Parareal methods for the evolution equations and control

Gabriel Turinici

Inria Rocquencourt and CERMICS-ENPC

+ J.L.Lions, Y.Maday

+ G.Bal

+ L. Baffico, G. Zerah, S. Bernard

- Goal: parallelize in the time direction the simulation of evolution PDE or control.

- Framework: we suppose that enough CPU power (i.e. many processors) is available and that the only relevant quantity is the physical time (= "wall clock time") required to solve the given equations with predefined accuracy.

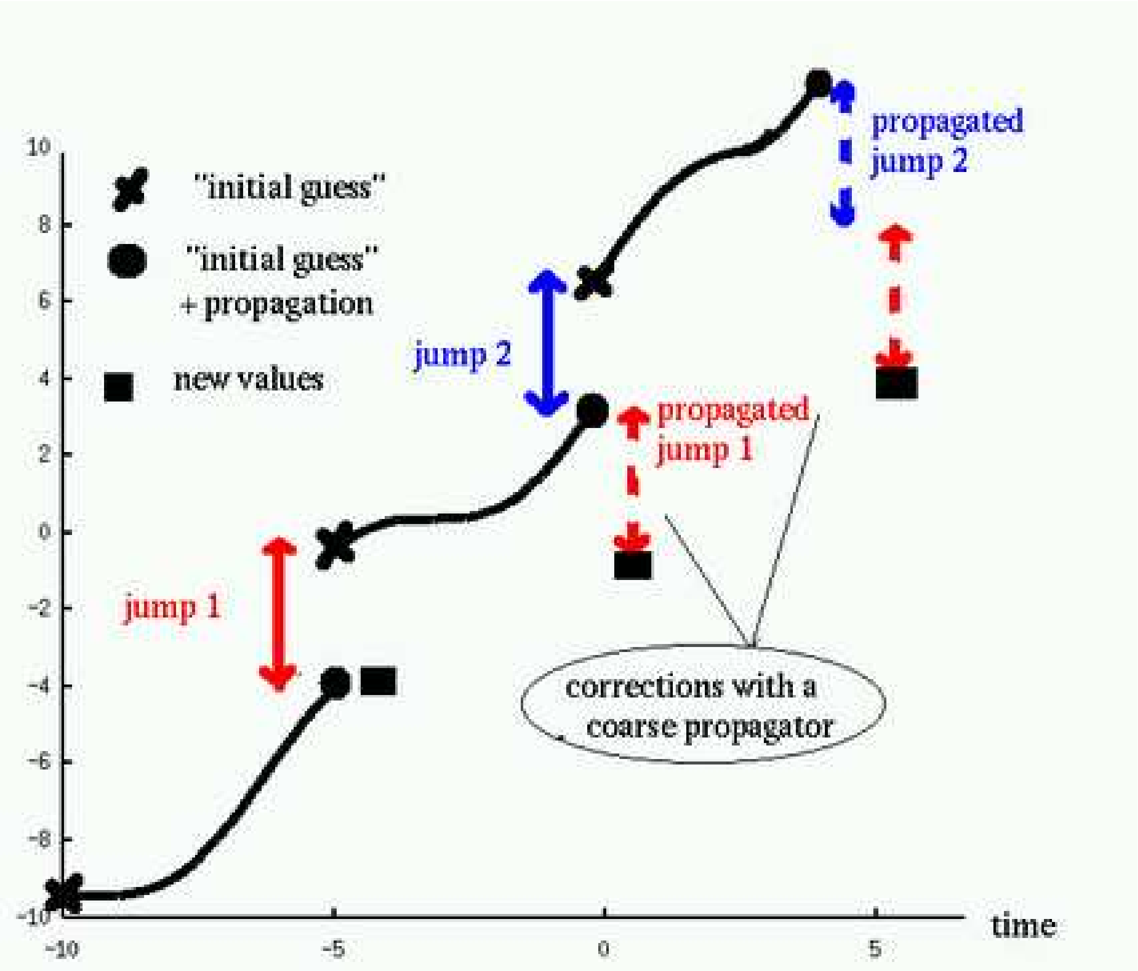
- Target applications: real time simulation of complex systems

▷ *J.-L. Lions, Y.Maday & G.T. A "parareal" in time discretization of PDEs, C.R.A.S. Sér. I, 332 (2001), 7, pp 661-668*

▷ *G.Bal & Y.Maday : A "parareal" in time discretization for nonlinear PDE with application to the pricing of an American put , Proc. of the workshop on Domain Decomposition, Zurich, Switzerland, LNCSE Series, Springer 2001*

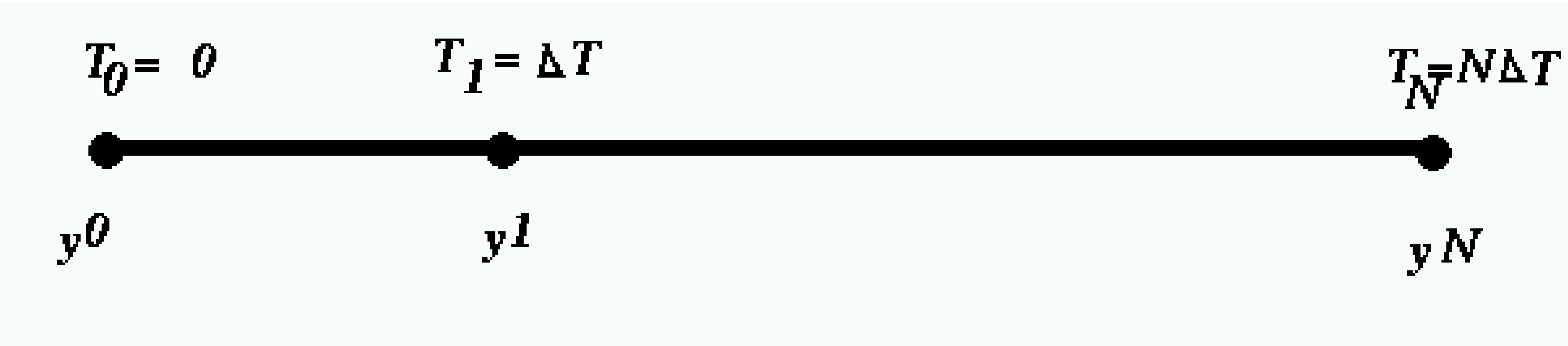
▷ *Y.Maday & G.T.: A "parareal" in time procedure for the control of partial differential equations, C.R.A.S. 2002, to appear.*

Principle of the parareal scheme



Parareal scheme formulation

Equation to solve: $\begin{cases} \frac{\partial y}{\partial t} + Ay = f,\, t \in [0, T] \\ y(0) = y_0 \end{cases}$



Exact solution: y^n is the exact propagation from y^{n-1}, n=1,...N-1.

To be found: y^1,...,y^{N-1} , then solution is obtained by parallel propagation on subintervals.

Parareal scheme: linear operator version, implicit

Iteration 0: **Coarse Euler scheme:**

$$\begin{cases} \frac{Y_0^{n+1} - Y_0^n}{\Delta T} + AY_0^{n+1} = 0 \\ Y^0 = y_0 \end{cases}$$

Propagation in parralel

$$\begin{cases} \frac{\partial y_0^n}{\partial t} + Ay_0^n = 0 \\ y_0^n(t = T^n) = Y_0^n \end{cases}$$

Iteration $k \geq 1$:

Computation of the jumps $S_k^n = y_k^{n-1}(T^n) - Y_k^n$

Propagation of the jumps:

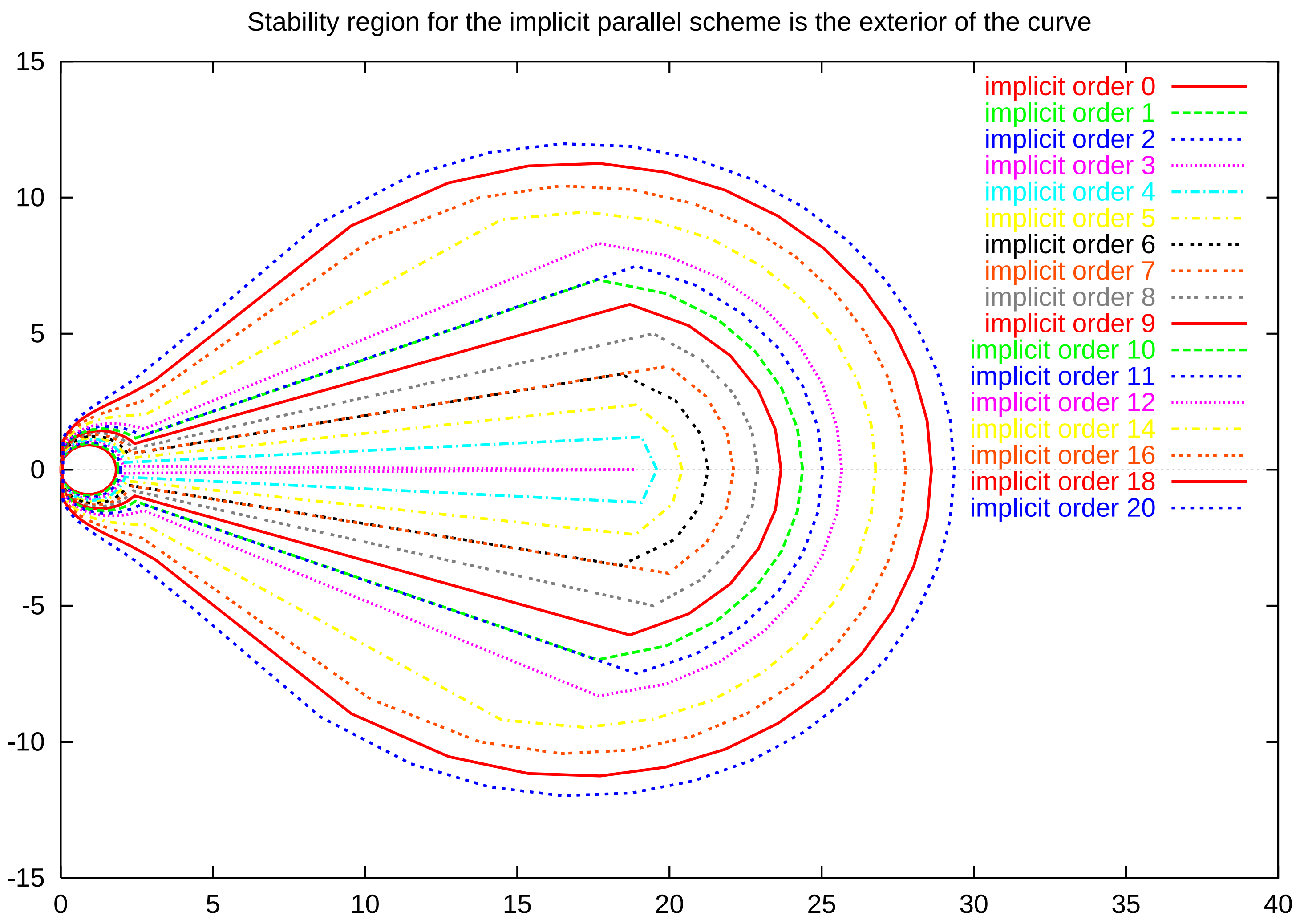
$$\begin{cases} \frac{\delta_k^{n+1}-\delta_k^n}{\Delta T} + A\delta_k^{n+1} = \frac{S_k^n}{\Delta T} \\ \delta_k^0 = 0 \end{cases}$$

Corrections $Y_{k+1}^n = y_k^{n-1}(T^n) + \delta_k^n$

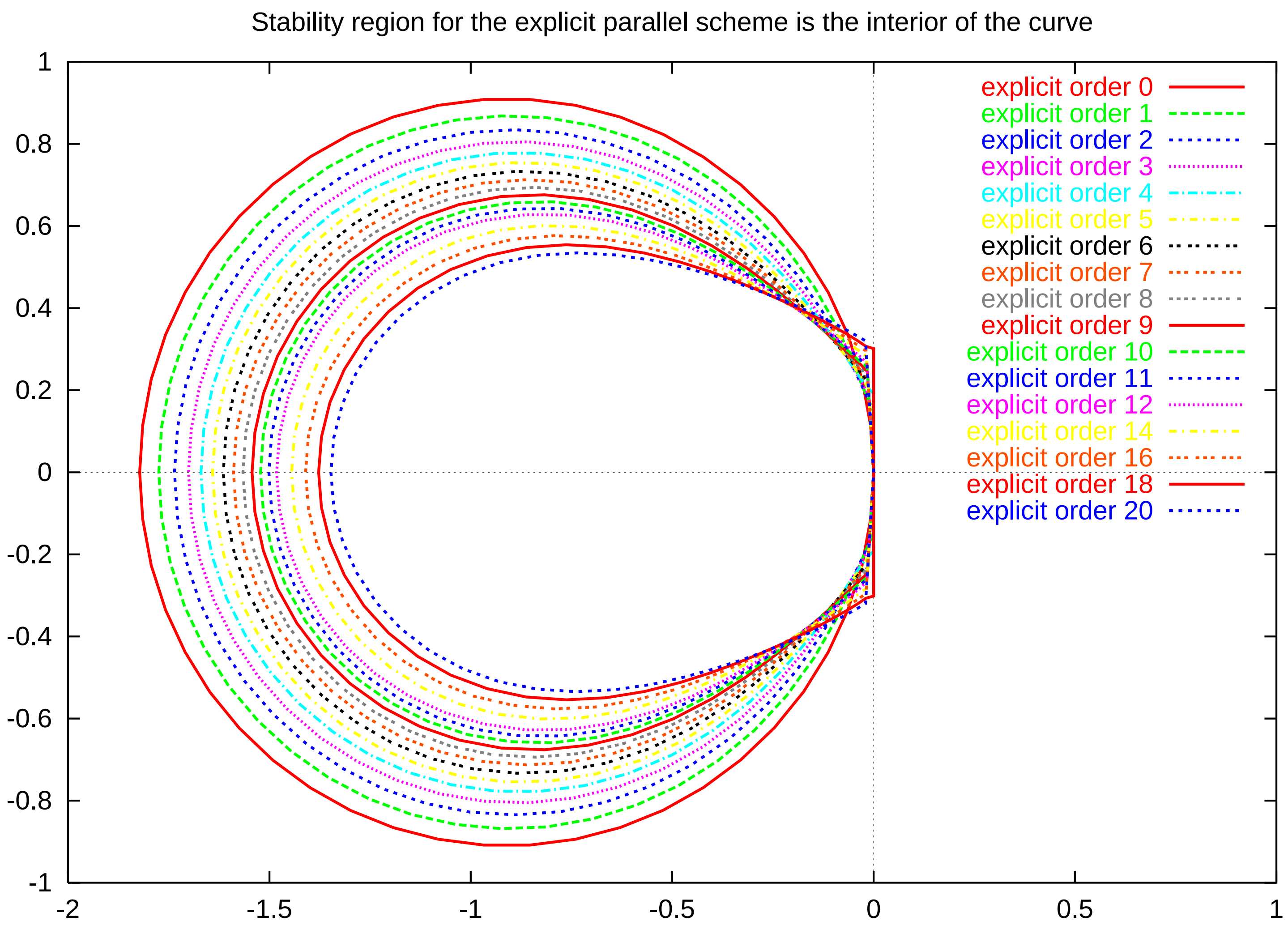
Propagations in parallel

$$\begin{cases} \frac{\partial y_{k+1}^n}{\partial t} + Ay_{k+1}^n = 0 \\ y_{k+1}^n(t = T^n) = Y_{k+1}^n \end{cases}$$

Theorem: $|Y_k^n - y(T^n)| + \max_{t\in[T^n,T^{n+1}]} |y_k^n - y(t)| \leq c(\Delta T)^k$

