Journal of Universal Computer Science, vol. 4, no. 2 (1998), 90-98
submitted: 25/9/97, accepted: 1/11/97, appeared: 28/2/98

Global Error Visualization

R. Aïd, L. Testard, G. Villard
(Laboratoire LMC/IMAG, Rene.Aid@imag.fr)

Abstract: A classical error estimation technique is used together with an interactive visualization software to validate numerical solutions of ordinary differential equations in the complex field. Examples showing how a modification of the integration path can help to reduce the absolute value of the global error committed by the integration process are provided.
Key Words: Ordinary Differential Equations, numerical methods, Richardson extrapolation, computer arithmetic, visualization.

1 Introduction

The aim of this paper is to show how classical global error estimation techniques together with powerful visualization software can be used to help validating numerical solutions of ordinary differential equations in the complex field.

A numerical integration code applied to an *ordinary differential equation* (ODE) makes use of a local error control to provide values within the range of a given tolerance. This local error control is achieved using a *local error estimate*, *i.e.* an estimation of the error committed within a single step. The code then guarantees that this estimation will be made smaller than the given tolerance.

The justification of this very classical approach, is that it leads to a *global error* – the "true error" – proportional to the tolerance [Ste80, Hig91]. As a result, it is then expected that if the tolerance is small, the solution will have some significant figures of the exact solution.

Nevertheless, for some equations and methods such as very unstable ODEs, stiff systems integrated with unappropriate numerical methods or systems integrated in the vicinity of a singular point, this may not be confirmed in practice. An absolute tolerance set to 10^{-3} does not guaranty that the global error is less than 10^{-3}. Because of the size of the constant of proportionality, in order for one significant figure to be guaranteed, the user should set the tolerance to a stringent value.

This is typically the case when the flow possesses singular points. Indeed, the closer the solutions are to the singular points, the more stringent the tolerance should be. If the user is interested in a set of solutions parameterized, for instance, by initial conditions, the maximum global error committed on each solution may greatly differ from each other depending on their getting too close to the singular points. Hence, to get the whole set of solutions within a given range of accuracy, different tolerances should be assigned to each initial conditions.

A global control of the integration would be of great help. Unfortunately, it is well-known that such a control is not in general possible due to the instability factors of the ODE ([SW76] and [Sha94], pp 76-78 for a comment on Airy's equation) and the limited arithmetic precision of computers [AL97].

Nevertheless, using a *global error estimation technique* together with an *interactive visualization tool* of the families of solutions, it is possible to assess the whole flow to stay within the range of a singled fixed tolerance by modifying the integration parameters (integration path, initial conditions, tolerances).

The paper is organized as follows. Section 2 presents the problem of estimating the global error during numerical integrations of ODEs. Our experiments will be based on *Richardson extrapolation* [Hen62]. Relying on this estimation, the visualization method described in §3, will enable users to study the regions of the flow where the global error is out of the range of the tolerance and to compute once again the corresponding solutions. We explain why the experimental validation of these integrations may rely on the use of an appropriate visualization tool. Two typical applications are examined in §4 and we show how the quality of the solutions can be improved. This will be illustrated by some graphical outputs[1].

2 Global error estimation of numerical methods for ODEs

We will assume that we integrate a differential equation of one complex variable:

$$\begin{aligned} \dot{y}(x) &= f(x, y(x)), \\ y(x_0) &= y_0, \end{aligned} \tag{1}$$

$x \in \mathcal{P}$, where $\mathcal{P} \subset \mathbb{C}$ is a complex path. It will also be assumed that f is holomorphic in the vicinity of the initial condition. A numerical method provides approximations y_i at $x_{i+1} = x_i + h_i$ of $y(x_i)$, $i = 0, \ldots, M$ along the path. The difference $E_i = y_i - y(x_i)$ denotes the *global error* committed by the method at x_i. The values $\hat{E}_i$ are said to be a *valid* estimation of E_i of relative order $r > 0$ when $E_i = \hat{E}_i\,(1 + \mathbf{O}(H^r))$ where $H = \max_i |h_i|$.

The theoretical justification of the validity of estimation techniques relies on the existence and on the structure of the asymptotic expansion of the global error.

