MAP551 - PC4: Numerical Integration of Ordinary Differential Equation (Part II)

Te SUN

October 2018

1 Stability diagrams

If we compare the four diagram (1) for the four Runge-Kutta methods of order 1, 2, 3 and 4, we can see the surface of the stable domain is increasing with the order, which means that the scheme has better stability propriety : easier to converge.

We can see clearly that the stability diagram of fourth order method admit z with positive real part number. However, when the real part of the coefficient before the term u^n leads to an absence of **contraction mapping**. That cause **an explosive expansion of the analytic solution**. Although we have lost the stability of the real solution, we can still maintain the stability of numerical resolution. Therefore, we might lose the accuracy when applying this method to the zone where analytic solution explodes (exponentially). (**2.1**)

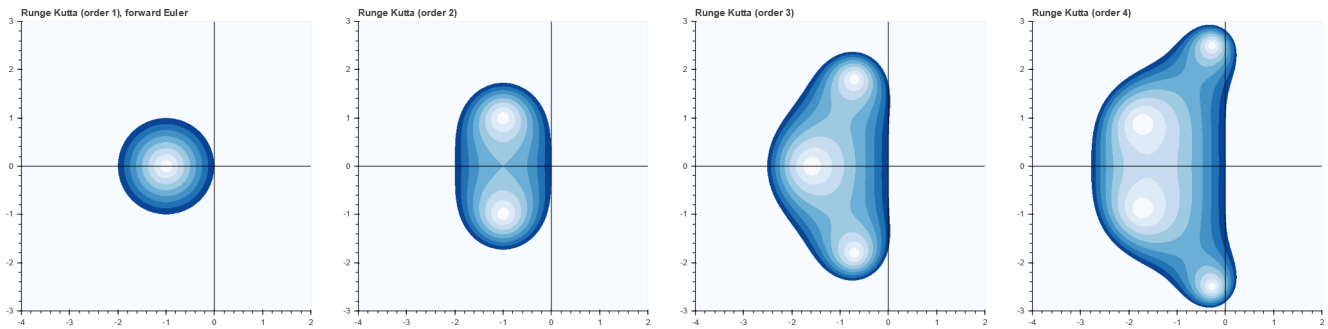
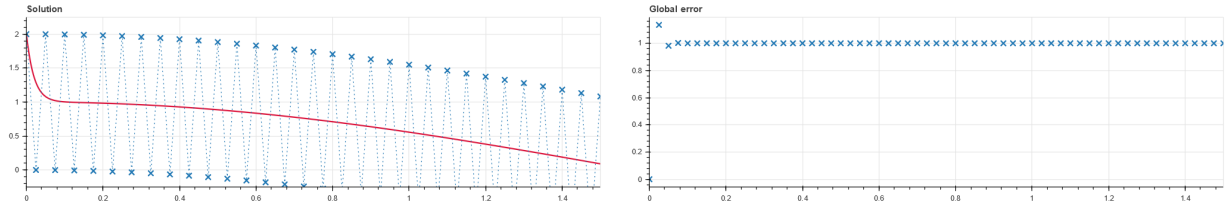


Figure 1 – Stability domain with different order of the method Runge-Kutta

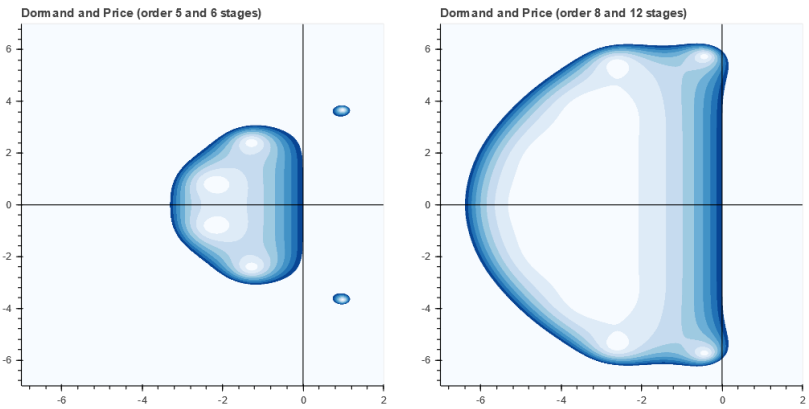
We can use Euler explicit method to study this phenomenon : When the time step chosen close to the boundary of stability domain, we still have a (very slow) convergence in long term. The stability then can not guarantee an accurate integration of the system.(**2.2**)



Numerical solution Global Error

Figure 2 – Explicit method solving Curtiss and Hirschfelder model with $\Delta t = 0.024$ and $K = 80$

Compared with RK methods (Fig 1, Fig 3), the DOPRI methods extend the stability domain to -6 (real part), yet, the RK method with fourth order supplies us with a stability domain from -3 to 0.2 (real part). **As a result, with higher order and more complex stages, the numerical resolution has a stronger stability propriety**. (**2.3**)



DOPRI with 5^{th} order, 6 stages DOPRI with 8^{th} order, 12 stages

Figure 3 – Explicit method solving Curtiss and Hirschfelder model with $\Delta t = 0.024$ and $K = 80$

2 High order RK methods for BZ reaction dynamics integration

The dynamics of the oscillating reaction discovered by Belousov and Zhabotinsky, can be modeled through the so-called Brusselator model depending on two parameters :

$$\begin{cases} \dfrac{dy_1}{dt} = 1 - (b+1)y_1 + ay_1^2 y_2 \\ \dfrac{dy_2}{dt} = by_1 - ay_1^2 y_2 \end{cases}$$

For the purpose of illustrating the concepts, we use $a = 1$, $b = 3$, $y_1(0) = 1.5$ and $y_2(0) = 3$. The interval to be studied is set $[0, 20]$.

