Time-parallel solution of the eddy current problem

Iryna Kulchytska-Ruchka[1,2],
Sebastian Schöps[1,2], Herbert De Gersem[1,2]

[1] *Graduate School of Computational Engineering,*
[2] *Institut für Theorie Elektromagnetischer Felder, Technische Universität Darmstadt*

4th STEAM Collaboration Meeting, Darmstadt

Outline

Introduction

- Motivation
- The eddy current problem

Parallel-in-time solution

- The Parareal method for IVPs
- Numerical example: coaxial cable model

Fourier basis for time-periodic systems

- Coarse solution by spectral collocation
- Numerical results: coaxial cable model
- Systems with nonsmooth excitations

Conclusions and outlook

Overview

Introduction

- Motivation
- The eddy current problem

Parallel-in-time solution

- The Parareal method for IVPs
- Numerical example: coaxial cable model

Fourier basis for time-periodic systems

- Coarse solution by spectral collocation
- Numerical results: coaxial cable model
- Systems with nonsmooth excitations

Conclusions and outlook

Motivation

→ Transient FEM simulation

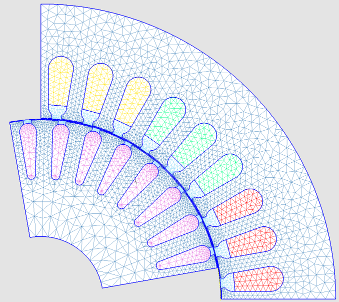


Fig.: Cross-section of an induction machine (J. Gyselinck).

Motivation

→ Transient FEM simulation

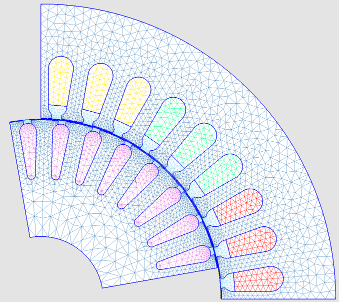


Fig.: Cross-section of an induction machine (J. Gyselinck).

→ System evolution in time

Motivation

→ Transient FEM simulation

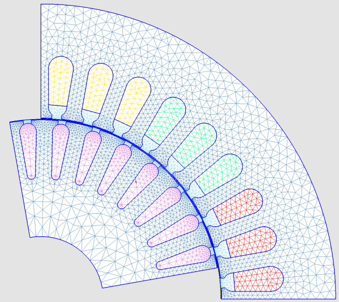
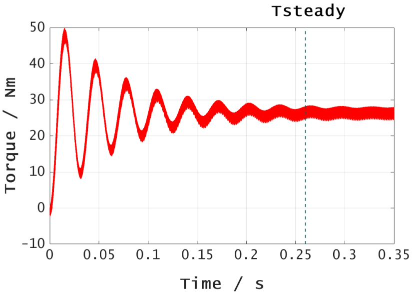


Fig.: Cross-section of an induction machine (J. Gyselinck).



→ System evolution in time

Motivation

→ Transient FEM simulation

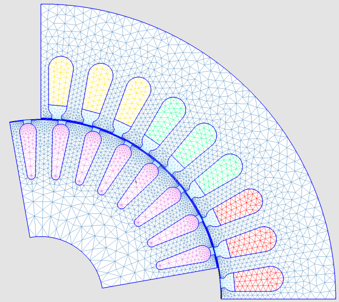
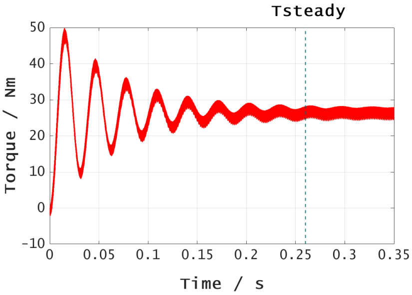


Fig.: Cross-section of an induction machine (J. Gyselinck).



- Long settling time till the steady state

→ System evolution in time

Motivation

→ Transient FEM simulation

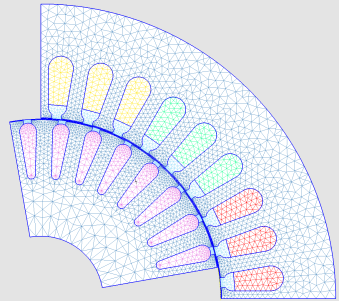
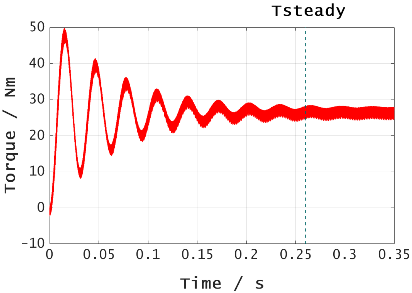


Fig.: Cross-section of an induction machine (J. Gyselinck).

→ System evolution in time



- Long settling time till the steady state
- Many time steps $\Longrightarrow$ time-consuming computation!

Overview

Introduction

- Motivation
- The eddy current problem

Parallel-in-time solution

- The Parareal method for IVPs
- Numerical example: coaxial cable model

Fourier basis for time-periodic systems

- Coarse solution by spectral collocation
- Numerical results: coaxial cable model
- Systems with nonsmooth excitations

Conclusions and outlook

The eddy current problem

Fundamentals of electromagnetism: Maxwell's equations.

Assumptions:

- Quasi-static regime: $|\mathbf{J}| \gg \left|\frac{\partial \mathbf{D}}{\partial t}\right|$;
- Neglect hysteresis.

The eddy current equation:

$$\sigma \frac{\partial \mathbf{A}}{\partial t} + \mathbf{curl}(\nu(|\,\mathbf{curl}\,\mathbf{A}|)\,\mathbf{curl}\,\mathbf{A}) = \mathbf{J}_s,$$

$\mathbf{A}$ – unknown magnetic vector potential;
$\mathbf{J}_s$ – impressed current density;
$\sigma,\ \nu$ – electric conductivity and magnetic reluctivity.

Semi-discrete problem

Solve the IVP in time:

$$\begin{aligned}\mathbf{M}\mathrm{d}_t\mathbf{u}(t) &= \mathbf{f}(t,\mathbf{u}), \qquad t \in \mathcal{I} := (0,T),\\ \mathbf{u}(0) &= \mathbf{u}_0,\end{aligned}$$

where $\mathbf{u} : \mathcal{I} \mapsto \mathbb{R}^{N_{\mathrm{dof}}}$ denotes the space-discretization of $\mathbf{A}$.

Semi-discrete problem

Solve the IVP in time:

$$\mathbf{M}\mathrm{d}_t\mathbf{u}(t) = \mathbf{f}(t,\mathbf{u}), \qquad t \in \mathcal{I} := (0, T),$$
$$\mathbf{u}(0) = \mathbf{u}_0,$$

where $\mathbf{u} : \mathcal{I} \mapsto \mathbb{R}^{N_{\text{dof}}}$ denotes the space-discretization of $\mathbf{A}$.

- $\mathbf{f}(t,\mathbf{u}) = -\mathbf{K}\mathbf{u}(t) + \mathbf{g}(t)$;
- $\mathbf{M}, \mathbf{K} \in \mathbb{R}^{N_{\text{dof}} \times N_{\text{dof}}}$ – mass and stiffness matrices;
- $\mathbf{g}(t)$ – excitation (e.g., impressed current).

