PinT methods for the monodomain model in cardiac electrophysiology

Giacomo Rosilho de Souza, Rolf Krause

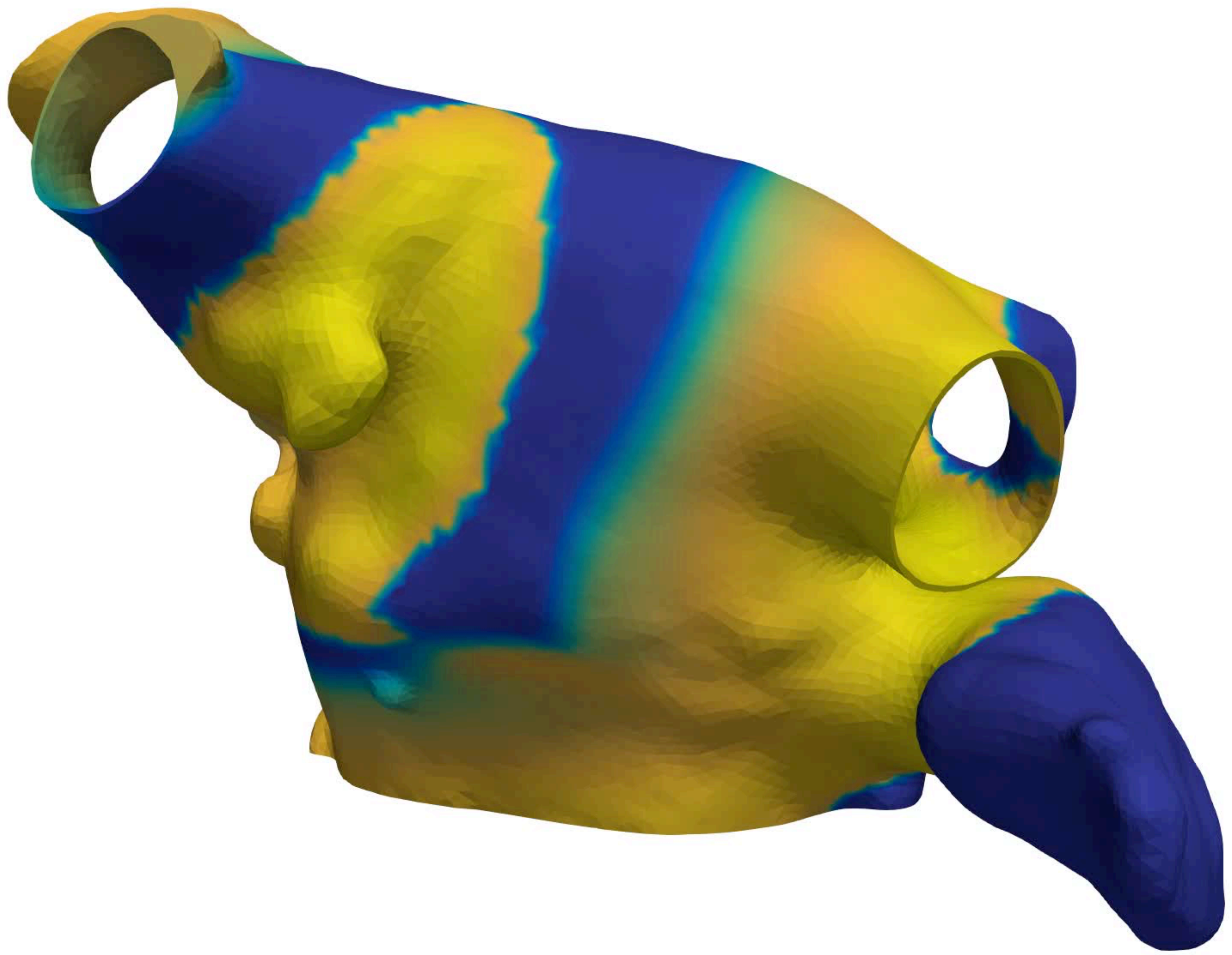
Università della Svizzera Italiana, Lugano, Switzerland

PinT workshop - Bruges - February 2024

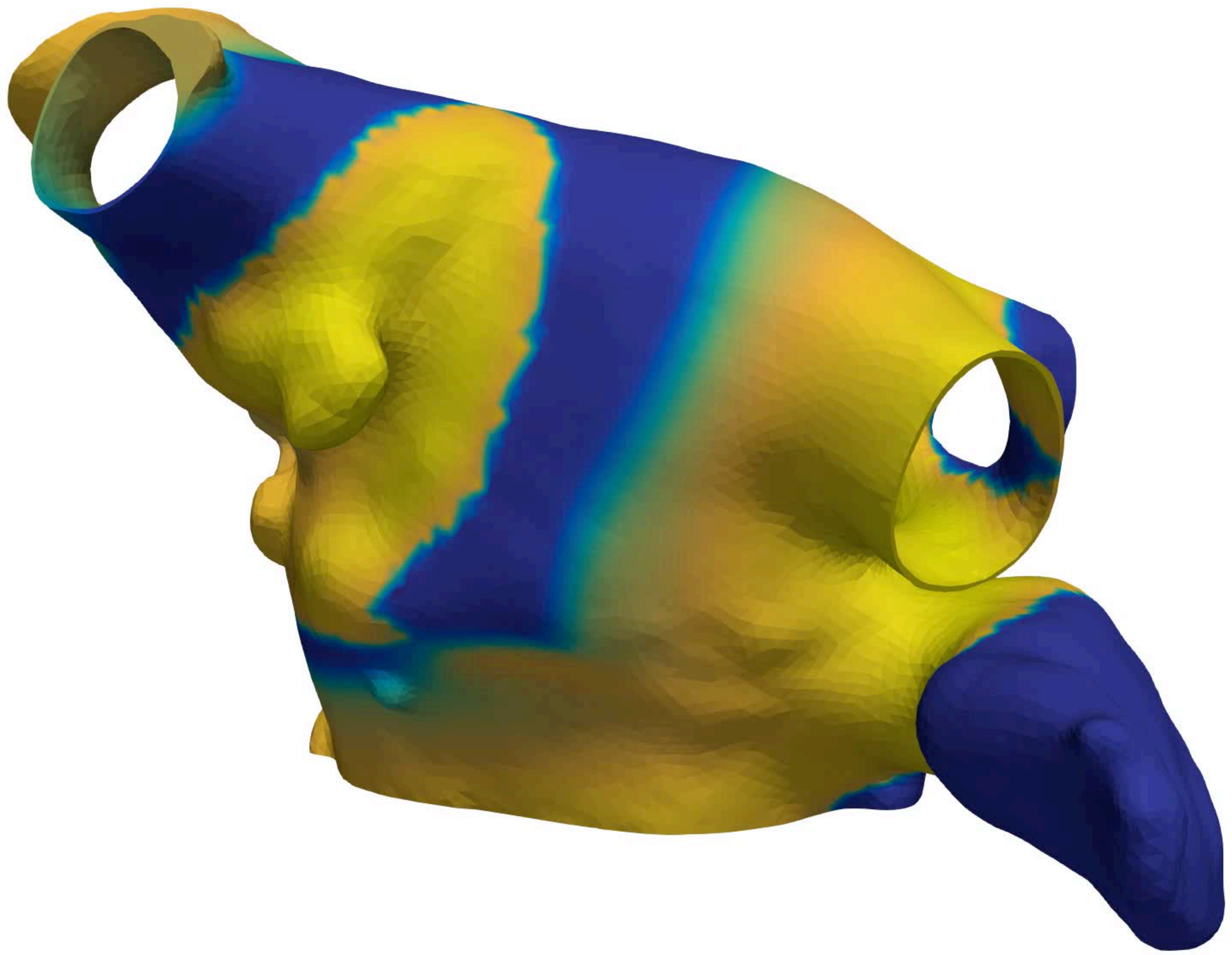
Index

- The monodomain equation and the serial way,
- The first attempt to PinT,
- The one that worked,
- Numerical results,
- Current challenges.

How does it look?



How does it look?



Monodomain model for cardiac electrophysiology

Monodomain model describes propagation of electric potential V in heart muscles:

$$
\begin{aligned}
\chi C_m V' &= \nabla \cdot (\sigma \nabla V) - \chi I_{ion}(V, z) && \text{in } \Omega \times [0,T],\\
z_E' &= g(V, z_e, z_E) && \text{in } \Omega \times [0,T],\\
z_e' &= \Lambda_e(V)(z_e - z_{e,\infty}(V)) && \text{in } \Omega \times [0,T],\\
&+ b.c.
\end{aligned}
$$

With V the electric potential and z_e, z_E the state variables of the ionic model.

$$
\begin{aligned}
V &: \Omega \times [0,T] \to \mathbb{R}\\
z_E &: \Omega \times [0,T] \to \mathbb{R}^{n_1}\\
z_e &: \Omega \times [0,T] \to \mathbb{R}^{n_2}
\end{aligned}
$$

A total of $N = 1 + n_1 + n_2$ state variables. In most applications $10 \leq N \leq 50$.

Total dofs $= N \cdot \#meshdofs$

Monodomain model for cardiac electrophysiology

Monodomain model describes propagation of electric potential V in heart muscles:

$$\begin{aligned}
\chi C_m V' &= \nabla\cdot(\sigma\nabla V) - \chi I_{ion}(V,z) && \text{in } \Omega\times[0,T],\\
z_E' &= g(V,z_e,z_E) && \text{in } \Omega\times[0,T],\\
z_e' &= \Lambda_e(V)(z_e - z_{e,\infty}(V)) && \text{in } \Omega\times[0,T],\\
&+b.c.
\end{aligned}$$

- PDE: diffusion + non stiff reaction
- Non stiff system of coupled ODEs:

$$\begin{cases} z_{E,1}' &= g_1(V,z_e,z_E)\\ &\vdots\\ z_{E,n_1}' &= g_{n_1}(V,z_e,z_E)\end{cases}$$

- Very stiff system of “uncoupled” ODEs:

$$\begin{cases} z_{e,1}' &= \lambda_1(V)(z_{e,1} - z_{\infty,1}(V))\\ &\vdots\\ z_{e,n_2}' &= \lambda_{n_2}(V)(z_{e,n_2} - z_{\infty,n_2}(V))\end{cases}$$

$$\Lambda_e(V) = \begin{pmatrix} \lambda_1(V) & & \\ & \ddots & \\ & & \lambda_{n_2}(V)\end{pmatrix}$$

$: \Omega\times[0,T]\to\mathbb{R}$

With V the electric potential and z_e, z_E the state variables of the ionic model.

$$V\colon \Omega\times[0,T]\to\mathbb{R}$$
$$z_E\colon \Omega\times[0,T]\to\mathbb{R}^{n_1}$$
$$z_e\colon \Omega\times[0,T]\to\mathbb{R}^{n_2}$$

A total of $N = 1 + n_1 + n_2$ state variables. In most applications $10 \le N \le 50$.

Total dofs = $N \cdot \#meshdofs$

Monodomain model for cardiac electrophysiology

After spatial discretization, we get the ODE system:

$$\begin{aligned} \chi C_m V' &= \nabla\cdot(\sigma\nabla V) - \chi I_{ion}(V,z) && \text{in } \Omega\times[0,T], \\ z_E' &= g(V,z_e,z_E) && \text{in } \Omega\times[0,T], \\ z_e' &= \Lambda_e(V)(z_e - z_{e,\infty}(V)) && \text{in } \Omega\times[0,T] \end{aligned} \quad \Longrightarrow \quad \begin{aligned} \mathbf{V}' &= A\mathbf{V} - I_{ion}(\mathbf{V},\mathbf{z}_E,\mathbf{z}_e), \\ \mathbf{z}_E' &= g_E(\mathbf{V},\mathbf{z}_E,\mathbf{z}_e), \\ \mathbf{z}_e' &= \Lambda_e(\mathbf{V})(\mathbf{z}_e - \mathbf{z}_{e,\infty}(\mathbf{V})). \end{aligned}$$

$$\begin{pmatrix} \mathbf{V}' \\ \mathbf{z}_E' \\ \mathbf{z}_e' \end{pmatrix} = \begin{pmatrix} A\mathbf{V} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -I_{ion}(\mathbf{V},\mathbf{z}_E,\mathbf{z}_e) \\ g_E(\mathbf{V},\mathbf{z}_E,\mathbf{z}_e) \\ 0 \end{pmatrix} + \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Lambda_e(V) \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \mathbf{z}_e - \mathbf{z}_{e,\infty}(\mathbf{V}) \end{pmatrix}}_{\Lambda(y)(y-y_\infty(y))}$$

With $y = (\mathbf{V},\mathbf{z}_E,\mathbf{z}_e)$:

$$y' = f_I(y) + f_E(y) + f_e(y)$$

Very <u>stiff</u>, but “<u>linear” and</u> <u>diagonal</u>.

e = exponential

Standard linear <u>diffusion</u> term.

