Periodic Orbits of Hybrid Systems and Parameter Estimation via AD

Eric Phipps[1], Richard Casey[2], and John Guckenheimer[3]

[1] Sandia National Laboratories, PO Box 5800 MS-0316, Albuquerque, NM 87185
etphipp@sandia.gov
[2] Projet COMORE, INRIA, 2004 route des Lucioles BP 93, 06902 Sophia-Antipolis, France rcasey@sophia.inria.fr
[3] Mathematics Department, Cornell University, Ithaca, NY 14853
gucken@cam.cornell.edu

Rhythmic, periodic processes are ubiquitous in biological systems; for example, the heart beat, walking, circadian rhythms and the menstrual cycle. Modeling these processes with high fidelity as periodic orbits of dynamical systems is challenging because

- (most) nonlinear differential equations can only be solved numerically
- accurate computation requires solving boundary value problems
- many problems and solutions are only piecewise smooth
- many problems require solving differential-algebraic equations
- sensitivity information for parameter dependence of solutions requires solving variational equations
- truncation errors in numerical integration degrade performance of optimization methods for parameter estimation.

In addition, mathematical models of biological processes frequently contain many poorly-known parameters, and the problems associated with this impedes the construction of detailed, high-fidelity models. Modelers are often faced with the difficult problem of using simulations of a nonlinear model, with complex dynamics and many parameters, to match experimental data. Improved computational tools for exploring parameter space and fitting models to data are clearly needed.

This paper describes techniques for computing periodic orbits in systems of hybrid differential-algebraic equations and parameter estimation methods for fitting these orbits to data. These techniques make extensive use of automatic differentiation to accurately and efficiently evaluate derivatives for time integration, parameter sensitivities, root finding and optimization. The boundary value problem representing a periodic orbit in a hybrid system of differential-algebraic equations is discretized via multiple-shooting using a high-degree Taylor series integration method [GM00, Phi03]. Numerical solutions to the

shooting equations are then estimated by a Newton process yielding an approximate periodic orbit. A metric is defined for computing the distance between two given periodic orbits which is then minimized using a trust-region minimization algorithm [DS83] to find optimal fits of the model to a reference orbit [Cas04].

There are two different yet related goals that motivate the algorithmic choices listed above. The first is to provide a simple yet powerful framework for studying periodic motions in mechanical systems. Formulating mechanically correct equations of motion for systems of interconnected rigid bodies, while straightforward, is a time-consuming error prone process. Much of this difficulty stems from computing the acceleration of each rigid body in an inertial reference frame. The acceleration is computed most easily in a redundant set of coordinates giving the spatial positions of each body: since the acceleration is just the second derivative of these positions. Rather than providing explicit formulas for these derivatives, automatic differentiation can be employed to compute these quantities efficiently during the course of a simulation. The feasibility of these ideas was investigated by applying these techniques to the problem of locating stable walking motions for a disc-foot passive walking machine [CGMR01, Gar99, McG91].

The second goal for this work was to investigate the application of smooth optimization methods to periodic orbit parameter estimation problems in neural oscillations. Others [BB93, FUS93, VB99] have favored non-continuous optimization methods such as genetic algorithms, stochastic search methods, simulated annealing and brute-force random searches because of their perceived suitability to the landscape of typical objective functions in parameter space, particularly for multi-compartmental neural models. Here we argue that a carefully formulated optimization problem is amenable to Newton-like methods and has a sufficiently smooth landscape in parameter space that these methods can be an efficient and effective alternative.

The plan of this paper is as follows. In Section 1 we provide a definition of hybrid systems that is the basis for modeling systems with discontinuities or discrete transitions. Sections 2, 3, and 4 briefly describe the Taylor series integration, periodic orbit tracking, and parameter estimation algorithms. For full treatments of these algorithms, we refer the reader to [Phi03, Cas04, CPG04]. The software implementation of these algorithms is briefly described in Section 5 with particular emphasis on the automatic differentiation software **ADMC++** . Finally, these algorithms are applied to the bipedal walking and Hodgkin-Huxley based neural oscillation problems discussed above in Section 6.

1 Hybrid Systems

An important feature of many practical nonlinear problems is the existence of discontinuities or discrete transitions in the problem's dynamics. For example,

in the bipedal walking problem presented below, an impact occurs each time a foot strikes the ground. Modelled as a plastic collision, these impacts create discontinuities in the external force on the system. When solving these systems numerically, it is important to not step over these discontinuities since this can create convergence problems for the numerical method. To this end, we model systems such as these as hybrid system and treat the discontinuities and/or transitions explicitly.

Informally, a hybrid system consists of a set of regions (called "charts") upon each of which a dynamical system is defined, typically an ODE or DAE. Charts are allowed to overlap, and may even belong to different spaces. Each chart V contains an open set U (called a "patch") whose boundary is contained within the union of the zero sets of a set of smooth scalar-valued functions (called "event functions"). It is assumed the closure of U is contained within V and that each event function is positive in U. A trajectory of the hybrid system starts with an initial point in some patch U in a chart V and evolves according to the dynamical system on V. This continues until the boundary of U is reached, at which point some event function g must be zero. Then a transition function is applied mapping that point to a new point in a new path $\bar{U}$ in a new chart $\bar{V}$. It is assumed these transitions are instantaneous. The evolution then continues according to the dynamical system on $\bar{V}$. A periodic orbit in such a system is merely a trajectory that returns to its starting point after some nonzero time T.