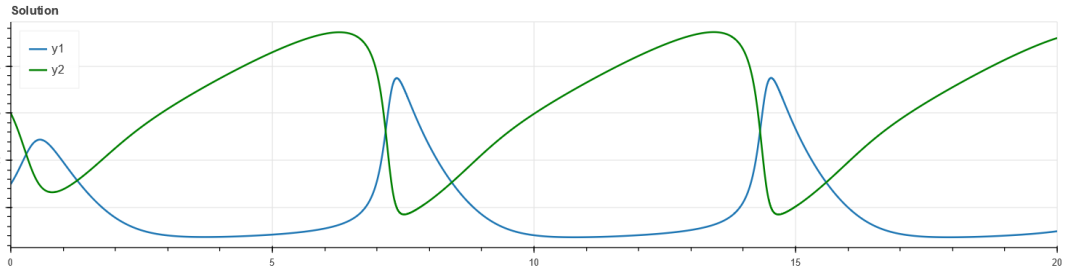


Figure 4 – Quasi-exact solution with RK of order 5

2.1 Stiffness ?

The system could be called as quasi-solution since the its high order of convergence to the real solution, which offers a relative high accuracy. Here, we take order 5 with controlled step size.

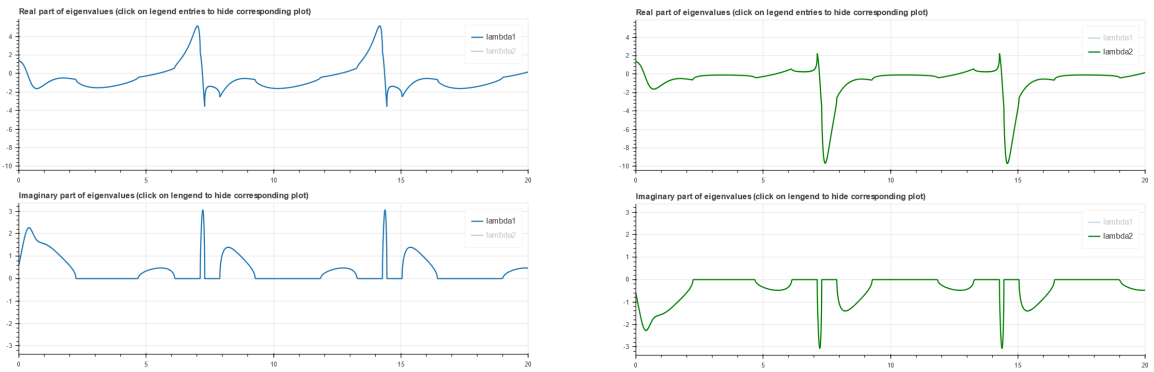
We take $f(y_1, y_2; t) = (\frac{dy_1}{dt}, \frac{dy_2}{dt})$, the Jacobian matrix of function f can be easily calculated as :

$$J = \begin{bmatrix} 2ay_1y_2 - (b+1) & ay_1^2 \\ b - 2ay_1y_2 & -ay_1^2 \end{bmatrix}$$

Therefore, the eigenvalues are solutions to the quadratic equation :

$$x^2 - (2ay_1y_2 - b - 1 - ay_1^2)x + ay_1^2 = 0$$

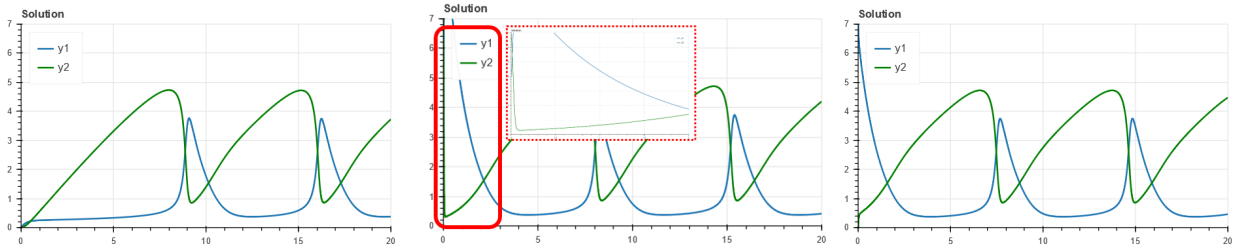
The system becomes stiff when $|Re(\lambda_i)|$ gets large. As we can see from Fig 4 and Fig 5, it is when y_1 and y_2 changes rapidly, the system takes on its stiffness. The order of magnitude in terms of variation of the dynamics lies essentially to λ_1 and λ_2, the variation has a same order of $\exp(\lambda_1 y_1)$ and $\exp(\lambda_2 y_2)$. (**3.1.1**)



λ_1, real part and imaginary part λ_2

Figure 5 – Representation of two eigenvalues of Jacobian matrix

When we try to play with different initial datum (Fig 6) for numerical resolution, we can observe that **on the limit cycle the stiffness preserves its own form as before**. But the initial time dynamics do change, and the time to reach the limit cycle varies as well. As for the case where $y_1(0), y_2(0) = 4, 7$, the system itself becomes extremely stiff at the beginning.(**3.1.2**)



$y_1(0), y_2(0) = 0, 0$ $y_1(0), y_2(0) = 4, 7$ $y_1(0), y_2(0) = 7, 0$

Figure 6 – Numerical resolution with different initial datum

2.2 Integration of the system using high order RK methods with fixed time steps

We have to set the time step small enough to let the eigenvalues stay in the convergence domain. As the same discussion in PC3, if the time steps are too large, the error explodes and the solution loses its stability.