I = implicit

Very expensive and highly non linear, but <u>non stiff</u>.

E = explicit

The standard way of solving it: IMEX-RL

In most applications it is solved with a splitting method based on:

- IMEX scheme for the parabolic part,
- Rush—Larsen scheme for the ionic model.

Basically, it's a festival of Euler schemes:

1. $y^1 = y_n + \Delta t\ \phi_1(\Delta t \Lambda(y_n))\, f_e(y_n)$, (exponential Euler)
2. $y^2 = y^1 + \Delta t\ f_E(y^1)$, (explicit Euler)
3. $y^3 = y^2 + \Delta t\ f_I(y^3)$, (implicit Euler)
4. $y_{n+1} = y^3$,

where $\phi_1(z) = \dfrac{e^z - 1}{z}$.

- Order $p = 1$ scheme.
- The severe stiffness of f_e is smoothed out thanks to ϕ_1.
- Stiffness from diffusion is dealt by the implicit Euler method.
- The exponential term is relatively cheap to evaluate due to its diagonal form.
- Higher order methods (up to three?) exist but with increasing complexity for compensating the splitting errors.
- This is the most used approach, due to robustness and simplicity.
- Let us design a PinT method based on this.

Spectral Deferred Correction (SDC)

Consider equation: $y' = f(y) = f_I(y) + f_E(y) + f_e(y)$

Collocation methods: Start from the approximation

$$y(t_i) = y_0 + \int_0^{t_i} f(y(s))\mathrm{d}s \approx y_0 + \int_0^{t_i} \sum_{j=1}^{m} \ell_j(s) f(y(t_j))\mathrm{d}s$$

And solve the discrete system:

$$\begin{aligned} y_i &= y_0 + \Delta t \sum_{j=1}^{m} a_{ij} f(y_j) \quad i = 1, \dots, m \\ &= y_0 + I(y)_i \end{aligned}$$

With

$$a_{ij} = \int_0^{c_i} \ell_j(s)\mathrm{d}s, \qquad I(y)_i = \Delta t \sum_{j=1}^{m} a_{ij} f(y_j) \,.$$

SDC[1]: Solves the discrete system iteratively:

i) Initial guess $\tilde{y}_i$,

ii) Error: $\delta_i = y_i - \tilde{y}_i$

iii) Residual $\varepsilon_i = \tilde{y}_i - y_0 - I(\tilde{y})_i$

iv) Derive error equation $\delta_i = \varepsilon_i + I(\tilde{y} + \delta)_i - I(\tilde{y})_i$.

Solve error equation approximatively: with $\hat{y}_i = \tilde{y}_i + \delta_i$

$$\begin{aligned} \hat{y}_{i+1} = \hat{y}_i + \Delta t \sum_{j=1}^{m} s_{ij} \left(f_I(\tilde{y}_j) + f_E(\tilde{y}_j) + f_e(\tilde{y}_j) \right) \\ + \Delta t_i \left(f_I(\hat{y}_{i+1}) + f_E(\hat{y}_i) + \varphi(\Delta t \Lambda(\hat{y}_i))\, f_e(\hat{y}_i) \right) \\ - \Delta t_i \left(f_I(\tilde{y}_{i+1}) + f_E(\tilde{y}_i) + \varphi(\Delta t \Lambda(\tilde{y}_i))\, f_e(\tilde{y}_i) \right) \end{aligned}$$

i.e. standard SI-SDC plus the exponential term.

[1]Dutt, A., Greengard, L., & Rokhlin, V. (2000). Spectral deferred correction methods for ordinary differential equations. BIT Numerical Mathematics, 40(2), 241–266.

Spectral Deferred Correction (SDC)

Università della Svizzera italiana

Consider equation: $y' = f(y) = f_I(y) + f_E(y) + f_e(y)$

Collocation methods: Start from the approximation

$$y(t_i) = y_0 + \int_0^{t_i} f(y(s))\mathrm{d}s \approx y_0 + \int_0^{t_i} \sum_{j=1}^{m} \ell_j(s) f(y(t_j))\mathrm{d}s$$

And solve the discrete system:

$$y_i = y_0 + \Delta t \sum_{j=1}^{m} a_{ij} f(y_j) \quad i = 1, \ldots, m$$
$$= y_0 + I(y)_i$$

With

$$a_{ij} = \int_0^{c_i} \ell_j(s)\mathrm{d}s, \qquad I(y)_i = \Delta t \sum_{j=1}^{m} a_{ij} f(y_j)\,.$$

SDC[1]: Solves the discrete system iteratively:

i) Initial guess $\tilde{y}_i$,

ii) Error: $\delta_i = y_i - \tilde{y}_i$

iii) Residual $\varepsilon_i = \tilde{y}_i - y_0 - I(\tilde{y})_i$

iv) Derive error equation $\delta_i = \varepsilon_i + I(\tilde{y} + \delta)_i - I(\tilde{y})_i$.

Solve error equation approximatively: with $\hat{y}_i = \tilde{y}_i + \delta_i$

$$\hat{y}_{i+1} = \hat{y}_i + \Delta t \sum_{j=1}^{m} s_{ij} \left(f_I(\tilde{y}_j) + f_E(\tilde{y}_j) + f_e(\tilde{y}_j) \right)$$
$$+ \Delta t_i \left(f_I(\hat{y}_{i+1}) + f_E(\hat{y}_i) + \varphi(\Delta t \Lambda(\hat{y}_i))\, f_e(\hat{y}_i) \right)$$
$$- \Delta t_i \left(f_I(\tilde{y}_{i+1}) + f_E(\tilde{y}_i) + \varphi(\Delta t \Lambda(\tilde{y}_i))\, f_e(\tilde{y}_i) \right)$$

i.e. standard SI-SDC plus the exponential term.

[1]Dutt, A., Greengard, L., & Rokhlin, V. (2000). Spectral deferred correction methods for ordinary differential equations. BIT Numerical Mathematics, 40(2), 241–266.

Stability of SDC coupled with IMEX-RL

For the time being, we only want to check whether the SDC+IMEX-RL sweeps converge or not.

Computational setup:

- Time discretization in pySDC[1],
- Radau IIA nodes, $m = 3$,
- Sweep till convergence.
- Consider three ionic models with increasing f_e stiffness:
  - ✤ Hodgkin-Huxley (HH), 4 vars.
  - ✤ Courtemanche-Ramirez-Nattel (CRN), 20 vars,
  - ✤ Ten Tusscher-Panfilov (TTP), 19 vars.
- Different step sizes.

	Stiffness	**SDC+IMEX-RL**		
	ρ_e	$\Delta t = 0.1$	$\Delta t = 0.05$	$\Delta t = 0.01$
HH	55	✓	✓	✓
CRN	130	✗	✗	✓
TTP	1000	✗	✗	✗

✓ = Converged

✗ = Unstable

Serial IMEX-RL is stable for all those combinations!

[1]Speck, R. (2019). Algorithm 997: PysDC—prototyping spectral deferred corrections. *ACM Transactions on Mathematical Software*, *45*(3).

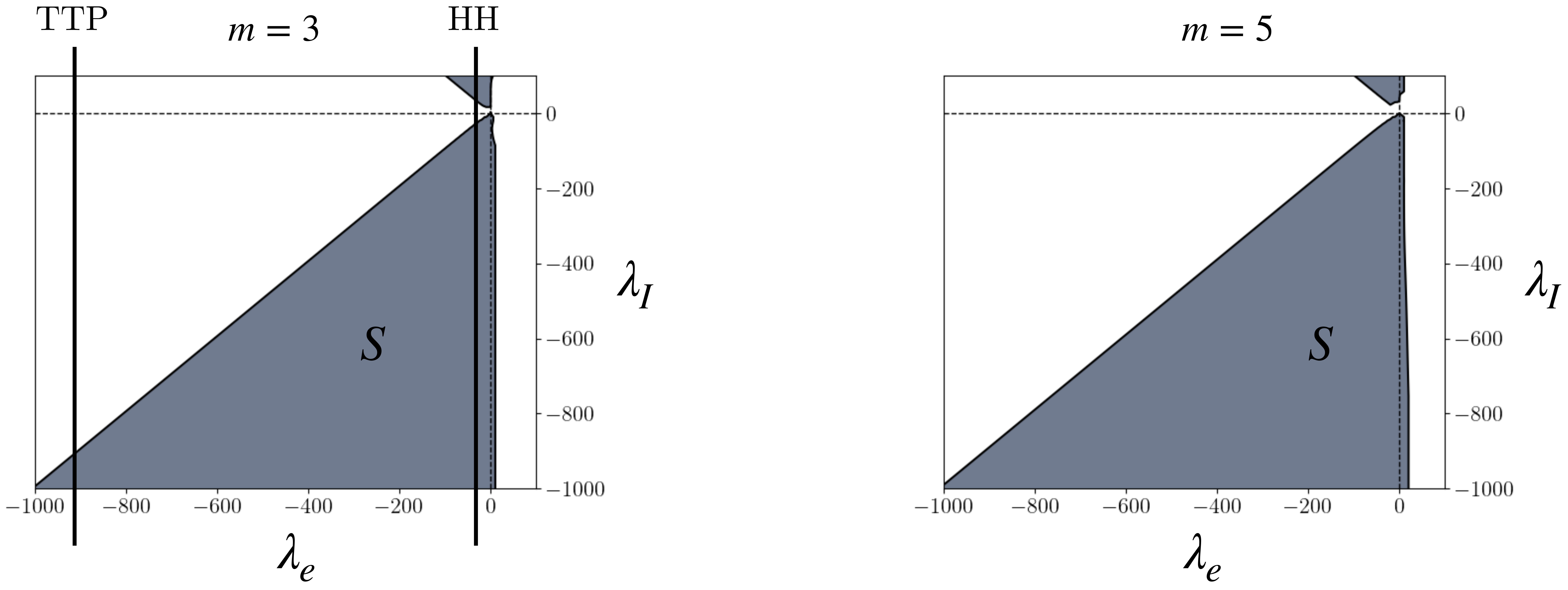
Stability analysis

- We solve the test equation $y' = \lambda_I y + \lambda_E y + \lambda_e y$
- With $\Delta t = 1$
- And display the stability domain for fixed $\lambda_E = -1$:

  $S = \{(\lambda_I, \lambda_e) \in \mathbb{R}^2 : |y(1)| \leq 1\}$
- Considering different combinations of
  - ✦ Number of collocation nodes,
  - ✦ Number of levels in MLSDC,
  - ✦ Number of parallel time steps in PFASST.
- We set the maximal number of sweeps/iterations to 5.

Stability analysis - SDC

Fix $\lambda_E = -1$ and let vary λ_I, λ_e in $[-1000{,}100]$. Let m be the number of collocation nodes.



$\Rightarrow$ SDC+IMEX-RL is stable only if the Laplacian is stiffer than the ionic model.

Stability analysis - SDC

A rough explanation:

$$\hat{y}_{i+1} = \hat{y}_i + \Delta t \sum_{j=1}^{m} s_{ij} \left(\lambda_I \tilde{y}_j + \lambda_E \tilde{y}_j + \lambda_e \tilde{y}_j \right) + \Delta t_i \left(\lambda_I \hat{y}_{i+1} + \lambda_E \hat{y}_i + \varphi(\Delta t \lambda_e) \lambda_e \hat{y}_i \right) - \Delta t_i \left(\lambda_I \tilde{y}_{i+1} + \lambda_E \tilde{y}_i + \varphi(\Delta t \lambda \tilde{y}_i) \lambda_e \tilde{y}_i \right)$$

$$\hat{y}_{i+1} = (1 - \Delta t \lambda_I)^{-1} \left(\hat{y}_i + \Delta t \sum_{j=1}^{m} s_{ij} \left(\lambda_I \tilde{y}_j + \lambda_E \tilde{y}_j + \lambda_e \tilde{y}_j \right) + \Delta t_i \left(\lambda_E \hat{y}_i + \varphi(\Delta t \lambda_e) \lambda_e \hat{y}_i \right) - \Delta t_i \left(\lambda_I \tilde{y}_{i+1} + \lambda_E \tilde{y}_i + \varphi(\Delta t \lambda \tilde{y}_i) \lambda_e \tilde{y}_i \right) \right)$$

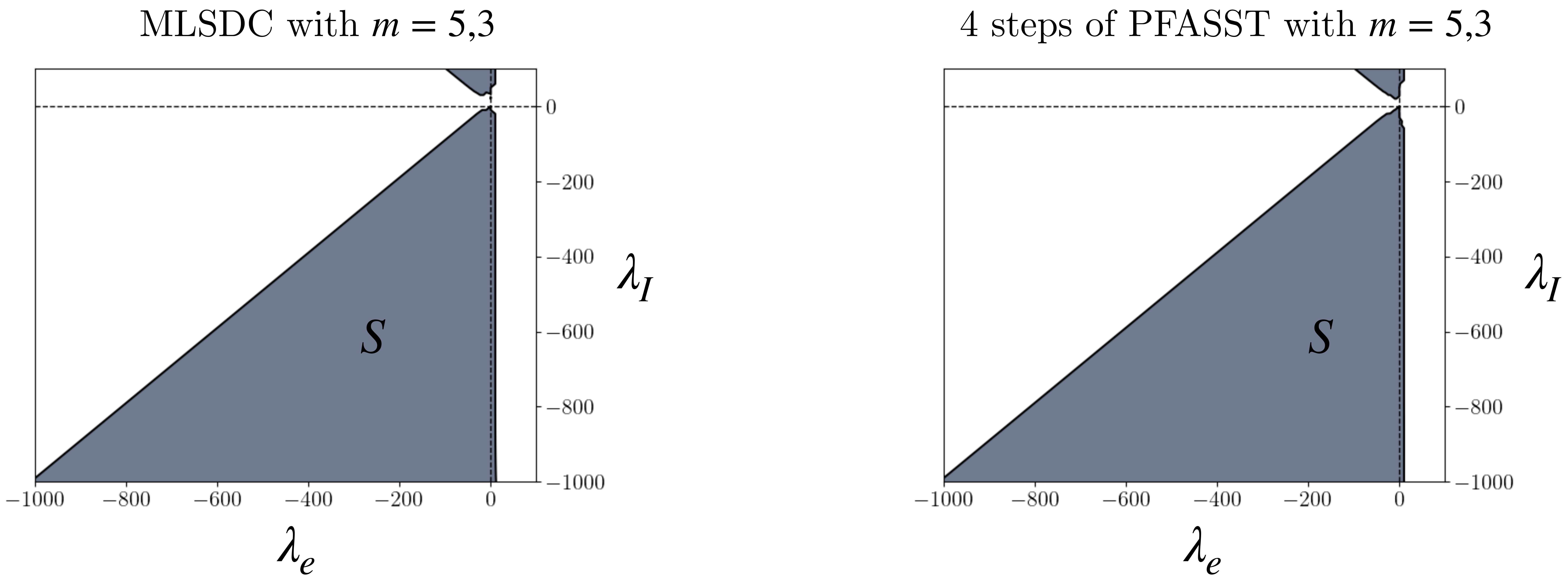
Therefore we have terms

$$(1 - \Delta t \lambda_I)^{-1} \Delta t \lambda_I \qquad \text{and} \qquad (1 - \Delta t \lambda_I)^{-1} \Delta t \lambda_e$$

$$\text{Stable for any } \lambda_I \qquad\qquad \text{Stable if } |\lambda_e| < |\lambda_I|$$

Stability analysis - MLSDC and PFASST

Fix $\lambda_E = -1$ and let vary λ_I, λ_e in $[-1000, 100]$. Let m be the number of collocation nodes.



Seems that stability is independent of number of levels and parallel steps.

Exponential Spectral Deferred Correction (ESDC)

Consider equation:

$$y' = f(y) = \Lambda y + N(y)\,.$$

SDC is based on collocation methods. Hence, on Picard formula:

$$y(t) = y_0 + \int_0^t f(y(s))\mathrm{d}s$$

and its discretization

$$y_i = y_0 + \Delta t \sum_{j=1}^{m} a_{ij} f(y_j) \quad i = 1,\dots,m,$$

with

$$a_{ij} = \int_0^{c_i} \ell_j(s)\mathrm{d}s \;\in\; \mathbb{R}$$

ESDC[1] is based on exponential Runge-Kutta methods. Hence, on variation-of-constants formula:

$$y(t) = y_0 + \int_0^t e^{(t-s)\Lambda}(\Lambda y_0 + N(y(s)))\mathrm{d}s$$

and its discretization

$$y_i = y_0 + \Delta t \sum_{j=1}^{m} a_{ij}(\Delta t\Lambda)(\Lambda y_0 + N(y_j)) \quad i = 1,\dots,m,$$

with

$$a_{ij}(\Delta t\Lambda) = \int_0^{c_i} e^{(c_i-s)\Delta t\Lambda}\ell_j(s)\mathrm{d}s \;\in\; \mathbb{R}^{d\times d}$$

[1]Buvoli, T. (2020). A class of exponential integrators based on spectral deferred correction. *SIAM Journal on Scientific Computing*, *42*(1), A1–A27.

ESDC for monodomain equation

- Write

$$\begin{aligned} y' &= f_I(y) + f_E(y) + f_e(y) \\ &= f_I(y) + f_E(y) + \Lambda(y)(y - y_\infty(y)) \\ &= \underbrace{\Lambda(y_n)y}_{\Lambda y} + \underbrace{f_I(y) + f_E(y) + \Lambda(y)(y - y_\infty(y)) - \Lambda(y_n)y}_{N(y)} \end{aligned}$$

- Since

$$\Lambda = \Lambda(y_n) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Lambda_e(V_n) \end{pmatrix}, \quad f_I(y) = \begin{pmatrix} * \\ 0 \\ 0 \end{pmatrix}, \quad f_E(y) = \begin{pmatrix} * \\ * \\ 0 \end{pmatrix}$$

Then

$$a_{ij}(\Delta t \Lambda) = \begin{pmatrix} a_{ij} & 0 & 0 \\ 0 & a_{ij} & 0 \\ 0 & 0 & a_{ij}(\Delta t \Lambda_e(V_n)) \end{pmatrix}$$

And

$$a_{ij}(\Delta t \Lambda)\, f_I(y_j) = a_{ij}\, f_I(y_j), \qquad a_{ij}(\Delta t \Lambda)\, f_E(y_j) = a_{ij}\, f_E(y_j),$$

hence f_I, f_E terms are integrated with the standard SDC method, while f_e with ESDC.

Checking for convergence in ESDC iterations

	Stiffness	IMEX-RL + ESDC		
	ρ	$\Delta t = 0.1$	$\Delta t = 0.05$	$\Delta t = 0.01$
HH	55	✓	✓	✓
CRN	130	✓	✓	✓
TTP	1000	✓	✓	✓

✓ = Converged

When computing residuals, ESDC treats the ionic model f_e variables exponentially. This is the main difference with respect to SDC.