For practical purposes, hybrid systems are implemented by including a discrete state s which determines which chart the hybrid system currently belongs to. All functions defining the hybrid system in that chart (vector field, event functions, etc.) take this state as an additional argument. Derivatives with respect to this state are never computed, enabling our automatic differentiation algorithms to operate only on smooth functions. For the algorithms discussed below, we assume the ODE or DAE on each chart is analytic and the event functions and transition functions are C^1 on their domain of definition.

2 Taylor Series Integration

Moore [Moo62, MDJS64] and Barton, Willers, and Zahar [BWZ71, Ric71] implemented general Taylor series methods for computing solutions to ODE iniital value problems in the 1960s and 1970s, followed by work of Corliss and Chang [CC82] and Griewank et al [GJU96]. Guckenheimer and Meloon [GM00] extended these methods to solve boundary value problems for locating periodic orbits of ODEs. At each step of a numerical integration, a degree d truncated Taylor polynomial solution $x(t) = \sum_{k=0}^{d} x_k t^k$ is generated using the Taylor polynomial mode of automatic differentiation [Gri00, Ral81]. In [GM00], Taylor series coefficients were generated using the ADOL-C package [GJU96]. Here, we generate Taylor coefficients using a new package

ADMC++ described in section 5. Typically, we set $d = 40$. Step sizes are estimated by examining the growth rates of Taylor coefficients.

Several authors have extended the Taylor series technique to computing numerical solutions to initial-value problems in DAEs. Chang and Corliss [CC94] describe computing Taylor series solutions to DAEs representing simple mechanical systems. Pryce [Pry98] and Nedialkov and Pryce [NP03] show how to compute Taylor series coefficients for arbitrary DAEs using Pryce's structural analysis [Pry01]. Here we assume the DAE has been converted to an ODE on a constraint manifold:

$$\begin{aligned} \dot{x} &= f(x), \\ F(x) &= 0. \end{aligned} \tag{1}$$

This can be done either explicitly by providing formulas for f and F, semi-automatically using Pryce's structural analysis [Pry01], or implicitly using automatic differentiation and knowledge of the structure of the DAE as is done for the mechanical system example in Section 6. Given a consistent initial condition x_0 such that $F(x_0) = 0$, an approximate Taylor polynomial solution $p(t)$ to the ODE initial-value problem $\dot{x} = f(x)$, $x(0) = x_0$ can be generated in the standard way, and a step size h computed as described above. While $F(p(h))$ will be quite small because of the high order of Taylor series methods, it will not be zero, in general. Moreover, this constraint error typically grows quadratically in the number of time steps [HW96]. This can be remedied by simply projecting each time step back onto the constraints $F = 0$.

What distinguishes our work is the connection of this method to computing Taylor series solutions to reduced ODEs written in terms of local parameterizations of the constraint manifold $\mathcal{M} = \{x : F(x) = 0\}$. In particular, let $f : \mathbf{R}^n \to \mathbf{R}^n$ and $F : \mathbf{R}^m \to \mathbf{R}^m$ be analytic, $F(x_0) = 0$, and assume $\ker DF(x_0)$ has dimension m. Define $p = n - m$, then by the Implicit Function Theorem there are neighborhoods $\mathcal{A} \subset \mathbf{R}^p$ of $0 \in \mathbf{R}^p$ and $\mathcal{B} \subset \mathcal{M}$ of x_0 such that the mapping $\psi : \mathcal{A} \to \mathcal{B}$ defined implicitly by

$$x = \psi(y) : \begin{cases} U_0^T(x - x_0) - y = 0 \\ F(x) = 0 \end{cases} \tag{2}$$

is well-defined and analytic on $\mathcal{A}$. Here the columns of $U_0 \in \mathbf{R}^{n \times p}$ form an orthonormal basis for $\ker D_x F(x_0)$. Clearly $\psi(0) = x_0$ and $F(\psi(y)) = 0$ for each $y \in \mathcal{A}$. Such a mapping is referred to tangent space projection in the literature [PR91, PY91, Yen93] and derives from locally projecting the manifold onto its tangent space. It can be shown that the DAE (1) yields the ODE

$$\dot{y} = U_0^T f(\psi(y)), \quad y(0) = 0, \quad y \in \mathcal{A}. \tag{3}$$

Note that evaluating the right-hand side of the above differential equation requires the solution to the nonlinear system (2), so computing the Taylor coefficients of (3) directly would be quite difficult. However, if $x(t) = \sum_{i=0}^{\infty} x_i t^i$

