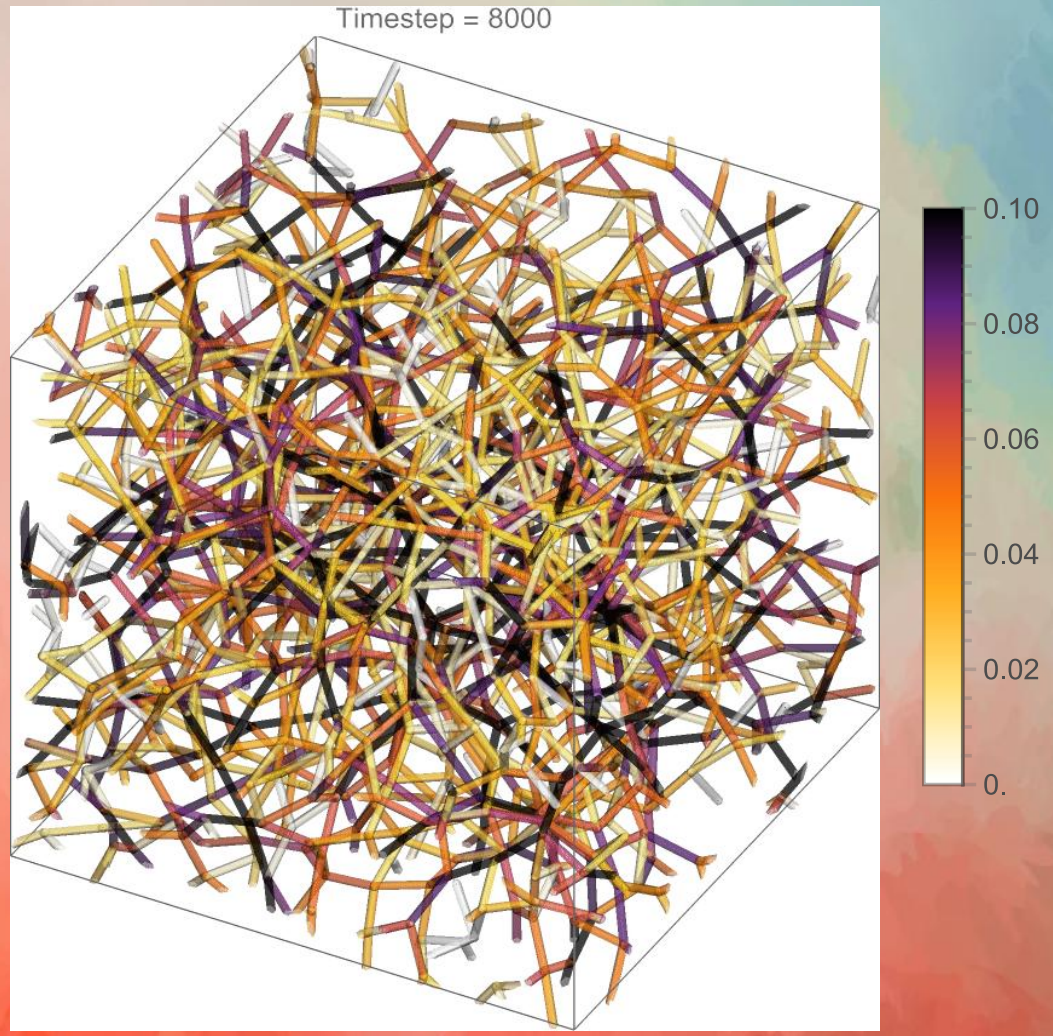


ElasticNetworks.jl

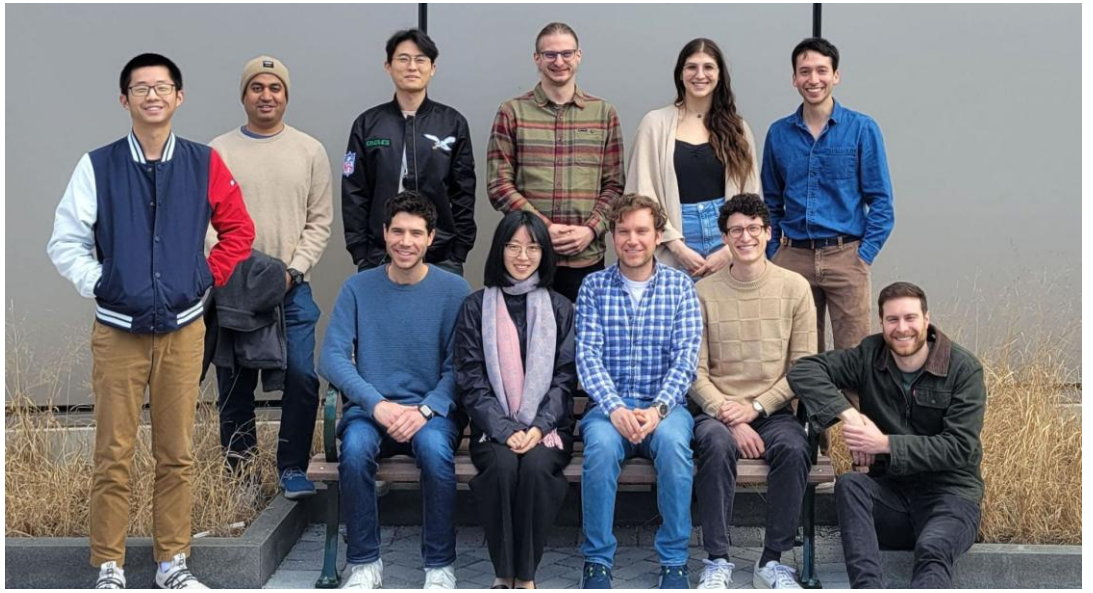
Infrastructure for metamaterials research & design

Haina Wang

University of Pennsylvania

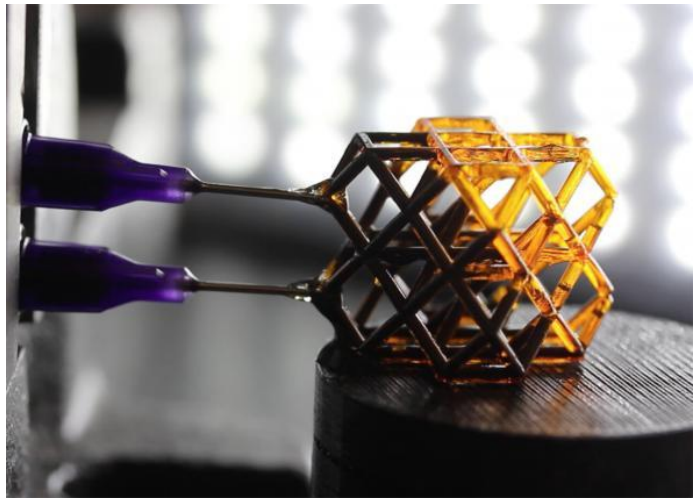
About me

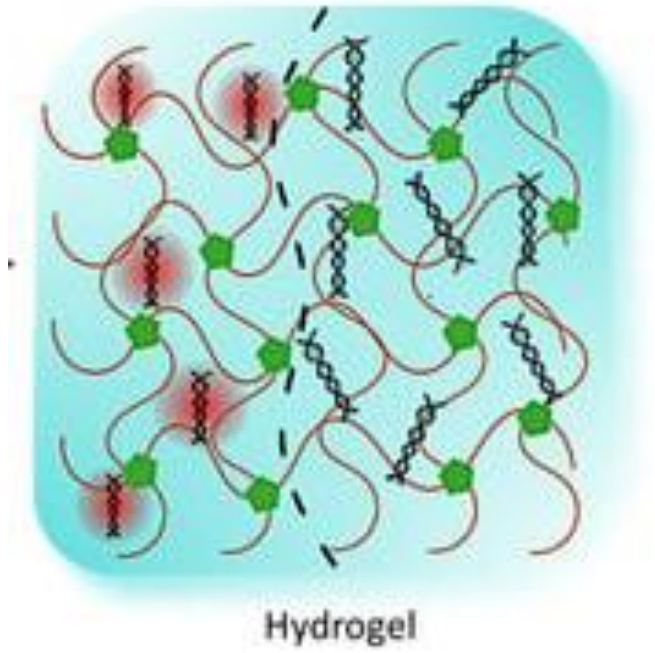
- Postdoc fellow at Center for Soft and Living Matter @ Penn
- Princeton *24 – Statistical mechanics, chemical physics, metamaterials design
- Penn – Biophysics, biodynamics modeling
- Julia programmer since 2021. Monte-Carlo and molecular dynamics simulations, optimization, geometry

