Parallel-in-time multirate explicit stabilized method for the monodomain model in cardiac electrophysiology

Giacomo Rosilho de Souza, Simone Pezzuto, Rolf Krause
Università della Svizzera Italiana

CMBE 2022 - Milano

Contents

- Hybrid Parareal Spectral Deferred Correction,
- Explicit stabilized methods,
- Application to the monodomain model.

Spectral Deferred Correction method[1]

Consider

$$y' = f(y), \qquad y(0) = y_0$$

and an approximation $\tilde{y}(t)$ to the solution $y(t)$. Let

$$\delta(t) = y(t) - \tilde{y}(t)$$

be the error and

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

the residual. Then

$$\delta(t_2) = \delta(t_1) + \int_{t_1}^{t_2} f(\tilde{y}(s) + \delta(s)) - f(\tilde{y}(s))\mathrm{d}s$$
$$+\varepsilon(t_2) - \varepsilon(t_1)\,.$$

Spectral Deferred Correction (SDC) method:

- Fix collocation points $c_1, \ldots, c_s$ in $[t_n, t_n + \Delta t]$,
- Compute approximations $\tilde{y}_i$ at c_i,

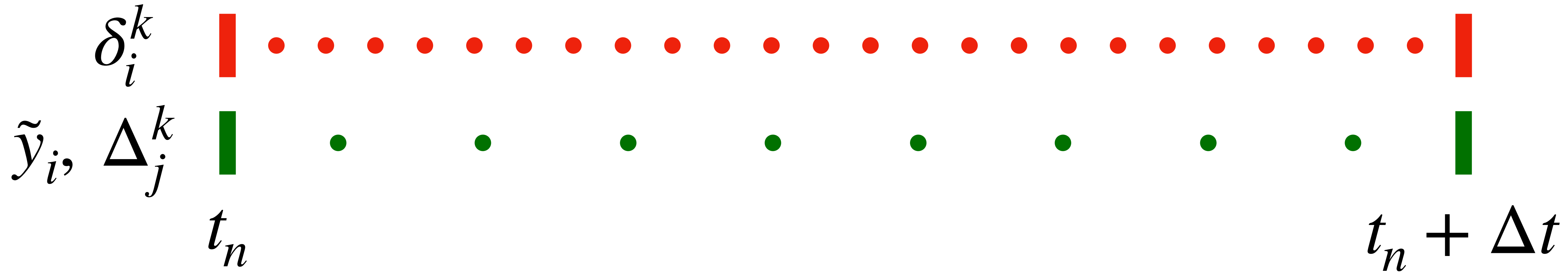
Then iterate on:

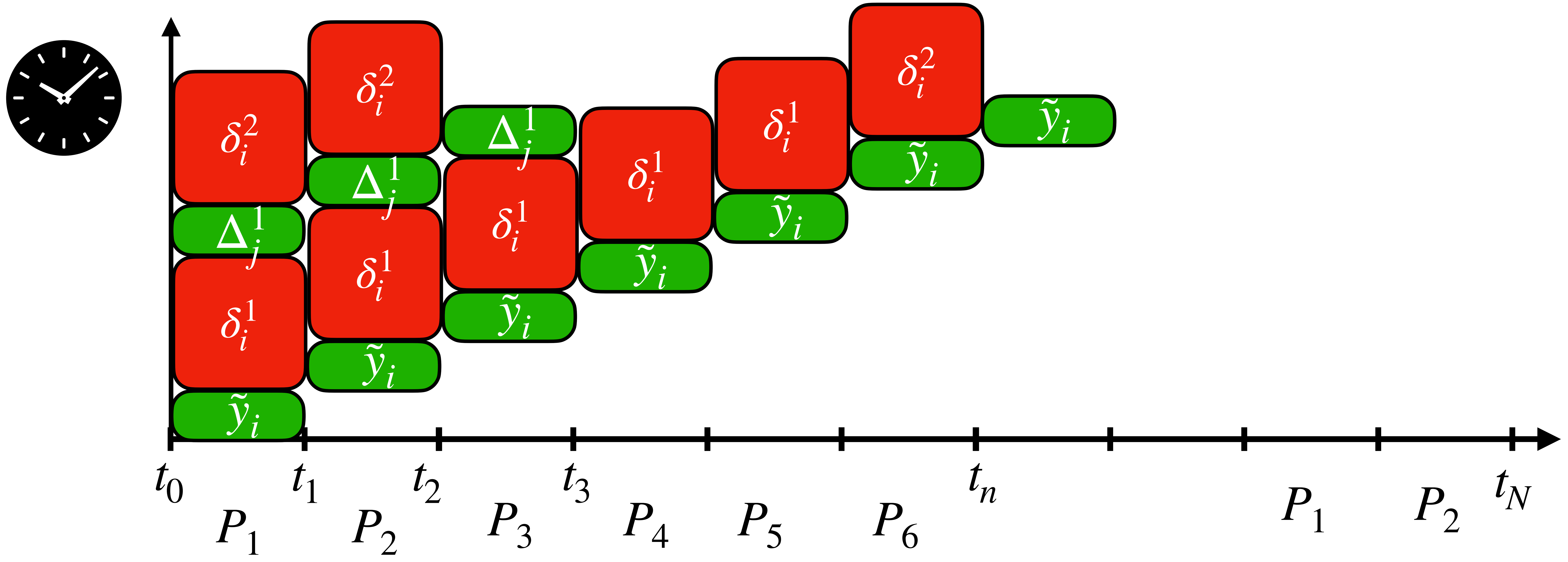
- Interpolate and form $\tilde{y}(t) = \sum L_i(t)\tilde{y}_i$,
- Approximate $\varepsilon(t)$ with care,
- Compute δ_i and correct $\tilde{y}_i + \delta_i \to \tilde{y}_i$.

[1]Dutt, A., Greengard, L., Rokhlin, V. (2000). BIT Numerical Mathematics, 40(2).

Hybrid Parareal Spectral Deferred Correction method[2]

Parareal SDC:

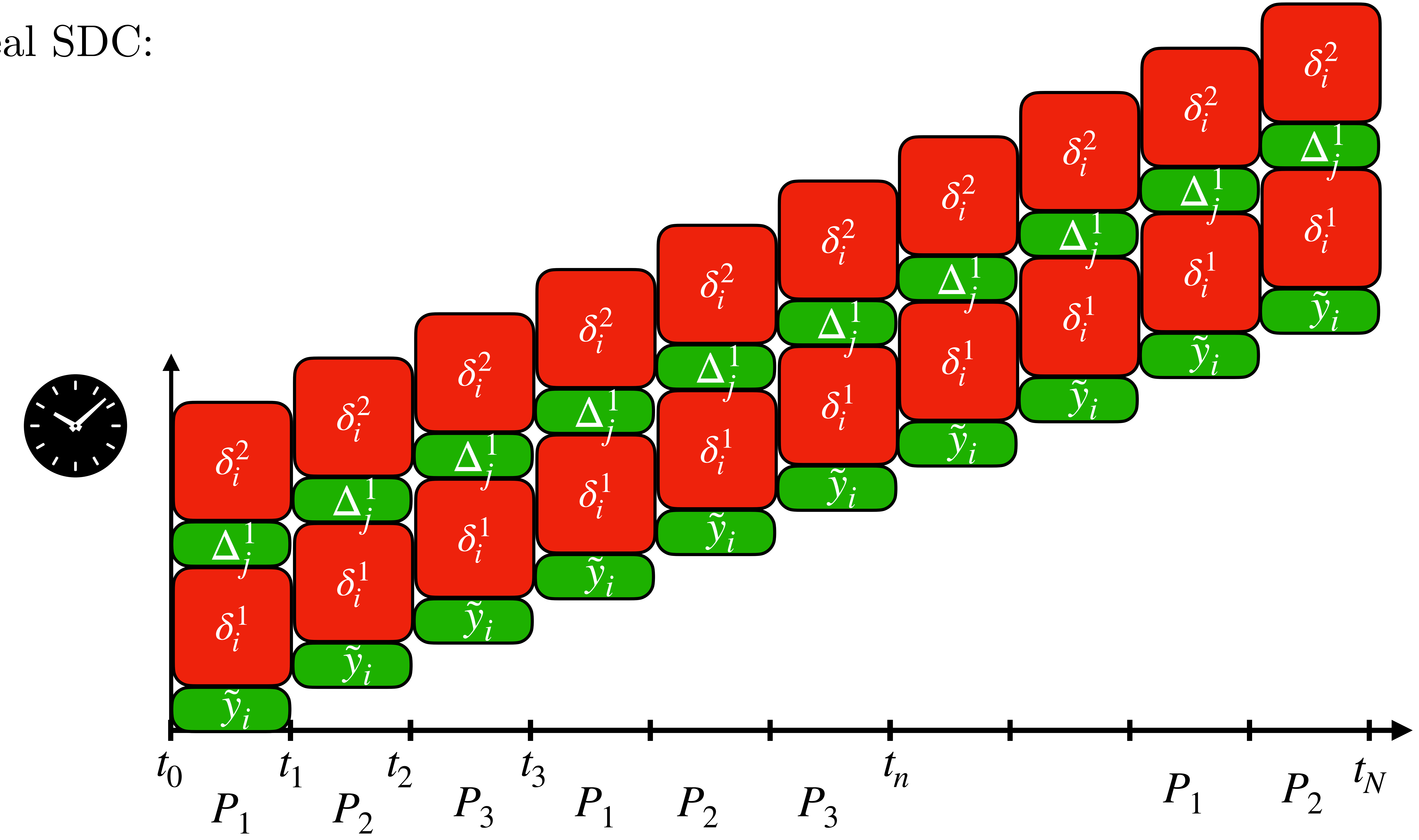




[2] Minion, M., Williams, S. 2008, 2010.

Parareal Spectral Deferred Correction method[2]

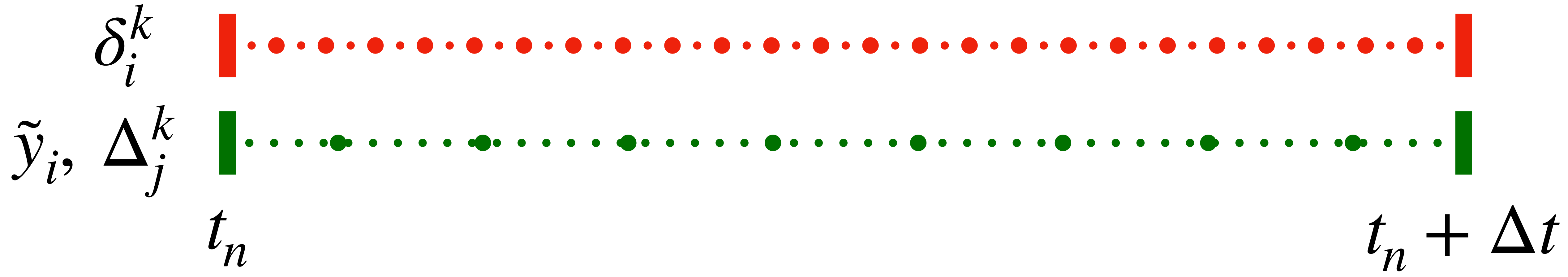
Parareal SDC:

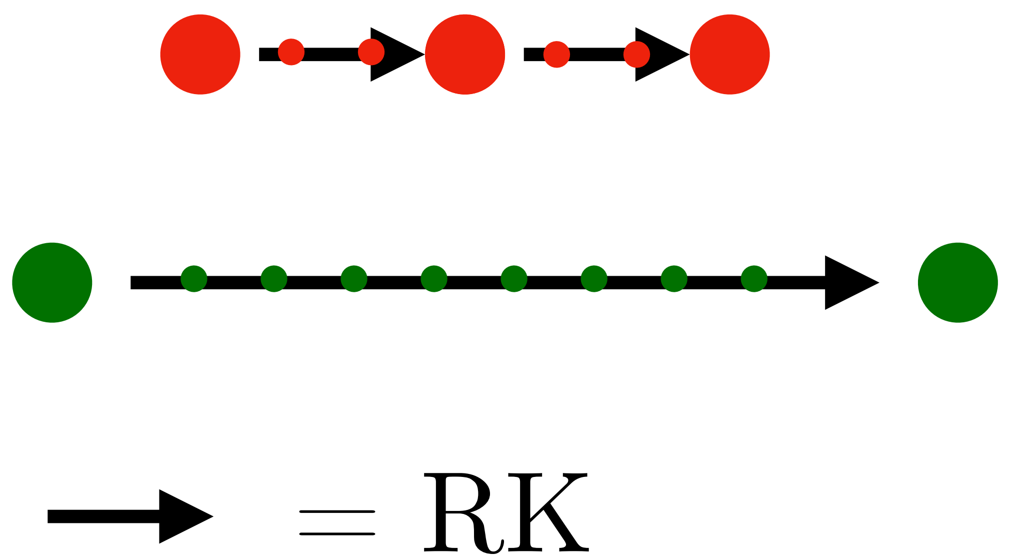


[2] Minion, M., Williams, S. 2008, 2010.

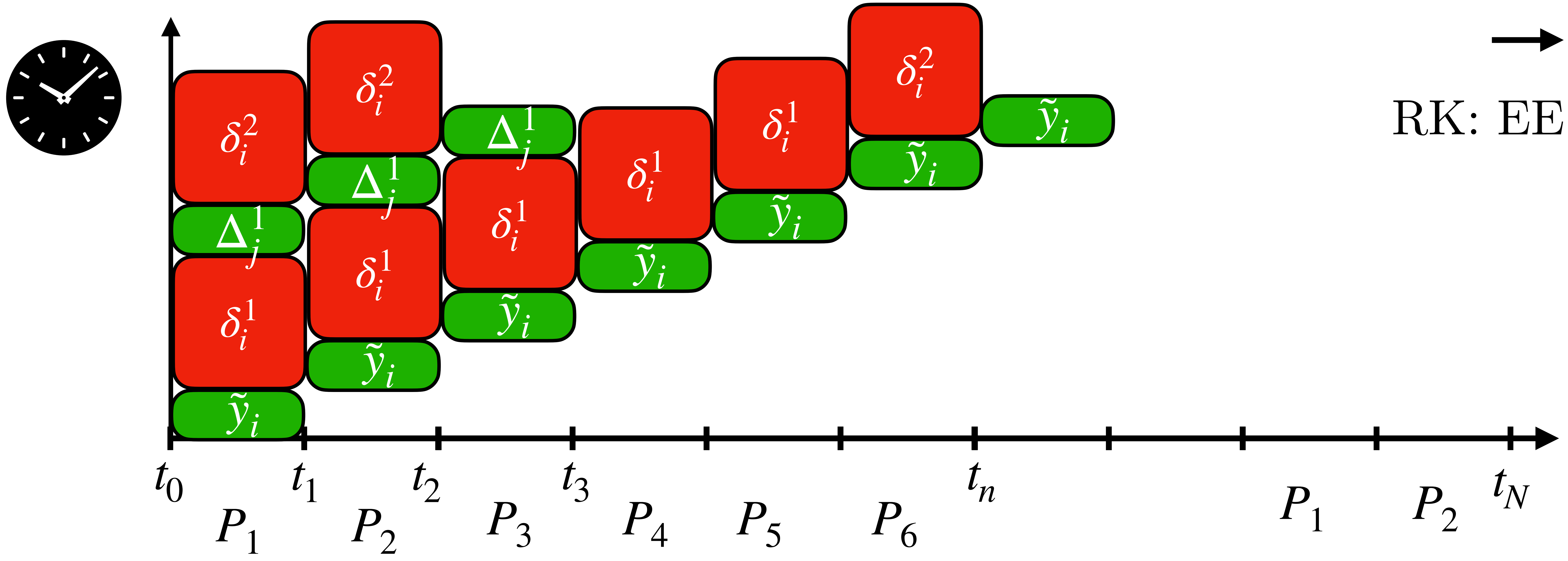
Parareal Spectral Deferred Correction method[2]

Parareal SDC:



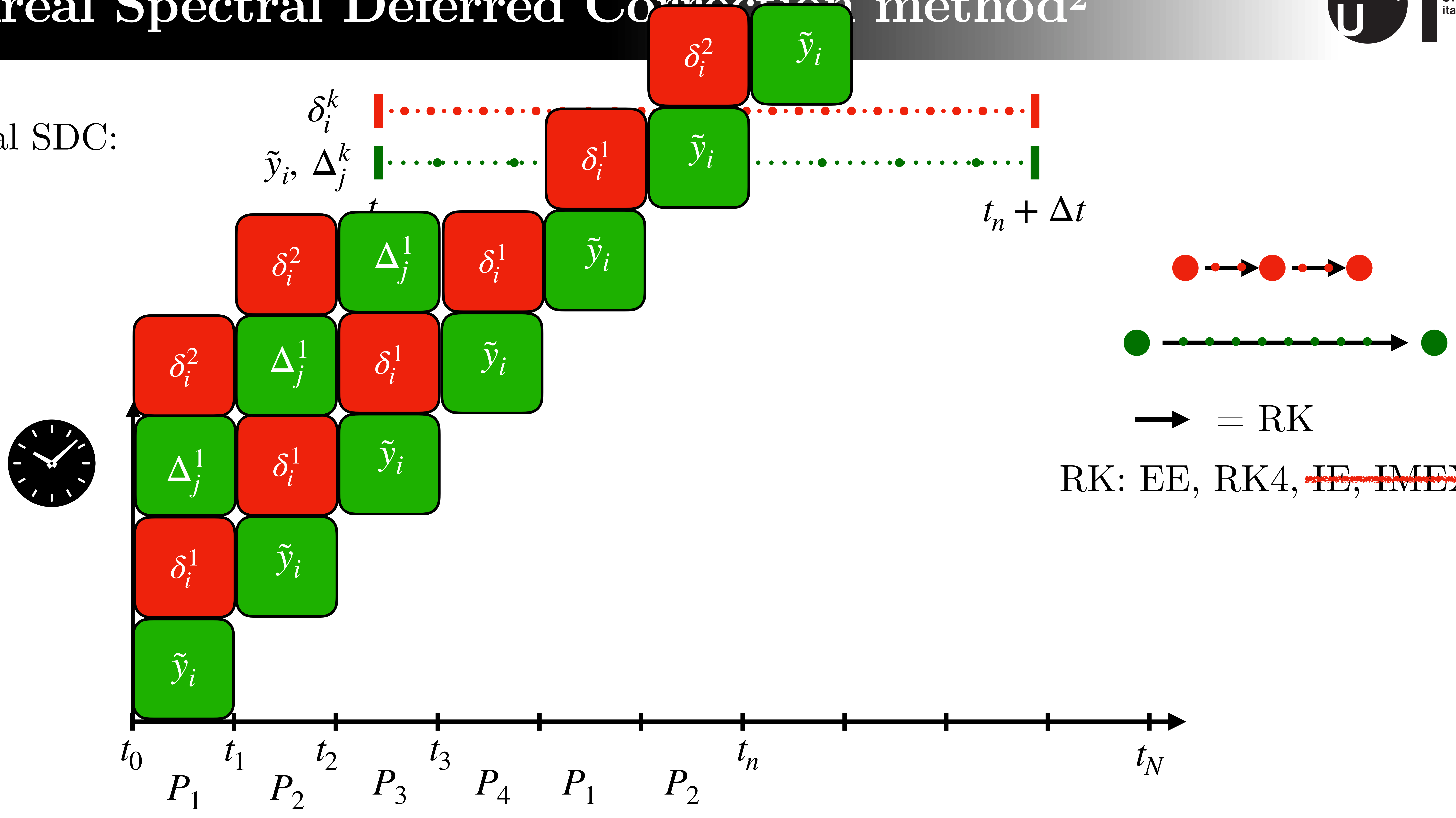


RK: EE, RK4, ~~IE, IMEX,...~~



Parareal Spectral Deferred Correction method²

Parareal SDC:



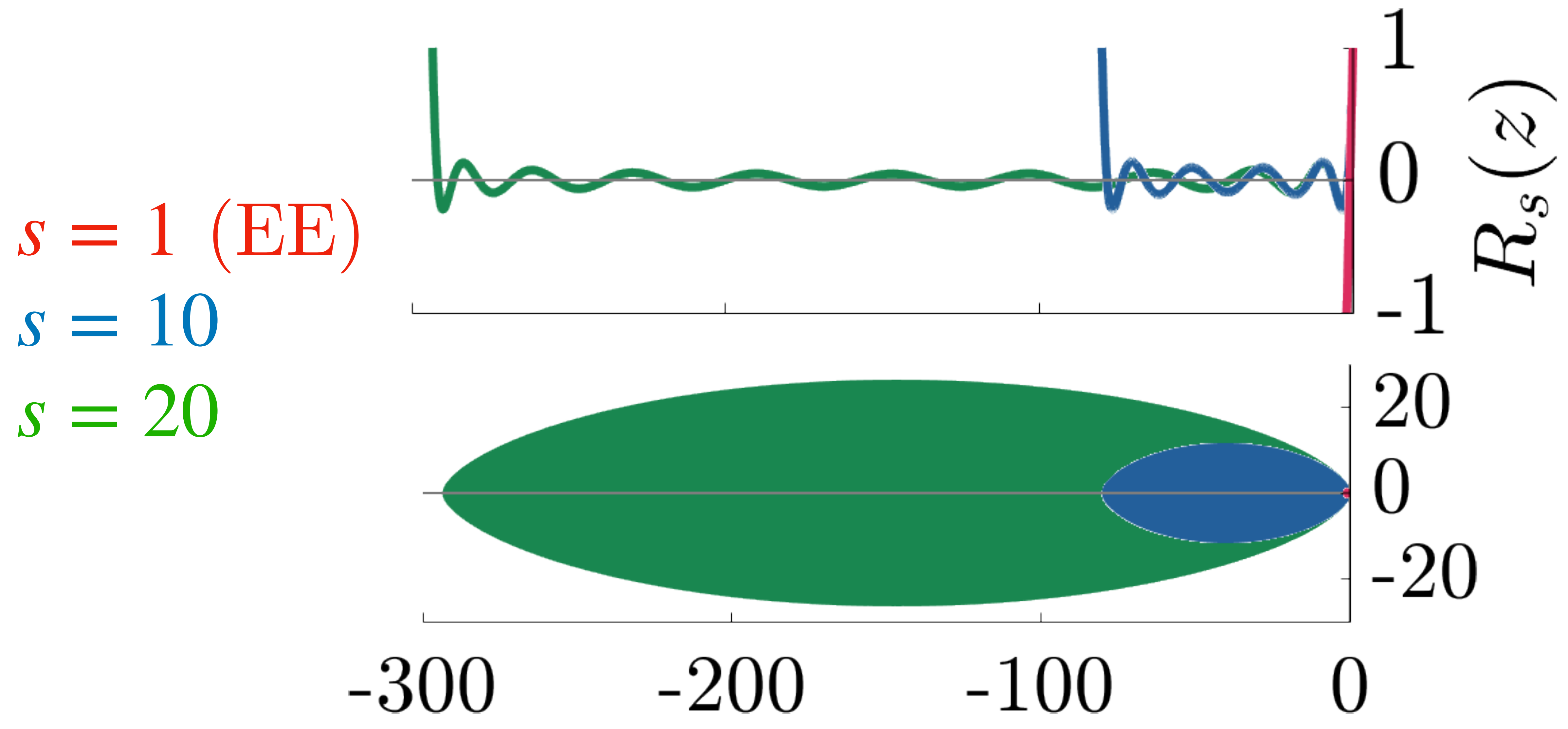
RK: EE, RK4, ~~IE, IMEX,...~~

The Second Kind Runge-Kutta-Chebyshev method

One step of RKU is given by

$$k_0 = y_0, \qquad k_1 = k_0 + \mu_1 \Delta t\, f(k_0),$$
$$k_j = \nu_j\, k_{j-1} + \kappa_j\, k_{j-2} + \mu_j \Delta t\, f(k_{j-1}), \quad j = 2, \ldots, s,$$
$$y_1 = k_s,$$

with s satisfying $\Delta t\, \rho(\partial f / \partial y) \leq (2/3)s(s+2)$.



- No step size restriction: just increase s.
- Fully explicit,
- There is a multirate version[3] for

$$y' = f_F(y) + f_S(y)\,.$$

  Good for multiscale ionic models or nonuniform grids, for instance.
- Works in mixed-precision arithmetic[4] (also in multirate). Good for CPU, memory, and energy savings in HPC computations.
- All flavors are straightforward to implement.

[3] Abdulle, A., Grote M., Rosilho G. 2022. *Math. Comput. (in press)*. [4] Croci M., Rosilho G. 2022. *J. Comput. Phys.* 464.

The Parareal SDC RKU method

- Fix collocation points $c_1, \dots, c_s$ in $[t_n, t_n + \Delta t]$ (Lobatto, Radau,...),
- Compute approximations $\tilde{y}_i$ at c_i with RKU.

Then iterate on:

- Define $\tilde{y}(t) = \sum L_i(t)\tilde{y}_i$,
- Approximate $\varepsilon(t) \approx \sum L_i(t)\varepsilon(c_i)$. $\varepsilon(c_i)$ computed with Lobatto, Radau,.. quadrature rules.
- Compute δ_i at $c_1, \dots, c_s$ solving the error equation with RKU

$$\delta(t_2) = \delta(t_1) + \int^{t_2} \big(f(\bar{y}(s) + \delta(s)) - f(\bar{y}(s))\big)\, \mathrm{d}s + \varepsilon(t_2) - \varepsilon(t_1)\,.$$

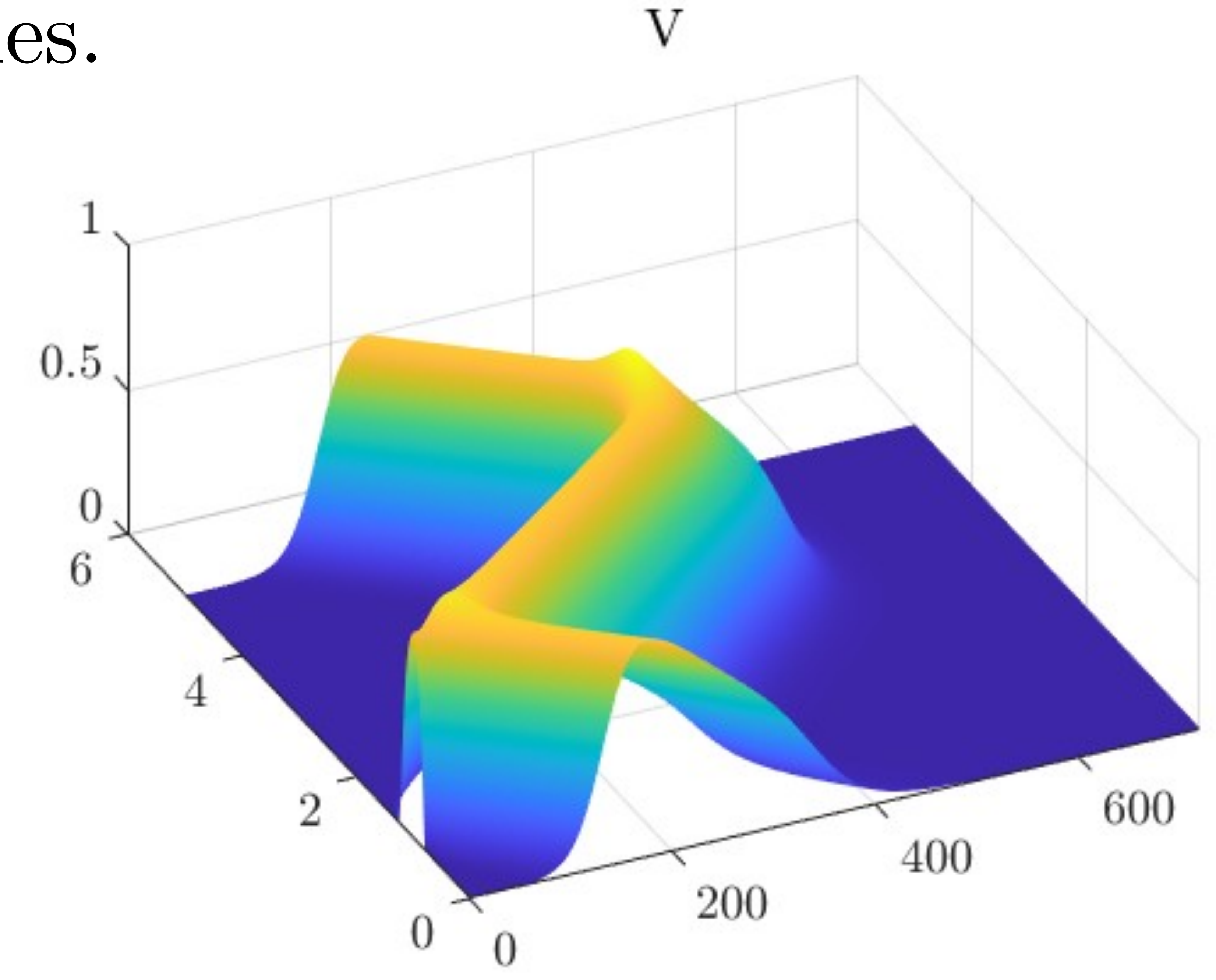
$$d_0 = \delta_i, \qquad d_1 = d_0 + \mu_1 \Delta t\, (f(\tilde{y}^0 + d_0) - f(\tilde{y}^0)) + \varepsilon^1 - \varepsilon^0,$$
$$d_j = \nu_j\, d_{j-1} + \kappa_j\, d_{j-2} + \mu_j \Delta t\, (f(\tilde{y}^{j-1} + d_{j-1}) - f(\tilde{y}^{j-1})) + \varepsilon^j - \nu_j \varepsilon^{j-1} - \kappa_j \varepsilon^{j-2},$$
$$\delta_{i+1} = d_s,$$

Numerical Experiment

Consider $\Omega = [0,5]cm$, $T = 720ms$ and

$$\begin{aligned} \partial_t u &= \nu \Delta u - I_{ion}(u,z) + I_s(t), && \text{in } \Omega \times [0,T] \\ z' &= g(u,z), && \text{in } \Omega \times [0,T] \end{aligned}$$

With periodic boundary conditions on u, $\nu = 10^{-3}$, I_{ion}, g an ionic model and z its state variables.

