AUTODESK UNIVERSITY

ES15357

Dynam(o)ite Your Rebar Design

Dieter Vermeulen - Autodesk

Learning Objectives

- Discuss the power of computational design for reinforcement detailing
- Understand the basic rebar modeling solutions in Revit
- Learn how to make your own automated rebar design with Dynamo
- Learn how to apply computational rebar design in your Revit models with Dynamo

Description

Increasingly, our Building Information Modeling (BIM) projects require the delivery of detailed models. In the reinforced concrete industry, it is becoming a standard to create fabrication-ready models as manufacturing-centric approaches are entering the architecture, engineering, and construction (AEC) world. In this class, you will discover the effect when computational design meets reinforcement detailing. Do you want to get introduced to the world of computational rebar design? Then this session will familiarize you with the Autodesk, Inc., technology for complex reinforcement detailing with Revit software and Dynamo software. In this powerful 90 minutes, you will see a few examples, from the automation of basic rebar design over radial fan-shaped rebars to complex irregular elements. After this session, you will be inspired by the power and dynamics of parametric and computational design for complex problems. This session features Revit Structure and Dynamo.

Your AU Expert

Working as a Technical Sales Specialist AEC for the Northern European region at Autodesk, **Dieter Vermeulen** is specialized in the products of the structural solutions portfolio. Within that domain he supports the channel partners and customers with workflow solutions, especially - but not limited to - for design and engineering. With Revit, Robot Structural Analysis and Dynamo as his sidekicks, he is offering BIM workflow solutions covering the building process from design over analysis to fabrication for steel and concrete constructions. He has over 15 years of experience in the structural engineering business, starting his career in 2000 at Jacobs Engineering in Belgium.

Twitter	www.twitter.com/BIM4Struc	Team Blog	www.autodesk.typepad.com/bimtoolbox/
LinkedIn	www.linkedin.com/in/dietervermeulen	YouTube	www.youtube.com/user/RevitbeyondBIM
Blog	www.revitbeyondbim.wordpress.com		

Table of Contents

Computational Design

What is computational design?

The last few years we all experienced the transition from traditional Computer Aided Design (CAD) to Building Information Modelling (BIM). As our projects are getting more complex, and need more design solutions, the current BIM focused workflow is challenged again. The answer on that is **Computational Design**. This new design approach represents a profound shift in design thinking and methods. Representation is being replaced by simulation, and the crafting of objects is moving towards the generation of integrated systems through designer-authored computational processes.

While there is a particular history of such an approach in the building industry, its relative newness requires the continued progression of new modes of design thinking for the designers and engineers of the 21st century.

Literally, "computational design" means "giving shape with the means of the power of a computer and scripting".

Autodesk Dynamo

Autodesk gives an answer to this new challenge in our design world. This solution is called **Autodesk® Dynamo** (open source) and **Autodesk® Dynamo Studio** (Subscription). Dynamo lets designers and engineers create visual logic to explore parametric designs and automate tasks. It helps you to solve challenges faster by designing workflows that drive the geometry and behavior of design models. With Dynamo you will extend your designs into interoperable workflows for documentation, fabrication, coordination, simulation, and analysis.

Getting started

For better understanding of this handout, a basic knowledge of Dynamo and Revit is advised. A comprehensive list of learning resources is listed at the end of this handout.

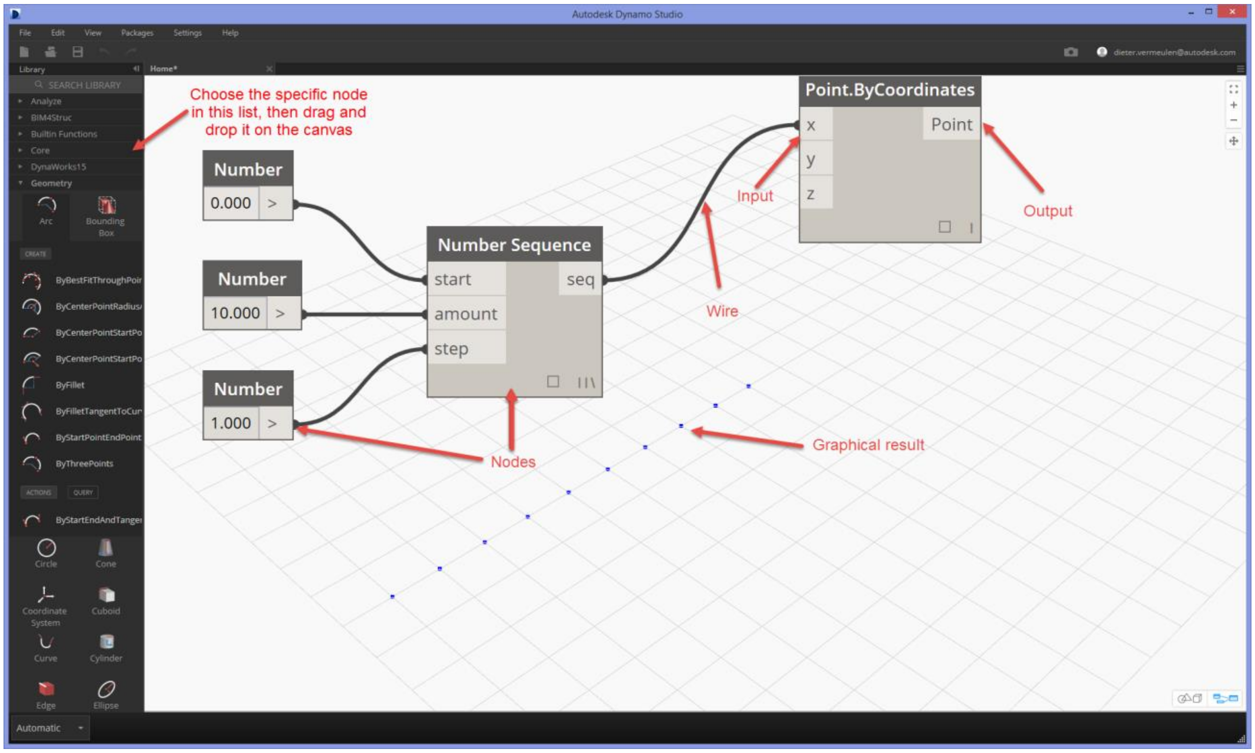


FIG. 1 - DYNAMO GRAPHICAL USER INTERFACE

Used software

Autodesk Software

The software that is used to work out the examples in the next chapters:

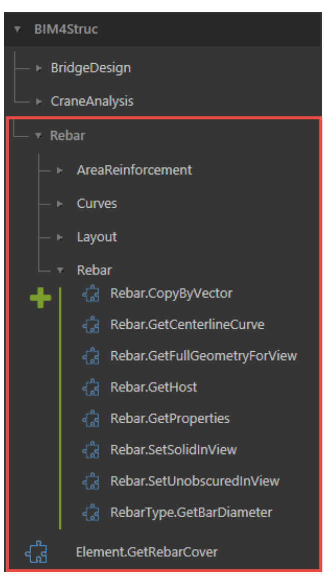
- **Autodesk Revit 2017.1**
  Used for setting up the base models
  More info on: http://www.autodesk.com/products/revit-family/overview

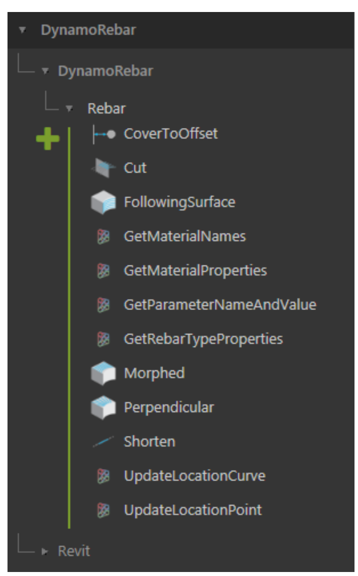
- **Autodesk Dynamo 1.2.0**
  Free open-source tool used for computational design approach of the presented construction problems.
  More info on: http://www.dynamobim.org

Dynamo Packages

- **BIM4Struc.Rebar**
  The BIM4Struc.Rebar package offers nodes that increase your productivity for Rebar modelling in Revit. The nodes don't only allow you to get data from existing rebar such as sketch lines, geometrical properties, … but they also give access to automation of Rebar tasks (from the API) such as representation settings of rebar, rebar layout rules setting... Combining these nodes with the rebar creation nodes (from the Dynamo for Rebar package) make it possible to create almost any complex rebar model. Look at the "Extra" folder for some examples that can be used with the Dynamo Player.

- **Dynamo for Rebar v. 1.2.0**
  Free package (plugin) for Dynamo for authoring geometrically complex rebar models in Revit, developed and maintained by Thornton Tomasetti's CORE studio.
  More info on: https://github.com/tt-acm/DynamoForRebar.
  The nodes are explained with examples in this handout.





The Dynamo packages can be found in Dynamo, by performing the next steps:

1. In Dynamo, go to Packages > Search for a Package.
2. In the next dialog find the appropriate package and install using the ‘arrow’ button.

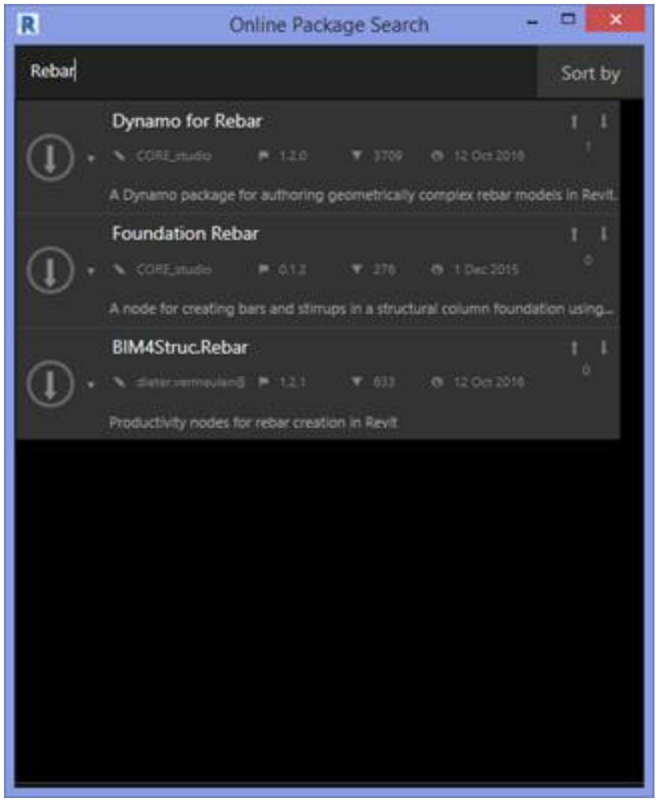


FIG. 2 - DYNAMO PACKAGE MANAGER

REBAR PRODUCTIVITY EXAMPLES

Rebar Productivity Examples

The next explained scripts are small examples that can be used to increase productivity in your rebar design.

