Algorithm 1 Plate Creator

procedure PLATE CREATOR ▷ Create a series of WARP3D input files
  Read lists of $\frac{a}{t}$, $\frac{a}{c}$, $\frac{E}{S_{\text{ys}}}$, n values from configuration
  Read t, S_{ys}, global .elt file from configuration
  for all $\frac{a}{c}$ values **do**
    for all $\frac{a}{t}$ values **do**
      $(a, c, L, W, S_{\text{in}}, S_{\text{out}}) \leftarrow$ SET GEOMETRY(global .elt file, $\frac{a}{c}$, $\frac{a}{t}$, t)
      generic mesh $\leftarrow$ BUILD MESH(.elt file, $a, c, t, L, W, S_{\text{in}}, S_{\text{out}}$)
      for all $\frac{E}{S_{\text{ys}}}$ values **do**
        $E \leftarrow (\frac{E}{S_{\text{ys}}})(S_{\text{ys}})$
        for all n values **do**
          WARP3D input $\leftarrow$ SET MATERIAL(generic mesh, E, S_{ys}, n)
        end for
      end for
    end for
  end for
end procedure

Algorithm 2 Set Geometry

procedure SET GEOMETRY(global .elt file, $a/c, a/t, t$)
  $a \leftarrow (a/t)(t)$
  $c \leftarrow a/(a/c)$
  if $c > t$ **then**
    $W \leftarrow 5c$
  else
    $W \leftarrow 5t$
  end if
  model type $\leftarrow$ GET TYPE(global .elt file)
  if model type = 'bending' **then**
    $S_{\text{in}} \leftarrow W$
    $S_{\text{ou}} \leftarrow 2W$
    if $2W > 1.1S_{\text{out}}$ **then**
      $L \leftarrow 2W$
    else
      $L \leftarrow 1.1S_{\text{out}}$
    end if
  else
    $S_{\text{in}} \leftarrow$ Null
    $S_{\text{out}} \leftarrow$ Null
    $L \leftarrow 2W$
  end if
  return $(a, c, W, L, S_{\text{in}}, S_{\text{out}})$
end procedure

Algorithm 3 Get Type

procedure GET TYPE(global .elt file)
  if '"*use bottom load pin plate ty ' found in 'Notes:' field **then**
    if 'RigidSurfaceData_Radius' found twice **then**
      if 'RigidSurfaceData_PinLocation' found twice **then**
        model type $\leftarrow$ 'bending'
      else
        model type $\leftarrow$ 'invalid'
      end if
    end if
  else
    model type $\leftarrow$ 'tension'
  end if
end procedure

Algorithm 4 Build Mesh

procedure BUILD MESH(global .elt file, a, c, t, L, W, S_{inner}, S_{outer})
  model .elt file ← GET ELT FILENAME(global .elt file, a, c, L, W, t)
  Copy global .elt file to model .elt file
  if $S_{\text{in}} \neq$ Null **and** $S_{\text{out}} \neq$ Null **then**
    ADJUST ROLLER POSITIONS(model .elt file, S_{in}, S_{out})
  end if
  Run FEACrack program on model .elt file, using a, $2c$, t, L, and W
  return generic mesh file ▷ 'tens_ac1.0_at0.8_L10.00_W05.00_wrp.inp'
end procedure

Algorithm 5 Get Elt Filename

procedure GET ELT FILENAME(global .elt file, a, c, L, W, t)
  prefix ← global .elt file basename ▷ remove .elt extension from file
  middle ← '_ac($\frac{a}{c}$)_at($\frac{a}{t}$)_L(L)_W(W)'
  suffix ← '.elt'
  return prefix + middle + suffix ▷ concatenate terms
end procedure

Algorithm 6 Adjust Roller Positions

procedure ADJUST ROLLER POSITIONS(model .elt file, S_{in}, S_{out})
  if first 'RigidSurfaceData_PinLocation' found **then**
    Change z value of location to S_{in}
  end if
  if second 'RigidSurfaceData_PinLocation' found **then**
    Change z value of location to S_{out}
  end if
end procedure

Algorithm 7 Set Material

procedure SET MATERIAL(generic mesh, E, S_{ys}, n)
  if 'stress-strain curve 1' found **then**
    change stress-strain curve data to LPPL(E, S_{ys}, n)
  end if
end procedure

Algorithm 8 LPPL

procedure LPPL(E, S_{ys}, n)
$\epsilon_{\text{pl1}} \leftarrow (0.001, 0.002, \cdots, 0.008)$
$\epsilon_{\text{pl2}} \leftarrow (0.013, 0.018, 0.023, 0.028)$
$\epsilon_{\text{pl3}} \leftarrow (0.038, 0.048, \cdots, 0.108)$
$\epsilon_{\text{ys}} \leftarrow \frac{S_{\text{ys}}}{E}$
$\epsilon \leftarrow \epsilon_{\text{ys}} + (\epsilon_{\text{pl1}}, \epsilon_{\text{pl2}}, \epsilon_{\text{pl3}})$
$\sigma \leftarrow S_{\text{ys}}(\frac{\epsilon}{\epsilon_{\text{ys}}})^{\frac{1}{n}}$
return ϵ, σ
end procedure

Algorithm 9 Get Model Filename

procedure GET MODEL FILENAME(generic mesh, E, S_{ys}, n)
prefix $\leftarrow$ global .elt file basename ▷ remove .elt extension from file
middle $\leftarrow$ '_ac($\frac{a}{c}$)_at($\frac{a}{t}$)_L(L)_W(W)'
suffix $\leftarrow$ '.elt'
return prefix + middle + suffix ▷ concatenate terms
end procedure