Transient Finite Element and Finite Difference Methods

SULAIMAN, ABDUL-HAFIZ AKANMU

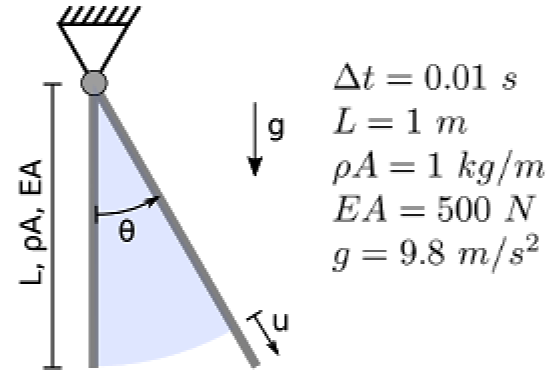
Department of Computational Engineering,

Ruhr University Bochum

Student-ID: 108024251117

June, 2025

Generalized-Alpha Method



For an elastic pendulum, the equation of motion is given as below

$$\frac{\rho A L^3}{3}\ddot{\theta} + \frac{\rho A g L^2}{2}\sin(\theta) = 0 \quad (1)$$

$$\frac{\rho A L}{3}\ddot{u} + \frac{EA}{L}u = 0 \qquad (2)$$

Assuming small amplitude vibrations i.e., $sin(\theta) \approx \theta$, eqn. (1) becomes

$$\frac{\rho A L^3}{3}\ddot{\theta} + \frac{\rho A g L^2}{2}\theta = 0$$

Subject to initial conditions;

$$u_0 = -\frac{L}{5}, \quad \dot{u_0} = 0, \quad \theta_0 = 0, \quad \dot{\theta_0} = \sqrt{\frac{g}{6L}}$$

Problem 1

The Semi-discrete matrix form $\mathbf{M}\ddot{\mathrm{U}} + \mathbf{C}\dot{\mathrm{U}} + \mathbf{K}\mathrm{U} \;=\; \mathbf{0}$, where $\mathbf{C}$ assumes the Rayleigh damping $\mathrm{C} = \alpha_1\mathbf{M} + \alpha_2\mathbf{K}$ is expanded as below:

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}\begin{bmatrix} \ddot{\theta} \\ \ddot{u} \end{bmatrix} + C\left(\alpha_1\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} + \alpha_2\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}\right) + \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}\begin{bmatrix} \theta \\ u \end{bmatrix} \;=\; \mathbf{0}$$

Solving for all entries,

$m_{11} = 1/3, m_{12} = m_{21} = 0; m_{22} = 1/3; k_{11} = 49/10; k_{12} = k_{22} = 0; k_{22} = 500$, one obtains:

$$\begin{bmatrix} 1/3 & 0 \\ 0 & 1/3 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{u} \end{bmatrix} + C\left(\alpha_1 \begin{bmatrix} \alpha_1/3 + 49\alpha_2/10 & 0 \\ 0 & \alpha_1/3 + 500\alpha_2 \end{bmatrix} + \right) + \begin{bmatrix} 49/10 & 0 \\ 0 & 500 \end{bmatrix} \begin{bmatrix} \theta \\ u \end{bmatrix} = \mathbf{0}$$

Problem 2

Using the Chung and Hulbert, 1993, Generalised Newmark – alpha scheme with optimal details as follow:

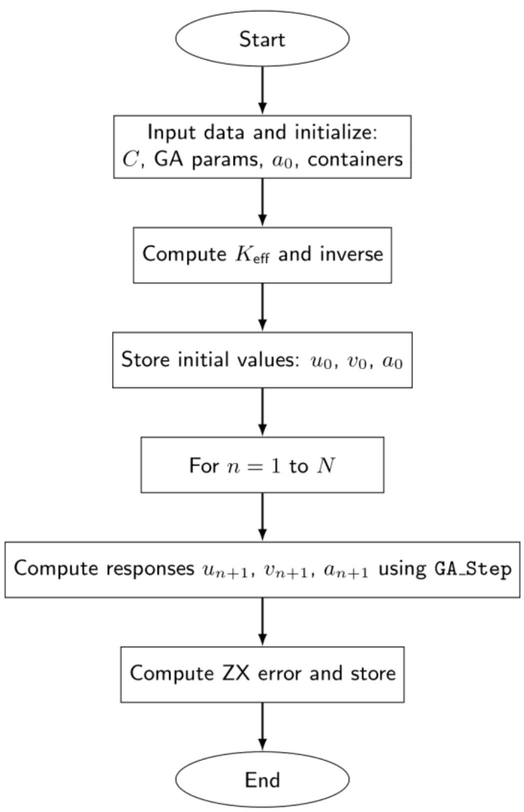
$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}, \quad \beta = \frac{1}{4}\left[1 - \alpha_m + \alpha_f\right]^2, \quad 1/2 - \alpha_m + \alpha_f$$

In a so-called **GA_Step** in Matlab script.

