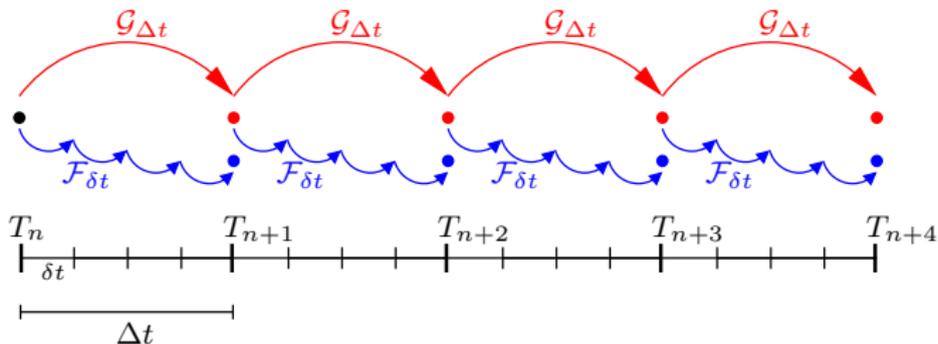


# Parallel-in-time integration - a possible way to increase parallelism?

Daniel Ruprecht

Institute of Computational Science  
University of Lugano (USI)

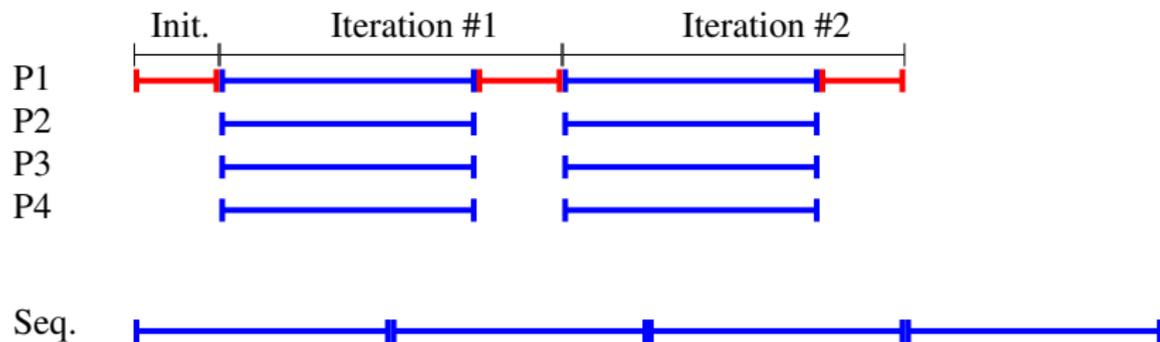
# Parallel-in-Time Integration



## Two Propagators

- *Coarse propagator*  $\mathcal{G}_{\Delta t}$  : larger time step, lower order, simplified equations
- *Fine propagator*  $\mathcal{F}_{\delta t}$  : determines accuracy of solution

# Parallel-in-Time Integration



## Two Propagators

- *Coarse propagator*  $\mathcal{G}_{\Delta t}$  : larger time step, lower order, simplified equations
- *Fine propagator*  $\mathcal{F}_{\delta t}$  : determines accuracy of solution

# Parareal

## Ingredients:

$$y_{n+1} = \mathcal{G}_{\Delta t}(y_n, T_{n+1}, T_n) \quad \text{coarse propagator}$$

$$y_{n+1} = \mathcal{F}_{\delta t}(y_n, T_{n+1}, T_n) \quad \text{fine propagator}$$

## Idea:

$$y_{n+1} = \underbrace{\mathcal{F}_{\delta t}(y_n)}_{\text{predictor}} - \underbrace{\mathcal{G}_{\Delta t}(y_n)}_{\text{corrector}} + \underbrace{\mathcal{G}_{\Delta t}(y_n)}_{\text{corrector}}$$

## Iteration:

$$y_{n+1}^{k+1} = \underbrace{\mathcal{F}_{\delta t}(y_n^k, T_{n+1}, T_n) - \mathcal{G}_{\Delta t}(y_n^k, T_{n+1}, T_n)}_{\text{predictor}} + \underbrace{\mathcal{G}_{\Delta t}(y_n^{k+1}, T_{n+1}, T_n)}_{\text{corrector}}$$

## Convergence:

$$y_n^k \rightarrow \mathcal{F}_{\delta t}(y_0, T_n, T_0), \quad k \rightarrow N_c$$