SDC residual given by

$$y(t) = y_0 + \int_0^t f(y(s))\mathrm{d}s$$

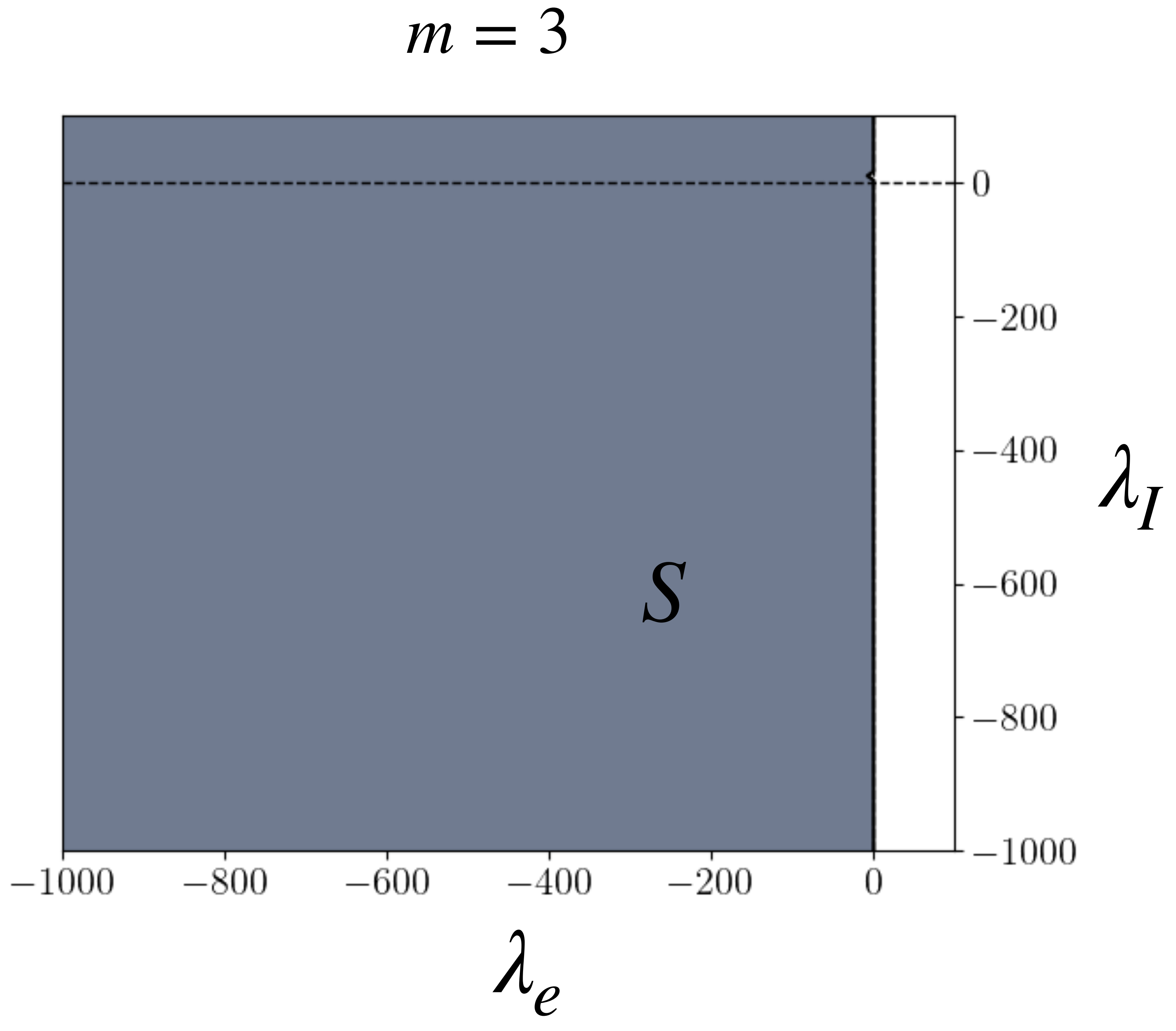
ESDC residual given by

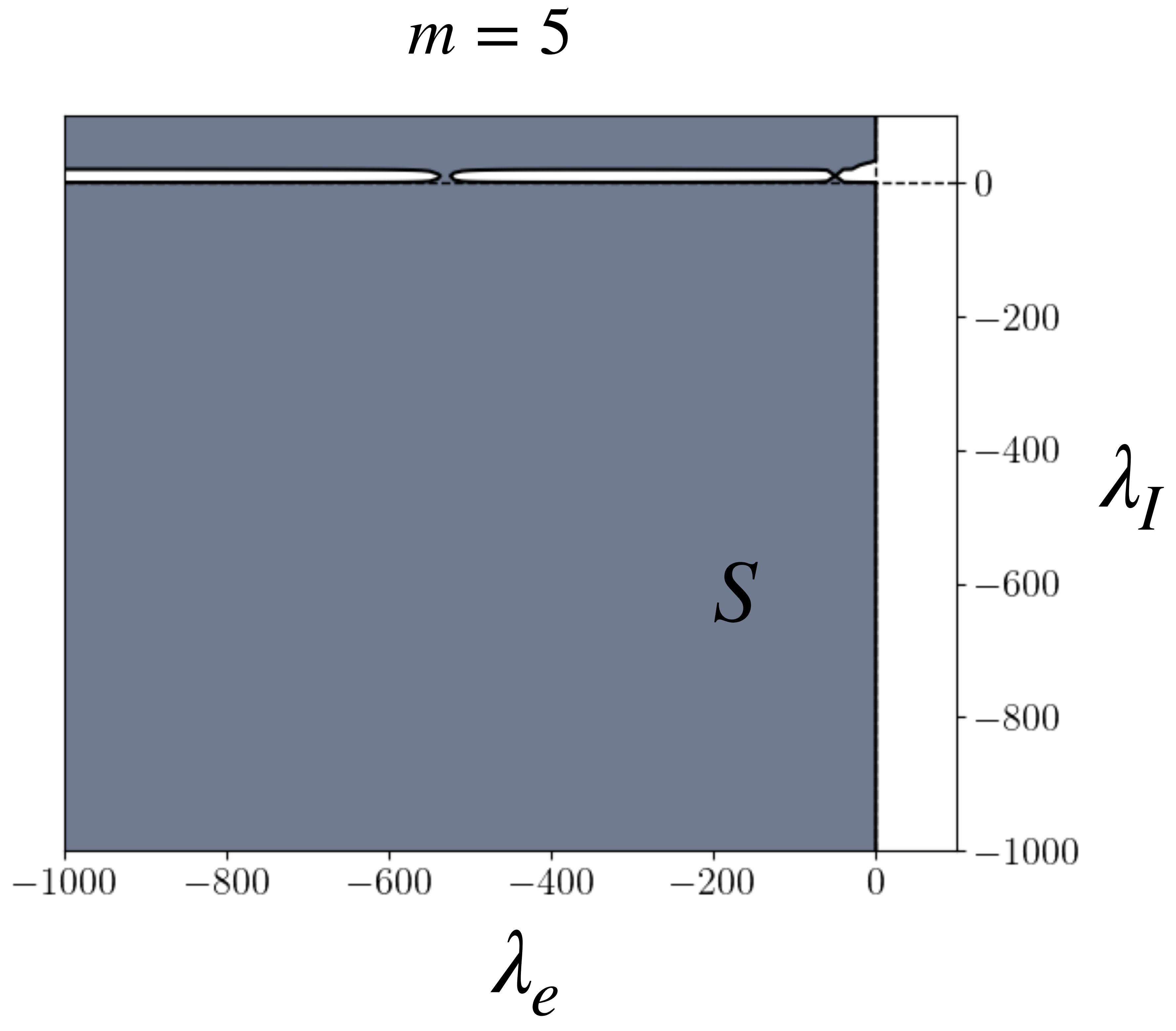
$$y(t) = y_0 + \int_0^t \boxed{e^{(t-s)\Lambda}}(\Lambda y_0 + N(y(s)))\mathrm{d}s$$

Stability analysis - ESDC

Università della Svizzera italiana

Fix $\lambda_E = -1$ and let vary λ_I, λ_e in $[-1000{,}100]$. Let m be the number of collocation nodes.



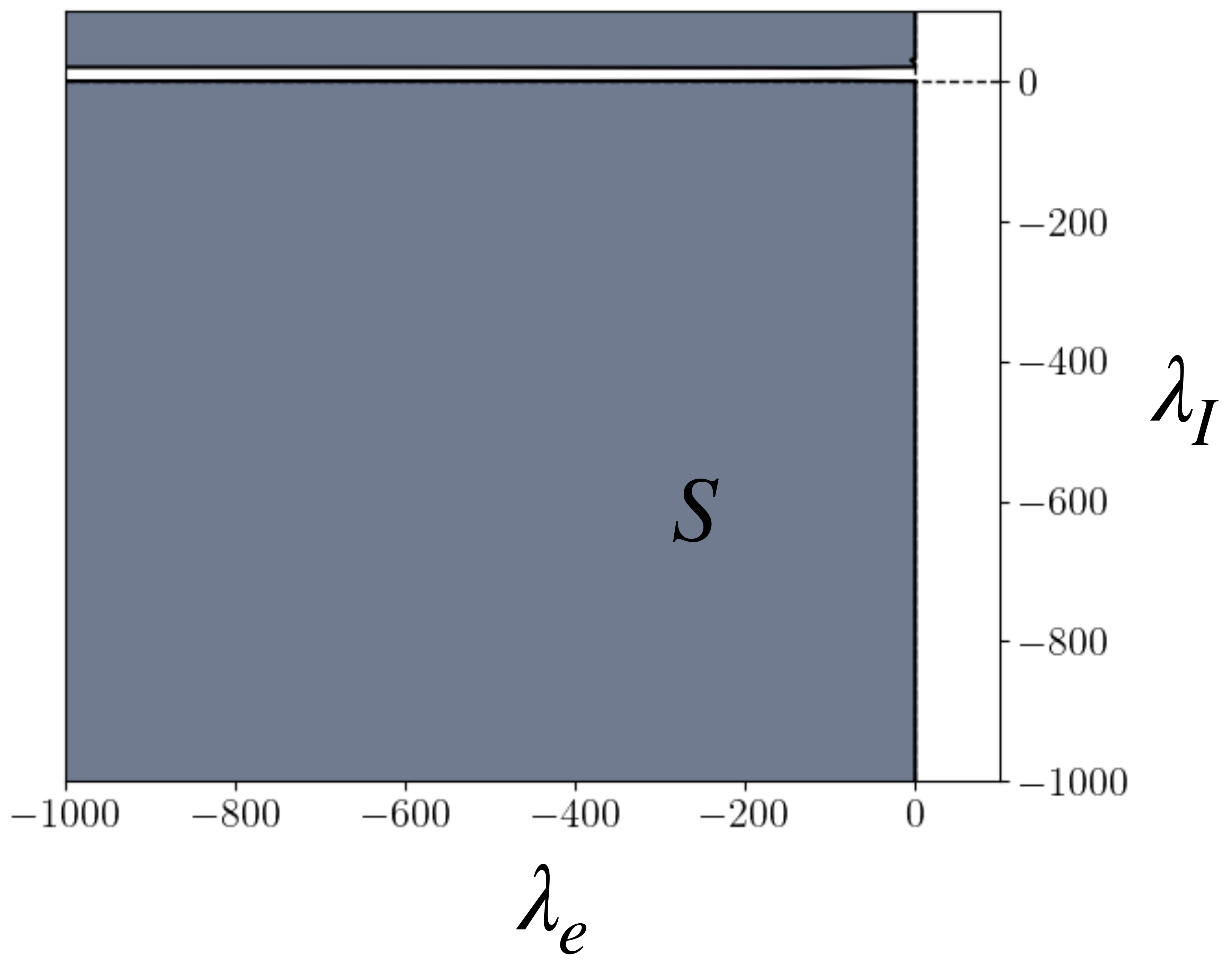


Exponential terms are stable independently of the Laplacian.

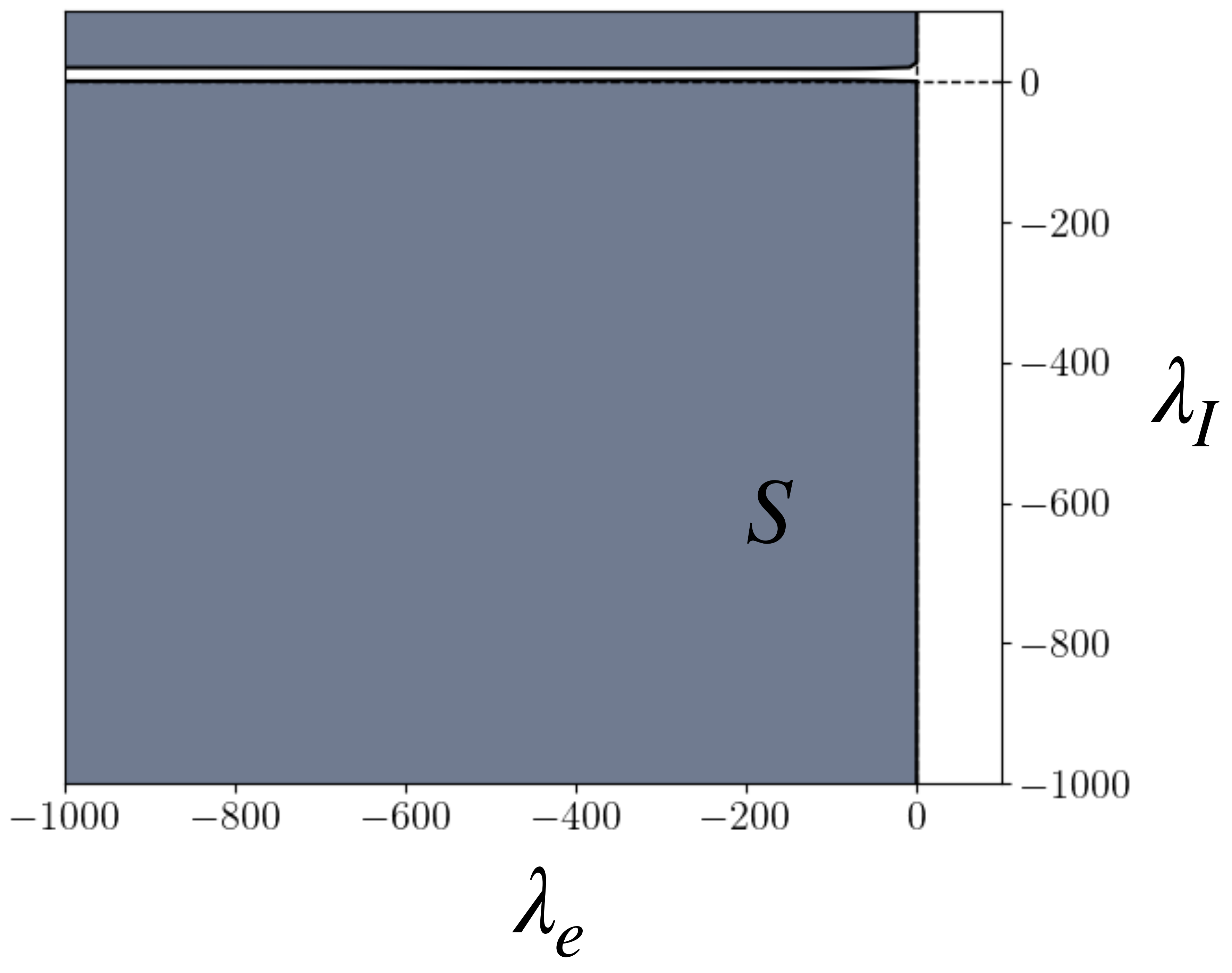
Stability analysis - MLESDC and PFASST

Fix $\lambda_E = -1$ and let vary λ_I, λ_e in $[-1000, 100]$. Let m be the number of collocation nodes.

