Integral-preserving integrators

2004 J. Phys. A: Math. Gen. 37 L489

(http://iopscience.iop.org/0305-4470/37/39/L01)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **37** (2004) L489–L495 PII: S0305-4470(04)82709-4

LETTER TO THE EDITOR

Integral-preserving integrators

D I McLaren[1] **and G R W Quispel**[2]

[1] Department of Mathematics, La Trobe University, Victoria 3086, Australia
[2] Department of Mathematics and Centre of Excellence for Mathematics and Statistics of Complex Systems, La Trobe University, Victoria 3086, Australia

E-mail: D.McLaren@latrobe.edu.au and R.Quispel@latrobe.edu.au

Received 1 July 2004, in final form 20 August 2004
Published 15 September 2004
Online at stacks.iop.org/JPhysA/37/L489
doi:10.1088/0305-4470/37/39/L01

Abstract
Ordinary differential equations having a first integral may be solved numerically using one of several methods, with the integral preserved to machine accuracy. One such method is the discrete gradient method. It is shown here that the order of the method can be bootstrapped repeatedly to higher orders of accuracy. The method is illustrated using the Henon–Heiles system.

PACS numbers: 02.60.Gh, 02.60.Nm

1. Introduction

Since about 1990 there has been a great deal of interest in geometric integration, the numerical solution of differential equations while preserving one or more (geometric) properties exactly (i.e. up to round-off error) [1–3, 8, 9, 13]. This has led to symplectic integrators [13], integral-preserving integrators [7], volume-preserving integrators [11, 14], integrators that preserve Lyapunov functions [7], foliations [10], Poisson structure [6], Lie group structure [4], etc.

In this letter, we study the preservation of first integrals (such as energy, momentum, angular momentum, etc) by linear-gradient methods (an alternative, the projection method, is described in [3]). The traditional method of obtaining integral-preserving integrators (IPIs) of higher order of accuracy, is to first construct a second-order IPI, and then to use Yoshida's composition method [15] to obtain higher order IPIs. The purpose of this letter is to introduce a more efficient alternative method.

2. Background

An ordinary differential equation with a first integral $I(x)$,

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x), \qquad \text{with} \quad f(x)\cdot\nabla I(x) = 0, \quad x \in \mathbb{R}^n, \tag{1}$$

can be written[3] [12, 7] in the form

$$\frac{\mathrm{d}x}{\mathrm{d}t} = S(x)\cdot\nabla I(x), \qquad x\in\mathbb{R}^n, \tag{2}$$

where S is a skew-symmetric $n\times n$ matrix. An integral-preserving discrete version of this is

$$\frac{(x'-x)}{\tau} = \tilde{S}(x,x',\tau)\bar{\nabla}I(x,x') \tag{3}$$

where x, x' denote x_n resp. x_{n+1}, and where $\tilde{S}$ is a skew-symmetric matrix satisfying (for consistency)

$$\tilde{S}(x,x',\tau) = S(x) + O(\tau).$$

The general discrete gradient $\bar{\nabla}I$ is defined by

$$(x'-x)\cdot\bar{\nabla}I(x',x) := I(x') - I(x) \tag{4}$$

and may be expanded in the form

$$\bar{\nabla}I(x,x') = \nabla I + B(x)(x'-x) + (x'-x)^T\mathbf{M}(x)(x'-x) + O(\|x'-x\|^3). \tag{5}$$

Substitution of (5) into (4) leads to

$$B_{ij}+B_{ji} = I_{ij}, \qquad \text{and} \qquad M_{ijk}+M_{jki}+M_{kij} = \tfrac{1}{2}I_{ijk}. \tag{6}$$

In this letter, subscript indices will take their usual meaning as labels of vector, matrix or tensor components, except for those involving the integral I, in which case $I_i := \frac{\partial I}{\partial x_i}$, $I_{ij} := \frac{\partial^2 I}{\partial x_i \partial x_j}$, etc. Further, a repeated index in an expression will imply summation over that index.

