New time-stepping prediction formula for an atmospheric chemical kinetics integrator

Franck Vigilant
Computer and Applied Mathematics Institute of Grenoble,
Joseph Fourier University Domaine Universitaire, BP 53, 38041 Grenoble cedex 9
fckvigilant@yahoo.fr

Abstract

The stiff ODE solver of the atmospheric chemical kinetics of BRAMS [1] uses a standard step size prediction formula. In this paper, we introduce a method to evaluate prediction efficiency. Then, we compare two prediction schemes on three different configurations. Our observations on specific behaviour of each scheme lead us to propose a new scheme. It offers up to 10% gain on total number of integration loops.

Keywords: *stiff problem, time-stepping, atmospheric chemical kinetics.*

1. Introduction

Atmospheric chemical kinetics belongs to stiff ODE problems. As a matter of fact, species concentration evolve with different order of magnitude. Rosenbrock methods are widely used to integrate the related ODE system and end by solving linear system $\mathbb{A}x = b$, [2], [4]. The stiffness ratio S (defined as $S = max_{i=1\cdots m}|\lambda_i|/min_{i=1\cdots m}|\lambda_i|$) may be of an order $\mathcal{O}(10^{13})$ [5] for chemical kinetics and varies in space and time. Then, we have to adapt the step size in a automatic way to control its own progress in integrating the solution. Step sizes are small when the solution gradient is large and large when the solution gradient is small. Multiples try is sometimes needed to adapt the step-time reach an acceptable error.

2. Prediction formulae

Standard step size prediction formulae are based on the error estimation. This estimation is usually computed using two approximations of the solution y_1 (of order p) and $\hat{y}_1$ (of order q, typically $q = p - 1$). The error is then proportionnal to $(\Delta t)^p$.

Current implementation of the automatic step-time control uses the following formula (eq. 1) presented by Hairer [3] (see also [4]):

$$dt_{new} = dt * min\Big(Fact_{max}, max\Big(Fact_{min}, \frac{Fact_{safe}}{Err^{\frac{1}{\tilde{p}+1}}}\Big)\Big) \quad (1)$$

where $Fact_{max}$, $Fact_{min}$ and $Fact_{safe}$ are user-defined parameters to improve probability of step-size acceptance and limit too fast decrease and increase of Δt. p is the order of the error tolerance.

We propose here to implement another time step control scheme (eq. 2) (see Freed [5]) and analyse actual performance respect to the usual [4] step size prediction formula.

His prediction scheme formula is:

$$dt_{new} = dt * \begin{cases} min\Big[Fact_{max}, \Big(\frac{Fact_{max}}{Fact_{n+1}}\Big)^{\frac{0,7}{p+1}} \Big(\frac{Fact_n}{Fact_{max}}\Big)^{\frac{0,4}{p+1}}\Big], \\ \text{if } Fact_{n+1} \leq Fact_{max} \\ max\Big[Fact_{min}, \Big(\frac{Fact_{max}}{Fact_{n+1}}\Big)^{\frac{1}{p}}\Big)\Big], \\ \text{if } Fact_{n+1} > Fact_{max} \end{cases} \quad (2)$$

where $Fact_{n+1}$ the error estimation of step $n + 1$.

2.1. Data Analysis

We will focus on one set of data input and perform one fixe integration step. We report two parameters: the number of total intermediate 'chemical' loops and the the number of tries to get an accepted step-time for a given error tolerance. Let's define the success step time prediction pass rate for try r ($SR(\Delta^r t)$) as follow:

$$SR(\Delta^r t) = \frac{\text{number of Resize Loop to pass in } r \text{ try}}{\text{number of ReSize loop}} \quad (3)$$

We observe a maximum of three tries to compute a correct step-time. We define the number of Δt resizing as follows: $\Delta^1 t$: passed after the first resizing, $\Delta^2 t$: passed after two resizing, $\Delta^3 t$: passed after three resizing. The total number of Δt failing is then $\Delta^1 t + 2 * \Delta^2 t + 3 * \Delta^3 t$.

	Standard Scheme	New Scheme
$SR(\Delta^1 t)$	$40,0\%$	100%
$SR(\Delta^2 t)$	$38,6\%$	
$SR(\Delta^3 t)$	$21,4\%$	
$SR(Overall)$	100%	100%

Table 1. Success rate for 1^{st}, 2^{nd} and 3^{rd} time Resize success loop in function of control steptime size scheme

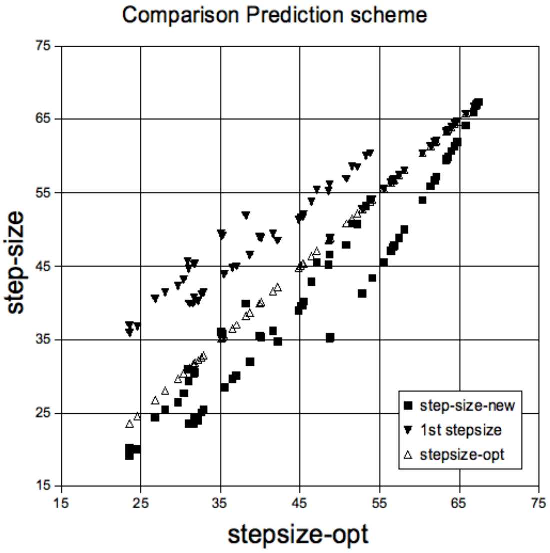

Figure 1. Comparison between standard prediction formula (successfull StepSize prediction (white triangles) and failed (black triangles) and the new prediction formula (black squares).

From the figure 1, we clearly understand why the new formula has a better success rate. Prediction is very severe and rarely compute a higher or equivalent steptime size than the standard formula (most of the points are in the graph zone below $y = x$ line).

Let's remark here the following, stepsize control formula correct a failing steptime but also give and output for the next computed steptime. Subsequently, it controls the whole integration behaviour. Thus, we must monitor not only the number of internal loop (α) but also the number of intermediate integration step (means successfully integrated) (β) to perform the integration between t and $t + \mathrm{d}t$. Then, comparing the term $\alpha + \beta$ for each formula give a rather good idea of their efficiency.

The next table 2 shows that the real improvement of the new scheme is very good to reduce the number of internal loop but ... null!! (0.5%) for the overall number of loops.

Formula	nb initial fails	nb internal loops	nb intg. steps	total nb loops
eq. 1	70	127	12533	12660
eq. 2	70	70	12529	12599
Diff. (%)	0	44.9%	$< 0.1\%$	0.5%

Table 2. Comparison of number total of loops for both formulae.

Remarks: 1/ 'initial fails' refers to the first integration fail (computed error greater than tolerance) for a given bock i. 2/ 'intg. steps' refers to total actual number of intermediate step time Δt^{CHEM} to integrate from $t + \Delta t^{BRAMS}$ for all the blocks. 3/ 'total nb loops' refers the number of times we go through the Rosenbrock algorithm (excluding Jacobian computation, not re-done within the internal loop).

As we have seen, the new formula is able to do a first-time-right step time prediction in our case. It predicts a smaller step time after a failure than the current formula. The potential drawback is to compute much more intermediate steptimes to perform the full integration. Here, here balance is not exciting at all. In addition, we have a sampling of fails very weak (0.5%) which does not help to conclude.

To discard our candidate we need to do further tests by integrating during a longer period or more critical data for instance. As we do not have at disposal another set of data to modify slightly the problem to stress the model and do additionnal observations.

2.2. Second configuration

We will stress the model by tightenning error tolerance limits. Instead of looking for new set of data which needs further effort for integration (kind of top-to-bottom problem) we propose to set more strict condition for integration error tolerance (kind of bottom-to-top problem). Currently, error computation is driven by two user-parameters $ATol$ and $RTol$ (absolute and relative error defined by species).

