Computation with continuous-time dynamical systems

P.-A. Absil*

23 March 2005

In this note we review the concept of using continuous-time dynamical systems (described by ordinary differential equations) to solve computational problems.

Many scientific computing problems (such as weather prediction, structural analysis, electrical networks analysis) strongly rely on matrix computation algorithms (linear system solving, eigenvalue decomposition, singular value decomposition, matrix nearness problems, balancing of linear systems, joint diagonalization of matrices...). These algorithms often assume the form of successive iteration,

$$x(k+1) = G(x(k)), \tag{1}$$

which can be viewed as a dynamical system, where the *state* x depends on the "time" k that takes integer values. Equation (1) is thus a *discrete-time (DT) system.* A sequence of points $\{x(k)\}_{k=-\infty}^{\infty}$ satisfying (1) is called the *orbit* of G based at $x(0)$. A simple example of DT dynamical system is the power method,

$$x(k+1) = Ax(k), \tag{2}$$

which computes the dominant eigenvector of the matrix A, i.e., the orbit $x(k)$ converges to an eigendirection of A as k goes to infinity. This and other iterations for matrix computation problems can be found in [GV96].

Interestingly, there exist "natural" physical systems the analysis of which leads directly to a DT dynamical system like (1). An example is the *Bouncing Ball* described in [GH83, §2.4], where a ball repeatedly impacts a sinusoidally vibrating table. This yields a two-dimensional system (because the state of the ball can be determined by the impact time and the velocity at impact) exhibiting rich dynamics (stable and unstable points of period one and higher, bifurcations, chaotic behaviour). However, most physical dynamical systems are continuous-time (CT) systems, which can be described by ordinary differential equations (ODEs) or partial differential equations (PDEs). Here we will only consider ODEs, which assume the form

$$\dot{x}(t) = f(x(t)) \tag{3}$$

where $\dot{x}$ denotes the derivative of x with repect to time. One of the simplest systems that can be described by (3) is the pendulum; see e.g. [Kha96, §1.1] for this and other examples.

CT and DT systems have crucial differences. The question of existence and uniqueness of solutions is an important issue for CT systems (see [HS74, Ch. 8]), not for DT systems. In spite of this, CT systems are arguably easier to analyse than DT systems. A fundamental reason is that the orbits of (classical) CT systems are continuous curves in the state space,

*School of Computational Science, Florida State University, Tallahassee FL 32306-4120, USA.

while the orbits of DT systems are sequences of points, which are more difficult to "track down". This distinction has important topological consequences [GH83]. For example, while a CT system needs at least three state variables to exhibit chaotic behaviour, DT systems of only one variable can be chaotic (see e.g. the *logistic map* $x(k+1) = ax(k)(1-x(k))$ [May76]).

This partly explains a long-standing interest in the numerical analysis community for CT versions of iterative processes. The ideal case is when the orbit of the CT system iterpolates the orbit of the DT system. A simple example is the *power flow*

$$\dot{x} = Bx \tag{4}$$

whose orbits are given by $x(t) = \exp(Bt)x(0)$. It is easy to see that if $x_D(k)$ is an orbit of the power method (2) and $x_C(t)$ is an orbit of the power flow (4) with $x_D(0) = x_C(0)$ and $A = \exp(B)$, then $x_C(k) = x_D(k)$ for all integer k. The result that ignited interest in such flows was when iterates of the unshifted QR-algorithm (which is closely related to the power method; see [Wat82]) were shown to be unit time samples of a particular Lax-pair equation [Fla74, Sym82, DNT83, Nan85].

Oftentimes, however, this interpolation property does not hold and the CT counterpart is simply "related" to the DT system. The relation may be that the DT system is the iteration obtained by applying a numerical integration scheme to the CT system. For example, the Newton iteration $x(k+1) = x(k) - (f'(x(k)))^{-1}f(x(k))$, for finding a zero of the function f, may be regarded as one explicit Euler step with unit steplength applied to the CT system $\dot{x} = -(f'(x))^{-1}f(x)$ [Chu88]. The CT flow may turn out to have interesting properties, quite different from the DT counterpart. For example, in [MA03], a CT flow related to the Rayleigh quotient iteration was shown to visit all eigenvectors of the given matrix in finite time. Such CT flows may also be used to study the asymptotic behaviour of their discrete counterpart, refering to the theory of Ljung [Lju77] and Kushner and Clark [KC78]; see for example [OK85].

