Tutorial on the contact problem in GiD

Marko Leskovar

March 19, 2020

1 Introduction

The focus of this tutorial is to show all the necessary steps in the set-up of a GiD input file that is used in FEM contact analysis. GiD only acts as a pre-processor while the actual calculation is carried out in the MATLAB script. Setting up a contact problem is no different from setting up a classic FEM plate in membrane action analysis, the only difference here is that we also specify the contact surfaces (lines).

2 Problem Type

When first starting the GiD, the user must specify the problem type (Figure 1). Select *Data* → *Problem Type* → *Load...* and find the folder where the MATLAB code is saved and then select the folder *problemTypeGiD/matlab*. This is a custom problem type made specifically for the interface between GiD and MATLAB. User is free to add more features by expanding the scripts in folder *matlab.gid*.

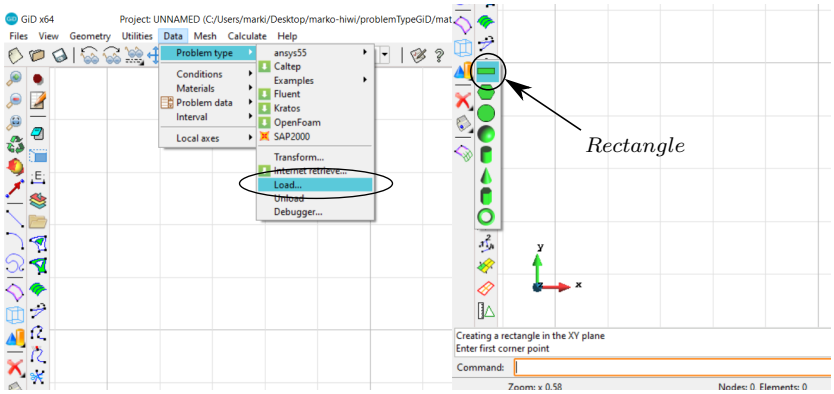

Figure 1: Problem type and geometry selection.

3 Geometry

To create geometry, select *create object* → *rectangle*. You can draw the rectangle by clicking on the drawing plane or specifying coordinates of the edges

(Figure 1). Coordinates must be in the format: **x y z**. They must contain white space between each coordinate. Enter the first point **−3 0** in the command line and confirm with *esc*. Now enter the second point **3 2** and again confirm with *esc*. Now we can create a circle from coordinate center **0 0** with the normal in positive Z direction and radius **1**.

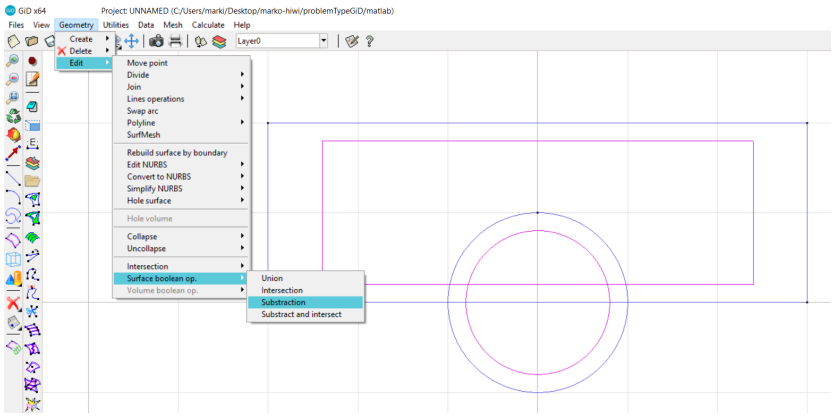

Figure 2: Subtracting circle from rectangle.

We have two unconnected surfaces as shown in Figure 2. Next step is to subtract the circle from the rectangle to get the desired shape. Click on *Geometry* → *Edit* → *Surface Boolean op.* → *Substraction*. Now first select the surface to subtract from (rectangle), confirm with *esc* and select the subtracting surface (circle). Our geometry is now complete (Figure 3). At this point it is recommended to save the project.

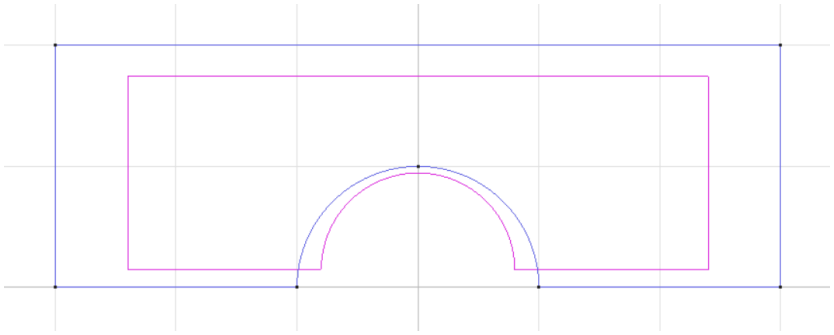

Figure 3: Final geometry.

If you are unsure what to click or how to use certain commands, take a look at console output above the command line. Just be aware that simple *ctrl+Z* undo command does not exist in GiD.

4 Boundary conditions

To specify the Dirichlet boundary conditions go to *Data* → *Conditions* → *Constrains*. Select *lines* (line icon) as selection type and select *Structure-Dirichlet*. Fix both **x** and **y** coordinate on bottom left line as shown in Figure 4. Please be aware that if you again select *Structure-Dirichlet* boundary conditions on the same line or node it overwrites the previous selection.

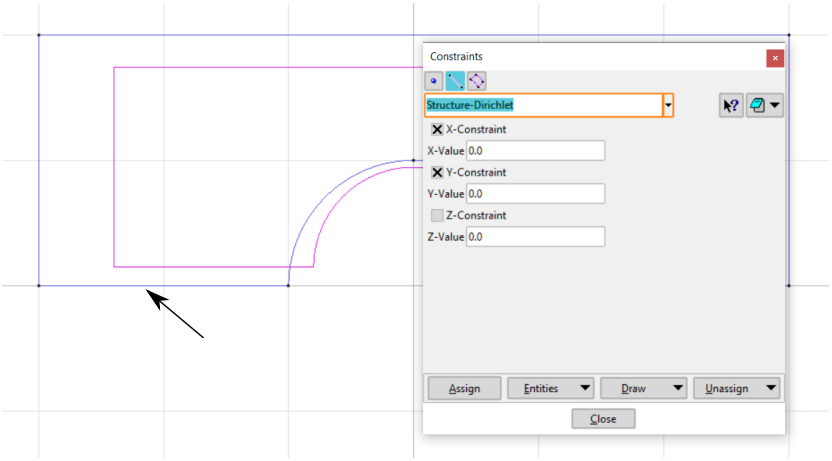

Figure 4: Applied Dirichlet boundary conditions.

5 Loads

To apply loads on the structure go to *Data* → *Conditions* → *Loads*. Now we need to specify the function handle to the load computation function that is implemented in the MATLAB script. Change the default *functionHandle* to the *computeConstantVerticalLoad* and apply the load on the top line according to Figure 5.

The function *computeConstantVerticalLoad* is self-explanatory as it only applies constant vertical load on the structure. The magnitude of the load must be specified in the MATLAB function itself. Another useful function for specifying loads is *computeConstantHorizontalLoad*. For more options please check the functions in folder *FEMPlateInMembraneActionAnalysis/loads* or create new functions in a similar manner.

6 Contact Nodes

To select the possible contact surfaces, go to *Data* → *Conditions* → *Contact* and select the whole boundary as possible contact line like shown in Figure 6. This is the safest option if we are not sure where the contact will actually occur.

If on the other hand we already know the exact contact line, we can only specify that as a possible contact and significantly reduce the computation time. Be

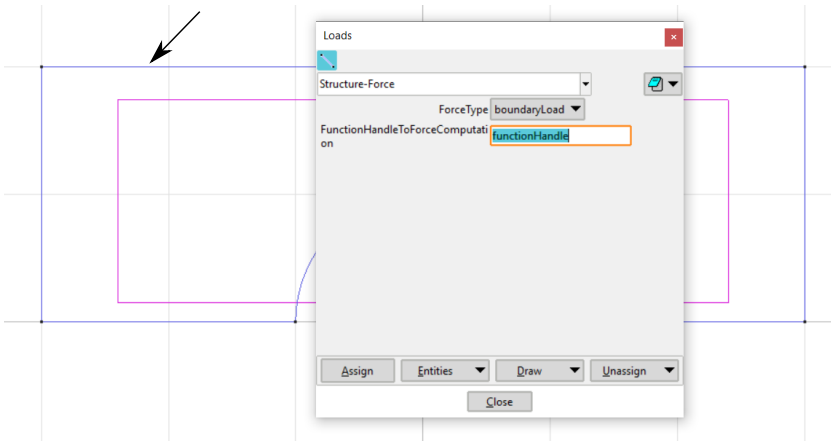

Figure 5: Applied constant vertical load.

