Interactive Digital Hearts: Over 25 Years of Web-Based Cardiac Simulation and Visualization via a Web Browser

Flavio H Fenton[1], Abouzar Kaboudian[1], Elizabeth M Cherry[1]

[1] Georgia Institute of Technology, Atlanta, GA, USA

Abstract

For over 25 years, TheVirtualHeart.org has provided interactive cardiac electrophysiology simulations for education, research, and visualization. Beginning with Java applets, the platform has evolved through JavaScript, WebGL, and most recently WebGPU, enabling increasingly complex simulations to run interactively in a web browser without specialized software. The platform now includes more than 20 electrophysiological models spanning single cells, one- and two-dimensional tissue, and patient's anatomically correct three-dimensional atrial and ventricular simulations, in addition to experimental optical-mapping data and a virtual-reality museum. GPU acceleration and high-performance numerical methods now allow even detailed ionic models with tens of coupled differential equations to be simulated at or near real time while users interact directly with the tissue. TheVirtualHeart.org demonstrates how modern web technologies can democratize access to advanced cardiac modeling, providing a unified and interactive framework for teaching, hypothesis testing, and dissemination of complex electrophysiological phenomena.

1. Introduction

Interactive visualization has played an important role in nonlinear dynamics and cardiac electrophysiology. In the early 1990s, numerical simulations were commonly performed on Unix scientific workstations using the X Window System. During graduate work, one of us (F.H.F.) encountered Michael Creutz's Xtoys[1], which demonstrated how numerical computation and visualization could be combined for interactive exploration of dynamical systems. This motivated F.H.F. to develop *X-Spiral* in 1994 (Fig. 1A), an X11 interface with Fortran solvers for two-dimensional FitzHugh–Nagumo (FHN)[2] and Karma[3] excitable-media models. Spiral waves could be studied interactively, with parameters modified and tissue stimulated or ablated directly with the mouse.

A second important step came with the emergence of interactive programs embedded within web pages. In 1997, after encountering Ajiki's *OrbitViewer*, an interactive Java applet developed for three-dimensional visualization of solar-system orbits [4], F.H.F. was struck by the ability to manipulate a scientific visualization directly within a web browser and, importantly, to run the same application across different computer platforms. The application used a Java applet, suggesting that the same technology could be used to make interactive simulations of cardiac dynamics broadly accessible through the Web. After learning Java, he developed an applet to explore cardiac restitution maps interactively (Fig. 1B), allowing the pacing period to be varied manually with the mouse and the onset of alternans through a period-doubling bifurcation to be observed directly. This was soon extended to one- and two-dimensional simulations of propagating waves (Fig. 1C-D), spiral waves, and spatiotemporal dynamics by combining Java and Fortran for fast simulations[5]. In 1999, F.H.F bought the *TheVirtualHeart.org* domain and created the repository to make these simulations freely accessible through a web browser. Beginning in 2000, the project became a collaborative effort between E.M.C and F.H.F and expanded from simplified excitable-media models to increasingly detailed models of cardiac electrophysiology. As new ionic models became available, interactive implementations were added for ventricular and atrial cells and tissues, eventually ranging from reduced models with only a few variables to detailed ionic descriptions with tens of state variables[6]. The objective remained the same: rather than presenting only static figures or prerecorded simulations, users could interact directly with the underlying mathematical models, modify parameters, stimulate tissue, initiate or terminate waves and arrhythmias in real time, and immediately observe the resulting dynamics. The platform also evolved with emerging visualization and web technologies. In 2004, after learning VRML[7] and X3D FHF created a virtual museum containing interactive 3D cardiac structures, simulations, and exhibits (Fig. 1E-F). we also explored an emerging approach for distributing interactive three-dimensional scientific content by embedding 3D cardiac structures and simulations directly within PDF documents, work that received

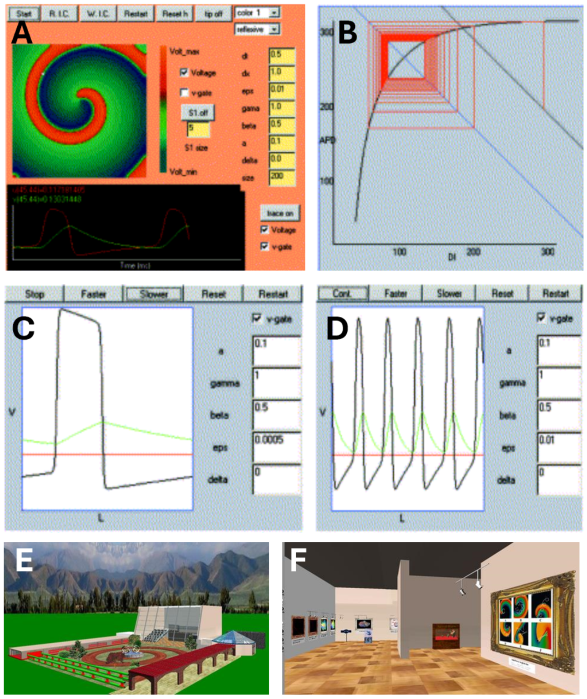

Figure 1. A) Spiral wave in the Karma model using *X-Spiral*. B) First *TheVirtualHeart.org* Java applet, showing an interactive restitution map and the development of alternans. C–D) One-dimensional FHN simulations. E–F) Exterior and interior views of the interactive 3D Virtual Heart Museum.

first prize in the Technical Publishing category of the first Acrobat 3D PDF Contest in 2008. As browsers evolved, Java applets were replaced by native JavaScript, with GPU acceleration introduced through WebGL beginning in 2011 and, more recently, WebGPU. *TheVirtualHeart.org* also includes experimental movies of voltage and calcium propagation from animal and human hearts.

Over more than 25 years, these developments have transformed *TheVirtualHeart.org* from a collection of relatively simple interactive Java simulations into a browser-based computational environment spanning single-cell models, one- and two-dimensional tissue, anatomically realistic three-dimensional hearts, electromechanics, and immersive visualization. In this work, we describe the evolution of this platform and illustrate how advances in web and GPU technologies have made increasingly complex cardiac simulations directly accessible and interactive across computing devices.

2. Methods

Evolution of the Web-Based Computational Platform. *TheVirtualHeart.org* project has evolved continuously with web technologies over more than 25 years. The earliest interactive simulations were implemented as Java applets, which allowed cardiac electrophysiology models to be executed directly within a web browser. As browser vendors progressively discontinued support for plug-in architectures, particularly during 2015–2017, Java applets became impractical for web-based scientific applications. The simulations were therefore progressively migrated to JavaScript, allowing them to run natively in modern browsers without external plug-ins or software installation. Beginning in 2011, we also developed GPU-accelerated implementations for research and teaching using CUDA C and CUDA FORTRAN[8, 9], exploiting graphic cards to perform highly parallelalizable, the numerical calculations required in large cardiac reaction–diffusion simulations. In 2012 we then transitioned to WebGL shaders programs, allowing not only faster simulations[10] but also visualization and interactivity at the same time within a web browser.

A central feature of this approach is its platform independence: simulations are distributed as web pages and can run, without compilation or installation, across operating systems and hardware ranging from desktop and laptop computers to tablets and mobile devices, subject to browser and GPU capabilities. The same framework supports models spanning single cells (0D) and one-, two-, and three-dimensional tissue simulations, including anatomically realistic cardiac geometries and detailed ionic models. Interactive visualization is performed concurrently with the simulation, allowing users to modify parameters, apply stimuli, initiate or terminate arrhythmias, and immediately observe the resulting dynamics. Three-dimensional and immersive visualization has also evolved from early VRML implementations to GPU-accelerated interactive 3D and virtual-reality environments. GPU parallelization is combined with optimized numerical integration methods that permit stable and accurate time steps typically in the order of 0.1 ms, allowing even complex ionic models, with tens of variables, to remain interactive.

3. Results

- **Interactive Maps and Mechanisms of Alternans.** Interactive discrete maps on *TheVirtualHeart.org* are able to easily illustrate nonlinear mechanisms underlying cardiac alternans and complex rhythms, including APD restitution maps exhibiting period-doubling, pitchfork and border-collision bifurcations, and biphasic or supernormal restitution. Complementary calcium maps demonstrate calcium-driven alternans arising from nonlinear intracellular calcium cycling. Users can manipulate parameters and directly observe transitions from stable rhythms to alternans and chaos.
- **Interactive Cardiac Cell Models.** A much larger collection allows users to investigate the dynamics of individ-

ual excitable cells. The models span generic excitable systems such as FitzHugh–Nagumo, Barkley, Karma, Fenton–Karma, and Mitchell–Schaeffer, as well as detailed ionic descriptions of neuronal, Purkinje, sinoatrial, atrial, and ventricular cells. Representative cardiac models include the Noble (4 variables), Beeler–Reuter (8v), Luo–Rudy (8v), Courtemanche–Ramirez–Nattel (21v), Nygren et al. (29v), ten Tusscher–Noble–Noble–Panfilov (17v), Iyer–Mazhari–Winslow (67v), and O'Hara–Virag–Varro–Rudy (41v) models, among others. The collection therefore spans several generations of cardiac electrophysiological modeling, from early descriptions containing only a few dynamical variables to modern ionic models containing dozens of coupled state variables. The most recent implementations extend this progression to the 93-variable Tomek et al. model. Importantly, these are not simply demonstrations of prerecorded action potentials. Users can change pacing protocols and model parameters and visualize membrane voltage together with individual ionic currents, gating variables, and ionic concentrations. Thus, the programs function like small interactive electrophysiology experiments where research questions can be investigated.