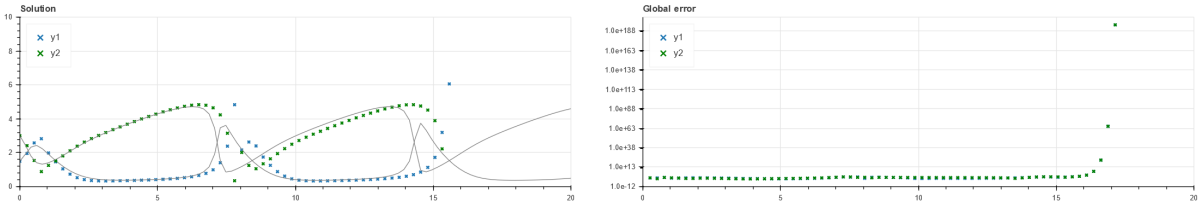


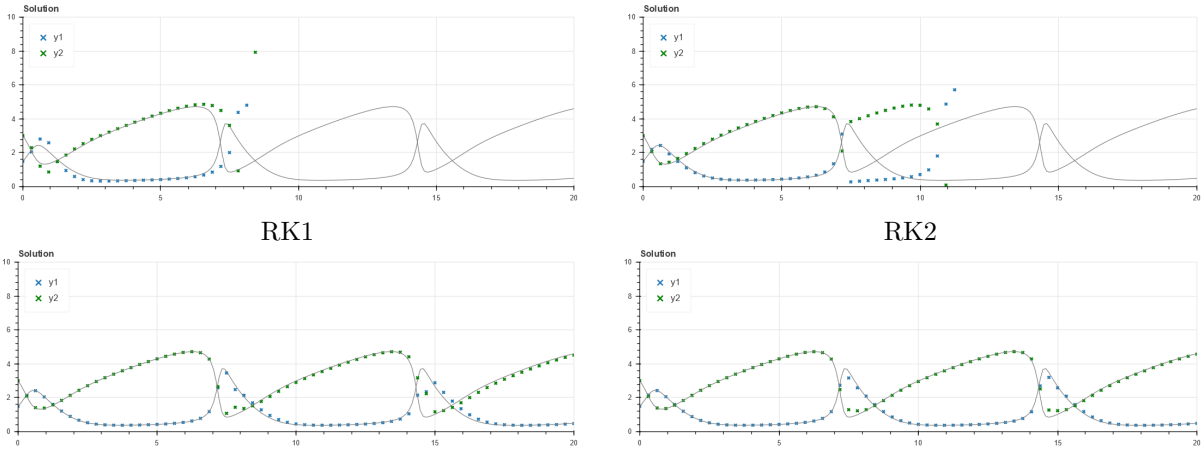
Figure 7 – Numerical resolution with RK2, fixed time step

The stability is valid only in the negative plane of complex coordination plane. When the real part of the eigenvalues is positive, although we observe a stability for numerical solution, it have lost its meaning to discuss. So the stability criterion is valid only when the dynamics of system do not allow an exponential increase. (**3.2.1**)

From PC 3, we know that, **order of the method** and the **stiffness** will influence on the stability of the scheme. We conclude also from PC3 that the **initial conditions** exert an effect on the stiffness of the system, leading to an impact on the stability at last. (**3.2.2**)

From both figure 8 and 9, we can firstly deduct that from order 1 to 4, **the stability becomes stronger with a fixed time steps by increasing the order**. As well, from the error diagram, effectively, we can tell the order of different method. It is interesting to notice the **flat slope** (black curve) at the last phase of convergence for RK4. Around 10^{-13}, the numerical solution seems to fail. Actually, this is due to the limit of float accuracy of computer. The tolerance we can set for the "ivp" which we consider as the exact solution can only be above 10^{-14}. As a result, global error can be reduced around level 10^{-13}. Therefore, we can no more tell the performance between so called exact solution and RK4 method.(**3.2.3**)

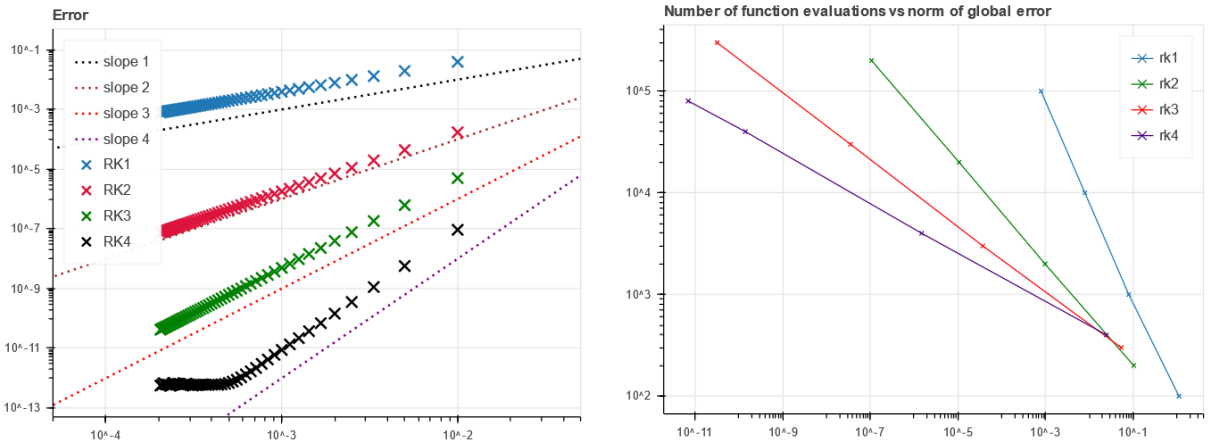
As for the computational cost, we can take the example when we set the global error tolerance 10^{-5}. Then the evaluation times for RK4 : 1.1×10^3 ; RK3 : 1.3×10^3 ; RK2 : 1.1×10^4. (The evaluation number is too huge for RK1...). Therefore, we can see the elevation of performance for different method in terms of computational cost. Yet, if we count the computational time-cost, the result could be a little different from those we explained, as the complex of steps for different order can also contribute to the time consumption. (**3.2.3**)



RK1 RK2

RK3 RK4

Figure 8 – Stability - Numerical resolution by RK with different order, fixed time step $n_t = 65$



Global Error with order 2 norm Computational cost

Figure 9 – Error and Cost

Once again, if we focus on the error diagram of Fig 9, we realized different numerical solution for number of time steps from 1000 to 50000 during a time period [0,10]. **When we increase the number of time steps, the global error tends to decrease**. And from the Fig 10, it is obvious that the global error increase on the stiff zone. **This error increment is also out of the stability of the method.** If we look at Fig 8, the deviation or the divergence of numerical solution (RK1 and RK2) takes place where stiffness appears. (**3.2.4**)

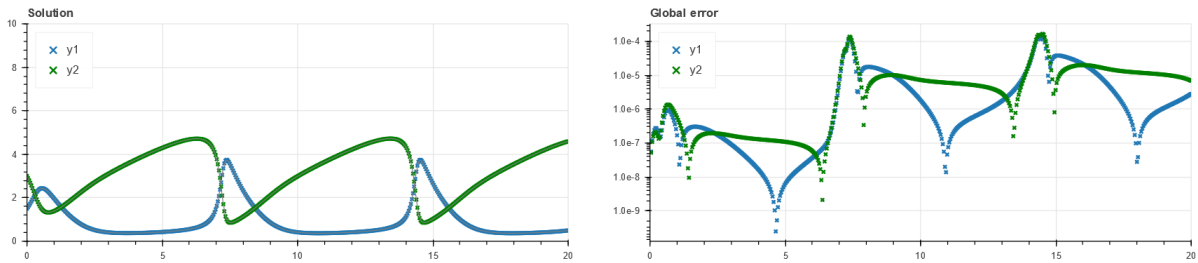


Figure 10 – Numerical solution with RK4

If we try to deduct from Fig 11, we might believe that the global error accumulate through time. Actually when we enlarge the interval for time, the error tends to stabilize instead of exploding. Yet, after each stiff zone, the error comes back. **That is thanks to the fact that the system is dissipative**, therefore, the accumulated global error can be decreased during time. As we have seen in PC 2, if the system is conservative, the error explodes no matter what the method is. (**3.2.5**)

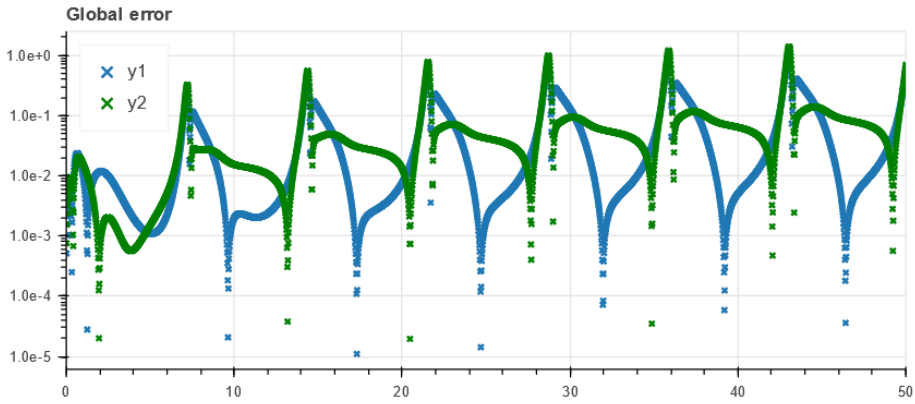
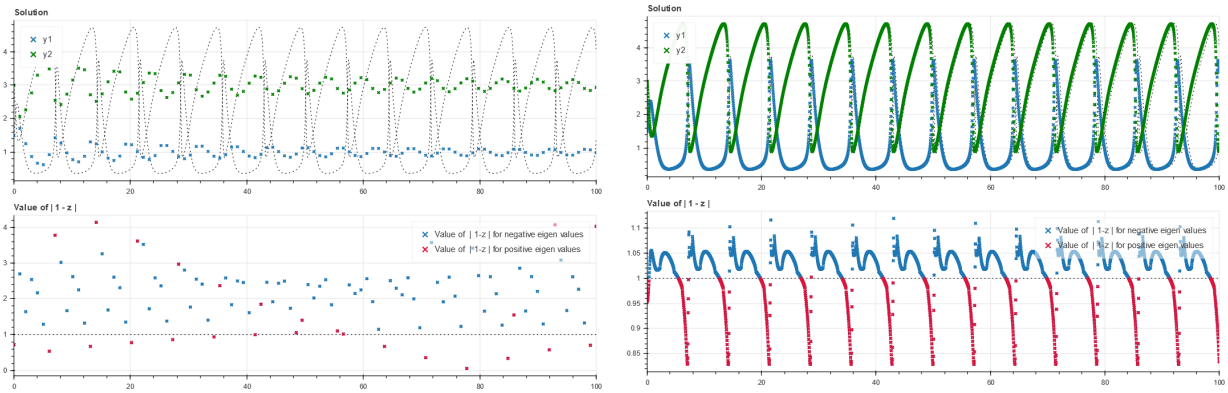


Figure 11 – Global Error with time step 1/60 , $t \in [0, 50]$

When we apply implicit Euler, the solution does have an absolute stable propriety, but the solution itself does not represent the system correctly anymore. **When we have eigenvalues that are larger than 1, the dynamical system should be characterized with an explosive increasing tendency. But when we look at the case where δt time step equals to 1, while eigenvalues are bigger than 1 (red points), there are quantities of points of converge function that are smaller than 1, which is against the real system's behavior.** When we increase the steps number, there is very few (or there is no more) case like that, convergence function passes below 1 as long as eigenvalue is positive. The conjecture can be made as : **When equations have eigenvalues with positive real part, the system is convergent if the $|1 - z|$ (convergence function) < 1.** (**3.2.6**)

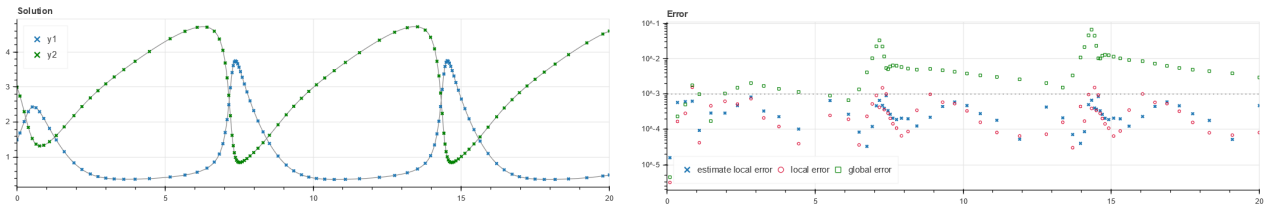


$\delta t = 1$ $\delta t = 1/30$

Figure 12 – Numerical solution with implicit Euler for $t \in [0, 100]$

2.3 Integration of the system using adaptive time stepping

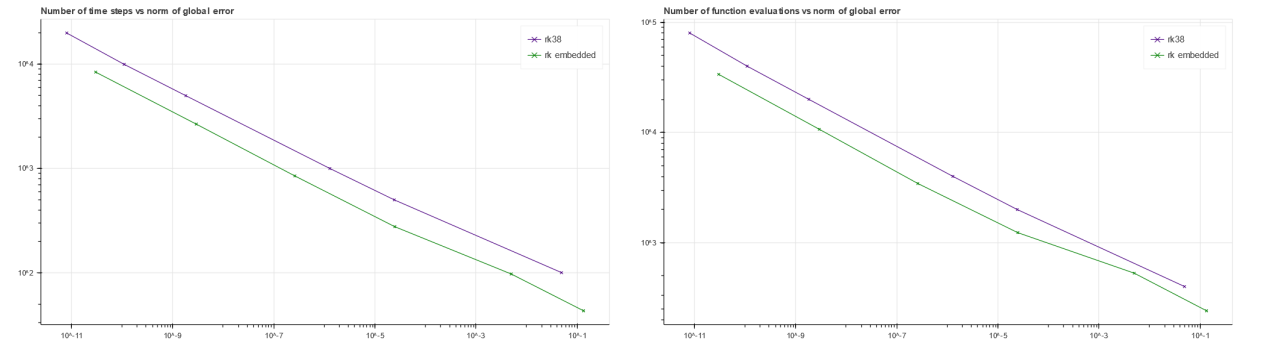
The Figure 13 shows that the adaptive time stepping method has finer time steps when the system gets stiff while the RK with fixed time step have uniform time steps though time.



Solution representation Error terms

Figure 13 – Numerical resolution with adaptive time steps and error : local error (red), global error (green), estimate local error (blue)

At the same time, left diagram in Figure 14 represents the relation between number of time steps and global error. **If we only concentrate on the time steps number, it is evident that adaptive method prevail.** (**3.3.1**) However, when we apply adaptive method, **it is potentially possible that the current evaluation would be rejected. Therefore, we need an extra efforts to proceed. The number of evaluations might be a more objective criterion to analyze the computational cost.** Then, another result is shown on right of the Figure 14, when the global error (or tolerance) is large, the gap between two methods narrows, which is produced from the fact that there are lots of rejected cases and consequently re-evaluations.(**3.3.2**)



time steps in function of global error numbers of function evaluations VS global error

Figure 14 – Error vs Time

It might be interesting to mention that, we took a four steps RK method, so the total number of evaluations should be around the fourth times bigger than number of total time steps if no step is rejected. With this intuition, we construct a tabular as below, thus, **RR(T), the rejected rates in terms of time steps and RR(Eva) : that calculated with number of function evaluations are almost equal.**

