Notes on numerical integration of ODE's

1 Introduction

These notes draw *heavily* from chapter 3 of [1], chapter 16 of [2] and chapter 7 of [3]. A nice quick overview is also provided by `http://en.wikipedia.org/wiki/Numerical_ordinary_differential_equations`.

2 Integration by Taylor series

Problems involving ordinary differential equations (ODE's) can always be reduced to the study of sets of first-order differential equations. For example the second-order equation

$$\frac{d^2x}{dt^2} = f(t)\frac{dx}{dt} + g(t) \tag{2.1}$$

can be turned into the first order system

$$\begin{aligned} \frac{dx}{dt} &= y \\ \frac{dy}{dt} &= f(t)\, y + g(t) \end{aligned} \tag{2.2}$$

Thus it is sufficient to discuss numerical schemes for the first order system of equations

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}\,(\boldsymbol{x}, t) \tag{2.3}$$

in a time interval for $t > t_o$ and with initial condition

$$\boldsymbol{x}(t_o) \equiv \boldsymbol{x}_{t_o} = \boldsymbol{x}_o \tag{2.4}$$

We will suppose in what follows the vector field $\boldsymbol{f}(\boldsymbol{x}, t)$ smooth (i.e. analytic) in $\mathbb{R}^d \times \mathbb{R}$. The equation (2.3) can be couched into the integral form m

$$\boldsymbol{x}_t = \boldsymbol{x}_o + \int_{t_o}^{t} ds\, f\,(\boldsymbol{x}_s, s) \tag{2.5}$$

It is expedient to introduce the change of integration variable

$$s = t - u \quad \& \quad ds = -du \tag{2.6}$$

so that

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{x}_o + \int_0^{t-t_o} du\, f\,(\boldsymbol{x}_{t-u}, t-u) \\ &= \boldsymbol{x}_o + (t - t_o)\, f\,(\boldsymbol{x}_o, t_o) - \int_0^{t-t_o} du\, u\, \frac{df}{du}\,(\boldsymbol{x}_{t-u}, t-u) \\ &= \boldsymbol{x}_o + (t - t_o)\, f\,(\boldsymbol{x}_o, t_o) + \frac{(t-t_o)^2}{2}\frac{df}{dt}\,(\boldsymbol{x}_o, t_o) + \int_0^{t-t_o} du\, \frac{u^2}{2}\, \frac{d^2 f}{du^2}\,(\boldsymbol{x}_{t-u}, t-u) \end{aligned} \tag{2.7}$$

where

$$\frac{d\boldsymbol{f}}{dt}\,(\boldsymbol{x}_o,t_o) = \boldsymbol{f}\,(\boldsymbol{x}_o,t_o)\cdot(\partial_{\boldsymbol{x}}\boldsymbol{f})\,(\boldsymbol{x}_o,t_o) + (\partial_t\boldsymbol{f})\,(\boldsymbol{x}_o,t_o) \tag{2.8}$$

Iterating a further step we get into

$$\begin{aligned}\boldsymbol{x}_t &= \boldsymbol{x}_o + (t-t_o)\, f\,(\boldsymbol{x}_o,t_o)\\ &+ \frac{(t-t_o)^2}{2}\frac{d\boldsymbol{f}}{dt}\,(\boldsymbol{x}_o,t_o) + \frac{(t-t_o)^3}{6}\frac{d^2\boldsymbol{f}}{dt^2}\,(\boldsymbol{x}_o,t_o) - \int_0^{t-t_o} du\,\frac{u^3}{6}\,\frac{d^3\boldsymbol{f}}{du^3}\,(\boldsymbol{x}_{t-u},t-u)\end{aligned} \tag{2.9}$$

In other words the hypothesis of analyticity guarantees that integrating (2.3) is equivalent to generate the coefficients of the Taylor expansion of $\boldsymbol{f}$ around the point $(\boldsymbol{x}_o\,,t_o)$. This result can be used to construct numerical integration schemes of ODE's.

