Dynamics of Continuous, Discrete and Impulsive Systems
Series **B:** *Applications & Algorithms* **21 (2014)** 137-148

NUMERICAL ACCURACY OF TWO BENCHMARK MODELS OF WALKING: THE RIMLESS SPOKED WHEEL AND THE SIMPLEST WALKER

Pranav A. Bhounsule
Disney Research Pittsburgh
4720 Forbes Avenue, Lower Level, Suite 110
Pittsburgh, PA, 15213 USA.

Abstract. The eigenvalues of the Jacobian of the return map are used to quantify the stability of discrete dynamical systems, such as, the rimless wheel and the simplest walker. The accuracy this Jacobian, usually obtained by finite differencing, depends on the step size. Even with the most optimal step size, only moderate accuracy is obtained. Here, we obtain the Jacobian by numerically integrating the gradient of the equations of motion. For the rimless wheel, our eigenvalue estimate is accurate to 12 significant digits and is better than 9 significant digits obtained by finite differencing with optimal step size obtained by Coleman [Dynamics of Continuous, Discrete and Impulsive Systems Series B, 16, 2009]. We first show that our method is able to produce the eigenvalues accurate to 12 significant digits obtained by known analytical solution for the rimless wheel. This benchmark calculation then permits us to make the claim that the eigenvalues of the simplest walker, for which the analytical solution is unknown, obtained using our method are accurate to 12 significant digits.
Keywords. Walking, Simplest walker, Rimless wheel, fixed point, Poincaré return map, stride function, periodic motion, discrete dynamics, intermittent contact, eigen-values.

1 Introduction

Our main motivation is to improve the accuracy of eigenvalue computation of passive dynamic walking machines like those reported by Coleman [3]. Passive dynamic walking (a term coined by McGeer [5]), refers to machines that exploit their natural dynamics (mass, inertia, placement of center of mass, etc.) to walk downhills without any control input or energy input.

The eigenvalues of the Jacobian of the return map quantify the stability of the system. The system is stable, neutrally stable or unstable depending on whether the maximum of the moduli of the eigenvalues of the Jacobian of the return map is less than 1, equal to 1 or greater than 1 respectively. Accurate eigenvalue computation is especially important when the system is close to being unstable [8].

Coleman [3] computes the Jacobian of return map by finite differencing. The accuracy of his computation depends on the appropriate tuning of the step size; which if too big, leads to truncation errors or if too small, leads to roundoff errors. Coleman gives a method of tuning the step size based on a tradeoff between truncation and round-off errors that achieves the best numerical accuracy.

In this paper, we obtain the Jacobian of the continuous dynamics by integrating the gradient of the continuous dynamics, and we obtain the Jacobian of the discrete dynamics by finding the gradient of the event function and the discrete jump condition assuming they are first order continuous. The product of the Jacobian of the continuous dynamics and the discrete dynamics gives the Jacobian of the return map. The accuracy of our computation depends on the error tolerance of the numerical integration and of the root finder; both of which can be set arbitrarily high but within machine precision. Our method has the following advantages over finite differencing: 1) There is no parameter tuning involved, and, 2) We obtain an accuracy close to machine precision by appropriately setting the error tolerance for the numerical integration and root finder. For the rimless wheel problem, our eigenvalues are accurate to 12 significant digits and better than 9 digits accuracy reported by Coleman using finite differencing.

We also develop numerical code to analyze legged machines and to compute the eigenvalues. In our analysis, we follow an approach similar to Coleman. We proceed by first writing numerical code to analyze the rimless wheel whose analytical solutions are already known. Next, using the same code but changing the equations of motion to correspond to the simplest walker, we give benchmark solutions for the simplest walker for which the analytical solution is unknown. To the best of our knowledge, we have not seen benchmark solutions for the simplest walker reported anywhere; although, this is a well studied problem. We also provide MATLAB code that we used to generate the results in this paper.

The other applications of this paper are:

1. The methods and code provided here can be used as a tutorial for getting started with legged system simulation. The models presented here have the following advantages: i) They are simple as they have only 1 or 2 degrees of freedom. ii) The equations of motion can be easily derived by hand or by using a symbolic algebra package. iii) They move passively under the influence of gravity and hence there is no need to develop a control law. (iv) It is fairly easy to incorporate actuation in these models to test control strategies.

2. The benchmark models can be used to test accuracy of simulations of higher degree of freedom models by reducing them to special cases (for example, see Appendix B in [1]).

3. Some control theorists have tried optimizing for the eigenvalues to compute control laws that induce asymptotically stable walking [7]. By using accurate eigenvalues we expect these smooth optimization methods to converge to an optimum faster and not get stuck in a local optima.