is a solution to $\dot{x} = f(x)$, $x(0) = x_0$, and $y(t) = \sum_{i=0}^{\infty} y_i t^i$ is a solution to (3), then we must have $y(t) = U_0^T(x(t) - x_0)$. By comparing Taylor coefficient we then find $y_0 = 0$ and $y_i = U_0^T x_i$ for $i > 0$. Since the truncated Taylor polynomial coefficients agree with the infinite expansions up to degree d, this relation holds for the truncated solutions as well. Moreover, given a step size h sufficiently small so that $\tilde{y}(h) \in \mathcal{A}$, the next point in the integration is given by $x_h = \psi(y_h)$ where $y_h = \sum_{i=0}^{d} y_i h^i$. It can be shown that this is equivalent to solving

$$\begin{aligned} U_0^T(x - \tilde{x}(h)) &= 0 \\ F(x) &= 0. \end{aligned} \tag{4}$$

where $x_h = \sum_{i=0}^{d} x_i h^i$. Hence, Taylor series integration of the reduced ODE (3) is equivalent to Taylor series integration of the original ODE $\dot{x} = f(x)$, $x(0) = x_0$ where after each time step the projection $x_h = \Pi(\tilde{x}(h))$ given by the solution to (4) is applied.

Note that these techniques are easily extended to hybrid systems by looking for sign changes in all of the event functions defined for a given chart. If an event function g changes sign over one step of the integration, the time of the event can easily be found by applying Newton's method to the scalar equation $g(x(t)) = 0$. There are simple formulas for computing derivatives of the Taylor polynomial solution with respect to the initial conditions and model parameters [Gri00, Phi03] that are important for the periodic orbit and parameter estimation techniques discussed in the following sections.

3 Periodic Orbits

Periodic orbits of a hybrid system of DAEs are trajectories that return to their starting point after some time T. We implement multiple shooting methods, using Newton's method as a root solver, to locate periodic orbits in systems where each patch has the same dimension n. Incorporation of Newton's method into multiple shooting boundary value solvers requires formulation of regular systems of equations whose roots represent the periodic orbit. The periodic orbit is discretized by selecting a set of points, times and discrete states

$$D = \{(x_i, t_i, s_i), 0 \leq i \leq N\}$$

on the periodic orbit that satisfy two properties:

1. All of the points of the orbit which lie on an event surface are included in D, and
2. $x_0 = x_N$.

In addition, we usually fix t_0. We denote by E the set of indices of the x_i which lie on event surfaces and by e the number of elements in E. The equations that characterize D as a discrete closed orbit are then

$$\begin{aligned}
\Phi(x_{i-1}, t_i - t_{i-1}) &= x_i, && && i-1, i \notin E,\\
\Phi(h(x_{i-1}), t_i - t_{i-1}) &= x_i, && && i-1 \in E, i \notin E,\\
\Phi(x_{i-1}, t_i - t_{i-1}) &= x_i, && g(x_i) = 0, && i-1 \notin E, i \in E,\\
\Phi(h(x_{i-1}), t_i - t_{i-1}) &= x_i, && g(x_i) = 0, && i-1, i \in E,
\end{aligned}$$

where h is the transition function applied to x_{i-1} and g is the event function that vanishes at x_i. In writing these equations, we have suppressed the changes of discrete state that take place at transitions and use the same symbol Φ to denote the flows on the patches containing the trajectory segments.

These equations are underdetermined if there are indices that do not lie in E. The location of the corresponding points on their trajectories has one degree of freedom that is not fixed by the equation $\Phi(x_{i-1}, t_i - t_{i-1}) = x_i$ since, given (x_{i-1}, t_{i-1}) this consists of n equations for the $n+1$ variables (x_i, t_i). Altogether, with $x_0 = x_N$ and t_0 fixed, there are $nN + e$ equations in the $(n+1)N$ variables (x_i, t_i). For a hyperbolic periodic orbit, these equations are a regular system defining a smooth manifold P of dimension $N - e$ [Phi03]. We accept this fact and use a version of Newton's method that is suitable for computing points on P, exploiting the fact that we "know" the tangent space to P. Moving the point (x_i, t_i) infinitesimally along its orbit yields a tangent vector to P that has components $f(x_i)$ and 1 in the appropriate locations of D. Insisting that the Newton updates be orthogonal to these vectors yields a regular system of equations to be solved for the updates. This strategy subsumes the definition of an explicit "phase condition" in the case of a system of ODEs that is not hybrid. The regular system of equations can be viewed as defining a residual function R whose roots, obtained via Newton's method, represent periodic orbits. Jacobian derivatives of R required by Newton's method are computed with the methods mentioned in the previous section.

This constitutes a "bare-bones" multiple shooting solver for periodic orbits of a hybrid system. The sequences of events along the periodic orbit to be calculated are specified in advance, and no attempt is made to modify these in the search for a periodic orbit. Similarly the number of mesh points is fixed and there is no attempt to adapt the mesh to improve the condition number of the Jacobians for Newton's method.

4 Parameter Estimation

We now present an optimization method for estimating parameters for periodic orbit data. For simplicity, we restrict our attention to systems of autonomous ODEs $\dot{x} = f(x, \lambda)$. The extension to hybrid systems is straightforward. Here $\lambda \in \mathbf{R}^p$ is a set of free parameters we wish to vary in order to find a "best fit" of a periodic orbit to empirical data. The method is based upon an objective function that measures the distance between closed curves. We apply trust-region based optimization methods to minimize this objective

function over the space of free parameters λ, assuming that the parameters are restricted to a region in which there is a family of periodic orbits that depend smoothly upon the parameters. Here, we only seek a local minimum, and the regularity of the family of periodic orbits makes this an unconstrained minimization problem.

Assume a time-series $\delta_r = [(t_0, x_0), \ldots, (t_N, x_N)]$ representing a reference periodic orbit is provided (e.g., from empirical data) with period $t_N = T_r$ and a discrete orbit $\delta_c = [(s_0, y_0), \ldots, (s_N, y_N)]$ with period $s_N = T_c$ representing an orbit in the model is computed via the periodic orbit algorithms discussed above. We assume the relative phase offset of the orbits is zero (i.e., $f(y_0)^T(x_0 - y_0) = 0$) and that the orbit mesh points have been computed at the same scaled times, $t_i/T_r = s_i/T_c = \bar{t}_i$, $i = 0, \ldots, N$. We define the distance between the two discrete orbits as

$$d(\lambda) = \sum_{i=1}^{N-1} \|x_i(\bar{t}_i) - y_i(\bar{t}_i, \lambda)\|_2^2 \ (\bar{t}_i - \bar{t}_{i-1}) + \left(\log\left(\frac{T_r}{T_c(\lambda)}\right)\right)^2 \qquad (5)$$

where the dependence on the free parameters λ has been made explicit. The first term in this formula is a Riemann approximation to the L_2 distance between the orbits, and the second term takes into account the discrepancy between the periods. The derivative of $d(\lambda)$ can be computed directly from the defining periodic orbit equations presented in Section 3 using automatic differentiation (see [Cas04]) for more details). While it would be possible to compute the second derivative $\nabla^2 d(\lambda)$ analytically as well using AD, we found a finite-difference approximation by differencing $\nabla d(\lambda)$ to be sufficient to investigate the feasibility of these algorithms.

With the objective function $d(\lambda)$ in hand, we applied trust-region minimization algorithms to find a best fit for the free parameters λ. Trust-region methods are a powerful class of Newton-like methods for solving unconstrained minimization problems that use a quadratic model for the objective function, but constrain each iterate to stay in some local neighborhood of the previous iterate. We implemented a method called the hook step (or "optimal" step) method [Heb73, Mor77, Sor82]. We followed the algorithms presented in [DS83] with minor adjustments to make the algorithm less likely to decrease the trust-region radius [Cas04].

5 Software

Implementations of Taylor series integration, periodic orbit location, and parameter estimation algorithms rely heavily on automatic differentiation to quickly and accurately compute derivatives of the underlying equations. We required an AD package that provides forward, reverse, and Taylor polynomial mode derivative calculations of matrix-valued functions and chose MATLAB as the framework for implementing these algorithms. We sought run times

roughly equivalent to hand-coding the corresponding derivative calculations in MATLAB itself. For the bipedal walking problem discussed in the next section, we also required at least third-degree tensor derivatives of Taylor polynomial coefficients using both the forward and reverse mode (generating the coefficients of the governing DAE and converting it to an ODE on a manifold requires two derivatives of the Lagrangian and constraints and then a third derivative to estimate the derivative of the flow for the periodic orbit location algorithm). While many AD packages satisfy some of these requirements, none at the time of this work satisfied all of them. We therefore created a custom AD package named **ADMC++** to implement the required derivative computations.

ADMC++ is an operator overloading-based AD package for differentiating matrix-valued functions written in MATLAB. For efficiency reasons, all derivative computations are carried out externally to MATLAB in compiled object code originally written in C++ using the MATLAB MEX® interface. Using the MATLAB programming language, a MATLAB class called **amatrix** is provided which overloads many of the matrix-level MATLAB intrinsic functions (e.g., $+$, $*$, $.*$, $/$, $\backslash$ (linear solve), $'$ (transpose), $(\cdot,\cdot)$ (indexing), etc.). By evaluating a function on **amatrix** objects, a computational trace [Gri00] is generated representing the expression graph of the function. The trace data structure is not stored in the MATLAB workspace, but rather is created in external memory through the MEX interface.

Once a trace has been generated, derivatives are computed by looping through the trace in either the forward or reverse directions, and C++ classes are provided for tangent, adjoint, and Taylor polynomial derivatives. A C++ matrix class is provided that implements any required matrix operations using the same underlying routines as MATLAB (LAPACK and BLAS). Traces of the tangent and adjoint computations can be created allowing the computation of arbitrarily high-degree tensor derivatives. The design of this library is very similar to that of FADBAD/TADIFF [BS96], except the underlying data type is a matrix instead of a scalar and the library provides a MATLAB interface.

By taking the MATLAB interpreter out of the forward and reverse sweeps of the trace, we are able to improve performance for these derivative computations, especially for the Taylor polynomial calculations that cannot be completely vectorized to eliminate MATLAB loops. This requires us to evaluate and differentiate each supported MATLAB intrinsic. Given the very large number of possibly differentiable MATLAB intrinsics, this is an arduous task indeed, and only a limited number of operations are currently supported.