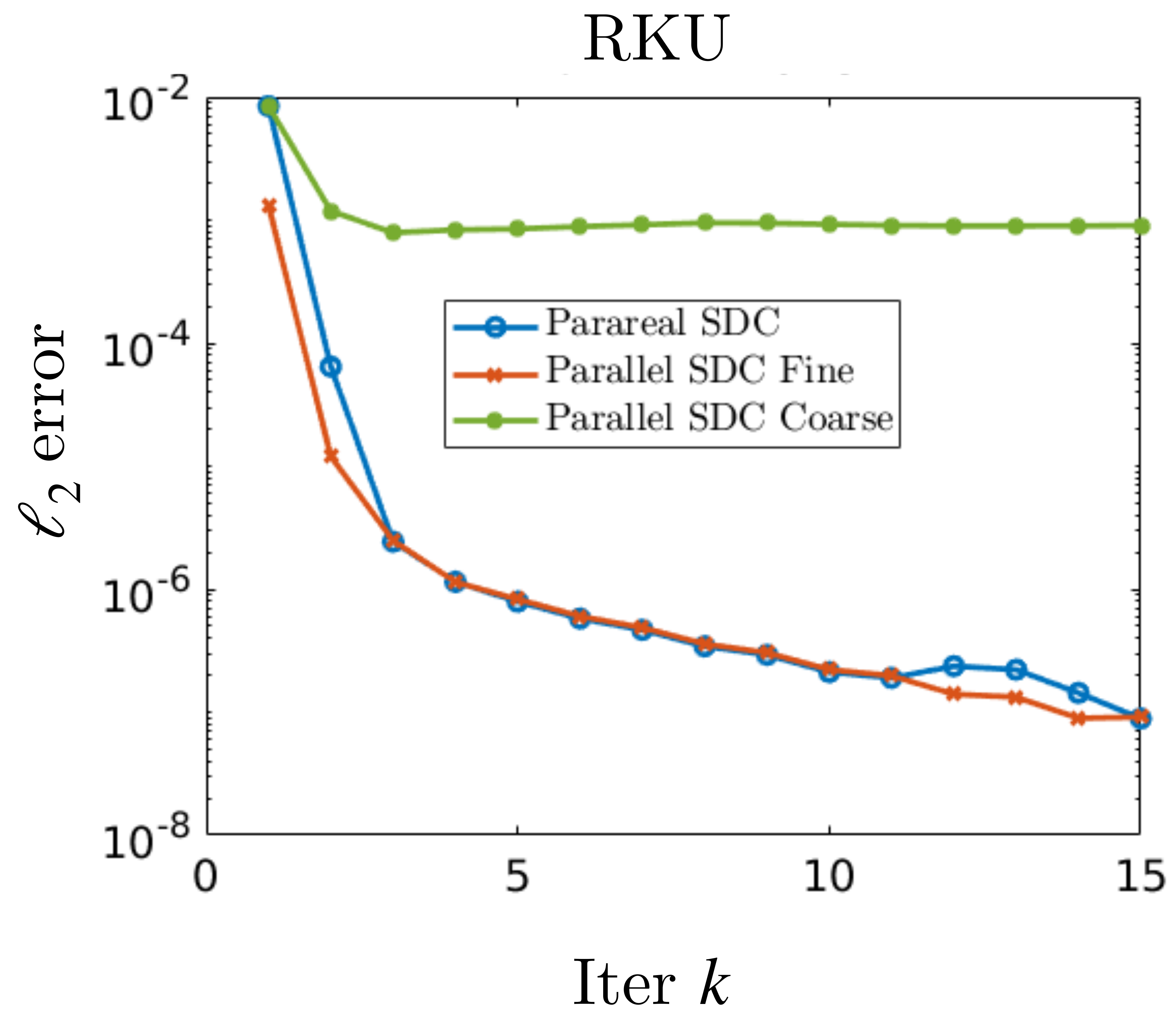


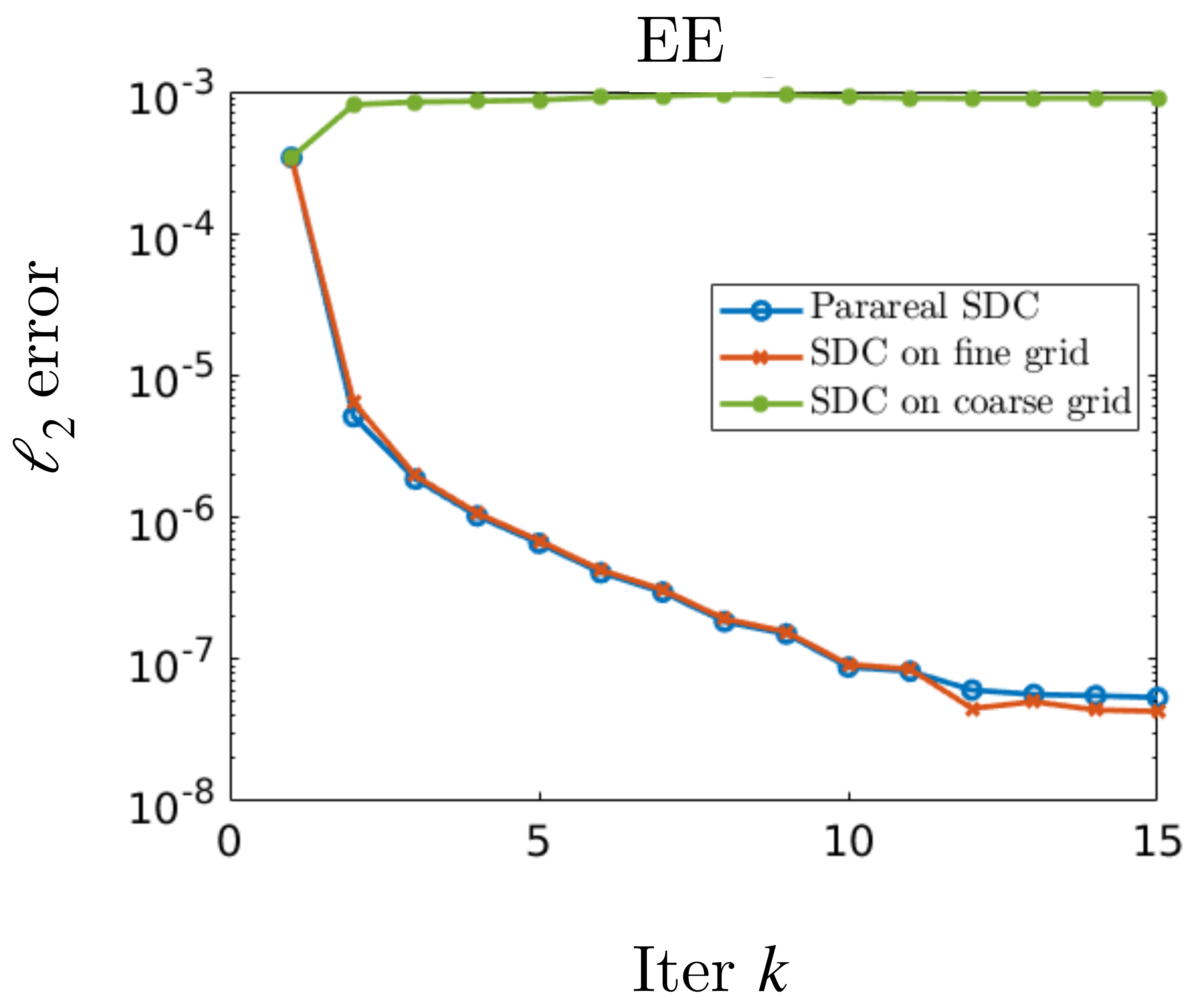
- Discretize with finite differences,
- Solve with Parareal SDC using EE, RKU, and mRKU.

We use $240 \times 3ms$ subintervals (cores)

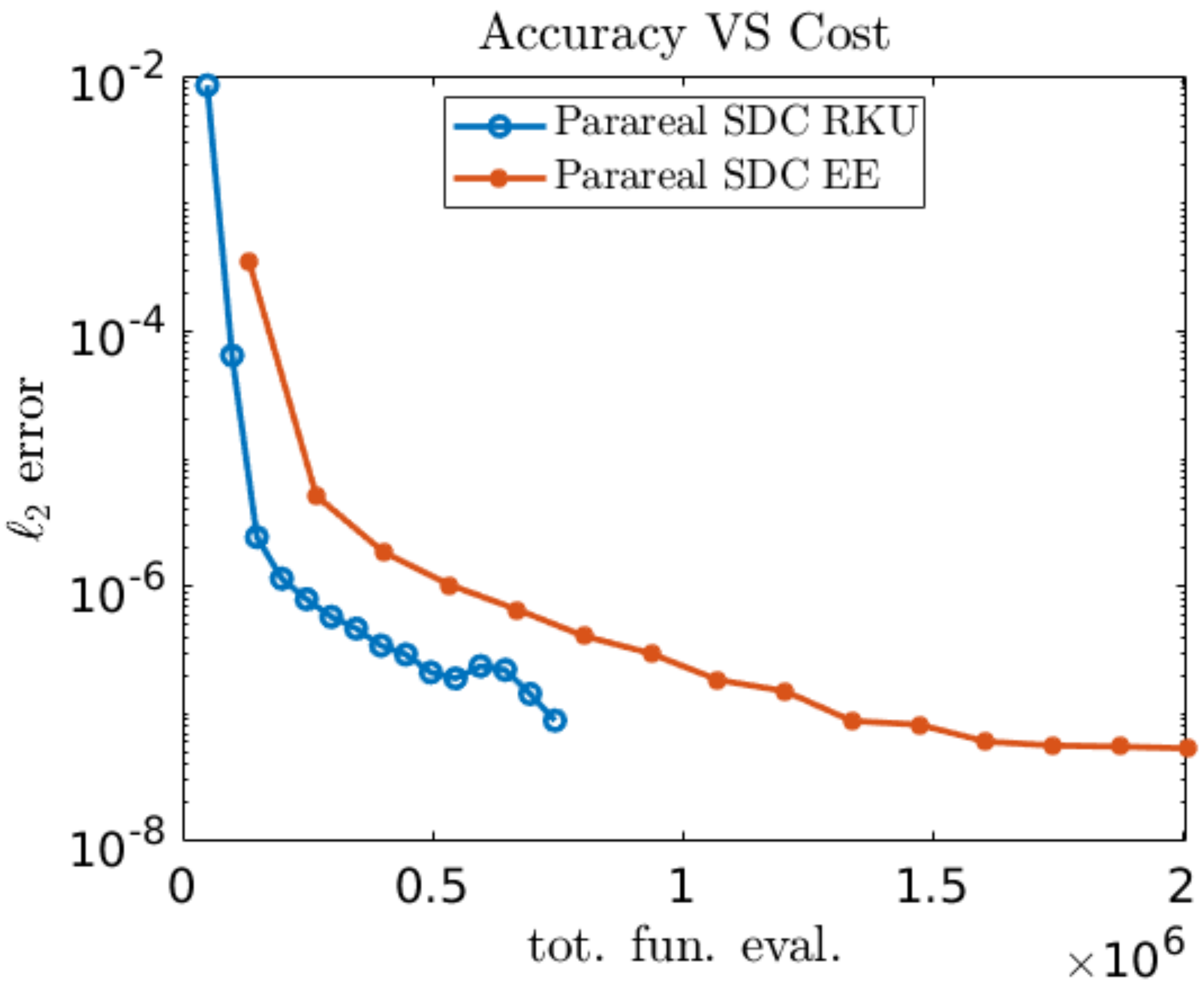
- 6 Lobatto collocation nodes on coarse grid,
- 40 Lobatto collocation nodes on fine grid.
- 15 Parareal iterations.

Numerical Experiment





Numerical Experiment



Costs per iter per time slice $[t_n, t_n + \Delta t]$

On coarse grid • • •

Cost EE: ≈ 269

Cost RKU: ≈ 58

On fine grid ......

Cost EE: ≈ 289

Cost RKU: ≈ 149

Multirate RKU method

Consider

$$y' = f_F(y) + f_S(y), \qquad y(0) = y_0,$$

with f_F stiff but cheap and f_S mildly stiff but expensive.

For RKU, the number of costly f_S evaluations is dictated by a few stiff terms in f_F.

We solve the *modified problem*

$$y_\eta' = f_\eta(y_\eta), \qquad y(0) = y_0,$$

With $\eta \geq 0$ a parameter used to tune the stiffness. For $\eta = \mathcal{O}(\rho_S^{-1})$ and the stiffness of f_η is same as f_S.

