High precision solvers for autonomous systems of differential equations based on Markov jump processes

Flavius Guiaş

Fachhochschule Dortmund
University of Applied Sciences and Arts

MCQMC 2018, Rennes, 6.07.2018

A general stochastic numerical scheme for ODE systems

Features:

- efficiently applicable to (large) ODE systems originating from spatial discretization of PDEs
- *first approximation:* stochastic direct simulation method (dODE)
- the *full path simulation* in (dODE) is used for increasing the precision (variance reduction) by:
  - Picard iterations
  - Runge-Kutta steps
  - a combination of both
- method with a simple, universal structure, based on the *predictor-corrector* principle
- performance comparable to standard deterministic solvers

The general principle of Picard approximations

ODE system $\dot{X} = F(X),\ F = (F_i)_{i=1}^n$ written in an integral form on a small time interval:

$$X(t+h) = X(t) + \int_t^{t+h} F(X(s))\, ds.$$

Given the approximation $\bar{X}(t) \approx X(t)$, the goal is to compute the approximation $\bar{X}(t+h)$ at the next point of the time discretization grid.
If we have a predictor $\tilde{X}(s)$ on the whole interval $[t, t, +h]$, then $\bar{X}(t+h)$ is given by the Picard–iteration:

$$\bar{X}(t+h) = \bar{X}(t) + \int_t^{t+h} F(\tilde{X}(s))\, ds.$$

Stochastic Picard approximation

In our method we compute the predictor $\tilde{X}(s)$ as a proper Markov jump process.

Then

$$I(\tau) := \int_t^\tau F(\tilde{X}(s))ds$$

is the integral of a step function which is updated after each jump of the process $\tilde{X}(s)$.

The Runge–Kutta principle

- iterations of the form:

$$X^*(t+h) = X^*(t) + \int_t^{t+h} Q(s)\,ds$$

  where $Q(s)$ is a polynomial which interpolates some intermediate values of $F(\tilde{X}(\cdot))$ or $F(\bar{X}(\cdot))$.
- the directly simulated process $\tilde{X}$ or the Picard–approximations $\bar{X}$ are evaluated here at some few equidistant points between t and $t+h$.
- the integral is computed by an exact quadrature formula:

$$\int_t^{t+h} Q(s) = h\sum_{i=1}^{m} b_i k_i$$

  where k_i are suitable approximations for $F(\tilde{X}(t+hc_i))$ or $F(\bar{X}(t+hc_i))$, $0 \le c_i \le 1$.

The Runge–Kutta principle: variants

- $m = 2$ and $c_1 = 0, c_2 = 1$: the trapezoidal rule: $h\left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)$ (RK2)
  $\rightarrow$ integrates exactly the linear interpolant for the two nodes.
- $m = 3$ and $c_1 = 0, c_2 = 1/2, c_3 = 1$: Simpson's 1/3 (or Kepler's) rule: $h\left(\frac{1}{6}k_1 + \frac{4}{6}k_2 + \frac{1}{6}k_3\right)$ (RK3)
  $\rightarrow$ integrates exactly the quadratic interpolation polynomial for the three nodes, but in this particular case also all cubic polynomials.
- $m = 4$ and $c_1 = 0, c_2 = 1/3, c_3 = 2/3, c_4 = 1$: Simpson's 3/8 rule: $h\left(\frac{1}{8}k_1 + \frac{3}{8}k_2 + \frac{3}{8}k_3 + \frac{1}{8}k_4\right)$ (RK4)
  $\rightarrow$ integrates exactly the cubic interpolation polynomial for the four nodes.

Remarks

- in the usual deterministic Runge–Kutta schemes k_i are computed according to the well-known Butcher-tableau
- in our stochastic approach the values of the k_i are based on a stochastic approximation (Markov jump process or its Picard iterate)

Stochastic Picard/Runge–Kutta approximation

EXAMPLE: the exponential function solves $X' = X$, $X(0) = 1$

- Markov jump process $\tilde{X}$ with jumps of fixed size $\frac{1}{N}$
- adapted time steps, inverse proportional to $N \cdot \tilde{X}$

STOCHASTIC ALGORITHM:

$t = 0, \quad \tilde{X}(0) = \bar{X}(0) = 1$

while $(t < t_*)$

{

- $\Delta t = -\log U/(N \cdot X)$ adapted time steps, U: u.d. on [0,1]
- $t = t + \Delta t$
- $\bar{X} = \bar{X} + \Delta t \cdot \tilde{X}$ update the Picard integral