Theorem 1 ([Gra64]) *Suppose that the integration of (1) is carried using a stable one-step method of order $p \geq 1$ with constant step-size integration h. Then, there exists smooth functions e_k, $k = p, \ldots, N-1$ such that uniformly on $[0, T]$,*

$$y_i - y(x_i) = h^p\, e_p(x_i) + \ldots + h^{N-1}\, e_{N-1}(x_i) + \mathbf{O}(h^N). \tag{2}$$

Using the existence of the asymptotic expansion in Theorem 1, a straightforward estimation formula can be deduced. In parallel with the original integration with step size h, another integration is carried out with step size $h/2$. Denoting the value obtained in this way at time t_i by y^*_{2i}, we have :

$$\begin{aligned} y_i &= y(x_i) + h^p\, e_p(x_i) + \mathbf{O}(h^{p+1}), \\ y^*_{2i} &= y(x_i) + (h/2)^p\, e_p(x_i) + \mathbf{O}(h^{p+1}). \end{aligned}$$

And, we get directly :

$$R_i := \frac{y_i - y^*_{2i}}{1 - 2^{-p}} = E_i + \mathbf{O}(h^{p+1}). \tag{3}$$

[1] These are available on a colour plate page 4.

A large number of numerical tests have shown that Richardson's estimator behaves well as long as it is used away from the *maximum possible precision* [AL97]. This quantity depends both on the equation which is to be solved and on the precision arithmetic. To give an idea of this phenomenon, the Figure 1 shows the behavior of the global error and R_i using DOPRI5(4) [HNW87] when the tolerance varies on the real ODE:

$$\begin{aligned} y' &= 10\,(y - x^2), \\ y(0) &= \quad 0.02. \end{aligned}$$

on $[0, 2]$. Its solution is $y(x) = 0.02 + 0.2\,x + x^2$. This is a very unstable problem. It can be seen that from tolerance 10^{-3} to 10^{-8}, the global error decreases linearly in logarithmic scale. In this zone, the estimation R_i is reliable. But, no more precision is gained after tolerance 10^{-8}. In that zone, R_i underestimates the global error and still decreases linearly in logarithmic scale.

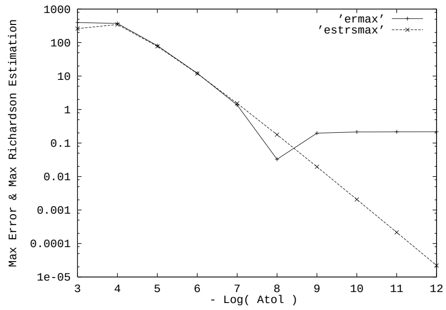


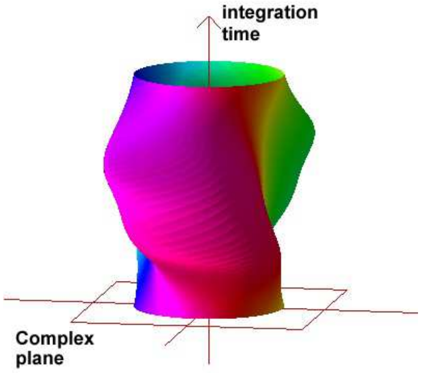
Figure 1: Compared behavior of maximum Richardson estimation and maximum global error

The maximum possible precision depends both on the instability of the equation and of the arithmetic of the computer.

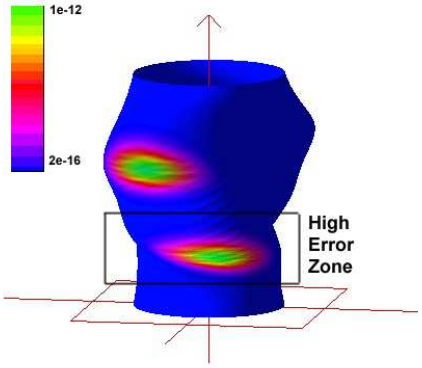
An interest of complex ODEs is that they also allow to modify the integration path, as long as the endpoints of the path are preserved. Hence, as the limit of accuracy of a numerical method is sooner reached in the zones where the flow varies rapidly, and typically this is the case in the vicinity of a singular point, it is possible to reduce the global error by changing the integration path rather than setting the tolerance to a more stringent value. The integration process is also all the more efficient.

This remark led us to consider that an interactive tool for the integration of complex ODE could be useful to assign a set of solutions within a given tolerance. The idea is to proceed in two steps. In the first step, the user specifies both the integration path, the set of initial conditions and the tolerance. Using the global error estimation, the program returns the zones of the flow where the maximum estimation has occured. Then, in a second step, by offering to the user the possibility to change locally the integration path, it is possible to get away from the singular points.

