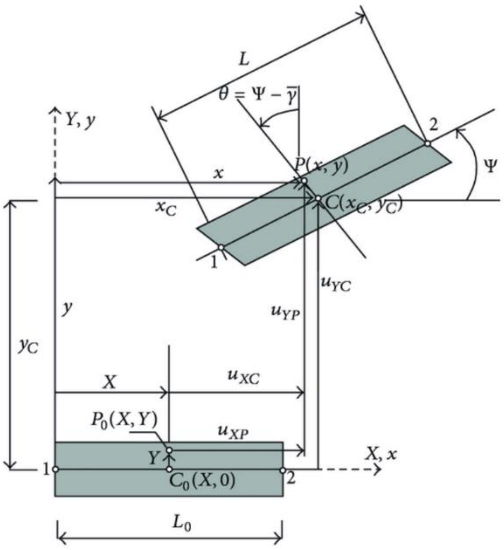
Shape-programmable magnetic soft matter using FEM

The aim of the project is to analyse the static behaviour of magnetic soft matter providing a solution in 2D of the desired magnetic field in order to obtain programmable shapes. It has been considered only a subgroup of soft robots build as parallelepiped (or inchworm).

In this report is presented a method that can be used to model the behaviour of the robots through plane beam element with FE formulation, assuming that:

1. The non-linear static analysis is considered. Only geometric non-linearity is considered. Material is considered to homogeneous, isotropic and linear elastic. Dynamical effects are neglected.
2. Deformations are large to use finite deformation theory (Geometric non-linearity).
3. *Bernoulli-Euler (BE) Model.* It accounts for bending moment effects on stresses and deformations. Transverse shear forces are recovered from equilibrium but their effect on beam deformations is ignored (the strain energy due to shear stresses is neglected).



The element kinematics of a plane beam is completely defined with the functions,

- *Axial displacement uX (X)*
- *Transverse displacement uY (X)*
- *Cross section rotation θZ (X) ≡ θ(X)*

Here *X* denotes the longitudinal coordinate in the reference configuration.

Two-node plane beam element are formulated using the Total Lagrangian (TL) kinematic description, if we assume the shear distortion is very small, it gives the simplified Lagrangian description of the motion,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X + u_x - Ysin\theta \\ u_y + Ycos\theta \end{bmatrix}$$

In the linear case, considering d as the displacement field and F the external load vector, algebraic equations of form $Kd = F$ are solved, but for non-linear systems equations need to be linearized around equilibrium point and solution must be sought by iterative procedure. For non-linear system either force or displacement can be a controlling parameter. Here load control is used, an iteration processes where the external load is increased at the start of the increment, by directly increasing the external force vector F_{ext} .

The problems encountered regard the computation of the internal forces within the iteration loop needed in order to obtain the residual(unbalance) vector and update the displacement field that account for the deformation of the beam. The internal force vector is obtained by taking the first variation of the internal energy with respect to the node displacements, exploiting the strain-displacement relationship matrix(or kinematic matrix). This expression is evaluated by a one point Gauss integration rule.

The implementation has been performed on Matlab, here follows a description of the incremental - iterative solution procedure.

The residual can be written as

$$R = F_{int} - F_{ext}$$

where F_{ext} and F_{int} are external and internal forces respectively. The increment in displacement is evaluated as

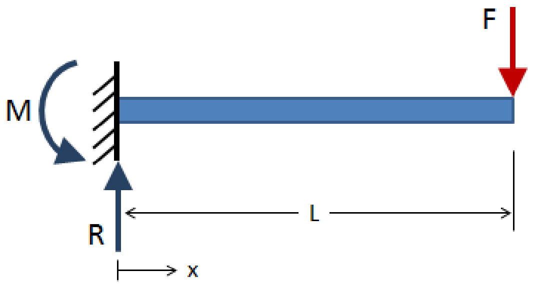
$$\Delta d_i = K_T^{-1} R_i$$

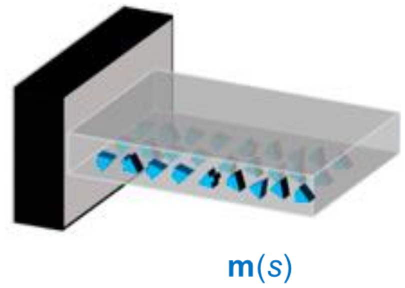
Where K_T is the tangent stiffness matrix, the updating step brings to

$$d_{i+1} = d_i + \Delta d_i$$

this is the equilibrium solution at a certain step load after iterating until residual becomes less than tolerance level (a quantity fixed a priori).

To simplify the implementation it has been used K, the linear stiffness matrix, in place of K_T. In this case the algorithm converges linearly. For a more refined model a tangent stiffness matrix can be also assembled by taking into account the Material Stiffness Matrix and the Geometric Stiffness Matrix, leading to a quadratic convergence.





The external load vector is computed at each step by the magnetic formulation solving at each node the following set of equation,

$$F_{ext} = \begin{bmatrix} (\boldsymbol{m} \cdot \nabla)\boldsymbol{B} \\ \boldsymbol{m} \times \boldsymbol{B} \end{bmatrix}$$

The results are partially correct if we deal with not so large displacement. A more accurate model for the internal forces is needed.

Recommendation for future works:

1. Checking and modifying the incremental program procedure and the Newton-Raphson iterations.
2. Including the tangent stiffness matrix K_T

31/07/2019 Pietro Pacciani