Research Update: Converting From Linear to Exponential Springs

Daniel Miles

October 20, 2025

1 Introduction

The original Julia simulation employed a 5×5 lattice of masses connected by linear springs, both nearest-neighbor and diagonal connections.

The conversion to exponential springs introduces nonlinear dynamics while preserving the fundamental conservation laws. This document details the mathematical foundations, derivations, and implementation changes required for this conversion.

2 Linear Spring Model (Original Implementation)

2.1 Force Law

The linear spring model follows Hooke's law for small displacements:

$$\vec{F} = -k(\vec{r} - \vec{r}_0) = -k\vec{d} \tag{1}$$

where:

- $\vec{F}$ is the spring force vector
- k is the spring constant (N/m)
- $\vec{r}$ is the current position vector
- $\vec{r}_0$ is the equilibrium position vector
- $\vec{d} = \vec{r} - \vec{r}_0$ is the displacement vector

For a 2D system, this becomes:

$$\vec{F} = -k\vec{d} = -k(d_x\hat{\imath} + d_y\hat{\jmath}) \tag{2}$$

2.2 Potential Energy

The potential energy function for a linear spring is:

$$U(\vec{r}) = \frac{1}{2}k|\vec{d}|^2 = \frac{1}{2}k(d_x^2 + d_y^2) \tag{3}$$

2.3 Force-Energy Relationship Verification

We can verify the force-energy relationship using:

$$\vec{F} = -\nabla U = -\frac{\partial U}{\partial \vec{r}} \tag{4}$$

Taking the gradient:

$$\frac{\partial U}{\partial d_x} = k d_x \tag{5}$$

$$\frac{\partial U}{\partial d_y} = k d_y \tag{6}$$

Therefore:

$$\vec{F} = -k(d_x \hat{\imath} + d_y \hat{\jmath}) = -k\vec{d} \tag{7}$$

This confirms the consistency between force law and potential energy.

3 Exponential Spring Model (New Implementation)

3.1 Force Law

The exponential spring model uses a force law that grows exponentially with displacement magnitude:

$$\vec{F} = k(\exp(\alpha|\vec{d}|) - 1)\frac{\vec{d}}{|\vec{d}|} \tag{8}$$

where:

- k is the spring constant (N)
- α is the exponential decay parameter (m^{-1})
- $|\vec{d}| = \sqrt{d_x^2 + d_y^2}$ is the displacement magnitude
- $\frac{\vec{d}}{|\vec{d}|}$ is the unit vector in the displacement direction

3.2 Potential Energy Derivation

To find the potential energy function, we integrate the force law. For a radially symmetric potential, we work in terms of the displacement magnitude $r = |\vec{d}|$.

The radial component of force is:

$$F_r = k(\exp(\alpha r) - 1) \tag{9}$$

The potential energy is found by integrating:

$$U(r) = -\int F_r \, dr = -\int k(\exp(\alpha r) - 1) \, dr \tag{10}$$

Evaluating the integral:

$$U(r) = -k\left[\frac{\exp(\alpha r)}{\alpha} - r\right] + C \tag{11}$$

Setting the boundary condition $U(0) = 0$:

$$0 = -k\left[\frac{1}{\alpha} - 0\right] + C \tag{12}$$

$$C = \frac{k}{\alpha} \tag{13}$$

Therefore, the potential energy function is:

$$U(r) = \frac{k}{\alpha}(\exp(\alpha r) - \alpha r - 1) \tag{14}$$

3.3 Force-Energy Relationship Verification

We verify that $F_r = -\frac{dU}{dr}$:

$$\frac{dU}{dr} = \frac{k}{\alpha}(\alpha \exp(\alpha r) - \alpha) = k(\exp(\alpha r) - 1) \tag{15}$$

Therefore:

$$F_r = -\frac{dU}{dr} = -k(\exp(\alpha r) - 1) \tag{16}$$

This matches our force law with the appropriate sign convention (force points toward equilibrium for positive displacements).

