







**Modification to FAAMeshClassLib.vb**

|  |
| --- |
| **ReadLoaFile (replaced by sub CreateRgdParms)**  'ACLoad(iAC).XGrid = CSng(LineInput(1))  'ACLoad(iAC).YGrid = CSng(LineInput(1))  ReDim ACLoad(iAC).XGrid(ACLoad(iAC).NSlabs), ACLoad(iAC).YGrid(ACLoad(iAC).NSlabs)  For iSlab = 1 To ACLoad(iAC).NSlabs  For Each m In reTwoNumPara.Matches(LineInput(1))  ACLoad(iAC).XGrid(iSlab) = CSng(m.Groups("para1").Value)  ACLoad(iAC).YGrid(iSlab) = CSng(m.Groups("para2").Value)  Next  Next iSlab |
| **NodeBC**  'Call NodeBC() |
| **NodebyPart**  Dim iPartSld, IdxSldTemp As Integer, TypSldSmTemp As String  Dim NdSldX1Temp, NdSldY1Temp, NdSldZ1Temp, NdSldX2Temp, NdSldY2Temp, NdSldZ2Temp As Integer  Dim SldX1Temp, SldY1Temp, SldZ1Temp, SldX2Temp, SldY2Temp, SldZ2Temp As Single  Dim iPartBC, BXYZTemp, BRXYZTemp As Integer  Dim NdBCX1Temp, NdBCY1Temp, NdBCZ1Temp, NdBCX2Temp, NdBCY2Temp, NdBCZ2Temp As Integer  Dim BCX1Temp, BCY1Temp, BCZ1Temp, BCX2Temp, BCY2Temp, BCZ2Temp As Single  'GoTo iNodeloop  GoTo iPartSldlabel    iPartSldlabel:  IdxSldTemp = 0  TypSldSmTemp = Nothing  For iPartSld = 1 To NPartSld  If PARTSLD(iPartSld).IdxPart = iPart And \_  iXCtr >= PARTSLD(iPartSld).SldNdIdxCtrIni(1) And iXCtr <= PARTSLD(iPartSld).SldNdIdxCtrEnd(1) And \_  iYCtr >= PARTSLD(iPartSld).SldNdIdxCtrIni(2) And iYCtr <= PARTSLD(iPartSld).SldNdIdxCtrEnd(2) And \_  iZCtr >= PARTSLD(iPartSld).SldNdIdxCtrIni(3) And iZCtr <= PARTSLD(iPartSld).SldNdIdxCtrEnd(3) Then  IdxSldTemp = PARTSLD(iPartSld).SldIdx  TypSldSmTemp = PARTSLD(iPartSld).SldSmTyp  NdSldX1Temp = PARTSLD(iPartSld).SldNdIdxCtrIni(1)  NdSldY1Temp = PARTSLD(iPartSld).SldNdIdxCtrIni(2)  NdSldZ1Temp = PARTSLD(iPartSld).SldNdIdxCtrIni(3)  NdSldX2Temp = PARTSLD(iPartSld).SldNdIdxCtrEnd(1)  NdSldY2Temp = PARTSLD(iPartSld).SldNdIdxCtrEnd(2)  NdSldZ2Temp = PARTSLD(iPartSld).SldNdIdxCtrEnd(3)  SldX1Temp = PART(iPart).CoordXCtr(NdSldX1Temp)  SldX2Temp = PART(iPart).CoordXCtr(NdSldX2Temp)  SldY1Temp = PART(iPart).CoordYCtr(NdSldY1Temp)  SldY2Temp = PART(iPart).CoordYCtr(NdSldY2Temp)  SldZ1Temp = PART(iPart).CoordZCtr(NdSldZ1Temp)  SldZ2Temp = PART(iPart).CoordZCtr(NdSldZ2Temp)  GoTo iPartBClabel  End If  Next iPartSld  