Two Time Parallel Algorithms for Optimality Systems

Norbert Tognon

Supervisor : Julien Salomon

INRIA Paris & Sorbonne University (France)

Laval University, Québec, Canada

Outline

Outline

Optimal control problem

- Let us consider the following Cauchy problem on $(0, T)$,

$$\begin{aligned} \dot{y}(t) - f\,(y(t)) &= \nu(t), \\ y(0) &= y_i. \end{aligned}$$

- $y, \nu \in \mathbb{R}^r$.
- y_i: the initial state,
- y_{tg}: the target state.

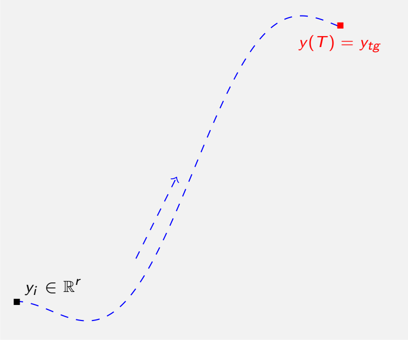
Assumption

We assume that this equation is controllable, .i.e, the application $\nu \longmapsto y(T)$ is surjective.

Optimal control problem

Find the optimal control ν such that

$$y(T) = y_{tg}.$$



Optimal control problem

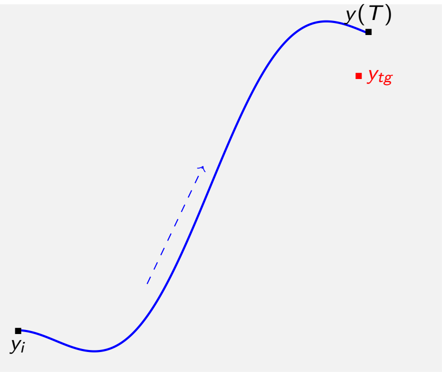
- The optimal control problem becomes the following minimization problem.

Minimization problem

$$\min_{\nu} \mathcal{J}(\nu) := \frac{1}{2}\|y(T)-y_{tg}\|^2+\frac{\alpha}{2}\int_0^T \|\nu\|^2(t)dt,$$

subject to

$$\dot{y}(t) - f(y(t)) = \nu(t), t \in (0, T)$$
$$y(0) = y_i.$$



Optimality system

- Using an adjoint variable λ, the Lagrange operator becomes

$$\mathfrak{L}(\nu, y, \lambda) = \mathcal{J}(\nu) - \int_0^T (\dot{y} - f(y) - \nu)^T \cdot \lambda dt.$$

- Taking $\nabla\mathfrak{L} = 0$, we get the optimality system

$$\begin{cases} \dot{y} - f(y) = \nu \\ y(0) = y_i, \end{cases} \qquad \begin{cases} \dot{\lambda} + \nabla f^T(y) \cdot \lambda = 0 \\ \lambda(T) = y(T) - y_{tg}, \end{cases}$$

$$\alpha\nu = -\lambda.$$

Optimality system

$$\begin{cases} \dot{y} = f(y) - \frac{1}{\alpha}\lambda, \\ \dot{\lambda} = -\nabla f^T(y) \cdot \lambda, \end{cases} \tag{1}$$

with the initial and final conditions $y(0) = y_i$ and $\lambda(T) = y(T) - y_{tg}$ respectively.

Outline

Parareal Algorithm

- Let us consider the following initial value problem

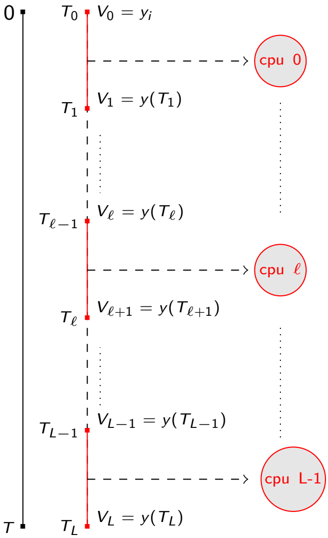
$$\begin{cases} \dot{y} = f(y) \quad t \in (0, T) \\ y(0) = y_i. \end{cases}$$