3 Euler scheme

The simplest integration scheme is the *Euler method*. First we partition the *finite* time interval $T := t - t_o$ into n sub-interval of equal size

$$\delta t := \frac{t-t_o}{n} \tag{3.1}$$

so that

$$t_k = t_o + k\,\delta\,t \quad \& \quad t = t_n = t_o + n\,\delta\,t \tag{3.2}$$

The quantity δt is often referred to as the *mesh* size of the discretization. If δt is "sufficiently" small i.e. n is large enough we can approximate

$$\boldsymbol{x}_{t_{k+1}} \simeq \boldsymbol{x}_{t_k} + \boldsymbol{f}\,(\boldsymbol{x}_{t_k}\,,t_k)\;\delta\,t \tag{3.3}$$

The symbol $\simeq$ here means that the left hand side equals the right hand side if we neglect terms of order $O\left(\delta t^2\right)$. In such a case, we can estimate the flow generated by the ordinary differential equation with the one of the discrete map

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \boldsymbol{f}(\boldsymbol{y}_k\,,t_k)\,\delta t \tag{3.4}$$

In one dimension or for each vector component of $\boldsymbol{x}\in\mathbb{R}^d$, the Euler scheme is the following recursion algorithm:

Algorithm 1 Euler

```
xvar = $x_o$
tvar = $t_o$
tfin =$t$
npartinions = $n$
mesh = (tfin-tinit)/n
for $k = 1, n$ do
    xvar = xvar + f(xvar,tvar) * mesh
    tvar= tvar+mesh
    print tvar xvar
end for
```

The statement **print** means that the outcome of the calculation is sent to some output (e.g. stored into a data file). The *local discretization error*

$$l_k = x_{t_k} - y_k \tag{3.5}$$

and the *global discretization error*

$$e_n = x_{t_n} - y_n \tag{3.6}$$

are standard measures of the accuracy of the approximation. A reliable numerical integration should provide an estimate for this errors. This is usually done by checking the convergence of the results to a given value versus the mesh size. In order to examine the dependence of the *global discretization error* upon the mesh it is convenient to consider exponential changes of the mesh size

$$\delta t_i = \frac{t - t_o}{n^i} \qquad i = 1, 2, \dots \tag{3.7}$$

and define

$$\boldsymbol{y}_{k+1}^{(i)} = \boldsymbol{y}_k^{(i)} + \boldsymbol{f}\left(\boldsymbol{y}_k^{(i)}, t_k\right) \tag{3.8}$$

Then one can study

$$\Delta_i = \ln ||\boldsymbol{y}_{n^{i+1}}^{(i+1)} - \boldsymbol{y}_{n^i}^{(i)}|| \tag{3.9}$$

versus $\ln \delta t_i$. The reason for introducing logarithms is that variations of order of magnitude in the Δ_i are reflected in change of slope in logarithmic scale. An alternative way to proceed to estimate errors, is to compare at fixed mesh the results of the Euler scheme and of an higher order scheme. The *order of a scheme* is defined as follows. A method is said to converge with *order* $\gamma \in \mathbb{N}$ if there exists a constant $K < \infty$ such that the global discretization error satisfies the bound

$$||e_{n+1}|| < K \, (\delta t)^{\gamma} \qquad \forall \, \delta t \in [0, \delta_{\star} t] \tag{3.10}$$

The Euler scheme can be proved to have order 1. Intuitively the order of the scheme can be thought as specified by the highest order term of the Taylor series matching the increment of the discrete map defining the approximation scheme in the limit of vanishing mesh size.

3.1 Limitations of the Euler scheme: stiffness

There are several reasons that Eulers method is not recommended for practical use, among them,

1. the method is not very accurate when compared to other at equivalent mesh size.

2. the method is not very stable.