A Taylor series integration package **TSINT** and multiple-shooting periodic orbit package **TSPO** have also been written that implement the algorithms discussed above for a wide variety of ODEs, DAEs and hybrid systems. These packages are written entirely in MATLAB and are dependent upon **ADMC++** for all required derivative computations. Further details on these packages can be found elsewhere [Phi03, CPG04].

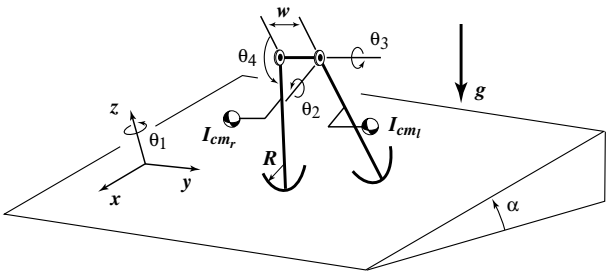

Fig. 1. Schematic diagram of the disc foot passive walker. Drawing based on a diagram of the McGeer walker given in [Gar99].

6 Application to Bipedal Walking

Our first application of these techniques is a periodic orbit location problem in a rigid body mechanical system. The disc-foot passive walker [CGMR01, Gar99, McG91] sketched in Figure 1 serves as a simple model of bipedal walking. It consists of two rigid body legs with unit mass and length connected at the hip by frictionless pin joints and separated by a distance w. The foot of each leg consists of a thin disc of radius R. The walker is placed on a flat plane inclined by an angle α from horizontal. It is assumed one leg is in contact with the ground at all times (the stance leg) while the other swings freely (the swing leg). The foot of the stance leg is in rolling contact with the ground at all times. The only external force on the system other than the contact forces at the stance foot contact point is gravity. Stable walking motions in these passive machines shed light on the ability of humans to walk in a stable manner.

Instantaneously, the system has 4 degrees of freedom: 3 rotation angles of the stance leg around the contact point and 1 rotation angle of the swing leg around the hip axis. Deriving ODE equations of motion of the system in terms of these 4 angles is straightforward, yet algebraically is quite complicated. Our goal was to see how much automatic differentiation could simplify the process of generating mechanically correct equations of motion, but still be able to compute periodic motions of the system to an equivalent level of accuracy as could be obtained from the original ODE system. We formulated the equations of motion of the system as a set of Euler-Lagrange differential-algebraic equations [Gol80, MT95]:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = -G^T(x)\lambda - D_x h(x)^T \mu \tag{6a}$$

$$G(x)\dot{x} + \tilde{g}(x) = 0 \tag{6b}$$

$$h(x) = 0 \tag{6c}$$

where L is the Lagrangian (kinetic minus potential energy). Here x represents the set of generalized coordinates for the system consisting of 3 coordinates for the center of mass position of each leg and 3 Euler angles of rotation around the center of mass for each leg (a total of 12 coordinates). The rolling contact of the stance leg on the ramp gives 2 velocity constraints (Equation 6b) by requiring the instantaneous velocity of the stance leg contact point be zero. The pin joint at the hip and the requirement that the bottom of the stance foot lie on the ramp gives a total of 6 position constraints (Equation 6c). The quantities λ and μ in Equation 6a are undetermined multipliers that must be computed along with the solution. Given algebraic formulas for the Lagrangian L, position constraint function h, and velocity constraint function $g(x, \dot{x}) = G(x)\dot{x} + \tilde{g}(x)$, automatic differentiation is used to compute the necessary derivatives appearing in the DAE, drastically simplifying the amount of programming effort required to compute them. The system is clearly hybrid with two states. Each chart has one event function given by the height of the swing foot above the ramp, and one transition function. The transition function is derived by considering angular momentum conservation around the new contact point [Phi03].

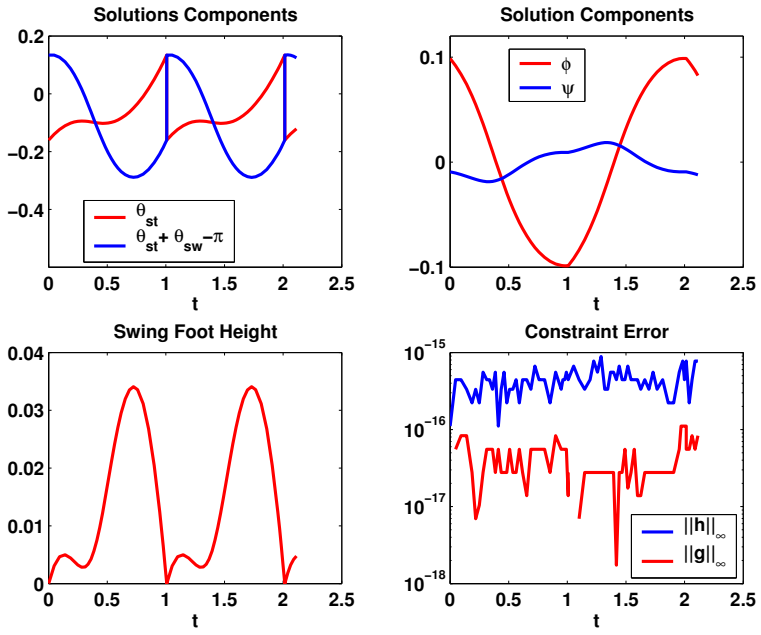

Fig. 2. Periodic trajectory of the disc foot walker for one complete step of the system (two strides of the walker). (a) Stance leg angle θ_{st} and swing leg angle measured from the ramp normal $\theta_{st} + \theta_{sw} - \pi$. (b) Swing foot height e. (c) Steer ϕ and lean ψ. (d) L_∞ norm of constraint error for DAE solution.