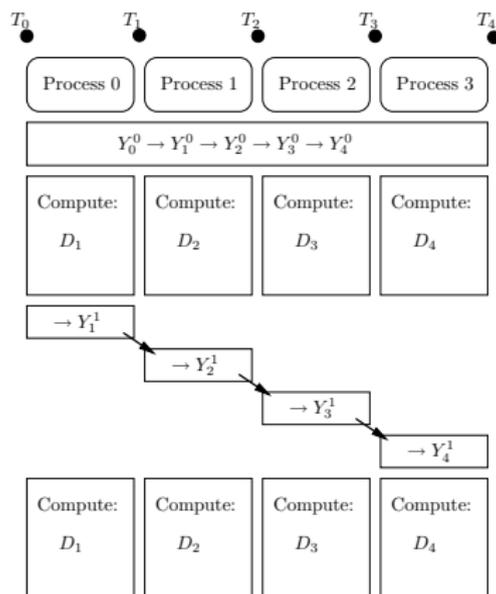
## Reference

[Lions et al. (2001)]

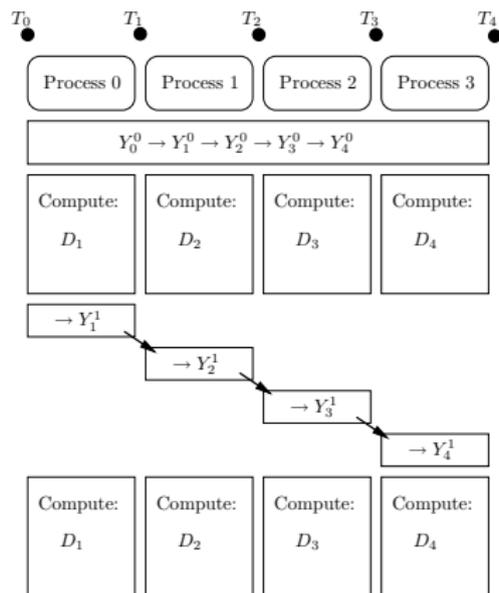
# Algorithm

```

{ ## Initialization (Sequential) ## }
 $Y_0^0 = y_0$ 
for  $i = 1$  to  $N_c$  do
   $Y_i^0 = \mathcal{G}_{\Delta t} (Y_{i-1}^0, T_i, T_{i-1})$ 
end for
{ ## Iteration ## }
 $k := 0$ 
repeat
  { # Predictor (Parallel) # }
  for  $i = 1$  to  $N_c$  do
     $D_i := \mathcal{F}_{\delta t} (Y_{i-1}^k, T_i, T_{i-1}) -$ 
       $\mathcal{G}_{\Delta t} (Y_{i-1}^k, T_i, T_{i-1})$ 
  end for
  { # Corrector (Sequential) # }
   $Y_0^{k+1} = y_0$ 
  for  $i = 1$  to  $N_c$  do
     $Y_i^{k+1} := \mathcal{G}_{\Delta t} (Y_{i-1}^{k+1}, T_i, T_{i-1}) + D_i$ 
  end for
   $k := k + 1$ 
until  $k = N_{it}$ 
  
```



# Speedup Estimate



$$s \approx \frac{N_t \tau_f}{N_c \tau_c + N_{it} \left( N_c \tau_c + \frac{N_t}{N_c} \tau_f \right)} = \frac{1}{(1 + N_{it}) \frac{N_c}{N_t} \frac{\tau_c}{\tau_f} + \frac{N_{it}}{N_c}} \leq \frac{N_c}{N_{it}}$$

# Krylov-Subspace-Enhanced Parareal

- Evolution of  $\mathcal{F}$  is known on

$$S^k := \text{span} \left\{ Y_i^{k'} : i = 1, \dots, N_c, k' \leq k \right\}$$

- Enhance  $\mathcal{G}$  to

$$\mathcal{K}_{\Delta t}(Y) := \mathcal{G}_{\Delta t} \left( (\mathbf{I} - \mathbf{P}^k) Y \right) + \mathcal{F}_{\delta t} \left( \mathbf{P}^k Y \right)$$

- Requires QR decomposition in every correction step.
- But: *works for hyperbolic systems!*

## References

- introduced by [Farhat et al. (2006), Cortial and Farhat (2008)]
- formulation taken from [M. Gander and M. Petcu (2008)]

## Explicit coarse integrator

- Efficiency requires  $\Delta t \gg \delta t$ , but  $\mathcal{G}_{\Delta t}$  must be stable.
- Original works rely on implicit scheme, but who wants to implement this in COSMO???
- But: Partially split scheme (Wicker, Skamarock) with divergence damping works, too:

$$y_t = F_{\text{fast}}(y) + F_{\text{slow}}(y)$$

- Allows for coarse CFL numbers  $C_{\text{coarse}} \gg 1$ .