Overview

Introduction

- Motivation
- The eddy current problem

Parallel-in-time solution

- The Parareal method for IVPs
- Numerical example: coaxial cable model

Fourier basis for time-periodic systems

- Coarse solution by spectral collocation
- Numerical results: coaxial cable model
- Systems with nonsmooth excitations

Conclusions and outlook

The Parareal method: splitting of the time interval

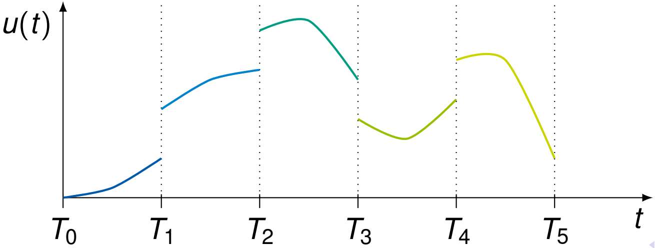
Partitioning the time interval into *N* windows (e.g., one per core) yields

$$
\begin{cases}
\mathbf{M}\mathrm{d}_t\mathbf{u}_0 = \mathbf{f}(t, \mathbf{u}_0), & \mathbf{u}_0(T_0) = \mathbf{U}_0, & t \in (T_0, T_1], \\
\mathbf{M}\mathrm{d}_t\mathbf{u}_1 = \mathbf{f}(t, \mathbf{u}_1), & \mathbf{u}_1(T_1) = \mathbf{U}_1, & t \in (T_1, T_2], \\
\quad\vdots & & \\
\mathbf{M}\mathrm{d}_t\mathbf{u}_{N-1} = \mathbf{f}(t, \mathbf{u}_{N-1}), & \mathbf{u}_{N-1}(T_{N-1}) = \mathbf{U}_{N-1}, & t \in (T_{N-1}, T_N],
\end{cases}
$$

The Parareal method: splitting of the time interval

Partitioning the time interval into N windows (e.g., one per core) yields

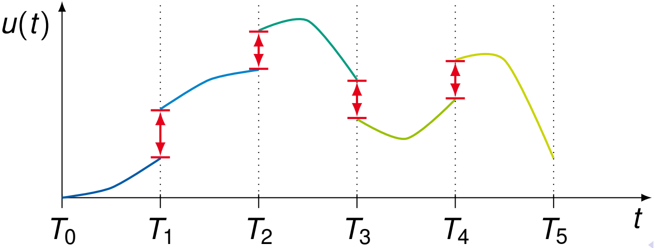
$$\begin{cases} \mathbf{M}\mathrm{d}_t\mathbf{u}_0 = \mathbf{f}(t,\mathbf{u}_0), & \mathbf{u}_0(T_0) = \mathbf{U}_0, & t \in (T_0, T_1], \\ \mathbf{M}\mathrm{d}_t\mathbf{u}_1 = \mathbf{f}(t,\mathbf{u}_1), & \mathbf{u}_1(T_1) = \mathbf{U}_1, & t \in (T_1, T_2], \\ \qquad\vdots & & \\ \mathbf{M}\mathrm{d}_t\mathbf{u}_{N-1} = \mathbf{f}(t,\mathbf{u}_{N-1}), & \mathbf{u}_{N-1}(T_{N-1}) = \mathbf{U}_{N-1}, & t \in (T_{N-1}, T_N], \end{cases}$$



The Parareal method: splitting of the time interval

Partitioning the time interval into N windows (e.g., one per core) yields

$$\begin{cases} \mathbf{M}\mathrm{d}_t\mathbf{u}_0 = \mathbf{f}(t,\mathbf{u}_0), & \mathbf{u}_0(T_0) = \mathbf{U}_0, & t \in (T_0, T_1], \\ \mathbf{M}\mathrm{d}_t\mathbf{u}_1 = \mathbf{f}(t,\mathbf{u}_1), & \mathbf{u}_1(T_1) = \mathbf{U}_1, & t \in (T_1, T_2], \\ \quad\vdots & & \\ \mathbf{M}\mathrm{d}_t\mathbf{u}_{N-1} = \mathbf{f}(t,\mathbf{u}_{N-1}), & \mathbf{u}_{N-1}(T_{N-1}) = \mathbf{U}_{N-1}, & t \in (T_{N-1}, T_N], \end{cases}$$



Parareal as the Newton-Raphson method (I)

Let $\mathcal{F}(t, T_i, U)$ be the solution operator of the IVP on $(T_i, T_{i+1}]$, for $i = 0, \ldots, N-1$, which propagates the initial value U in time.

Parareal as the Newton-Raphson method (I)

Let $\mathcal{F}(t, T_i, U)$ be the solution operator of the IVP on $(T_i, T_{i+1}]$, for $i = 0, \ldots, N-1$, which propagates the initial value U in time.

Matching conditions can be satisfied by root finding

$$\begin{cases} \mathbf{U}_1 - \mathcal{F}(T_1, T_0, \mathbf{U}_0) = 0, \\ \vdots \\ \mathbf{U}_{N-1} - \mathcal{F}(T_{N-1}, T_{N-2}, \mathbf{U}_{N-2}) = 0 \end{cases}$$

Parareal as the Newton-Raphson method (I)

Let $\mathcal{F}(t, T_i, U)$ be the solution operator of the IVP on $(T_i, T_{i+1}]$, for $i = 0, \ldots, N-1$, which propagates the initial value U in time.

Matching conditions can be satisfied by root finding

$$\begin{cases} \mathbf{U}_1 - \mathcal{F}(T_1, T_0, \mathbf{U}_0) = 0, \\ \vdots \\ \mathbf{U}_{N-1} - \mathcal{F}(T_{N-1}, T_{N-2}, \mathbf{U}_{N-2}) = 0 \end{cases}$$

or, equivalently,

$$\mathbf{F}(\mathbf{U}) = 0, \quad \text{with} \quad \mathbf{U}^T = \left(\mathbf{U}_0^T, \mathbf{U}_1^T, ..., \mathbf{U}_i^T, ..., \mathbf{U}_{N-1}^T\right).$$

M. J. Gander and E. Hairer, *Nonlinear convergence analysis for the parareal algorithm*, Domain Decomposition Methods in Science and Engineering XVII, Springer Berlin Heidelberg, 2008.

Parareal as the Newton-Raphson method (II)

The Newton-Raphson iteration: for $k = 0, 1, \dots$

$$\mathbf{U}_0^{(k+1)} = u_0,$$

$$\mathbf{U}_n^{(k+1)} = \mathcal{F}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right) + \frac{\partial \mathcal{F}(T_n, T_{n-1}, \mathbf{U})}{\partial \mathbf{U}} \left(\mathbf{U}_{n-1}^{(k+1)} - \mathbf{U}_{n-1}^{(k)}\right),$$

where $n = 1, \dots, N-1$.