- Subdivision of $(0, T)$ into L sub-intervals:

$$T_0 = 0 < T_1 < \ldots, \cdots < T_L = T.$$

- Sub-problems on $(T_\ell, T_{\ell+1})$:

$$\begin{cases} \dot{y}_\ell = f(y_\ell) \\ y_\ell(T_\ell) = V_\ell, \end{cases}$$



Parareal algorithm

- Together with the matching condition

$$
\begin{aligned}
V_0 - y_i =& 0 \\
V_1 - y_0(T_1, V_0) =& 0 \\
&\vdots \\
V_L - y_{L-1}(T_L, V_{L-1}) =& 0.
\end{aligned}
$$

- We obtain the following nonlinear equation

$$
\Psi(V) := \begin{pmatrix} V_0 - y_i \\ V_1 - y_0(T_1, V_0) \\ V_2 - y_1(T_2, V_1) \\ \vdots \\ V_L - y_{L-1}(T_L, V_{L-1}) \end{pmatrix} = 0, \quad \text{where} \quad V = \begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ \vdots \\ V_L \end{pmatrix}.
$$

Parareal algorithm

- Newton method gives

$$J_\Psi(V^k)\left(V^{k+1}-V^k\right)=-\Psi(V^k),$$

where J_Ψ is the Jacobian matrix of Ψ, a triangular matrix. This becomes

$$\begin{aligned}V_0^{k+1}=&y_i,\\ V_{\ell+1}^{k+1}=&y_\ell(T_{\ell+1},V_\ell^k)+\frac{\partial y_\ell}{\partial_{V_\ell}}(T_{\ell+1},V_\ell^k)\left(V_\ell^{k+1}-V_\ell^k\right),\ \ell=0,\dots,L-1.\end{aligned}$$

- The Parareal idea

$$y_\ell(T_{\ell+1},V_\ell^k)=\mathcal{P}(V_\ell^k),$$

$\mathcal{P}$ is a solution propagator on sub-intervals,

$$\frac{\partial y_\ell}{\partial_{V_\ell}}(T_{\ell+1},V_\ell^k)\left(V_\ell^{k+1}-V_\ell^k\right)\approx\mathcal{G}(V_\ell^{k+1})-\mathcal{G}(V_\ell^k),$$

$\mathcal{G}$ is a solution propagator cheaper than $\mathcal{P}$.

- Parareal algorithm, [J.-L. Lions et al, 2001], [M. Gander et al, 2007]

$$\begin{aligned}V_0^{k+1}=&y_i,\\ V_{\ell+1}^{k+1}=&\mathcal{P}(V_\ell^k)+\mathcal{G}(V_\ell^{k+1})-\mathcal{G}(V_\ell^k),\ \ \ell=0,\dots,L-1.\end{aligned}$$

Outline

ParaOpt algorithm

Optimality system

Solve on $(0, T)$ the couple of equations

$$\begin{cases} \dot{y} = f(y) - \frac{1}{\alpha}\lambda, \\ y(0) = y_i, \end{cases} \qquad \begin{cases} \dot{\lambda} = -\nabla f^T(y) \cdot \lambda, \\ \lambda(T) = y(T) - y_{tg}. \end{cases}$$

using Parareal idea.

ParaOpt algorithm

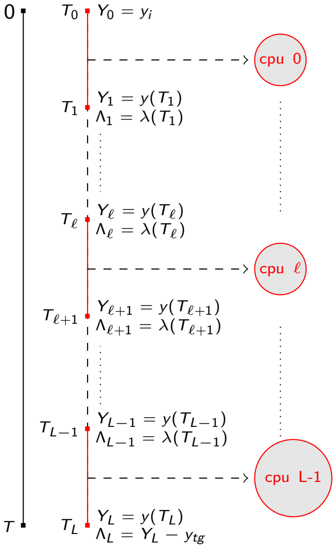
- Partition of $(0, T)$ into $(T_\ell, T_{\ell+1})$, $T_0 = 0$, $T_\ell = \ell \Delta T$ and $\ell = 0, \dots, L-1$.