4. Our computation of the Jacobian of the Poincaré map for systems with discontinuities can be extended to linearize the equations of motion of such systems. This is useful for developing model based controllers and model based state estimators.

2 Problem Description

Figure 1 (a) shows the 2-D rimless wheel and (b) shows the simplest walker. We will analyze these two passive dynamic walking models. Figure 1 (c) shows a typical step of the simplest walker. The simplest walker starts at (i), the instance at which the front leg has just made contact with the floor and the trailing leg has just begin to swing. At this instant, we denote the time as t_0 and the state of the system as $\boldsymbol{x_0}$. The walker moves from (i) at time t_0 to (v) at time t_1 as per the ordinary differential equation, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$. At (v), the system reaches the event given by $\boldsymbol{h}(\boldsymbol{x}(t_1)) = 0$. This event corresponds to instance when the swing leg is going to touch the ground. At this event, there is a discrete jump in the state variables due to collision of the swing leg and the legs exchange roles; the swing leg becomes the new stance leg and vice versa. We describe the support exchange by the equation, $\boldsymbol{x}^{+}(t_1) = \boldsymbol{g}(\boldsymbol{x}(t_1))$ and the systems ends in the state (vi). We summarize the complete set of equations that describe a walking step below:

$$
\begin{aligned}
t = t_0 : &\qquad \boldsymbol{x}(t_0) = \boldsymbol{x_0}, && (1)\\
t_0 \le t \le t_1 : &\qquad \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), && (2)\\
t = t_1 : &\qquad h(\boldsymbol{x}(t_1)) = 0, && (3)\\
t = t_1 : &\qquad \boldsymbol{x}^{+}(t_1) = \boldsymbol{g}(\boldsymbol{x}(t_1)) && (4)
\end{aligned}
$$

Using equations 1 to 4, we can also define a function $\boldsymbol{S}(\boldsymbol{x})$ (called the stride map) that maps the state at beginning of the step to the end of the step as follows,

$$
\boldsymbol{x}^{+}(t_1) = \boldsymbol{S}(\boldsymbol{x(t_0)}) = \boldsymbol{S}(\boldsymbol{x_0}) \tag{5}
$$

For our benchmark calculations we do the following: i) Find the steady state solution of the system. That is, we want to find the initial state $\boldsymbol{x_0}$ that will result in $\boldsymbol{x_0} = \boldsymbol{S(x_0)}$. $\boldsymbol{x_0}$ is the period one fixed point of the stride function and the resulting map is called the Poincaré return map. ii) Evaluate the stability of the steady state solution. We obtain this by computing the eigenvalues of the linearized map $(\boldsymbol{S_x}(\boldsymbol{x_0}))$. A system is stable if the maximum of the moduli of the eigenvalues of $\boldsymbol{S_x}$ is less than 1 and unstable if the biggest eigenvalue is greater than 1. If the biggest eigenvalue is equal to 1 than the system is neutrally stable [9]. The next section describes a numerical method for computing accurate values of the Jacobian $\boldsymbol{S_x}$.

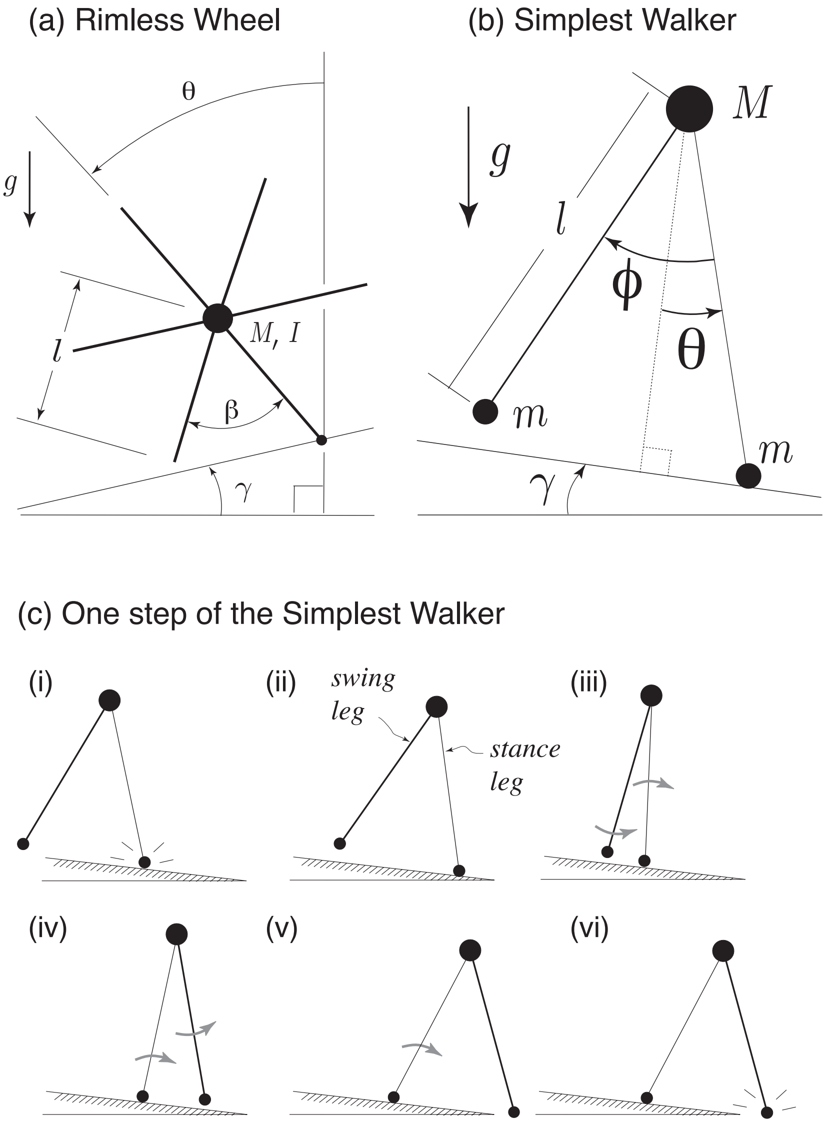

Figure 1: (a) Rimless wheel analyzed by Coleman. Figure source is [3]. (b) Simplest walker analyzed by Garcia et al. (c) The simplest walker starts in (i), the state in which the front leg is the stance leg and the trailing leg is the swing leg. The walker moves under the influence of gravity and momentum through (ii) to (v) as shown. We ignore scuffing of the swing leg with the ground. Finally in (vi), the swing leg collides with the ground and becomes the new stance leg. At this point, we have a complete gait cycle with the state configuration (vi) being the same as (i). Figure source for (b) and (c) is Garcia et al. [4] and has been used with permission.

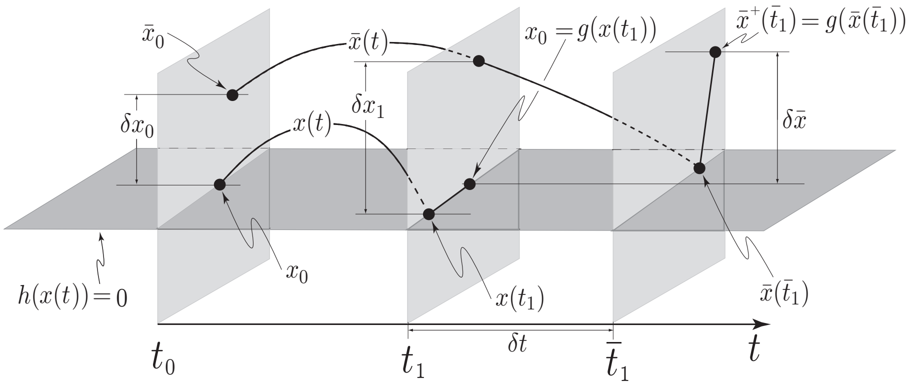

Figure 2: Graphical illustration of the evolution of the period one fixed point and of a small perturbation about the fixed point. The steady state solution starts at the fixed point, $\boldsymbol{x_0}$, and evolves along $\boldsymbol{x}(t)$. At time t_1, this trajectory intersects the surface $h = 0$, followed by the jump condition $\boldsymbol{g}$ that brings the state back to the initial condition $\boldsymbol{x_0}$. A perturbed initial condition, $\boldsymbol{\bar{x}_0} = \boldsymbol{x_0} + \delta \boldsymbol{x_0}$, on the other hand, evolves along $\boldsymbol{\bar{x}}(t)$. At time $\bar{t}_1 \neq t_1$, this perturbed trajectory intersects the surface $h = 0$, followed by a jump condition $\boldsymbol{g}$, and finally ending in the state $\boldsymbol{\bar{x}}^+ \neq \boldsymbol{x_0} \neq \boldsymbol{\bar{x}_0}$.

2.1 Derivation of the Jacobian of the Poincaré map

We will find the Jacobian of the stride function at the fixed point, $\boldsymbol{S_x}(\boldsymbol{x_0})$. Referring to figure 2, we rewrite the Jacobian as follows:

$$\begin{aligned} \boldsymbol{S_x} &= \frac{\delta \boldsymbol{\bar{x}}}{\delta \boldsymbol{x_0}} \\ &= \underbrace{\frac{\delta \boldsymbol{x_1}}{\delta \boldsymbol{x_0}}}_{\boldsymbol{S_x^c}} \underbrace{\frac{\delta \boldsymbol{\bar{x}}}{\delta \boldsymbol{x_1}}}_{\boldsymbol{S_x^d}} \end{aligned} \tag{6}$$

In the above expression, $\boldsymbol{S_x^c}$ is the Jacobian of the continuous dynamics, and $\boldsymbol{S_x^d}$ is the Jacobian of the discontinuous dynamics. We derive an expression for the two Jacobians next.

2.1.1 Jacobian of continuous dynamics ($\boldsymbol{S_x^c}$):

We substitute $\boldsymbol{x}$ with $\boldsymbol{\chi}$ in equation 2 to get, $\dot{\boldsymbol{\chi}} = \boldsymbol{f}(\boldsymbol{\chi})$. Next, we differentiate this expression with respect to $\boldsymbol{x}$ [1] to get,

$$\frac{d}{dt}\left(\frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{x}}\right) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{x}}. \tag{7}$$

In the above equation we have used the fact that the order of differentiation is commutative to interchange $\boldsymbol{x}$ and t.

Now put $\boldsymbol{F} = \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{x}}$ in equation 7 and noting that $\boldsymbol{F}(\boldsymbol{x_0}) = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix, we get the following matrix differential equation,

$$t_0 \le t \le t_1 : \qquad \dot{\boldsymbol{F}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\boldsymbol{F}, \qquad \boldsymbol{F}(\boldsymbol{x_0}) = \boldsymbol{I}. \tag{8}$$

We obtain the Jacobian of the continuous case by integrating equation 8 till time t_1 to get,

$$\boldsymbol{S^c_x} = \frac{\delta \boldsymbol{x_1}}{\delta \boldsymbol{x_0}} = \boldsymbol{F}(\boldsymbol{x}(t_1)) \tag{9}$$

2.1.2 Jacobian of discontinuous dynamics($\boldsymbol{S^d_x}$):

Using terms defined in figure 2, we do a first order Taylor series expansion of the term $h(\bar{\boldsymbol{x}}(\bar{t}_1)) = 0$.

$$\begin{aligned}
0 = h(\bar{x}(\bar{t}_1)) &= h(\bar{\boldsymbol{x}}(t_1 + \delta t)) \\
&= h(\bar{\boldsymbol{x}}(t_1) + \boldsymbol{f}(\boldsymbol{x}(t_1))\delta t) \\
&= h(\boldsymbol{x}(t_1) + \delta \boldsymbol{x_1} + \boldsymbol{f}(\boldsymbol{x}(t_1))\delta t) \\
&= \cancelto{0}{h(\boldsymbol{x}(t_1))} + \boldsymbol{h_x}(\boldsymbol{x}(t_1))\Big[\delta \boldsymbol{x_1} + \boldsymbol{f}(\boldsymbol{x}(t_1))\delta t\Big] \\
&= \boldsymbol{h_x}(\boldsymbol{x}(t_1))\Big[\delta \boldsymbol{x_1} + \boldsymbol{f}(\boldsymbol{x}(t_1))\delta t\Big] \qquad (10)
\end{aligned}$$

In the final two expressions, $\boldsymbol{h_x}$ is the Jacobians of $\boldsymbol{h}$ with respect to the state $\boldsymbol{x}$. Using equation 10, we can solve for δt to get,

$$\delta t = -\frac{\boldsymbol{h_x}(\boldsymbol{x}(t_1))\delta \boldsymbol{x_1}}{\boldsymbol{h_x}(\boldsymbol{x}(t_1))\boldsymbol{f}(\boldsymbol{x}(t_1))}. \tag{11}$$

Next, using terms defined in figure 2, we do a first order Taylor series

[1] For systems of the form $\dot{\boldsymbol{\chi}} = \boldsymbol{f}(\boldsymbol{\chi}) = \boldsymbol{A^{-1}}(\boldsymbol{\chi})\boldsymbol{b}(\boldsymbol{\chi})$, using the fact that $\boldsymbol{A^{-1}A} = \boldsymbol{I}$, we can write $\frac{\partial \boldsymbol{A^{-1}}}{\partial \boldsymbol{x}} = -\boldsymbol{A^{-1}}\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{x}}\boldsymbol{A^{-1}}$ and thus, $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \boldsymbol{A^{-1}}(-\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{x}}\boldsymbol{A^{-1}b} + \frac{\partial \boldsymbol{b}}{\partial \boldsymbol{x}})$. This way we do not have to evaluate $\frac{\partial \boldsymbol{A^{-1}}}{\partial \boldsymbol{x}}$, which is a computationally expensive calculation.