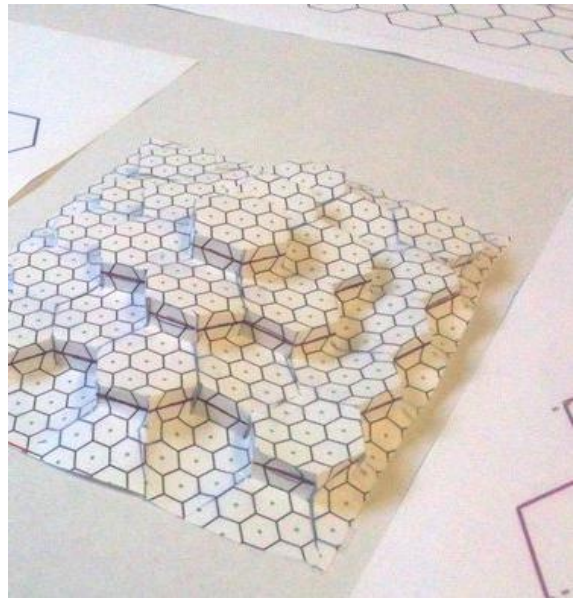


What are metamaterials?

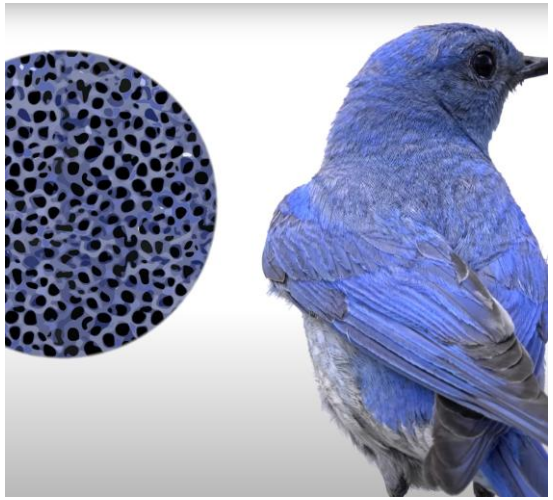
- **Natural or engineered materials whose properties arise from meso- or macro-scale structure—not chemical composition on the molecular and atomic level**
  - 3D printed structures
  - Knitted fabrics
  - Hydrogels
  - Kirigami

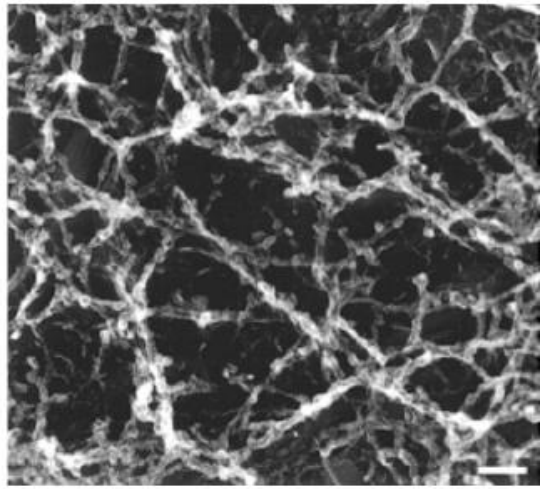


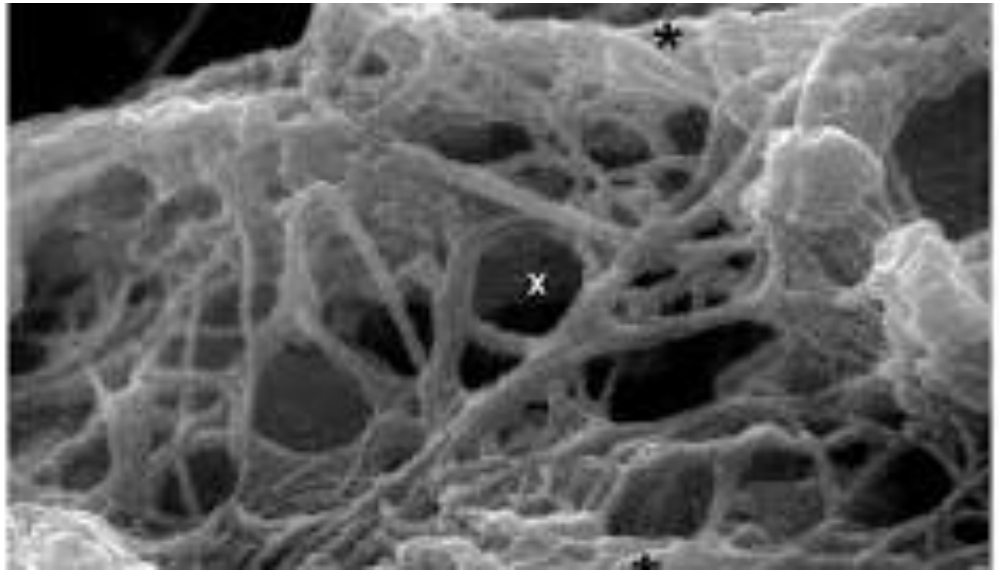




Biology has metamaterials

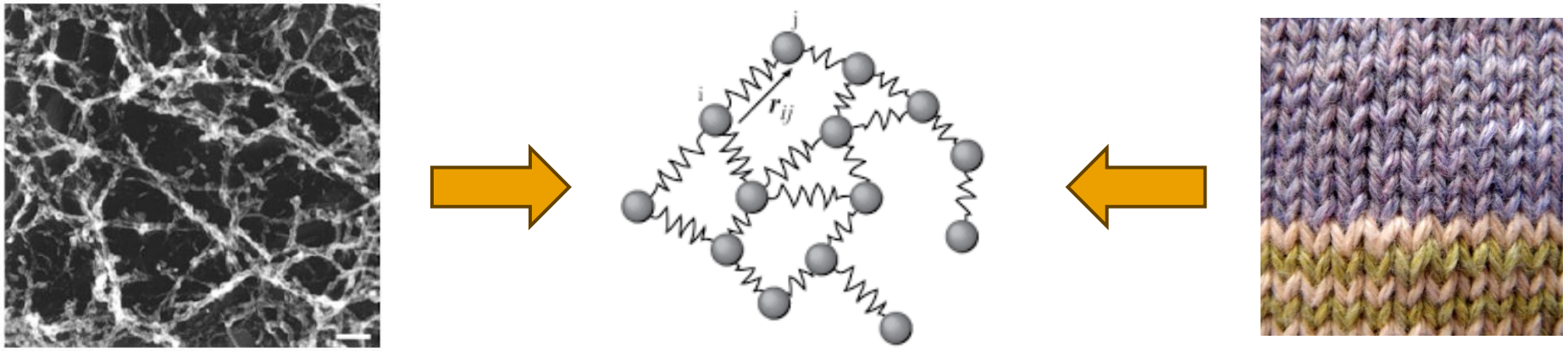






- The actin cortex, a network made of actin filaments, lies just beneath the cell membrane. It supports cell shape and regulates mechanic responses.
- The collagen network in the extracellular matrix bind cells together in tissues.
- Complex nanostructures made of beta-keratin create air channels that diffract light, resulting in structural color in birds.

Mechanical metamaterials can be modeled as elastic networks



- Elastic network have been “rediscovered” many times in biology and civilizations alike…
  - Lightweight yet stiff, saves materials
  - Highly programmable structures down to individual edges and nodes
  - “Many more is more different”
- Need a general infrastructure for elastic networks.
  - See also JuSFEM.jl for continuum elastic problems.

ElasticNetworks.jl

Vision & Mission

A go-to platform for the materials science community to model, design, and study metamaterials based on elastic networks.

Features

- **Customizable topology** with Graphs.jl
- **Automatic differentiation** for exact linear-response tensors
- **Periodic boundary conditions** with deformable unit cells and flexibility to connect node i to arbitrary image of node j.
- **Topology tools** that delete and add edges, split and merge nodes
- **Creators of common networks including** cubic, diamond, Erdos-Renyi
- **Visualization** with Makie
- **I/O** with JLD2.jl

What can you do with ElasticNetwork.jl?

- Compute linear response moduli.
- Explore how topology affects elasticity.
- Prototype and iterate for materials design.

📘 Quick Start

```
using ElasticNetworks

net = diamond1000(10.0, 0.05)
fig = visualize_net(net)
```

Creates a diamond network with 1000 nodes. Each edge has a strain of 0.05.
Visualize the network.

Data structure: The Network struct

ElasticNetworks.Network — Type

```
mutable struct Network
```

Models a Hookean spring network.

Fields

- `g::SimpleGraph` : Graph specifying the connectivity of the network.
- `basis::Matrix{Float64}` : Basis for the nodes.
- `points::Matrix{Float64}` : Coordinates of the nodes in space.
- `rest_lengths::Dict{Graphs.SimpleGraphs.SimpleEdge{Int64}, Float64}` : Rest lengths of the spring edges in the network.
- `image_info::Dict{Graphs.SimpleGraphs.SimpleEdge{Int64}, Vector{Int}}` : Specifies which image of node `j` node `i` is connected to in periodic boundary conditions.
- `youngs::Dict{Graphs.SimpleGraphs.SimpleEdge{Int64}, Float64}` : Young's modulus of the spring edges, defining their stiffness.

source

Demo: Create and modify networks

```
using ElasticNetworks
```
✓ 1m 36.4s

```
l = 10 #box side length
ϵ = 0.05 #edge prestrain
diamond_net = diamond1000(l, ϵ)
visualize_net(diamond_net)
```
✓ 0.3s

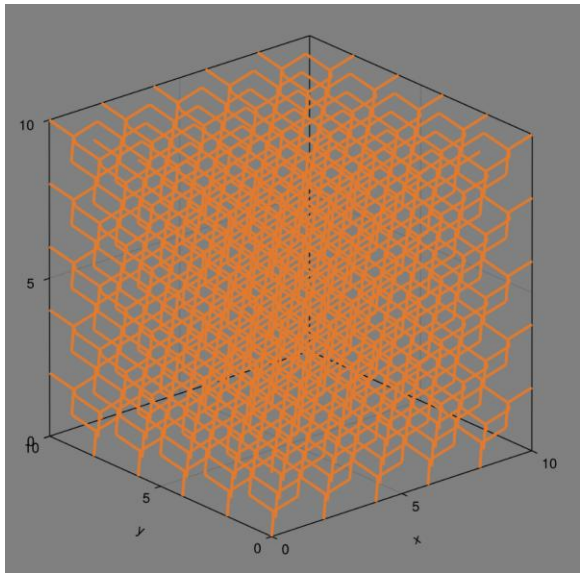
```
collect(edges(diamond_net.g))
```
✓ 1.0s

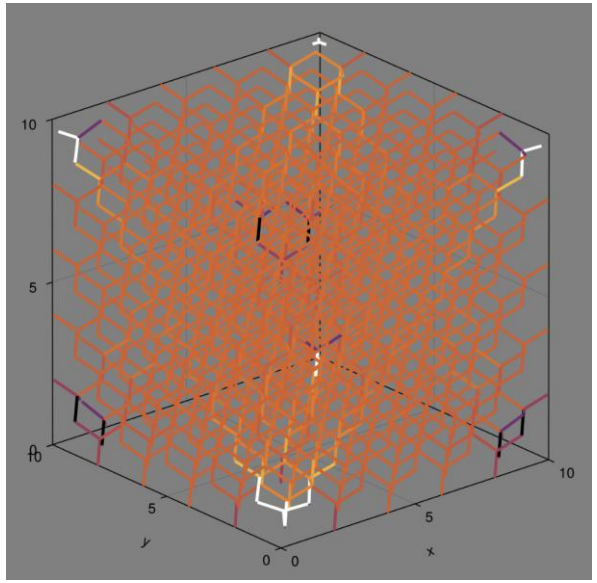
```
2000-element Vector{Graphs.SimpleGraphs.SimpleEdge{Int64}}:
 Edge 1 => 501
 Edge 1 => 746
 Edge 1 => 775
 Edge 1 => 980
```

```
ElasticNetworks.rem_edge!(diamond_net, 1, 501)
```
✓ 0.2s

```
1.0
```

```
relax!(diamond_net)
visualize_net(diamond_net)
```

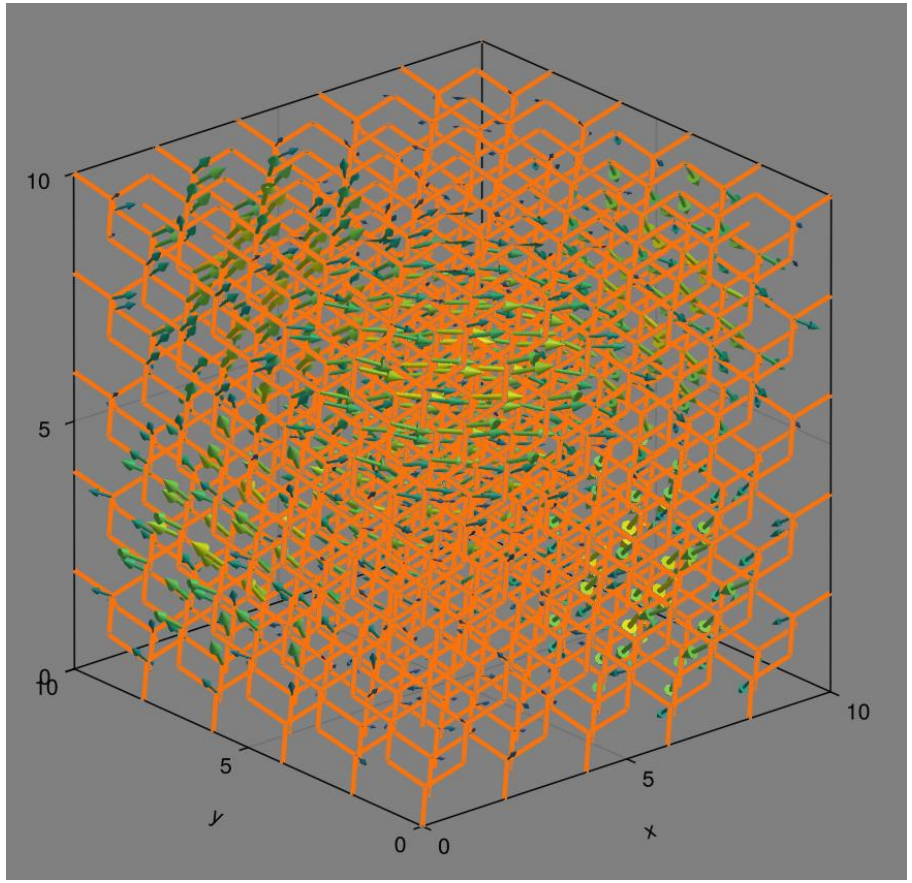




Demo: Compute and visualize vibrational modes

```
energy_hessian(diamond_net)
✓ 0.0s

3000×3000 Matrix{Float64}:
 177.054        -1.55599e-9   -1.5558e-9  …    0.0            0.0
  -1.55599e-9  177.054        -1.5558e-9        0.0            0.0
  -1.5558e-9    -1.5558e-9   177.054            0.0            0.0
   0.0           0.0           0.0              0.0            0.0
   0.0           0.0           0.0              0.0            0.0
   0.0           0.0           0.0         …    0.0            0.0
   0.0           0.0           0.0              0.0            0.0
   0.0           0.0           0.0              0.0            0.0
   0.0           0.0           0.0              0.0            0.0
   0.0           0.0           0.0              0.0            0.0
   ⋮                                    ⋱
   0.0           0.0           0.0              0.0            0.0
   0.0           0.0           0.0              0.0            0.0
   0.0           0.0           0.0              0.0            0.0
   0.0           0.0           0.0              0.0            0.0
   0.0           0.0           0.0         …    0.0            0.0
   0.0           0.0           0.0              0.0            0.0
   0.0           0.0           0.0              2.30735e-10   -1.24403e-9
   0.0           0.0           0.0            177.054          2.30585e-10
   0.0           0.0           0.0              2.30585e-10  177.054
```