### Discretization

- $\mathcal{G}$  : split forw. Euler/Forward-Backward, 1st order advective flux
- $\mathcal{F}$  : Runge-Kutta-3, 6th order advective fluxes
- Finite Volumes on rectangular cells

$$\partial_t q_i = -\frac{F_{i+1/2,j} - F_{i-1/2,j}}{\Delta x} - \frac{G_{i,j+1/2} - G_{i,j-1/2}}{\Delta y}$$

# Numerical example

- **Model:**

$$\mathbf{u}_t + \mathbf{U} \cdot \nabla \mathbf{u} + c_s \nabla \pi = 0$$

$$\pi_t + \mathbf{U} \cdot \nabla \pi + c_s \nabla \cdot \mathbf{u} = 0$$

with  $c_s = 30$ .

- **Advection:**  $\mathbf{U} = (U, V)$  with

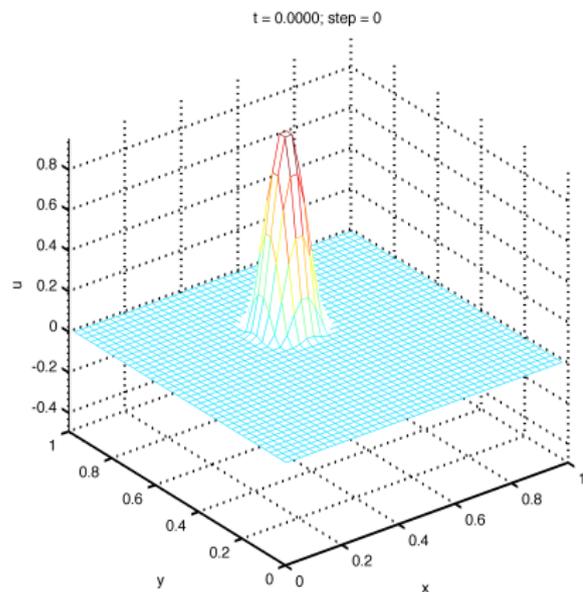
$$U = -\omega (y - 0.5)$$

$$V = \omega (x - 0.5),$$

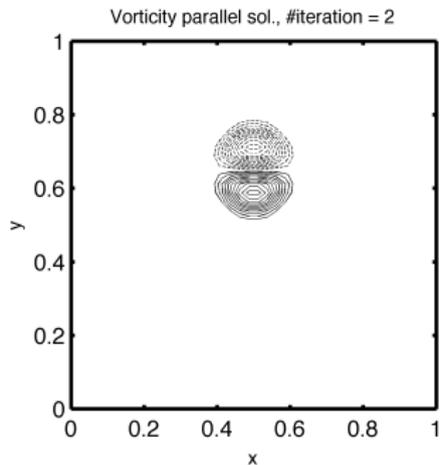
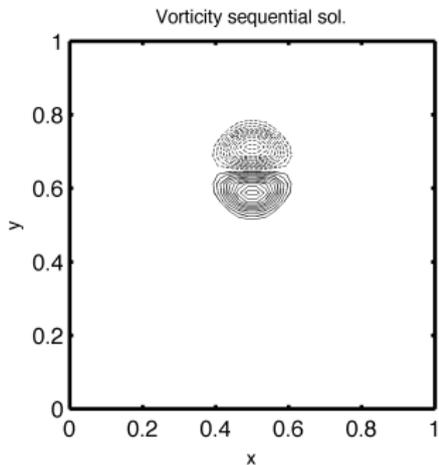
$\max(\mathbf{U}) \approx 1.5$ , and  $\omega = \pi$ .

- $C_c = 4$ ,  $N_{\text{sound}} = 5$ ,  $C_f = 0.2$ .

