PARALLÉLISME EN TEMPS:
Résultats de Convergence

Martin J. Gander
martin.gander@math.unige.ch

Université de Genève

Juin, 2007

Convergence for the Heat Equation

Corollary

The parareal algorithm applied to the heat equation $u_t = \Delta u$ discretized with an L-stable method in time converges superlinearly on bounded time intervals,

$$\max_{1\leq n\leq N} ||u(t_n) - U_n^k||_2 \leq \frac{\gamma_s^k}{k!}\prod_{j=1}^{k}(N-j)\max_{1\leq n\leq N}||u(t_n) - U_n^0||_2,$$

where the constant $\gamma_s < 1$ is universal for each L-stable method. On unbounded time intervals the convergence is linear,

$$\sup_{n>0} ||u(t_n) - U_n^k||_2 \leq \gamma_l^k \sup_{n>0}||u(t_n) - U_n^0||_2,$$

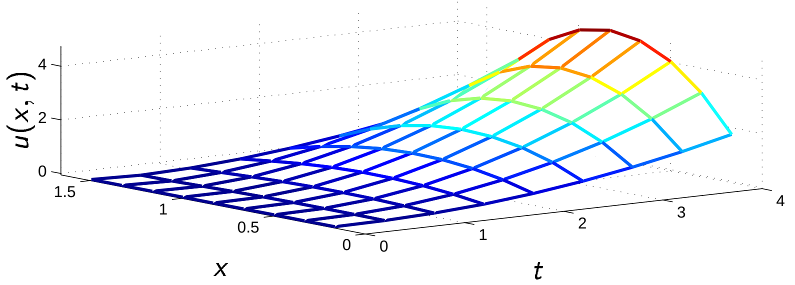
where $\gamma_l < 1$ is universal for each L-stable method.

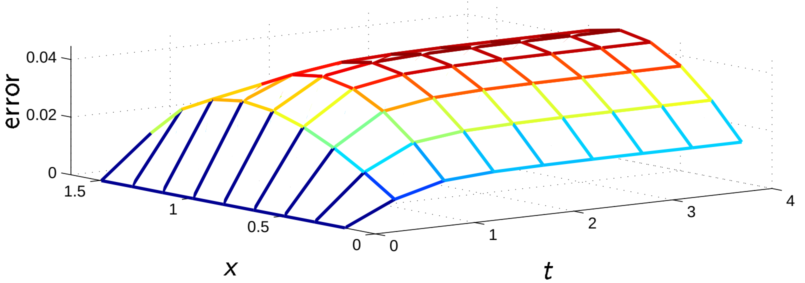
Convergence Constants for the Heat Equation

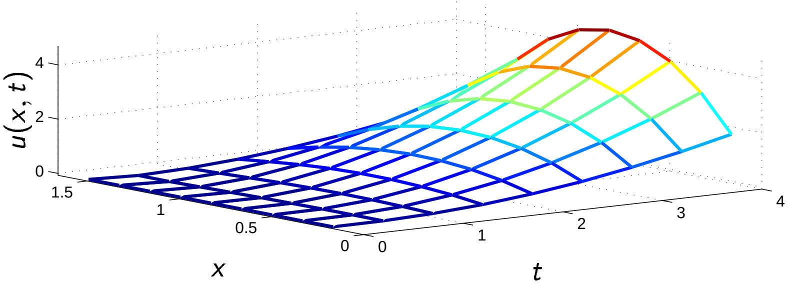
The convergence constants γ_s and γ_l can be computed for each L-stable method:

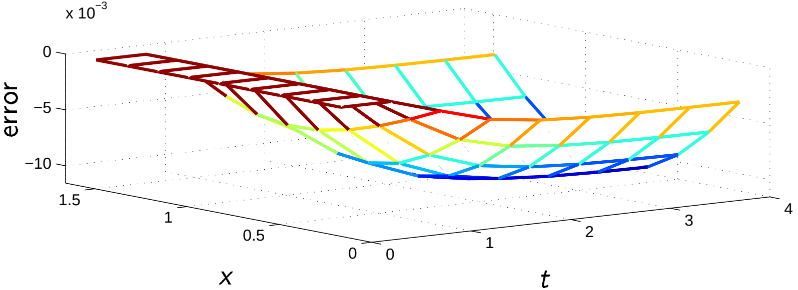
method	order	γ_s	γ_l
BE	1	0.2036321888	0.2984256075
SDIRK 3.1	3	0.1717941220	0.2338191487
SDIRK 3.2	3	0.2073822267	0.1718033767
Radau IIA	5	0.0634592650	0.0677592165

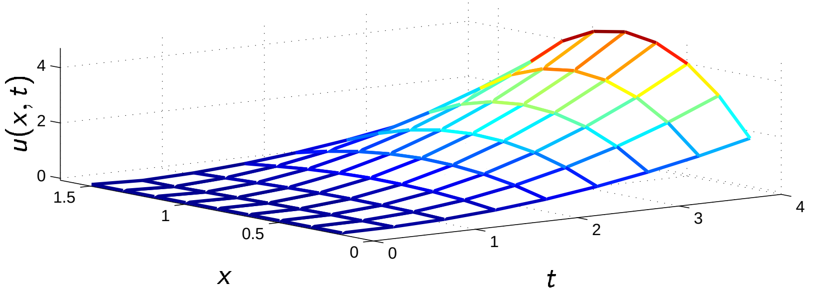
Note that higher order methods lead to faster convergence of the parareal algorithm.

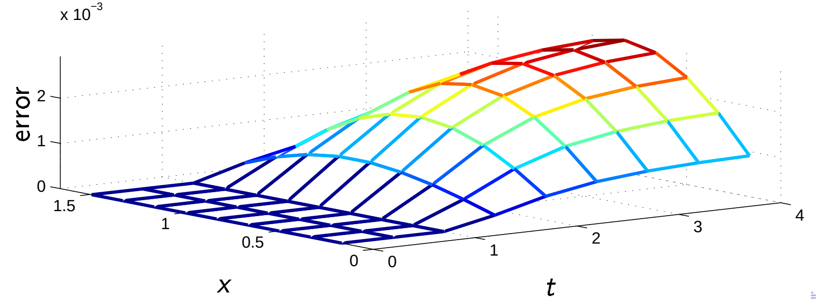

u(x, t)
4
2
0
1.5
1
0.5
0
0
1
2
3
4
x
t

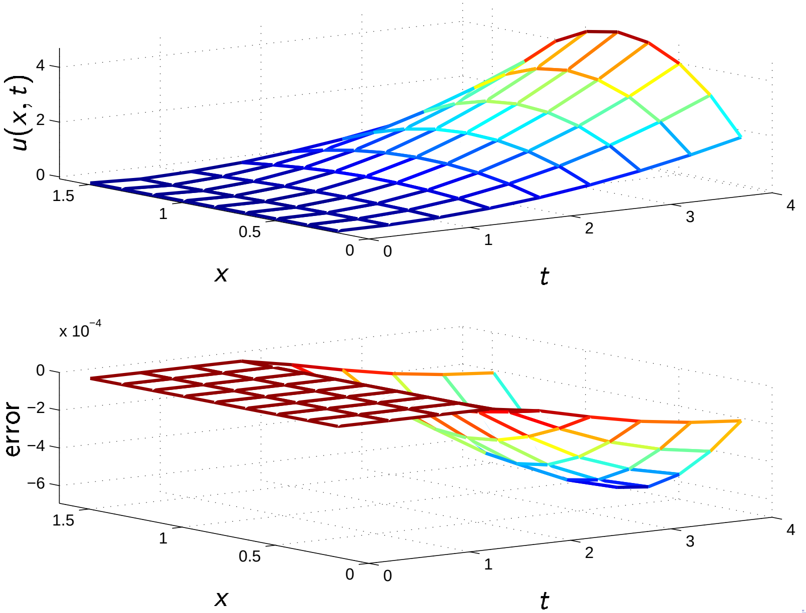

error
0.04
0.02
0
1.5
1
0.5
0
0
1
2
3
4
x
t

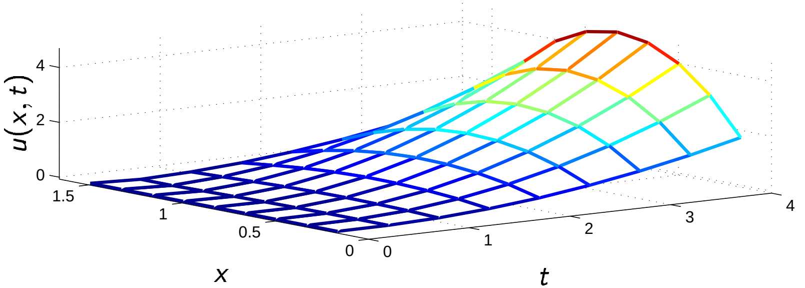

$u(x,t)$
4
2
0
1.5
1
0.5
0
0
1
2
3
4
x
t

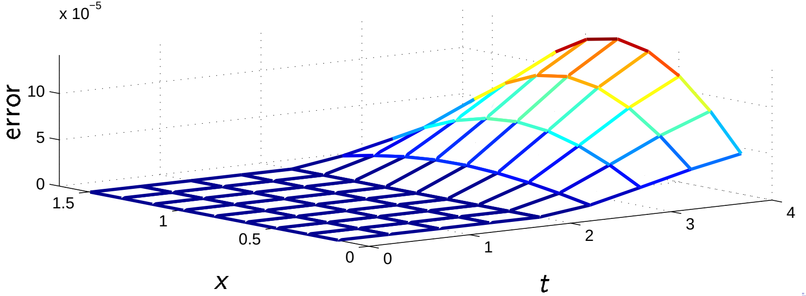

x 10^{-3}
0
−5
−10
error
1.5
1
0.5
0
0
1
2
3
4
x
t

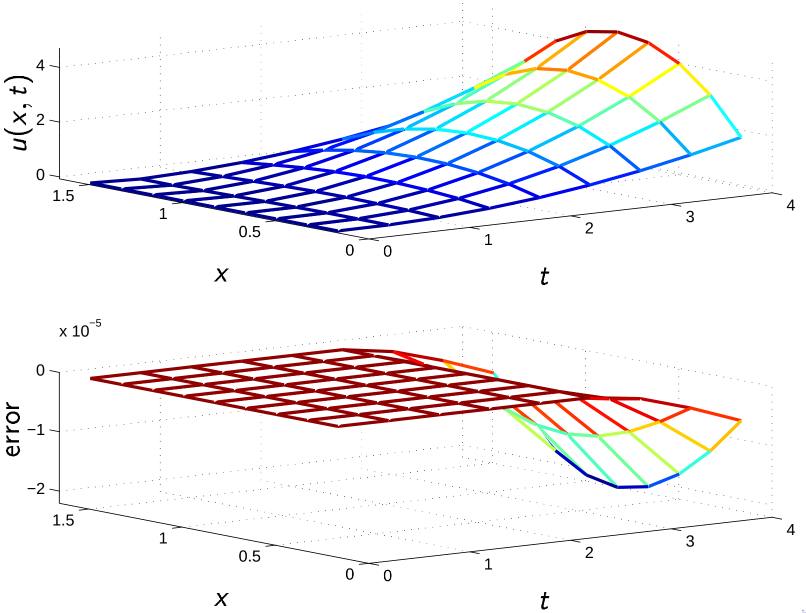

u(x, t)
4
2
0
1.5
1
0.5
0
0
1
2
3
4
x
t

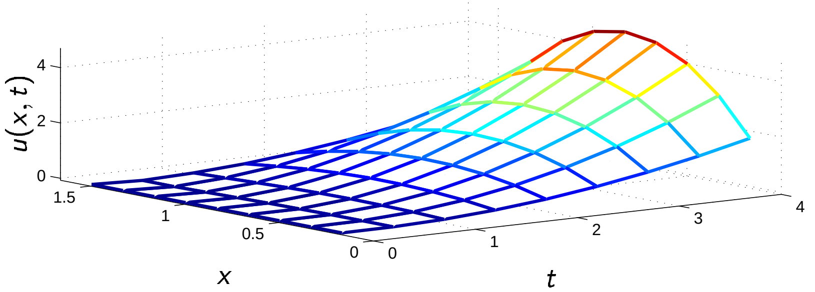

x 10^-3
error
2
1
0
1.5
1
0.5
0
0
1
2
3
4
x
t

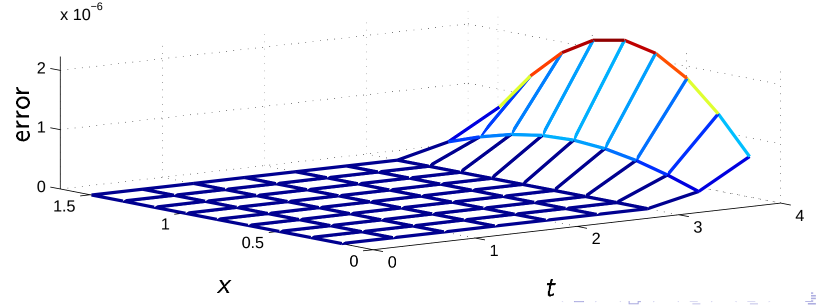

u(x, t)
4
2
0
1.5
1
0.5
0
x
0
1
2
3
4
t
x 10^−4
0
−2
−4
−6
error
1.5
1
0.5
0
x
0
1
2
3
4
t

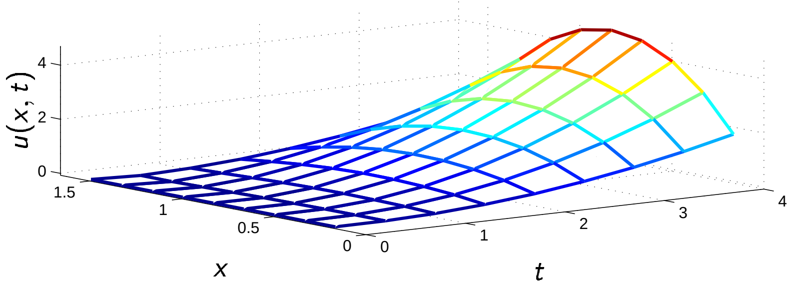

u(x, t)
4
2
0
1.5
1
0.5
0
0
1
2
3
4
x
t

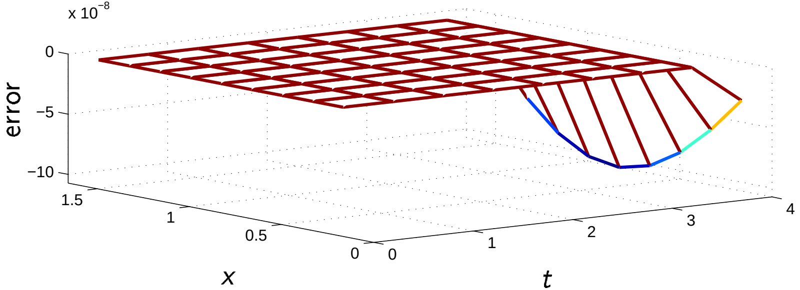

x 10^-5
error
10
5
0
1.5
1
0.5
0
0
1
2
3
4
x
t

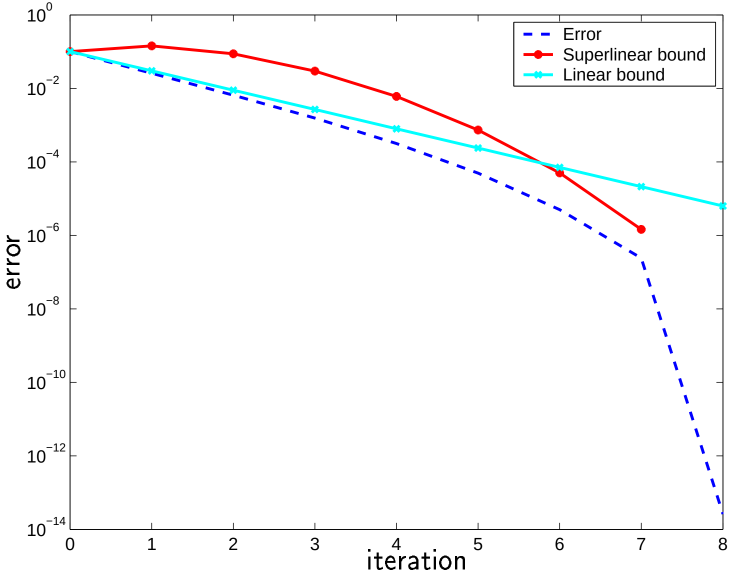

u(x, t)
4
2
0
1.5
1
0.5
0
0
1
2
3
4
x
t
x 10^−5
0
−1
−2
error
1.5
1
0.5
0
0
1
2
3
4
x
t

