Numerical Integration of Differential Equations on Manifolds

Dr. Kaushila Nandan Srivastava, Research Scholar, LNMU Darbhanga.

Abstract

This paper is concerned with the problem of developing numerical inte- gration algorithms for differential equations that, when viewed as equations in some Euclidean space, naturally evolve on some embedded submanifold. It is desired to con- struct algorithms whose iterates also evolve on the same manifold. These algorithms can therefore be viewed as integrating ordinary differential equations on manifolds. The basic method "decouples" the computation of flows on the submanifold from the numerical integration process. It is shown that two classes of single-step and multistep algorithms can be posed and analyzed theoretically, using the concept of "freezing" the coefficients of differential operators obtained from the defining vector field. Ex- plicit third-order algorithms are derived, with additional equations augmenting those of their classical counterparts, obtained from "obstructions" defined by nonvanishing Lie brackets.

Key words, numerical integration, manifold, differential equation flow, lie algebra, algorithm, symbolic computation, frozen coefficients.

1. Introduction

1.1. Numerical Algorithms and Constraints

The flow of many systems of ordinary differential equations preserves certain con- straints. For example, a conservative mechanical system preserves energy, a rotating rigid body in space preserves angular momentum, a Hamiltonian system of equations preserves the symplectic structure, and the configuration variables of a spherical pen- dulum are confined to a sphere. As these examples illustrate, the dynamical variables may be constrained by a conservation law, a manifold or a group structure. More generally, there may be some mixed algebraic-differential equation constraining the dynamic variables. A general-purpose numerical algorithm usually does not preserve these constraints, although there is a large variety of algorithms that do. Without trying to be complete we mention the following: Perhaps the oldest algorithms of this type are those that preserve energy, see Chorin, Hughes, Marsden, and McCracken and the references contained there for examples and further discussion of algorithms of this type. Algorithms that preserve symplectic invariants are also important and quite old. Our interest here is to introduce a class of numerical algorithms that naturally evolve on a constraint manifold. The basic idea is to define the numerical algorithm using certain primitive flows, obtained from the original problem, which have the property that they can be integrated numerically to arbitrarily high order. It turns out that this provides sufficient structure to solve our problem and at the same time is general enough to find application to a variety of interesting examples as described below.

2. Classical Numerical Integration Algorithms

In this section we shall review some classical numerical integration algorithms for a system of differential equations written in the form

= F(z),zER ~, z(0) = zo. (9)

Note that we have not considered time-dependent vector fields F, as we did in equation (I), in order to simplify the notation. However, extension to time~ependent systems is straightforward. We first consider the Runge-Kutta methods described by the following equations•

2.1. Classical (Explicit) Runge-Kutta Algorithms

vl(z) = z,

v2(z) = z + hc2,~F(z),

t.'3(z) = z + h(c3,1F(z) + c3,2F(I.'2(z))), (10)

vr(Z) = z + h(cr,lF(z) + Cr,2F(v2(z)) + "" + cr.~-IF(v~-t(Z))),

Zk+l = Uk+J(zk, h)

= Zk + h(clF(zk) + c2F(v2(z~)) + ... + c~F(vr(Zk))).

Numerical Integration of ODEs on Manifolds.

2.2. Classical (Explicit) Multistep Algorithms

Zk+l = Uk+I(Zl,,Zk-I, ... Zk-r)

(i5)

= Zk + h(c~oF(zk) + alF(zk-1) + ... + arF(Zk-r)).

We say that the algorithm (15) has r + 1 steps. Implicit algorithms may be obtained by including terms

harF(zk-r), r < 0

in the expression (15). In these cases we must rearrange the update equations in some way in order to solve for the most recent iterate. Again, for any particular choice of the constants ok, 0 -< k --- r, which in future we refer to as the constants o~, we may define the order of the algorithm.

3. Numerical Integration Algorithms Adapted to Frames

In this section we shall define our generalizations of the (explicit) Runge-Kutta and multistep algorithms described in the previous section. The adaptation to implicit algorithms is self-evident, even if this implementation would be more problematic than their classical counterparts. In this section we shall deal with an ordinary differential equation described by the equations

Numerical Integration of ODEs on Manifolds

3.1 Explicit Runge-Kutta Algorithms Adapted to a Frame

vl(p) = p, F~(h)(z) = Fp(z),

uz(h, p) = ehC~'~F~p, F~(h)(z) = F~z(h,p)(Z),

~'3(h, p) = ehc3'2F~ehc3'lFpp, F3p(h)(z) -= Fv3(h,p)(Z), (19)

vr(h, p) = e hc .... IF~'-~ e hc"r-2F~-2 ... e hcr'lF~ p, F~p(h)(z) = Fvr(h,p)(Z),

r r-I uk+l(h, p) = ehc~Fpe hc"~Fp ...ehc~F~ P,

Zk+l = uk+t(h, zk) (20)

Note that Vk (0, p) = p, so that F~ (0) = Fp, and the update rule (20) is defined by flows of the vector field F with frozen coefficients. It follows that if the vector fields El ..... En are tangent to a submanifold Ms, as described in the introduction, then the iterates zk will also evolve on Ms, assuming that z0 E M~.

3.2. Explicit Multistep Algorithm Adapted to a Frame

j+l, eh,,Jo&k eh,~{ F=k_i ... eh,~l F j+l uJ(h, Zk, Zk-1 ..... Zk-,', U~) = z~_~u k , (0_< j--< /-1),

u~ = Zk, (22)

= u°(h, zk-1 .....

