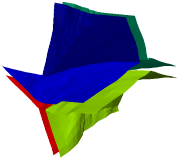
- Asynchronous iterations *[Chazan and Miranker, 1969]*

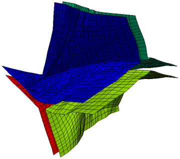
+ parallel serial iterations
⇒ no speedup limit

+ non-deterministic inter-process communication ⇒ fault-tolerance

+ dynamic adaptation to unbalanced load



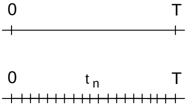




Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation
- Domain decomposition methods



+ static load balancing
– serial parallel iterations
– non fault-tolerant

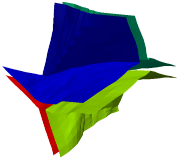
- Asynchronous iterations *[Chazan and Miranker, 1969]*

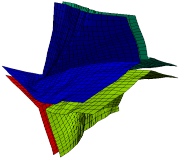
+ parallel serial iterations
⇒ no speedup limit

+ non-deterministic inter-process communication ⇒ fault-tolerance

+ dynamic adaptation to unbalanced load

– low convergence rate





SystemX CentraleSupélec

Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation
- Domain decomposition methods

+ static load balancing
– serial parallel iterations
– non fault-tolerant

- Asynchronous iterations *[Chazan and Miranker, 1969]*

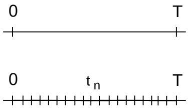
+ parallel serial iterations
⇒ no speedup limit

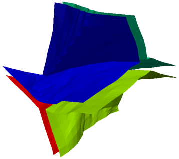
+ non-deterministic inter-process communication ⇒ fault-tolerance

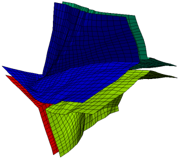
+ dynamic adaptation to unbalanced load

– low convergence rate

Eventual consistency across interfaces?
(convergence conditions)



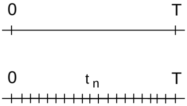




Introduction

Parallel computing

- Speedup limit *[Amdahl, 1967]*
- But also: load balancing and fault-tolerance
- Numerical simulation
- Domain decomposition methods



+ static load balancing
– serial parallel iterations
– non fault-tolerant

- Asynchronous iterations *[Chazan and Miranker, 1969]*

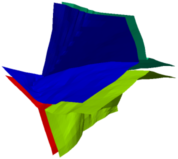
+ parallel serial iterations
⇒ no speedup limit

+ non-deterministic inter-process communication ⇒ fault-tolerance

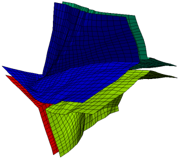
+ dynamic adaptation to unbalanced load

– low convergence rate

Eventual consistency across interfaces?
(convergence conditions)



Consistency reached?
(convergence detection)



Outline of the presentation

01 Asynchronous iterative methods

02 Space domain decomposition

03 Time domain decomposition

04 Asynchronous convergence detection

05 Conclusion

Outline

01 Asynchronous iterative methods

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods ⇒ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods ⇒ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

Convergence condition (sufficient and necessary)

$$\rho(M^{-1}N) < 1$$

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods ⇒ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

Convergence condition (sufficient and necessary)

$$\rho(M^{-1}N) < 1$$

Parallel computing with p processors, $p \leq n$

$$f(x) = \begin{bmatrix} f_1(x) & \cdots & f_p(x) \end{bmatrix}^\mathsf{T}$$

$$x = \begin{bmatrix} x_1 & \cdots & x_p \end{bmatrix}^\mathsf{T}$$

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods $\Rightarrow$ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

Convergence condition (sufficient and necessary)

$$\rho(M^{-1}N) < 1$$

Parallel computing with p processors, $p \leq n$

$$f(x) = \begin{bmatrix} f_1(x) & \cdots & f_p(x) \end{bmatrix}^\mathsf{T}$$

$$x = \begin{bmatrix} x_1 & \cdots & x_p \end{bmatrix}^\mathsf{T}$$

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods ⇒ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

Convergence condition (sufficient and necessary)

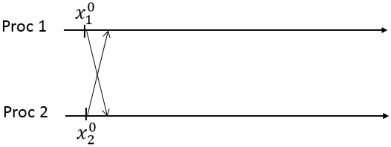
$$\rho(M^{-1}N) < 1$$

Parallel computing with p processors, $p \leq n$

$$f(x) = \begin{bmatrix} f_1(x) & \cdots & f_p(x) \end{bmatrix}^\mathsf{T}$$

$$x = \begin{bmatrix} x_1 & \cdots & x_p \end{bmatrix}^\mathsf{T}$$

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$



1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods $\Rightarrow$ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

Convergence condition (sufficient and necessary)

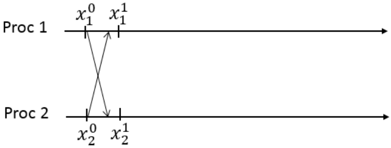
$$\rho(M^{-1}N) < 1$$

Parallel computing with p processors, $p \leq n$

$$f(x) = \begin{bmatrix} f_1(x) & \cdots & f_p(x) \end{bmatrix}^{\mathsf{T}}$$

$$x = \begin{bmatrix} x_1 & \cdots & x_p \end{bmatrix}^{\mathsf{T}}$$

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$



$$x_1^1 := f_1(x_1^0, x_2^0) \qquad x_2^1 := f_2(x_1^0, x_2^0)$$

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods $\Rightarrow$ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

Convergence condition (sufficient and necessary)

$$\rho(M^{-1}N) < 1$$

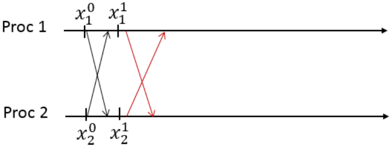
Parallel computing with p processors, $p \leq n$

$$f(x) = \begin{bmatrix} f_1(x) & \cdots & f_p(x) \end{bmatrix}^\mathsf{T}$$

$$x = \begin{bmatrix} x_1 & \cdots & x_p \end{bmatrix}^\mathsf{T}$$

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

speedup limit



$$x_1^1 := f_1(x_1^0, x_2^0) \qquad x_2^1 := f_2(x_1^0, x_2^0)$$

wait wait

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods ⇒ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

Convergence condition (sufficient and necessary)

$$\rho(M^{-1}N) < 1$$

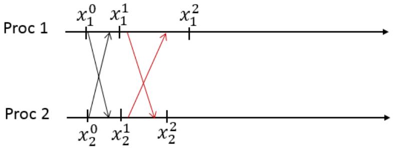
Parallel computing with p processors, $p \leq n$

$$f(x) = \begin{bmatrix} f_1(x) & \cdots & f_p(x) \end{bmatrix}^{\mathsf{T}}$$

$$x = \begin{bmatrix} x_1 & \cdots & x_p \end{bmatrix}^{\mathsf{T}}$$

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

speedup limit



$$x_1^1 := f_1(x_1^0, x_2^0) \qquad x_2^1 := f_2(x_1^0, x_2^0)$$

wait wait

$$x_1^2 := f_1(x_1^1, x_2^1) \qquad x_2^2 := f_2(x_1^1, x_2^1)$$

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods $\Rightarrow$ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

Convergence condition (sufficient and necessary)

$$\rho(M^{-1}N) < 1$$

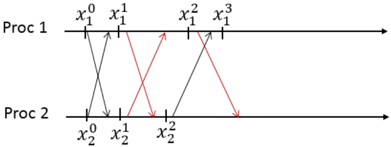
Parallel computing with p processors, $p \leq n$

$$f(x) = \begin{bmatrix} f_1(x) & \cdots & f_p(x) \end{bmatrix}^\mathsf{T}$$

$$x = \begin{bmatrix} x_1 & \cdots & x_p \end{bmatrix}^\mathsf{T}$$

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

speedup limit



$x_1^1 := f_1(x_1^0, x_2^0)$ $\quad$ $x_2^1 := f_2(x_1^0, x_2^0)$

wait $\quad$ wait

$x_1^2 := f_1(x_1^1, x_2^1)$ $\quad$ $x_2^2 := f_2(x_1^1, x_2^1)$

$x_1^3 := f_1(x_1^2, x_2^2)$ $\quad$ wait

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods $\Rightarrow$ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

Convergence condition (sufficient and necessary)

$$\rho(M^{-1}N) < 1$$

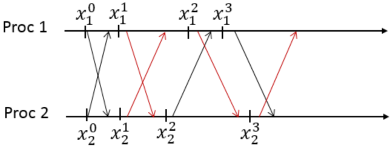
Parallel computing with p processors, $p \leq n$

$$f(x) = \begin{bmatrix} f_1(x) & \cdots & f_p(x) \end{bmatrix}^\mathsf{T}$$

$$x = \begin{bmatrix} x_1 & \cdots & x_p \end{bmatrix}^\mathsf{T}$$

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

speedup limit



$x_1^1 := f_1(x_1^0, x_2^0)$	$x_2^1 := f_2(x_1^0, x_2^0)$
wait	wait
$x_1^2 := f_1(x_1^1, x_2^1)$	$x_2^2 := f_2(x_1^1, x_2^1)$
$x_1^3 := f_1(x_1^2, x_2^2)$	wait
wait	$x_2^3 := f_2(x_1^2, x_2^2)$

1 - Asynchronous iterative methods

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Splitting

$$A = M - N$$

Mapping

$$f(x) := M^{-1}Nx + M^{-1}b$$

Fixed-point problem

$$Ax = b \iff x = f(x)$$

Iterative methods $\Rightarrow$ sequence $\{x^k\}_{k\in\mathbb{N}}$:

$$x^{k+1} = f(x^k)$$

Convergence from any initial vector x^0

$$\lim_{k\to\infty} x^k = x^*, \quad f(x^*) = x^*$$

Convergence condition (sufficient and necessary)

$$\rho(M^{-1}N) < 1$$

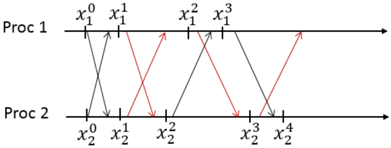
Parallel computing with p processors, $p \leq n$

$$f(x) = \begin{bmatrix} f_1(x) & \cdots & f_p(x) \end{bmatrix}^\mathsf{T}$$

$$x = \begin{bmatrix} x_1 & \cdots & x_p \end{bmatrix}^\mathsf{T}$$

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

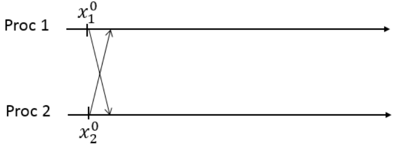
speedup limit


x_1^0 x_1^1 x_1^2 x_1^3
Proc 1
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4

$x_1^1 := f_1(x_1^0, x_2^0)$	$x_2^1 := f_2(x_1^0, x_2^0)$
wait	wait
$x_1^2 := f_1(x_1^1, x_2^1)$	$x_2^2 := f_2(x_1^1, x_2^1)$
$x_1^3 := f_1(x_1^2, x_2^2)$	wait
wait	$x_2^3 := f_2(x_1^2, x_2^2)$
wait	$x_2^4 := f_2(x_1^3, x_2^3)$

1 - Asynchronous iterative methods

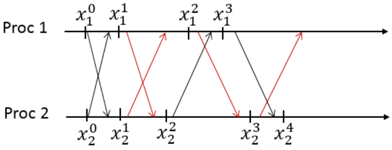
Asynchronous iterations



Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

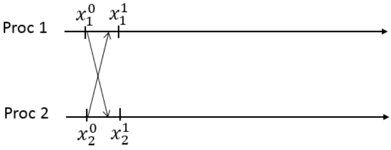
delay ⇒ speedup limit


x_1^0 x_1^1 x_1^2 x_1^3
Proc 1
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4

$x_1^1 := f_1(x_1^0, x_2^0)$	$x_2^1 := f_2(x_1^0, x_2^0)$
wait	wait
$x_1^2 := f_1(x_1^1, x_2^1)$	$x_2^2 := f_2(x_1^1, x_2^1)$
$x_1^3 := f_1(x_1^2, x_2^2)$	wait
wait	$x_2^3 := f_2(x_1^2, x_2^2)$
wait	$x_2^4 := f_2(x_1^3, x_2^3)$

1 - Asynchronous iterative methods

Asynchronous iterations

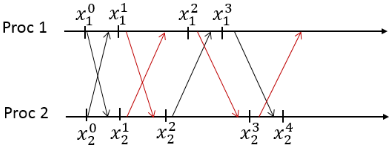


$x_1^1 := f_1(x_1^0, x_2^0) \quad x_2^1 := f_2(x_1^0, x_2^0)$

Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \dots, x_p^k), \quad \forall i \in \{1, \dots, p\}$$

delay ⇒ speedup limit

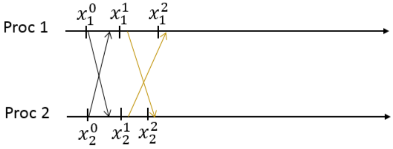


$x_1^1 := f_1(x_1^0, x_2^0)$	$x_2^1 := f_2(x_1^0, x_2^0)$
wait	wait
$x_1^2 := f_1(x_1^1, x_2^1)$	$x_2^2 := f_2(x_1^1, x_2^1)$
$x_1^3 := f_1(x_1^2, x_2^2)$	wait
wait	$x_2^3 := f_2(x_1^2, x_2^2)$
wait	$x_2^4 := f_2(x_1^3, x_2^3)$

1 - Asynchronous iterative methods

Asynchronous iterations

delay ⇒ low convergence rate

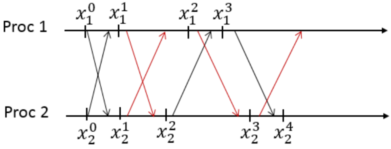


$$\begin{array}{ll} x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 := f_2(x_1^0, x_2^0) \\ x_1^2 := f_1(x_1^1, x_2^0) & x_2^2 := f_2(x_1^0, x_2^1) \end{array}$$

Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

delay ⇒ speedup limit

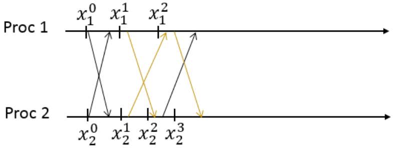


$$\begin{array}{cc} x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 := f_2(x_1^0, x_2^0) \\ \text{wait} & \text{wait} \\ x_1^2 := f_1(x_1^1, x_2^1) & x_2^2 := f_2(x_1^1, x_2^1) \\ x_1^3 := f_1(x_1^2, x_2^2) & \text{wait} \\ \text{wait} & x_2^3 := f_2(x_1^2, x_2^2) \\ \text{wait} & x_2^4 := f_2(x_1^3, x_2^3) \end{array}$$

1 - Asynchronous iterative methods

Asynchronous iterations

delay ⇒ low convergence rate



$$x_1^1 := f_1(x_1^0, x_2^0) \qquad x_2^1 := f_2(x_1^0, x_2^0)$$

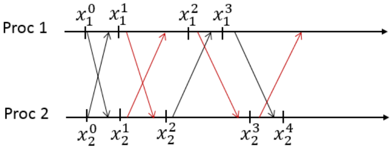
$$x_1^2 := f_1(x_1^1, x_2^0) \qquad x_2^2 := f_2(x_1^0, x_2^1)$$

$$x_2^3 := f_2(x_1^1, x_2^2)$$

Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

delay ⇒ speedup limit



$$x_1^1 := f_1(x_1^0, x_2^0) \qquad x_2^1 := f_2(x_1^0, x_2^0)$$

wait wait

$$x_1^2 := f_1(x_1^1, x_2^1) \qquad x_2^2 := f_2(x_1^1, x_2^1)$$

$$x_1^3 := f_1(x_1^2, x_2^2)$$

wait

wait

$$x_2^3 := f_2(x_1^2, x_2^2)$$

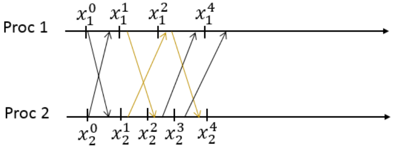
wait

$$x_2^4 := f_2(x_1^3, x_2^3)$$

1 - Asynchronous iterative methods

Asynchronous iterations

delay ⇒ low convergence rate

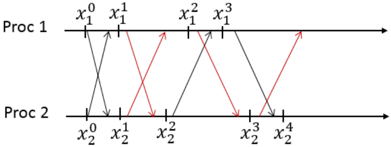

Proc 1
x_1^0 x_1^1 x_1^2 x_1^4
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4

$$\begin{array}{ll}
x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 := f_2(x_1^0, x_2^0) \\
x_1^2 := f_1(x_1^1, x_2^0) & x_2^2 := f_2(x_1^0, x_2^1) \\
x_1^3 := x_1^2 & x_2^3 := f_2(x_1^1, x_2^2) \\
x_1^4 := f_1(x_1^3, x_2^2) & x_2^4 := f_2(x_1^2, x_2^3)
\end{array}$$

Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

delay ⇒ speedup limit

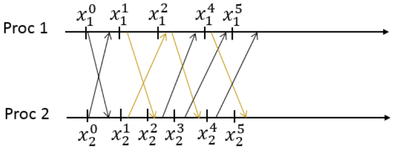

Proc 1
x_1^0 x_1^1 x_1^2 x_1^3
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4

$$\begin{array}{ll}
x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 := f_2(x_1^0, x_2^0) \\
\text{wait} & \text{wait} \\
x_1^2 := f_1(x_1^1, x_2^1) & x_2^2 := f_2(x_1^1, x_2^1) \\
x_1^3 := f_1(x_1^2, x_2^2) & \text{wait} \\
\text{wait} & x_2^3 := f_2(x_1^2, x_2^2) \\
\text{wait} & x_2^4 := f_2(x_1^3, x_2^3)
\end{array}$$