MLESDC with $m = 5,3$



4 steps of PFASST with $m = 5,3$



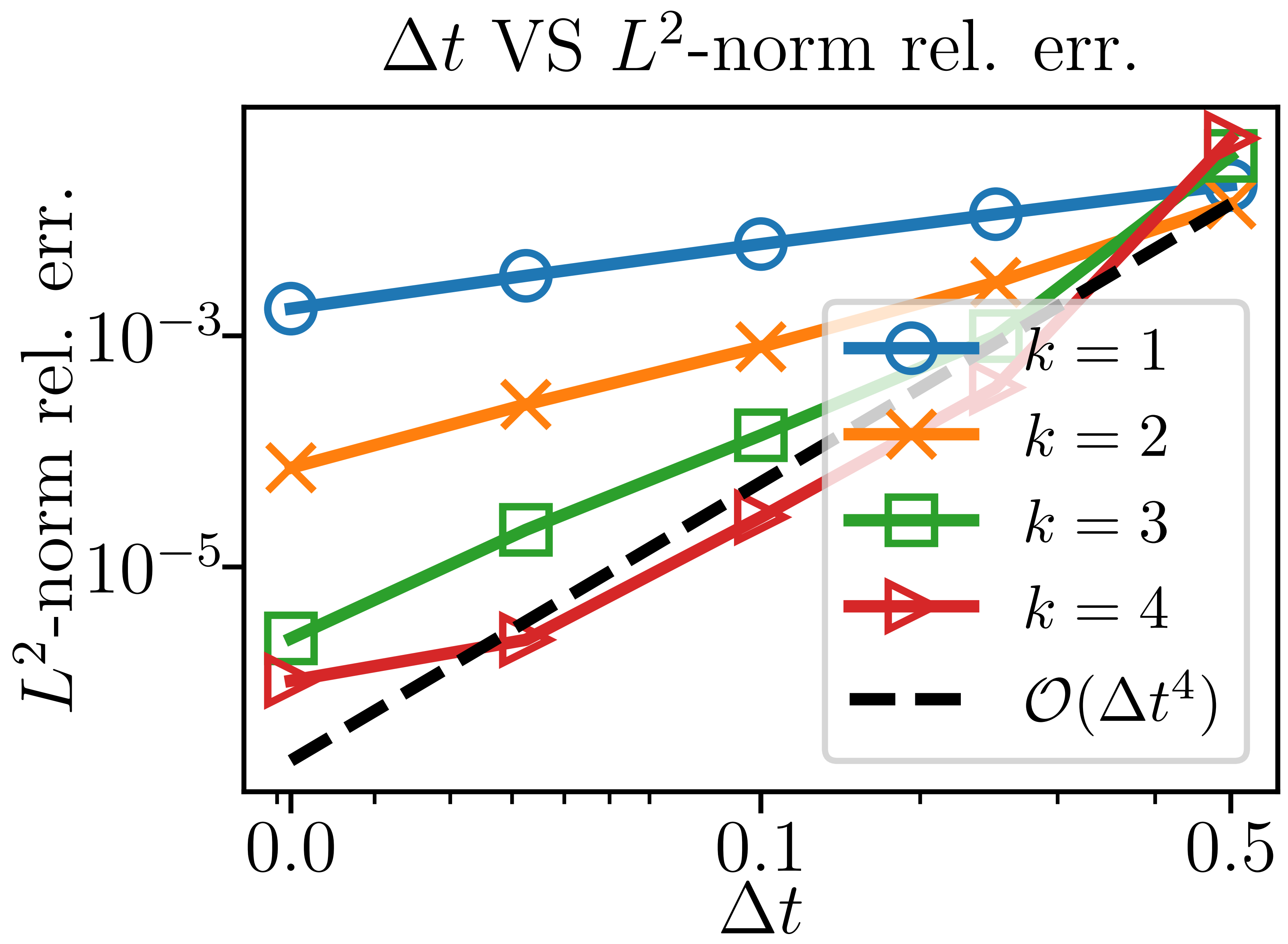
Again stability seems to not depend on number of levels or parallel steps.

Convergence experiments

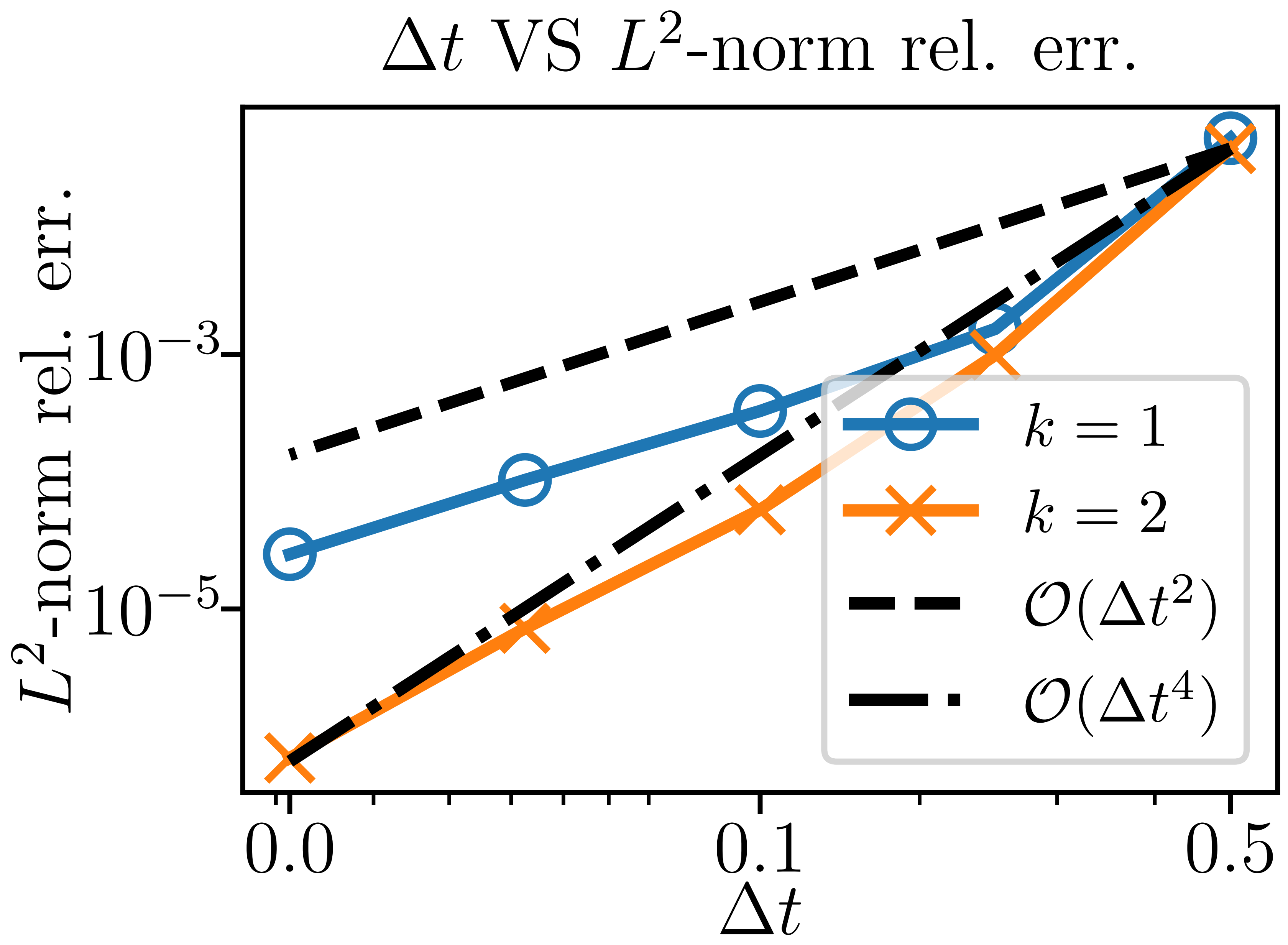
Solve Monodomain equation with

- TTP ionic model (stiffest one),
- ESDC with one or two levels,
- Check expected order of convergence, as function of iteration number k.

One level, $m = 4$.



Two levels, with space-time coarsening, $m = 4{,}2$.

