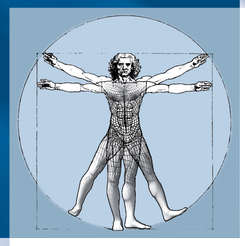
SimTech Cluster of Excellence

University of Stuttgart
Germany



Data-driven time parallelism with application to reduced order models

Lukas Brencher[1], Kevin Carlberg[2], Bernard Haasdonk[1], Andrea Barth[1]

[1] IANS, University of Stuttgart, [2] Sandia National Laboratories, Livermore

Motivation

Model order reduction lowers the computational costs of dynamical systems in *Cores* $\times$ *hours*, but does not generate speedup regarding the wall-time as spatial parallelism is quickly saturated. Therefore, one can apply time-parallelism to reduced order models (ROMs). A new method is presented [1, 2], which employs the time-evolution data collected during the offline (training) stage of the ROM to compute an accurate *coarse propagator* of the Parareal framework.

Time parallelism

In parallel-in-time methods, the time interval of the problem is divided into time-sub-intervals. To construct sub-problems, which can be computed in parallel, initial conditions are introduced to each sub-interval. This parallel computation yields to *jumps* in the final solution, which are corrected iteratively.

In the Parareal framework [4], two propagation operators are used for the computation, which differ in the used time discretization:

- $\mathcal{G}(T_{\bar{m}+1}, T_{\bar{m}}, \mathbf{X}_{\bar{m}})$ provides a *coarse* approximation (serial)
- $\mathcal{F}(T_{\bar{m}+1}, T_{\bar{m}}, \mathbf{X}_{\bar{m}})$ provides a *fine* approximation (parallel)

For Parareal iteration k do the correction iteration for the final solution $\mathbf{X}_{\bar{m}+1}^{k+1}$ until convergence:

$$\mathbf{X}_{\bar{m}+1}^{k+1} = \mathcal{F}(T_{\bar{m}+1}, T_{\bar{m}}, \mathbf{X}_{\bar{m}}^{k}) + \underbrace{\mathcal{G}(T_{\bar{m}+1}, T_{\bar{m}}, \mathbf{X}_{\bar{m}}^{k+1}) - \mathcal{G}(T_{\bar{m}+1}, T_{\bar{m}}, \mathbf{X}_{\bar{m}}^{k})}_{\text{jump}}$$

Forecasting

The forecasting method introduced in [3] aims to predict the unknown state at future time steps. Therefore, it employs the data of the ROM offline stage and the previously computed time steps to forecast the unknown state. After a time-evolution basis is computed via (thin) SVD of the ROM training snapshots, the forecast coefficients are obtained by solving the least-squares problem

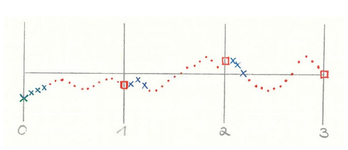
$$\mathbf{z}_j = \underset{\mathbf{z}\in\mathbb{R}^a}{\operatorname{argmin}} \|\mathbf{Z}(m,\alpha)\Xi_j\mathbf{z} - \mathbf{Z}(m,\alpha)\underline{h}(\mathbf{x}_j)\|$$

Here, α is the number of previous time steps used for the computation (*Memory*), $\mathbf{Z}$ is the sampling matrix, which extracts entries of a given vector and $\underline{h}$ 'unrolls' the time according to its discretization. This procedure is applied locally on each sub-interval of the time parallelism.

Data-driven coarse propagator

In the following illustration of the data-driven coarse propagator, the forecast memory is set to $\alpha = 3$.

1. Serial computation of the first initial guess $\mathbf{X}_{\bar{m}+1}^{0} = \mathcal{G}(T_{\bar{m}+1}, T_{\bar{m}}, \mathbf{X}_{\bar{m}}^{0})$

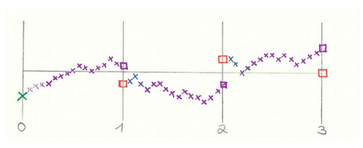


$\times$ - value computed on fine grid via time-integration used for $\mathcal{F}$

$\cdots$ - computed forecast

$\square$ - value of forecast at end of sub-interval

2. Parallel computation of the fine approximations $\mathcal{F}(T_{\bar{m}+1}, T_{\bar{m}}, \mathbf{X}_{\bar{m}}^{k})$



$\square$ - initial value of subinterval computed by forecast

$\times$ - value computed on fine grid via $\mathcal{F}$

$\square$ - value of fine approximation $\mathcal{F}$ at end of sub-interval

3. Serial computation of the correction step

$$\mathbf{X}_{\bar{m}+1}^{k+1} = \mathcal{F}(T_{\bar{m}+1}, T_{\bar{m}}, \mathbf{X}_{\bar{m}}^{k}) + \mathcal{G}(T_{\bar{m}+1}, T_{\bar{m}}, \mathbf{X}_{\bar{m}}^{k+1}) - \mathcal{G}(T_{\bar{m}+1}, T_{\bar{m}}, \mathbf{X}_{\bar{m}}^{k})$$

$\square$ - value computed by fine propagator $\mathcal{F}$

$\square$ - value computed by coarse propagator $\mathcal{G}$ during last forecast

