|  |
| --- |
|  |

Samson Mano

saminnx@gmail.com

<https://sites.google.com/site/samsoninfinite/>

[https://github.com/Samson-Mano](%20https://github.com/Samson-Mano)

Position based dynamics

Formulation of PBD for compliant constrained dynamics

# Constraint dynamics

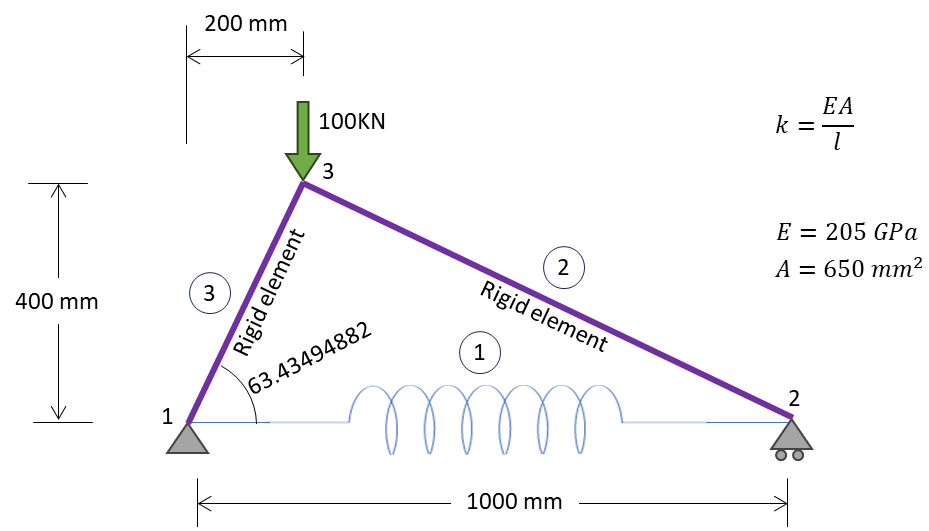
Constrained dynamics is a fundamental concept in physics simulations, particularly in the realms of computer graphics where it is employed to realistically model the behavior of both rigid and soft bodies. Constrained dynamics refers to the study and simulation of physical systems whose motion is restricted in some way by constraints. A constraint can be anything that limits the freedom of a system to move, such as joints, springs, or collision with other bodies.

## Algorithm

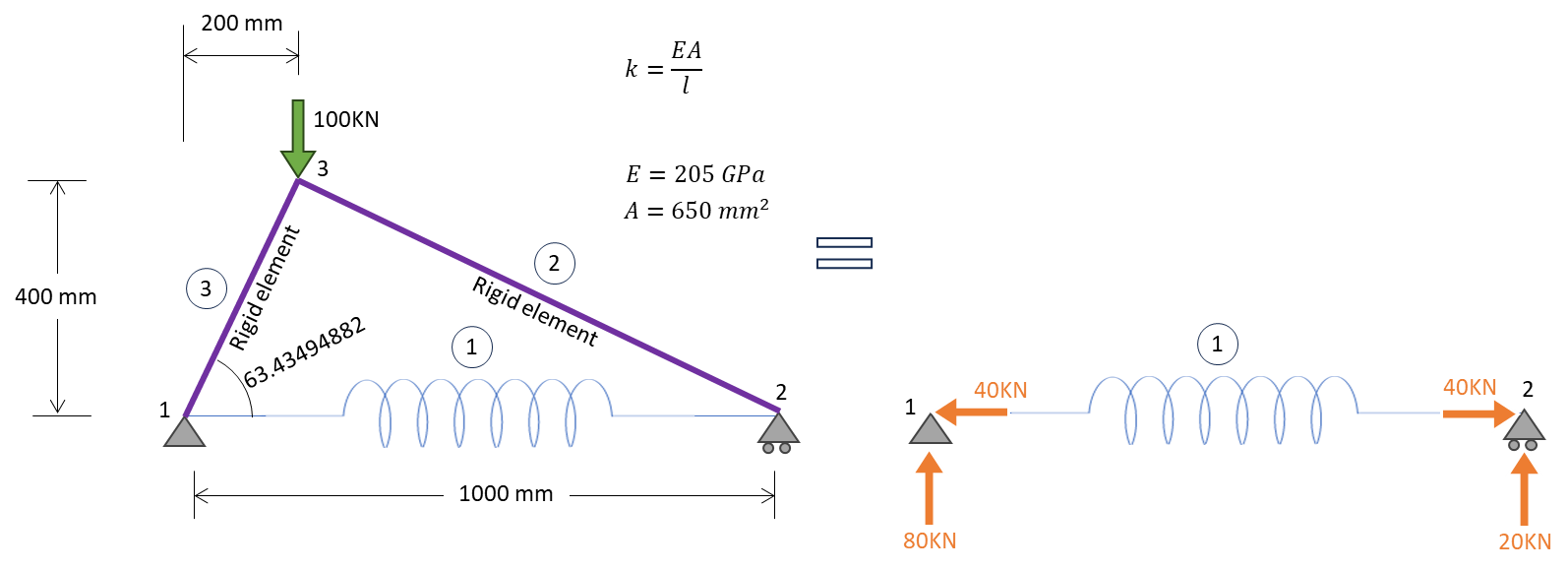
|  |
| --- |
| Algorithm: XPBD: Position-Based Simulation of Compliant Constrained Dynamics by Miles Macklin, Matthias M¨uller, Nuttapong Chentanez.  Input: Initial positions (xi0), velocities (vi0), masses (mi), external force function fext, time step Δt, solver iteration count  Output: Updated positions and velocities of particles  1. Initialize each particle i  **For each** particle i **do**  xi = xi0 // Set initial position  vi = vi0 // Set initial velocity  wi = 1 / mi // Compute inverse mass  **End For**  2. Simulation Loop  **Repeat**  // Update velocities with external force  **For each** particle i **do**  vi = vi + Δt \* wi \* fext(t, xi)  **End For**  // Update positions based on new velocities  **For each** particle i **do**  pi = xi + Δt \* vi  **End For**  // Solve constraints specified times  **For** 1 to solveriteration **do**  projectConstraints(C\_1, C\_2, ..., C\_M, p\_1, p\_2, ..., p\_N)  **End For**  // Correct velocities and update positions  **For each** particle i **do**  vi = (pi - xi) / Δt  xi = pi  End For  **Until** simulation ends |

