PinT Multirate Explicit Stabilized Methods

Giacomo Rosilho de Souza, Simone Pezzuto, Rolf Krause

Università della Svizzera Italiana, Lugano, Switzerland









PinT 2022 - CIRM - Marseille

Contents

Parareal Spectral Deferred Correction and Multirate Explicit Stabilized methods

- Hybrid Parareal Spectral Deferred Correction,
- Explicit stabilized methods,
- 



&

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,

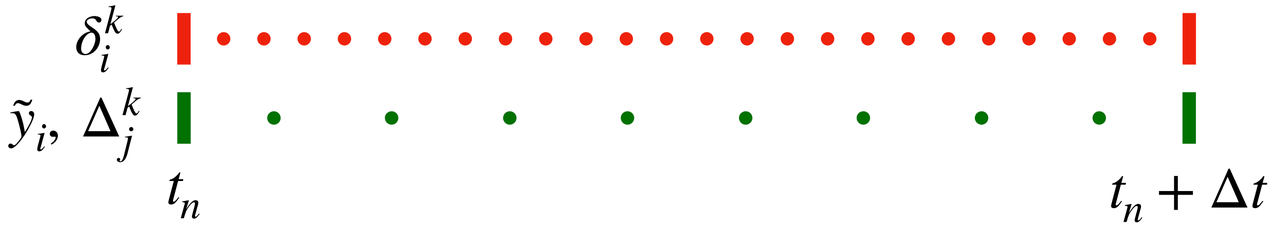
Then iterate on:

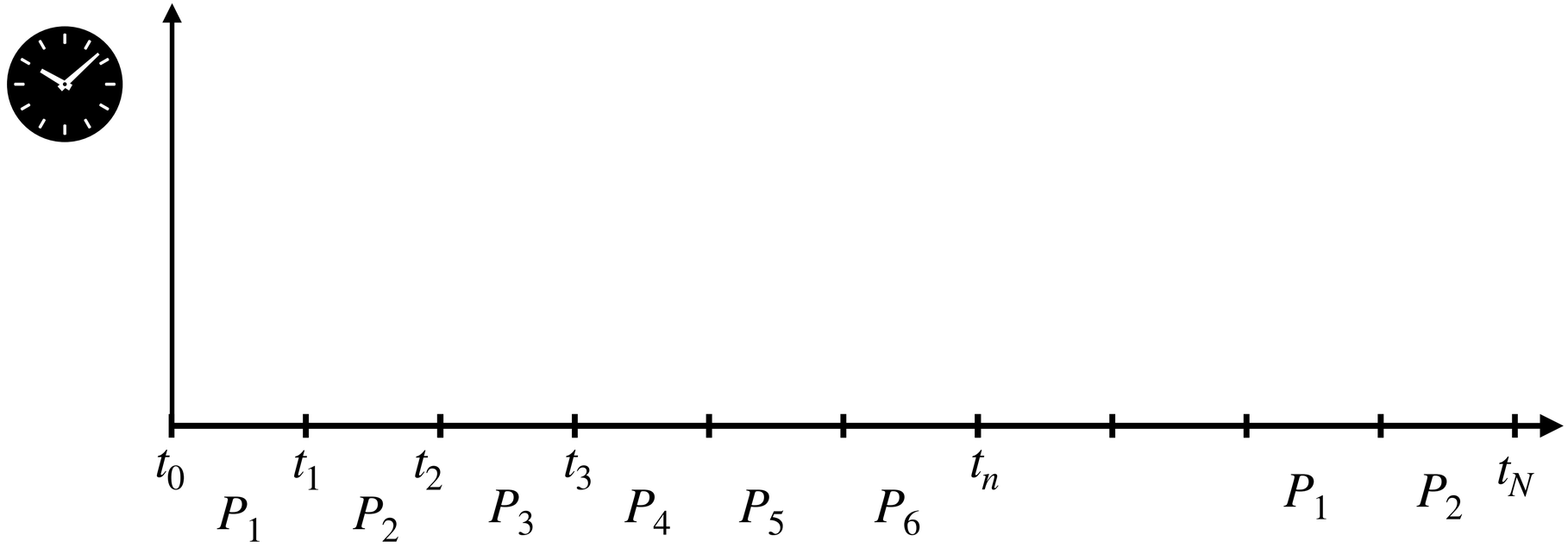
- Interpolate and form $\tilde{y}(t) = \sum L_i(t)\tilde{y}_i$,
- Approximate $\varepsilon(t)$ with <u>care</u>,
- Compute δ_i and correct $\tilde{y}_i + \delta_i \to \tilde{y}_i$.

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

Hybrid Parareal Spectral Deferred Correction method[2]

Parareal SDC:

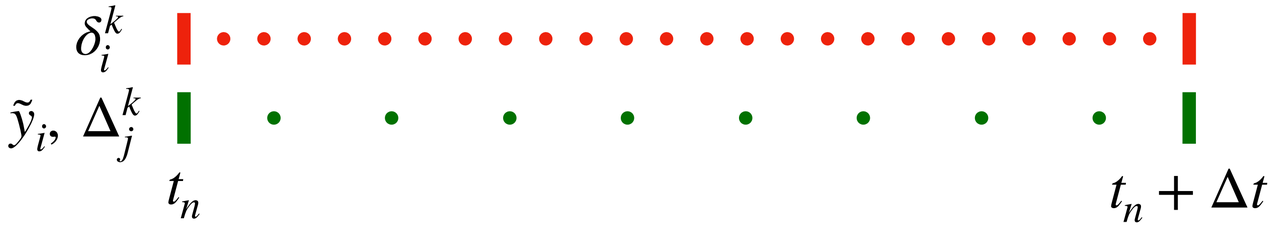


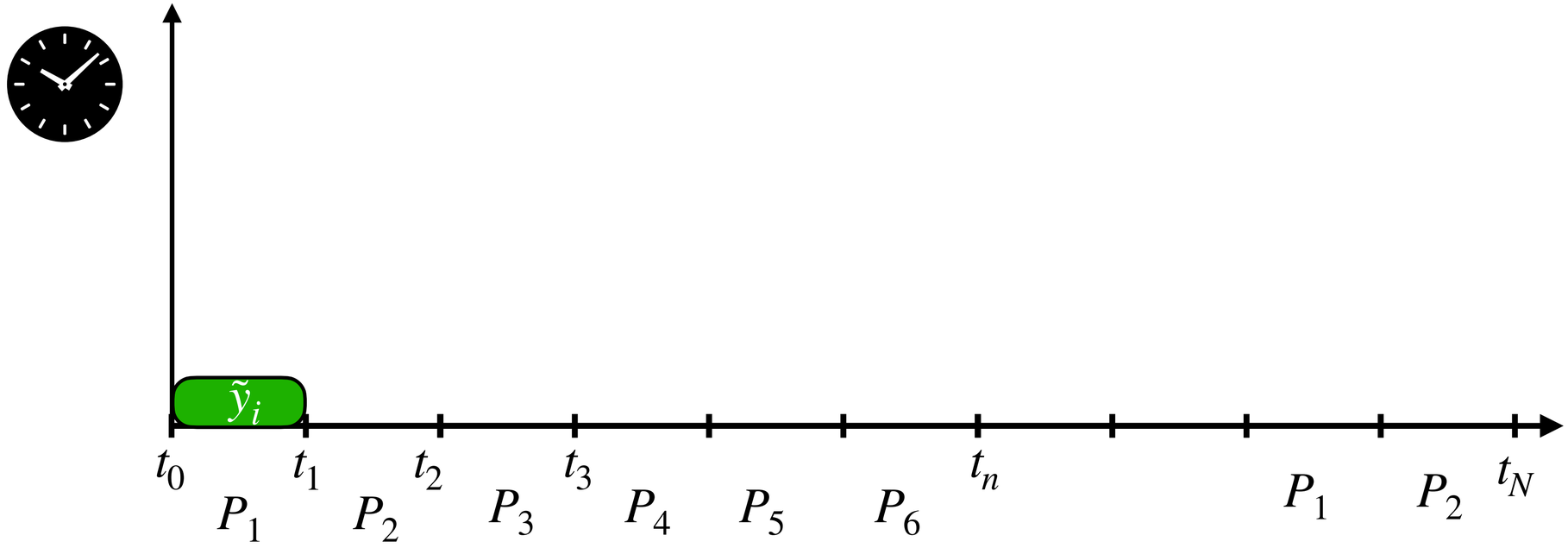


[2] Minion, M., Williams, S. 2008, 2010.

Hybrid Parareal Spectral Deferred Correction method[2]

Parareal SDC:

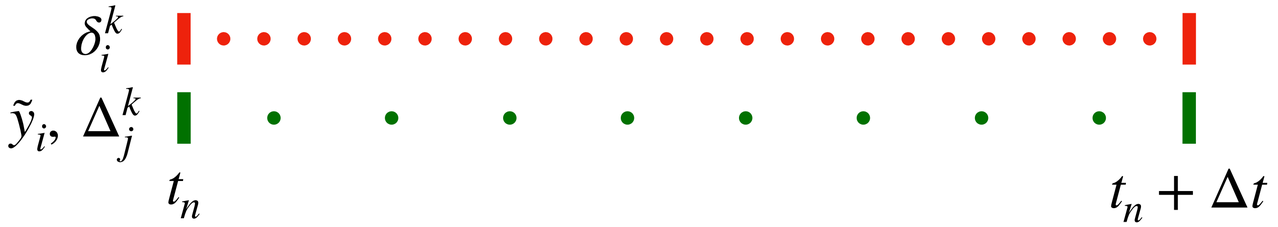


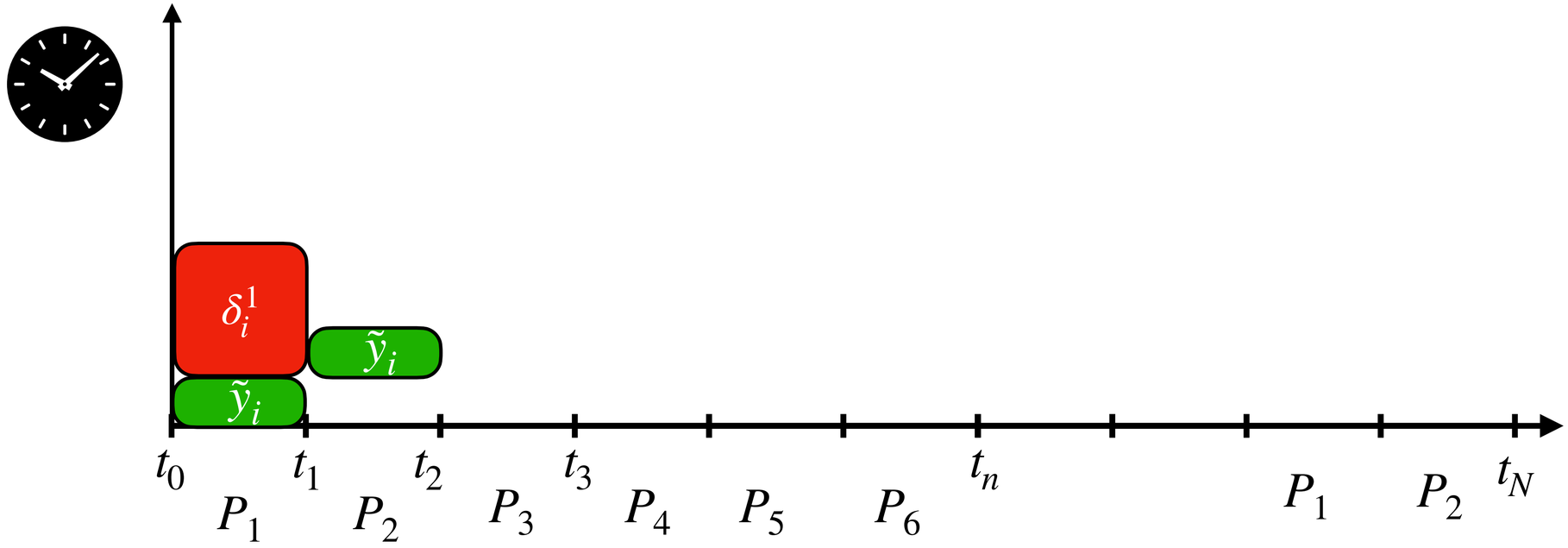


[2] Minion, M., Williams, S. 2008, 2010.

Hybrid Parareal Spectral Deferred Correction method[2]

Parareal SDC:

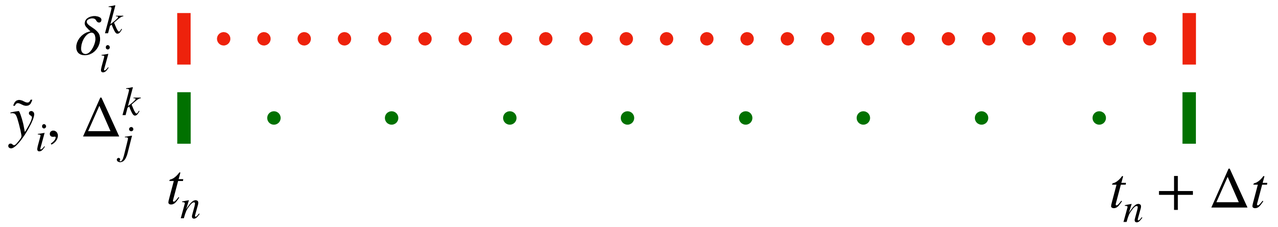


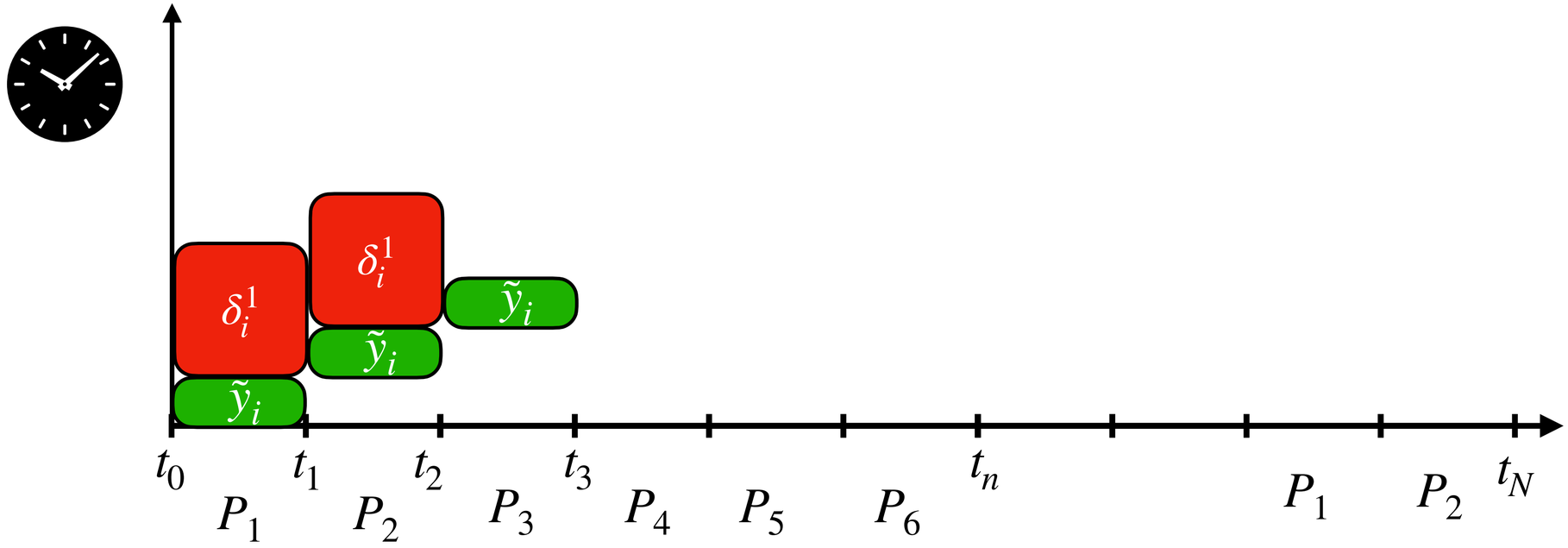


[2] Minion, M., Williams, S. 2008, 2010.

Hybrid Parareal Spectral Deferred Correction method[2]

Parareal SDC:

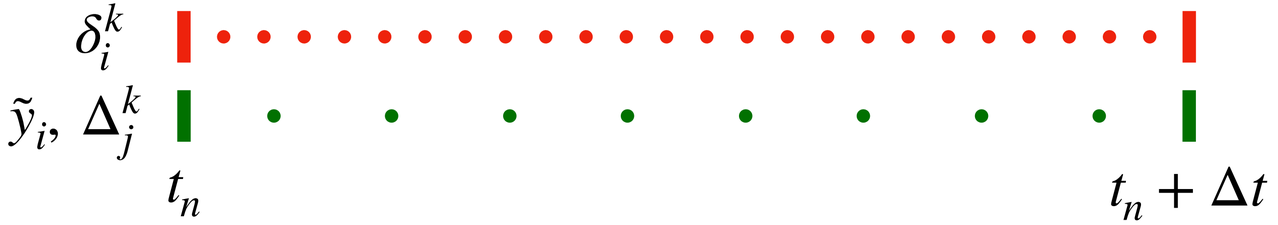


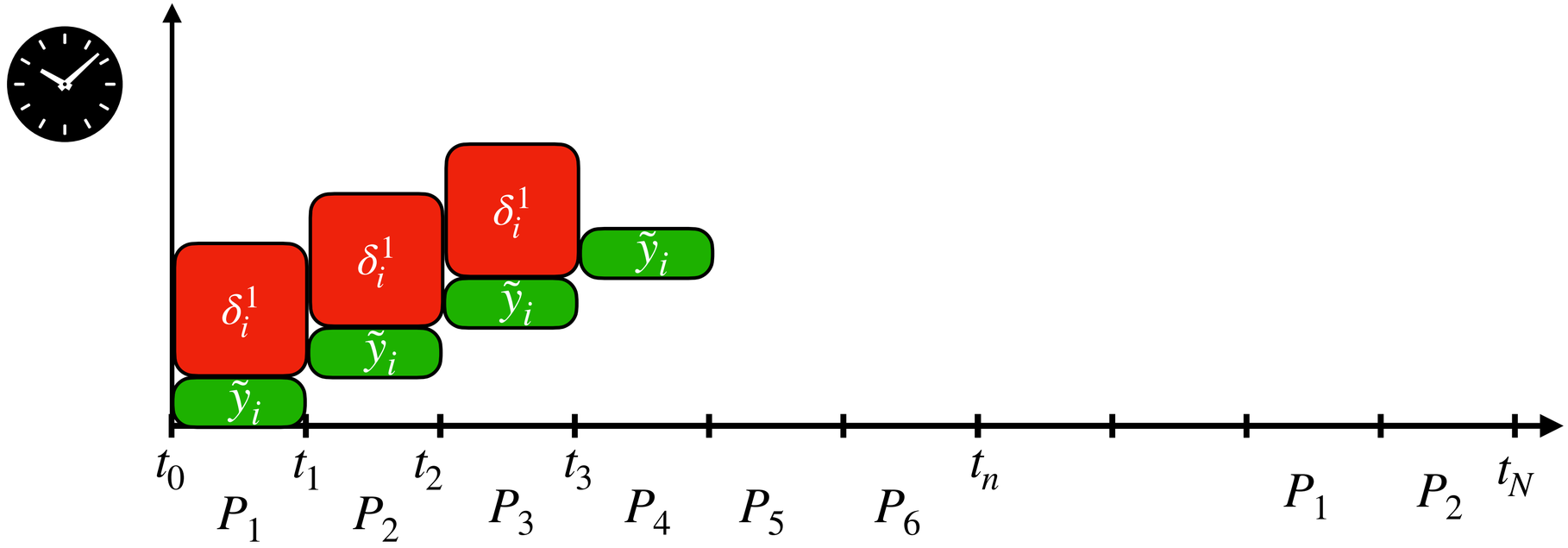


[2] Minion, M., Williams, S. 2008, 2010.

Hybrid Parareal Spectral Deferred Correction method[2]

Parareal SDC:

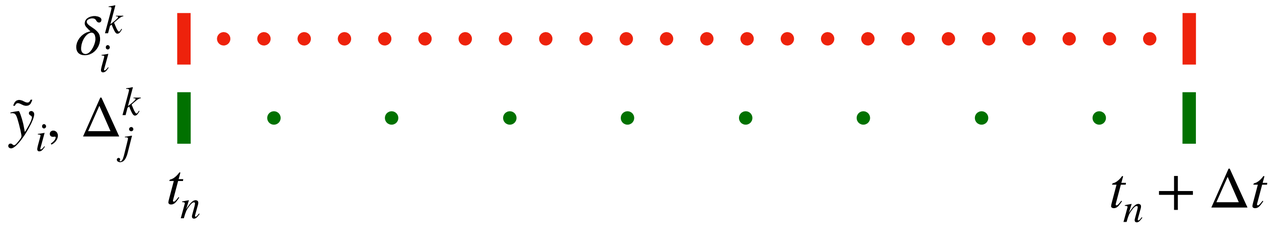


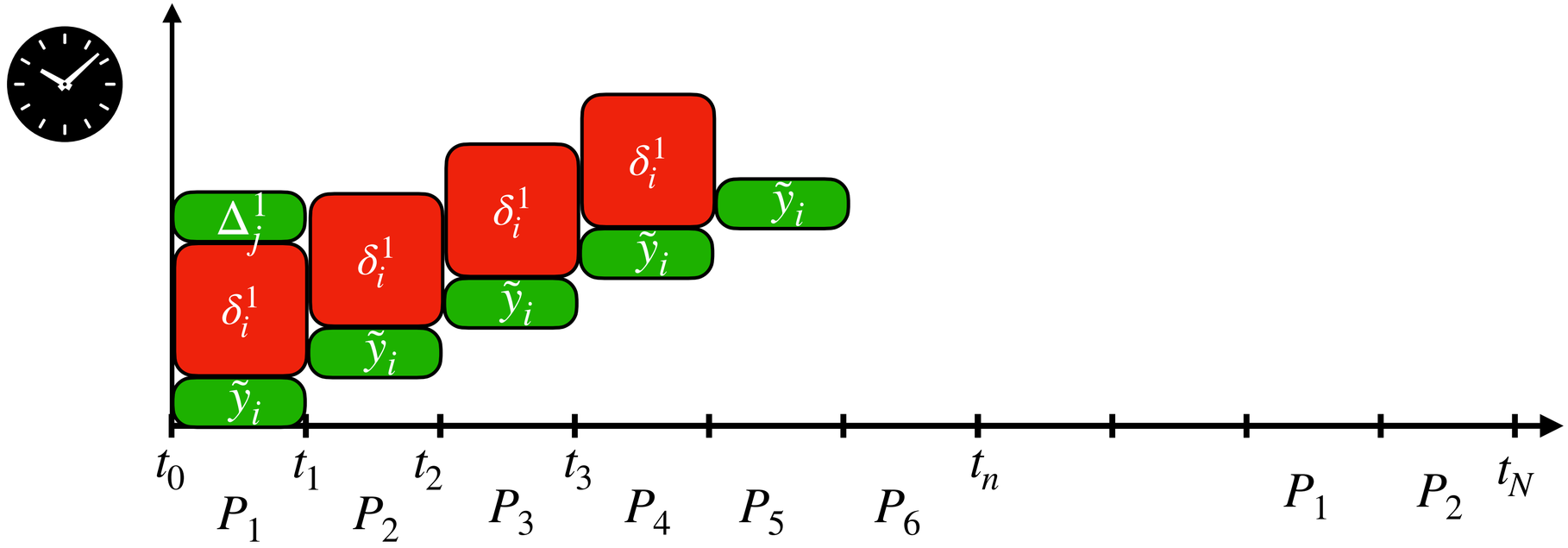


[2] Minion, M., Williams, S. 2008, 2010.

Hybrid Parareal Spectral Deferred Correction method[2]

Parareal SDC:

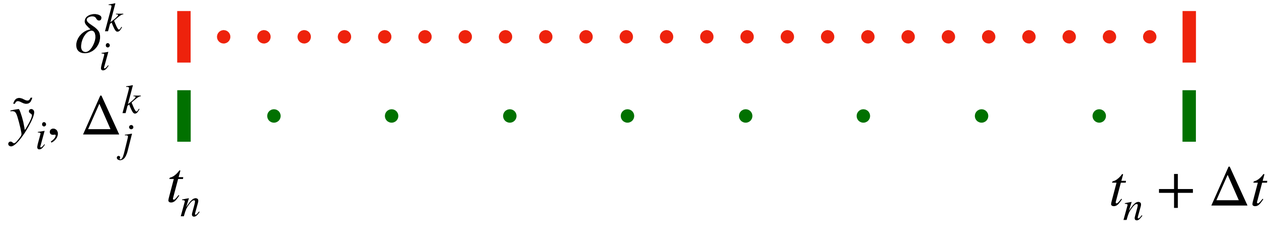


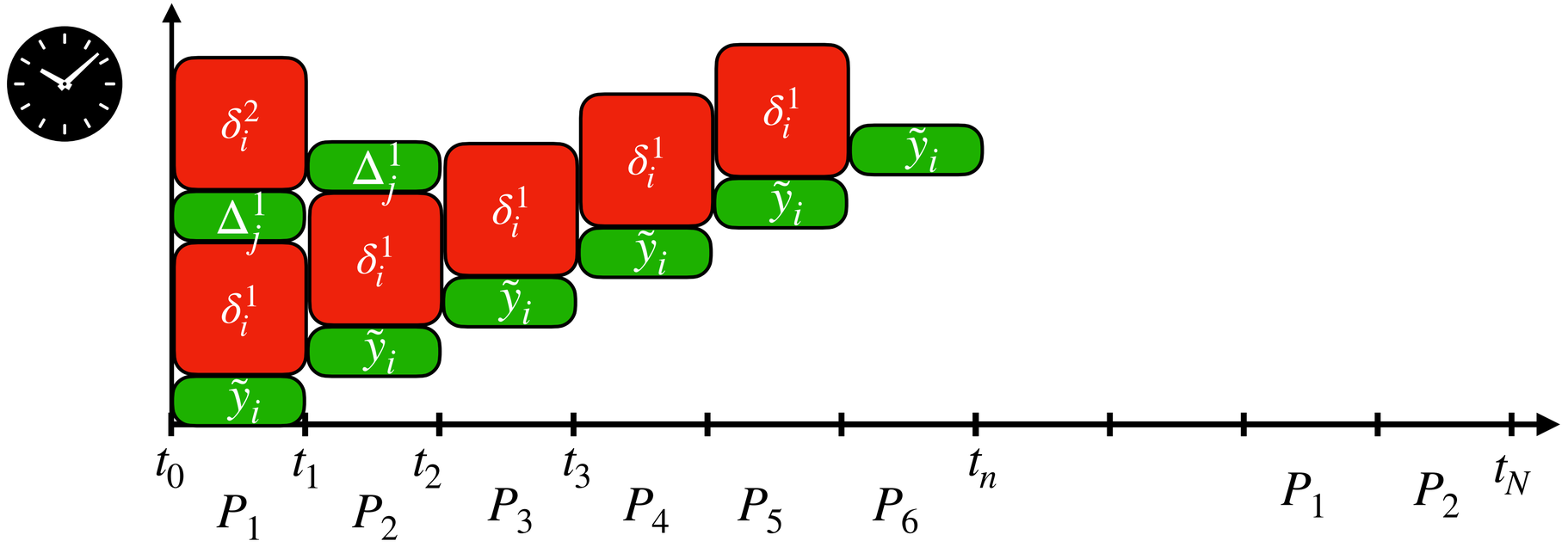


[2] Minion, M., Williams, S. 2008, 2010.

Hybrid Parareal Spectral Deferred Correction method[2]

Parareal SDC:

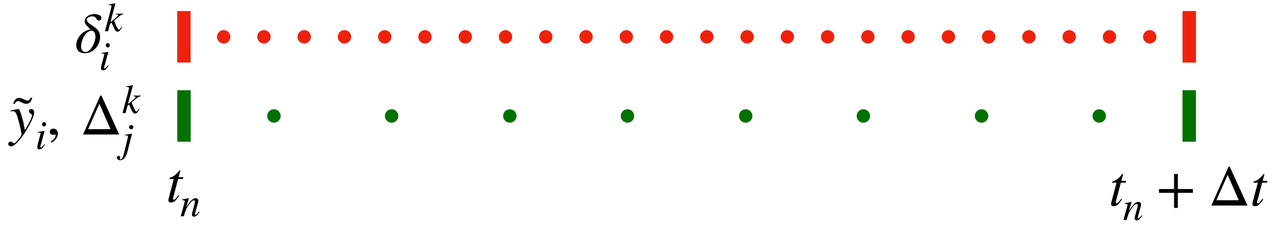


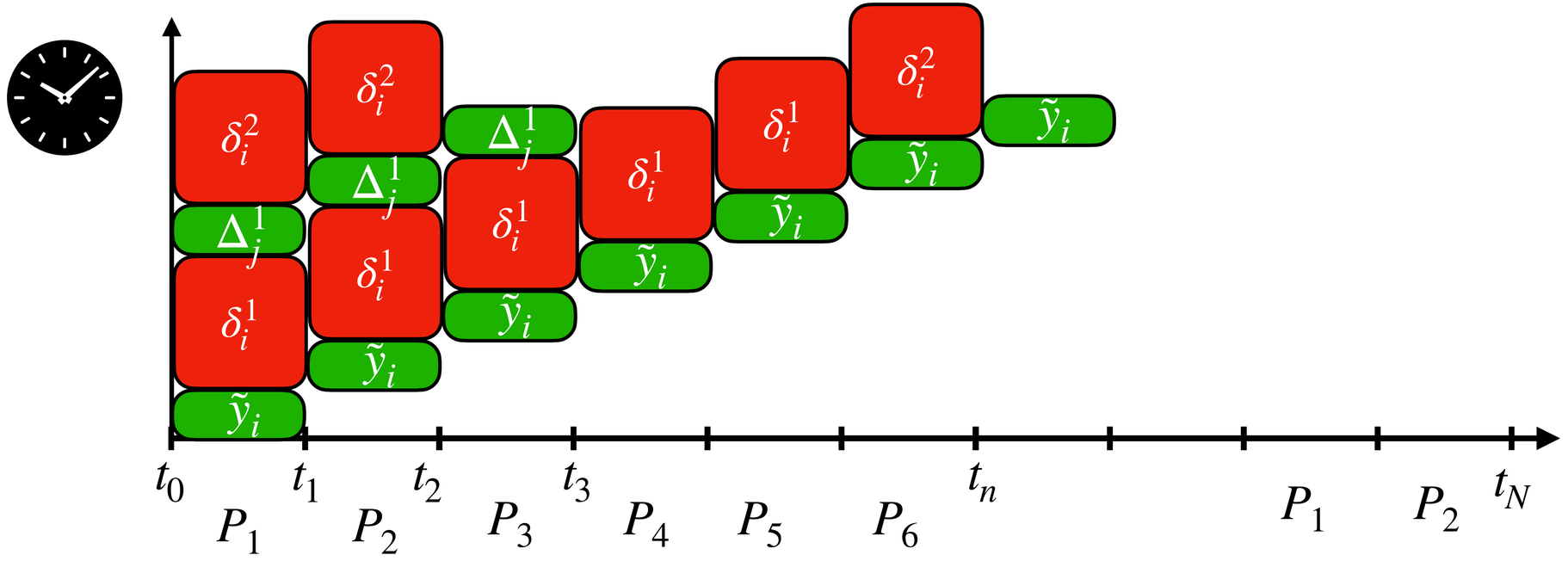


[2] Minion, M., Williams, S. 2008, 2010.

Hybrid Parareal Spectral Deferred Correction method[2]

Parareal SDC:

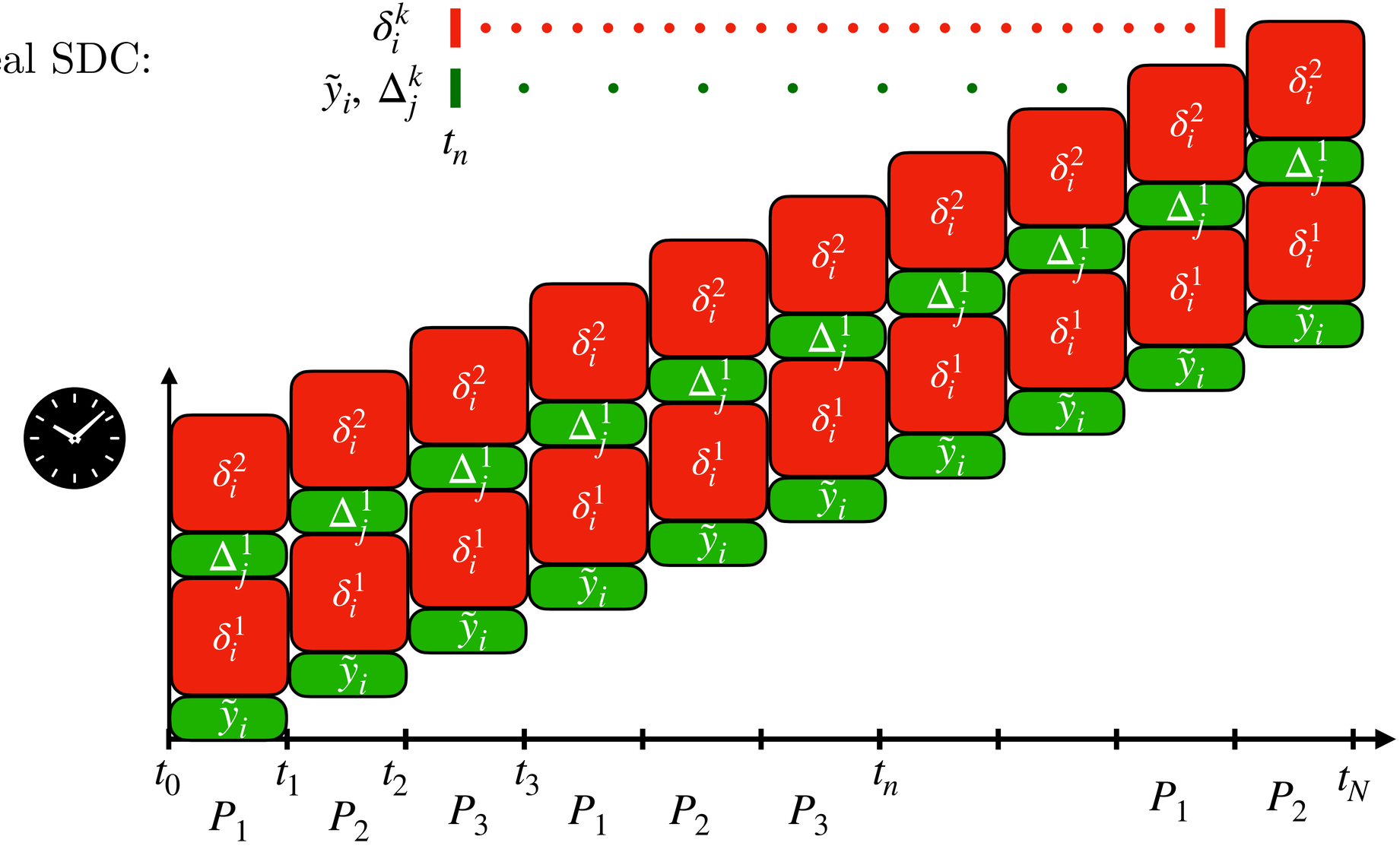




[2] Minion, M., Williams, S. 2008, 2010.