1 - Asynchronous iterative methods

Asynchronous iterations

delay ⇒ low convergence rate

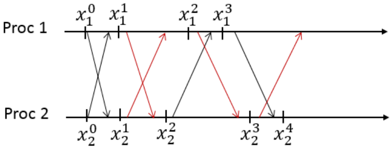


$$\begin{array}{ll}
x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 := f_2(x_1^0, x_2^0) \\
x_1^2 := f_1(x_1^1, x_2^0) & x_2^2 := f_2(x_1^0, x_2^1) \\
x_1^3 := x_1^2 & x_2^3 := f_2(x_1^1, x_2^2) \\
x_1^4 := f_1(x_1^3, x_2^2) & x_2^4 := f_2(x_1^2, x_2^3) \\
x_1^5 := f_1(x_1^4, x_2^3) & x_2^5 := f_2(x_1^2, x_2^4)
\end{array}$$

Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \dots, x_p^k), \quad \forall i \in \{1, \dots, p\}$$

delay ⇒ speedup limit



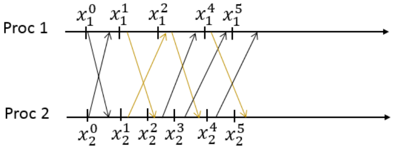
$$\begin{array}{cc}
x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 := f_2(x_1^0, x_2^0) \\
\text{wait} & \text{wait} \\
x_1^2 := f_1(x_1^1, x_2^1) & x_2^2 := f_2(x_1^1, x_2^1) \\
x_1^3 := f_1(x_1^2, x_2^2) & \text{wait} \\
\text{wait} & x_2^3 := f_2(x_1^2, x_2^2) \\
\text{wait} & x_2^4 := f_2(x_1^3, x_2^3)
\end{array}$$

1 - Asynchronous iterative methods

Asynchronous iterations

$$x_i^{k+1} = f_i(x_1^{\tau_1^i(k)}, \dots, x_p^{\tau_p^i(k)}), \quad \forall i \in P^k$$
$$x_i^{k+1} = x_i^k, \quad \forall i \notin P^k$$

delay ⇒ low convergence rate


Proc 1
Proc 2
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

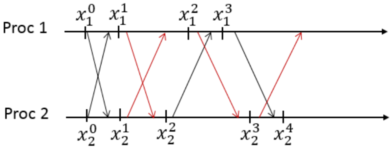
$$\begin{array}{ll}
x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 := f_2(x_1^0, x_2^0) \\
x_1^2 := f_1(x_1^1, x_2^0) & x_2^2 := f_2(x_1^0, x_2^1) \\
x_1^3 := x_1^2 & x_2^3 := f_2(x_1^1, x_2^2) \\
x_1^4 := f_1(x_1^3, x_2^2) & x_2^4 := f_2(x_1^2, x_2^3) \\
x_1^5 := f_1(x_1^4, x_2^3) & x_2^5 := f_2(x_1^2, x_2^4)
\end{array}$$

$$P^k \subset \{1, \dots, p\}, \qquad \tau_j^i(k) \leq k$$

Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \dots, x_p^k), \quad \forall i \in \{1, \dots, p\}$$

delay ⇒ speedup limit


Proc 1
Proc 2
x_1^0 x_1^1 x_1^2 x_1^3
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4

$$\begin{array}{cc}
x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 := f_2(x_1^0, x_2^0) \\
\text{wait} & \text{wait} \\
x_1^2 := f_1(x_1^1, x_2^1) & x_2^2 := f_2(x_1^1, x_2^1) \\
x_1^3 := f_1(x_1^2, x_2^2) & \text{wait} \\
\text{wait} & x_2^3 := f_2(x_1^2, x_2^2) \\
\text{wait} & x_2^4 := f_2(x_1^3, x_2^3)
\end{array}$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

Asynchronous iterations

$$\begin{aligned} x_i^{k+1} &= f_i(x_1^{\tau_1^i(k)}, \dots, x_p^{\tau_p^i(k)}), & \forall i \in P^k \\ x_i^{k+1} &= x_i^k, & \forall i \notin P^k \end{aligned}$$

Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \dots, x_p^k), \quad \forall i \in \{1, \dots, p\}$$

Convergence condition (necessary and sufficient)

$$\rho(M^{-1}N) < 1$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

Asynchronous iterations

$$\begin{aligned} x_i^{k+1} &= f_i(x_1^{\tau_1^i(k)}, \dots, x_p^{\tau_p^i(k)}), & \forall i \in P^k \\ x_i^{k+1} &= x_i^k, & \forall i \notin P^k \end{aligned}$$

Convergence condition (necessary and sufficient)
[Chazan and Miranker, 1969]

$$\rho(|M^{-1}N|) < 1$$

Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \dots, x_p^k), \quad \forall i \in \{1, \dots, p\}$$

Convergence condition (necessary and sufficient)

$$\rho(M^{-1}N) < 1$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

Asynchronous iterations

$$\begin{aligned} x_i^{k+1} &= f_i(x_1^{\tau_1^i(k)}, \ldots, x_p^{\tau_p^i(k)}), & \forall i \in P^k \\ x_i^{k+1} &= x_i^k, & \forall i \notin P^k \end{aligned}$$

Convergence condition (necessary and sufficient)
[Chazan and Miranker, 1969]

$$\rho(M^{-1}N) \leq \quad \rho(|M^{-1}N|) < 1$$

Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

Convergence condition (necessary and sufficient)

$$\rho(M^{-1}N) < 1$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

Asynchronous iterations

$$\begin{aligned} x_i^{k+1} &= f_i(x_1^{\tau_1^i(k)}, \ldots, x_p^{\tau_p^i(k)}), && \forall i \in P^k \\ x_i^{k+1} &= x_i^k, && \forall i \notin P^k \end{aligned}$$

Convergence condition (necessary and sufficient)
[Chazan and Miranker, 1969]

$$\rho(M^{-1}N) \leq \quad \rho(|M^{-1}N|) < 1$$

Synchronous iterations

$$x_i^{k+1} = f_i(x_1^k, \ldots, x_p^k), \quad \forall i \in \{1, \ldots, p\}$$

Convergence condition (necessary and sufficient)

$$\rho(M^{-1}N) < 1$$

General fixed-point problems

$$f^{(k)}(x, x, \ldots, x) = x, \ \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \ m \in \mathbb{N}^*$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \ldots, x) = x, \ \ \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \ \ m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \ \ \forall k$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \dots, x) = x, \ \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \ m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \ \forall k$$

[Miellou, 1975] (sufficient)

$$|f(x) - f(y)| \leq T|x - y|$$

$$T \geq O, \ \rho(T) < 1, \ |x| = (|x_1|, \dots, |x_p|)$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \ldots, x) = x, \;\; \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \;\; m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \;\; \forall k$$

[Miellou, 1975] (sufficient)

$$|f(x) - f(y)| \leq T|x - y|$$

$$T \geq O, \;\; \rho(T) < 1, \;\; |x| = (|x_1|, \ldots, |x_p|)$$

[El Tarazi, 1982] (sufficient)

$$\|f(x) - f(y)\|_\infty^w \leq \alpha \|x - y\|_\infty^w$$

$$w > 0, \;\; \alpha < 1, \;\; \|x\|_\infty^w = \max_i |x_i|/w_i$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \dots, x) = x, \;\; \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \;\; m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \;\; \forall k$$

[Miellou, 1975] (sufficient)

$$|f(x) - f(y)| \leq T|x - y|$$

$$T \geq O, \;\; \rho(T) < 1, \;\; |x| = (|x_1|, \dots, |x_p|)$$

[El Tarazi, 1982] (sufficient)

$$\|f(x) - f(y)\|_\infty^w \leq \alpha\|x - y\|_\infty^w$$

$$w > 0, \;\; \alpha < 1, \;\; \|x\|_\infty^w = \max_i |x_i|/w_i$$

[Bertsekas, 1983] (sufficient)

$$f(S^{(t)}) \subset S^{(t+1)} \subset S^{(t)}$$

$$S^{(t)} = S_1^{(t)} \times \cdots \times S_p^{(t)}, \;\; \lim_{t \to \infty} S^{(t)} = \{x^*\}$$

1 - Asynchronous iterative methods

- Linear problems

$$Ax = b \iff M^{-1}Nx + M^{-1}b = x$$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \ldots, x) = x, \quad \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \quad m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \quad \forall k$$

$$m = 1$$

[Miellou, 1975] (sufficient)

$$|f(x) - f(y)| \leq T|x - y|$$

$$T \geq O, \quad \rho(T) < 1, \quad |x| = (|x_1|, \ldots, |x_p|)$$

[El Tarazi, 1982] (sufficient)

$$\|f(x) - f(y)\|_\infty^w \leq \alpha \|x - y\|_\infty^w$$

$$w > 0, \quad \alpha < 1, \quad \|x\|_\infty^w = \max_i |x_i|/w_i$$

[Frommer and Szyld, 1994] (sufficient)

$$\|f^{(k)}(x) - f^{(k)}(y)\|_\infty^w \leq \alpha \|x - y\|_\infty^w, \quad \forall k$$

$$w > 0, \quad \alpha < 1, \quad \|x\|_\infty^w = \max_i |x_i|/w_i$$

[Bertsekas, 1983] (sufficient)

$$f(S^{(t)}) \subset S^{(t+1)} \subset S^{(t)}$$

$$S^{(t)} = S_1^{(t)} \times \cdots \times S_p^{(t)}, \quad \lim_{t \to \infty} S^{(t)} = \{x^*\}$$

1 - Asynchronous iterative methods

- Linear problems

$$Ax = b \iff M^{-1}Nx + M^{-1}b = x$$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \dots, x) = x, \quad \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \quad m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \quad \forall k$$

[Miellou, 1975] (sufficient)

$$|f(x) - f(y)| \leq T|x - y|$$

$$T \geq O, \quad \rho(T) < 1, \quad |x| = (|x_1|, \dots, |x_p|)$$

[El Tarazi, 1982] (sufficient)

$$\|f(x) - f(y)\|_\infty^w \leq \alpha \|x - y\|_\infty^w$$

$$w > 0, \quad \alpha < 1, \quad \|x\|_\infty^w = \max_i |x_i|/w_i$$

[Bertsekas, 1983] (sufficient)

$$f(S^{(t)}) \subset S^{(t+1)} \subset S^{(t)}$$

$$S^{(t)} = S_1^{(t)} \times \cdots \times S_p^{(t)}, \quad \lim_{t \to \infty} S^{(t)} = \{x^*\}$$

$$m = 1$$

[Frommer and Szyld, 1994] (sufficient)

$$\|f^{(k)}(x) - f^{(k)}(y)\|_\infty^w \leq \alpha \|x - y\|_\infty^w, \quad \forall k$$

$$w > 0, \quad \alpha < 1, \quad \|x\|_\infty^w = \max_i |x_i|/w_i$$

[Frommer and Szyld, 2000] (sufficient)

$$f^{(k)}(S^{(t)}) \subset S^{(t+1)} \subset S^{(t)}, \quad \forall k$$

$$S^{(t)} = S_1^{(t)} \times \cdots \times S_p^{(t)}, \quad \lim_{t \to \infty} S^{(t)} = \{x^*\}$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \ldots, x) = x, \;\; \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \;\; m \in \mathbb{N}^*$$

$m = 1, \quad f^{(k)} \equiv f, \;\; \forall k$

$m \geq 1, \quad f^{(k)} \equiv f, \;\; \forall k$

[Miellou, 1975] (sufficient): $|.|$-contraction

[El Tarazi, 1982] (sufficient): $\|.\|_\infty^w$-contraction

[Bertsekas, 1983] (sufficient): $\{S^{(t)}\}$-contraction

$m = 1$

[Frommer & Szyld, 1994] (sufficient): $\|.\|_\infty^w$-contraction, $\forall k$

[Frommer & Szyld, 2000] (sufficient): $\{S^{(t)}\}$-contraction, $\forall k$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \ldots, x) = x, \quad \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \quad m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \quad \forall k$$

[Miellou, 1975] (sufficient): $|.|$-contraction

[El Tarazi, 1982] (sufficient): $\|.\|_\infty^w$-contraction

[Bertsekas, 1983] (sufficient): $\{S^{(t)}\}$-contraction

$$m \geq 1, \quad f^{(k)} \equiv f, \quad \forall k$$

$$X := (x^{(1)}, \ldots, x^{(m)}), \quad Y := (y^{(1)}, \ldots, y^{(m)})$$

[Baudet, 1978] (sufficient)

$$|f(X) - f(Y)| \leq T \max\{|x^{(1)} - y^{(1)}|, \ldots, |x^{(m)} - y^{(m)}|\}$$

$$T \geq O, \quad \rho(T) < 1, \quad (\max\{|x|, |y|\})_i = \max\{|x_i|, |y_i|\}$$

$$m = 1$$

[Frommer & Szyld, 1994] (sufficient): $\|.\|_\infty^w$-contraction, $\forall k$

[Frommer & Szyld, 2000] (sufficient): $\{S^{(t)}\}$-contraction, $\forall k$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \dots, x) = x, \quad \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \quad m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \quad \forall k$$

[Miellou, 1975] (sufficient): $|.|$-contraction

[El Tarazi, 1982] (sufficient): $\|.\|_\infty^w$-contraction

[Bertsekas, 1983] (sufficient): $\{S^{(t)}\}$-contraction

$$m = 1$$

[Frommer & Szyld, 1994] (sufficient): $\|.\|_\infty^w$-contraction, $\forall k$

[Frommer & Szyld, 2000] (sufficient): $\{S^{(t)}\}$-contraction, $\forall k$

$$m \geq 1, \quad f^{(k)} \equiv f, \quad \forall k$$

$$X := (x^{(1)}, \dots, x^{(m)}), \quad Y := (y^{(1)}, \dots, y^{(m)})$$

[Baudet, 1978] (sufficient)

$$|f(X) - f(Y)| \leq T \max\{|x^{(1)} - y^{(1)}|, \dots, |x^{(m)} - y^{(m)}|\}$$

$$T \geq O, \quad \rho(T) < 1, \quad (\max\{|x|, |y|\})_i = \max\{|x_i|, |y_i|\}$$

[El Tarazi, 1982] (sufficient)

$$\|f(X) - x^*\|_\infty^w \leq \alpha \max\{\|x^{(l)} - x^*\|_\infty^w\}_{1 \leq l \leq m}$$

$$w > 0, \quad \alpha < 1, \quad \|x\|_\infty^w = \max_i |x_i|/w_i$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \ldots, x) = x, \quad \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \quad m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \quad \forall k$$

[Miellou, 1975] (sufficient): $|.|$-contraction

[El Tarazi, 1982] (sufficient): $\|.\|_\infty^w$-contraction

[Bertsekas, 1983] (sufficient): $\{S^{(t)}\}$-contraction

$$m = 1$$

[Frommer & Szyld, 1994] (sufficient): $\|.\|_\infty^w$-contraction, $\forall k$

[Frommer & Szyld, 2000] (sufficient): $\{S^{(t)}\}$-contraction, $\forall k$

$$m \geq 1, \quad f^{(k)} \equiv f, \quad \forall k$$

$$X := (x^{(1)}, \ldots, x^{(m)}), \quad Y := (y^{(1)}, \ldots, y^{(m)})$$

[Baudet, 1978] (sufficient): $|.|$-contraction
[El Tarazi, 1982] (sufficient)

$$\|f(X) - x^*\|_\infty^w \leq \alpha \max\{\|x^{(l)} - x^*\|_\infty^w\}_{1 \leq l \leq m}$$

$$w > 0, \quad \alpha < 1, \quad \|x\|_\infty^w = \max_i |x_i|/w_i$$

[Frommer & Szyld, 1998] (sufficient)

$$\|f(x, y) - x^*\|_\infty^w \leq \alpha \max\{\|x - x^*\|_\infty^w, \|y - x^*\|_\infty^w\}$$

$$w > 0, \quad \alpha < 1, \quad \|x\|_\infty^w = \max_i |x_i|/w_i$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \dots, x) = x, \ \ \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \ \ m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \ \ \forall k$$

[Miellou, 1975] (sufficient): $|.|$-contraction

[El Tarazi, 1982] (sufficient): $\|.\|^w_\infty$-contraction

[Bertsekas, 1983] (sufficient): $\{S^{(t)}\}$-contraction

$$m = 1$$

[Frommer & Szyld, 1994] (sufficient): $\|.\|^w_\infty$-contraction, $\forall k$

[Frommer & Szyld, 2000] (sufficient): $\{S^{(t)}\}$-contraction, $\forall k$

$$m \geq 1, \quad f^{(k)} \equiv f, \ \ \forall k$$

$$X := (x^{(1)}, \dots, x^{(m)}), \ \ Y := (y^{(1)}, \dots, y^{(m)})$$

[Baudet, 1978] (sufficient): $|.|$-contraction
[El Tarazi, 1982] (sufficient)

$$\|f(X) - x^*\|^w_\infty \leq \alpha \max\{\|x^{(l)} - x^*\|^w_\infty\}_{1 \leq l \leq m}$$

$$w > 0, \ \ \alpha < 1, \ \ \|x\|^w_\infty = \max_i |x_i|/w_i$$

[Frommer & Szyld, 1998] (sufficient) $\quad m = 2$

$$\|f(x, y) - x^*\|^w_\infty \leq \alpha \max\{\|x - x^*\|^w_\infty, \|y - x^*\|^w_\infty\}$$

$$w > 0, \ \ \alpha < 1, \ \ \|x\|^w_\infty = \max_i |x_i|/w_i$$

1 - Asynchronous iterative methods

- Linear problems
  $Ax = b \iff M^{-1}Nx + M^{-1}b = x$

[Chazan and Miranker, 1969] (necessary and sufficient): $\rho(|M^{-1}N|) < 1$

- General fixed-point problems

$$f^{(k)}(x, x, \dots, x) = x, \quad \forall k \in \mathbb{N}, \quad f^{(k)} : E^m \mapsto E, \quad m \in \mathbb{N}^*$$

$$m = 1, \quad f^{(k)} \equiv f, \quad \forall k$$

[Miellou, 1975] (sufficient): $|.|$-contraction

[El Tarazi, 1982] (sufficient): $\|.\|_\infty^w$-contraction

[Bertsekas, 1983] (sufficient): $\{S^{(t)}\}$-contraction

$$m = 1$$

[Frommer & Szyld, 1994] (sufficient): $\|.\|_\infty^w$-contraction, $\forall k$

[Frommer & Szyld, 2000] (sufficient): $\{S^{(t)}\}$-contraction, $\forall k$

$$m \geq 1, \quad f^{(k)} \equiv f, \quad \forall k$$

[Baudet, 1978] (sufficient): $|.|$-contraction

$$|f(X) - f(Y)| \leq T \max\{|x^{(1)} - y^{(1)}|, \dots, |x^{(m)} - y^{(m)}|\}$$

$$T \geq O, \quad \rho(T) < 1, \quad (\max\{|x|, |y|\})_i = \max\{|x_i|, |y_i|\}$$