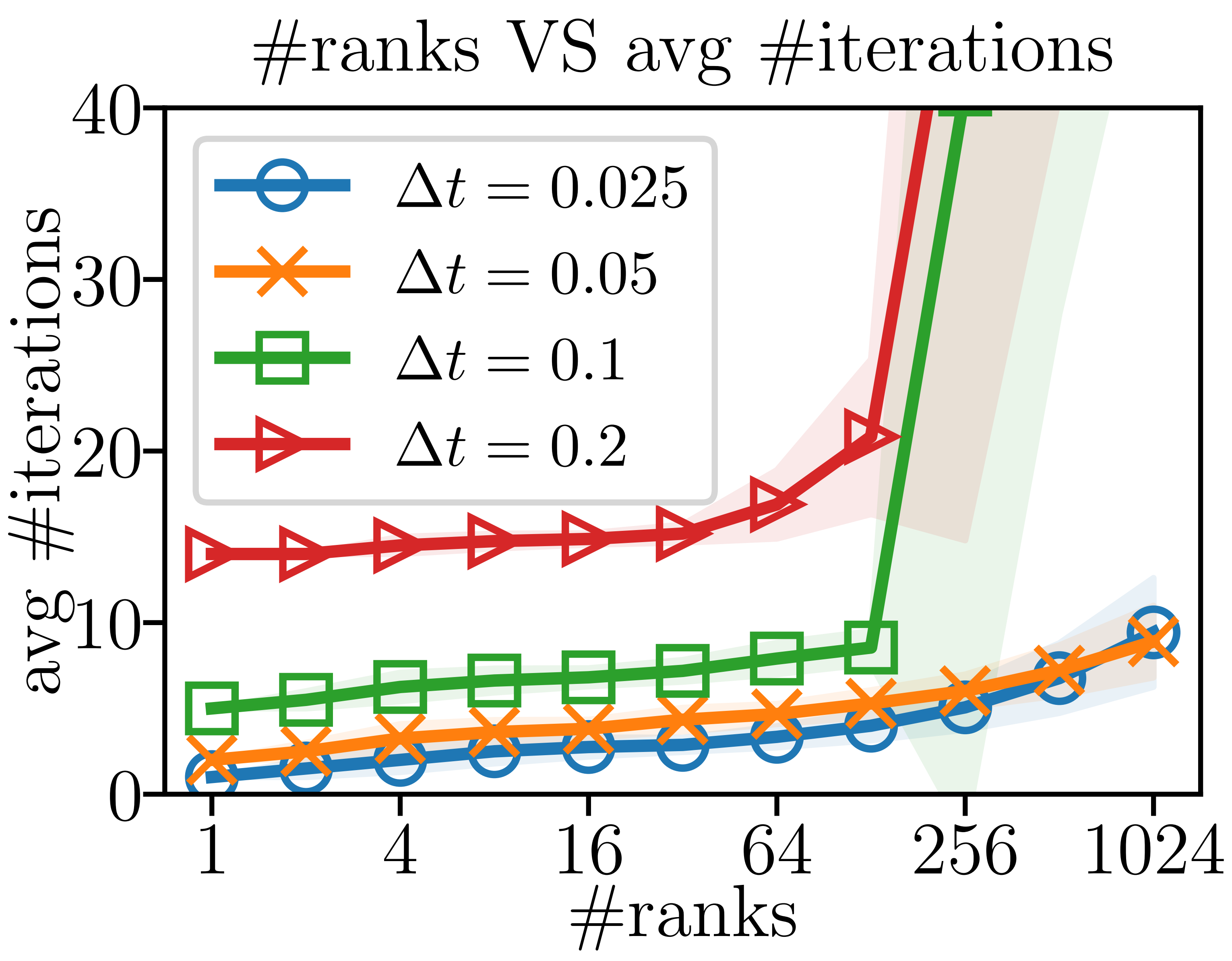


PFASST: iterations to convergence

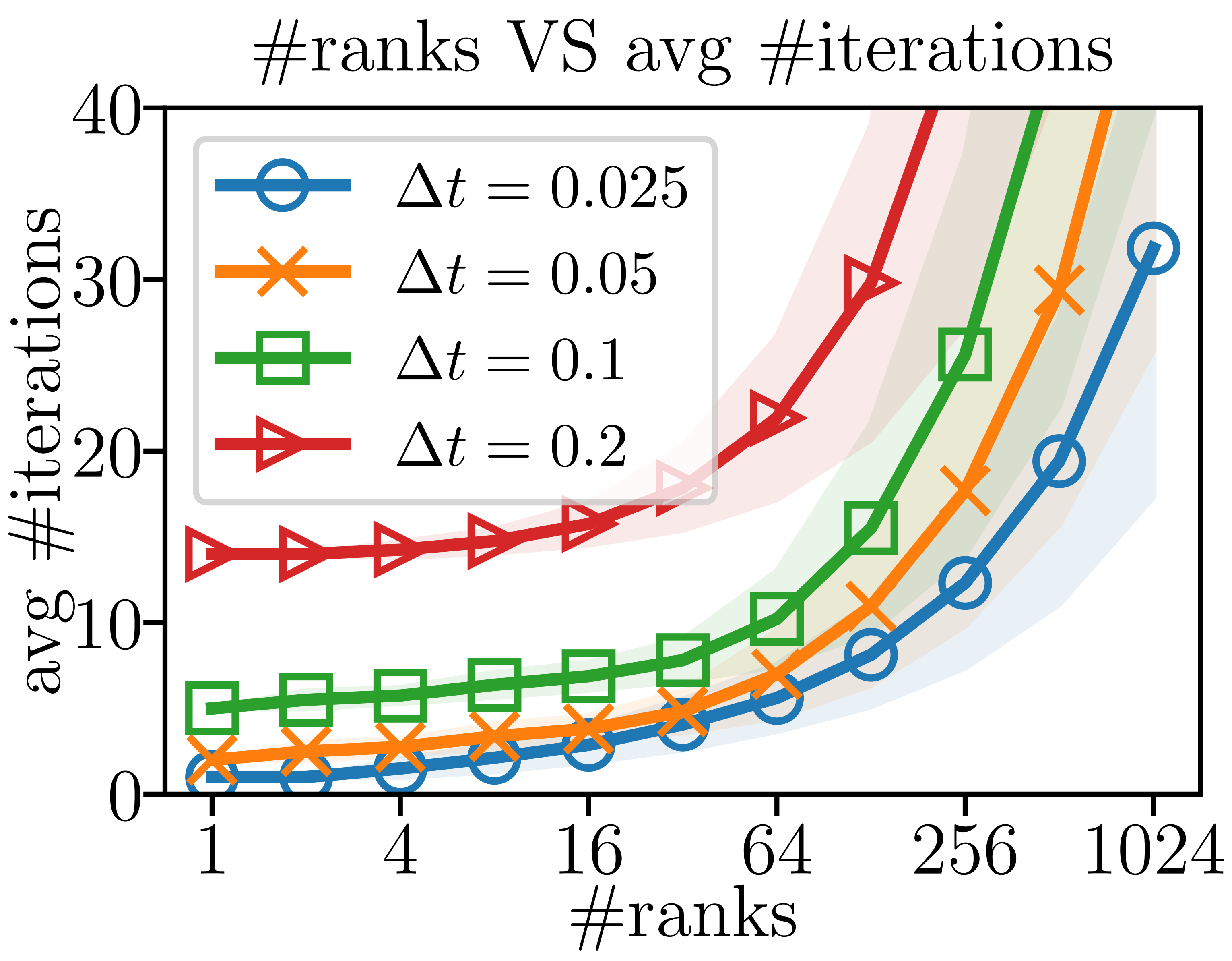
Number of iterations VS Number of processors

- TTP ionic model,
- 1D problem with fine spatial mesh: $\Delta x = 0.05$ mm.
- **Time coarsening only.**

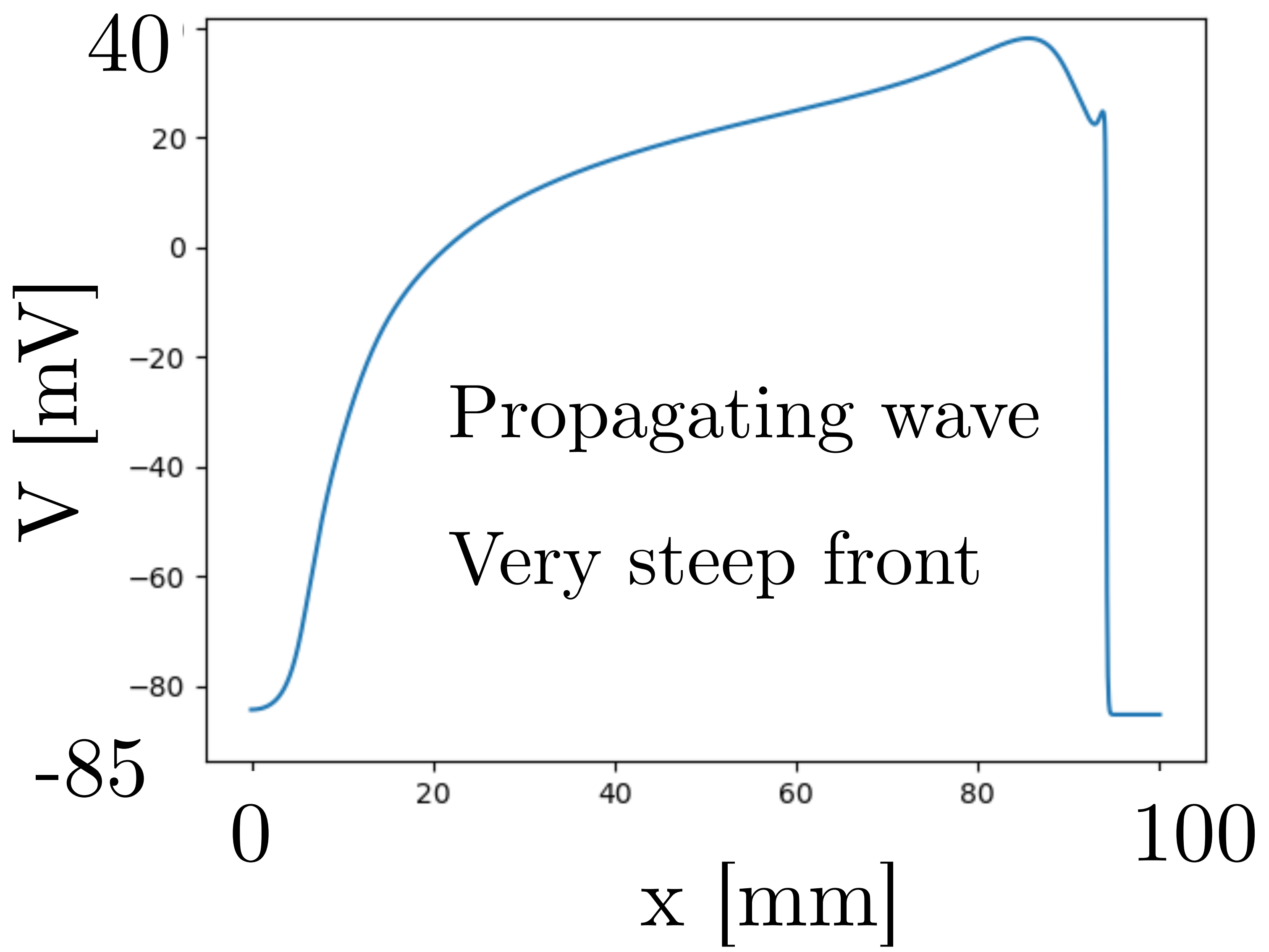
Two levels: $m = 6{,}4$



Three levels: $m = 6{,}4{,}2$



Typical solution

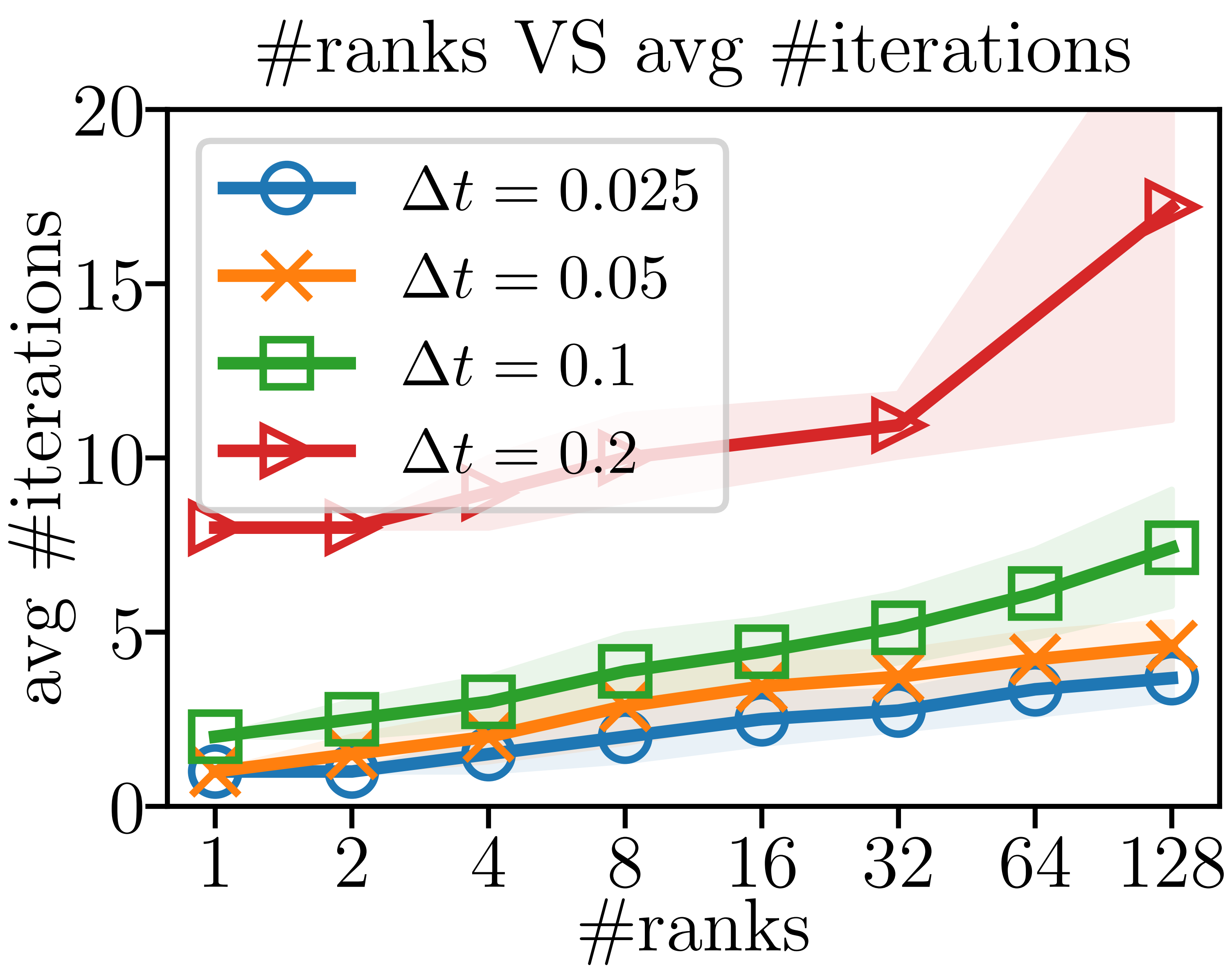


PFASST: iterations to convergence

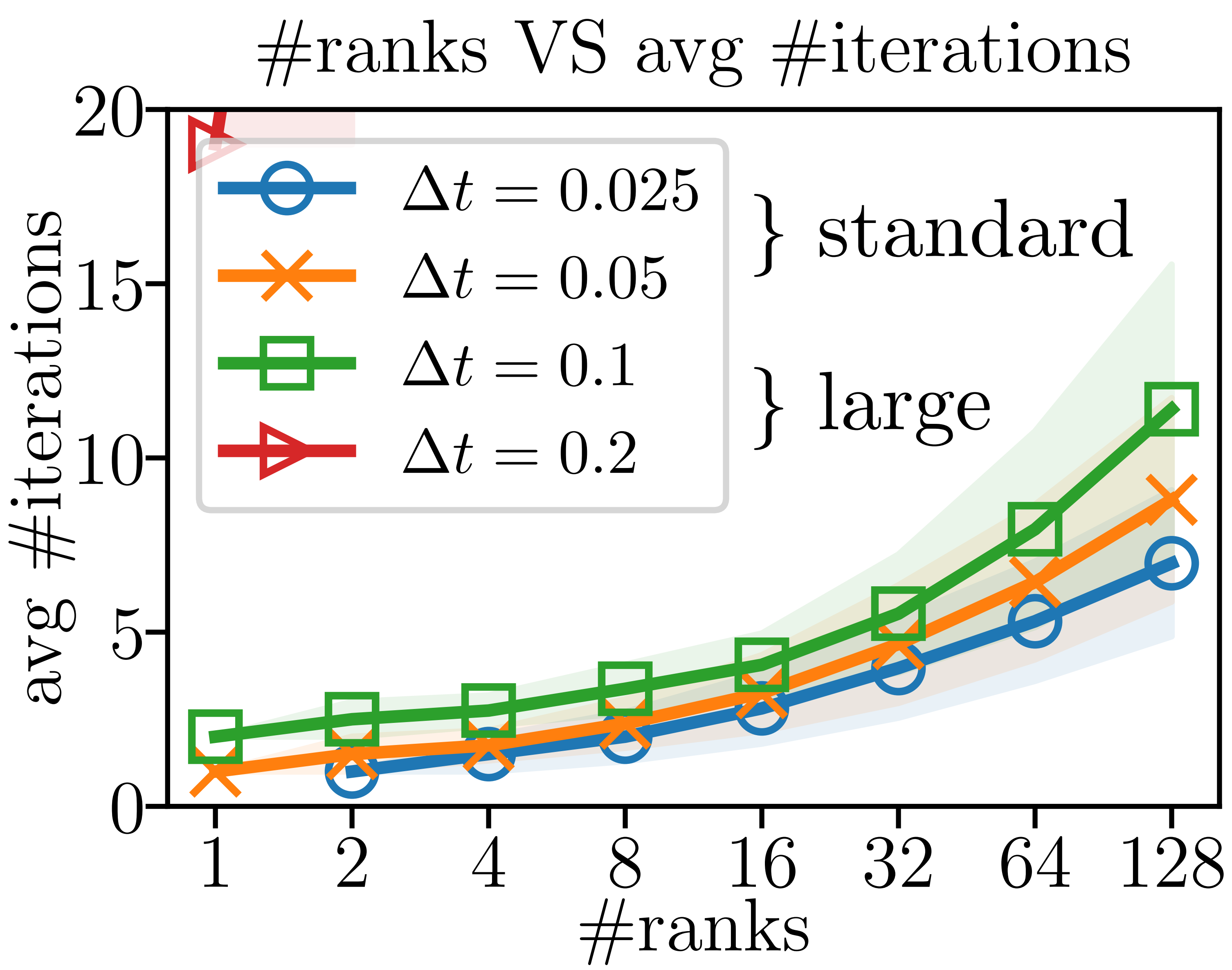
Number of iterations VS Number of processors