- Subproblem notation on $(T_\ell, T_{\ell+1})$

$$\begin{cases} \dot{y}_\ell = f(y_\ell) - \frac{1}{\alpha}\lambda_\ell \\ y_\ell(T_\ell) = Y_\ell, \end{cases}$$

and

$$\begin{cases} \dot{\lambda}_\ell = -\nabla f^T(y_\ell) \cdot \lambda_\ell \\ \lambda_\ell(T_{\ell+1}) = \Lambda_{\ell+1} \end{cases}$$



ParaOpt algorithm

- Introduce the solution operators to solve the subproblems.

$$
\begin{aligned}
y(T_{\ell+1}) =& \mathcal{P}(Y_\ell, \Lambda_{\ell+1}) \\
\lambda(T_\ell) =& \mathcal{Q}(Y_\ell, \Lambda_{\ell+1}).
\end{aligned}
$$

- The solutions must match at interfaces, which lead to the equations

$$
\begin{array}{rcl|rcl}
Y_0 - y_i & = & 0 & & & \\
Y_1 - \mathcal{P}(Y_0, \Lambda_1) & = & 0 & \Lambda_1 - \mathcal{Q}(Y_1, \Lambda_2) & = & 0 \\
Y_2 - \mathcal{P}(Y_1, \Lambda_2) & = & 0 & \Lambda_2 - \mathcal{Q}(Y_2, \Lambda_3) & = & 0 \\
& \vdots & & & \vdots & \\
Y_L - \mathcal{P}(Y_{L-1}, \Lambda_L) & = & 0 & \Lambda_L - Y_L + y_{tg} & = & 0.
\end{array}
$$

ParaOpt algorithm

- Collecting the unknowns in the vector X we obtain the nonlinear system

$$\mathcal{F}(X) := \begin{pmatrix} Y_0 - y_i \\ Y_1 - \mathcal{P}(Y_0, \Lambda_1) \\ Y_2 - \mathcal{P}(Y_1, \Lambda_2) \\ \vdots \\ Y_L - \mathcal{P}(Y_{L-1}, \Lambda_L) \\ \Lambda_1 - \mathcal{Q}(Y_1, \Lambda_2) \\ \Lambda_2 - \mathcal{Q}(Y_2, \Lambda_3) \\ \vdots \\ \Lambda_L - Y_L + y_{tg} \end{pmatrix} = 0, \quad X := \begin{pmatrix} Y_0 \\ \vdots \\ \vdots \\ Y_L \\ \hline \Lambda_1 \\ \vdots \\ \vdots \\ \Lambda_L \end{pmatrix}. \tag{2}$$

ParaOpt algorithm

Newton method

$$\mathcal{F}'\left(X^k\right)\left(X^{k+1}-X^k\right)=-\mathcal{F}\left(X^k\right),$$

where

$$\mathcal{F}'(X) = \left(\begin{array}{cccc|cccc}
I & & & & & & & \\
-\mathcal{P}_y(Y_0,\Lambda_1) & I & & & -\mathcal{P}_\lambda(Y_0,\Lambda_1) & & & \\
 & -\mathcal{P}_y(Y_1,\Lambda_2) & I & & & \mathcal{P}_\lambda(Y_1,\Lambda_2) & & \\
 & & \ddots & \ddots & & & \ddots & \\
 & & -\mathcal{P}_y(Y_{L-1},\Lambda_L) & I & & & & -\mathcal{P}_\lambda(Y_{L-1},\Lambda_L) \\
\hline
 & -\mathcal{Q}_y(Y_1,\Lambda_2) & & & I & -\mathcal{Q}_\lambda(Y_1,\Lambda_2) & & \\
 & & \ddots & & & \ddots & \ddots & \\
 & & -\mathcal{Q}_y(Y_{L-1},\Lambda_L) & & & & I & -\mathcal{Q}_\lambda(Y_{L-1},\Lambda_L) \\
 & & & -I & & & & I
\end{array}\right).$$