Parareal as the Newton-Raphson method (II)

The Newton-Raphson iteration: for $k = 0, 1, \ldots$

$$\mathbf{U}_0^{(k+1)} = u_0,$$

$$\mathbf{U}_n^{(k+1)} = \mathcal{F}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right) + \frac{\partial \mathcal{F}(T_n, T_{n-1}, \mathbf{U})}{\partial \mathbf{U}} \left(\mathbf{U}_{n-1}^{(k+1)} - \mathbf{U}_{n-1}^{(k)}\right),$$

where $n = 1, \ldots, N-1$.

How to calculate the derivative?

Parareal as the Newton-Raphson method (II)

The Newton-Raphson iteration: for $k = 0, 1, \dots$

$$\mathbf{U}_0^{(k+1)} = u_0,$$

$$\mathbf{U}_n^{(k+1)} = \mathcal{F}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right) + \frac{\partial \mathcal{F}(T_n, T_{n-1}, \mathbf{U})}{\partial \mathbf{U}}\left(\mathbf{U}_{n-1}^{(k+1)} - \mathbf{U}_{n-1}^{(k)}\right),$$

where $n = 1, \dots, N-1$.

How to calculate the derivative?

Cheap approximation by a coarse propagator $\mathcal{G}$:

$$\frac{\partial \mathcal{F}(T_n, T_{n-1}, \mathbf{U})}{\partial \mathbf{U}}\left(\mathbf{U}_{n-1}^{(k+1)} - \mathbf{U}_{n-1}^{(k)}\right) \approx$$

$$\approx \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}\right) - \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right).$$

Parareal as the Newton-Raphson method (II)

The Newton-Raphson iteration: for $k = 0, 1, \ldots$

$$\mathbf{U}_0^{(k+1)} = u_0,$$
$$\mathbf{U}_n^{(k+1)} = \mathcal{F}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right) + \frac{\partial \mathcal{F}(T_n, T_{n-1}, \mathbf{U})}{\partial \mathbf{U}} \left(\mathbf{U}_{n-1}^{(k+1)} - \mathbf{U}_{n-1}^{(k)}\right),$$

where $n = 1, \ldots, N-1$.

How to calculate the derivative?

Cheap approximation by a coarse propagator $\mathcal{G}$:

$$\frac{\partial \mathcal{F}(T_n, T_{n-1}, \mathbf{U})}{\partial \mathbf{U}} \left(\mathbf{U}_{n-1}^{(k+1)} - \mathbf{U}_{n-1}^{(k)}\right) \approx$$
$$\approx \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}\right) - \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right).$$

Parareal as the Newton-Raphson method (III)

For $k = 0, 1, \dots$ and $n = 1, \dots, N$ solve

$$\mathbf{U}_0^{(k+1)} = u_0,$$
$$\mathbf{U}_n^{(k+1)} = \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}) + \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}) - \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}).$$

Parareal as the Newton-Raphson method (III)

For $k = 0, 1, \ldots$ and $n = 1, \ldots, N$ solve

$$\mathbf{U}_0^{(k+1)} = u_0,$$
$$\mathbf{U}_n^{(k+1)} = \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}) + \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}) - \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}).$$

Propagators:

- Fine $\tilde{\mathbf{U}}_n^{(k)} := \mathcal{F}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: e.g., backward Euler's method;
- Coarse $\bar{\mathbf{U}}_n^{(k)} := \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: lower-order scheme or the same time-integrator as $\mathcal{F}$ but with coarser discretization.

Parareal as the Newton-Raphson method (III)

For $k = 0, 1, \ldots$ and $n = 1, \ldots, N$ solve

$$\mathbf{U}_0^{(k+1)} = u_0,$$
$$\mathbf{U}_n^{(k+1)} = \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}) + \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}) - \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}).$$

Propagators:

- Fine $\tilde{\mathbf{U}}_n^{(k)} := \mathcal{F}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: e.g., backward Euler's method;
- Coarse $\bar{\mathbf{U}}_n^{(k)} := \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: lower-order scheme or the same time-integrator as $\mathcal{F}$ but with coarser discretization.

$\longrightarrow$ Parallel solution of fine problems;

Parareal as the Newton-Raphson method (III)

For $k = 0, 1, \ldots$ and $n = 1, \ldots, N$ solve

$$\mathbf{U}_0^{(k+1)} = u_0,$$
$$\mathbf{U}_n^{(k+1)} = \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}) + \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}) - \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}).$$

Propagators:

- Fine $\tilde{\mathbf{U}}_n^{(k)} := \mathcal{F}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: e.g., backward Euler's method;
- Coarse $\bar{\mathbf{U}}_n^{(k)} := \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: lower-order scheme or the same time-integrator as $\mathcal{F}$ but with coarser discretization.

$\longrightarrow$ Parallel solution of fine problems;

$\longrightarrow$ Sequential solution of coarse problems.

Considering the periodic constraint: PP-IC

PP-IC: periodic parareal algorithm with initial value coarse problem:

For $k = 0, 1, \dots$ and $n = 1, \dots, N$ solve

$$\mathbf{U}_0^{(k+1)} = \mathbf{U}_N^{(k)},$$
$$\mathbf{U}_n^{(k+1)} = \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}) + \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}) - \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}).$$

Considering the periodic constraint: PP-IC

PP-IC: periodic parareal algorithm with initial value coarse problem:

For $k = 0, 1, \dots$ and $n = 1, \dots, N$ solve

$$\mathbf{U}_0^{(k+1)} = \mathbf{U}_N^{(k)},$$
$$\mathbf{U}_n^{(k+1)} = \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}) + \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}) - \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}).$$

Propagators:

- Fine $\tilde{\mathbf{U}}_n^{(k)} := \mathcal{F}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: e.g., backward Euler's method;
- Coarse $\bar{\mathbf{U}}_n^{(k)} := \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: lower-order scheme or the same time-integrator as $\mathcal{F}$ but with coarser discretization.

Considering the periodic constraint: PP-IC

PP-IC: periodic parareal algorithm with initial value coarse problem:

For $k = 0, 1, \dots$ and $n = 1, \dots, N$ solve

$$\mathbf{U}_0^{(k+1)} = \mathbf{U}_N^{(k)},$$

$$\mathbf{U}_n^{(k+1)} = \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}) + \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}) - \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}).$$

Propagators:

- Fine $\tilde{\mathbf{U}}_n^{(k)} := \mathcal{F}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: e.g., backward Euler's method;
- Coarse $\bar{\mathbf{U}}_n^{(k)} := \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: lower-order scheme or the same time-integrator as $\mathcal{F}$ but with coarser discretization.

$\longrightarrow$ Parallel solution of fine problems;

Considering the periodic constraint: PP-IC

PP-IC: periodic parareal algorithm with initial value coarse problem:

For $k = 0, 1, \ldots$ and $n = 1, \ldots, N$ solve

$$\mathbf{U}_0^{(k+1)} = \mathbf{U}_N^{(k)},$$

$$\mathbf{U}_n^{(k+1)} = \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}) + \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}) - \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}).$$

Propagators:

- Fine $\tilde{\mathbf{U}}_n^{(k)} := \mathcal{F}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: e.g., backward Euler's method;
- Coarse $\bar{\mathbf{U}}_n^{(k)} := \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}\right)$: lower-order scheme or the same time-integrator as $\mathcal{F}$ but with coarser discretization.

⟶ Parallel solution of fine problems;

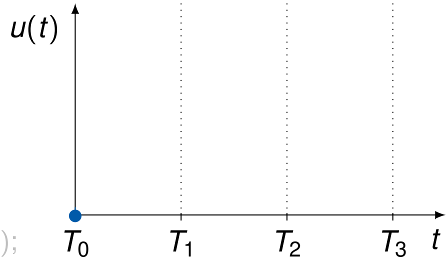
⟶ Sequential solution of coarse problems.

The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```

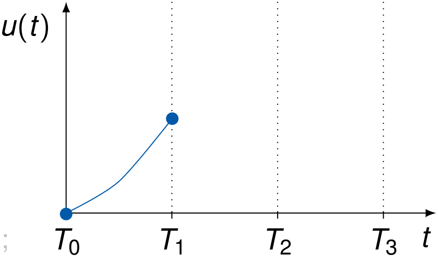
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



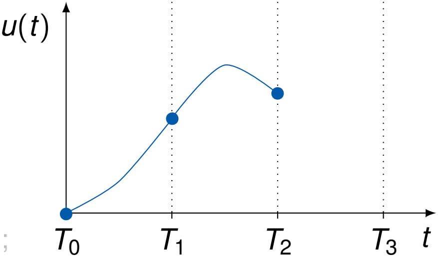
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



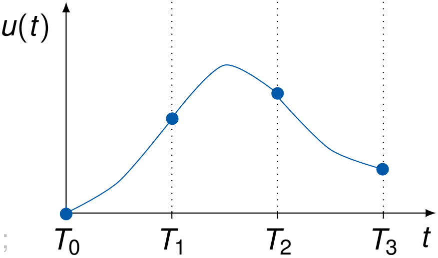
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



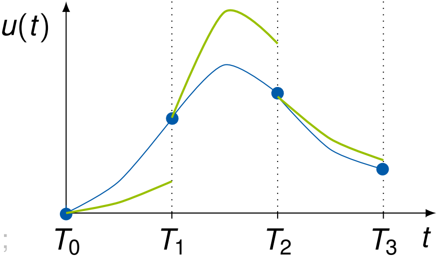
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



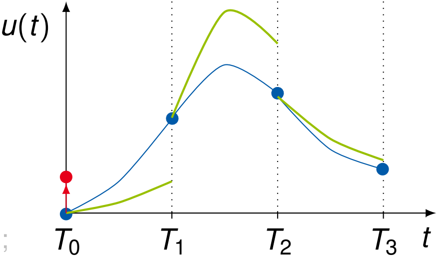
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



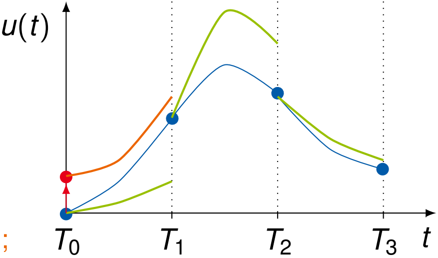
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



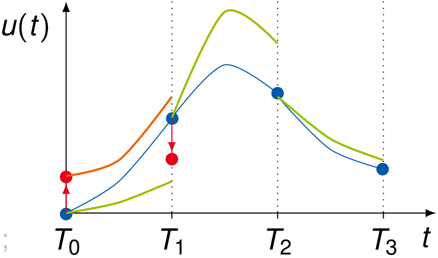
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



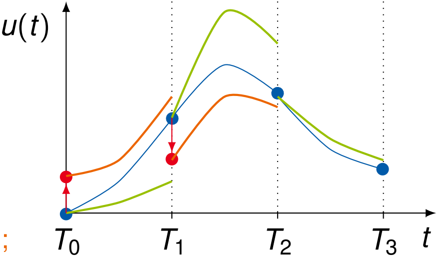
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



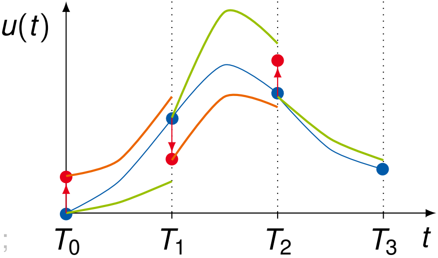
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



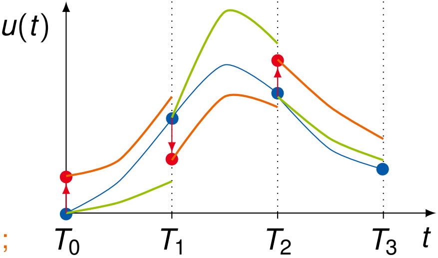
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



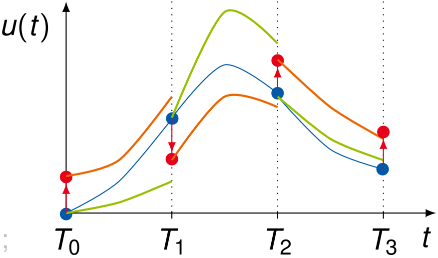
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



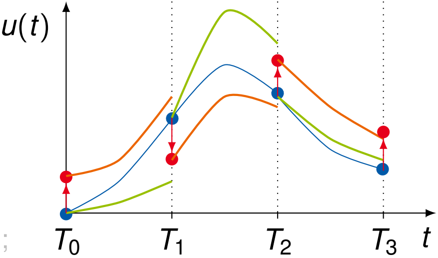
The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



The PP-IC algorithm

```
initialize: $\mathbf{U}_0^{(0)} \leftarrow \mathbf{U}_0$ and $k \leftarrow 0$;
for $n \leftarrow 1$ to $N$ do
    $\mathbf{U}_n^{(0)} \leftarrow \bar{\mathbf{U}}_n^{(0)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$;
end
while $\|error^{(k)}\| > tol$ do
    parfor $n \leftarrow 1$ to $N$ do
        $\tilde{\mathbf{U}}_n^{(k)} \leftarrow \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)})$;
    end
    $\mathbf{U}_0^{(k+1)} \leftarrow \mathbf{U}_N^{(k)}$;
    for $n \leftarrow 1$ to $N$ do
        $\bar{\mathbf{U}}_n^{(k+1)} \leftarrow \mathcal{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)})$;
        $\mathbf{U}_n^{(k+1)} \leftarrow \tilde{\mathbf{U}}_n^{(k)} + \bar{\mathbf{U}}_n^{(k+1)} - \bar{\mathbf{U}}_n^{(k)}$;
    end
    increment: $k \leftarrow k + 1$;
end
```



Overview

Introduction

- Motivation
- The eddy current problem

Parallel-in-time solution

- The Parareal method for IVPs
- Numerical example: coaxial cable model

Fourier basis for time-periodic systems

- Coarse solution by spectral collocation
- Numerical results: coaxial cable model
- Systems with nonsmooth excitations

Conclusions and outlook

Numerical example: coaxial cable model

For $t \in [0, T]$ find $\mathbf{u}(t) \in \mathbb{R}^{N_{dof}}$ s.t.

$$\begin{aligned} \mathbf{M}\mathrm{d}_t\mathbf{u}(t) + \mathbf{K}\mathbf{u}(t) &= \mathbf{X}i(t), \\ \mathbf{u}(0) &= \mathbf{u}(T), \end{aligned}$$

space-discrete eddy current problem with $N_{dof} = 2269$.

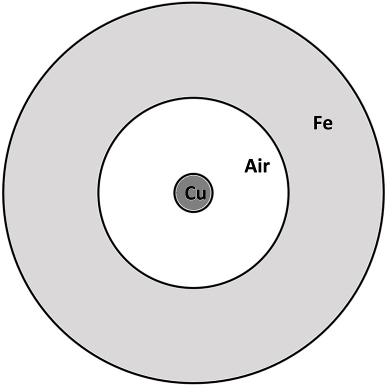


Fig.: Wire inside of a steel tube.

Numerical example: coaxial cable model

For $t \in [0, T]$ find $\mathbf{u}(t) \in \mathbb{R}^{N_{dof}}$ s.t.

$$\begin{aligned} \mathbf{M}\mathrm{d}_t\mathbf{u}(t) + \mathbf{K}\mathbf{u}(t) &= \mathbf{X}i(t), \\ \mathbf{u}(0) &= \mathbf{u}(T), \end{aligned}$$

space-discrete eddy current problem with $N_{dof} = 2269$.

$T = 0.02$, $\omega = 2\pi/T$,
$i(t) = 100 \sin(\omega t)$

Dimensions: $r_{Cu} = 0.254cm$,
$r_{Air} = 1.27cm$, $r_{Fe} = 2.54cm$.

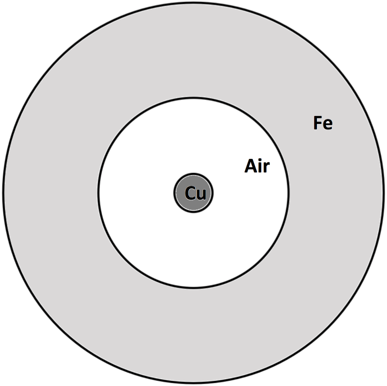


Fig.: Wire inside of a steel tube.

Coaxial cable model: convergence PP-IC

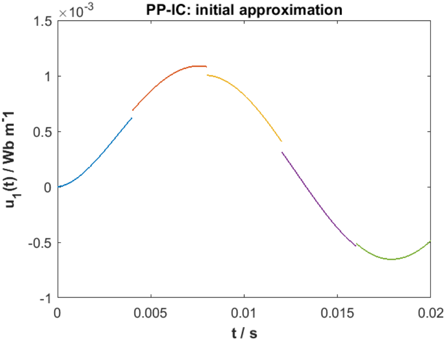


Fig.: PP-IC: initial approximation.

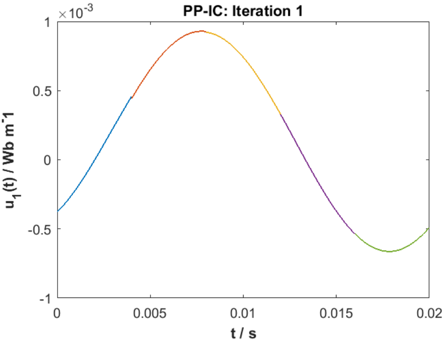


Fig.: Solution after the 1st iteration.

Iteration	1	50	125	200	260
Rel. error	$2.3 \cdot 10^{-1}$	$8.0 \cdot 10^{-4}$	$7.3 \cdot 10^{-5}$	$6.7 \cdot 10^{-6}$	$9.9 \cdot 10^{-7}$

Coaxial cable model: speed-up PP-IC

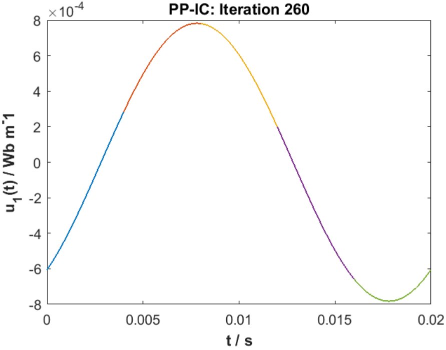


Fig.: PP-IC: solution within *tol* $\approx 10^{-6}$.

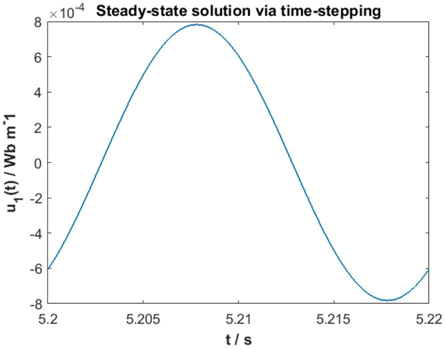


Fig.: Sequential steady-state solution.

Sequential time:	35.6 min;	Number of periods: 261
PP-IC time:	1.5 min	$\longrightarrow$ 23 times faster.

Overview

Introduction

- Motivation
- The eddy current problem

Parallel-in-time solution

- The Parareal method for IVPs
- Numerical example: coaxial cable model

Fourier basis for time-periodic systems

- Coarse solution by spectral collocation
- Numerical results: coaxial cable model
- Systems with nonsmooth excitations

Conclusions and outlook

Time-periodic problem to be solved: PP-PC

PP-PC: periodic parareal algorithm with periodic coarse problem:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \dots & -\mathcal{G}\left(T_N, T_{N-1}, \cdot\right) \\ -\mathcal{G}\left(T_1, T_0, \cdot\right) & \mathbf{I} & & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & -\mathcal{G}\left(T_{N-1}, T_{N-2}, \cdot\right) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0^{(k+1)} \\ \mathbf{U}_1^{(k+1)} \\ \vdots \\ \mathbf{U}_{N-1}^{(k+1)} \end{bmatrix} =$$

$$= \begin{bmatrix} \mathcal{F}\left(T_N, T_{N-1}, \mathbf{U}_{N-1}^{(k)}\right) - \mathcal{G}\left(T_N, T_{N-1}, \mathbf{U}_{N-1}^{(k)}\right) \\ \mathcal{F}\left(T_1, T_0, \mathbf{U}_0^{(k)}\right) - \mathcal{G}\left(T_1, T_0, \mathbf{U}_0^{(k)}\right) \\ \vdots \\ \mathcal{F}\left(T_{N-1}, T_{N-2}, \mathbf{U}_{N-2}^{(k)}\right) - \mathcal{G}\left(T_{N-1}, T_{N-2}, \mathbf{U}_{N-2}^{(k)}\right) \end{bmatrix}$$

Time-periodic problem to be solved: PP-PC

PP-PC: periodic parareal algorithm with periodic coarse problem:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \dots & -\mathcal{G}\left(T_N, T_{N-1}, \cdot\right) \\ -\mathcal{G}\left(T_1, T_0, \cdot\right) & \mathbf{I} & & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & -\mathcal{G}\left(T_{N-1}, T_{N-2}, \cdot\right) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0^{(k+1)} \\ \mathbf{U}_1^{(k+1)} \\ \vdots \\ \mathbf{U}_{N-1}^{(k+1)} \end{bmatrix} =$$

$$= \begin{bmatrix} \mathcal{F}\left(T_N, T_{N-1}, \mathbf{U}_{N-1}^{(k)}\right) - \mathcal{G}\left(T_N, T_{N-1}, \mathbf{U}_{N-1}^{(k)}\right) \\ \mathcal{F}\left(T_1, T_0, \mathbf{U}_0^{(k)}\right) - \mathcal{G}\left(T_1, T_0, \mathbf{U}_0^{(k)}\right) \\ \vdots \\ \mathcal{F}\left(T_{N-1}, T_{N-2}, \mathbf{U}_{N-2}^{(k)}\right) - \mathcal{G}\left(T_{N-1}, T_{N-2}, \mathbf{U}_{N-2}^{(k)}\right) \end{bmatrix}$$

⟶ Large size and inconvenient structure!