[El Tarazi, 1982] (sufficient): $\|.\|_\infty^w$-contraction

$$\|f(X) - x^*\|_\infty^w \leq \alpha \max\{\|x^{(l)} - x^*\|_\infty^w\}_{1 \leq l \leq m}$$

$$w > 0, \quad \alpha < 1, \quad \|x\|_\infty^w = \max_i |x_i|/w_i$$

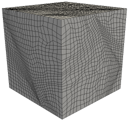
[Frommer & Szyld, 1998] $m = 2$

02 Space domain decomposition

2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$



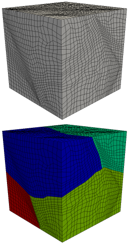
2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains

$$\begin{bmatrix} A_{1,1} & O & A_{1,\Gamma} \\ O & A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,2} & A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} \end{bmatrix}$$



2 - Space domain decomposition

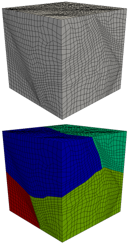
Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains

$$\begin{bmatrix} A_{1,1} & O & A_{1,\Gamma} \\ O & A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,2} & A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} \end{bmatrix}$$

$$\begin{aligned} A_{1,1}x_1 + A_{1,\Gamma}x_\Gamma &= b_1 \\ A_{2,2}x_2 + A_{2,\Gamma}x_\Gamma &= b_2 \\ A_{\Gamma,1}x_1 + A_{\Gamma,2}x_2 + \left(A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)}\right) x_\Gamma &= b_\Gamma^{(1)} + b_\Gamma^{(2)} \end{aligned}$$



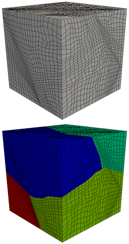
2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains

$$\begin{bmatrix} A_{1,1} & O & A_{1,\Gamma} \\ O & A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,2} & A^{(1)}_{\Gamma,\Gamma} + A^{(2)}_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b^{(1)}_\Gamma + b^{(2)}_\Gamma \end{bmatrix}$$



$$\begin{aligned} A_{1,1}x_1 + A_{1,\Gamma}x_\Gamma &= b_1 \\ A_{2,2}x_2 + A_{2,\Gamma}x_\Gamma &= b_2 \\ A_{\Gamma,1}x_1 + A_{\Gamma,2}x_2 + \left(A^{(1)}_{\Gamma,\Gamma} + A^{(2)}_{\Gamma,\Gamma}\right) x_\Gamma &= b^{(1)}_\Gamma + b^{(2)}_\Gamma \end{aligned}$$

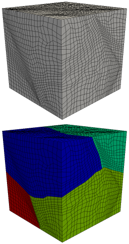
$$\begin{aligned} x_2 &= A^{-1}_{2,2} \left(b_2 - A_{2,\Gamma}x_\Gamma\right) \\ A_{\Gamma,1}x_1 + \left(A^{(1)}_{\Gamma,\Gamma} + A^{(2)}_{\Gamma,\Gamma}\right) x_\Gamma &= b^{(1)}_\Gamma + b^{(2)}_\Gamma - A_{\Gamma,2}x_2 \end{aligned}$$

2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains



$$\begin{bmatrix} A_{1,1} & O & A_{1,\Gamma} \\ O & A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,2} & A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} \end{bmatrix}$$

$$A_{1,1}x_1 + A_{1,\Gamma}x_\Gamma = b_1$$

$$A_{2,2}x_2 + A_{2,\Gamma}x_\Gamma = b_2$$

$$A_{\Gamma,1}x_1 + A_{\Gamma,2}x_2 + \left(A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)}\right) x_\Gamma = b_\Gamma^{(1)} + b_\Gamma^{(2)}$$

$$x_2 = A_{2,2}^{-1}\left(b_2 - A_{2,\Gamma}x_\Gamma\right)$$

$$A_{\Gamma,1}x_1 + \left(A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)}\right) x_\Gamma = b_\Gamma^{(1)} + b_\Gamma^{(2)} - A_{\Gamma,2}x_2$$

$$\begin{bmatrix} A_{1,1} & A_{1,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} - A_{\Gamma,2}x_2 \end{bmatrix}$$

2 - Space domain decomposition

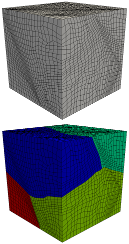
Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains

$$\begin{bmatrix} A_{1,1} & A_{1,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} - A_{\Gamma,2}x_2 \end{bmatrix}$$

$$\begin{bmatrix} A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,2} & A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_2 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_2 \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} - A_{\Gamma,1}x_1 \end{bmatrix}$$



2 - Space domain decomposition

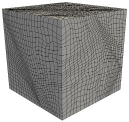
Problem

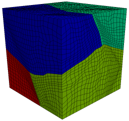
$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains

$$\begin{bmatrix} A_{1,1} & A_{1,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} - A_{\Gamma,2}x_2 \end{bmatrix}$$

$$\begin{bmatrix} A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,2} & A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_2 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_2 \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} - A_{\Gamma,1}x_1 \end{bmatrix}$$





2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains

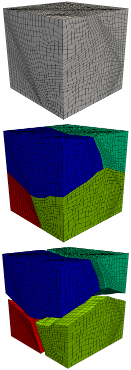
$$\begin{bmatrix} A_{1,1} & A_{1,\Gamma} \\ A_{\Gamma,1} & A^{(1)}_{\Gamma,\Gamma} + A^{(2)}_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} x_1 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ b^{(1)}_\Gamma + b^{(2)}_\Gamma - A_{\Gamma,2}x_2 \end{bmatrix}$$

$$\begin{bmatrix} A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,2} & A^{(1)}_{\Gamma,\Gamma} + A^{(2)}_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} x_2 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_2 \\ b^{(1)}_\Gamma + b^{(2)}_\Gamma - A_{\Gamma,1}x_1 \end{bmatrix}$$

Parallel solutions

$$\begin{bmatrix} A_{1,1} & A_{1,\Gamma} \\ A_{\Gamma,1} & A^{(1)}_{\Gamma,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} x_1 \\ x^{(1)}_\Gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ b^{(1)}_\Gamma + \lambda^{(1)}_\Gamma \end{bmatrix}$$

$$\begin{bmatrix} A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,2} & A^{(2)}_{\Gamma,\Gamma} + \Lambda^{(2)}_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} x_2 \\ x^{(2)}_\Gamma \end{bmatrix} = \begin{bmatrix} b_2 \\ b^{(2)}_\Gamma + \lambda^{(2)}_\Gamma \end{bmatrix}$$



2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains

$$\begin{bmatrix} A_{1,1} & A_{1,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} - A_{\Gamma,2}x_2 \end{bmatrix}$$

$$\begin{bmatrix} A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,2} & A_{\Gamma,\Gamma}^{(1)} + A_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_2 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_2 \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} - A_{\Gamma,1}x_1 \end{bmatrix}$$

Parallel solutions

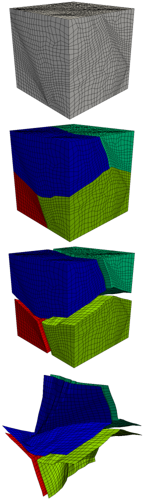
$$\begin{bmatrix} A_{1,1} & A_{1,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,\Gamma}^{(1)} + \Lambda_{\Gamma,\Gamma}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_\Gamma^{(1)} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_\Gamma^{(1)} + \lambda_\Gamma^{(1)} \end{bmatrix}$$

$$\begin{bmatrix} A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,2} & A_{\Gamma,\Gamma}^{(2)} + \Lambda_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_2 \\ x_\Gamma^{(2)} \end{bmatrix} = \begin{bmatrix} b_2 \\ b_\Gamma^{(2)} + \lambda_\Gamma^{(2)} \end{bmatrix}$$

Consistency across interface Γ

$$x_\Gamma^{(1)} = x_\Gamma^{(2)}$$

$$\lambda_\Gamma^{(1)} - \Lambda_{\Gamma,\Gamma}^{(1)} x_\Gamma^{(1)} = -\left(\lambda_\Gamma^{(2)} - \Lambda_{\Gamma,\Gamma}^{(2)} x_\Gamma^{(2)}\right)$$



2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains
Parallel solutions

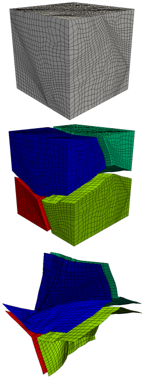
$$\begin{bmatrix} A_{1,1} & A_{1,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,\Gamma}^{(1)} + \Lambda_{\Gamma,\Gamma}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_\Gamma^{(1)} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_\Gamma^{(1)} + \lambda_\Gamma^{(1)} \end{bmatrix}$$

$$\begin{bmatrix} A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,2} & A_{\Gamma,\Gamma}^{(2)} + \Lambda_{\Gamma,\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_2 \\ x_\Gamma^{(2)} \end{bmatrix} = \begin{bmatrix} b_2 \\ b_\Gamma^{(2)} + \lambda_\Gamma^{(2)} \end{bmatrix}$$

Consistency across interface Γ

$$x_\Gamma^{(1)} = x_\Gamma^{(2)}$$

$$\lambda_\Gamma^{(1)} - \Lambda_{\Gamma,\Gamma}^{(1)} x_\Gamma^{(1)} = -\left(\lambda_\Gamma^{(2)} - \Lambda_{\Gamma,\Gamma}^{(2)} x_\Gamma^{(2)}\right)$$



2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains
Parallel solutions

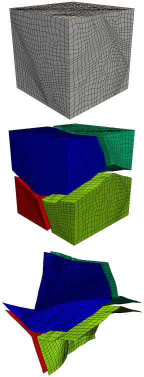
$$\begin{bmatrix} A_{1,1} & A_{1,\Gamma} \\ A_{\Gamma,1} & A^{(1)}_{\Gamma,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} x_1 \\ x^{(1)}_{\Gamma} \end{bmatrix} = \begin{bmatrix} b_1 \\ b^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma} \end{bmatrix}$$

$$\begin{bmatrix} A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,2} & A^{(2)}_{\Gamma,\Gamma} + \Lambda^{(2)}_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} x_2 \\ x^{(2)}_{\Gamma} \end{bmatrix} = \begin{bmatrix} b_2 \\ b^{(2)}_{\Gamma} + \lambda^{(2)}_{\Gamma} \end{bmatrix}$$

Consistency across interface Γ

$$x^{(1)}_{\Gamma} = x^{(2)}_{\Gamma}$$

$$\lambda^{(1)}_{\Gamma} - \Lambda^{(1)}_{\Gamma,\Gamma} x^{(1)}_{\Gamma} = -\left(\lambda^{(2)}_{\Gamma} - \Lambda^{(2)}_{\Gamma,\Gamma} x^{(2)}_{\Gamma}\right)$$



$$x_1 = A^{-1}_{1,1}\left(b_1 - A_{1,\Gamma} x^{(1)}_{\Gamma}\right)$$

$$A_{\Gamma,1} x_1 + \left(A^{(1)}_{\Gamma,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma}\right) x^{(1)}_{\Gamma} = b^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma}$$

2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains
Parallel solutions

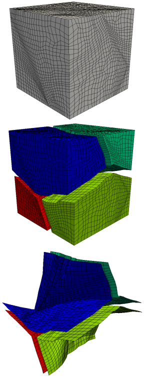
$$\begin{bmatrix} A_{1,1} & A_{1,\Gamma} \\ A_{\Gamma,1} & A^{(1)}_{\Gamma,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} x_1 \\ x^{(1)}_{\Gamma} \end{bmatrix} = \begin{bmatrix} b_1 \\ b^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma} \end{bmatrix}$$

$$\begin{bmatrix} A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,2} & A^{(2)}_{\Gamma,\Gamma} + \Lambda^{(2)}_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} x_2 \\ x^{(2)}_{\Gamma} \end{bmatrix} = \begin{bmatrix} b_2 \\ b^{(2)}_{\Gamma} + \lambda^{(2)}_{\Gamma} \end{bmatrix}$$

Consistency across interface Γ

$$x^{(1)}_{\Gamma} = x^{(2)}_{\Gamma}$$

$$\lambda^{(1)}_{\Gamma} - \Lambda^{(1)}_{\Gamma,\Gamma} x^{(1)}_{\Gamma} = -\left(\lambda^{(2)}_{\Gamma} - \Lambda^{(2)}_{\Gamma,\Gamma} x^{(2)}_{\Gamma}\right)$$



$$x_1 = A^{-1}_{1,1}\left(b_1 - A_{1,\Gamma} x^{(1)}_{\Gamma}\right)$$

$$A_{\Gamma,1} x_1 + \left(A^{(1)}_{\Gamma,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma}\right) x^{(1)}_{\Gamma} = b^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma}$$

$$\left(A^{(1)}_{\Gamma,\Gamma} - A_{\Gamma,1} A^{-1}_{1,1} A_{1,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma}\right) x^{(1)}_{\Gamma} = b^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma} - A_{\Gamma,1} A^{-1}_{1,1} b_1$$

2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains
Parallel solutions

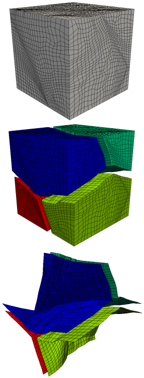
$$\left(A_{\Gamma,\Gamma}^{(1)} - A_{\Gamma,1}A_{1,1}^{-1}A_{1,\Gamma} + \Lambda_{\Gamma,\Gamma}^{(1)}\right) x_{\Gamma}^{(1)} = b_{\Gamma}^{(1)} + \lambda_{\Gamma}^{(1)} - A_{\Gamma,1}A_{1,1}^{-1}b_1$$

$$\left(A_{\Gamma,\Gamma}^{(2)} - A_{\Gamma,2}A_{2,2}^{-1}A_{2,\Gamma} + \Lambda_{\Gamma,\Gamma}^{(2)}\right) x_{\Gamma}^{(2)} = b_{\Gamma}^{(2)} + \lambda_{\Gamma}^{(2)} - A_{\Gamma,2}A_{2,2}^{-1}b_2$$

Consistency across interface Γ

$$x_{\Gamma}^{(1)} = x_{\Gamma}^{(2)}$$

$$\lambda_{\Gamma}^{(1)} - \Lambda_{\Gamma,\Gamma}^{(1)}x_{\Gamma}^{(1)} = -\left(\lambda_{\Gamma}^{(2)} - \Lambda_{\Gamma,\Gamma}^{(2)}x_{\Gamma}^{(2)}\right)$$



2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains
Parallel solutions

$$\left(A^{(1)}_{\Gamma,\Gamma} - A_{\Gamma,1}A^{-1}_{1,1}A_{1,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma}\right) x^{(1)}_{\Gamma} = b^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma} - A_{\Gamma,1}A^{-1}_{1,1}b_1$$

$$\left(A^{(2)}_{\Gamma,\Gamma} - A_{\Gamma,2}A^{-1}_{2,2}A_{2,\Gamma} + \Lambda^{(2)}_{\Gamma,\Gamma}\right) x^{(2)}_{\Gamma} = b^{(2)}_{\Gamma} + \lambda^{(2)}_{\Gamma} - A_{\Gamma,2}A^{-1}_{2,2}b_2$$

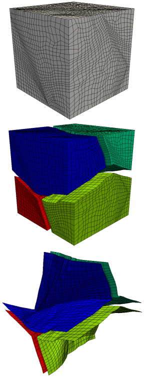
$$\left(S^{(1)}_{\Gamma,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma}\right) x^{(1)}_{\Gamma} = d^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma}$$

$$\left(S^{(2)}_{\Gamma,\Gamma} + \Lambda^{(2)}_{\Gamma,\Gamma}\right) x^{(2)}_{\Gamma} = d^{(2)}_{\Gamma} + \lambda^{(2)}_{\Gamma}$$

Consistency across interface Γ

$$x^{(1)}_{\Gamma} = x^{(2)}_{\Gamma}$$

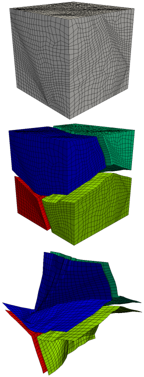
$$\lambda^{(1)}_{\Gamma} - \Lambda^{(1)}_{\Gamma,\Gamma}x^{(1)}_{\Gamma} = -\left(\lambda^{(2)}_{\Gamma} - \Lambda^{(2)}_{\Gamma,\Gamma}x^{(2)}_{\Gamma}\right)$$



2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$



Two subdomains
Parallel solutions

$$\left(A_{\Gamma,\Gamma}^{(1)} - A_{\Gamma,1}A_{1,1}^{-1}A_{1,\Gamma} + \Lambda_{\Gamma,\Gamma}^{(1)}\right) x_\Gamma^{(1)} = b_\Gamma^{(1)} + \lambda_\Gamma^{(1)} - A_{\Gamma,1}A_{1,1}^{-1}b_1$$
$$\left(A_{\Gamma,\Gamma}^{(2)} - A_{\Gamma,2}A_{2,2}^{-1}A_{2,\Gamma} + \Lambda_{\Gamma,\Gamma}^{(2)}\right) x_\Gamma^{(2)} = b_\Gamma^{(2)} + \lambda_\Gamma^{(2)} - A_{\Gamma,2}A_{2,2}^{-1}b_2$$

$$\left(S_{\Gamma,\Gamma}^{(1)} + \Lambda_{\Gamma,\Gamma}^{(1)}\right) x_\Gamma^{(1)} = d_\Gamma^{(1)} + \lambda_\Gamma^{(1)}$$
$$\left(S_{\Gamma,\Gamma}^{(2)} + \Lambda_{\Gamma,\Gamma}^{(2)}\right) x_\Gamma^{(2)} = d_\Gamma^{(2)} + \lambda_\Gamma^{(2)}$$