ParaOpt algorithm

- Like in Parareal the remaining expensive fine grid $\mathcal{P}(Y_\ell, \Lambda_{\ell+1})$ and $\mathcal{Q}(Y_\ell, \Lambda_{\ell+1})$ can now all be performed in parallel.
- Parareal Idea,

$$\begin{aligned}
\mathcal{P}_y(Y_\ell^k, \Lambda_{\ell+1}^k)(Y_\ell^{k+1} - Y_\ell^k) &\approx \mathcal{P}^G(Y_\ell^{k+1}, \Lambda_{\ell+1}^k) - \mathcal{P}^G(Y_\ell^k, \Lambda_{\ell+1}^k),\\
\mathcal{P}_\lambda(Y_\ell^k, \Lambda_{\ell+1}^k)(\Lambda_{\ell+1}^{k+1} - \Lambda_{\ell+1}^k) &\approx \mathcal{P}^G(Y_\ell^k, \Lambda_{\ell+1}^{k+1}) - \mathcal{P}^G(Y_\ell^k, \Lambda_{\ell+1}^k),\\
\mathcal{Q}_\lambda(Y_\ell^k, \Lambda_{\ell+1}^k)(\Lambda_{\ell+1}^{k+1} - \Lambda_{\ell+1}^k) &\approx \mathcal{Q}^G(Y_\ell^k, \Lambda_{\ell+1}^{k+1}) - \mathcal{Q}^G(Y_\ell^k, \Lambda_{\ell+1}^k),\\
\mathcal{Q}_y(Y_\ell^k, \Lambda_{\ell+1}^k)(Y_\ell^{k+1} - Y_\ell^k) &\approx \mathcal{Q}^G(Y_\ell^{k+1}, \Lambda_{\ell+1}^k) - \mathcal{Q}^G(Y_\ell^k, \Lambda_{\ell+1}^k).
\end{aligned}$$

ParaOpt algorithm

- Derivative Parareal idea [Gander & Hairer 2014]:

$$\mathcal{P}_y(Y_\ell^k, \Lambda_{\ell+1}^k)(Y_\ell^{k+1} - Y_\ell^k) \approx \mathcal{P}_y^G(Y_\ell^k, \Lambda_{\ell+1}^k)(Y_\ell^{k+1} - Y_\ell^k),$$
$$\mathcal{P}_\lambda(Y_\ell^k, \Lambda_{\ell+1}^k)(\Lambda_{\ell+1}^{k+1} - \Lambda_{\ell+1}^k) \approx \mathcal{P}_\lambda^G(Y_\ell^k, \Lambda_{\ell+1}^k)(\Lambda_{\ell+1}^{k+1} - \Lambda_{\ell+1}^k),$$
$$\mathcal{Q}_\lambda(Y_\ell^k, \Lambda_{\ell+1}^k)(\Lambda_{\ell+1}^{k+1} - \Lambda_{\ell+1}^k) \approx \mathcal{Q}_\lambda^G(Y_\ell^k, \Lambda_{\ell+1}^k)(\Lambda_{\ell+1}^{k+1} - \Lambda_{\ell+1}^k),$$
$$\mathcal{Q}_y(Y_\ell^k, \Lambda_{\ell+1}^k)(Y_\ell^{k+1} - Y_\ell^k) \approx \mathcal{Q}_y^G(Y_\ell^k, \Lambda_{\ell+1}^k)(Y_\ell^{k+1} - Y_\ell^k).$$

  - Application to the Jacobian $\mathcal{F}'$ by solving linear subproblems in parallel using a coarse solver.
  - $\mathcal{J}^G$ is the coarse approximation of $\mathcal{F}'$.

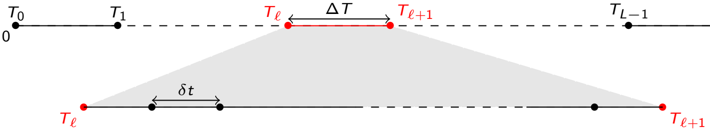
Paraopt algorithm

$$\mathcal{J}^G\left(X^k\right)\left(X^{k+1} - X^k\right) = -\mathcal{F}\left(X^k\right).$$

⟶ ***ParaOpt*** *: M. Gander, F. Kwok, J. Salomon, ParaOpt: A Parareal algorithm for optimal control systems, 2020.*

ParaOpt Algorithm

- Let $\delta t = T/M_0$, $\Delta T = M\delta t$, $M_0 = ML$ and $t_n = n\delta t, n = 0, \ldots, M_0$.



