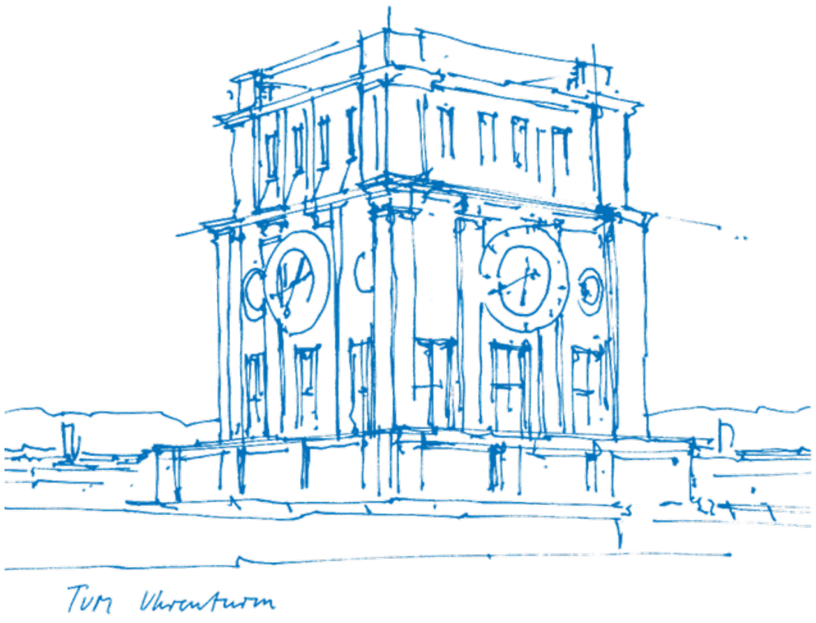
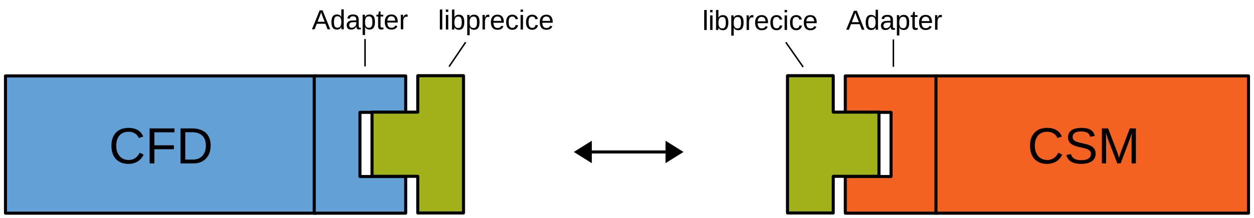
Time Stepping for Partitioned Multi-Physics

Benjamin Rüth
Technical University of Munich
Informatics
Chair of Scientific Computing in Computer Science
Jülich, 20. October 2017



Partitioned multi-physics

preCICE[1]



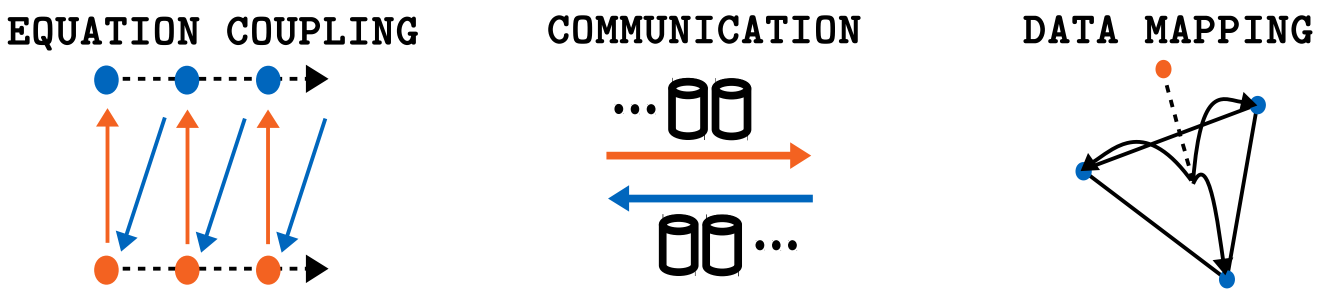
preCICE adapters for connecting solvers (e.g. OpenFOAM and CalculiX[2])

Resources

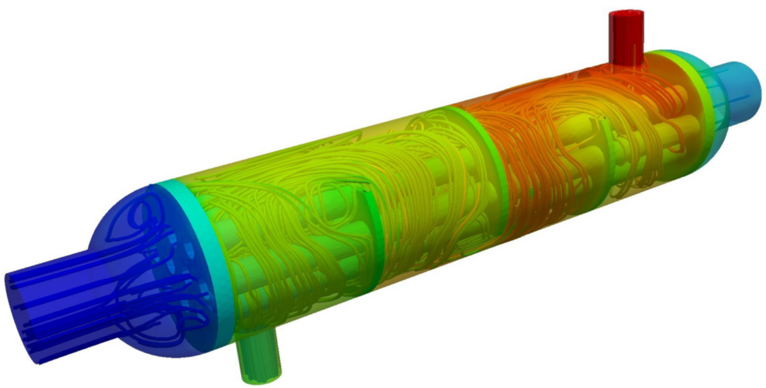
- http://www.precice.org
- written in C++
- API for other languages available (Python, Fortran)
- OpenSource, LGPL (https://github.com/precice)

[1] *Bungartz, H.-J., et al.(2016). preCICE – A Fully Parallel Library for Multi-Physics Surface Coupling.*
[2] *Uekermann, B., et al. (2017). Official preCICE Adapters for Standard Open-Source Solvers.*

Partitioned multi-physics
preCICE



- **Equation coupling:**
  quasi-Newton acceleration schemes
- **Communication:**
  fully parallel, MPI or TCP/IP
- **Data Mapping:**
  nearest neighbor/projection, radial basis function interpolation



Shell and tube heat exchanger[1]

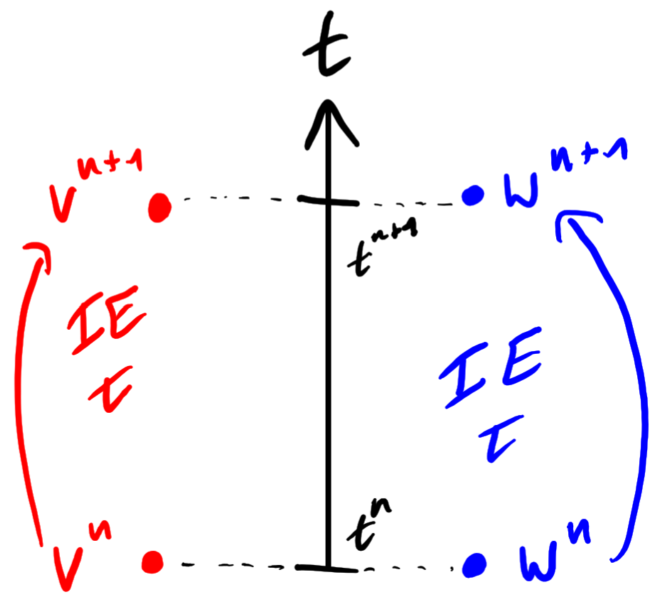
[1] *Cheung Yau, L. (2016). Conjugate Heat Transfer with the Multiphysics Coupling Library preCICE.*

Partitioned multi-physics

Time-stepping challenges

Simple

Participants **A** and **B** use identical timestep size and (high-order) solvers.



Partitioned multi-physics

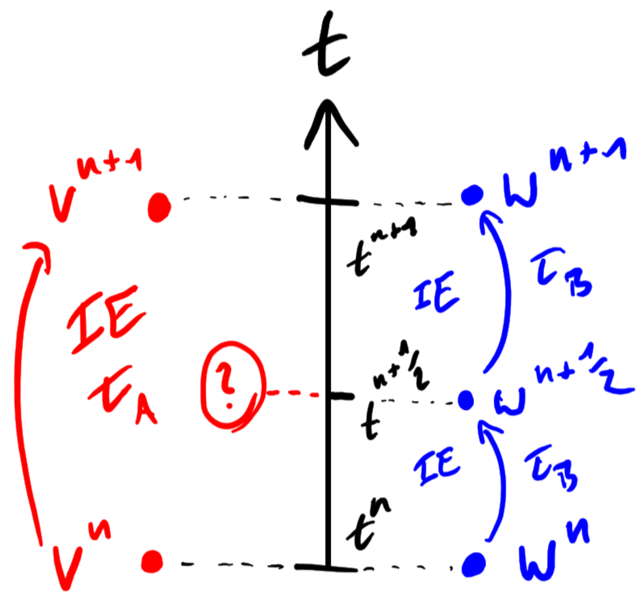
Time-stepping challenges

Simple

Participants **A** and **B** use identical timestep size and (high-order) solvers.

Subcycling

Participant **A** uses a time step size twice as big as the time step size of participant **B**
$\tau_A = 2\tau_B$.



Partitioned multi-physics

Time-stepping challenges

Simple

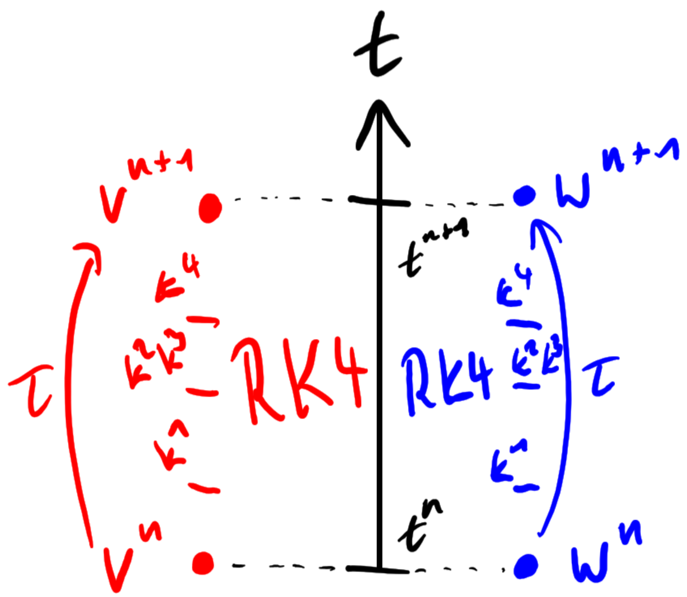
Participants **A** and **B** use identical timestep size and (high-order) solvers.

Subcycling

Participant **A** uses a time step size twice as big as the time step size of participant **B** $\tau_A = 2\tau_B$.

Substepping

Runge Kutta 4 needs function evaluations at $t^{n+\frac{1}{2}}$, which are not directly accessible.



Partitioned multi-physics

Time-stepping challenges

Simple

Participants **A** and **B** use identical timestep size and (high-order) solvers.

Subcycling

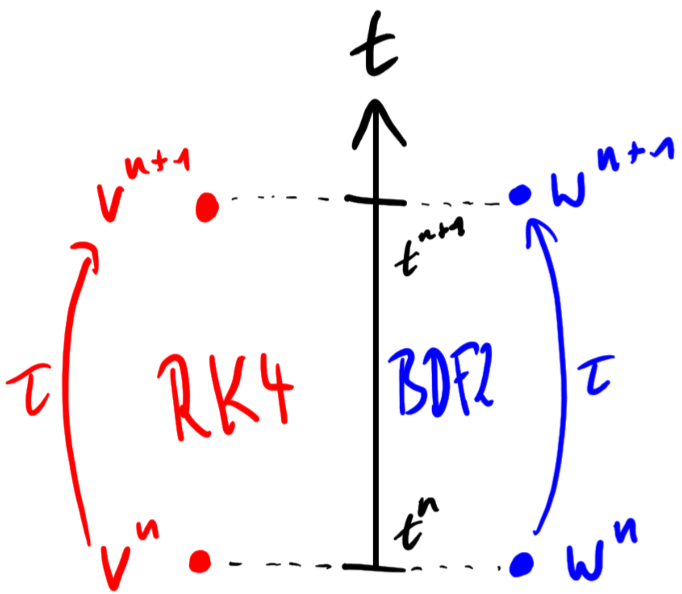
Participant **A** uses a time step size twice as big as the time step size of participant **B** $\tau_A = 2\tau_B$.

Substepping

Runge Kutta 4 needs function evaluations at $t^{n+\frac{1}{2}}$, which are not directly accessible.

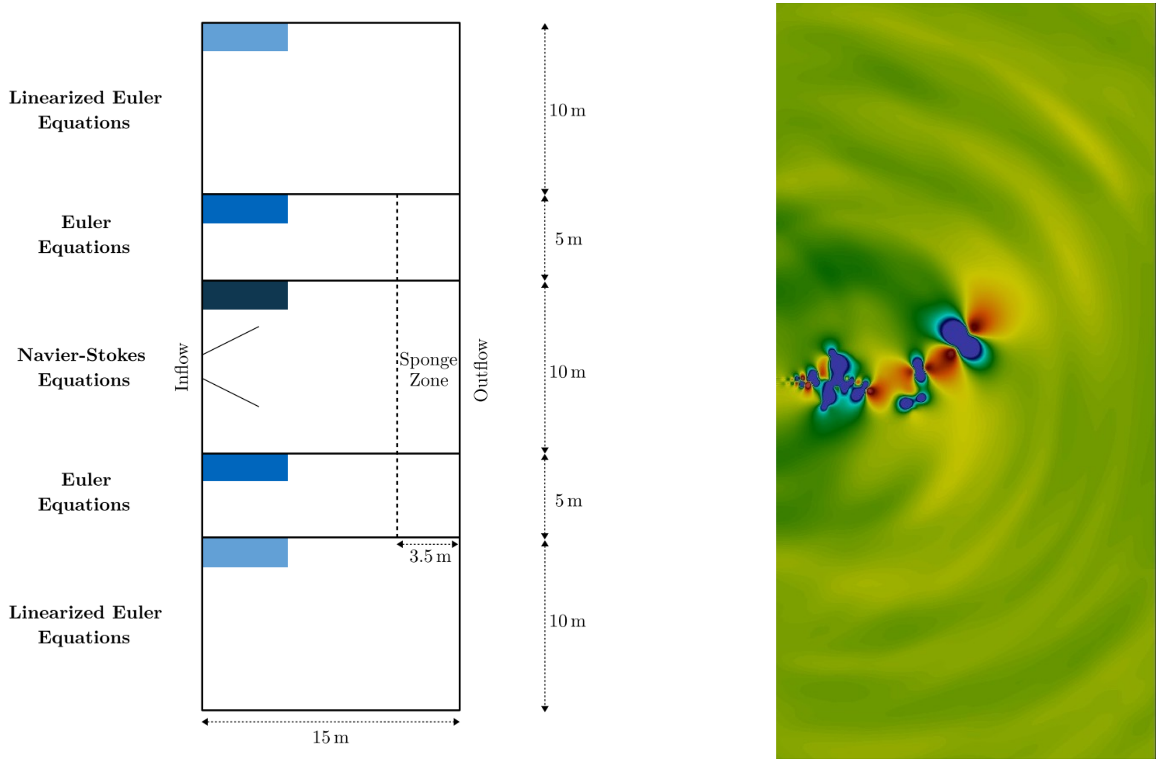
Inhomogeneous time stepping

Participants use different time stepping schemes.



Partitioned multi-physics

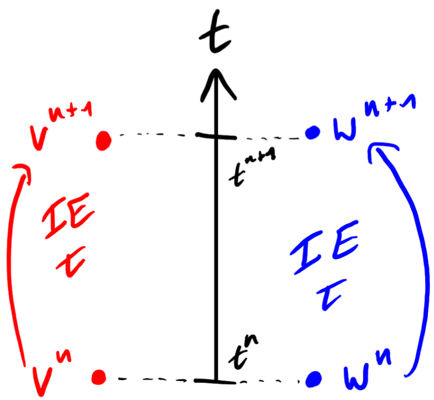
Time-stepping challenges



Three-Field Flow Coupling around a 2D Subsonic Free Jet[1]

[1] *Uekermann, B. (2016). Partitioned Fluid-Structure Interaction on Massively Parallel Systems.*

Order degradation for simple time-stepping



- Convergence order cannot be maintained[1]
- Order degradation to $\mathcal{O}(\tau)$
- Reproduce and quantify this effect
- Show up possible solutions

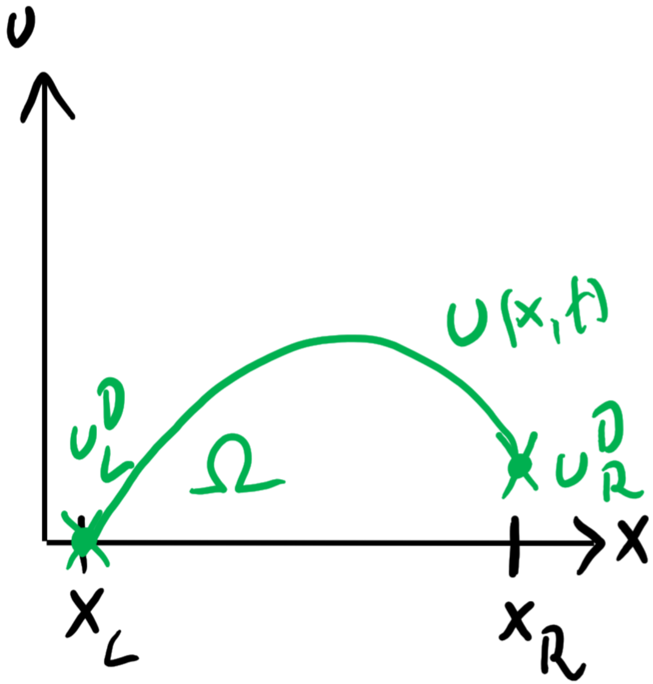
[1] *Blom, D. S., et al.(2015). On parallel scalability aspects of strongly coupled partitioned fluid-structure-acoustics interaction.*

Order degradation for simple time-stepping

1D heat transport problem

Heat Transport equation

$$\frac{\partial u(x,t)}{\partial t} = \alpha \frac{\partial^2 u(x,t)}{\partial x^2}, x \in \Omega, t \in \mathbb{R}^+$$



Dirichlet boundary conditions

$$u(x = x_L, t) = u_L^D, u(x = x_R, t) = u_R^D$$

Initial condition

$$u(x, t = 0) = u_0(x)$$

Order degradation for simple time-stepping

Partitioned heat transport equation

Left heat transport equation

$$\frac{\partial}{\partial t}v(x,t) = \alpha\frac{\partial^2}{\partial x^2}v(x,t), x \in \Omega_L, t \in \mathbb{R}^+$$

$$u_L(x_L,t) = u_L^D,\ \frac{\partial}{\partial x}v(x_C,t) = u_C^N(t)$$

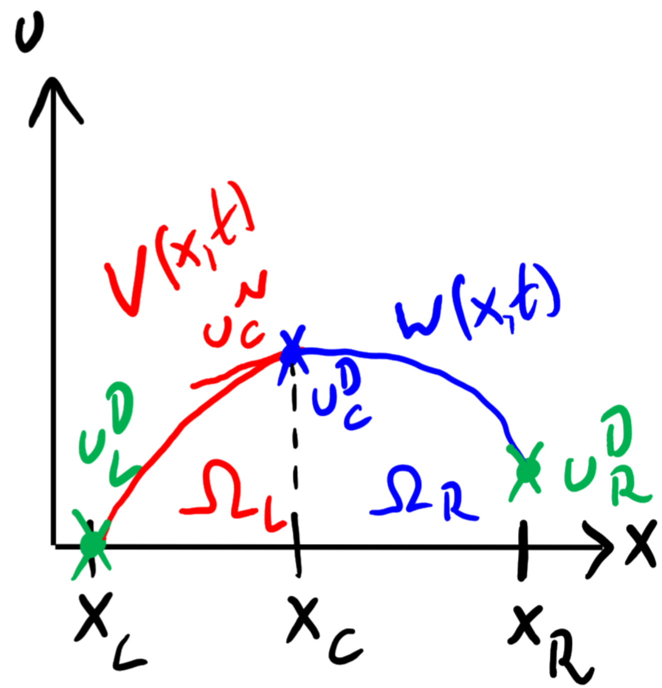
$$v(x,0) = u_0(x)$$

Right heat transport equation

$$\frac{\partial}{\partial t}w(x,t) = \alpha\frac{\partial^2}{\partial x^2}w(x,t), x \in \Omega_R, t \in \mathbb{R}^+$$

$$w(x_C,t) = u_C^D(t),\ w(x_R,t) = u_R^D$$

$$w(x,0) = u_0(x)$$



Order degradation for simple time-stepping

Time stepping

Explicit Euler

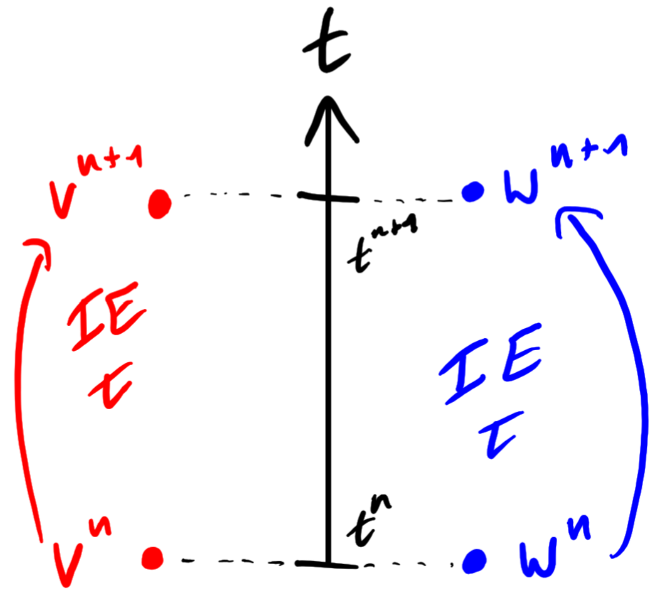
$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{n} + \tau f(\boldsymbol{u}^{n}, t_n)$$

Implicit Euler

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{n} + \tau f(\boldsymbol{u}^{n+1}, t_{n+1})$$

Trapezoidal Rule

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{n} + \frac{\tau}{2}\left[f(\boldsymbol{u}^{n}, t_n) + f(\boldsymbol{u}^{n+1}, t_{n+1})\right]$$



Order degradation for simple time-stepping

Coupling schemes

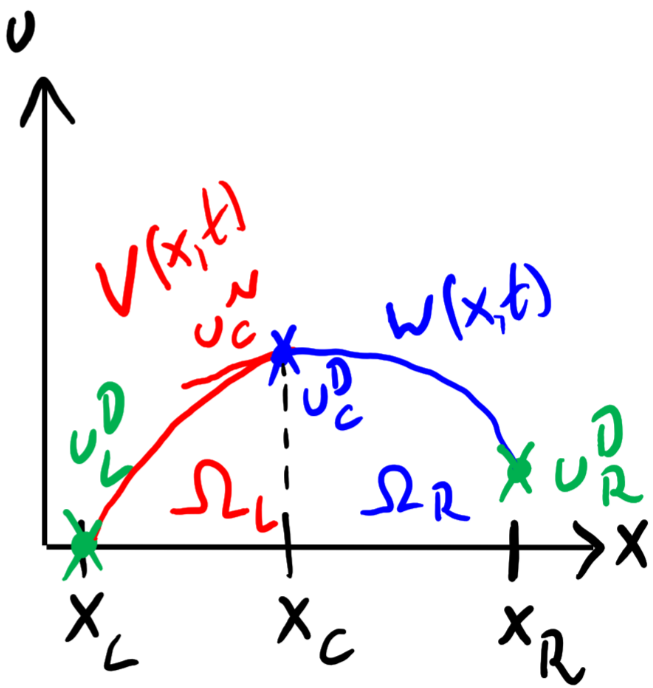
Dirichlet-Neumann coupling

Boundary condition for v

$$u_C^N = \frac{\partial}{\partial x} w(x_C)$$

Boundary condition for w

$$u_C^D = v(x_C)$$



Order degradation for simple time-stepping

Coupling schemes

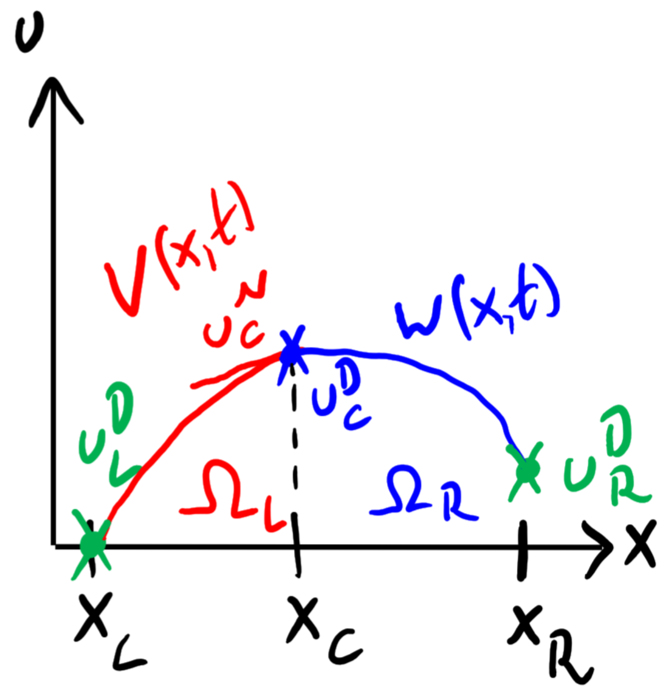
Dirichlet-Neumann coupling

Boundary condition for v

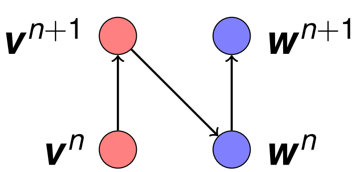
$$u_C^N = \frac{\partial}{\partial x} w(x_C)$$

Boundary condition for w

$$u_C^D = v(x_C)$$



Explicit coupling



Order degradation for simple time-stepping

Coupling schemes

Dirichlet-Neumann coupling

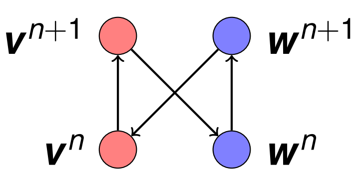
Boundary condition for v

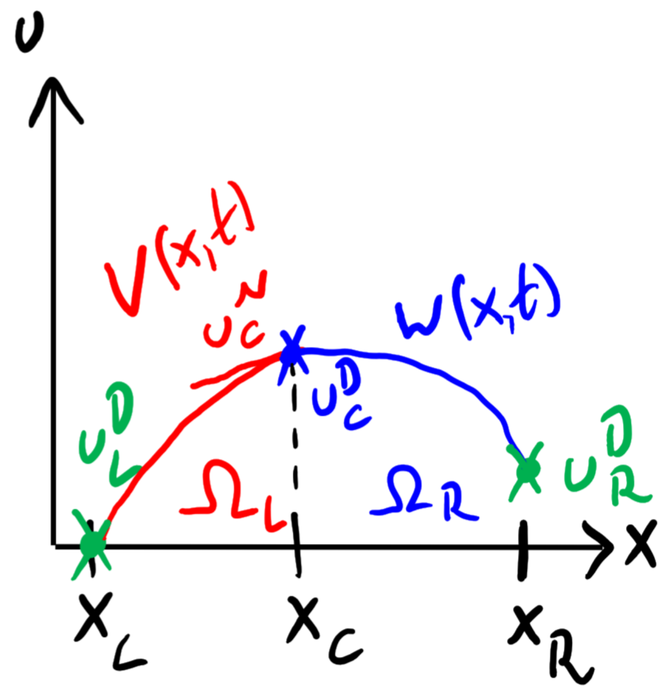
$$u_C^N = \frac{\partial}{\partial x} w(x_C)$$