$$\bar{X}(t + \Delta t) = \bar{X}(t) + \int_t^{t+\Delta t} \tilde{X}(s)\, ds$$

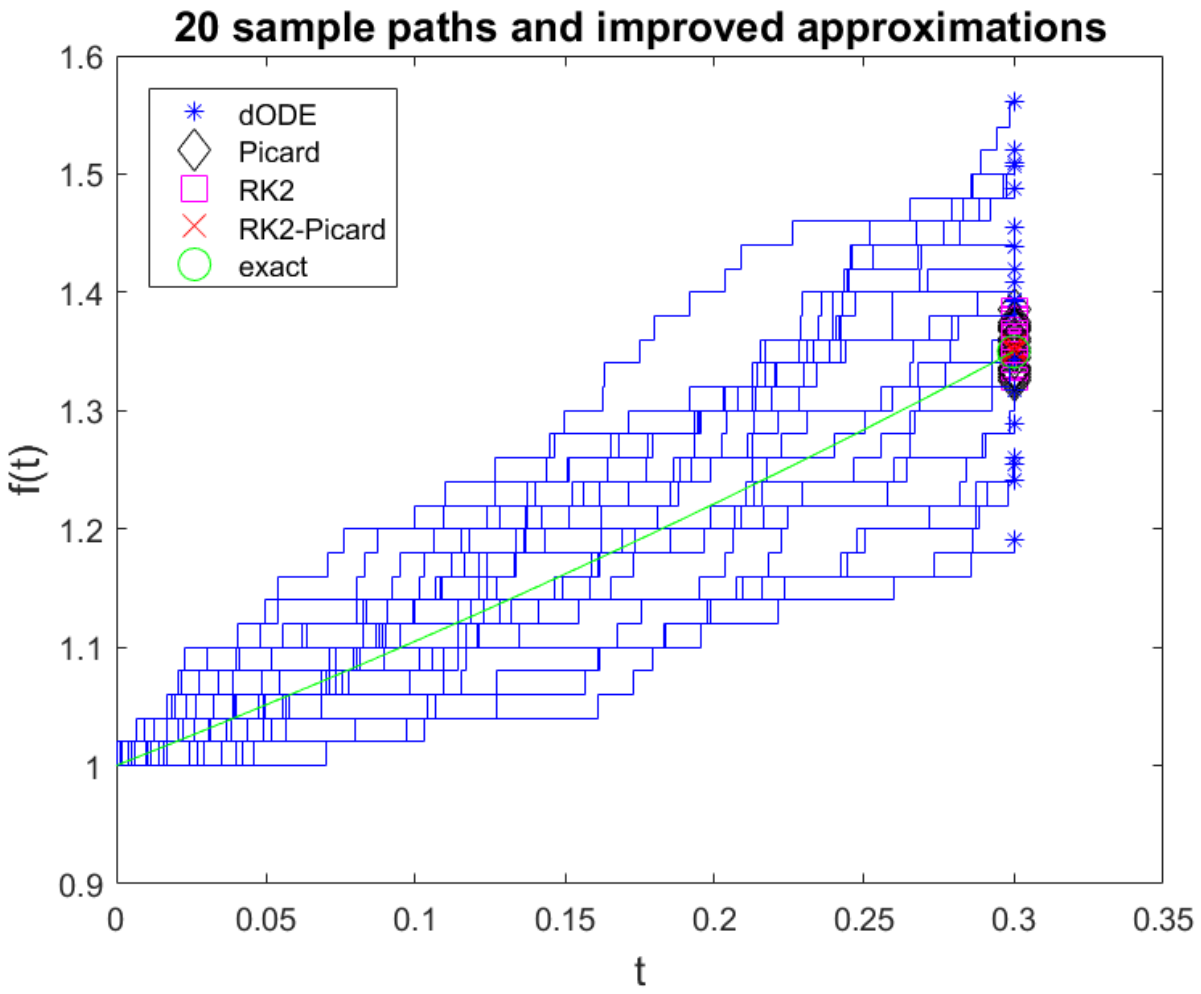
- $\tilde{X} = \tilde{X} + \frac{1}{N}$ update the jump process

}

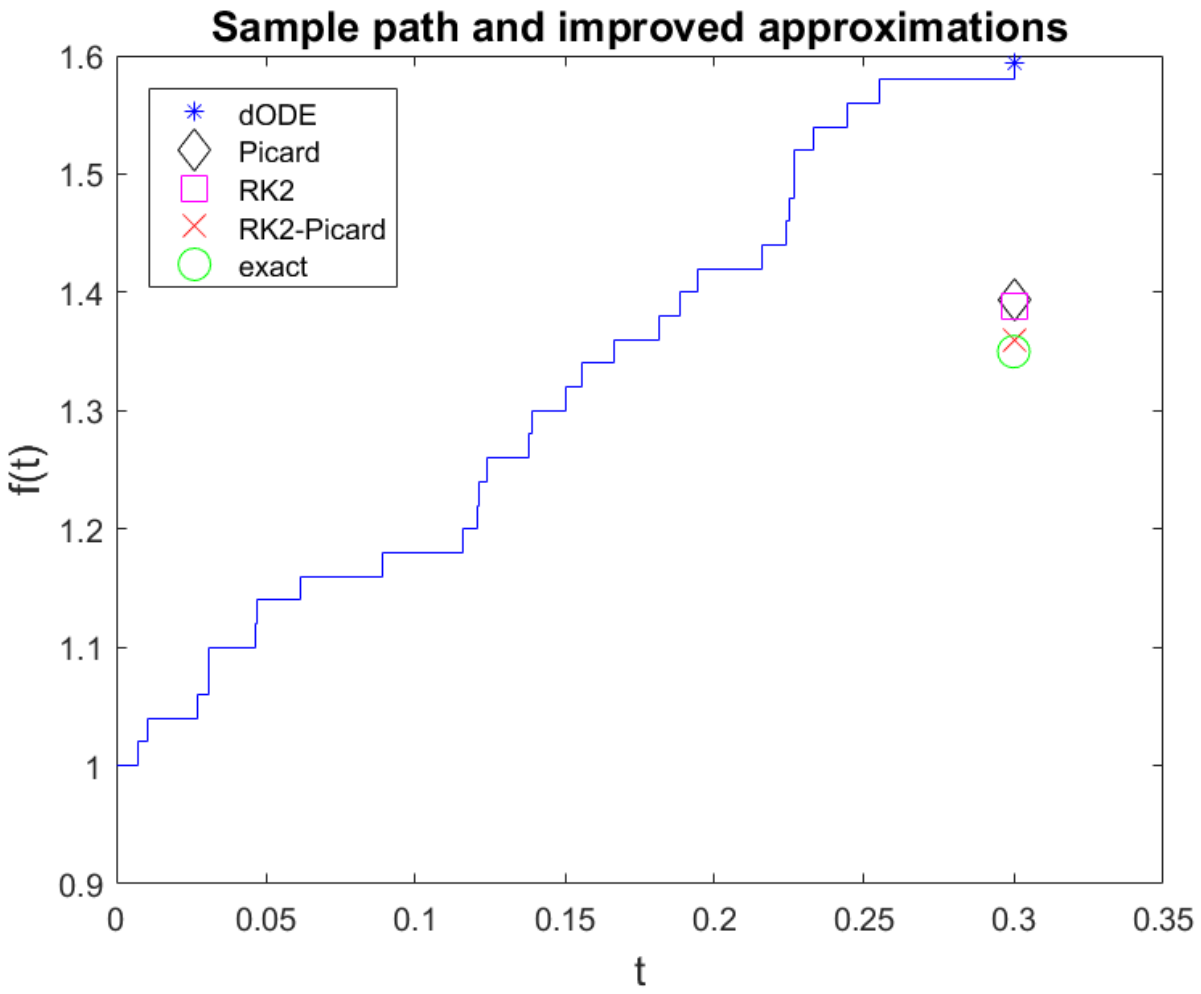
$X^* = \tilde{X}(0) + 0.5 \cdot t \cdot (\tilde{X}(0) + \tilde{X})$ RK2-approximation