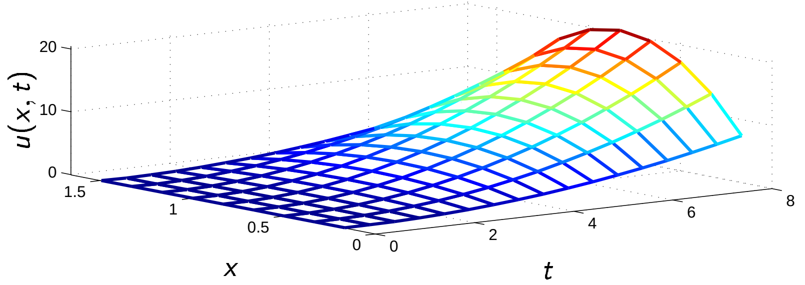

u(x, t)
4
2
0
1.5
1
0.5
0
0
1
2
3
4
x
t

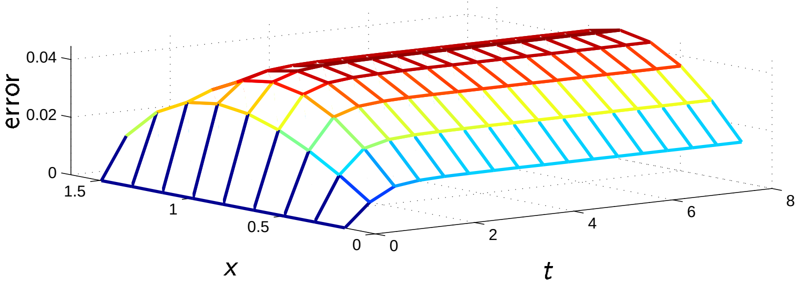

x 10^-6
error
2
1
0
1.5
1
0.5
0
0
1
2
3
4
x
t

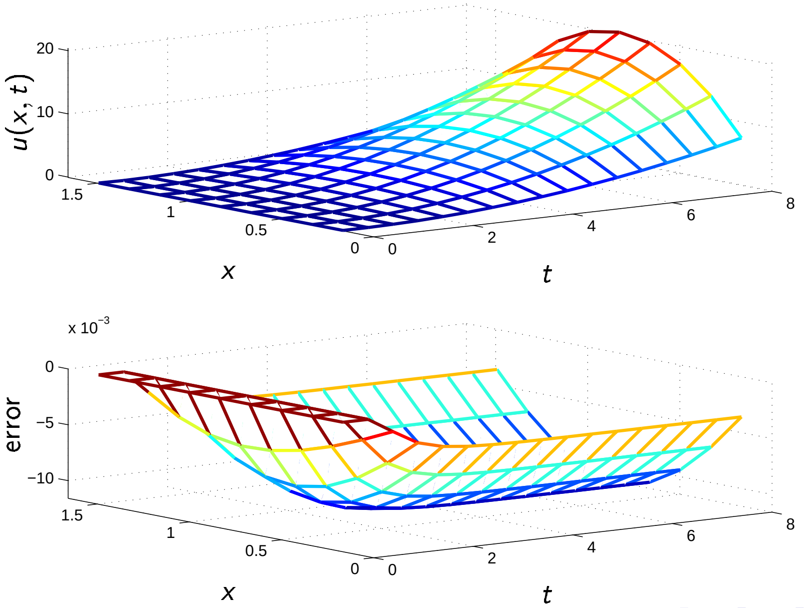

u(x, t)
4
2
0
1.5
1
0.5
0
0
1
2
3
4
x
t

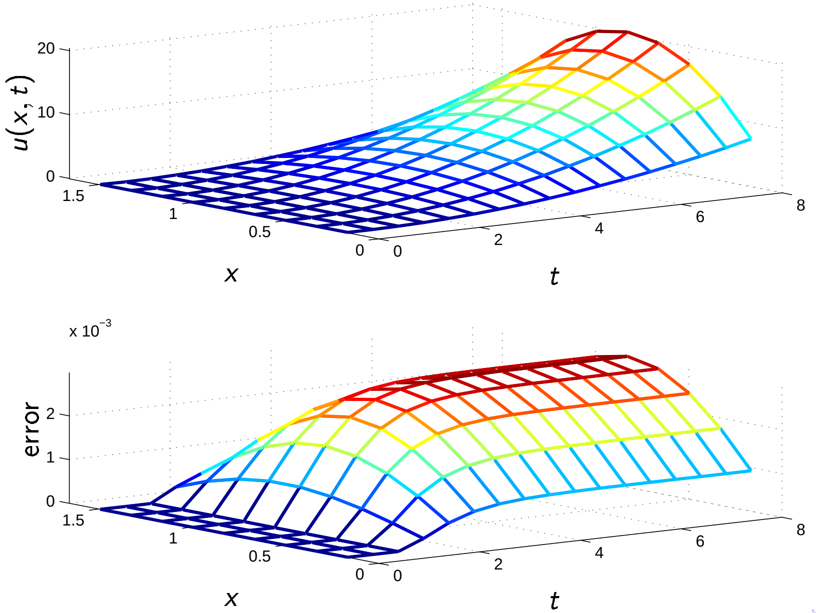

x 10⁻⁸
0
−5
−10
error
1.5
1
0.5
0
0
1
2
3
4
x
t

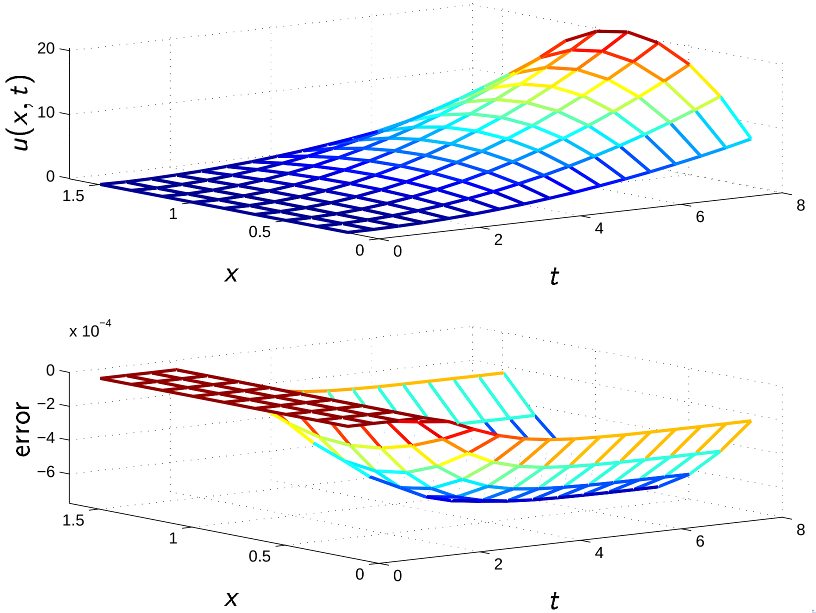

Error
Superlinear bound
Linear bound
error
iteration
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
10^-14
0
1
2
3
4
5
6
7
8

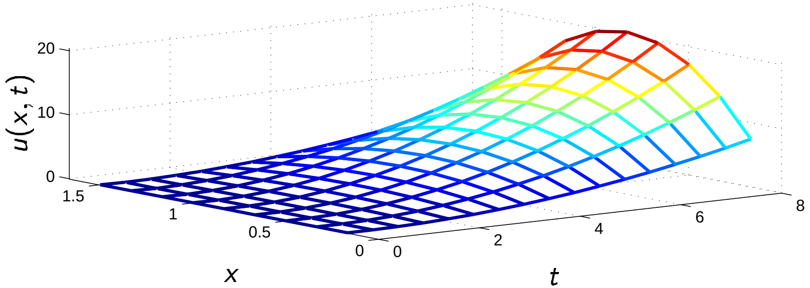

u(x, t)
20
10
0
1.5
1
0.5
0
x
0
2
4
6
8
t

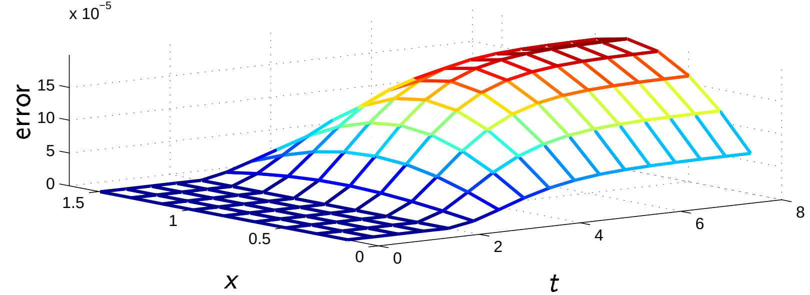

error
0.04
0.02
0
1.5
1
0.5
0
x
0
2
4
6
8
t

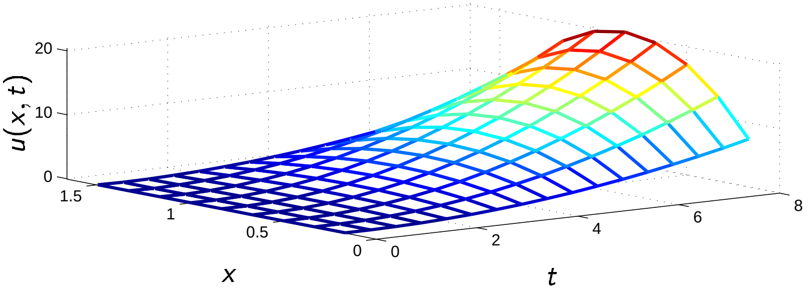

20
10
0
$u(x,t)$
1.5
1
0.5
0
x
0
2
4
6
8
t
x 10^−3
0
−5
−10
error
1.5
1
0.5
0
x
0
2
4
6
8
t

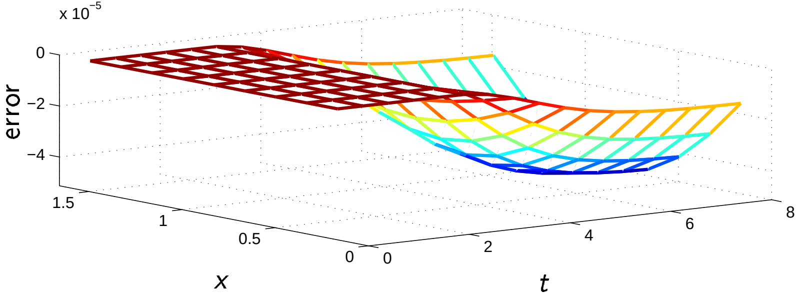

u(x, t)
20
10
0
1.5
1
0.5
0
x
0
2
4
6
8
t
x 10^-3
error
2
1
0
1.5
1
0.5
0
x
0
2
4
6
8
t

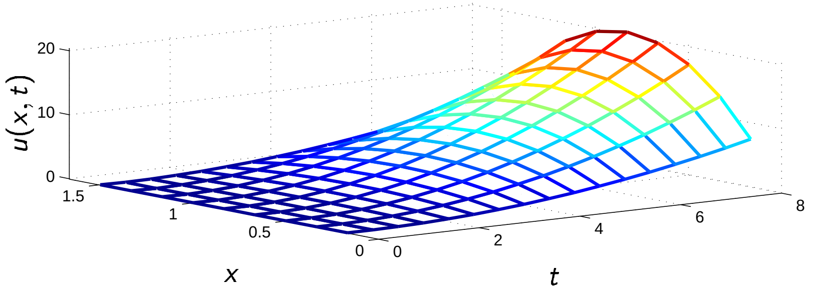

20
10
0
$u(x,t)$
1.5
1
0.5
0
x
0
2
4
6
8
t
x 10^{-4}
0
−2
−4
−6
error
1.5
1
0.5
0
x
0
2
4
6
8
t