Boundary condition for w

$$u_C^D = v(x_C)$$

Implicit coupling

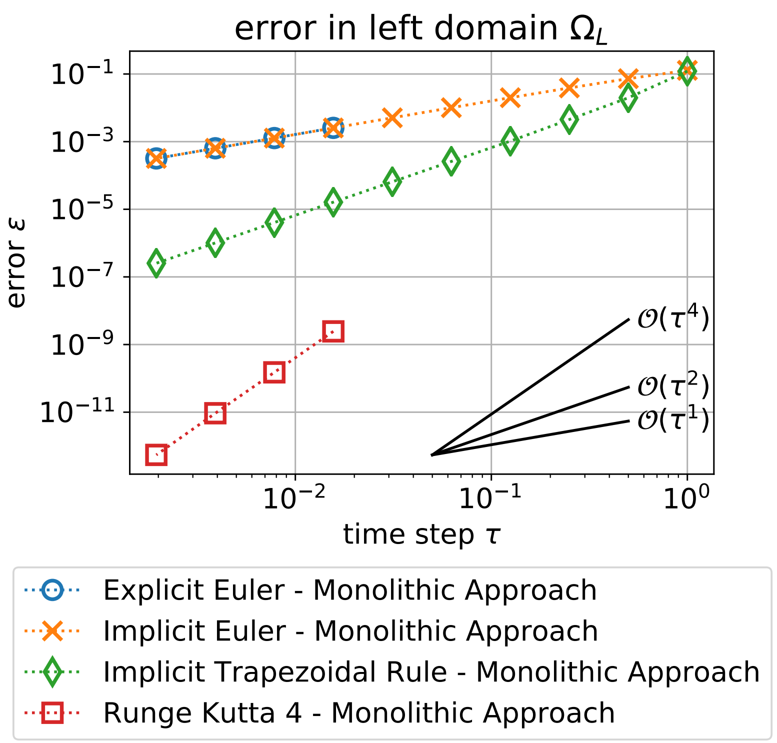




Order degradation for simple time-stepping

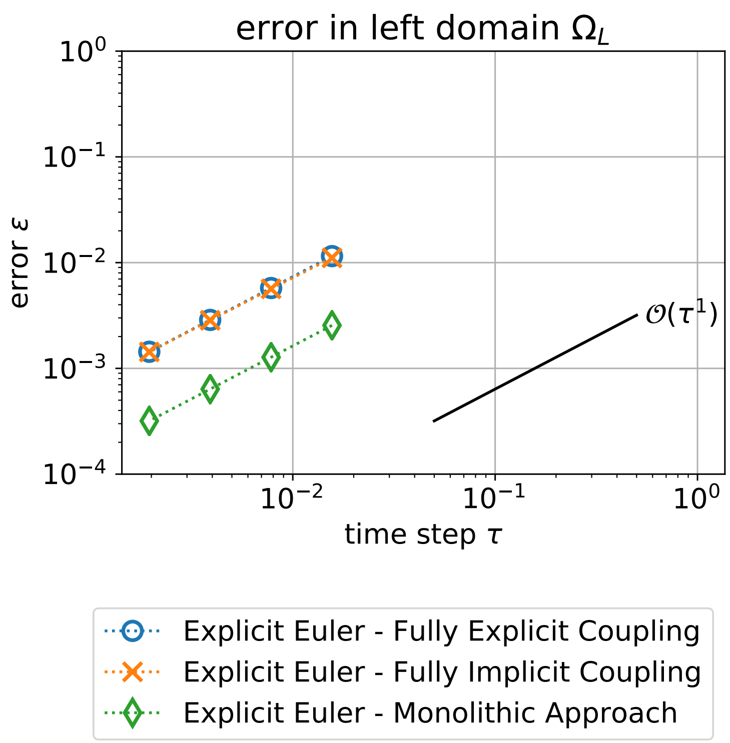
Convergence order in time

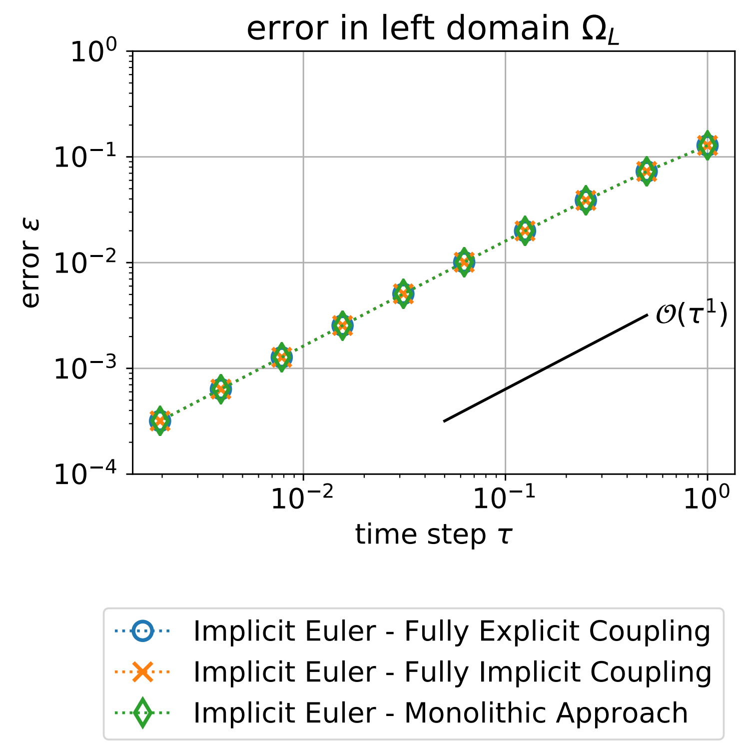
- use constant spatial meshwidth h
- refine temporal meshwidth τ
- compare to monolithic reference solution $\boldsymbol{u}^n$ with fine τ



Order degradation for simple time-stepping

Explicit and implicit Euler

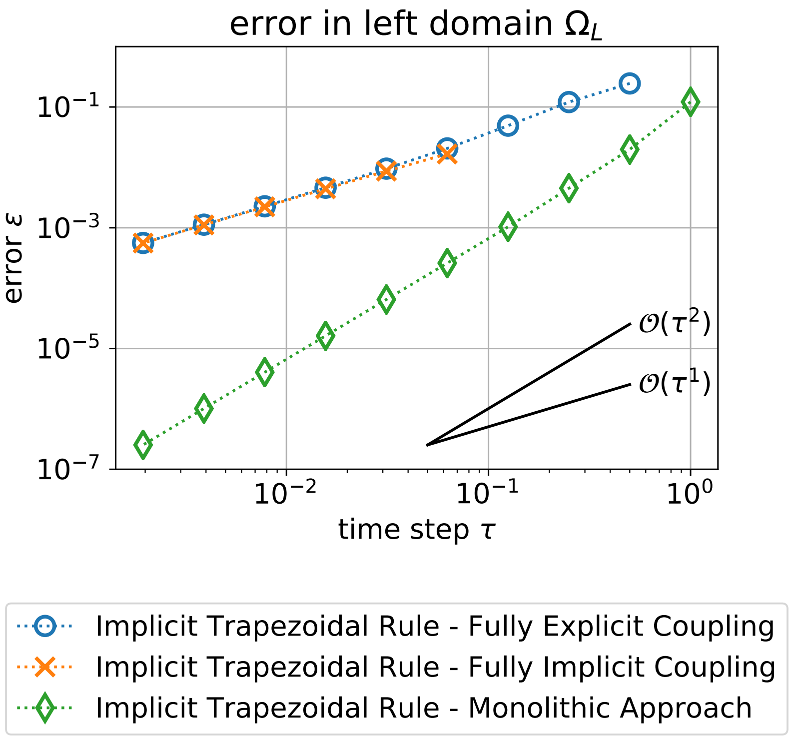




Order degradation for simple time-stepping

Trapezoidal rule

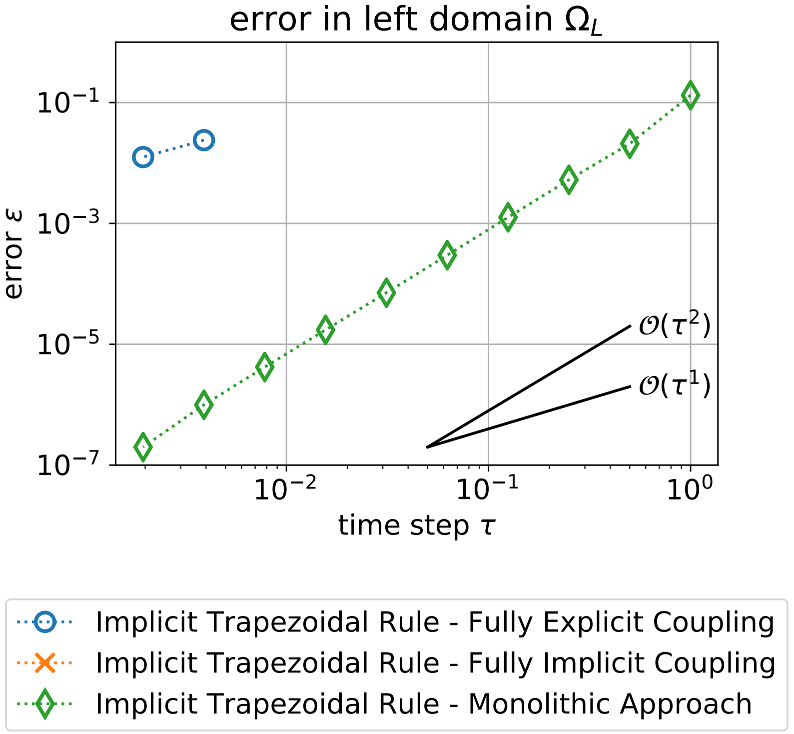
- order reduced to $\mathcal{O}(\tau)$
- $h = 0.2$
- stability problems for Fully implicit coupling