Rebar Representation Setting

With *Dynamo Player* in Revit you can run several Dynamo scripts that don't need manual input (for now). In this example we use the *Set Rebar Solid in View* example. This will collect all the Structural Rebar objects from the Revit model and set their visibility setting to "Solid" for the current view.

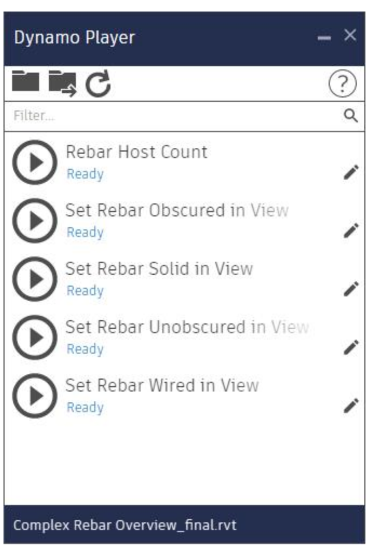
DATASETS

REVIT

Any Revit model containing Structural Rebar

DYNAMO

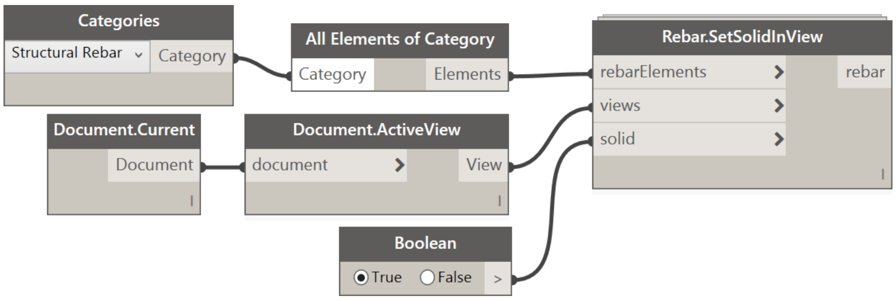
Set Rebar Solid in View.dyn
Set Rebar Wired in View.dyn
Set Rebar Obscured in View.dyn
Set Rebar Unobscured in View.dyn



Step 1 – Create the DYN script

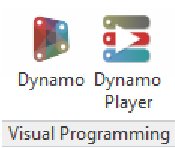
First you need to build up your Dynamo script. Like the one below for instance.

This one used the custom node *Rebar.SetSolidInView* from the **BIM4Struc.Rebar** package. It collects all the Structural Rebar elements, and set their visibility to solid in the current view.

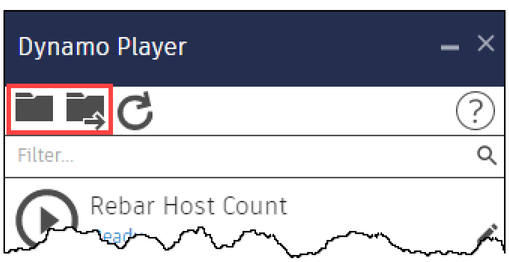


Step 2 – Initialize Dynamo Player

Dynamo Player is available in Revit 2017.1 on the *Manage tab*.

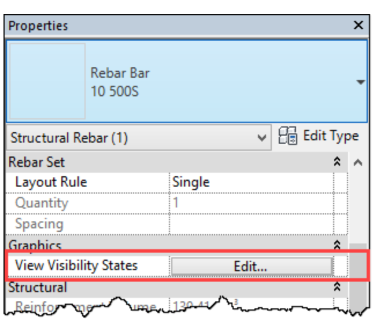


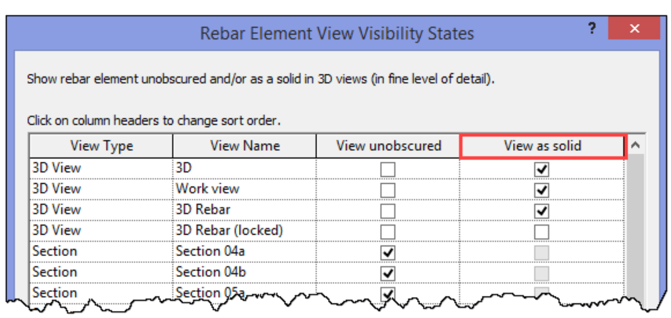
Now save/copy the file in the folder that is used to organize the Dynamo Player. You can find this folder via the Dynamo Player.



Step 3 – Run the script

Then click the  button next to your script. The script automated now the process below fur the current view.





Rebar Host Count

The “Host Count” parameter is available for a group of rebars in a schedule in Revit. When you disable the *Itemize every instance* parameter in the Revit schedule “Rebar Schedule”, under the Sorting/Grouping settings, then you get a total value for the host count.

With this script Dynamo allows to store the Host Count value into a project parameter of all the single rebar objects, so that you can track the value in extended workflows (e.g. construction coordination, field management …)