Parareal Spectral Deferred Correction method[2]

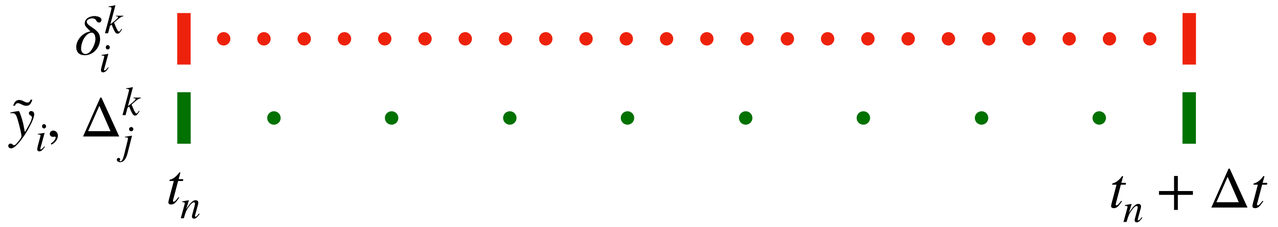
Parareal SDC:

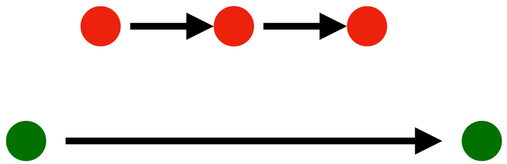


[2] Minion, M., Williams, S. 2008, 2010.

Parareal Spectral Deferred Correction method^2

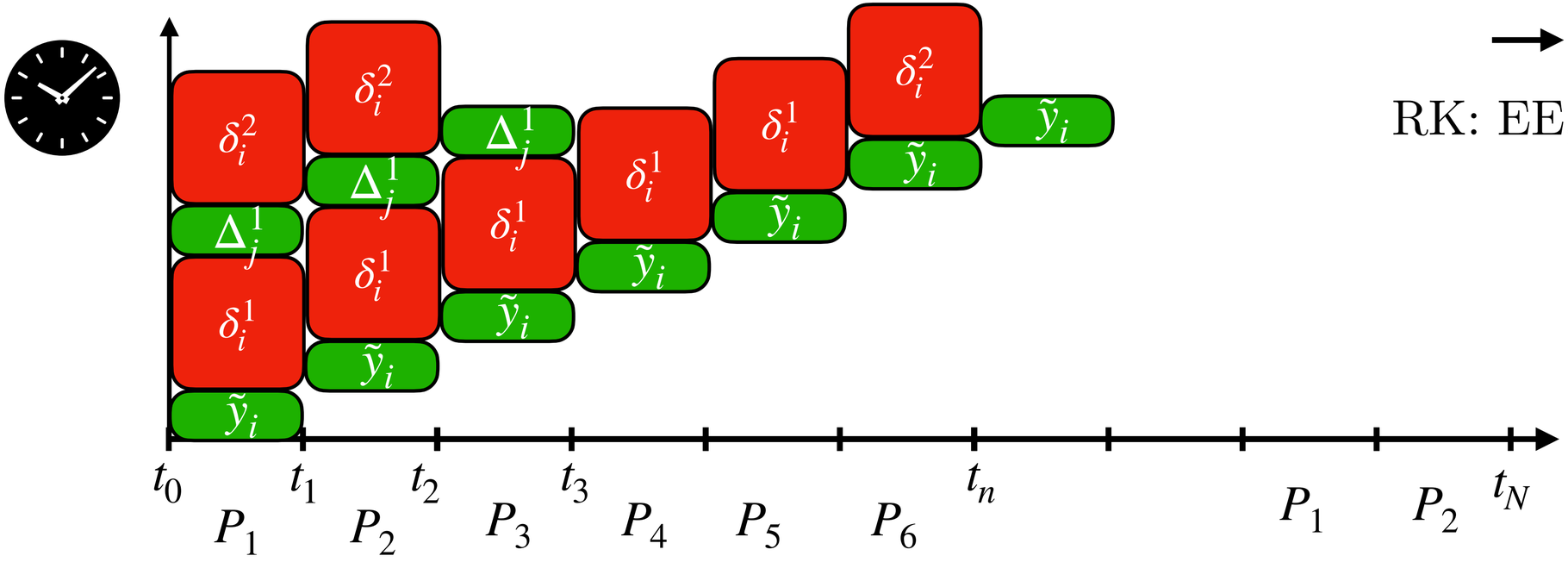
Parareal SDC:





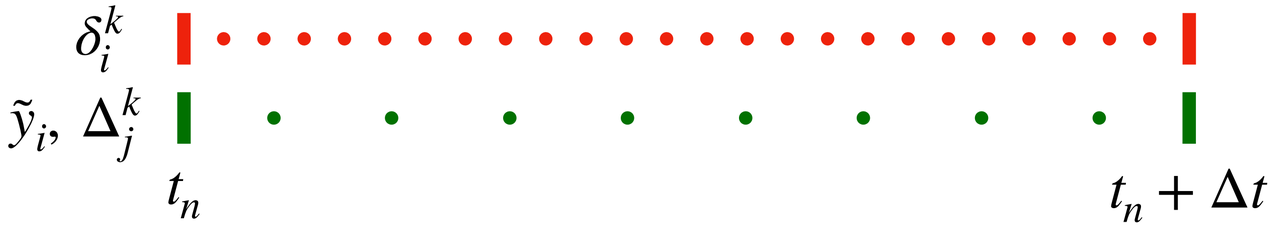
$\rightarrow$ = RK

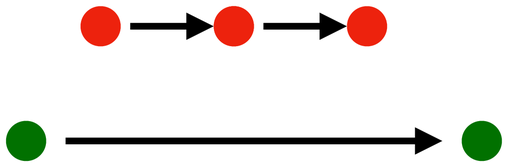
RK: EE, RK4, IE, IMEX,...



Parareal Spectral Deferred Correction method^2

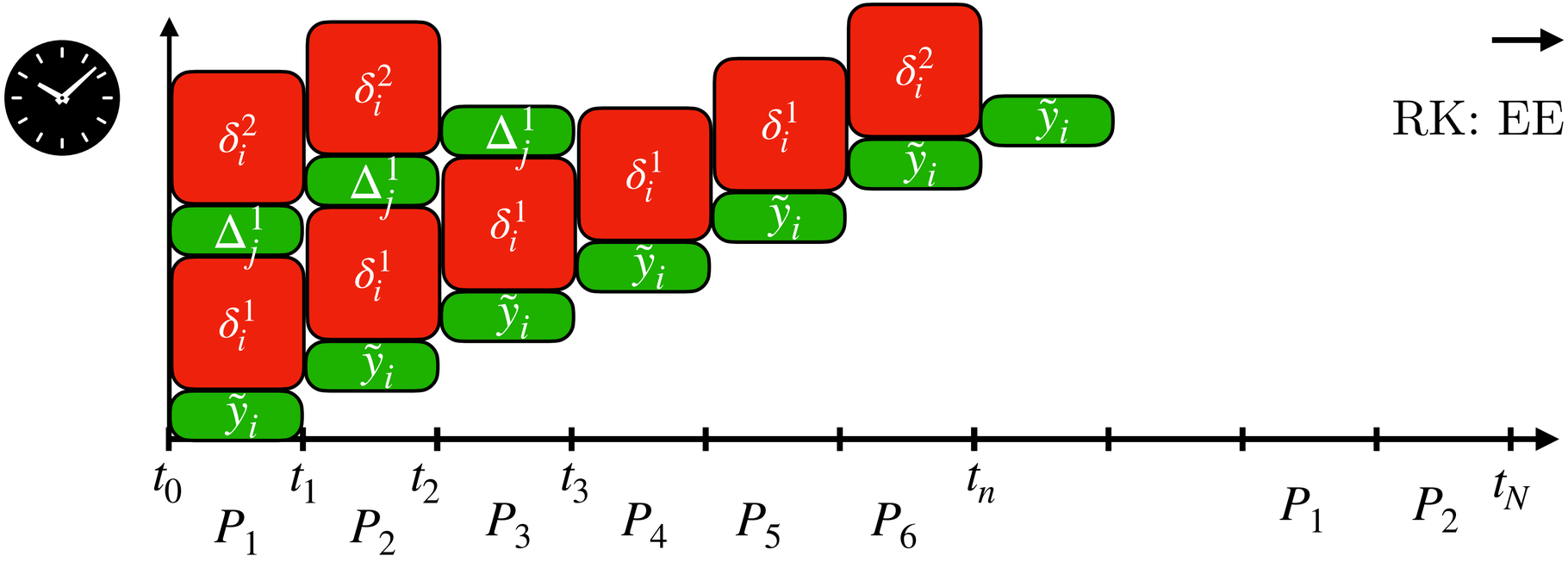
Parareal SDC:





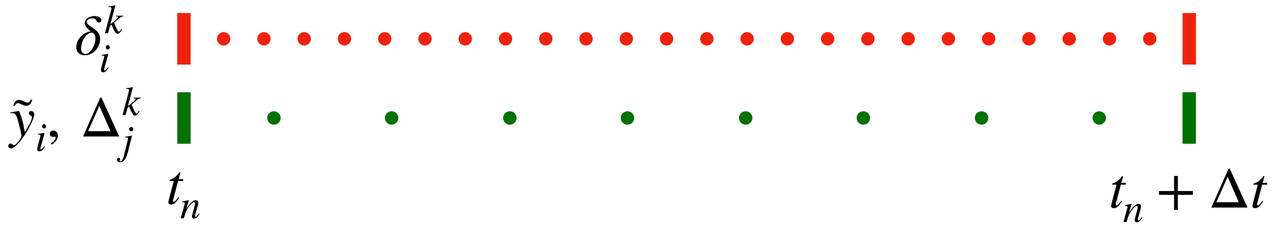
$\rightarrow$ = RK

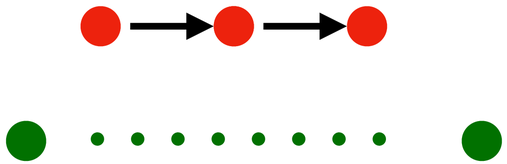
RK: EE, RK4, ~~IE, IMEX~~,...



Parareal Spectral Deferred Correction method^2

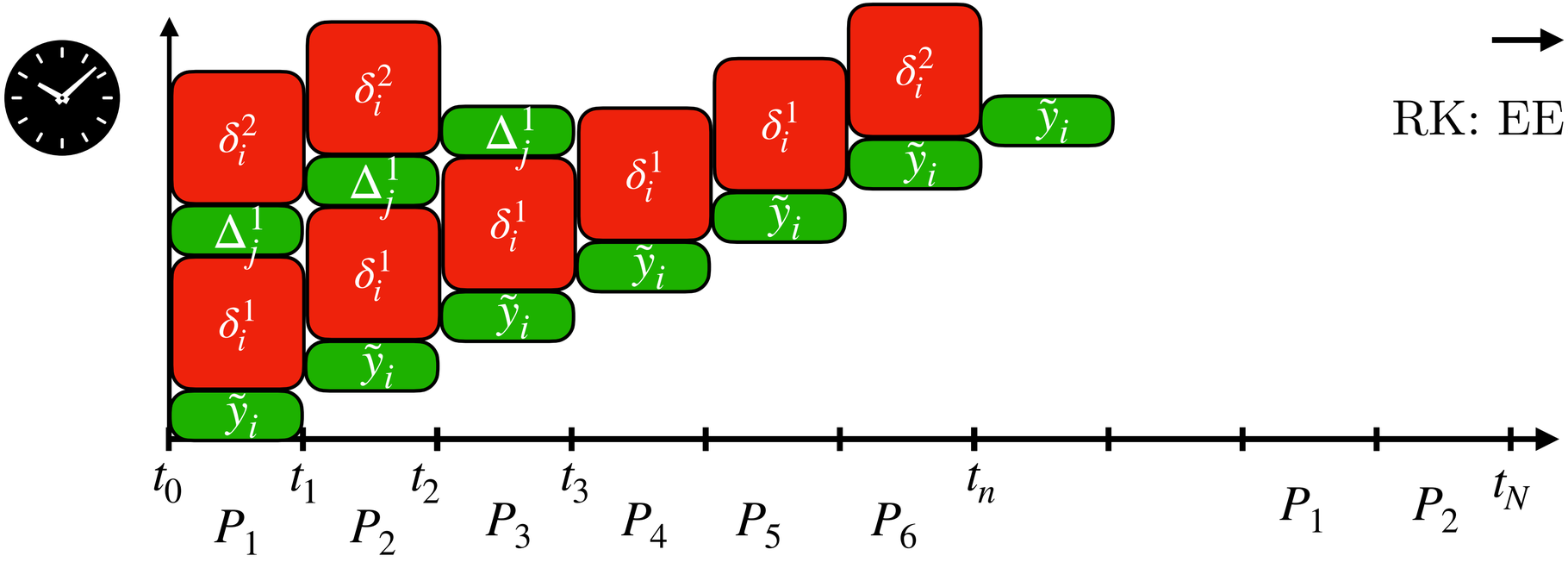
Parareal SDC:





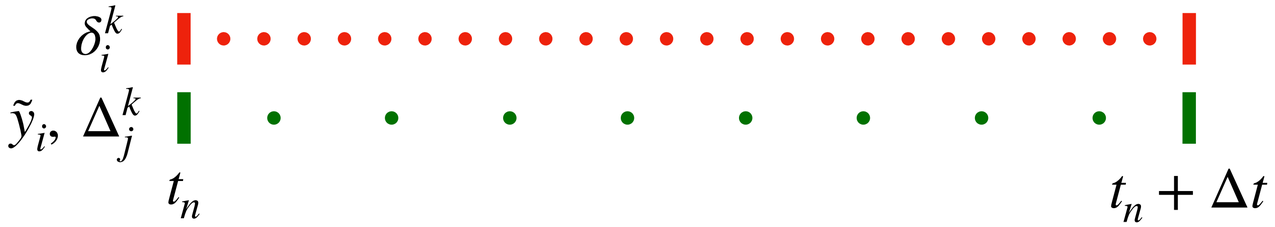
$\rightarrow$ = RK

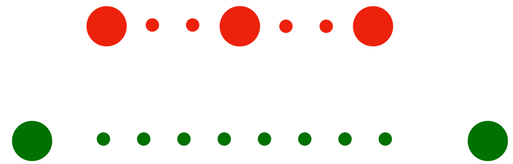
RK: EE, RK4, ~~IE, IMEX~~,...



Parareal Spectral Deferred Correction method^2

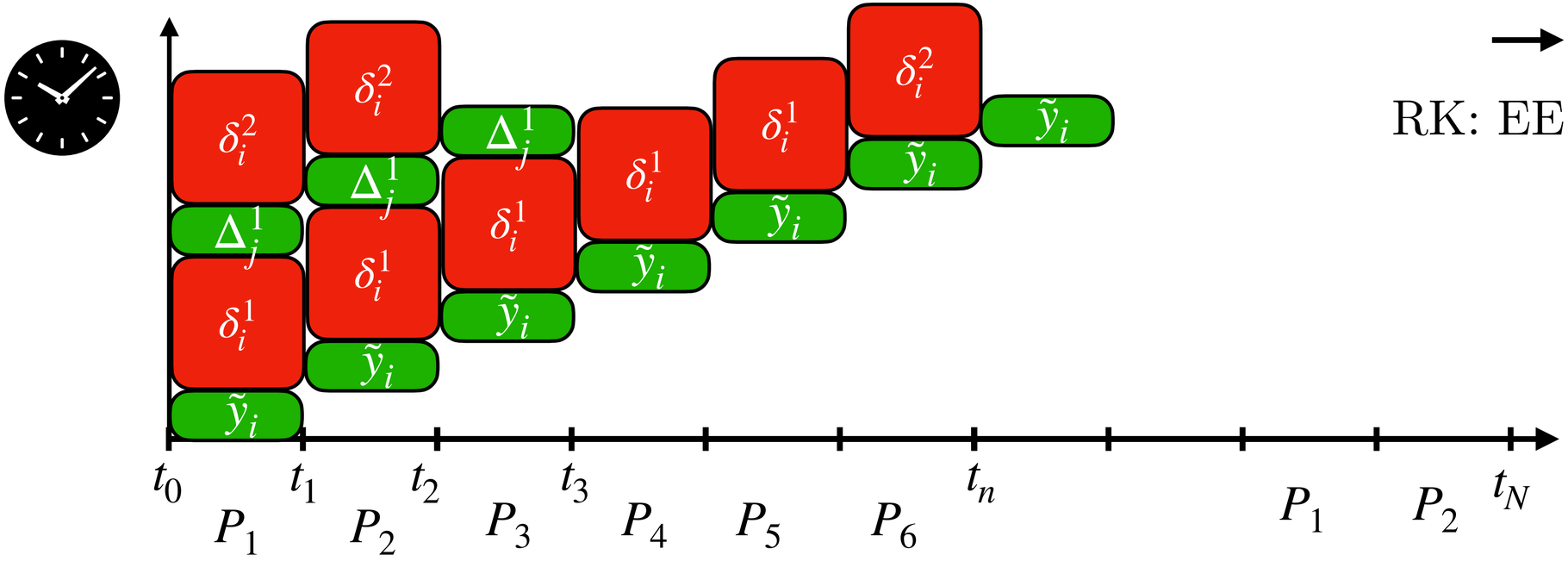
Parareal SDC:





$\longrightarrow$ = RK

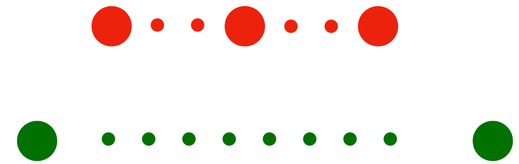
RK: EE, RK4, ~~IE, IMEX~~,...