Order degradation for simple time-stepping

Trapezoidal rule

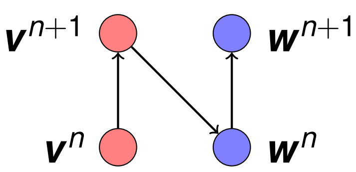
- order reduced to $\mathcal{O}(\tau)$
- $h = 0.01$
- stability problems for Fully implicit coupling
- stability problems for Fully explicit coupling

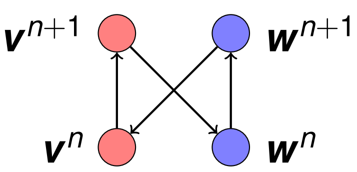


Order degradation for simple time-stepping

Trapezoidal rule

	update scheme	stability	order
fully explicit	$\boldsymbol{v}^{n+1} = \boldsymbol{v}^{n} + \frac{\tau}{2}\left[f_v(\boldsymbol{v}^{n}, t_n) + f_v(\boldsymbol{v}^{n+1}, t_{n+1})\right]$ $\boldsymbol{w}^{n+1} = \boldsymbol{w}^{n} + \frac{\tau}{2}\left[f_w(\boldsymbol{w}^{n}, t_n) + f_w(\boldsymbol{w}^{n+1}, t_{n+1})\right]$	depends on τ	$\mathcal{O}(\tau)$
fully implicit	$\boldsymbol{v}^{n+1} = \boldsymbol{v}^{n} + \frac{\tau}{2}\left[f_v(\boldsymbol{v}^{n}, t_n) + f_v(\boldsymbol{v}^{n+1}, t_{n+1})\right]$ $\boldsymbol{w}^{n+1} = \boldsymbol{w}^{n} + \frac{\tau}{2}\left[f_w(\boldsymbol{w}^{n}, t_n) + f_w(\boldsymbol{w}^{n+1}, t_{n+1})\right]$	depends on τ	$\mathcal{O}(\tau)$

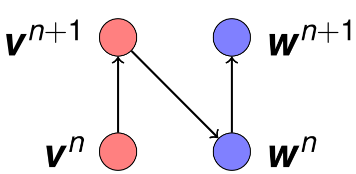


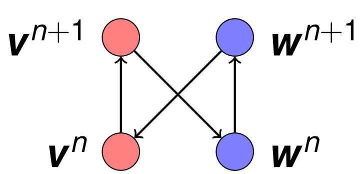


Order degradation for simple time-stepping

Trapezoidal rule

	update scheme	stability	order
fully explicit	$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \frac{\tau}{2}\left[f_v(\boldsymbol{v}^n, t_n, c_n) + f_v(\boldsymbol{v}^{n+1}, t_{n+1}, c_n)\right]$ $\boldsymbol{w}^{n+1} = \boldsymbol{w}^n + \frac{\tau}{2}\left[f_w(\boldsymbol{w}^n, t_n, c_{n+1}) + f_w(\boldsymbol{w}^{n+1}, t_{n+1}, c_{n+1})\right]$	depends on τ	$\mathcal{O}(\tau)$
fully implicit	$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \frac{\tau}{2}\left[f_v(\boldsymbol{v}^n, t_n, c_{n+1}) + f_v(\boldsymbol{v}^{n+1}, t_{n+1}, c_{n+1})\right]$ $\boldsymbol{w}^{n+1} = \boldsymbol{w}^n + \frac{\tau}{2}\left[f_w(\boldsymbol{w}^n, t_n, c_{n+1}) + f_w(\boldsymbol{w}^{n+1}, t_{n+1}, c_{n+1})\right]$	depends on τ	$\mathcal{O}(\tau)$





Order degradation for simple time-stepping

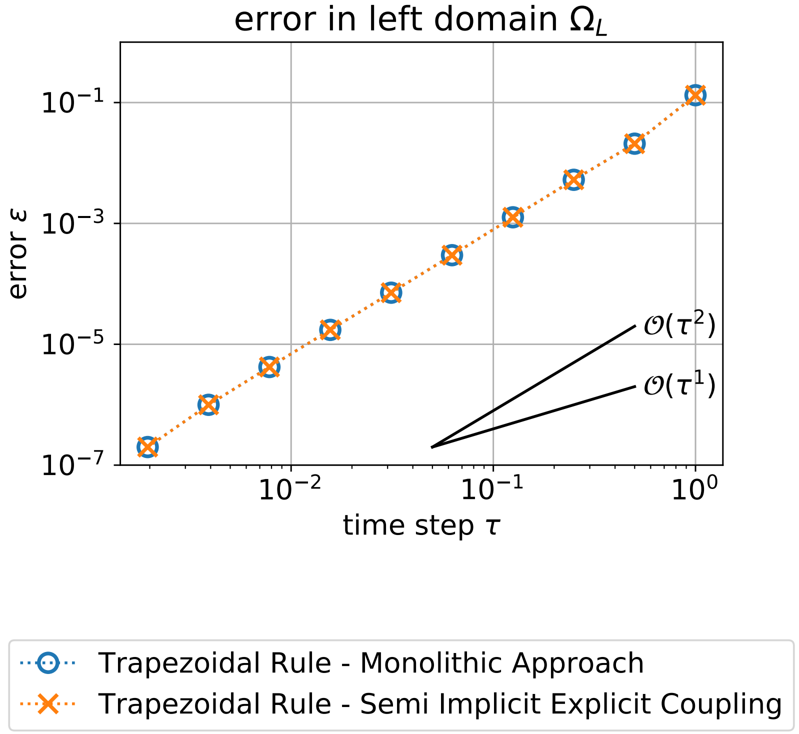
Trapezoidal rule

	update scheme	stability	order
fully explicit	$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \frac{\tau}{2}\left[f_v(\boldsymbol{v}^n, t_n, c_n) + f_v(\boldsymbol{v}^{n+1}, t_{n+1}, c_n)\right]$ $\boldsymbol{w}^{n+1} = \boldsymbol{w}^n + \frac{\tau}{2}\left[f_w(\boldsymbol{w}^n, t_n, c_{n+1}) + f_w(\boldsymbol{w}^{n+1}, t_{n+1}, c_{n+1})\right]$	depends on τ	$\mathcal{O}(\tau)$
fully implicit	$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \frac{\tau}{2}\left[f_v(\boldsymbol{v}^n, t_n, c_{n+1}) + f_v(\boldsymbol{v}^{n+1}, t_{n+1}, c_{n+1})\right]$ $\boldsymbol{w}^{n+1} = \boldsymbol{w}^n + \frac{\tau}{2}\left[f_w(\boldsymbol{w}^n, t_n, c_{n+1}) + f_w(\boldsymbol{w}^{n+1}, t_{n+1}, c_{n+1})\right]$	depends on τ	$\mathcal{O}(\tau)$
semi explicit-implicit	$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \frac{\tau}{2}\left[f_v(\boldsymbol{v}^n, t_n, c_n) + f_v(\boldsymbol{v}^{n+1}, t_{n+1}, c_{n+1})\right]$ $\boldsymbol{w}^{n+1} = \boldsymbol{w}^n + \frac{\tau}{2}\left[f_w(\boldsymbol{w}^n, t_n, c_n) + f_w(\boldsymbol{w}^{n+1}, t_{n+1}, c_{n+1})\right]$	???	???

Order degradation for simple time-stepping

Semi explicit-implicit coupling

- order $\mathcal{O}\left(\tau^2\right)$ maintained for semi explicit-implicit coupling
- no stability problems for semi explicit-implicit coupling
- $h = 0.01$

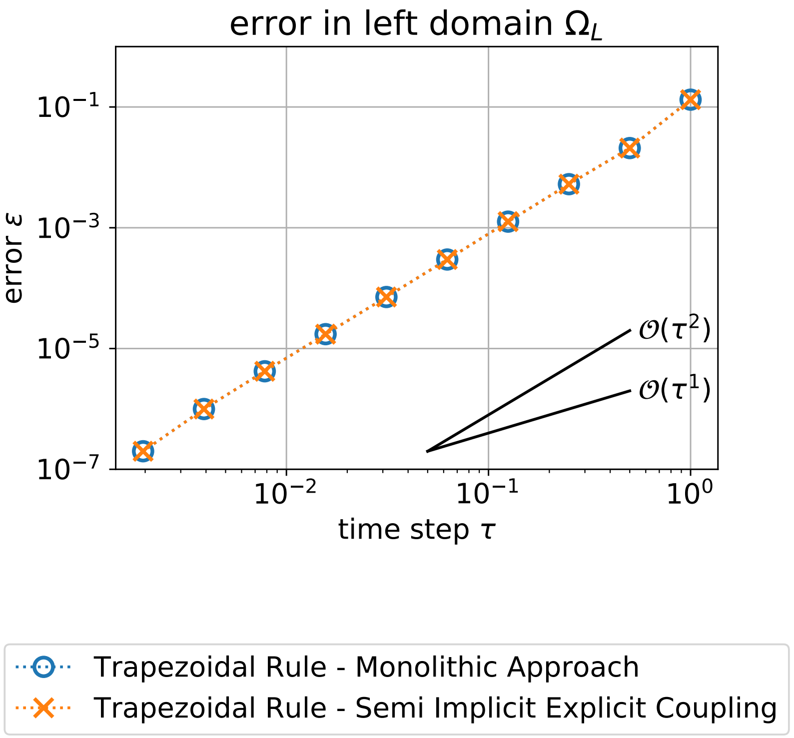


Order degradation for simple time-stepping

Semi explicit-implicit coupling

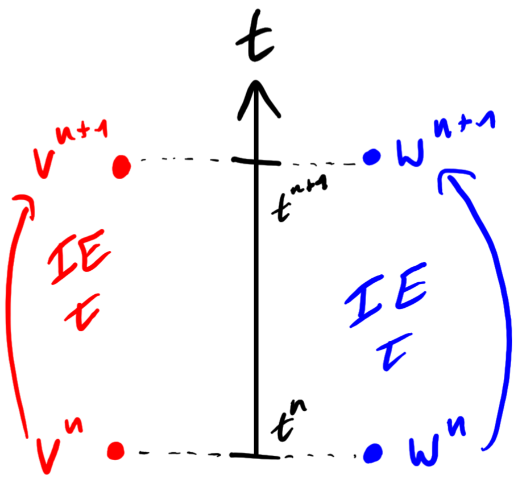
- order $\mathcal{O}\left(\tau^2\right)$ maintained for semi explicit-implicit coupling
- no stability problems for semi explicit-implicit coupling
- $h = 0.01$

	stability	order
fully explicit	depends on τ	$\mathcal{O}(\tau)$
fully implicit	depends on τ	$\mathcal{O}(\tau)$
semi explicit-implicit	unconditionally	$\mathcal{O}\left(\tau^2\right)$



Conclusions and Outlook

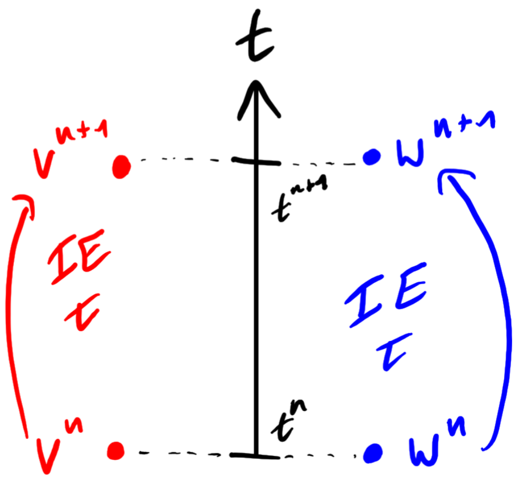
Order degradation of trapezoidal rule



- order degradation to $\mathcal{O}(\tau)$ for the trapezoidal rule with standard coupling schemes
- order $\mathcal{O}(\tau^2)$ using a specialized coupling scheme

Conclusions and Outlook

Order degradation of trapezoidal rule



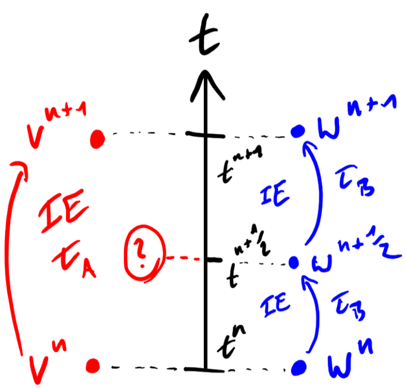
coupling	time-stepping	order
semi	Trapezoidal rule	$\mathcal{O}(\tau^2)$
predictor	Heun	$\mathcal{O}(\tau^2)$
predictor	Runge Kutta 2	$\mathcal{O}(\tau^2)$
interpolated	Midpoint rule	$\mathcal{O}(\tau^2)$
???	Runge Kutta 4	$\mathcal{O}(\tau)$

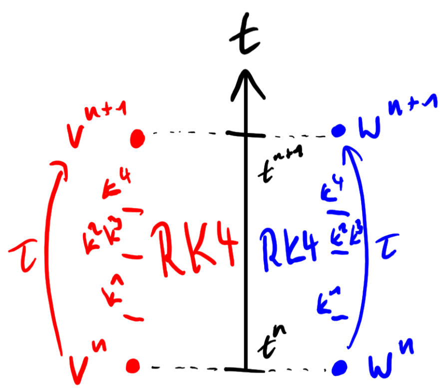
- order degradation to $\mathcal{O}(\tau)$ for the trapezoidal rule with standard coupling schemes
- order $\mathcal{O}(\tau^2)$ using a specialized coupling scheme
- similar experiments for other higher order schemes

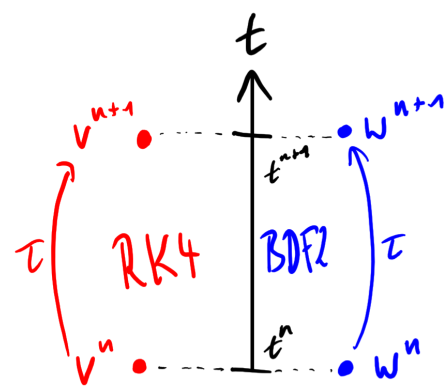
Conclusions and Outlook

Partitioned multi-physics time stepping

Today	Outlook
identical timesteps	subcycling
simple schemes	substepping
identical time stepping	inhomogeneous time stepping
$\mathcal{O}\left(\tau^2\right)$	Higher order
taylored schemes	general solution strategy
1D heat transport problem	real-world scenario