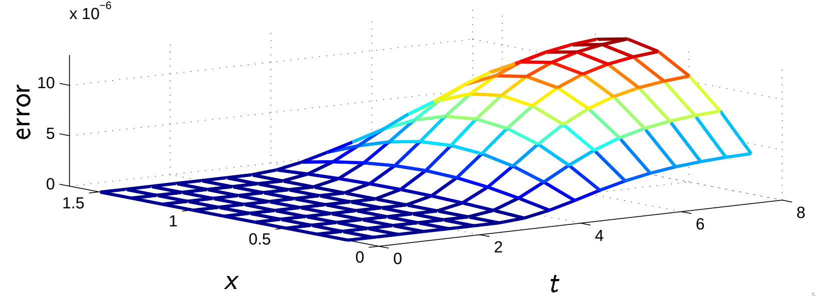

20
10
0
u(x, t)
1.5
1
0.5
0
x
0
2
4
6
8
t

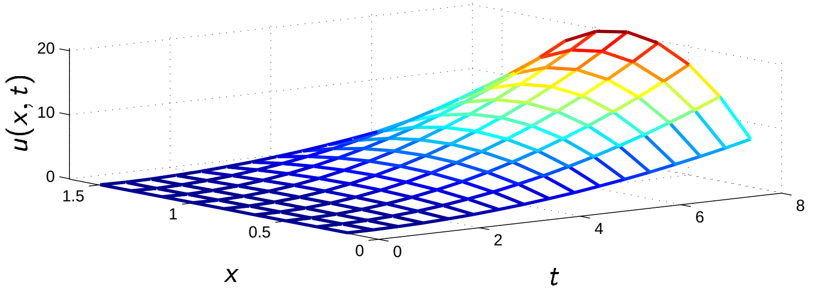

x 10^-5
15
10
5
0
error
1.5
1
0.5
0
x
0
2
4
6
8
t

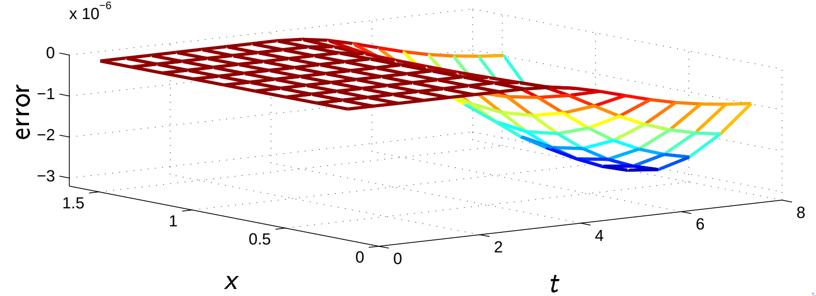

20
10
0
u(x, t)
1.5
1
0.5
0
x
0
2
4
6
8
t

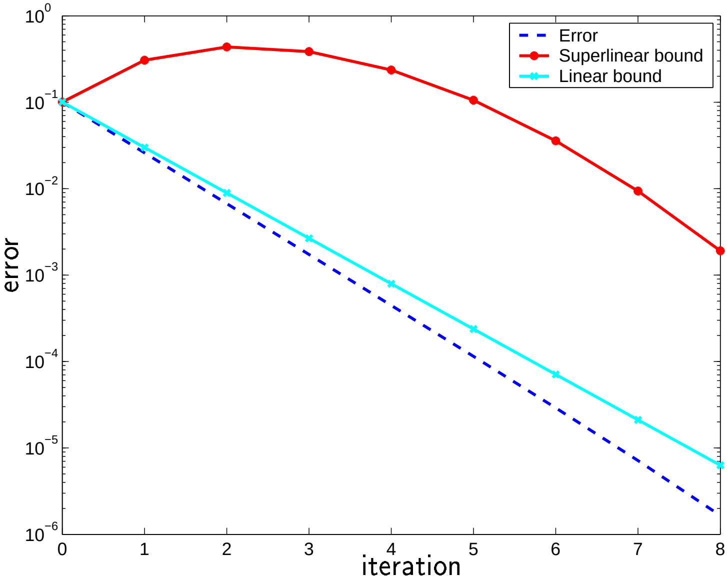

x 10^-5
0
−2
−4
error
1.5
1
0.5
0
x
0
2
4
6
8
t


u(x, t)
20
10
0
1.5
1
0.5
0
x
0
2
4
6
8
t

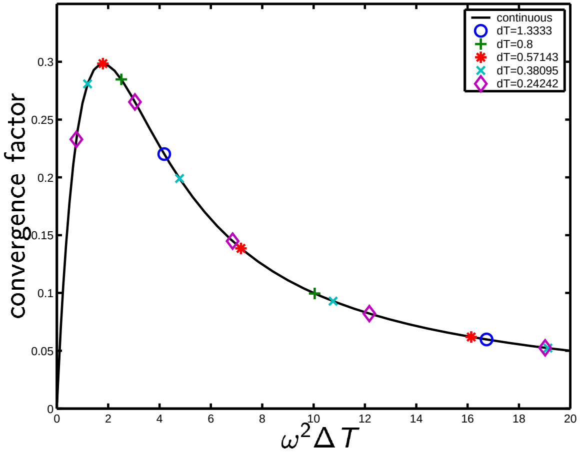

x 10^-6
error
10
5
0
1.5
1
0.5
0
x
0
2
4
6
8
t


20
10
0
u(x, t)
1.5
1
0.5
0
x
0
2
4
6
8
t


x 10^-6
0
−1
−2
−3
error
1.5
1
0.5
0
x
0
2
4
6
8
t

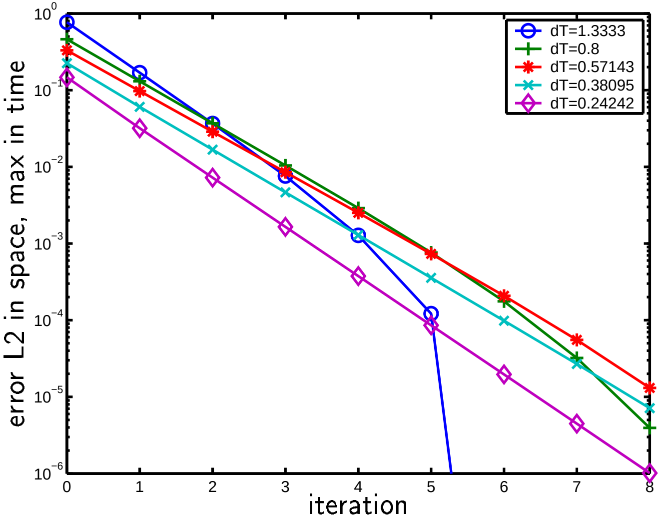

Error
Superlinear bound
Linear bound
error
iteration
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0
1
2
3
4
5
6
7
8

The Results are Upper Bounds



$$\gamma_s := \sup_{\omega \in \mathbb{Z}} |e^{-\omega^2 \Delta T} - \beta(\omega)|, \quad \omega \text{ Fourier parameter in space}$$

Corresponding Numerical Experiments



Convergence for pure Advection Problems

Corollary

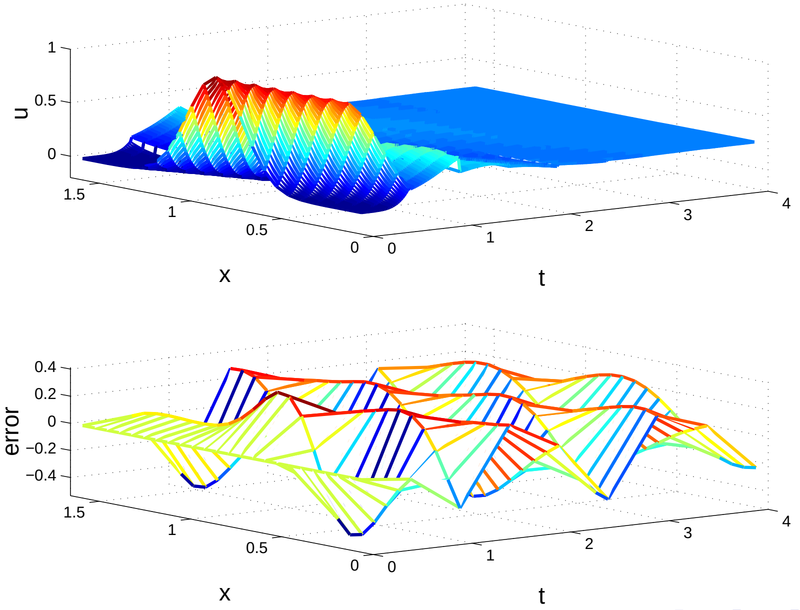
The parareal algorithm applied to the advection equation $u_t = u_x$ with backward Euler in time converges superlinearly on bounded time intervals,

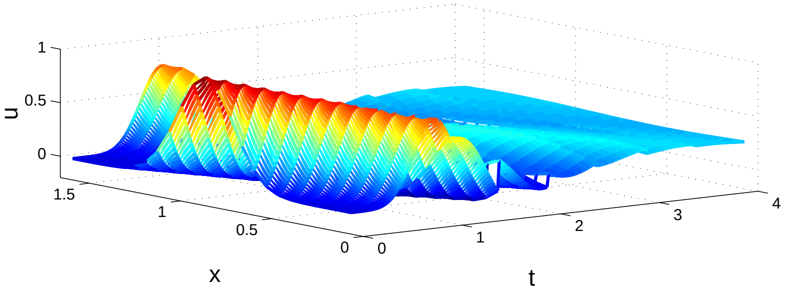
$$\max_{1 \leq n \leq N} ||u(t_n) - U_n^k||_2 \leq \frac{\alpha_s^k}{k!} \prod_{j=1}^{k} (N-j) \max_{1 \leq n \leq N} ||u(t_n) - U_n^0||_2,$$

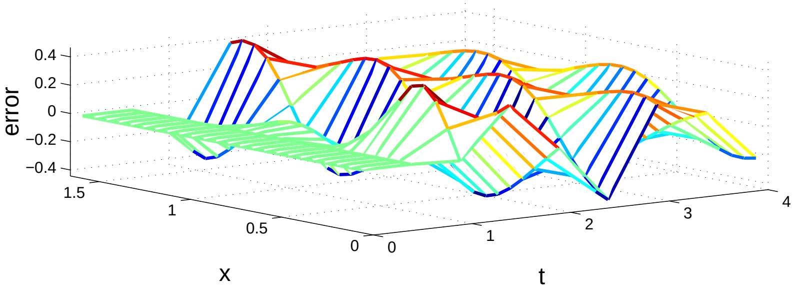
where the constant α_s is universal, $\alpha_s = 1.224353426$.

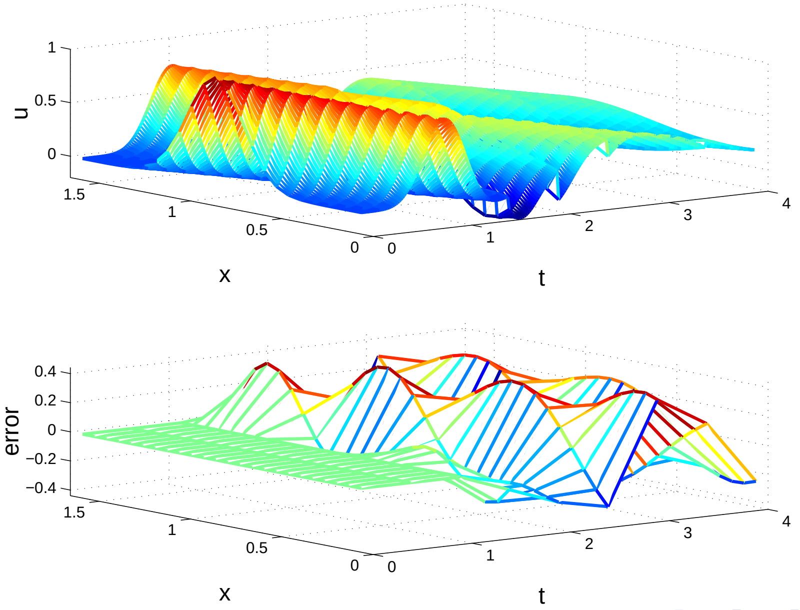
Remarks:

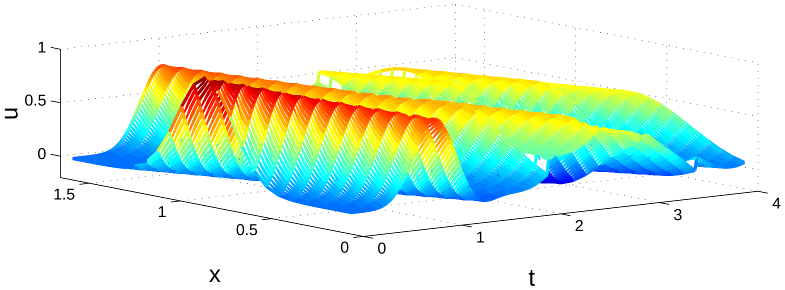
- No convergence result for unbounded time intervals.
- As soon as more than N iterations are needed, the method looses all interest.

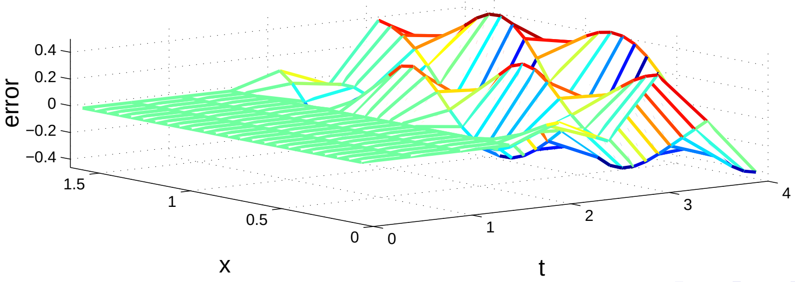

u
1
0.5
0
1.5
1
0.5
0
x
0
1
2
3
4
t
error
0.4
0.2
0
−0.2
−0.4
1.5
1
0.5
0
x
0
1
2
3
4
t

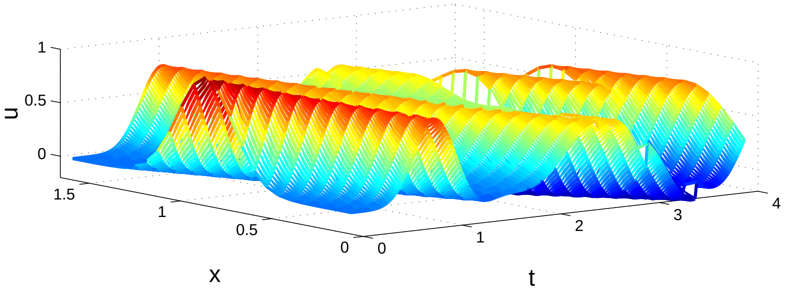

1
0.5
0
u
1.5
1
0.5
0
x
0
1
2
3
4
t

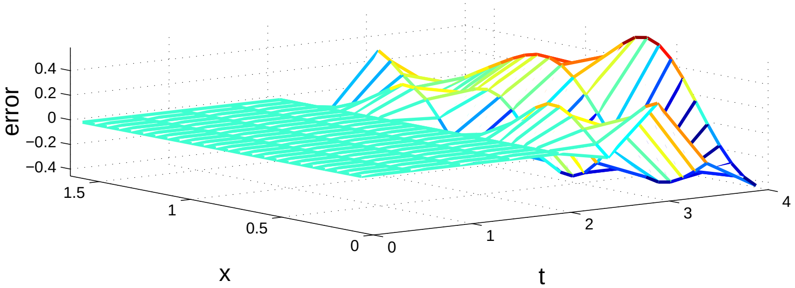

0.4
0.2
0
−0.2
−0.4
error
1.5
1
0.5
0
x
0
1
2
3
4
t

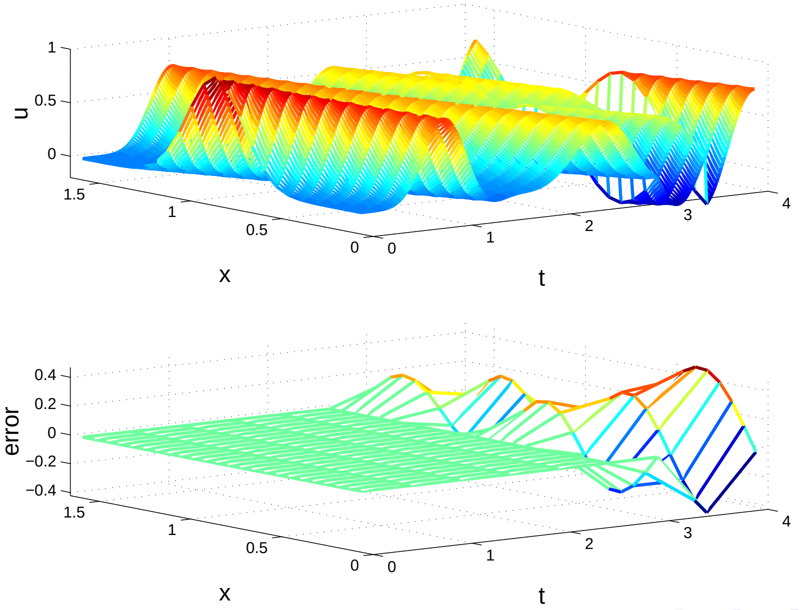

u
1
0.5
0
1.5
1
0.5
0
x
0
1
2
3
4
t
error
0.4
0.2
0
−0.2
−0.4
1.5
1
0.5
0
x
0
1
2
3
4
t