DATASETS

REVIT

Rebar Host Count.rvt

DYNAMO

00 Rebar Host Count.dyn

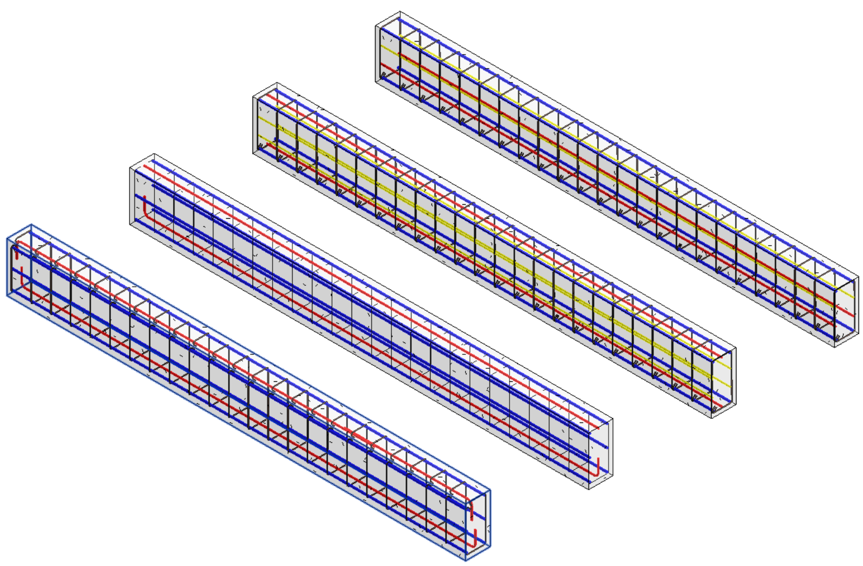


FIG. 3 - REINFORCED CONCRETE BEAM SAMPLES

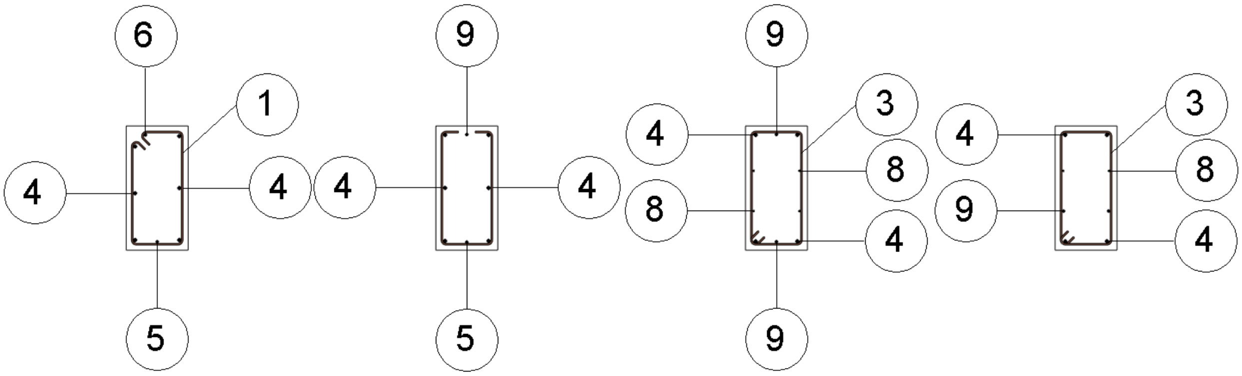


FIG. 4 - REBAR NUMBER TAGS

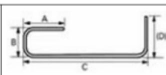
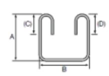
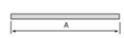
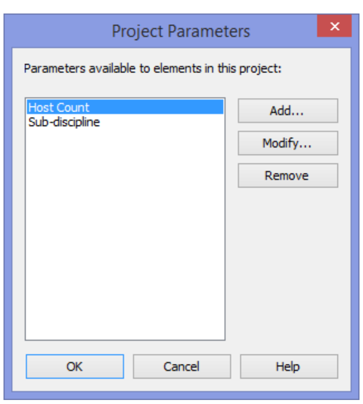
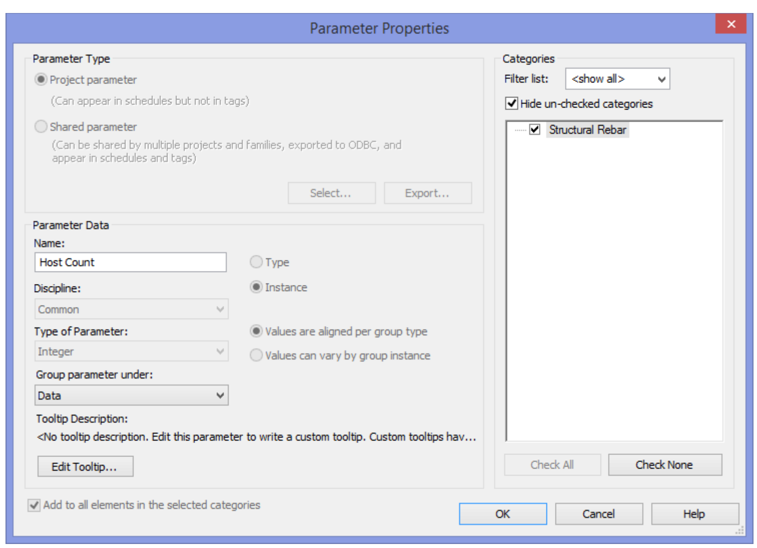
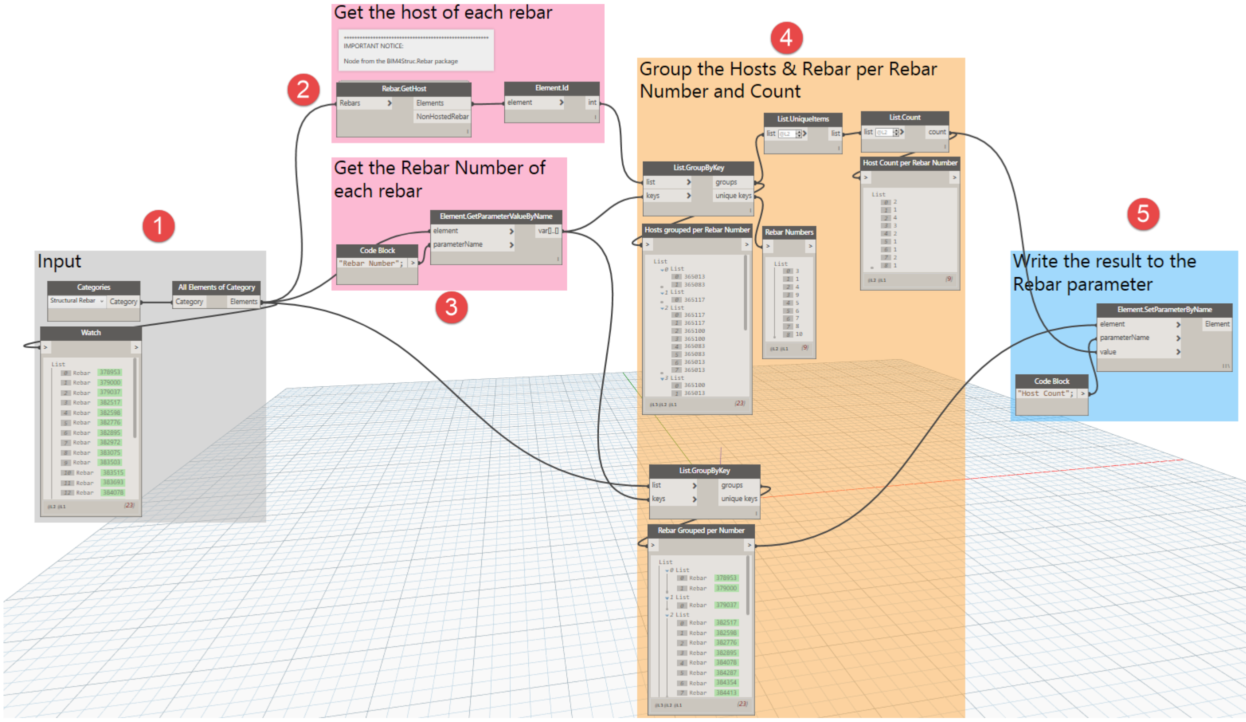
Rebar Schedule				
Rebar Number	Bar Diameter	Shape Image	Host Count	Bar Length
1	8 mm		1	1575 mm
3	8 mm		2	1675 mm
4	20 mm		4	5550 mm

FIG. 5 - REBAR SCHEDULE WITH TOTALS FOR HOST COUNT

Important Note:
Before you start with the script you need to create a project parameter called “Host Count” (integer type) in your Revit project.







(1) Get all the elements from the category "Structural Rebar"

(2) Each rebar element is hosted. The custom node *Rebar.GetHost* (from the **BIM4Struc.Rebar** package), returns the family instance that hosts the rebar. With the *Element.ID* node you get the Revit ID of each instance.

(3) The host count is based on the "Rebar Number" parameter value of each Structural Rebar in Revit. This node returns this value for each rebar. The results are the keys for the grouping of hosts and rebar in step (4).

(4) Group all the host elements and the rebar elements according to the keys defined by the rebar number. The hosts per group are then counted.

(5) In this step the calculated total hosts are now written in the project parameter "Host Count" of each Structural Rebar.

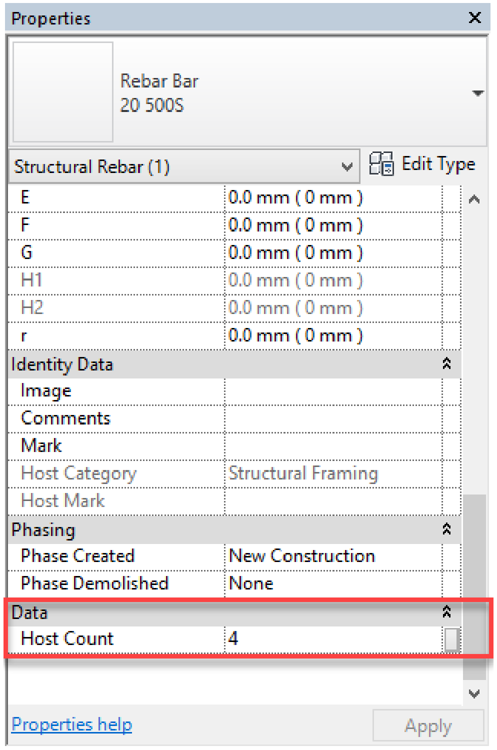


FIG. 6 - THE RESULT FOR EACH REBAR AFTER THE HOST COUNT SCRIPT

This script can also be used in the "Dynamo Player" panel of Revit.

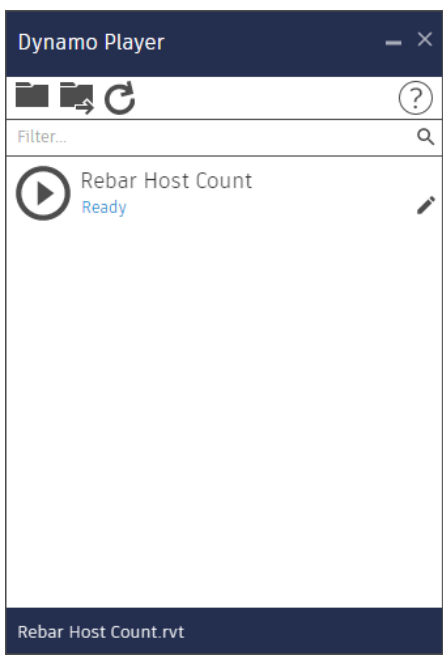


FIG. 7 - REBAR HOST COUNT WITH DYNAMO PLAYER

Rebar Centerline

This example shows how to get the sketch geometry of a selected rebar from Revit into Dynamo.

DATASETS

REVIT

Rebar Productivity.rvt

DYNAMO

01 Get Rebar Centerline.dyn

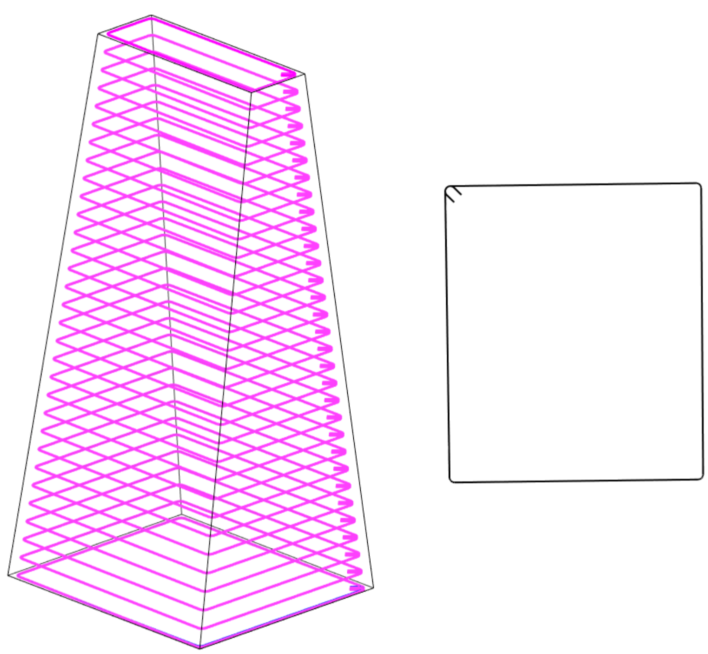
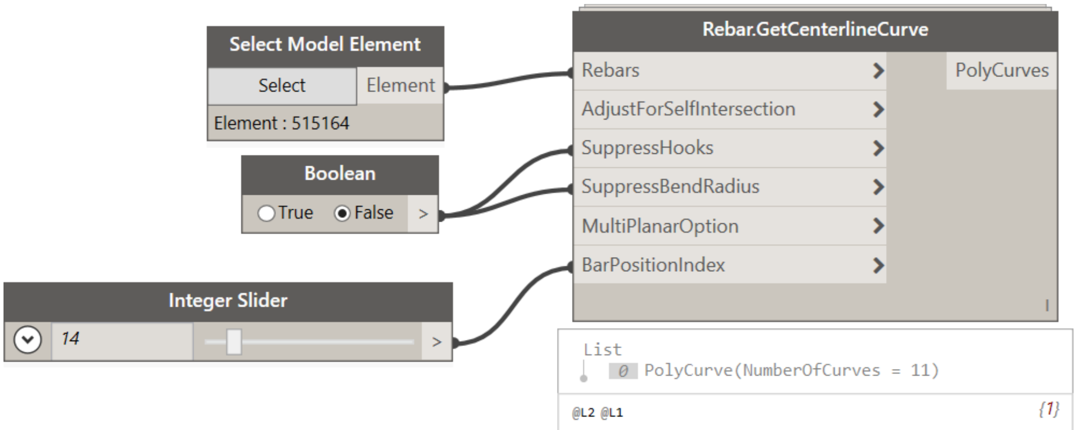


FIG. 8 - GET THE SKETCH GEOMETRY OF A SELECTED REBAR



Select a Structural Rebar in Revit (this can be a single bar, or a rebar set), the node *Rebar.GetCenterlineCurve* (from the **BIM4Struc.Rebar** package) reads the element and returns a Dynamo polycurve.