Appendix

Other 2nd order schemes

- Explicit Heun – predictor

$$\begin{aligned} \boldsymbol{u}_p, c_p &= \boldsymbol{u}^n + \tau f(\boldsymbol{u}^n, t_n, c_n) \\ \boldsymbol{u}^{n+1} &= \boldsymbol{u}^n + \frac{\tau}{2}\left[f(\boldsymbol{u}^n, t_n, c_n) + f(\boldsymbol{u}_p, t_{n+1},, c_p)\right] \end{aligned}$$

- Explicit Runge Kutta 2 – predictor

$$\begin{aligned} k_1 &= \tau f(\boldsymbol{u}^n, t_n, c_n) \\ \boldsymbol{u}_p, c_p &= \boldsymbol{u}^n + \frac{1}{2}k_1 \\ k_2 &= \tau f\left(\boldsymbol{u}_p, t_{n+\frac{1}{2}}, c_p\right) \\ \boldsymbol{u}^{n+1} &= \boldsymbol{u}^n + k_2 \end{aligned}$$

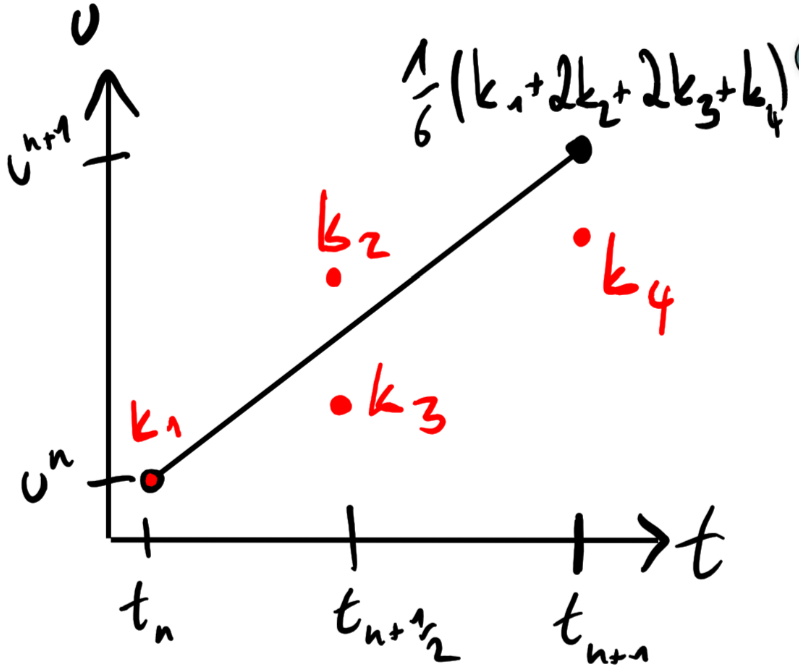
- Implicit midpoint rule – interpolation

$$\boldsymbol{u}^{n+1}, c_{n+1} = \boldsymbol{u}^n + \tau f\left(\frac{\boldsymbol{u}^n + \boldsymbol{u}^{n+1}}{2}, t_{n+\frac{1}{2}}, \frac{c_n + c_{n+1}}{2}\right)$$

Appendix

Runge Kutta 4

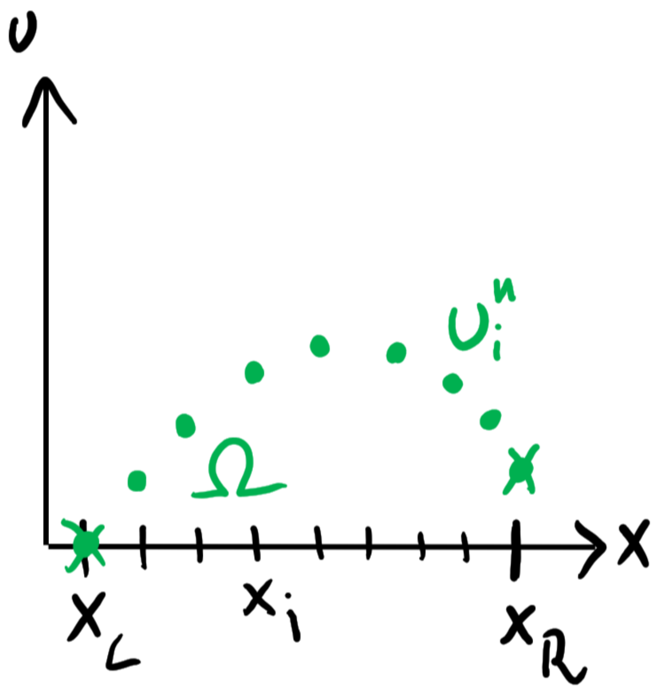
$$
\begin{aligned}
k_1 &= \tau f(\boldsymbol{u}^n, t_n) \\
k_2 &= \tau f\left(\boldsymbol{u}^n + \frac{1}{2}k_1, t_{n+\frac{1}{2}}\right) \\
k_3 &= \tau f\left(\boldsymbol{u}^n + \frac{1}{2}k_2, t_{n+\frac{1}{2}}\right) \\
k_4 &= \tau f\left(\boldsymbol{u}^n + k_3, t_n\right) \\
\boldsymbol{u}^{n+1} &= \boldsymbol{u}^n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$



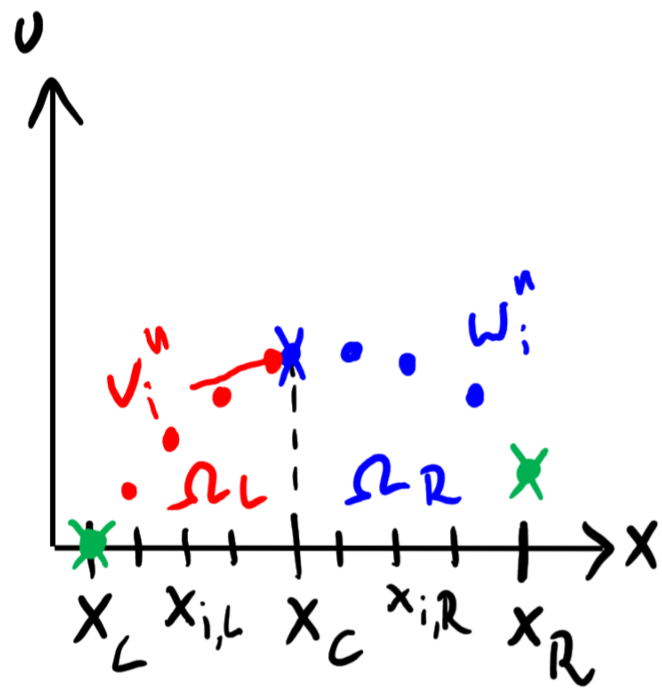
Appendix

Spatial discretization & coupling condition

Monolithic solution



Partitioned solution



Appendix

Spatial discretization & coupling condition

Dirichlet-Neumann coupling

Boundary condition for w

$$u_C^D(t) = v(x_C)$$

Boundary condition for v

$$u_C^N(t) = \frac{\partial}{\partial x} w(x_C)$$

Coupling error

We only consider the error for the left participant

$$\varepsilon^n = \left| \sum_i u_i^n - v_i^n \right|, \text{ for } x_i \in \Omega_L$$

Coupled solution