Parallel-in-time Algorithms for Data Assimilation Problems

Felix Kwok

Hong Kong Baptist University

Big Data Challenges for Predictive Modeling of Complex Systems

HKU, November 26, 2018
Collaborators

- Martin J. Gander (Geneva)
- Julien Salomon (Inria Paris)
- Sebastián Reyes Riffo (Paris-Dauphine)
Outline

Introduction

Assimilation over Finite Time

Assimilation over Infinite Time
Data assimilation

**Goal:** Given a physical model *with unknown or uncertain* parameters, use *observed data* to approximate the exact trajectory.

(Source: UK Met Office)
Example: Contaminant Tracking

**Problem:** Track the concentration of contaminants $y$, governed by the advection-diffusion equation

$$\dot{y} - \nabla \cdot (c \nabla y + b y) = Bu$$

in $\Omega$

+ initial and boundary conditions.
Example: Contaminant Tracking

**Variant I.** Unknown forcing function \( u \):

\[
\min_{u \in \mathcal{U}} \frac{\nu}{2} \int_0^T \| u \|^2 dt + \frac{1}{2} \int_0^T \| C y(t; u) - \hat{y} \|^2 dt
\]

subject to

\[
\frac{\partial y}{\partial t} - \nabla \cdot (c \nabla y + by) = B u
\]
Example: Contaminant Tracking
**Example: Contaminant Tracking**

**Variant II.** Unknown initial conditions $y(0)$:

$$\min_{v \in \mathcal{V}} \frac{1}{2} \|v - y(0)\|^2 + \frac{\gamma}{2} \int_0^T \|Cy(t; v) - \hat{y}\|^2 dt$$

subject to

$$\frac{\partial y}{\partial t} - \nabla \cdot (c \nabla y + b y) = Bu,$$

$$y(0) = v.$$
Variant III. What if $T \to \infty$?

$$\min_{v \in \mathcal{Y}} \frac{1}{2} \| v - y(0) \|^2 + \frac{\gamma}{2} \int_0^T \| Cy(t; v) - \hat{y} \|^2 dt$$

subject to

$$\frac{\partial y}{\partial t} - \nabla \cdot (c \nabla y + b y) = Bu,$$

$$y(0) = v.$$
Other Applications

- Weather forecasting
- Bio-medicine: Drug administration in chemotherapy (Jackson & Byrne 2000, Rockne et al. 2010, Corwin et al. 2013, ...)
- Oil & Gas: Oil field management optimization, data assimilation, history matching (Smart Field Consortium at Stanford, ...)
- ...
Challenges

- Huge system size, globally coupled
- Stiff nonlinear problems
- Real-time requirements
- ...
Parallel Computing

- Supercomputer with highest peak rate: Summit (ORNL, US, 2,397,824 cores, 200,794.9 Tflops/s)
- Supercomputer with largest number of cores: TaihuLight (China, 10,649,600 cores, 125,345.9 TFlops/s)
Optimality System (with tracking)

For the problem

$$
\min_u J[u] = \frac{1}{2} \int_0^T \|y(t) - y_d\|^2 \, dt + \frac{\nu}{2} \int_0^T \|u(t)\|^2 \, dt
$$

s.t. \quad \dot{y}(t) = f(y(t)) + u(t), \quad t \in (0, T).

A Lagrange multiplier argument shows that the optimality conditions are

$$
\begin{align*}
\dot{y} &= f(y) + \nu^{-1} \lambda, \\
y(0) &= y_{\text{init}},
\end{align*}
$$

$$
\begin{align*}
\dot{\lambda} &= -(f'(y))^T \lambda + y - y_d, \\
\lambda(T) &= 0.
\end{align*}
$$

Coupled forward-backward problem

Huge system after discretization!
Divide time horizon \((0, T)\) into “subdomains” \(I_i = (T_{i-1}, T_i)\)

Subdomain problem \((y_i(t), \lambda_i(t))\) on \(I_i\) well defined when \(y(T_{i-1})\) and \(\lambda(T_i)\) are given

Multiple shooting: solve for intermediate states \(Y_i = y(T_i)\) and \(\Lambda_i = \lambda(T_i)\) such that **continuity conditions** are satisfied:

\[
y_i(T_i) = y_{i+1}(T_i), \quad \lambda_i(T_{i+1}) = \lambda_{i+1}(T_{i+1})
\]
Decomposition in time

- Lagnese & Leugering (2003): Wave equation
- Heinkenschloss (2005): Parabolic problems
- Barker & Stoll (2013/15): Heat and Stokes equations
Optimized Schwarz for Control (Gander & K., DD22 proceedings)

