Journal of Computational Science 75 (2024) 102198

Contents lists available at ScienceDirect

Journal of Computational Science

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

Mesoscale material modeling with memoryless isotropic point particles

Erik Strand [a,*], Filippos Tourlomousis [b,c], Neil Gershenfeld [a]

[a] MIT Center for Bits and Atoms, United States of America
[b] Superlabs, Marousi, Attica, Greece
[c] National Centre for Scientific Research Demokritos, Agia Paraskevi, Attica, Greece

ARTICLE INFO

Keywords:
Particle systems
Discrete element method
Molecular dynamics
Machine learning
Anisotropy
Hysteresis
Failure
Composites

ABSTRACT

There has been a proliferation of particle systems developed to model complex systems. These are attractive because they are mesh-free, avoiding issues associated with solver remeshing and convergence. They have however fragmented into niches, using increasingly complex particles that introduce internal degrees of freedom and external solver coupling. We show that, contrary to prior assumptions in the literature, memoryless isotropic point particles can model material properties including anisotropy, hysteresis, and failure solely through the statistics of their distributions. The resulting models offer compact code that is straightforward to accelerate and port, can span between micro- and macro-structure, require few parameters to set up a simulation, and unlike high-dimensional machine learning models they use a low-dimensional representation that can be efficiently learned. Rather than deriving them as approximations to either molecular dynamics or partial differential equations we investigate how these models can be found directly, and illustrate this with both qualitative comparisons of phenomenology and quantitative comparisons of predictions.

1. Introduction

Molecular dynamics (MD) is fundamental, starting from an atomic description, however it is not applicable on macroscopic time or length scales; the frontier is milliseconds [1]. Partial differential equations (PDEs) are also considered to be fundamental, however they are an approximation that can break down, and their use requires selecting both governing equations and a solver. The solvers are themselves complex systems that can introduce unphysical dynamics, can require repeated remeshing to follow changing geometries, and can fail to converge.

Particle systems are a mesh-free alternative that have proliferated in application domains including Direct Simulation Monte Carlo (DSMC) [2], Discrete Element Method (DEM) [3], Dissipative Particle Dynamics (DPD) [4], the Lattice Boltzmann Method (LBM) [5], Lattice Gas Hydrodynamics (LGH) [6], the Material Point Method (MPM) [7], Movable Cellular Automata (MCA) [8], Parallel Particle–Mesh (PPM) [9], Peridynamics (PD) [10], Position Based Dynamics (PBD) [11], and Smoothed Particle Hydrodynamics (SPH) [12]. These can be derived as a coarse-graining of molecular dynamics, or as a discrete approximation to continuum PDEs. They all share particles in common, but differ in their internal degrees of freedom and the solvers that they are coupled to.

We ask here whether mesoscale particles can be found directly as an equally fundamental representation, rather than assumed as an approximation. We take as a test case modeling materials spanning between their micro and macro structure, and choose to use the simplest common denominator of particle models: memoryless isotropic point particles (MIPS). MIPS is not new; rather, it abstracts the essential features of a particle system.

MIPS trades computational efficiency for simplicity. This approach can be thought of as a simulation analogy to the role of RISC versus CISC in computer instruction sets [13]. Increasingly complex instructions were displaced by the use of a small set of simple instructions that could be efficiently executed, moving the complexity into the compilers that call them. Similarly, MIPS replaces complex solvers with simple dynamics, and moves the complexity into the particle distributions.

The feasability of this approach rests on the increasing performance and availability of computational accelerators that can take advantage of the massive parallelism of MIPS. We will show that this can reduce millions of lines of code in existing solvers to hundreds; benefits of this simplicity include the ease of specifying a model, a reduction in the number of parameters required to learn a model, the ease of porting to new languages and architectures, and the effective equivalence between the description of a model and its computational implementation.

The force laws are not directly observable because they govern the nonlinear collective behavior of the particles. They can instead

* Corresponding author.
E-mail address: erik.strand@cba.mit.edu (E. Strand).

https://doi.org/10.1016/j.jocs.2023.102198
Received 1 March 2023; Received in revised form 5 November 2023; Accepted 4 December 2023
Available online 13 December 2023

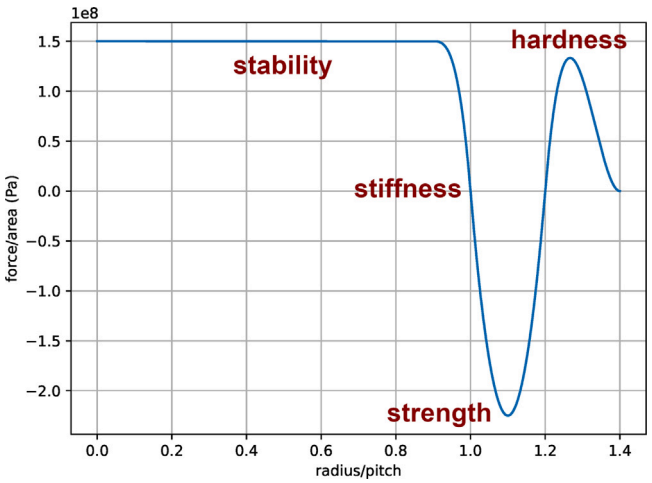

Fig. 1. Sample force law and its effective properties.

be found qualitatively by matching phenomenology, and quantitatively by matching indirect experimental observations. We illustrate these in the following section, followed by applications to modeling anisotropy, hysteresis, and failure, and conclude with implications for future hardware and software.

2. Methods

2.1. Particle system physics

Our models consist of a collection of point particles subject to a pairwise, isotropic force law. Each particle's state is described entirely by its position, velocity, and mass. In this work we assume that all particle masses are identical and constant, though this assumption could be relaxed.

The equations of motion of the system are simply Newton's laws. The force law prescribes the magnitude of the force that acts radially between two interacting particles as a function of the distance between them. Equivalently, one can view the pairwise interactions as determined by a one dimensional potential function. For efficiency, we assume the force law vanishes beyond a certain cutoff distance. This assumption could be relaxed if a long-range force approximation method were used, such as the fast multipole method [14].

Most macroscopic geometry descriptions are not particle based, so a translation procedure is needed to generate the particles and set their initial positions. In this work, we pack particles in the densest sphere-packing with equivalent sites: a face-centered cubic (FCC) lattice [15]. As such the possible initial particle positions are specified by the lattice pitch. The particle masses are set to achieve the desired bulk material density, and for all the simulations discussed in this paper the initial particle velocities are zero.

The choice of FCC particle packing effectively also imposes constraints on the force laws that govern particle interactions. In particular, it is usually desired that forces vanish at the lattice pitch so that the initial state is stable. Stiffness is determined by the slope at that point; below the lattice pitch the repulsive force is rolled off for numerical stability. Above it there is a restoring force which represents strength, and finally a repulsive force out to a cutoff at the next nearest-neighbor distance that corresponds to hardness. These features are visible in the example force law depicted in Fig. 1.

If all particles are initialized to lattice positions, and interaction forces vanish at the lattice pitch, then all particles remain stationary indefinitely. To study strains, displacements, etc. boundary conditions must be applied. We apply boundary conditions of three forms: position constraints, velocity constraints, and force constraints. Position and velocity constraints simply override the values computed by the integrator. In some situations this can lead to instabilities, since the forces that should be determining particle velocities and positions become uncoupled from the system and may grow without bound. For this reason, it is sometimes preferable to use a force constraint even if one's intent is to hold a particle in a certain location. This result can be achieved using a simple control law, e.g. a proportional–derivative (PD) controller, that applies the force needed to keep the particle in position.

2.2. Implementation

The complete simulation sequence entails populating CAD files with particles, assigning properties to those particle, applying boundary conditions, integrating the dynamics, and aggregating and visualizing the results. Because of the pre- and post-processing effort this requires with existing molecular dynamics packages such as LAMMPS [16], these steps were implemented in dedicated workflows which were ported to multiple accelerated computing environments for comparison.

A suite of modular C++ programs were written to facilitate composing simulations. These communicated in BASH scripts through memory pipes; on an Intel Xeon E5-2686 the bandwidth between the modules was on the order of 400 Mbps. GPU acceleration used CUDA kernels [17]; these were able to update on the order of a hundred million particles per second on one NVIDIA V100. This version used a few thousand lines of code; for comparison, NASTRAN requires over a million lines [18].

To further simplify the code it was rewritten using Python accelerators. One version used JAX [19] which automates the parallelization of array operations, and another version used Taichi [20] which automates the parallelization of kernel outer loops. These reduced the code to hundreds of lines, and delivered performance that was almost comparable to the hand-tuned CUDA kernels, updating tens of millions of particle per second on a V100.

The most expensive computational step is accumulating the forces. Naively this is an $O(N^2)$ calculation, but because of the cutoff of the force law it becomes $O(N)$ with a spatial sort. This was parallelized with atomic operations for the bin occupancy, and the bin storage was dynamically allocated with a cumulative sum prior to the radix sort, sharing a buffer equal in length to the number of particles rather than the much larger sample volume. If some margin is added to the cutoff distance the resulting neighbor list can be reused in subsequent time steps as long as no particle has moved far enough to form a new interaction pair. This requires storing the positions of all particles at the time of the last sort, but allows the cost of the operation to be amortized over several timesteps.

For the integrator we compared use of both symplectic velocity Verlet and semi-implicit Euler methods [21], with similar results. A higher-order method was not used because of the discontinuities in these simulations. For numerical stability information must not propagate faster than the speed of sound, which for a linear array of masses and springs is equal to the square root of the elastic force divided by the mass [21]. In the MKS units used here that is the slope of the radially normalized force law in Pascals divided by the particle mass in kilograms. That can be converted to a time step by dividing the lattice pitch by the velocity; for typical values used here that gives:

$$10^{-4}\ \text{m} * \sqrt{10^{3}\ \text{kg/m}^3\ /\ 10^{9}\ \text{Pa}} = 10^{-7}\ \text{s} \quad (1)$$

This is in good agreement with the observed stability limit, and for these mesoscale effective particles it is much longer than the 10^{-15} s time steps that are typical in molecular dynamics. For a million particle simulation on one V100 GPU that corresponds to:

$$10^{6}\ \text{particles}/(10^{8}\ \text{particles/clock s}\ *\ 10^{-7}\ \text{sim s}) = 10^{5}\ \text{clock s/sim s} \quad (2)$$

(about a day per simulation second).

If inertial forces are not important a larger fictitious mass allows longer time steps to be used. And a dissipative term was added to the

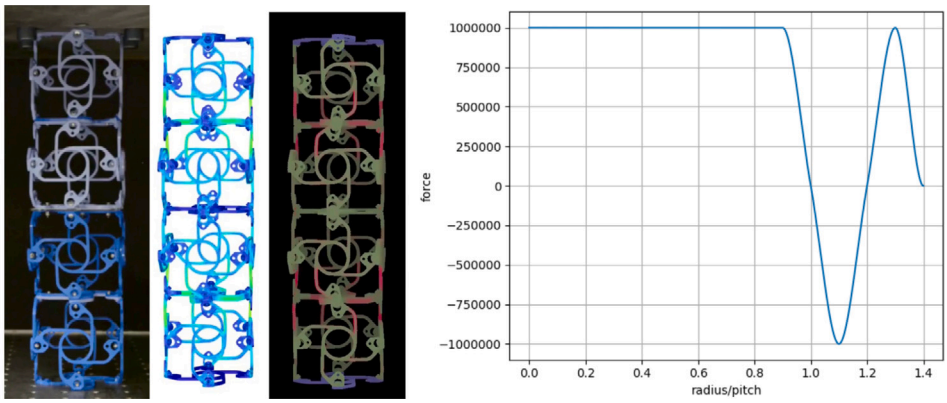

Fig. 2. Comparison of simulating deformation of a chiral metamaterial, showing the agreement between experiment (left), NASTRAN (center), and MIPS using the force law (right).

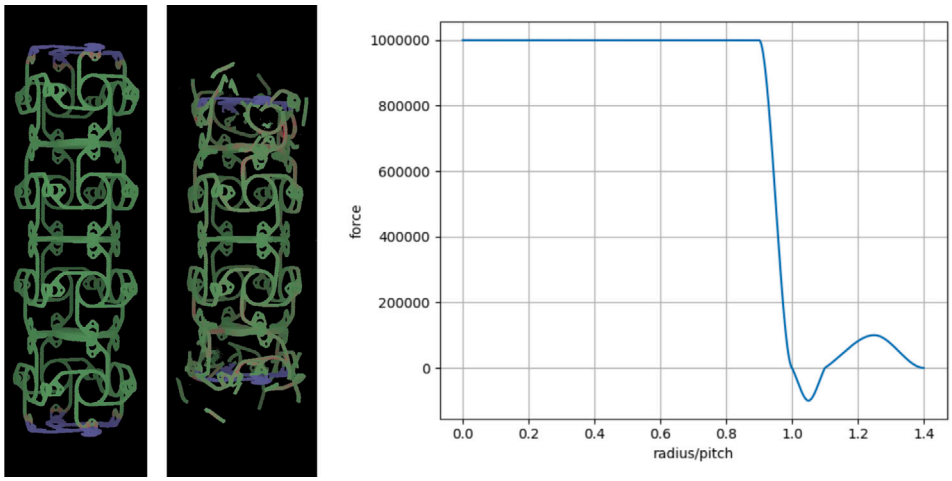

Fig. 3. The simulation in Fig. 2 rerun with a force law to model a brittle material.

force law; in addition to modeling internal and external friction, it is needed for numerical stability if the boundary conditions put energy into the system. To maintain Galilean invariance, we make this term linearly proportional to the difference in radial velocity between the particles in each interacting pair.

To initialize the particle positions, we must test if each FCC lattice site within the simulation volume is in the interior or exterior of the relevant geometry. This process is easily parallelized, so it is not a significant bottleneck even with hundreds of millions of particles. In this work, we use functional (signed distance field) and mesh based geometry descriptions. The former trivially supports interior/exterior queries; for the latter they are implemented by ray casting. The lattice pitch should be small relative to the minimum feature sizes in the simulated geometries; as long as this constraint is satisfied we found that simulation results are not sensitive to the specific value.

3. Phenomenology

To illustrate the phenomenology that is possible with these simple models, Fig. 2 shows the agreement between experiment, NASTRAN Finite Element Analysis (FEA) (using the default Fusion 360 solver settings [22]), and MIPS for simulating the deformation of a chiral metamaterial [23]. It used thousands rather than millions of lines of code to reproduce the FEA results, and demonstrates how isotropic particles can be used to model anisotropic properties (to be further developed below). Figs. 3 and 4 show the same simulation run again, just changing a few points in the force law to model brittle and viscous materials, which conventionally require different types of solvers.

4. Finding force laws

Because these effective force laws are not directly observable, to make quantitative predictions we train them by searching over simulations of experimental results. An example model was trained on the elastic and plastic deformation of Delrin under tension. We started with experimental stress–strain data collected on a material testing machine, as shown in Fig. 5. The curve in this chart is the entirety of our training data. We then modeled the same experiment virtually, allowing us to collect a force-versus-displacement curve akin to the experimental reference. Finally, we defined an objective function that computes the L2 difference between the experimental and simulated curves, and ran an optimization routine to find the force laws that minimize this difference.

Setting up the virtual experiment consists of defining the coupon geometry, and setting up boundary conditions and measurements. In this case, we used a functional representation of the coupon geometry that we constructed from the ASTM documentation [24]. The boundary conditions fixed the positions of a layer of particles on the bottom of the coupon, and enforced movement at a constant rate of a layer of particles at the top. In this experiment the boundary conditions were enforced by applying forces to particles rather than setting the positions directly, but this technique is not essential to the work. The forces applied were determined by a proportional–derivative controller, with a proportional coefficient one order of magnitude greater than the maximum stiffness in the force law, and a derivative coefficient of equal magnitude to the dissipative term in the MIPS model.

We used cardinal 2D B-splines of order 3 to model the force law. For convenience, the force law was broken into two splines: a repulsive

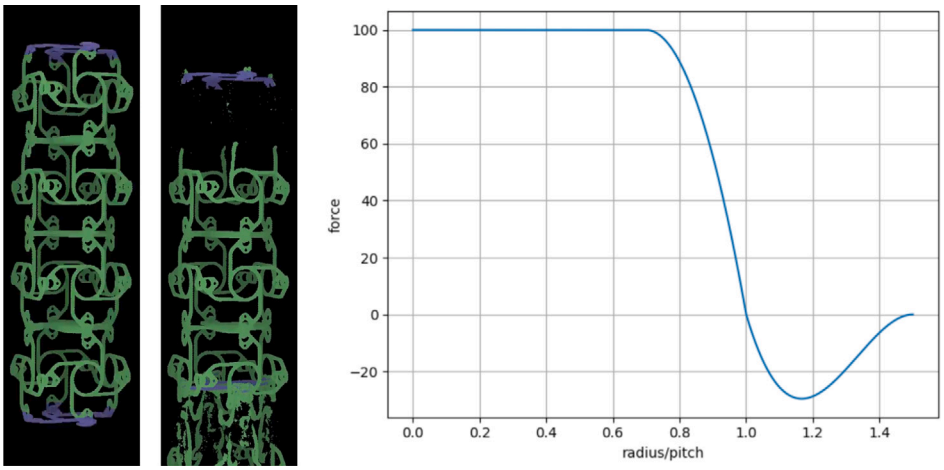

Fig. 4. The simulation in Fig. 2 rerun with a force law to model a viscous material.

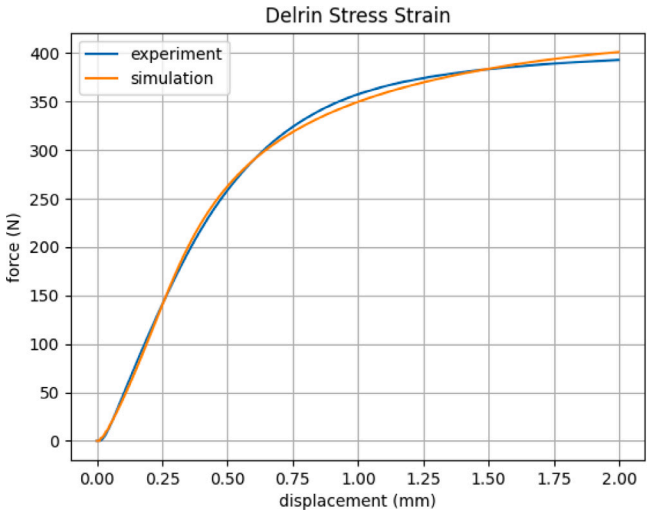

Fig. 5. Experimental and simulated stress–strain curves for Delrin.

portion with five control points acting from a distance of zero to one lattice pitch, and an attractive portion with six control points acting from one to three lattice pitches. To ensure stability, both functions were constrained to a force of zero at a distance of one lattice pitch. The repulsive force was further constrained to level off between the first and second control points, and the attractive force was constrained to vanish at three lattice pitches. The tangents at the joining point were controlled by an angle parameter to ensure continuity. The net result was thirteen free parameters. For computational efficiency, we maintained nearest lattice neighbors for computation of the attractive forces.