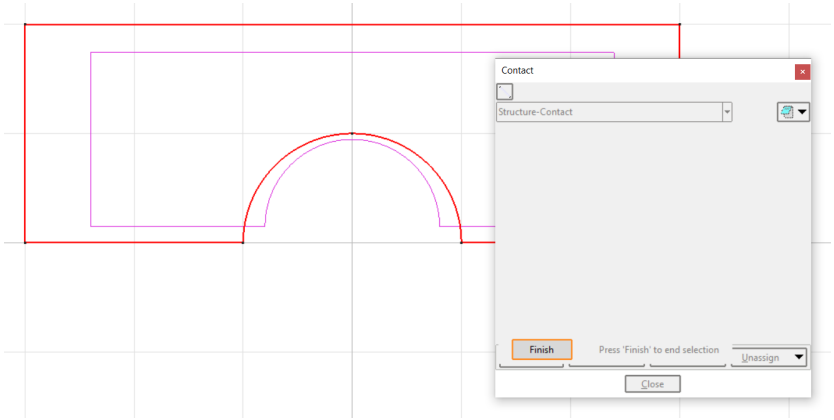

Figure 6: Defined possible contact boundary.

aware that if the magnitude of applied load is too big the resulting displacements are not linear. This in turn can cause problems after the first solver iteration where no contact nodes are detected due to violation of linear gap function.

7 Domains

We need to specify the domain of the mesh elements and nodes that will be generated in the GiD output file. Go to *Data* → *Conditions* → *Domains*, choose *Structure-Nodes* from the dropdown menu and select the whole surface according to Figure 7. Confirm with *esc*. Now select *Structure-Elements* and repeat the previous step.

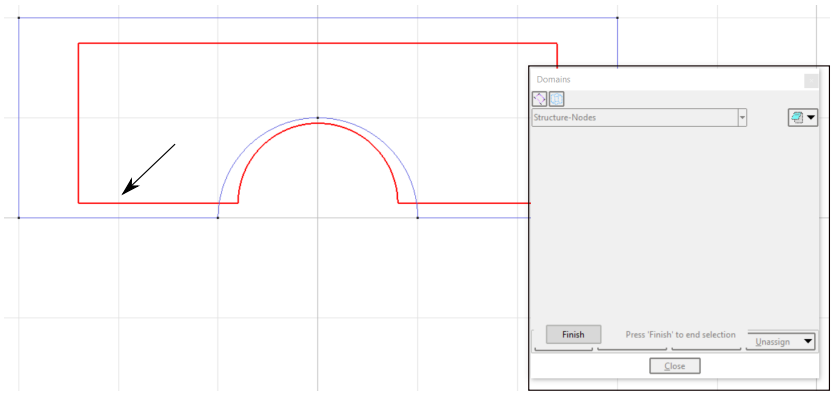

Figure 7: Defined domain for the mesh elements and nodes.

8 Materials

To select material, go to *Data* → *Materials* → *Solids*. Here we can select between two default materials *Steel* and *Aluminum* from the drop-down menu (Figure 8) or just change the given parameters to adjust material properties. To apply the material to the geometry click *Assign* → *Surfaces* and select the surface of interest. User can also expand the materials selection by editing the scripts in folder *matlab.gid*.

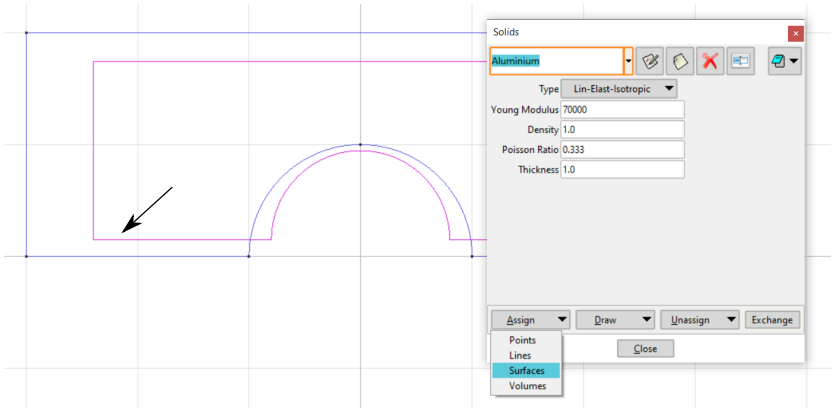

Figure 8: Define material over surfaces.