Compute the linear response with infinitesimal strain theory

- Strain is the deformation to the material.
- Stress is the force per area as a result of the strain = first derivative of elastic energy to the strain.
- Elastic modulus is stress/strain = **second derivative of elastic energy to the strain**.
- The setting is perfect for **automatic differentiation**!

To calculate the edge strains in eq. A1 we take a two step approach where we first calculate the affine strain induced by the strain tensor ϵ and then obtain the non-affine strain from the forces that result from the affine deformation. The non-affine strain is obtained from

$$M_{\alpha\beta}\delta r_{\alpha}^{NA} = f_{\beta} \qquad \text{(A4)}$$

Marco A. Galvani Cunha

Galvani Cunha, Crocker, Liu, PRRearch (2024)

```julia
function moduli(net::Network)
    relax!(net)
    basis, points, edge_nodes, rls, iis, youngs = net_info_primitive(net)
    deformed_bases = Dict{String, Function}()
    deformed_bases["1111"] = ϵ -> ([ϵ 0 0; 0 0 0; 0 0 0] + I)*basis
    deformed_bases["2222"] = ϵ -> ([0 0 0; 0 ϵ 0; 0 0 0] + I)*basis
    deformed_bases["3333"] = ϵ -> ([0 0 0; 0 0 0; 0 0 ϵ] + I)*basis
    deformed_bases["1212"] = ϵ -> ([0 ϵ 0; ϵ 0 0; 0 0 0] + I)*basis
    deformed_bases["1313"] = ϵ -> ([0 0 ϵ; 0 0 0; ϵ 0 0] + I)*basis
    deformed_bases["2323"] = ϵ -> ([0 0 0; 0 0 ϵ; 0 ϵ 0] + I)*basis
    deformed_bases["1122"] = ϵ -> ([ϵ 0 0; 0 ϵ 0; 0 0 0] + I)*basis
    deformed_bases["1133"] = ϵ -> ([ϵ 0 0; 0 0 0; 0 0 ϵ] + I)*basis
    deformed_bases["2233"] = ϵ -> ([0 0 0; 0 ϵ 0; 0 0 ϵ] + I)*basis
    n = nv(net.g)
    H = zeros(3*n, 3*n)
    hessian!(H, basis, points, edge_nodes, rls, iis, youngs)
    function make_curry_function(component)
        function curry(ϵ)
            deformed_basis = deformed_bases[component](ϵ[1])
            F = -gradient(deformed_basis, points, edge_nodes, rls, iis, youngs)
            nonaffine_displacements = qr(H, Val(true)) \ F
            return elastic_energy(deformed_basis, points + nonaffine_displacements, edge_nodes, rls, iis, youngs)
        end
        return curry
    end
    v = det(basis)
    c1111 = ForwardDiff.hessian(make_curry_function("1111"), [0])[1]/v
    c2222 = ForwardDiff.hessian(make_curry_function("2222"), [0])[1]/v
    c3333 = ForwardDiff.hessian(make_curry_function("3333"), [0])[1]/v
    c1212 = ForwardDiff.hessian(make_curry_function("1212"), [0])[1]/(4*v)
    c1313 = ForwardDiff.hessian(make_curry_function("1313"), [0])[1]/(4*v)
    c2323 = ForwardDiff.hessian(make_curry_function("2323"), [0])[1]/(4*v)
    c1122 = ForwardDiff.hessian(make_curry_function("1122"), [0])[1]/(2*v) - (c1111 + c2222)/2
    c1133 = ForwardDiff.hessian(make_curry_function("1133"), [0])[1]/(2*v) - (c1111 + c3333)/2
    c2233 = ForwardDiff.hessian(make_curry_function("2233"), [0])[1]/(2*v) - (c2222 + c3333)/2
    B = 1/9*(c1111 + c2222 + c3333 + 2*(c1122 + c1133 + c2233))
    G = 1/15*(3*(c1212 + c1313 + c2323) + c1111 + c2222 + c3333 - c1122 - c1133 - c2233)
    println("B = $B \n G = $G \n c1111 = $c1111 \n c2222 = $c2222 \n c3333 = $c3333 \n c1212 = $c1212 \n c1313 = $c1313 \n c2323 = $c2323 \n c1122 = $c1122 \n c1133 = $c113
    return B, G, c1111, c2222, c3333, c1212, c1313, c2323, c1122, c1133, c2233
end
```

These are different ways to deform the box infinitesimally...

Minimize energy under the deformed basis:
A single step of Newton-Raphson suffices because strain is infinitesimal!

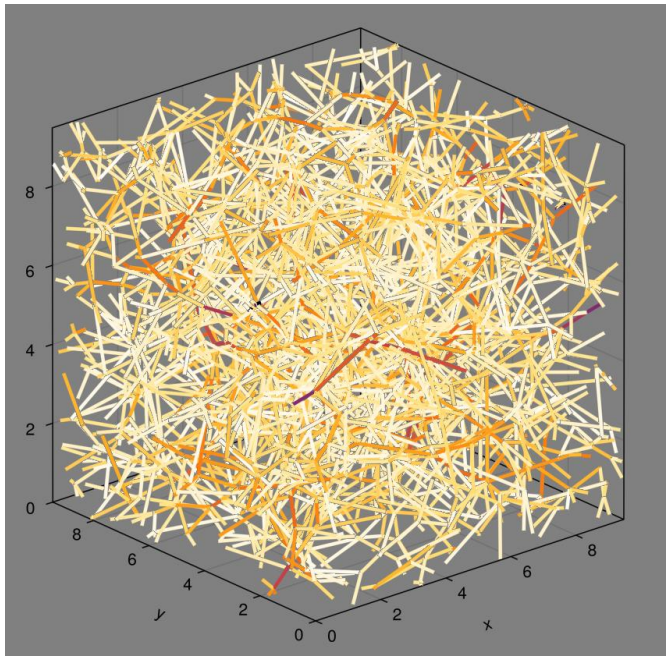
Compute the components of the stiffness tensor

Compute bulk and shear moduli from the components

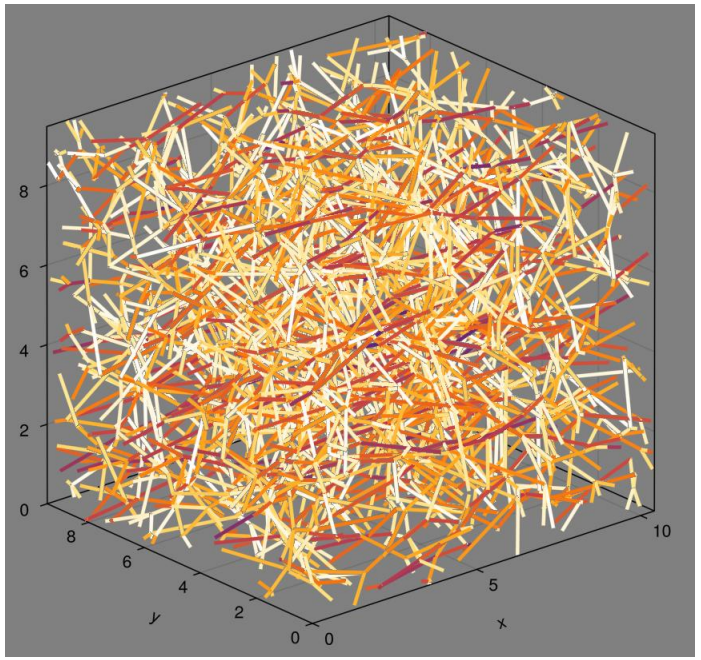
Demo: Strain stiffening

- For many biopolymer networks, their elastic modulus increases when the material is subject to **finite** strain, even if all edge components can be considered as Hookean springs.