expansion of the term $\boldsymbol{g}(\bar{\boldsymbol{x}}(\bar{t}_1))$.

$$\begin{aligned}
\boldsymbol{g}(\bar{\boldsymbol{x}}(\bar{t}_1)) &= \boldsymbol{g}(\bar{\boldsymbol{x}}(t_1+\delta t)) \\
&= \boldsymbol{g}(\bar{\boldsymbol{x}}(t_1)+\boldsymbol{f}(\boldsymbol{x}(t_1))\delta t) \\
&= \boldsymbol{g}(\boldsymbol{x}(t_1)+\delta\boldsymbol{x_1}+\boldsymbol{f}(\boldsymbol{x}(t_1))\delta t) \\
&= \boldsymbol{g}(\boldsymbol{x}(t_1))+\boldsymbol{g_x}(\boldsymbol{x}(t_1))\Big[\delta\boldsymbol{x_1}+\boldsymbol{f}(\boldsymbol{x}(t_1))\delta t\Big] \\
&= \boldsymbol{g}(\boldsymbol{x}(t_1))+\boldsymbol{g_x}(\boldsymbol{x}(t_1))\delta\boldsymbol{x_1}+\boldsymbol{g_x}(\boldsymbol{x}(t_1))\boldsymbol{f}(\boldsymbol{x}(t_1))\delta t. \qquad (12)
\end{aligned}$$

In the final two expressions, $\boldsymbol{g_x}$ is the Jacobians of $\boldsymbol{g}$ with respect to the state $\boldsymbol{x}$.

From figure 2 we can compute the differential $\delta\bar{\boldsymbol{x}}$ as follows:

$$\delta\bar{\boldsymbol{x}} = \boldsymbol{g}(\bar{\boldsymbol{x}}(\bar{t}_1)) - \boldsymbol{g}(\boldsymbol{x}(t_1)) \qquad (13)$$

Now putting equation 12 in equation 13 we get,

$$\delta\bar{\boldsymbol{x}} = \boldsymbol{g_x}(\boldsymbol{x}(t_1))\delta\boldsymbol{x_1}+\boldsymbol{g_x}(\boldsymbol{x}(t_1))\boldsymbol{f}(\boldsymbol{x}(t_1))\delta t. \qquad (14)$$

Next we substitute the value of δt from equation 11 into equation 14 and re-arrange to get the Jacobian of the discontinuous dynamics,

$$\boldsymbol{S^d_x} = \frac{\delta\bar{\boldsymbol{x}}}{\delta\boldsymbol{x_1}} = \boldsymbol{g_x}(\boldsymbol{x}(t_1)) - \frac{\boldsymbol{g_x}(\boldsymbol{x}(t_1))\boldsymbol{f}(\boldsymbol{x}(t_1))}{\boldsymbol{h_x}(\boldsymbol{x}(t_1))\boldsymbol{f}(\boldsymbol{x}(t_1))}\boldsymbol{h_x}(\boldsymbol{x}(t_1)) \qquad (15)$$

2.1.3 Jacobian of the complete system

By substituting equations 9 and 15 in equation 6, we obtain the Jacobian ($\boldsymbol{S_x}$) of the complete system and is given below

$$\boldsymbol{S_x} = \boldsymbol{S^c_x}\boldsymbol{S^d_x} = \boldsymbol{F}(x(t_1))\boldsymbol{g_x}(\boldsymbol{x}(t_1))\left\{\boldsymbol{I} - \frac{\boldsymbol{f}(\boldsymbol{x}(t_1))\boldsymbol{h_x}(\boldsymbol{x}(t_1))}{\boldsymbol{h_x}(\boldsymbol{x}(t_1))\boldsymbol{f}(\boldsymbol{x}(t_1))}\right\} \qquad (16)$$

3 Benchmark results for rimless wheel

3.1 Equations of motion

Figure 1 (a) shows the 2-D rimless wheel analyzed by Coleman. The center of the mass of the rimless wheel is at intersection of the spokes. The mass is M, inertia about center of mass is I, leg length is ℓ, number of spokes is n, gravity is g and the slope of the ramp is γ. We calculate the inter-spoke angle β from the number of spokes, that is, $\beta = 2\pi/n$.

The equations of motion described below for the rimless wheel are taken from Coleman [3]. The angle between the spoke touching the ramp and the vertical direction is θ. Note that time is non-dimensionalized with $\sqrt{g/\ell}$ in the equations below.

We define the constants λ and μ as below:

$$\lambda^2 = \frac{M\ell^2}{I + M\ell^2}, \quad \mu = 1 + \lambda^2(\cos\left(\frac{2\pi}{n}\right) - 1). \tag{17}$$

We can now write the equations of motion using the above constants as follows:

$$\boldsymbol{x} = \begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix}, \qquad \boldsymbol{f} = \begin{bmatrix}\dot{\theta} \\ \lambda^2 \sin(\theta + \gamma)\end{bmatrix}, \qquad \boldsymbol{f_x} = \begin{bmatrix}0 & 1 \\ \lambda^2 \cos(\theta + \gamma) & 0\end{bmatrix},$$
$$h = \theta^- - \pi/n, \qquad \boldsymbol{h_x} = \begin{bmatrix}1 & 0\end{bmatrix}, \qquad \boldsymbol{g} = \begin{bmatrix}-\theta^- \\ \mu\dot{\theta}^-\end{bmatrix}, \qquad \boldsymbol{g_x} = \begin{bmatrix}-1 & 0 \\ 0 & \mu\end{bmatrix}. \tag{18}$$

3.2 Analytical solution

The fixed point can be calculated analytically and is given by [3]:

$$\boldsymbol{x_0} = \begin{bmatrix}\theta_0 \\ \dot{\theta}_0\end{bmatrix} = \begin{bmatrix}-\frac{\pi}{n} \\ \{\frac{4\mu^2\lambda^2 \sin\frac{\pi}{n} \sin\alpha}{1-\mu^2}\}^{\frac{1}{2}}\end{bmatrix} \tag{19}$$

Similarly the analytical expression for the Jacobian, $\boldsymbol{S_x}(\boldsymbol{x_0})$, is given by:

$$\boldsymbol{S_x}(\boldsymbol{x_0}) = \begin{bmatrix}0 & 0 \\ -\frac{\mu\lambda^2 \sin(\alpha - \frac{\pi}{n})}{2\sqrt{\frac{\lambda^2 \sin\frac{\pi}{n} \sin\alpha}{1-\mu^2}}} & \mu^2\end{bmatrix} \tag{20}$$

The eigen-values, σ_1 and σ_2, of the Jacobian of equation 20 are given by:

$$\sigma_1 = 0, \qquad \sigma_2 = \mu^2. \tag{21}$$

Our benchmark will be for the parameter set $\lambda^2 = 2/3$, $\alpha = 0.2$ and $n = 6$. Putting values of λ and μ in equation 17 gives, $\mu = 2/3$. Further, the fixed point from equation 19 is $\boldsymbol{x_0} = [-0.5235987755982, 0.460341126609458]$. The eigenvalues from equation 21 are, $\sigma_1 = 0$ and $\sigma_2 = \mu^2 = 4/9 = 0.4444444444444\bar{4}$.

3.3 Numerical Solution

To compute the fixed point we develop a simulation in MATLAB. At the core of the simulation is the stride function $\boldsymbol{S}(\boldsymbol{x_0})$ (see equation 5), that takes the state of the system after heel-strike, $\boldsymbol{x_0}$, and returns the state after the next heel-strike. This involves integrating the equations of motion for one step given by equation 2 up to the event given by equation 3 and application of the jump condition given by equation 4. We use a variable order Adams-Bashforth-Moulton (*ode113* in MATLAB) with the relative and absolute tolerance of 10^{-13}. We do heel-strike detection using MATLAB's event detector within *ode113*, and is accurate to machine accuracy. To find the fixed point $\boldsymbol{x_0}$ that satisfies equation 5, we use a MATLAB's root finder function *fsolve* and specify an accuracy of 10^{-12} on the root.

One of the limitations of *fsolve* is that it needs a good initial guess ($\boldsymbol{x_0}$) for convergence. Generating a good initial guess for the rimless wheel is easy. This is because, though there are two states, the wheel angle θ_0 is always π/n at heel-strike (see $\boldsymbol{g}$ in equation 18). Thus, we need to search for only the wheel angular rate, $\dot{\theta}_0$. We have found that a forward simulation in conjunction with an animation can speed up the search of an initial guess for the root finder.