Consistency across interface Γ

$$x_\Gamma^{(1)} = x_\Gamma^{(2)}$$
$$\lambda_\Gamma^{(1)} - \Lambda_{\Gamma,\Gamma}^{(1)} x_\Gamma^{(1)} = -\left(\lambda_\Gamma^{(2)} - \Lambda_{\Gamma,\Gamma}^{(2)} x_\Gamma^{(2)}\right)$$

Primal approach: $\lambda_\Gamma^{(i)} = \left(S_{\Gamma,\Gamma}^{(i)} + \Lambda_{\Gamma,\Gamma}^{(i)}\right) x_\Gamma^{(i)} - d_\Gamma^{(i)}$

Dual approach: $x_\Gamma^{(i)} = \left(S_{\Gamma,\Gamma}^{(i)} + \Lambda_{\Gamma,\Gamma}^{(i)}\right)^{-1} \left(d_\Gamma^{(i)} + \lambda_\Gamma^{(i)}\right)$

2 - Space domain decomposition

Problem

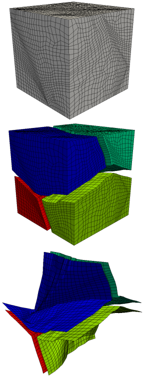
$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains

Parallel solutions

$$\left(A^{(1)}_{\Gamma,\Gamma} - A_{\Gamma,1}A^{-1}_{1,1}A_{1,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma}\right) x^{(1)}_{\Gamma} = b^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma} - A_{\Gamma,1}A^{-1}_{1,1}b_1$$

$$\left(A^{(2)}_{\Gamma,\Gamma} - A_{\Gamma,2}A^{-1}_{2,2}A_{2,\Gamma} + \Lambda^{(2)}_{\Gamma,\Gamma}\right) x^{(2)}_{\Gamma} = b^{(2)}_{\Gamma} + \lambda^{(2)}_{\Gamma} - A_{\Gamma,2}A^{-1}_{2,2}b_2$$



$$\left(S^{(1)}_{\Gamma,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma}\right) x^{(1)}_{\Gamma} = d^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma}$$

$$\left(S^{(2)}_{\Gamma,\Gamma} + \Lambda^{(2)}_{\Gamma,\Gamma}\right) x^{(2)}_{\Gamma} = d^{(2)}_{\Gamma} + \lambda^{(2)}_{\Gamma}$$

Consistency across interface Γ

$$x^{(1)}_{\Gamma} = x^{(2)}_{\Gamma}$$

$$\lambda^{(1)}_{\Gamma} - \Lambda^{(1)}_{\Gamma,\Gamma}x^{(1)}_{\Gamma} = -\left(\lambda^{(2)}_{\Gamma} - \Lambda^{(2)}_{\Gamma,\Gamma}x^{(2)}_{\Gamma}\right)$$

Primal approach: $\lambda^{(i)}_{\Gamma} = \left(S^{(i)}_{\Gamma,\Gamma} + \Lambda^{(i)}_{\Gamma,\Gamma}\right) x^{(i)}_{\Gamma} - d^{(i)}_{\Gamma}$

Dual approach: $x^{(i)}_{\Gamma} = \left(S^{(i)}_{\Gamma,\Gamma} + \Lambda^{(i)}_{\Gamma,\Gamma}\right)^{-1} \left(d^{(i)}_{\Gamma} + \lambda^{(i)}_{\Gamma}\right)$

2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains
Parallel solutions

$$\left(S^{(1)}_{\Gamma,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma}\right) x^{(1)}_{\Gamma} = d^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma}$$

$$\left(S^{(2)}_{\Gamma,\Gamma} + \Lambda^{(2)}_{\Gamma,\Gamma}\right) x^{(2)}_{\Gamma} = d^{(2)}_{\Gamma} + \lambda^{(2)}_{\Gamma}$$

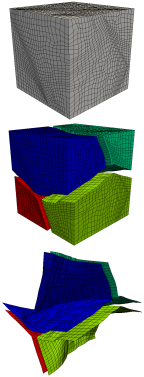
Consistency across interface Γ

$$x^{(1)}_{\Gamma} = x^{(2)}_{\Gamma}$$

$$\lambda^{(1)}_{\Gamma} - \Lambda^{(1)}_{\Gamma,\Gamma} x^{(1)}_{\Gamma} = -\left(\lambda^{(2)}_{\Gamma} - \Lambda^{(2)}_{\Gamma,\Gamma} x^{(2)}_{\Gamma}\right)$$

Primal approach

$$\lambda^{(i)}_{\Gamma} = \left(S^{(i)}_{\Gamma,\Gamma} + \Lambda^{(i)}_{\Gamma,\Gamma}\right) x^{(i)}_{\Gamma} - d^{(i)}_{\Gamma}$$



2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Two subdomains

Parallel solutions

$$\left(S^{(1)}_{\Gamma,\Gamma} + \Lambda^{(1)}_{\Gamma,\Gamma}\right) x^{(1)}_{\Gamma} = d^{(1)}_{\Gamma} + \lambda^{(1)}_{\Gamma}$$

$$\left(S^{(2)}_{\Gamma,\Gamma} + \Lambda^{(2)}_{\Gamma,\Gamma}\right) x^{(2)}_{\Gamma} = d^{(2)}_{\Gamma} + \lambda^{(2)}_{\Gamma}$$

Consistency across interface Γ

$$x^{(1)}_{\Gamma} = x^{(2)}_{\Gamma}$$

$$\lambda^{(1)}_{\Gamma} - \Lambda^{(1)}_{\Gamma,\Gamma} x^{(1)}_{\Gamma} = -\left(\lambda^{(2)}_{\Gamma} - \Lambda^{(2)}_{\Gamma,\Gamma} x^{(2)}_{\Gamma}\right)$$

Primal approach

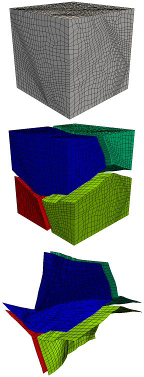
$$\lambda^{(i)}_{\Gamma} = \left(S^{(i)}_{\Gamma,\Gamma} + \Lambda^{(i)}_{\Gamma,\Gamma}\right) x^{(i)}_{\Gamma} - d^{(i)}_{\Gamma}$$

Interface problem: Schur complement inversion

$$\left(S^{(1)}_{\Gamma,\Gamma} + S^{(2)}_{\Gamma,\Gamma}\right) x^{(1)}_{\Gamma} = d^{(1)}_{\Gamma} + d^{(2)}_{\Gamma}$$

$$\left(S^{(1)}_{\Gamma,\Gamma} + S^{(2)}_{\Gamma,\Gamma}\right) x^{(2)}_{\Gamma} = d^{(1)}_{\Gamma} + d^{(2)}_{\Gamma}$$

$$\left(S^{(1)}_{\Gamma,\Gamma} + S^{(2)}_{\Gamma,\Gamma}\right) x_{\Gamma} = d^{(1)}_{\Gamma} + d^{(2)}_{\Gamma}$$



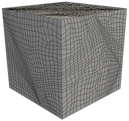
2 - Space domain decomposition

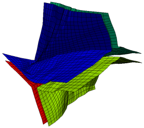
Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion

$$\sum_i S^{(i)}_{\Gamma,\Gamma} x_\Gamma = \sum_i d^{(i)}_\Gamma$$





2 - Space domain decomposition

Problem

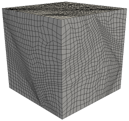
$$Ax = b, \quad x \in \mathbb{C}^n$$

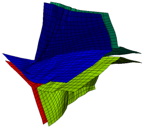
Interface problem: Schur complement inversion

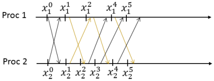
$$\sum_i S^{(i)}_{\Gamma,\Gamma} x_\Gamma = \sum_i d^{(i)}_\Gamma$$

Asynchronous iterative solution

$$\sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$






Proc 1
Proc 2
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

2 - Space domain decomposition

Problem

$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion

$$\sum_i S^{(i)}_{\Gamma,\Gamma} x_\Gamma = \sum_i d^{(i)}_\Gamma$$

Asynchronous iterative solution

$$\sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

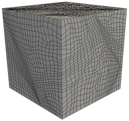
Classical splittings (M, N): (block-)Jacobi, Gauss-Seidel, SOR, ...

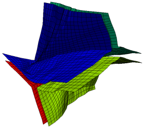
- Minimum requirement: $\mathtt{diag} \sum_i S^{(i)}_{\Gamma,\Gamma}$
- $\Rightarrow$ explicit computation of $\sum_i S^{(i)}_{\Gamma,\Gamma}$
- $\Rightarrow$ vector decomposition

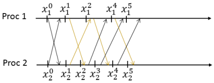
$$x_\Gamma := \begin{bmatrix} (x_\Gamma)_1 & \cdots & (x_\Gamma)_p \end{bmatrix}^\mathsf{T}$$

instead of primal domain decomposition approach

$$x_\Gamma := x^{(1)}_\Gamma = \cdots = x^{(p)}_\Gamma$$







2 - Space domain decomposition

Problem

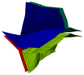
$$Ax = b, \quad x \in \mathbb{C}^n$$

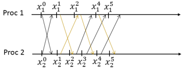
Interface problem: Schur complement inversion

Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S_{\Gamma,\Gamma}^{(i)} = M - N$$

Classical splittings (M, N) $\Rightarrow$ vector decomposition





2 - Space domain decomposition

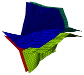
Problem

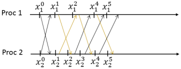
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) ⇒ vector decomposition




Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Proposition 1

$$M := \gamma I$$

Sufficient conditions for vector decomposition

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\qquad \gamma \geq \max\ \operatorname{diag} S_{\Gamma,\Gamma}$

2 - Space domain decomposition

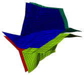
Problem

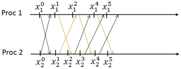
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) $\Rightarrow$ vector decomposition




Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Proposition 1

$$M := \gamma I$$

Sufficient conditions for vector decomposition

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max \text{ diag } S_{\Gamma,\Gamma}$

2 - Space domain decomposition

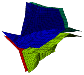
Problem

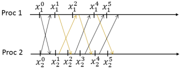
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S_{\Gamma,\Gamma}^{(i)} = M - N$$

Classical splittings (M, N) ⇒ vector decomposition




Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Proposition 1

$$M := \gamma I$$

Sufficient conditions for vector decomposition

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\gamma \geq \max \text{ diag } S_{\Gamma,\Gamma}$

$\mathcal{M}$-matrix
$(S_{\Gamma,\Gamma})_{i,j\neq i} \leq 0, \quad S_{\Gamma,\Gamma}^{-1} \geq O$
$(S_{\Gamma,\Gamma})_{i,j\neq i} \leq 0, \quad S_{\Gamma,\Gamma}$ symmetric and positive definite

[Crabtree and Haynsworth, 1969]

A is an $\mathcal{M}$-matrix $\implies$ $S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix

2 - Space domain decomposition

Problem

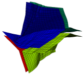
$$Ax = b, \quad x \in \mathbb{C}^n$$

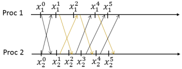
Interface problem: Schur complement inversion

Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) ⇒ vector decomposition




Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Proposition 1

$$M := \gamma I$$

Remark

$$A \text{ is an } \mathcal{M}\text{-matrix} \implies A_{\Gamma,\Gamma} := \sum_i A^{(i)}_{\Gamma,\Gamma} \geq S_{\Gamma,\Gamma}$$

Sufficient conditions for vector decomposition

$$S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}, \qquad \gamma \geq \max \ \text{diag}\, S_{\Gamma,\Gamma}$$

⇒ **Sufficient**

$$S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}, \qquad \gamma \geq \max \ \text{diag}\, A_{\Gamma,\Gamma}$$

$\mathcal{M}$-matrix

$(S_{\Gamma,\Gamma})_{i,j\neq i} \leq 0, \quad S^{-1}_{\Gamma,\Gamma} \geq O$

$(S_{\Gamma,\Gamma})_{i,j\neq i} \leq 0, \quad S_{\Gamma,\Gamma}$ symmetric and positive definite

[Crabtree and Haynsworth, 1969]

$$A \text{ is an } \mathcal{M}\text{-matrix} \implies S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}$$

2 - Space domain decomposition

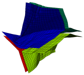
Problem

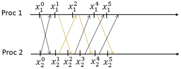
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) ⇒ vector decomposition




Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Proposition 1

$$M := \gamma I$$

Proposition 2

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

Sufficient conditions for vector decomposition

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max \operatorname{diag} S_{\Gamma,\Gamma}$

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max \operatorname{diag} A_{\Gamma,\Gamma}$

[Crabtree and Haynsworth, 1969]

A is an $\mathcal{M}$-matrix $\implies$ $S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix

2 - Space domain decomposition

Problem

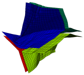
$$Ax = b, \quad x \in \mathbb{C}^n$$

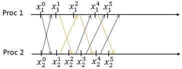
Interface problem: Schur complement inversion

Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) ⇒ vector decomposition




x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 1
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Proposition 1

$$M := \gamma I$$

Sufficient conditions for vector decomposition

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max \text{ diag } S_{\Gamma,\Gamma}$

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max \text{ diag } A_{\Gamma,\Gamma}$

[Crabtree and Haynsworth, 1969]

A is an $\mathcal{M}$-matrix $\implies$ $S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix

Proposition 2

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

Sufficient conditions for vector decomposition

A is an $\mathcal{M}$-matrix, $\quad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma})$ is an $\mathcal{M}$-splitting

2 - Space domain decomposition

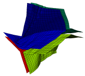
Problem

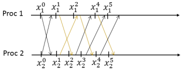
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S_{\Gamma,\Gamma}^{(i)} = M - N$$

Classical splittings (M, N) $\Rightarrow$ vector decomposition




Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Proposition 1

$$M := \gamma I$$

Sufficient conditions for vector decomposition

$$S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}, \qquad \gamma \geq \max \operatorname{diag} S_{\Gamma,\Gamma}$$

$$S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}, \qquad \gamma \geq \max \operatorname{diag} A_{\Gamma,\Gamma}$$

[Crabtree and Haynsworth, 1969]

$$A \text{ is an } \mathcal{M}\text{-matrix} \implies S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}$$

Remark

$$A_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix} \implies \text{block-Jacobi is an } \mathcal{M}\text{-splitting}$$

Proposition 2

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

Sufficient conditions for vector decomposition

$$A \text{ is an } \mathcal{M}\text{-matrix}, \qquad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma}) \text{ is an } \mathcal{M}\text{-splitting}$$

$\mathcal{M}$-splitting
$M_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad N_{\Gamma,\Gamma} \geq O$

2 - Space domain decomposition

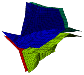
Problem

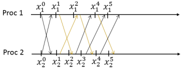
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) ⇒ vector decomposition




Proc 1
Proc 2
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Proposition 1

$$M := \gamma I$$

Sufficient conditions for vector decomposition

$$S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}, \qquad \gamma \geq \max \text{ diag } S_{\Gamma,\Gamma}$$

$$S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}, \qquad \gamma \geq \max \text{ diag } A_{\Gamma,\Gamma}$$

[Crabtree and Haynsworth, 1969]

$$A \text{ is an } \mathcal{M}\text{-matrix} \implies S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}$$

Remark

$$A_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix} \implies \text{block-Jacobi is an } \mathcal{M}\text{-splitting}$$

Proposition 2

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

Sufficient conditions for vector decomposition

$$A \text{ is an } \mathcal{M}\text{-matrix}, \qquad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma}) \text{ is an } \mathcal{M}\text{-splitting}$$

$\mathcal{M}$-splitting
$M_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad N_{\Gamma,\Gamma} \geq O$

⇒ **Sufficient**

$$A \text{ is an } \mathcal{M}\text{-matrix}, \qquad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma}) \text{ is a block-Jacobi}$$

2 - Space domain decomposition

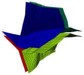
Problem

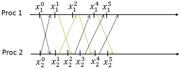
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) $\Rightarrow$ vector decomposition




Proc 1
Proc 2
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Proposition 1

$$M := \gamma I$$

Sufficient conditions for vector decomposition

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max \text{ diag } S_{\Gamma,\Gamma}$

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max \text{ diag } A_{\Gamma,\Gamma}$

[Crabtree and Haynsworth, 1969]

A is an $\mathcal{M}$-matrix $\implies$ $S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix

Proposition 2

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

Sufficient conditions for vector decomposition

A is an $\mathcal{M}$-matrix, $\quad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma})$ is an $\mathcal{M}$-splitting

A is an $\mathcal{M}$-matrix, $\quad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma})$ is a block-Jacobi

- Equivalent sufficient conditions for synchronous convergence of primal Schur decomposition
- What about asynchronous primal Schur decomposition?

