Article:

Borau, C. orcid.org/0000-0002-3784-1140, Chisholm, R. orcid.org/0000-0003-3379-9042, Richmond, P. orcid.org/0000-0002-4657-5518 et al. (1 more author) (2024) An agent-based model for cell microenvironment simulation using FLAMEGPU2. Computers in Biology and Medicine, 179. 108831. ISSN 0010-4825

https://doi.org/10.1016/j.compbiomed.2024.108831

Computers in Biology and Medicine 179 (2024) 108831

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

An agent-based model for cell microenvironment simulation using FLAMEGPU2

C. Borau [a,b], R. Chisholm [c], P. Richmond [c], D. Walker [c,*]

[a] Multiscale in Mechanical and Biological Engineering (M2BE), Mechanical Engineering Dept, University of Zaragoza, Zaragoza, Spain
[b] Centro Universitario de la Defensa, Zaragoza, Spain
[c] Department of Computer Science and Insigneo Institute of in Silico Medicine, University of Sheffield, Sheffield, UK

ARTICLE INFO

Keywords:
Agent-based model
FLAMEGPU2
GPU
Cell microenvironment
Extracellular matrix

ABSTRACT

This work presents an advanced agent-based model developed within the FLAMEGPU2 framework, aimed at simulating the intricate dynamics of cell microenvironments. Our primary objective is to showcase FLAMEGPU2's potential in modelling critical features such as cell-cell and cell-ECM interactions, species diffusion, vascularisation, cell migration, and/or cell cycling. By doing so, we provide a versatile template that serves as a foundational platform for researchers to model specific biological mechanisms or processes. We highlight the utility of our approach as a microscale component within multiscale frameworks. Through four example applications, we demonstrate the model's versatility in capturing phenomena such as strain-stiffening behaviour of hydrogels, cell migration patterns within hydrogels, spheroid formation and fibre reorientation, and the simulation of diffusion processes within a vascularised and deformable domain. This work aims to bridge the gap between computational efficiency and biological fidelity, offering a scalable and flexible platform to advance our understanding of tissue biology and engineering.

1. Introduction

The extracellular matrix (ECM) is a complex network of proteins and molecules that surround cells within tissues, providing mechanical support, biochemical signals, and a scaffold for cell attachment and migration [1,2]. It forms the structural framework of tissues and plays a fundamental role in cellular behaviour, tissue development, and overall tissue integrity. Moreover, the ECM facilitates cell communication, essential for tissue homeostasis and repair. Trying to mimic this, hydrogels are widely used both in cell culture for in vitro experiments and in tissue engineering, due to their proven biocompatibility properties and their efficacy in substituting biological tissues [3]. Hydrogels share similar characteristics with the ECM, such as their three-dimensional porous structure, water content, and biocompatibility. Like the ECM, hydrogels can be tailored to have specific mechanical properties [4], which can influence cell behaviour and tissue formation. Additionally, hydrogels can be functionalised with bioactive molecules, such as growth factors, to guide cell differentiation and tissue-specific functions. Moreover, their ability to absorb and retain water allows hydrogels to maintain a hydrated environment for cells, promoting their survival and functionality [5]. By mimicking the ECM, hydrogels offer a promising platform for creating artificial tissues and supporting cell growth. Understanding the intricate interactions of cells within the ECM is crucial for advancing our knowledge of tissue biology, development, and diseases, as well as for guiding the design of biomaterials for tissue engineering and regenerative medicine applications [6,7].

However, in vitro experiments can be expensive and time-consuming, particularly for complex or long-term studies, limiting the number of scenarios that can be explored. Moreover, external factors in the experimental environment may lead to uncontrolled variables, potentially influencing the outcomes and introducing confounding effects, leading to uncertainties in the validity and reliability of the findings. In recent years, numerous computational models of the ECM have been developed [8–10], complementing and extending actual experiments to understand its mechanical behaviour and how it affects cell-matrix interactions [11,12]. Some of these works [13] have gone further by providing a computational framework for simulating cell-ECM interactions by generating random fibre networks that closely resemble actual collagen type I gels in terms of key geometric characteristics, including connectivity distribution, free-fibre length, and orientation correlation between adjacent fibres. These microstructural

---

* Corresponding author.
E-mail address: d.c.walker@sheffield.ac.uk (D. Walker).

https://doi.org/10.1016/j.compbiomed.2024.108831
Received 12 March 2024; Received in revised form 5 June 2024; Accepted 29 June 2024

features have been identified as the primary factors influencing the mechanical properties of fibre networks [14]. These sophisticated models provide valuable insights into the complex behaviour of hydrogels under varying physiological conditions and offer a deeper understanding of their mechanical, chemical, and biological properties. However, integrating such models with individualised cell behaviour may prove challenging, often needing multiscale approaches or hybrid models combining continuum and particle modelling [8,10,11,15].

The use of agent-based models (ABMs) in the biomedical field, especially at the cellular level, has been on the rise in recent years. This has been due to their flexibility and the development of both hardware and software that has combined to enable the simulation of increasing numbers of agents. These models, usually categorised as *microscopic*, due to their direct focus on emergent behaviour arising from the behaviour and interactions of individual agents in a bottom-up methodology, provide a means for investigating intricate phenomena that would otherwise be challenging to analyse using a top-down approach, such as traditional Partial Differential Equations (PDEs). A variety of cell-agent-based model frameworks (Chaste [16], PhysiCell [17], TiSim/CellSys [18], CBMOS [19], Gell [20], and FLAMEGPU2 [21] among others) have originated by extending solutions from particular workflows, incorporating a range of functionalities that allow users to dedicate their time to their research questions rather than to code design and implementation [22]. While some of these platforms, such as PhysiCell, have garnered great popularity, with a substantial and dedicated user community, they are computationally limited when the number of agents increases beyond the tens of thousands, as they are usually based on OpenMP (Open Multiprocessing) and/or MPI (Message Passing Interface) approaches for parallelisation. On the other hand, Gell, CBMOS and FLAMEGPU2 were built to work on graphical processing units (GPUs), allowing simulations to scale to millions of agents whilst remaining computationally viable. CBMOS, for example, leverages CuPy's rapid GPU vector operations for efficient computation, achieving simulation speeds up to 30 times faster than its CPU counterpart. However, its capability to manage large systems is constrained by the platform's design in terms of GPU memory usage, which can surpass 16 GB. In contrast, Gell, while less versatile, addresses these limitations in large-scale hybrid cell simulations effectively, achieving speed increases up to 150 times faster than similar simulations using multi-core PhysiCell, while also being more memory-efficient. It should be noted that PhysiCell_GPU [23] is an ongoing project designed to migrate portions of its serial CPU code to run on GPUs using OpenACC. Although this approach has certain drawbacks, including redundant data transfers between the host and the GPU, it may soon be integrated into the core PhysiCell software. In the same way, FLAMEGPU2 is in active development, and while it does not include predefined and detailed ECM or cell-related functions (as it is a general-purpose platform, rather than being focused on biomedical applications) it provides the necessary tools to do so.

The primary objective of this study is to demonstrate the potential of FLAMEGPU2 as a powerful tool in the biomedical field for simulating the cell-tissue microenvironment, encompassing critical features such as cell-ECM and cell-cell mechanical interactions, species diffusion, cell migration, and cell cycling. By doing so, we aim to provide a foundational template that not only showcases FLAMEGPU2's capabilities in handling complex biological simulations, but also serves as a versatile starting point for researchers without specialised GPU programming expertise, looking to model specific mechanisms or processes within the cell microenvironment. We anticipate that future studies can build on this template by incorporating more detailed models of cellular behaviour, biochemical signalling pathways, and mechanical properties of the ECM. In summary, this work intends to bridge the gap between computational efficiency and biological fidelity, in terms of freedom and versatility, offering a scalable and flexible platform for advancing our understanding of tissue biology and engineering.

2. Methods

In this section, we describe the model structure, as well as the main interactions and mathematical equations that govern it. Importantly, we present it as a base model to build upon, including the basic functionalities to act as a virtual cell laboratory that takes advantage of the power of GPU computation. In this first version, we include ECM, cell and vascularisation agents that interact with each other and with the domain boundaries. All agents can be activated/deactivated to perform different types of experiments. Each of the agent behaviours subsequently presented can be easily adapted and tailored to the user needs with a minimal programming knowledge of Python and/or C++.

2.1. FLAMEGPU2 implementation

The model has been implemented using the FLAMEGPU2 framework [21], a generalised ABM and simulation system which allows not only the abstraction of the GPU's complexity away from the end user, but also provides significant speedup over alternative CPU ABM frameworks. FLAMEGPU2 uses a state-based representation, which permits modellers to describe an agent-based system as a state machine with internal memory that is updated by functions. It is worth noting that serial agent-based simulations typically loop through agents, executing their behaviours sequentially, followed by shuffling agents to avoid order-induced bias. Conversely, the approach of state-based representation is inherently parallel. Where bias may have existed in the serial equivalent due to competition, a conflict resolution approach is required [24]. In FLAMEGPU2, this typically requires agents to use functions to bid for resources with bias resolved in the selection process by using some form of agent prioritisation (which in most cases is a randomly assigned priority value). The state-based description of a model requires a collection of agents, each having a defined set of variables, states and functions. These functions enable agents to transition between states by updating their memory through predefined mathematical operations. Within FLAMEGPU2, the order of execution of agent functions can be specified explicitly using layers. Note that functions within the same layer may execute in parallel, and therefore have strict limitations to avoid race conditions (situations where concurrent processes lead to unpredictable outcomes, due to shared resource conflicts). Additionally, inter-agent communication is facilitated by global message lists, which can be written to, and read from, by agents. These message lists provide an abstraction for indirect agent communication. Furthermore, FLAMEGPU offers the possibility to define and represent environment variables (or properties) that do not belong to any specific agent instance. FLAMEGPU2's macro environment properties serve as an alternative to messages, enabling lightweight indirect communication between agents through globally accessible variables that can be atomically manipulated (e.g. individual cells consuming nutrients). It should be noted that the order of atomic operations is nondeterministic and determined by the GPU's execution model, which may affect reproducibility of results.

Any model implemented within the FLAMEGPU2 framework must follow the aforementioned rules and therefore adhere to a general layout. Fig. 1 shows the present model layout which can be considered as a standard template, with the exception of the function dependency or layer structure, which constitutes the core of the model as it determines function priority and call order. The type and initial number of agents must be configured prior to the simulation. Their starting positions and states can be defined both programmatically or loaded from a pre-existing file. For the rest of the simulation, a first block of layers must be defined to retrieve all agent locations. Notably, all agents can spatially move in 3D but the ECM agents are additionally constrained to a regular grid (Fig. 2). This not only influences the mechanical behaviour, as explained later, but has an important impact on computation speed as it determines the type of messages that may be used. In a second block, we compute the mechanical interactions between the different

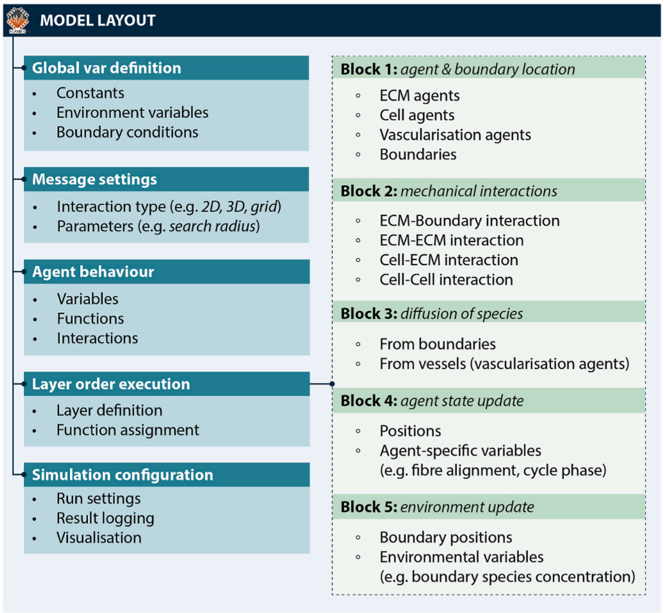

Fig. 1. Model layout. The model is implemented upon FLAMEGPU2 following a basic setup that can be fully customised. Firstly, global variables affecting the environment and boundary conditions must be defined. Then, message settings such as the type of interaction between agents as well as their basic parameters (e.g. search radius) must be established, followed by the definition of each agent's behaviour and the functions to be executed in each layer. This is the core of the model as it determines function priority and call order (i.e. functions in layer 1 will be called before those in layer 2, but they can be called in any order within the layer). In this sense, the model defines a first block (1) of layers to retrieve all agents' positions, a second block (2) to compute the mechanical interactions between different types of agents, a third block (3) to compute (if present) the diffusion of species and two final blocks (4,5) to update the states of both the agents (e.g. cell migration, cell division, ECM deformation, fibre alignment, etc.) and the environment (e.g. boundary positions or species concentration). Finally, a series of simulation settings must be configured depending on user needs such as parallelisation, result logging or live-visualisation.

kinds of agents present in the simulation. A third block computes (if present) the diffusion of species, from the domain boundaries or from the vascularisation agents, through the ECM grid. In the last two blocks, we update the states of both the agents (e.g. cell migration, cell division, ECM deformation, fibre alignment, etc.) and the environment (e.g. boundary positions or species concentration).

In the following sub-sections, we describe in detail the mechanisms or interactions defined within the aforementioned layers of the model. Importantly, once the ABM layout is set, adding or modifying agent behaviour is remarkably straightforward, providing an excellent tool to test and study a wide variety of emerging phenomena.

2.2. Model assumptions

This section, together with Fig. 2, aims to provide a broad vision of the model hypotheses and simplifications before expanding the details of each agent and their interactions.

- The ECM is conceptualised as a 3D regular grid of ECM agents confined within specified domain boundaries (a virtual prism).
- ECM agents are modelled as point masses, which simplifies their interaction by focusing only on translational motion and ignoring any volumetric properties or rotational dynamics. This assumption streamlines the calculation of forces and motion updates.
- The viscoelastic properties of the ECM are modelled through a spring-damper system connecting every pair of adjacent ECM agents in a Moore neighbourhood.
- The elastic constant for each ECM agent depends on various local factors such as collagen concentration, matrix degradation, fibre orientation, and deformation.
- Parameters like elastic or damper constants are normalised by mass, simplifying the computation and assuming uniform density or mass distribution across ECM agents, which may not capture variations in biological tissue properties. This can, however, be alleviated by using custom factors modifying properties of individual ECM agents (e.g. pre-loading an heterogenous spatial distribution of ECM constants).
- Force transmission between ECM agents is affected by the directionality of fibres, with parallel fibres transmitting force more effectively than perpendicular ones.
- Fibre direction is dynamically updated based on the direction of maximum stretching, reflecting real-time reorientation and reorganisation in response to mechanical loads.
- Cells are modelled as distinct types of agents and unlike ECM agents, are not confined to a fixed grid.
- Cell agents interact with ECM agents and other cell agents via constrained search radii, creating localised effects.
- A repulsive force profile is introduced between cell and ECM agents to prevent physical overlap.
- Mechanical interactions between cells and ECM are modelled using a spring-damper system, taking into account both the fibres and the cell body orientation.
- The orientation of cell agents is updated over time to align with the direction of maximum stretching, reflecting adaptive responses to mechanical stimuli.
- Cell agents' migration is modulated by external forces, orientation and random vectors.
- Cell cycle is simplified to phases G1, S, G2 and M. Duration and behaviour of cells in each phase is not defined by default.
- The default behaviour of cell agents includes linear growth until division in the M phase.
- The timing of cell division within the M phase is assumed to follow a uniform distribution, modified by the time elapsed to ensure that division occurs exactly once during this phase.
- Different diffusing species are modelled as non-interacting by default, which simplifies the complexity of biological interactions. Should the simulation of competitive, or should synergistic diffusion effects among multiple species be required, these interactions must be explicitly programmed by the researcher.
- Vascularisation agents are included to represent blood vessels or specific source points. These agents act as constant sources of species concentration, diffusing to their surroundings as per the reaction-diffusion dynamics. This assumption simplifies the role of vascularisation but effectively integrates it into the broader diffusion model.
- The mobility of vascularisation agents is constrained by their attachment to the nearest ECM agent, implying that any deformation in the ECM directly translates to the vascularisation.

2.3. Domain boundaries

Before further describing the main agent interactions, it is necessary to define the simulation domain. This domain is a virtual prism of any size (a cube by default), whose limits are naturally defined by 6 faces or planes. Different conditions can be individually set in each of these faces, including concentration of multiple species and different mechanical behaviours. These affect each agent type differently, as will be described in the relevant section. In general, these planes can act as: i) sticky surfaces: once an agent gets close enough, it gets stuck for the rest of the simulation; ii) sliding surfaces: impeding an agent passing in perpendicular directions; iii) visco-elastic surfaces: using the ECM-ECM interaction model; iv) periodic surfaces: agents migrating through them