iPartBClabel:  BXYZTemp = 0  BRXYZTemp = 7 'default non-rotational node  For iPartBC = 1 To NPartBC  If PARTBC(iPartBC).IdxPart = iPart And \_  iXCtr >= Math.Max(PARTBC(iPartBC).BCNdIdxCtrIni(1), 1) And iXCtr <= Math.Max(PARTBC(iPartBC).BCNdIdxCtrEnd(1), 1) And \_  iYCtr >= Math.Max(PARTBC(iPartBC).BCNdIdxCtrIni(2), 1) And iYCtr <= Math.Max(PARTBC(iPartBC).BCNdIdxCtrEnd(2), 1) And \_  iZCtr >= Math.Max(PARTBC(iPartBC).BCNdIdxCtrIni(3), 1) And iZCtr <= Math.Max(PARTBC(iPartBC).BCNdIdxCtrEnd(3), 1) Then ' Math.Max function to change 0 to 1  BXYZTemp = CInt(PARTBC(iPartBC).StrBCValue.Substring(0, 1)) \* CInt(2 ^ 2) \_  + CInt(PARTBC(iPartBC).StrBCValue.Substring(1, 1)) \* CInt(2 ^ 1) \_  + CInt(PARTBC(iPartBC).StrBCValue.Substring(2, 1)) \* CInt(2 ^ 0)  BRXYZTemp = CInt(PARTBC(iPartBC).StrBCValue.Substring(3, 1)) \* CInt(2 ^ 2) \_  + CInt(PARTBC(iPartBC).StrBCValue.Substring(4, 1)) \* CInt(2 ^ 1) \_  + CInt(PARTBC(iPartBC).StrBCValue.Substring(5, 1)) \* CInt(2 ^ 0)  NdBCX1Temp = Math.Max(PARTBC(iPartBC).BCNdIdxCtrIni(1), 1)  NdBCY1Temp = Math.Max(PARTBC(iPartBC).BCNdIdxCtrIni(2), 1)  NdBCZ1Temp = Math.Max(PARTBC(iPartBC).BCNdIdxCtrIni(3), 1)  NdBCX2Temp = Math.Max(PARTBC(iPartBC).BCNdIdxCtrEnd(1), 1)  NdBCY2Temp = Math.Max(PARTBC(iPartBC).BCNdIdxCtrEnd(2), 1)  NdBCZ2Temp = Math.Max(PARTBC(iPartBC).BCNdIdxCtrEnd(3), 1)  BCX1Temp = PART(iPart).CoordXCtr(NdBCX1Temp)  BCX2Temp = PART(iPart).CoordXCtr(NdBCX2Temp)  BCY1Temp = PART(iPart).CoordYCtr(NdBCY1Temp)  BCY2Temp = PART(iPart).CoordYCtr(NdBCY2Temp)  BCZ1Temp = PART(iPart).CoordZCtr(NdBCZ1Temp)  BCZ2Temp = PART(iPart).CoordZCtr(NdBCZ2Temp)  GoTo iNodeloop  End If  Next iPartBC  'Node sliding  If XND > (Math.Min(SldX1Temp, SldX2Temp) - ErrTol) \* ACLoad(1).ScaleX And XND < (Math.Max(SldX1Temp, SldX2Temp) + ErrTol) \* ACLoad(1).ScaleX And  YND > (Math.Min(SldY1Temp, SldY2Temp) - ErrTol) \* ACLoad(1).ScaleY And YND < (Math.Max(SldY1Temp, SldY2Temp) + ErrTol) \* ACLoad(1).ScaleY And  ZND > (Math.Min(SldZ1Temp, SldZ2Temp) - ErrTol) And ZND < (Math.Max(SldZ1Temp, SldZ2Temp) + ErrTol) Then  Node(iNode).IdxSld = IdxSldTemp  Node(iNode).TypSldSm = TypSldSmTemp  End If  'Node BC  If XND > (Math.Min(BCX1Temp, BCX2Temp) - ErrTol) \* ACLoad(1).ScaleX And XND < (Math.Max(BCX1Temp, BCX2Temp) + ErrTol) \* ACLoad(1).ScaleX And  YND > (Math.Min(BCY1Temp, BCY2Temp) - ErrTol) \* ACLoad(1).ScaleY And YND < (Math.Max(BCY1Temp, BCY2Temp) + ErrTol) \* ACLoad(1).ScaleY And  ZND > (Math.Min(BCZ1Temp, BCZ2Temp) - ErrTol) And ZND < (Math.Max(BCZ1Temp, BCZ2Temp) + ErrTol) Then  Node(iNode).BXYZ = BXYZTemp  Node(iNode).BRXYZ = BRXYZTemp  End If  If ACLoad(1).IndxSy = 1 Then 'symmetric to y-axial  If Math.Abs(Node(iNode).Y - 0) < ErrTol Then  Node(iNode).BXYZ = 2  End If  End If  'infinite element BC  For iNode = 1 To NNode  If Math.Abs(Node(iNode).Z - ZMin) < ErrTol Then  Node(iNode).BXYZ = 7  End If  Next iNode |
| **NodeMerge**  'Dim iPart As Integer  'Dim iPartSld As Integer  'Dim NdSldX1Temp, NdSldY1Temp, NdSldZ1Temp, NdSldX2Temp, NdSldY2Temp, NdSldZ2Temp As Integer  'Dim X1Temp, Y1Temp, Z1Temp, X2Temp, Y2Temp, Z2Temp As Single  'For iPartSld = 1 To NPartSld  ' iPart = PARTSLD(iPartSld).IdxPart  ' NdSldX1Temp = PARTSLD(iPartSld).SldNdIdxCtrIni(1)  ' NdSldY1Temp = PARTSLD(iPartSld).SldNdIdxCtrIni(2)  ' NdSldZ1Temp = PARTSLD(iPartSld).SldNdIdxCtrIni(3)  ' NdSldX2Temp = PARTSLD(iPartSld).SldNdIdxCtrEnd(1)  ' NdSldY2Temp = PARTSLD(iPartSld).SldNdIdxCtrEnd(2)  ' NdSldZ2Temp = PARTSLD(iPartSld).SldNdIdxCtrEnd(3)  ' X1Temp = PART(iPart).CoordXCtr(NdSldX1Temp)  ' X2Temp = PART(iPart).CoordXCtr(NdSldX2Temp)  ' Y1Temp = PART(iPart).CoordYCtr(NdSldY1Temp)  ' Y2Temp = PART(iPart).CoordYCtr(NdSldY2Temp)  ' Z1Temp = PART(iPart).CoordZCtr(NdSldZ1Temp)  ' Z2Temp = PART(iPart).CoordZCtr(NdSldZ2Temp)  ' For iNode = 1 To NNode  ' If Node(iNode).X > (Math.Min(X1Temp, X2Temp) - ErrTol) \* ACLoad(1).ScaleX And Node(iNode).X < (Math.Max(X1Temp, X2Temp) + ErrTol) \* ACLoad(1).ScaleX And  ' Node(iNode).Y > (Math.Min(Y1Temp, Y2Temp) - ErrTol) \* ACLoad(1).ScaleY And Node(iNode).Y < (Math.Max(Y1Temp, Y2Temp) + ErrTol) \* ACLoad(1).ScaleY And  ' Node(iNode).Z > (Math.Min(Z1Temp, Z2Temp) - ErrTol) And Node(iNode).Z < (Math.Max(Z1Temp, Z2Temp) + ErrTol) Then  ' Node(iNode).IdxSld = PARTSLD(iPartSld).SldIdx  ' Node(iNode).TypSldSm = PARTSLD(iPartSld).SldSmTyp  ' End If  ' Next  'Next iPartSld |
|  |