## Lagrange Multiplier MPC

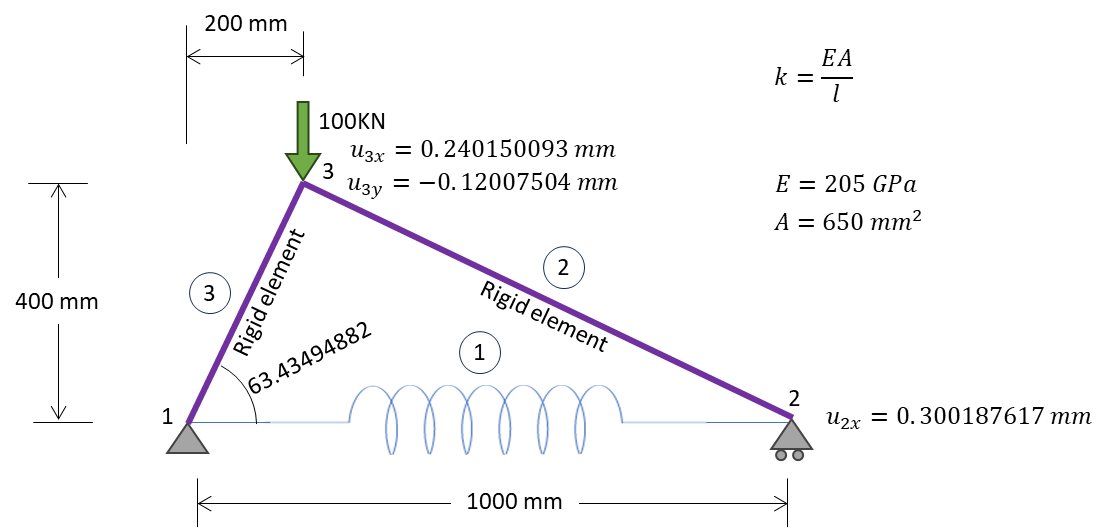
Inclined rigid element.



|  |  |  |
| --- | --- | --- |
|  | Element stiffness matrix |  |
|  | The direction cosines, |  |
|  | Direction cosine transformation matrix, |  |
|  | Element stiffness matrix | 1.0 |
|  | For element 1, |  |
|  |  | 2.0 |
|  | For element 2, |  |
|  |  | 3.0 |
|  | For element 3, |  |
|  |  |  |
|  |  | 4.0 |
|  | Global stiffness matrix | 5.0 |
|  | Appending the constraints to the global stiffness equation gives, |  |
|  |  | 6.0 |
|  | Adjunction process: Lagrange multiplier augmented bordered stiffness matrix |  |
|  |  |  |
|  | After applying the essential boundary condition. The matrix reduces to |  |
|  |  | 7.0 |
|  | Applying the stiffness values |  |
|  |  |  |
|  |  | 8.0 |
|  | Solving the above equation gives |  |
|  |  | 9.0 |
|  |  |  |
|  |  |  |



|  |  |  |
| --- | --- | --- |
|  | Exact solution is given by, |  |
|  |  |  |
|  |  | 1.0 |
|  | Rigid element 2 |  |
|  |  |  |
|  |  |  |
|  | Substituting and | 2.0 |
|  | Rigid element 3 |  |
|  |  |  |
|  |  |  |
|  | Substituting and | 3.0 |
|  | Solving eqn 2.0 and 3.0 |  |
|  |  |  |
|  |  |  |



## Generalized Eigen Value problem – Lagrange multiplier MPC

Below is the free vibration generalized eigen value problem of a dynamic system without damping.

|  |  |  |
| --- | --- | --- |
|  | Generalized eigen value problem of mass – spring system | 1.0 |
|  | MPC – Lagrange Multiplier method, multiplier – augmented system, bordered equation is, |  |
|  |  | 2.0 |
|  | The solution to the eigen value problem defined in eqn 2 is complicated by the presence of many null elements in the involved matrices. We can use change of variables to overcome this difficulty. |  |
|  | Change of variable | 3.0 |
|  |  | 4.0 |
|  | Substitute the eqn 4.0 in eqn 2.0 | 5.0 |
|  | Pre-multiply the eqn 5.0, by the below |  |
|  |  | 6.0 |
|  | On a side note, Is the following expression valid? |  |
|  | Then, | 7.0 |
|  | Substitute eqn 7.0 in eqn 6.0 Mass matrix null lower block. |  |
|  | Simplifying the above equation gives the below standard eigen value problem. | 8.0 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |