CHAOS LAB
Complex Heart Arrhythmias and other Oscillating Systems

Shattering the Supercomputer Barrier of Computationally Expensive Problems by Personal Computers

A WebGL 2.0 Approach

Abouzar Kaboudian abouzar.kaboudian@physics.gatech.edu

Elizabeth Cherry echerry30@gatech.edu

Flavio H. Fenton flavio.fenton@physics.gatech.edu

Abstract

Over the past decades, as our understanding of the fundamental principles governing physical systems has developed, and subsequently, our mathematical models have advanced, the complexity and the computational costs of these models have increased significantly. This is a common challenge in various fields from fractal studies, to fluid mechanics, crystal growth, or even volcanic lava flows. Many of these mathematical models, can benefit from parallel computing to split the work among multiple processors to achieve the solutions faster.

In this work, the simulation of various complex models that normally require huge computational power are made possible on personal computers and even cell-phones through a browser-based technology called WebGL 2.0. Our in-house library, Abubu.js, enables easy implementation of user friendly and interactive computational tools by using WebGL 2.0 on personal computers and even cell-phones.

Power of GPUs

Recently, the advantages of GPU over CPU processing have been established for many areas of science, including biological systems. Graphics processing units (GPUs), now standard in most devices, contain hundreds if not thousands of processor cores which allow massive parallelization. This, effectively, turn GPU enabled devices into personal supercomputers. Using the WebGL, we have been able to exploit the highly parallel and multi-core capabilities of GPUs to achieve extremely fast simulations of complex models in 2D and 3D.

WebGL

WebGL stands for Web Graphics Library which is a JavaScript API for rendering interactive 2D and 3D computer graphics within any compatible web browser without the use of plug-ins.

WebGL enables us to carry out high-performance parallel simulations over the Web on a local personal computer or a laptop with the following advantages:

- the code will be independent of the operating system;
- there will be no need to install any additional plug-in as long as the user has a modern browser, such as Google Chrome or FireFox;
- users do not need to compile the code;
- the code is instantly available over the Web and the user can run the code by just accessing a URL;
- updates and fixes can be easily sent to all users as the program is over the web;
- JavaScript and HTML enable us to easily interact with the simulation, change parameters, and see the results in real-time;

Fractals

Complex return maps are classical examples of problems that can benefit from parallel computations. We have successfully applied our techniques to study complex return maps and their fractal nature.

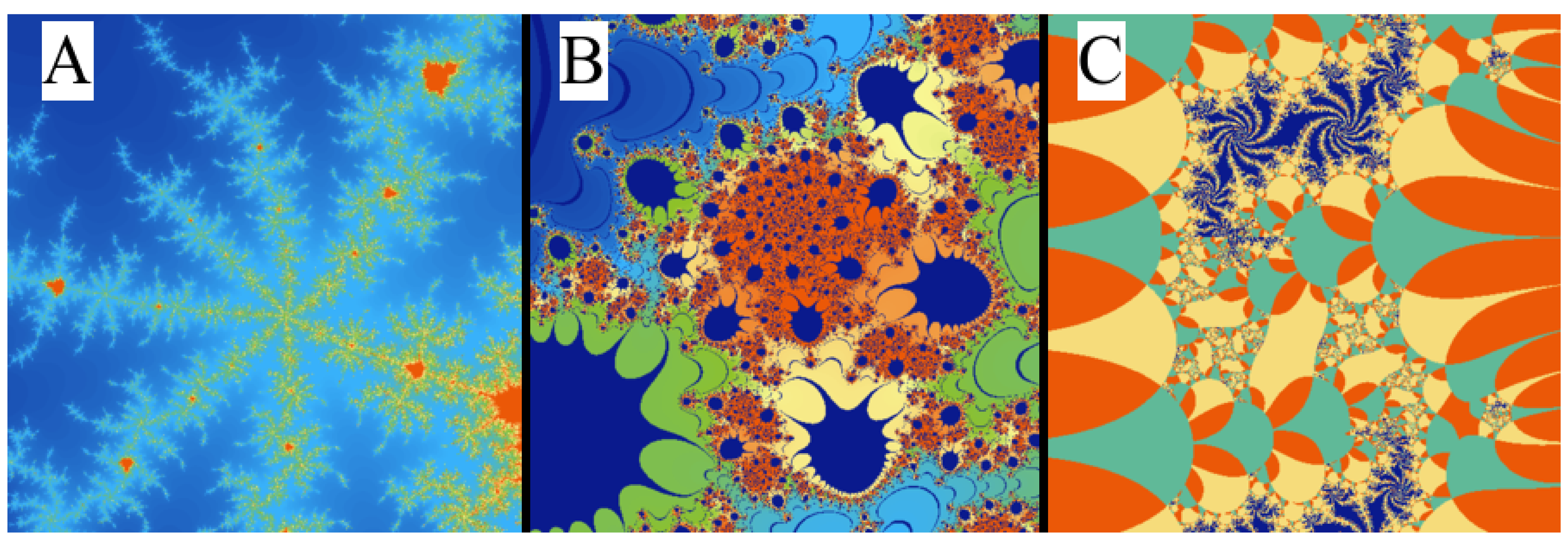

Figure 1. Three fractals studied using WebGL 2.0 applications developed using Abubu.js: **A)** a branch of the infamous Mandelbrot set that shows multiple copies of the set; **B)** a biomorph-like region identified in the infamous Julia set. **C)** a depiction of a return map revealing biomorph and Mandala like features at the same time. For more information see reference [1].