9 Mesh

The only thing left to do is to generate finite element mesh. The simplest way is to go to *Mesh* → *Generate Mesh...* and specify the element size (Figure 9). User can also choose between CST and Quadrilateral elements, both are implemented in MATLAB script and work with the contact analysis problem. For more details about the mesh generation please look at the official GiD documentation.

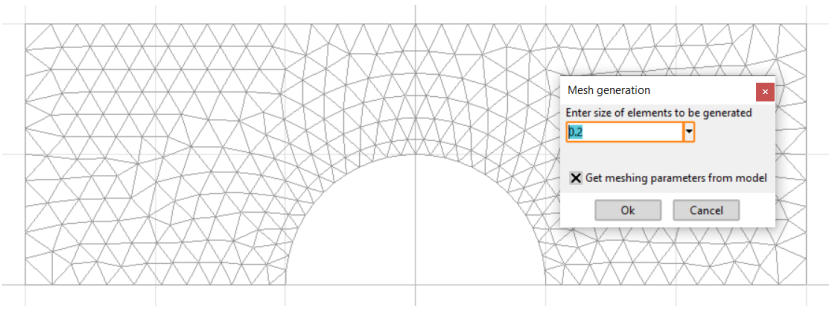

Figure 9: Mesh generation with mesh size.

10 Analysis type and solver setup

Go to *Data* → *Problem Data* → *Structural Analysis* to access the detailed setup of the analysis type and solver. Here we can choose between plane stress or plane strain and further specify if we have a steady state or a transient analysis. Many more settings are possible, but not all work (are not needed) with the FEM contact analysis. In the scope of this tutorial we can leave all the setting on their default options.

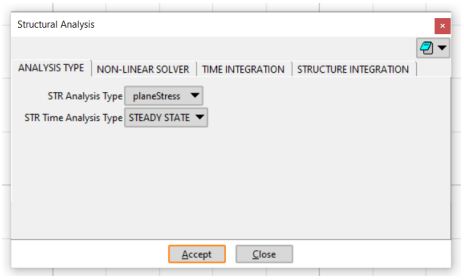

Figure 10: Analysis type and solver setup.

11 Generation of input file and analysis

The setup is now completed, go to *Calculation* → *Calculation (F5)* or just click *F5* to prepare the input file (Figure 11) to be used in MATLAB. This file has the same name as our project and extension *.dat*. You can also open it with any text editor to check or adjust the data. Please place this file into folder *inputGiD/FEMContactLinearPlateInMembraneAction* and add new *caseName* with the same name in the MATLAB main driver located in */main/main_contactMechanicsAnalysis*.

12 Rigid boundary

User must manually create the rigid boundary in the MATLAB script like shown in Figure 12. Rigid boundary consists of multiple segments that are indepen-

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
%   Structural Boundary Value Problem                                          %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

STRUCTURE_ANALYSIS
 ANALYSIS_TYPE,planeStress

STRUCTURE_MATERIAL_PROPERTIES
 DENSITY,1.0
 YOUNGS_MODULUS,1e5
 POISSON_RATIO,0.3
 THICKNESS,1.0

STRUCTURE_NLINEAR_SCHEME
 NLINEAR_SCHEME,NEWTON_RAPHSON
 NO_LOAD_STEPS,1
 TOLERANCE,1e-9
 MAX_ITERATIONS,100

STRUCTURE_TRANSIENT_ANALYSIS
 SOLVER STEADY_STATE
```

Figure 11: GiD input file in detail.

dent from each other and do not have to be connected. It is recommended for individual segments to be longer than the size of mesh elements. Orientation of each segment has to be respected by the user so that the normal vectors are orientated towards the body of interest. To create the rigid boundary please look at the MATLAB script for examples. You can simply add segments by specifying start and end points or choose to create a circular boundary via the included function.

```
%% Rigid wall- line  [(x0,y0) ; (x1,y1)]

% different line segments for different cases
if strcmp(caseName,'example_01_bridge')

    % Either define bottom contact line segment and add it to the segments
    segments.points(:,:,1) = [-0.5, -0.5; 1.2,-0.1];
    segments.points(:,:,2) = [1.2, -0.1; 2,-0.1];
    segments.points(:,:,3) = [2, -0.1; 4.5,-0.5];

    % ...or define a circular contact boundary
    center = [2,-4.1];
    radius = 4;
    startAngle = 3*pi/4;
    endAngle = pi/4;
    nSegments = 20;

    % Create circular segments
%     segments = createCircleSegments(center,radius,startAngle,endAngle,nSegments);
```

Figure 12: Part of MATLAB script where the rigid boundary is defined.