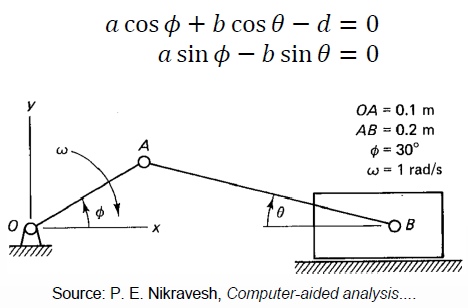
Using Newton-Raphson’s Method on Crank Mechanism

# Introduction

The task was about creating a program in MATLAB, that plots displacement, velocity and acceleration of a connecting rod and a piston of a Slider Crank Mechanism.

# Solution for constant

, constrain equations.

Solution for ,

Jacobian.

. First time derivative (velocity), where and is constant.

This can be also represented in a form , where the terms associated with the known crank velocity are moved to the right side:

. Solution for .

. Second time derivative (acceleration).

This can be also represented in a form , where the terms associated with the known crank acceleration and the quadratic velocity are moved to the right side:

. Solution for .

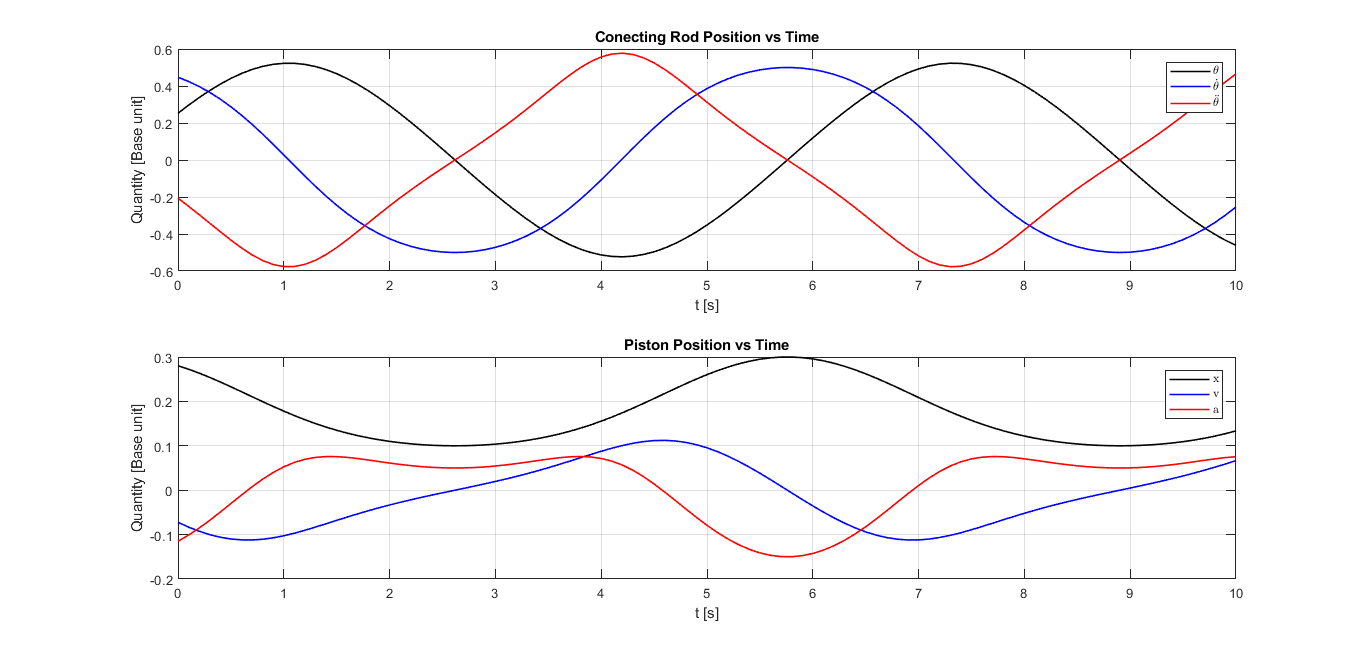
# Program

A program Crank\_Mechanism.m was created, which calls function NR\_method.m. In the beginning of the program, parameters and time vector is defined. Next, initial guess and tolerance for the solution of the equations is set. The vectors of displacement *x*, velocity *v* and acceleration *aa* are initialized with zeros to match size of the time vector.

After that, function *for* is used to call one by one each element of the vector *phi*, which corresponds to the angle of rotation of the crankshaft and save it to corresponding row of each wanted vectors *x*, *v* and *aa*. These vectors are computed using function NR\_method.m, which uses Newton-Raphson’s method to solve systems of nonlinear algebraic equations. Above mentioned equations are used in its input. Finally, these vectors are saved to extra variables and then plotted.

# Results

The results for constant angular velocity , time interval (0, 10 s), , tolerance and initial guess looks like this:



# Conclusions

All sources were pushed to GitHub. Link to the repository is: https://github.com/q2493/HW\_3\_Trusina.git

The correct results very depend on initial guess. For example, with initial guess solution diverge:

