

## A Fast Eulerian Approach for Computation of Global Isochrons in High Dimensions\*

Miles Detrixhe<sup>†</sup>, Marion Doubeck<sup>†</sup>, Jeff Moehlis<sup>†</sup>, and Frédéric Gibou<sup>‡</sup>

**Abstract.** We present a novel Eulerian numerical method to compute global isochrons of a stable periodic orbit in high dimensions. Our approach is to formulate the asymptotic phase as a solution to a first-order boundary value problem and solve the resulting Hamilton–Jacobi equation with the parallel fast sweeping method. All isochrons are then given as isocontours of the phase. We apply this method to the Hodgkin–Huxley equations and a model of a dopaminergic neuron which exhibits mixed mode oscillations. Our results show that this Eulerian scheme is an efficient, accurate method for computing the asymptotic phase of a periodic dynamical system. Furthermore, by computing the phase on a Cartesian grid, it is simple to compute the gradient of phase, and thus compute an “almost phaseless” target set for the purposes of desynchronization of a system of oscillators.

**Key words.** isochrons, neuron models, mixed mode oscillation (MMO), desynchronization, Hamilton–Jacobi, parallel, high dimensions

**AMS subject classifications.** 37C10, 65P99, 37N25

**DOI.** 10.1137/140998615

**1. Introduction.** A powerful technique for analyzing the dynamics of nonlinear oscillators is phase reduction, with a single variable for an oscillator describing the phase of the oscillation with respect to some reference state [28, 50, 18, 20]. The resulting reduction in the dimensionality and complexity of a system often retains enough information to yield a useful understanding of the system’s dynamics (see, e.g., [26, 2, 16, 6, 32]) and can allow for the implementation of phase-based control algorithms (see, e.g., [43, 33, 48]).

Phase reduction is obtained through the partition of the basin of attraction of a stable periodic orbit into isochrons, which are sets of points that give trajectories which converge with the same phase on the periodic orbit [49]. Isochrons can be interpreted as the level sets of an appropriate phase function. Since isochrons capture the phase properties of the system, they provide a useful characterization of the system’s dynamics and its response to control inputs.

Various methods for computing isochrons have been developed, with recent approaches including [15, 37, 30, 19]. While such numerical techniques hold promise, they are challenged by oscillator models with high dimensionality. The main problem stems from the “curse of

\*Received by the editors December 5, 2014; accepted for publication (in revised form) by H. Osinga May 4, 2016; published electronically August 11, 2016.

<http://www.siam.org/journals/siads/15-3/99861.html>

**Funding:** This work was supported by the National Science Foundation grant NSF-1363243 and Office of Naval Research grant ONR N00014-11-1-0027.

<sup>†</sup>Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070 ([mdetrixhe@engineering.ucsb.edu](mailto:mdetrixhe@engineering.ucsb.edu), [mariondoubeck@yahoo.fr](mailto:mariondoubeck@yahoo.fr), [moehlis@engineering.ucsb.edu](mailto:moehlis@engineering.ucsb.edu)).

<sup>‡</sup>Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070, and Department of Computer Science, University of California, Santa Barbara, CA 93106-5110 ([fgibou@engineering.ucsb.edu](mailto:fgibou@engineering.ucsb.edu)).

dimensionality” as termed by Bellman [4]: the computational cost increases exponentially with the dimension of the mathematical model. Existing computational approaches are severely limited in the dimensionality of the problems they can feasibly analyze.

The two general approaches for solving for isochrons are to use either Lagrangian or Eulerian methods. Lagrangian methods aim to compute the isochron explicitly, i.e., identify all points that belong to a particular isochron. An Eulerian method computes the isochrons implicitly as isocontours of the phase. Eulerian methods have the advantage that once the phase field is computed, *all isochrons are known implicitly*. Explicit representations of the isochrons can then be obtained by using an isocontouring algorithm during the postprocessing stage (e.g., with MATLAB or ParaView). Often, the research goal is to have a picture of the isochrons in a specific region. Lagrangian methods that propagate the isochron from the boundary data will not have knowledge of a particular isochron’s point of intersection with the orbit a priori. Further, Eulerian methods make it simple to compute derivatives of the phase with finite differences. This can be useful for control applications and is demonstrated in our results. Lagrangian methods, however, have the advantage that by computing a single isochron at high accuracy, very fine details can be resolved (e.g., near the phaseless set).

Such grid-based methods, however, are limited by memory and computational expense with increasing dimension. Applications that require the solution at only a few points can avoid the use of a grid and employ Lagrangian methods to problems with much higher dimensionality. Here we focus on applications where it is important to compute isochrons in the entire domain, for which Lagrangian methods are much less efficient.

In this paper, we introduce a robust Eulerian computational method that enables the efficient computation of isochrons in dimensions higher than is possible with current state-of-the-art techniques. Our Eulerian technique has the added benefit that it allows for the simple computation of the almost phaseless set for the purpose of desynchronizing control. We mitigate the curse of dimensionality by first formulating the isochrons as isocontours of the solution of a Hamilton–Jacobi (HJ) PDE and then coupling an efficient algorithm for solving the HJ equation with parallel computer hardware. We demonstrate the method by analyzing a three-dimensional (3D) model of a neuron which exhibits mixed mode oscillations and the four-dimensional (4D) Hodgkin–Huxley (HH) model for neural dynamics and a particular two-dimensional (2D) reduction. The results can be used to better understand the geometry and phase dynamics of each system.

Much research has been done on solving static (time-independent) HJ problems. A marching approach for solving linear PDEs was developed in [1]. Both the method of Tsitsiklis [45] and the fast marching method (FMM) [40] were designed to solve the eikonal equation, which is a specific static HJ equation. The FMM has been adapted into a family of ordered upwind methods (OUMs) [42, 41] that can be applied to many static HJ problems. These are single-pass methods in which each grid point is computed in the order of its dependence on the boundary data. The computational complexity of these methods is  $O(N \log N)$ , where  $N$  is the total number of grid points.

The fast sweeping method (FSM) [52] was also developed to solve the eikonal equation and was generalized to solve many other static HJ equations [44, 24, 38]. The FSM is an  $O(N)$  method that does not exactly compute the dependence graph of all grid points as does the FMM. Instead, the FSM builds on an older approach for linear PDEs [29] and HJ equations

[5] which uses alternating Gauss–Seidel iterations. The FSM updates grid points in several alternating orderings that, in a finite number of sweeps, will propagate information along all possible directions of dependence to the entire domain. The number of sweeps is not strictly grid independent for general PDEs (as shown in Remark 4.4 in [7]). However, in practice the number of iterations to convergence is bounded by the geometry of the characteristic curves of the underlying PDE. For this reason, the method is said to be of computational complexity of  $O(N)$ , even though a proof only exists for a subset of eikonal equations.

All of the above fast methods require grid points to be updated in a specific order, so parallelization is not trivial. The fast iterative method [21] is a parallel marching method for the eikonal equation that gives good speedups on CPUs and GPUs for certain problems. Two parallel sweeping methods were developed in [53], but are limited. The first is limited to a small number of parallel threads, and the second drastically increases the number of iterations to converge. An efficient parallel sweeping method for the eikonal equation in arbitrary dimensions was developed in [13]. In addition to sweeping and marching methods, there exists a family of two-scale methods [7] and their parallel versions [8] designed to combine the best features of marching and sweeping, while also exhibiting excellent parallel scalability.

We are currently developing a more powerful parallel FSM [12] designed for use on modern supercomputers. The HJ problem being considered here could be solved with a marching, sweeping, or two-scale method, but for the above reason, we chose to develop the FSM for the computation of isochrons. The significance of this work is that by combining the HJ formulation with a highly parallel algorithm, we can solve problems that are intractable with previously existing methods.

This paper is organized as follows. We first rigorously set up the mathematical framework and provide definitions in section 2. In section 3, we formulate the asymptotic phase as a solution to an HJ boundary value problem. Then we describe the FSM for the computation of isochrons in section 4 and the parallelization in section 5. In section 6 we provide the various dynamical system models we will study, including the HH model for neural dynamics. We then present and discuss the numerical results of the example problems in section 7 before discussing the computational expense in section 8. Finally, we make concluding remarks in section 9.

**2. Definitions.** Consider an autonomous vector field

$$(1) \quad \frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n \quad (n \geq 2),$$

having a stable hyperbolic periodic orbit  $\mathbf{x}^\gamma(t)$  with period  $T$ . We define the set of all points on the stable orbit as  $\Gamma$  and the basin of attraction as  $\mathcal{B}$ . For each point  $\mathbf{x}^*$  in  $\mathcal{B}$  there exists a unique  $\theta(\mathbf{x}^*)$  such that

$$(2) \quad \lim_{t \rightarrow \infty} \left| \mathbf{x}(t) - \mathbf{x}^\gamma \left( t + \frac{T}{2\pi} \theta(\mathbf{x}^*) \right) \right| = 0,$$

where  $\mathbf{x}(t)$  is a trajectory starting with the initial point  $\mathbf{x}^*$ . The function  $\theta(\mathbf{x})$  is called the *asymptotic phase* of  $\mathbf{x}$  and takes values in  $[0, 2\pi)$ . Other conventions, related to this through a simple rescaling, define the asymptotic phase to take values in  $[0, T)$  or in  $[0, 1)$ . Let  $\gamma_0 \in \Gamma$  be

the point where the phase is zero. Our convention is to choose  $\gamma_0$  as the global maximum of  $\Gamma$  with respect to the first coordinate. An *isochron* is a level set of  $\theta(\mathbf{x})$ , that is, the collection of all points in the basin of attraction of  $\mathbf{x}^\gamma$  with the same asymptotic phase. Isochrons extend the notion of phase of a stable periodic orbit to the basin of attraction of the periodic orbit. It is conventional to define isochrons so that the phase of a trajectory on the periodic orbit advances linearly in time, so that

$$(3) \quad \frac{d\theta}{dt} = \frac{2\pi}{T}$$

both on and off the periodic orbit. Points at which isochrons of a periodic orbit cannot be defined form the *phaseless set*.

Isochrons can be shown to exist for any stable hyperbolic periodic orbit. They are codimension one manifolds as smooth as the vector field, and transversal to the periodic orbit  $\mathbf{x}^\gamma$ . Their union covers an open neighborhood of  $\mathbf{x}^\gamma$ . This can be proved directly by using the implicit function theorem [9, 14] and is also a consequence of results on normally hyperbolic invariant manifolds [46].

The “almost phaseless set” of a system is a region (often near the phaseless set) in which the isochrons are sufficiently close for small perturbations (in practice, ambient noise) to lead to phase randomization. A measure of the “closeness” of isochrons is the magnitude of the gradient of phase:  $|\nabla\theta|$ . The threshold on  $|\nabla\theta|$  which defines the almost phaseless set will depend on the system and the perturbations it is subjected to. For the purposes of this paper, we define the almost phaseless set  $\mathcal{A}$  as

$$(4) \quad \mathcal{A} = \{\mathbf{x} \in \mathcal{B} \mid |\nabla\theta| > \tau\},$$

where  $\tau$  is an application-dependent threshold value.

**3. Hamilton–Jacobi formulation.** We study systems of first-order ordinary differential equations (1), where the dynamics  $\mathbf{F}$  are given by the specific model and  $\mathbf{x}$  is the state. Since  $\mathbf{x}^\gamma(t)$  is a hyperbolic attractor, as  $t \rightarrow \infty$  the state converges to the periodic orbit. We can assume after a sufficiently long time, denoted by  $T_\infty$ , that the state will be arbitrarily close to the orbit for any initial condition not in the phaseless set and in a finite region of interest  $\Omega'$  within the basin of attraction:

$$\inf_{\mathbf{y} \in \Gamma} \|\mathbf{x}(T_\infty) - \mathbf{y}\| \leq \epsilon \quad \forall \mathbf{x}_0 \in \Omega' \subseteq \mathcal{B},$$

where  $\epsilon$  is a small parameter. To compute isochrons, it becomes advantageous to formulate the problem as a static HJ boundary value problem. We do this by applying the chain rule to (3),

$$\frac{d\theta}{dt} = \frac{\partial\theta}{\partial\mathbf{x}} \cdot \frac{d\mathbf{x}}{dt} = \frac{2\pi}{T},$$

and substituting (1):

$$\nabla\theta \cdot \mathbf{F}(\mathbf{x}) = \frac{2\pi}{T}.$$

The above is a first-order hyperbolic PDE. Since the characteristics emanate from any point near the periodic orbit to the rest of  $\mathcal{B}$ , the above equation will have a unique solution for all points in the basin of attraction if we impose boundary conditions on a small neighborhood near  $\Gamma$ . We define this set as

$$(5) \quad \Gamma^+ = \{\mathbf{x} \mid \inf_{\mathbf{y} \in \Gamma} \|\mathbf{x} - \mathbf{y}\| \leq r\},$$

where  $r$  is a small parameter. Finally, the time-independent (static) HJ formulation for the asymptotic phase is given:

$$(6) \quad \begin{aligned} \nabla \theta \cdot \mathbf{F}(\mathbf{x}) &= \frac{2\pi}{T} \quad \forall \mathbf{x} \in \mathcal{B}, \\ \theta(\gamma) &= f(\gamma) \quad \forall \gamma \in \Gamma^+, \end{aligned}$$

where  $f$  is the phase along the periodic orbit and can be found by evolving a single trajectory according to (1) until it is sufficiently close to  $\Gamma$  and applying the definition of asymptotic phase from section 2. Since isochrons are just contours of constant phase, by solving (6), we solve for all isochrons simultaneously. Any isochron can be visualized by simply taking an isocontour of the function  $\theta(\mathbf{x})$ .

**4. Fast sweeping method.** A numerical method for solving static HJ equations must have two parts. The first is a consistent discretization of the underlying PDE, and the second is a method for solving the system of equations. In this section, we describe the Godunov finite difference discretization and the iterative FSM for computing the numerical solution.

**4.1. Discretization.** In this section we describe the discretization in two dimensions and discuss the extension to higher dimensions. Consider a domain  $\Omega = [x_{min}, x_{max}] \times [y_{min}, y_{max}] \in \mathbb{R}^2$  discretized by a uniform Cartesian grid with  $N_x + 1$  and  $N_y + 1$  equally spaced grid points in the  $x$  and  $y$  directions, respectively. The discrete locations of the points on the grid are given by  $x_i = x_{min} + i\Delta x$  and  $y_j = y_{min} + j\Delta y$ , where  $\Delta x, \Delta y$  are the grid spacings. We will use the notation  $\theta_{i,j}$  to denote  $\theta(x_i, y_j)$ .

Equation (6) can be rewritten in two dimensions as

$$(7) \quad \frac{\partial \theta}{\partial x} F_1 + \frac{\partial \theta}{\partial y} F_2 = \frac{2\pi}{T},$$

where the components of the vector  $\mathbf{F}$  are  $\{F_1, F_2\}$ . The Godunov discretization of the Hamilton–Jacobi–Bellman (HJB) equation for optimal control was introduced in [10] and later used in [24]. Our equation is a specific case of the HJB equation, so we use the same discretization. In fact, (6) is a linear PDE with characteristic directions prescribed by  $\mathbf{F}$ , so the discretization becomes much simpler. The first-order forward and backward finite differences for the approximation of derivatives are

$$(8) \quad \begin{aligned} D_+^x \theta &= \frac{\theta_{i+1,j} - \theta_{i,j}}{\Delta x}, & D_-^x \theta &= \frac{\theta_{i,j} - \theta_{i-1,j}}{\Delta x}, \\ D_+^y \theta &= \frac{\theta_{i,j+1} - \theta_{i,j}}{\Delta y}, & D_-^y \theta &= \frac{\theta_{i,j} - \theta_{i,j-1}}{\Delta y}. \end{aligned}$$

The significance of the Godunov discretization is the notion of the “upwind” direction. The derivative approximation is chosen such that information propagates from the correct characteristic direction. In this case, information is flowing backwards along the characteristics of the time-dependent problem of (1). That is, we initialize at the limit cycle (where all states converge as  $t \rightarrow \infty$ ) and sweep information backwards along the characteristics to the rest of the domain. That is, if  $F_1$  is positive, we use forward difference  $D_+^x$ , and if  $F_1$  is negative, the backward difference approximation  $D_-^x$  is used. The Godunov numerical discretization of (7) is

$$(9) \quad D_+^x \theta F_1^+ + D_-^x \theta F_1^- + D_+^y \theta F_2^+ + D_-^y \theta F_2^- - \frac{2\pi}{T} = 0,$$

where we use the notation  $a^+ = \max(0, a)$  and  $a^- = \min(0, a)$ . Equation (9) is the discrete form to be solved simultaneously for all grid points. The following section details the method for solving this system of equations.

**Remark 1.** *A caveat must be made regarding finite difference approximations of the derivative of phase. Since a phase is a periodic quantity and is constrained to exist in  $[0, 2\pi)$ , one must account for that when computing derivatives. We do this at each point by checking that all values in the stencil are continuous. If they are not, then we temporarily add either  $2\pi$  or  $-2\pi$  to the value so that the function is continuous for all points in the numerical stencil and compute derivatives with the temporary values. In two dimensions, the continuity is enforced by computing the temporary value  $\tilde{\theta}$  as follows:*

$$(10) \quad \tilde{\theta}_{k,l} = \begin{cases} \theta_{k,l} + 2\pi & \text{if } |\theta_{k,l} + 2\pi - \theta_{i,j}| < |\theta_{k,l} - \theta_{i,j}|, \\ \theta_{k,l} - 2\pi & \text{if } |\theta_{k,l} - 2\pi - \theta_{i,j}| < |\theta_{k,l} - \theta_{i,j}|, \\ \theta_{k,l} & \text{otherwise} \end{cases}$$

$$\forall (k, l) \in \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}.$$

**Remark 2.** *Equation (6) is true for all  $\mathbf{x} \in \mathcal{B}$ , and in section 4.1, we define the computational domain  $\Omega$  as a box. For this method to produce a robust solution,  $\Omega \subseteq \mathcal{B}$  is required. There exist dynamical systems for which that requirement is not met. For instance, the Morris–Lecar model [34] of a barnacle giant muscle fiber is an often studied periodic system. When the parameters are such that the Morris–Lecar equations are near the homoclinic bifurcation [39], the basin of attraction shrinks so that there does not exist a bounding box of the periodic orbit that is a subset of the basin of attraction. However, for many practical systems, including the Morris–Lecar model away from the bifurcation, this requirement is met and our method can be applied.*

**4.2. Initialization.** The algorithm begins by initializing the points that lie in the boundary set  $\Gamma^+$  with a known solution and assigning an arbitrary value to all other points. This requires computing the phase on the periodic orbit  $f(\gamma)$ . We do this by selecting any initial point  $x_0 \in \mathcal{B}$  and numerically integrating (1) for sufficient time ( $T_\infty$ ) such that the trajectory is near the periodic orbit. This can be done with any standard numerical ODE integration software. For all examples herein, we use the MATLAB solver `ode15s`. Then  $\gamma_0$  is chosen

from all points in the trajectory according to its definition in section 2. Finally, with  $\gamma_0$  as the initial point, (1) is integrated once more, forward in time for exactly one period. This directly provides  $T$ , and the phase on the orbit is simply computed by its definition:  $\theta = \frac{2\pi t}{T}$ .

Since the method is first-order accurate, a first-order accurate approximation of the phase for points in  $\Gamma^+$  is sufficient. We choose to extend the phase from the orbit to the nearby points by constant extrapolation from the nearest point on the orbit:

$$(11) \quad \begin{aligned} \theta_{i,j} &= f(\gamma_{min}(\mathbf{x}_{i,j})) & \forall \mathbf{x}_{i,j} \in \Gamma^+, \\ \theta_{i,j} &= 2\pi & \forall \mathbf{x}_{i,j} \notin \Gamma^+, \end{aligned}$$

where

$$\gamma_{min}(\mathbf{x}_{i,j}) = \arg \min_{\gamma \in \Gamma} (\|\mathbf{x}_{i,j} - \gamma\|).$$

For  $\gamma_{min}$  to be unique,  $\Delta x$  must be sufficiently small to resolve sharp features in  $\Gamma$ . In practice we choose a neighborhood width on the order of  $\Delta x$ , i.e., in (5),  $r = \Delta x$ . For a formal proof of the accuracy of the approximation in (11), see Appendix A.

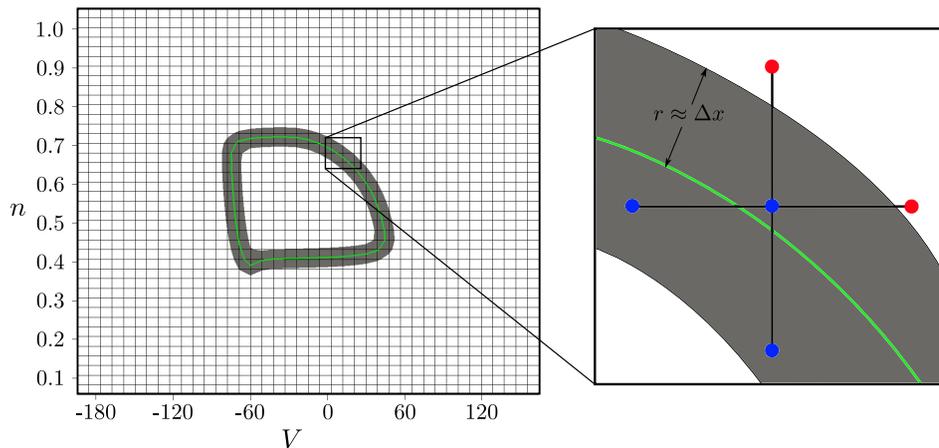
Figure 1 illustrates the initialization procedure on a 2D and a 3D example.<sup>1</sup> In the figure, the stable orbits are indicated by green curves and  $\Gamma^+$  is shaded gray. The insets show grid points within  $\Gamma^+$  in blue and those outside  $\Gamma^+$  in red. All points are initialized according to (11). The figure graphically shows that initializing in a tube with  $r \approx \Delta x$  creates a band of boundary points from which the computed solution can propagate.

**4.3. Iterative updating.** The FSM is an iterative method that takes advantage of the underlying causality of the HJ PDE. Information propagates along characteristic curves from the boundary data to the rest of the domain. The FSM updates grid points in a specific order that allows for information to propagate through the entire domain along a single characteristic direction with a single sweep. The ordered list of grid point vertices for each of the four sweeping orderings for the 2D case is given below:

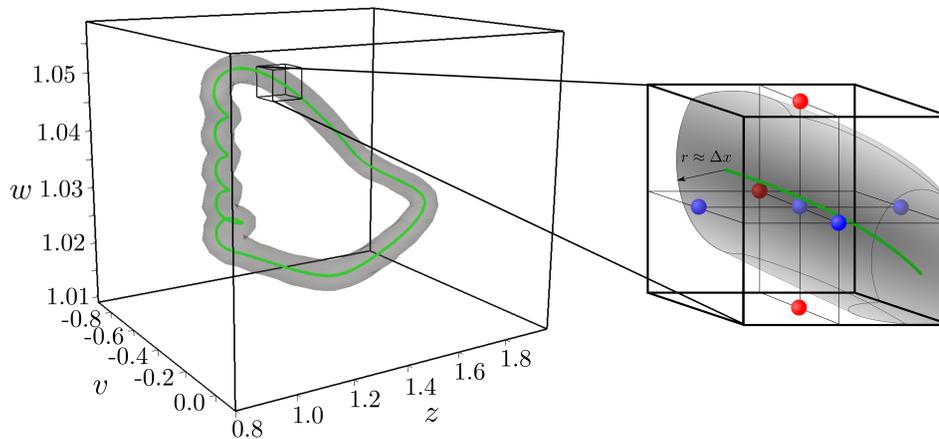
$$(12) \quad \begin{array}{llll} 1 & i = \{0, 1, \dots, N_x\}, & j = \{0, 1, \dots, N_y\}, & \nearrow \\ 2 & i = \{N_x, N_x - 1, \dots, 0\}, & j = \{0, 1, \dots, N_y\}, & \nwarrow \\ 3 & i = \{N_x, N_x - 1, \dots, 0\}, & j = \{N_y, N_y - 1, \dots, 0\}, & \swarrow \\ 4 & i = \{0, 1, \dots, N_x\}, & j = \{N_y, N_y - 1, \dots, 0\}. & \searrow \end{array}$$

The arrows indicate that a particular sweep accesses grid points in that general direction and propagates the solution along a family of characteristic curves. The arrows  $\nearrow, \nwarrow, \swarrow, \searrow$  denote a family of characteristic curves with vector components of sign  $(+, +), (-, +), (-, -), (+, -)$ , respectively. A visualization of this procedure is shown in Figure 2. The example problem has a circular periodic orbit shown in green, and its characteristic directions are indicated by the black vectors. The problem is fully specified in section 6.1. Notice that after each sweep, information has propagated to grid points along the corresponding

<sup>1</sup>The specifics of the two dynamical systems will be introduced in section 6.



(a) Initialization of a 2D example (reduced HH model from section 6.2).



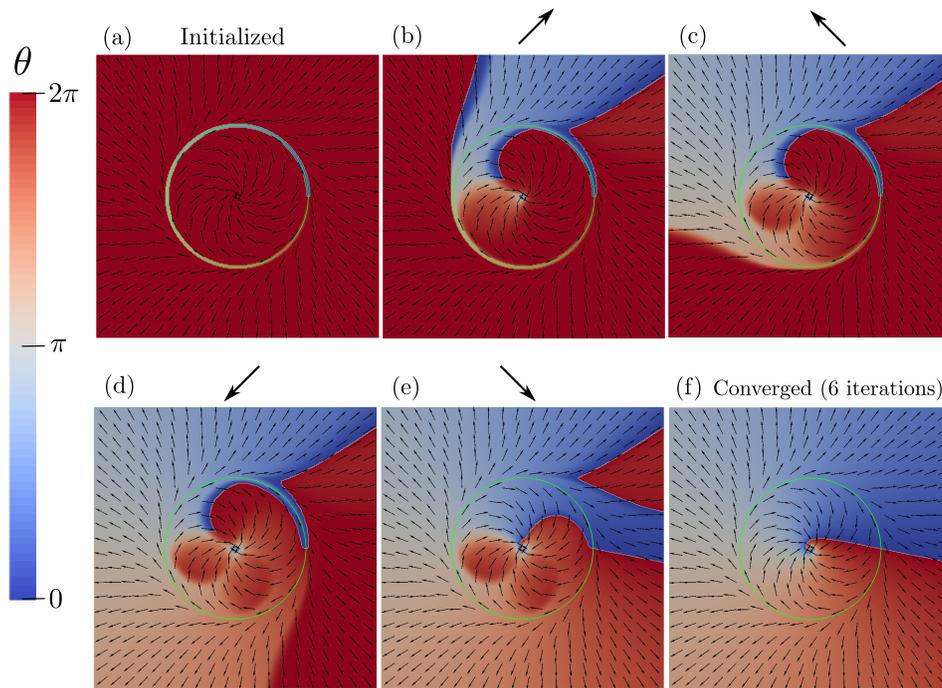
(b) Initialization of a 3D example (Wilson–Callaway model from section 6.3).

**Figure 1.** Illustration of the initialization procedure on a (a) 2D and (b) 3D example. The stable orbits are indicated by the green curves. The nearby region  $\Gamma^+$  is shaded in gray. Discrete grid points are initialized by (11) and shown in the inset as blue (inside  $\Gamma^+$ ) or red (outside  $\Gamma^+$ ).

characteristic directions. This particular example was computed on a grid with  $N_x = N_y = 320$  and converged in six iterations. Each of the sweeping directions is performed during each iteration. So there are 4, 8, and 16 sweeps in each iteration in two, three, and four dimensions, respectively.

The formula to update each grid point comes directly from the discretization (9). Substituting equations (8) and rearranging, we get the following 2D update formula:

$$\begin{aligned} \frac{\theta_{i+1,j} - \theta_{i,j}}{\Delta x} F_1^+ + \frac{\theta_{i,j} - \theta_{i-1,j}}{\Delta x} F_1^- + \frac{\theta_{i,j+1} - \theta_{i,j}}{\Delta y} F_2^+ + \frac{\theta_{i,j} - \theta_{i,j-1}}{\Delta y} F_2^- - \frac{2\pi}{T} &= 0, \\ -\theta_{i,j} \left( \frac{|F_1|}{\Delta x} + \frac{|F_2|}{\Delta y} \right) + \frac{\theta_{i+1,j} F_1^+ - \theta_{i-1,j} F_1^-}{\Delta x} + \frac{\theta_{i,j+1} F_2^+ - \theta_{i,j-1} F_2^-}{\Delta y} - \frac{2\pi}{T} &= 0, \end{aligned}$$



**Figure 2.** Visualization of the solution procedure of the FSM on an example problem (to be introduced in section 6.1) with its circular periodic orbit shown in green. The characteristic directions are indicated by the black vector field. Panel (a) shows the initialized domain. Panels (b), (c), (d), and (e) show the updated solution after the ↗, ↖, ↙, ↘ sweeps, respectively, during the first iteration. Panel (f) shows the solution after convergence (6 iterations). This particular grid is  $320^2$ .

$$(13) \quad \theta_{i,j} = \frac{\theta_{i+1,j}F_1^+ - \theta_{i-1,j}F_1^- + \theta_{i,j+1}F_2^+ - \theta_{i,j-1}F_2^- - \frac{2\pi}{T}}{\left(\frac{|F_1|}{\Delta x} + \frac{|F_2|}{\Delta y}\right)}$$

The higher dimensional update formulas can be derived similarly. For the sake of brevity, the only additional update formula we provide is for the 3D case:

$$(14) \quad \theta_{i,j,k} = \frac{\theta_{i+1,j,k}F_1^+ - \theta_{i-1,j,k}F_1^- + \theta_{i,j+1,k}F_2^+ - \theta_{i,j-1,k}F_2^- + \theta_{i,j,k+1}F_3^+ - \theta_{i,j,k-1}F_3^- - \frac{2\pi}{T}}{\left(\frac{|F_1|}{\Delta x} + \frac{|F_2|}{\Delta y} + \frac{|F_3|}{\Delta z}\right)}$$

The illustration in Figure 2 is typical for a fast sweeping problem. On the first sweep, information is only propagated from the boundary in the ↗ direction, but the update formula is applied to all grid points. This leads to a significant number of unnecessary computations. In [3], the authors propose the locking sweeping method (LSM) as a variant on the FSM. The algorithm only updates grid points with neighbors that have changed since the previous iteration. This method is simple to implement and provides a significant reduction in total computations. In practice we implement the LSM variant on the FSM.

We sweep over all the grid points iteratively applying the update formula, with alternating orderings until convergence is reached. Our criterion for convergence is that no value changes more than a small threshold between two successive iterations. In practice we choose a threshold that is of the order of machine epsilon.

**Remark 3.** *In certain cases, points on the boundary of the domain must be treated in a special manner. If the following inequality is satisfied, then all characteristics of (1) point into the computational domain and the update formulas can be applied unmodified:*

$$(\mathbf{F} \cdot \hat{\mathbf{n}} \leq 0) \quad \forall \mathbf{x} \in \partial\Omega.$$

Here,  $\hat{\mathbf{n}}$  is an outward-pointing vector normal to the computational boundary  $\partial\Omega$ . However, if the above inequality is not satisfied for a grid point, then characteristics point out of the domain and the update formula is undefined since it depends on a point outside the domain. In this case, we set the offending component of  $\mathbf{F}$  equal to 0 and update as usual. Finally, we must check that this modification does not propagate incorrect solutions to the region in which we are interested. Without loss of generality, say we are interested in isochrons in a region  $\Omega' \subset \Omega$ . The method will converge if all trajectories originating in  $\Omega'$  never leave  $\Omega$  for all  $t > 0$ . This can be done computationally by sampling points along  $\partial\Omega'$  and testing their trajectories against  $\Omega$ . In practice, we use an ad hoc method, setting  $\Omega' = \Omega$  and checking whether  $\mathbf{F} \cdot \hat{\mathbf{n}} \leq 0$ . If that inequality is not satisfied, then we increase the size of  $\Omega$  incrementally until the  $\theta$  values inside converge to within a given threshold. Note that  $\theta$  will not converge near the phaseless set, so the convergence check should only apply to values not near the phaseless set. We acknowledge that there can exist pathological cases where an infinitely large domain would be necessary to compute accurate isochrons, but this was not the case for any of the examples we tested.

**4.4. Algorithm overview.** Algorithm 1 shows a pseudocode listing of the entire algorithm for the 2D problem. The extension to higher dimensions has been covered in the text.

**5. Parallelization.** There exists an efficient parallelization of the FSM. The computational cost and memory footprint of the FSM are both  $O(N_x^d)$ , where  $N_x = N_y = \dots$ . The exponential growth with dimension is referred to as the curse of dimensionality. For problems in many dimensions, the computational cost quickly exceeds the capabilities of a single CPU. By using state-of-the-art hardware (e.g., large distributed memory clusters and/or graphics processing units (GPUs)) along with our novel algorithm, we can compute isochrons for many-dimensional dynamical systems that were intractable using existing methods.

In [13], the authors developed the parallel FSM for the eikonal equation. The parallel method is based on the dependence tree of points on a uniform grid linked by a 5-point stencil. Since our discretization for the computation of isochrons uses a 5-point stencil (in two dimensions), the parallel method can be directly applied to this application. We have implemented this parallel method and use it for all of the numerical examples in this article. A numerical study of the parallel performance is included in section 8. A hybrid method [12] is currently in development and will allow for parallelization of this and other problems at a much larger scale.

---

**Algorithm 1** Solve (6) in two dimensions with the FSM.

---

▷ Initialization

$T_\infty = 100$  ▷ Find a sufficiently large time a priori  
 $\mathbf{x}_0 = \{0, 0\}$  ▷ Choose an (arbitrary) initial condition inside  $\mathcal{B}$   
 $\mathbf{x}(t) = \text{integrate\_dynamics}(T_\infty, \mathbf{x}_0)$  ▷ Compute trajectory from  $\mathbf{x}_0$  until  $t = T_\infty$   
 $\gamma_0 = \text{compute\_point\_of\_zero\_phase}(\mathbf{x}(t))$  ▷ See definition of  $\gamma_0$  in section 2  
 $[f, T] = \text{integrate\_single\_orbit}(\gamma_0)$

**for all**  $\mathbf{x}_{i,j} \in \Omega$  **do**

**if**  $\mathbf{x}_{i,j} \in \Gamma^+$  **then**

$\gamma_{min} = \text{compute\_nearest\_point\_on\_orbit}(\mathbf{x}_{i,j})$  ▷ See (11)  
 $\theta_{i,j} = f(\gamma_{min})$

**else**

$\theta_{i,j} = 2\pi$

**end if**

**end for**

▷ Sweeping

**for** iteration = 0 : max\_iterations **do**

$\theta_{old} = \theta$

**for** ordering = 1 : 4 **do** ▷ 4 sweeping orderings in 2D

choose\_sweep\_direction(ordering) ▷ Sweep in correct order (12)

**for all**  $\mathbf{x}_{i,j} \in \Omega$  **do**

**if**  $\mathbf{x}_{i,j} \notin \Gamma^+$  **then**

$\theta(i, j) = \text{update\_function}(\mathbf{x}_{i,j}, \theta_{i-1,j}, \theta_{i+1,j}, \theta_{i,j-1}, \theta_{i,j+1})$  ▷ Equation (13)

**end if**

**end for**

**end for**

**if** convergence( $\theta, \theta_{old}$ ) **then**

break

**end if**

**end for**

---

## 6. Example systems.

**6.1. Simple circular orbit.** The first system we study is one which has an analytic solution, which we will use to validate the computational method. This particular example comes from [50]. Written in polar coordinates  $(r, \phi)$ , the dynamics are

$$\begin{aligned}\dot{r} &= 5r^2(1 - r), \\ \dot{\phi} &= r.\end{aligned}$$

This system has a circular periodic orbit with radius 1 and is centered at the origin. The orbit is a hyperbolic attractor with  $\mathcal{B} = \mathbb{R}^2 \setminus \{(0, 0)\}$ . The period is  $T = 2\pi$  and the asymptotic

phase is given exactly by

$$(15) \quad \theta(r, \phi) = \phi - \frac{1}{5r} + 0.2.$$

A coordinate transform to the  $x$ - $y$  plane gives the dynamics

$$(16) \quad \mathbf{F} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 5r^2(1-r)\cos\phi - r^2\sin\phi \\ 5r^2(1-r)\sin\phi + r^2\cos\phi \end{pmatrix},$$

where the standard polar coordinate definitions apply:  $r = r(x, y) = \sqrt{x^2 + y^2}$  and  $\phi = \phi(x, y) = \arctan(y/x)$ .

**6.2. Hodgkin–Huxley models.** In 1952, Alan Hodgkin and Andrew Huxley presented a mathematical model for the generation of action potentials for a squid giant axon based on the dynamical interplay between ionic conductances and electrical activity [17]. This model, which is widely used in active research on the dynamics of neurons, consists of the following ODEs:

$$(17) \quad \begin{cases} C \frac{dV}{dt} = I_b + I - \bar{g}_{Na} m^3 h (V - V_{Na}) - \bar{g}_K n^4 (V - V_K) - g_L (V - V_L), \\ \frac{dn}{dt} = \alpha_n(V)(1-n) - \beta_n(V)n, \\ \frac{dm}{dt} = \alpha_m(V)(1-m) - \beta_m(V)m, \\ \frac{dh}{dt} = \alpha_h(V)(1-h) - \beta_h(V)h, \end{cases}$$

where

$$\begin{aligned} \alpha_n(V) &= \frac{0.01(V+55)}{1 - \exp[-(V+55)/10]}, & \beta_n(V) &= 0.125 \exp[-(V+65)/80], \\ \alpha_m(V) &= \frac{0.1(V+40)}{1 - \exp[-(V+40)/10]}, & \beta_m(V) &= 4 \exp[-(V+65)/18], \\ \alpha_h(V) &= 0.07 \exp[-(V+65)/20], & \beta_h(V) &= \frac{1}{1 + \exp[-(V+35)/10]}. \end{aligned}$$

Here  $V$  is the transmembrane potential (the voltage inside the axon minus that outside the axon),  $I$  is the current injected into the neuron from a microelectrode, and  $n$ ,  $m$ , and  $h$  are dimensionless gating variables with values in  $[0, 1]$ , which correspond to the state of the membrane's ion channels.  $I_b$  is the baseline current, which represents the effects of other parts of the brain on the neuron and can be viewed as a bifurcation parameter that controls whether the neuron is in an excitable or an oscillatory regime. In these equations, voltages are measured in  $mV$ , current density in  $\mu A/cm^2$ , capacitance density in  $\mu F/cm^2$ , and time in  $ms$ .

$\bar{g}_{Na}, \bar{g}_k, g_L$  are the conductances of the sodium, potassium, and leakage channels, respectively.  $V_{Na}, V_K, V_L$  represent respective reversal potentials for these channels, and  $C$  the constant membrane capacitance.

In a previous study [25], the 4D Hodgkin–Huxley (HH) equations were reduced to a set of two coupled (but still highly nonlinear) ODEs. This reduction leads to equations whose dynamics approximate the dynamics of the full HH equations, and which allow an intuitive understanding of the mathematics of action potential generation without the need to solve the full 4D model. The equations for the gating variables  $n$ ,  $m$ , and  $h$  can be written as

$$(18) \quad \tau_x(V) \frac{dx}{dt} = x_\infty(V) - x,$$

where

$$(19) \quad x_\infty(V) = \frac{\alpha_x(V)}{\alpha_x(V) + \beta_x(V)} \quad \text{and} \quad \tau_x(V) = \frac{1}{\alpha_x(V) + \beta_x(V)},$$

with  $x \in \{n, m, h\}$ . It is found that the time constant  $\tau_m$  is much smaller than  $\tau_n$  and  $\tau_h$  over the entire relevant range of  $V$ . Thus,  $m$  evolves faster than  $n$  or  $h$ . This suggests that  $m(t)$  in the HH equations can be reduced to a function of the voltage and given by its time-asymptotic value:  $m_\infty(V(t))$ . This approximation has been shown to be reasonable through numerical simulations. Next, it is observed numerically that when the solutions to the HH equations correspond to periodic action potentials, the following equation approximately holds:

$$(20) \quad n(t) + h(t) \approx 0.8.$$

Note that (20) should be viewed as an observation; it has no rigorous mathematical or biological basis. With these simplifications, the following 2D system of equations is obtained:

$$(21) \quad \begin{aligned} C \frac{dV}{dt} &= I_b + I - \bar{g}_{Na}[m_\infty(V)]^3(0.8 - n)(V - V_{Na}) \\ &\quad - \bar{g}_K n^4(V - V_K) - g_L(V - V_L), \\ \frac{dn}{dt} &= \alpha_n(V)(1 - n) - \beta_n(V)n. \end{aligned}$$

We use the following parameter values for all numerical experiments for both HH models:  $I_b = 10$ ,  $I = 0$ ,  $\bar{g}_{Na} = 120$ ,  $\bar{g}_k = 36$ ,  $g_L = 0.3$ ,  $V_{Na} = 50$ ,  $V_K = -77$ ,  $V_L = -54.4$ ,  $C = 1$ .

**6.3. Wilson–Callaway model.** We also consider a version of the Wilson–Callaway model of the dopaminergic neuron [47]. The authors proposed a model that describes the firing dynamics of dopaminergic neurons in the mammalian brain stem with a chain of strongly coupled oscillators. In [31], the authors studied the Wilson–Callaway model for a range of parameters and found that oscillators in a regime near the Andronov–Hopf bifurcation exhibit mixed mode oscillatory behavior. The authors of [27] study the Wilson–Callaway model for the case of two compartments (a 4D system). They note the strong diffusive coupling and

reduce the system to a 3D model to study its dynamics in the context of canard-induced MMOs. We study this 3D system:

$$(22) \quad \begin{cases} \epsilon \frac{dv}{dt} = a(E_2 - v)[f(v) - z], \\ \frac{dz}{dt} = \bar{\omega}g_1(v)(E_1 - v) - \frac{\omega_1^2 + \omega_2^2}{\tau(\omega_1 + \omega_2)}z - \frac{\omega_1 - \omega_2}{2\tau}w, \\ \frac{dw}{dt} = -\frac{\omega_1\omega_2}{\tau\bar{\omega}} \left( \frac{\omega_1 - \omega_2}{\omega_1 + \omega_2}z + w \right), \end{cases}$$

where

$$f(v) = \frac{g_1(v)(E_1 - v) + \bar{g}_3(E_3 - v)}{a(E_2 - v)} + \frac{b}{a},$$

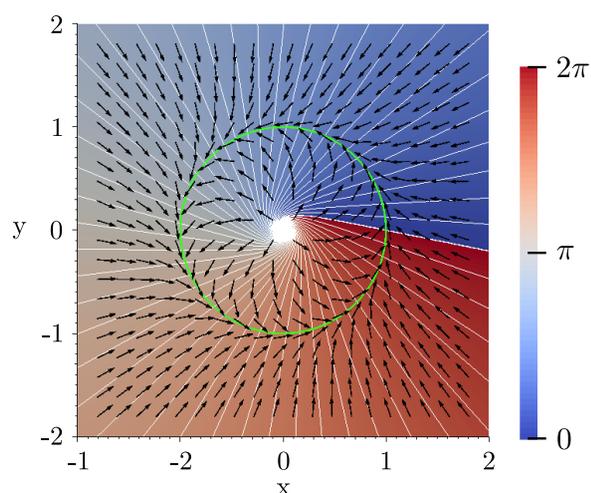
and

$$g_1(v) = \frac{\bar{g}_1}{1 + \exp -(v - c_1)/c_2}.$$

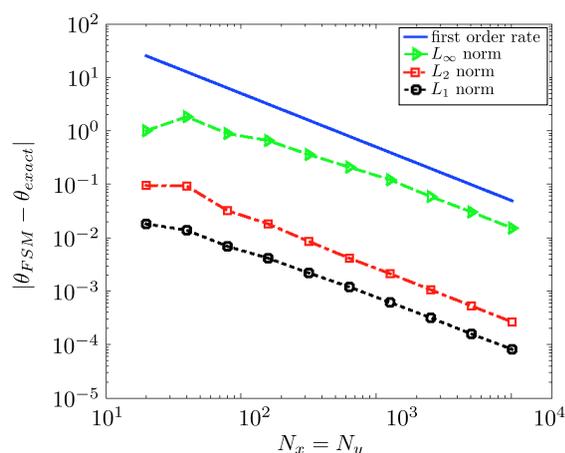
For a rigorous description of the system and its reduction to three dimensions, see [27]. For all numerical experiments herein, we use the following parameters:  $a = -0.9569$ ,  $b = -0.7241$ ,  $c_1 = -0.35$ ,  $c_2 = 0.07$ ,  $d = 4444$ ,  $E_1 = 1.0$ ,  $E_2 = -0.9$ ,  $E_3 = -0.5$ ,  $\bar{g}_1 = 0.8$ ,  $\bar{g}_3 = 1.0$ ,  $\epsilon = 0.013$ ,  $\omega_1 = 1.0$ ,  $\omega_2 = 16.0$ ,  $\bar{\omega} = (\omega_1 + \omega_2)/2$ .

**7. Numerical results.** In this section we present numerical results for the four example systems described in section 6. The first example is one with an analytic solution for the purpose of validating the computational method. It is described in section 6.1. The second and third examples are the 2D and 4D HH models discussed in section 6.2. The fourth example is the 3D system with MMOs from section 6.3. In each case, we present the numerical solution and discuss the results. The timing data and other relevant details of each of these tests is given in Table 2 at the end of section 8.

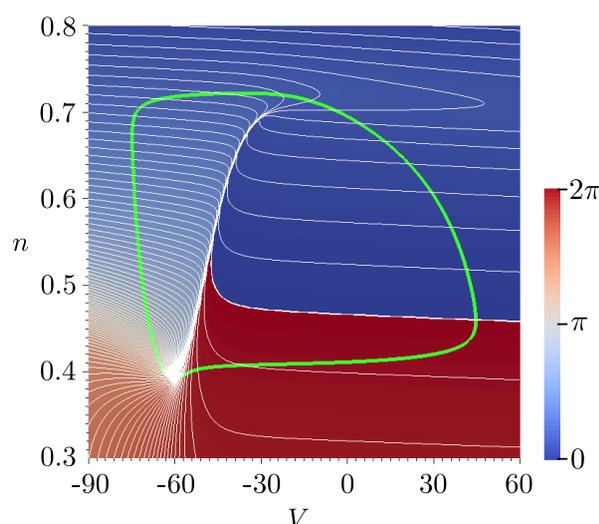
**7.1. 2D analytic example.** To confirm that the numerical scheme is indeed consistent and first-order accurate, we consider a problem with a known exact solution. We described this system in section 6.1. Figure 3 shows the exact solution to the example problem. Notice that in this example, the phaseless set is the origin. We applied our computational technique to this problem and compared the results with the exact solution given by (15). A solution does not exist at the origin, so we compute errors at all points outside of a small neighborhood near the phaseless set ( $\sqrt{x^2 + y^2} > 0.05$ ). Figure 4 shows that as the grid is refined, the computed solution converges to the exact solution. In the asymptotic regime, the error in the maximum norm decreases at a first-order rate. This confirms that the method is consistent and first-order accurate (i.e.,  $\text{error} \propto \Delta x^1$ ).



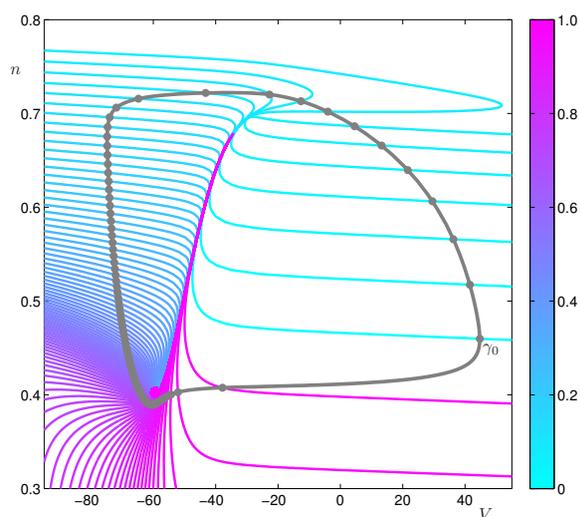
**Figure 3.** The exact solution to the first example problem as indicated by the color. The periodic orbit is indicated by the green line, and the characteristic directions are shown as black vectors. Also shown are 50 evenly spaced isochrons (white).



**Figure 4.** Error in computational solution as a function of grid size. The blue line is included for reference and indicates a first-order convergence rate.



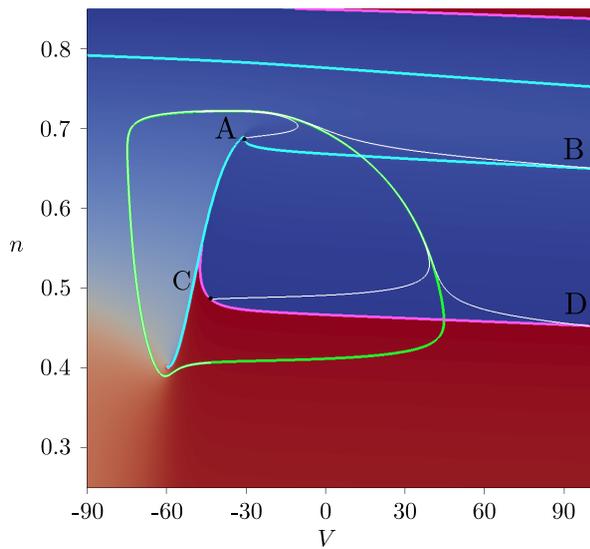
**Figure 5.** 100 evenly spaced isochrons of the 2D HH model computed with the FSM on a  $1600^2$  grid.



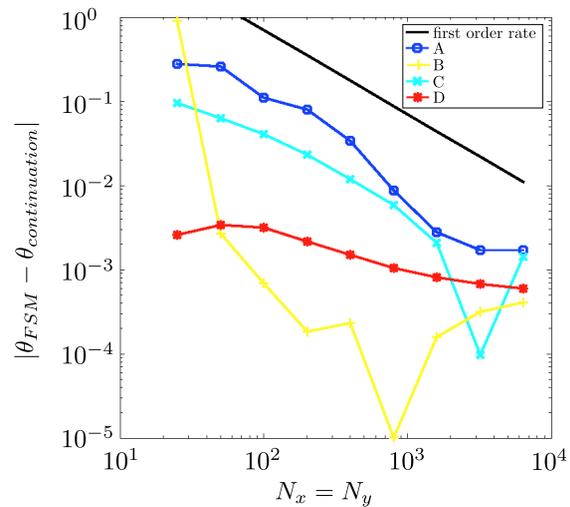
**Figure 6.** Isochrons of the 2D HH model as computed with the continuation-based method of [37].

**7.2. 2D Hodgkin–Huxley model.** We applied the method to the reduced HH model of (21). Figure 5 shows the asymptotic phase for the computational domain along with 100 evenly spaced isochrons and the periodic orbit. The results from the method of [37] are shown in Figure 6 for comparison. The two results are in good agreement.

For further validation of our method’s numerical accuracy, we did an error convergence



**Figure 7.** Four test points at which our method is compared to the continuation-based method of [37]. A and B exist on the  $\theta = 0.1\pi$  isochron (cyan), and C and D exist on the  $\theta = 0$  isochron (magenta). The backward streamlines from each point are also shown.



**Figure 8.** The magnitude of difference in phase between our method and the continuation-based method of [37] as the grid is refined. The black line indicates a first-order convergence rate.

analysis on the reduced HH model. We obtained the isochron data as computed by the continuation-based method of [37]. Because the continuation method gives the isochrons explicitly and our method does so implicitly, we cannot compare the two methods directly. However, for any point in  $\Omega$ , we can interpolate our grid-valued results to give a value for phase. In this manner we can compute a difference in phase between our method and the continuation-based method for a point on an isochron. We chose four points at which to measure the difference between the two methods:  $A = (-30.69, 0.692)$ ,  $B = (100, 0.650)$ ,  $C = (-43.53, 0.486)$ ,  $D = (100, 0.452)$ . A and B are along the  $\theta = 0$  isochron, while C and D are on the  $\theta = 0.1\pi$  isochron. At each point we computed the phase of our method using bilinear interpolation from the surrounding grid points. Figure 7 shows the four test points, their respective streamlines, the two relevant isochrons, and the periodic orbit. Figure 8 shows the absolute value of difference in phase of the two methods as the computational grid is refined. The black line is for reference and indicates a first-order convergence rate. The figure indicates that our method converges to the same solution as that of the continuation-based method at roughly a first-order rate. The convergence stops when the absolute error reaches a value of  $\approx 10^{-3}$ . This is, presumably, the error in the benchmark method.

We acknowledge that the Lagrangian approach of [37] allows for very detailed computation of isochrons as they spiral toward the phaseless set. Computing these fine details with our approach on a uniform grid would be computationally expensive, but it may be possible using adaptively refined grids. Adaptive grid techniques for computing isochrons with our Eulerian approach is a topic we are pursuing, but is outside the scope of this paper.

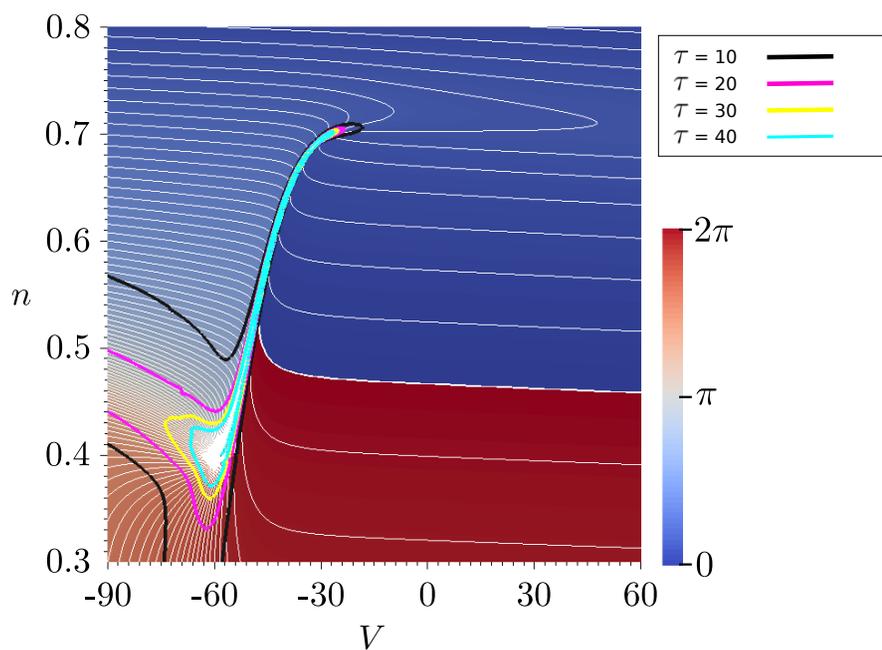
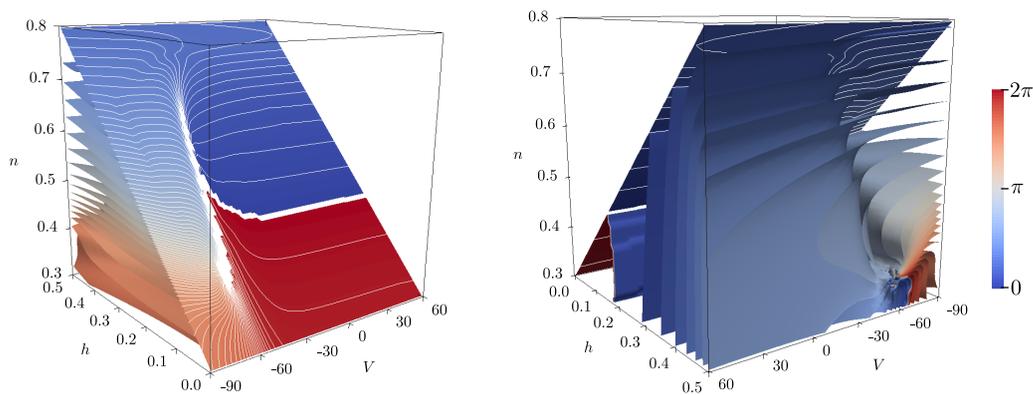


Figure 9. The outlines of the almost phaseless set for a range of threshold values  $\tau$ .

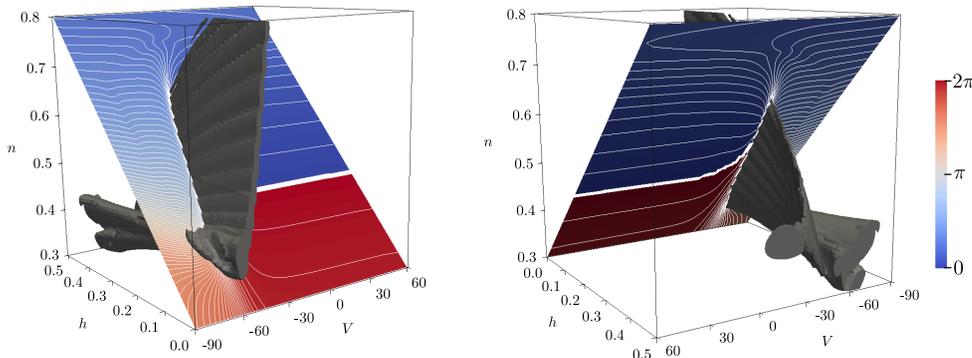
One application that depends on knowledge of the isochrons of the system is desynchronization of neurons. In [11, 35], the authors present optimal methods for driving the system to a region near the phaseless set. Once there, noise leads to phase randomization. This is sufficient because points near the phaseless set lie inside the almost phaseless set. Figure 9 shows the almost phaseless set for a range of threshold values. One of the advantages of our Eulerian method for computing phase is that once the phase is known on the Cartesian grid, it is simple to compute the gradient (and its magnitude) with finite differences. This method of computing the almost phaseless set may produce a larger target set and, as a result, a means for calculating a more useful optimal control for desynchronizing the system.

**7.3. 4D Hodgkin–Huxley model.** We also applied the method to the 4D HH model of (17). The tests were done at the highest resolution possible with the workstation we used ( $160^4$  grid points). Figure 10 shows two views of the isochrons projected onto a reduced subspace. The isochrons are projected onto  $m = m_\infty(V)$ ; i.e., we show the isochrons for  $\{(V, n, m, n) | m = m_\infty(V)\}$ . For the 2D slice, the isochrons are projected onto  $m = m_\infty(V)$  and  $h = 0.8 - n$ , i.e.,  $\{(V, n, m, h) | m = m_\infty(V), h = 0.8 - n\}$ . We note that the isochron geometry on the 2D subspace is similar to that of the isochrons of the reduced 2D system of Figure 5. The rotated view in Figure 10 (right) shows interesting isochron structure that is lost in the 2D reduced model.

With knowledge of the asymptotic phase of the 4D HH model, we can also compute the almost phaseless set according to (4). Figure 11 shows the almost phaseless set projected onto the 3D subspace. For reference, the 2D reduced subspace and the isochrons are included in the



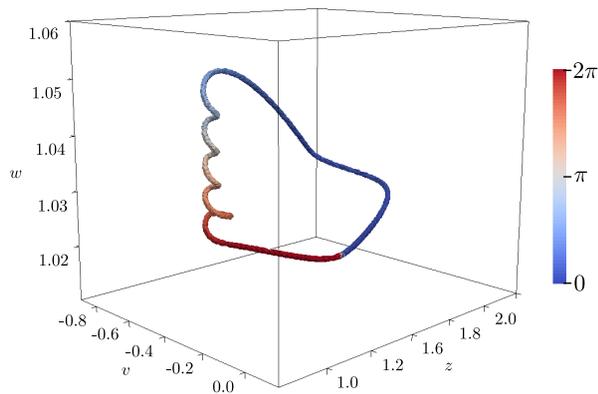
**Figure 10.** Two views of the isochrons of the 4D HH model projected onto the  $(V, n, m_\infty(V), h)$  subspace. The diagonal 2D slice is the projection onto the  $(V, n, m_\infty(V), h(n))$  subspace. 100 equally spaced isochrons are shown on the 2D slice, and every fourth isochron (25 total) is shown in three dimensions. The accompanying animation (99861.01.gif [local/web 1.68MB]) shows the isochrons in three dimensions with a rotating view.



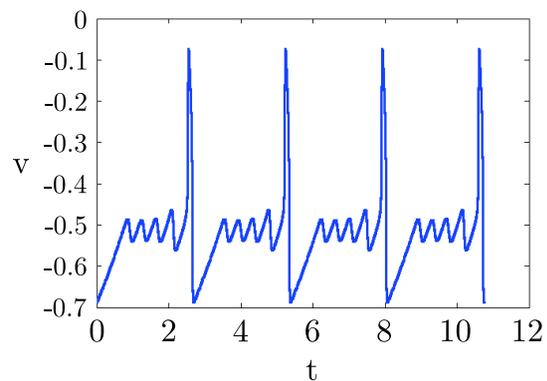
**Figure 11.** Two views of the almost phaseless set (gray) of the 4D HH model projected onto the  $(V, n, m_\infty(V), h)$  subspace for  $\tau = 30$ . The diagonal 2D slice is the projection onto the  $(V, n, m_\infty(V), h(n))$  subspace. 100 equally spaced isochrons are shown on the 2D slice. The accompanying animation (99861.02.gif [local/web 2.94MB]) shows the almost phaseless set for a varying threshold value.

figure. It can be seen that the intersection of the almost phaseless set in Figure 11 with the 2D subspace qualitatively agrees with the almost phaseless set of the 2D model shown in Figure 9. However, the almost phaseless set does vary in the  $h$  direction. Even though it is difficult to visualize the geometry of the isochrons and almost phaseless set in dimensions greater than three, it is simple to compute them and describe them numerically. For the application of desynchronizing neurons (e.g., for possible DBS treatment of Parkinson's disease [35]) the knowledge of this higher dimensional target set may allow for a more useful optimal control stimulus to be computed.

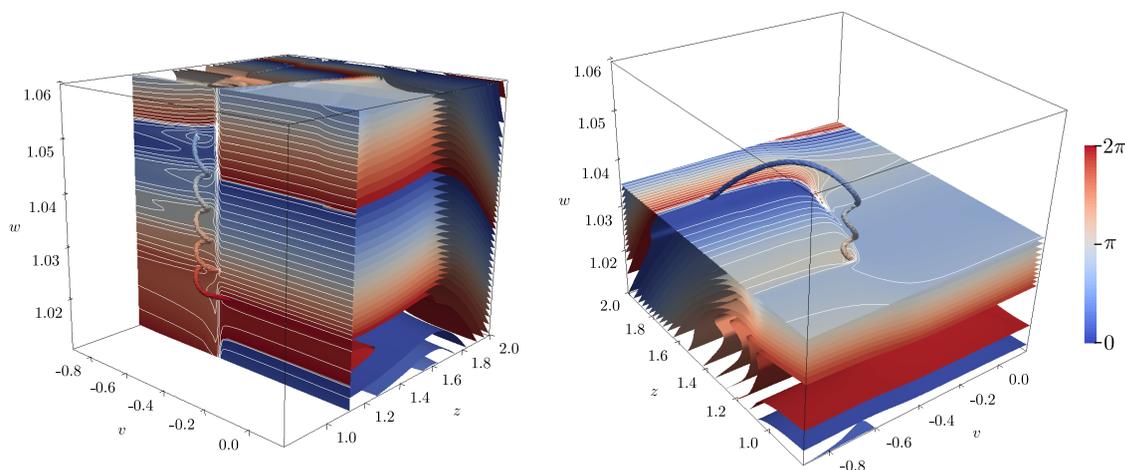
**7.4. Wilson–Callaway model.** The model of a dopaminergic neuron from section 6.3 has more complex firing dynamics than the HH model; it exhibits MMOs. This is a good example



**Figure 12.** Stable periodic orbit of the Wilson–Callaway model. Color indicates asymptotic phase.



**Figure 13.** Voltage-time series for the limit cycle of the 3D Wilson–Callaway model.

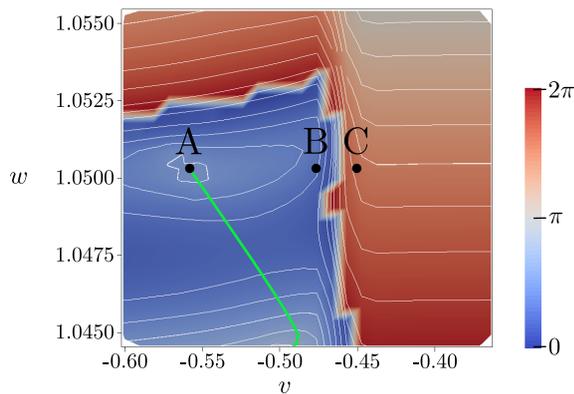


**Figure 14.** Two cutaway views of the isochrons of the 3D Wilson–Callaway model. The periodic orbit is shown as a tube and colored by phase. 20 equally spaced isochrons are shown on each 2D slice, and 20 isochrons are shown in the 3D half-space.

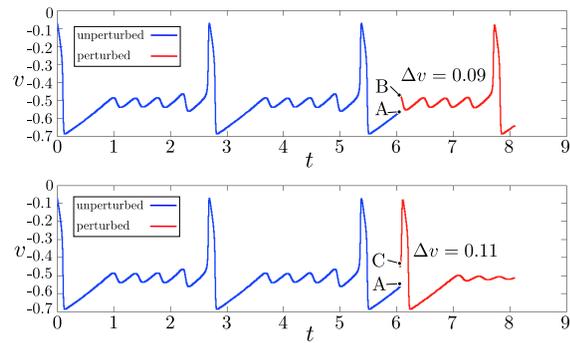
to show the versatility of the method. Figure 12 shows the limit cycle and the phase along it, and Figure 13 shows the voltage-time series. This system is characterized by a large, fast voltage spike followed by a series of slow spikes with smaller magnitude.

Figure 14 shows two cutaway views of the isochrons of the system superimposed with the periodic orbit. The cuts in Figure 14 (left) and 14 (right) are located at  $z = 1.1163$  and  $w = 1.035$ , respectively. The isochrons on the 2D plane cuts are shown in white, and the isochrons in the rest of the cutaway domain are colored by phase.

This picture of the isochrons of the Wilson–Callaway model allows us to better understand



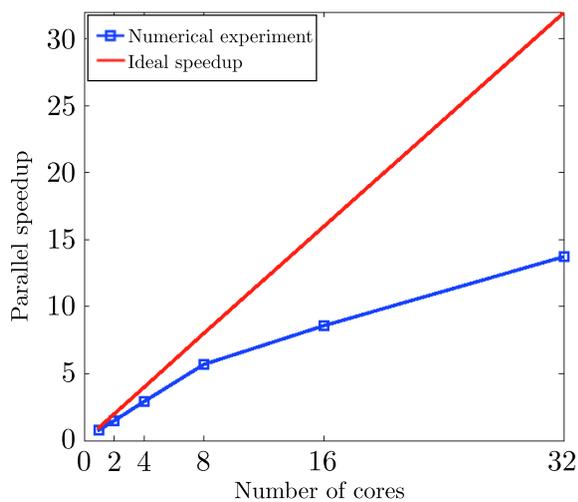
**Figure 15.** Closeup view of isochrons near the limit cycle on a slice ( $z = 1.1163$ ). The limit cycle intersects the plane at A and is represented by the green curve. Point A is on the limit cycle with phase  $\theta = \frac{\pi}{2}$ , point B represents a small perturbation ( $\Delta v = 0.09$ ) from the limit cycle, and point C represents a slightly larger perturbation ( $\Delta v = 0.11$ ).