Using numerical simulation we compute the fixed point to be
$\boldsymbol{x_0} = [-0.523598775598299, 0.460341126609455]$, which is accurate to 12 decimal places (based on root finder tolerance) when compared with the analytical solution.

To compute the Jacobian of the stride-function $\boldsymbol{S}(\boldsymbol{x})$, we used two methods: 1) A central difference with a perturbation size of 10^{-5}, and, 2) The formulae 16 and which is accurate to root finder tolerance.

Using a numerical perturbation of 10^{-5} we computed the eigenvalues to be
$\sigma_1 = 0.000000000883332$ and $\sigma_2 = 0.444444444604858$, which are accurate to 9 decimal places. Similarly using integration and the formulae 16 derived in equation 2.1, we computed the eigenvalues to be $\sigma_1 = 0$ and $\sigma_2 = 0.444444444444440$, the latter of which is accurate to 13 decimal places.

4 Benchmark results for the simplest walker

4.1 Equations of motion

Figure 1 (b) shows the simplest walker analyzed by Garcia et al. The model has a point mass M at the hip, point feet of mass m, leg length is ℓ, gravity is g, and ramp slope is γ. Garcia et al. did two simplifications: 1) They nondimensionalized time with $\sqrt{g/\ell}$, and 2) They analyzed the limiting case, $m/M \rightarrow 0$. The net effect of these two simplifications is that the equations of motion have only one free parameter, the slope γ. The equations of motion for the simplest walker are from [4].

$$
\begin{aligned}
\boldsymbol{x} &= \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix}, \qquad
\boldsymbol{f} = \begin{bmatrix} \dot{\theta} \\ \sin(\theta-\gamma) \\ \dot{\phi} \\ \sin(\theta-\gamma) + \dot{\theta}^2\sin(\phi) - \cos(\theta-\gamma)\sin(\phi) \end{bmatrix}, \\
\boldsymbol{f_x} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ \cos(\theta-\gamma) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \cos(\theta-\gamma)+\sin(\theta-\gamma)\sin(\phi) & 2\dot{\theta}\sin(\phi) & \dot{\theta}^2\cos(\phi)-\cos(\theta-\gamma)\cos(\phi) & 0 \end{bmatrix}, \\
h &= \phi^- - 2\theta^-, \quad \boldsymbol{h_x} = \begin{bmatrix} -2 & 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{g} = \begin{bmatrix} -\theta^- \\ \cos(2\theta^-)\dot{\theta}^- \\ -2\theta^- \\ \{1-\cos(2\theta^-)\}\cos(2\theta^-)\dot{\theta}^- \end{bmatrix}, \\
\boldsymbol{g_x} &= \begin{bmatrix} -1 & 0 & 0 & 0 \\ -2\sin(2\theta^-)\dot{\theta}^- & \cos(2\theta^-) & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 2\sin(2\theta^-)\{2\cos(2\theta^-)-1\}\dot{\theta}^- & \{1-\cos(2\theta^-)\}\cos(2\theta^-) & 0 & 0 \end{bmatrix}.
\end{aligned}
\tag{22}
$$

The simplest walker has no analytical solution. We use the numerical code developed during the analysis of the rimless wheel to compute the fixed point and evaluate the eigenvalues. Our benchmark are for $\gamma = 0.009$.

4.2 Numerical solution

We use the method described in section 3.3 to find the fixed point and stability of the walker. To generate an initial guess for the root solver we do the following. We note that the swing leg angle and rate of the robot after heel-strike (see **g** in 22) is dependent only on the stance leg angle and rate. Thus our search space for the initial state is $\boldsymbol{x_0} = [\theta_0, \dot{\theta}_0, 2\theta_0, \{1-\cos(\theta)\}\dot{\theta}_0]$, where our search is for the 2 parameters; θ_0 and $\dot{\theta}_0$. Like the rimless wheel, we use the forward simulation in conjunction with an animation to speed up the search for the values of the 2 parameters, θ_0 and $\dot{\theta}_0$.

For the slope $\gamma = 0.009$, we found two fixed points. This matches with the results obtained by Garcia et al. Table 1 row 1, shows the fixed point we found and are accurate to 12 decimal places. Further, we computed the eigenvalues using central difference with a step size of 10^{-5} and which are accurate tow 5 decimal places and are listed in Table 1, row 2 (λ_{perturb}). Finally, using the analytically derived Jacobian we computed the eigenvalues accurate to 12 decimal places and which are listed in Table 1, row 3 ($\lambda_{\text{analytical}}$).