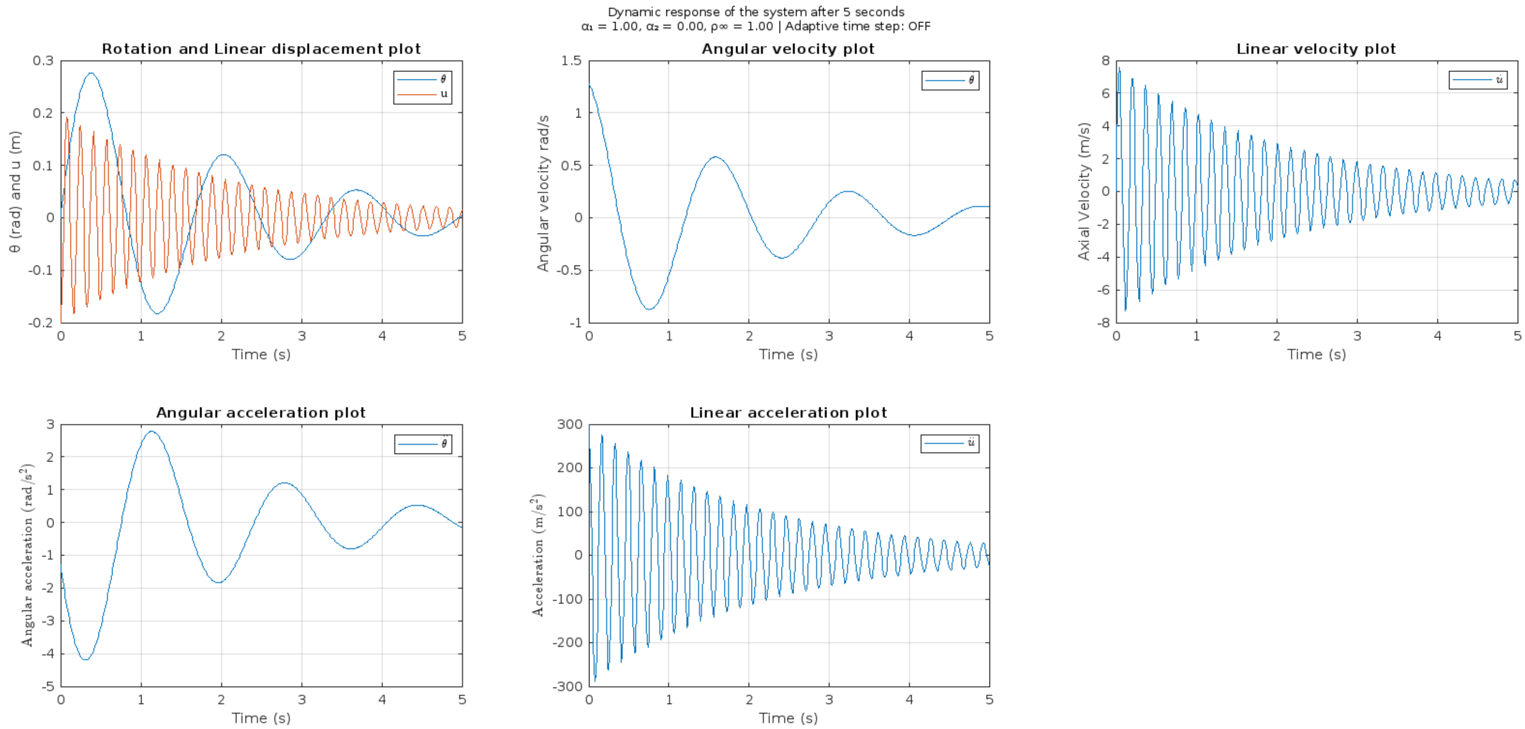
Problem 3

After implementing the dynamic step, it is possible to write a conditional for loop algorithm that solves the system in **Task 1** in a constant Δt of 0.01s for 5 seconds.