The *averaged force* is defined as

$$f_\eta(y) = \frac{1}{\eta}\left(u(\eta) - y\right)$$

With *auxiliary solution* u given by

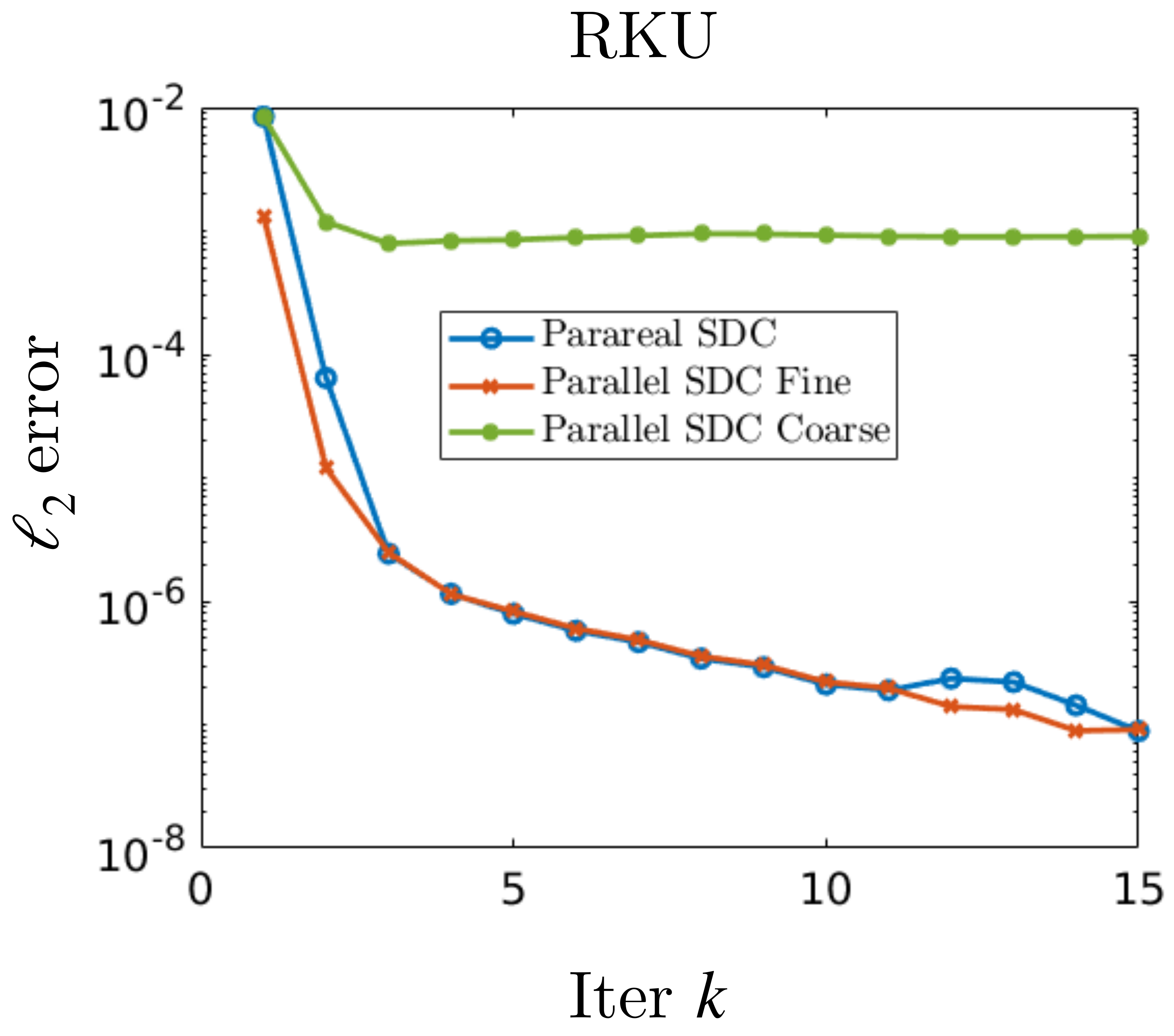
$$u' = f_F(u) + f_S(y), \qquad u(0) = y\,.$$

The multirate RKU method is given by:

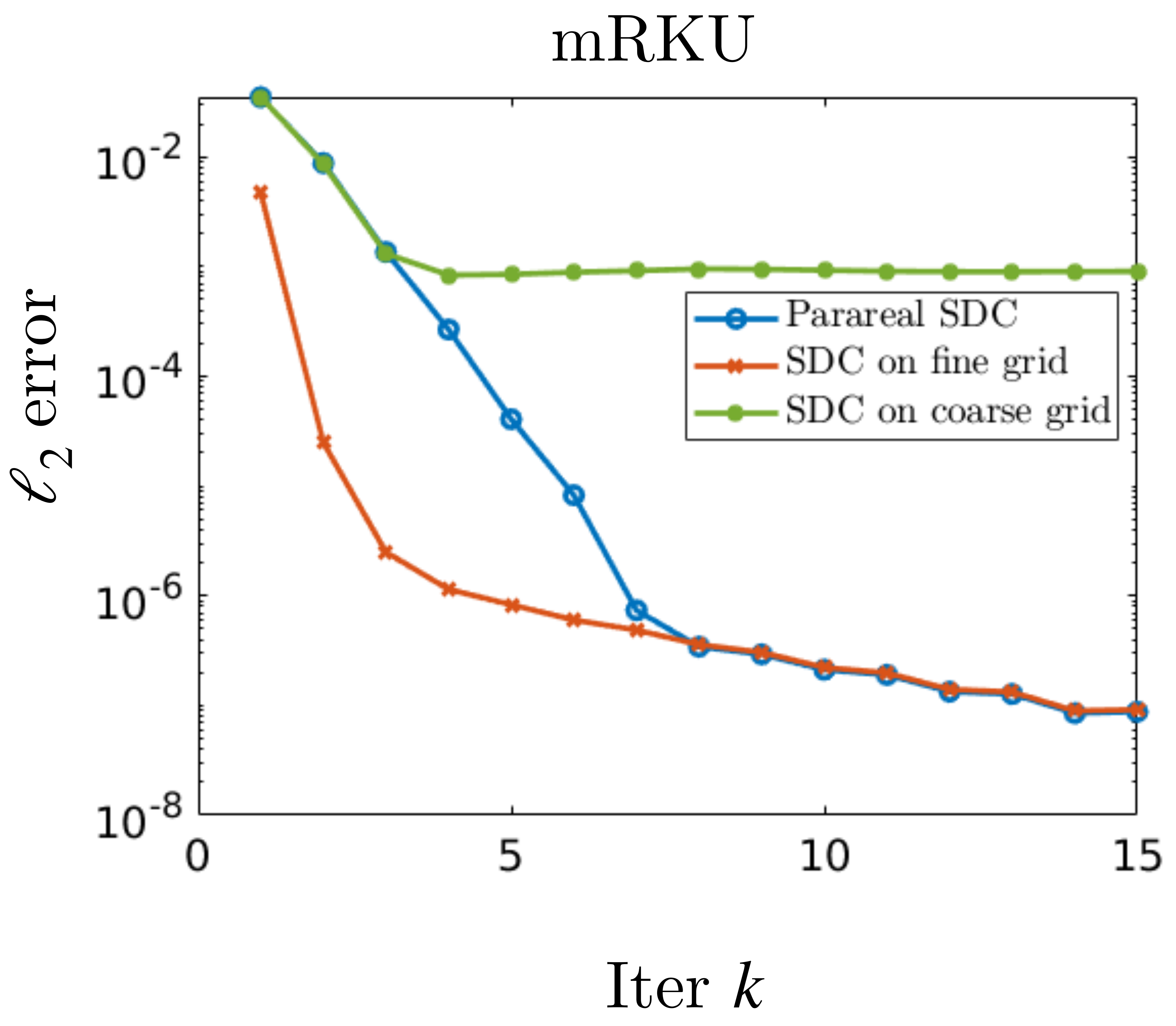
- Integrate $y_\eta' = f_\eta(y_\eta)$ with a RKU method.
- To evaluate f_η solve $u' = f_F(u) + f_S(y)$ with another RKU method.

