Advances in Engineering Software 98 (2016) 1–11

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Application of smoothed particle hydrodynamics to modeling mechanisms of biological tissue

Andrey Palyanov[a,b,c,*], Sergey Khayrulin[a,b,c], Stephen D. Larson[b]

[a] A.P. Ershov Institute of Informatics Systems, Novosibirsk, Russia
[b] OpenWorm Project, San Diego, CA, USA
[c] Novosibirsk State University, Novosibirsk, Russia

ARTICLE INFO

Article history:
Received 6 August 2015
Revised 3 February 2016
Accepted 8 March 2016
Available online 31 March 2016

MSC:
68
70
74
76
92

Keywords:
Computational fluid dynamics
PCISPH
Biomechanics
Elastic matter
Liquid impermeable elastic membranes
Contractile matter
High-performance parallel computing
Open science

ABSTRACT

A prerequisite for simulating the biophysics of complex biological tissues and whole organisms are computational descriptions of biological matter that are flexible and can interface with materials of different viscosities, such as liquid. The landscape of software that is easily available to do such work is limited and lacks essential features necessary for combining elastic matter with simulations of liquids. Here we present an open source software package called Sibernetic, designed for the physical simulation of biomechanical matter (membranes, elastic matter, contractile matter) and environments (liquids, solids and elastic matter with variable physical properties). At its core, Sibernetic is built as an extension to Predictive–Corrective Incompressible Smoothed Particle Hydrodynamics (PCISPH). Sibernetic is built on top of OpenCL, making it possible to run simulations on CPUs or GPUs, and has 3D visualization support built on top of OpenGL. Several test examples of the software running and reproducing physical experiments, as well as performance benchmarks, are presented and future directions are discussed.

1. Introduction

In recent years, investigations of the physical simulation of tissues and whole organisms has increased in interest. Tissues that have been simulated include cardiac muscle [7,18], brain tissue [10], and liver tissue [9]. Methods have been developed for virtual surgery [8,25], including blood flow through deformable tissues [23,24] (Appendix B) and some platforms have been created in the service of simulating artificial embodied autonomous agents, such as AnimatLab [5]. While these efforts have produced important results and publicly available computer code that is available online, there are still significant gaps in the landscape of software implementations of biological tissue simulators that inhibit further research in this space. This is often because the mechanical physics of biological entities is different from the mechanical physics of engineered devices that are built from conventional solid materials, and simulation engines have more often handled the latter case than the former.

Biological entities often have physical features that are not found in standard mechanical devices such as elastic matter for an outer shell (skin or cell membranes) and internal reservoirs of liquids or gels (blood, brain fluid, cytoplasm). Parts responsible for active movement (such as muscles) also require elastic matter that can contract on demand, which we call contractile matter. Biological entities often interact with liquids or gels that are incompressible in an external environment as well, which means that the surfaces of the elastic matter interact with the surfaces of the liquids.

At their core, physical simulations must decompose the mechanical physics of their object of interest into components and implement algorithms that are capable of reproducing them. Because we are interested in simulating the physical dynamics of complex biological tissues, we sought an algorithmic modeling

---

* Corresponding author at: A.P. Ershov Institute of Informatics Systems, Russian Academy of Sciences, Siberian Branch, Acad. Lavrentjev pr. 6, Novosibirsk 630090, Russia. Tel.: +79139577150.

E-mail addresses: palyanov@iis.nsk.su (A. Palyanov), skhayrulin@iis.nsk.su (S. Khayrulin), stephen@openworm.org (S.D. Larson).

http://dx.doi.org/10.1016/j.advengsoft.2016.03.002
0965-9978/

approach that would enable arbitrarily complex tissues to be digitally reproduced as simply as possible. During our investigation for such an approach, we came across smoothed-particle hydrodynamics (SPH).

Initially developed for astrophysical modeling and simulation, smoothed-particle hydrodynamics is a computational method for simulating fluids [19]. It represents fluids as a system of interacting elements (referred to as particles) rather than as meshes. The movement of particles is based on the efficient calculation of the forces and pressures each particle experiences every time step. Each particle has an associated spatial distance h (the smoothing distance) over which the forces that act upon it are estimated by a smoothing kernel function.