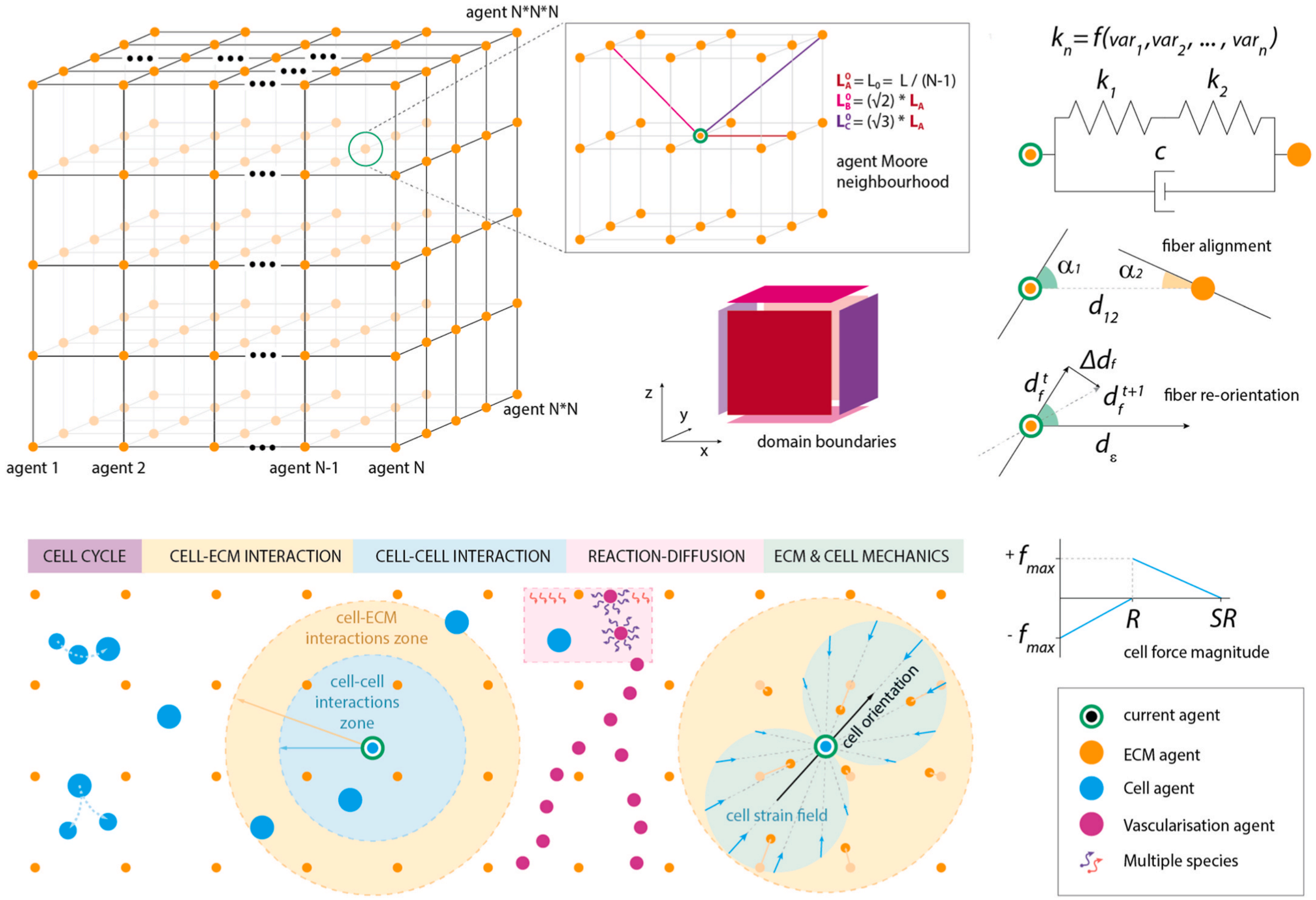

Fig. 2. Visual description of the ABM model. Top row, left, shows the lattice grid representing the ECM. Each of the ECM agents interacts with its Moore neighbourhood. Thus, the equilibrium length of the spring-damper system joining the agents depends on the relative position of the agents. In turn, the ECM spring constant may depend on other variables such as collagen concentration, degradation or relative fibre alignment (eq. (1)). The simulation domain is surrounded by boundary planes which can be configured to interact differently with agents (sticky, periodic, and sliding) and subject to motion, both external (linear, oscillatory, parallel/perpendicular) and due to agents' action (elastic). Bottom row shows other agents that may be included in the simulations, such as cells or vascularisation agents. The model includes basic functionality to account for cell cycling (cell growth and division), as well as cell-ECM (yellow shade) and cell-cell interactions (blue shade), mechanical or otherwise. Cell forces are aligned with their orientation, therefore creating heterogeneous strain fields (green shade). Cells tend to cluster if within a cell-cell interaction zone. Overlap is avoided by defining a repulsive force profile when the distance between cell agents is less than the sum of their radii (eq. (7)). Diffusion of multiple species from boundaries and/or vascularisation agents is computed following a forward-Euler scheme (eq. (14)).

re-enter the domain from the opposite plane; and v) moving surfaces: whose position over time is defined by a function (by default, linear and oscillatory movements are included). The combination of these boundary conditions provides a very flexible environment for simulating a wide variety of experiments, from traction-compression assays to multiscale models within a much bigger domain.

2.4. ECM-ECM interaction

The ECM is represented by a 3D regular grid of ECM agents which is surrounded by the domain boundaries (Fig. 2 top left). The viscoelastic mechanical behaviour of the ECM is modelled using a spring-damper system that connects every pair of ECM agents. This system comprises two springs in series, representing the elastic behaviour of the matrix, in parallel with a damper that accounts for the viscosity of the media. Importantly, every agent interacts only with its immediate Moore neighbourhood. In other words, every time an ECM-ECM interaction is evaluated, the current agent will receive messages from its 26 neighbours. The elastic constant of each agent ($k_{ECM,i}$) may depend on different factors such as collagen concentration ($[C_{col}]$), matrix degradation (md, ranging from 0 to 1), fibre orientation (α) and deformation (ε), which is reflected on eq. (1):

$$k_{ECM,i}(\varepsilon,\alpha,[C_{col}],md) = (1-md)\frac{[C_{col}]}{[C^0_{col}]}\cos(\alpha_i)\,k^0_{ECM}\begin{cases} e^{\frac{\varepsilon}{b_0}} & \varepsilon \le 0 \\ 1 & 0 < \varepsilon \le \varepsilon_s \\ e^{\frac{(\varepsilon-\varepsilon_s)}{s_0}} & \varepsilon_s < \varepsilon \end{cases} \quad \text{eq. 1}$$

The collagen concentration is defined relative to a base value (C^0_{col}) in order to facilitate parameter calibration in different conditions (i.e. model parameters can be adjusted for a specific gel and used as a reference for different tests). Strain-stiffening is a well-known phenomenon in hydrogels [25,26], involving a linear elastic regime followed by an exponential increase in stiffness as the fibres are subjected to higher levels of stretching. Following the equation proposed by Ref. [27] the mechanical behaviour of the ECM agents is assumed to display three distinct regimes: buckling, straightening, and stretching. The elastic constant is therefore multiplied by a function characterised using only four parameters: (i) a buckling coefficient (b_0), which accounts for an exponential reduction in fibre stiffness during compression; (ii) linear stiffness (k^0_{ECM}); (iii) critical strain (ε_s) at which strain stiffening commences; and (iv) an exponential strain stiffening coefficient (s_0).

Each of the ECM agents has assigned a fibre direction to further represent how well the force can be spatially transmitted. This unitary direction (d_f) is projected ($cos\ (\alpha)$) on the direction joining every pair of agents (d_{12}) affecting each of the $k_{ECM,i}$ (*eq. (1)* and Fig. 2 top right). Thus, two agents with highly parallel oriented fibres will transmit force more effectively than two with perpendicular ones. Additionally, the fibre direction of each agent is updated at a given rate (δ) towards the direction of maximum stretching (d_ε) within every ABM iteration (Fig. 2 top right), reflecting the fibre-reorientation and reorganisation depending on loads (see Supp.Movie1). This is described in eq. (2):

$$\Delta d_f = \delta\left(d_f^t \times \left(d_\varepsilon \times d_f^t\right)\right)\Delta t \qquad \text{eq. 2}$$

Supplementary video related to this article can be found at https://doi.org/10.1016/j.compbiomed.2024.108831

The force acting between two ECM agents separated by a distance (d) can be therefore calculated as follows:

$$\left|\overrightarrow{f}_{ECM-ECM}\right| = \left(d - L_*^0\right)\left(\frac{k_{ECM_1}k_{ECM_2}}{k_{ECM_1} + k_{ECM_2}}\right) + c_{ECM}\left|\overrightarrow{v}_{rel}\right| \qquad \text{eq. 3}$$

where L_*^0 is the equilibrium length that depends on the relative position of the agents within the virtual grid (regardless their current 3D position) and the number of agents – domain size relationship (Fig. 2 top middle), c_{ECM} is the ECM damping constant (agent independent) and $\left|\overrightarrow{v}_{rel}\right|$ the relative speed between the interacting agents projected on the direction joining them (d_{12}).

2.5. Cell-ECM interaction

Cells are represented by a different type of agent whose communication with other agents does not rely on a fixed grid, but on different search radii (SR) in 3D (one for each type of agent as shown in Fig. 2 bottom left). In other words, cells only interact with those ECM agents within a certain range ($SR^{cell-ECM}$), exerting forces depending on the distance between them (d) and the cell agents' orientation (d_{or}), which create heterogeneous strain fields around them (see in Fig. 2 bottom right). The mechanical model is similar to the one described for the ECM (spring-damper system with its corresponding constants k_{cell} and c_{cell}), with the difference that a repulsive profile is added when the distance is smaller than the cell agent's radius (R_{cell}) to avoid overlap. Therefore, cell-ECM forces are computed following the equation:

$$\left|\overrightarrow{f}_{cell-ECM}\right| = \begin{cases} \left(SR^{cell-ECM} - d\right)\left(\dfrac{k_{cell}k_{ECM}}{k_{cell} + k_{ECM}}\right) + c_{cell}\left|\overrightarrow{v}_{rel}\right| & d \geq R_{cell} \\ \dfrac{f_{max}^{cell-ECM}}{R_{cell}} d - f_{max}^{cell-ECM} + c_{cell}\left|\overrightarrow{v}_{rel}\right| & d < R_{cell} \end{cases} \qquad \text{eq. 4}$$

where $f_{max}^{cell-ECM}$ is:

$$f_{max}^{cell-ECM} = \left(SR^{cell-ECM} - R_{cell}\right)\left(\frac{k_{cell}k_{ECM}}{k_{cell} + k_{ECM}}\right) \qquad \text{eq. 5}$$

Similar to the ECM agents, the orientation of cell agents aligns over time with the direction of maximum stretching with a given rate (δ) (that may be different between cells and ECM agents):

$$\Delta d_{or} = \delta\left(d_{or}^t \times \left(d_\varepsilon \times d_{or}^t\right)\right)\Delta t \qquad \text{eq. 6}$$

2.6. Cell-cell interaction

Cell agents interact with other cell agents within a certain range ($SR^{cell-cell}$) and attract each other linearly with distance (d) until they are in contact ($d = \left(R_{cell_1} + R_{cell_2}\right)$), when the force profile switches to repulsion to avoid overlap (eq. (7)) (see Fig. 2 bottom). Since, unlike the ECM agents, cell agents' movement is unrestricted, they will tend to form clusters according to the following equations. Nevertheless, the equilibrium is broken by adding random movements, preferentially in their direction of orientation (d_{or}) (see *ECM deformation and cell migration* section).

$$\left|\overrightarrow{f}_{cell-cell}\right| = \begin{cases} \left(SR^{cell-cell} - d\right)\left(\dfrac{k_{cell_1}k_{cell_2}}{k_{cell_1} + k_{cell_2}}\right) + c_{cell}\left|\overrightarrow{v}_{rel}\right| & d \geq \left(R_{cell_1} + R_{cell_2}\right) \\ \dfrac{f_{max}^{cell-cell}}{\left(R_{cell_1} + R_{cell_2}\right)} d - f_{max}^{cell-cell} + c_{cell}\left|\overrightarrow{v}_{rel}\right| & d < \left(R_{cell_1} + R_{cell_2}\right) \end{cases} \qquad \text{eq. 7}$$

where $f_{max}^{cell-cell}$ is:

$$f_{max}^{cell-cell} = \left(SR^{cell-cell} - \left(R_{cell_1} + R_{cell_2}\right)\right)\left(\frac{k_{cell_1}k_{cell_2}}{k_{cell_1} + k_{cell_2}}\right) \qquad \text{eq. 8}$$

2.7. Cell cycle

The cell cycle is a complex process due to its intricate regulation and coordination of numerous molecular events. The presented implementation provides minimal functionality that can be easily expanded for specific user needs. Importantly, FLAMEGPU automatically integrates the possibility of adding/removing agents (agent birth/death) on the fly, and it is up to the user to define the conditions under which such events take place. In the current model, cell agents include a variable defining their current phase in the cell cycle (G1, S, G2, M) [28] as well as an internal clock that determines when they switch to a different phase. Users must define the duration of each phase according to the cell type, and the behaviour of the cell in each of the states. As stated before, including specific behaviours for different types of cells is out of the scope of the present work. However, we include, by default, linear cell growth and cell division at some point in M. When a cell divides, two daughter cells, of half the radius, appear along the direction of the mother's orientation (d_{or}) (the original mother cell agent ceases to exist).

Once a cell agent enters the M phase, we compute the probability of its division by drawing from a conditional uniform distribution, which ensures that the probability is independent of the simulation time step (Δt) and that the cell divides with probability 1 during this phase. To this end, we use a dynamic random procedure considering for each time step (t) that the probability of cell division is expressed by eq. (9), a uniform distribution in $\left\{1, \ldots, M_{length}\right\}$ (where M_{length} is the duration of the M phase) conditioned to no occurrence in $\{1, \ldots, t-1\}$ (i.e. truncated uniform). In other words, we compute the conditional probability of cell division occurring at current time ($P\left(D_t^c\right)$) provided that division has not happened so far in M phase (see Supp.Movie2).

$$P\left(D_t^c\right) = P\left(D_t^c \mid \bigcap_{i=1}^{t-1} D_i^c\right) = \frac{\frac{1}{M_{length}}}{\frac{\left(M_{length} - t + 1\right)}{M_{length}}} \qquad \text{eq. 9}$$

Supplementary video related to this article can be found at https://doi.org/10.1016/j.compbiomed.2024.108831

2.8. ECM deformation and cell migration

ECM agents are modelled as point masses subjected to inertia, but lack volume and are not subject to rotational forces. Hence, all the forces acting upon them (i.e. other ECM agents, cell agents and/or boundaries) are utilised to update each ECM agent's position at the end of every ABM step following the equation of motion (eq. (10) and eq. (11)). Parameters involved in computing the forces, such as elastic or damper constants, are normalised by mass for simplicity:

$$\overrightarrow{v}^{t}_{ECM} = \overrightarrow{v}^{t-1}_{ECM} + \left(\overrightarrow{f}_{boundary-ECM} + \overrightarrow{f}_{cell-ECM} + \overrightarrow{f}_{ECM-ECM}\right)\Delta t \qquad \text{eq. 10}$$

$$\overrightarrow{x}^{t+1}_{ECM} = \overrightarrow{x}^{t}_{ECM} + \overrightarrow{v}^{t}_{ECM}\Delta t \qquad \text{eq. 11}$$

Cell migration is computed in a similar fashion but includes two extra velocity vectors to account for directed migration ($\overrightarrow{v}_{prot}$) aligned with cell agent's orientations (d_{or}) and randomness ($\overrightarrow{v}_{rand}$). The size of such vectors can be tuned to achieve different migration patterns (see Supp. Movie3).

$$\overrightarrow{v}^{t}_{cell} = \overrightarrow{v}^{t-1}_{cell} + \overrightarrow{v}^{t}_{prot} + \overrightarrow{v}^{t}_{rand} + \overrightarrow{f}_{cell-cell}\Delta t \qquad \text{eq. 12}$$

$$\overrightarrow{x}^{t+1}_{cell} = \overrightarrow{x}^{t}_{cell} + \overrightarrow{v}^{t}_{cell}\Delta t \qquad \text{eq. 13}$$

Supplementary video related to this article can be found at https://doi.org/10.1016/j.compbiomed.2024.108831

2.9. Transport of nutrients

By taking advantage of the 3D grid representing the ECM, transport of species is included in the model by solving the reaction-diffusion equation at each point of the grid following a forward Euler scheme:

$$\frac{\partial C}{\partial t} = \frac{C^{t+1}_{i,j,k} - C^{t}_{i,j,k}}{\Delta t} = D\left(\left(\frac{C^{t}_{i+1,j,k} - 2C^{t}_{i,j,k} + C^{t}_{i-1,j,k}}{\Delta x^2}\right) + \left(\frac{C^{t}_{i,j+1,k} - 2C^{t}_{i,j,k} + C^{t}_{i,j-1,k}}{\Delta y^2}\right) + \left(\frac{C^{t}_{i,j,k+1} - 2C^{t}_{i,j,k} + C^{t}_{i,j,k-1}}{\Delta z^2}\right) + R^{t}_{i,j,k}\right) \qquad \text{eq. 14}$$

where $\Delta x, \Delta y, \Delta z$ are the distances between neighbour ECM agents, D the diffusion coefficient, C the concentration and R the source/sink term at each point. It is worth noting that this equation is solved for any number of species, and that those species can diffuse from either the boundaries, the vascularisation agents (see *Vascularisation* section) or both (see Supp.Movie4). In the present model, different species don't interact, but that functionality could be easily implemented by the user, tailored to their needs.

Supplementary video related to this article can be found at https://doi.org/10.1016/j.compbiomed.2024.108831

2.10. Vascularisation

Vascularisation agents are included to represent vessels or specific source points to account for more complex systems that cannot simplify the diffusion of species conditions to the six flat domain surfaces. Hence, these agents act as a constant source of species concentration, which diffuses to their surroundings following eq. (14). Importantly, for simplicity, vascularisation agents' movements are tied to the closest ECM agent. Hence, ECM deformations translate to the vascularisation in an affine way (see Supp.Movie5). This feature, together with the possibility of combining the diffusion of multiple species, makes our platform very attractive for multiscale systems that require a very detailed simulation of the microenvironment, but also macroscopic variables such as nutrient availability or mechanical inputs (see Results - Example 4).

Supplementary video related to this article can be found at https://doi.org/10.1016/j.compbiomed.2024.108831

3. Results

In this section, we show four different example applications of our model to demonstrate its potential and versatility. Our intention is not to validate specific experiments with particular parameters, but rather to demonstrate how the model can be adapted to a wide range of simulation conditions. In the first simulation study, we perform a rheology assay with only ECM agents present, capturing the strain-stiffening behaviour of hydrogels. In the second, we simulate a large domain (in the order of hundreds of microns) representing the hydrogel within a microfluidic device where cell agents migrate and interact with ECM agents. In the third, we simulate the formation of spheroids within a hydrogel and their interactions with the ECM. Finally, we show the potential of combining diffusion of multiple species from different sources (i.e. vessels and boundaries). It is worth noting that matrix degradation mechanisms have not been explicitly implemented for these example cases, so its parameter value (*md*) is set to 0. Additionally, we present a computational efficiency and scalability study of our model to highlight its speed and potential use for large scale problems.

3.1. Example application 1: simulating strain-stress curve of hydrogels of different concentrations

As previously stated, collagen hydrogels, commonly used for cell culture experiments, exhibit strain-stiffening behaviour due to the structural arrangement of the fibres within the gel matrix. In Ref. [29], this behaviour was experimentally measured in hydrogels with three different collagen concentrations (Fig. 1). To capture this same strain-stress relationship, we simulated a shear-stress assay, with a small grid of 10 × 10 × 10 ECM agents with initial random fibre orientation, by imposing a horizontal displacement of the top boundary face of the domain, while constraining the bottom one. Then, we computed the shear modulus (G), following:

$$G = \frac{F/A}{\Delta x/l} \qquad \text{eq. 15}$$

where F was the force measured in the direction parallel to the movement of the top face of area A, and $\Delta x/l$ the deformation in the stretching direction (See Supp.Fig1 and Supp. Movie1). We then adjusted the parameters used in eq (1), namely: the linear stiffness (k^0_{ECM}), the buckling coefficient (b_0), the critical strain (ε_s) and the strain-stiffening coefficient (s_0). Matrix degradation (*md*) was set to 0 in these simulations. It is worth noting that, in this scenario, the magnitude of the measured forces is proportional to the number of agents and the value of k^0_{ECM}, which is in turn multiplied by C_{col}/C^0_{col} to account for the collagen concentration. C^0_{col} was set to 2.5 mg/mL and C_{col} was set accordingly for the three different cases (2.5, 4 and 6 mg/mL respectively). A sensitivity analysis was performed to select the final values of the remaining parameters involved (see Supp. Fig. 2).

For all concentrations, the experimental and computational curves follow a similar trend, showcasing an initial region of lower modulus followed by a steep ascent that depends on collagen concentration. The slight deviations between the experimental and computational curves may be attributed to the simplifications inherent in the computational model or to experimental variability, but this study shows the potential of the model to be tailored to specific research needs. In fact, we managed to fit the three curves by only varying C_{col}, as we are simulating the same material (collagen) at different concentrations, but finer adjustments could be achieved by varying the remaining parameters independently.

3.2. Example application 2: simulating cell migration within hydrogels

Microfluidic devices, in combination with hydrogels, are widely used in the field of biomedical research, serving as powerful tools for precise condition manipulation and control in a variety of applications, including cell culture, drug screening, and the study of cellular responses to different stimuli [30]. For this reason, we simulated cell migration within a hydrogel using common experimental setups in terms of gel size and experiment duration [12,31]. In particular, we simulated the migration of 1000 cell agents within a rectangular domain of 2000 ×

1300 × 300 μm. This domain, representing collagen, was formed by a regular grid of ECM agents placed every 20 μm, resulting in a total of 101 × 66 × 16 = 106,656 ECM agents. We ran 5000 simulation steps, with each step representing 1.5 min, totalling approximately 5 days.

Fig. 4A displays the entirety of the domain at an arbitrary time point, demonstrating how the ECM grid has been deformed by the actions of cell agents. Concurrently, Fig. 4B provides a detailed view of the fibre orientation, captured at the same time point. We tuned $\left|\vec{v}_{prot}\right|$ and $\left|\vec{v}_{rand}\right|$ ratio ($\left|\vec{v}_{prot}\right| \approx f * \left|\vec{v}_{rand}\right|$) within the cell agents to illustrate two extreme cases of movement: purely random motion ($f = 0$) and highly directed motion ($f = 2$).This parameter (f) has the strongest impact on the migration pattern (i.e. Brownian, persistent) and may be used as a starting point to adjust the simulation depending on cell types or other conditions, achieving the desired directionality ratios (see Sup. Material and Supp.Fig.3). Further refinement can be achieved by tweaking cell-cell and cell-ECM interactions.

