Mechanics of Flexible Structures & Soft Robots
MAE-263F
Fall 2025

Homework 1

Due Date: 11:59 PM, 10/15/2025

You should create a single GitHub repository for this class and share it with the instructor (`khalidjm@seas.ucla.edu`). All the homeworks, reports, presentations, and proposal should be uploaded to this repository. Do not create a separate repository for each assignment. Within your repository, create a separate folder for each assignment (e.g., `homework_1`, `homework_2`, `homework_3`, `homework_4`, `homework_5`, `proposal`, `midterm_report`, and `final_report`).

Submission Instructions: Your submission on BruinLearn should only contain the URL to your GitHub repository. Your GitHub repository should include the following items:

1. **Report in PDF format:** Submit a report in `.pdf` format (file name should be `Homework1_LASTNAME.pdf`, replacing `LASTNAME` with your last name) addressing the questions asked in the deliverables. Include all the plots and figures requested in the assignment and discuss them in the report. See the syllabus for formatting requirements. As stated in the syllabus, you must use one of the provided templates.
2. **Source code:** The submission should include one main file named exactly as `HomeworkX.[ext]` (where `X` is the homework number) along with any additional files (e.g., functions or text files) as necessary. The main file should require no more than a single command to run or one click for execution.
3. **README file:** Add a `README.md` file on GitHub to provide clear instructions on how to run your code and describe the purpose of each file included in your repository.

Fig. 1 shows two different networks, (a) and (b). The configuration at time $t = 0$ is shown. All quantities are in SI units: mass (`m` in the figure) in kilograms (kg) and spring stiffness k in newtons per meter (N/m).
Boundary conditions: Nodes 0 and 2 are fixed.

Spring energy and strain. For each spring k connecting nodes i and j, the energy is

$$E_k = \frac{1}{2} k \, \ell_k \, \varepsilon^2, \tag{1}$$

where ℓ_k is the rest length (Euclidean distance between node i and node j at $t = 0$), and

$$\varepsilon = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\ell_k} - 1. \tag{2}$$

Use the functions `gradEs` and `hessEs` to obtain the gradient and Hessian of the spring energy with respect to (x_i, y_i, x_j, y_j).

Input files. Create the following text files to describe the network at $t = 0$.

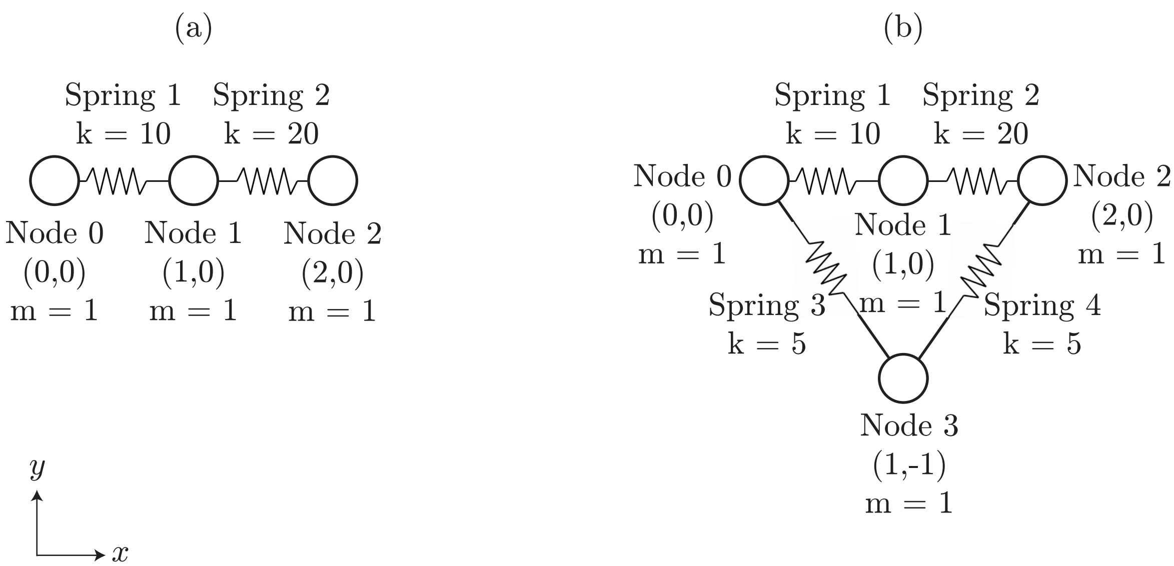

Figure 1: Network of Springs.

`nodes.txt` One node per line, two numbers (x and y):

```
0, 0
1, 0
2, 0
```

For example, for (a) with three nodes as above, the first line `0, 0` means node 0 has $x_0 = 0$, $y_0 = 0$ at $t = 0$.

`springs.txt` One spring per line, three numbers (i, j, k):

```
0, 1, 10
1, 2, 20
```

For example, the first line `0, 1, 10` means spring 0 connects node 0 (x_0, y_0) and node 1 (x_1, y_1) with stiffness $k = 10\,\mathrm{N}$.

Task. Write a simulator using *implicit Euler* to simulate network (b) from $t = 0$ to $t = 100\,\mathrm{s}$.

- Plot the shape of the spring network at $t = \{0, 0.1, 1, 10, 100\}\,\mathrm{s}$. Include axis labels with *units*, legends if needed, and a title indicating the time.
- As a function of time, plot the y-coordinates of all *free* nodes (i.e., nodes 1 and 3). Include labels with *units*.

Your submission only needs to simulate network (b).

Report questions. Answer the following in your report (you may include and upload any auxiliary scripts used to generate results):

1. **Pseudocode and code structure:** Write pseudocode for your simulator describing its main logic and workflow. Briefly describe the main functions and scripts in your implementation (2–3 sentences each), including their inputs and outputs. Create a simple block diagram showing how the functions or scripts interact (i.e., which one calls which). There is no strict format requirement—use your best judgment to make it understandable to someone with an undergraduate-level engineering background.

2. How do you choose an appropriate time step size Δt?

3. Simulate the spring network using *Explicit Euler* for $t \in [0, 1]$ s. If it becomes unstable, reduce Δt and try again. If it still fails even at $\Delta t = 10^{-6}$ s, state this in the report. Explain which method (implicit vs. explicit) is preferable for this spring network and why.

4. The simulation with implicit Euler appears to reach a static state (numerical damping). Read about the Newmark–β family of time integrators and explain how such integrators can mitigate such artificial damping.

5. **(Ungraded) Newmark–β Simulation:** Simulate the spring networks in (a) and (b) using the Newmark–β method and demonstrate that the artificial damping observed in the implicit Euler method is eliminated.