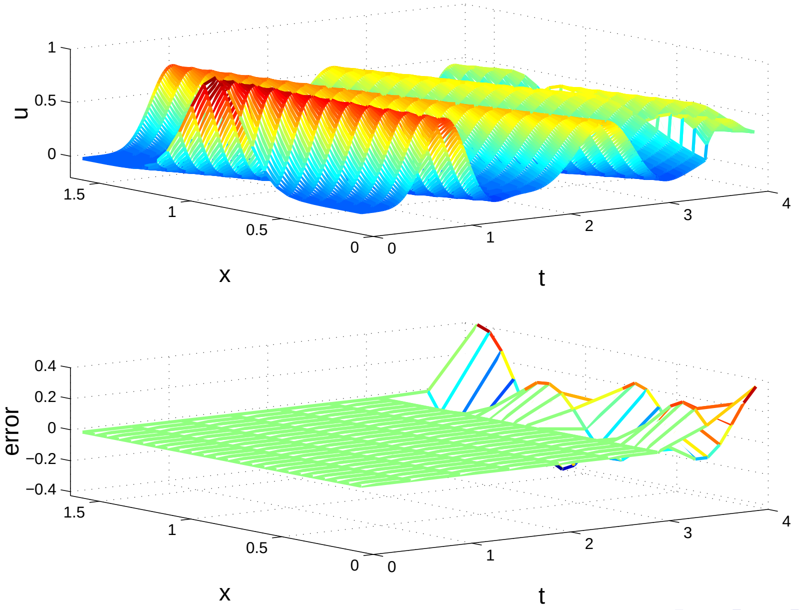

u
1
0.5
0
1.5
1
0.5
0
x
0
1
2
3
4
t

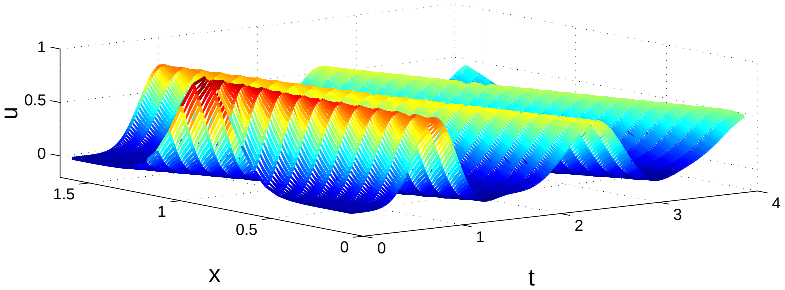

error
0.4
0.2
0
−0.2
−0.4
1.5
1
0.5
0
x
0
1
2
3
4
t

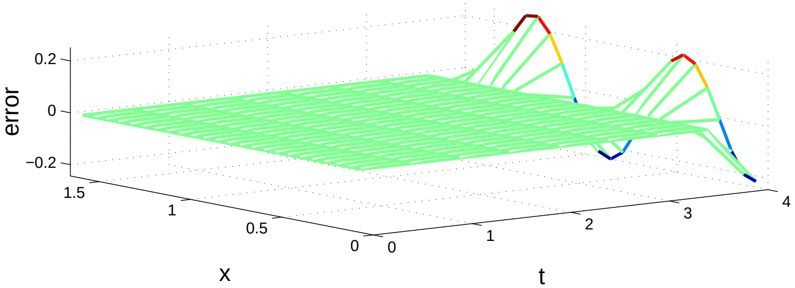

u
1
0.5
0
1.5
1
0.5
0
x
0
1
2
3
4
t

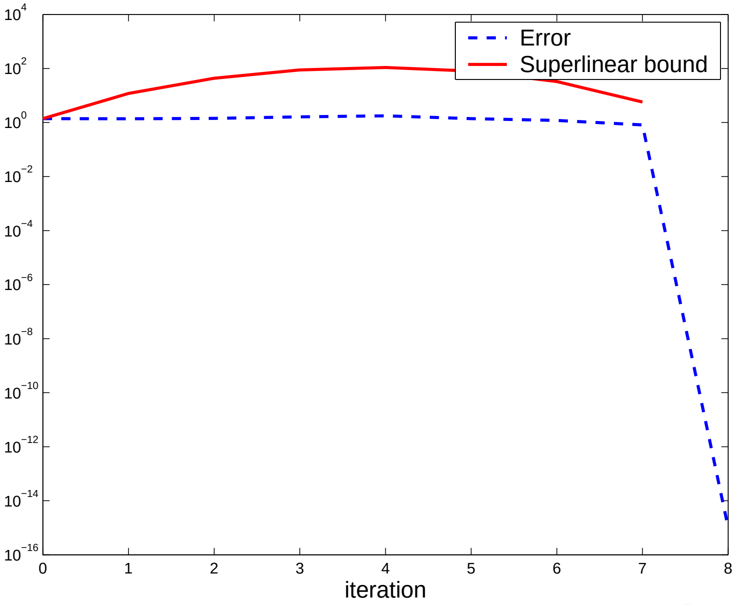

error
0.4
0.2
0
−0.2
−0.4
1.5
1
0.5
0
x
0
1
2
3
4
t


u
1
0.5
0
1.5
1
0.5
0
x
0
1
2
3
4
t
error
0.4
0.2
0
−0.2
−0.4
1.5
1
0.5
0
x
0
1
2
3
4
t


1
0.5
0
u
1.5
1
0.5
0
x
0
1
2
3
4
t
0.4
0.2
0
−0.2
−0.4
error
1.5
1
0.5
0
x
0
1
2
3
4
t


u
1
0.5
0
1.5
1
0.5
0
x
0
1
2
3
4
t


error
0.2
0
−0.2
1.5
1
0.5
0
x
0
1
2
3
4
t


Error
Superlinear bound
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
0
1
2
3
4
5
6
7
8
iteration

Speedup

We define the speedup of the parareal algorithm by N/k, where N is the number of processors (=number of coarse intervals), and k is the number of iterations to achieve a given precision ε.

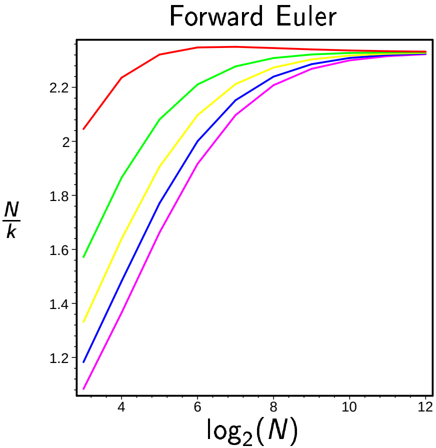
To quantify the speedup of the parareal algorithm, we need to study for $k < N$ the function

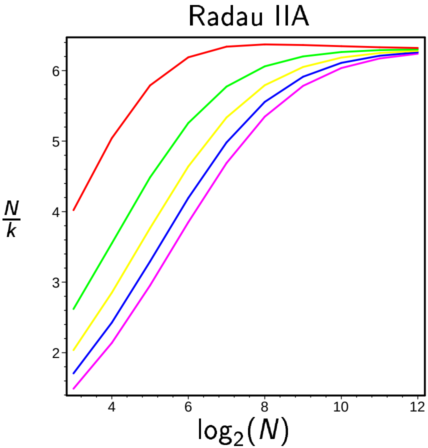
$$f(\gamma, N, k) := \frac{\gamma^k}{k!}\prod_{j=1}^{k}(N-j) \leq \frac{\gamma^k}{k!}\frac{N^N}{e^k(N-k)^{(N-k)}}.$$

Goal: for a given γ from an L-stable method, and a desired precision ε, find N, such that the speedup N/k is maximized.

Speedup for the Heat Equation

For given precision $\varepsilon \in \{1/10, 1/100, \ldots, 1/100000\}$, the speedup N/k as a function of the number of coarse intervals N:





Speedup for the Advection Equation ?

We have

$$\lim_{k\to N} f(\gamma, N, k) \leq \frac{\gamma^N N^N}{N! e^N} =: \bar{f}(\gamma, N),$$

and hence for a given N, speedup is possible if

$$\bar{f}(\gamma, N) < 1.$$

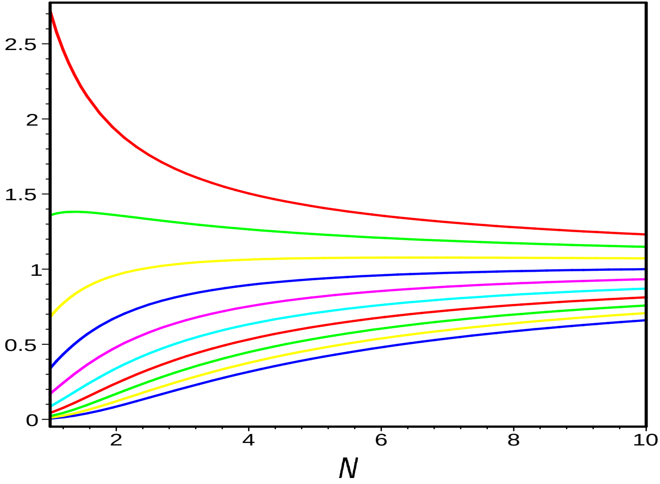
The limiting case $\bar{f}(\gamma, N) = 1$ defines the function

$$\gamma = \gamma(N) = \frac{e(N!)^{\frac{1}{N}}}{N}.$$

and for γ-values above this curve, speedup is not possible.

Possible Speedup Depeding on γ

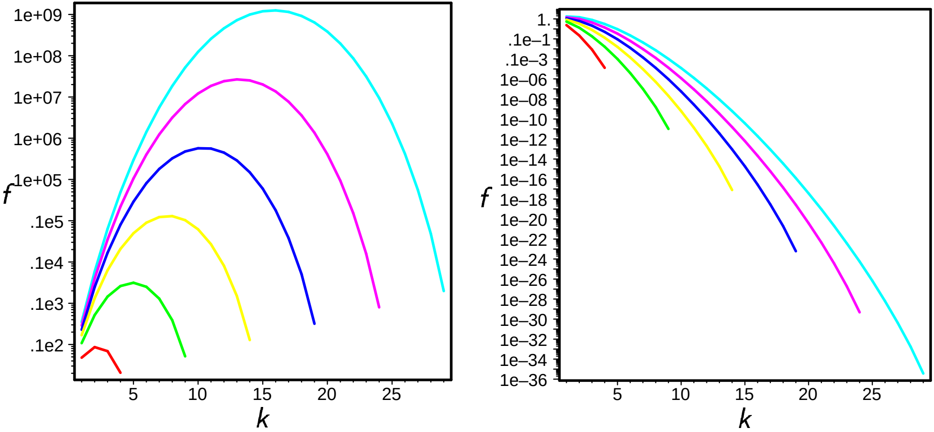
Curves $\bar{f}(\gamma, N) = \varepsilon$ for $\varepsilon = \{1, 1/2, 1/4, \ldots 1/512\}$.



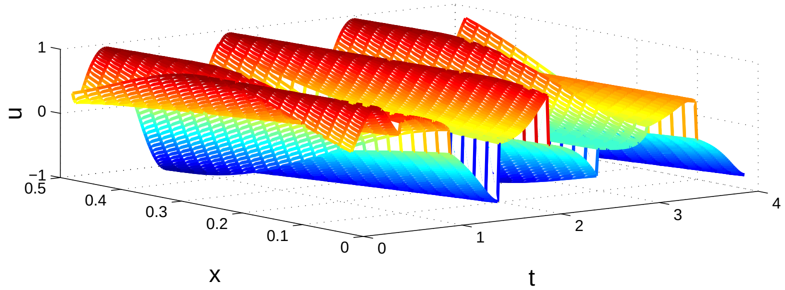
Note: the limit for N large is 1 for all ε.

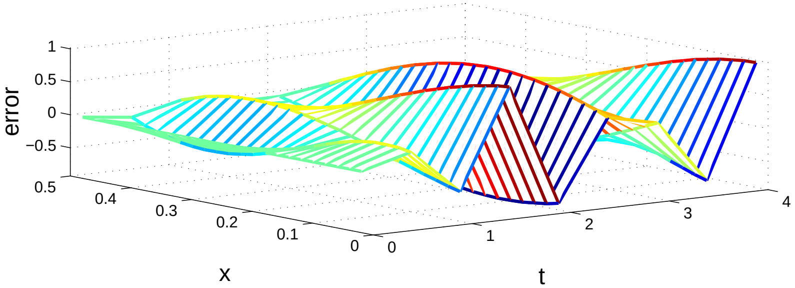
Speedup Comparison between Advection and Diffusion

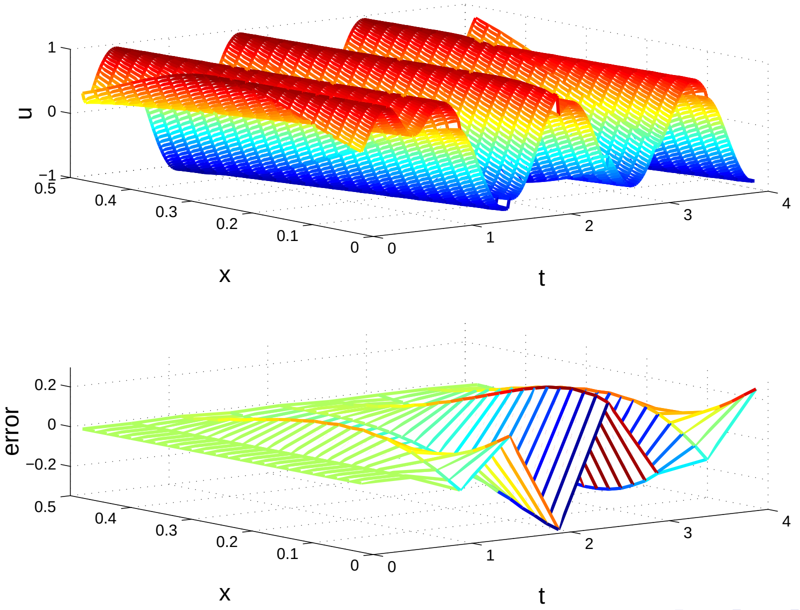
Convergence factors for $N = 5, 10, 15, 20, 25, 30$:

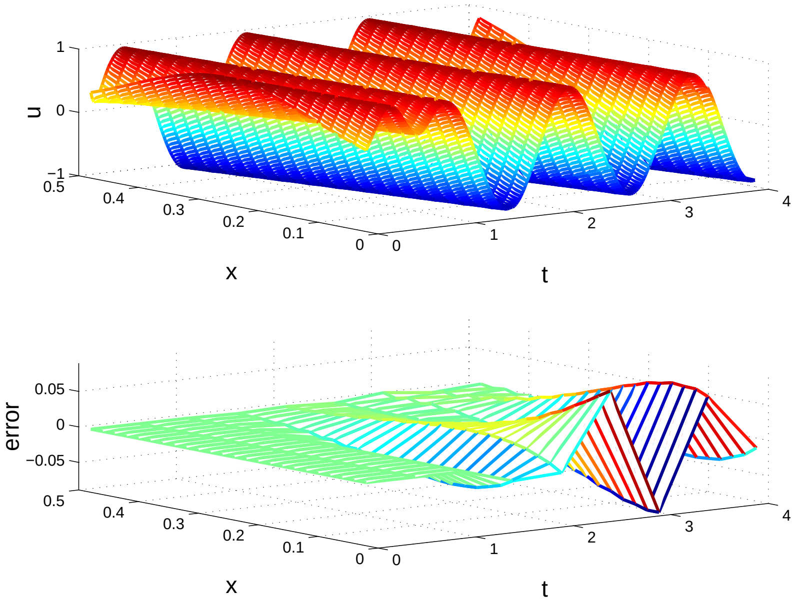


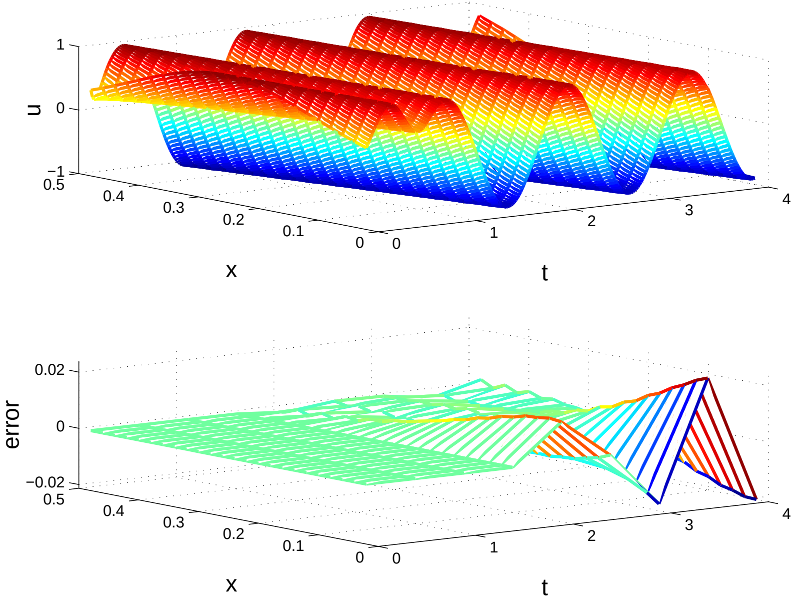
Advection case on the left, diffusion case on the right.