$X_1 = \bar{X}(0) + 0.5 \cdot t \cdot (\bar{X}(0) + \bar{X})$ Picard-RK2-approximation

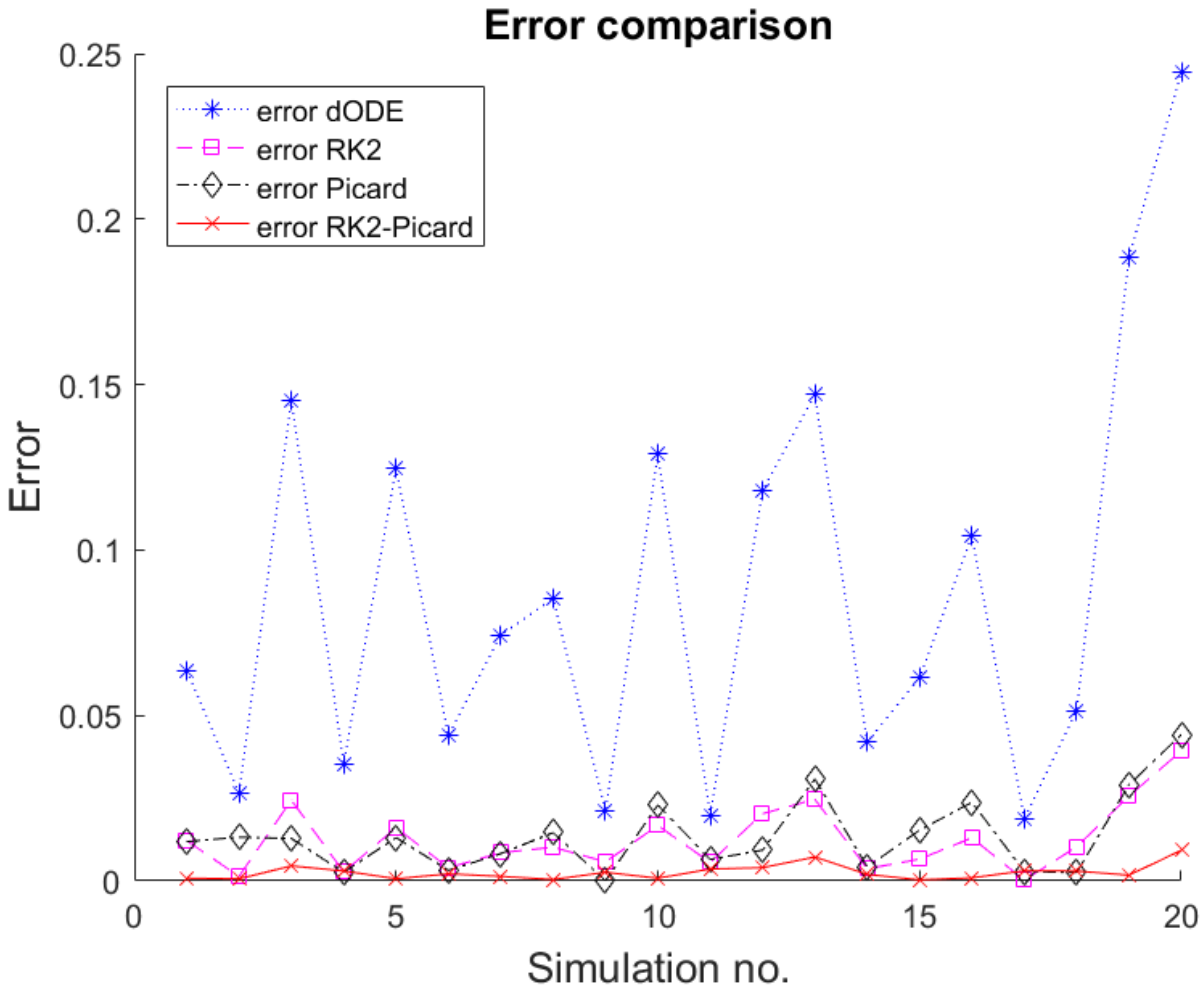
20 SAMPLE PATHS OF $X(\cdot)$ FOR $N = 50$ AND THE IMPROVED APPROXIMATIONS AT $t = t_* = 0.3$



A SAMPLE PATH OF $X(\cdot)$ FOR $N = 50$ AND THE IMPROVED APPROXIMATIONS AT $t = t_* = 0.3$



Error profiles from 20 simulations



Direct simulation of ODEs

Discretization scheme with fixed increment $\frac{1}{N}$ for the **scalar equation** $\dot{x} = F(x)$:

$$x(t + \Delta t) = x(t) + \frac{1}{N} sign(F(x(t)).$$

Consistency requirement: $\Delta t = (N|F(x(t))|)^{-1}$ (if $F(x(t)) \neq 0$).

Remarks:

- change of the quantity $x(t)$ with the fixed increment $\frac{1}{N}$
- the time step Δt is adapted
- approach does not work for **systems** in a deterministic framework, since the time steps required for the single components are different

DIRECT SIMULATION OF ODE SYSTEMS

ODE Systems: $\dot{x}_i = F_i(x),\ i = 1, \ldots, n$, ($F$ Lipschitz continuous).

- stochastic approach: change the components sequentially, by sampling with given frequencies
- selection of the component which is updated according to a proper probability distribution

Approximation by Markov jump process with transitions

$$x \mapsto x + \frac{1}{N} sign(F_i(x)) e_i \text{ at rate } N|F_i(x)|$$

and waiting time parameter

$$\lambda = N \sum_{i=1}^{n} |F_i(x)|$$

Direct ODE simulation algorithm (dODE)

While $t \leq t_{max}$ do:

1. Given the state vector $x(t)$ of the process at time t:
2. Select a component x_i with probability proportional to $|F_i(x(t))|$.
3. The time step $\Delta t = -\log U/\lambda$ with U uniformly distributed on $(0, 1)$ is then exponentially distributed with parameter λ.
4. Update the value of the selected component: $x_i \mapsto x_i + \frac{1}{N} sign(F_i(x))$ and set the new time as $t = t + \Delta t$.
5. Update the values of $F_j(x(t))$ for all j (for which $F_j(x(t))$ depends explicitly on the changed component x_i in step (2)).
6. GOTO 1.

Dynamics of Markov jump processes

Transitions of a Markov jump process X with state space (E, d) (polish space, e.g. $\mathbb{R}^n$ or space of Radon measures):

$$X \mapsto X' \text{ at rate } r(X, dX')$$

Martingale representation in terms of the infinitesimal generator Λ:

$$\Psi(X(t)) = \Psi(X(0)) + \int_0^t (\Lambda\Psi)(X(s))ds + M_\Psi(t)$$

for all bounded continuous functions Ψ defined on E with the deterministic trend

$$(\Lambda\Psi)(X(t)) = \int_E (\Psi(X') - \Psi(X))r(X, dX') \tag{1}$$

and trendless stochastic noise (martingale) M_Ψ i.e. $E[M_\psi(t) \,|\, \mathcal{F}_s] = M_\psi(s)$ for $s \leq t$.

Dynamics of Markov jump processes

- family of stochastic processes $X^N(\cdot)$, with values in $D([0,\infty),E)$: set of right continuous with left limits, E-valued functions defined on $[0,\infty)$.
- convergence defined by the *Skorokhod topology* (weaker than the topology of uniform convergence and takes in account also the distance in time between the jumps).
- relative compactness property: the probability distributions $\mathcal{L}(X^{N_k}(\cdot))$ on the space $D([0,\infty),E)$ of a subsequence $X^{N_k}(\cdot)$ converge weakly to the law of a limit process as $N \to \infty$.
- If $E[(M^N(t))^2] \to 0$ (+other conditions), then the limit process solves the deterministic equation

$$\Psi(X(t)) = \Psi(X(0)) + \int_0^t \Lambda\Psi(X(s))ds$$

for any $\Psi \in C_b(E)$ (or in a given subclass).

Remarks:

- Stochastic simulation method - Gillespie ’76-...’06 ...
- Direct ODE simulation method - G’10
- In general inefficient for large ODE systems, due to step (5)
- Affordable for discretized PDE’s (FD-Methods), also on nonuniform grids

IMPROVING THE EFFICIENCY OF THE METHOD

- develop fast sampling algorithms based on Huffman trees
  Problem: sampling from a large list of probabilities $\{p_i\}_{i=1}^n$ subjected to changes after every step

  possible methods:

  - acceptance-rejection (not adequate in this context)
  - inverse transform (*linear search*, usually $\sim O(n)$ operations/step)
  - inverse transform based on **Huffman trees** (*binary search*, $O(\log n)$ operations/step)

- **increasing the convergence order by exploiting the full path simulation of (dODE)**

Picard Iterations

- simulate $\tilde{x}_N(s)$ for $s \in [t, t+\Delta t]$ according to (dODE) with initial value $x_N(t)$
- let

$$x_N(t+\Delta t) = x_N(t) + \int_t^{t+\Delta t} F(\tilde{x}_N(s))ds$$

- computationally: $I(\tau) := \int_t^{\tau} F(\tilde{x}_N(s))ds$ is the integral of a step function which is updated after each jump of the process $\tilde{x}_N(s)$
- comparison: **Euler method**

$$x(t+\Delta t) = x(t) + F(x(t))\Delta t = x(t) + \int_t^{t+\Delta t} F(x(t))ds.$$

Picard Iterations: improvement of the convergence order

δ: upper bound with high probability of the martingale part: $P(\sup \|M_N(s)\| \leq \delta) \geq 1 - \varepsilon$

Proposition

For the (dODE) simulation starting with $\tilde{x}_N(t) = X(t)$ *we have*

$$\|\tilde{x}_N(t + \Delta t) - X(t + \Delta t)\| \leq \delta e^{L\Delta t}, \tag{2}$$

while for the scheme improved by the Picard iteration on $[t, t + \Delta t]$ *we have*

$$\|x_N(t + \Delta t) - X(t + \Delta t)\| \leq \delta \cdot L\Delta t \cdot e^{L\Delta t}, \tag{3}$$

where L *ist the Lipschitz constant of* F*.*

Adaptive stochastic Runge–Kutta Methods

Instead of prescribing a time step h, fix a number M of jumps of the Markov process. h is in this case the sum of all lengths of jump intervals since the previous step and is an automatically adapted time step for (RK).

Variants for the (RK3) scheme in considering the intermediate time step (for (RK4) use an analogue procedure):

- (A1) after $M/2$ jumps. The two time intervals between 0 and $M/2$, respectively $M/2$ and M jumps have not necessarily the same length. Instead of the quadrature formulae, use therefore an exact integration of the corresponding quadratic interpolation polynomial.
- (A2) let h_1 be the time elapsed after $M/2$ jumps and consider the intermediate value at this point. Let $h = 2h_1$ and proceed by the exact quadrature formula.

Runge-Kutta-steps: Improvement of the convergence order

δ: upper bound with high probability of the martingale part: $P(\sup \|M_N(s)\| \leq \delta) \geq 1 - \varepsilon$

Proposition

For (dODE)-Runge-Kutta:

$$\|x_N(\Delta t) - X(\Delta t)\| \leq \delta \frac{L\Delta t}{6}[4e^{L\Delta t/2} + e^{L\Delta t}], \tag{4}$$

For (dODE)-Runge-Kutta with Picard iterations at $t_{i-1} + \Delta t/2$ and t_i:

$$\|x_N(\Delta t) - X(\Delta t)\| \leq \delta \frac{(L\Delta t)^2}{6}[3e^{L\Delta t/2} + \frac{L\Delta t}{4}e^{L\Delta t}]. \tag{5}$$

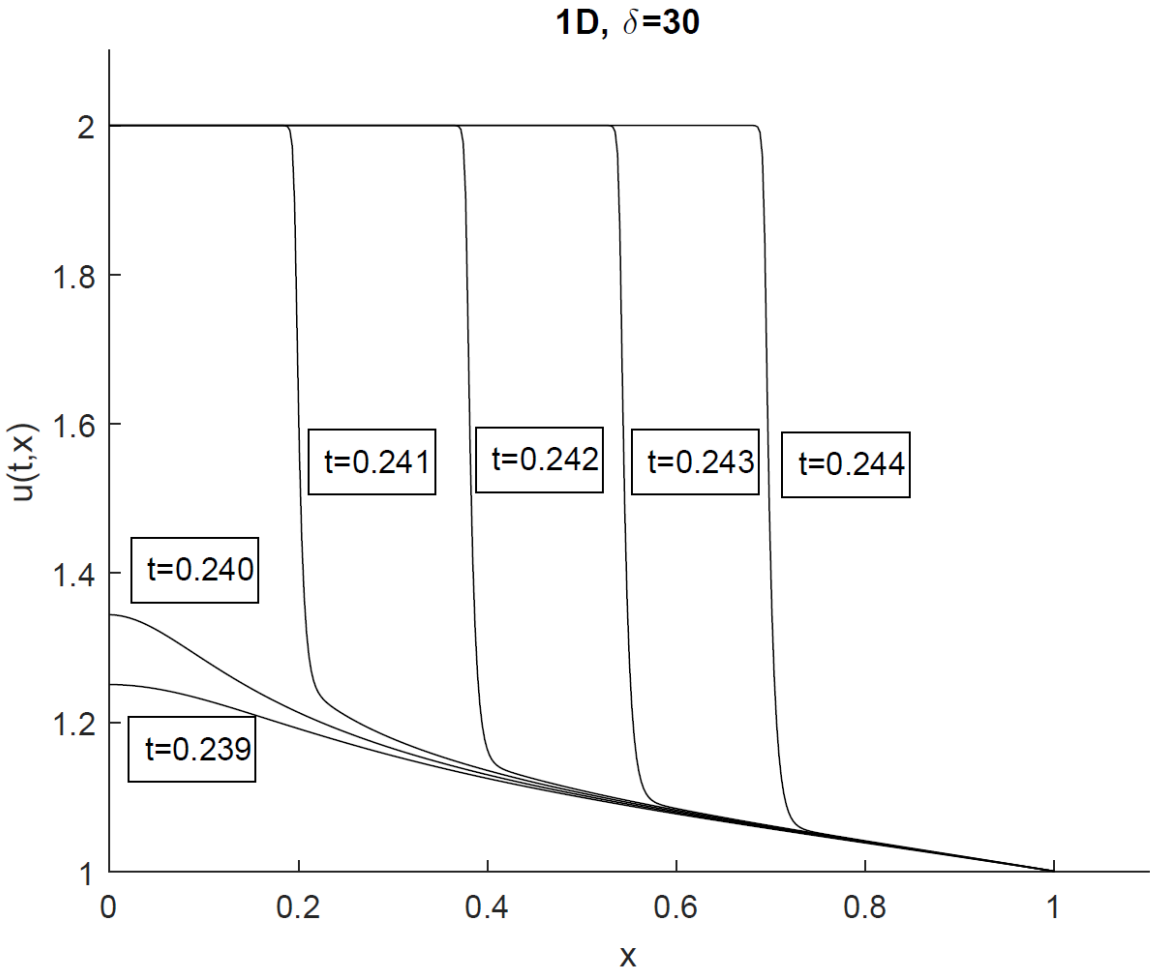
Test problem:

A fast reaction-diffusion equation modeling an *ignition process:*

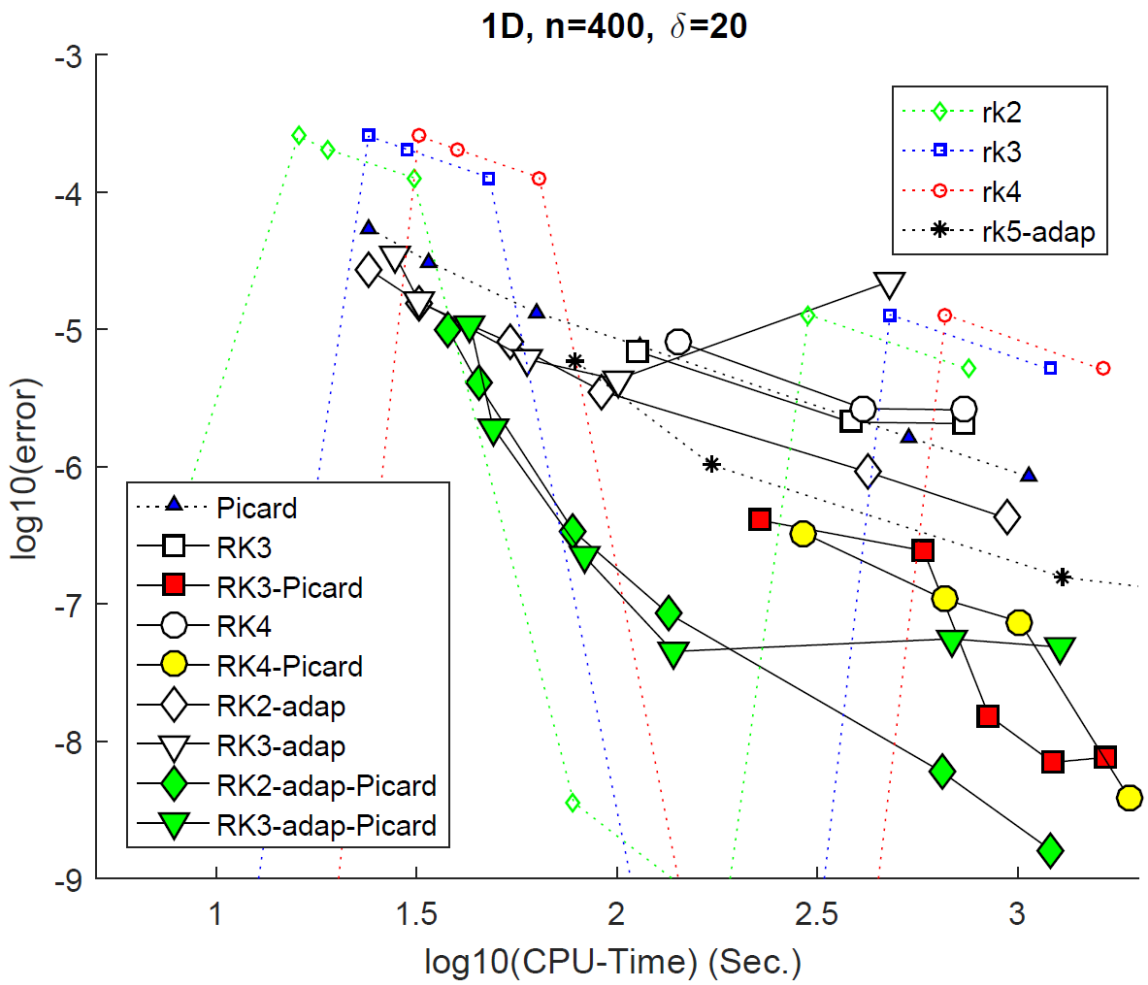
$$\frac{du}{dt} = \Delta u + \frac{5e^{\delta}}{\delta}(2-u)\exp\left(-\frac{\delta}{u}\right)$$

with initial condition $u_0 \equiv 1$

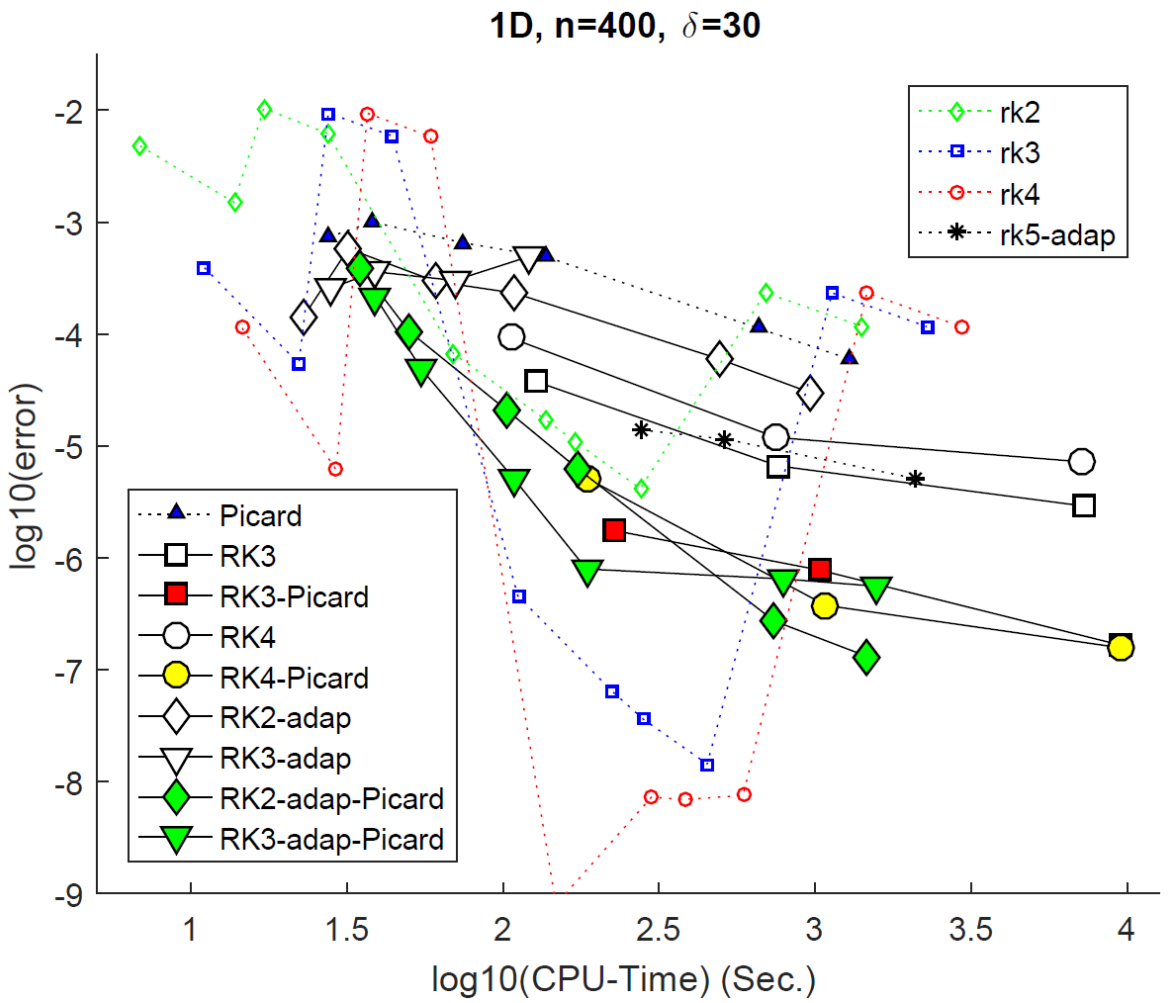
- either on $(0,1)$ with boundary conditions $\partial_\nu u(0) = 0$ and $u(1) = 1$,
- or on the square $(0,1)^2$, with boundary conditions $\partial_\nu u = 0$ if $x = 0$ or $y = 0$ and $u = 1$ if $x = 1$ or $y = 1$.
- ODE-system by finite-difference discretization with uniform grids



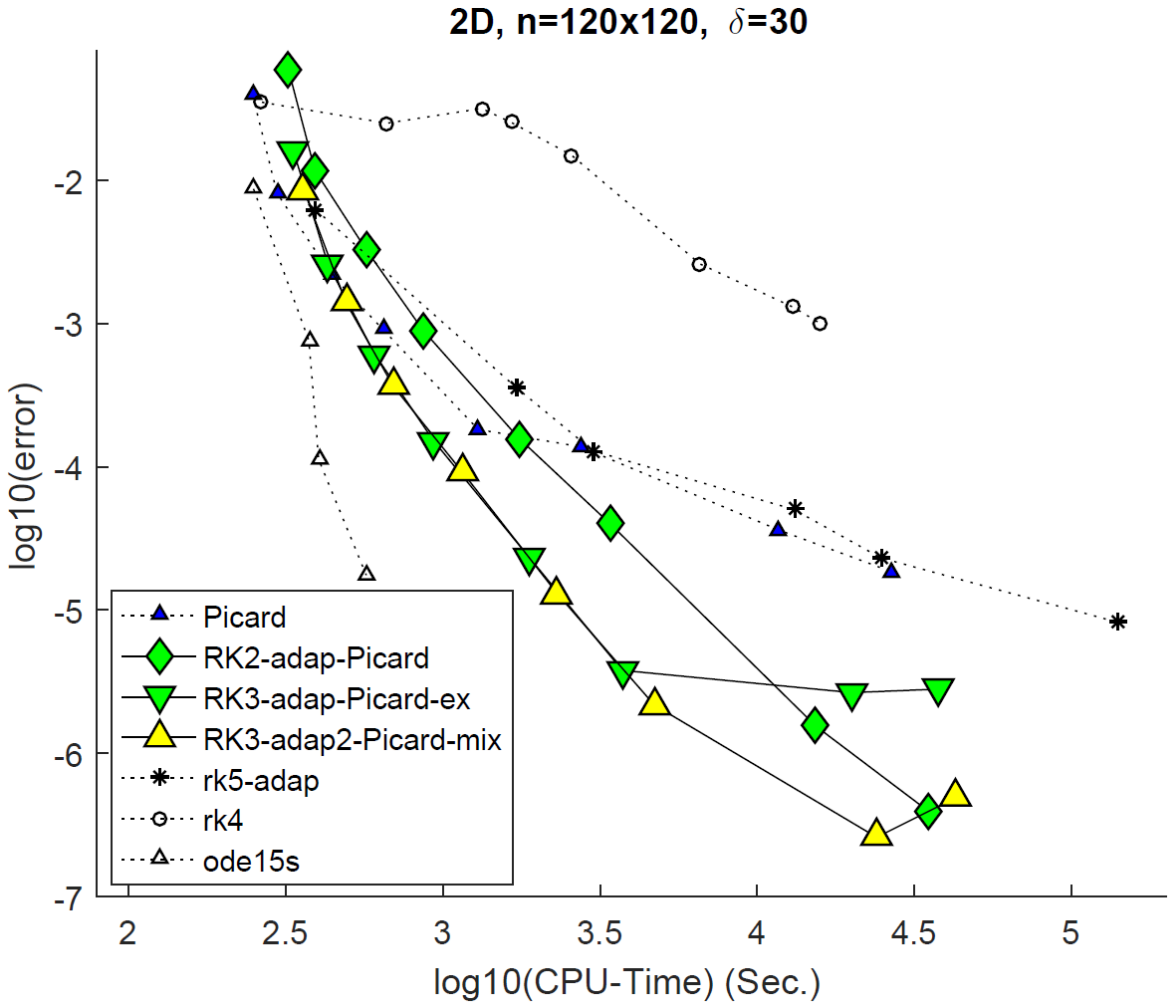
Solution in one space dimension for $\delta = 30$.



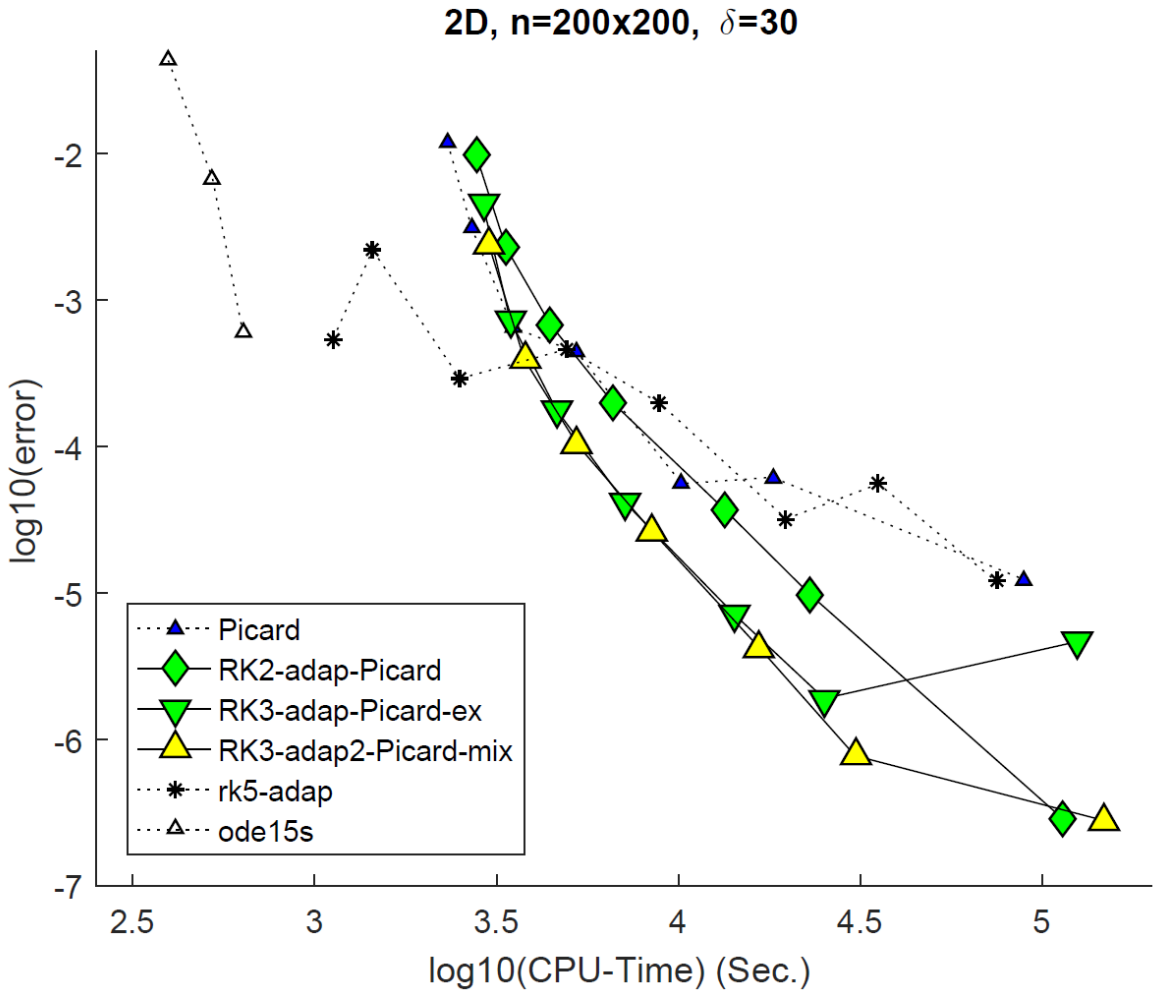
Efficiency comparison in 1D $\delta = 20$, $n = 400$



Efficiency comparison in 1D $\delta = 30$, $n = 400$



Efficiency comparison in 2D $\delta = 30$, $n = 120 \times 120$



Efficiency comparison in 2D $\delta = 30$, $n = 200 \times 200$

Reference

F. Guiaş, P. Eremeev. Improving the stochastic direct simulation method with applications to evolution partial differential equations. *Appl. Math. Comput.*, 289 : 353–370, 2016.

Outlook

- PDE discretization with adaptive grids in space dimensions $d \geq 2$
- Finite-element discretization $\rightarrow$ linearly implicit ODE systems $A(u) \cdot u' = F(u)$

Thank you for your attention!