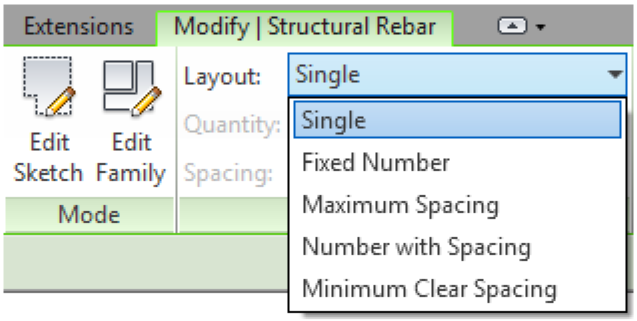
If the hooks and bend radius are not suppressed (set to “false”) then the node returns a full sketch geometry of the rebar (not a solid!).

The port “BarPositionIndex” is used when you select a Rebar Set. This option allows to indicate which element from the Rebar Set you want to return.

AUTODESK UNIVERSITY

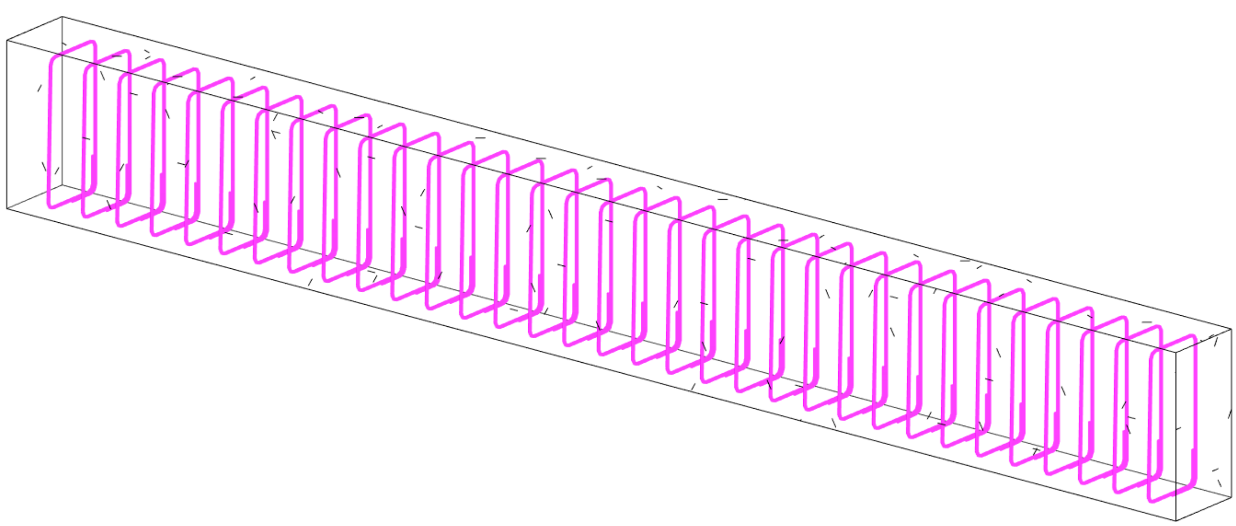
Rebar Layout Rules

With the Layout Rule a Structural Rebar can be organized in a Rebar Set, according to several layouts in Revit.



With this script you can automate this setting with Dynamo.

DATASETS	
REVIT	*Rebar Productivity.rvt*
DYNAMO	*02a Rebar Layout Rules.dyn*



The nodes in this script are available in the **BIM4Struc.Rebar** package.

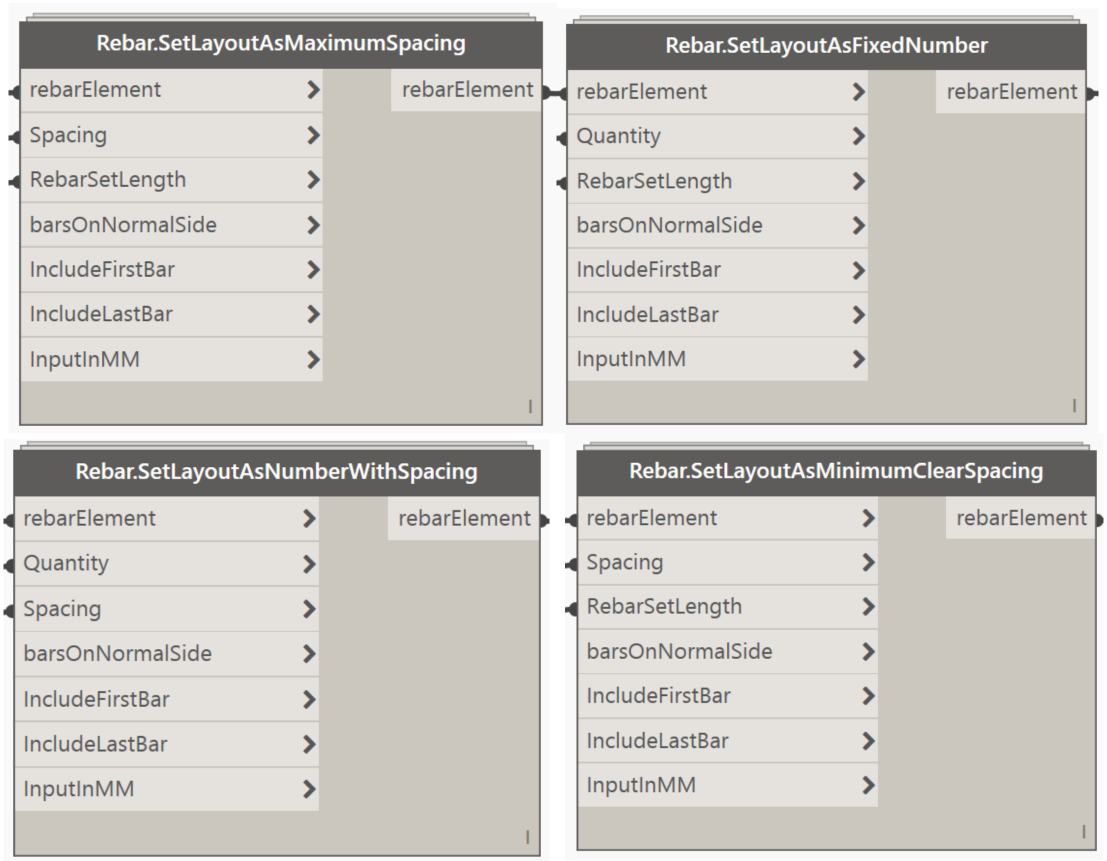
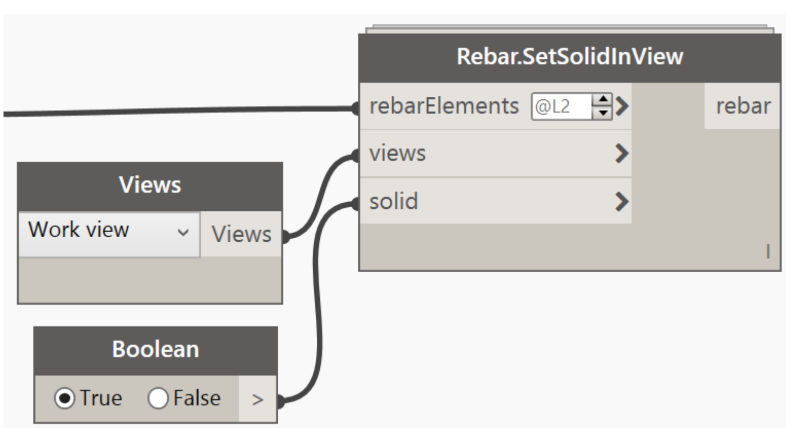


FIG. 9 - REBAR LAYOUT NODES FROM BIM4STRUC.REBAR PACKAGE

The port *barsOnNormalSide* identifies if the bars of the rebar set are on the same side of the rebar plane indicated by the normal vector.

The values are by default expressed in [mm]. In case you want to use Imperial units [feet], then you need to set the *InputInMM* port to “False”.



With the *Rebar.SetSolidInView* you ensure that the rebar is set as solid in the selected view.

Automatic Rebar Set Creation

The *SetLayoutRule* nodes from the **BIM4Struc.Rebar** package, allow you to create scripts that create rebar elements with different zones in a selected host.

This example shows how to apply distributed rebar in multiple zones with a minimum spacing for each zone for a beam or column.