3.4 Vector Form Implementation

For implementation in 2D, we need the vector components:

$$\vec{F} = F_r \frac{\vec{d}}{|\vec{d}|} = k(\exp(\alpha|\vec{d}|) - 1)\frac{\vec{d}}{|\vec{d}|} \tag{17}$$

The components are:

$$F_x = k(\exp(\alpha|\vec{d}|) - 1)\frac{d_x}{|\vec{d}|} \tag{18}$$

$$F_y = k(\exp(\alpha|\vec{d}|) - 1)\frac{d_y}{|\vec{d}|} \tag{19}$$

4 Model Comparison

4.1 Small Displacement Behavior

For small displacements ($\alpha r \ll 1$), we can expand the exponential:

$$\exp(\alpha r) \approx 1 + \alpha r + \frac{(\alpha r)^2}{2} + \dots \tag{20}$$

The exponential force becomes:

$$F_r \approx k\left(\alpha r + \frac{(\alpha r)^2}{2}\right) \approx k\alpha r \tag{21}$$

This shows that for small displacements, the exponential spring behaves linearly with effective spring constant $k_{eff} = k\alpha$.

4.2 Large Displacement Behavior

For large displacements ($\alpha r \gg 1$):

$$F_r \approx k \exp(\alpha r) \tag{22}$$

The force grows exponentially, providing much stronger resistance to large deformations compared to linear springs.

4.3 Energy Storage Comparison

The potential energy storage differs significantly:

Linear: $U_{linear} = \frac{1}{2} k_{linear} r^2$ (quadratic growth)

Exponential: $U_{exp} = \frac{k}{\alpha}(\exp(\alpha r) - \alpha r - 1)$ (exponential growth)

5 Implementation Changes

5.1 Parameter Modifications

The following parameters were modified in the Julia code:

Listing 1: Original linear spring parameters

```julia
# Original linear spring parameters
const K_COUPLING   = 100.0         # N m^-1
const K_DIAGONAL   = 50.0          # N m^-1
```

Listing 2: New exponential spring parameters

```julia
# New exponential spring parameters
const K_COUPLING   = 100.0         # N
const K_DIAGONAL   = 50.0          # N
const ALPHA_COUPLING = 10.0       # m^-1
const ALPHA_DIAGONAL = 10.0       # m^-1
```

Note that the spring constants now have units of force (N) rather than force per unit length (N/m).

5.2 Spring Force Function Changes

Original Linear Implementation:

Listing 3: Original linear spring force function

```julia
function spring_force_2d(pos1, pos2, k)
    displacement = pos2 - pos1
    return k * displacement      # Hooke's law in 2D
end
```

New Exponential Implementation:

Listing 4: New exponential spring force function

```julia
function spring_force_2d(pos1, pos2, k, alpha)
    displacement = pos2 - pos1
    distance = norm(displacement)

    if distance < 1e-12
        return zeros(2)
    end

    direction = displacement / distance
    force_magnitude = k * (exp(alpha * distance) - 1.0)

    return force_magnitude * direction
end
```

5.3 Potential Energy Function Changes

Original Linear Implementation:

Listing 5: Original linear potential energy calculation

```
function potential_energy_2d_with_diagonals(pos_matrix)
    pe = 0.0
    # For each spring connection:
    displacement = pos_matrix[:, k2] - pos_matrix[:, k1]
    pe += 0.5 * K_COUPLING * sum(displacement.^2)
    return pe
end
```

New Exponential Implementation:

Listing 6: New exponential potential energy calculation

```
function potential_energy_2d_with_diagonals(pos_matrix)
    pe = 0.0
    # For each spring connection:
    displacement = pos_matrix[:, k2] - pos_matrix[:, k1]
    distance = norm(displacement)
    pe += (K_COUPLING / ALPHA_COUPLING) *
          (exp(ALPHA_COUPLING * distance) -
           ALPHA_COUPLING * distance - 1.0)
    return pe
end
```

