MAE263F Homework 1

Jiandong Gu[1]

I. Plot results

The shape of mass-spring system:

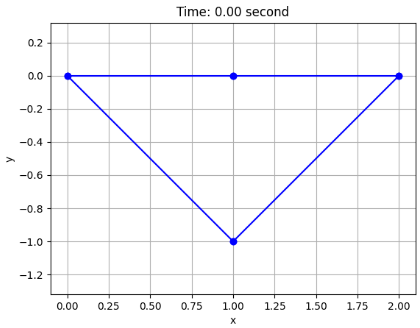

Fig. 1. T = 0s

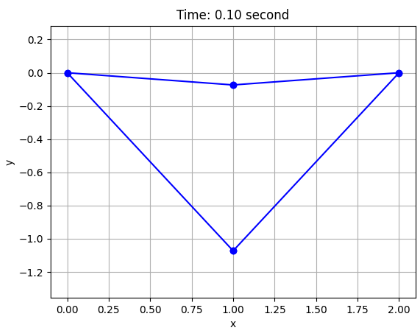

Fig. 2. T = 0.1s

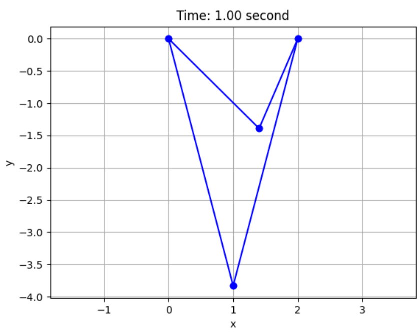

Fig. 3. T = 1s

The shape of y-coordinate of node1 and node3:

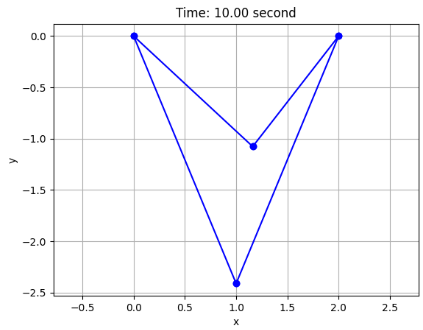

Fig. 4. T = 10s

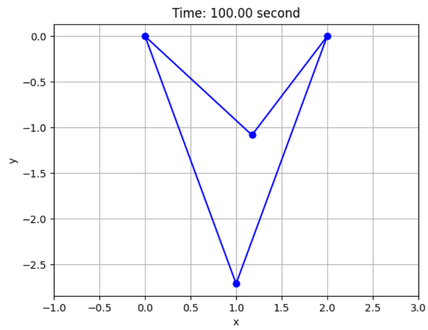

Fig. 5. T = 1001s

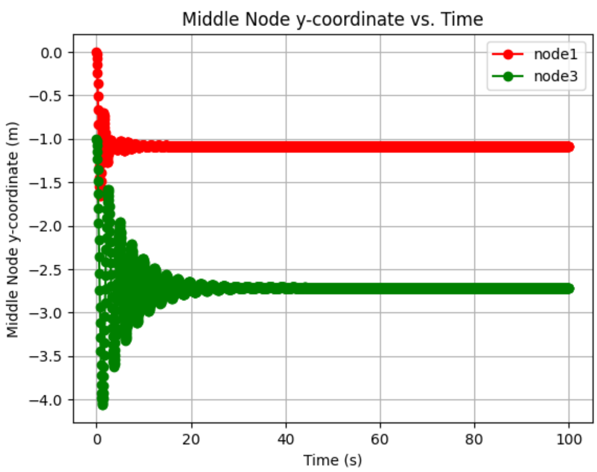

Fig. 6. y-coordinate of nodes

II. Report question 1

A. Pseudocode

Algorithm 1 Multi-spring simulation

Input: nodes.txt, springs.txt, Free_DOF
Output: Position as a function of time $\mathbf{X(t)}$

Set start position and velocity $\mathbf{x_{old}}$, $\mathbf{u_{old}}$
Set time interval **dt** and end time $\mathbf{t_e}$
Calculate time series $\mathbf{T[0:t_e,dt]}$
for $\mathbf{t_{new}}$ in T **do**
  Given $\mathbf{x_{old}}$, $\mathbf{u_{old}}$
  Using Newton-Raphson method to solve $\mathbf{x_{new}}$, $\mathbf{u_{new}}$
  while error $>$ **eps do**
    Calculate $\mathbf{F_{spring}}$ (Gradian matrix of Es $\mathbf{G_{es}}$)
    Calculate $\mathbf{J_{spring}}$ (Hessian matrix of Es $\mathbf{H_{es}}$)
    $\mathbf{F = F_{interia} + F_{spring} - F_{ext}}$
    $\mathbf{J = J_{interia} + J_{spring} - J_{ext}}$
    Solve Δx with $\mathbf{F,J}$
    Renew $\mathbf{x_{new}}$, $\mathbf{u_{new}}$
    Calculate **error** = $\mathbf{F(x_{new})}$
  end while
  Save $\mathbf{x_{new}}$, $\mathbf{u_{new}}$ to $\mathbf{X(t)}$, $\mathbf{U(t)}$
end for
return $\mathbf{X(t)}$