For $k = 1, 2, \ldots$, solve on each $(\alpha_j, \beta_j)$

\[
\begin{align*}
\dot{y}_j^k & = f(y_j^k) + \nu^{-1} \lambda_j^k \\
\dot{\lambda}_j^k & = -(f'(y_j^k))^T \lambda_j^k + (y_j^k - y_d),
\end{align*}
\]

on $(\alpha_j, \beta_j)$,

with boundary conditions

\[
\begin{align*}
y_j^k(\alpha_j) - q_j \lambda_j^k(\alpha_j) & = y_{j-1}^{k-1}(\alpha_j) - q_j \lambda_{j-1}^{k-1}(\alpha_j), \\
\lambda_j^k(\beta_j) + p_j y_j^k(\beta_j) & = \lambda_{j+1}^{k-1}(\beta_j) + p_j y_{j+1}^{k-1}(\beta_j).
\end{align*}
\]
For \( p, q \neq 0 \), this is equivalent to

\[
\min \frac{1}{2} \int_{\alpha_j}^{\beta_j} \| y(t; u) - \hat{y} \|^2 + \nu \int_{\alpha_j}^{\beta_j} \| u \|^2 \\
+ \frac{p_j}{2} \| y(\beta_j; u) - p_j^{-1} g_{j+1}^{k-1} \|^2 + \frac{1}{2q_j} \| y(\alpha_j; u) - h_{j-1}^{k-1} \|^2
\]

where

\[
g_{j+1}^{k-1} = \lambda_{j+1}^{k-1}(\beta_j) + p_j y_{j+1}^{k-1}(\beta_j), \quad h_{j-1}^{k-1} = y_{j-1}^{k-1}(\alpha_j) - q_j \lambda_{j-1}^{k-1}(\alpha_j)
\]

- For \( p = q = 0 \), this reduces to Dirichlet transmission conditions.
For \( p, q \neq 0 \), this is equivalent to

\[
\min \left( \frac{1}{2} \int_{\alpha_j}^{\beta_j} \| y(t; u) - \hat{y} \|^2 + \frac{\nu}{2} \int_{\alpha_j}^{\beta_j} \| u \|^2 \right. \\
+ \frac{p_j}{2} \| y(\beta_j; u) - p_j^{-1} g_j^{k-1} \|^2 + \frac{1}{2q_j} \| y(\alpha_j; u) - h_j^{k-1} \|^2 
\]

where

\[
g_j^{k-1} = \lambda_j^{k-1}(\beta_j) + p_j y_j^{k-1}(\beta_j), \quad h_j^{k-1} = y_j^{k-1}(\alpha_j) - q_j \lambda_j^{k-1}(\alpha_j) 
\]

- For \( p = q = 0 \), this reduces to Dirichlet transmission conditions
- Minimization problem with small changes in boundary conditions \( \implies \) solvers available!
For $k = 1, 2, \ldots$, solve on each $(\alpha_j, \beta_j)$

\[
\begin{align*}
\dot{y}_j^k &= f(y_j^k) + \nu^{-1}\lambda_j^k \\
\dot{\lambda}_j^k &= -(f'(y_j^k))^T \lambda_j^k + (y_j^k - y_d),
\end{align*}
\]

on $(\alpha_j, \beta_j)$, with boundary conditions

\[
\begin{align*}
y_j^k(\alpha_j) - q_j \lambda_j^k(\alpha_j) &= y_{j-1}^{k-1}(\alpha_j) - q_j \lambda_{j-1}^{k-1}(\alpha_j), \\
\lambda_j^k(\beta_j) + p_j y_j^k(\beta_j) &= \lambda_{j+1}^{k-1}(\beta_j) + p_j y_{j+1}^{k-1}(\beta_j).
\end{align*}
\]

- Convergence for which values of $p_j$ and $q_j$?
- How to choose $p_j$ and $q_j$ to optimize convergence?
Dirichlet Case \((p = q = 0)\)

**Theorem (No uncertainties in ICs)**

For two subdomains with overlap \(L \geq 0\), the parallel Schwarz method for two subdomains converges with the estimate

\[
\rho \leq \frac{e^{-L\sqrt{d_{\text{min}}^2 + \nu - 1}}}{\sqrt{1 + \nu d_{\text{min}}^2 + \nu^{1/2} d_{\text{min}}}},
\]

where \(d_{\text{min}} > 0\) is the smallest eigenvalue of \(A\).