Parareal Spectral Deferred Correction method^2

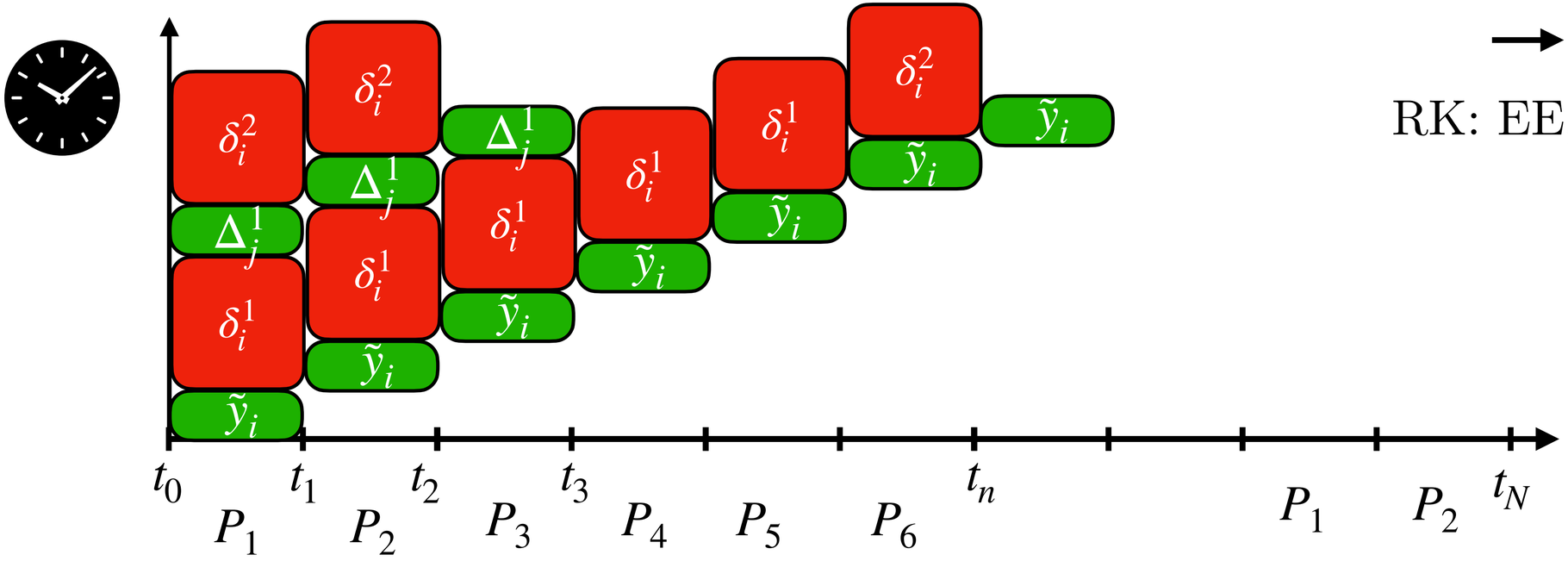
Parareal SDC:





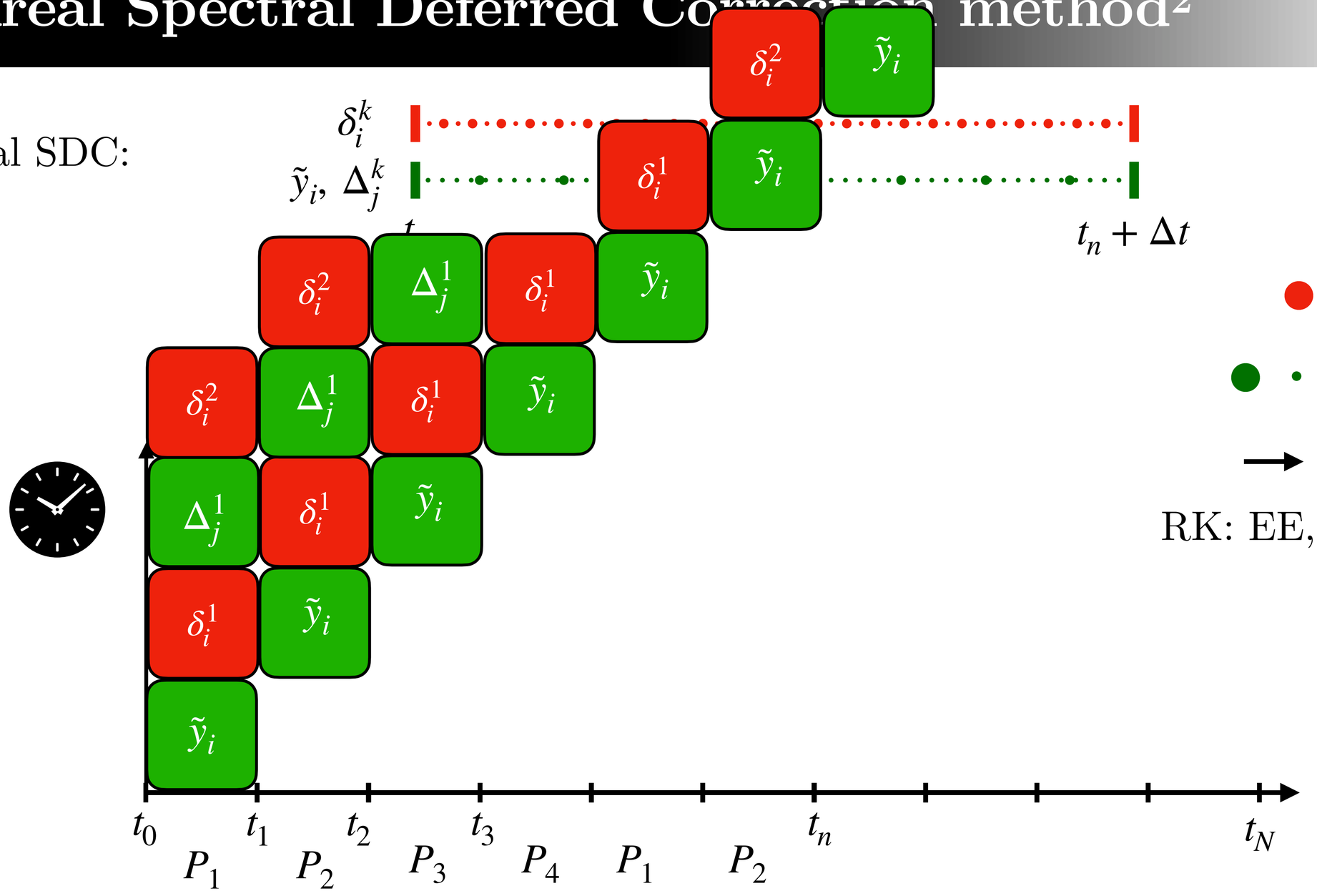
$\longrightarrow$ = RK

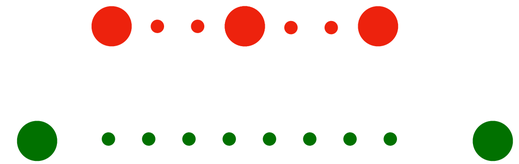
RK: EE, RK4, ~~IE, IMEX~~,...



Parareal Spectral Deferred Correction method^2

Parareal SDC:





$\longrightarrow$ = RK

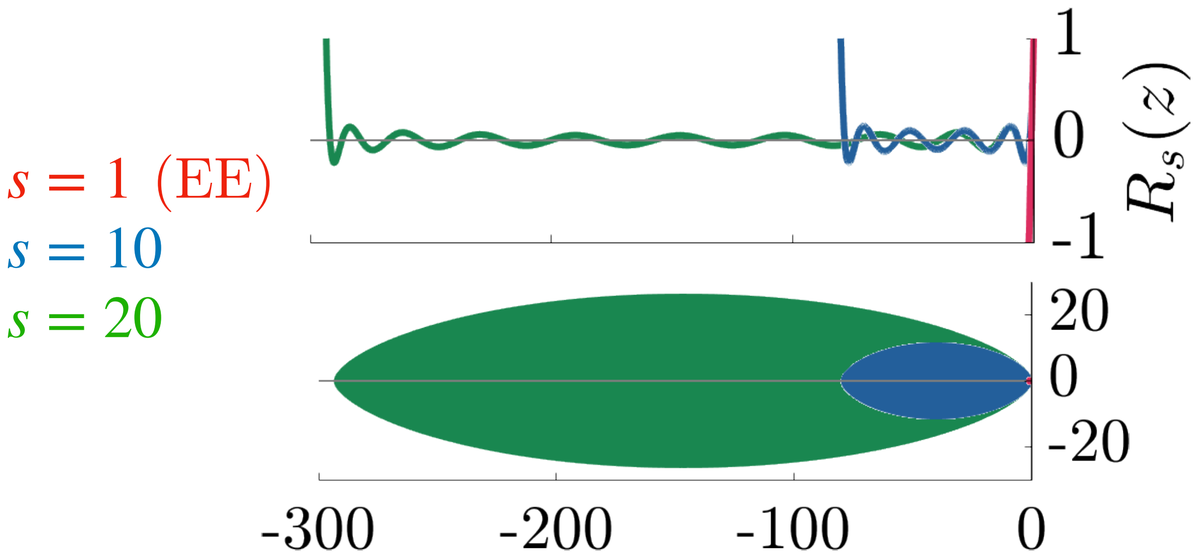
RK: EE, RK4, ~~IE, IMEX~~,...

The Runge-Kutta-Chebyshev methods[3]

One step of RKU for $y' = f(y)$ 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 for

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

  For multiscale ionic models or nonuniform grids, for instance.
- Works in mixed-precision arithmetic. Good for CPU, memory, and energy savings in HPC computations.
- All flavors are straightforward to implement.

[3] Van der Houwen, Sommeijer, Verwer, Lebedev, Abdulle, Medovikov, Vilmart, Rosilho, Almuslimani,...

SDC explicit stabilized method

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

Then iterate on:

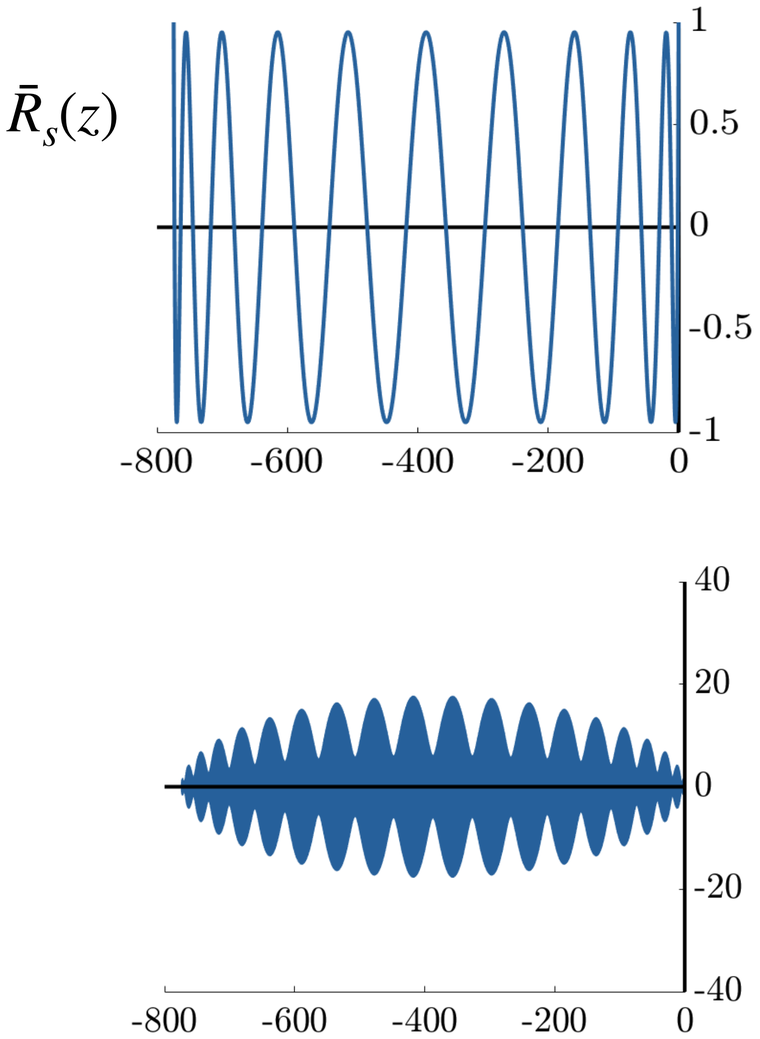
- Define $\tilde{y}(t) = \sum L_i(t)\tilde{y}_i$,
- Approximate $\varepsilon(t) \approx \sum L_i(t)\varepsilon(c_i)$. $\varepsilon(c_i)$ computed with quadrature rules.
- Compute $\delta_i - \varepsilon_i$ at $c_1, \ldots, c_s$ solving the error equation with RKU, HSRKU, mRKU,....

$$\delta(c_{i+1}) - \varepsilon(c_{i+1}) = \delta(c_i) - \varepsilon(c_i) + \int_{c_i}^{c_{i+1}} \left(f(\tilde{y}(s) + \delta(s)) - f(\tilde{y}(s)) \right) \mathrm{d}s,$$

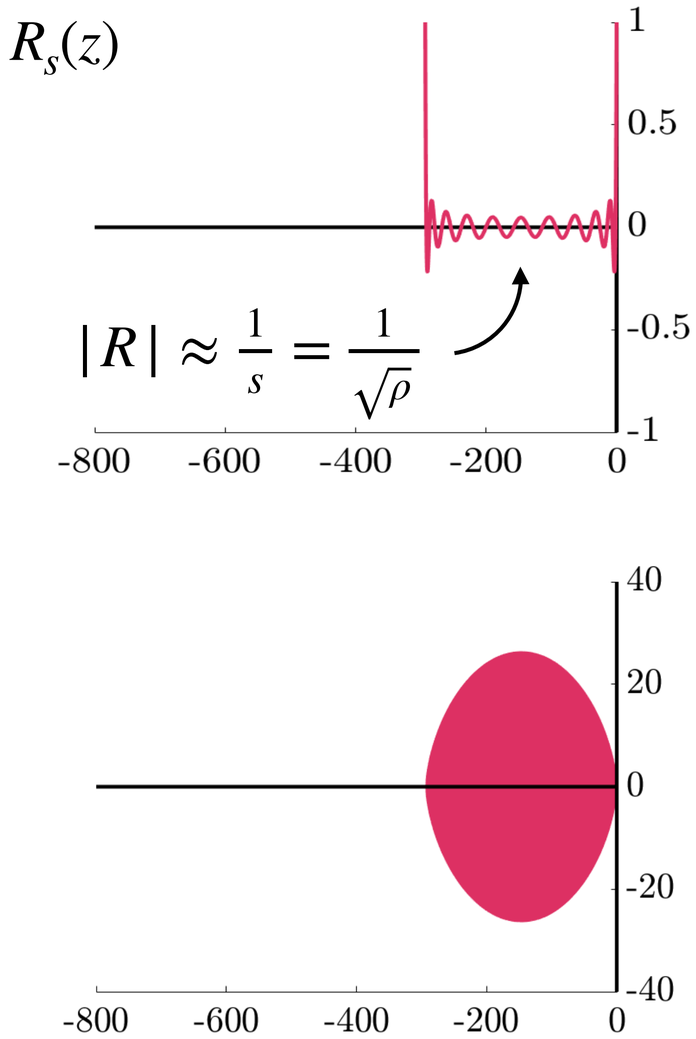
$$y(c_{i+1}) = \tilde{y}(c_{i+1}) + \delta(c_{i+1}) = \tilde{y}(c_{i+1}) + \varepsilon(c_{i+1}) + \left(\delta(c_{i+1}) - \varepsilon(c_{i+1}) \right).$$