- $\nu_c = 0.3$ ,  $\nu_f = 0.01$



# Comparison of vorticity



## Run times and speed up

threads	#it.	diff.	par.	speed up
6	1	$5.6 \times 10^{-2}$	3.8 s	4.0
6	2	$1.5 \times 10^{-2}$	7.7 s	2.0
6	3	$6.0 \times 10^{-3}$	11.9 s	1.3

Table:  $N_x = N_y = 40$ . Sequential run time  $\tau_{\text{seq}} = 15.2$  s.

threads	#it.	diff.	par.	speed up
6	1	$5.7 \times 10^{-2}$	29.9 s	4.0
6	2	$1.3 \times 10^{-2}$	61.1 s	2.0
6	3	$4.5 \times 10^{-3}$	93.3 s	1.3

Table:  $N_x = N_y = 80$ . Sequential run time  $\tau_{\text{seq}} = 119.8$  s.

# Residuals

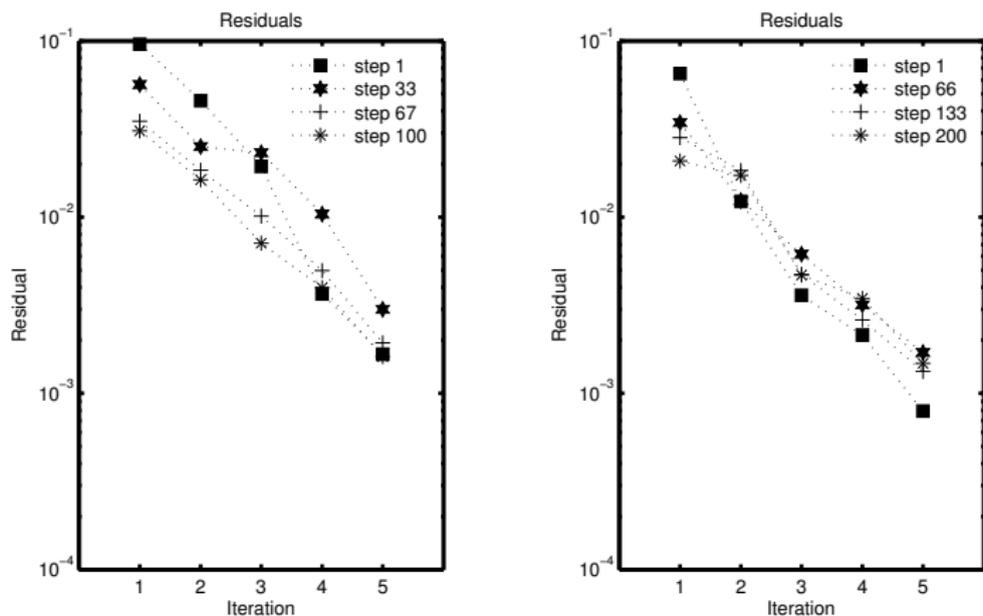
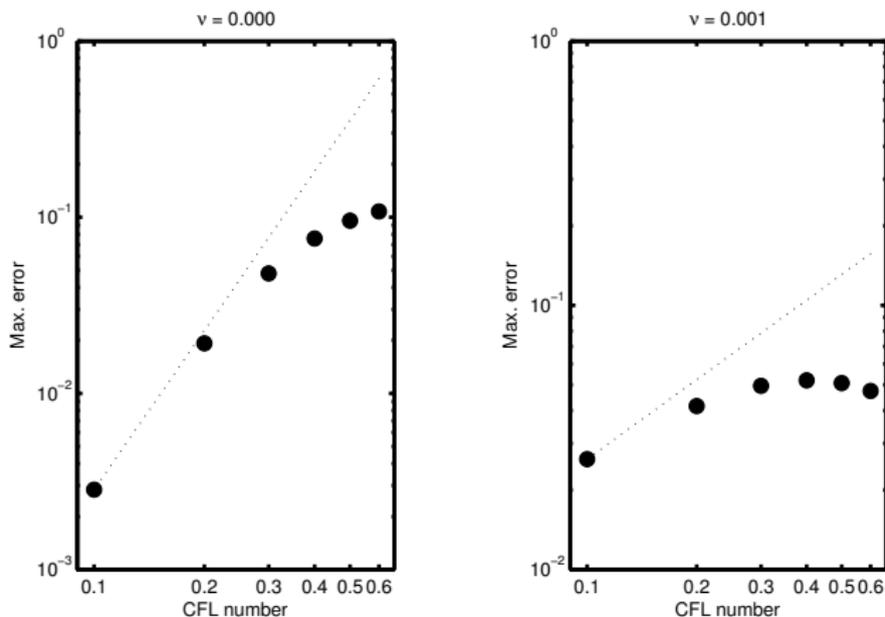