Interactive fluid flow simulations

The Lattice Boltzmann Method is a parallel friendly approach for simulating fluid flow. We have successfully applied this method to simulate external fluid flow around obstacles.

Figure 2. the WebGL 2.0 implementation of the LBM method provides an interactive simulation of the flow around obstacles that can be added interactively during simulation. For more information on these simulations see reference [2].

Real-Time 2D Simulations

Using an NVIDIA Titan V graphics card, we could achieve 3,100 time steps of solution on a 512x512 grid for the **41-variable** OVVR model [3]. This means we could solve

$$512 \times 512 \times 41 \times 3,100 \approx 3.3 \times 10^{10}$$

ODEs per second! This is the order of computational power required to do real-time simulations in 2D, and near real-time simulations in 3D.

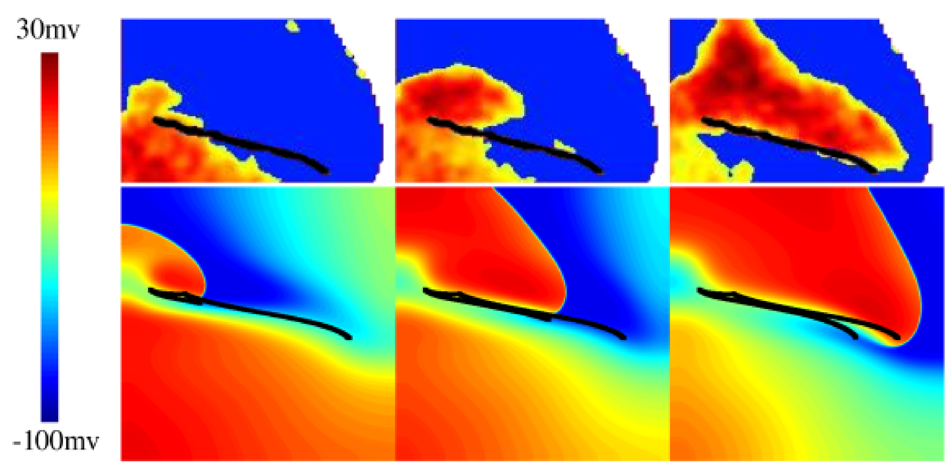

Figure 3. Reproduction of optically observed data (top) in numerical simulations of the **41-variable** OVVR [3] model.

Supercomputing of cardiac models on realistic 3D structures using Cell-Phones

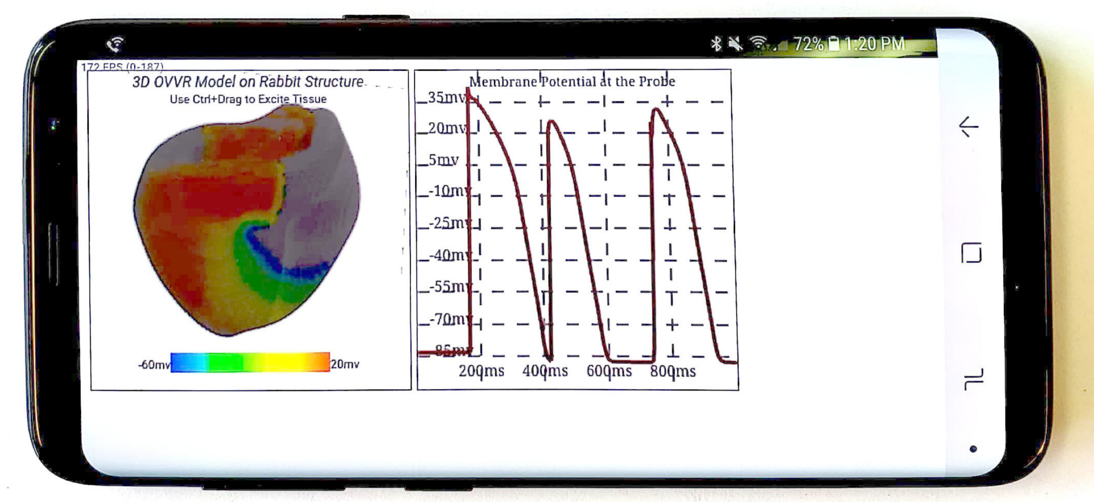

Figure 4. 3D simulations of the **41-variable** OVVR [3] model on a Galaxy S8 cell-phone.

Now, using WebGL, it is possible to model realistic models of cardiac tissue both in 2D and 3D not only using a personal computer but also using a cell-phone. This can help the clinicians in diagnosing cardiac problems, researches in advancing the frontiers in global system studies, or simply teaching students how re-entry and breakup can lead to fibrillation.

Acknowledgments

This work was partially funded by NSF grants CNS-1446675 & CNS-1446312.

References

[1] Abouzar Kaboudian, Elizabeth M Cherry, and Flavio H Fenton. Large-scale interactive numerical experiments of chaos, solitons and fractals in real time via gpu in a web browser. *Chaos, Solitons & Fractals*, 121:6–29, 2019.

[2] Abouzar Kaboudian, Elizabeth M Cherry, and Flavio H Fenton. Real-time interactive simulations of large-scale systems on personal computers and cell phones: Toward patient-specific heart modeling and other applications. *Science advances*, 5(3):eaav6019, 2019.

[3] Thomas O'Hara, László Virág, András Varró, and Yoram Rudy. Simulation of the undiseased human cardiac ventricular action potential: model formulation and experimental validation. *PLoS Comput Biol*, 7(5):e1002061, 2011.

For more information, visit *https://abubujs.org*