The second pathology arises in the treatment of *"stiff"* systems of differential equations. Dictionary definitions of the word "stiff" refer to concept like "being not easily bent", "rigid" and "stubborn". In the context of ODE's a problem is said to be stiff if [3]

> *A problem is stiff if the solution being sought varies slowly, but there are nearby solutions that vary rapidly, so the numerical method must take small steps to obtain satisfactory results.*

[2] provides the following example.

$$\begin{aligned} \frac{dx_1}{dt} &= a\,x_1 + b\,x_2 \\ \frac{dx_2}{dt} &= -(a+c)\,x_1 - (b+c)\,x_2 \end{aligned} \tag{3.11}$$

with

$$c = O(1)\;>\;0 \qquad \& \qquad b-a = O(10^3)\;>\;0 \tag{3.12}$$

Independently of the value of the parameters the system is explicitly integrable. The orthonormal change of variables

$$X := \frac{x_1 + x_2}{\sqrt{2}} \quad \& \quad x = \frac{x_1 - x_2}{\sqrt{2}} \tag{3.13}$$

partially diagonalizes the system

$$\begin{array}{lcl} \frac{d}{dt}(x_1 + x_2) = -c\,(x_1 + x_2) & & \frac{d}{dt}X = -c\,X \\ & \Rightarrow & \\ \frac{d}{dt}(x_1 - x_2) = (2\,a + c)\,x_1 + (2\,b + c)\,x_2 & & \frac{d}{dt}x = (a+b+c)\,X + (a-b)\,x \end{array} \tag{3.14}$$

Remark: a systematic theory for the analytic integration of linear ODE's with constant coefficients

$$\frac{d\boldsymbol{x}}{dt} = \mathsf{A}\boldsymbol{x} \tag{3.15}$$

proceeds from similarity transformations

$$\boldsymbol{x} = \mathsf{O}\,\boldsymbol{y} \qquad \det \mathsf{O} \neq 0 \tag{3.16}$$

such that

$$\frac{d\boldsymbol{y}}{dt} = \mathsf{O}^{-1}\,\mathsf{A}\,\mathsf{O}\boldsymbol{y} \quad | \quad \mathsf{O}^{-1}\,\mathsf{A}\,\mathsf{O} = \operatorname{diag}\mathsf{A} \tag{3.17}$$

where diag means the full diagonalization of A or at least its reduction to Jordan form. The variables (3.13) reduce the equivalent of the matrix A for the system (3.11) to triangular form so that the system can be readily integrated

$$\begin{aligned} X_t &= X_o\,e^{-c\,t} \\ x_t &= x_o\,e^{(a-b)\,t} + (a+b+c)\,X_o \int_0^t ds\,e^{(a-b)\,(t-s)}e^{-c\,s} \end{aligned} \tag{3.18}$$

Performing the integral gives

$$x_t = x_o\,e^{(a-b)\,t} + \frac{a+b+c}{b-a-c}X_o\left\{e^{-c\,t} - e^{(a-b)\,t}\right\} \tag{3.19}$$

Going back to the original variables the solution of (3.11) versus initial conditions $(x_{1;o}\,, x_{2;o})$ reads

$$\begin{aligned} x_1 &= \frac{X_t + x_t}{\sqrt{2}} = \frac{b(x_{1;o} + x_{2;o})\,e^{-c\,t} - [(a+c)\,x_{1;o} + b\,x_{2;o}]e^{(a-b)\,t}}{b-a-c} \\ x_2 &= \frac{X_t + x_t}{\sqrt{2}} = \frac{-\,(a+c)\,(x_{1;o} + x_{2;o})\,e^{-c\,t} + [(a+c)\,x_{1;o} + b\,x_{2;o}]e^{(a-b)\,t}}{b-a-c} \end{aligned} \tag{3.20}$$

Using now the hypotheses (3.12) with

$$c = \frac{1}{\tau} \qquad \& \qquad b - a = \frac{1000}{\tau} \tag{3.21}$$

we see that in order to observe the decay of the exponential $e^{(a-b)\,t}$ we need a mesh size

$$\delta t \,\ll\, \frac{\tau}{1000} \tag{3.22}$$

Failing to satisfy (3.22) may compromise the stability of the numerical integration scheme. The reason is that for any matrix A the components of the map

$$\boldsymbol{y}_{k+1} = \mathsf{A}\,\boldsymbol{y}_k \tag{3.23}$$

with solution

$$\boldsymbol{y}_n = \mathsf{A}^n\,\boldsymbol{y}_o \tag{3.24}$$

tends to zero as n tends to infinity if the *largest eigenvalue* of A has magnitude less than unity. The Euler scheme for

$$\frac{d\boldsymbol{x}}{dt} = -\mathsf{C}\boldsymbol{x} \tag{3.25}$$

with C a positive definite matrix corresponds to the map

$$\boldsymbol{y}_{k+1} = (1 - \mathsf{C}\,\delta t)\,\boldsymbol{y}_k \tag{3.26}$$

