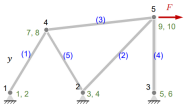
UML STRUCTURAL SOLVER



THIS UML DIAGRAM REPRESENTS
THE OPP STRUCTURE THAT HAS BEEN
FOLLOWED TO DESIGN A
STRCUTURAL NUMERIC SOLVER OF
ARTICULATED
MECHANICAL STRUCTURES

n_{d} : Problem dimension, i.e. $n_{\mathrm{d}} = 2$ (2D).
n_{el} : Total number of bars.
n_{nod} : Total number of nodes (joints).
n_{ne} : Number of nodes in a bar, i.e. $n_{\mathrm{ne}} = 2$.
n_{i} : Degrees of freedom per node, i.e. $n_{\mathrm{i}} = 2$.
n_{dof} : Total number of degrees of freedom, i.e. $n_{\mathrm{dof}} = n_{\mathrm{nod}} \times n_{\mathrm{i}}$.

$\mathbf{x}$: Nodal coordinates array ($n_{\mathrm{nod}} \times n_{\mathrm{d}}$):

$\mathbf{T}_{\mathrm{n}}$: Nodal connectivity table ($n_{\mathrm{el}} \times n_{\mathrm{ne}}$). Example:

$\mathbf{T}_{\mathrm{d}}$: Degrees of freedom connectivity table ($n_{\mathrm{el}} \times (n_{\mathrm{ne}} \times n_{\mathrm{i}})$).

GeometryComputer
geometry
obj = GeometryComputer(cParms)
obj.compute()

StifnessMatrixComputer
stifnessMatrix
obj = StifnessMatrixComputer(cParms)
obj.compute()

ElementalStifnessMatrixComputer
elementalStifnessMatrix
obj = ElementalStifnessMatrixComputer(cParms)
obj.compute()

SplitConditionComputer
splitedCondition
obj = SplitConditionComputer(cParms)
obj.compute()

ForceComputer
externalForces
obj = forceComputer(cParms)
obj.compute()

SplitStifnessMatrizComputer
splitedStifnessMatrix
obj = SplitStifnessMatrizComputer(cParms)
obj.compute()

SplitExternalForceComputer
splitedForce
obj =SplitExternalForceComputer(cParms)
obj.compute()

ElementGeometryComputer
lenght
se
ce
obj =ElementGeometryComputer(cParms)
obj.compute()

DisplacementReactionComputer
displacement
reactionForces
obj = DisplacementReactionComputer(cParms)
obj.compute()

SplitComputer
splitedCondition
splitedStifnessMatrix
splitedForce
obj =SplitComputer(cParms)
obj.compute()

DirectSolver
obj.solve()

SystemSolver
B
D
obj =SystemSolver(cParms)
obj.solve()

MatrixBsolver
B
splitedB
obj =MatrixBsolver(cParms)
obj.solve()

IterativeSolver
obj.solve()

structureComputer
Fext
u
KG
FB
obj = structureComputer(cParms)
obj.compute()

$$\begin{bmatrix} AA & AB \\ AC & AD \end{bmatrix} \cdot \begin{bmatrix} BA \\ BB \end{bmatrix} = \begin{bmatrix} CA \\ CB \end{bmatrix} \cdot \begin{bmatrix} DA \\ 0 \end{bmatrix}$$

$$[A] \cdot [B] = [C] + [D]$$

MatrixDsolver
D
splitedD
obj =MatrixDsolver(cParms)
obj.solve()

StressStrainComputer
stress
strain
obj = StressStrainComputer(cParms)
obj.compute()

bucklingFailureComputer
FB
obj = forceComputer(cParms)
obj.compute()

matrixRotationComputer
le
Re
obj =solverMethodCompute(cParms)
obj.compute()