B. Main functions and scripts

- **gradEs(xk, yk, xkp1, ykp1, l_k, k)** Calculate the gradian of spring Energe. Inputs are positions, stiffness and rest length. Output is gradian matrix of spring force
- **hessEs(xk, yk, xkp1, ykp1, l_k, k)** Calculate the hessian of spring Energe. Inputs are positions, stiffness and rest length. Output is hessian matrix of spring force
- **getForceJacobian(x_new, x_old, u_old, stiffness_matrix, index_matrix, m, dt, l_k)** Calculate the force when it is at t_new. Inputs are positions, stiffness, connections and rest length. Output is the force and force Jacobian
- **Cell: Preparation at t = 0**
Setting of initial states of objects
- **myInt(t_new, x_old, u_old, free_DOF, stiffness_matrix, index_matrix, m, dt)** Use implicit method to solve x_new and u_new. Inputs are positions, velocities, and fixed parameters of mass-spring system. Outputs are x_new and u_new
- **explicit_int(x_old, u_old, free_DOF, stiffness_matrix, index_matrix, m, dt)** Use explicit method to solve x_new and u_new. Inputs are positions, velocities, and fixed parameters of mass-spring system. Outputs are x_new and u_new
- **Cell: Main simulation loop**
Main function of this project. Solving position as a function of time through looping.

C. Block diagram

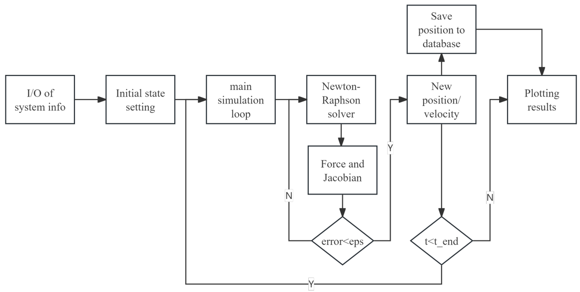

Fig. 7. Block diagram

III. Report Question 2

Considering the existing numerical damping in this problem, I choose Δt using the standards below.

Firstly, the Δt should reflect the virtual damping; Secondly, the Δt should hint the shape of output where there is no damping. Thirdly, the total simulation time should be limited.

The experiments are carried provided when eps is fixed to 1e-6 and simulation time to 10s. Adjust Δt to get different shapes of output.

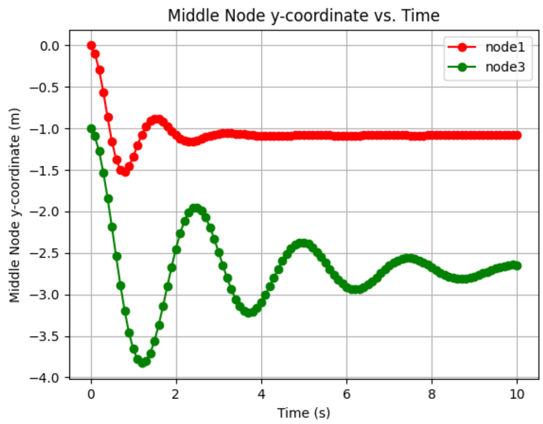

Fig. 8. Δt = 0.1s

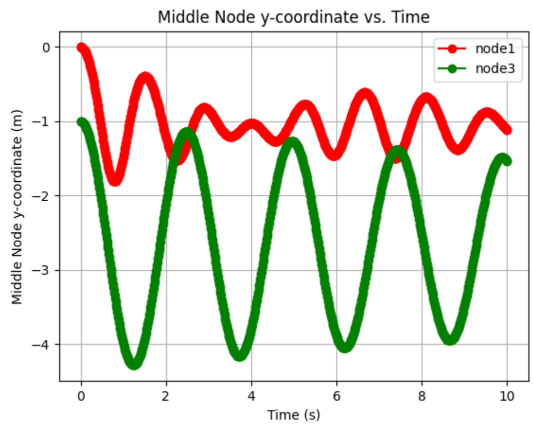

Fig. 9. Δt = 0.01s

As shown in Fig 8,9,10, when Δt decreases:

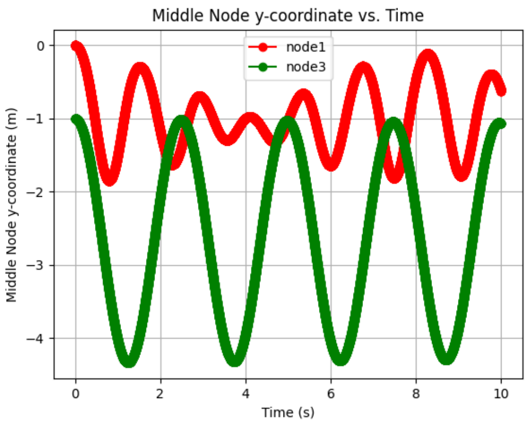

Fig. 10. Δt = 0.001s

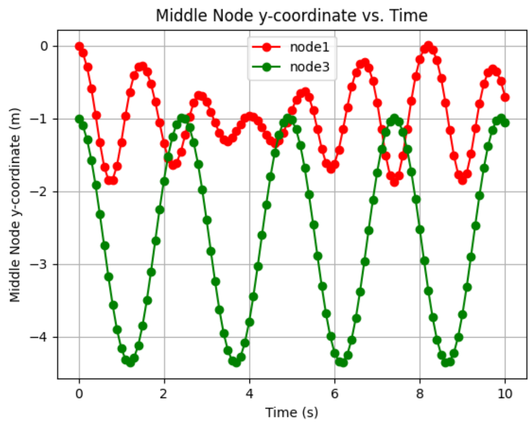

Fig. 12. Explicit method: Δt = 0.1s, $t \in [0,10]$

- The numerical damping has a tendency to be mitigated. As in Fig 10, in this picture the amplitude of output slightly decreases as time goes on. But in the other pictures, the y-coordinates tends to be stable around a specific number.
- The simulation time increases. Though not provided in picture, the simulation time using $\Delta t = \{0.1, 0.01, 0.001\}$ are respectively $\{0.49s, 1.88s, 13.8s\}$, which is the same with common sense.
- The output period is the same in all three figures. Take the y-coordinate of node3 as an example. It is obvious that the period can be described by $cos(\frac{2\pi t}{\sqrt{5}})$.

So, the summary of how to choose Δt is that Δt should be in range [0.01,0.1]. In the code I use $\Delta t = 0.05s$ for implicit method as a default value.

IV. Report Question 3

Using explicit method to solve this problem. Due to the fact that the period of position can't be reflected when $t \in [0,1]$, I also provided figure when $t \in [0,10]$ shown in Fig 12. The result shows that explicit method is a better choice in this problem.

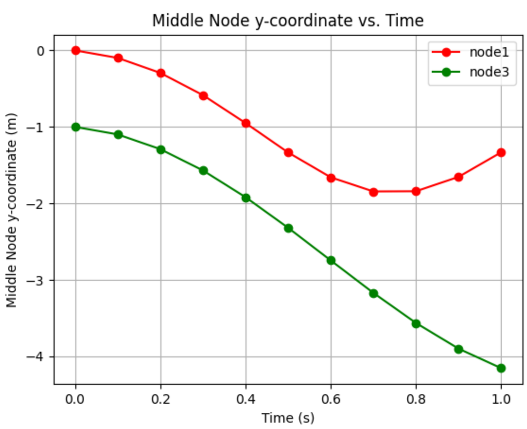

Fig. 11. Explicit method: Δt = 0.1s, $t \in [0,1]$

A. *Why explicit method*

Given the output of y-coordinate is a periodic function, there will be distortion if the sample frequency is less than twice output frequency (Nyquist sampling theory). To prove this, I record the shape of output when $\Delta t = \{0.25s, 0.4s, 0.8s\}$.