PP-PC: backward Euler's method as coarse solver

Assume $y_n^{(k+1)} := \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}\right)$ is defined by the backward Euler's method:

$$\underbrace{\left(\frac{\mathbf{M}}{\Delta t} + \mathbf{K}\right)}_{=:\mathbf{Q}} y_n^{(k+1)} - \underbrace{\frac{\mathbf{M}}{\Delta t}}_{=:\mathbf{C}} \mathbf{U}_{n-1}^{(k+1)} = g(T_n)$$

for $n = 1, \ldots, N$ with $\Delta t = T/N$.

PP-PC: backward Euler's method as coarse solver

Assume $y_n^{(k+1)} := \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}\right)$ is defined by the backward Euler's method:

$$\underbrace{\left(\frac{\mathbf{M}}{\Delta t} + \mathbf{K}\right)}_{=:\mathbf{Q}} y_n^{(k+1)} - \underbrace{\frac{\mathbf{M}}{\Delta t}}_{=:\mathbf{C}} \mathbf{U}_{n-1}^{(k+1)} = g(T_n)$$

for $n = 1, \ldots, N$ with $\Delta t = T/N$.The system reads:

$$\underbrace{\begin{bmatrix} \mathbf{Q} & & & -\mathbf{C} \\ -\mathbf{C} & \mathbf{Q} & & \\ & \ddots & \ddots & \\ & & -\mathbf{C} & \mathbf{Q} \end{bmatrix}}_{=:\mathbf{G}} \begin{bmatrix} \mathbf{U}_0^{(k+1)} \\ \mathbf{U}_1^{(k+1)} \\ \vdots \\ \mathbf{U}_{N-1}^{(k+1)} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{Q}\mathcal{F}\left(T_N, T_{N-1}, \mathbf{U}_{N-1}^{(k)}\right) - \mathbf{C}\mathbf{U}_{N-1}^{(k)}) \\ \mathbf{Q}\mathcal{F}\left(T_1, T_0, \mathbf{U}_0^{(k)}\right) - \mathbf{C}\mathbf{U}_0^{(k)} \\ \vdots \\ \mathbf{Q}\mathcal{F}\left(T_{N-1}, T_{N-2}, \mathbf{U}_{N-2}^{(k)}\right) - \mathbf{C}\mathbf{U}_{N-2}^{(k)} \end{bmatrix}}_{=:\mathbf{r}^{(k)}}$$

PP-PC: backward Euler's method as coarse solver

Assume $y_n^{(k+1)} := \mathcal{G}\left(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}\right)$ is defined by the backward Euler's method:

$$\underbrace{\left(\frac{\mathbf{M}}{\Delta t} + \mathbf{K}\right)}_{=:\mathbf{Q}} y_n^{(k+1)} - \underbrace{\frac{\mathbf{M}}{\Delta t}}_{=:\mathbf{C}} \mathbf{U}_{n-1}^{(k+1)} = g(T_n)$$

for $n = 1, \ldots, N$ with $\Delta t = T/N$.The system reads:

$$\underbrace{\begin{bmatrix} \mathbf{Q} & & & -\mathbf{C} \\ -\mathbf{C} & \mathbf{Q} & & \\ & \ddots & \ddots & \\ & & -\mathbf{C} & \mathbf{Q} \end{bmatrix}}_{=:\mathbf{G}} \begin{bmatrix} \mathbf{U}_0^{(k+1)} \\ \mathbf{U}_1^{(k+1)} \\ \vdots \\ \mathbf{U}_{N-1}^{(k+1)} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{Q}\mathcal{F}\left(T_N, T_{N-1}, \mathbf{U}_{N-1}^{(k)}\right) - \mathbf{C}\mathbf{U}_{N-1}^{(k)}) \\ \mathbf{Q}\mathcal{F}\left(T_1, T_0, \mathbf{U}_0^{(k)}\right) - \mathbf{C}\mathbf{U}_0^{(k)} \\ \vdots \\ \mathbf{Q}\mathcal{F}\left(T_{N-1}, T_{N-2}, \mathbf{U}_{N-2}^{(k)}\right) - \mathbf{C}\mathbf{U}_{N-2}^{(k)} \end{bmatrix}}_{=:\mathbf{r}^{(k)}}$$

Coarse solution via the spectral basis

$\mathbf{U}_0, \dots \mathbf{U}_{N-1}$ are values of a **periodic function** at $t_0, \dots, t_{N-1}$.

Coarse solution via the spectral basis

$\mathbf{U}_0, \ldots \mathbf{U}_{N-1}$ are values of a **periodic function** at $t_0, \ldots, t_{N-1}$.

$\longrightarrow$ can be expanded into the Fourier series over the period $[0, T]$:

Coarse solution via the spectral basis

$\mathbf{U}_0, \dots \mathbf{U}_{N-1}$ are values of a **periodic function** at $t_0, \dots, t_{N-1}$.

$\longrightarrow$ can be expanded into the Fourier series over the period $[0, T]$:

$$\mathbf{U}(t) = \sum_{m \in \mathcal{M}} \hat{\mathbf{U}}_m \exp\left(\imath \omega_m t\right),$$

with the frequencies $\omega_m = 2\pi m / T$, and $\mathcal{M} := \{-N/2 + 1, \dots, N/2\}$.

Coarse solution via the spectral basis

$\mathbf{U}_0, \ldots \mathbf{U}_{N-1}$ are values of a **periodic function** at $t_0, \ldots, t_{N-1}$.

$\longrightarrow$ can be expanded into the Fourier series over the period $[0, T]$:

$$\mathbf{U}(t) = \sum_{m \in \mathcal{M}} \hat{\mathbf{U}}_m \exp\left(\imath \omega_m t\right),$$

with the frequencies $\omega_m = 2\pi m / T$, and $\mathcal{M} := \{-N/2 + 1, \ldots, N/2\}$.

At the kth iteration

$$\mathbf{U}_n^{(k+1)} = \sum_{m \in \mathcal{M}} \hat{\mathbf{U}}_m^{(k+1)} \exp\left(\imath \omega_m t_n\right), \quad n = 0, \ldots, N-1.$$

Coarse solution via the spectral basis

$\mathbf{U}_0, \ldots \mathbf{U}_{N-1}$ are values of a **periodic function** at $t_0, \ldots, t_{N-1}$.