Some explicit stabilized methods

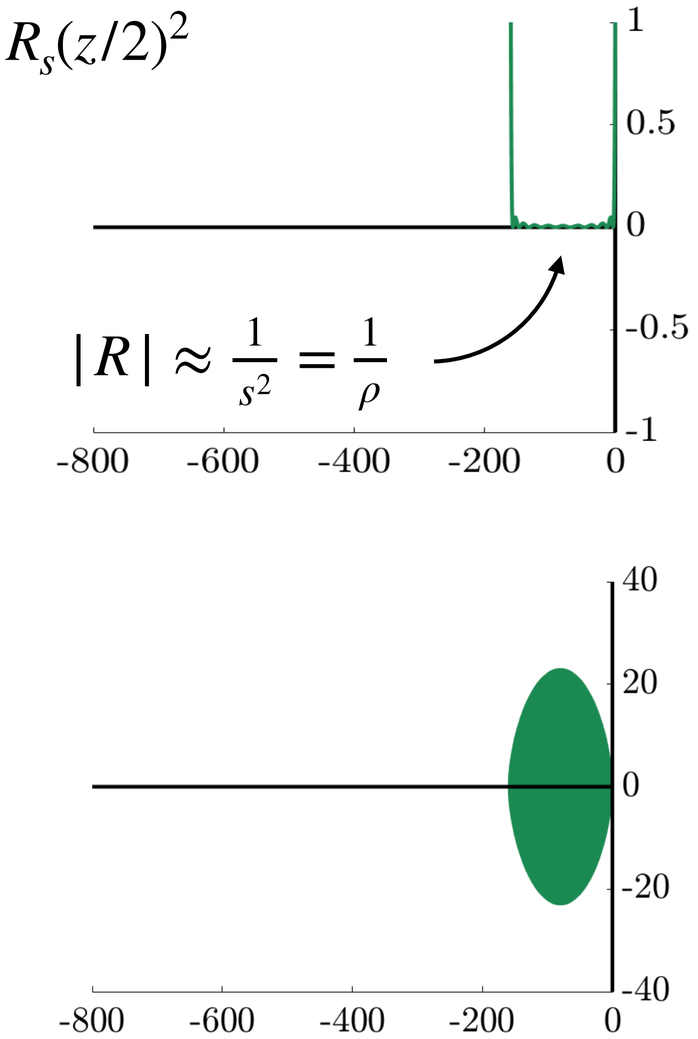
RKC $s = 20$



RKU $s = 20$



HSRKU $s = 10$ twice

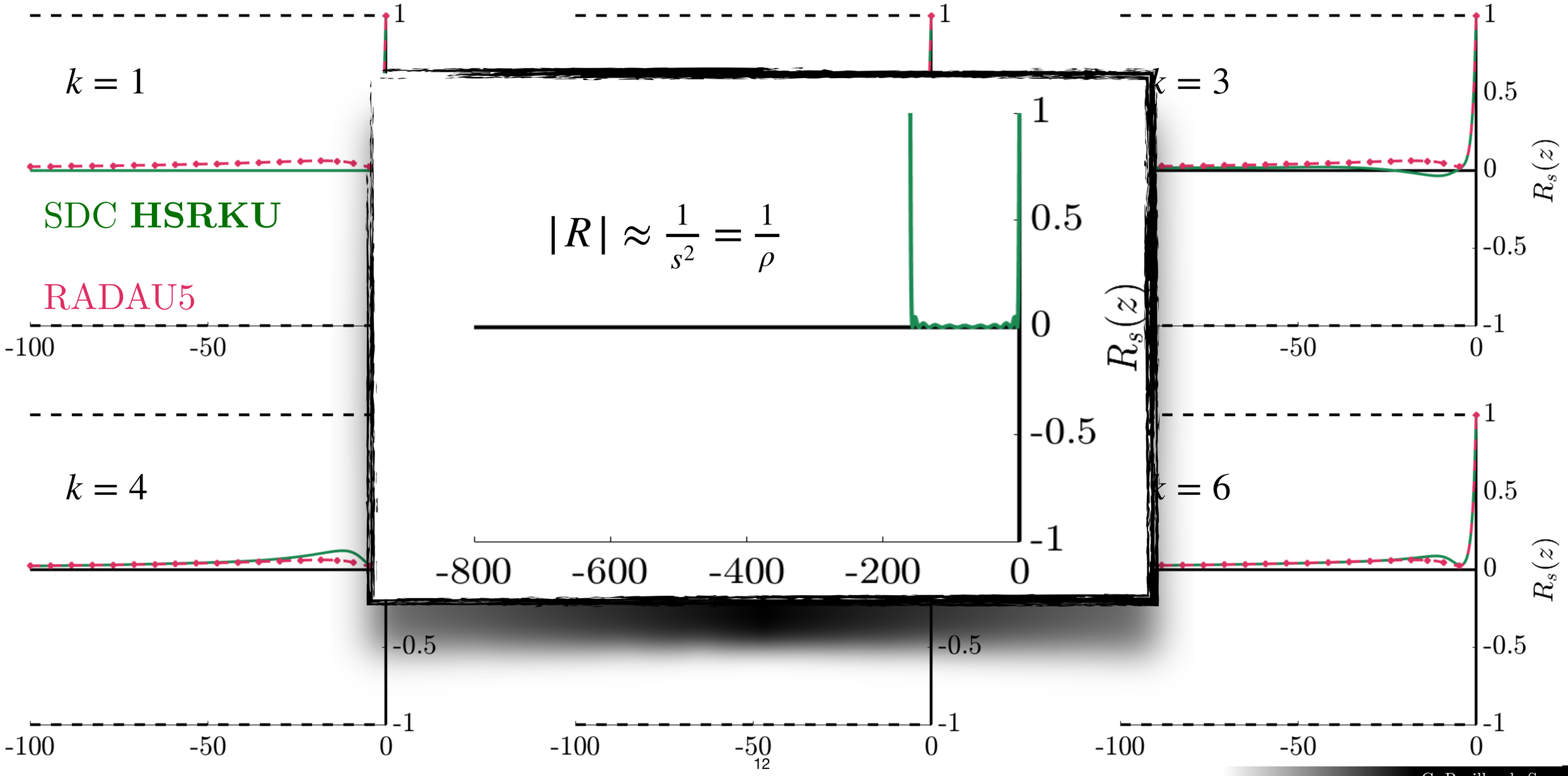


A numerical stability study

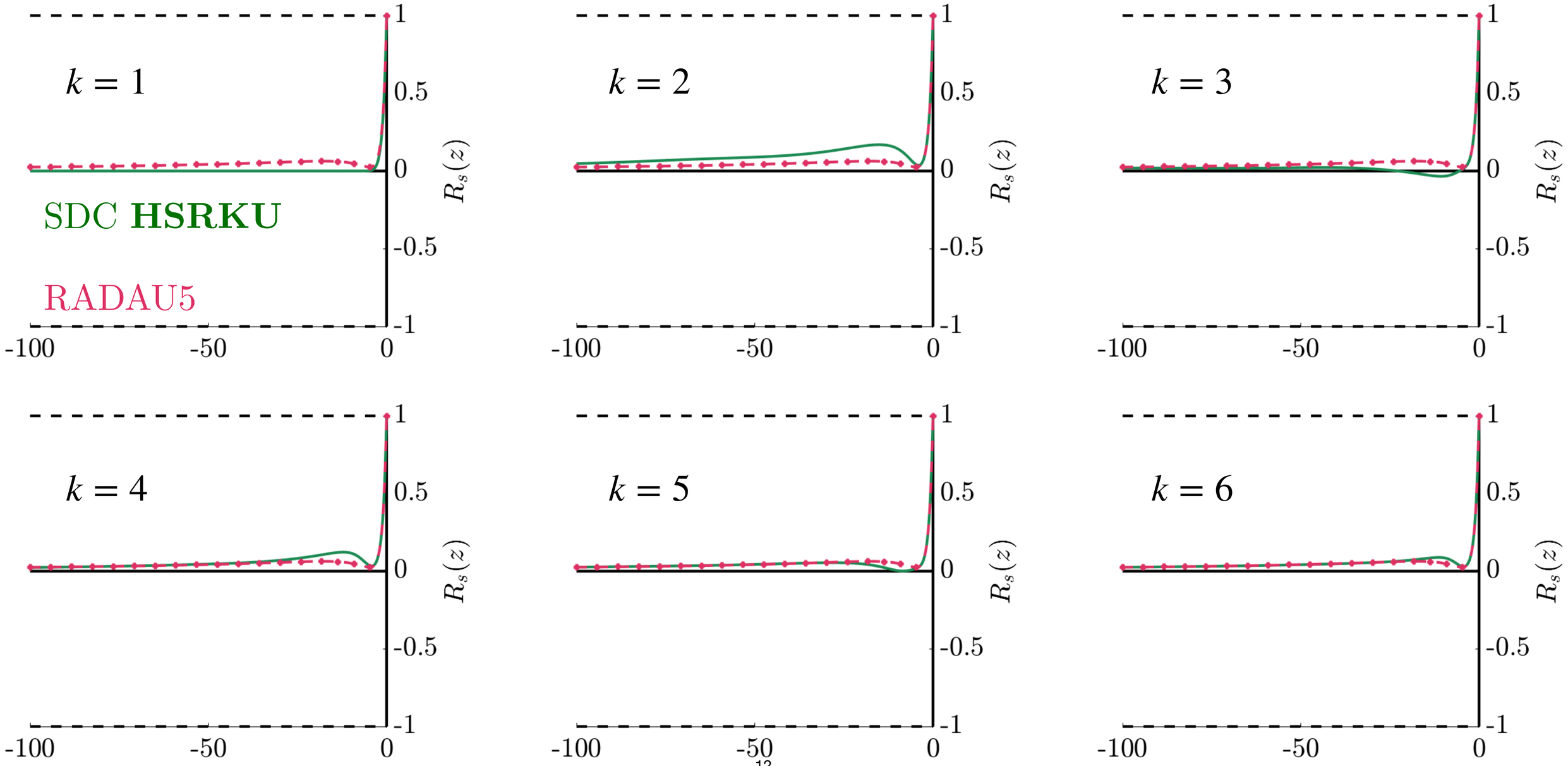
Compute (numerically) the stability polynomial of SDC using stabilized methods

- Consider Dahlquist test equation $y' = \lambda y$,
- And the serial SDC method (not hybrid parareal SDC),
- Fix 3 Radau collocation points c_1, c_2, c_3 (underlying collocation method is RADAU5),
- Solve $y' = \lambda y$ with SDC for different λ, obtaining the stability polynomial,
- We do it using the three explicit stabilized methods RKC, RKU, HSRKU,
- And different iteration numbers.