- **Propagation and Reentry in One-Dimensional Tissue.** Many complex cellular models are also implemented in one-dimensional cables and rings, extending single-cell dynamics to propagating tissue. Users can stimulate arbitrary locations to create or terminate waves and reentry and investigate refractoriness, conduction block, and spatially concordant or discordant alternans. Dedicated programs additionally explore liminal length, strength-duration relationships, space and time constants, membrane capacitance and conductances, and S1–S2 initiation and conduction block, with parameters and stimulation protocols adjustable during simulation.
- **GPU-Accelerated Two-Dimensional Cardiac Tissue.** The transition from JavaScript CPU calculations to GPU computation through WebGL, and more recently WebGPU, greatly increased the spatial and model complexity that could remain interactive. Two-dimensional programs include classical excitable-media models such as FitzHugh–Nagumo and Karma, simplified cardiac models, and increasingly detailed ionic descriptions including Noble, Beeler–Reuter, Luo–Rudy, and more recent human ventricular models. Simulations can be performed in simple square domains as well as anatomical tissue geometries such as ventricular slices. Direct mouse stimulation allows waves and spiral waves to be initiated or terminated while parameters can be modified and the resulting dynamics visualized as the simulation is running. With GPU acceleration, even substantially more complex ionic descriptions can be simulated interactively. For example, the recent 93-variable Tomek et al. model can be integrated on a 500×500 tissue domain while the resulting electrical activity is calculated and visualized concurrently, as show in Fig 2A. This corresponds to approximately $500 \times 500 \times 93 = 23.25$ million dynamical state variables being evolved at every time step. The ability to interactively evolve more than 23 million state variables within an ordinary web browser illustrates the substantial increase in computational scale made possible by browser-accessible GPU parallelism and the progression of *TheVirtualHeart.org* from its original Java applets to modern WebGL- and WebGPU-based simulations.
- **Virtual Museum and Interactive Three-Dimensional Cardiac Structures.** The extension from two-dimensional visualization to interactive three-dimensional cardiac structures began with the development of the Virtual Heart Museum using VRML and, subsequently, X3D. The virtual museum allows users to navigate through a three-dimensional environment containing cardiac structures, educational material, and interactive exhibits. These early implementations demonstrated the potential of the Web not only for distributing simulations, but also for allowing users to explore cardiac anatomy and computational results within interactive three-dimensional environments. The original VRML and X3D implementations, however, depend on technologies and external browser plug-ins that are no longer widely supported. We are therefore currently converting and extending these three-dimensional structures using WebGPU. The objective is to make the virtual museum and its interactive cardiac structures accessible directly within modern web browsers, without external plug-ins, while also providing substantially improved rendering and interaction. We anticipate making the first version of this new implementation available in early 2027.
- **Interactive Simulations in Three-Dimensional Heart Structures.** In collaboration with Abouzar Kaboudian, we developed WebGL-based methods for interactive GPU simulation of cardiac reaction–diffusion models in anatomically detailed hearts, including patient-specific geometries[11–13]. Using Abubu.js, a WebGL-based computational framework developed by our group to simplify the implementation of GPU-accelerated simulations, computation and visualization occur through the browser, allowing users to rotate and inspect the heart, modify parameters, and stimulate tissue to initiate or terminate three-dimensional arrythmias, without compiled software or computing clusters, as shown in Fig.2 B-C. Over the past several years, we have continued to develop numerical and computational methods to increase the speed and accuracy of these simulations. An important difficulty in representing anatomically realistic hearts with Cartesian grids is that much of the surrounding three-dimensional computational domain contains no cardiac tissue. Sparse-grid load balancing and data-compression methods were therefore developed to avoid calculations and memory opera-

tions in these empty regions and to distribute the computational work more efficiently among GPU cores[14, 15]. More recent compressed Cartesian-grid implementations reduce GPU memory requirements by approximately an order of magnitude and can accelerate calculations by as much as a factor of 20 compared with the corresponding uncompressed implementation. Improved discretization schemes also increase the accuracy with which thin anatomical structures can be represented without requiring prohibitively fine grids [15]. Together with improved numerical integration schemes and the transition from WebGL to WebGPU, these developments are continuing to increase the complexity of ionic models and anatomical structures that can be simulated interactively. Consequently, simulations that historically required specialized high-performance computing resources can now be performed on widely available hardware through a web browser, providing a practical route toward interactive patient-specific simulations and virtual-heart applications.

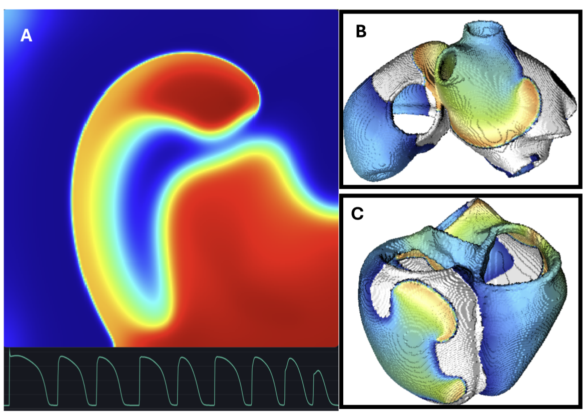

Figure 2. A) Two-dimensional simulation of the 93-variables 2026 Tomek et al. model (epicardial parameters) on a 400×400 grid ($dt = 0.1$ ms, $dx = 0.25$ mm) in WebGPU. One spiral rotation ($T \sim 225$ ms) requires ~ 4 s of wall-clock time ($\sim 18\times$ slower than real time) on a ThinkPad laptop. B) Three-variable model fitted to patient data in a patient-specific 3D atrial anatomy with one spiral wave in WebGL. C) 41-variables 2011 O'Hara–Virág–Varró–Rudy model in a patient-specific 3D ventricular anatomy with three spiral waves in WebGL.

Acknowledgments

This study was supported by NIH grant number 2R01HL143450–05 and NSF DMS-2505837.

References

[1] Creutz M. Xtoys: Cellular automata on xwindows. Nuclear Physics B Proceedings Supplements 1996;47:846–849.

[2] Izhikevich EM, FitzHugh R. Fitzhugh-Nagumo model. Scholarpedia 2006;1(9):1349.

[3] Karma A. Spiral breakup in model equations of action potential propagation in cardiac tissue. Physical Review Letters 1993;71(7):1103.

[4] Ajiki O. Orbitviewer, 1996 URL https:// www.astroarts.co .jp/products/orbitviewer/ index.html. Interactive Java applet for three-dimensional visualization of solar-system orbits.

[5] Fenton FH, Cherry EM, Hastings HM, Evans SJ. Real-time computer simulations of excitable media: Java as a scientific language and as a wrapper for C and FORTRAN programs. Biosystems 2002;64(1-3):73–96.

[6] Fenton FH, Cherry EM. Models of cardiac cell. Scholarpedia 2008;3(8):1868.

[7] Ames AL, Nadeau DR, Moreland JL. The VRML 2.0 Sourcebook. 2nd edition. New York: John Wiley & Sons, 1997. ISBN 0-471-16507-7.

[8] Bartocci E, Cherry EM, Glimm J, Grosu R, Smolka SA, Fenton FH. Toward real-time simulation of cardiac dynamics. In Proceedings of the 9th International Conference on Comp. Meth. in Sys. Bio. 2011; 103–112.

[9] Bartocci E, et al. Teaching cardiac electrophysiology modeling to undergraduate students: laboratory exercises and gpu programming for the study of arrhythmias and spiral wave dynamics. Advances in physiology education 2011; 35(4):427–437.

[10] Kaboudian A, Velasco-Perez HA, Iravanian S, Shiferaw Y, Cherry EM, Fenton FH. A comprehensive comparison of GPU implementations of cardiac electrophysiology models. In From Reactive Systems to Cyber-Physical Systems. Springer, 2019; 9–34.

[11] Kaboudian A, Cherry EM, Fenton FH. Large-scale interactive numerical experiments of chaos, solitons and fractals in real time via GPU in a web browser. Chaos Solitons Fractals 2019;121:6–29.

[12] Kaboudian A, Cherry EM, Fenton FH. Real-time interactive simulations of large-scale systems on personal computers and cell phones: Toward patient-specific heart modeling and other applications. Science advances 2019; 5(3):eaav6019.

[13] Kaboudian A, Cherry EM, Fenton FH. Real-time interactive simulations of complex ionic cardiac cell models in 2d and 3d heart structures with GPUs on personal computers. In 2021 Computing in Cardiology (CinC), volume 48. IEEE, 2021; 1–4.

[14] Kaboudian A, Cherry EM, Fenton FH. GPU load balancing using sparse cartesian grids: Making interactive WEBGL simulations of complex ionic models even faster on 3d heart structures. In 2023 Computing in Cardiology (CinC), volume 50. IEEE, 2023; 1–4.

[15] Kaboudian A, Gray RA, Uzelac I, Cherry EM, Fenton FH. Fast interactive simulations of cardiac electrical activity in anatomically accurate heart structures by compressing sparse uniform cartesian grids. Computer methods and programs in biomedicine 2024;257:108456.

Address for correspondence:
Flavio H. Fenton ffentonn@physics.gatech.edu