We propose to study three cases:

1. Case 1: $ATol = ATol^0$ and $RTol = RTol^0$ the standard case.
2. Case 2: $ATol = 0,1.ATol^0$ and $RTol = 0,1.RTol^0$.
3. Case 3: $ATol = 0,01.ATol^0$ and $RTol = 0,01.RTol^0$.

Case	initial fails	internal loops	integ. steps	total nb loops
1	70	127	12533	12660
2	1024	2362	14619	16981
3	6989	15256	37942	53198

Table 3. Comparison of number total of loops for standard formula for three different sets of $(ATol, RTol)$.

Initial results show that the standard formula (let's call it formula (a)) leads the same or more amount of total loops (see tables 3 and 5). From these results we can also forecast the ability of formula (b) to predict a 'rather good' value of steptime when the current steptime is not compliant with error tolerance and the ability of formula (a) to predict a 'rather good' value of steptime when the current steptime is compliant with error tolerance (by 'rather good' we mean here a probability, that we do not evaluate here (!), to lead to an accepted steptime).

From the previous observation, it is easy to make a step forward reminding the structure of formula (b) (two formulae with a condition on accepted or rejected value).

We propose a new formula, formula (c), combining both ability of formula (a) and formula (b). Means: formula (a) if the current steptime is accepted, formula (b) if it is rejected. To have a complete set we add, for comparison only, a fourth formula which will be the pending of formula (c): formula (b) if the current steptime is accepted, formula (a) if it is rejected. As expected, the new formula (c) gives the best results (see table 4) with 10% gain on total number of loops.

Case	diff. of total loops formula (a) vs. (b)	diff. of total loops formula (a) vs. (c)	diff. of total loops formula (a) vs. (d)
1	0.5%	0.4%	0%
2	6.2%	5.35%	0.8%
3	6.7%	9.97%	0.98%

Table 4. Comparison of number total of loops between formulae (a), (b), (c) and (d).

3. Conclusion

We have presented four sets of formulae. Standard formula appears to be an optimum when looking for the greater step-time size. Our formula (c) shows up promessing results with about 10% 'less work' for the tightest conditions. It is then our favorite candidate. Our choice of stressing the data (throught $ATol$ and $RTol$ parameters) give a good confidence on our output results. Similar performances are expected with other sets of data with similar comparable stiffness. We recommand to perform a one-day integration period to compare standard and new scheme and to quantify the corresponding gain.

Case	initial fails	internal loops	integ. steps	total nb loops
1	70	70	12529	12599
2	1072	1413	14513	15926
3	10389	12253	37394	49647

Table 5. Comparison of number total of loops for formula (b) for three different sets of $(ATol, RTol)$.

Case	initial fails	internal loops	integ. steps	total nb loops
1	70	70	12536	12606
2	1024	1365	14706	16071
3	6936	8800	39092	47892

Table 6. Comparison of number total of loops for formula (c) for three different sets of $(ATol, RTol)$.

Case	initial fails	internal loops	integ. steps	total nb loops
1	70	127	12530	12657
2	1027	2366	14479	16845
3	7981	16300	36374	52674

Table 7. Comparison of number total of loops for formula (d) for three different sets of $(ATol, RTol)$.

References

[1] S.FREITAS. *Chemistry at the Brazilian developements on the Regional Atmospheric Modeling System*. PowerPoint Presentation - April 2008

[2] J.G VERWER, *A Second-Order Rosenbrock Method Applied to Photochemical Dispersion Problems.* SIAM journal on scientific computing. 1999, vol. 20, no4, pp. 1456-1480 (32 ref.).

[3] E. HAIRER & G. WANNER *Solving Ordinary Differential Equations I: Non-stiff problems.*, Springer-Verlag, Berlin, 1991.

[4] A. SANDU. *Benchmarking Stiff ODE Solvers for Atmospheric Chemistry Problems II: Rosenbrock Solvers.* Computational mathematics. N° 90/1996.

[5] A.D FREED & AL. *Development and applications of a Rosenbrock integrator.* Nasa Technical Memorandum 4709. 1996.