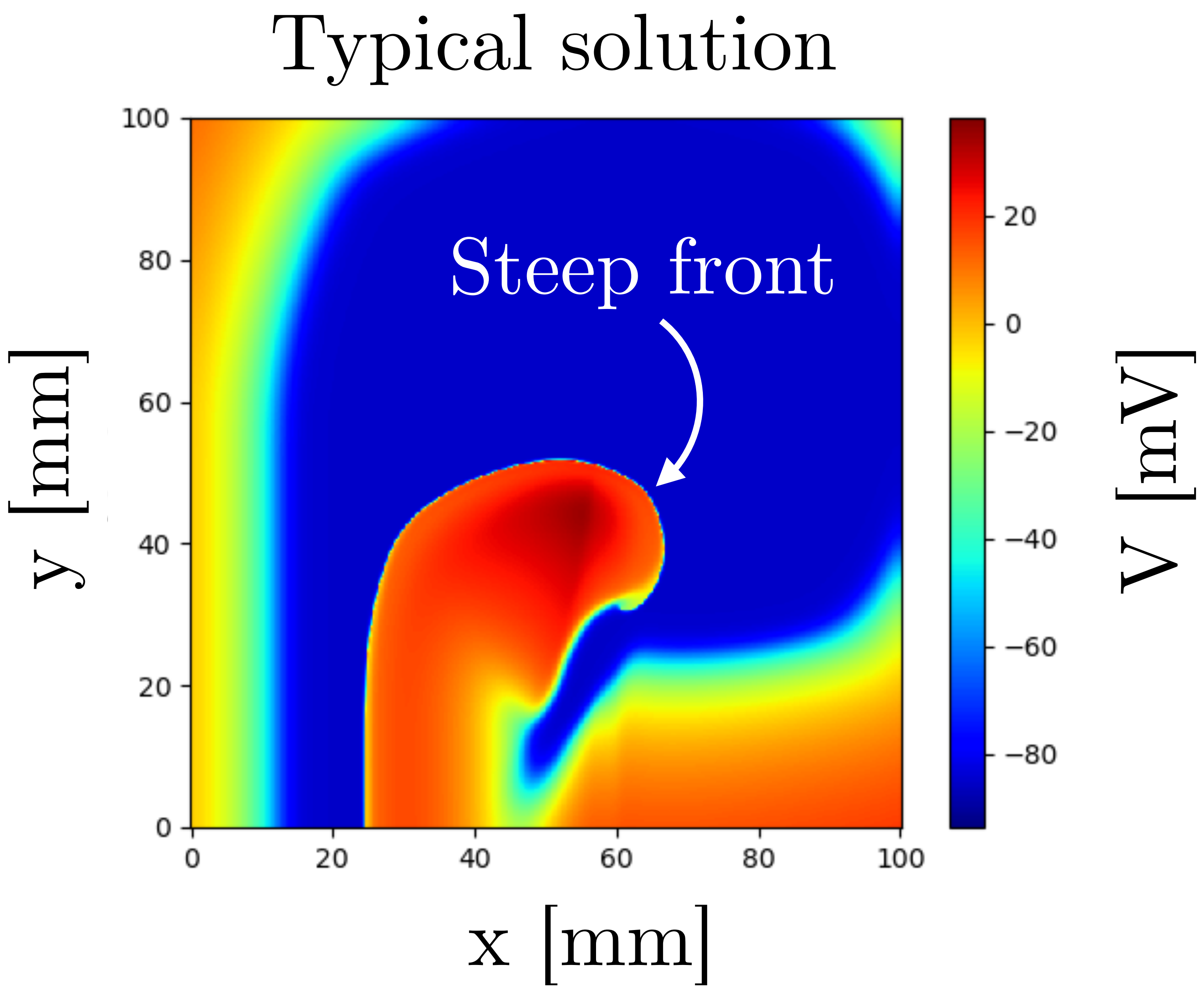
- TTP ionic model,
- 2D problem with standard mesh size: $\Delta x = 0.2$ mm.
- **Time coarsening only.**

Two levels: $m = 6{,}4$



Three levels: $m = 6{,}4{,}2$





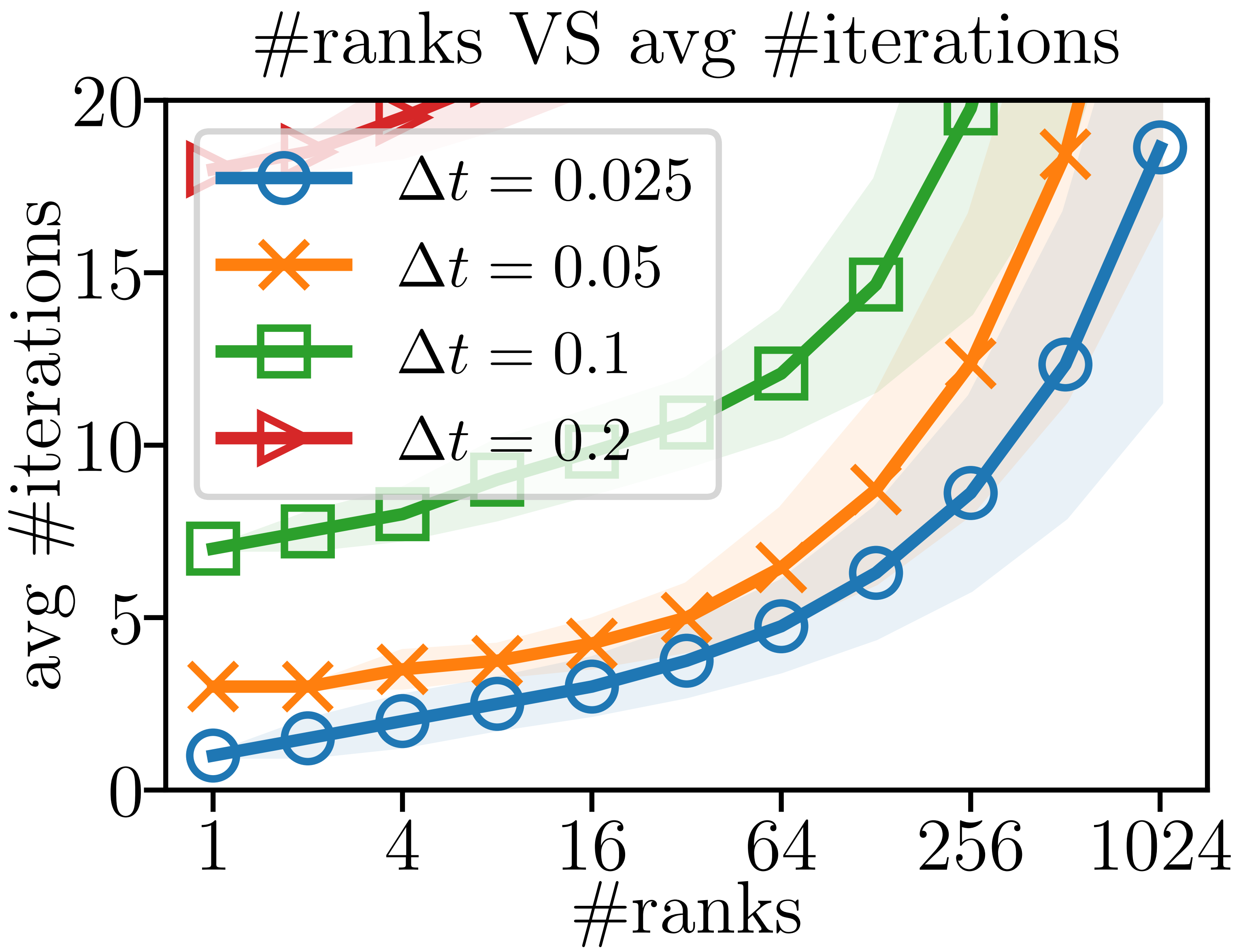
PFASST: iterations to convergence

Number of iterations VS Number of processors

- TTP ionic model,
- 1D problem with fine spatial mesh: $\Delta x = 0.05$ mm.
- Transfer operator: via FFT.