- Runge-Kutta method (A, b^T, c) given by the following Butcher table

$$
\begin{array}{c|cccc}
c_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
c_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
c_n & a_{n,1} & a_{n,2} & \cdots & a_{n,n} \\
\hline
 & b_1 & b_2 & \cdots & b_n
\end{array}
$$

ParaOpt algorithm

- Discrete cost functional becomes

$$\mathfrak{J}_{\delta t}(\nu) = \frac{1}{2}\left\|y_{M_0} - y_{tg}\right\|^2 + \frac{\alpha}{2}\delta t \sum_{n=0}^{M_0-1}\sum_{j=1}^{s} b_j \left\|\nu_{n,j}\right\|^2, \tag{3}$$

with $y_{M_0} = y(T)$ and $\nu_{n,j} = \nu(t_n + c_j \delta t)$.

- Discrete constraint

$$\dot{y} = f(y) + \nu \Longrightarrow y_{n+1} = y_n + \delta t \sum_{j=1}^{s} b_j g_j$$

where

$$g_i = f\left(y_n + \delta t \sum_{j=1}^{s} a_{i,j} g_j\right) + \nu_{n,i}.$$

ParaOpt Algorithm

- We consider the linear dynamic

$$\dot{y}(t) = \mathcal{L}y(t) + \nu(t).$$

- The discrete optimality system becomes

$$\begin{array}{rcl|rcl}
Y_0 - y_i & = & 0 & & & \\
Y_1 - \mathcal{P}(Y_0, \Lambda_1) & = & 0 & \Lambda_1 - \mathcal{Q}(Y_1, \Lambda_2) & = & 0 \\
Y_2 - \mathcal{P}(Y_1, \Lambda_2) & = & 0 & \Lambda_2 - \mathcal{Q}(Y_2, \Lambda_3) & = & 0 \\
 & \vdots & & & \vdots & \\
Y_L - \mathcal{P}(Y_{L-1}, \Lambda_L) & = & 0 & \Lambda_L - Y_L + y_{tg} & = & 0.
\end{array}$$

$$\begin{aligned}
\mathcal{P}(Y_\ell, \Lambda_{\ell+1}) =& \mathcal{S}_{\delta t} Y_\ell - \frac{1}{\alpha}\mathcal{R}_{\delta t}\Lambda_{\ell+1} \\
\mathcal{Q}(Y_\ell, \Lambda_{\ell+1}) =& \mathcal{S}_{\delta t}^T \Lambda_{\ell+1}.
\end{aligned}$$

- $\mathcal{S}_{\delta t}$ is the approximation of the exponential propagator from T_ℓ to $T_{\ell+1}$. And $\mathcal{R}_{\delta t}$ the approximation of the integral of the product between exponential propagator and its transpose over $(T_\ell, T_{\ell+1})$.

Discrete optimality system

- The function $\mathcal{F}_{\delta t}(X) := \mathcal{M}_{\delta t} X - \mathbf{f} = 0$;

$$\mathcal{M}_{\delta t} = \left(\begin{array}{cccc|cccc} I & & & & 0 & & & \\ -\mathcal{S}_{\delta t} & \ddots & & & \mathcal{R}_{\delta t}/\alpha & \ddots & & \\ & \ddots & \ddots & & & \ddots & & 0 \\ & & -\mathcal{S}_{\delta t} & I & & & & \mathcal{R}_{\delta t}/\alpha \\ \hline & & & & I & -\mathcal{S}_{\delta t}^T & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \ddots & -\mathcal{S}_{\delta t}^T \\ & & & -I & & & & I \end{array}\right), \quad \mathbf{f} = \begin{pmatrix} y_i \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ -y_{tg} \end{pmatrix}.$$

ParaOpt algorithm

Theorem

Let ΔT, δt be fixed and p the order of the Runge-Kutta method. Then the solution $X = \begin{pmatrix} Y \\ \Lambda \end{pmatrix}$ of the linear system $\mathcal{M}_{\delta t} X = \mathbf{f}$ satisfies

$$\|Y_\ell - y(T_\ell)\| \leq c_Y \delta t^p \quad \text{and} \quad \|\Lambda_\ell - \lambda(T_\ell)\| \leq c_\Lambda \delta t^p, \quad \ell = 1, \ldots, L.$$

The constants $c_Y > 0$ and $c_\Lambda > 0$ are independent on δt.