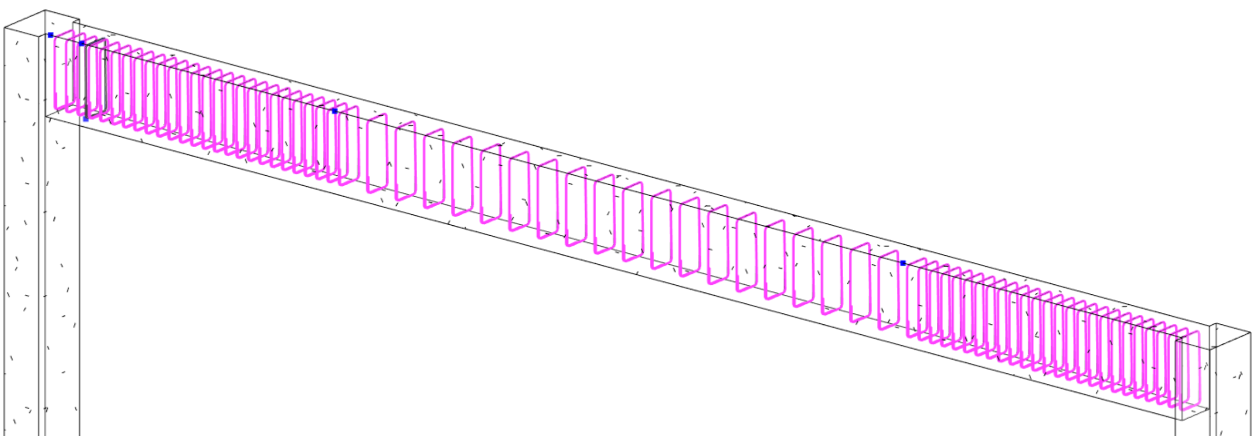
DATASETS

REVIT

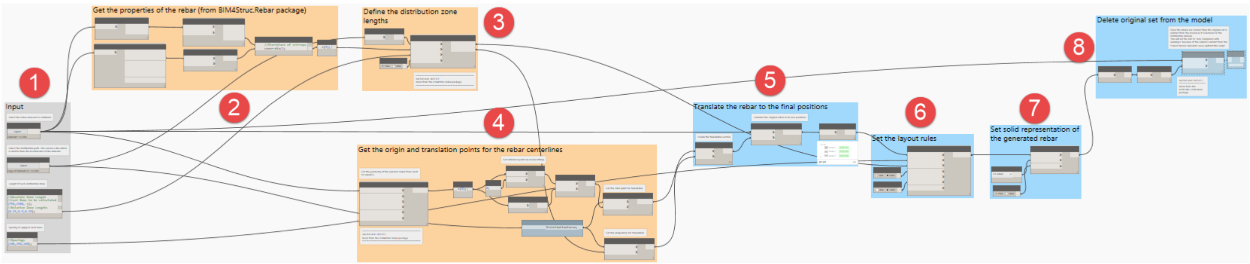
Rebar Productivity.rvt

DYNAMO

02b Automatic Rebar Set Creation.dyn



General Overview



The script consists of 8 main parts:

(1) Get the original rebar and the distribution guide curve

(2) Get the geometrical properties of the rebar

(3) Define the distribution zones

(4) Define the source and target translation points for the different zones

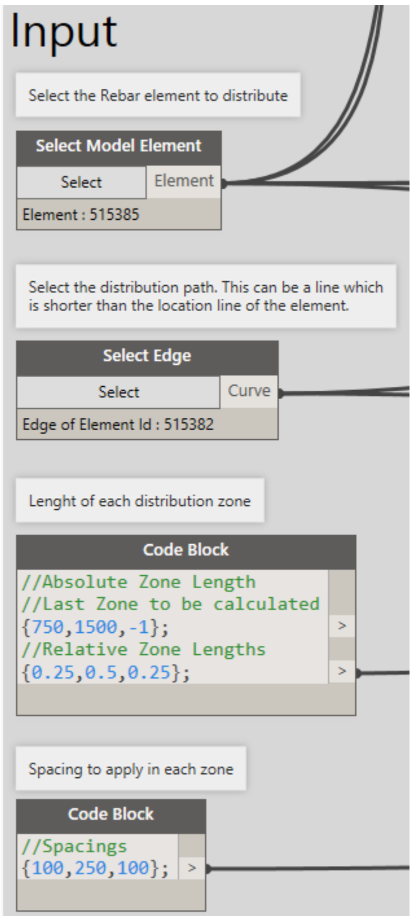
(5) Translate the rebar to the different zones

(6) Set the layout rule for each zone

(7) Optionally set the rebar in solid representation

(8) Optionally delete the original defined stirrup in the model

Step 1 – Input Geometry



First select the rebar geometry from a single stirrup in the Revit model, which needs to be distributed.

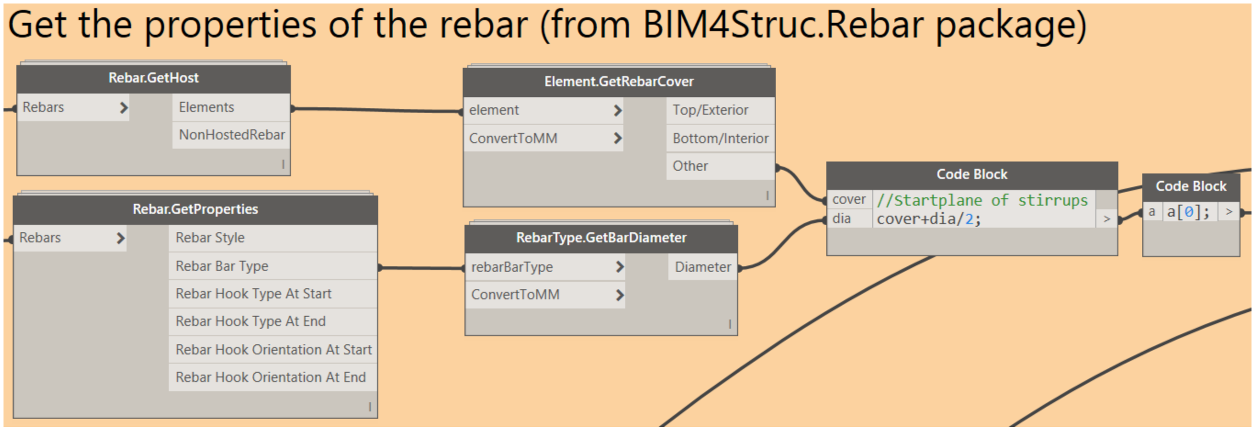
Select the edge along which the rebar needs to get distributed. Beware that Layout Rules can only be applied on straight zone, so in case the edge is a curve, the distribution won't work as desired.

In the first code block you can define the length of each zone. Make sure the sum of the zone lengths is not exceeding the total cut length of the beam or column. You can choose between relative or absolute zone values.

When you set the last zone length = -1, then the script will define the last length automatically based on the first zones and the total length of the beam or column.

In the second code block, you need to define the desired maximum spacing for each zone. Make sure the number of spacing values is equal to the number of zone you defined.

Step 2 – Get the Rebar Properties



The nodes used in this group are available with the **BIM4Struc.Rebar** package.

They detect the family instance that hosts the original rebar and also the properties of the rebar (such as the Rebar Bar Type). The *Element.GetRebarCover* will return the cover value for all faces of the selected Revit element (the host in this case). By default the value is returned in [mm]. If you set the “ConvertToMM” input port to “False” the node will return the value in feet.

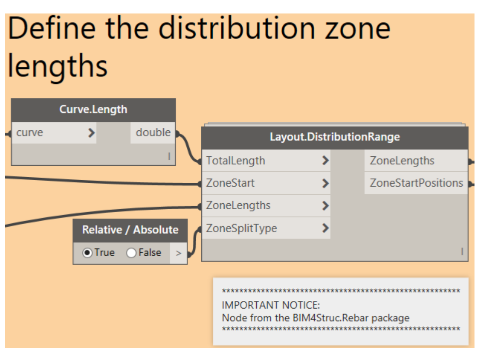
At the end of this group you get the start plane as a cover to center of the rebar.

Step 3 – Definition of the distribution zones

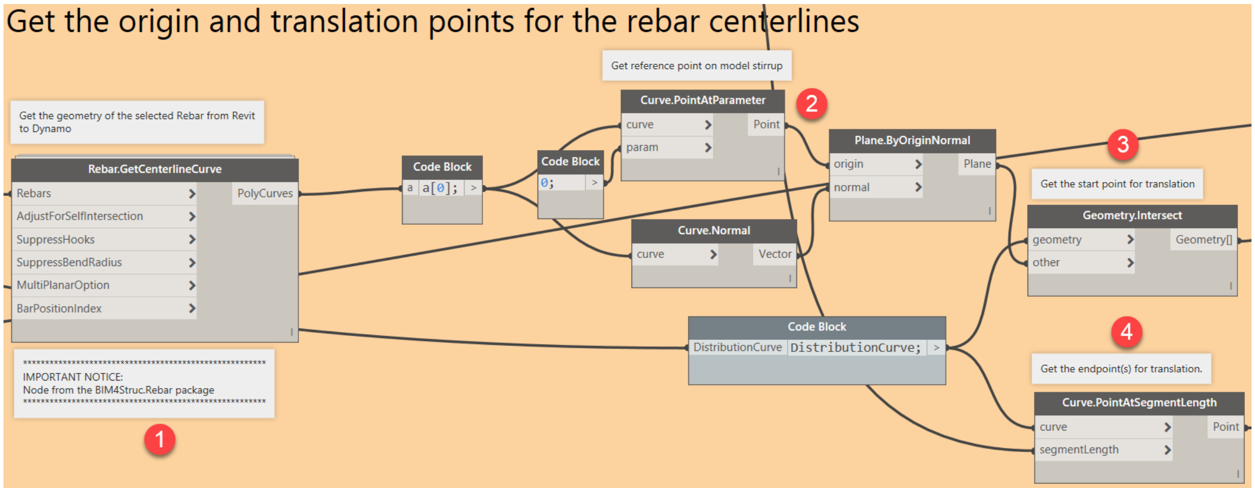
The distribution edge (selected in Step 1), also defines the start and end point of the full distribution.

With the custom node *Layout.DistributionRange* (from the **BIM4Struc.Rebar** package), you get the length and the start position of each zone. The calculation is based on the input values from Step 1.

- **TotalLength** = the total length of all zones together
- **ZoneStart** = the start position of the first zone, as absolute value (e.g. Cover + Half Diameter)
- **ZoneLengths** = The length of each zone (relative or absolute). In case the last zone length is indicated as “-1” this custom node will calculate the real value.
- **ZoneSplitType** = defines whether the zone dimensions are set as relative (true) or absolute (false) values.



Step 4 – Define translation points



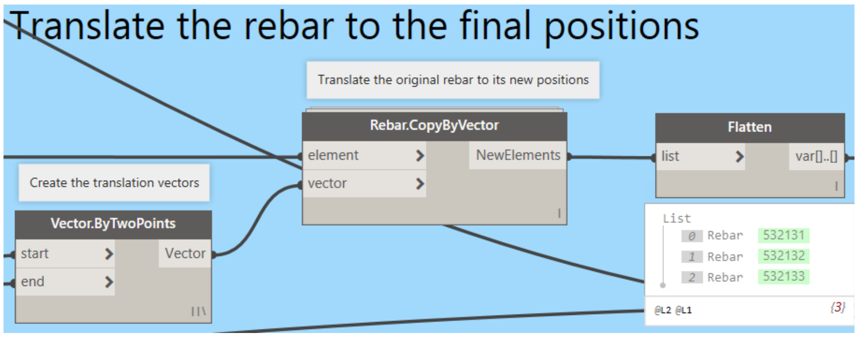
(1) Use the *Rebar.*GetCenterlineCurve node (**BIM4Struc.Rebar** package) to get the sketch lines from the selected stirrup.

(2) Get a reference point on the first segment of the resulting geometry and create a plane.

(3) Use the plane to intersect with the distribution curve, to get the reference point for the stirrup on this curve. This is the origin point for translation.

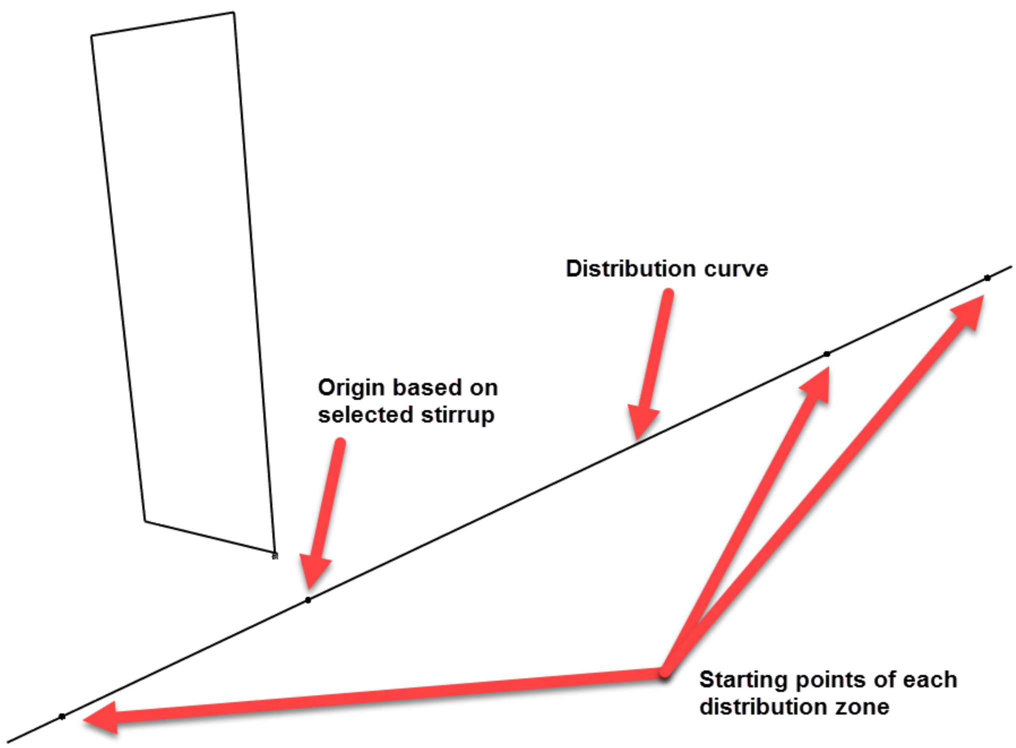
(4) Get the target points for translation by creating the zone starting points on the curve, with the output from the *Layout.DistributionRange* node.

Step 5 – Translate rebar to the specified zones



First a vector is created where "start" is the origin point of the selected rebar and the "end" is based on the defined translation points from [Step 4](#).

With the *Rebar.CopyByVector* (from **BIM4Struc.Rebar** package) the rebar is copied in the Revit project according to the defined translation vectors.



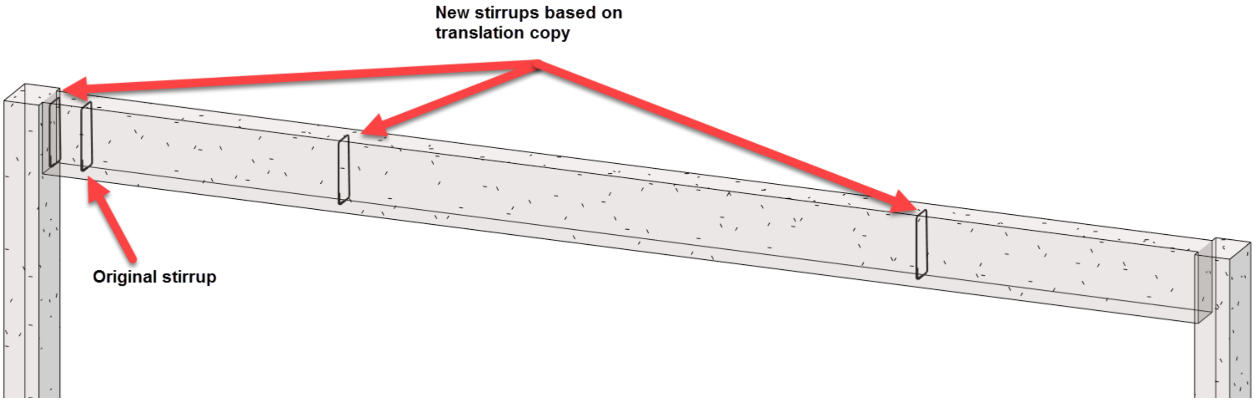
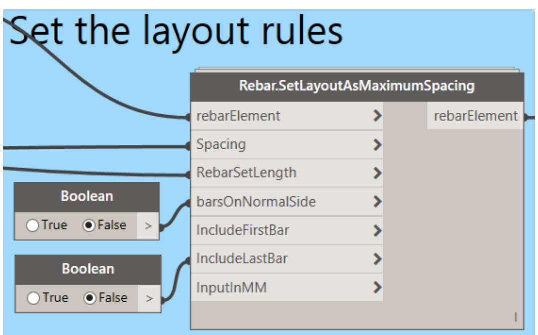


FIG. 10 - REBAR TRANSLATED / COPIED BY VECTORS IN REVIT

Step 6 – Set the Layout Rule for each zone

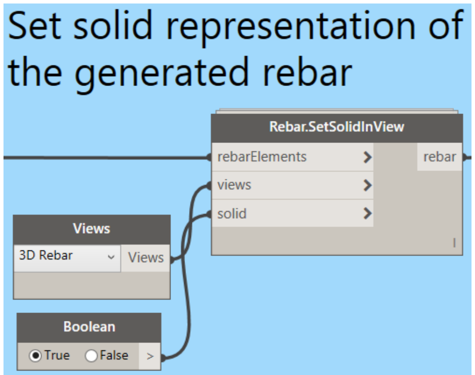
Once all the rebar are positioned at the start of each zone, the “Layout Rule” can be applied. The spacing is defined in Step 1. The “RebarSetLength” is defined as the length of each zone in Step 3.

In this case the “IncludeLastBar” option is set to “False” to avoid overlapping bars from two adjacent zones.



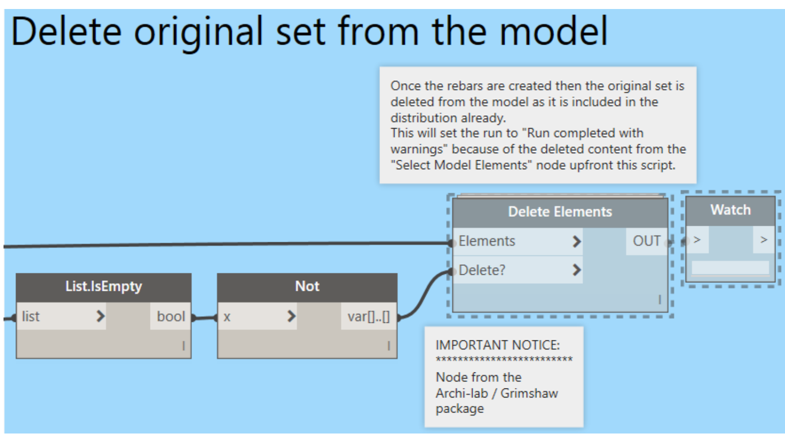
Step 7 – Set the solid representation of the rebar

In this optional step you can set the representation of the generated rebar to solid in the selected view.



Step 8 – Delete original stirrup

Optionally you can choose to delete the original base stirrup (as this one is not needed anymore). This is done with the node *Delete Elements* (from the **Archi-lab / Grimshaw** package), which is triggered by a boolean, that results in "True" when the output of the *Rebar.SetSolidInView* from previous step, is <u>not empty</u>, meaning that all nodes have been successfully executed.



Blended Shapes

With the “Varying Rebar Set” function, Revit allows to generate rebar with varying lengths in shapes such as blended forms.

In this example the primary rebar and stirrups in a blended shape are automated with a Dynamo script, driven by this “Varying Rebar Set” feature.

An additional challenge is to have different oriented hooks for each primary rebar, point the anchorage hooks toward the center of the base plane of the form.