Several CT systems with computational properties have also been proposed that do not stem directly from a DT version. A celebrated example is the *double bracket flow* [Bro91]

$$\dot{A} = [A, [A, N]] \tag{5}$$

where $[A, B] = AB - BA$ denotes the commutator. For suitably chosen N, the orbit $A(t)$ converges to a diagonal matrix whose diagonal elements are the eigenvalues of $A(0)$. The same equation (5) can be used to sort lists and solve linear programming problems [Bro91]. Other examples can be found in [WE88, Chu88, CD90, HM94, GM97, Prz03, AS04] and references therein.

Reciprocally to the above-mentioned trend to consider CT counterparts to DT systems, much effort is geared towards deriving iterative methods that efficiently approximate the orbits of CT systems (see [HM94, Ise02, Cas04]) with a view towards implementing these iterations on digital computers. An emerging alternative, however, is to make use of non-classical computer technologies which operate in CT, such as VLSI devices implementing neural networks [HKP91].

References

[AS04] P.-A. Absil and R. Sepulchre, *Continuous dynamical systems that realize discrete optimization on the hypercube*, Systems Control Lett. **52** (2004), no. 3-4, 297–304.

[Bro91] R. W. Brockett, *Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems*, Linear algebra appl. **146** (1991), 79–91.

[Cas04] Fernando Casas, *Numerical integration methods for the double-bracket flow*, J. Comput. Appl. Math. **166** (2004), no. 2, 477–495.

[CD90] M. T. Chu and K. R. Driessel, *The projected gradient method for least squares matrix approximations with spectral constraints*, SIAM Journal Numerical Analysis **27** (1990), no. 4, 1050–1060.

[Chu88] M. T. Chu, *On the continuous realization of iterative processes*, SIAM Review **30** (1988), no. 3, 375–387.

[DNT83] P. Deift, T. Nanda, and C. Tomei, *Ordinary differential equations for the symmetric eigenvalue problem*, SIAM Journal Numerical Analysis **20** (1983), no. 1, 1–22.

[Fla74] H. Flashka, *The Toda latice, II. Existence of integrals*, Physical Review B **9** (1974), no. 4, 1924–1925.

[GH83] John Guckenheimer and Philip Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, Applied Mathematical Sciences, vol. 42, Springer-Verlag, New York, 1983.

[GM97] Neil H. Getz and Jerrold E. Marsden, *Dynamical methods for polar decomposition and inversion of matrices*, Linear Algebra Appl. **258** (1997), 311–343.

[GV96] G. H. Golub and C. F. Van Loan, *Matrix computations, third edition*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 1996.

[HKP91] John Hertz, Anders Krogh, and Richard G. Palmer, *Introduction to the theory of neural computation*, Santa Fe Institute Studies in the Sciences of Complexity. Lecture Notes, I, Addison-Wesley Publishing Company Advanced Book Program, Redwood City, CA, 1991, With forewords by Jack Cowan and Christof Koch.

[HM94] U. Helmke and J. B. Moore, *Optimization and dynamical systems*, Springer, 1994.

[HS74] M. W. Hirsch and S. Smale, *Differential equations, dynamical systems, and linear algebra*, Academic Press, 1974.

[Ise02] Arieh Iserles, *On the discretization of double-bracket flows*, Found. Comput. Math. **2** (2002), no. 3, 305–329.

[KC78] H. J. Kushner and D. S. Clark, *Stochastic approximation methods for constrained and unconstrained systems*, Applied Mathematical Sciences, vol. 26, Springer-Verlag, 1978.

[Kha96] H. K. Khalil, *Nonlinear systems, second edition*, Prentice Hall, 1996.

[Lju77] L. Ljung, *Analysis of recursive stochastic algorithms*, IEEE Trans. Automatic Control **22** (1977), no. 4, 551–575.

[MA03] R. Mahony and P.-A. Absil, *The continuous-time Rayleigh quotient flow on the sphere*, Linear Algebra Appl. **368C** (2003), 343–357.

[May76] R. H. May, *Simple mathematical models with very complicated dynamics*, Nature **261** (1976), 459–467.

[Nan85] T. Nanda, *Differential equations and the QR algorithm*, SIAM Journal Numerical Analysis **22** (1985), no. 2, 310–321.

[OK85] E. Oja and J. Karhunen, *On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix*, J. Math. Anal. Appl. **106** (1985), no. 1, 69–84.

[Prz03] M. Przybylska, *Isospectral-like flows and eigenvalue problem*, Future Generation Computer Systems **19** (2003), no. 7, 1165–1175.

[Sym82] W. W. Symes, *The QR algorithm and scattering for the finite nonperiodic Toda lattice*, Physica D **4** (1982), no. 2, 275–280.

[Wat82] D. S. Watkins, *Understanding the QR algorithm*, SIAM Review **24** (1982), no. 4, 427–440.

[WE88] D. S. Watkins and L. Elsner, *Self-similar flows*, Linear algebra and its applications **110** (1988), 213–242.