⟶ can be expanded into the Fourier series over the period $[0, T]$:

$$\mathbf{U}(t) = \sum_{m \in \mathcal{M}} \hat{\mathbf{U}}_m \exp\left(\imath \omega_m t\right),$$

with the frequencies $\omega_m = 2\pi m / T$, and $\mathcal{M} := \{-N/2+1, \ldots, N/2\}$.

At the kth iteration

$$\mathbf{U}_n^{(k+1)} = \sum_{m \in \mathcal{M}} \hat{\mathbf{U}}_m^{(k+1)} \exp\left(\imath \omega_m t_n\right), \quad n = 0, \ldots, N-1.$$

⟶ into the PP-PC system.

Frequency domain solution of the coarse problem

Let **F** denote the discrete Fourier transform matrix:

$$\mathbf{F}_{pq} = \frac{1}{\sqrt{N}} \exp(-\imath \omega_p t_q)$$

and $\tilde{\mathbf{F}} = \mathbf{F} \otimes \mathbf{I}$, with $\mathbf{I} \in \mathbb{R}^{N_{dof} \times N_{dof}}$.

Frequency domain solution of the coarse problem

Let **F** denote the discrete Fourier transform matrix:

$$\mathbf{F}_{pq} = \frac{1}{\sqrt{N}} \exp(-\imath \omega_p t_q)$$

and $\tilde{\mathbf{F}} = \mathbf{F} \otimes \mathbf{I}$, with $\mathbf{I} \in \mathbb{R}^{N_{dof} \times N_{dof}}$. Restriction to the spectral basis in the discrete PP-PC system is equivalent to the solution of

$$\underbrace{\left(\tilde{\mathbf{F}}\,\mathbf{G}\,\tilde{\mathbf{F}}^H\right)}_{=:\hat{\mathbf{G}}} \hat{\mathbf{U}}^{(k+1)} = \underbrace{\tilde{\mathbf{F}}\mathbf{r}^{(k)}}_{=:\hat{\mathbf{r}}^{(k)}}$$

for $\hat{\mathbf{U}}^{(k+1)^T} = \left[\hat{\mathbf{U}}_{-N/2+1}^{(k+1)^T}, \dots, \hat{\mathbf{U}}_{N/2}^{(k+1)^T}\right]$ in the frequency domain.

Frequency domain solution of the coarse problem

Let **F** denote the discrete Fourier transform matrix:

$$\mathbf{F}_{pq} = \frac{1}{\sqrt{N}} \exp(-\imath \omega_p t_q)$$

and $\tilde{\mathbf{F}} = \mathbf{F} \otimes \mathbf{I}$, with $\mathbf{I} \in \mathbb{R}^{N_{dof} \times N_{dof}}$. Restriction to the spectral basis in the discrete PP-PC system is equivalent to the solution of

$$\underbrace{\left(\tilde{\mathbf{F}}\ \mathbf{G}\ \tilde{\mathbf{F}}^H\right)}_{=:\hat{\mathbf{G}}} \hat{\mathbf{U}}^{(k+1)} = \underbrace{\tilde{\mathbf{F}}\mathbf{r}^{(k)}}_{=:\hat{\mathbf{r}}^{(k)}}$$

for $\hat{\mathbf{U}}^{(k+1)^T} = \left[\hat{\mathbf{U}}^{(k+1)^T}_{-N/2+1}, \dots, \hat{\mathbf{U}}^{(k+1)^T}_{N/2}\right]$ in the frequency domain.

$\mathbf{G}$ – block-circulant → $\hat{\mathbf{G}}$ – block-diagonal:

Frequency domain solution of the coarse problem

Let **F** denote the discrete Fourier transform matrix:

$$\mathbf{F}_{pq} = \frac{1}{\sqrt{N}} \exp(-\imath \omega_p t_q)$$

and $\tilde{\mathbf{F}} = \mathbf{F} \otimes \mathbf{I}$, with $\mathbf{I} \in \mathbb{R}^{N_{dof} \times N_{dof}}$. Restriction to the spectral basis in the discrete PP-PC system is equivalent to the solution of

$$\underbrace{\left(\tilde{\mathbf{F}}\ \mathbf{G}\ \tilde{\mathbf{F}}^H\right)}_{=:\hat{\mathbf{G}}} \hat{\mathbf{U}}^{(k+1)} = \underbrace{\tilde{\mathbf{F}}\mathbf{r}^{(k)}}_{=:\hat{\mathbf{r}}^{(k)}}$$

for $\hat{\mathbf{U}}^{(k+1)^T} = \left[\hat{\mathbf{U}}_{-N/2+1}^{(k+1)^T}, \ldots, \hat{\mathbf{U}}_{N/2}^{(k+1)^T}\right]$ in the frequency domain.

$\mathbf{G}$ – block-circulant → $\hat{\mathbf{G}}$ – block-diagonal: $\hat{\mathbf{G}}_{pp} = \mathbf{Q} - \mathbf{C} \exp(-\imath \Delta t \omega_p)$.

Frequency domain solution of the coarse problem

Solve for each harmonic component independently:

$$\Big\{\mathbf{Q} - \mathbf{C}\exp(-\imath\Delta t\omega_p)\Big\}\hat{\mathbf{U}}_p^{(k+1)} = \hat{\mathbf{r}}_p^{(k)},\ p = -N/2+1,\dots,N/2.$$

Frequency domain solution of the coarse problem

Solve for each harmonic component independently:

$$\left\{\mathbf{Q}-\mathbf{C}\exp(-\imath\Delta t\omega_p)\right\}\hat{\mathbf{U}}_p^{(k+1)}=\hat{\mathbf{r}}_p^{(k)},\ p=-N/2+1,\dots,N/2.$$

$\longrightarrow$ Allows parallelization on the coarse level: one needs to solve N separate systems of N_{dof} equations.

Frequency domain solution of the coarse problem

Solve for each harmonic component independently:

$$\left\{\mathbf{Q} - \mathbf{C}\exp(-\imath\Delta t\omega_p)\right\}\hat{\mathbf{U}}_p^{(k+1)} = \hat{\mathbf{r}}_p^{(k)},\ p = -N/2+1,\ldots,N/2.$$

$\longrightarrow$ Allows parallelization on the coarse level: one needs to solve N separate systems of N_{dof} equations.

Solution in the time domain is obtained by the inverse Fourier transformation:

$$\mathbf{U}^{(k+1)} = \tilde{\mathbf{F}}^H\hat{\mathbf{U}}^{(k+1)}.$$

Frequency domain solution of the coarse problem

Solve for each harmonic component independently:

$$\left\{\mathbf{Q} - \mathbf{C}\exp(-\imath \Delta t \omega_p)\right\}\hat{\mathbf{U}}_p^{(k+1)} = \hat{\mathbf{r}}_p^{(k)},\ p = -N/2+1, \ldots, N/2.$$

⟶ Allows parallelization on the coarse level: one needs to solve N separate systems of N_{dof} equations.

Solution in the time domain is obtained by the inverse Fourier transformation:

$$\mathbf{U}^{(k+1)} = \tilde{\mathbf{F}}^H \hat{\mathbf{U}}^{(k+1)}.$$

⟶ Can be calculated using the fast Fourier transform algorithm;

⟶ Matrices $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{F}}^H$ do not have to be explicitly constructed.