We compared the DAE system with a MATLAB ODE model written by Garcia [Gar98]. The DAE system consisted of approximately 60 lines of MATLAB code that whereas the ODE model had approximately 240 lines and was much harder to derive and verify. To compare the accuracy of these methods, the periodic orbit algorithms using Taylor series integration discussed in section 3 were applied using parameters and initial conditions found in [CGMR01]. At these parameter values, the system has a stable periodic orbit (Figure 2). For both systems, 3 mesh points were used in the periodic orbit solver (1 for each transition plus 1 mesh point not lying on an event surface). An error tolerance of 1.0e-16 was used in the Taylor series integration, along with an event solver tolerance of 1.0e-15. For the DAE calculation, the projection solver tolerance was set to 5.0e-16. For the DAE system, the initial residual of the periodic orbit equations was 6.9e-5 and 2 Newton iterations were required to reduce the residual to 1.3e-15. For the ODE system, the initial residual was 3.5e-5 and also took 2 Newton iterations to reduce the residual to 1.8e-15. The DAE calculation took approximately 12 times longer than the ODE calculations. The L_2 distance between these orbits was calculated as discussed in Section 4 and was found to be approximately 3.9e-14. The L_2 distance between the initial orbit for the ODE periodic orbit calculation and the final orbit was calculated as well, and was found to be approximately 3.2e-5. The eigenvalues of the return map for both the ODE and DAE periodic orbits were also calculated. The largest difference between eigenvalues was found to be approximately 7.6e-14 with the magnitude of the largest eigenvalue equal to 0.70418256213669 (ODE). This agrees with the eigenvalue of $(0.8391560)^2$ given in [CGMR01] to 2.4e-7, very near the expected error provided in [CGMR01]. These results show the DAE Taylor series periodic orbit method has nearly the same accuracy as the ODE method. They provide strong independent verification of the results in [CGMR01] regarding the existence of a stable walking motion.

We next apply the techniques described in Section 4 to the Hodgkin-Huxley model [HH52]. The Hodgkin-Huxley (H-H) equations model electrical excitability of squid axon and are the archetype of conductance-based models of neural oscillations. They constitute a four-dimensional vector field with several parameters that produces periodic oscillations in some parameter regimes. We used the H-H equations as a test-case for the parameter estimation algorithm described earlier, using $(\bar{g}_{\mathrm{Na}}, \bar{g}_K, I_{\mathrm{ext}})$ as active parameters. Their "standard values" are $(120, 36, -20)$. We began by fitting the Hodgkin-Huxley model to an ideal reference orbit generated using the H-H equations at perturbed parameters values $(140, 36, -20)$. This gave a reference orbit with period $T_r = 11.2082849$ ms. We took an approximation δ_r to this reference orbit with $N = 30$ meshpoints, and used the trust-region algorithm to look for an optimal fit of the Hodgkin-Huxley model to δ_r, starting from the standard H-H parameter values. These parameter values give a starting δ_c with $T_c = 1.4574003\mathrm{e}{+01}$. The iteration converged in 8 steps, computing an optimal value of $d(\lambda_*) = 1.2483041\mathrm{e}{-12}$ and $\|\mathrm{relgrad}(\lambda_*)\|_\infty = 7.9200748\mathrm{e}{-}$

10 , with parameters $\lambda_* = (1.3999986e{+}02, 3.5999964e{+}01, -1.9999989e{+}01)$. and period $T_* = 11.2082847966$. These values are very close to the reference values.

To examine the effects of noise in the reference orbit on the convergence of the parameter estimation algorithm, we fixed the starting data at the standard Hodgkin-Huxley parameter values and added Gaussian noise to the reference data. For each run, we replaced the voltage time-series $\{V_i^r\}_{i=1}^N$ for the reference orbit with $\{N(V_i^r, \sigma)\}_{i=1}^N$, where $N(\mu, \sigma)$ is the normal distribution with mean μ and standard deviation σ. As the variance of the noise increased, the algorithm still converges to a minimum, but the minimum is increasingly farther away from the noise-free minimum, both in terms of optimal parameter values and the minimum value of $d(\lambda)$ achieved. The results of these computations are summarized in Table 1 and Figure 3.

Table 1. Results for noisy reference data with increasing variance. The distance $d(\lambda_*)$, the norm of the relative gradient and optimal values found for the active parameters are shown at each value of the variance σ^2.