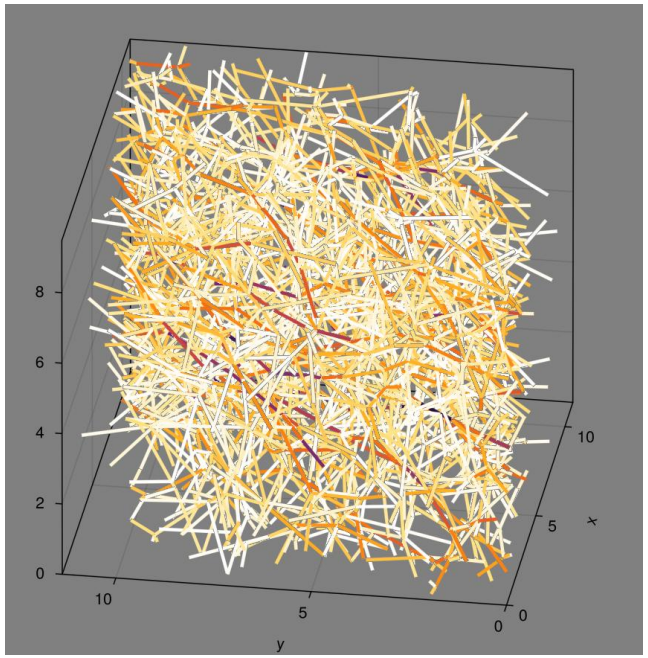
Undeformed

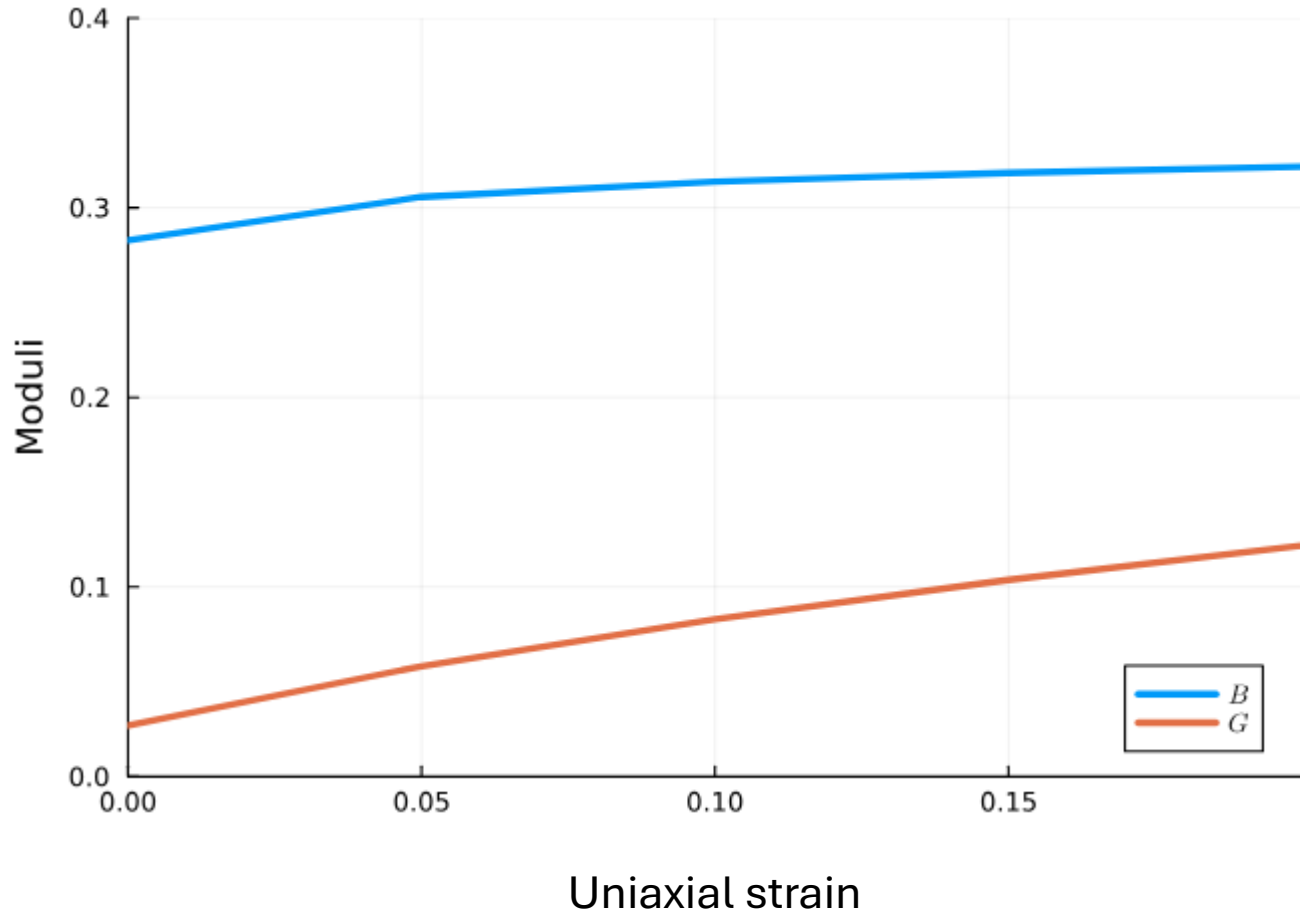


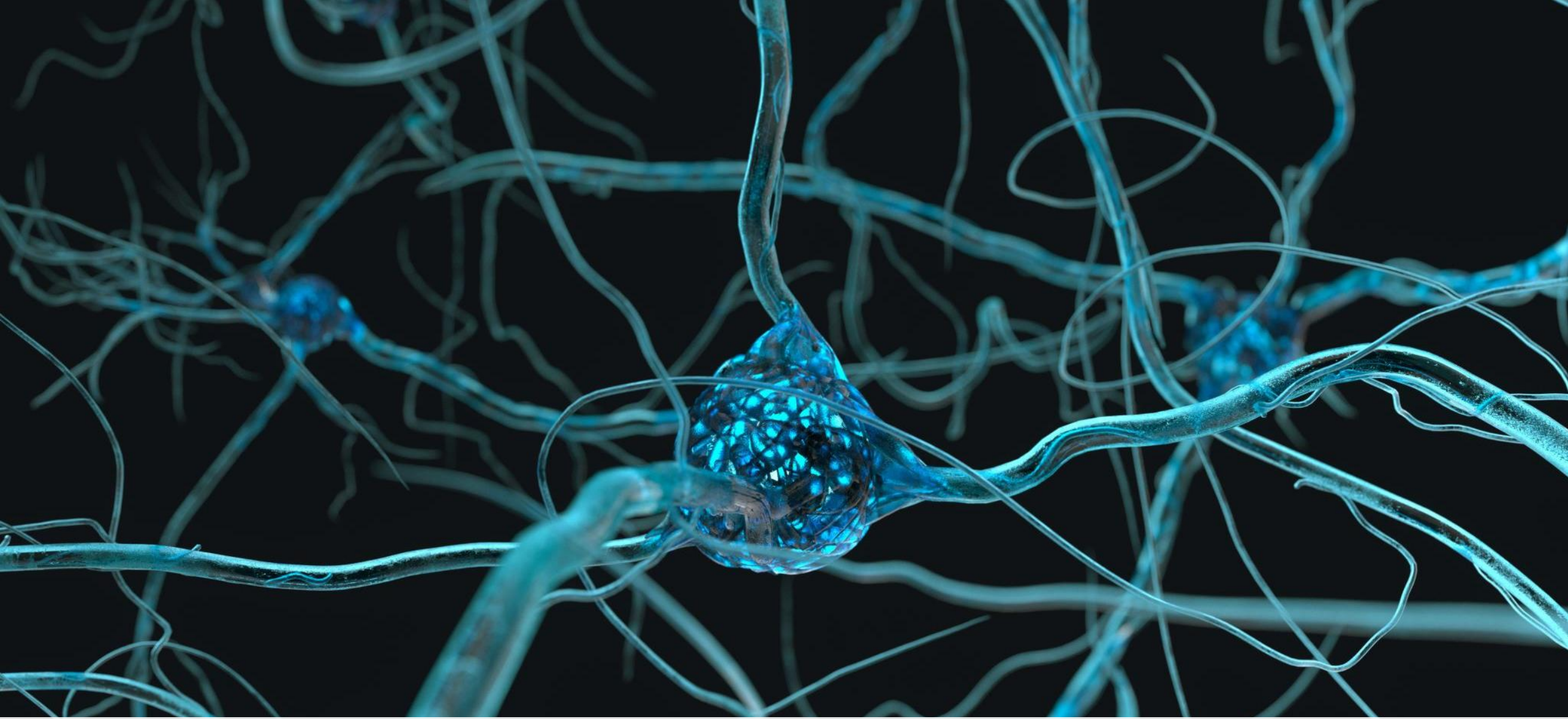
10% dilation



20% shear



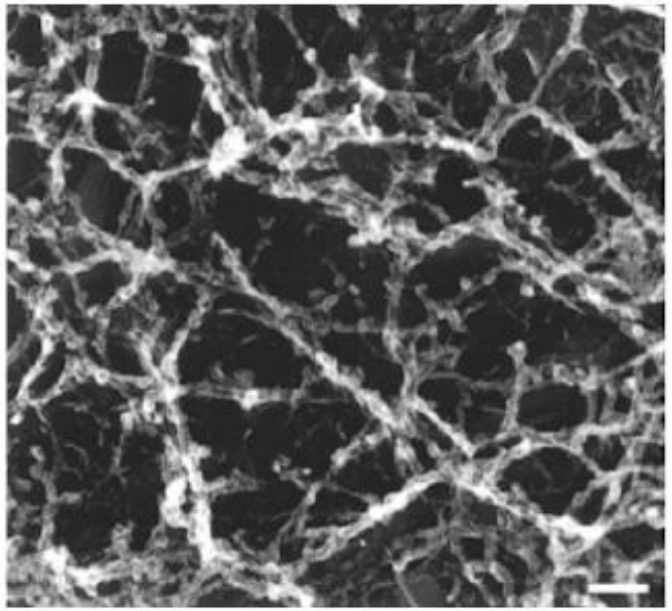

0.4
0.3
0.2
0.1
0.0
Moduli
0.00
0.05
0.10
0.15
B
G
Uniaxial strain



Research enabled by this package:
Rigidity homeostasis of the actin cortex

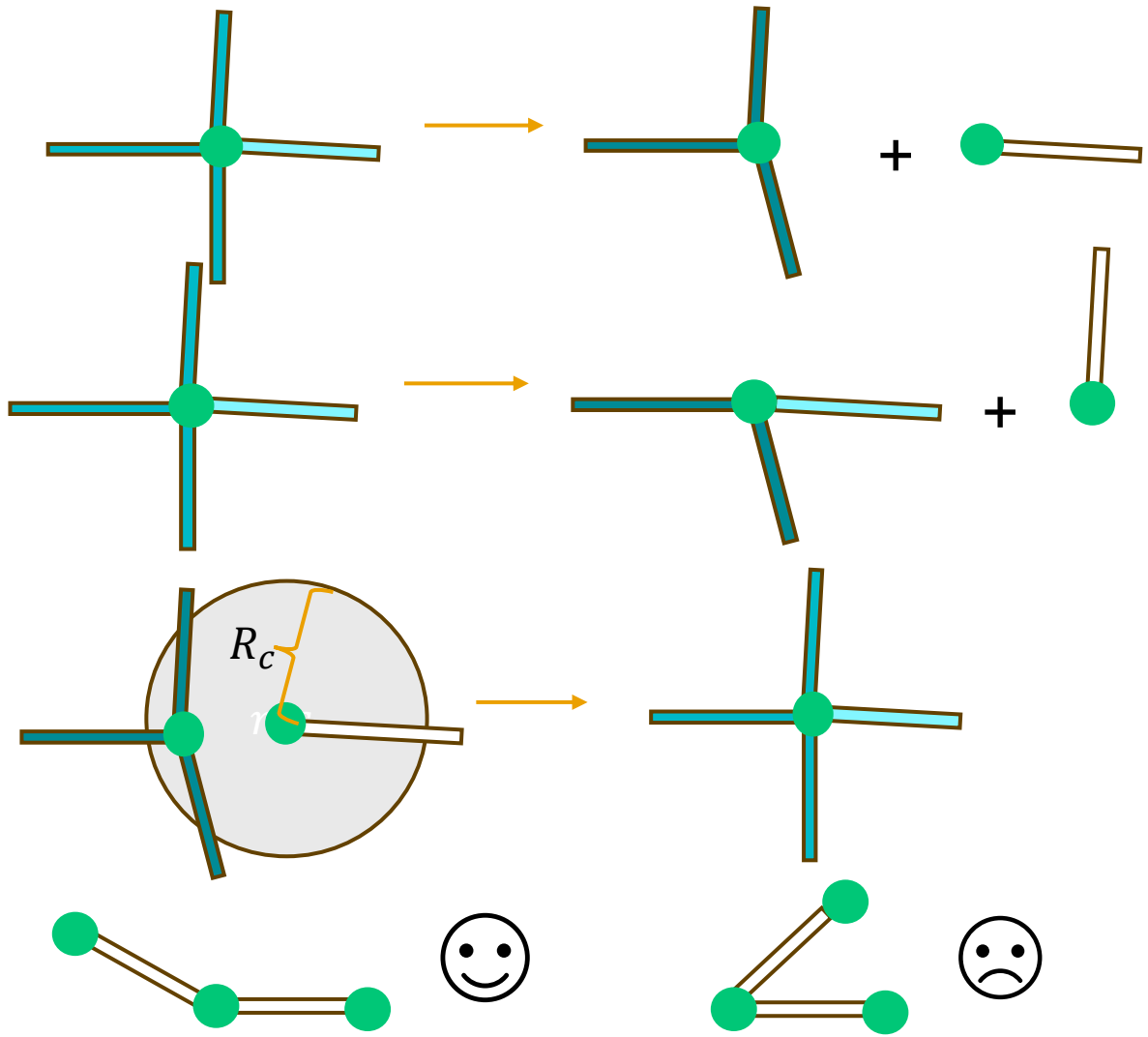
Can we learn from biology to build functional metamaterials?

- The actin cortex maintains rigidity even when its architecture constantly changes…
  - Rigid at average coordination number $3 < \bar{z} < 4$, below Maxwellian isostaticity $\bar{z} = 6$: tensegrity
  - Regulated by many actin-binding proteins: Motors, severing proteins and crosslinkers
  - Filaments and crosslinkers undergo fast remodeling (turnover); τ~seconds



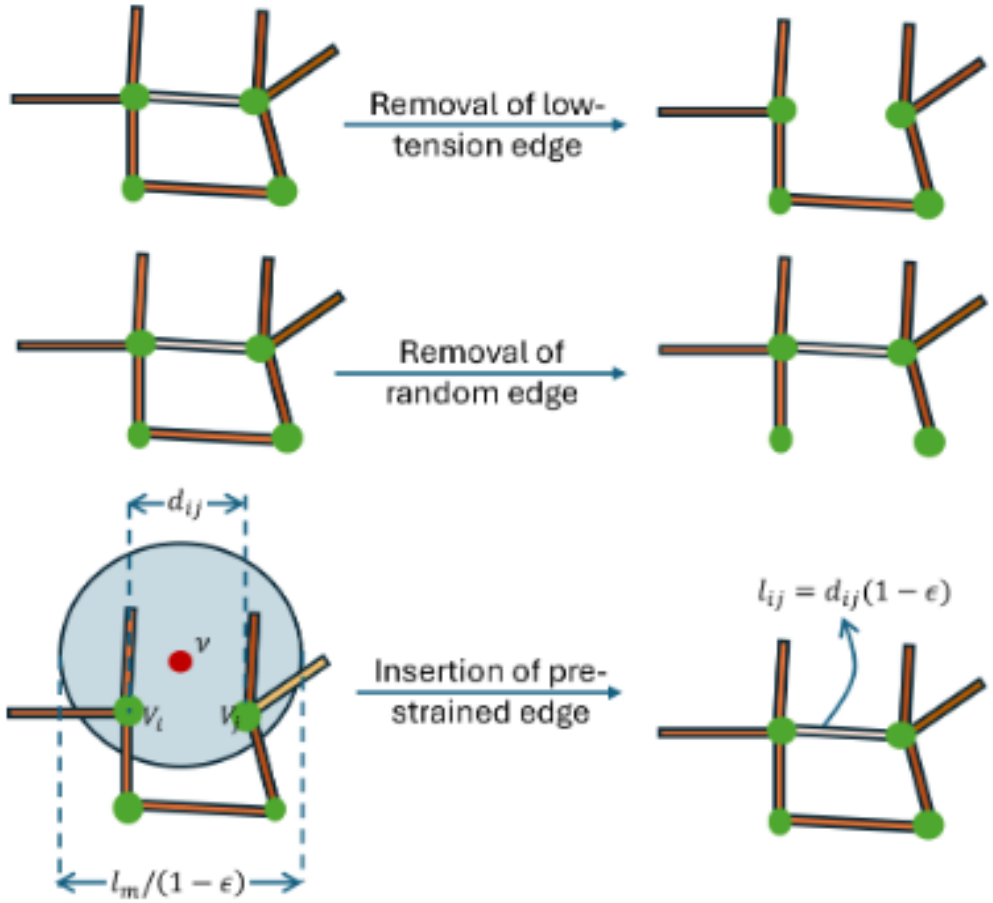
Stossel et al. *Nature* 2001

The package enables us to build network models that behave like the actin cortex.



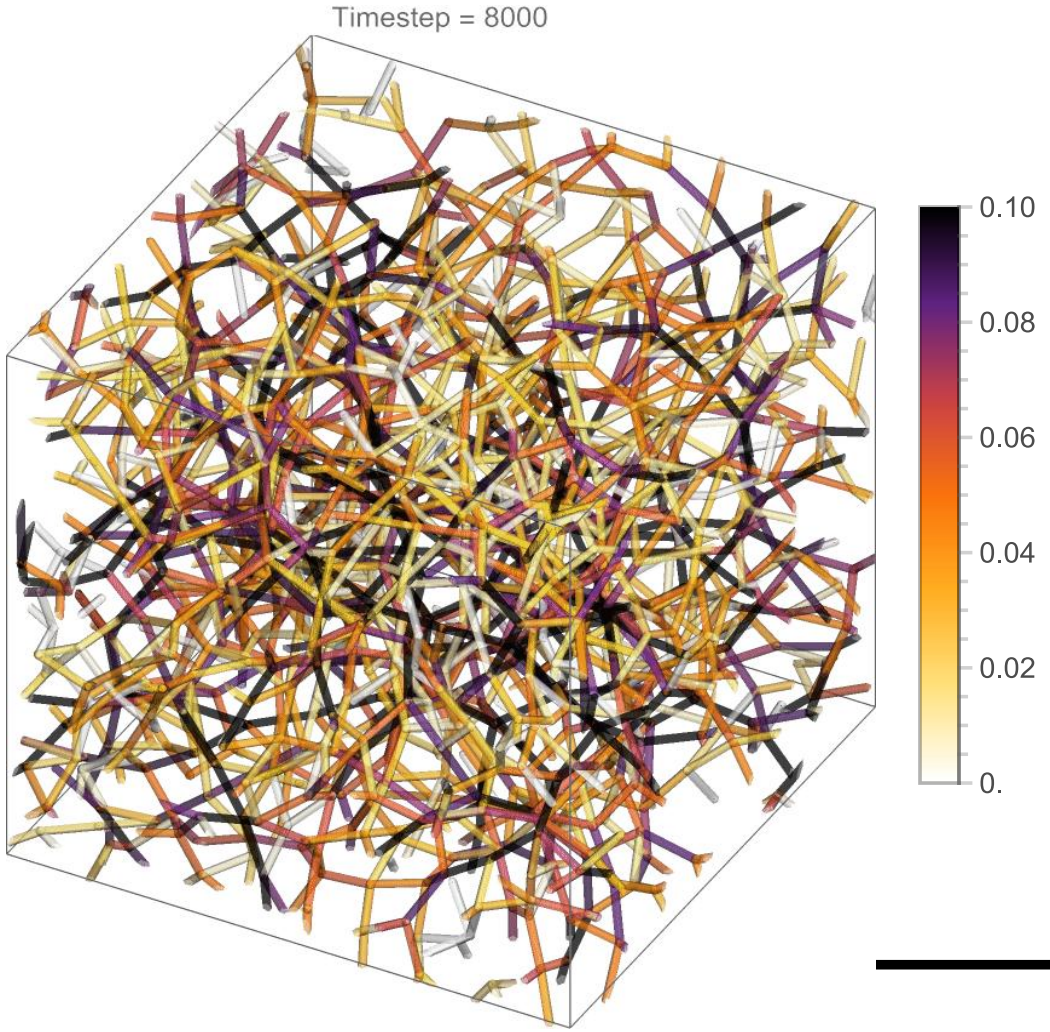
- **Force-sensitive disassembly.** A node splits with probability $k_V dt$, if the tension on one of its edges < threshold Θ
- **Small fraction of random disassembly.** All nodes can be randomly split with probability $k_r dt$, where $k_r \ll k_V$
- **Assembly + Energy injection.** A $z = 1$ node merges with its nearest $z \leq 3$ node in a capture sphere of radius R_c, with probability $k_m dt$
- **Steric repulsion.** To orient edges at large angles

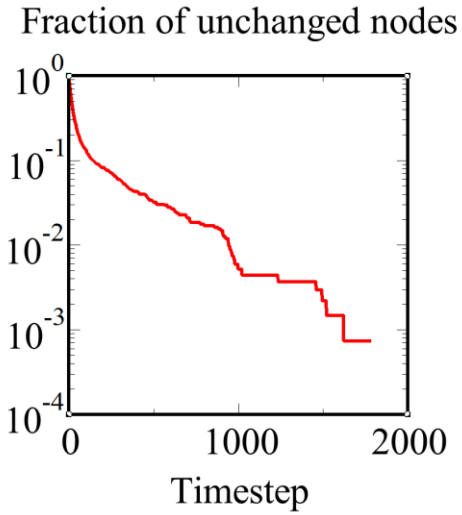
The package enables us to build network models that behave like the actin cortex.



- **Tension-inhibited pruning:** An edge under low tension is always removed.
- **Random pruning**: An edge can be randomly removed with a small probability regardless of its tension.
- **Insertion of prestrained edges.** Energy is injected.