2 - Space domain decomposition

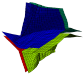
Problem

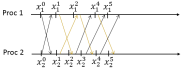
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) $\Rightarrow$ vector decomposition




Proc 1
Proc 2
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Propositions

$$M := \gamma I$$

Asynchronous vector / Synchronous primal Schur

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max\ \text{diag}\, S_{\Gamma,\Gamma}$

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max\ \text{diag}\, A_{\Gamma,\Gamma}$

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

Asynchronous vector / Synchronous primal Schur

A is an $\mathcal{M}$-matrix, $\quad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma})$ is an $\mathcal{M}$-splitting

A is an $\mathcal{M}$-matrix, $\quad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma})$ is a block-Jacobi

Asynchronous primal Schur decomposition

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

2 - Space domain decomposition

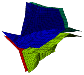
Problem

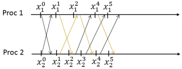
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) $\Rightarrow$ vector decomposition




Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Propositions

$$M := \gamma I$$

Asynchronous vector / Synchronous primal Schur

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\gamma \geq \max$ diag $S_{\Gamma,\Gamma}$
$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\gamma \geq \max$ diag $A_{\Gamma,\Gamma}$

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

Asynchronous vector / Synchronous primal Schur

A is an $\mathcal{M}$-matrix, $(M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma})$ is an $\mathcal{M}$-splitting
A is an $\mathcal{M}$-matrix, $(M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma})$ is a block-Jacobi

Asynchronous primal Schur decomposition

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

$$A = \widetilde{M} - \widetilde{N}, \qquad \widetilde{M} = \begin{bmatrix} A_{1,1} & O & O \\ O & A_{2,2} & O \\ O & O & M_{\Gamma,\Gamma} \end{bmatrix}$$

2 - Space domain decomposition

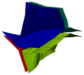
Problem

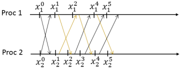
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) $\Rightarrow$ vector decomposition




Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Propositions

$$M := \gamma I$$

Asynchronous vector / Synchronous primal Schur

$$S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}, \qquad \gamma \geq \max \ \text{diag}\, S_{\Gamma,\Gamma}$$
$$S_{\Gamma,\Gamma} \text{ is an } \mathcal{M}\text{-matrix}, \qquad \gamma \geq \max \ \text{diag}\, A_{\Gamma,\Gamma}$$

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

Asynchronous vector / Synchronous primal Schur

$$A \text{ is an } \mathcal{M}\text{-matrix}, \qquad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma}) \text{ is an } \mathcal{M}\text{-splitting}$$
$$A \text{ is an } \mathcal{M}\text{-matrix}, \qquad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma}) \text{ is a block-Jacobi}$$

Asynchronous primal Schur decomposition

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

$$A = \widetilde{M} - \widetilde{N}, \qquad \widetilde{M} = \begin{bmatrix} A_{1,1} & O & O \\ O & A_{2,2} & O \\ O & O & M_{\Gamma,\Gamma} \end{bmatrix}$$

Theorem

$$\rho(|\widetilde{M}^{-1}\widetilde{N}|) < 1, \quad |\sum_i M^{-1}_{\Gamma,\Gamma} N^{(i)}_{\Gamma,\Gamma}| = \sum_i |M^{-1}_{\Gamma,\Gamma} N^{(i)}_{\Gamma,\Gamma}|$$

2 - Space domain decomposition

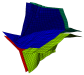
Problem

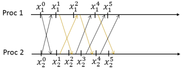
$$Ax = b, \quad x \in \mathbb{C}^n$$

Interface problem: Schur complement inversion
Asynchronous iterative solution

$$S_{\Gamma,\Gamma} := \sum_i S^{(i)}_{\Gamma,\Gamma} = M - N$$

Classical splittings (M, N) ⇒ vector decomposition




Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

Propositions

$$M := \gamma I$$

Asynchronous vector / Synchronous primal Schur

$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max$ diag $S_{\Gamma,\Gamma}$
$S_{\Gamma,\Gamma}$ is an $\mathcal{M}$-matrix, $\quad \gamma \geq \max$ diag $A_{\Gamma,\Gamma}$

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

Asynchronous vector / Synchronous primal Schur

A is an $\mathcal{M}$-matrix, $\quad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma})$ is an $\mathcal{M}$-splitting
A is an $\mathcal{M}$-matrix, $\quad (M_{\Gamma,\Gamma}, N_{\Gamma,\Gamma})$ is a block-Jacobi

Asynchronous primal Schur decomposition

$$M := M_{\Gamma,\Gamma}, \qquad A_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N_{\Gamma,\Gamma}$$

$$A = \widetilde{M} - \widetilde{N}, \qquad \widetilde{M} = \begin{bmatrix} A_{1,1} & O & O \\ O & A_{2,2} & O \\ O & O & M_{\Gamma,\Gamma} \end{bmatrix}$$

Theorem

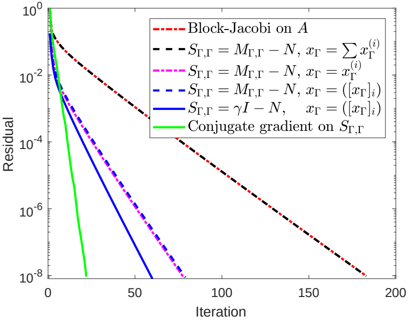
$$\rho(|\widetilde{M}^{-1}\widetilde{N}|) < 1, \qquad |\sum_i M^{-1}_{\Gamma,\Gamma} N^{(i)}_{\Gamma,\Gamma}| = \sum_i |M^{-1}_{\Gamma,\Gamma} N^{(i)}_{\Gamma,\Gamma}|$$

$$N^{(i)}_{\Gamma,\Gamma} = \alpha_i N_{\Gamma,\Gamma}, \qquad \sum_i \alpha_i = 1, \qquad \alpha_i > 0$$

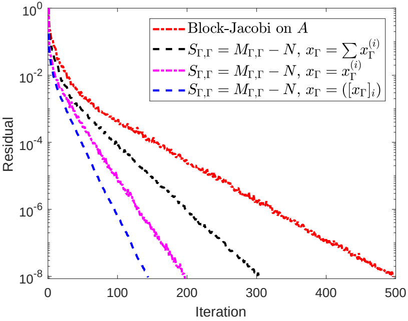
2 - Space domain decomposition

- 2D Laplacian model problem, 16 subdomains
- P^k and τ_j^i pre-generated (same for all methods)
- $S_{\Gamma,\Gamma} = M_{\Gamma,\Gamma} - N$, with $x_\Gamma = \sum_i x_\Gamma^{(i)}$: slight improvement of *[Magoulès & Venet, 2018]*

$$P^k = \{1, \dots, 16\}, \qquad \tau_j^i(k) = k$$



$$P^k \subset \{1, \dots, 16\}, \qquad k - 4 \leq \tau_j^i(k) \leq k$$



Outline

03 Time domain decomposition

3 - Time domain decomposition

Partial differential equation

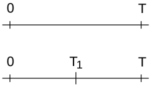
$$\delta(u(s,t),s,t) = 0, \quad t \in [0,T], \ \ s \in \Omega$$

$u(\Omega, 0)$ given

3 - Time domain decomposition

Partial differential equation

$$\delta(u(s,t),s,t) = 0, \quad t \in [0,T], \ \ s \in \Omega$$
$$u(\Omega,0) \text{ given}$$



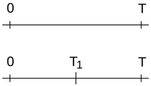
Two time intervals

$$\delta(u_0(s,t),s,t) = 0, \quad t \in [0, T_1] \qquad \delta(u_1(s,t),s,t) = 0, \quad t \in [T_1, T]$$
$$u_0(\Omega,0) = u(\Omega,0) \qquad u_1(\Omega,T_1) = u_0(\Omega,T_1)$$

3 - Time domain decomposition

Partial differential equation

$$\delta(u(s,t),s,t) = 0, \quad t \in [0,T], \ \ s \in \Omega$$
$$u(\Omega,0) \text{ given}$$



Two time intervals

$$\delta(u_0(s,t),s,t) = 0, \quad t \in [0,T_1] \qquad \delta(u_1(s,t),s,t) = 0, \quad t \in [T_1,T]$$
$$u_0(\Omega,0) = u(\Omega,0) \qquad u_1(\Omega,T_1) = u_0(\Omega,T_1)$$

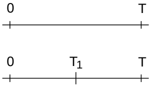
Parallel solutions

$$\widetilde{u}_0 \equiv u_0 \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1,T]$$
$$\lambda_0(\Omega) = u_0(\Omega,0) \qquad \widetilde{u}_1(\Omega,T_1) = \lambda_1(\Omega)$$

3 - Time domain decomposition

Partial differential equation

$$\delta(u(s,t),s,t) = 0, \quad t \in [0,T], \quad s \in \Omega$$
$$u(\Omega,0) \text{ given}$$



Two time intervals

$$\delta(u_0(s,t),s,t) = 0, \quad t \in [0,T_1] \qquad \delta(u_1(s,t),s,t) = 0, \quad t \in [T_1,T]$$
$$u_0(\Omega,0) = u(\Omega,0) \qquad u_1(\Omega,T_1) = u_0(\Omega,T_1)$$

Parallel solutions

$$\widetilde{u}_0 \equiv u_0 \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1,T]$$
$$\lambda_0(\Omega) = u_0(\Omega,0) \qquad \widetilde{u}_1(\Omega,T_1) = \lambda_1(\Omega)$$

Consistency across interface T_1

$$\lambda_1(\Omega) = \widetilde{u}_0(\Omega,T_1)$$

3 - Time domain decomposition

Partial differential equation

$$\delta(u(s,t),s,t) = 0, \quad t \in [0,T], \ \ s \in \Omega$$
$$u(\Omega,0) \text{ given}$$

Two time intervals

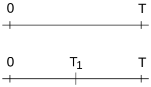
$$\delta(u_0(s,t),s,t) = 0, \quad t \in [0,T_1] \qquad \delta(u_1(s,t),s,t) = 0, \quad t \in [T_1,T]$$
$$u_0(\Omega,0) = u(\Omega,0) \qquad u_1(\Omega,T_1) = u_0(\Omega,T_1)$$

Parallel solutions

$$\widetilde{u}_0 \equiv u_0 \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1,T]$$
$$\lambda_0(\Omega) = u_0(\Omega,0) \qquad \widetilde{u}_1(\Omega,T_1) = \lambda_1(\Omega)$$

Consistency across interface T_1

$$\lambda_1(\Omega) = \widetilde{u}_0(\Omega,T_1)$$



Parareal approach:
Two-level time discretization
[Lions et al, 2001]

3 - Time domain decomposition

Partial differential equation
Two time intervals

$$\delta(u_0(s,t),s,t) = 0, \quad t \in [0, T_1] \qquad \delta(u_1(s,t),s,t) = 0, \quad t \in [T_1, T]$$

$$u_0(\Omega, 0) = u(\Omega, 0) \qquad u_1(\Omega, T_1) = u_0(\Omega, T_1)$$

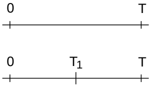
Parallel solutions

$$\widetilde{u}_0 \equiv u_0 \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1, T]$$

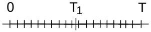
$$\lambda_0(\Omega) = u_0(\Omega, 0) \qquad \widetilde{u}_1(\Omega, T_1) = \lambda_1(\Omega)$$

Consistency across interface T_1

$$\lambda_1(\Omega) = \widetilde{u}_0(\Omega, T_1)$$



Parareal approach:
Two-level time discretization
[Lions et al, 2001]



Fine time discretization

$$\alpha(\widetilde{u}_0(s,t_{n+1})) = \beta(\widetilde{u}_0(s,t_n)) \qquad \alpha(\widetilde{u}_1(s,t_{n+1})) = \beta(\widetilde{u}_1(s,t_n))$$

$\Rightarrow$ parallel fine propagator

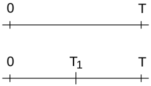
$$F(\lambda_0) := \widetilde{u}_0(\Omega, T_1) \qquad F(\lambda_1) := \widetilde{u}_1(\Omega, T)$$

3 - Time domain decomposition

Partial differential equation

Two time intervals

$$\delta(u_0(s,t),s,t)=0,\quad t\in[0,T_1] \qquad \delta(u_1(s,t),s,t)=0,\quad t\in[T_1,T]$$
$$u_0(\Omega,0)=u(\Omega,0) \qquad u_1(\Omega,T_1)=u_0(\Omega,T_1)$$



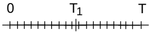
Parallel solutions

$$\widetilde{u}_0 \equiv u_0 \qquad \delta(\widetilde{u}_1(s,t),s,t)=0,\quad t\in[T_1,T]$$
$$\lambda_0(\Omega)=u_0(\Omega,0) \qquad \widetilde{u}_1(\Omega,T_1)=\lambda_1(\Omega)$$

Parareal approach:
Two-level time discretization
[Lions et al, 2001]

Consistency across interface T_1

$$\lambda_1(\Omega)=\widetilde{u}_0(\Omega,T_1)$$



Fine time discretization

$$\alpha(\widetilde{u}_0(s,t_{n+1}))=\beta(\widetilde{u}_0(s,t_n)) \qquad \alpha(\widetilde{u}_1(s,t_{n+1}))=\beta(\widetilde{u}_1(s,t_n))$$

$\Rightarrow$ parallel fine propagator

$$F(\lambda_0):=\widetilde{u}_0(\Omega,T_1) \qquad F(\lambda_1):=\widetilde{u}_1(\Omega,T)$$

Coarse time discretization

$$\alpha(u(s,T_1))=\beta(u(s,0))$$
$$\alpha(u(s,T))=\beta(u(s,T_1))$$

$\Rightarrow$ serial coarse propagator

$$G(u(\Omega,0)):=\widetilde{u}(\Omega,T_1),\quad G(\widetilde{u}(\Omega,T_1))=\widetilde{u}(\Omega,T)$$

3 - Time domain decomposition

Partial differential equation
Two time intervals
Parallel solutions

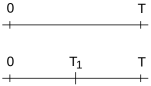
$$\delta(\widetilde{u}_0(s,t),s,t) = 0, \quad t \in [0, T_1] \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1, T]$$

$$\lambda_0(\Omega) \text{ given} \qquad \widetilde{u}_1(\Omega, T_1) = \lambda_1(\Omega)$$

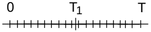
Consistency across interface T_1

$$\lambda_1(\Omega) = \widetilde{u}_0(\Omega, T_1)$$

Parareal iterations



Parareal approach:
Two-level time discretization
[Lions et al, 2001]



$$F(\lambda_0) := \widetilde{u}_0(\Omega, T_1)$$
$$F(\lambda_1) = \widetilde{u}_1(\Omega, T)$$

$$G(u(\Omega, 0)) := \widetilde{u}(\Omega, T_1)$$
$$G(\widetilde{u}(\Omega, T_1)) = \widetilde{u}(\Omega, T)$$

3 - Time domain decomposition

Partial differential equation
Two time intervals
Parallel solutions

$$\delta(\widetilde{u}_0(s,t),s,t) = 0, \quad t \in [0, T_1] \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1, T]$$

$$\lambda_0(\Omega) \text{ given} \qquad \widetilde{u}_1(\Omega, T_1) = \lambda_1(\Omega)$$

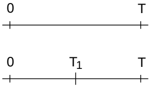
Consistency across interface T_1

$$\lambda_1(\Omega) = \widetilde{u}_0(\Omega, T_1)$$

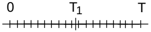
Parareal iterations

$$\lambda_0^0 := u_0(\Omega, 0)$$

$$\lambda_1^0 := G(\lambda_0^0) \qquad \text{wait} \qquad [0, T_1]$$



Parareal approach:
Two-level time discretization
[Lions et al, 2001]



$$F(\lambda_0) := \widetilde{u}_0(\Omega, T_1)$$

$$F(\lambda_1) = \widetilde{u}_1(\Omega, T)$$

$$G(u(\Omega, 0)) := \widetilde{u}(\Omega, T_1)$$

$$G(\widetilde{u}(\Omega, T_1)) = \widetilde{u}(\Omega, T)$$

Prediction

3 - Time domain decomposition

Partial differential equation
Two time intervals
Parallel solutions

$$\delta(\widetilde{u}_0(s,t),s,t) = 0, \quad t \in [0, T_1] \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1, T]$$

$$\lambda_0(\Omega) \text{ given} \qquad \widetilde{u}_1(\Omega, T_1) = \lambda_1(\Omega)$$

Consistency across interface T_1

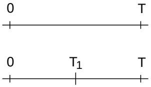
$$\lambda_1(\Omega) = \widetilde{u}_0(\Omega, T_1)$$

Parareal iterations

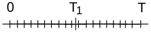
$$\lambda_0^0 := u_0(\Omega, 0)$$

$$\lambda_1^0 := G(\lambda_0^0) \qquad \text{wait} \qquad [0, T_1]$$

$$\lambda_2^0 := G(\lambda_1^0) \qquad [T_1, T]$$



Parareal approach:
Two-level time discretization
[Lions et al, 2001]



$$F(\lambda_0) := \widetilde{u}_0(\Omega, T_1)$$

$$F(\lambda_1) = \widetilde{u}_1(\Omega, T)$$

$$G(u(\Omega, 0)) := \widetilde{u}(\Omega, T_1)$$

$$G(\widetilde{u}(\Omega, T_1)) = \widetilde{u}(\Omega, T)$$

Prediction

3 - Time domain decomposition

Partial differential equation

Two time intervals

Parallel solutions

$$\delta(\widetilde{u}_0(s,t),s,t)=0, \quad t\in[0,T_1] \qquad \delta(\widetilde{u}_1(s,t),s,t)=0, \quad t\in[T_1,T]$$

$$\lambda_0(\Omega)\ \text{given} \qquad \widetilde{u}_1(\Omega,T_1)=\lambda_1(\Omega)$$

Consistency across interface T_1

$$\lambda_1(\Omega)=\widetilde{u}_0(\Omega,T_1)$$

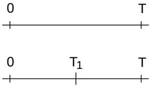
Parareal iterations

$$\lambda_0^0 := u_0(\Omega,0)$$

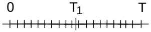
$$\lambda_1^0 := G(\lambda_0^0) \qquad \text{wait} \qquad [0,T_1]$$

$$\lambda_2^0 := G(\lambda_1^0) \qquad [T_1,T]$$

$$F(\lambda_0^0) \qquad F(\lambda_1^0) \qquad [0,T_1]\ [T_1,T]$$



Parareal approach:
Two-level time discretization
[Lions et al, 2001]



$$F(\lambda_0) := \widetilde{u}_0(\Omega,T_1)$$

$$F(\lambda_1) = \widetilde{u}_1(\Omega,T)$$

$$G(u(\Omega,0)) := \widetilde{u}(\Omega,T_1)$$

$$G(\widetilde{u}(\Omega,T_1)) = \widetilde{u}(\Omega,T)$$

Prediction

3 - Time domain decomposition

Partial differential equation

Two time intervals

Parallel solutions

$$\delta(\widetilde{u}_0(s,t),s,t)=0,\quad t\in[0,T_1] \qquad \delta(\widetilde{u}_1(s,t),s,t)=0,\quad t\in[T_1,T]$$

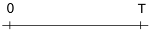
$$\lambda_0(\Omega)\text{ given} \qquad \widetilde{u}_1(\Omega,T_1)=\lambda_1(\Omega)$$