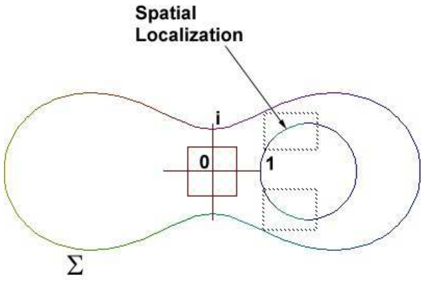
Figure 2: Colour plate



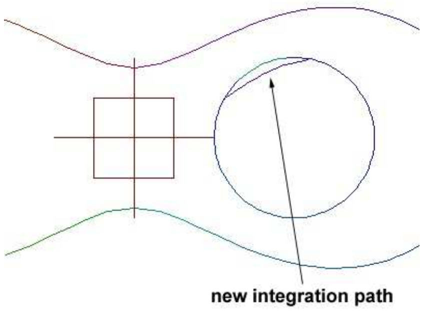
(2a) The solution surface



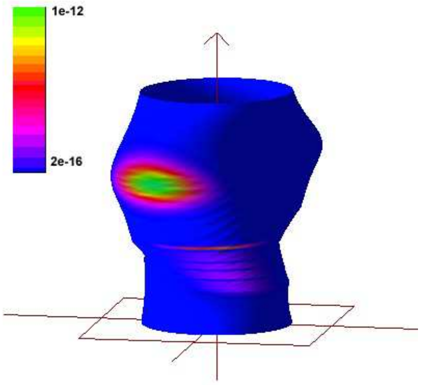
(2b) Global Error visualization



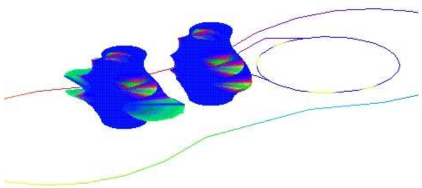
(2c) Spatial localization



(2d) New integration path



(2e) New Global Error Visualization



(2f) A bad case

3 Global Error Visualization

We have seen in the previous section that the visualization of a valid global error estimation during the integration of a complex ODE, may help end-users to improve the results of experiments using *user feedback*. The efficient visualization of data lying in the complex field may be a difficult problem, especially for the global error. Visualization in the complex field may be difficult because of the limited number of dimensions the human eye can appreciate. Moreover, the global error is generally not significant on account of its values, but its variations and localization are attached to a given quantity, such as the solution itself, and end-users who want to visualize such quantities will generally be dealing with these two problems. The GANJ graphical environment [Tes97] that we have developed in this context, implements some key features to handle these questions.

From a global point of view, human perception is limited by four dimensions: the classical three dimensions of space, plus additional data such as colours of points in the three-dimensional space. We will thus assume that we can visualize quantities embedded in $\mathbb{R}^3 \times I$, where I is a finite interval which may be coded by a colour formalism [Ric88]. The solutions resulting from integrations of a complex ODE are elements of $\mathbb{C}^2$, and need *four* dimensions for their visualization. In this particular case, we can nevertheless use a visualization technique derived from the *extended phase portraits* [BO78]. This visualization technique provides a way to visualize a *group* of solutions, specified by different initial conditions or by different values of a parameter.

If we want to integrate the differential equation (1) where x is taken on a complex path $\mathcal{P}$, and $y_0 \in \gamma \subset \mathbb{C}$ (the initial conditions is the pair (x_0, y_0)). Suppose the path $\mathcal{P}$ is parameterized by a diffeomorphism Φ. We can write

$$x = \Phi(t).$$

The real parameter t will be called *integration time*. We can perform the substitution in the equation (1)

$$\frac{dy}{dx} = \frac{1}{\Phi'(t)} \frac{d\tilde{y}}{dt},$$

where $\tilde{y}(t) = y(\Phi(t))$. Doing so, we can rewrite the equation (1) as

$$\frac{1}{\Phi'(t)} \frac{d\tilde{y}}{dt} = f(\Phi(t), \tilde{y}(t)).$$

If we now write $\tilde{y}(t) = y_1(t) + iy_2(t)$ and

$$\Phi'(t) f(\Phi(t), \tilde{y}(t)) = f_1(t, y_1, y_2) + i f_2(t, y_1, y_2),$$

we obtain a 2×2 real dynamical system

$$\begin{cases} \dfrac{dy_1}{dt} = f_1(t, y_1, y_2), \\[2ex] \dfrac{dy_2}{dt} = f_2(t, y_1, y_2). \end{cases} \tag{4}$$