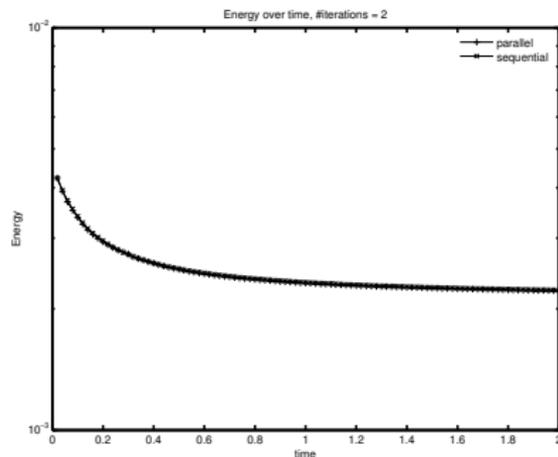
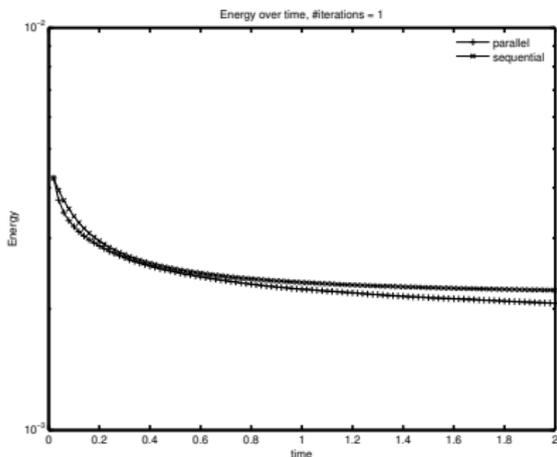
Figure: **Left:**  $N = 40$ . **Right:**  $N = 80$ .

# Error from time discretization



**Figure:** Maximum difference of sequential solution against reference solution with  $C = 0.01$ . **Left:** No divergence damping. **Right:**  $\nu = 0.001$ .

# Energy



**Figure:** Evolution of total energy over time for parallel and sequential solution.  
**Left:** After one iteration. **Right:** After two iterations.

## Energy

$$E := \frac{1}{2} (u^2 + v^2 + \pi^2) \Rightarrow \frac{DE}{Dt} + c_s \nabla \cdot (\mathbf{u}\pi) = 0$$

# Conclusions and Questions

## Conclusions

- 1 KSE - Parareal is applicable for 2D linear acoustic-advection.
- 2 Use of purely explicit propagators is possible.
- 3 Speed up is achieved but parallel efficiency is inherently moderate.
- 4 "Easy" to implement if  $\mathcal{F}$ ,  $\mathcal{G}$  are available.

## Questions

- 1 What about more complex models, e.g. fully nonlinear compressible Euler eq.?
- 2 Implementation in a hybrid MPI (spatial) / Open MP (temporal) approach ?
- 3 Reduced models (e.g. sound-proof) in  $\mathcal{G}$  for even larger coarse time steps?

# References

- 

C. Farhat and J. Cortial and C. Dastillung and H. Bavestrello,  
*Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses*,  
*Int. J. Numer. Meth. Engng.*, 67 (2006), pp. 697–724.
- 

J. Cortial and C. Farhat,  
*A time-parallel implicit method for accelerating the solution of non-linear structural dynamics*,  
*Int. J. Numer. Meth. Engng.*, 77 (2008), pp. 451–470.
- 

M. Gander and M. Petcu  
*Analysis of a Krylov subspace enhanced parareal algorithm for linear problems*  
*ESAIM Proc.*, 25 (2008), pp. 114–129.
- 

J.-L. Lions and Y. Maday and G. Turinci,  
*A "parareal" in time discretization of PDE's*  
*C. R. Acad. Sci. Sér. I Math.*, 332 (2001), pp. 661-668.

## Run times of segments

threads	<i>#it.</i>	coarse	fine	QR
6	1	1.1 s	2.6 s	0.07 s
6	2	2.2 s	5.3 s	0.25 s
6	3	3.4 s	7.9 s	0.57 s

**Table:** Run time spent in different parts of the algorithm.  $N_x = N_y = 40$ .

threads	<i>#it.</i>	coarse	fine	QR
6	1	8.5 s	20.4 s	0.6 s
6	2	17.5 s	40.8 s	2.0 s
6	3	26.6 s	61.2 s	4.6 s

**Table:** Run time spent in different parts of the algorithm.  $N_x = N_y = 80$ .

# Time Series