- We introduce the coarse time step $\Delta t = \Delta T / N$ with $\delta t \leq \Delta t \leq \Delta T$.

ParaOpt algorithm

$$\mathcal{M}_{\Delta t}\left(X^{k+1} - X^k\right) = -\left(\mathcal{M}_{\delta t} X^k - \mathbf{f}\right) \Longleftrightarrow X^{k+1} = \left(I - \mathcal{M}_{\Delta t}^{-1} \mathcal{M}_{\delta t}\right) X^k + \mathcal{M}_{\Delta t}^{-1} \mathbf{f}.$$

ParaOpt algorithm

ParaOpt algorithm

$$\mathcal{M}_{\Delta t}^{-1}\mathcal{M}_{\delta t}X = \mathcal{M}_{\Delta t}^{-1}\mathbf{f}.$$

- Implicit Euler method $(A, b^T, c) = (1, 1, 1)$.

Theorem (Kwok et al)

Let $\Delta T, \Delta t, \delta t$ and α be fixed. Then for all $\mathcal{L} < 0$, the spectral radius of $\left(I - \mathcal{M}_{\Delta t}^{-1}\mathcal{M}_{\delta t}\right)$ satisfies

$$\max_{\mathcal{L}<0} \rho(\mathcal{L}) \leq \frac{0.79\Delta t}{\alpha + \sqrt{\alpha \Delta t}} + 0.3.$$

Thus, if $\alpha > 0.4544\Delta t$, then the linear Paraopt algorithm converges.

ParaOpt algorithm

- More general case.

Stability condition

We assume that the dynamic and the Runge-Kutta method satisfy

$$\|\mathcal{S}_{\delta t}\| \leq 1.$$

Theorem (N. et al)

Let $\Delta T, \Delta t, \delta t$ be given and p the order of the Runge-Kutta method. Then the spectral radius of the iteration matrix $\left(I - \mathcal{M}_{\Delta t}^{-1}\mathcal{M}_{\delta t}\right)$ satisfies

$$\rho \leq \frac{c_\rho}{\Delta T}(1+\alpha^{-1})(\Delta t - \delta t)\Delta t^{p-1}$$

where c_ρ is a positive constant.

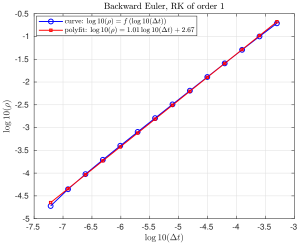
ParaOpt algorithm

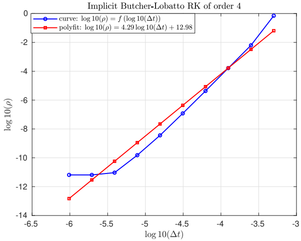
- The convergence factor satisfies

$$\rho \leq c_\rho^0 \Delta t^p$$

- Implicit Euler method
  $A = 1,\ b = 1,\ c = 1$ of order 1 on top.
- Butcher-Lobatto RK method order 4 on bottom

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 \\ 1 & 0 & 1 & 0 \\ \hline & 1/6 & 2/3 & 1/6 \end{array}.$$





Outline

ParaExp algorithm

- Let us consider here the evolution equation

$$\dot{y}(t) = \mathcal{L}y(t) + \nu(t), \ \ y(0) = y_i \ \text{ on } \ (0, T).$$

- We introduce a partitioning of $(0, T)$ in L time intervals $(T_{\ell-1}, T_\ell)$ with $\ell = 1, \dots, L$.
- Sub-problems

  In-homogeneous sub-problems :

$$\dot{z}_\ell(t) = \mathcal{L}z_\ell(t) + \nu(t), \ \ z_\ell(T_{\ell-1}) = 0, \ \text{ on } \ (T_{\ell-1}, T_\ell),$$