$\square$ - value of the new forecast computed by $\mathcal{G}$

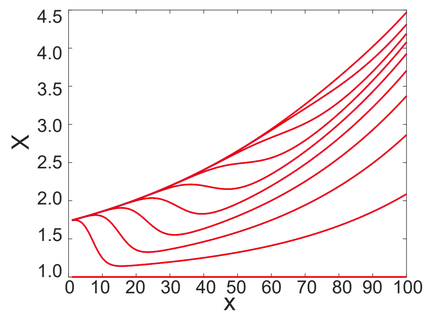
$\times$ - corrected initial value of next sub-interval

Numerical experiments

- Burgers equation

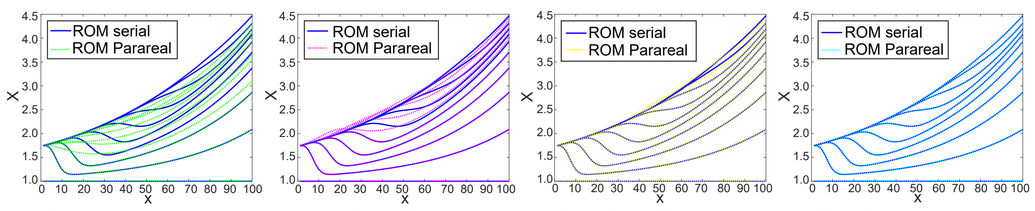
$$\frac{\partial X(x;t)}{\partial t} + \frac{1}{2}\frac{\partial (X^2(x;t))}{\partial x} = 0.02e^{bx}$$
$$X(0;t) = a \quad \forall t > 0$$
$$X(x;0) = 1 \quad \forall x \in [0,100]$$

- $(a,b) \in [2.5,3.5]\times[0.02,0.03]$
- time $t \in [0,25]$



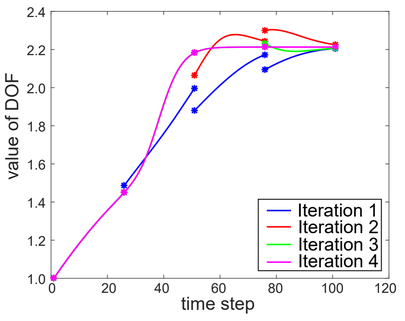
Solution of the Burgers equation. Each curve is the solution at a specific time.

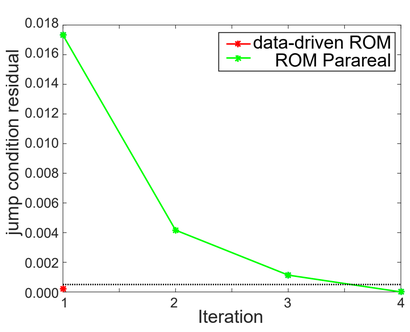
The subsequent illustration demonstrates the behavior of the solution during the iterations of the time-parallelism.



Iteration 1 of 4. Iteration 2 of 4. Iteration 3 of 4. Iteration 4 of 4.

To exemplify the convergence of the time parallelism, the figure shown alongside shows a specific degree of freedom during the parallel iterations. Furthermore, the jumps introduced by the time parallelism are visualized.





This illustration shows the jump condition residual used for the method's convergence over the time parallel iterations. Note that the proposed method converged after the first iteration, while the Parareal method requires 4 of 4 iterations to satisfy the convergence criterion.

The following table illustrates that the proposed method generates speedup compared to the Parareal framework in [4].

used CPUs	Dirichlet	Source	FOM Parareal	ROM Parareal	data-driven ROM $\alpha = 4$	data-driven ROM $\alpha = 8$
4	3.2004	0.0272	**0.1439**	**0.1979**	**0.1102**	**0.0668**
4	2.9325	0.0236	0.1329	0.2084	0.1104	0.0670
4	3.4385	0.0290	0.1314	0.2102	0.1279	0.0768
8	3.2004	0.0272	**0.0744**	**0.1261**	**0.0572**	**0.0843**
8	2.9325	0.0236	0.0924	0.1143	0.0583	0.0850
8	3.4385	0.0290	0.0755	0.1160	0.0579	0.0840

Simulation time (sec) for different online points and forecast memory α.

Conclusions

With an accurate forecast of the unknown, the proposed data-driven method converges in less iterations of the time parallelism, which leads to computational speedup measured in walltime.

Extensions are in progress:

- local basis update
- global initial forecast
- dominant reduced coordinate prediction for improved stability

References

[1] Lukas Brencher.
Leveraging spatial and temporal data for time-parallel model reduction.
Bachelor's thesis, University of Stuttgart, 2015.

[2] Kevin Carlberg, Lukas Brencher, Bernard Haasdonk, and Andrea Barth.
Data-driven time parallelism with application to reduced-order models.
in progress, 2016.

[3] Kevin Carlberg, Jaideep Ray, and Bart van Bloemen Waanders.
Decreasing the temporal complexity for nonlinear, implicit reduced-order models by forecasting.
Computer Methods in Applied Mechanics and Engineering, 289:79–103, 2015.

[4] J Lions, Yvon Maday, and Gabriel Turinici.
A "'parareal"' in time discretization of PDEs.
Comptes Rendus de l'Academie des Sciences Series I Mathematics, 332(7):661–668, 2001.