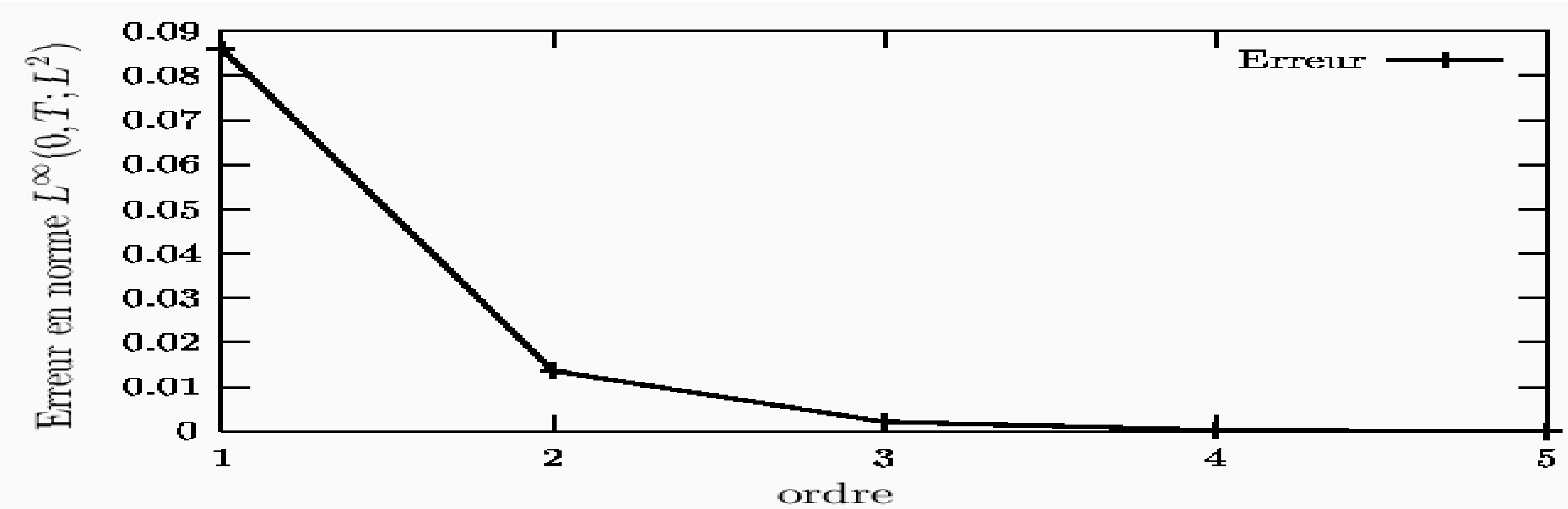


Stability region for the implicit parallel scheme is the exterior of the curve
15
10
5
0
-5
-10
-15
0
5
10
15
20
25
30
35
40
implicit order 0
implicit order 1
implicit order 2
implicit order 3
implicit order 4
implicit order 5
implicit order 6
implicit order 7
implicit order 8
implicit order 9
implicit order 10
implicit order 11
implicit order 12
implicit order 14
implicit order 16
implicit order 18
implicit order 20



Stability region for the explicit parallel scheme is the interior of the curve
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-2
-1.5
-1
-0.5
0
0.5
1
explicit order 0
explicit order 1
explicit order 2
explicit order 3
explicit order 4
explicit order 5
explicit order 6
explicit order 7
explicit order 8
explicit order 9
explicit order 10
explicit order 11
explicit order 12
explicit order 14
explicit order 16
explicit order 18
explicit order 20

Numerical results $\begin{cases} \frac{\partial u}{\partial t} - \Delta u = A\sin(\alpha t) \ on \ B(0,1) \subset R^2 \\ u(t=0) = \tan(x^2+y^2-1) \end{cases}$