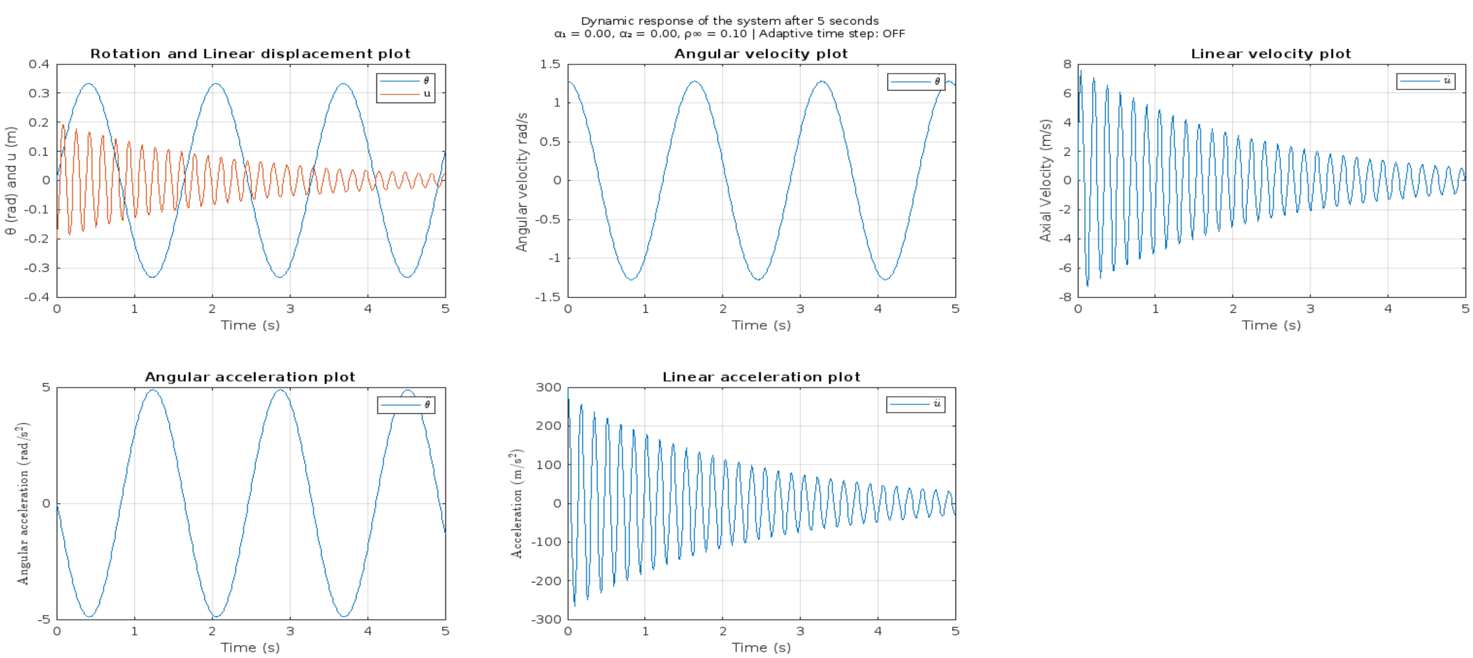
The implementation is as below;



3a) Dynamic response for $\alpha_1 = 1; \alpha_2 = 0 \; and \; \rho_\infty = 1,$



3b) Dynamic response for $\alpha_1 = 0; \alpha_2 = 0 \; and \; \rho_\infty = 0.1,$



Differences in analysis

1) **Damping of responses with lower frequency:** There is a significant difference in the rotational displacement of 3a with $\alpha_1 = 1$ and 3b with $\alpha_1 = 0$. This is because of α_1 in the Rayleigh damping formula. α_1 is inversely proportional to frequency and is associated with the mass matrix – the inertia term responsible for moving a system slowly when accelerated. Because of the high frequency of the displacement, it tends to not be greatly affected by this term.
2) **Numerical damping of the iteration scheme:** This manifests in the reduction of values (amplitudes) for higher frequency modes. From system 3b, even though there is no mathematical physical damping factor, there is still a significant reduction of amplitudes for the displacement towards the 5 seconds timeline. This is due to smaller spectral radius at infinity, ρ_∞ for system b.

→ Values after 5 seconds for both systems

3a) $\begin{pmatrix} \theta \\ u \end{pmatrix} = \begin{pmatrix} 0.0043 \\ 0.0167 \end{pmatrix} rad, m;$ 3b) $\begin{pmatrix} \theta \\ u \end{pmatrix} = \begin{pmatrix} 0.1021 \\ 0.0221 \end{pmatrix} rad, m$

Problem 4

The error indicator by Zienkiewicz and Xie, which has been formulated as

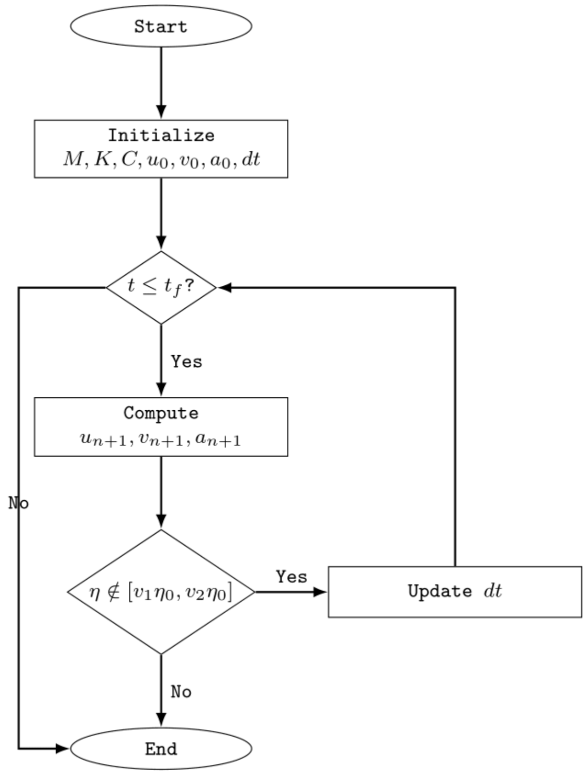
$e_{n+1}^{zx} = \left(\frac{6\beta-1}{6}\right)[\ddot{u}_{n+1} - \ddot{u}_n]\Delta t^2 + O(\Delta t^4)$, has been implemented in the Matlab script provided.

Problem 5

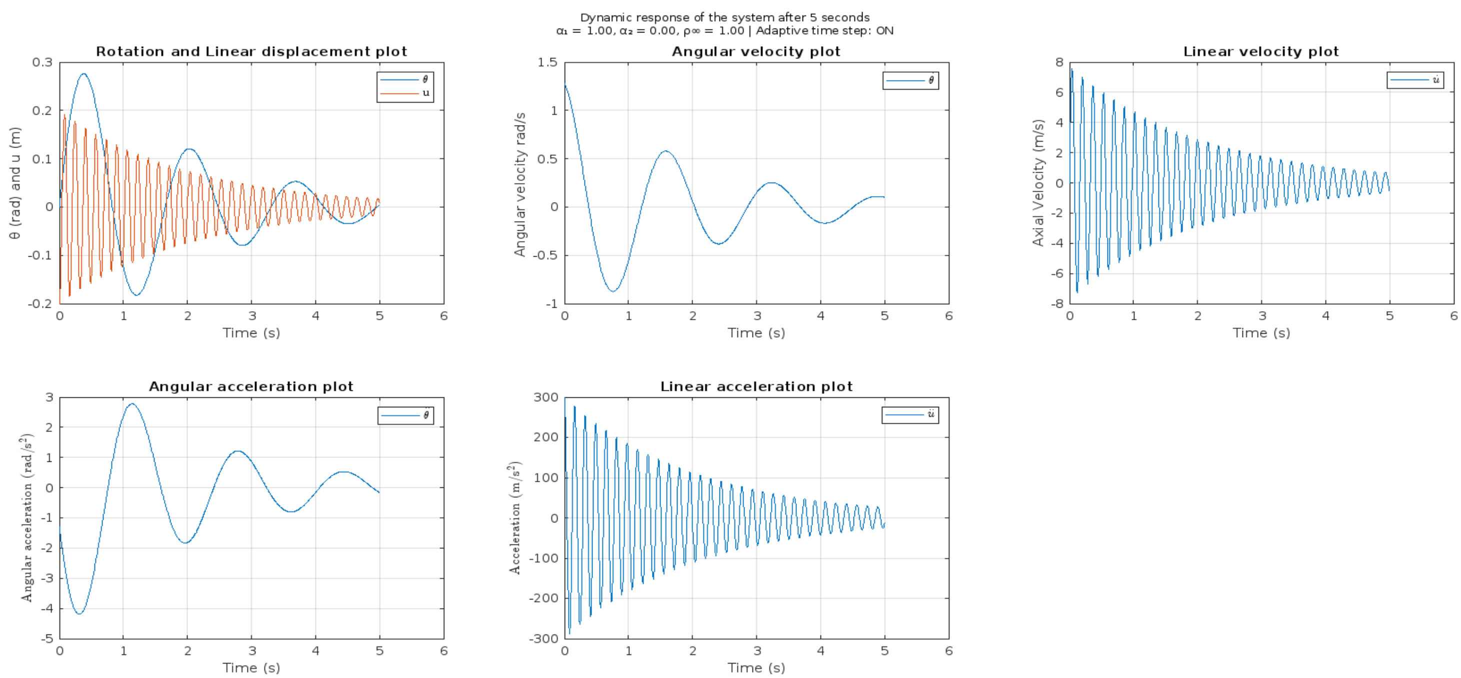
An implementation extension for adaptive time stepping is made. Using error bounds, where $v1 = 1.0, v2 = 10$ and $\eta_e = 1.0x10^{-3}$, adaptive steps in time change, Δt, was made as $\Delta t_{new} = \Delta t_{old} \times \sqrt{\eta_e/\eta}$. Implementation is as in the Matlab script.

Problem 6

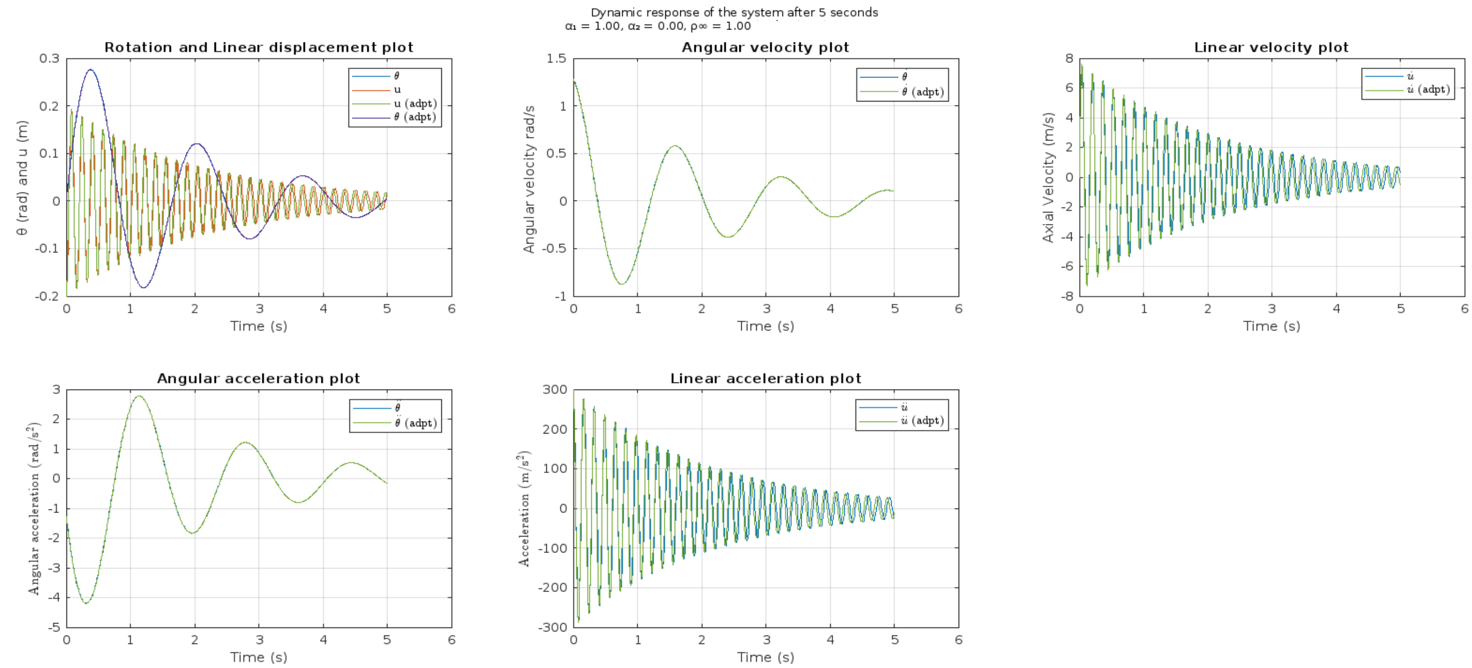
An aggregation of the above algorithms is then plugged together for the adaptive algorithm as below.



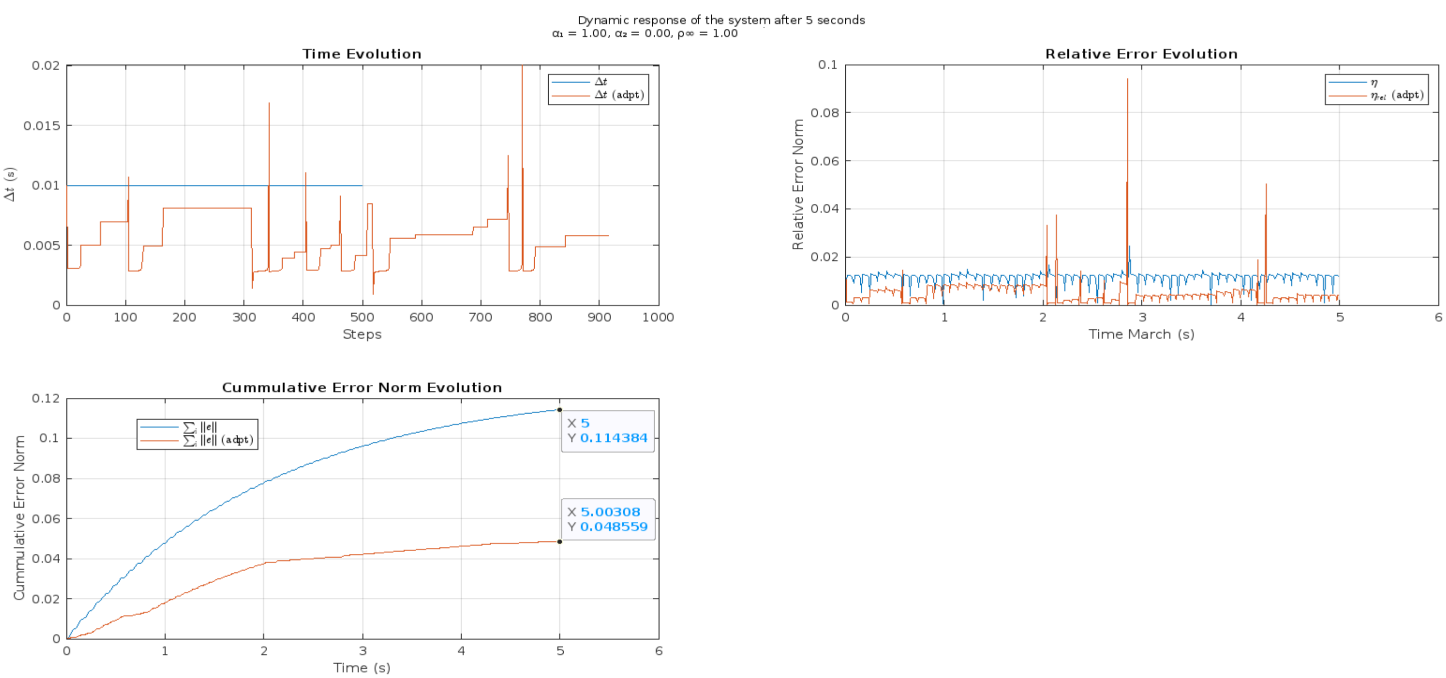
Adaptive Dynamic response for $\alpha_1 = 1; \alpha_2 = 0$ *and* $\rho_\infty = 1$,



Comparison with Task Problem(a)



In responses with high frequency (linear displacement, linear velocity and acceleration), there is a small, but visible shift in both solutions. The constant time step is seen taking a forward lead in time in all cases with a difference that started initially as zero. For the responses with lower frequency, there is almost no visible difference in the results of both methods.



For time steps, the constant solution took a deterministic **500** steps from the initial conditions, but for the adaptive time step, the 5 seconds mark was reached at **917** steps after the initial condition as time step 0.

The relative error norm over time for the constant step seems to maximize steadily at 0.015 with few higher spikes (not higher than 0.024). However, the relative error norm for the adaptive step is lower for most cases with few higher spikes. The spikes in the case are way higher than the constant ones and are followed by lower relative error norms in time due to the heavy penalization of the error formula.

Expectedly, the Cumulative Error Norm (CEN)of the adaptive case is lower than that of the constant case. The CEN for the constant case being **0.1144**, while that of the adaptive case is **0.0486** according to the Zienkiewicz and Xie error indicator. This is a 57.5% difference to the contact case.