In this paper we shall be mostly concerned with the case I = 2, although the general case may be analyzed using similar techniques. In the Euclidean case (2t) we easily establish that

• j+l .....)

= h f(Zk) + a{f(Zk-1) + "'" + o~F(Zk-r) + ~k "

4. Simultaneous Approximation of Flows by Flows of Fixed Frame

In this section we study the problem of determining when a mapping ~, as in equation (24), approximates all of the vector fields F, relative to a fixed frame E = [El ..... En}, to order q. In particular, we are interested in computing the Taylor series expansions about h = 0 of the expressions e hF p and ~(h, p). Thus, we must find necessary and sufficient conditions for the following relations to hold:

dk h--0 dk P) h dhJ:~(ehFp) - dhk~bo~(h, , 1 < k< q

5. Calculus of Parameter-Dependent Vector Fields

The definition of the differential operator A(h) is given in terms of the mapping ~, as described in equation (24), as follows:

d -~¢, o ~t,(h, p) = (A(h)O) o ~t,(h, p).

The purpose of this section is to characterize A(h) and its derivatives in terms of the operators Gk(h, p) that define ~. We have the following expansion:

c~ ~(etGk(h' p) p) = Z(Gk (h, p)i ~)(p)(ti/i!),

i=0

with the series converging for small ttl and appropriate conditions on the coefficient functions of Gk(h, p) = ~'=1 g~(h, p)Ej. Note that Gk(0, p) is the zero vector field. Rather than use the geometric interpretation of (p, t) ~ etG~(h'p)p

6. Specific Examples of Algorithms on Manifolds

In this section we shall apply the analysis of the previous two sections to obtain third-order Runge-Kutta and multistep algorithms, adapted to a frame E, as explained in Section 3. The analysis will also demonstrate that third-order algorithms are the first instance of algorithms where the non-Euclidean structure of the frame E, plays a nontrivial role; or in other words, all second-order Euclidean algorithms are second- order with respect to any frame. To apply the analysis of Sections 5 and 4 we need only compute the derivatives Gk(0) (~ using the specific form of the algorithm in question. We first consider the Runge-Kutta algorithms. For third-order algorithms we consider three-stage algorithms as described by equations (19) and (20), explicitly:

ul(p) = p, Flp(h)(z) = Vp(z),

uz(h, p) = ehc2,1F~,p, FZ(h)(z) = Fu~(h,p)(Z),

u3(h, P) = ehc~,2F~ ehc3,1r~ p, F3(h)(z) = V~(h,p)(Z),

Zk + l = e hc3F3zk e hc2F~k ehClF~ Zk = uk+l(h, Zk).

Thus, as in the definition of ~ for the Runge-Kutta algorithms, we obtain

Gl(h, p) = hc3F3(h), Ge(h, p) = hc2F2(h), G3(h, p) = hclFlp(h).

7. Concluding Remarks

Clearly the work above represents only the beginning of a vast program of work, replicating the usual analysis of numerical integration algorithms for ODEs, Butcher [2,3], and others [19.20]. However as is clear from Section 6, any attempt to an- alyze algorithms of order greater than three will be very complex, and our current work is aimed at precisely this problem. We make some observations concerning this issue

7.1. Computing the Constraint Equations

7.2. Solving the Constraint Equations

7.3. Integration of the Flow

References

1. M. Austin, P. S. Krishnaprasad, and L.-S. Wang, Symplectic and Almost Poisson In- tegration of Rigid Body Systems, Proceedings of the 1991 International Conference on Computational Engineering Science, ed. S. Atluvi, Melbourne, Australia, August, (1991).

2. J. C. Butcher, An Order Bound for Runge-Kutta Methods, SIAM J. Numerical Analysis, Vol. 12, pp. 304-315, (I975).

3. J. C. Butcher, The Numerical Analysis of Ordinary Differential Equations, John Wiley, (1986).

4. R Channell, Symptectic Integration for Particles in Electric and Magnetic Fields, (Accel- erator Theory Note, No. AT-6: ATN-86-5). Los Alamos National Laboratory, (1986).

5. E Channell, and C. Scovel, Symplectic Integration of Hamiltonian Systems, submitted to Nonlinearity, June, 1988.

6. A. Chorin, T. J. R. Hughes, J. E. Marsden, and M. McCracken, Product Formulas and Numerical Algorithms, Comm. Pure and Appl. Math., Vol. 31, pp. 205-256, (1978).

7. R E. Crouch, Spacecraft Altitude Control and Stabilization: Application of Geometric Control to Rigid Body Models, [EEE Transactions on Automatic Control, Vol. AC-29, pp. 321-331, (1986).

8. R E. Crouch, and R. L. Grossman, The Explicit Computation of Integration Algorithms and First Integrals for Ordinary Differential Equations with Polynomial Coefficients Us- ing Trees, Proceedings of the 1992 International Symposium of Algebraic and Symbolic Computation, ACM, (1992).

9. R Crouch, R. Grossman, and R. G. Larson, Computations Involving Differential Operators and Their Actions on Functions, Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation, ACM, (1991).

10. P. Crouch, R. Grossman, and R. Larson, Trees, Bialgebras, and Intrinsic Numerical Integra- tors, (Laboratory for Advanced Computing Technical Report, # LAC90-R23). University of Illinois at Chicago, May, (1990).