Time Parallelization and Optimal Control : Paraopt

Norbert Tognon

INRIA Paris

RJCAF (Cinquième Édition)

1. Parareal Algorithm

2. Optimal control problem

3. Paraopt Algorithm

Parareal Algorithm

Let us consider the following initial value problem

$$\begin{cases} \dot{y}(t) = f(y(t)) \;\; t \in [0; T] \\ y(0) = y^0 \end{cases} \tag{1}$$

- We consider the subdivision of the time interval $[0; T]$ into L subintervals:

$$T_0 = 0 < T_1 < \dots, \dots < T_L = T.$$

- Solving the original problem on the subintervals,

$$\begin{cases} \dot{y}_\ell = f(y_\ell(t)) \\ y_\ell(T_\ell) = Y_\ell \end{cases} \quad t \in [T_\ell; T_{\ell+1}],\; \ell = 0, 1, \dots, L-1,$$

Paraeal Algorithm

with the matching condition which satisfies

$$\begin{cases} Y_{\ell+1} = \mathcal{F}(Y_\ell) \\ Y_0 = y^0 \end{cases} \quad \ell = 0, \ldots, L-1.$$

- For $Y = \left(Y_0^T, Y_1^T, \ldots, Y_L^T\right)^T$, the matching condition becomes

$$\Psi(Y) := \begin{pmatrix} Y_0 - y^0 \\ Y_1 - \mathcal{F}(Y_0) \\ Y_2 - \mathcal{F}(Y_1) \\ \vdots \\ Y_L - \mathcal{F}(Y_{L-1}) \end{pmatrix} = 0.$$

- $\mathcal{F}$ the solution operator propagates y from T_ℓ to $T_{\ell+1}$.

Parareal Algorithm

- Newton method

$$\Psi'\left(Y^k\right)\left(Y^{k+1}-Y^k\right)=-\Psi(Y^k).$$

- Componentwise

$$\begin{aligned} Y_0^{k+1} &= y^0 \\ Y_{\ell+1}^{k+1} &= \mathcal{F}\left(Y_\ell^k\right)+\partial_{Y_\ell}\mathcal{F}\left(Y_\ell^k\right)\left(Y_\ell^{k+1}-Y_\ell^k\right),\ \ell=0,\dots,L-1. \end{aligned}$$

- The parareal algorithm reads

$$\begin{aligned} Y_0^{k+1} &= y^0 \\ Y_{\ell+1}^{k+1} &= \mathcal{F}\left(Y_\ell^k\right)+\mathcal{G}\left(Y_\ell^{k+1}\right)-\mathcal{G}\left(Y_\ell^k\right),\ \ell=0,\dots,L-1. \end{aligned}$$

- $\mathcal{F}$ is more accurate than $\mathcal{G}$.

M. Gander and S. Vandewalle "Analysis of the Parareal time-parallel time-integration method", SIAM 2007

Optimal control problem

Problem: control on a fixed, bounded interval $[0, T]$.
Cost functional

$$J(c) = \frac{1}{2}\|y(T) - y_{target}\|^2 + \frac{\alpha}{2}\int_0^T c^2(t)dt,$$

- α: a fixed regularization parameter;
- y_{target}: the target state;
- y: state function is described by the equation

$$\begin{cases} \dot{y}(t) = f(y(t)) + c(t), t \in [0; T] \\ y(0) = y^0. \end{cases} \tag{2}$$

Optimal control problem

- Define the *Lagrange operator*

$$\mathcal{L}(y, \lambda, c) = J(c) - \int_0^T \langle \lambda(t), (\dot{y}(t) - f(y(t)) - c(t)) \rangle \, dt.$$

- The optima are characterized by the Euler-Lagrange equations $\nabla \mathcal{L} = 0$.

$\rightarrow$ **Elimination of c:**

$$\begin{cases} \dot{y} = f(y) - \dfrac{\lambda}{\alpha}, & t \in [0, T] \\ \dot{\lambda} = -f'(y)^T \lambda, \end{cases} \quad \text{with} \quad \begin{cases} y(0) = y^0 \\ \lambda(T) = y(T) - y_{target}. \end{cases}$$

Paraopt Algorithm

- Boundary value problems notations

$$\begin{cases} \dot{y}_\ell = f(y_\ell) - \frac{\lambda_\ell}{\alpha} \\ \dot{\lambda}_\ell = -f'(y_\ell)^T \lambda_\ell. \end{cases} \quad \text{with} \quad \begin{cases} y(T_\ell) = Y_\ell \\ \lambda(T_{\ell+1}) = \Lambda_{\ell+1} \end{cases}$$

  on the subinterval $[T_\ell, T_{\ell+1}]$.