5 Conclusion and extensions

Our main goal was to benchmark the eigenvalues of the Jacobian of the return map for discrete dynamical systems. We obtain the Jacobian of the continu-

	Stable solution	Unstable solution
$\boldsymbol{x_0}$	$\begin{bmatrix} 0.200310900544287 \\ -0.199832473004977 \\ 0.400621801088574 \\ -0.015822999948318 \end{bmatrix}$	$\begin{bmatrix} 0.193937369810184 \\ -0.203866927442010 \\ 0.387874739620369 \\ -0.015144260853192 \end{bmatrix}$
λ_{perturb}	$\begin{bmatrix} 0 \\ 0.000000001586465 \\ -0.190099639402167 - i0.557599274284362 \\ -0.190099639402167 + i0.557599274284362 \end{bmatrix}$	$\begin{bmatrix} -0.000000000000002 \\ -0.000000005231481 \\ 0.459589047035257 \\ 4.003865226079296 \end{bmatrix}$
$\lambda_{\text{analytical}}$	$\begin{bmatrix} 0 \\ -0.000000000000211 \\ -0.190099841069399 - i0.557598776679489 \\ -0.190099841069399 + i0.557598776679489 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.000000000000023 \\ 0.459589589797698 \\ 4.003864358929642 \end{bmatrix}$

Table 1: Fixed points (first row and denoted by $\boldsymbol{x_0}$), eigenvalues using central difference (second row and denoted by λ_{perturb}), eigenvalues by numerically propagating the Jacobian (third row and denoted by $\lambda_{\text{analytic}}$) for the simplest walker for slope, $\gamma = 0.009$. The fixed points are accurate to 12 decimal places. The eigenvalues computed by central difference and with perturbation size of 10^{-5} and are accurate to 5 decimal places while the eigenvalue computed by numerically propagating the Jacobian via integration are accurate to 12 decimal places.

ous dynamics by integrating the gradient of the continuous dynamics, and we obtain the Jacobian of the discrete dynamics by finding the gradient of the event function and the discrete jump condition assuming they are first order continuous. The product of these two Jacobians gives the Jacobian of the return map. We checked our eigenvalue computation against the analytical solution available for the rimless wheel. We then gave benchmark results for the simplest walker whose analytical solution is unknown. In the process of benchmarking the eigenvalues, we also developed numerical code to analyze the rimless wheel and the simplest walker.

This paper and accompanying code can be used as a tutorial for getting started with legged systems. Some possible extensions to the models presented in this paper are: 1) Add hip and ankle actuation to the simplest walker to test control strategies for level ground walking [2]. 2) Add an actuated upper body to the simplest walker and use the upper body to power walking on level ground [6]. 3) Use the benchmark results to test optimization code (for examples see Appendix D in [1]).

6 Online supplementary material

The MATLAB code that was used to generate the results in this paper can be found online:

- Rimless Wheel: `http://ruina.tam.cornell.edu/~pab47/rimlesswheel.m` OR `http://tiny.cc/pranavb_rimlesswheel`

- Simplest Walker: `http://ruina.tam.cornell.edu/~pab47/simplestwalker.m` OR `http://tiny.cc/pranavb_simplestwalker`

References

[1] P.A. Bhounsule. *A controller design framework for bipedal robots: trajectory optimization and event-based stabilization.* PhD thesis, Cornell University, 2012.

[2] P.A. Bhounsule. Control of a compass gait walker based on energy regulation using ankle push-off and foot placement. *Robotica*, 2014 (in press).

[3] M. Coleman. Numerical accuracy case studies of two systems with intermittent dynamics: A 2D rimless spoked wheel and a 3D passive-dynamics model of walking. *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms*, 16:59–87, 2009.

[4] Mariano Garcia, Anindya Chatterjee, Andy Ruina, and Michael Coleman. The simplest walking model: Stability, complexity, and scaling. *ASME J. of Biomech. Eng.*, 120(2):281–288, 1998.

[5] Tad McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990.

[6] Terumasa Narukawa, Masaki Takahashi, and Kazuo Yoshida. Efficient walking with optimization for a planar biped walker with a torso by hip actuators and springs. *Robotica*, 29(4):641–648, 2011.

[7] C. Shih, J.W. Grizzle, and C. Chevallereau. Asymptotically stable walking of a simple underactuated 3d bipedal robot. In *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pages 2766–2771. IEEE, 2007.

[8] G. Steisberg. Stability of a passive dynamic hopper. unpublished.

[9] S. Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry and engineering.* Perseus Books Group, 2001.

Received September 2013; revised April 2014.

email: journal@monotone.uwaterloo.ca
http://monotone.uwaterloo.ca/∼journal/