The order of accuracy of an integral-preserving integrator (IPI) based on (3) is determined by $\tilde{S}$ and by the choice of discrete gradient $\bar{\nabla}I(x,x')$, i.e. by $\tilde{S}$ and the matrix B, the tensor $\mathbf{M}$, and higher order parts of $\bar{\nabla}I$. If a discrete gradient $\bar{\nabla}I$ for which $\bar{\nabla}I(x,x')\neq\bar{\nabla}I(x',x)$, and skew matrix $\tilde{S} = S(x)$ are used, the IPI obtained from (3) is first order.

3. Bootstrapping to higher order

3.1. From first order to second order

We demonstrate a method for 'bootstrapping' the order of an IPI. Note that from now on we restrict to systems having a constant matrix $S(x) = S$, for example Hamiltonian systems. Starting with the first-order integrator

$$\frac{(x'-x)}{\tau} = S_1\bar{\nabla}I(x,x'), \tag{7}$$

where $S_1 := S$, substitution of $(x'-x)$ from (7) into the second term of (5) gives the approximation

$$\bar{\nabla}I(x,x') = (Id + \tau BS)\nabla I(x) + O(\tau^2). \tag{8}$$

Differentiation of (1) for the case of constant S gives

$$\ddot{x} = S\mathcal{H}S\nabla I \tag{9}$$

where $\mathcal{H}$ is the Hessian, $\mathcal{H}_{ij} := \frac{\partial^2 I}{\partial x_i \partial x_j}$. Then

$$(x'-x)_{\text{exact}} - (x'-x)_{\text{1st order IPI}} = \tau^2 SQS\nabla I + O(\tau^3) \tag{10}$$

[3] Under some technical conditions that are generically satisfied (see [7]).

with skew matrix $Q(x) := \frac{1}{2}\mathcal{H}(x) - B(x)$. The RHS of (10) provides a correction term that leads to the second-order integrator

$$\frac{(x'-x)}{\tau} = S_2(x)\bar{\nabla} I(x,x'). \tag{11}$$

Here,

$$S_2(x) := S + \tau S Q(x) S. \tag{12}$$

3.2. From second order to third order

The second-order integral-preserving integrator (11) can similarly be bootstrapped to third order. We obtain

$$(x'-x)_{\text{exact}} - (x'-x)_{\text{2nd order IPI}} = \tau^3 R \nabla I + O(\tau^4) \tag{13}$$

with

$$R := SQSQS - \tfrac{1}{12} S\mathcal{H}S\mathcal{H}S + E \tag{14}$$

where

$$E_{kn} := S_{ki} P_{ijm} S_{jl} I_l S_{mn} \tag{15}$$

with

$$P_{ijm} := \tfrac{1}{6} I_{ijm} - M_{ijm}. \tag{16}$$

It is interesting to note that E is not necessarily skew.

Neverthess, it follows from (6) that

$$P_{ijm} + P_{jmi} + P_{mij} = 0, \tag{17}$$

and hence

$$(\nabla I)^T E \nabla I = 0. \tag{18}$$

We thus obtain a third-order IPI as follows:

$$\frac{x'-x}{\tau} = S_3(x,x')\bar{\nabla} I(x,x'), \tag{19}$$

with

$$S_3 := S_2 + \tau^2 \bar{R} = S + \tau SQS + \tau^2 \bar{R} \tag{20}$$

where

$$\bar{R} := SQSQS - \tfrac{1}{12} S\mathcal{H}S\mathcal{H}S + \bar{E} \tag{21}$$

and

$$\bar{E}_{kn} := S_{ki} P_{ijm} S_{jl} (\bar{\nabla} I)_l S_{mn}. \tag{22}$$

With this definition, even though S_3 will not be skew, we will have

$$(\bar{\nabla} I)^T S_3 \bar{\nabla} I = 0, \tag{23}$$

confirming that (19) is an IPI.