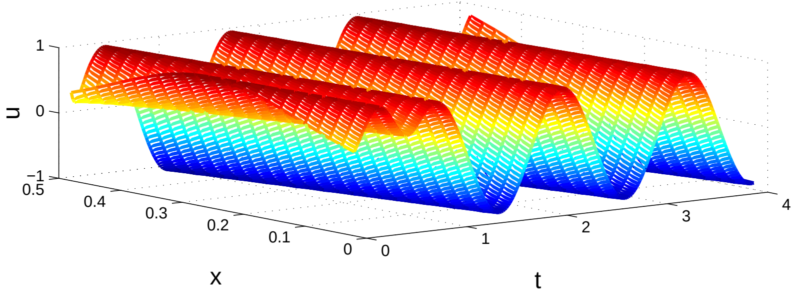

u
1
0
−1
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t

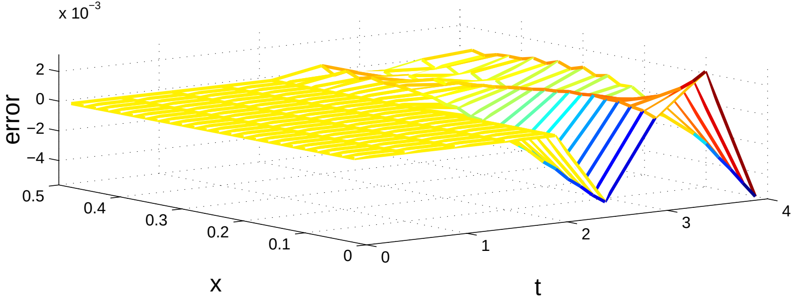

error
1
0.5
0
−0.5
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t

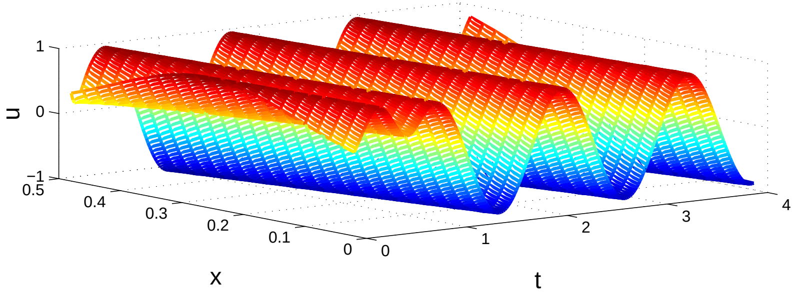

u
1
0
−1
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t
error
0.2
0
−0.2
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t

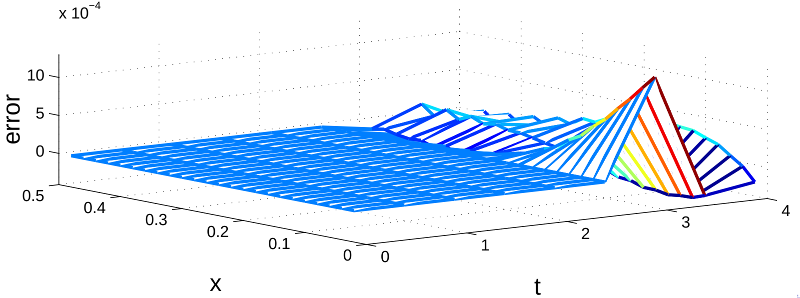

u
1
0
−1
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t
error
0.05
0
−0.05
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t

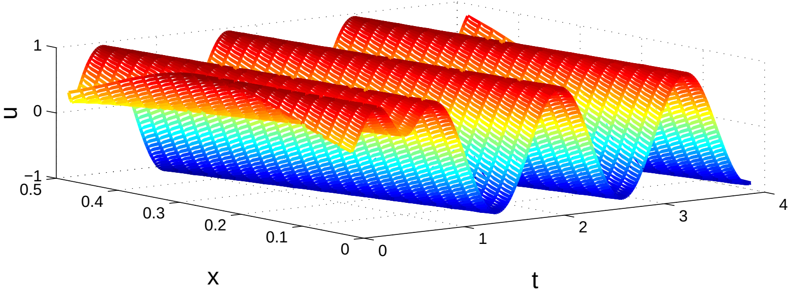

1
0
−1
u
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t
0.02
0
−0.02
error
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t

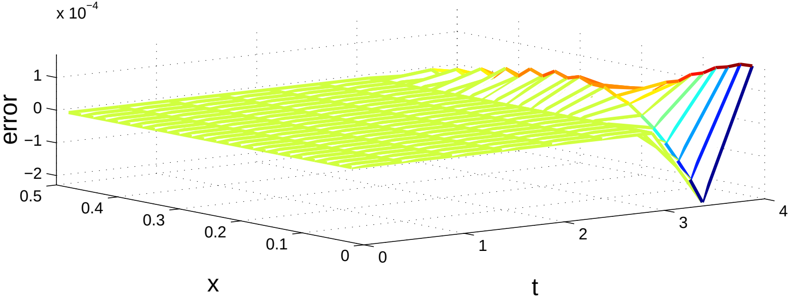

1
0
−1
u
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t

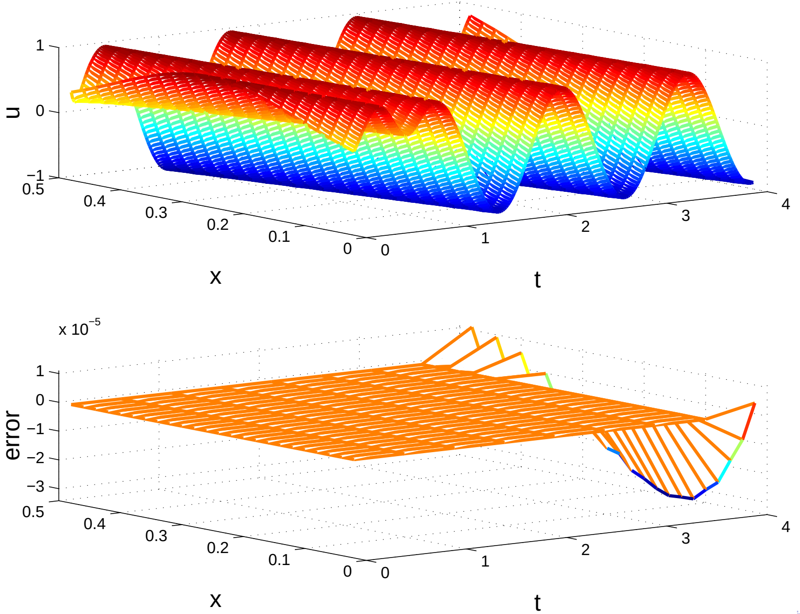

x 10−3
2
0
−2
−4
error
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t

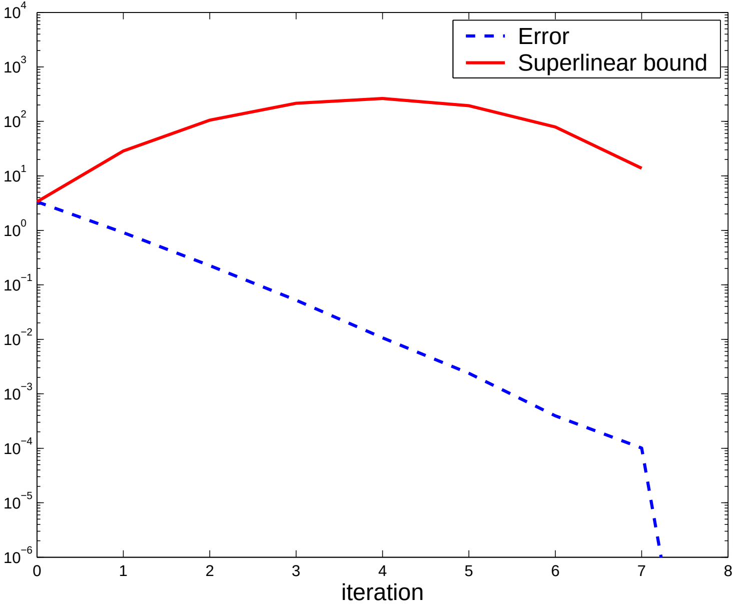

u
1
0
−1
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t


x 10^{-4}
error
10
5
0
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t


1
0
−1
u
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t


x 10^{-4}
1
0
−1
−2
error
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t


1
0
−1
u
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t
x 10^−5
1
0
−1
−2
−3
error
0.5
0.4
0.3
0.2
0.1
0
x
0
1
2
3
4
t


Error
Superlinear bound
10^4
10^3
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
0
1
2
3
4
5
6
7
8
iteration

A General Convergence Result

For the non-linear IVP $u' = f(u), \quad u(t_0) = u_0$.

Theorem (G, Hairer 2006)

Let $F(t_{n+1}, t_n, U_n^k)$ denote the exact solution at t_{n+1} and $G(t_{n+1}, t_n, U_n^k)$ be a one step method with local truncation error bounded by $C_1 \Delta T^{p+1}$. If

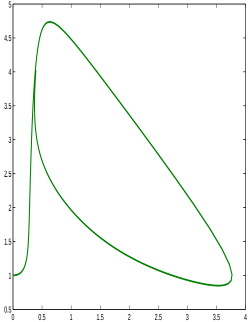
$$|G(t+\Delta T, t, x) - G(t+\Delta T, t, y)| \leq (1 + C_2 \Delta T)|x-y|,$$

then

$$\max_{1\leq n\leq N} |u(t_n) - U_n^k| \leq \frac{C_1 \Delta T^{k(p+1)}}{k!}(1+C_2\Delta T)^{N-1-k}\prod_{j=1}^{k}(N-j)\max_{1\leq n\leq N}|u(t_n)-U_n^0|$$
$$\leq \frac{(C_1 T)^k}{k!} e^{C_2(T-(k+1)\Delta T)}\Delta T^{pk}\max_{1\leq n\leq N}|u(t_n)-U_n^0|.$$

Numerical experiments: Brusselator

$$\begin{aligned} \dot{x} &= A + x^2 y - (B+1)x \\ \dot{y} &= Bx - x^2 y, \end{aligned}$$

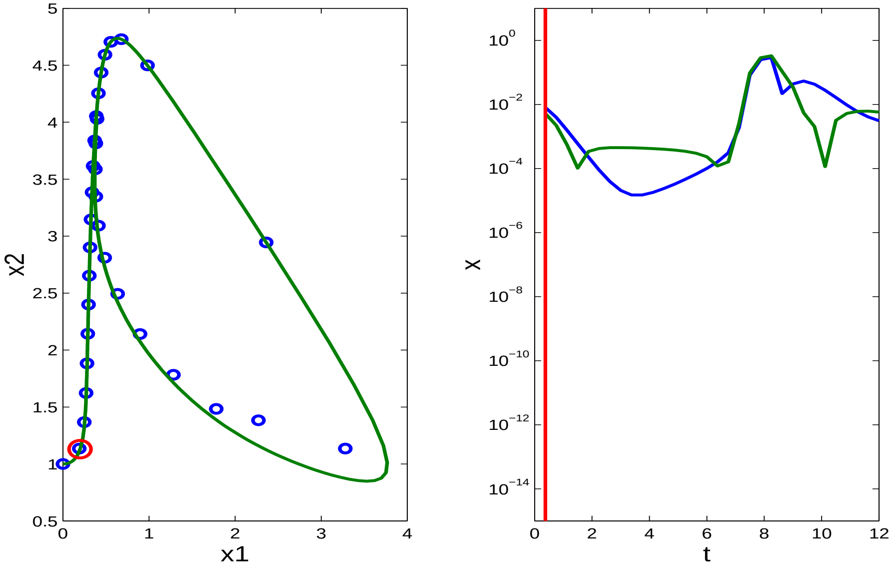


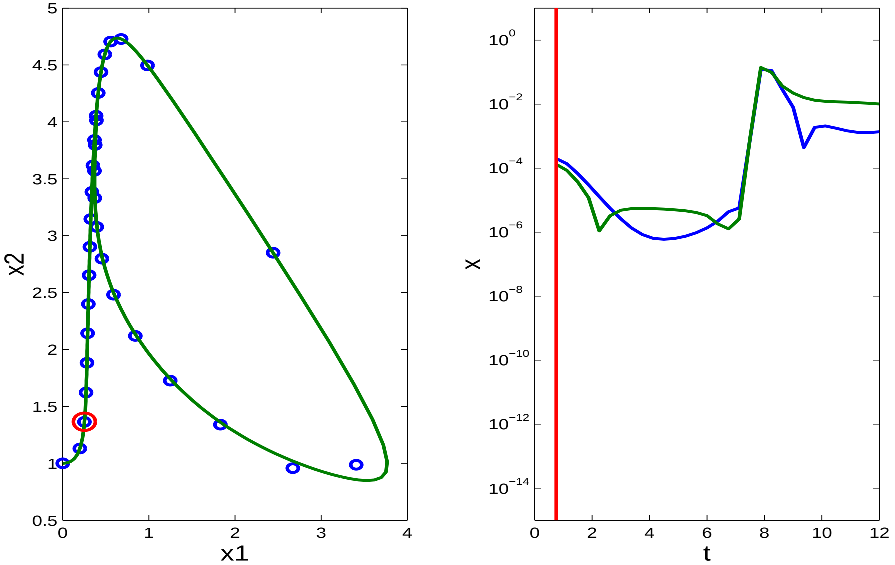
Parameters: $A = 1$ and $B = 3$, $B > A^2 + 1 \Longrightarrow$ limit cycle.

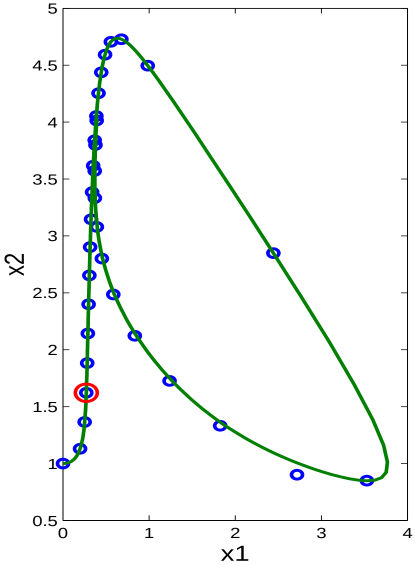
Initial conditions: $x(0) = 0$, $y(0) = 1$.

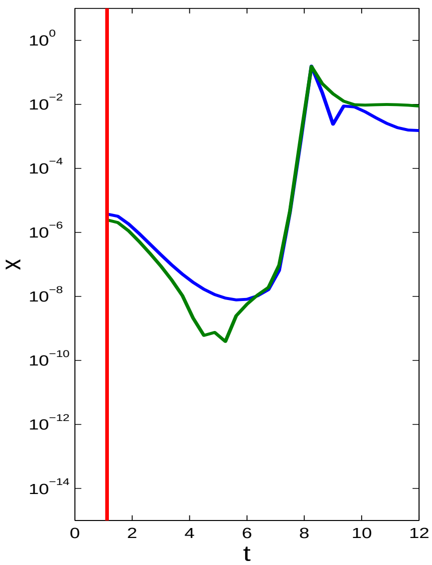
Simulation time: $t \in [0, T = 12]$

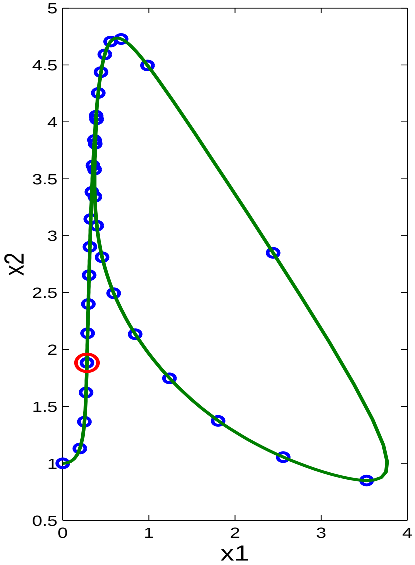
Discretization: Fourth order Runge Kutta, $\Delta T = \frac{T}{32}$, $\Delta t = \frac{T}{320}$.

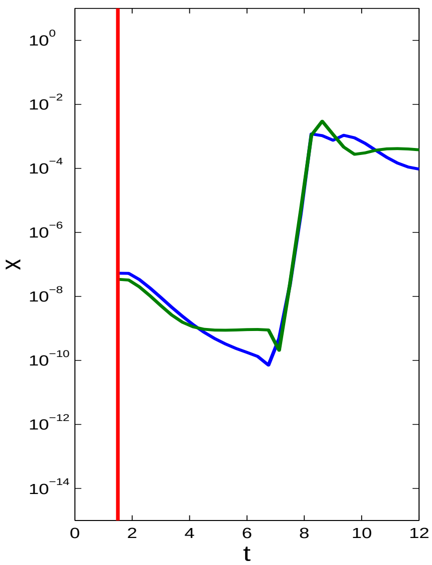

x2
x1
x
t

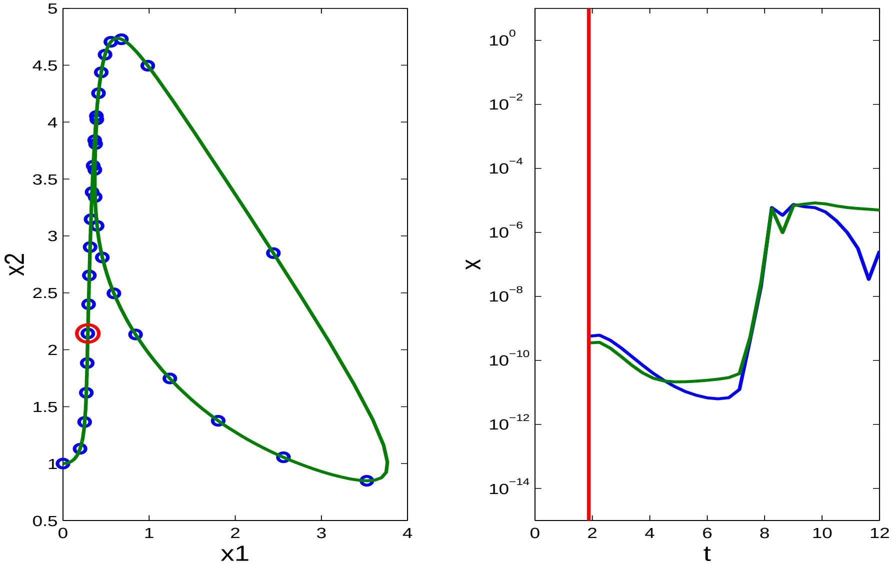

x1
x2
t
x

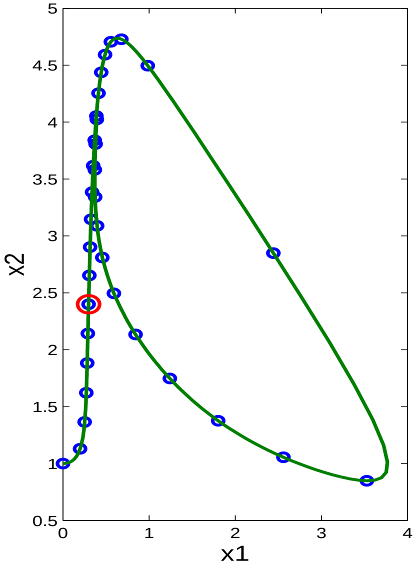

5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
1
2
3
4
x2
x1

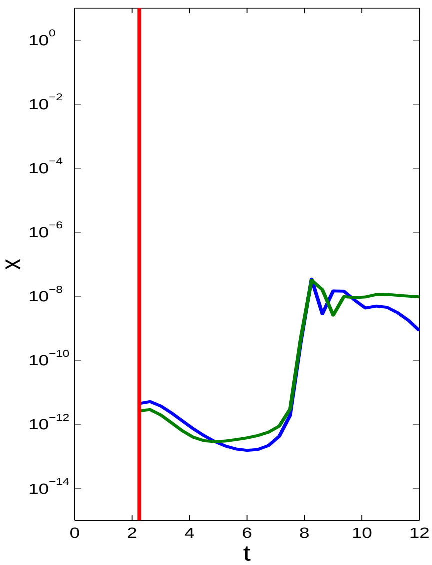

10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
0
2
4
6
8
10
12
x
t

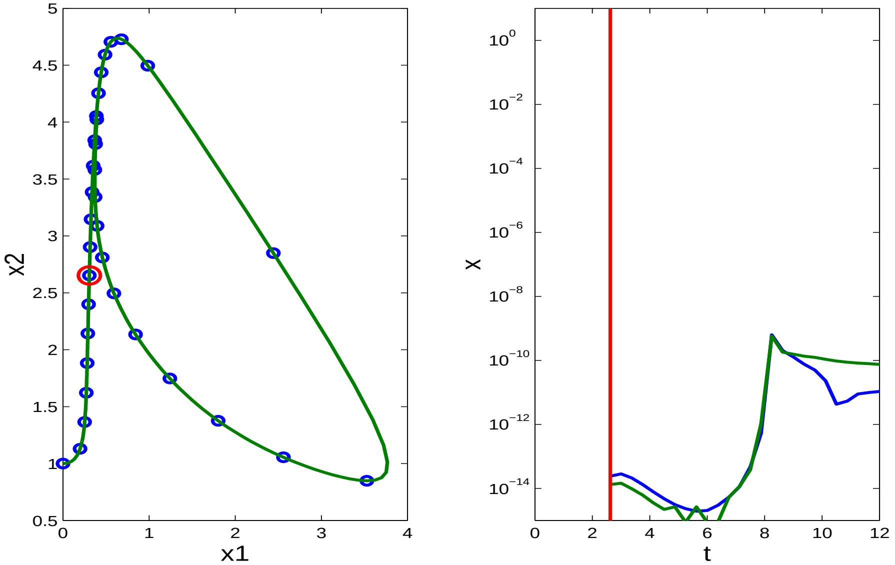

x2
x1
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
1
2
3
4

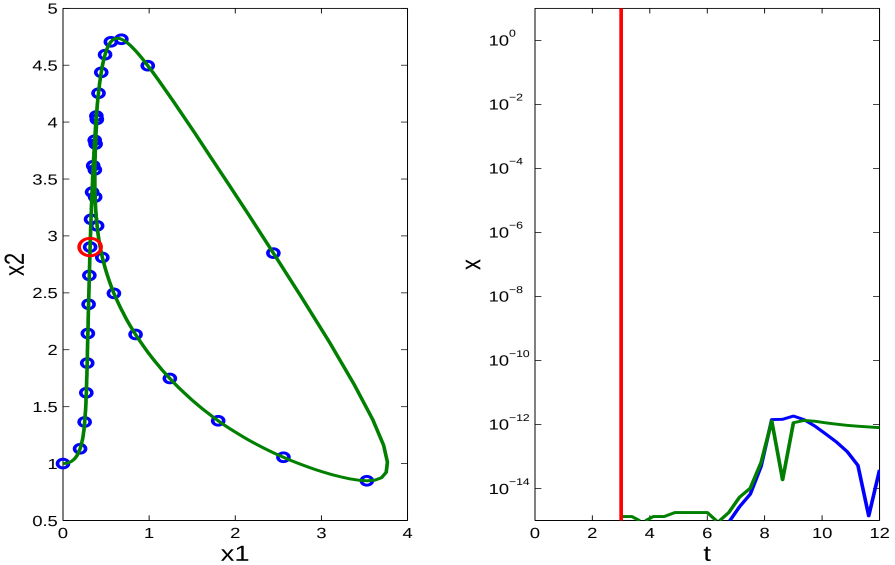

x
t
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
0
2
4
6
8
10
12


x2
x1
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
1
2
3
4
x
t
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
0
2
4
6
8
10
12


5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
1
2
3
4
x2
x1


10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
0
2
4
6
8
10
12
x
t

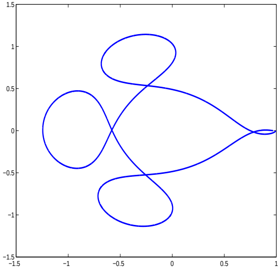

x2
x1
0
1
2
3
4
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
x
t
10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
0
2
4
6
8
10
12


x2
x1
0
1
2
3
4
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
x
t
0
2
4
6
8
10
12
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}

Numerical experiments: Arenstorf orbit

$$\ddot{x} = x + 2\dot{y} - b\frac{x+a}{D_1} - a\frac{x-b}{D_2}$$
$$\ddot{y} = y - 2\dot{x} - b\frac{y}{D_1} - a\frac{y}{D_2},$$

$$D_1 = ((x+a)^2+y^2)^{(3/2)},\ D_2 = ((x-b)^2+y^2)^{(3/2)}$$



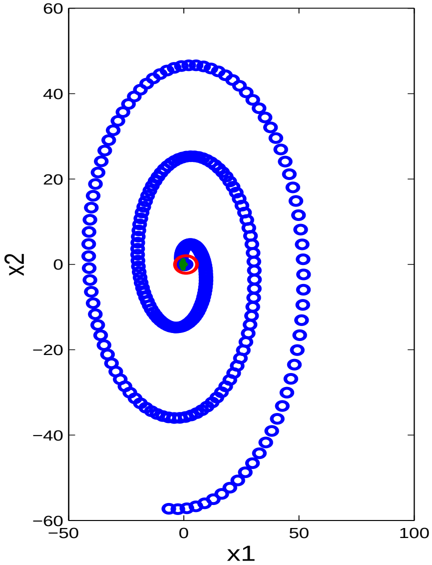
Parameters: $a = 0.012277471,\ b = 1 - a.$

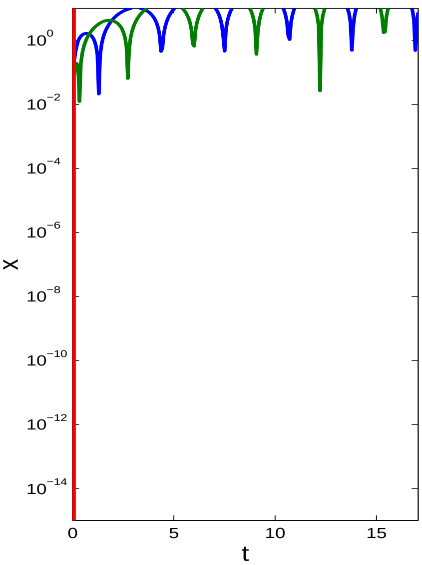
Initial conditions: $x(0) = 0.994,\ \dot{x} = 0,$
$y(0) = 0,\ \dot{y}(0) = -2.00158510637908$

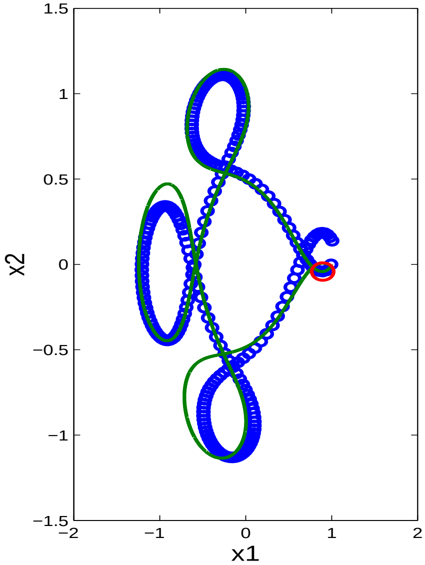
Simulation time: $t \in [0, T = 17.06]$

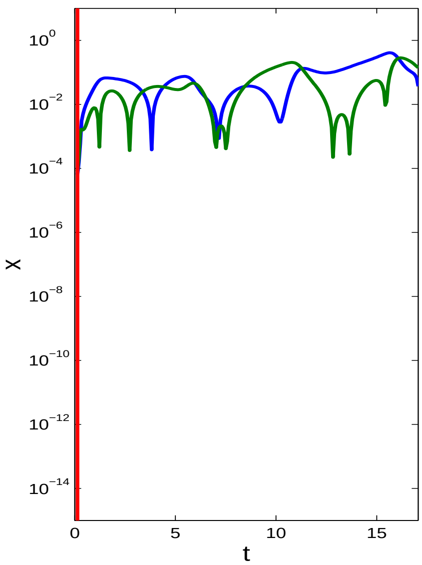
Discretization: Forth order Runge Kutta, $\Delta T = \frac{T}{250},\ \Delta t = \frac{T}{10000}.$

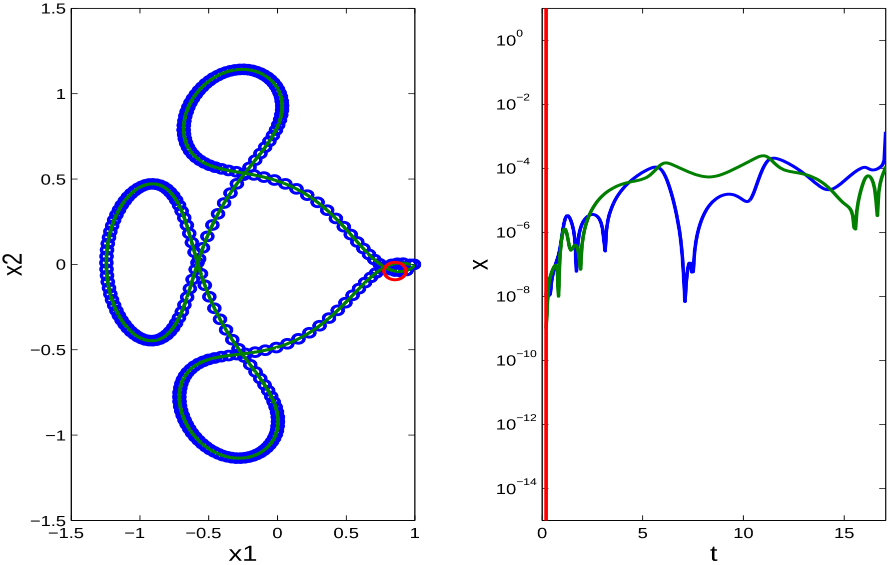
See also Saha, Stadel and Tremaine, a parallel integration method for solar system dynamics, 1997