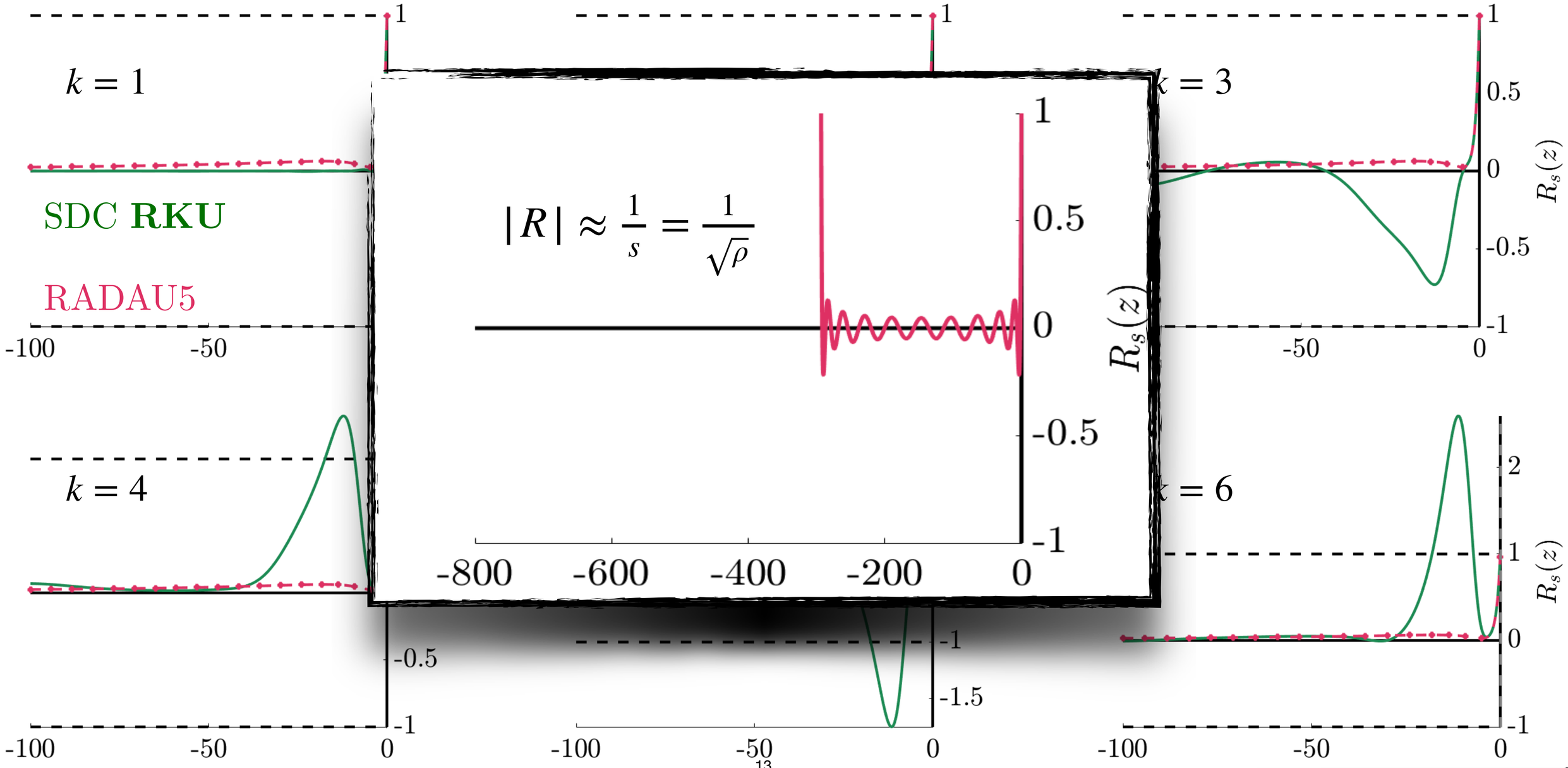
A numerical stability study: HSRKU



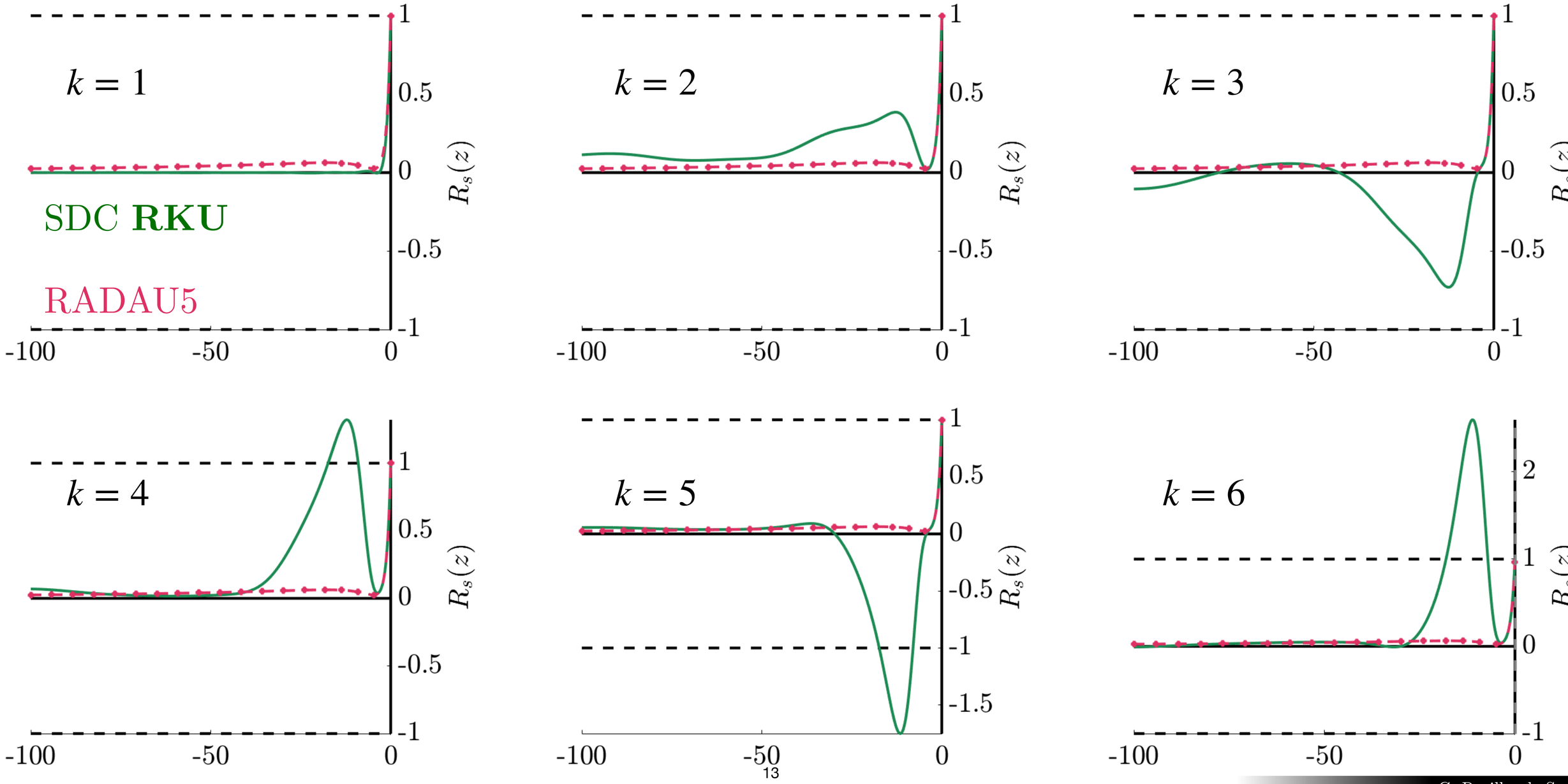
A numerical stability study: HSRKU



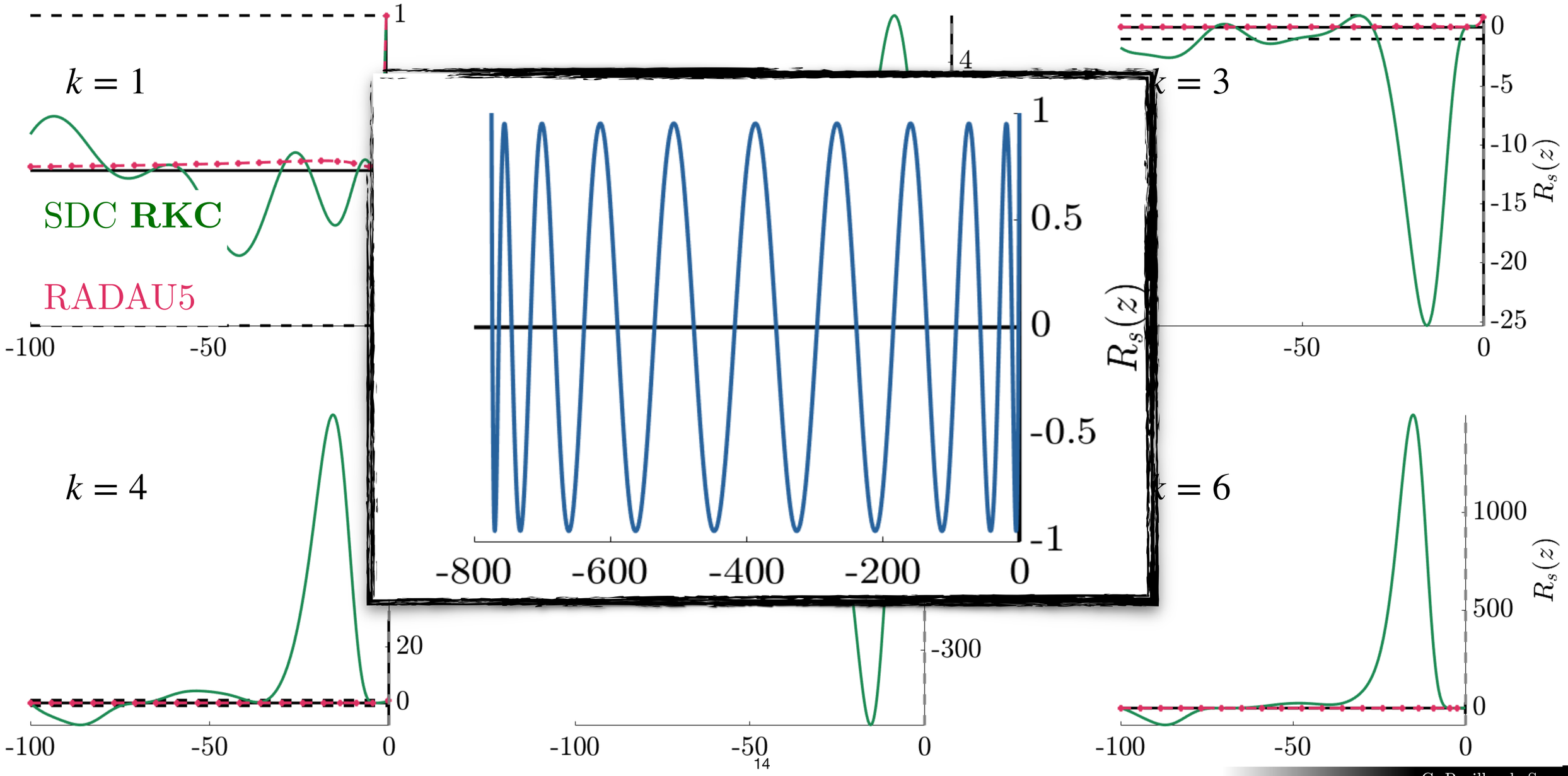
A numerical stability study: RKU



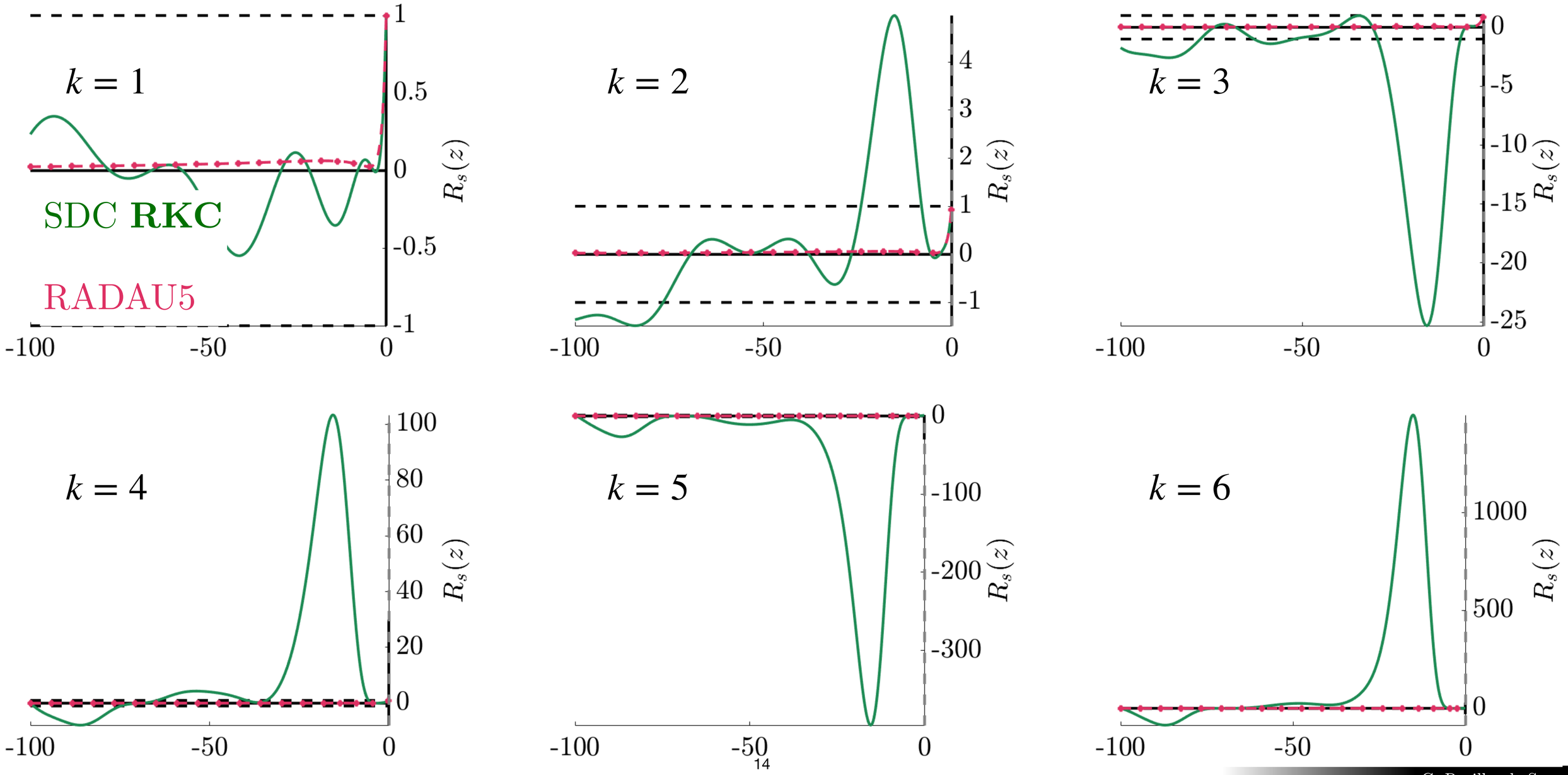
A numerical stability study: RKU



A numerical stability study: RKC



A numerical stability study: RKC



A numerical stability study

Why HSRKU works and RKU, RKC not? In SDC at some point we compute

$$\varepsilon(c_i) = y_0 + \int_{t_n}^{c_i} f(\tilde{y}(s))\mathrm{d}s - \tilde{y}_i \approx y_0 + \Delta t \sum_{j=1}^{s} a_{ij} f(\tilde{y}_j) - \tilde{y}_i$$

$$= y_0 + \sum_{j=1}^{s} a_{ij}\ z\ \tilde{y}_j - \tilde{y}_i = y_0 + \sum_{j=1}^{s} a_{ij}\ zR_j(z)\ \tilde{y}_{j-1} - \tilde{y}_i$$

A numerical stability study

Why HSRKU works and RKU, RKC not? In SDC at some point we compute

$$\varepsilon(c_i) = y_0 + \int_{t_n}^{c_i} f(\tilde{y}(s))\mathrm{d}s - \tilde{y}_i \approx y_0 + \Delta t \sum_{j=1}^{s} a_{ij} f(\tilde{y}_j) - \tilde{y}_i$$

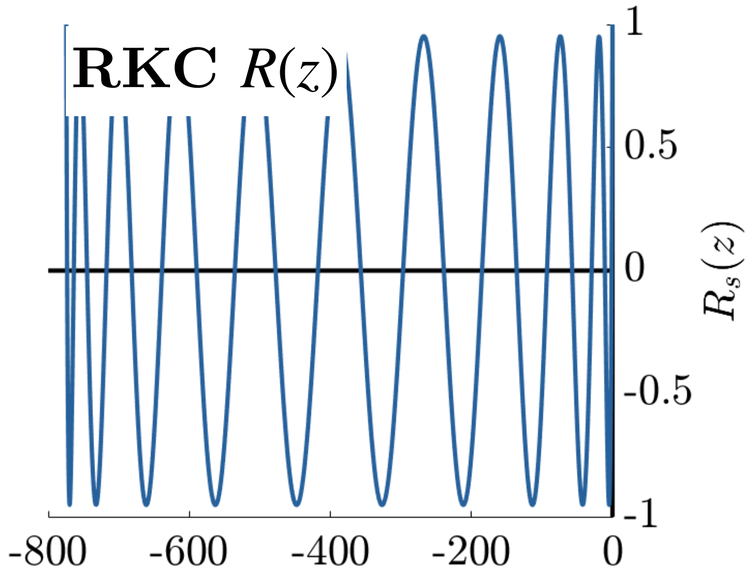
$$= y_0 + \sum_{j=1}^{s} a_{ij}\ z\ \tilde{y}_j - \tilde{y}_i = y_0 + \sum_{j=1}^{s} a_{ij}\ zR_j(z)\ \tilde{y}_{j-1} - \tilde{y}_i$$

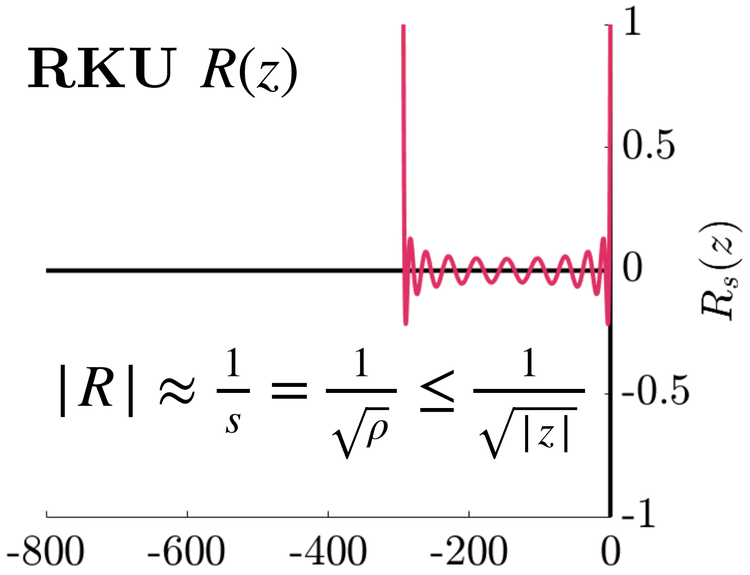
A numerical stability study

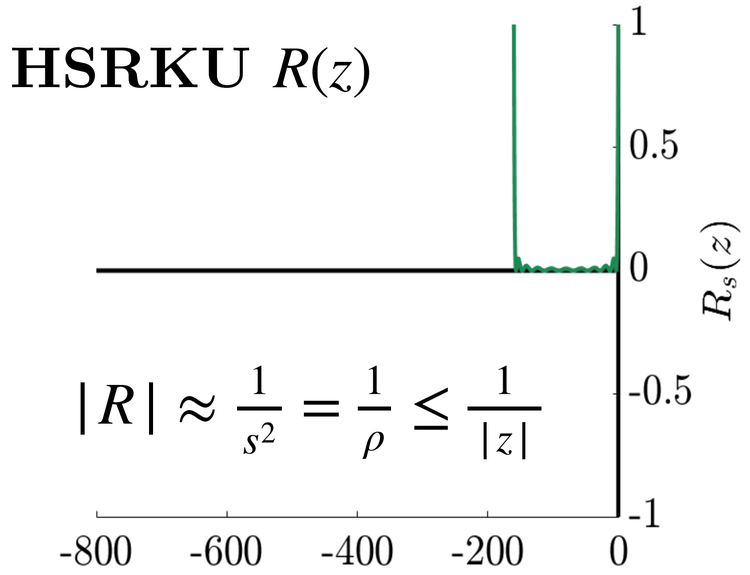
Why HSRKU works and RKU, RKC not? In SDC at some point we compute

$$\varepsilon(c_i) = y_0 + \int_{t_n}^{c_i} f(\tilde{y}(s))\mathrm{d}s - \tilde{y}_i \approx y_0 + \Delta t \sum_{j=1}^{s} a_{ij} f(\tilde{y}_j) - \tilde{y}_i$$

$$= y_0 + \sum_{j=1}^{s} a_{ij}\, z\, \tilde{y}_j - \tilde{y}_i = y_0 + \sum_{j=1}^{s} a_{ij}\, zR_j(z)\, \tilde{y}_{j-1} - \tilde{y}_i$$





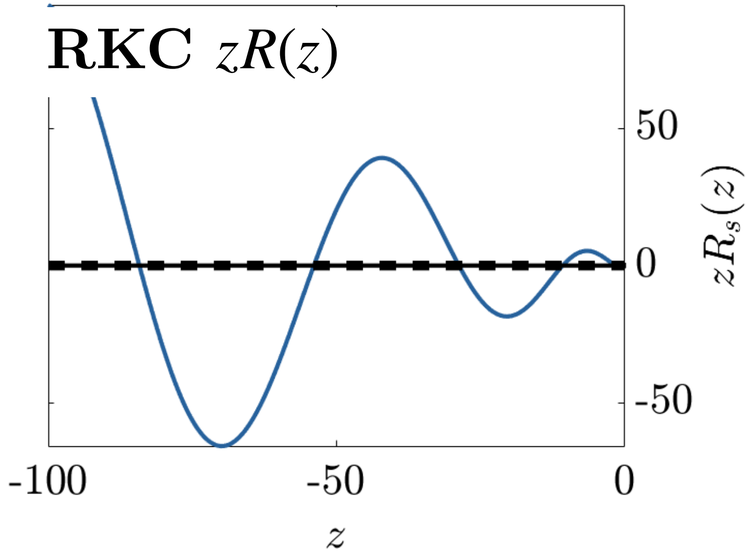


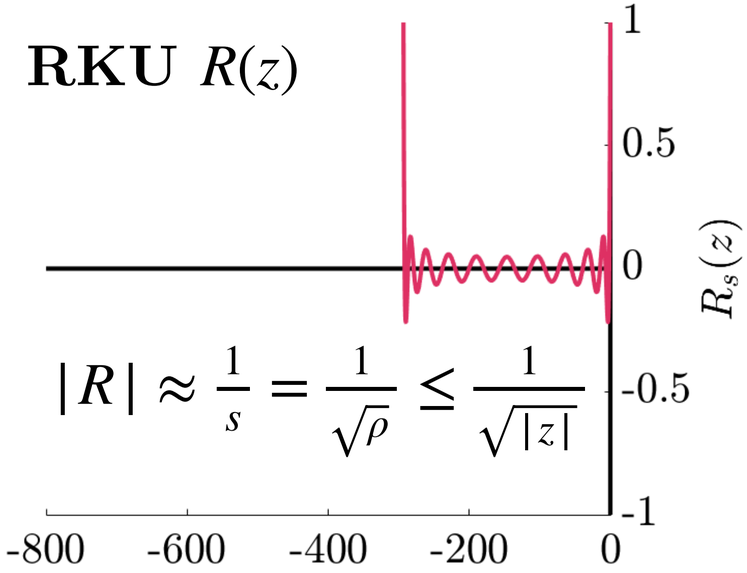
A numerical stability study

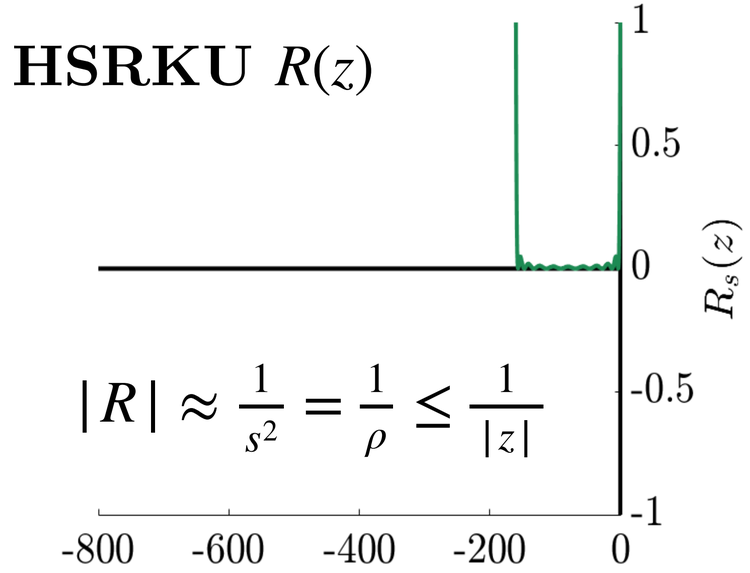
Why HSRKU works and RKU, RKC not? In SDC at some point we compute

$$\varepsilon(c_i) = y_0 + \int_{t_n}^{c_i} f(\tilde{y}(s))\mathrm{d}s - \tilde{y}_i \approx y_0 + \Delta t \sum_{j=1}^{s} a_{ij} f(\tilde{y}_j) - \tilde{y}_i$$

$$= y_0 + \sum_{j=1}^{s} a_{ij}\ z\ \tilde{y}_j - \tilde{y}_i = y_0 + \sum_{j=1}^{s} a_{ij}\ zR_j(z)\ \tilde{y}_{j-1} - \tilde{y}_i$$