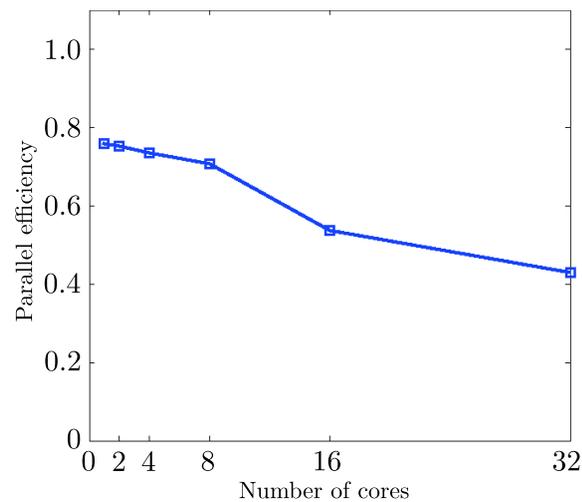
**Figure 16.** Evolution of the Wilson–Callaway model subject to a perturbation in the  $+v$  direction with two different magnitudes.

the behavior of the system. For example, we can look closely at the isochrons near the limit cycle and predict the effect that a perturbation will have on the system. Figure 15 shows the isochrons on a subset of our computational domain on a slice  $z = 1.1163$ . The periodic orbit is shown going through the slice at point A. Point A lies on the intersection of the limit cycle with the  $z = 1.1163$  plane and has phase  $\theta = \frac{\pi}{2}$  and coordinates  $(v, z, w) = (-0.563, 1.1163, 1.0505)$ . Point B represents a perturbation of  $\Delta v = 0.09$ ; i.e.,  $v(B) - v(A) = 0.09$ . Similarly, point C is the result of perturbing the system by  $\Delta v = 0.11$  from point A. By looking at the phase and isochrons surrounding the three points, we can predict the effect these perturbations will have on the evolution of the system. The perturbation from A to B does not result in a large change in the asymptotic phase, so we predict that it will have a small effect on the evolution of the system. However, the perturbation from A to C results in a significant jump in phase and crosses a concentrated region of isochrons. This small perturbation will immediately move the system to a phase near the spike at  $\theta = 2\pi$ . We carried out these perturbation experiments numerically. Their results are shown in Figure 16. We first evolve the system along the limit cycle for two full periods, and then, when  $\theta = \frac{\pi}{2}$ , we apply the perturbation. We then evolve the system for one more period. The small perturbation ( $\Delta v = 0.09$ ) does indeed have a small effect on the evolution of the system. Also, as predicted by the isochron analysis, the slightly larger perturbation has a dramatic effect on the system, immediately initiating the spike.

**8. Computational expense.** Here, we provide details about the efficiency of the numerical method. All tests were run on a single compute node of the Stampede compute cluster at the Texas Advanced Computing Center. Each standard node has two Intel Xeon E5-2680 processors with 32GB of host memory, and each processor has eight cores. Large memory nodes have 32 cores and 1TB of host memory. The code was written in C++ and parallelized with OpenMP.



**Figure 17.** Parallel speedup for the 4D HH model on a uniform grid with  $140^4$  points.



**Figure 18.** Parallel efficiency for the 4D HH model on a uniform grid with  $140^4$  points.

One can characterize the viability of a parallel method by analyzing the parallel speedup and comparing it to the ideal case. Parallel speedup is given by

$$(23) \quad S = \frac{T_1}{T_p},$$

where  $T_1$  is the execution time of the serial method and  $T_p$  is the execution time of the parallel method on  $p$  cores. For the ideal case,  $S = p$ . Parallel efficiency is given by

$$(24) \quad E_p = \frac{S_p}{p} = \frac{T_1}{pT_p},$$

where  $S_p$  is the speedup on  $p$  cores.

Since the 4D HH model is the most computationally expensive test in this paper, we analyze the parallel speedup and efficiency for that example. We conducted parallel speedup tests on a large memory node of stampede with a computational grid of  $140^4$  grid points. Figure 17 shows the parallel speedup on up to 32 cores with a maximum speedup of about 14. The parallel speedup is very good on up to 8 cores and continues to increase on 16 and 32 cores. Figure 18 shows that parallel efficiency is nearly 80% up to 8 cores before dropping off to a still useful 43% on 32 cores. The reduction in efficiency on 16 and 32 cores is related to the hardware architecture. Communication costs between processors are higher than within each 8-core processor.

The following is an analysis of the total computational resources used in a numerical experiment. These results are for the 4D HH model over a range of problem sizes on 16 cores. Table 1 shows the execution time and memory used to compute the asymptotic phase for the 4D HH model. For this problem, the memory required becomes prohibitive before the computation time becomes exceedingly large. In order to solve larger problems, one would

**Table 1***Computational resources used to compute the asymptotic phase on the 4D HH model.*

Number of grid points	Execution time (s)	Memory (GB)
$20^4$	1.46	0.0038
$40^4$	17.25	0.061
$60^4$	77.86	0.311
$80^4$	247.7	0.983
$100^4$	541.6	2.40
$120^4$	1094	4.97

**Table 2***Implementation details and timing results for the numerical experiments conducted throughout this work.*

Equation	Dimensions	$\Omega$	$\Omega'$	Grid points	Iterations	Threads	Time (s)	Figure(s)
(16)	$d = 2$	$[-2, 2]^2$	$\Omega' = \Omega$	$N = 320^2$	7	16	0.20	2
(21)	$d = 2$	$[-195, 165]$ $\times [0.06, 1.05]$	$[-150, 90]$ $\times [0.15, 0.95]$	$N = 1600^2$	71	16	69.7	5, 7, 9
(17)	$d = 4$	$[-0.0020, 0.0015]$ $\times [0.025, 1.117]$ $\times [-0.106, 0.218]$ $\times [-0.321, 0.850]$	$[-90, 60]$ $\times [0.30, 0.80]$ $\times [-0.03, 0.15]$ $\times [0.00, 0.50]$	$N = 160^4$	80	32	27,100	10, 11
(22)	$d = 3$	$[-1.92, 1.16]$ $\times [-0.39, 3.26]$ $\times [0.96, 1.11]$	$[-0.9, 0.15]$ $\times [0.8, 2.0]$ $\times [1.01, 1.06]$	$N = 320^3$	100	16	4865	12, 14, 15

have to make use of a compute node with a large amount of memory or employ a domain decomposition strategy. A hybrid method employing domain decomposition and shared memory parallelization is outside the scope of this paper, but is something we are actively researching [12].

Finally, Table 2 provides details about the numerical experiments discussed in the previous section.

**9. Conclusion.** We have introduced a novel numerical method to compute global isochrons of dynamical systems. This method is significant in that it computes all isochrons as isocontours of the solution of a Hamilton–Jacobi PDE, and it does so with an efficient method that can utilize state-of-the-art parallel computer hardware. Our method has an advantage over existing Lagrangian methods (e.g., [37, 19]) in that it computes all isochrons implicitly and the mathematics and numerics are simple to extend to high dimensions. The continuation method [37] does, however, make it possible to compute very fine features of isochrons

(e.g., near the phaseless set) and is embarrassingly parallel in that numerous isochrons can be computed simultaneously. A future topic of interest is to explore this Eulerian technique on adaptive tree-based grids to better compute the fine details that are currently more practical with Lagrangian methods. Another possible area of future research is to explore using higher order discretizations (e.g., ENO and WENO schemes [36, 23, 22]) to obtain a more accurate solution. This was studied for some static HJ equations in [51]. These techniques present their own challenges, including the need for a better approximation of  $\theta$  near  $\Gamma$ , a larger stencil, and an increased number of iterations to convergence which could all have implications for an efficient parallel implementation.

Since our methodology is easily extendable to high dimensions, it enables us to analyze both the 4D Hodgkin–Huxley model and the 2D reduced model. We used our results to compute the almost phaseless sets of both models. This information could impact control strategies for the desynchronization of a population of neurons for the treatment of Parkinson's disease. We also computed the isochrons of a 3D model of a dopaminergic neuron with MMOs to show the versatility of this model. It is our hope that this efficient and parallel method will allow for the analysis of more complex and higher dimensional models.

**Appendix A. Proof of initialization procedure accuracy.** In section 4.2, we presented our technique for initializing the FSM with constant extrapolation of the asymptotic phase from the periodic orbit to nearby grid points. Here, we prove that this method is first-order accurate, i.e., the error is  $O(\Delta x)$ .

Consider the static HJ formulation (equation (6)) for asymptotic phase of a dynamical system with a periodic orbit. Without loss of generality, assume isotropic grid spacing  $\Delta x = \Delta y = h$  and that the system exists in  $\mathbb{R}^2$ . Figure 19 depicts a point  $\mathbf{x}_{i,j}$  near the periodic orbit  $\Gamma$  and the corresponding nearest point on the orbit  $\gamma_{min}$ . We choose to represent these points in the orthonormal reference frame  $(n, \tau)$  with basis vectors normal to  $\Gamma$  and tangent to  $\Gamma$  and their origin at  $\gamma_{min}$ . The dynamics  $F$  necessarily lie tangent to  $\Gamma$  because it is a hyperbolic attractor. To verify the accuracy of the approximation used to initialize the method (equation (11)), we begin by constructing a Taylor expansion of  $\theta(n, \tau)$  at the origin and evaluating at our test point  $x_{i,j}$ :

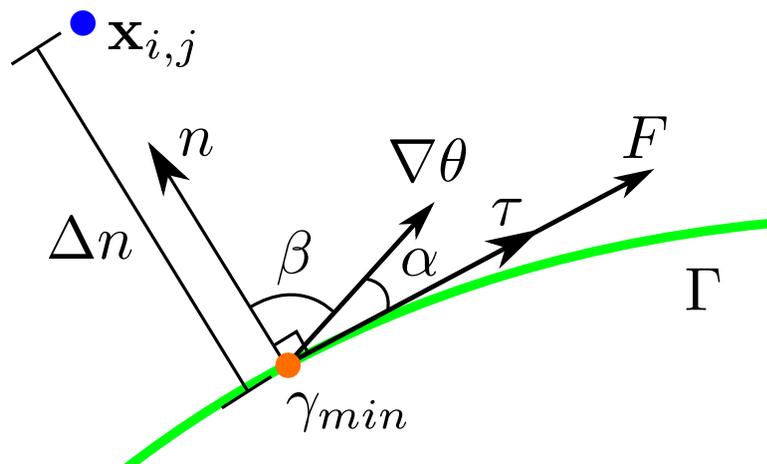
$$(25) \quad \theta(n, \tau) = \theta(0, 0) + n \left. \frac{\partial \theta}{\partial n} \right|_{(n, \tau)} + \tau \left. \frac{\partial \theta}{\partial \tau} \right|_{(n, \tau)} + \text{H.O.T.}$$

We assume that the curve  $\Gamma$  is sampled by points  $\gamma$  such that the distance between successive points is much smaller than  $h$ . Then we can state that the test point in the  $(n, \tau)$  coordinate system is equal to  $\mathbf{x}_{i,j} = (\Delta n, 0)$ :

$$(26) \quad \theta(\Delta n, 0) = \theta(0, 0) + \Delta n \frac{\partial \theta}{\partial n} + \text{H.O.T.}$$

From (6), the definition of the inner product, and the relationship  $\cos \alpha = \sin \beta$ , we have

$$\frac{2\pi}{T} = \nabla \theta \cdot F,$$



**Figure 19.** Schematic of a test point  $\mathbf{x}_{i,j}$  near the periodic orbit represented in a coordinate system normal and tangent to  $\Gamma$ .

$$(27) \quad \begin{aligned} \frac{2\pi}{T} &= \|\nabla\theta\| \|F\| \cos \alpha, \\ \|\nabla\theta\| &= \frac{2\pi}{T \|F\| \sin \beta}. \end{aligned}$$