The model is then fit by minimizing the mean squared error between the experimental and simulated stress–strain curves. To solve this optimization problem, we used the covariance matrix adaptation evolution strategy (CMA-ES) [25]. This gradient-free optimization algorithm balances local and global search by iteratively refining a multivariate Gaussian distribution that models the relationship between the independent and dependent variables (in this case, the design parameters and the value of the objective function). We used 16 samples per generation and a relative convergence threshold of 0.01 in the objective value, with the c-cmaes software package [26]. Fig. 6 depicts the optimization progress for the simulated force law depicted in Fig. 5.

The simulation is in principle differentiable, so one could use a gradient based method such as gradient descent. However, for the low-dimensional design spaces we use, gradient based search is not necessary. Furthermore, eschewing gradients decouples the optimization problem from the simulation time. When computing gradients through a simulation, the chain rule must be applied at each time step. After thousands of steps, the gradients tend to vanish or grow without bound. For this reason, optimization over differentiable simulations are usually limited to the order of 1000 time steps [27]. In contrast, we commonly optimize over tens or hundreds of thousands of time steps.

Being data driven, it is natural to compare our method to machine learning based techniques. Our method involves orders of magnitude fewer parameters than those involving neural networks, thus enabling the use of much smaller training datasets. Physics Informed Neural Networks (PINNs) [28] are a powerful deep learning framework for solving ODEs and PDEs. They can also be trained with few training data, but only if the governing equations of the system are known directly so that residuals can be included in the objective function. In contrast, our method requires very little training data and can be used when very little is also known about the governing equations of the system.

5. Hysteresis

In PDE based numeric simulations and more complex particle methods, memory kernels are commonly employed to maintain history dependent state [29]. Sophisticated techniques have been developed to derive these memory kernels, such as the Mori-Zwanzig formalism [30]. The underlying assumption in such methods is that the history dependence should be accounted for within the governing equations of the model. This stands in contrast to fundamental physics theories, in which the forces acting on a system depend only on its current state.

We find that history dependence naturally emerges via the evolution of the model state of MIPS models. Fig. 7 depicts a comparison of experimental and simulated cyclic loading of an ASTM D638 coupon. The force law used in the MIPS simulation was trained on a single stress–strain curve (specifically, the experimental data depicted in Fig. 5). When cyclically loaded, rather than extended linearly, the same experimental setup produces a cyclic stress–strain curve. The resulting hysteresis loop resembles that of the equivalent physical experiment (Fig. 7). This is notable given that the data used to train the model included no samples taken under cyclic loading. The history dependence emerges as a result of particle rearrangement. The energy being added to the system is dissipated by the damping term discussed in Section 4.

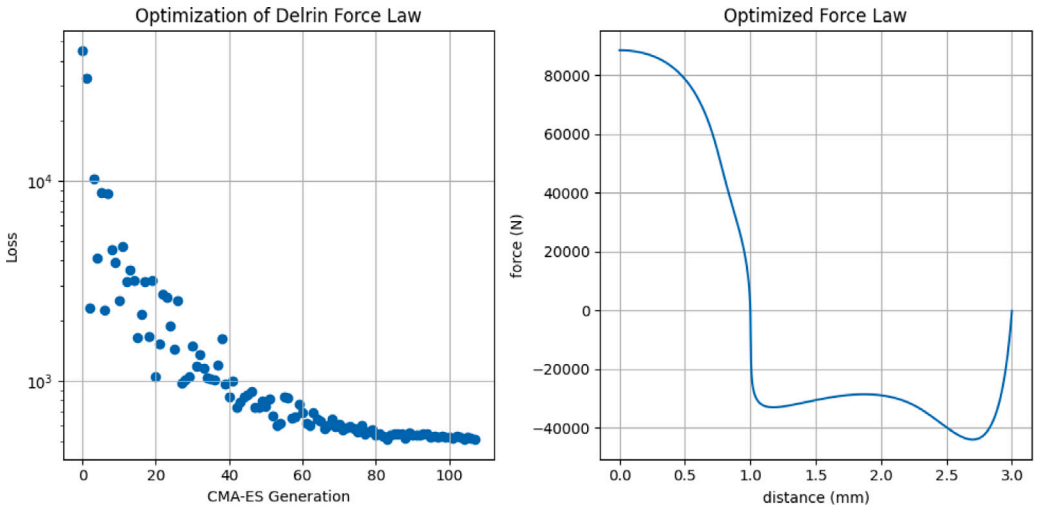

Fig. 6. Left: Objective function value vs. CMA-ES generation. Right: Optimized force law.

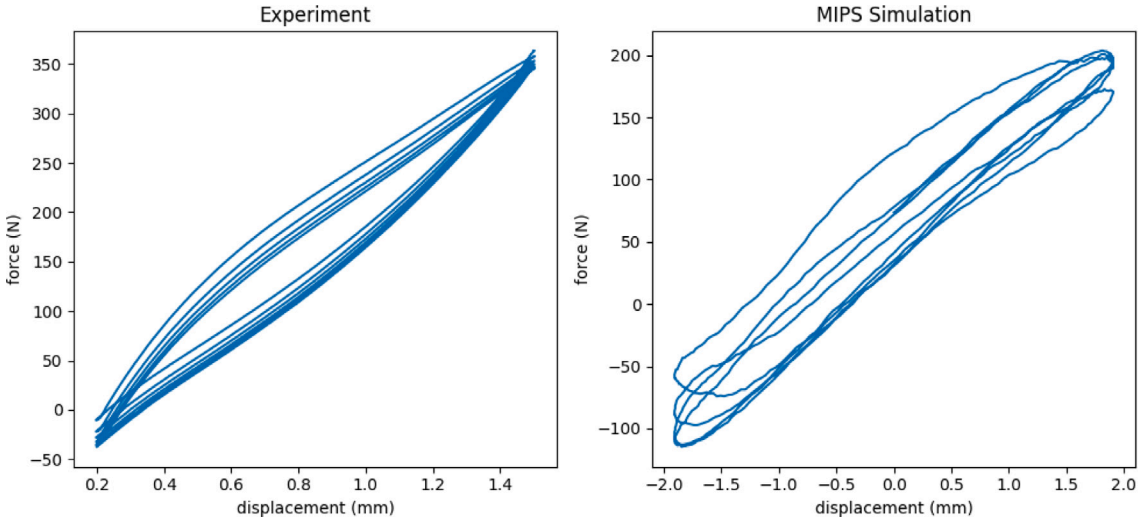

Fig. 7. Experimental (left) and simulated (right) cyclic loading of Delrin. The simulated model exhibits hysteresis despite being trained only on non-cyclic data.

6. Anisotropy

For any particular crystal structure, relationships between the interatomic potentials and the resulting elastic constants can be derived [31]. The result is generally a loss of degrees of freedom in the elastic constants, which imposes constraints on the stiffness tensor [32]. For the FCC lattices used in these studies, $c_{44} = c_{12}$. This would appear to indicate that MIPS cannot be used to model materials for which this constraint does not hold.

The situation is different when bonds may be formed and break dynamically. The molecular dynamics literature contains a wide variety of bond order potentials, such as the Tersoff potential [33], the Brenner potential [34], and the EDIP potential [35]. These potentials still express the binding energy as a sum of pairwise interactions, but each term incorporates a bond order parameter that is a function of the neighborhood of the particle pair. In full generality the bond order functions can include chiral or directional dependence, opening the door to anisotropic interactions.

MIPS offers a simpler approach to anisotropy that bypasses the constraints of rigid lattice structures but also eschews the additional model and computational complexity introduced by bond order potentials. This is accomplished by annotating each particle with a particle type that controls the strength of its interactions. In the simplest case, particles are divided into a strongly interacting group and a weakly interacting group. All pairwise particle interactions use a force law of the same shape, but the resulting force is scaled based on the particle types of the two involved particles. As a result, the bulk material properties depend on the distribution of positions of the annotated particles. This allows anisotropy to be embedded in geometry, rather than the governing equations.

This approach is inspired by mechanical and electronic metamaterials [23,36], which utilize geometry to achieve bulk behavior impossible for a homogeneous block of any of their constituent materials. (For example, Fig. 2 depicts a chiral metamaterial.) The principle also has roots in materials science: Ti64, a titanium alloy common in aerospace and medical applications, exhibits anisotropy as a result of the relative proportion and spatial distribution of grains with two different crystal structures.

Fig. 8 demonstrates a case study of an anisotropic tensile response. We apply tensile stresses along two orthogonal axes to an axis-aligned

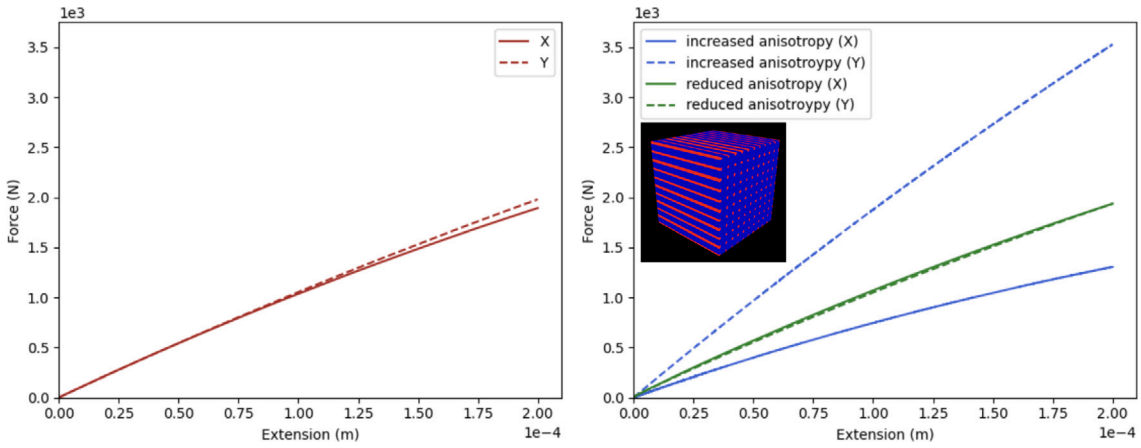

Fig. 8. Left: anisotropic tensile response of MIPS particles on a FCC lattice with a uniform force law distribution. Right: reducing and increasing the anisotropy through variation of the particle force law spatial distribution (illustrated in the inset).

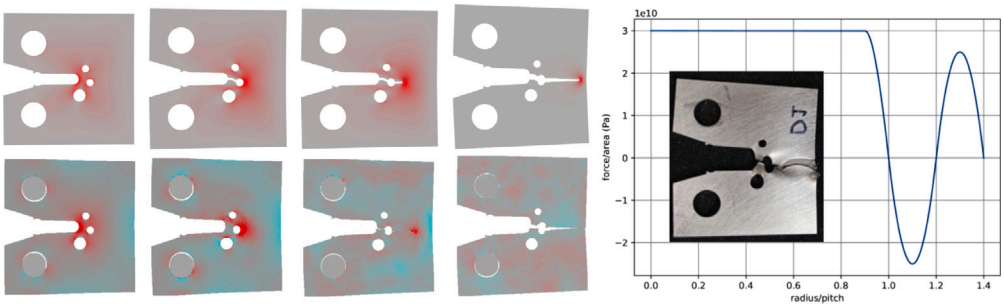

Fig. 9. Crack propagation in the Sandia fracture challenge solved by COMSOL FEA (top) and MIPS (bottom) using the force law (right) compared with experiment (inset, from [37]).

cubic block of material 1 cm on each side. As a baseline, we use the same pairwise potential for all particle interactions. The potential was chosen to produce a stiffness of about 1 GPa. Some anisotropy is evident due to the FCC lattice structure. To alter the anisotropic response, we divide the particles into strongly and weakly interacting groups. To increase the anisotropy, we distribute particles in the strongly interacting group along fibers running parallel to the Y axis. Particles in the strongly interacting group experience forces twenty times stronger than the baseline, while particles in the weakly interacting group experience forces a factor of one half weaker. To decrease the anisotropy, we orient the fibers of strongly interacting particles parallel to the X axis, and scale strongly interacting particle forces by a factor of 3.53, and weakly interacting particle forces by a factor of 0.82.

This example illustrates the principle of tuning bulk material anisotropy based on particle distributions; future work will investigate searching over force law assignments to match arbitrary stiffness tensors.

7. Failure

In this section we investigate how MIPS can model failure, comparing both the mechanisms found and the modeling effort. The first test case is fracture-induced failure of ductile materials, comparing the path of crack propagation in the Sandia challenge [37] between experiment, MIPS, and FEA simulations with Phase Field Modeling (PFM).

PFM solves an ordinary differential equation for tracking the sharp interface of the crack, whose geometry is represented in a diffusive way without sharp discontinuities by a scalar field that smoothly transits the phase of the non-damaged material to the phase of the damaged material. PFMs are favored for modeling the branching and merging of multiple cracks over alternative modeling techniques where the crack is treated as a physical discontinuity introducing external criteria for fracture and additional algorithmic steps to track the fracture shape [38].

Quasi-static crack propagation simulation of the Sandia challenge was performed with COMSOL Multiphysics, shown at the top of Fig. 9. The Sandia challenge scenario was set up in COMSOL assuming plane stress condition and following the modeling steps described in the holed plate application example [39], whose results are benchmarked against previously published literature [40,41]. Loading is applied through prescribed displacements to the boundaries of the two larger holes. Steel AISI 4340, as provided by COMSOL's materials library (Young's modulus 205 GPa, Poisson's ratio 0.28, and critical energy release rate 2280 J/m^2), was used in the simulations due to its high ductility.

A phase field approximation was made after the sharp crack geometry, and cracking was incorporated into the domain material. The set up of the phase field damage model does not include a damage threshold, meaning that all material points subjected to tension are damaged. The material input to the damage model is the critical energy release rate, which was chosen to be within the range of 1500–2500 J/m^2, whereas the tensile strength is determined by the damage evolution function and the evolution of the phase field, which is dependent on the internal length scale setting which controls the width of the localized phase field and was set to 0.25 mm.

From a usability perspective, the FEA simulation workflow for fracture-induced failure has many limitations. The most important point is the need to tune the solver configuration to achieve convergence. We used a linear elastic model with damage to account for tensile cracking. The scalar phase field evolution is modified by the internal length scale parameter that controls the width of the crack interface. To properly resolve the crack without unstable material behavior, significant time and effort is required of the user to define an appropriate mesh that is locally refined according to the expected crack trajectory, in combination with an efficient and stable solver configuration.

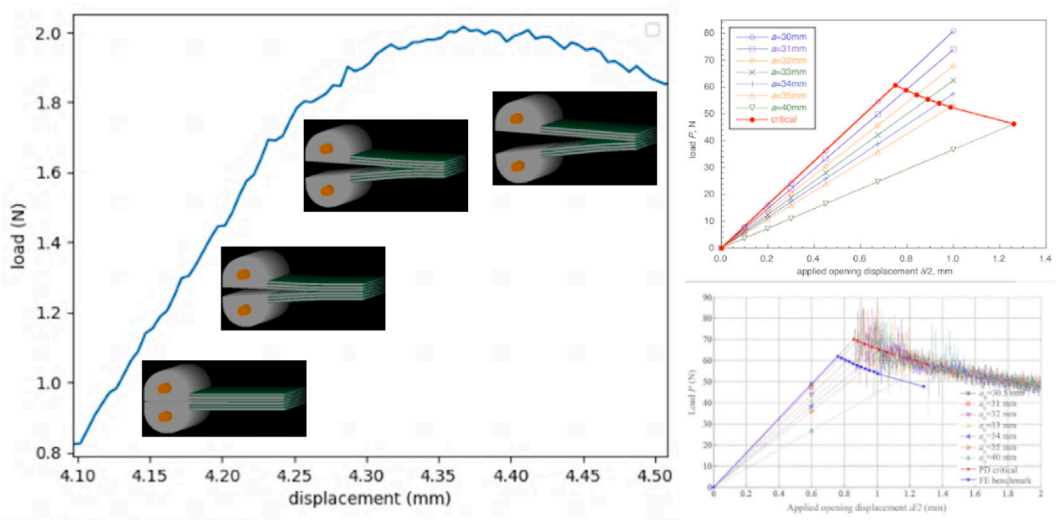

Fig. 10. MIPS simulation of the delamination of a double cantilever beam showing characteristic decreasing load with increasing delamination length (left), compared with load–displacement curve critical paths found from families of virtual crack closure technique [44] (right top) and peridynamic [45] (right bottom) simulations.

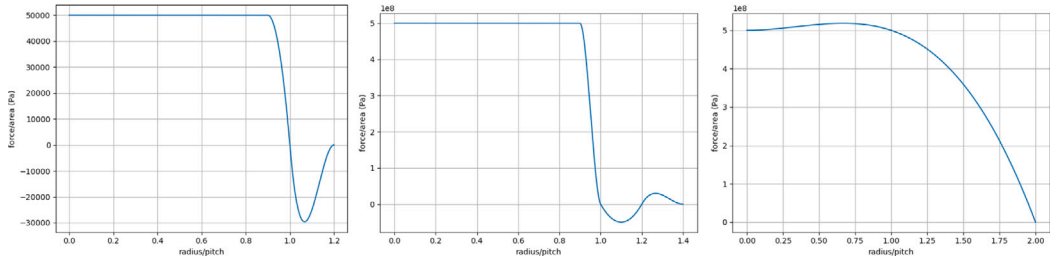

Fig. 11. Resin (left), fiber (middle), and repulsive (right) force laws used for the composite DCB simulation.

Depending on the user expertise with FEA and PFM, the metric of user time can range from days to months. The simulation step becomes even more overwhelming with the multiple semi-empirical material models for damage, whose number of parameters vary and need to be determined during an intermediate calibration step before the actual simulation task begins. This is evident by the many materials models reported in the Sandia challenge [42,43].

Compared to this, the bottom of Fig. 9 shows the MIPS solution to the same problem. The strain field is plotted, with red showing tension and blue compression. This used 5 million 100 μm particles, and the simulation required on the order of 15 min on a V100 GPU. The strain distribution and crack path is similar to both experiment (from [37], shown in the inset) and the COMSOL simulation. Unlike PFM there was no need for MIPS to explicitly follow the crack; all that was specified was the force law shown on the right and the boundary conditions that displaced the clevis pins.

A second test case was modeling composite failure, which is urgently needed for their wider adoption [46]; the challenge is the need to span between their micro- and macro-structure [47,48]. The goal here was to simulate the delamination of a double cantilever beam (DCB). Fig. 10 shows a MIPS simulation using layers of stiff fiber and softer resin particles and the boundary condition imposed by again displacing clevis pins, with the force laws shown in Fig. 11.

This simulation shows the characteristic behavior of an initial increase in the load, followed by a decrease as the delamination length grows. Fig. 10 compares DCB load–displacement curves showing that behavior found from families of virtual crack closure technique [44] and peridynamic [45] simulations tracing the critical load/displacement path. Again MIPS is able to do this without needing to explicitly follow the delamination front or break bonds at the interface.

8. Conclusion

We have shown how MIPS can trade computational effort for simplicity, reproducing results from much more complex algorithms in challenging material modeling areas including anisotropy, hysteresis, and failure. It effectively equates the description of a model and its computational solution; future work will develop this as a quantitative starting point rather than just a numerical approximation. This approach is directly applicable to other short-range interactions such as rheology and hydrodynamics, and potentially to long-range interactions such as electricity and magnetism through the exchange of virtual particles.

Benefits of the MIPS simplification include significant reductions in the user effort to set up a simulation, in the number of parameters required to train a model, and in the size of the code and the effort required to accelerate a computation and to port it to new computational architectures.

The latter point contains the most significant implication, the prospect of a corresponding hardware simplification. Aspects of existing high performance computing architectures are not just wasted on MIPS, they are obstacles to its efficient implementation. Their memory hierarchy and interconnect topology serve as bottlenecks to particle propagation and interaction; what is needed is a much simpler 3D lattice of locally connected particle processors [49]. This reduces the requirement to introduce emerging device technologies for a range of high-performance computing applications, with the potential to reduce one of the most significant scaling limits, power consumption, by orders of magnitude [50].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We gratefully acknowledge extensive valuable conversations with Vesselin Yamakov and Nelson Vieira de Carvalho that posed many of the problems studied here and helped guide their investigation. This work was funded under DARPA award HR00112090066, an AWS Machine Learning Research Award, and by the CBA Consortium.

References

[1] D.E. Shaw, R.O. Dror, J.K. Salmon, J. Grossman, K.M. Mackenzie, J.A. Bank, C. Young, M.M. Deneroff, B. Batson, K.J. Bowers, et al., Millisecond-scale molecular dynamics simulations on Anton, in: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, 2009, pp. 1–11.
[2] E. Oran, C. Oh, B. Cybyk, Direct simulation Monte Carlo: recent advances and applications, Annu. Rev. Fluid Mech. 30 (1998) 403.
[3] M. Jebahi, D. André, I. Terreros, I. Iordanoff, Discrete Element Method to Model 3D Continuous Materials, John Wiley & Sons, 2015.
[4] P. Espanol, P.B. Warren, Perspective: Dissipative particle dynamics, J. Chem. Phys. 146 (15) (2017) 150901.
[5] A. Mohamad, Lattice Boltzmann Method, Vol. 70, Springer, 2011.
[6] J.-P. Rivet, J.P. Boon, Lattice Gas Hydrodynamics, in: Cambridge Nonlinear Science Series, Cambridge University Press, 2001, http://dx.doi.org/10.1017/CBO9780511524707.
[7] J. Fern, A. Rohe, K. Soga, E. Alonso, The Material Point Method for Geotechnical Engineering: A Practical Guide, CRC Press, 2019.
[8] S. Psakhie, Y. Horie, G. Ostermeyer, S.Y. Korostelev, A.Y. Smolin, E. Shilko, A. Dmitriev, S. Blatnik, M. Špegel, S. Zavšek, Movable cellular automata method for simulating materials with mesostructure, Theor. Appl. Fract. Mech. 37 (1–3) (2001) 311–334.
[9] I.F. Sbalzarini, J.H. Walther, M. Bergdorf, S.E. Hieber, E.M. Kotsalis, P. Koumoutsakos, PPM–A highly efficient parallel particle–mesh library for the simulation of continuum systems, J. Comput. Phys. 215 (2) (2006) 566–588.
[10] E. Madenci, E. Oterkus, Peridynamic theory, in: Peridynamic Theory and its Applications, Springer, 2014, pp. 19–43.
[11] M. Müller, B. Heidelberger, M. Hennix, J. Ratcliff, Position based dynamics, J. Vis. Commun. Image Represent. 18 (2) (2007) 109–118.
[12] C.A.D. Fraga Filho, C.A.D. Fraga Filho, Castro, Smoothed Particle Hydrodynamics, Springer, 2019.
[13] T. Jamil, RISC versus CISC, IEEE Potentials 14 (3) (1995) 13–16.
[14] V. Rokhlin, Rapid solution of integral equations of classical potential theory, J. Comput. Phys. (ISSN: 0021-9991) 60 (2) (1985) 187–207, http://dx.doi.org/10.1016/0021-9991(85)90002-6.
[15] J.H. Conway, N.J.A. Sloane, Sphere Packings, Lattices and Groups, Vol. 290, Springer Science & Business Media, 2013.
[16] A.P. Thompson, H.M. Aktulga, R. Berger, D.S. Bolintineanu, W.M. Brown, P.S. Crozier, P.J. in't Veld, A. Kohlmeyer, S.G. Moore, T.D. Nguyen, et al., LAMMPS-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales, Comput. Phys. Comm. 271 (2022) 108171.
[17] CUDA, 2013, https://developer.nvidia.com/cuda-toolkit, accessed July 2023.
[18] NASTRAN, 2023, https://github.com/nasa/NASTRAN-95, accessed July 2023.
[19] JAX, 2023, https://github.com/google/jax, accessed July 2023.
[20] Taichi, 2023, https://github.com/taichi-dev/taichi, accessed July 2023.
[21] N.A. Gershenfeld, N. Gershenfeld, The Nature of Mathematical Modeling, Cambridge University Press, 1999.
[22] Fusion 360, 2023, https://www.autodesk.com/products/fusion-360/overview, accessed July 2023.
[23] B. Jenett, C. Cameron, F. Tourlomousis, A.P. Rubio, M. Ochalek, N. Gershenfeld, Discretely assembled mechanical metamaterials, Sci. Adv. 6 (47) (2020) eabc9943.
[24] Standard test method for tensile properties of plastics, 2014, http://dx.doi.org/10.1520/D0638-14.
[25] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.
[26] N. Hansen, c-cmaes, 2023, https://github.com/CMA-ES/c-cmaes.
[27] Y. Hu, J. Liu, A. Spielberg, J.B. Tenenbaum, W.T. Freeman, J. Wu, D. Rus, W. Matusik, ChainQueen: A real-time differentiable physical simulator for soft robotics, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 6265–6271, http://dx.doi.org/10.1109/ICRA.2019.8794333.
[28] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[29] R.W. Zwanzig, Memory effects in irreversible thermodynamics, Phys. Rev. 124 (1961) 983–992.
[30] C. Hijón, P. Español, E. Vanden-Eijnden, R. Delgado-Buscalioni, Mori-zwanzig formalism as a practical computational tool, Faraday Discuss. 144 (2010) 301–322, discussion 323–45, 467–81.
[31] R.A. Johnson, Relationship between two-body interatomic potentials in a Lattice model and elastic constants, Phys. Rev. B 6 (1972) 2094–2100, http://dx.doi.org/10.1103/PhysRevB.6.2094.
[32] M. Born, K. Huang, M. Lax, Dynamical theory of crystal lattices, Amer. J. Phys. 23 (7) (1955) 474.
[33] J. Tersoff, New empirical approach for the structure and energy of covalent systems, Phys. Rev. B 37 (1988) 6991–7000, http://dx.doi.org/10.1103/PhysRevB.37.6991, URL https://link.aps.org/doi/10.1103/PhysRevB.37.6991.
[34] D.W. Brenner, Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films, Phys. Rev. B 42 (1990) 9458–9471, http://dx.doi.org/10.1103/PhysRevB.42.9458, URL https://link.aps.org/doi/10.1103/PhysRevB.42.9458.
[35] M.Z. Bazant, E. Kaxiras, J.F. Justo, Environment-dependent interatomic potential for bulk silicon, Phys. Rev. B 56 (1997) 8542–8552, http://dx.doi.org/10.1103/PhysRevB.56.8542, URL https://link.aps.org/doi/10.1103/PhysRevB.56.8542.
[36] N.I. Landy, S. Sajuyigbe, J.J. Mock, D.R. Smith, W.J. Padilla, Perfect metamaterial absorber, Phys. Rev. Lett. 100 (2008) 207402, http://dx.doi.org/10.1103/PhysRevLett.100.207402, URL https://link.aps.org/doi/10.1103/PhysRevLett.100.207402.
[37] K. Pack, M. Luo, T. Wierzbicki, Sandia fracture challenge: blind prediction and full calibration to enhance fracture predictability, Int. J. Fract. 186 (1) (2014) 155–175.
[38] R. Spatschek, E. Brener, A. Karma, Phase field modeling of crack propagation, Phil. Mag. 91 (1) (2011) 75–95.
[39] COMSOL holed plate application example, 2023, https://www.comsol.com/model/brittle-fracture-of-a-holed-plate-89321, accessed July 2023.
[40] B.L. Boyce, S.L. Kramer, H.E. Fang, T.E. Cordova, M.K. Neilsen, K. Dion, A.K. Kaczmarowski, E. Karasz, L. Xue, A.J. Gross, et al., The Sandia Fracture Challenge: blind round robin predictions of ductile tearing, Int. J. Fract. 186 (2014) 5–68.
[41] S. Zhou, T. Rabczuk, X. Zhuang, Phase field modeling of quasi-static and dynamic crack propagation: COMSOL implementation and case studies, Adv. Eng. Softw. 122 (2018) 31–49.
[42] M. Ambati, T. Gerasimov, L. De Lorenzis, A review on phase-field models of brittle fracture and a new fast hybrid formulation, Comput. Mech. 55 (2015) 383–405.
[43] C. Miehe, M. Hofacker, F. Welschinger, A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits, Comput. Methods Appl. Mech. Engrg. 199 (45–48) (2010) 2765–2778.
[44] R. Krueger, An Approach to Assess Delamination Propagation Simulation Capabilities in Commercial Finite Element Codes, Tech. Rep. TM-2008-215123, NASA, 2008.
[45] Y. Hu, N. De Carvalho, E. Madenci, Peridynamic modeling of delamination growth in composite laminates, Compos. Struct. 132 (2015) 610–620.
[46] R. Talreja, J. Varna, Modeling Damage, Fatigue and Failure of Composite Materials, Elsevier, 2015.
[47] V.K. Goyal, E.R. Johnson, C.G. Davila, Irreversible constitutive law for modeling the delamination process using interfacial surface discontinuities, Compos. Struct. 65 (3–4) (2004) 289–305.
[48] J. Fish, G.J. Wagner, S. Keten, Mesoscopic and multiscale modelling in materials, Nat. Mater. 20 (6) (2021) 774–786.
[49] Z. Fredin, J. Zemanek, C. Blackburn, E. Strand, A. Abdel-Rahman, P. Rowles, N. Gershenfeld, Discrete integrated circuit electronics (DICE), in: 2020 IEEE High Performance Extreme Computing Conference (HPEC), IEEE, 2020, pp. 1–8.
[50] L.C. Blackburn, E. Golden, A. Wynn, A. Wagner, N. Gershenfeld, Design and simulation of phase synchronizer for adiabatic quantum flux parametron circuits, IEEE Trans. Appl. Supercond. 33 (5) (2023) 1–5.