- We denote

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

Paraopt Algorithm

- The optimality system is enriched:

$$
\begin{array}{rclrcl}
Y_0 - y^0 & = & 0 & & & \\
Y_1 - \mathcal{P}(Y_0, \Lambda_1) & = & 0 & \Lambda_1 - \mathcal{Q}(Y_1, \Lambda_2) & = & 0 \\
Y_2 - \mathcal{P}(Y_1, \Lambda_2) & = & 0 & \Lambda_2 - \mathcal{Q}(Y_2, \Lambda_3) & = & 0 \\
& \vdots & & & \vdots & \\
Y_L - \mathcal{P}(Y_{L-1}, \Lambda_L) & = & 0 & \Lambda_L - Y_L + Y_{target} & = & 0
\end{array}
\tag{3}
$$

That is : **a system of boundary value subproblems, satisfying matching conditions**.

Paraopt Algorithm

- We obtain the nonlinear equation

$$\Phi \begin{pmatrix} Y \\ \Lambda \end{pmatrix} := \begin{pmatrix} Y_0 - y_{init} \\ Y_1 - \mathcal{P}(Y_0, \Lambda_1) \\ Y_2 - \mathcal{P}(Y_1, \Lambda_2) \\ \vdots \\ Y_L - \mathcal{P}(Y_{L-1}, \Lambda_L) \\ \Lambda_1 - \mathcal{Q}(Y_1, \Lambda_2) \\ \Lambda_2 - \mathcal{Q}(Y_2, \Lambda_3) \\ \vdots \\ \Lambda_L - Y_L + y_{target} \end{pmatrix} = 0, \quad \text{where} \begin{pmatrix} Y \\ \Lambda \end{pmatrix} = \begin{pmatrix} Y_0 \\ \vdots \\ Y_L \\ \Lambda_1 \\ \vdots \\ \Lambda_L \end{pmatrix}.$$

Paraopt Algorithm

- Newton's method:

$$\Phi'\begin{pmatrix} Y^k \\ \Lambda^k \end{pmatrix}\begin{pmatrix} Y^{k+1}-Y^k \\ \Lambda^{k+1}-\Lambda^k \end{pmatrix} = -\Phi\begin{pmatrix} Y^k \\ \Lambda^k \end{pmatrix},$$

where the Jacobian matrix of Φ is given by

$$\Phi'\begin{pmatrix} Y \\ \Lambda \end{pmatrix} =$$

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

Paraopt Algorithm

- Coarse approximation of the Jacobian using "Derivative Parareal" idea

Which concretely corresponds to:

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

- $\mathcal{P}_Y^G, \mathcal{Q}_Y^G, \mathcal{P}_\Lambda^G$ and $\mathcal{Q}_\Lambda^G$ are coarse approximation of $\mathcal{P}_Y, \mathcal{Q}_Y, \mathcal{P}_\Lambda$ and $\mathcal{Q}_\Lambda$.
- The computation of $\mathcal{P}_Y^G, \mathcal{Q}_Y^G, \mathcal{P}_\Lambda^G$ and $\mathcal{Q}_\Lambda^G$ imply linear problems.

M. Gander and E. Hairer "Analysis for parareal algorithms applied to Hamiltonian differential equations", JCAM 2014.

Paraopt Algorithm

- Inexact Newton method

$$(\Phi')^G \begin{pmatrix} Y^k \\ \Lambda^k \end{pmatrix} \begin{pmatrix} Y^{k+1} - Y^k \\ \Lambda^{k+1} - \Lambda^k \end{pmatrix} = -\Phi \begin{pmatrix} Y^k \\ \Lambda^k \end{pmatrix}.$$

- $\mathcal{P}_Y^G, \mathcal{Q}_Y^G, \mathcal{P}_\Lambda^G$ and $\mathcal{Q}_\Lambda^G$ are performed in parallel over subintervals using the coarse time discretization.
- $\mathcal{P}$ and $\mathcal{Q}$ are performed in parallel over subintervals using the fine time discretization.