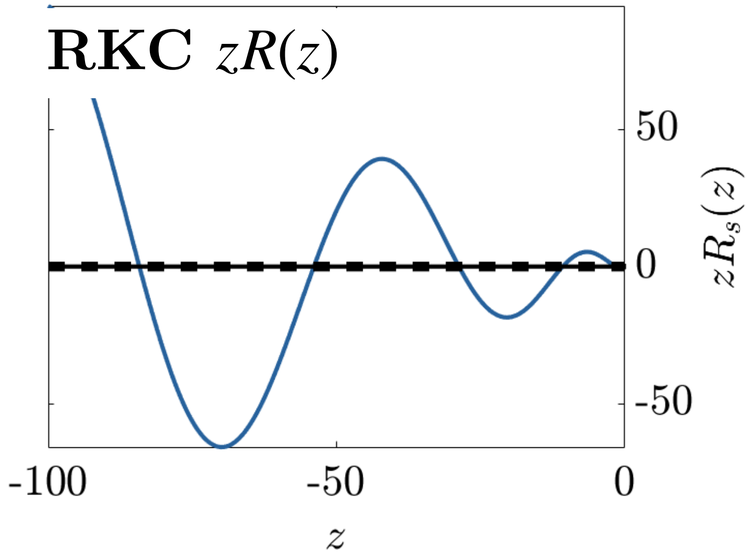


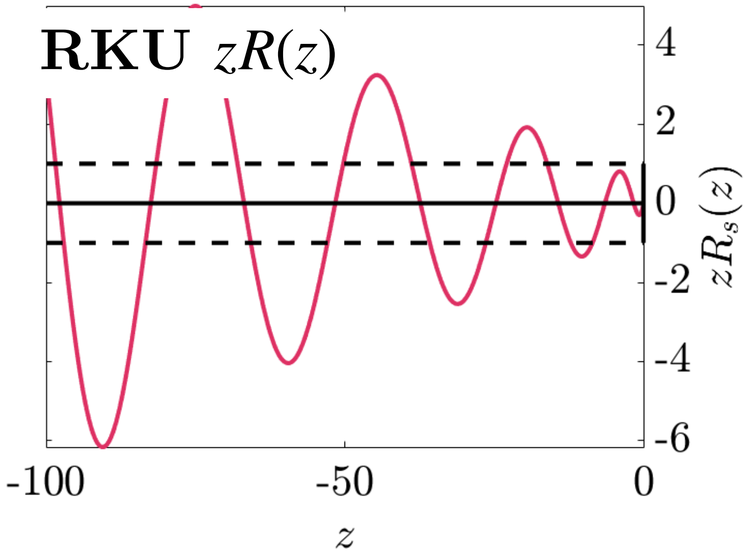
A numerical stability study

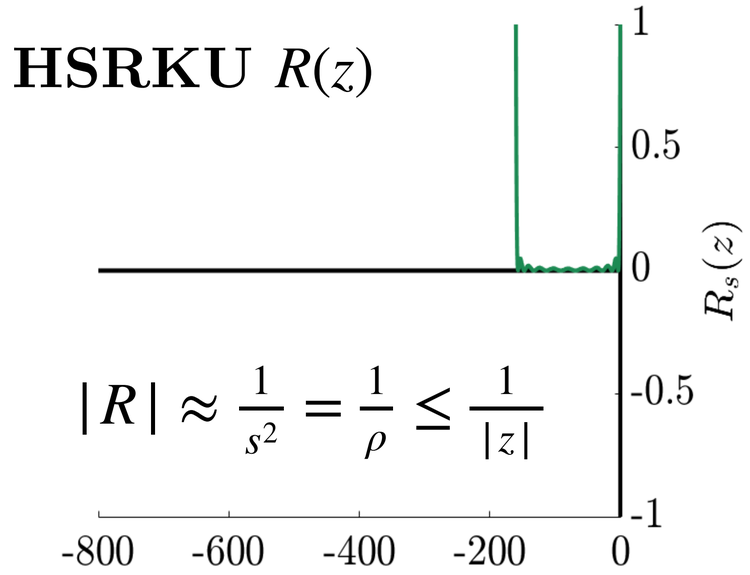
Why HSRKU works and RKU, RKC not? In SDC at some point we compute

$$\varepsilon(c_i) = y_0 + \int_{t_n}^{c_i} f(\tilde{y}(s))\mathrm{d}s - \tilde{y}_i \approx y_0 + \Delta t \sum_{j=1}^{s} a_{ij} f(\tilde{y}_j) - \tilde{y}_i$$

$$= y_0 + \sum_{j=1}^{s} a_{ij}\, z\, \tilde{y}_j - \tilde{y}_i = y_0 + \sum_{j=1}^{s} a_{ij}\, zR_j(z)\, \tilde{y}_{j-1} - \tilde{y}_i$$





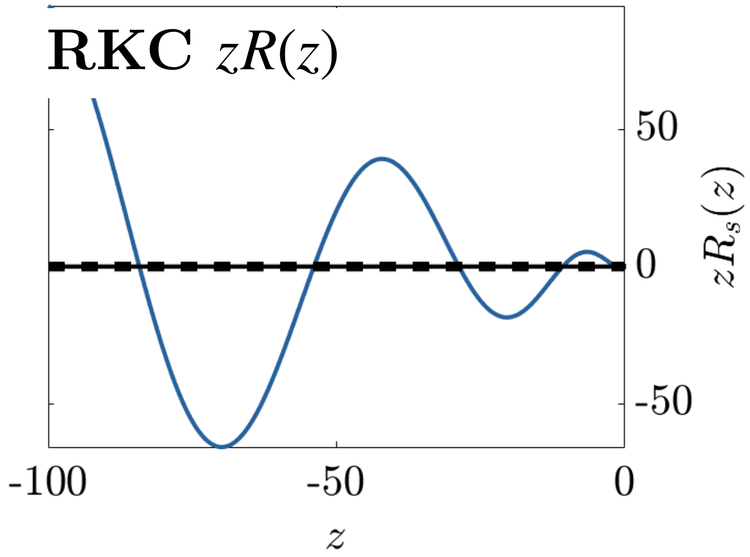


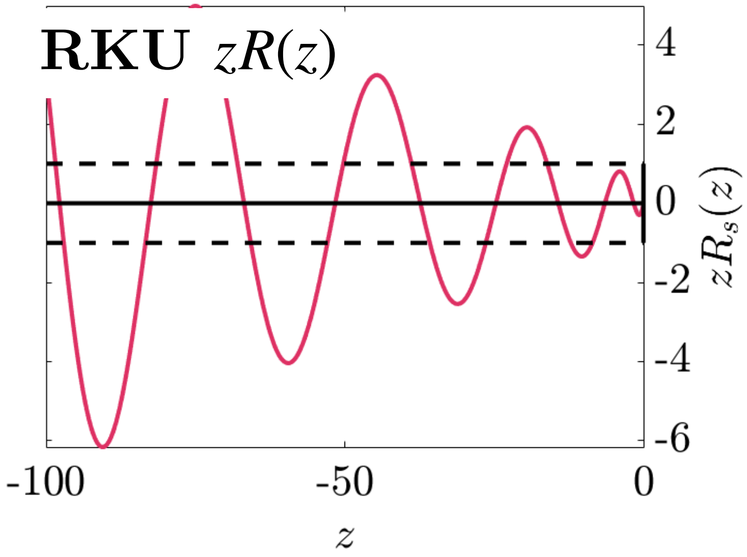
A numerical stability study

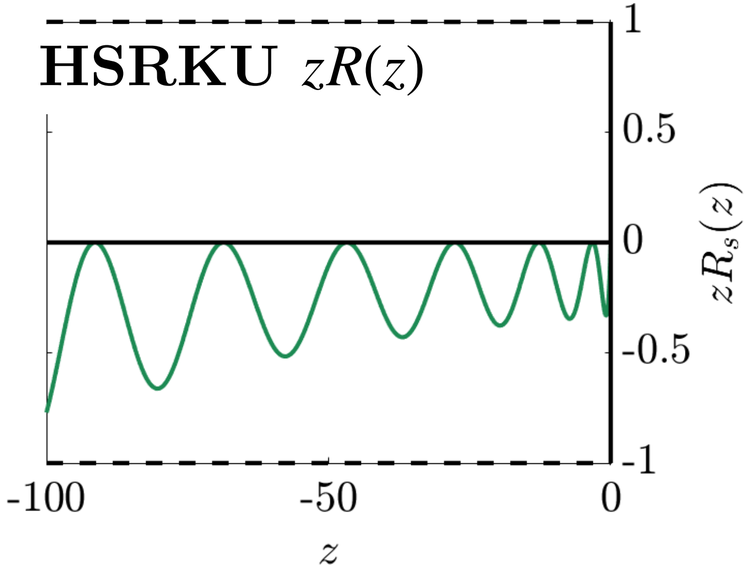
Why HSRKU works and RKU, RKC not? In SDC at some point we compute

$$\varepsilon(c_i) = y_0 + \int_{t_n}^{c_i} f(\tilde{y}(s))\mathrm{d}s - \tilde{y}_i \approx y_0 + \Delta t \sum_{j=1}^{s} a_{ij} f(\tilde{y}_j) - \tilde{y}_i$$

$$= y_0 + \sum_{j=1}^{s} a_{ij}\, z\, \tilde{y}_j - \tilde{y}_i = y_0 + \sum_{j=1}^{s} a_{ij}\, zR_j(z)\, \tilde{y}_{j-1} - \tilde{y}_i$$





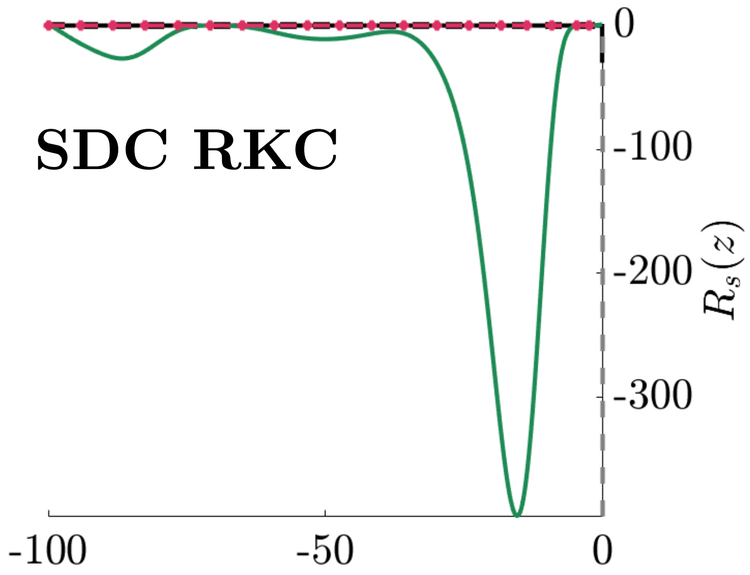


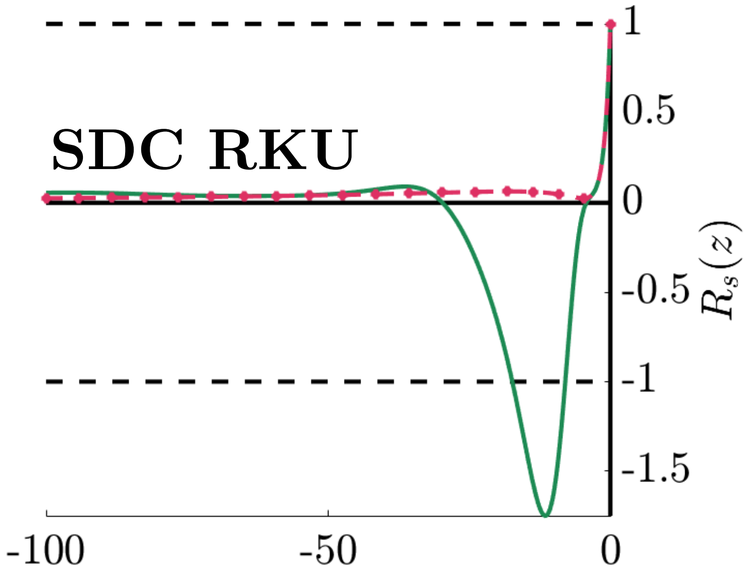
A numerical stability study

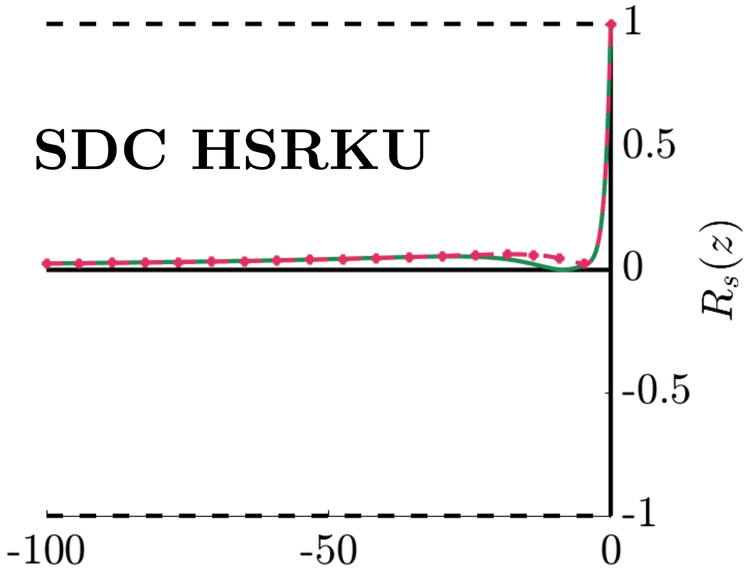
Why HSRKU works and RKU, RKC not? In SDC at some point we compute

$$\varepsilon(c_i) = y_0 + \int_{t_n}^{c_i} f(\tilde{y}(s))\mathrm{d}s - \tilde{y}_i \approx y_0 + \Delta t \sum_{j=1}^{s} a_{ij} f(\tilde{y}_j) - \tilde{y}_i$$

$$= y_0 + \sum_{j=1}^{s} a_{ij}\, z\, \tilde{y}_j - \tilde{y}_i = y_0 + \sum_{j=1}^{s} a_{ij}\, zR_j(z)\, \tilde{y}_{j-1} - \tilde{y}_i$$





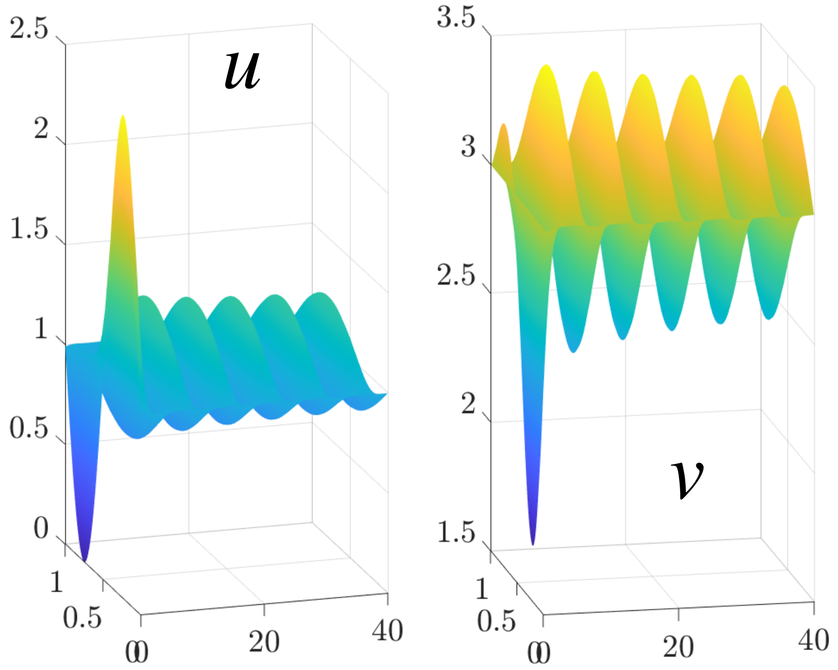


Numerical Experiment on Brusselator problem

Consider $\Omega = [0,1]$, $T = 40$ and

$$\partial_t u = \nu \Delta u + 1 + u^2 v - 4u, \qquad \text{in } \Omega \times [0,T]$$
$$\partial_t v = \nu \Delta v + 3u - u^2 v, \qquad \text{in } \Omega \times [0,T]$$

With Dirichlet and $\nu = 1/20$.