Fig. 4C shows the relative cell trajectories at the end of the simulation. As would be expected, the spread is much higher in the persistent case, as cells tend to continue migrating in their orientation direction (d_{or}). Mean squared displacement (MSD) of the tracked trajectories was obtained and used to fit a power law ($MSD(t) = \gamma t^{\alpha}$) to determine the kind of motion ($\alpha < 1$ for confined motion, $\alpha = 1$ for Brownian or purely diffusive motion and $\alpha > 1$ for directed or persistent motion). Fig. 4D compares the MSD of both migratory cases confirming the theory with the α values obtained (0.96 ± 0.18 and 1.87 ± 0.13 for Brownian and persistent cases respectively). Similarly, along with the type of migration, cell velocity can be modulated by setting a reference value. The combination of both parameters (directionality and expected speed) results in a range of mean (instantaneous) and effective (measured from the starting to the endpoint) speeds (v_{mean} and v_{eff}, respectively) as depicted in Fig. 4E. As an example, the speeds in these simulations were varied within the physical range expected for fibroblasts migrating in collagen [31].

3.3. Example application 3: spheroid formation and fibre reorientation

Spheroids are 3D aggregates of cells that closely mimic the architecture and function of tissue in vivo and are integral to the understanding of cellular organisation, differentiation, and proliferation [32]. The formation of these spheroids and their interaction with the ECM's fibrous network is of particular interest in tissue engineering and cancer research, as it provides insight into the spatial dynamics of cell growth and the biomechanical properties of the ECM. Our platform may be used to replicate this complex interplay by employing more advanced functionalities (i.e. cell cycle) to capture the emergent properties of spheroid structuring and ECM adaptation. To showcase this, we employed a cubical domain of 100 × 100 × 100 μm with 8 cells randomly placed in the central region, and we allowed them to grow for 200 h.

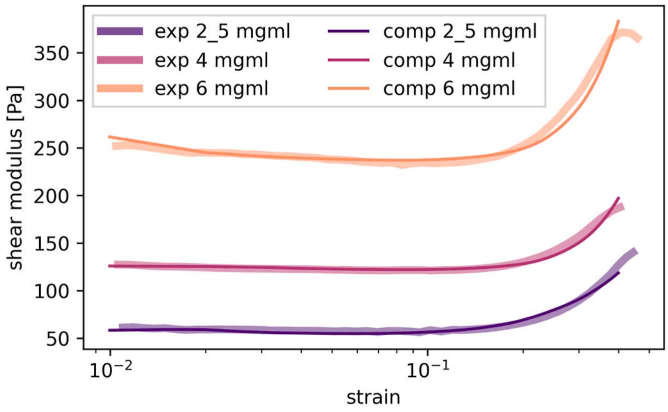

Fig. 3. Graphical comparison of the shear modulus (Pa) as a function of applied strain for collagen at varying concentrations (2.5, 4, and 6 mg/mL), obtained through both experimental ('exp') measurement from Ref. [29] and computational simulation ('comp') from this work.

Fig. 5 illustrates the temporal progression of collagen fibre alignment in relation to the forming spheroid. As the simulation progresses, there is a marked alignment of fibres orienting towards the spheroid, culminating in near-perfect alignment by the end of the simulation period, as depicted in the right panel of Fig. 5. This phenomenon is consistent with empirical findings that demonstrate the ability of tumour spheroids to reorient surrounding collagen fibres, extending radially up to five times the spheroid's radius - well beyond the immediate zone of tumour expansion and cellular migration [33]. While the specific dynamics of fibre realignment and the intricate interactions among cells, or between cells and the ECM fibres, were not the focus of this simulation, given its simplified representation of a highly complex process, the results nonetheless underscore the versatility of our computational framework in modelling and understanding such biological phenomena.

3.4. Example application 4: simulating diffusion of multiple species within a vascularised deformable domain

In this fourth example, we present a simple simulation that captures the diffusion of multiple species within a (randomly) vascularised and deformable domain of 1 mm^3. This aims to show the potential of our framework to compute highly detailed simulations that could constitute a single component within a larger, multiscale problem. A similar approach has been used previously to simulate the evolution of neuroblastoma tumour [34], but including only cell agents at the micro-level. These agents were modelled in deep detail (including signalling pathways, mutations etc.), but the role of the ECM (i.e. a drag term affecting cell motion) and the available nutrients (i.e. constant concentration of oxygen) were greatly simplified. Including both ECM agents and vascularisation allows for a more nuanced exploration of cellular dynamics within a tumour environment, providing a robust platform for understanding the interplay between cells, ECM architecture, and nutrient diffusion facilitated by vascularisation. While the detailed characterisation of these behaviours falls beyond the scope of the present work, we employ an illustrative example as a proof of concept. In this example, Species A diffuses from the vessels to the ECM, and Species B diffuses from one of the boundaries.

Fig. 6 shows the concentration of species A (Fig. 6, first row) and B (Fig. 6, second row) over time (t = 0,1 and 5 min). It is worth noting that, in these simulations, there are no cell agents and the species do not interfere with one another in any way. Therefore, their diffusion profiles are unaltered by the presence of other species or agents. The graph (Fig. 6, right) compares the concentration profiles of species A and B over time at two distinct points within the domain, to visualise more easily the spatial heterogeneity that can be achieved by using this approach. In fact, although only two species were simulated in this example, there is no limit in the number of species diffusing simultaneously (with their own particular dynamics) and/or interacting with the ECM or cell agents Furthermore, our framework facilitates the investigation of more intricate scenarios by accommodating domain deformability (i.e. displacements arising from the macroscale within a multiscale problem, or deformations induced by the cell agents that exert forces on the boundaries), all while integrating with the diffusion calculations which are linked to the ECM grid, in turn linked to the vascularisation agents (see Supp.Movie 5).

3.5. Computational efficiency and scalability

To evaluate the computational efficiency and scalability of our model, we performed simulations with an increasing number of agents on two different devices: a laptop and a High-Performance Computing

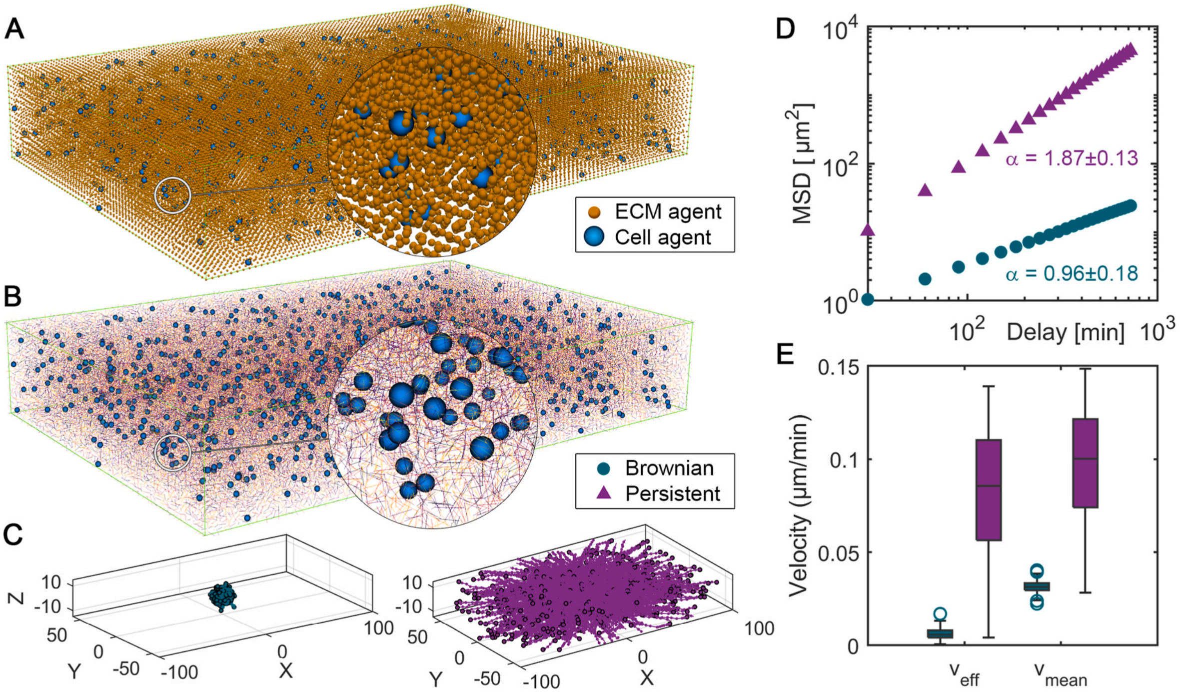

Fig. 4. A) Depicts the overall domain at a selected time point, illustrating the deformation of the ECM grid as a result of interaction with cell agents (blue), with ECM agents (orange). B) Provides a focused view on a cluster of cell agents within the ECM, highlighting the proximity and potential interactions between cells and ECM fibres. C) Shows trajectories of cell agents with two types of migration patterns: Brownian (random) and persistent (directed), represented in teal and purple respectively. D) Mean squared displacement (MSD) against time delay for both migration types, with the slope of the line indicating the nature of each movement pattern - Brownian ($\alpha \approx 1$) or superdiffusive ($\alpha > 1$). E) Comparison of effective (v_{eff}) and mean (v_{mean}) velocities of cell agents, illustrating the range of observed speeds, found significantly different (***), within the simulated ECM environment. ANOVA tests were performed among the cell migration data sets, and statistical significance was assumed when $p < 0.001$ (***), $p < 0.01$ (**) or $p < 0.05$ (*).

(HPC) cluster (University of Sheffield's Stanage HPC system). The technical specifications of these devices were as follows:

- Laptop: Intel i7-10710U@1.10GHz (4.7 GHz Boost), NVIDIA GeForce GTX 1650
- HPC Cluster: AMD EPYC 7413@2.65GHz (3.6 GHz Boost), NVIDIA H100 PCIe

We carried out two types of studies:

First, we simulated a cubical domain with a fixed size of 1 mm^3 divided in a regular grid of N_{ECM} x N_{ECM} x N_{ECM} ECM agents where N_{cell} cell agents were randomly placed. We varied these quantities up to a maximum combined total of 2 million agents. Specifically, we tested combinations of:

N_{cell} = 1000; 10,000; 100,000; 1,000,000

N_{ECM} = 10; 20; 30; 40; 50; 60; 70; 80; 90; 100

This range of agent numbers was chosen to cover scenarios from in-vitro to in-vivo studies, allowing us to assess the performance and scalability of our model across a wide range of biological complexities.

Secondly, we simulated a cubical domain with a fixed density of agents, and we tested the model efficiency by multiplying iteratively 8-fold the volume of the domain, starting with 1 mm^3 up to 512 mm^3. The initial total agent numbers for this test were N_{cell} = 10,000 and N_{ECM} = 30 (a combined total of 37,000 agents), which increased up to N_{cell} = 5,120,000 and N_{ECM} = 240 (a combined total of 18,940,000 agents) in the largest case. This secondary set of simulations aims to demonstrate that the code's efficiency is more dependent on model parameters than on the number of agents.

In all the cases, diffusion of two different species from boundaries was present, and there were no vascularisation agents.

The duration of the simulations is influenced by both the initialisation phase and the number of simulated steps. Consequently, we focused on two primary parameters: (i) initialisation time, which encompasses the time required to initialise the FLAME GPU library and configure the simulation parameters, position and initialise all agents' states, functions, and variables; and (ii) average step time, defined as the time needed to execute a single model time step, averaged over 100 consecutive steps. Notably, this implementation of a Python model on FLAMEGPU2 has the drawback of higher initialisation times compared to a pure C++ model, due primarily to the use of run time compilation of the GPU agent functions. Additionally, any host (CPU) initialisation functions occurring within the Python code incur the overhead of Python being an interpreted language. However, once the model is running, the utilisation of GPUs on HPC systems results in significant speed improvements, achieving step times that can be orders of magnitude faster than on regular machines, especially for larger simulations.

A summary of the results is shown in Fig. 7, and a full listing of the recorded timings can be found in Supp. Table 1. For the first set of simulations (with varying agent density), the initialisation time for the H100 (Fig. 7A) ranged from 0.276s to 273.607s and increased gradually with the number of cell and ECM agents. In contrast, the GTX1650 (Fig. 7E) showed a wider range (from 0.358s to 526.322s), with a steeper increase as the number of agents increased. The average step time for the H100 (Fig. 7B–G) remained relatively low, ranging from 7.4e-4s to approximately 0.1511s, corresponding with the most demanding cases. Conversely, the GTX1650 (Fig. 7F and G) experienced much higher step times, from 0.0021 to about 5.124 s, indicating more significant performance degradation with larger agent numbers. In the same way, the total simulation time (Fig. 7C) for both GPUs increased exponentially with the number of agents, but the GTX1650 shows significantly longer times, particularly beyond 1×10^6 agents. In fact,

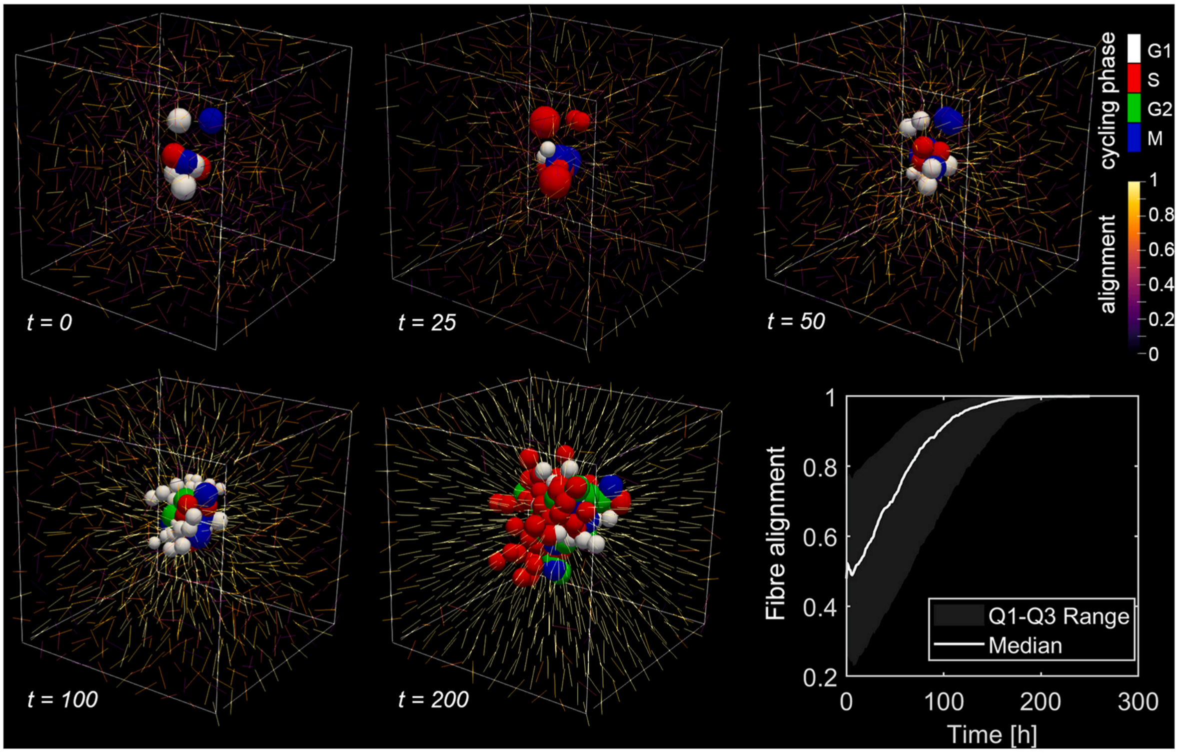

Fig. 5. Temporal Evolution of collagen fibre reorientation in response to spheroid formation. Panels depict the simulation at time points t = 0, t = 25, t = 50, t = 100, and t = 200 h, showing the initial state and subsequent stages of spheroid development and associated collagen fibre alignment. Initially, fibres are randomly oriented (t = 0). As the spheroid forms, a gradual reorientation of fibres towards the spheroid becomes evident (t = 100), leading to a pronounced alignment at t = 200, as seen in the rightmost panel. The colour gradient on this panels represents the degree of fibre alignment over time, from low (dark purple) to high (bright yellow). In addition, cell agents are coloured as a function of their cycle phase (1:G1, 2:S, 3:G2, 4:M). The accompanying graph quantifies the fibre alignment, with the shaded area indicating the interquartile range (Q1–Q3) and the solid line representing the median, reflecting the overall increase in alignment over time (see Supp.Movie2).

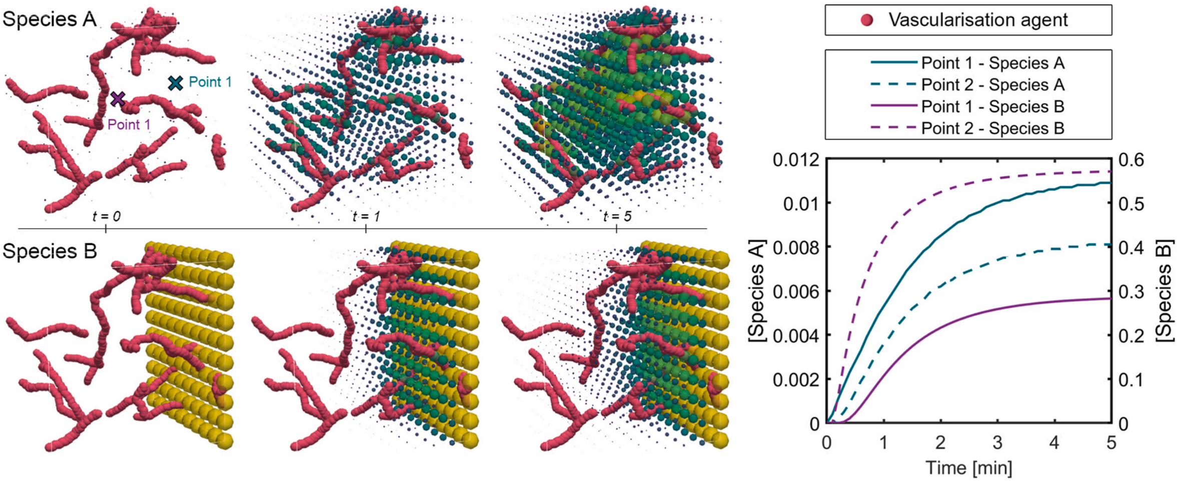

Fig. 6. Simulation of diffusion of multiple species within a vascularised domain. The top row (left) illustrates the diffusion of Species A at time points t = 0, t = 1, and t = 5 min, with the source being the vascularisation agents (pink). The bottom row (left) shows the diffusion of Species B, which source is one of the domain boundaries. The graph on the right quantitatively compares the concentration profiles of Species A and B over time at two distinct points within the domain: Point 1 (dotted lines) and Point 2 (solid lines), demonstrating the detailed spatial heterogeneity that can be achieved.

the slowest simulations, indicated by markers (Fig. 7C–G) highlighted with a coloured box, correspond to cases with $N_{CELL} = 1 \times 10^6$ and varying numbers of ECM agents. These simulations exhibit a significant increase in both total simulation time and average step time (over 7 times higher) compared to those with similar number of agents but composed by $N_{ECM}^3 = 1 \times 10^6$ and different numbers of cell agents. This difference, also illustrated by the predominantly vertical contour lines in Fig. 7B and F, arises because ECM agents utilise grid communication (each interacting with just their 26 neighbours), whereas cell agents employ 3D spatial communication, using a search radius to identify other agents for interaction. The method of agent communication is generally the largest factor impacting performance in FLAME GPU models [21].