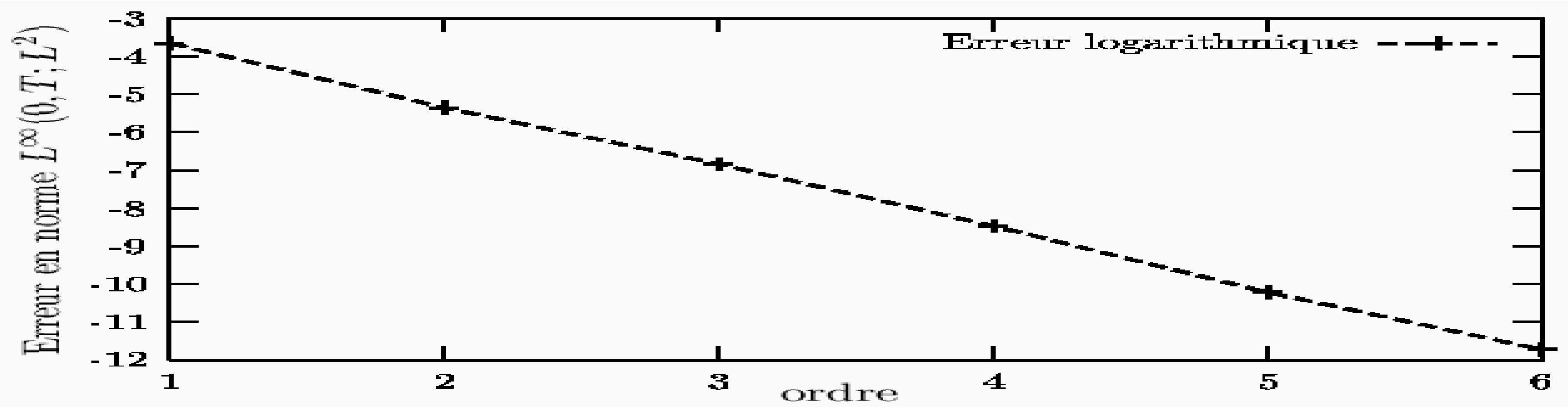


$$T = 50,\ \Delta T = 0.1,\ \delta t = 0.004,\ A = 5,\ \alpha = 10$$

Error of the sequential resolution $\delta t = 0.6 \cdot 10^{-2}$ is recovered by parareal scheme at iteration $k = 3$.

Numerical results $\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} - \lambda \Delta u + \beta u^3 = A \sin(\alpha t) \ on \ B(0,1) \subset R^2 \\ u(t=0) = \tan(x^2+y^2-1) \end{cases}$

Jump propagation by linearisation $\begin{cases} \frac{\delta_k^{n+1} - \delta_k^n}{\Delta T} + \frac{\partial \delta_k^{n+1}}{\partial x} - \lambda \Delta \delta_k^{n+1} + 3\beta(y_k^{n+1}(T^n)^2 \delta_k^{n+1} = \frac{S_k^n}{\Delta T} \\ \delta_k^0 = 0 \end{cases}$



$$T = 50,\ \Delta T = 0.1,\ \delta t = 0.004,\ A = 5,\ \alpha = 10$$

Error of the sequential resolution is recovered by parareal scheme at iteration $k = 4$.(speed-up =18)

Parareal scheme: non-linear operator version

Implicit propagation schemes: $\begin{array}{c} G = (I + A \cdot \Delta T)^{-1} \, coarse\, propagator \\ F = (I + A \cdot \delta T)^{-n} \, fine\, propagator \end{array}$

Jump evolution: $\begin{cases} \delta_k^{n+1} = G(\delta_k^n + S_k^n)\;(*) \\ S_k^n = FY_k^{n-1} - Y_k^n \end{cases}$

Update: $Y_{k+1}^n = FY_k^{n-1} - \delta_k^n$

Rewrite jump evolution (*): $Y_{k+1}^{n+1} - FY_k^n = G(Y_{k+1}^n - FY_k^{n-1} + FY_k^{n-1} - Y_k^n)$

$$Y_{k+1}^{n+1} = GY_{k+1}^n + FY_k^n - GY_k^n$$

No properties of A are used any more (differentiability-cf. pricing of an American put-, linearity ..)

Parareal scheme applied to control

Equations for a solution (evolution): $Y^{n+1} = FY^n$

$$\underbrace{\begin{pmatrix} I & & & 0 \\ -F & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -F & I \end{pmatrix}}_{M} \underbrace{\begin{pmatrix} Y^0 \\ \vdots \\ \vdots \\ Y^N \end{pmatrix}}_{Y} = \underbrace{\begin{pmatrix} y_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{b}$$

Parareal scheme: $Y_k^{n+1} = GY_{k+1}^n + FY_k^n - GY_k^n$

$$\begin{pmatrix} I & & & 0 \\ -G & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -G & I \end{pmatrix} Y^{k+1} = \begin{pmatrix} 0 & & & 0 \\ F-G & \ddots & & \\ & \ddots & \ddots & \\ 0 & & F-G & 0 \end{pmatrix} Y^k + b \ ;$$

denote $\tilde{M} = \begin{pmatrix} I & & & 0 \\ -G & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -G & I \end{pmatrix}$