σ^2	$d(\lambda_*)$	$\|\text{relgrad}(\lambda_*)\|_\infty$	$\bar{g}_{\text{Na}*}$	$\bar{g}_{K*}$	$I_{\text{ext}*}$
0	1.2483e-12	7.9201e-10	1.4000e+02	3.6000e+01	-2.0000e+
0.1	7.6301e-03	3.3220e-09	1.3767e+02	3.5362e+01	-1.9793e+01
0.5	3.8202e-02	6.8566e-10	1.3505e+02	3.4650e+01	-1.9563e+01
2.0	1.5318e-01	1.4228e-05	1.3098e+02	3.3551e+01	-1.9213e+01
4.0	3.0699e-01	1.5196e-06	1.2814e+02	3.2792e+01	-1.8977e+01
16.0	1.2361	6.6402e-08	1.2110e+02	3.0973e+01	-1.8448e+01

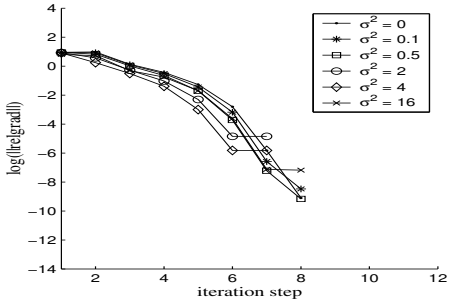

Fig. 3. Effects of noisy δ_r on convergence to $\nabla d(\lambda_*) \approx 0$. The trust-region algorithm is seen to converge to local minima even for large variance noise in δ_r. See the output in Table 1 for more details.

Note that even for large variance σ, the convergence of the trust-region algorithm is indicated by the small values of relgrad(λ_*). For example, with

$\sigma = 4$ we found a local minimum with $d(\lambda_*) = 1.2361$, $\|\text{relgrad}(\lambda_*)\|_\infty =$ 6.6402e-08 and optimal parameter values $\lambda_* =$ (1.2110e+02, 3.0973e+01, -1.8448e+01). These results indicate that the optimization algorithm is robust with respect to noise in the reference data: a local minimum of the objective function is still found with a a small value for $\|\text{relgrad}\|_\infty$, indicating good convergence. Moreover, the parameters for the minima with noisy data approach the noise-free reference data as $\sigma \to 0$.

7 Conclusions

Taylor series integration and automatic differentiation provide a powerful set of tools for computing periodic orbits in hybrid systems of ODEs and DAEs, and for parameter estimation to fit these orbits to data. Taylor series integration allows numerical trajectories to be computed to high accuracy with large step sizes and provides dense output for accurate event location. Through the use of tangent space parameterization, the standard Taylor series algorithm for ODEs has a simple extension to DAEs formulated as an ODE on a constraint manifold. Furthermore, automatic differentiation can be employed to simplify the conversion of the DAE to an ODE on a manifold, and as shown in the bipedal walking example, simplify the derivation of the governing DAE itself. Taylor series integration coupled with automatic differentiation provides a simple mechanism for computing the derivative of the numerical flow of the ODE or DAE with respect to initial conditions and model parameters. These properties, coupled with a simple multiple-shooting framework allow the accurate computation of periodic orbits using very coarse discretizations in an efficient manner. Computing these orbits accurately is critical for further analysis such as parameter estimation, since loss in accuracy directly leads to a loss in accuracy degrades the performance of Gauss-Newton optimization algorithms. Maintaining accuracy in the periodic orbit computation is necessary to ensure smoothness of the objective function and the amenability of Newton-based optimization methods for these parameter estimation problems.

A new automatic differentiation library **ADMC++** was presented facilitating the derivative and Taylor polynomial calculations required to implement these algorithms. The library provides the forward, reverse, and Taylor polynomial automatic differentiation modes for functions written in MATLAB but performs all derivative calculations in compiled object code for efficiency. All three modes can be combined to produce arbitrarily high-degree tensor derivatives of Taylor polynomial coefficients.

References

[BB93] U. S. Bhalla and J. M. Bower. Exploring parameter space in detailed single neuron models: simulations of the mitral and granule cells of the olfactory bulb. *Journal of Neurophysiology*, 69(6):1948–1964, 1993.

[BM] A. Back, J. Guckenheimer and M. Myers. *A Dynamical Simulation Facility for Hybrid Systems. Lecture Notes in Compter Science.* 736, 255-267, 1993.

[BS] K. S. Brown and J. P. Sethna. Statistical mechanics approaches to models with many poorly known parameters. *Physical Review E.* 68, 21904–21904(9), 2003.

[BS96] C. Bendtsen and O. Stauning. FADBAD, a flexible C++ package for automatic differentiation. Technical Report IMM–REP–1996–17, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, August 1996.

[BWZ71] D. Barton, I. M. Willers, and R. V. M. Zahar. The automatic solution of systems of ordinary differential equations by the method of Taylor series. *The Computer Journal*, 14(3):243–248, 1971.

[Cas04] Richard James Casey. *Periodic Orbits in Neural Models: Sensitivity Analysis and Algorithms for Parameter Estimation.* PhD thesis, Cornell University, 2004.

[CC82] George Corliss and Y. F. Chang. Solving ordinary differential equations using Taylor series. *ACM Transactions on Mathematical Software*, 8(2):114–144, 1982.