The predominant dependence of computation time of cell agent

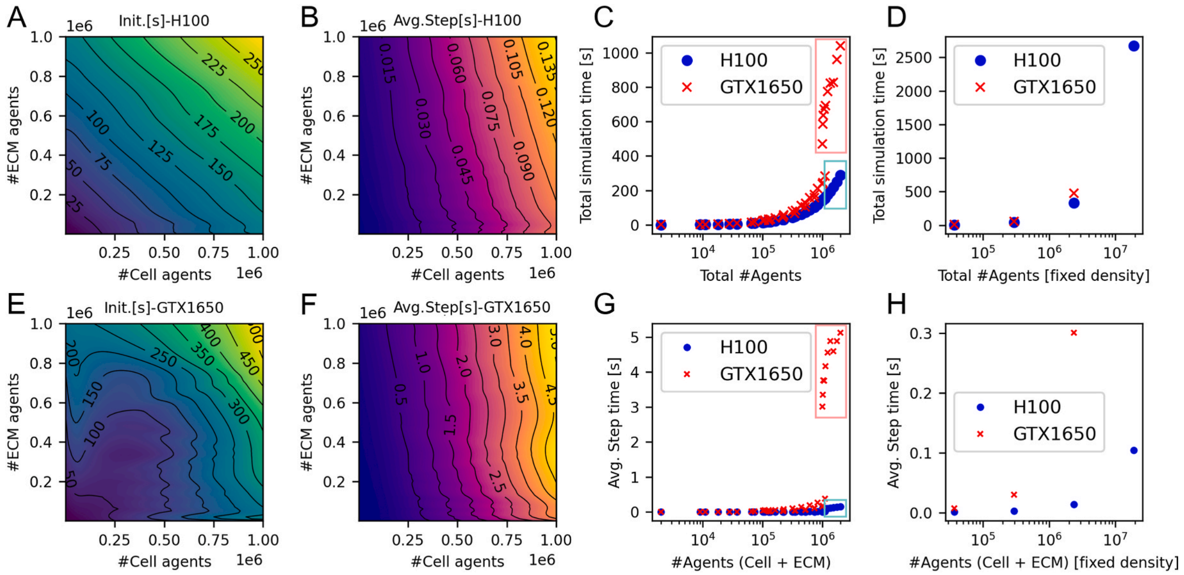

Fig. 7. Comparison of simulation timings between NVIDIA H100 PCle and GTX1650 GPUs. A, E) Initialisation time [s] of the model using the H100 PCle and GTX1650 GPUs respectively: time required to compile and configure the simulation parameters, position and initialise all agents' states, functions, and variables. B, F) Average step time [s] of the model using the H100 PCle and GTX1650 GPUs respectively: time needed to execute a single model time step, averaged over 100 consecutive steps. C) Total simulation time [s] vs. total number of agents, D) Total simulation time [s] vs. total number of agents in fixed density conditions G) Average step time [s] vs. total number of agents. H) Average step time [s] vs. total number of agents in fixed density conditions. In C) and F), coloured boxes highlight markers corresponding to cases with $N_{CELL} = 1e6$ and varying numbers of ECM agents.

number has two direct implications: i) since ECM density has less impact, it gives the researcher more freedom to choose the level of detail for the mechanical problem; and ii) the cell search radius is a critical parameter that depends on the cell type and/or the simulated problem. In these tests, its value was set to 50 μm, resulting in an exponential increase in the number of interactions for cases with 1×10^6 cells/mm^3 compared to those with 1×10^5 cells/mm^3 or less. To confirm this, we conducted a second series of simulations with a fixed agent density, increasing the total number of agents to nearly 19 million. Importantly, the density was chosen so that the search radius did not play a significant role. The biggest simulation was executed only on the HPC system, as the GTX1650 lacked memory capacity. As it can be seen in Fig. 7D, the total simulation time was similar for both GPUs, as over 90% of the total time corresponds to model initialisation (which is carried out mostly on CPU). It's important to note that for longer-running simulations, the impact of this initialisation time would be amortised, leading to a more significant performance advantage for the GPUs during the actual simulation process. Fig. 7H highlights how the H100 maintains an average step time significantly below that of the GTX1650. In fact, the average step time on H100 for a simulation of 18,94 million agents was 0.1 s, three times lower than the average step time of a simulation with 6,84 million agents on the GTX1650. In both cases, times grow exponentially with the number of agents, with no step changes as in the previous series, demonstrating the raw capabilities of the framework and its potential scalability to enormous problems.

A deeper analysis of the FLAMEGPU2 framework performance and scalability can be found in the original work [21].

4. Discussion

Computational simulations are emerging as essential tools in overcoming the limitations of traditional experimental imaging, facilitating the visualisation of cellular behaviours in three dimensions. These models complement experimental techniques, offering scalable, cost-effective means to hypothesise, predict, and guide experimental designs in studying complex biological processes. In this work, we explore the development and application of an innovative agent-based model platform designed for simulating cell microenvironments using the FLAMEGPU2 framework. This model demonstrates a foundational approach intended to be built upon, specifically tailored for modelling the complexities of cell microenvironments. It is worth noting that the examples presented in this work are intended to showcase the capabilities of the model rather than to validate specific experimental conditions. In fact, the examples involving spheroid formation and the integration of diffusion with vascularisation present complexities that would require specialised experimental data and additional modelling efforts that are beyond the scope of this manuscript.

The utilisation of FLAMEGPU2 offers significant computational advantages, enabling extremely rapid calculations that enhance the model's efficiency and applicability to a broad range of scenarios. In fact, one of the core strengths of this platform is its highly efficient GPU utilisation. FLAMEGPU2 has been meticulously crafted from the ground up with a focus on efficient agent communication and scalability. It utilises hand-written and finely-tuned CUDA code, distinguishing it from other approaches that rely on pragma-based standards such as OpenACC. Although OpenACC is known for its ease of use and portability, it often fails to deliver peak performance in complex applications due to redundant data transfers between host and GPU, as well as inefficient conflict resolution algorithms. Furthermore, FLAMEGPU2 offers extensive support for running ensembles across multiple GPUs and nodes, facilitating easy integration with high-performance computing (HPC) environments, which is crucial for scalability, as we have demonstrated in our performance study. Nevertheless, we have also shown that a very large number of agents can be effectively simulated on personal computers, highlighting the potential for conducting complex simulations in more accessible and cost-effective environments, thereby broadening the scope for research in this field.

Aside from the advantages of the underlying architecture, our model offers significant flexibility in incorporating any type of custom agents, including customisable templates for ECM, cell, and vascularisation agents, and managing intricate interactions between them. We also provide comprehensive control over domain boundary conditions, features that are typically restricted in other solutions. Basic events, such as agent birth and death, are seamlessly integrated within our messaging scheme, granting researchers complete autonomy over the conditions triggering them. Unlike alternatives like PhysiCell or Gell, which tend to oversimplify the mechanical behaviour of the ECM by reducing it to a drag term in the migration equation, our approach considers its viscoelastic properties. Moreover, we account for factors like local fibre orientation and matrix degradation. The mechanical equations in our model are not only customisable but can also be expanded by researchers as needed. However, it is important to note that this degree of flexibility and customisation adds a layer of complexity and requires additional effort from the researcher.

However, we believe that this flexibility is crucial for accurately replicating the dynamic and complex nature of biological systems, as illustrated by the examples provided in this work, ranging from simulating cell migration within microfluidic devices to simulating the tumour microenvironment. In fact, our model's adaptability might be particularly relevant for simulating phenomena where vascularisation plays a critical role. For example, vascular networks are crucial for tumour growth and progression, affecting the delivery of nutrients and the removal of waste. Our model's ability to accurately represent these processes enhances its value, not only helping understand the underlying biological dynamics, but providing a highly detailed microscale component within a broader macroscale analysis, such as a finite element (FE) simulation. This integration would allow for a dynamic exchange of information, where the macroscale FE model might provide overarching boundary conditions and global parameters to the microscale agent-based model. Conversely, insights from the microscale, regarding cellular behaviours, molecular diffusion, ECM degradation or deformation, would inform adjustments and updates at the macroscale level. This reciprocal flow of information would enhance the predictive accuracy and relevance of simulations across different spatial scales, offering a comprehensive understanding of complex biological systems. The applicability of such an agent-based model, developed using FLAMEGPU2, within a multiscale approach has been already tested by the authors in a previous study [34], showing promising results in personalised medicine.

While our framework marks a significant step forward, it presents some limitations and potential areas for improvement. Currently, FLAMEGPU2 lacks a robust user community and the comprehensive suite of default behaviours found in more established frameworks like PhysiCell. These frameworks offer a wide variety of pre-defined models for cell phenotype, diffusing substrates, signalling factors, and biologically driven sub-models for processes such as cell cycling, apoptosis, and necrosis, which are not inherently included in our model. These built-in functionalities significantly reduce the time required for setup and development, and ensure standardisation across studies. In fact, these models are often well-tested and validated, providing a reliable foundation that facilitates the comparison of results and enhances the reproducibility of research. Additionally, this ease of use makes advanced simulations more accessible, especially for those who may not have deep coding expertise or familiarity with complex modelling techniques. In contrast, our model, while not excessively complex, demands a higher level of technical proficiency and a deeper understanding of modelling principles, making it more challenging for unexperienced researchers. Finally, another constraint is the requirement for a CUDA compatible GPU, which may not be readily available in many research laboratories. This dependency could limit the model's accessibility and utility across the broader scientific community, potentially hindering its adoption and integration into existing research workflows.

5. Conclusions

Despite these limitations, our model represents a significant advancement in the simulation of cell microenvironments, providing a powerful tool for researchers to explore and understand the complex dynamics of cellular interactions. In fact, the potential for incorporating submodels into our framework opens avenues for modelling more complex scenarios that were not explored in this work. The FLAMEGPU2 submodel feature allows for the integration of nested models, offering a versatile tool for simulating iterative algorithms for conflict resolution problems, or complex interactions that require different temporal resolutions, such as the diffusion of species and cell migration, or intricate ECM interactions. The ability to map agent variables between models and share environment properties enhances the framework's flexibility, enabling the construction of sophisticated simulations that can address a wide range of biological questions. This capability significantly expands the model's applicability, allowing for the detailed simulation of biological processes that are challenging to capture with traditional modelling approaches. Future developments should focus on enhancing the model's versatility, incorporating more templates for detailed biological processes, and expanding its accessibility to foster a wider user community.

Availability of data and materials

The work presented here was built upon the FLAMEGPU2 framework. The code is publicly available via: https://doi.org/10.5281/zenodo.10804507. To run it, an NVIDIA graphics card is needed and FLAMEGPU2 must be installed. Instructions can be found at https://flamegpu.com/and their release archive: https://zenodo.org/doi/10.5281/zenodo.5428984.

CRediT authorship contribution statement

C. Borau: Writing – original draft, Methodology, Formal analysis, Conceptualization. **R. Chisholm:** Software, Methodology. **P. Richmond:** Writing – review & editing, Supervision. **D. Walker:** Writing – review & editing, Supervision.

Declaration of competing interest

None declared.

Acknowledgements

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826494.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2024.108831.

References

[1] D.A. Cruz Walma, K.M. Yamada, The extracellular matrix in development, Development 147 (10) (May 2020), https://doi.org/10.1242/DEV.175596.
[2] B. Yue, Biology of the extracellular matrix: an overview, J. Glaucoma 23 (8) (Dec. 2014) S20, https://doi.org/10.1097/IJG.0000000000000108.
[3] A.S. Hoffman, Hydrogels for biomedical applications, Adv. Drug Deliv. Rev. 64 (SUPPL) (Dec. 2012) 18–23, https://doi.org/10.1016/J.ADDR.2012.09.010.
[4] Y.S. Zhang, A. Khademhosseini, Advances in engineering hydrogels, Science 356 (6337) (1979), https://doi.org/10.1126/SCIENCE.AAF3627/ASSET/FF981CF9-F818-45D8-A693-CA10D4ABEE12/ASSETS/GRAPHIC/356_AAF3627_F5.JPEG. May 2017.
[5] M. Hu, Z. Ling, X. Ren, Extracellular matrix dynamics: tracking in biological systems and their implications, J. Biol. Eng. 16 (1) (May 2022) 1–13, https://doi.org/10.1186/S13036-022-00292-X, 2022 16:1.
[6] L.T. Saldin, M.C. Cramer, S.S. Velankar, L.J. White, S.F. Badylak, Extracellular matrix hydrogels from decellularized tissues: structure and function, Acta Biomater. 49 (Feb. 2017) 1–15, https://doi.org/10.1016/J.ACTBIO.2016.11.068.
[7] S. Uriel, et al., Extraction and assembly of tissue-derived gels for cell culture and tissue engineering, Tissue Eng. C Methods 15 (3) (Sep. 2009) 309–321, https://doi.org/10.1089/TEN.TEC.2008.0309.
[8] V. Ayyalasomayajula, B. Pierrat, P. Badel, A computational model for understanding the micro-mechanics of collagen fiber network in the tunica adventitia, Biomech. Model. Mechanobiol. 18 (5) (Oct. 2019) 1507–1528, https://doi.org/10.1007/S10237-019-01161-1/FIGURES/17.

[9] B. Lee, et al., A three-dimensional computational model of collagen network mechanics, PLoS One 9 (11) (Nov. 2014), https://doi.org/10.1371/JOURNAL.PONE.0111896.

[10] H. Wong, J.M. Crowet, M. Dauchez, S. Ricard-Blum, S. Baud, N. Belloy, Multiscale modelling of the extracellular matrix, Matrix Biol. 13 (Feb. 2022) 100096, https://doi.org/10.1016/J.MBPLUS.2021.100096.

[11] M.C. Kim, Y.R. Silberberg, R. Abeyaratne, R.D. Kamm, H.H. Asada, Computational modeling of three-dimensional ECMrigidity sensing to guide directed cell migration, Proc. Natl. Acad. Sci. U. S. A. 115 (3) (Jan. 2018) E390–E399, https://doi.org/10.1073/PNAS.1717230115/SUPPL_FILE/PNAS.1717230115.SM08.AVI.

[12] N. Movilla, C. Borau, C. Valero, J.M. García-Aznar, Degradation of extracellular matrix regulates osteoblast migration: a microfluidic-based study, Bone 107 (2018), https://doi.org/10.1016/j.bone.2017.10.025.

[13] J.F. Eichinger, et al., A computational framework for modeling cell–matrix interactions in soft biological tissues, Biomech. Model. Mechanobiol. 20 (5) (Jun. 2021) 1851–1870, https://doi.org/10.1007/S10237-021-01480-2, 2021 20:5.

[14] I. Davoodi Kermani, M. Schmitter, J.F. Eichinger, R.C. Aydin, C.J. Cyron, Computational study of the geometric properties governing the linear mechanical behavior of fiber networks, Comput. Mater. Sci. 199 (Nov. 2021) 110711, https://doi.org/10.1016/J.COMMATSCI.2021.110711.

[15] F. Merino-Casallo, M.J. Gomez-Benito, R. Martinez-Cantin, J.M. Garcia-Aznar, A mechanistic protrusive-based model for 3D cell migration, Eur. J. Cell Biol. 101 (3) (Jun. 2022) 151255, https://doi.org/10.1016/J.EJCB.2022.151255.

[16] F.R. Cooper, et al., Chaste: cancer, heart and soft tissue environment, J. Open Source Softw. 5 (47) (Mar. 2020) 1848, https://doi.org/10.21105/JOSS.01848.

[17] A. Ghaffarizadeh, R. Heiland, S.H. Friedman, S.M. Mumenthaler, P. Macklin, PhysiCell: an open source physics-based cell simulator for 3-D multicellular systems, PLoS Comput. Biol. 14 (2) (Feb. 2018) e1005991, https://doi.org/10.1371/JOURNAL.PCBI.1005991.

[18] S. Hoehme, D. Drasdo, A cell-based simulation software for multi-cellular systems, Bioinformatics 26 (20) (Oct. 2010) 2641–2642, https://doi.org/10.1093/BIOINFORMATICS/BTQ437.

[19] S. Mathias, A. Coulier, A. Hellander, CBMOS: a GPU-enabled Python framework for the numerical study of center-based models, BMC Bioinf. 23 (1) (Dec. 2022), https://doi.org/10.1186/S12859-022-04575-4.

[20] J. Du, Y. Zhou, L. Jin, K. Sheng, Gell: a GPU-powered 3D hybrid simulator for large-scale multicellular system, PLoS One 18 (7) (Jul. 2023), https://doi.org/10.1371/JOURNAL.PONE.0288721.

[21] P. Richmond, R. Chisholm, P. Heywood, M.K. Chimeh, M. Leach, Flame GPU 2: a framework for flexible and performant agent based simulation on GPUs, Software Pract. Ex. 53 (8) (Aug. 2023) 1659–1680, https://doi.org/10.1002/SPE.3207.

[22] J. Pleyer, C. Fleck, Agent-based models in cellular systems, Front. Physiol. 10 (Jan) (2023), https://doi.org/10.3389/FPHY.2022.968409/FULL.

[23] M. Stack, P. Macklin, R. Searles, S. Chandrasekaran, J.C. Carver, K. Morris, OpenACC acceleration of an agent-based biological simulation framework, Comput. Sci. Eng. 24 (5) (Sep. 2022) 53–63, https://doi.org/10.1109/MCSE.2022.3226602.

[24] M. Yang, P. Andelfinger, W. Cai, A. Knoll, Evaluation of conflict resolution methods for agent-based simulations on the GPU, in: SIGSIM-PADS 2018 - Proceedings of the 2018 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, May 2018, pp. 129–132, https://doi.org/10.1145/3200921.3200940.

[25] H. Wang, X. Xu, Continuum elastic models for force transmission in biopolymer gels, Soft Matter 16 (48) (Dec. 2020) 10781–10808, https://doi.org/10.1039/D0SM01451F.

[26] S. Motte, L.J. Kaufman, Strain stiffening in collagen I networks, Biopolymers 99 (1) (Jan. 2013) 35–46, https://doi.org/10.1002/BIP.22133.

[27] J. Steinwachs, et al., Three-dimensional force microscopy of cells in biopolymer networks, Nat. Methods 13 (2) (Dec. 2015) 171–176, https://doi.org/10.1038/nmeth.3685, 2015 13:2.

[28] Z. Wang, Regulation of cell cycle progression by growth factor-induced cell signaling, Cells 10 (12) (Dec. 2021), https://doi.org/10.3390/CELLS10123327.

[29] C. Valero, H. Amaveda, M. Mora, J.M. García-Aznar, Combined experimental and computational characterization of crosslinked collagen-based hydrogels, PLoS One 13 (4) (Apr. 2018) e0195820, https://doi.org/10.1371/JOURNAL.PONE.0195820.

[30] C.B. Goy, R.E. Chaile, R.E. Madrid, Microfluidics and hydrogel: a powerful combination, React. Funct. Polym. 145 (Dec. 2019) 104314, https://doi.org/10.1016/J.REACTFUNCTPOLYM.2019.104314.

[31] Y. Juste-Lanas, et al., 3D collagen migration patterns reveal a SMAD3-dependent and TGF-β1-independent mechanism of recruitment for tumour-associated fibroblasts in lung adenocarcinoma, Br. J. Cancer 128 (6) (Dec. 2022) 967–981, https://doi.org/10.1038/s41416-022-02093-x, 2022 128:6.

[32] B.L. Yen, C.C. Hsieh, P.J. Hsu, C.C. Chang, L.T. Wang, M.L. Yen, Three-dimensional spheroid culture of human mesenchymal stem cells: offering therapeutic advantages and in vitro glimpses of the in vivo state, Stem Cell. Transl. Med. 12 (5) (May 2023) 235, https://doi.org/10.1093/STCLTM/SZAD011.

[33] H.E. Balcioglu, B. Van De Water, E.H.J. Danen, Tumor-induced remote ECM network orientation steers angiogenesis, Sci. Rep. 6 (1) (Mar. 2016) 1–12, https://doi.org/10.1038/srep22580, 2016 6:1.

[34] C. Borau, et al., A multiscale orchestrated computational framework to reveal emergent phenomena in neuroblastoma, Comput. Methods Progr. Biomed. 241 (Nov. 2023) 107742, https://doi.org/10.1016/J.CMPB.2023.107742.