3.3. From third order to fourth order

To gain a fourth-order integrator, one can make the composition $\phi_{\frac{\tau}{2}} \circ \phi^{-1}_{-\frac{\tau}{2}}$, where ϕ is the third-order integrator (19). An alternative would be to bootstrap from third to fourth order. We hope to report on this in the future.

4. A choice of non-symmetric discrete gradient

A suitable discrete gradient is that due to Itoh and Abe [5]:

$$\bar{\nabla}_1 I(x, x') := \begin{pmatrix} \frac{I(x_1', x_2, x_3, x_4, \ldots, x_n) - I(x_1, x_2, x_3, x_4, \ldots, x_n)}{x_1' - x_1} \\ \frac{I(x_1', x_2', x_3, x_4, \ldots, x_n) - I(x_1', x_2, x_3, x_4, \ldots, x_n)}{x_2' - x_2} \\ \vdots \\ \frac{I(x_1', x_2', x_3', x_4', \ldots, x_n') - I(x_1', x_2', \ldots, x_{n-1}', x_n)}{x_n' - x_n} \end{pmatrix}. \tag{24}$$

In this case,

$$B_{ij} = \begin{cases} 0 & \text{if} \quad i < j \\ \frac{1}{2} I_{ii} & \text{if} \quad i = j \\ I_{ji} & \text{if} \quad i > j. \end{cases} \tag{25}$$

Regarding the tensor $\mathbf{M}$, each of the symmetric matrices M_k, $k = 1, \ldots, n$, is defined by

$$M_{kij} = \begin{cases} 0 & \text{for} \quad i, j > k \quad \text{if} \quad k \leqslant n-1 \\ \frac{1}{2} I_{kii} & \text{for} \quad i = 1, 2, \ldots, k-1, \quad j = i \\ \frac{1}{6} I_{iii} & \text{if} \quad i = k, \quad j = i \\ \frac{1}{2} I_{kij} & \text{for} \quad j = 2, 3 \ldots, k-1, \quad i = 1, 2, \ldots, j-1 \\ \frac{1}{4} I_{kik} & \text{for} \quad i = 1, 2 \ldots, k-1, \quad j = k \\ M_{kji} & \text{(symmetric).} \end{cases} \tag{26}$$

Conditions (6) are satisfied by (25) and (26), respectively.

5. Numerical experiments

To demonstrate the advantages of these new integrators we will present a comparison of the discrete mechanics fourth-order integrator (DM) derived by the bootstrap process described above with that found using Yoshida's method [15] (referred to as Yo) to increase the accuracy of a second-order discrete-gradient integrator to fourth order. The second-order integrator used here was [12]

$$\frac{x' - x}{\tau} = S \bar{\nabla}_3 I(x, x'), \tag{27}$$

where

$$\bar{\nabla}_3 I(x, x') := \frac{\bar{\nabla}_1 I(x, x') + \bar{\nabla}_1 I(x', x)}{2}. \tag{28}$$

Both DM and Yo are applied to the (Hamiltonian) four-dimensional Henon–Heiles (H–H) system:

$$\begin{aligned} \dot{x}_1 &= x_3 & \dot{x}_2 &= x_4 \\ \dot{x}_3 &= -x_1 - 2x_1 x_2 & \dot{x}_4 &= -x_2 - x_1^2 + x_2^2 \end{aligned} \tag{29}$$

and first integral (the Hamiltonian)

$$H = \tfrac{1}{2}\left(x_1^2 + x_2^2 + x_3^2 + x_4^2\right) + x_1^2 x_2 - \tfrac{1}{3} x_2^3. \tag{30}$$