DATASETS

REVIT

Rebar Productivity.rvt

DYNAMO

03 Blended Shapes.dyn

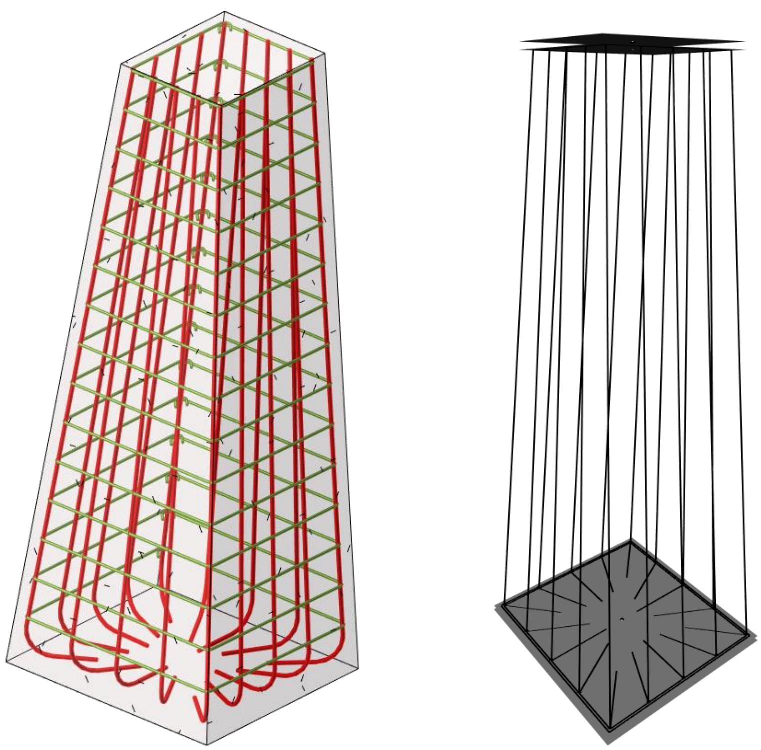
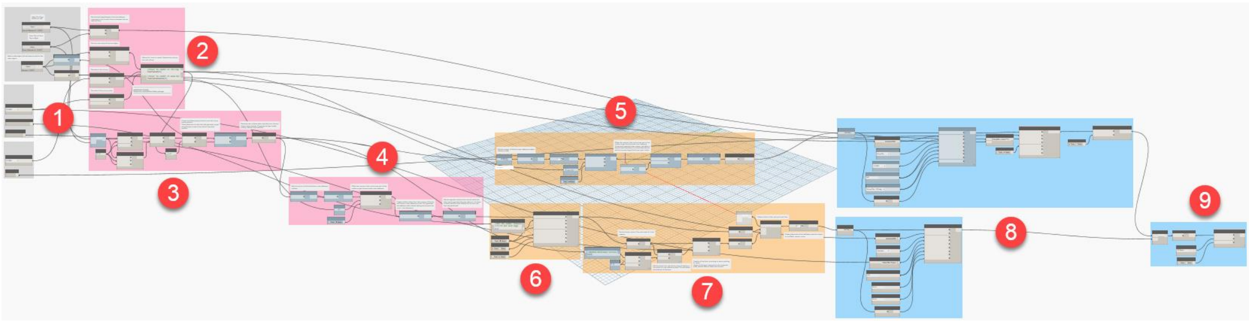


FIG. 11 - REBAR DISTRIBUTION IN BLENDED SHAPE

General Overview



(1) Input & settings

(2) Define rebar cover

(3) Define start plane stirrup

(4) Define cover to center plane for primary bars

(5) Create stirrup sketch

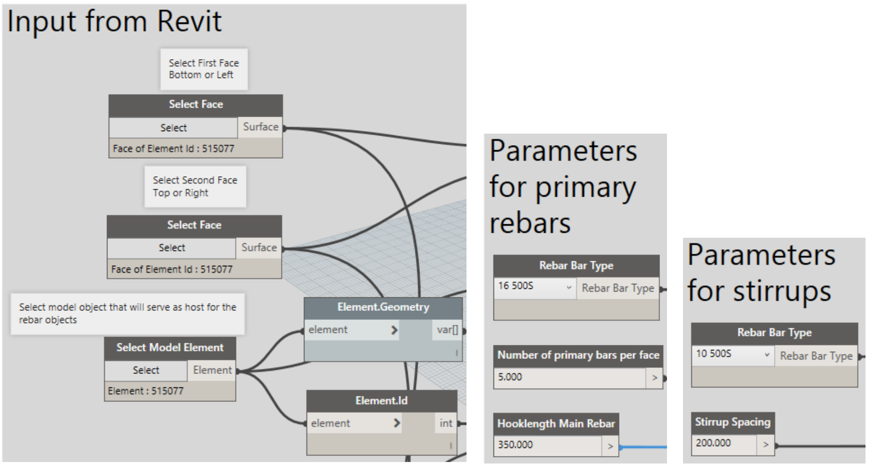
(6) Create primary rebar centerlines

(7) Create primary rebar hook lines

(8) Create rebar in Revit

(9) Set solid view of Rebar in Revit

Step 1 – Input base geometry and settings

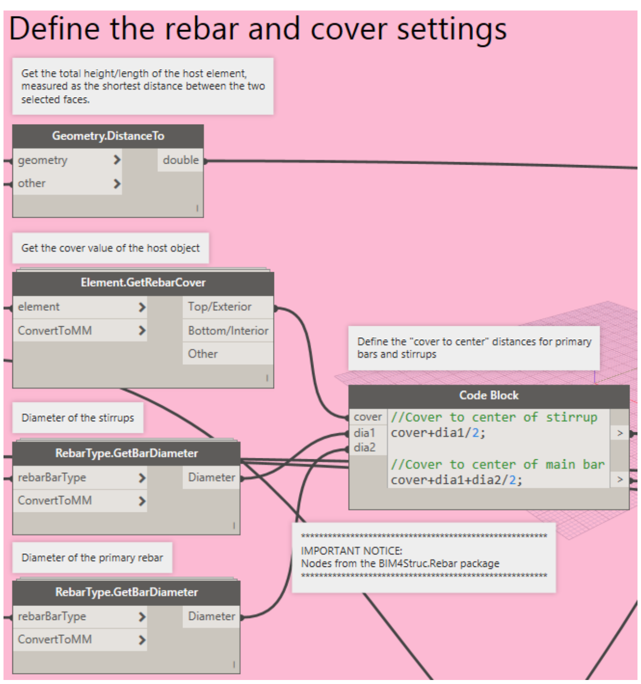


The script needs three inputs from Revit:

1. The bottom (column) or left (beam) face of the blended form
2. The top (column) or right (beam) face
3. The family instance that hosts the rebar

Other than that you also need to set the diameter and layout for the primary bars and stirrups.

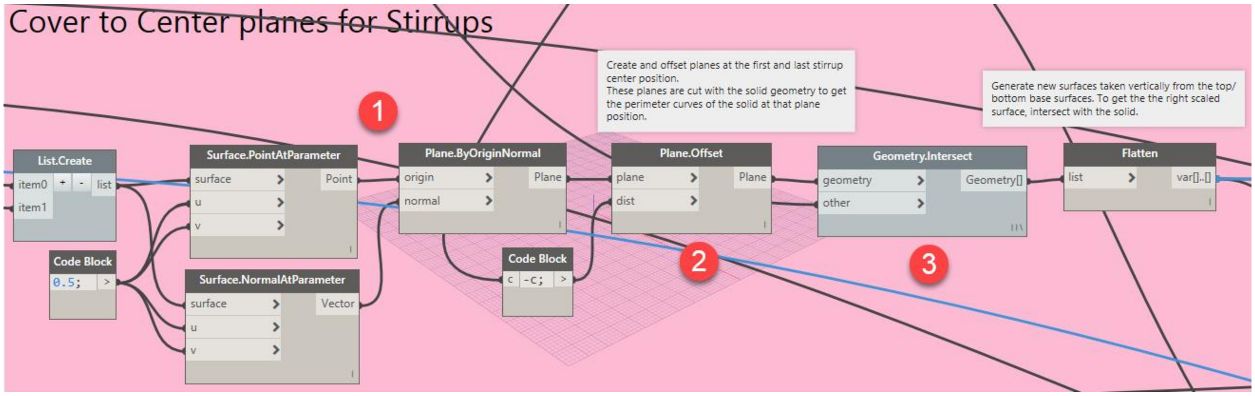
Step 2 – Define Rebar Cover



With these nodes from the **BIM4Struc.Rebar** package, you can get the cover value of the host elements, as well as the diameter of the selected rebar bar types.

These values result in the calculation for the cover to center to the stirrup and the main bar. The main bar needs to be enclosed by the stirrup, thus that cover to center value is greater.

Step 3 – Define Cover to Center Plane Stirrup



(1) Create planes, based on the normal of the bottom and top face.

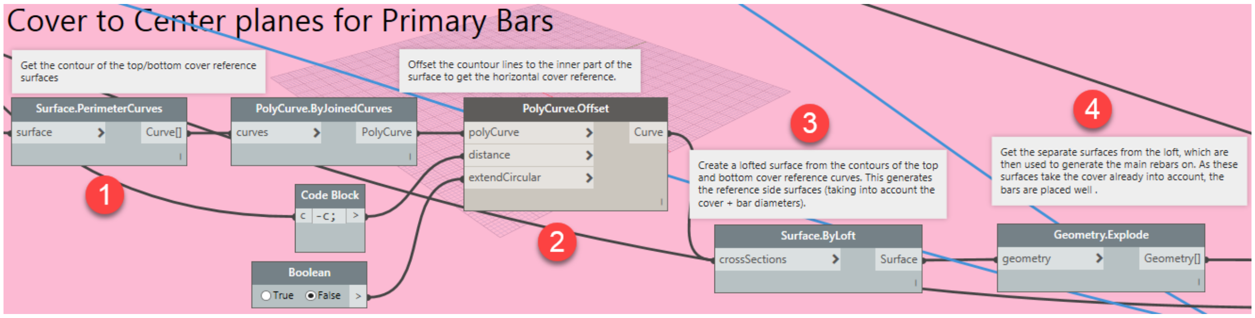
(2) The planes get offset with the cover to center value for the stirrup.

(3) The resulting planes are then intersected with the solid geometry. The results are two surfaces defining the cover to center planes for the stirrups at the bottom and top.

For this operation we use a *Plane* to be sure that the full solid geometry gets intersected (as planes are infinite). When using the original surface, you risk not to cut the full solid geometry in case the blend is smaller at the bottom then at the top.

FIG. 12 - COVER REFERENCE PLANES STIRRUPS

Step 4 – Define cover to center plane for primary bars



(1) The resulting surfaces from previous step are decomposed to their perimeter curves and joined together in one polycurve per surface.

(2) The polycurves now get offset again with the cover to center for the primary bars.

(3) When these resulting polycurves are lofted you get polysurface representing the center planes for the primary bars.

(4) The polysurface is exploded in separate surfaces for the creation of primary (longitudinal) bars along each face.