Model achieves rigidity homeostasis with complete node turnover

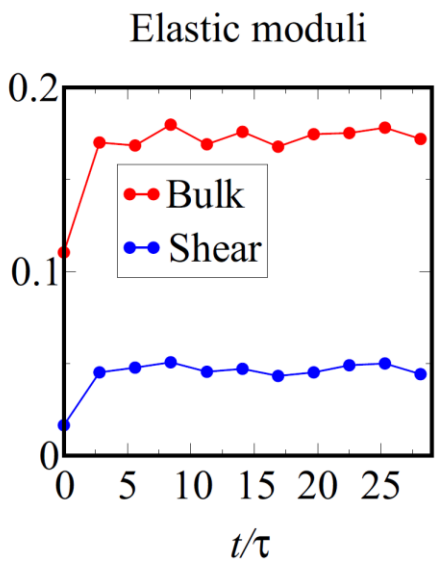


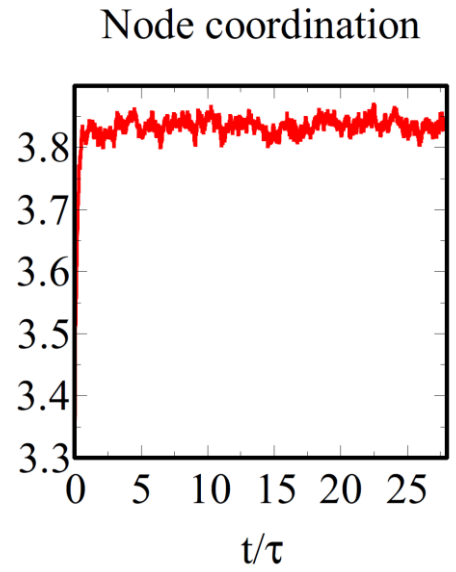


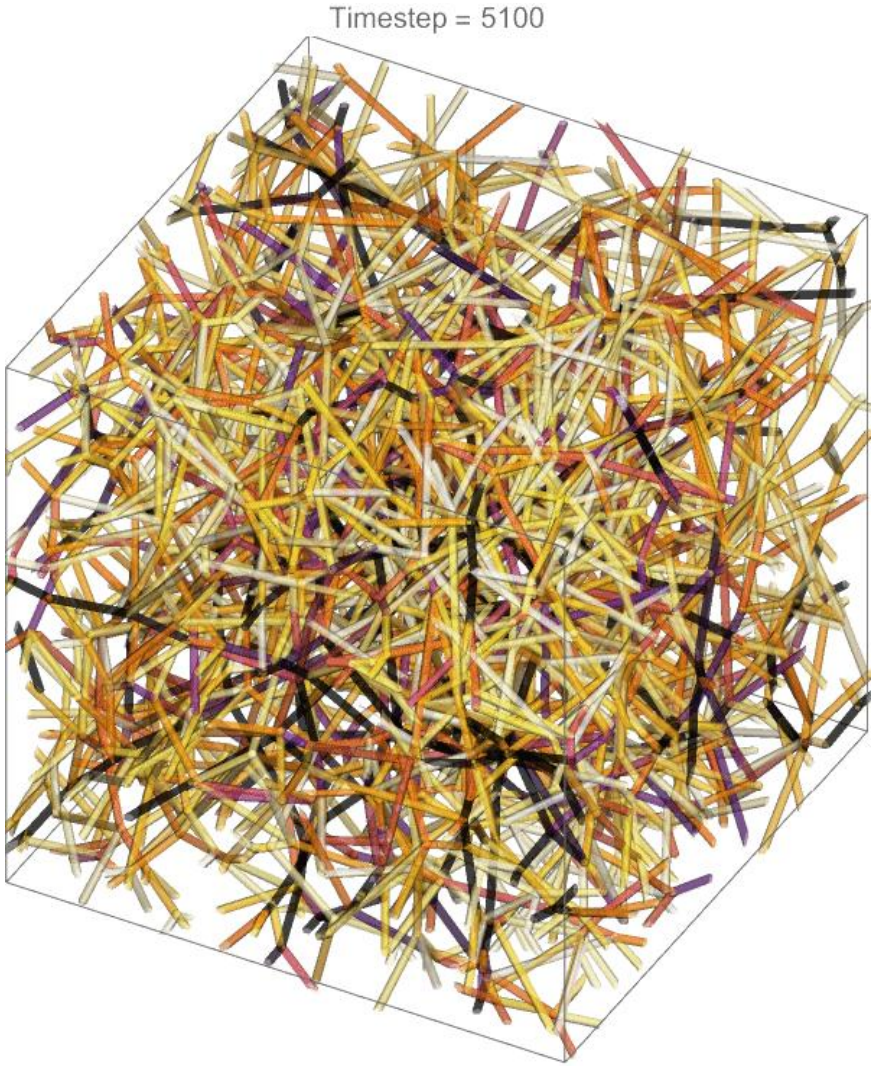
Exponential decay of node labels

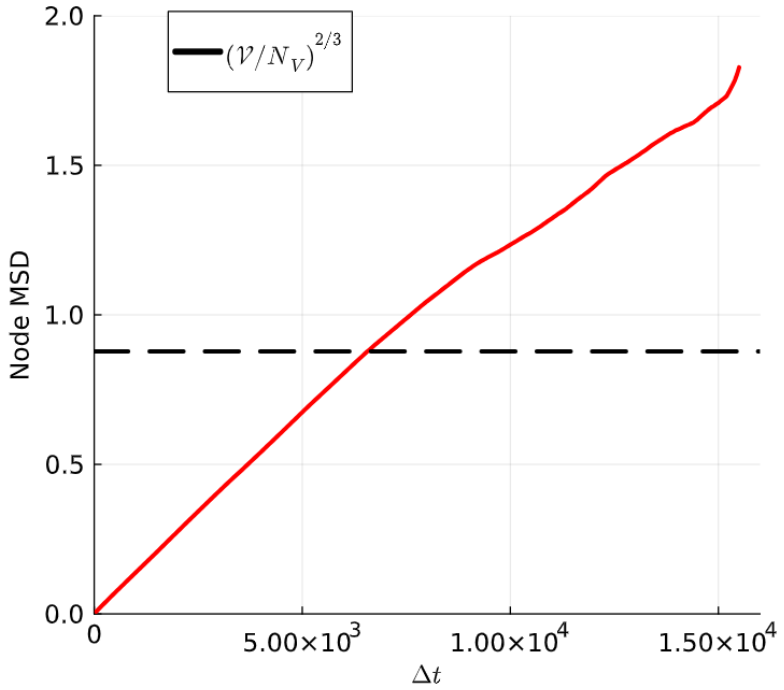
Turnover time scale τ = 355 timesteps

$t_{1/2}$ = 246 timesteps






Timestep = 5100


$(\mathcal{V}/N_V)^{2/3}$
Node MSD
2.0
1.5
1.0
0.5
0.0
0
5.00×10^3
1.00×10^4
1.50×10^4
Δt

Future work

- Other boundary conditions than periodic
- 2D networks
- Nonlinear (nonharmonic) edges
- Bending rigidity

Please consider collaborating or contributing!! 💗

Acknowledgements

Preprint on arXiv soon!

Rigidity homeostasis of the actin cortex emerging from mechano-sensitive filament and crosslinker dynamics

Haina Wang,[1] Marco A. Galvani Cunha,[1] John C. Crocker,[2] and Andrea J. Liu[1]

[1] *Department of Physics and Astronomy, University of Pennsylvania*
[2] *Department of Chemical and Biomolecular Engineering, University of Pennsylvania*

hainaresearch.com

Andrea J. Liu

John C. Crocker

Marco A. Galvani Cunha

Center for Soft & Living Matter
University of Pennsylvania

National Science Foundation
MRSEC
Materials Research Science and Engineering Centers

8 Things You Can Do

to Promote Human Rights in Science, Engineering, and Medicine

1. Highlight the Importance of Free Speech and the Dangers of Political Censorship for Work
2. Use your Expertise to Help Address Human Rights Challenges
3. Host a Discussion on Human Rights and/or Integrate Human Rights into your Teaching
4. Notify Human Rights Monitoring/Advocacy Bodies if you Suspect Rights Violations
5. Urge Institution to Assist Colleagues Fleeing Persecution Through Fellowships and Other Means
6. When Planning Travel, Learn about the State of Human Rights in your Destination
7. Write to U.S. and International Officials in Support of Colleagues
8. Stay Informed by Attending Human Rights Events

+ Ask me for this slide for you to share in your talks!

www.nationalacademies.org/8-things-you-can-do