[CC94] Y. F. Chang and G. Corliss. ATOMFT: solving ODEs and DAEs using taylor series. *Computers Math. Applic.*, 28(10–12):209–233, 1994.

[CGMR01] Michael J. Colemen, Mariano Garcia, Katja Mombaur, and Andy Ruina. Prediction of stable walking for a toy that cannot stand. *Physical Review E*, 64, 2001.

[CPG04] Richard Casey, Eric Phipps, and John Guckenheimer. Periodic orbit computation: Hybrid systems and parameter estimation. *In preparation*

[DS83] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equation.* SIAM, Philadelphia, 1983.

[FUS93] W. R. Foster, L. H. Ungar, and J. S Schwaber. Significance of conductances in Hodgkin-Huxley models. *Journal of Neurophysiology*, 70(6):2502–2518, 1993.

[Gar98] Mariano Garcia. Disc foot walker simulation code. Personal Communication, 1998.

[Gar99] Mariano Sylvio Garcia. *Stability, Scaling, and Chaos in Passive-Dynamic Gait Models.* PhD thesis, Cornell University, 1999.

[GJU96] A. Griewank, D. Juedes, and J. Utke. ADOL-C, A Package for the Automatic Differentiation of Algorithms Written in C/C++ . *ACM Trans. Math. Software*, 22:131–167, 1996. `http://www.math.tu-dresden.de/wir/project/adolc`.

[GM00] John Guckenheimer and Brian Meloon. Computing periodic orbits and their bifurcations with automatic differentiation. *SIAM J. Sci. Comput.*, 22(3):951–985, 2000.

[Gol80] Herbert Goldstein. *Classical Mechanics.* Addison Wesley, Reading, Second edition, 1980.

[Gri00] Andreas Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* Frontiers in Applied Mathematics. SIAM, Philadelphia, 2000.

[Heb73] M. D. Hebden. An algorithm for minimization using exact second derivatives. Technical report, A.E.R.E., Harwell, England, 1973. Report TP515.

[HH52] Hodgkin, A. L. and Huxley, A. F. A quantitative description of membrane current and its applications to conduction and excitation in nerve. *J. Physiol.* (Lond.), 116 (1952), 500–544.

[HW96] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II.* Number 14 in Springer Series in Computation Mathematics. Springer, Berlin, Second edition, 1996.

[McG91] T. McGeer. Passive dynamic biped catalogue. In *Proceedings of the 2nd International Symposium of Experimental Robotics*, New York, 1991. Springer-Verlag.

[MDJS64] R. E. Moore, J. A. Davidson, H. R. Jaske, and S. Shayer. DIFEQ integration routine—user's manual. Technical Report LMSC 6-90-64-6, Lockheed Missilies and Space Co., Palo Alto, CA, 1964.

[Moo62] R. E. Moore. *Interval Arithmetic and Automatic Error Analysis in Digital Computing.* PhD thesis, Department of Computer Science, Stanford University, 1962.

[Mor77] J. J. Moré. *The Levenberg-Marquardt Algorithm: Implementation and Theory*, volume 630 of *Lecture Notes in Mathematics*, pages 105–116. Springer-Verlag, New York, 1977.

[MT95] Jerry B. Marion and Stephen T. Thornton. *Classical Dynamics of Particles and Systems.* Saunders College Publishing, Fort Worth, Fourth edition, 1995.

[NP03] Nedialko S. Nedialkov and John D. Pryce. Solving differential-algebraic equations by taylor series (i): Computing taylor coefficients. Preprint

[Phi03] Eric Todd Phipps. *Taylor Series Integration of Differential-algebraic Equations: Automatic Differentiation as a Tool for Simulating Rigid Body Mechanical Systems.* PhD thesis, Cornell University, 2003.

[PR91] Florian A. Potra and Werner C. Rheinboldt. On the numerical solution of Euler-Lagrange equations. *Mech. Struct. & Mach.*, 19(1):1–18, 1991.

[Pry98] J. D. Pryce. Solving high-index daes by taylor series. *Numerical Algorithms*, 19:195–211, 1998.

[Pry01] J. D. Pryce. A simple structural analysis method for daes. *BIT*, 41(2):364–294, 2001.

[PY91] Florian A. Potra and Jeng Yen. Implicit numerical integration for Euler-Lagrange equations via tangent space parameterization. *Mech. Struct. & Mach.*, 19(1):77–98, 1991.

[Ral81] L. B. Rall. *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Computer Science.* Springer Verlag, Berlin, 1981.

[Ric71] John R. Rice, editor. *Mathematical Software*, chapter Taylor Series Methods for Ordinary Differential Equations—An Evaluation, pages 369–390. Academic Press, New York, 1971.

[Sor82] D. C. Sorensen. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19:409–426, 1982.

[VB99] M. C. Vanier and J. M. Bower. A comparative survey of automated parameter-search methods for compartmental neural models. *Journal of Computational Neuroscience*, 7:149–171, 1999.

[Yen93] Jeng Yen. Constrained equations of motion in multibody dynamics as ODEs on manifolds. *SIAM J. Numer. Anal.*, 30(2):553–568, April 1993.