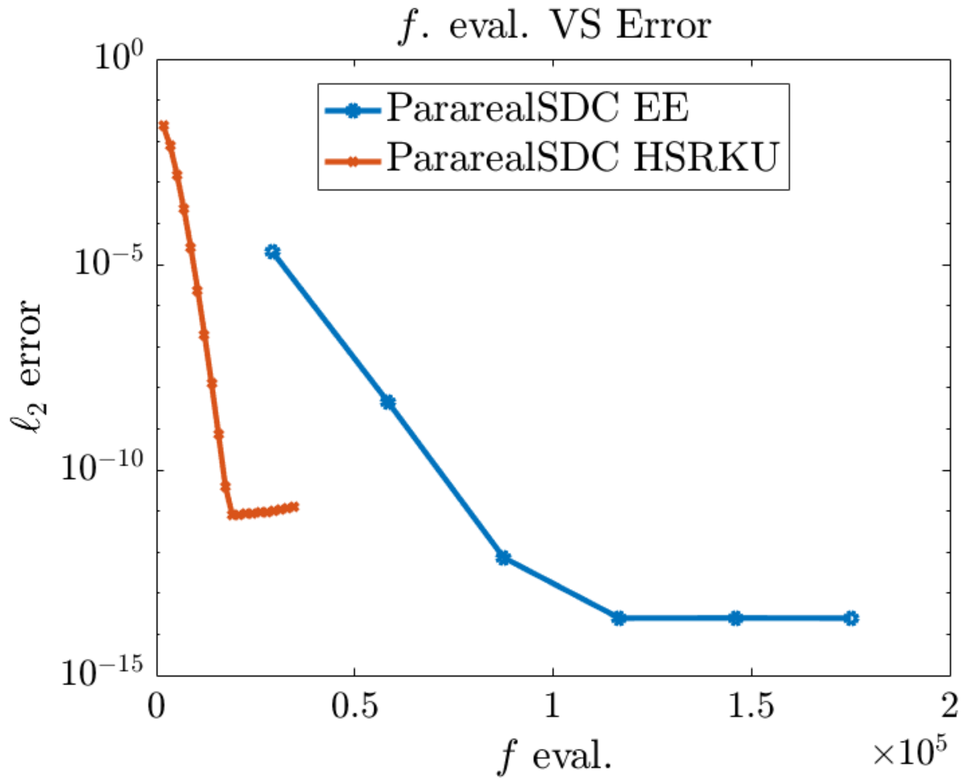


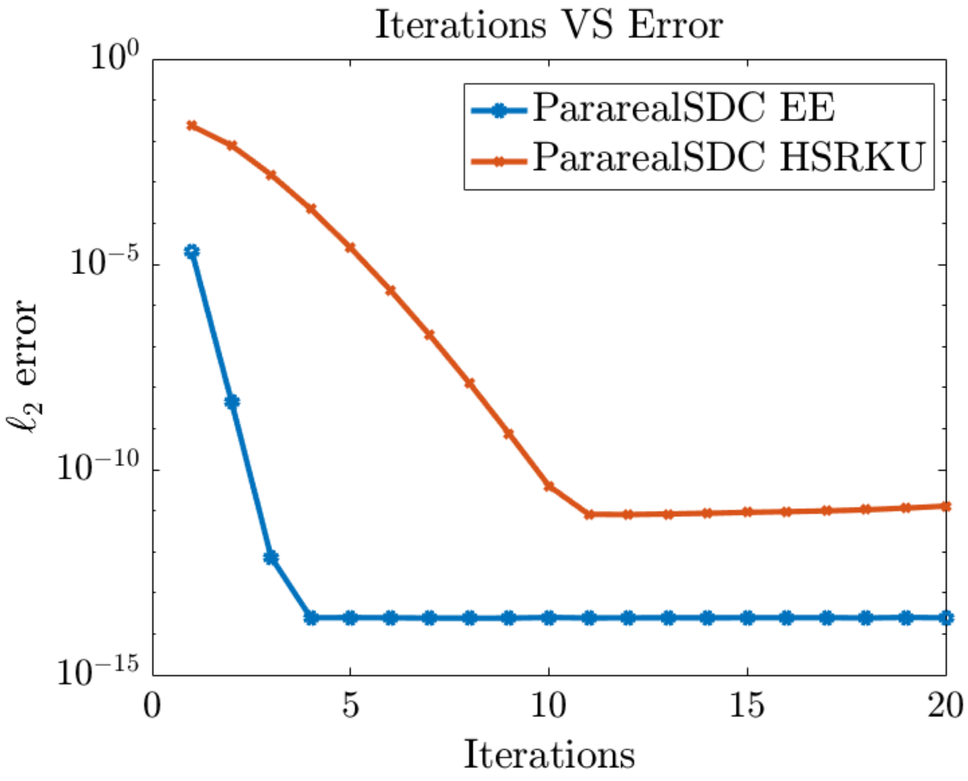
- Discretize with finite differences $h = 1/500$.
- Solve with Parareal SDC using EE, HSRKU, mHSRKU.

We use 80 subintervals (cores) with

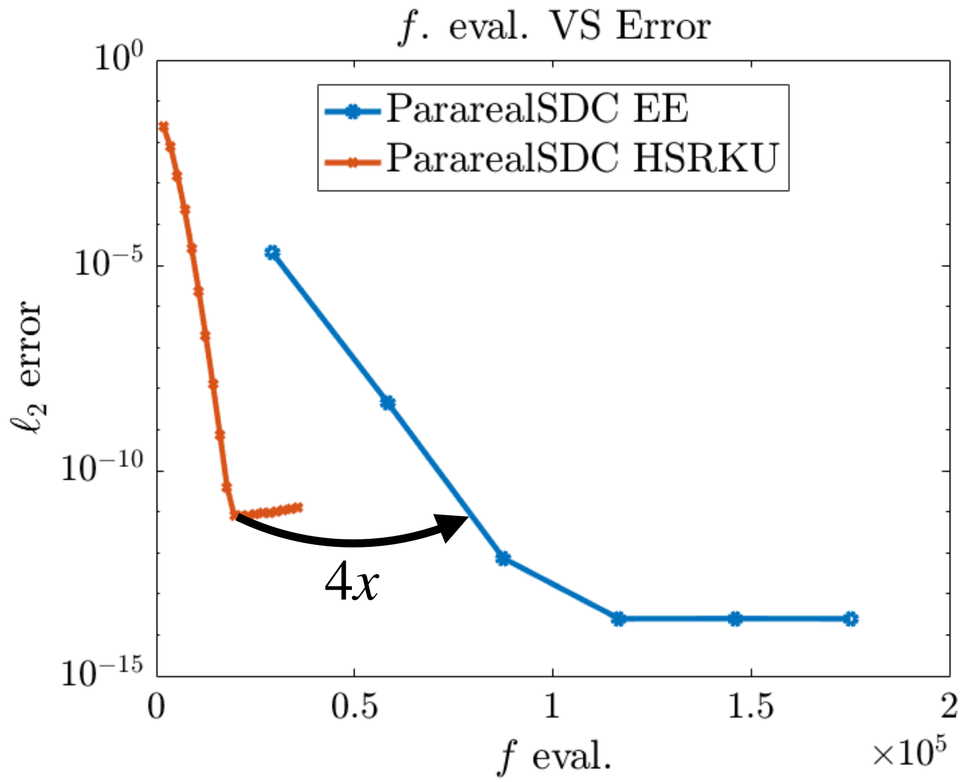
- 3 Radau collocation nodes on coarse grid,
- 20 Radau collocation nodes on fine grid.

Numerical Experiment on Brusselator problem





Numerical Experiment on Brusselator problem



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

On coarse grid • • •

Cost EE: $\approx 1.5e4$

Cost HSRKU: $\approx 5e2$ $(x30)$

On fine grid ·····

Cost EE: $\approx 1.5e4$

Cost HSRKU: $\approx 1.2e3$ $(x12)$

Multirate stabilized 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

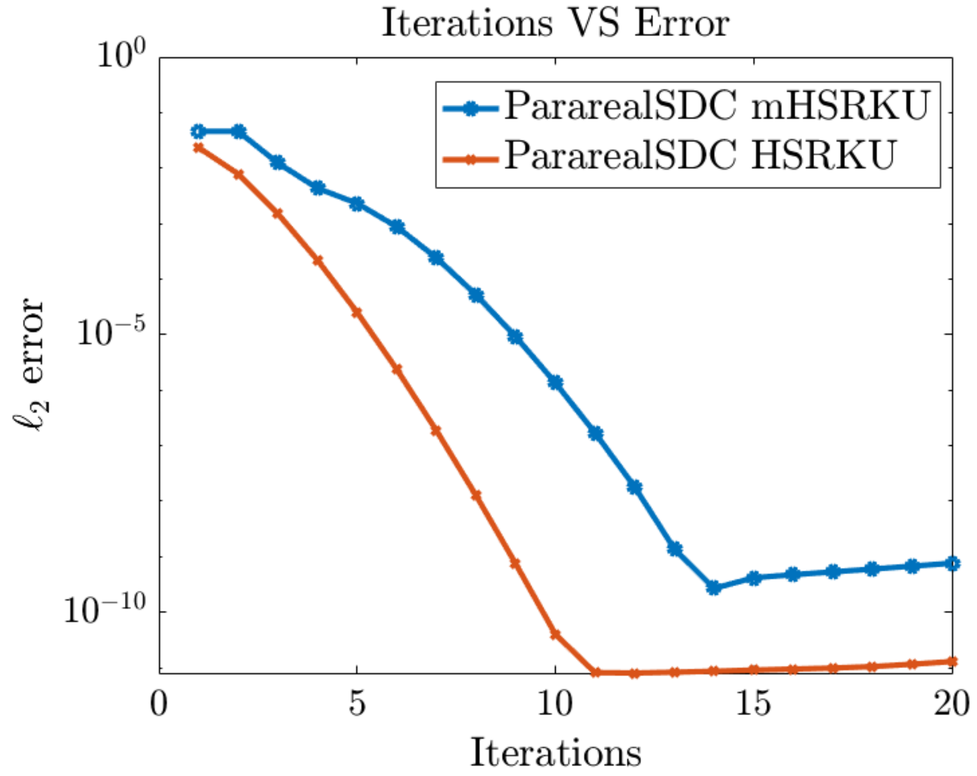
$$u' = f_F(u) + f_S(y), \qquad u(0) = y\,.$$

The multirate RKU method is given by:

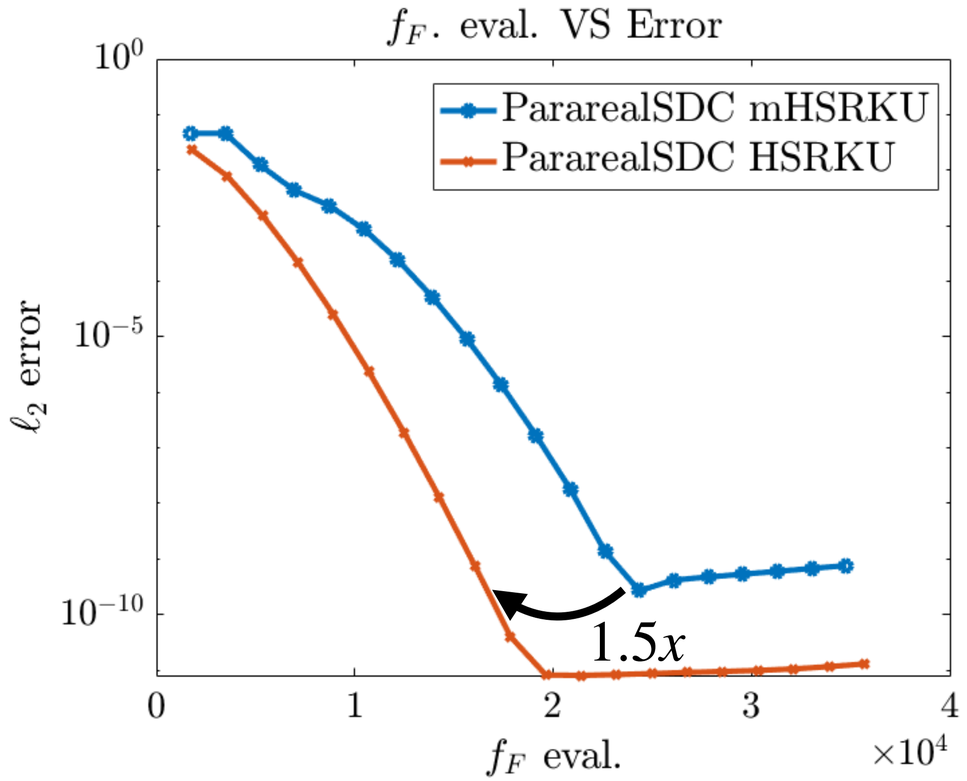
- Integrate $y'_\eta = f_\eta(y_\eta)$ with a stabilized method.
- To evaluate f_η solve $u' = f_F(u) + f_S(y)$ with another stabilized method.

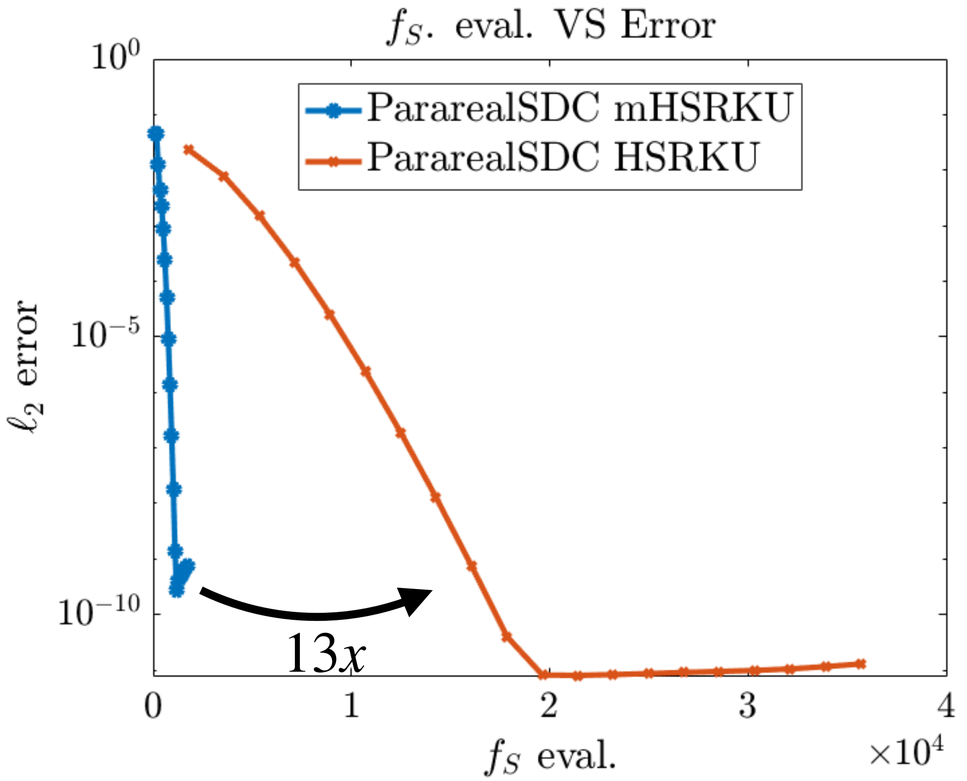
Numerical Experiment on Brusselator problem

$$f(u,z) = \begin{pmatrix} \nu\Delta u + 1 + u^2 v - 4u \\ \nu\Delta v + 3u - u^2 v \end{pmatrix} \quad f_F(u,z) = \begin{pmatrix} \nu\Delta u \\ \nu\Delta v \end{pmatrix} \quad f_S(u,z) = \begin{pmatrix} 1 + u^2 v - 4u \\ 3u - u^2 v \end{pmatrix}$$



Numerical Experiment on Brusselator problem





Conclusions

The first results are fine but there's much room for improvement.

TODOs

- Better understanding of stability,
- Maybe stabilize

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

  instead of overstabilize $\tilde{y}(t)$?
- Some serious numerical experiments,
- Comparison with implicit schemes.

A short advertising...

International Multigrid Conference

August 22-26

Università della Svizzera Italiana

Lugano, Switzerland

img2022.usi.ch



The end

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.