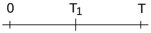
Consistency across interface T_1

$$\lambda_1(\Omega)=\widetilde{u}_0(\Omega,T_1)$$

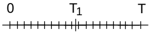
Parareal iterations

$\lambda_0^0 := u_0(\Omega,0)$		
$\lambda_1^0 := G(\lambda_0^0)$	wait	$[0,T_1]$
	$\lambda_2^0 := G(\lambda_1^0)$	$[T_1,T]$
$F(\lambda_0^0)$	$F(\lambda_1^0)$	$[0,T_1]$ $[T_1,T]$
$\lambda_0^1 := \lambda_0^0$		
$\lambda_1^1 := G(\lambda_0^1)+F(\lambda_0^0)-G(\lambda_0^0)$	wait	$[0,T_1]$





Parareal approach:
Two-level time discretization
[Lions et al, 2001]



$$F(\lambda_0):=\widetilde{u}_0(\Omega,T_1)$$

$$F(\lambda_1)=\widetilde{u}_1(\Omega,T)$$

$$G(u(\Omega,0)):=\widetilde{u}(\Omega,T_1)$$

$$G(\widetilde{u}(\Omega,T_1))=\widetilde{u}(\Omega,T)$$

Prediction

Gap

3 - Time domain decomposition

Partial differential equation

Two time intervals

Parallel solutions

$$\delta(\widetilde{u}_0(s,t),s,t) = 0, \quad t \in [0, T_1] \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1, T]$$

$$\lambda_0(\Omega) \text{ given} \qquad \widetilde{u}_1(\Omega, T_1) = \lambda_1(\Omega)$$

Consistency across interface T_1

$$\lambda_1(\Omega) = \widetilde{u}_0(\Omega, T_1)$$

Parareal iterations

$$\lambda_0^0 := u_0(\Omega, 0)$$

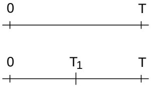
$$\lambda_1^0 := G(\lambda_0^0) \qquad \text{wait} \qquad [0, T_1]$$

$$\lambda_2^0 := G(\lambda_1^0) \qquad [T_1, T]$$

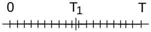
$$F(\lambda_0^0) \qquad F(\lambda_1^0) \qquad [0, T_1]\ [T_1, T]$$

$$\lambda_0^1 := \lambda_0^0$$

$$\lambda_1^1 := G(\lambda_0^1) + F(\lambda_0^0) - G(\lambda_0^0) \qquad \text{wait} \qquad [0, T_1]$$



Parareal approach:
Two-level time discretization
[Lions et al, 2001]



$$F(\lambda_0) := \widetilde{u}_0(\Omega, T_1)$$

$$F(\lambda_1) = \widetilde{u}_1(\Omega, T)$$

$$G(u(\Omega, 0)) := \widetilde{u}(\Omega, T_1)$$

$$G(\widetilde{u}(\Omega, T_1)) = \widetilde{u}(\Omega, T)$$

Prediction

Gap

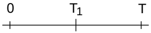
Corrected prediction

3 - Time domain decomposition

Partial differential equation
Two time intervals
Parallel solutions

$$\delta(\widetilde{u}_0(s,t),s,t) = 0, \quad t \in [0, T_1] \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1, T]$$

$$\lambda_0(\Omega) \text{ given} \qquad \widetilde{u}_1(\Omega, T_1) = \lambda_1(\Omega)$$

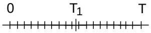


Consistency across interface T_1

$$\lambda_1(\Omega) = \widetilde{u}_0(\Omega, T_1)$$

Parareal approach:
Two-level time discretization
[Lions et al, 2001]

Parareal iterations



$\lambda_0^0 := u_0(\Omega, 0)$			
$\lambda_1^0 := G(\lambda_0^0)$	wait	$[0, T_1]$	
	$\lambda_2^0 := G(\lambda_1^0)$	$[T_1, T]$	$F(\lambda_0) := \widetilde{u}_0(\Omega, T_1)$
$F(\lambda_0^0)$	$F(\lambda_1^0)$	$[0, T_1]$ $[T_1, T]$	$F(\lambda_1) = \widetilde{u}_1(\Omega, T)$
$\lambda_0^1 := \lambda_0^0$			
$\lambda_1^1 := G(\lambda_0^1) + F(\lambda_0^0) - G(\lambda_0^0)$	wait	$[0, T_1]$	$G(u(\Omega, 0)) := \widetilde{u}(\Omega, T_1)$
	$\lambda_2^1 := G(\lambda_1^1) + F(\lambda_1^0) - G(\lambda_1^0)$	$[T_1, T]$	$G(\widetilde{u}(\Omega, T_1)) = \widetilde{u}(\Omega, T)$

Prediction
Gap
Corrected prediction

3 - Time domain decomposition

Partial differential equation
Two time intervals
Parallel solutions

$$\delta(\widetilde{u}_0(s,t),s,t) = 0, \quad t \in [0, T_1] \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1, T]$$

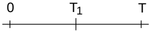
$$\lambda_0(\Omega) \text{ given} \qquad \widetilde{u}_1(\Omega, T_1) = \lambda_1(\Omega)$$

Consistency across interface T_1

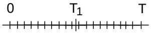
$$\lambda_1(\Omega) = \widetilde{u}_0(\Omega, T_1)$$

Parareal iterations

$\lambda_0^0 := u_0(\Omega, 0)$		
$\lambda_1^0 := G(\lambda_0^0)$	wait	$[0, T_1]$
	$\lambda_2^0 := G(\lambda_1^0)$	$[T_1, T]$
$F(\lambda_0^0)$	$F(\lambda_1^0)$	$[0, T_1]$ $[T_1, T]$
$\lambda_0^1 := \lambda_0^0$		
$\lambda_1^1 := G(\lambda_0^1) + F(\lambda_0^0) - G(\lambda_0^0)$	wait	$[0, T_1]$
	$\lambda_2^1 := G(\lambda_1^1) + F(\lambda_1^0) - G(\lambda_1^0)$	$[T_1, T]$
$\lambda_0^2 := \lambda_0^1$	$F(\lambda_1^1)$	$[T_1, T]$



Parareal approach:
Two-level time discretization
[Lions et al, 2001]



$$F(\lambda_0) := \widetilde{u}_0(\Omega, T_1)$$

$$F(\lambda_1) = \widetilde{u}_1(\Omega, T)$$

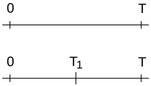
$$G(u(\Omega, 0)) := \widetilde{u}(\Omega, T_1)$$

$$G(\widetilde{u}(\Omega, T_1)) = \widetilde{u}(\Omega, T)$$

Prediction
Gap
Corrected prediction

3 - Time domain decomposition

Partial differential equation
Two time intervals
Parallel solutions



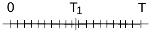
$$\delta(\widetilde{u}_0(s,t),s,t)=0,\quad t\in[0,T_1]\qquad\qquad \delta(\widetilde{u}_1(s,t),s,t)=0,\quad t\in[T_1,T]$$

$$\lambda_0(\Omega)\text{ given}\qquad\qquad \widetilde{u}_1(\Omega,T_1)=\lambda_1(\Omega)$$

Consistency across interface T_1

$$\lambda_1(\Omega)=\widetilde{u}_0(\Omega,T_1)$$

Parareal approach:
Two-level time discretization
[Lions et al, 2001]



Parareal iterations

$\lambda_0^0 := u_0(\Omega,0)$			
$\lambda_1^0 := G(\lambda_0^0)$	wait	$[0, T_1]$	
	$\lambda_2^0 := G(\lambda_1^0)$	$[T_1, T]$	$F(\lambda_0) := \widetilde{u}_0(\Omega,T_1)$
$F(\lambda_0^0)$	$F(\lambda_1^0)$	$[0, T_1]$ $[T_1, T]$	$F(\lambda_1) = \widetilde{u}_1(\Omega,T)$
$\lambda_0^1 := \lambda_0^0$			
$\lambda_1^1 := G(\lambda_0^1)+F(\lambda_0^0)-G(\lambda_0^0)$	wait	$[0, T_1]$	$G(u(\Omega,0)) := \widetilde{u}(\Omega,T_1)$
	$\lambda_2^1 := G(\lambda_1^1)+F(\lambda_1^0)-G(\lambda_1^0)$	$[T_1, T]$	$G(\widetilde{u}(\Omega,T_1)) = \widetilde{u}(\Omega,T)$
$\lambda_0^2 := \lambda_0^1$	$F(\lambda_1^1)$	$[T_1, T]$	
$\lambda_1^2 := \lambda_1^1$	$\lambda_2^2 := G(\lambda_1^2)+F(\lambda_1^1)-G(\lambda_1^1)$	$[T_1, T]$	

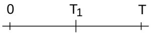
Prediction
Gap
Corrected prediction

3 - Time domain decomposition

Partial differential equation
Two time intervals
Parallel solutions

$$\delta(\widetilde{u}_0(s,t),s,t) = 0, \quad t \in [0, T_1] \qquad \delta(\widetilde{u}_1(s,t),s,t) = 0, \quad t \in [T_1, T]$$

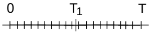
$$\lambda_0(\Omega) \text{ given} \qquad \widetilde{u}_1(\Omega, T_1) = \lambda_1(\Omega)$$



Consistency across interface T_1

$$\lambda_1(\Omega) = \widetilde{u}_0(\Omega, T_1)$$

Parareal approach:
Two-level time discretization
[Lions et al, 2001]



Parareal iterations

$$\lambda_0^0 := u_0(\Omega, 0)$$

$$\lambda_1^0 := G(\lambda_0^0) \qquad \text{wait} \qquad [0, T_1]$$

$$\lambda_2^0 := G(\lambda_1^0) \qquad [T_1, T] \qquad F(\lambda_0) := \widetilde{u}_0(\Omega, T_1)$$

$$F(\lambda_0^0) \qquad F(\lambda_1^0) \qquad [0, T_1]\ [T_1, T] \qquad F(\lambda_1) = \widetilde{u}_1(\Omega, T)$$

$$\lambda_0^1 := \lambda_0^0$$

$$\lambda_1^1 := G(\lambda_0^1) + F(\lambda_0^0) - G(\lambda_0^0) \qquad \text{wait} \qquad [0, T_1] \qquad G(u(\Omega, 0)) := \widetilde{u}(\Omega, T_1)$$

$$\lambda_2^1 := G(\lambda_1^1) + F(\lambda_1^0) - G(\lambda_1^0) \qquad [T_1, T] \qquad G(\widetilde{u}(\Omega, T_1)) = \widetilde{u}(\Omega, T)$$

$$\lambda_0^2 := \lambda_0^1 \qquad F(\lambda_1^1) \qquad [T_1, T]$$

$$\lambda_1^2 := \lambda_1^1 \qquad \lambda_2^2 := G(\lambda_1^2) + F(\lambda_1^1) - G(\lambda_1^1) \qquad [T_1, T]$$

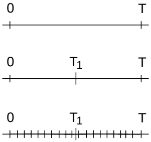
$$\lambda_i^{k+1} = G(\lambda_{i-1}^{k+1}) + F(\lambda_{i-1}^{k}) - G(\lambda_{i-1}^{k}), \qquad [T_{i-1}, T_i]$$

Prediction
Gap
Corrected prediction

3 - Time domain decomposition

Partial differential equation
N time intervals
Parallel solutions and consistency across interfaces T_i
Parareal iterations

$$\lambda_i^{k+1} = G(\lambda_{i-1}^{k+1}) + F(\lambda_{i-1}^{k}) - G(\lambda_{i-1}^{k}), \qquad [T_{i-1}, T_i]$$



$$\delta(\widetilde{u}_i(s,t), s, t) = 0$$
$$\widetilde{u}_i(\Omega, T_i) = \lambda_i(\Omega)$$

Parareal approach:
Two-level time discretization
[Lions et al, 2001]

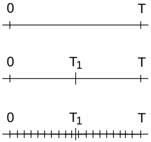
$$F(\lambda_i) = \widetilde{u}_i(\Omega, T_{i+1})$$
$$G(\widetilde{u}(\Omega, T_i)) = \widetilde{u}(\Omega, T_{i+1})$$

Prediction G
Gap $F - G$
Corrected prediction $G+$

3 - Time domain decomposition

Partial differential equation
N time intervals
Parallel solutions and consistency across interfaces T_i
Parareal iterations

$$\lambda_i^{k+1} = G(\lambda_{i-1}^{k+1}) + F(\lambda_{i-1}^{k}) - G(\lambda_{i-1}^{k}), \qquad [T_{i-1}, T_i]$$



Theorem 1
PDE extension of ODE result
[Gander & Vandewalle, 2007]

Iterative error

$$\|\lambda^k - \lambda^*\|_\infty \leq \alpha^k \|\lambda^0 - \lambda^*\|_\infty$$

$$\alpha = \frac{1-\theta^N}{1-\theta}\|F-G\|, \ \theta \neq 1, \ \theta \geq \|G\|$$

$$\delta(\widetilde{u}_i(s,t), s, t) = 0$$

$$\widetilde{u}_i(\Omega, T_i) = \lambda_i(\Omega)$$

Parareal approach:
Two-level time discretization
[Lions et al, 2001]

$$F(\lambda_i) = \widetilde{u}_i(\Omega, T_{i+1})$$

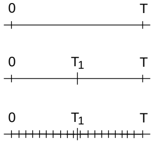
$$G(\widetilde{u}(\Omega, T_i)) = \widetilde{u}(\Omega, T_{i+1})$$

Prediction G
Gap $F - G$
Corrected prediction $G+$

3 - Time domain decomposition

Partial differential equation
N time intervals
Parallel solutions and consistency across interfaces T_i
Parareal iterations

$$\lambda_i^{k+1} = G(\lambda_{i-1}^{k+1}) + F(\lambda_{i-1}^{k}) - G(\lambda_{i-1}^{k}), \qquad [T_{i-1}, T_i]$$



$$\delta(\widetilde{u}_i(s,t), s, t) = 0$$

$$\widetilde{u}_i(\Omega, T_i) = \lambda_i(\Omega)$$

Theorem 1
PDE extension of ODE result
[Gander & Vandewalle, 2007]

Iterative error

$$\|\lambda^k - \lambda^*\|_\infty \leq \alpha^k \|\lambda^0 - \lambda^*\|_\infty$$

$$\alpha = \frac{1-\theta^N}{1-\theta}\|F-G\|, \ \theta \neq 1, \ \theta \geq \|G\|$$

Sufficient convergence condition for
$\|G\| < 1$ **(stability region)**

$$\|G\| + \|F-G\| < 1 + \|G\|^N \|F-G\|$$

Parareal approach:
Two-level time discretization
[Lions et al, 2001]

$$F(\lambda_i) = \widetilde{u}_i(\Omega, T_{i+1})$$

$$G(\widetilde{u}(\Omega, T_i)) = \widetilde{u}(\Omega, T_{i+1})$$

Prediction G
Gap $F - G$
Corrected prediction $G+$

3 - Time domain decomposition

Partial differential equation

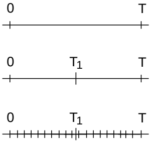
N time intervals

Parallel solutions and consistency across interfaces T_i

Parareal iterations

$$\lambda_i^{k+1} = G(\lambda_{i-1}^{k+1}) + F(\lambda_{i-1}^{k}) - G(\lambda_{i-1}^{k}), \qquad [T_{i-1}, T_i]$$

$$\lambda_i^{k+1} = G\left(\lambda_{i-1}^{\tau_{i-1}^{i,1}(k+1)}\right) + F\left(\lambda_{i-1}^{\tau_{i-1}^{i,2}(k)}\right) - G\left(\lambda_{i-1}^{\tau_{i-1}^{i,2}(k)}\right)$$



$$\delta(\widetilde{u}_i(s,t), s, t) = 0$$

$$\widetilde{u}_i(\Omega, T_i) = \lambda_i(\Omega)$$

Theorem 1

PDE extension of ODE result

[Gander & Vandewalle, 2007]

Iterative error

$$\|\lambda^k - \lambda^*\|_\infty \leq \alpha^k \|\lambda^0 - \lambda^*\|_\infty$$

$$\alpha = \frac{1-\theta^N}{1-\theta}\|F - G\|, \;\; \theta \neq 1, \;\; \theta \geq \|G\|$$

Sufficient convergence condition for

$\|G\| < 1$ (stability region)

$$\|G\| + \|F - G\| < 1 + \|G\|^N \|F - G\|$$

Proposition

Parareal approach:

Two-level time discretization

[Lions et al, 2001]

$$F(\lambda_i) = \widetilde{u}_i(\Omega, T_{i+1})$$

$$G(\widetilde{u}(\Omega, T_i)) = \widetilde{u}(\Omega, T_{i+1})$$

Prediction G

Gap $F - G$

Corrected prediction $G+$

3 - Time domain decomposition

Partial differential equation

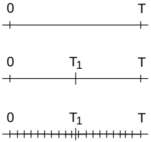
N time intervals

Parallel solutions and consistency across interfaces T_i

Parareal iterations

$$\lambda_i^{k+1} = G(\lambda_{i-1}^{k+1}) + F(\lambda_{i-1}^{k}) - G(\lambda_{i-1}^{k}), \qquad [T_{i-1}, T_i]$$

$$\lambda_i^{k+1} = G\left(\lambda_{i-1}^{\tau_{i-1}^{i,1}(k+1)}\right) + F\left(\lambda_{i-1}^{\tau_{i-1}^{i,2}(k)}\right) - G\left(\lambda_{i-1}^{\tau_{i-1}^{i,2}(k)}\right)$$



Theorem 1

PDE extension of ODE result

[Gander & Vandewalle, 2007]

Iterative error

$$\|\lambda^k - \lambda^*\|_\infty \leq \alpha^k \|\lambda^0 - \lambda^*\|_\infty$$

$$\alpha = \frac{1-\theta^N}{1-\theta}\|F-G\|, \ \theta \neq 1, \ \theta \geq \|G\|$$

Sufficient convergence condition for $\|G\| < 1$ (stability region)

$$\|G\| + \|F-G\| < 1 + \|G\|^N \|F-G\|$$

Proposition

Asynchronous convergence (sufficient)

$$\|G\| + \|F-G\| < 1$$

$$\delta(\widetilde{u}_i(s,t), s, t) = 0$$

$$\widetilde{u}_i(\Omega, T_i) = \lambda_i(\Omega)$$

Parareal approach:

Two-level time discretization

[Lions et al, 2001]

$$F(\lambda_i) = \widetilde{u}_i(\Omega, T_{i+1})$$

$$G(\widetilde{u}(\Omega, T_i)) = \widetilde{u}(\Omega, T_{i+1})$$

Prediction G

Gap $F - G$

Corrected prediction $G+$

3 - Time domain decomposition

Partial differential equation

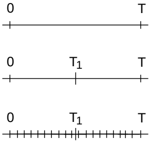
N time intervals

Parallel solutions and consistency across interfaces T_i

Parareal iterations

$$\lambda_i^{k+1} = G(\lambda_{i-1}^{k+1}) + F(\lambda_{i-1}^{k}) - G(\lambda_{i-1}^{k}), \qquad [T_{i-1}, T_i]$$

$$\lambda_i^{k+1} = G\left(\lambda_{i-1}^{\tau_{i-1}^{i,1}(k+1)}\right) + F\left(\lambda_{i-1}^{\tau_{i-1}^{i,2}(k)}\right) - G\left(\lambda_{i-1}^{\tau_{i-1}^{i,2}(k)}\right)$$



Theorem 1

PDE extension of ODE result

[Gander & Vandewalle, 2007]

Iterative error

$$\|\lambda^k - \lambda^*\|_\infty \leq \alpha^k \|\lambda^0 - \lambda^*\|_\infty$$

$$\alpha = \frac{1-\theta^N}{1-\theta}\|F-G\|, \; \theta \neq 1, \; \theta \geq \|G\|$$

Sufficient convergence condition for $\|G\| < 1$ (stability region)

$$\|G\| + \|F-G\| < 1 + \|G\|^N \|F-G\|$$

Proposition

Asynchronous convergence (sufficient)

$$\|G\| + \|F-G\| < 1$$

Theorem 2

$$\|\lambda^k - \lambda^*\|_\infty \leq \widetilde{\alpha}^{\tau(k)} \|\lambda^0 - \lambda^*\|_\infty$$

$$\widetilde{\alpha} = \|G\| + \|F-G\|, \quad \lim_{k\to\infty} \tau(k) = \infty$$

$$\delta(\widetilde{u}_i(s,t), s, t) = 0$$

$$\widetilde{u}_i(\Omega, T_i) = \lambda_i(\Omega)$$

Parareal approach:

Two-level time discretization

[Lions et al, 2001]

$$F(\lambda_i) = \widetilde{u}_i(\Omega, T_{i+1})$$

$$G(\widetilde{u}(\Omega, T_i)) = \widetilde{u}(\Omega, T_{i+1})$$

Prediction G

Gap $F - G$

Corrected prediction $G+$

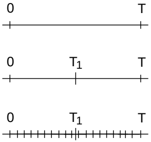
SystemX CentraleSupélec

3 - Time domain decomposition

Partial differential equation
N time intervals
Parallel solutions and consistency across interfaces T_i
Parareal iterations

$$\lambda_i^{k+1} = G(\lambda_{i-1}^{k+1}) + F(\lambda_{i-1}^{k}) - G(\lambda_{i-1}^{k}), \qquad [T_{i-1}, T_i]$$

$$\lambda_i^{k+1} = G\left(\lambda_{i-1}^{\tau_{i-1}^{i,1}(k+1)}\right) + F\left(\lambda_{i-1}^{\tau_{i-1}^{i,2}(k)}\right) - G\left(\lambda_{i-1}^{\tau_{i-1}^{i,2}(k)}\right)$$



Theorem 1
PDE extension of ODE result
[Gander & Vandewalle, 2007]

Iterative error

$$\|\lambda^k - \lambda^*\|_\infty \leq \alpha^k \|\lambda^0 - \lambda^*\|_\infty$$

$$\alpha = \frac{1-\theta^N}{1-\theta}\|F-G\|, \;\; \theta \neq 1, \;\; \theta \geq \|G\|$$

Sufficient convergence condition for $\|G\| < 1$ (stability region)

$$\|G\| + \|F-G\| < 1 + \|G\|^N \|F-G\|$$

Proposition

Asynchronous convergence (sufficient)

$$\|G\| + \|F-G\| < 1$$

Theorem 2

$$\|\lambda^k - \lambda^*\|_\infty \leq \widetilde{\alpha}^{\tau(k)} \|\lambda^0 - \lambda^*\|_\infty$$

$$\widetilde{\alpha} = \|G\| + \|F-G\|, \quad \lim_{k\to\infty} \tau(k) = \infty$$

Synchronous convergence for $N \to \infty$

$$\|G\| + \|F-G\| < 1$$

$$\delta(\widetilde{u}_i(s,t), s, t) = 0$$

$$\widetilde{u}_i(\Omega, T_i) = \lambda_i(\Omega)$$

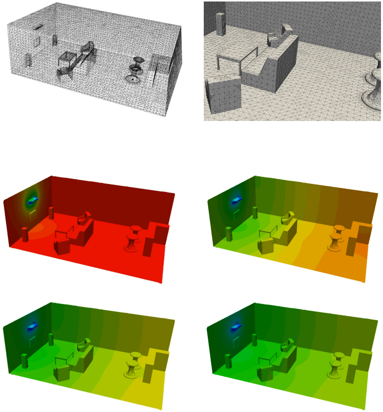
Parareal approach:
Two-level time discretization
[Lions et al, 2001]

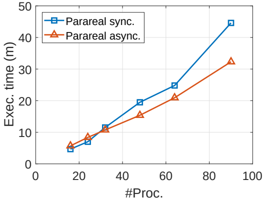
$$F(\lambda_i) = \widetilde{u}_i(\Omega, T_{i+1})$$

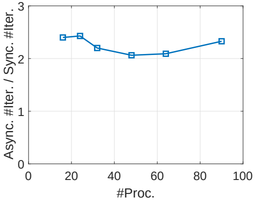
$$G(\widetilde{u}(\Omega, T_i)) = \widetilde{u}(\Omega, T_{i+1})$$

Prediction G
Gap $F - G$
Corrected prediction $G+$

3 - Time domain decomposition







$$t_{n+1} - t_n = 0.002, \qquad T_{i+1} - T_i = 0.2$$

$$N = \#\text{Proc}, \qquad T = N \times (T_{i+1} - T_i)$$

Outline

04 Asynchronous convergence detection

4 - Asynchronous convergence detection

Problem

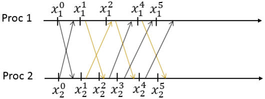
$$f(x) = x, \quad x \in E$$

Asynchronous iterations

$$x_i^{k+1} = f_i(x_1^{\tau_1^i(k)}, \ldots, x_p^{\tau_p^i(k)}), \quad i \in P^k$$

Convergence detection

$$\bar{x} := \left(x_1^{k_1}, \ldots, x_p^{k_p}\right) \simeq x^*$$

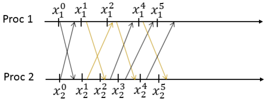


4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous iterations


Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

$$x_i^{k+1} = f_i(x_1^{\tau_1^i(k)}, \ldots, x_p^{\tau_p^i(k)}), \quad i \in P^k$$

Convergence detection

$$\bar{x} := \left(x_1^{k_1}, \ldots, x_p^{k_p}\right) \simeq x^*$$

Approach

- Finite time termination *[Bertsekas & Tsitsiklis, 1989], [El Baz, 1996], [Savari & Bertsekas, 1996], ...*
- Snapshot-based supervised termination *[Savari & Bertsekas, 1996]*
- Predicted termination (finite number of iterations) *[Evans & Chikohora, 1998]*
- Local convergence monitoring *[Bahi et al, 2005, 2008]*
- Nested-sets-based supervised termination *[Miellou et al, 2008]*

4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Approach

- Finite time termination *[Bertsekas & Tsitsiklis, 1989], [El Baz, 1996], [Savari & Bertsekas, 1996], ...*
- Snapshot-based supervised termination *[Savari & Bertsekas, 1996]*
- Predicted termination (finite number of iterations) *[Evans & Chikohora, 1998]*
- Local convergence monitoring *[Bahi et al, 2005, 2008]*
- Nested-sets-based supervised termination *[Miellou et al, 2008]*

	Intrusiveness	Centralization	Effectiveness	Messages size
Bertsekas & Tsitsiklis, 1989	altered iterations	–	formal	–
El Baz, 1996	altered iterations	–	formal	–
Savari & Bertsekas, 1996	altered iterations	–	formal	–
Savari & Bertsekas, 1996	non-intrusive	two reductions	exact	$\mathcal{O}(n)$
Evans & Chikohora, 1998	non-intrusive	no reduction	heuristic	0
Bahi *et al*, 2005	non-intrusive	one reduction	heuristic	$\mathcal{O}(1)$
Bahi *et al*, 2008	piggybacking	two reductions	heuristic	$\mathcal{O}(1)$
Miellou *et al*, 2008	–	–	formal	–

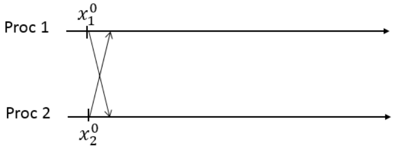
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*



- Initiator / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
  3. Start recording neighbors' messages
- On *marker* reception
  1. Stop recording corresponding neighbor's messages
- On computation message reception
  1. Record message

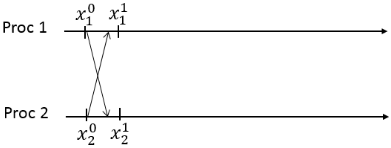
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*



$x_1^1 := f_1(x_1^0, x_2^0)$ $x_2^1 := f_2(x_1^0, x_2^0)$

- Initiator / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
  3. Start recording neighbors' messages
- On *marker* reception
  1. Stop recording corresponding neighbor's messages
- On computation message reception
  1. Record message

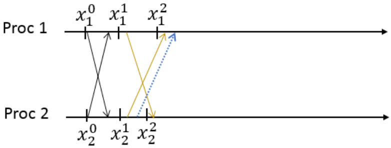
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*



$x_1^1 := f_1(x_1^0, x_2^0)$ x_2^1 $x_2^1 := f_2(x_1^0, x_2^0)$

$x_1^2 := f_1(x_1^1, x_2^0)$ $x_2^2 := f_2(x_1^0, x_2^1)$

- Initiator / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
  3. Start recording neighbors' messages
- On *marker* reception
  1. Stop recording corresponding neighbor's messages
- On computation message reception
  1. Record message

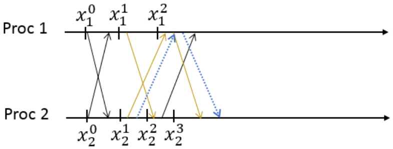
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*



$x_1^1 := f_1(x_1^0, x_2^0)$

$x_1^2 := f_1(x_1^1, x_2^0)$

x_1^2 $x_1^3 := x_1^2$

x_2^1 $x_2^1 := f_2(x_1^0, x_2^0)$

$x_2^2 := f_2(x_1^0, x_2^1)$

$x_2^3 := f_2(x_1^1, x_2^2)$ x_1^1

- Initiator / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
  3. Start recording neighbors' messages
- On *marker* reception
  1. Stop recording corresponding neighbor's messages
- On computation message reception
  1. Record message

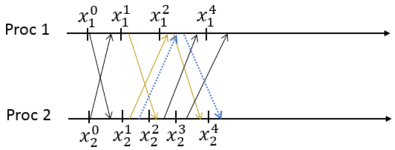
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*



$$\begin{array}{lllll} & x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 & x_2^1 := f_2(x_1^0, x_2^0) & \\ & x_1^2 := f_1(x_1^1, x_2^0) & & x_2^2 := f_2(x_1^0, x_2^1) & \\ x_1^2 & x_1^3 := x_1^2 & & x_2^3 := f_2(x_1^1, x_2^2) & x_1^1 \\ & x_1^4 := f_1(x_1^3, x_2^2) & & x_2^4 := f_2(x_1^2, x_2^3) & x_1^2 \end{array}$$

- Initiator / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
  3. Start recording neighbors' messages
- On *marker* reception
  1. Stop recording corresponding neighbor's messages
- On computation message reception
  1. Record message

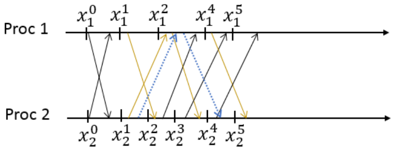
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*


x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 1
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

$$
\begin{array}{llll}
 & x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 & x_2^1 := f_2(x_1^0, x_2^0) \\
 & x_1^2 := f_1(x_1^1, x_2^0) & & x_2^2 := f_2(x_1^0, x_2^1) \\
x_1^2 & x_1^3 := x_1^2 & & x_2^3 := f_2(x_1^1, x_2^2) \quad x_1^1 \\
 & x_1^4 := f_1(x_1^3, x_2^2) & & x_2^4 := f_2(x_1^2, x_2^3) \quad x_1^2 \\
 & x_1^5 := f_1(x_1^4, x_2^3) & & x_2^5 := f_2(x_1^2, x_2^4)
\end{array}
$$

- Initiator / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
  3. Start recording neighbors' messages
- On *marker* reception
  1. Stop recording corresponding neighbor's messages
- On computation message reception
  1. Record message

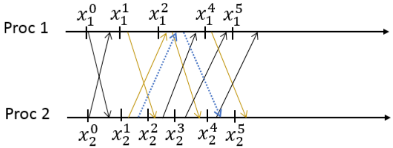
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*


x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 1
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

	$x_1^1 := f_1(x_1^0, x_2^0)$	x_2^1	$x_2^1 := f_2(x_1^0, x_2^0)$	
	$x_1^2 := f_1(x_1^1, x_2^0)$		$x_2^2 := f_2(x_1^0, x_2^1)$	
x_1^2	$x_1^3 := x_1^2$		$x_2^3 := f_2(x_1^1, x_2^2)$	x_1^1
	$x_1^4 := f_1(x_1^3, x_2^2)$		$x_2^4 := f_2(x_1^2, x_2^3)$	x_1^2
	$x_1^5 := f_1(x_1^4, x_2^3)$		$x_2^5 := f_2(x_1^2, x_2^4)$	
	$(x_1^2, ?)$		(x_1^2, x_2^1)	

- Initiator / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
  3. Start recording neighbors' messages
- On *marker* reception
  1. Stop recording corresponding neighbor's messages
- On computation message reception
  1. Record message

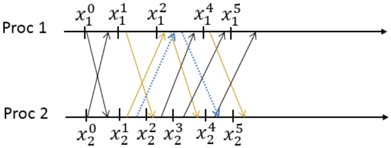
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*


x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 1
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

$x_1^1 := f_1(x_1^0, x_2^0)$

$x_1^2 := f_1(x_1^1, x_2^0)$

x_1^2 $x_1^3 := x_1^2$

$x_1^4 := f_1(x_1^3, x_2^2)$

$x_1^5 := f_1(x_1^4, x_2^3)$

$(x_1^2, ?)$

x_2^1 $x_2^1 := f_2(x_1^0, x_2^0)$

$x_2^2 := f_2(x_1^0, x_2^1)$

$x_2^3 := f_2(x_1^1, x_2^2)$ x_1^1

$x_2^4 := f_2(x_1^2, x_2^3)$ x_1^2

$x_2^5 := f_2(x_1^2, x_2^4)$

(x_1^2, x_2^1)

Proposition 1
Asynchronous iterations snapshot (AIS)

- Initiator / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
  3. Start recording neighbors' messages
- On *marker* reception
  1. Stop recording corresponding neighbor's messages
- On computation message reception
  1. Record message

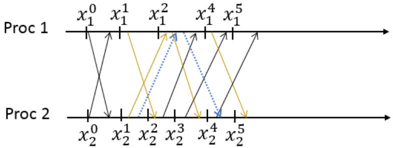
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*


x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 1
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

$$\begin{array}{lll}
 & x_1^1 & := f_1(x_1^0, x_2^0) \\
 & x_1^2 & := f_1(x_1^1, x_2^0) \\
x_1^2 & x_1^3 & := x_1^2 \\
 & x_1^4 & := f_1(x_1^3, x_2^2) \\
 & x_1^5 & := f_1(x_1^4, x_2^3) \\
 & (x_1^2, ?) &
\end{array}$$

$$\begin{array}{llll}
x_2^1 & x_2^1 & := f_2(x_1^0, x_2^0) & \\
 & x_2^2 & := f_2(x_1^0, x_2^1) & \\
 & x_2^3 & := f_2(x_1^1, x_2^2) & x_1^1 \\
 & x_2^4 & := f_2(x_1^2, x_2^3) & x_1^2 \\
 & x_2^5 & := f_2(x_1^2, x_2^4) & \\
 & (x_1^2, x_2^1) & &
\end{array}$$

Proposition 1
Asynchronous iterations snapshot (AIS)

- Local convergence / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
- On *marker* reception
  1. Stop recording corresponding neighbor's messages
- On computation message reception
  1. Record message

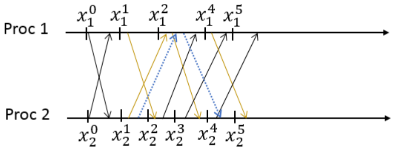
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*



$$\begin{array}{clccll}
 & x_1^1 := f_1(x_1^0, x_2^0) & & x_2^1 & x_2^1 := f_2(x_1^0, x_2^0) & \\
 & x_1^2 := f_1(x_1^1, x_2^0) & & & x_2^2 := f_2(x_1^0, x_2^1) & \\
x_1^2 & x_1^3 := x_1^2 & x_2^1 & & x_2^3 := f_2(x_1^1, x_2^2) & \\
 & x_1^4 := f_1(x_1^3, x_2^2) & & & x_2^4 := f_2(x_1^2, x_2^3) & x_1^2 \\
 & x_1^5 := f_1(x_1^4, x_2^3) & & & x_2^5 := f_2(x_1^2, x_2^4) & \\
 & (x_1^2, ?) & & & (x_2^1, x_2^1) &
\end{array}$$

Proposition 1
Asynchronous iterations snapshot (AIS)

- Local convergence / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
- On *marker* reception
  1. Record last corresponding neighbor's message
- On computation message reception
  1. Record message

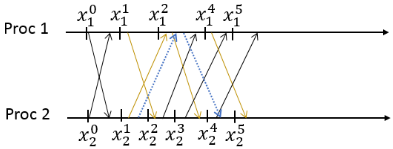
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*


x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 1
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

$$\begin{array}{lllll}
 & x_1^1 := f_1(x_1^0, x_2^0) & x_2^1 & x_2^1 := f_2(x_1^0, x_2^0) & \\
 & x_1^2 := f_1(x_1^1, x_2^0) & & x_2^2 := f_2(x_1^0, x_2^1) & \\
x_1^2 & x_1^3 := x_1^2 & x_2^1 & x_2^3 := f_2(x_1^1, x_2^2) & \\
 & x_1^4 := f_1(x_1^3, x_2^2) & & x_2^4 := f_2(x_1^2, x_2^3) & x_1^2 \\
 & x_1^5 := f_1(x_1^4, x_2^3) & & x_2^5 := f_2(x_1^2, x_2^4) & \\
 & (x_1^2, x_2^1) & & (x_1^2, x_2^1) &
\end{array}$$

Proposition 1
Asynchronous iterations snapshot (AIS)

- Local convergence / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
- On *marker* reception
  1. Record last corresponding neighbor's message

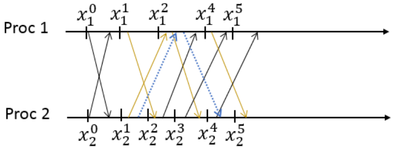
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*


x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 1
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

$$\begin{array}{lll}
 & x_1^1 := f_1(x_1^0, x_2^0) & \\
 & x_1^2 := f_1(x_1^1, x_2^0) & \\
x_1^2 & x_1^3 := x_1^2 & x_2^1 \\
 & x_1^4 := f_1(x_1^3, x_2^2) & \\
 & x_1^5 := f_1(x_1^4, x_2^3) & \\
 & (x_1^2, ?) &
\end{array}
\qquad
\begin{array}{lll}
x_2^1 & x_2^1 := f_2(x_1^0, x_2^0) & \\
 & x_2^2 := f_2(x_1^0, x_2^1) & \\
 & x_2^3 := f_2(x_1^1, x_2^2) & \\
 & x_2^4 := f_2(x_1^2, x_2^3) & x_1^2 \\
 & x_2^5 := f_2(x_1^2, x_2^4) & \\
 & (?, x_2^1) &
\end{array}$$

Proposition 1
Asynchronous iterations snapshot (AIS)

- Local convergence / First *marker* reception
  1. Record local state
  2. Send *marker* to neighbors
- On *marker* reception
  1. Record last corresponding neighbor's message

Non First-In-First-Out delivering
$\Rightarrow$ marker crossing computation message

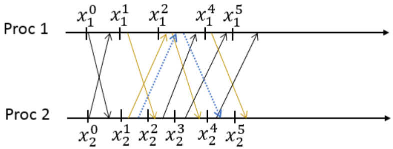
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*


Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

$$\begin{array}{llllll}
 & x_1^1 := f_1(x_1^0, x_2^0) & & x_2^1 & x_2^1 := f_2(x_1^0, x_2^0) & \\
 & x_1^2 := f_1(x_1^1, x_2^0) & & & x_2^2 := f_2(x_1^0, x_2^1) & \\
x_1^2 & x_1^3 := x_1^2 & x_2^1 & & x_2^3 := f_2(x_1^1, x_2^2) & \\
 & x_1^4 := f_1(x_1^3, x_2^2) & & & x_2^4 := f_2(x_1^2, x_2^3) & x_1^2 \\
 & x_1^5 := f_1(x_1^4, x_2^3) & & & x_2^5 := f_2(x_1^2, x_2^4) & \\
 & (x_1^2, x_2^1) & & & (x_1^2, x_2^1) &
\end{array}$$

Proposition 1
Asynchronous iterations snapshot (AIS)

Non FIFO case

- *[Savari & Bertsekas, 1996]*
  Embed computation message into marker
  ⇒ marker size: $\mathcal{O}(n)$

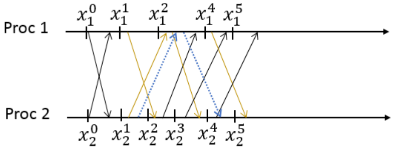
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*


Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

$$\begin{array}{llllll}
 & x_1^1 := f_1(x_1^0, x_2^0) & & x_2^1 & x_2^1 := f_2(x_1^0, x_2^0) & \\
 & x_1^2 := f_1(x_1^1, x_2^0) & & & x_2^2 := f_2(x_1^0, x_2^1) & \\
x_1^2 & x_1^3 := x_1^2 & x_2^1 & & x_2^3 := f_2(x_1^1, x_2^2) & \\
 & x_1^4 := f_1(x_1^3, x_2^2) & & & x_2^4 := f_2(x_1^2, x_2^3) & x_1^2 \\
 & x_1^5 := f_1(x_1^4, x_2^3) & & & x_2^5 := f_2(x_1^2, x_2^4) & \\
 & (x_1^2, x_2^1) & & & (x_1^2, x_2^1) &
\end{array}$$

Proposition 1
Asynchronous iterations snapshot (AIS)

Non FIFO case

- *[Savari & Bertsekas, 1996]*
  Embed computation message into marker
  $\Rightarrow$ marker size: $\mathcal{O}(n)$

Proposition 2
Non FIFO Asynchronous iterations snapshot (NFAIS)

Non FIFO characterization
A marker can cross at most *m* computation messages

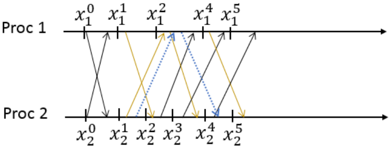
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*


x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 1
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

$$\begin{aligned}
& x_1^1 := f_1(x_1^0, x_2^0) \\
& x_1^2 := f_1(x_1^1, x_2^0) \\
x_1^2 \quad & x_1^3 := x_1^2 \qquad x_2^1 \\
& x_1^4 := f_1(x_1^3, x_2^2) \\
& x_1^5 := f_1(x_1^4, x_2^3) \\
& (x_1^2, x_2^1)
\end{aligned}$$

$$\begin{aligned}
x_2^1 \quad & x_2^1 := f_2(x_1^0, x_2^0) \\
& x_2^2 := f_2(x_1^0, x_2^1) \\
& x_2^3 := f_2(x_1^1, x_2^2) \\
& x_2^4 := f_2(x_1^2, x_2^3) \quad x_1^2 \\
& x_2^5 := f_2(x_1^2, x_2^4) \\
& (x_1^2, x_2^1)
\end{aligned}$$

Proposition 1
Asynchronous iterations snapshot (AIS)

Non FIFO case

- *[Savari & Bertsekas, 1996]*
  Embed computation message into marker
  ⇒ marker size: $\mathcal{O}(n)$

Proposition 2
Non FIFO Asynchronous iterations snapshot (NFAIS)

Non FIFO characterization
A marker can cross at most *m* computation messages

1. Send marker after *m* iterations under local convergence
2. Send flag-marker after *m* subsequent iterations
   → flag armed if continuous local convergence

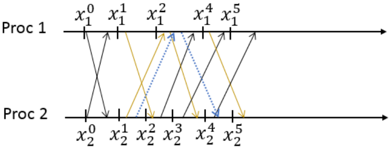
4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Distributed snapshot *[Chandy & Lamport, 1985]*



Proc 1
x_1^0 x_1^1 x_1^2 x_1^4 x_1^5
Proc 2
x_2^0 x_2^1 x_2^2 x_2^3 x_2^4 x_2^5

$$x_1^1 := f_1(x_1^0, x_2^0) \qquad x_2^1 \quad x_2^1 := f_2(x_1^0, x_2^0)$$

$$x_1^2 := f_1(x_1^1, x_2^0) \qquad x_2^2 := f_2(x_1^0, x_2^1)$$

$$x_1^2 \quad x_1^3 := x_1^2 \qquad x_2^1 \qquad x_2^3 := f_2(x_1^1, x_2^2)$$

$$x_1^4 := f_1(x_1^3, x_2^2) \qquad x_2^4 := f_2(x_1^2, x_2^3) \quad x_1^2$$

$$x_1^5 := f_1(x_1^4, x_2^3) \qquad x_2^5 := f_2(x_1^2, x_2^4)$$

$$(x_1^2, x_2^1) \qquad (x_1^2, x_2^1)$$

Proposition 1
Asynchronous iterations snapshot (AIS)

Non FIFO case

- *[Savari & Bertsekas, 1996]*
  Embed computation message into marker
  ⇒ marker size: $\mathcal{O}(n)$

Proposition 2
Non FIFO Asynchronous iterations snapshot (NFAIS)

Non FIFO characterization
A marker can cross at most *m* computation messages

1. Send marker after *m* iterations under local convergence
2. Send flag-marker after *m* subsequent iterations
   → flag armed if continuous local convergence

Practical case
Marker transmits faster than computation message
⇒ no need for flag-marker

4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Theorem

$$\bar{x} := \begin{bmatrix} \bar{x}_1^{(1)} & \cdots & \bar{x}_p^{(p)} \end{bmatrix}^{\mathsf{T}}$$

Global residual bound

$$r(\bar{x}) < \sigma(r(\bar{x}^{(1)}), \ldots, r(\bar{x}^{(p)})) + Cm\varepsilon$$

constant C depending on r, σ and p
ε used for local convergence

Proposition 1
Asynchronous iterations snapshot (AIS)

Non FIFO case

- *[Savari & Bertsekas, 1996]*
  Embed computation message into marker
  $\Rightarrow$ marker size: $\mathcal{O}(n)$

Proposition 2
Non FIFO Asynchronous iterations snapshot (NFAIS)

Non FIFO characterization
A marker can cross at most *m* computation messages

1. Send marker after *m* iterations under local convergence
2. Send flag-marker after *m* subsequent iterations
   $\longrightarrow$ flag armed if continuous local convergence

Practical case
Marker transmits faster than computation message
$\Rightarrow$ no need for flag-marker

4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Snapshot-based supervised termination

Theorem

$$\bar{x} := \begin{bmatrix} \bar{x}_1^{(1)} & \dots & \bar{x}_p^{(p)} \end{bmatrix}^T$$

Global residual bound

$$r(\bar{x}) < \sigma(r(\bar{x}^{(1)}), \dots, r(\bar{x}^{(p)})) + Cm\varepsilon$$

constant C depending on r, σ and p
ε used for local convergence

Corollary
$\| \|_\infty^w$–based residual

$$\sigma(r(\bar{x}^{(1)}), \dots, r(\bar{x}^{(p)})) < \varepsilon \implies r(\bar{x}) < \widetilde{\varepsilon}$$
$$\varepsilon = \widetilde{C}(m, w)\widetilde{\varepsilon}$$

Proposition 1
Asynchronous iterations snapshot (AIS)

Non FIFO case

- *[Savari & Bertsekas, 1996]*
  Embed computation message into marker
  $\Rightarrow$ marker size: $\mathcal{O}(n)$

Proposition 2
Non FIFO Asynchronous iterations snapshot (NFAIS)

Non FIFO characterization
A marker can cross at most m computation messages

1. Send marker after m iterations under local convergence
2. Send flag-marker after m subsequent iterations
   $\rightarrow$ flag armed if continuous local convergence

Practical case
Marker transmits faster than computation message
$\Rightarrow$ no need for flag-marker

4 - Asynchronous convergence detection

Problem

$$f(x) = x, \quad x \in E$$

Asynchronous convergence detection
Approach

- Finite time termination *[Bertsekas & Tsitsiklis, 1989], [El Baz, 1996], [Savari & Bertsekas, 1996], ...*
- Snapshot-based supervised termination *[Savari & Bertsekas, 1996]*, NFAIS
- Predicted termination (finite number of iterations) *[Evans & Chikohora, 1998]*
- Local convergence monitoring *[Bahi et al, 2005, 2008]*
- Nested-sets-based supervised termination *[Miellou et al, 2008]*

	Intrusiveness	Centralization	Effectiveness	Messages size
Bertsekas & Tsitsiklis, 1989	altered iterations	–	formal	–
El Baz, 1996	altered iterations	–	formal	–
Savari & Bertsekas, 1996	altered iterations	–	formal	–
Savari & Bertsekas, 1996	non-intrusive	two reductions	exact	$\mathcal{O}(n)$
Evans & Chikohora, 1998	non-intrusive	no reduction	heuristic	0
Bahi *et al*, 2005	non-intrusive	one reduction	heuristic	$\mathcal{O}(1)$
Bahi *et al*, 2008	piggybacking	two reductions	heuristic	$\mathcal{O}(1)$
Miellou *et al*, 2008	–	–	formal	–
NFAIS	non-intrusive	one reduction	exact	$\mathcal{O}(1)$

4 - Asynchronous convergence detection

NFAIS: $m = 1$

NBS: Non-blocking synchronization (snapshot without local convergence condition)

	Synchronous			NBS		
p	$r \times 10^7$	wt	k	$r \times 10^7$	wt	k
168	8.33	701	281916	5.03	**536**	346226
240	8.31	516	284118	6.18	**378**	366231
360	8.33	382	287557	5.72	**250**	355394
480	8.32	302	289933	5.40	**202**	406611
600	8.32	278	292163	5.23	**168**	432390

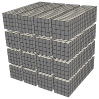
	Savari & Bertsekas, 1996			NFAIS		
p	$r \times 10^7$	wt	k	$r \times 10^7$	wt	k
168	6.55	641	319703	6.54	640	319349
240	6.52	462	342476	6.42	463	343295
360	6.71	310	335008	5.19	314	339204
480	6.43	249	380524	6.63	250	383745
600	6.55	207	404544	6.06	209	410621

Convection-diffusion

$$\frac{\partial u}{\partial t} - \nu \Delta u + a.\nabla u = s$$

Problem size

$$n = 185^3 = 6,331,625$$



Block-Jacobi splitting
+
Gauss-Seidel on blocks

Residual threshold

$$\varepsilon = 10^{-6}$$

p = number of processors
r = global residual after termination

OUTLINE

05 CONCLUSION

5 - Conclusion

- Asynchronous iterations theory
  → Contracting mappings
  → Matrix splittings, maximum norm and spectral radius

5 - Conclusion

- Asynchronous iterations theory
  → Contracting mappings
  → Matrix splittings, maximum norm and spectral radius
- Space domain decomposition
  → Primal approach
  → Two diagonal-free splittings of the interface Schur complement
  → Asynchronous convergence for vector and substructures-based decomposition
  → Journal article in progress

5 - Conclusion

- Asynchronous iterations theory
  → Contracting mappings
  → Matrix splittings, maximum norm and spectral radius
- Space domain decomposition
  → Primal approach
  → Two diagonal-free splittings of the interface Schur complement
  → Asynchronous convergence for vector and substructures-based decomposition
  → Journal article in progress
- Time domain decomposition
  → Parareal approach
  → PDE extension of ODE convergence condition
  → Asynchronous convergence for asymptotically equivalent condition
  → Journal article under review

5 - Conclusion

- Asynchronous iterations theory
  → Contracting mappings
  → Matrix splittings, maximum norm and spectral radius
- Space domain decomposition
  → Primal approach
  → Two diagonal-free splittings of the interface Schur complement
  → Asynchronous convergence for vector and substructures-based decomposition
  → Journal article in progress
- Time domain decomposition
  → Parareal approach
  → PDE extension of ODE convergence condition
  → Asynchronous convergence for asymptotically equivalent condition
  → Journal article under review
- Asynchronous convergence detection
  → Snapshot approach
  → Exact global residual evaluation
  → One reduction operation, data-free marker
  → Non-blocking synchronization
  → Peer-reviewed journal article
- ⇒ JACK2: MPI-based communication library
  → Peer-reviewed journal article, Proceedings
- And various other works and publications

5 - Conclusion

- Asynchronous iterations theory
  $\longrightarrow$ Contracting mappings
  $\longrightarrow$ Matrix splittings, maximum norm and spectral radius
- Space domain decomposition
  $\longrightarrow$ Primal approach
  $\longrightarrow$ Two diagonal-free splittings of the interface Schur complement
  $\longrightarrow$ Asynchronous convergence for vector and substructures-based decomposition
  $\longrightarrow$ Journal article in progress
- Time domain decomposition
  $\longrightarrow$ Parareal approach
  $\longrightarrow$ PDE extension of ODE convergence condition
  $\longrightarrow$ Asynchronous convergence for asymptotically equivalent condition
  $\longrightarrow$ Journal article under review
- Asynchronous convergence detection
  $\longrightarrow$ Snapshot approach
  $\longrightarrow$ Exact global residual evaluation
  $\longrightarrow$ One reduction operation, data-free marker
  $\longrightarrow$ Non-blocking synchronization
  $\longrightarrow$ Peer-reviewed journal article
- $\Longrightarrow$ JACK2: MPI-based communication library
  $\longrightarrow$ Peer-reviewed journal article, Proceedings
- And various other works and publications

Future work

- Substructures-based convergence of γI splitting
- Complete interface tearing
- Dual approach
- Coarse space correction

5 - Conclusion

- Asynchronous iterations theory
  → Contracting mappings
  → Matrix splittings, maximum norm and spectral radius
- Space domain decomposition
  → Primal approach
  → Two diagonal-free splittings of the interface Schur complement
  → Asynchronous convergence for vector and substructures-based decomposition
  → Journal article in progress
- Time domain decomposition
  → Parareal approach
  → PDE extension of ODE convergence condition
  → Asynchronous convergence for asymptotically equivalent condition
  → Journal article under review
- Asynchronous convergence detection
  → Snapshot approach
  → Exact global residual evaluation
  → One reduction operation, data-free marker
  → Non-blocking synchronization
  → Peer-reviewed journal article
- ⇒ JACK2: MPI-based communication library
  → Peer-reviewed journal article, Proceedings
- And various other works and publications

Future work

- Substructures-based convergence of γI splitting
- Complete interface tearing
- Dual approach
- Coarse space correction

Next goal
Habilitation à Diriger des Recherches

Thèse de Doctorat

Asynchronous domain decomposition methods for massively parallel computing

Préparée et soutenue par

Guillaume Gbikpi-Benissan

Devant le Jury composé de

Raphaël Couturier	(Rapporteur)
Fabienne Jezequel	(Examinatrice)
Pierre Spiteri	(Rapporteur)
Stéphane Vialle	(Examinateur)
Frédéric Magoulès	(Directeur de thèse)

18 décembre 2017