The directional derivative of  $\theta$  in the  $n$  direction is

$$(28) \quad \begin{aligned} \frac{\partial\theta}{\partial n} &= \nabla\theta \cdot \hat{n}, \\ \frac{\partial\theta}{\partial n} &= \|\nabla\theta\| \cos \beta, \end{aligned}$$

where  $\hat{n}$  is the normalized basis vector in the  $n$  direction. The error  $e$  in our approximation is given by

$$(29) \quad e = |\theta(\Delta n, 0) - \theta_{approx}|.$$

Using our constant extrapolation approximation  $\theta_{approx} = \theta(0, 0)$ , substituting equations (26), (27), and (28) into (29), and dropping the higher order terms gives the following expression for the error:

$$(30) \quad e = \left| \frac{2\pi \Delta n \cot \beta}{T \|F\|} \right|.$$

Recall that we choose the width of the narrow band  $r = h$ , so necessarily  $\Delta n \leq h$ .  $F$  is nonzero on  $\Gamma$  and independent of grid spacing. Equation (6) guarantees that the vectors  $\nabla\theta$  and  $F$  are nonorthogonal; therefore,  $\cot \beta$  is bounded and is also independent of grid spacing. Thus, the initialization procedure of (11) is first-order accurate in the asymptotic regime  $h \rightarrow 0$ :

$$(31) \quad e = h \frac{2\pi |\cot \beta|}{T \|F\|} = O(h).$$

**Acknowledgments.** The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper (<http://www.tacc.utexas.edu>).

## REFERENCES

- [1] D. ADALSTEINSSON AND J. SETHIAN, *A fast level set method for propagating interfaces*, J. Comput. Phys., 118 (1995), pp. 269–277.
- [2] P. ASHWIN AND J. SWIFT, *The dynamics of  $n$  weakly coupled identical oscillators*, J. Nonlinear Sci., 2 (1992), pp. 69–108, doi:10.1007/BF02429852.
- [3] S. BAK, J. McLAUGHLIN, AND D. RENZI, *Some improvements for the fast sweeping method*, SIAM J. Sci. Comput., 32 (2010), pp. 2853–2874, doi:10.1137/090749645.
- [4] R. BELLMAN, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.
- [5] M. BOUE AND P. DUPUIS, *Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control*, in Proceedings of the 36th IEEE Conference on Decision and Control, 4 (1997), pp. 667–695, doi:10.1109/CDC.1997.652377.
- [6] E. BROWN, J. MOEHLIS, AND P. HOLMES, *On the phase reduction and response dynamics of neural oscillator populations*, Neural Comput., 16 (2004), pp. 673–715, doi:10.1162/089976604322860668.
- [7] A. CHACON AND A. VLADIMIRSKY, *Fast two-scale methods for eikonal equations*, SIAM J. Sci. Comput., 34 (2012), pp. A547–A578, doi:10.1137/10080909X.
- [8] A. CHACON AND A. VLADIMIRSKY, *A parallel two-scale method for eikonal equations*, SIAM J. Sci. Comput., 37 (2015), pp. A156–A180, doi:10.1137/12088197X; preprint available from arXiv:1306.4743v1.
- [9] E. A. CODDINGTON AND N. LEVINSON, *Theory of Ordinary Differential Equations*, McGraw–Hill, New York, 1955.
- [10] M. CRANDALL AND P. LIONS, *Two approximations of solutions of Hamilton–Jacobi equations*, Math. Comp., 43 (1984), pp. 1–19, doi:10.2307/2007396.
- [11] P. DANZL, J. HESPANHA, AND J. MOEHLIS, *Event-based minimum-time control of oscillatory neuron models: Phase randomization, maximal spike rate increase, and desynchronization*, Biological Cybernet., 101 (2009), pp. 387–399, doi:10.1007/s00422-009-0344-3.
- [12] M. DETRIXHE AND F. GIBOU, *Hybrid massively parallel fast sweeping method for static Hamilton–Jacobi equations*, J. Comput. Phys., 322 (2016), pp. 199–223, doi:10.1016/j.jcp.2016.06.023.
- [13] M. DETRIXHE, F. GIBOU, AND C. MIN, *A parallel fast sweeping method for the eikonal equation*, J. Comput. Phys., 237 (2013), pp. 46–55, doi:10.1016/j.jcp.2012.11.042.
- [14] J. GUCKENHEIMER, *Isochrons and phaseless sets*, J. Math. Biol., 273 (1975), pp. 259–273, doi:10.1007/BF01273747.
- [15] A. GUILLAMON AND G. HUGUET, *A computational and geometric approach to phase resetting curves and surfaces*, SIAM J. Appl. Dyn. Syst., 8 (2009), pp. 1005–1042, doi:10.1137/080737666.
- [16] D. HANSEL, G. MATO, AND C. MEUNIER, *Phase dynamics for weakly coupled Hodgkin–Huxley neurons*, Europhys. Lett., 367 (1993), pp. 367–372, doi:0295-5075/23/5/011.
- [17] A. HODGKIN AND A. HUXLEY, *A quantitative description of membrane current and its application to conduction and excitation in nerve*, J. Physiol., 117 (1952), pp. 500–544, <http://www.ncbi.nlm.nih.gov/pmc/articles/pmc1392413/>.
- [18] F. C. HOPPENSTEADT AND E. M. IZHIKEVICH, *Weakly Connected Neural Networks*, Springer-Verlag, New York, 1997.
- [19] G. HUGUET AND R. DE LA LLAVE, *Computation of limit cycles and their isochrons: Fast algorithms and their convergence*, SIAM J. Appl. Dyn. Syst., 12 (2013), pp. 1763–1802, doi:10.1137/120901210.
- [20] E. M. IZHIKEVICH, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*, MIT Press, Cambridge, MA, 2007, [http://cns-pc62.bu.edu/cn510/Papers/Izhikevich\\_Ch8.pdf](http://cns-pc62.bu.edu/cn510/Papers/Izhikevich_Ch8.pdf).
- [21] W.-K. JEONG AND R. T. WHITAKER, *A fast iterative method for eikonal equations*, SIAM J. Sci. Comput., 30 (2008), pp. 2512–2534, doi:10.1137/060670298.
- [22] G.-S. JIANG AND D. PENG, *Weighted ENO schemes for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21 (2000), pp. 2126–2143, doi:10.1137/S106482759732455X.

- [23] G.-S. JIANG AND C.-W. SHU, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.
- [24] C.-Y. KAO, S. OSHER, AND Y.-H. TSAI, *Fast sweeping methods for static Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 42 (2005), pp. 2612–2632, doi:10.1137/S0036142902419600.
- [25] J. P. KEENER AND J. SNEYD, *Mathematical Physiology*, Springer, New York, 1998.
- [26] N. KOPELL AND G. B. ERMENTROUT, *Phase transitions and other phenomena in chains of coupled oscillators*, SIAM J. Appl. Math., 50 (1990), pp. 1014–1052, doi:10.1137/0150062.
- [27] M. KRUPA, N. POPOVIĆ, N. KOPELL, AND H. G. ROTSTEIN, *Mixed-mode oscillations in a three time-scale model for the dopaminergic neuron*, Chaos, 18 (2008), 015106, doi:10.1063/1.2779859.
- [28] Y. KURAMOTO, *Chemical Oscillations, Waves, and Turbulence*, Springer, Berlin, 1984, <http://link.springer.com/book/10.1007%2F978-3-642-69689-3>.
- [29] P. LESAINTE AND P.-A. RAVIART, *On a finite element method for solving the neutron transport equation*, in Mathematical Aspects of Finite Elements in Partial Differential Equations, Academic Press, New York, 1974, pp. 89–123.
- [30] A. MAUROY AND I. MEZIĆ, *On the use of Fourier averages to compute the global isochrons of (quasi)-periodic dynamics*, Chaos, 22 (2012), 033112, doi:10.1063/1.4736859.
- [31] G. S. MEDVEDEV AND J. E. CISTERNAS, *Multimodal regimes in a compartmental model of the dopamine neuron*, Phys. D, 194 (2004), pp. 333–356, doi:10.1016/j.physd.2004.02.006.
- [32] J. MOEHLIS, *Improving the precision of noisy oscillators*, Phys. D, 272 (2014), pp. 8–17, doi:10.1016/j.physd.2014.01.001.
- [33] J. MOEHLIS, E. SHEA-BROWN, AND H. RABITZ, *Optimal inputs for phase models of spiking neurons*, J. Comput. Nonlinear Dynam., 1 (2006), pp. 358–367, doi:10.1115/1.2338654.
- [34] C. MORRIS AND H. LECAR, *Voltage oscillations in the barnacle giant muscle fiber*, Biophys. J., 35 (1981), pp. 193–213, doi:10.1016/S0006-3495(81)84782-0.
- [35] A. NABI, M. MIRZADEH, F. GIBOU, AND J. MOEHLIS, *Minimum energy desynchronizing control for coupled neurons*, J. Comput. Neurosci., 34 (2013), pp. 259–271, doi:10.1007/s10827-012-0419-3.
- [36] S. OSHER AND C.-W. SHU, *High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 28 (1991), pp. 907–922, doi:10.1137/0728049.
- [37] H. M. OSINGA AND J. MOEHLIS, *Continuation-based computation of global isochrons*, SIAM J. Appl. Dyn. Syst., 9 (2010), pp. 1201–1228, doi:10.1137/090777244.
- [38] J. QIAN, Y.-T. ZHANG, AND H.-K. ZHAO, *A fast sweeping method for static convex Hamilton–Jacobi equations*, J. Sci. Comput., 31 (2007), pp. 237–271, doi:10.1007/s10915-006-9124-6.
- [39] J. RINZEL AND G. B. ERMENTROUT, *Analysis of neural excitability and oscillations*, in Methods in Neuronal Modeling, C. Koch and I. Segev, eds., MIT Press, Cambridge, MA, 1989, pp. 135–169.
- [40] J. A. SETHIAN, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. USA, 93 (1996), pp. 1591–1595, <http://www.pnas.org/content/93/4/1591.abstract>.
- [41] J. A. SETHIAN AND A. VLADIMIRSKY, *Ordered upwind methods for static Hamilton–Jacobi equations*, Proc. Natl. Acad. Sci. USA, 98 (2001), pp. 11069–11074, doi:10.1073/pnas.201222998.
- [42] J. A. SETHIAN AND A. VLADIMIRSKY, *Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms*, SIAM J. Numer. Anal., 41 (2003), pp. 325–363, doi:10.1137/S0036142901392742.
- [43] P. A. TASS, *Phase Resetting in Medicine and Biology*, Springer, New York, 1999.
- [44] Y.-H. R. TSAI, L.-T. CHENG, S. OSHER, AND H.-K. ZHAO, *Fast sweeping algorithms for a class of Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 41 (2003), pp. 673–694, doi:10.1137/S0036142901396533.
- [45] J. N. TSITSIKLIS, *Efficient algorithms for globally optimal trajectories*, IEEE Trans. Automat. Control, 40 (1995), pp. 1528–1538.
- [46] S. WIGGINS, *Normally Hyperbolic Invariant Manifolds in Dynamical Systems*, Springer, New York, 1994.
- [47] C. J. WILSON AND J. C. CALLAWAY, *Coupled oscillator model of the dopaminergic neuron of the substantia nigra*, J. Neurophysiol., 83 (2000), pp. 3084–3100, PubMed:10805703.
- [48] D. WILSON AND J. MOEHLIS, *Optimal chaotic desynchronization for neural populations*, SIAM J. Appl. Dyn. Syst., 13 (2014), pp. 276–305, doi:10.1137/120901702.
- [49] A. WINFREE, *Patterns of phase compromise in biological cycles*, J. Math. Biol., 95 (1974), doi:10.1007/BF02339491.

- [50] A. WINFREE, *The Geometry of Biological Time*, 2nd ed., Springer, New York, 2001.
- [51] Y.-T. ZHANG, H.-K. ZHAO, AND J. QIAN, *High order fast sweeping methods for static Hamilton-Jacobi equations*, J. Sci. Comput., 29 (2005), pp. 25–56, doi:10.1007/s10915-005-9014-3.
- [52] H. ZHAO, *A fast sweeping method for eikonal equations*, Math. Comp., 74 (2004), pp. 603–627.
- [53] H. ZHAO, *Parallel implementations of the fast sweeping method*, J. Comput. Math., 25 (2007), pp. 421–429.