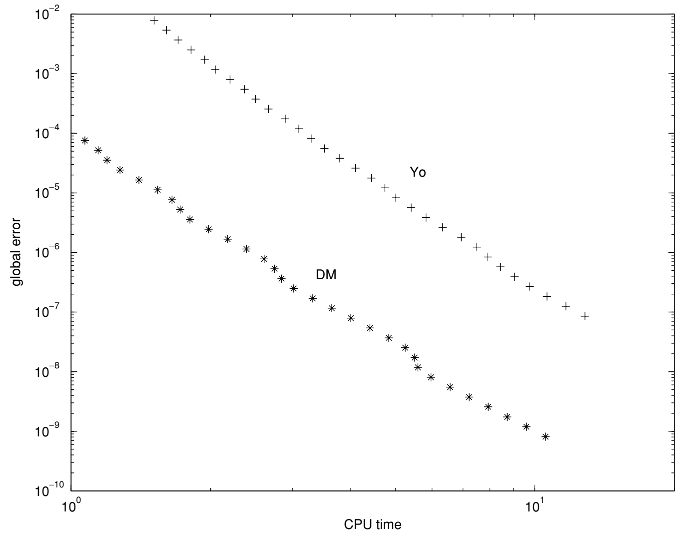

Figure 1. Global error versus CPU time for the two fourth-order integrators DM and Yo, for 31 step sizes—starting at $\tau = 0.08$ and reducing exponentially by a factor of 1.1. The system was integrated up to $t_{\max} = 10^4$ in each case.

In this case, the integrator DM is given by $\phi_{\frac{\tau}{2}} \circ \phi^{-1}_{-\frac{\tau}{2}}$, with ϕ_τ defined by $x' - x = \tau S_3 \bar{\nabla}_1 H$, where $\bar{\nabla}_1 H$ is given by equation (24):

$$\bar{\nabla}_1 H(x, x') = \begin{pmatrix} (x_1 + x_1')\left(\frac{1}{2} + x_2\right) \\ \frac{1}{2}(x_2 + x_2') - \frac{1}{3}\left(x_2^2 + x_2 x_2' + x_2'^2\right) + x_1'^2 \\ \frac{1}{2}(x_3 + x_3') \\ \frac{1}{2}(x_4 + x_4') \end{pmatrix} \tag{31}$$

and where S_3 is the 4×4 matrix

$$S_3(x, x') = \begin{bmatrix} 0 & A \\ -A & B \end{bmatrix}, \tag{32}$$

with

$$A = \begin{bmatrix} 1 + \frac{1}{12}\tau^2(1 + 2x_2) & \frac{1}{6}\tau^2 x_1 \\ \frac{1}{6}\tau^2 x_1 & 1 + \frac{1}{12}\tau^2(1 - 2x_2) \end{bmatrix} \tag{33}$$

and

$$B = \begin{bmatrix} -\frac{1}{6}\tau^2(x_4 + x_4') & -\tau x_1 - \frac{1}{6}\tau^2(x_3 + x_3') \\ \tau x_1 + \frac{1}{3}\tau^2(x_3 + x_3') & 0 \end{bmatrix}. \tag{34}$$

The initial conditions were $x_1 = x_2 = x_3 = x_4 = 0.12$.

A least-squares fit to the global error (E) versus step-size (τ) data for the bootstrapped third- and fourth-order integrators yields (for $t_{\max} = 10^4$), respectively,

$$E \approx 23.083\tau^{3.029} \qquad \text{and} \qquad E \approx 1.855\tau^{4.001}.$$

In figure 1 we show a comparison of global error versus CPU time (for a range of step sizes). In figure 2 we show global error versus time for the same step size on the one hand,