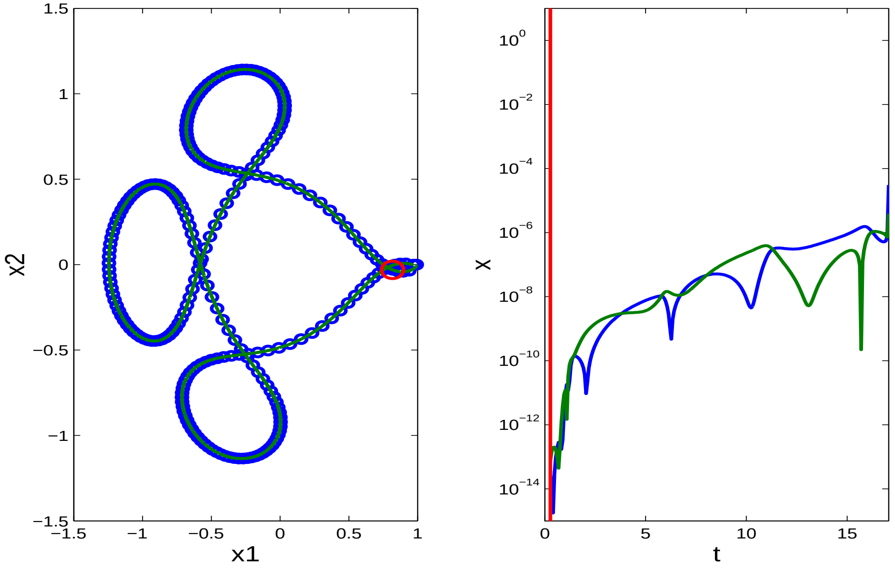

60
40
20
0
−20
−40
−60
−50
0
50
100
x1
x2

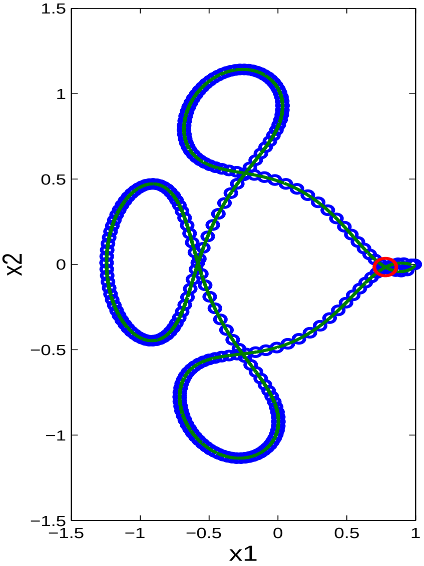

10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
0
5
10
15
t
x

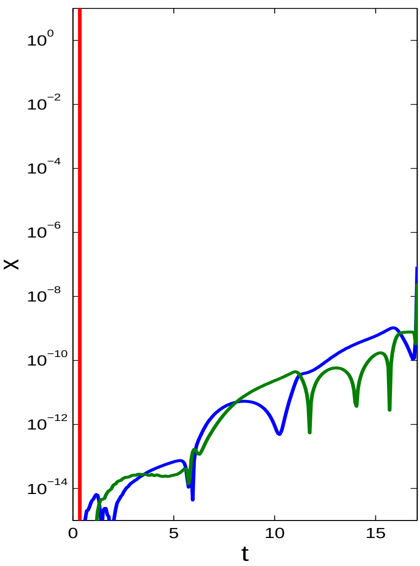

1.5
1
0.5
0
−0.5
−1
−1.5
−2
−1
0
1
2
x2
x1

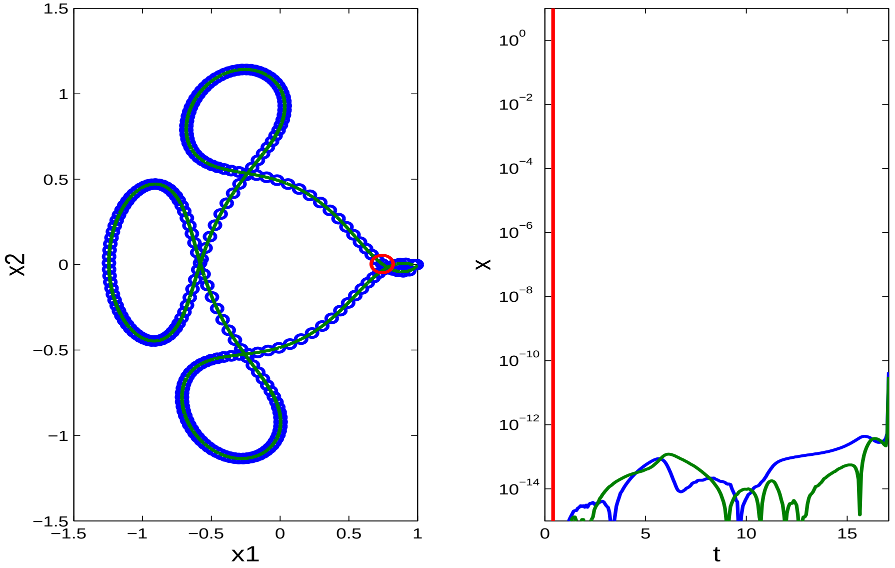

10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
0
5
10
15
x
t


1.5
1
0.5
0
−0.5
−1
−1.5
x2
−1.5
−1
−0.5
0
0.5
1
x1
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
x
0
5
10
15
t


1.5
1
0.5
0
−0.5
−1
−1.5
x2
−1.5
−1
−0.5
0
0.5
1
x1
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
x
0
5
10
15
t

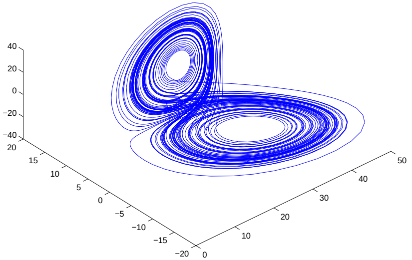

1.5
1
0.5
0
−0.5
−1
−1.5
x2
−1.5
−1
−0.5
0
0.5
1
x1


10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
x
0
5
10
15
t


1.5
1
0.5
0
−0.5
−1
−1.5
x2
−1.5
−1
−0.5
0
0.5
1
x1
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
x
0
5
10
15
t

Results for the Lorenz Equations

$$\begin{aligned}\dot{x} &= -\sigma x + \sigma y\\ \dot{y} &= -xz + rx - y\\ \dot{z} &= xy - bz\end{aligned}$$

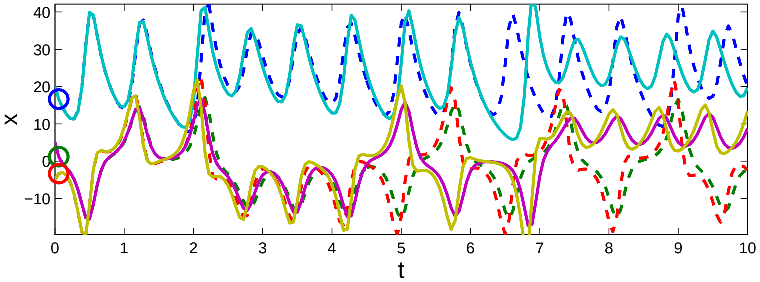


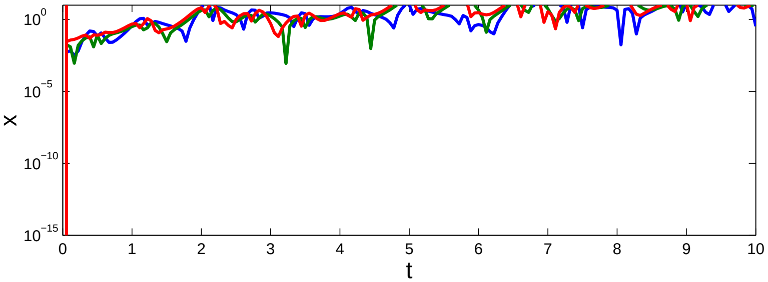
Parameters: $\sigma = 10$, $r = 28$ and $b = \frac{8}{3} \Longrightarrow$ chaotic regime.

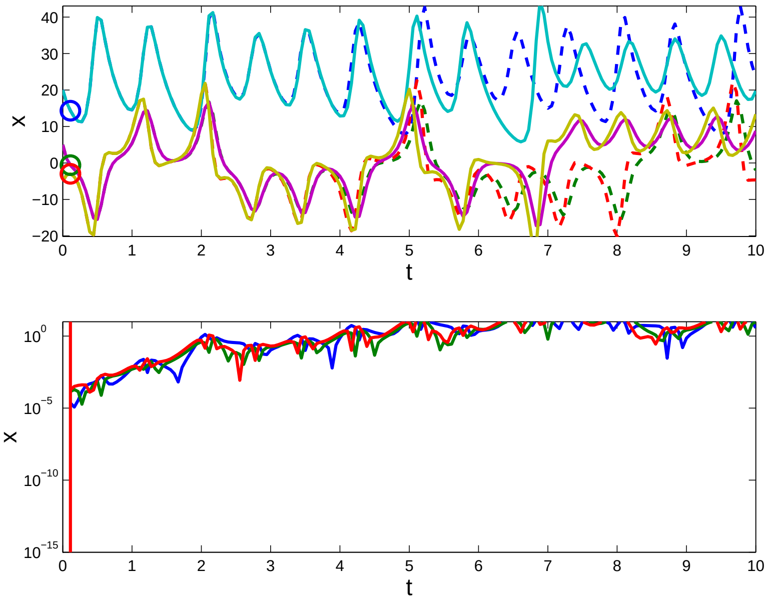
Initial conditions: $(x, y, z)(0) = (20, 5, -5)$

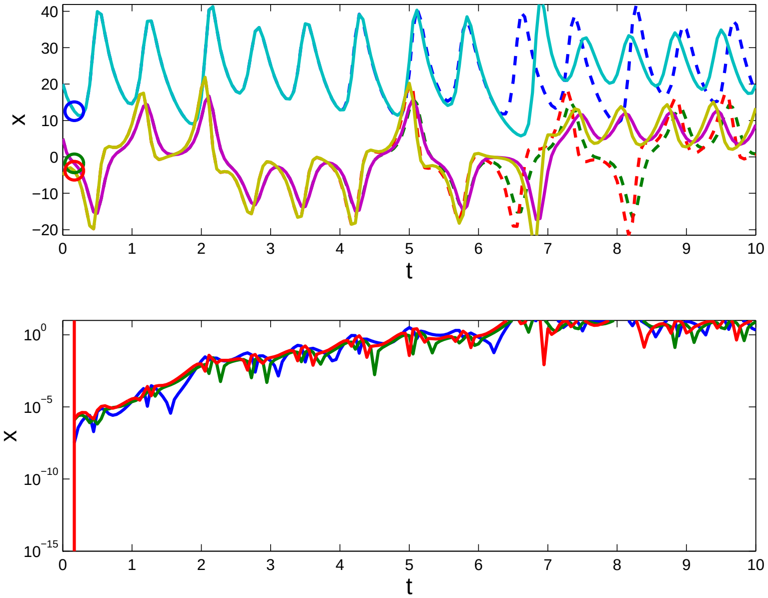
Simulation time: $t \in [0, T = 10]$

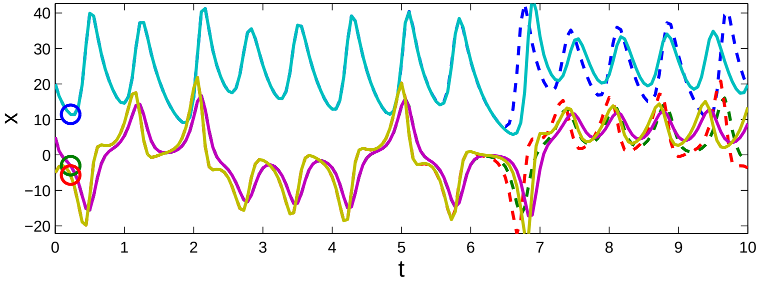
Discretization: Fourth order Runge Kutta, $\Delta T = \frac{T}{180}$, $\Delta t = \frac{T}{1800}$.

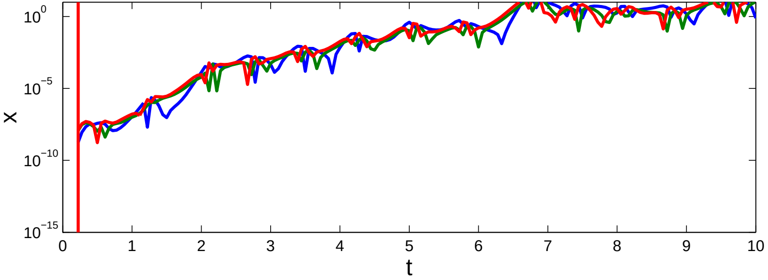

40
30
20
10
0
−10
x
0
1
2
3
4
5
6
7
8
9
10
t

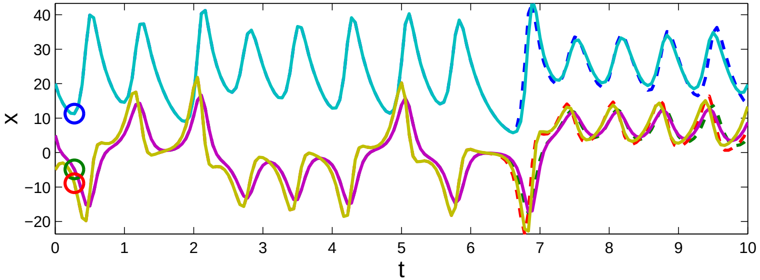

10^{0}
10^{-5}
10^{-10}
10^{-15}
x
0
1
2
3
4
5
6
7
8
9
10
t

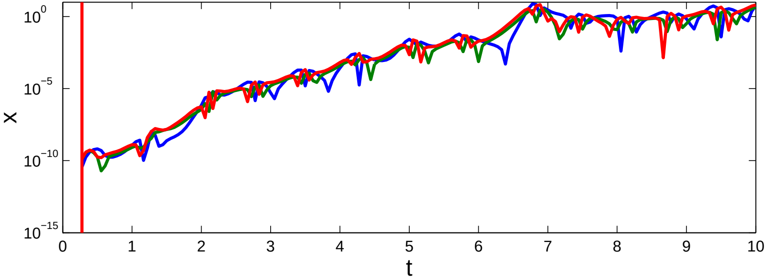

x
40
30
20
10
0
−10
−20
0 1 2 3 4 5 6 7 8 9 10
t
x
10^0
10^−5
10^−10
10^−15
0 1 2 3 4 5 6 7 8 9 10
t