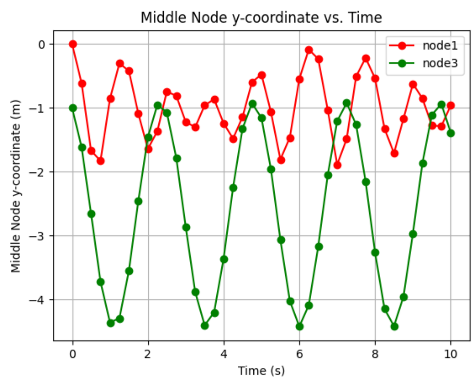

Fig. 13. $\Delta t = 0.25s$: the output of both nodes are recoverable

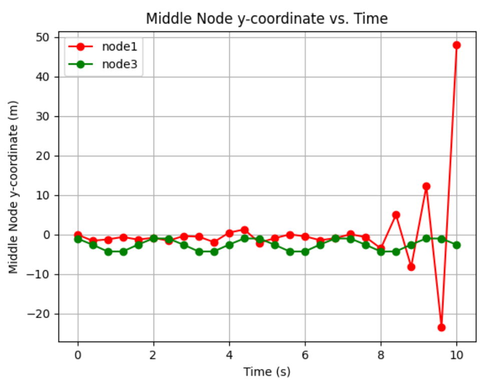

Fig. 14. $\Delta t = 0.4s$: only the output of node3 is recoverable

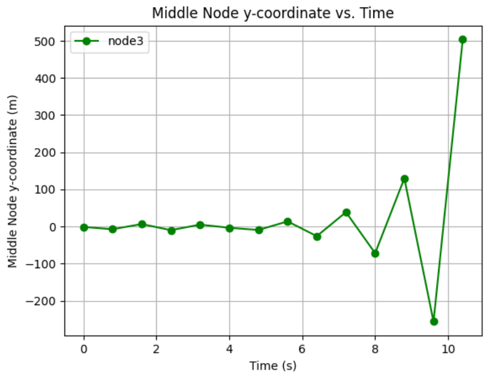

Fig. 15. $\Delta t = 0.8s$: the output of both nodes are not recoverable

According to the experiments using implicit method and direct solving of ODEs, the output frequency of node1 and node3 are respectively 0.9Hz and 0.45Hz. However, the experiments shows that the valid Δt is actually limited to a smaller range than expectation. The possible reason is that the sampling object is $\Delta^2 t$, though this is still an assumption and hasn't been proved.

Fig 13,14,15 show that explicit method works well when $\Delta t < 0.25s$. In another word, explicit method has advantages over implicit method including computing speed, no numerical damping and wider range of Δt in mass-spring system.

V. Report Question 4

The incremental equations of Newmark-β method can be written as:

$$u'_{n+1} = u'_n + \Delta t(1-\gamma)u''_n + \Delta t \gamma u''_{n+1} \tag{1}$$

$$u_{n+1} = u_n + \Delta t u'_n + (1-2\beta)\frac{(\Delta t)^2}{2}u''_n + \beta(\Delta t)^2 u''_{n+1} \tag{2}$$

As shown in Eq1,2, the newmark-β integrator introduces two parameters γ, β to allow adjusting artificial damping. When $\gamma = 1/2$, the integrator is used for numerical stability situations. The complete demonstration of stability is in [1].

Define a typical mass-spring-damping system using Newmark-beta integrator. M,C,K are respectively defined as mass matrix, damping matrix and stiffness matrix. Let state $q_n = [\dot{u}_n, u_n]$ and the state update matrix $A = H_1^{-1}H_0$:

$$H_0 = \begin{bmatrix} M-(1-\gamma)\Delta t C & -(1-\gamma)\Delta t K \\ -\left(\frac{1}{2}-\beta\right)\Delta t^2 C + \Delta t M & M - \left(\frac{1}{2}-\beta\right)\Delta t^2 K \end{bmatrix}$$
$$H_1 = \begin{bmatrix} M+\gamma\Delta t C & \gamma \Delta t K \\ \beta\Delta t^2 C & M+\beta\Delta t^2 K \end{bmatrix} \tag{3}$$

The characteristic equation is shown in Eq4.

$$\lambda^2 - \left(2-(\gamma+\frac{1}{2})\eta_i^2\right)\lambda + 1 - (\gamma - \frac{1}{2})\eta_i^2 = 0$$
$$\eta_i^2 = \frac{\omega_i^2 \Delta t^2}{1+\beta\omega_i^2\Delta t^2} \tag{4}$$

When C = 0, it means that there is no damping in this system. To avoid introducing new artificial damping into this system while maintain stability, the parameter γ should be set to 1/2. Then there is:

$$(\lambda-1)^2 = \eta_i^2 \tag{5}$$

And we can easily know from Eq5 that $|\lambda| = 1$

Please noted that this is not a complete demonstration of Newmark-beta integrator's stability. The original paper [2] has a more detailed analysis than this report.

References

[1] G. Lindfield and J. Penny, "Chapter 5 - solution of differential equations," in *Numerical Methods (Fourth Edition)* (G. Lindfield and J. Penny, eds.), pp. 239–299, Academic Press, fourth edition ed., 2019.

[2] N. M. Newmark, "A method of computation for structural dynamics," *Journal of the Engineering Mechanics Division*, vol. 85, no. 3, pp. 67–94, 1959.