The solutions of this dynamical system can be visualized on a phase portrait [BO78]. But this dynamical system is generally not autonomous, even if equation (1) was autonomous, and as a consequence different solution curves (issued from different initial conditions) can intersect, and the interpretation of the phase portrait can be obscured. Lifted orbits $\Omega(x_0, y_0)$can be defined in order to solve this problem

$$\Omega(t_0, y_1^0, y_2^0) = \{(y_1(t), y_2(t), t),\}$$

where y in the above definition is such that $y_1(t_0) = y_1^0$ and $y_2(t_0) = y_2^0$. The *extended phase portraits* are defined by (D being a subset of $\mathbb{R}^2$)

$$\mathcal{EPP}(D, t_0) = \{\Omega(t_0, y_1^0, y_2^0), (y_1^0, y_2^0) \in D\}.$$

Two lifted orbits belonging to an extended phase portrait *cannot* intersect, even if the system is not autonomous. A first consequence is that extended phase portraits can be efficiently used for the visualization of solutions of dynamical 2×2 real dynamical systems. Secondly, the solution of system (4) is the complex solution of equation (1), and thus the visualization of the solutions of system (4) is equivalent to the visualization of solutions of the complex equation (1). Such an extended phase portrait is shown on figure (2a): the domain D where the y_0's are taken is a circle centered around 0, and this domain can be seen on the figure by considering the point of altitude 0 (corresponding to the value of $t = 0$). The surfaces are obtained by connecting two adjacent solution curves (solutions issued from adjacent values of y_0 in γ).

This visualization method uses only three dimensions: the fourth, corresponding to the colour of the points, can be used to visualize values of another quantity attached to the solution. Using this method, we can visualize global error's module, as shown on figure (2b). The palette used for the quantification is displayed on the left of the figure, and we can visually detect the zones of the surface where this module is important.

Another advantage of this visualization method is that it enables the tracing of a solution when t varies, and so the position of the solution in the complex plane can be known for any value of x on the path $\mathcal{P}$, using the relation $x = \Phi(t)$. Furthermore, the repartition of the colours on the surface (concentration or diffusion of colours) enables visual detection of the dependence of the equation to the initial conditions.

The next section shows applications of this principle.

4 Applications

We now propose a visualization method in response to the questions of paragraph 2. In addition, we give an example of feedback the user can provide in order to improve the quality of the results. This method has been successfully implemented using the GANJ graphical environment.

The numerical integrations of (1) have been achieved using the explicit Runge-Kutta method of order 5 of Dormand and Prince (DOPRI5(4)). The local error estimate is performed using local extrapolation and current safety factors to smoothen the selection of the step-size have been implemented.

4.1 Time localization of Global Error

This technique provides a method to visualize the *time* variations of the estimated global error. Because the time can be read on the z-axis of the figures 2a and 2b, the variations of the global error estimated during the integration can be visualized, using the aforementioned technique. Because the global error data is discrete, its visualization needs the definition of a quantification mapping. This mapping can be of several types:

– a linear mapping:

$$[error_{min}, error_{max}] \longrightarrow [0, colourmax]$$

where $error_{min}$ (resp. $error_{max}$) is the minimum (resp. the maximum) of the module of the global error estimated during *one* integration. This mapping, and so the visualization technique associated to it, lets users find the time-zones where the module of the global error is high by reading z-values of points of colour $colourmax$. This technique is not useful anymore when several solutions are to be studied (these solutions can be be computed from several initial conditions), because users cannot perform global comparisons between different solutions.

– a linear mapping :

$$[error^*_{min}, error^*_{max}] \longrightarrow [0, colourmax]$$

where $error^*_{min}$ (resp. $error^*_{max}$) is the minimum (resp. the maximum) of the module of the global error estimated for *every* integrations (issued from every initial conditions). This *global quantification* mapping can prove useful for the localization of global extrema of global error's module, and the visualization method associated to it can show the time zones of highest error.

These mappings can be replaced by logarithmic ones, when variations of global error are important, or by refined ones, when users study the behavior near a precise value of y. The solution of the considered differential equation can therefore be *precisely* studied, by a time-refinement of the time-interval, near the zone where estimated global error's module is high (the green zones on the figure 2b).

4.2 Spatial localization of Global Error

Because time in the previous point is naturally correlated to x position by the formula $x = \Phi(t)$, it may prove useful to associate the times zones detected by the aforementioned technique with corresponding x values. Furthermore, because solutions values are supposed to be the same[2] when obtained after integration on two paths having the same endpoints, the initial integration path can then be deformed in order to avoid the spatial zones where high values of global error are estimated. For instance, one can avoid a zone "too close" from a singularity of the solution.