  Homogeneous sub-problems :

$$\dot{u}_\ell(t) = \mathcal{L}u_\ell(t), \ \ u_\ell(T_{\ell-1}) = z_{\ell-1}(T_{\ell-1}), \ \text{ on } \ (T_{\ell-1}, T_L),$$

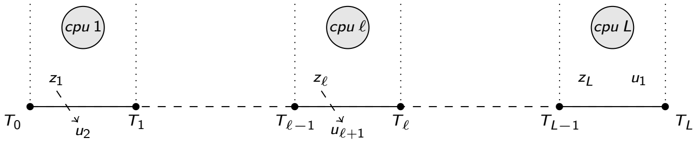
for $\ell = 1, \ u_1(T_0) = y_i$.

- Solution becomes

$$y_\ell(t) = z_\ell(t) + \sum_{j=1}^{\ell} u_j(t), \ \ t \in (T_{\ell-1}, T_\ell).$$

ParaExp algorithm

- The parallel in time idea:



- Approximation of the exponential: projection based methods and expansion based methods [Güttel and Gander, 2013].

Goal

Solve on $(0, T)$ the couple of equations

$$\begin{cases} \dot{y} = \mathcal{L}y - \frac{1}{\alpha}\lambda, \\ y(0) = y_i, \end{cases} \qquad \begin{cases} \dot{\lambda} = -\mathcal{L}^T \cdot \lambda, \\ \lambda(T) = y(T) - y_{tg}. \end{cases}$$

using ParaExp idea.

Outline

Time parallel algorithm based ParaExp

- Let $Y_\ell = y(T_\ell)$ and $\Lambda_\ell = \lambda(T_\ell)$, $\ell = 1, \ldots, L$.

Optimality System

$$\begin{cases} \dot{y} = \mathcal{L}y - \frac{1}{\alpha}\lambda, & \text{on } (0,T) \\ y(0) = y_i, \end{cases} \qquad \begin{cases} \dot{\lambda} = -\mathcal{L}^T \cdot \lambda, & \text{on } (0,T). \\ \lambda(T) = \Lambda_L \end{cases}$$

and $\Lambda_L = Y_L - y_{tg}$.

Time parallel formulation

In-homogeneous sub-problems :

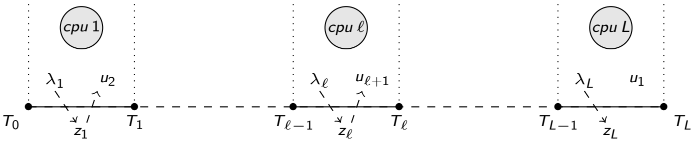
$$\dot{z}_\ell(t) = \mathcal{L}z_\ell(t) - \frac{1}{\alpha}\lambda_\ell(t), \;\; z_\ell(T_{\ell-1}) = 0, \;\; \text{on } (T_{\ell-1}, T_\ell),$$

Homogeneous sub-problems :

$$\begin{aligned} \dot{u}_\ell(t) =& \mathcal{L}u_\ell(t), \;\; u_\ell(T_{\ell-1}) = z_{\ell-1}(T_{\ell-1}), \;\; \text{on } (T_{\ell-1}, T_L), \\ \dot{\lambda}_\ell(t) =& -\mathcal{L}^T\lambda_\ell, \;\; \lambda_\ell(T_L) = \Lambda_L \;\; \text{on } (T_{\ell-1}, T_L), \end{aligned}$$

for $\ell = 1, \; u_1(T_0) = y_i$.

Time parallel algorithm based ParaExp



- Let $\mathcal{P}$ and $\mathcal{Q}$ be the exponential propagator of $\mathcal{L}$ and $-\mathcal{L}^T$ respectively.

$$\begin{aligned} u_\ell(t) =& \mathcal{P}_\ell(t) \cdot z_{\ell-1}(T_{\ell-1}), \text{ on } (T_{\ell-1}, T_L), \\ \lambda_\ell(t) =& \mathcal{Q}_\ell(t) \cdot \Lambda_L \text{ on } (T_{\ell-1}, T_L). \end{aligned}$$