Overview

Introduction
- Motivation
- The eddy current problem

Parallel-in-time solution
- The Parareal method for IVPs
- Numerical example: coaxial cable model

Fourier basis for time-periodic systems
- Coarse solution by spectral collocation
- Numerical results: coaxial cable model
- Systems with nonsmooth excitations

Conclusions and outlook

Coaxial cable model: convergence PP-PC

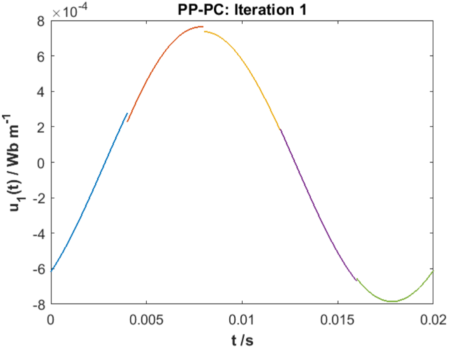


Fig.: Solution after the 1st iteration

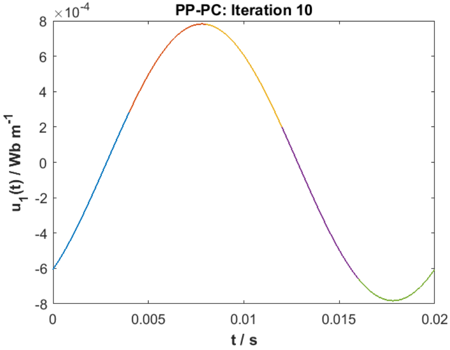
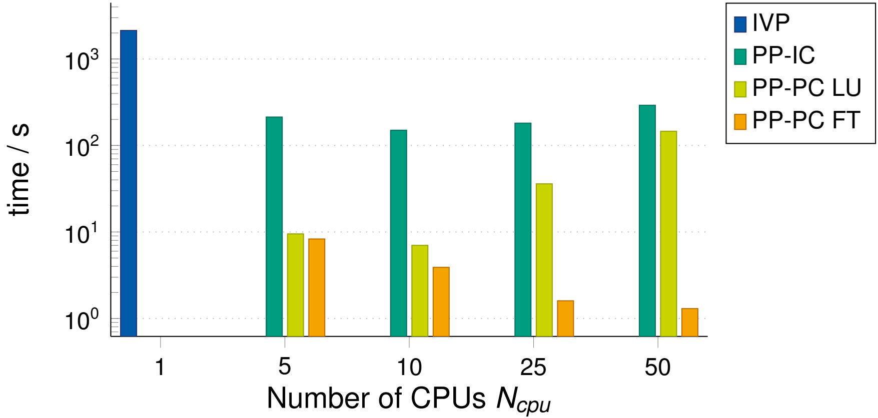


Fig.: Solution after the 10th iteration

Iteration	1	3	5	7	10
Rel. error	$2.0 \cdot 10^{-1}$	$6.3 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	$2.0 \cdot 10^{-5}$	$3.7 \cdot 10^{-7}$

Comparison of computational time: strong scaling



All the results are obtained for $N_{dof} = 2269$ degrees of freedom, rel. error $\approx 10^{-6}$.

Overview

Introduction

- Motivation
- The eddy current problem

Parallel-in-time solution

- The Parareal method for IVPs
- Numerical example: coaxial cable model

Fourier basis for time-periodic systems

- Coarse solution by spectral collocation
- Numerical results: coaxial cable model
- Systems with nonsmooth excitations

Conclusions and outlook

Systems with nonsmooth excitations

PWM (pulse width modulation): high-order components in the frequency spectra of exciting currents/voltages.

$$\mathbf{M}\mathrm{d}_t\mathbf{u}(t) = \bar{\mathbf{f}}(t,\mathbf{u}) + \tilde{\mathbf{f}}(t), \quad t \in \mathcal{I}$$
$$\mathbf{u}(0) = \mathbf{u}(T),$$

- $\bar{\mathbf{f}}$ smooth input;
- $\tilde{\mathbf{f}}$ piecewise continuous, square integrable on $\mathcal{I}$.

$\longrightarrow$ Very fine discretization required to capture the pulses.

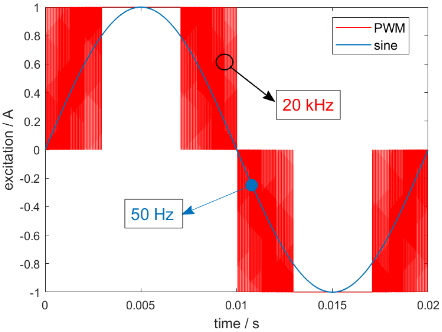
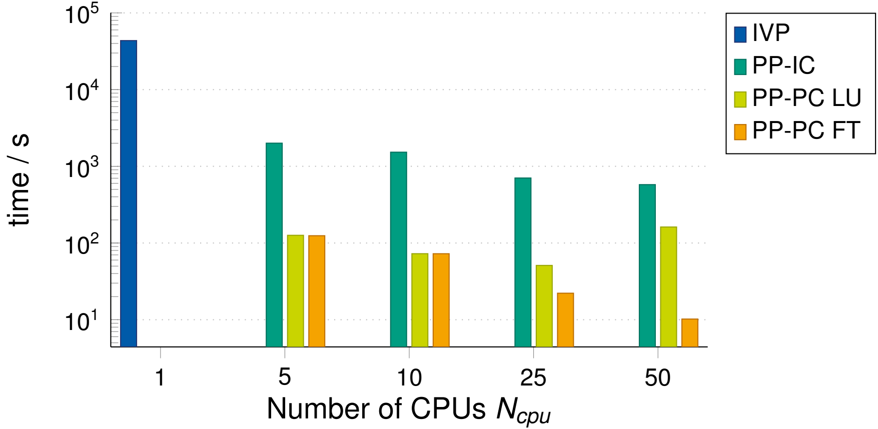


Fig.: PWM signal with a switching frequency of 20 kHz and a sine wave of 50 Hz.

Comparison of computational time: PWM input



All the results are obtained for $N_{dof} = 2269$ degrees of freedom, rel. error $\approx 10^{-6}$.

Conclusions and outlook

Conclusions

- Parareal method significantly accelerates convergence to the steady state;

Conclusions and outlook

Conclusions

- Parareal method significantly accelerates convergence to the steady state;

Outlook

- Consider nonlinear problems;
- Application to an electrical machine;

Thank you!

ACKNOWLEDGEMENT

This work has been supported by the Excellence Initiative of the German Federal and State Governments and the Graduate School of Computational Engineering at Technische Universität Darmstadt.

- S. Schöps, I. Niyonzima, and M. Clemens, *Parallel-in-time Simulation of the Eddy Current Problem using Parareal*, 21st Conference on the Computation of Electromagnetic Fields (COMPUMAG 2017), Daejeon, Korea, June 2017.
- M. J. Gander, Y.-L. Jiang, B. Song, and H. Zhang, *Analysis of two parareal algorithms for time-periodic problems*, SIAM J. Sci. Comput. 35 (5), 2013.
- J. Gyselinck et. al. *A General Method for the Frequency Domain FE Modeling of Rotating Electromagnetic Devices*, IEEE Trans. Magn. 39 (3), 2003.