Denoting by $c_\star$ the largest eigenvalue of C

$$c_\star := \max \mathrm{sp}\mathsf{C} \tag{3.27}$$

the condition for $||\boldsymbol{y}_k||$ to be bounded is therefore that

$$\max \mathrm{sp}\,\{1 - \mathsf{C}\,\delta t\} \,<\, 1 \qquad \Rightarrow \qquad \delta t \,<\, \frac{2}{c_\star} \tag{3.28}$$

The example shows the source of the instability of the Euler scheme: the mesh size must be carefully chosen in order to achieve convergence. Other integration schemes maybe, however, less sensitive to the mesh size even in the presence of “stiff” problems. A nice article on stiff systems can be found at
`http://www.scholarpedia.org/article/Stiff_systems`

4 Second order schemes

The *improved Euler* or *Heun scheme* is an example of second order scheme:

$$\begin{aligned} \bar{\boldsymbol{y}}_{k+1} &= \boldsymbol{y}_k + \boldsymbol{f}\,(\boldsymbol{y}_k, t_k)\;\delta t \\ \boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + \frac{1}{2}\left\{\boldsymbol{f}\,(\boldsymbol{y}_k, t_k) + \boldsymbol{f}\,(\bar{\boldsymbol{y}}_{k+1}, t_{k+1})\right\}\delta t \end{aligned} \tag{4.1}$$

Comparison with the solution of (2.3) by Taylor series is achieved by observing

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \boldsymbol{f}\,(\boldsymbol{y}_k, t_k)\;\delta t + \frac{(\delta t)^2}{2}\left\{\boldsymbol{f}\,(\boldsymbol{y}_k, t_k)\,\partial_{\boldsymbol{y}_k}\boldsymbol{f}\,(\boldsymbol{y}_k, t_k) + (\partial_{t_k}\boldsymbol{f})\,(\boldsymbol{y}_k, t_k)\right\} + \dots \tag{4.2}$$

Another second order scheme is the *two-step Adams-Bashforth* scheme

$$\boldsymbol{y}_{k+2} = \boldsymbol{y}_{k+1} + \frac{3\,\delta t}{2}\boldsymbol{f}\left(\boldsymbol{y}_{k+1}, t_{k+1}\right) - \frac{\delta t}{2}\boldsymbol{f}\left(\boldsymbol{y}_k, t_k\right) \tag{4.3}$$

since

$$\begin{aligned} \boldsymbol{y}_{k+2} - \boldsymbol{y}_{k+1} &= \delta t\,\boldsymbol{f}\left(\boldsymbol{y}_{k+1}, t_{k+1}\right) + \frac{\delta t}{2}\left\{\boldsymbol{f}\left(\boldsymbol{y}_{k+1}, t_{k+1}\right) - \boldsymbol{f}\left(\boldsymbol{y}_k, t_k\right)\right\} \\ &= \delta t\,\boldsymbol{f}\left(\boldsymbol{y}_{k+1}, t_{k+1}\right) + \frac{(\delta t)^2}{2}\left\{\boldsymbol{f}\left(\boldsymbol{y}_{k+1}, t_{k+1}\right)\partial_{\boldsymbol{y}_{k+1}}\boldsymbol{f}\left(\boldsymbol{y}_{k+1}, t_{k+1}\right) + \left(\partial_{t_{k+1}}\boldsymbol{f}\right)\left(\boldsymbol{y}_{k+1}, t_{k+1}\right)\right\} + \dots \end{aligned} \tag{4.4}$$