Global Error	RT	TS	RR(T)	NE	RR(Eva)
1.e-1	14	30	0.46(67)	177	0.47(50)
1.e-2	17	43	0.39(53)	241	0.40(11)
1.e-3	23	62	0.37(09)	341	0.37(5)
1.e-4	35	97	0.36(08)	529	0.36(34)
1.e-5	38	161	0.23(60)	797	0.23(75)
1.e-6	32	276	0.11(59)	1233	0.11(68)

RT : number of rejected time steps ; TS : total number of time steps
RR : rejected rates in terms of time steps (RT/TS) ; NE : number of function evaluations ;
RR(Eva) : Rejected rates in terms of number of evaluations : $NE/(4 \times TS) - 1$

The exact local error (erreur de troncature), can be calculated as :

$$\varepsilon_n = y(t_{n+1}) - \Big(y(t_n) + \Delta t \phi(t_n, y(t_n); \Delta t)\Big)$$

where ϕ is an evaluation function, and $y(t_i)$ stands for the exact solution at instant t_i.

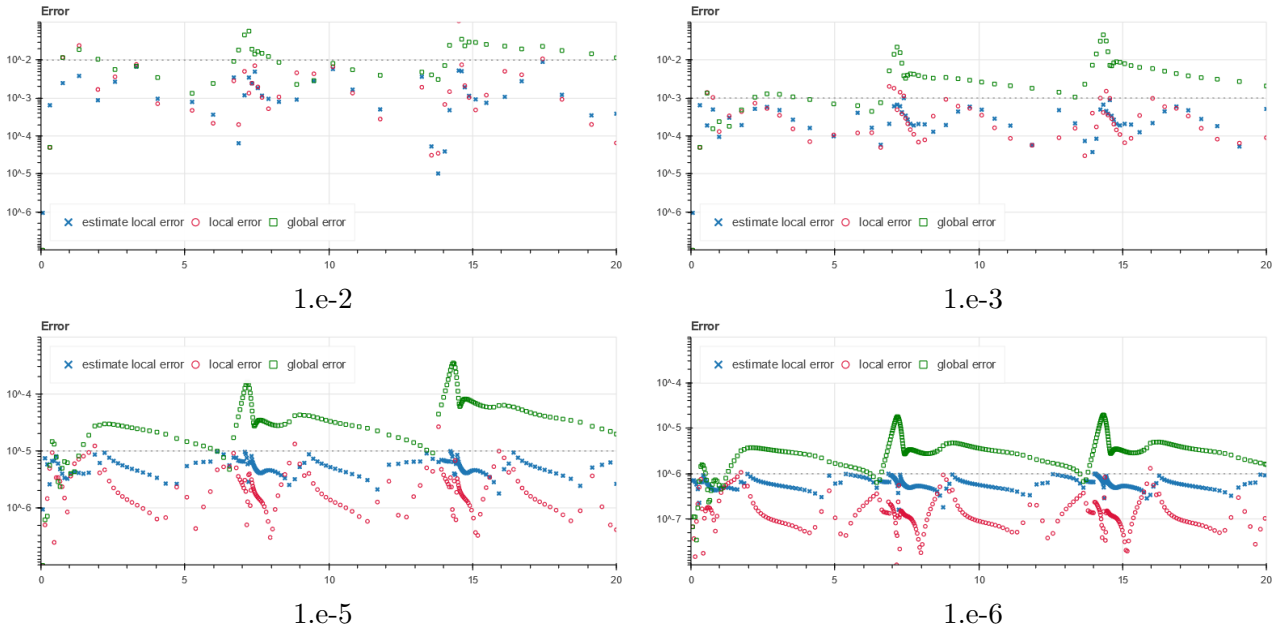


Figure 15 – Error : local error (red), global error (green), estimate local error (blue)

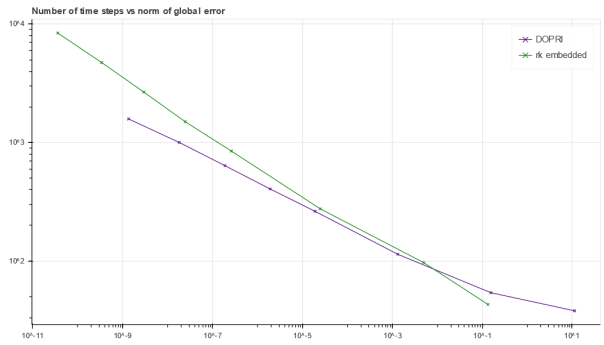
From figure 15, we can see the **estimated local error is (almost) above the local error, which will reduce the time step when there is a need (when the local error gets large, we should refine the time step to fit better to the real model.).** To this extent, the embedded method is efficient. (**3.3.3**) Although, **when the stiffness appears, the high order method behaves no better than the referent lower order method, the fact that they evolve at the same pace underestimates the exact local error.** Finally, it deviate the numerical solution from the exact one. (**3.3.3**)

Evolution of the number of steps and computational cost are shown at Fig 14. It is logical that when tolerance gets smaller, we will have a larger cost. And, as both methods is of 4^{th} order, they have a similar slope. (**3.3.4**)

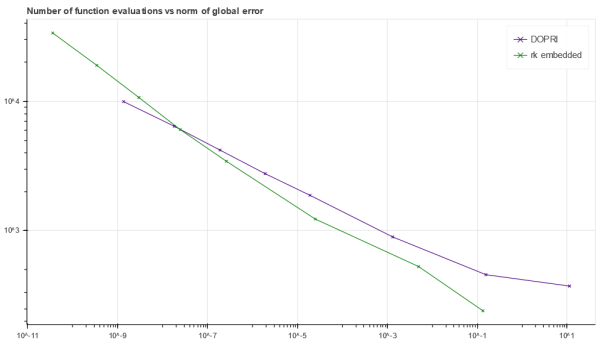
2.4 Integration of the system using Dormand and Price method

For each point in two diagram in Fig16, we demand the same tolerance.

It comes as no surprise that the DOPRI method has a (generally) stronger performance since it uses a higher order structure. But, if we look at the evaluation times, the DOPRI method evaluate $y(t_{n+1})$ much more than RK embedded when global error is relatively large, which means that given the same number of time steps, DOPRI method rejects more often the current Δt. It, therefore, introduces more efforts to evaluate the value of next y. (**3.4.1-3.4.3**)



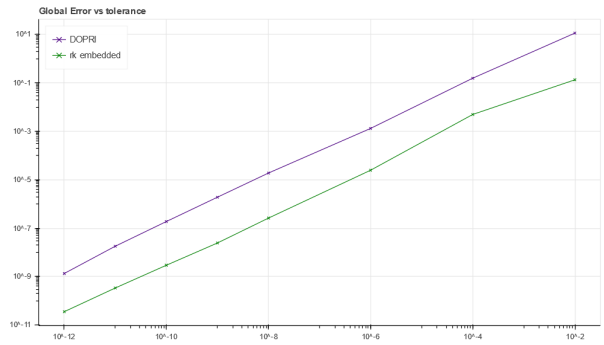
time steps in function of global error



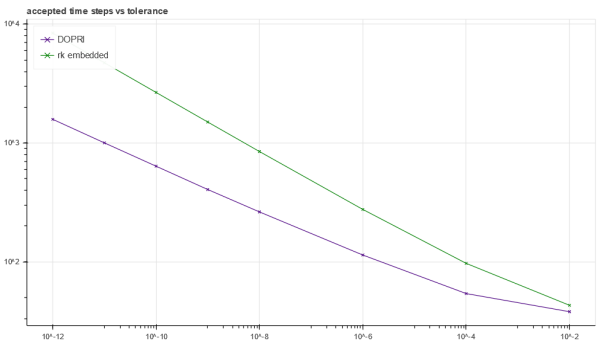
numbers of function evaluations VS global error

Figure 16 – Error vs Time for methods RK embedded and Dromand and Price

However, i**t is quiet astonishing when we concentrate on the left figure of Fig 17. As a higher order method, DOPRI has a larger global error compared with RK embedded with the same imposed tolerance. Fortunately, the number of time steps decrease shapely with the DOPRI method, which means that we saved a lot of time to compute.** Therefore, there exists a concurrence between the time consumption and global error. And from Fig 16, we can tell that the DOPRI method performs better only when the tolerance is extremely demanded. (**3.4.3**)



X :tol, Y :gloabl error



X :tol, Y :accepted time steps

Figure 17 – Error vs Time for methods RK embedded and Dromand and Price

3 The van der Pol oscillator

We consider the following problem :

$$\begin{cases} \mathrm{d}_t y_1 = y_2 \\ \mathrm{d}_t y_2 = \varepsilon\,(1 - y_1^2)\,y_2 - y_1 \quad \text{avec } \varepsilon > 0 \end{cases} \tag{1}$$

3.1 Stiffness

We could expect that ε adds to stiffness of the system as the larger ε is, the stiffer the system will be.

The stiffness comes from the fact that when $|y_1|$ differs from 1, the $\mathrm{d}_t y_2$ will be large, introducing a violent variation of y_2. **Stiffer the system is, less time we need to reach the limit cycle. The stiffness of initial transient depends essentially on the initial datum :** The closer to the extremum of periodic evolution the initial condition is, the less stiffer the transient is.(**4.1.1**)

When $\varepsilon = 1$, the dynamic is not as stiff as the system of last model is. That is because the real part of two eigenvalue is smaller than the previous case. When $\varepsilon = 20$, the real part of eigenvalues pass easily towards 20, which is larger than previous ones.

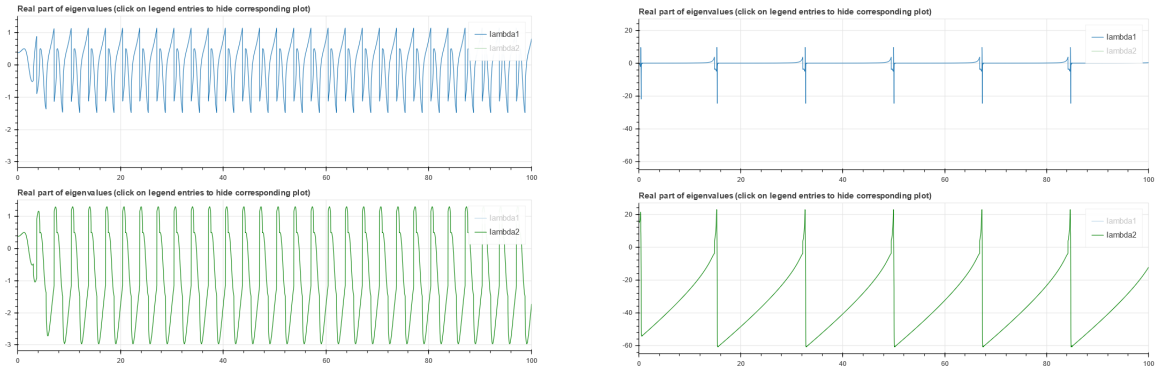
ε small : the stiffness is almost the same during one cycle ; ε large : the stiffness appears only in a very short term and is strong.

$$J = \begin{bmatrix} 0 & 1 \\ -2\varepsilon y_1 y_2 - 1 & \varepsilon(1 - y_1)^2 \end{bmatrix}$$

characteristic function :

$$x^2 - x\varepsilon(1 - y_1^2) + (2\varepsilon y_1 y_2 - 1)$$

When the imaginary part is not zero, the real part should be the order of $\mathcal{O}(\varepsilon)$

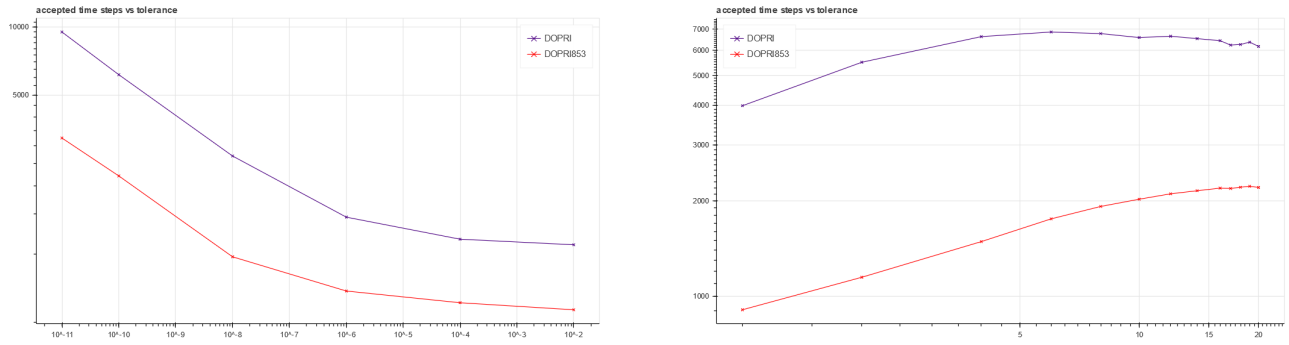


$\varepsilon = 1$, real part of eigenvalues $\varepsilon = 20$

Figure 18 – Representation of two eigenvalues of Jacobian matrix

The Figure 19 shows that (for a given stiffness) **when the tolerance is gradually and increasingly demanded, the number of time step increases as well. And the increment of ε implies an intensify of stiffness of the system. Then, (for a given tolerance) the time steps needed grow in number.**

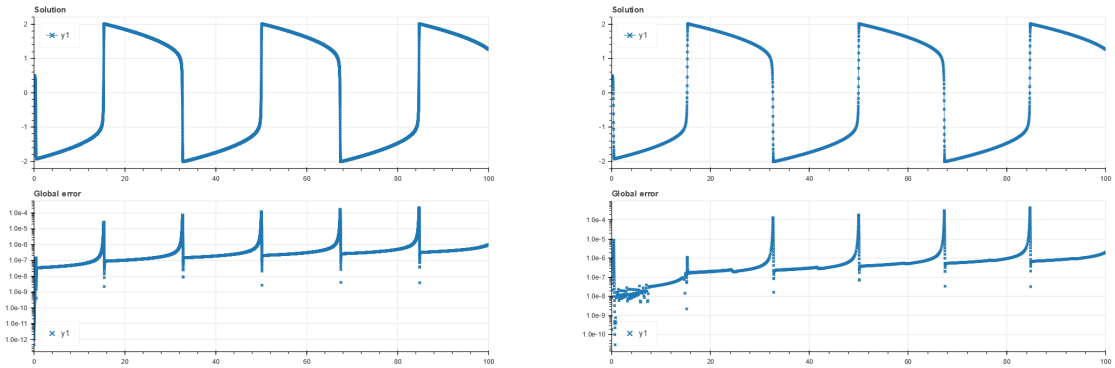
Comparing with DOPRI853, the classical DOPRI5 needs more time steps for the same stiffness and tolerance. As for the performance against stiffness, the computational cost of DOPRI853 is even 3 times less than DOPRI5. (**4.1.5**)



time steps in function of tolerance — time steps in function of ε

Figure 19 – Time steps of methods : DORPI 5 and DOPRI 853

Next, we would like to use RK4 method to reach the same level of tolerance. **As shown below, the RK4 needs 50000 uniform steps to achieve the goal, 34 times bigger than DOPRI5.** But we didn't count the different effort to evaluate y_{n+1}.(**4.1.4**)



number of steps=50000 — number of steps=1451

Figure 20 – Left : Runge-Kutta order 4 with fixed time steps ; Right : Dormand and Price method order 5

4 Conclusion (4.1.6)

Until then, we have studied different schemes : RK1-4, RK38, RK with adaptive time step, DOPRI5 and DOPRI853. Higher order methods usually have larger stable domain which resists better the stiffness. Adapting time step helps us to save unnecessary efforts by adding little extra function evaluations where the system is not stiff.

Classical RK facilitates the calculation. Rather than setting a tolerance, we can have direct control by adding time steps. Their problem appears when system gets stiff. Adaptive and higher order methods like DOPRI save us a lot of time, but the tolerance demanded is not always satisfied. Fortunately, they do not worse than the embedded lower order scheme.

Another scheme is the implicit one, it is so robust that it guarantees the A-stability. Difficulty appears on the stiff zone where solution stabilizes while the real system is exponentially increasing.