```
main.m
    Simulation start;
    Read input data :  geometry and connections;
    defSimParams.m : define simulation parameter;
    defSystemParams.m : build a system struct;
    InitialElement.m : initialization of each type of element;
    defConsParams.m : define boundary conndtions;
    Simulation loop
        While    (current time $\leq$ total time)
                 Apply current loading step;
                 Guess a new solution;
                 objfun.m
                     While    (error < tolerance)
                              getFs (if needed) : stretching force,  stretchingForce.m;
                              getFb (if needed) : bending force,  bendingForce.m;
                              getFt (if needed) : twisting force,  twistingForce.m;
                              getFo (if needed) : other force;
                              Include the inertia and damping effects (if needed);
                              Compute sum force and Hession;
                              Newton′s method for optmization;
                              Update DOF vector;
                              Update error;
                     end While
                 Upate time step;
                 plotSystem.m : plot the dynamic rendering;
                 Output data (if needed);
        end While
    Simulation end
```