Numerical Experiment

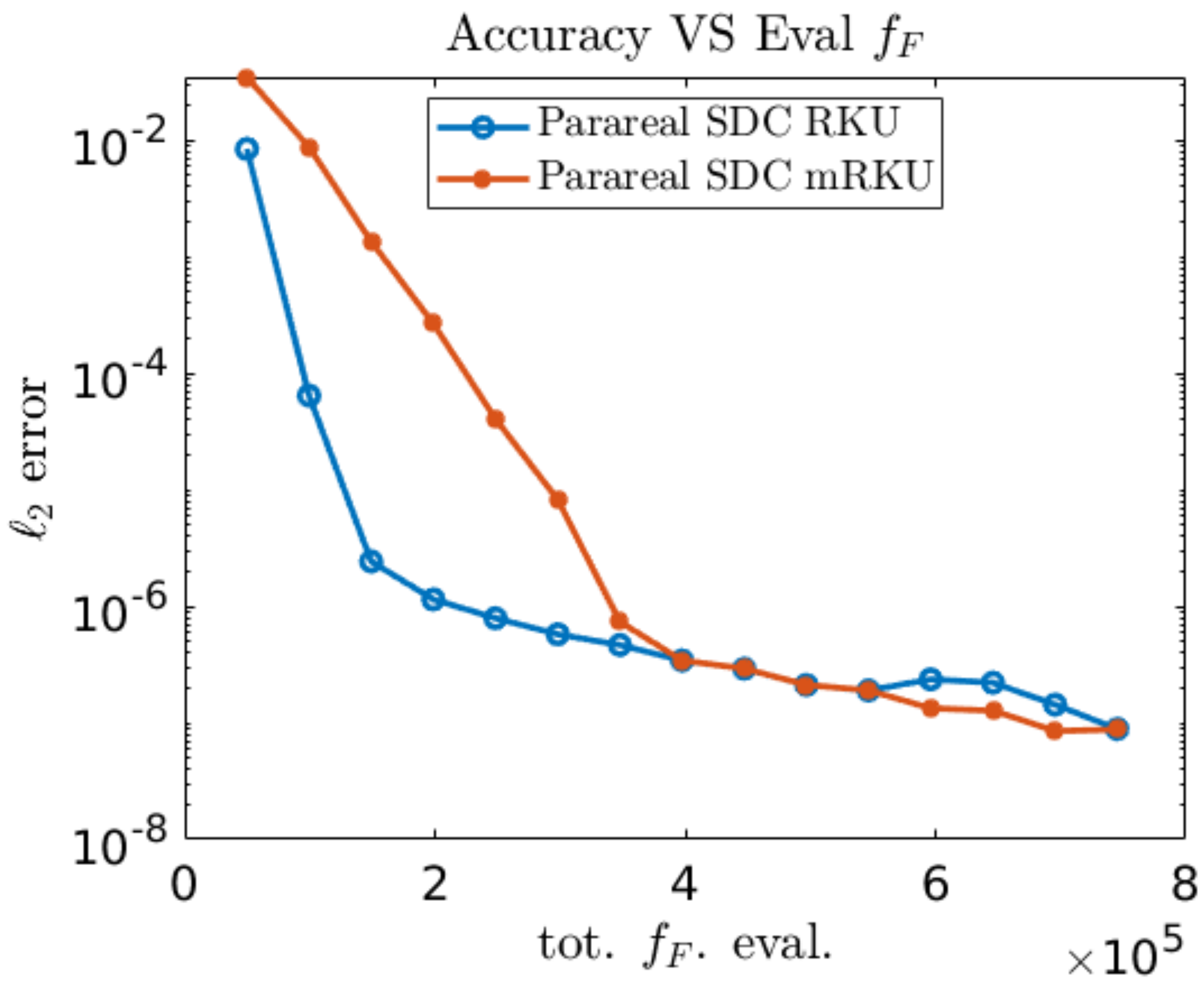
$$f(u,z) = \begin{pmatrix} \nu\Delta u - I_{ion}(u,z) + I_s(t) \\ g(u,z) \end{pmatrix}$$

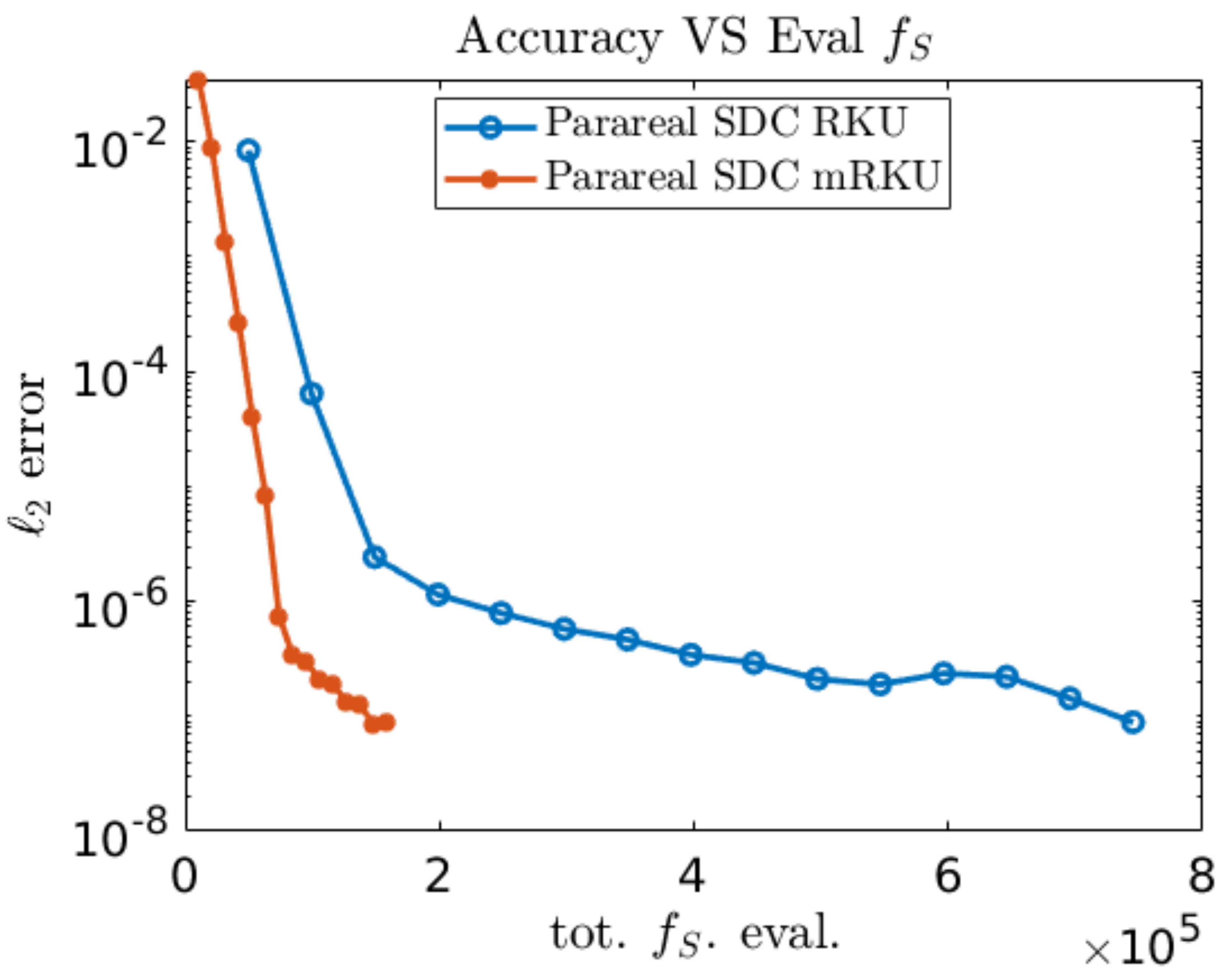


$$f_F(u,z) = \begin{pmatrix} \nu\Delta u \\ 0 \end{pmatrix} \quad f_S(u,z) = \begin{pmatrix} -I_{ion}(u,z) + I_s(t) \\ g(u,z) \end{pmatrix}$$



Numerical Experiment





Conclusions

F. Eval.	EE	RKU	mRKU
f_S	558	207	44
f_F	558	207	207

Thank you!

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.