- Denote by $\mathcal{R}$ defining by

$$z_\ell(t) = -\frac{1}{\alpha} \mathcal{R}_\ell(t) \cdot \Lambda_L \text{ on } (T_{\ell-1}, T_L).$$

Time parallel algorithm based ParaExp

Time parallel formulation

$$
\begin{aligned}
\lambda_\ell(t) =& \mathcal{Q}_\ell(t) \cdot \Lambda_L \quad \text{on} \quad (T_{\ell-1}, T_L) \\
z_\ell(t) =& -\frac{1}{\alpha}\mathcal{R}_\ell(t) \cdot \Lambda_L, \quad \text{on} \quad (T_{\ell-1}, T_\ell) \\
u_1(t) =& \mathcal{P}_1(t) \cdot y_i \quad \text{on} \quad (T_0, T_L), \\
u_\ell(t) =& -\frac{1}{\alpha}\mathcal{P}_\ell(t)\mathcal{R}_{\ell-1}(T_{\ell-1}) \cdot \Lambda_L \quad \text{on} \quad (T_{\ell-1}, T_L).
\end{aligned}
$$

Optimality system

$$
\begin{aligned}
Y_\ell =& \mathcal{P}_1(T_\ell) \cdot y_i - \frac{1}{\alpha}\left(\mathcal{R}_\ell(T_\ell) + \sum_{j=2}^{\ell} \mathcal{P}_j(T_\ell)\mathcal{R}_{j-1}(T_{j-1})\right) \cdot \Lambda_L, \quad \ell = 1, \dots, L, \\
\Lambda_\ell =& \mathcal{Q}_\ell(T_\ell) \cdot \Lambda_L, \quad \ell = 1, \dots, L-1 \\
\Lambda_L =& Y_L - y_{tg}.
\end{aligned}
$$

Time parallel algorithm based ParaExp

Algorithm

- Solve the linear

$$\mathcal{M} \cdot \Lambda_L - \mathbf{f} = 0,$$

where

$$\mathcal{M} := \left(I + \frac{1}{\alpha}\mathcal{R}_L(T_L) + \frac{1}{\alpha}\sum_{j=2}^{L}\mathcal{P}_j(T_L)\mathcal{R}_{j-1}(T_{j-1}) \right),$$

$$\mathbf{f} := \mathcal{P}_1(T_L) \cdot y_i - y_{tg}.$$

- Update the trajectory of the state Y_ℓ.

$$Y_\ell = \mathcal{P}_1(T_\ell) \cdot y_i - \frac{1}{\alpha}\left(\mathcal{R}_\ell(T_\ell) + \sum_{j=2}^{\ell}\mathcal{P}_j(T_\ell)\mathcal{R}_{j-1}(T_{j-1}) \right) \cdot \Lambda_L, \;\; \ell = 1, \dots, L.$$

Outline

Conclusion and Ongoing works

- Conclusion
  - Time parallel algorithm based Parareal: ParaOpt,
  - Some convergence results using Runge-Kutta method for a linear dynamic.
  - Time parallel algorithm based ParaExp.
- Ongoing works
  - Preconditioner for Time parallel algorithm based ParaExp.
  - Time parallel algorithm based ParaExp for the following optimality systems

$$\begin{cases} \dot{y} = \mathcal{L}y - \frac{1}{\alpha}\lambda \quad \text{on } (0,T), \\ y(0) = y_0, \end{cases} \qquad \begin{cases} \dot{\lambda} = -\mathcal{L}^T\lambda + (y - \hat{y}) \quad \text{on } (0,T) \\ \lambda(T) = -\gamma(y(T) - \hat{y}(T)). \end{cases}$$

  - Preconditioner to speed-up that algorithm.

References

- **M. Gander and S. Vandewalle, 2007**: Analysis of the Parareal, time-parallel time integration method.
- **M.Gander,F. Kwok and J. Salomon, 2020**: ParaOpt, a Parareal algorithm for optimality systems.
- **S. Güttel and M. Gander, 2013**: A parallel in time integrator for linear initial-value problem.
- **F. Kwok, J. Salomon, N. (in print)**: Convergence of the ParaOpt Algorithm for general Runge-Kutta time discretizations.

THANK YOU!