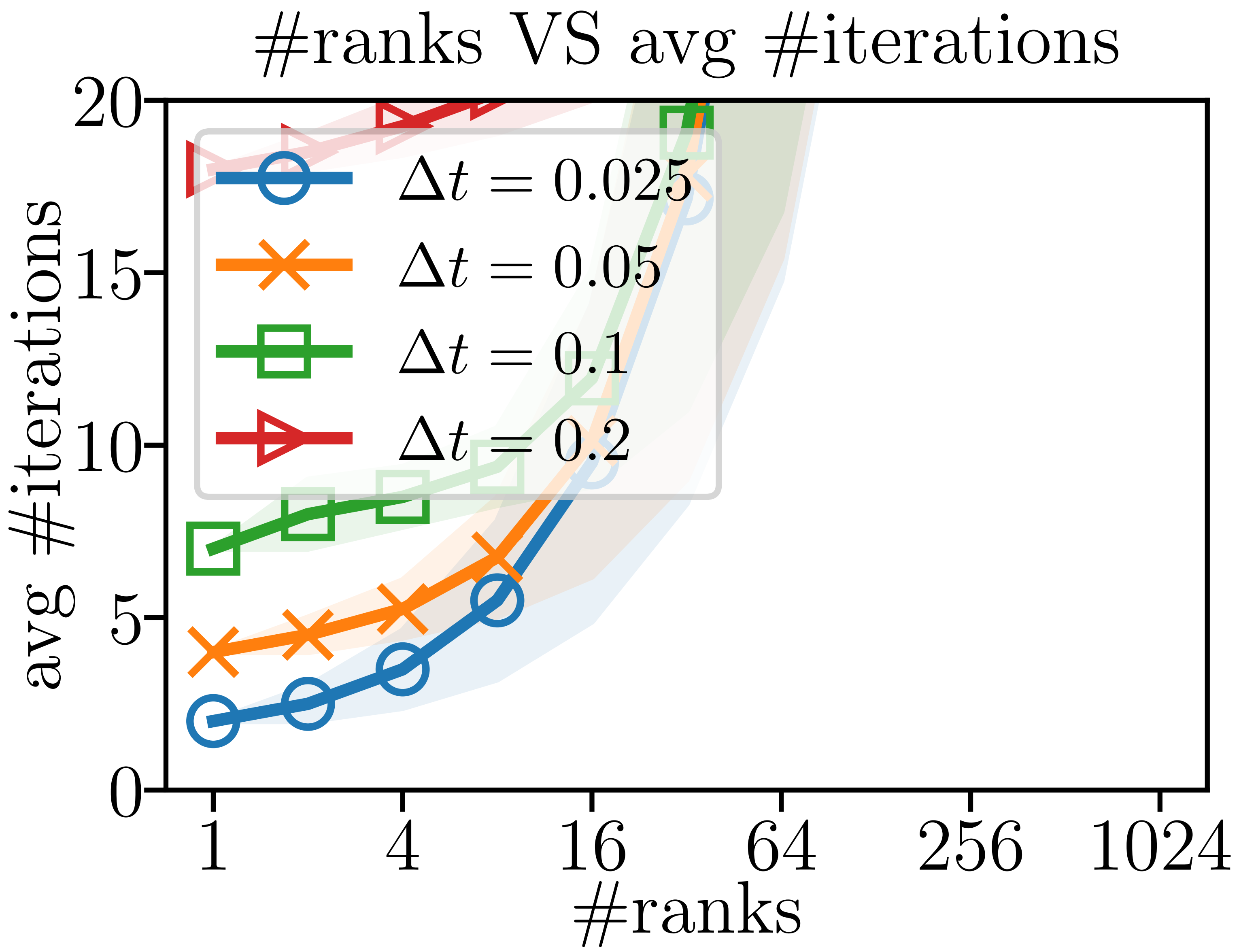
Two levels: $m = 6{,}3$,

Time coarsening

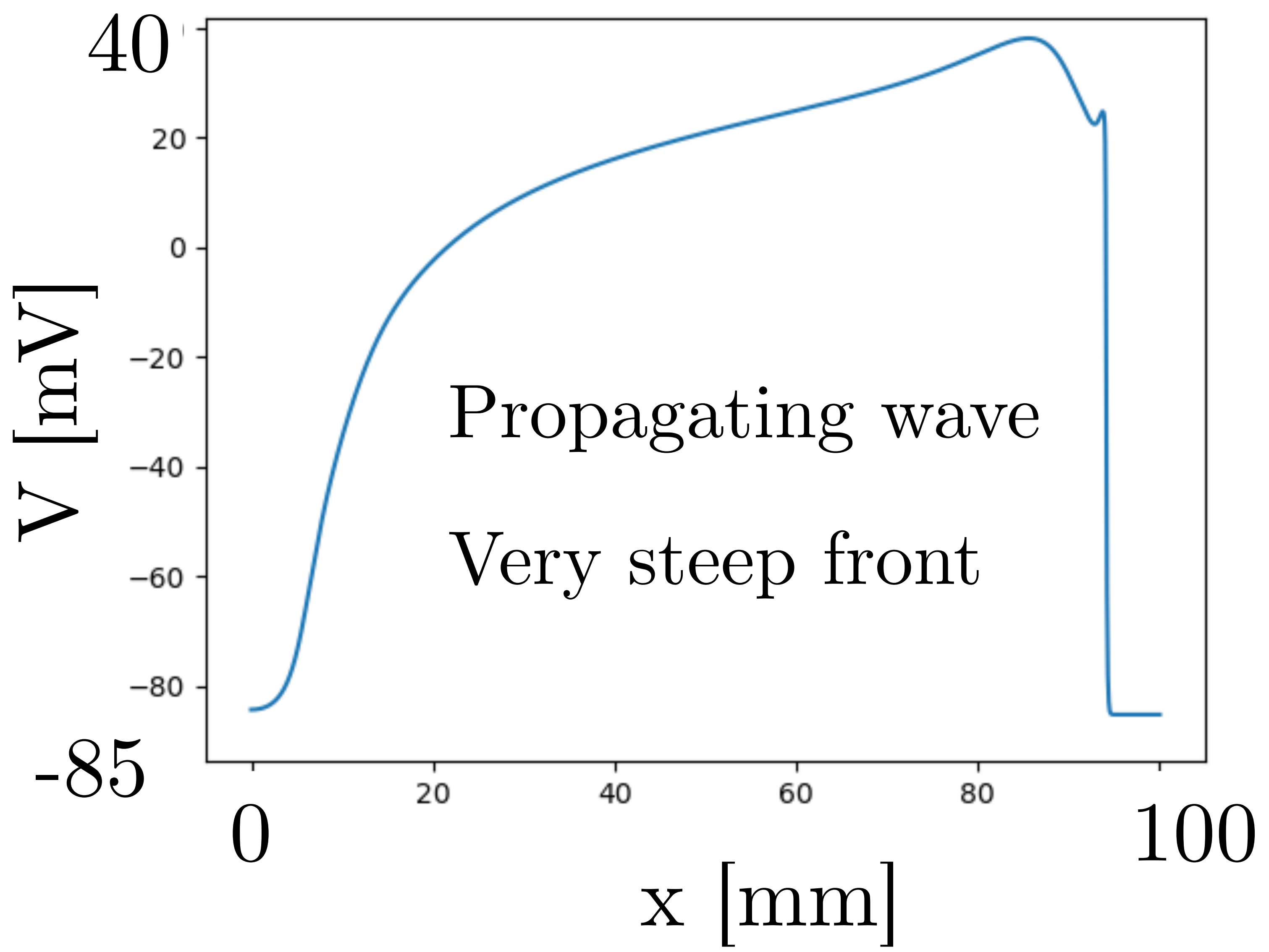


Two levels: $m = 6{,}3$,

Space-Time coarsening



Typical solution



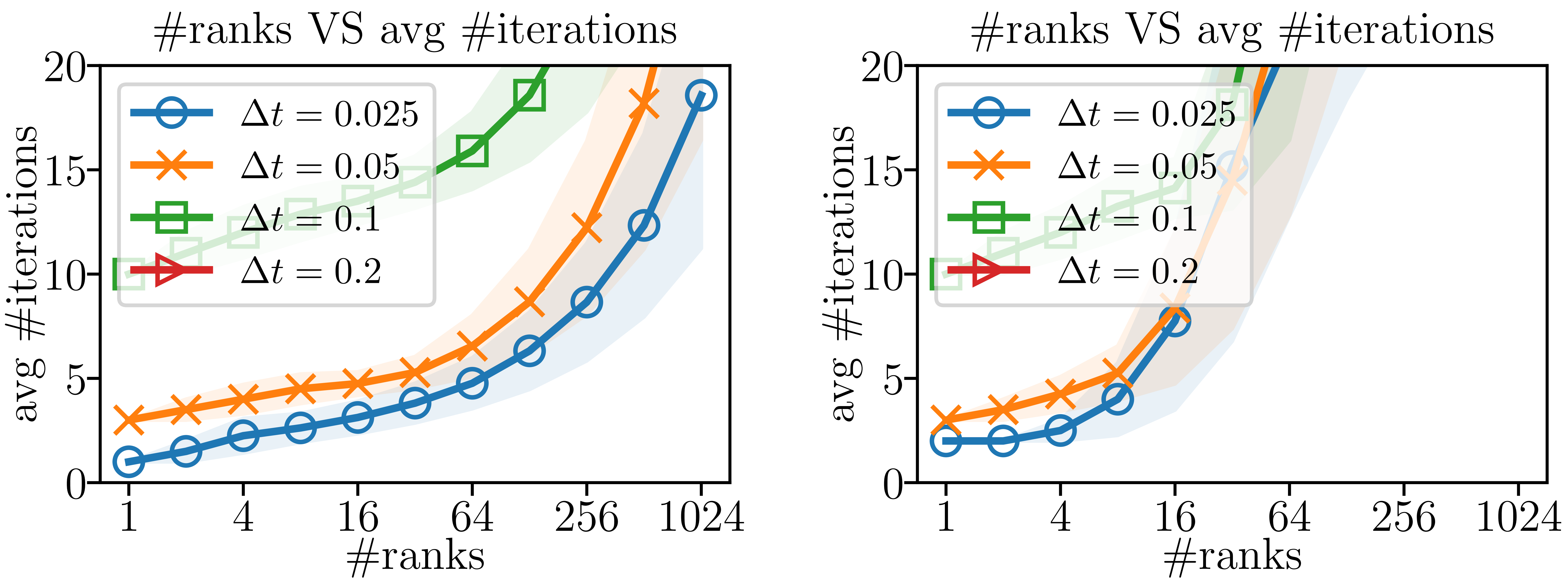
PFASST: iterations to convergence

Number of iterations VS Number of processors

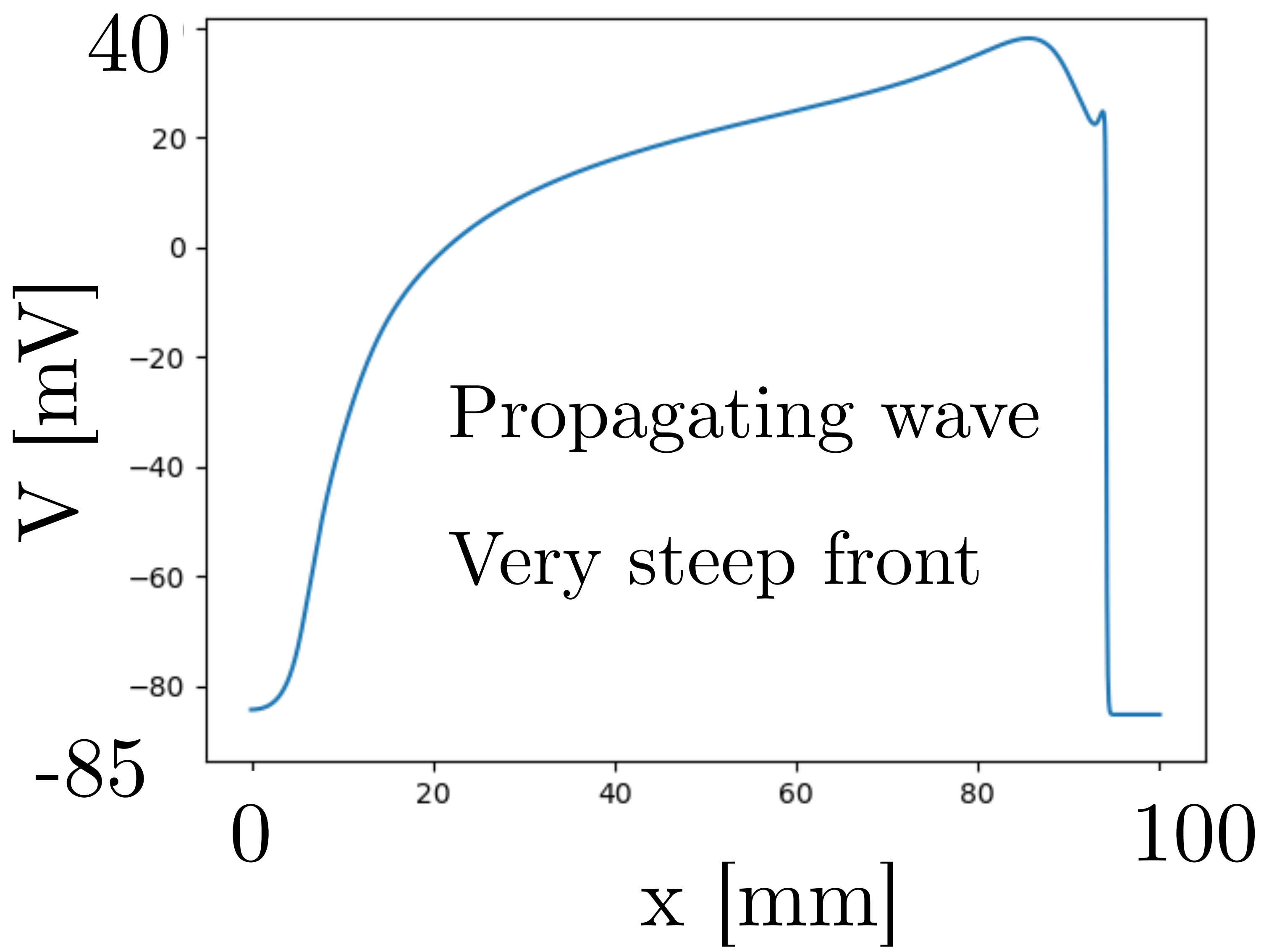
- TTP ionic model,
- 1D problem with very fine spatial mesh: $\Delta x = 0.0125$ mm.
- Transfer operator: via FFT.

Two levels: $m = 6,3$, **Time** coarsening

Two levels: $m = 6,3$, **Space-Time** coarsening



Typical solution



Thank you for your attention 😁

Funding: This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreements No 955495 (MICROCARD) and No 955701 (TIME-X). The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, Switzerland.