We illustrate this principle by the example of the differential equation

[2] except in the case when the solution of the differential equation is *many-valued*

$$y' = xy^2 \tag{5}$$

where x is taken on a circle centered on 2, of radius 1, and y_0 is taken on a circle, γ, centered on 0, of low radius (0.3). After integration of equation (5), a solution surface as shown on figure (2a) is obtained, and on figure (2b), the values of the solution are plotted with a colour corresponding to a global quantification mapping. The colour of the highest error zones is encoded in green, as shown on the colour palette displayed on the left of the figure.

A spatial localization of the global error is shown on figure (2c). The green zone on the integration path is defined by the set

$$\{x \in \mathcal{P}, \exists y_0 \in \gamma, |\hat{E}_i| > \delta\},$$

where $\hat{E}_i$ is an intermediate global error estimation during the integration process, at the i^{th} iteration, and δ an arbitrary number in $[error^*_{min}, error^*_{max}]$. The aim here is to deform the path in order that the integration variable x avoids the zones where the global error is high: for this particular example, the singularities position is known. 0 is a fixed singularity of the equation (5), and σ_1 and σ_2 are movable singularities, whose position depends on x_0 and y_0:

$$\sigma_j = (-1)^{j-1}\sqrt{x_0^2 + \frac{1}{y_0^2}}.$$

The figure (2c) shows the set $\Sigma = \{\sigma_1(x_0, y_0), y_0 \in \mathcal{P}\}$. For *some* value of y_0, the integration path gets close to Σ: the integration path can then be deformed to stand away from Σ. Figure (2d) shows a new user-specified path, where further integrations will be performed.

The figure (2e) shows the result of the global error estimation computed during the integration on the new user-specified path. The mapping used for the visualization is the same as the mapping used for the previous visualization, in order to compare the two integrations. The green zones have been reduced on the part of the surface corresponding to values of t where specifications of the new integration path have been performed.

This visual detection of high global error zones leads the user to specify integration paths that will avoid – or at least reduce – these zones.

Of course, it is also possible that some attempts to change the integration path increases the global error estimation. The figure (2f) gives an example of such a situation. The integration path has been modified in such a way that it gets closer to the singular points. Are represented both the global error estimations obtained before the modification of the integration path (on the right) and after (on the left). The green zones on the new global error estimation surface have been obtained using the same mapping as on the previous one. Since they are larger, it is clear here that this modification had the bad effect of increasing the global error estimation.

5 Conclusion

It is always a problem to validate the numerical outputs of a program. In the case of the solution of differential equations we have seen that many parameters should be considered. Indeed, the global error depends on the integration method, on the given equation and on the path of integration. The arithmetic error even if mainly related to the machine, also depends on the problem. Furthermore, the numerical data may lie in various mathematical domains (future investigations will concern Riemann surfaces). It is essential that all these parameters are studied simultaneously to gain the best understanding of the problem and the numerical results. We have pointed out that an efficient graphical tool appears to be necessary and how it can be used to improve the results in a mathematical context.

References

[AL97] R. Aïd and L. Levacher. Numerical investigations on global error estimation for ordinary differential equations. *J. of Comput. and Appl. Math.*, august 1997.

[BO78] C. M. Bender and S. A. Orszag. *Advanced Mathematical Methods for Scientists and Engineers.* McGraw-Hill, 1978.

[Gra64] W. B. Gragg. *Repeated extrapolation to the limit in the numerical solution of ordinary differential equations.* PhD thesis, University of California, 1964. see also SIAM J. Numer. Anal., ser. B, vol. 2, p. 384-403, 1965.

[Hen62] P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations.* John Wiley & Sons, Inc., 1962.

[Hig91] D. J. Higham. Global error versus tolerance for explicit Runge-Kutta methods. *IMA J. Numer. Anal.*, pages 457–480, 1991.

[HNW87] E. Hairer, S.P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems.* Springer-Verlag, 1987.

[Ric88] F. Richard. Computer graphics and complex ordinary differential equations. In *New Trends in Computer Graphics*, C. G. International, Genève, pages 515–524, 1988.

[Sha94] L. Shampine. *Numerical Solution of Ordinary Differential Equations.* Chapman and Hall, 1994.

[Ste80] H. J. Stetter. Tolerance proportionality in ODE-codes. In R. März, editor, *Proc. Second Conf. on Numerical Treatment of Ordinary Differential Equations.* Seminarberichte 32, 1980.

[SW76] L. F. Shampine and H. A. Watts. Global error estimation for ordinary differential equations. *ACM Trans. Math. Softw.*, 2:172–186, 1976.

[Tes97] Laurent Testard. *Calcul, Équations différentielles et visualisation en nombre complexes.* PhD thesis, INPG, 1997. en préparation.