$\tilde{M}Y^{k+1} = (M - \tilde{M})Y^k + b \Leftrightarrow \tilde{M}Y^{k+1} = \tilde{M}Y^k + (b - MY^k)$;

denote $Res^k = b - MY^k$

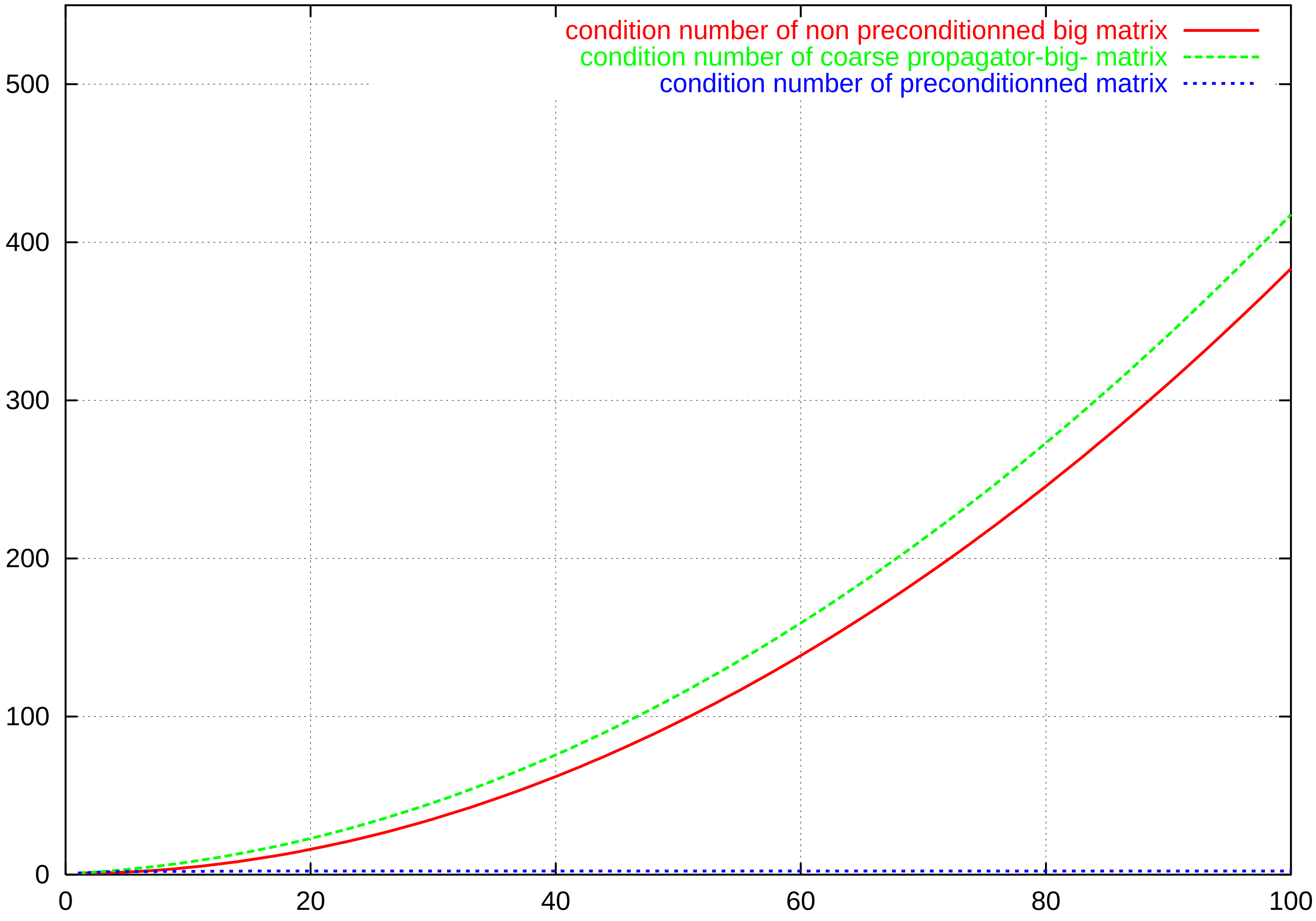
$$Y^{k+1} = Y^k + \tilde{M}^{-1}Res^k \ : \ \text{preconditioning}$$

Parareal control schemes

dynamical equations: $\begin{cases} \frac{\partial y}{\partial t} + Ay = Bv \\ y(t = 0) = y_0 \end{cases}$

target $y(T) = y_T$

—cf control note ...—


condition number of non preconditionned big matrix
condition number of coarse propagator-big- matrix
condition number of preconditionned matrix
500
400
300
200
100
0
0
20
40
60
80
100