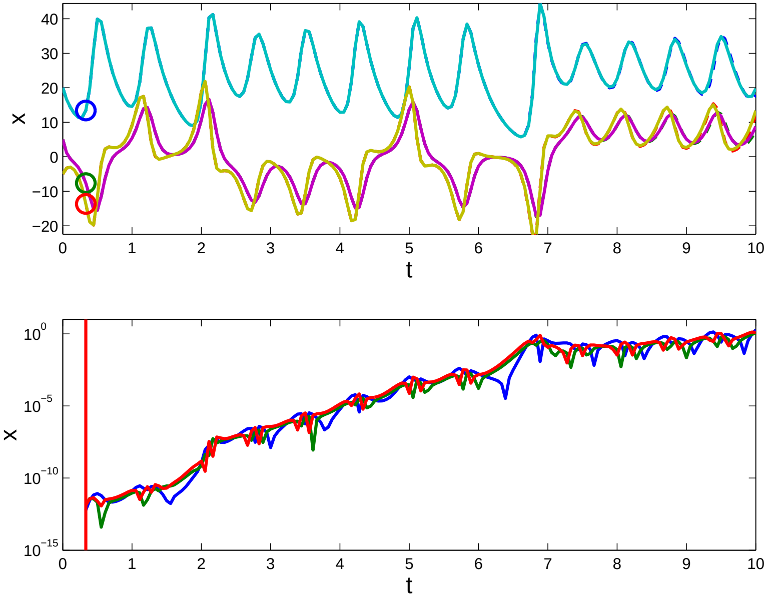

x
t
x
t

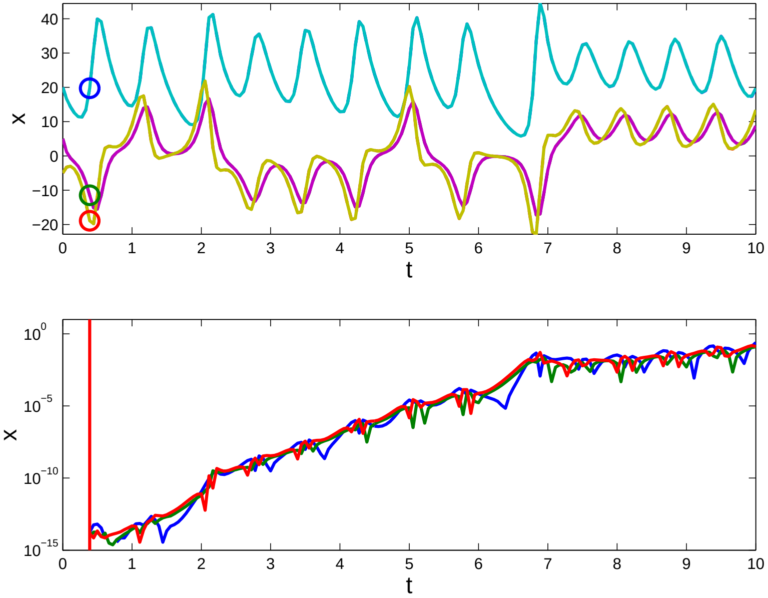

x
t
40
30
20
10
0
−10
−20
0
1
2
3
4
5
6
7
8
9
10

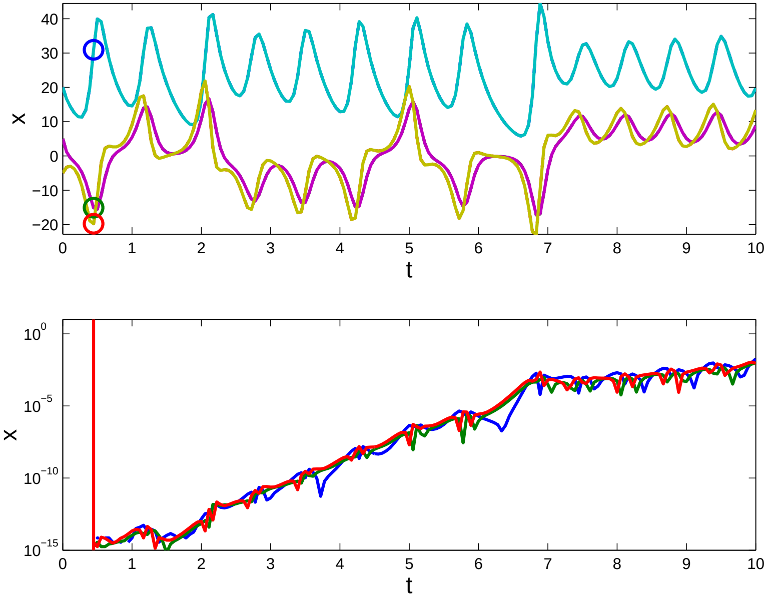

x
t
10^{0}
10^{-5}
10^{-10}
10^{-15}
0
1
2
3
4
5
6
7
8
9
10

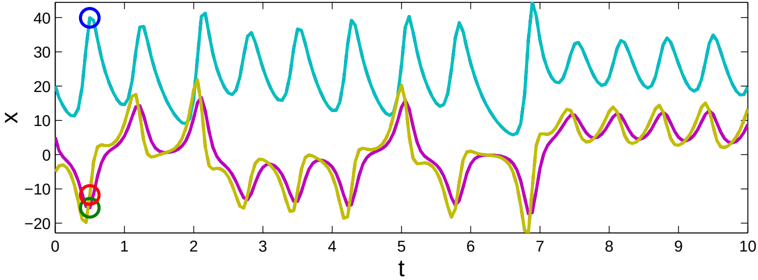

40
30
20
10
0
−10
−20
x
0
1
2
3
4
5
6
7
8
9
10
t

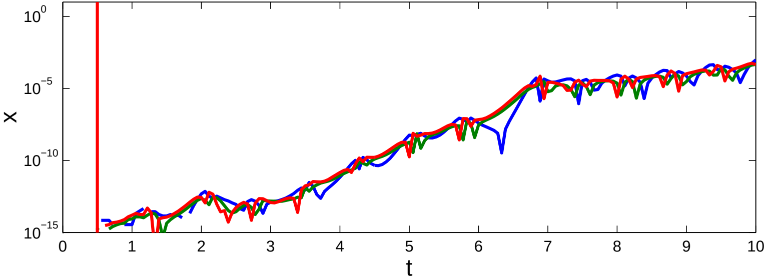

10^{0}
10^{-5}
10^{-10}
10^{-15}
x
0
1
2
3
4
5
6
7
8
9
10
t

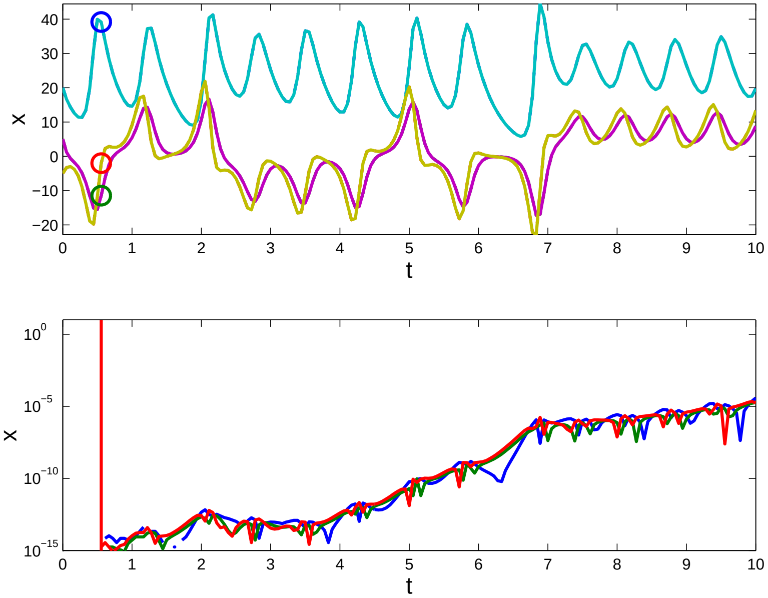

40
30
20
10
0
−10
−20
x
0 1 2 3 4 5 6 7 8 9 10
t
10^{0}
10^{-5}
10^{-10}
10^{-15}
x
0 1 2 3 4 5 6 7 8 9 10
t

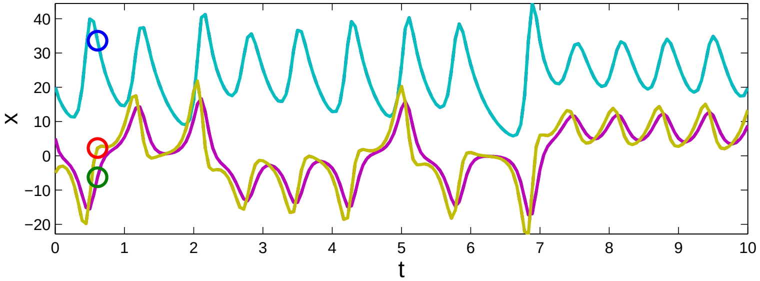

x
t
0
1
2
3
4
5
6
7
8
9
10
40
30
20
10
0
−10
−20
x
t
10^{0}
10^{-5}
10^{-10}
10^{-15}

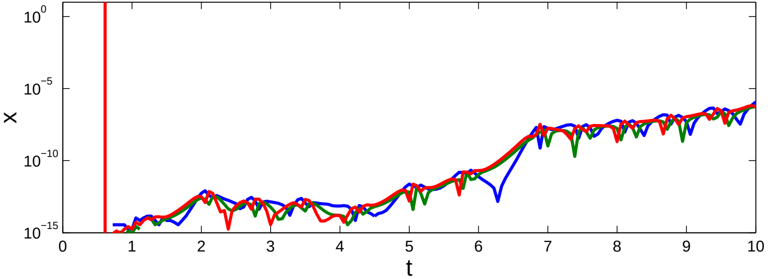

40
30
20
10
0
−10
−20
x
0 1 2 3 4 5 6 7 8 9 10
t
10^{0}
10^{-5}
10^{-10}
10^{-15}
x
0 1 2 3 4 5 6 7 8 9 10
t

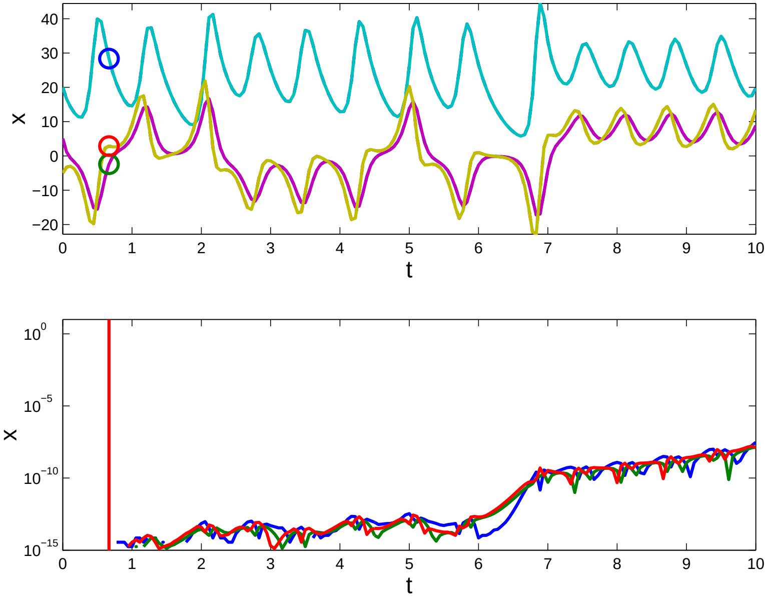

40
30
20
10
0
−10
−20
x
0
1
2
3
4
5
6
7
8
9
10
t

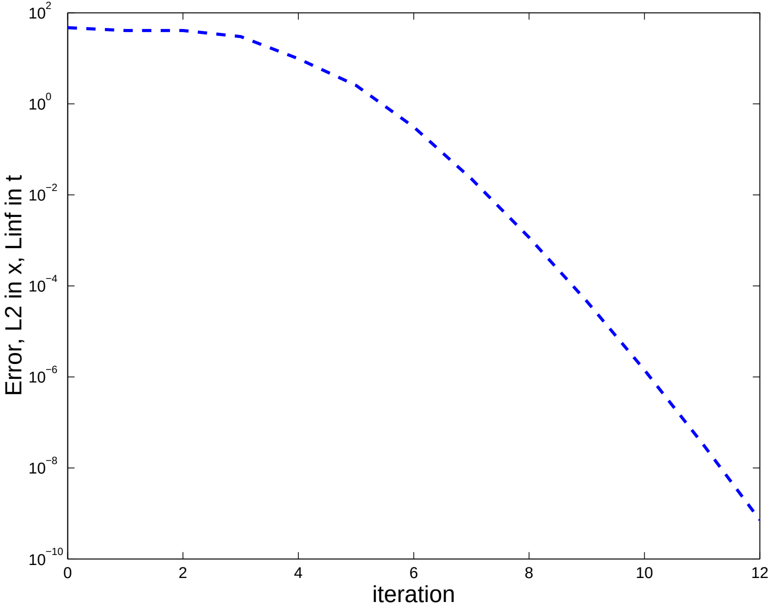

10^{0}
10^{-5}
10^{-10}
10^{-15}
x
0
1
2
3
4
5
6
7
8
9
10
t


x
t
40
30
20
10
0
−10
−20
0
1
2
3
4
5
6
7
8
9
10
10^0
10^{-5}
10^{-10}
10^{-15}


40
30
20
10
0
−10
−20
x
0
1
2
3
4
5
6
7
8
9
10
t


10^{0}
10^{-5}
10^{-10}
10^{-15}
x
0
1
2
3
4
5
6
7
8
9
10
t

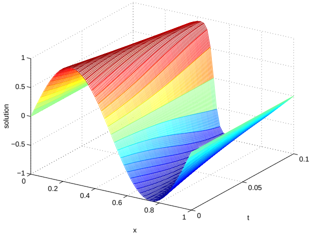

40
30
20
10
0
−10
−20
x
0
1
2
3
4
5
6
7
8
9
10
t
10^{0}
10^{-5}
10^{-10}
10^{-15}
x
t

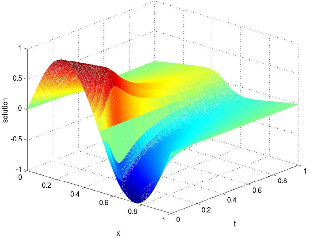

Error, L2 in x, Linf in t
iteration
10^2
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
0
2
4
6
8
10
12

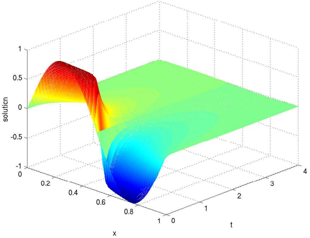
Numerical experiments for PDEs: Burgers equation

$$\begin{aligned} u_t + uu_x &= \nu u_{xx} \quad \text{in } \Omega = [0,1] \\ u(x,0) &= \sin(2\pi x) \end{aligned}$$

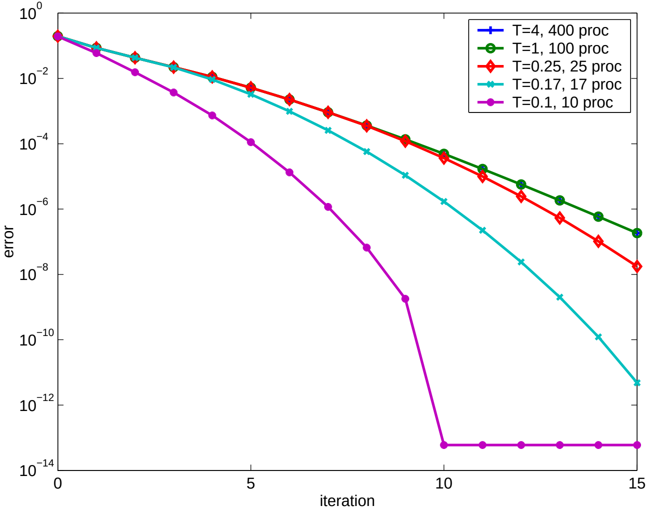
Viscosity $\nu = \frac{1}{50}$, homogeneous boundary conditions
Centered finite difference discretization, $\Delta x = \frac{1}{50}$
Backward Euler in time, $\Delta T = \frac{1}{10}$, $\Delta t = \frac{1}{100}$.







Burgers equation: convergence behavior



Further Applications and Results on Parareal

- **Oscillatory Problems:** Cortial, Farhat, Chandesris (2003, 2006)
- **Control of Quantum Systems:** Maday, Salomon, Turinici (2002, 2006)
- **Reservoir Simulation:** Garrido, Espedal, Fladmark (2003, 2005)
- **Navier-Stokes:** Fischer, Hecht, Maday (2003)
- **Stability Analysis:** Staff and Rønquist (2003)
- **Molecular Dynamics:** Baffico, Bernard, Maday, Turinici, Zerah (2002)
- **Finance:** Bal, Maday (2002)

Google hits for parareal algorithm (6.5.2007): 433

Conclusions

Parallel speedup in time is possible, but the speedup is more modest than in space.

Further results:

- Two multilevel versions of the algorithm.
- Understand hyperbolic case with boundary conditions, and extension by Farhat for general hyperbolic problems.

Future work:

- Analysis of Parareal for DAEs.
- Preservation of symplectic structure in Parareal.