- Method converges even without overlap
- Convergence independent of the spatial mesh parameter!
Dirichlet Case \((p = q = 0)\)

- **Case A:** \(\Omega_1 = (0, 1), \Omega_2 = (1, 3), \gamma = 0\)
- **Case B:** \(\Omega_1 = (0, 2.9), \Omega_2 = (2.9, 3), \gamma = 10\)
Dirichlet Case \((p = q = 0)\)

- Case A converges for all positive definite matrices
- Convergence slow if \(d_{\min} \ll 1\)
- Case B diverges if \(d_{\min} \lesssim 2\) (e.g. Neumann boundary)
Dirichlet Case ($p = q = 0$)

- **Case A:** $\Omega_1 = (0, 1), \Omega_2 = (1, 3), \gamma = 0$
- **Case B:** $\Omega_1 = (0, 2.9), \Omega_2 = (2.9, 3), \gamma = 10$
Optimized case, $p = q$

- **Case A:** $\Omega_1 = (0, 1)$, $\Omega_2 = (1, 3)$, $\gamma = 0$
- **Case B:** $\Omega_1 = (0, 2.9)$, $\Omega_2 = (2.9, 3)$, $\gamma = 10$
- Convergence for all frequencies
Numerical Example 1

- Governing PDE: $u_t = u_{xx}$ in $(x, t) \in (0, 1) \times (0, 3)$
- Discretization: Crank–Nicolson with $h = 1/32$ and $h = 1/64$
- Dirichlet or Neumann boundary conditions in space
- Two temporal subdomains: $\Omega_1 = (0, 1)$, $\Omega_2 = (1, 3)$
Numerical Example 1

- Mesh independent convergence
- Optimized conditions beneficial for Neumann case
Numerical Example 2

- 2D advection-diffusion equation
- Flow field obtained by Stokes equation
- Source (control) at centre of domain, observation at one point on boundary
- 736 dof in space, 64 time steps
- \( T = 32 \), split into 2, 4, 8, 16 equal subdomains
- Transmission conditions: \( p = q = 0.8563 \)
Numerical Example 2
Outline

Introduction

Assimilation over Finite Time

Assimilation over Infinite Time
Infinite Time Horizon

Consider the linear ODE system

\[ \dot{y} = Ay + Bu, \quad t > 0 \]
\[ y(0) = y_0. \]

Suppose we do not know the exact initial conditions \( y_0 \), but we have the observations \( y_d(t) = Cy(t) \) for all time \( t \).

**Goal:** find an approximation \( \tilde{y}(t) \) of \( y(t) \) such that \( \|\tilde{y}(t) - y(t)\| \to 0 \) as \( t \to \infty \).
Define the approximate trajectory $\tilde{y}(t)$ with modified dynamics:

$$\frac{d\tilde{y}}{dt} = A\tilde{y} + Bu(t) + L(y_d - C\tilde{y})$$

$$\tilde{y}(0) = \tilde{y}_0,$$

where $\tilde{y}_0$ is an approximation for the exact initial conditions $y_0$.

If $y_d = Cy$ is sampled from the exact trajectory, then the solution $\tilde{y}$ satisfies

$$y(t) - \tilde{y}(t) = e^{(A-CL)t}(y(0) - \tilde{y}(0)).$$
Luenberger Observer

\[ y(t) - \tilde{y}(t) = e^{(A-LC)t}(y(0) - \tilde{y}(0)) \]

**Theorem**

Let the system be *completely observable*. Then if \( \{\mu_i\} \) correspond to a set of eigenvalues of an \( n \times n \) real matrix, then there is a choice of \( L \) such that \( A - LC \) have eigenvalues \( \{\mu_i\} \).

- Construct \( L \) such that all the eigenvalues of \( A - LC \) lie on the open left half plane \( \implies e^{(A-LC)t} \to 0 \) as \( t \to \infty \)
- Standard algorithms to design \( L \): Routh’s or Hurwitz criterion, Ackermann’s formula, etc.
Questions

1. How large must $T$ be in order for $\|y(t) - \tilde{y}(t)\|$ to be smaller than a given tolerance?
2. How to run the assimilation in parallel?
Questions

1. How large must $T$ be in order for $\|y(t) - \tilde{y}(t)\|$ to be smaller than a given tolerance?

2. How to run the assimilation in parallel?
   - Pure initial value problems
   - Algorithms: Parareal, Paraexp, diagonalization, ... (cf. Gander, *50 Years of Time Parallel Time Integration*)
Résolution d’EDP par un schéma en temps « pararéel »

Jacques-Louis LIONS a, Yvon MADAY b, Gabriel TURINICI b,c

Résumé. On propose dans cette Note un schéma permettant de profiter d’une architecture parallèle pour la discrétisation en temps d’une équation d’évolution aux dérivées partielles. Cette méthode, basée sur un schéma d’Euler, combine des résolutions grossières et des résolutions fines et indépendantes en temps en s’inspirant de ce qui est classique en espace. La parallélisation qui en résulte se fait dans la direction temporelle ce qui est en revanche non classique. Elle a pour principale motivation les problèmes en temps réel, d’où la terminologie proposée de « pararéel ». © 2001 Académie des sciences/Éditions scientifiques et médicales Elsevier SAS

A “parareal” in time discretization of PDE’s

Abstract. The purpose of this Note is to propose a time discretization of a partial differential evolution equation that allows for parallel implementations. The method, based on an Euler scheme, combines coarse resolutions and independent fine resolutions in time in the same spirit as standard spacial approximations. The resulting parallel implementation is done in the non standard time direction. Its main goal concerns real time problems, hence the proposed terminology of “parareal” algorithm. © 2001 Académie des sciences/Éditions scientifiques et médicales Elsevier SAS
Parareal (Lions, Maday & Turinici 2001)

- Decompose $[0, T]$ into $N$ subintervals
- Unknowns: $Y_n \approx y(t_n)$
- Fine/coarse propagators: $P^{\delta t}(t_{n+1}; t_n, y_n), P^{\Delta t}(t_{n+1}; t_n, y_n)$
- For $k = 1, 2, \ldots$:
  1. Solve fine problems in parallel:
     \[ y^{k}_{n+1}(t_{n+1}) = P^{\delta t}(t_{n+1}; t_n, Y^{k}_n) \]
  2. Correct initial conditions using coarse propagator:
     \[ Y^{k+1}_{n+1} = y^{k}_{n+1}(t_{n+1}) + P^{\Delta t}(t_{n+1}; t_n, Y^{k+1}_{n}) - P^{\Delta t}(t_{n+1}; t_n, Y^{k}_n) \]
- Initial guesses for $Y^0_n$ obtained by coarse propagation
Example: Brusselator

- Model equation:

\[
\frac{dx}{dt} = A + x^2 y - (B + 1)x \\
\frac{dy}{dt} = Bx - x^2 y
\]

- Parameters: \( A = 1, B = 3, \) \\
  \( x(0) = 0, y(0) = 1 \)

- Use 4th order Runge-Kutta with 32 coarse steps and 320 fine steps
Example: Brusselator

- Method is exact in the first \( k \) intervals after \( k \) iterations
- Speedup only if \( k \ll N \)
Example: Brusselator

- Method is exact in the first $k$ intervals after $k$ iterations
- Speedup only if $k \ll N$
Example: Brusselator

- Method is exact in the first $k$ intervals after $k$ iterations
- Speedup only if $k \ll N$
Example: Brusselator

- Method is exact in the first $k$ intervals after $k$ iterations
- Speedup only if $k \ll N$
Example: Brusselator

- Method is exact in the first $k$ intervals after $k$ iterations
- Speedup only if $k \ll N$
Example: Brusselator

- Method is exact in the first $k$ intervals after $k$ iterations
- Speedup only if $k \ll N$
Example: Brusselator

- Method is exact in the first $k$ intervals after $k$ iterations
- Speedup only if $k \ll N$
Questions

1. How large must $T$ be in order for $\|y(t) - \tilde{y}(t)\|$ to be smaller than a given tolerance?
2. How to run the assimilation in parallel?
   $\Rightarrow$ Parareal
Diamond Strategy

**Algorithm:**

1. Choose some initial assimilation window $[0, T_0]$
2. For $\ell = 1, 2, \ldots,$
   3. Perform assimilation in parallel using Parareal
   4. Estimate assimilation error on $\ell$th interval
      
      \[ \text{Err}_\ell := \| y(\ell T_0) - \tilde{y}(\ell T_0) \| \]

5. If $\text{Err}_\ell < \text{Tol}$, done.
   Else, continue assimilation over $[\ell T_0, (\ell + 1) T_0]$

**Question:** How many Parareal iterations $k_\ell$ to perform in Step 3?
Problem: Parareal error eventually dominates!
How to Choose $k_\ell$

- At each Parareal step, calculate jumps $J^k_\ell$ across subintervals
- Use the jumps to estimate the parareal error at step $k$
- Continue Parareal iterations until

$$\text{Parareal Error} \leq \gamma e^{-\mu \ell T_0}$$

where $\mu I + A - LC$ has spectrum in the open left half plane, so that

$$e^{-\mu \ell T_0} \approx \text{Assim. Error.}$$
Numerical Example

\[ \frac{d\tilde{y}}{dt} = A\tilde{y} + Bu(t) + L(y_d - C\tilde{y}), \]

with

\[ A = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad L = \begin{bmatrix} 0.8 \\ -1.1 \end{bmatrix}. \]

- Exact trajectory observed: \( y_d = Cy \)
- Known input: \( u(t) = 3 + 0.5 \sin(0.75t) \)
- Assimilation performed over time windows of length \( T = 5 \)
- Each assimilation done using parareal with 20 subintervals
Parallelization, variable $k_\ell$
Summary

- Finite time assimilation
  - Coupled forward backward problem
  - Domain decomposition in time for parallelization
  - Mesh independent convergence, even without overlap
  - Use of additional Robin parameters enhances convergence

- Infinite time assimilation
  - Initial value problem using Luenberger observers
  - Use parareal for parallel speedup
  - Adjust Parareal tolerance to match assimilation error
Thank you!