Erik Strand is a Ph.D. student at the MIT Center for Bits and Atoms. His research interests include particle-based physics simulation methods, accelerated computing, and silicon compilers for emerging computational architectures. Prior to CBA, Erik studied mathematics and physics at the University of Chicago, and worked as a computational geometry software engineer in the San Francisco Bay Area. Outside of work you will find him playing music, working on interactive digital art projects, and enjoying the outdoors.

Filippos Tourlomousis is the Scientific Coordinator of The Laboratory for Autonomous Science at the National Center for Scientific Research "Demokritos". His research interests include the automation of science and robotics infrastructure for self-driving materials engineering labs of the future (a.k.a. "robot scientists"). Before that, he studied Experimental Fluid Mechanics at the von Karman Institute for Fluid Dynamics in Belgium and then completed his PhD in Advanced Biomanufacturing at the Stevens Institute of Technology, where he was awarded the Excellence in Research Award for his contributions in the area of rheology and processing of complex fluids. After that, he worked as A PostDoc Associate at The MIT Center for Bits and Atoms. His research has been featured in MIT News, The Economist, Wired, and the Vanguard of General Electrics. Outside of academia, he is an active entrepreneur and DJ in electronic music.

Prof. Neil Gershenfeld is the Director of MIT's Center for Bits and Atoms. He has been elected a Member of the National Academy of Engineering, a Fellow of the American Association for the Advancement of Science and of the American Physical Society, has been named one of Scientific American's 50 leaders in science and technology, as one of 40 Modern-Day Leonardos by the Museum of Science and Industry, one of Popular Mechanic's 25 Makers, has been selected as a CNN/Time/Fortune Principal Voice, and by Prospect/Foreign Policy as one of the top 100 public intellectuals. He is the author of books including Designing Reality, Fab, When Things Start To Think, The Nature of Mathematical Modeling, and The Physics of Information Technology. He is a co-founder of the Interspecies Internet, of the Science and Entertainment Exchange, and of a global network of over two thousand fab labs in 125 countries. He has a BA in Physics with High Honors from Swarthmore College, a Ph.D. in Applied Physics from Cornell University, honorary doctorates from Swarthmore College, Strathclyde University and the University of Antwerp, was a Junior Fellow of the Harvard University Society of Fellows, and a member of the research staff at Bell Labs.