J. Salomon et al "Paraopt: A Parareal algorithm for optimality systems", SIAM 2020.

Paraopt Algorithm

Dahlquist test equation

- Let us consider the following Dahlquist test problem

$$\begin{aligned} \dot{y}(t) =& \sigma y(t) + c(t), \\ y(0) =& y^0, \end{aligned}$$

where $\sigma > 0$.

- Setting $\delta t = \Delta T / M$, Euler explicit yields to

$$\begin{aligned} \mathcal{P}_{\delta t}(Y_{\ell-1}, \Lambda_\ell) =& -\beta_{\delta t} Y_{\ell-1} + \frac{1}{\alpha}\gamma_{\delta t}\Lambda_\ell \\ \mathcal{Q}_{\delta t}(Y_{\ell-1}, \Lambda_\ell) =& -\beta_{\delta t}\Lambda_\ell \end{aligned}$$

where $\beta_{\delta t} := (1 + \sigma\delta t)^{\frac{\Delta T}{\delta t}}$ and $\gamma_{\delta t} := \frac{\beta_{\delta t}^2 - 1}{\sigma(2+\sigma\delta t)}$.

Paraopt Algorithm

Dahlquist test equation

- Paraopt algorithm becomes

$$\mathcal{M}_{\Delta t}\left(X^{k+1}-X^{k}\right)=-\left(\mathcal{M}_{\delta t}X^{k}-b\right),$$

$$X=(Y,\Lambda)^{T},\quad b=\left(y_{init},0,\dots,0,-y_{target}\right)^{T},$$

$$\mathcal{M}_{\delta t} := \left(\begin{array}{cccc|cccc} 1 & & & & 0 & & & \\ -\beta_{\delta t} & 1 & & & \frac{\gamma_{\delta t}}{\alpha} & \ddots & & \\ & \ddots & \ddots & & & \ddots & & 0 \\ & & -\beta_{\delta t} & 1 & & & & \frac{\gamma_{\delta t}}{\alpha} \\ \hline & & & & 1 & -\beta_{\delta t} & & \\ & \ddots & & & & \ddots & \ddots & \\ & & & & & & 1 & -\beta_{\delta t} \\ & & & -1 & & & & 1 \end{array}\right).$$

- Δt and δt are coarse and fine time steps respectively over each subinterval $[T_\ell, T_{\ell+1}]$.

Paraopt Algorithm

Dahlquist test equation

- Setting $\tau = \beta_{\Delta t} - \gamma_{\Delta t}\frac{\beta_{\Delta t} - \beta_{\delta t}}{\gamma_{\Delta t} - \gamma_{\delta t}}$.
- $\tau > 1$.
- We assume that τ satisfies

$$(\tau - 1) \geq \frac{\beta_{\Delta t} - 1}{2(L-1)\beta_{\Delta t}}. \tag{4}$$

- Let $1 < \tau_0 < \tau$.
- Let $L_0 = \frac{(\beta_{\Delta t} - \tau)}{\gamma_{\Delta t}(\tau - \tau_0)}$.

Paraopt Algorithm

Dahlquist test equation

Theorem (N.)

Let $\sigma > 0, \alpha, T, L, \Delta t, \delta t$ and be given such that (4) is satisfied. If $L > \alpha L_0$ then the spectral radius of the iterate matrix satisfies

$$\rho < \sigma\left(\Delta t - \delta t\right)\left[\frac{1}{2} + \left(\frac{1}{2}\sigma(\Delta t - \delta t) + 1\right) e^{2\sigma\Delta T}\right].$$

- The spectral radius depends strongly on σ and $(\Delta t - \delta t)$.

References

M. Gander, F. Kwok and J. Salomon
Paraopt: A parareal algorithm for optimal control systems.
SIAM Journal on Scientific Computing, 42(5): A2773-A2802, 2020.

M. Gander, S. Vandewalle.
Analysis of the Parareal time-parallel time-integration method.
SIAM, 29(2) :556–578, 2007.

M. Gander, E. Hairer.
Analysis for parareal algorithms applied to Hamiltonian differential equations.
JCAM 259 :2-13 2014.

End

Thank You