Predictive–corrective incompressible SPH (PCISPH) is a recently-developed modification to SPH that enforces incompressibility of particles, keeping volumes of liquid constant despite the movement of their constituent particles, using a prediction correction scheme [22]. This method improves over other leading SPH algorithms such as weakly compressible SPH (WCSPH) [13] and incompressible SPH (ISPH) [21] by providing a balance between computational cost and quality of the solution (See Appendix A for more complete introduction). Work by Fedkiw and colleagues [12,20] used an approach that had equivalent accuracy and speed by coupling a particle level set method with SPH. However, instead of coupling multiple algorithms, PCISPH can be applied as a single, unified algorithm to simulate flexible, deformable and static rigid objects interacting with liquid flows [3] (Appendix B). This unified approach simplifies the computational method.

Wishing to utilize the SPH algorithm, we surveyed the landscape of available SPH implementations. However, we could not find any existing open source software to implement the PCISPH algorithm. Consequently, we created a software implementation of PCISPH and extended it with several enhancements that reproduce key features of biological tissue. We have named this collection of algorithms and computational methods ‘Sibernetic’.

1.1. Review of existing software

Before implementing Sibernetic, we first turned to other software systems that appeared to do what we needed and evaluated them. Table 1 contains the set of evaluated software with our criteria, including being open source, enabling parallel computation, using a particle based method, implementing incompressibility, implementing membranes, having contractile matter and usability. In this case, by usability, we mean a qualitative assessment of the ease of use of the open source library, where high means it was very easy and well maintained and documented, medium means it was somewhat maintained and partially documented, and low means it was not maintained and not well documented.

We did not find any software that met all criteria we needed. As such, we have implemented Sibernetic in order to provide the set of functionality we needed for biological modeling. In the rest of this manuscript, we describe Sibernetic and present extensions to the PCISPH algorithm.

2. Methods

Sibernetic enables the digital reconstruction and simulation of different forms of tissue from lattices of basic particles. In this section, we describe how we implemented different types of simulated matter as well as the software construction of the Sibernetic system.

2.1. Types of simulated matter

Four principle types of matter are simulated in the Sibernetic system (Fig. 1). This includes liquids with variable viscosity, elastic matter, contractile matter, and liquid-impermeable membranes. Matter is represented as individual particles, or lattices composed of particles, whose edges can either be rigid or flexible. These types of matter can be connected together into larger wholes such as simulated muscle tissue. Multiple kinds of tissue can be constructed from different lattice configurations. An additional configurable property is the handling of static boundaries at the limits of the simulated environment.

All of these types of matter rely on the tracking of the x, y, and z positions of particles, as well as the constraints between them.

2.2. Boundary handling

Boundary handling methods are implemented according to the approach of [14], which combines the idea of direct-forcing with a frozen-particles method. Thus, instead of implicit repulsing planes, boundaries are explicitly represented by non-moving particles provided with normal vectors, which enables the construction of complex spatial 3D configurations. This has the advantage, as stated by [14] that: “by incorporating density estimates at the boundary

Table 1
Comparison of software libraries that implements fluid dynamics algorithms.

Name	Stability	Mesh or Particle?	Incompressibility?	Open source?	Parallel implementation?	Membranes?	Contractile matter?
FLOW-3D[1]	High	Mesh	Yes	No	Yes; MPI	Yes	No
OpenFOAM[2]	High	Mesh	Yes	Yes	Yes; MPI	No	No
FLUID v2, v3[3]	Medium	Particle	No	Yes	Yes; CUDA	No	No
Interactive SPH Simulation [17]	N/A (no code available)	Particle	No	No	Yes; CUDA	No	No
ISPH[4]	Low (no official releases)	Particle	Yes	Yes	Yes	No	No
AnimatLab[5]	High	Both	No	Yes	No	No	Yes
OpenTissue[6]	Medium	Particle	No	Yes	No	No	No
Bullet-Fluid[7]	Low	Particle	No	Yes	Yes; OpenCL	No	No
PhysBAM[8]	Medium	Both	Yes	Yes	Yes; MPI	Yes	Yes
Sibernetic	**High**	**Particle**	**Yes**	**Yes**	**Yes; OpenCL**	**Yes**	**Yes**

[1] http://www.flow3d.com.

[2] http://openfoam.com.

[3] http://www.rchoetzlein.com/fluids3/.

[4] http://isph.sourceforge.net/.

[5] http://animatlab.com/.

[6] http://image.diku.dk/kenny/download/opentissue/gallery.html.

[7] http://bulletphysics.org/wordpress/.

[8] http://physbam.stanford.edu/.