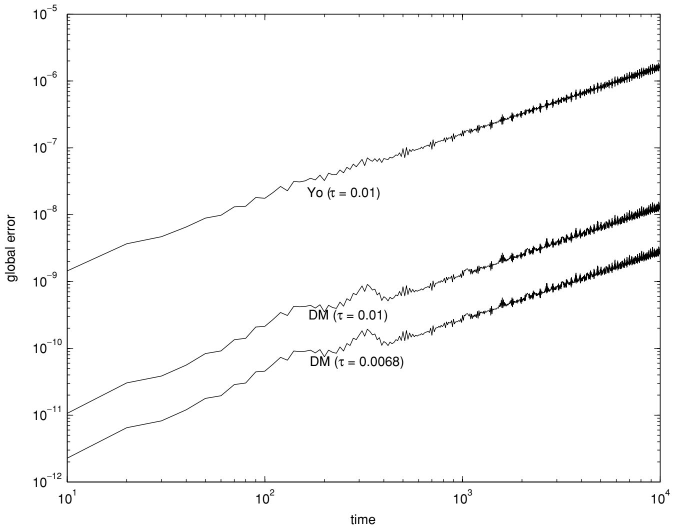

Figure 2. Global error versus time, using the same step size ($\tau = 0.01$) for both integrators, and (for DM) also $\tau = 0.0068$ to achieve the same work (CPU time) as for Yo with $\tau = 0.01$.

and also for the same work on the other hand. All computations were executed using double precision Fortran 77 on a 400 MHz Macintosh G4.

For the data in figure 1, the initial step size was $\tau = 0.08$, reduced by a factor of 1.1 before each repeat of the computations (done 30 times). In figure 2, step size $\tau = 0.01$ was used for the same step-size computations and in the Yoshida method case for the same-work computations. When the DM code was executed, a step size of $\tau = 0.0068$ was used in order to have the same total CPU time as for Yo with step size $\tau = 0.01$.

Acknowledgment

We are grateful to the Australian Research Council for financial support.

References

[1] Budd C J and Iserles A (ed) 1999 Geometric integration: numerical solution of differential equations on manifolds *Phil. Trans. R. Soc.* A **357** 943–1133

[2] Budd C J and Piggott M D 2003 *Geometric integration and its applications, Handbook of Numerical Analysis XI* ed P G Ciarlet and F Cucker (Amsterdam: North-Holland) pp 35–139

[3] Hairer E, Lubich C and Wanner G 2002 *Geometric Numerical Integration (Springer Series in Computational Mathematics* vol 31*)* (Berlin: Springer)

[4] Iserles A, Munthe-Kaas H Z, Norsett S P and Zanna A 2000 Lie-group methods *Acta Numer.* 215–365

[5] Itoh T and Abe K 1988 Hamiltonian-conserving discrete canonical equations based on variational difference quotients *J. Comput. Phys.* **77** 85–102

[6] McLachlan R I 1993 Explicit Lie–Poisson integration and the Euler equations *Phys. Rev. Lett.* **71** 3043–6

[7] McLachlan R I, Quispel G R W and Robidoux N 1999 Geometric integration using discrete gradients *Phil. Trans. R. Soc.* A **357** 1021–45

[8] McLachlan R I and Quispel G R W 2001 Six lectures on the geometric integration of ODEs *Foundations of Computational Mathematics* ed R A DeVore *et al* (Cambridge: Cambridge University Press) pp 155–210

[9] McLachlan R I and Quispel G R W 2002 Splitting methods *Acta Numer.* **11** 341–434

[10] McLachlan R I, Perlmutter M and Quispel G R W 2003 Lie group foliations: dynamical systems and integrators *Future Gener. Comput. Syst.* **19** 1207–19

[11] Quispel G R W 1995 Volume-preserving integrators *Phys. Lett.* A **206** 26–30

[12] Quispel G R W and Turner G S 1996 Discrete gradient methods for solving ODEs numerically while preserving a first integral *J. Phys. A: Math. Gen.* **29** L341–9

[13] Sanz-Serna J M and Calvo M P 1994 *Numerical Hamiltonian Problems* (London: Chapman and Hall)

[14] Zai-jiu S 1994 Construction of volume-preserving difference schemes for source-free systems via generating functions *J. Comput. Math.* **12** 265–72

[15] Yoshida H 1990 Construction of higher order symplectic integrators *Phys. Lett.* A **150** 262–8