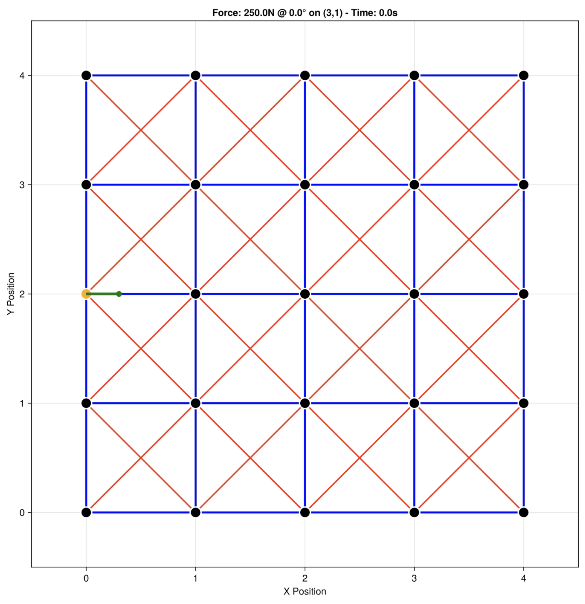

Figure 1: Schematic of the initial state of the 5×5 Mass-Spring Lattice with nearest neighbor and diagonal (X-pattern) connections. The location (3,1) at which external force is applied is indicated.

6 Energy Conservation

6.1 Conservation Principle

Energy conservation in the modified system follows the same fundamental principle:

$$E_{total} = T + V = \text{constant} + W_{external} \tag{23}$$

where:

- T is kinetic energy
- V is potential energy (now exponential)
- $W_{external}$ is work done by external forces

6.2 Numerical Accuracy

The exponential spring model maintains the same numerical accuracy for energy conservation because:

1. **Consistent Derivation**: The potential energy function is properly derived from the force law
2. **Symplectic Integration**: The Vern9 solver preserves energy to machine precision
3. **Proper Gradient**: The force is exactly $-\nabla U$

6.3 Verification

Energy conservation can be verified by monitoring:

$$\Delta E = |E(t) - E(0) - W_{external}(t)| < \epsilon \tag{24}$$

where ϵ is the numerical tolerance (typically 10^{-12} to 10^{-14}).

7 Numerical Considerations

7.1 Stability

The exponential force law can become very large for significant displacements. Care must be taken to:

1. Choose appropriate α values to prevent numerical overflow
2. Use adaptive time stepping for stability
3. Monitor displacement magnitudes during simulation

8 Simulation Results

The direct terminal output after running the script with the previously outlined parameters:

```
5×5 Lattice Mass-Spring System with Configurable Force
=================================================================
Physical Parameters:
  Total masses: 25
  DOF per mass: 2 (x, y)
  Total DOF: 50
  Nearest neighbor spring constant: 100.0 N
  Nearest neighbor alpha: 10.0 m^-1
  Diagonal spring constant: 50.0 N
  Diagonal alpha: 10.0 m^-1
  Mass: 1.0 kg

Spring Network:
  Nearest neighbors: horizontal & vertical connections
  Diagonal springs: X-pattern connections in each square
  Total connectivity: 8 neighbors per interior mass
  Spring type: Exponential

Configurable External Force:
  Magnitude: 250.0 N
  Angle: 0.0° (0°=right, 90°=up, 180°=left, 270°=down)
  Applied to: Mass at position (3, 1)
  Duration: 0.05 s
  Force components: Fx = 250.0 N, Fy = 0.0 N

Solver completed successfully!
Final time: 5.0 s
Time steps: 501

Final Summary:
==================================================
Force Configuration:
  Magnitude: 250.0 N
  Angle: 0.0°
  Target mass: (3, 1)
  Components: Fx = 250.0 N, Fy = 0.0 N

Spring Network Statistics:
  Nearest neighbor springs: 40
  Diagonal springs: 32
  Total springs: 72

Energy Conservation:
  Total work input: 24.24754391201347
  Final total energy: 24.247543945208772
  Energy error: 3.3195302506783264e-8
  Final % error: 1.369017110649983e-7%
Creating animation file: animations/lattice_anim.mp4
Animation saved to: animations/lattice_anim.mp4
```