5 Runge-Kutta scheme

An integration scheme often used in applications is the Runge-Kutta. Its simplest implementation is the its second order or mid-point version

$$\begin{aligned} \bar{\boldsymbol{y}}_{k+1} &= \boldsymbol{y}_k + \frac{1}{2}\boldsymbol{f}\left(\boldsymbol{y}_k, t_k\right)\,\delta t \\ \boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + \left\{\boldsymbol{f}\left(\boldsymbol{y}_k, t_k\right) + \boldsymbol{f}\left(\bar{\boldsymbol{y}}_{k+1}, t_k + \frac{\delta t}{2}\right)\right\}\delta t \end{aligned} \tag{5.1}$$

The most used version is, however, the fourth order Runge-Kutta scheme which reads

$$\begin{aligned} \bar{\boldsymbol{y}}_k^{(1)} &= \boldsymbol{f}\left(\boldsymbol{y}_k, t_k\right) \\ \bar{\boldsymbol{y}}_k^{(2)} &= \boldsymbol{f}\left(\boldsymbol{y}_k + \frac{\bar{\boldsymbol{y}}_k^{(1)}\,\delta t}{2}, t_k + \frac{\delta t}{2}\right) \\ \bar{\boldsymbol{y}}_k^{(3)} &= \boldsymbol{f}\left(\boldsymbol{y}_k + \frac{\bar{\boldsymbol{y}}_k^{(2)}\delta t}{2}, t_k + \frac{\delta t}{2}\right) \\ \bar{\boldsymbol{y}}_k^{(4)} &= \boldsymbol{f}\left(\boldsymbol{y}_k + \bar{\boldsymbol{y}}_k^{(3)}\delta t, t_{k+1}\right) \\ \boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + \frac{\bar{\boldsymbol{k}}_n^{(1)} + 2\,\bar{\boldsymbol{y}}_n^{(2)} + 2\,\bar{\boldsymbol{y}}_k^{(3)} + \bar{\boldsymbol{y}}_k^{(4)}}{6}\delta t \end{aligned} \tag{5.2}$$

6 Roundoff error

Beside the errors connatural to discretization there is another important source of discrepancies between numerical approximations and exact solutions. This source is due *round-off errors*. A round-off error is the difference between the calculated approximation of a number and its exact mathematical value. Numbers are represented on a computer with a finite number of digits. Increasing the number of digits allowed in a representation reduces the magnitude of possible round-off errors, but any representation limited to finitely many digits will still cause some degree of round-off error for uncountably many real numbers. Realistic estimates of *accumulated round-off errors* can be obtained by statistical analysis by assuming that *local round-off errors* are *identically distributed random variables*. In particular if

$$s = \#\,(\text{significant digits used to represent numbers}) \tag{6.1}$$

then the local round-off error maybe described as random variable ρ with *uniform distribution* in

$$[-5 \times 10^{-(s+1)}, 5 \times 10^{-(s+1)}] \tag{6.2}$$

In order to obtain a rough estimate of the accumulated round-off error affecting the n-th step of a numerical integration we can compute the typical size of the fluctuations of the sum

$$R = \sum_{i=1}^{n} \rho_i \tag{6.3}$$

around the average

$$\prec R_n \succ = 0 \tag{6.4}$$

This is given by the *variance*

$$\prec (R_n - \prec R_n \succ)^2 \succ = \prec \left(\sum_{i=1}^{n} \rho_i \right)^2 \succ = \sum_{i=1}^{n} \prec \rho_i^2 \succ = n \prec \rho_1^2 \succ \tag{6.5}$$

Note that in the above chain of equalities, the second follows from the first by the vanishing of the average (6.4), the third from the indipendence of each of the random variables in the sum, and the fourth by the assumption of identical distribution. The conclusion is that the estimated size of the round-off error grows after n steps as $\sqrt{n}$.

References

[1] P.E. Kloeden, E. Platen, H. Schurz,
Numerical solution of SDE through computer experiments
Springer, (1994). 1

[2] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery,
Numerical recipes in C: the art of scientific computing
Cambridge University Press (1992) and `www.nr.com`. 1, 4

[3] C. Moler,
Numerical Computing with MATLAB MathWorks web-book,
`http://www.mathworks.com/moler/index_ncm.html`. 1, 3