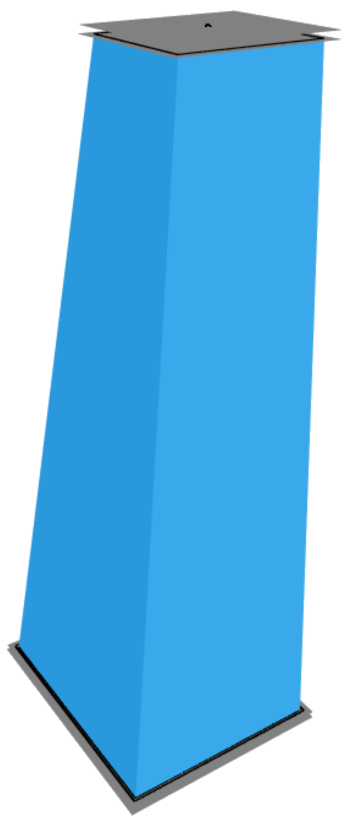
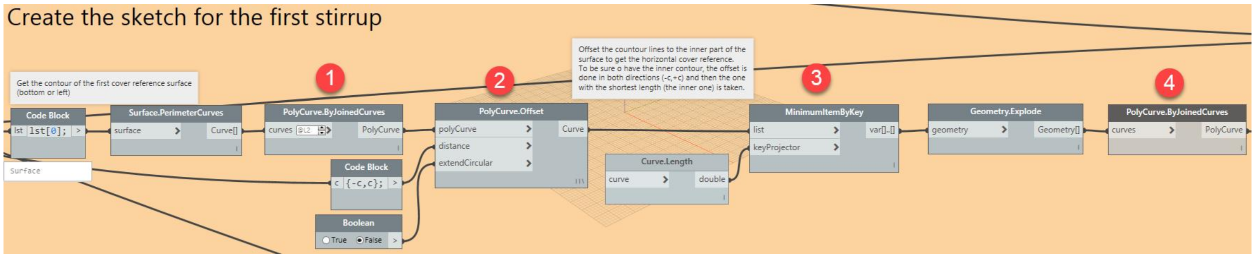


FIG. 13 - COVER REFERENCE PLANES PRIMARY BARS

Step 5 – Create Stirrup Sketch

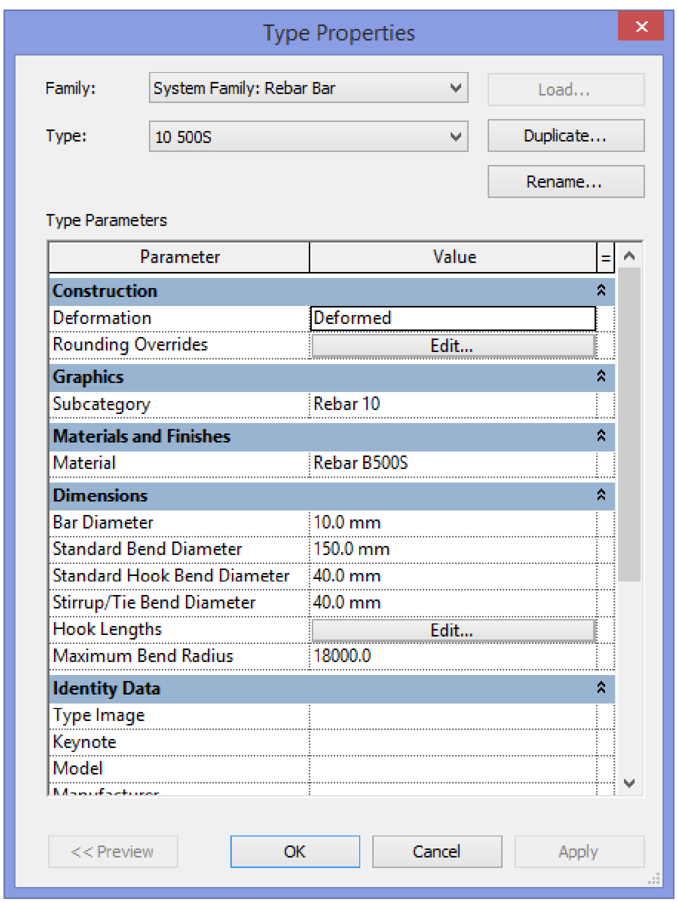


(1) The perimeter curves of the bottom cover reference is taken, and joined into a polycurve

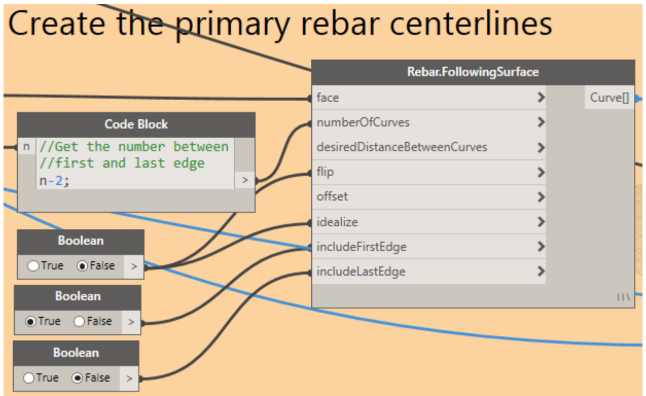
(2) This polycurve is then offset with the cover to center distance for stirrups. The offset is done in + and – direction, as it is not always sure which direction you need to offset to stay inside the geometry. It depends on the surface normal of the blend shape.

(3) The right polycurve can be found by taking the one with the minimal length.

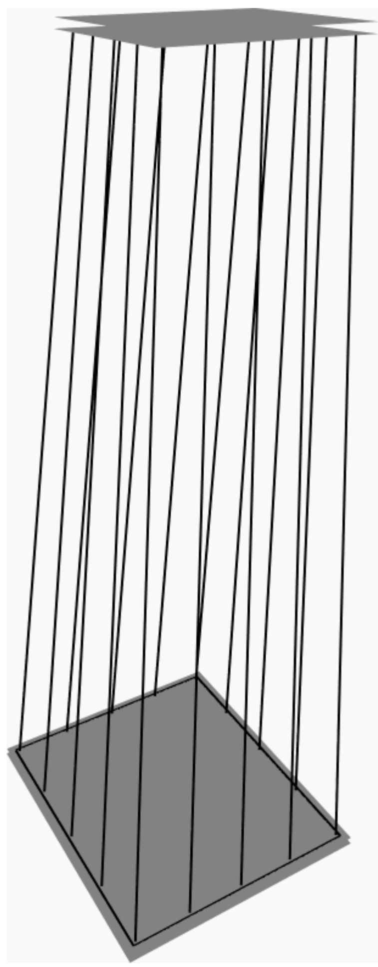
(4) The new offset is joined back into a polycurve to be used for rebar creation in Revit then. You don't need to take into account the bend radius, as this is defined through the properties of the rebar bar type in Revit.



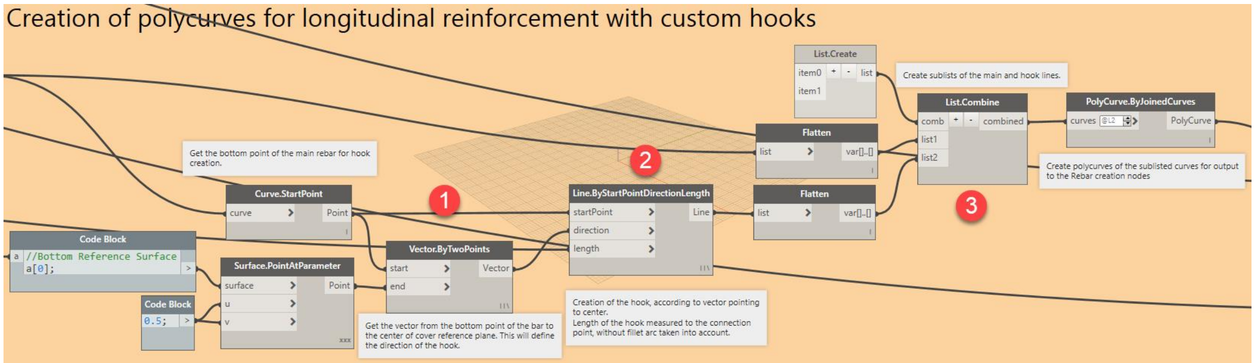
Step 6 - Create Primary rebar centerlines



The *Rebar.FollowingSurface* node (from the **Dynamo for Rebar** package) creates a set of curves following the geometry of a selected surface (most polysurfaces will also work). It divides the surface in one dimension - either U or V - regularly. You can define the number of divisions (or optionally, a distance to divide the surface by), and the direction of the curves.



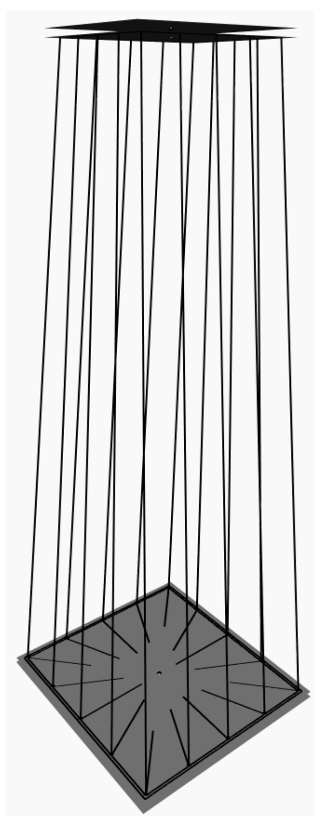
Step 7 – Create primary rebar hook lines



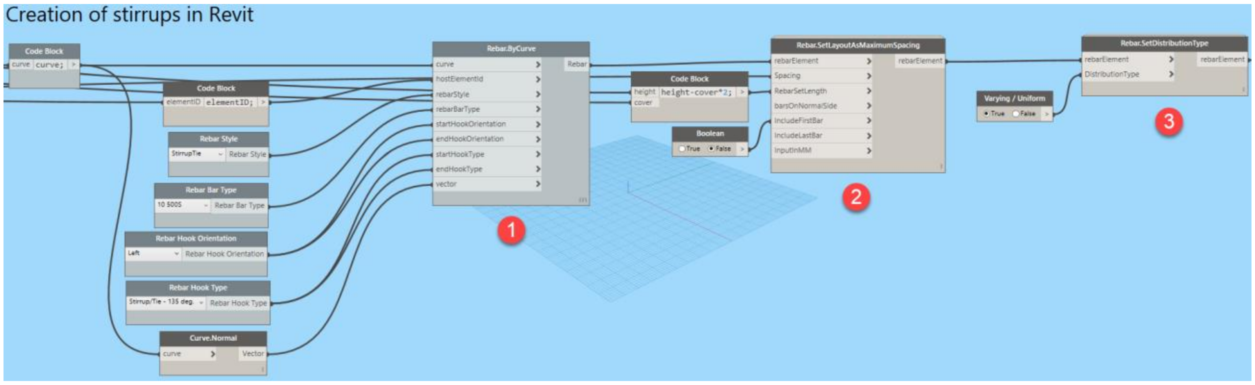
(1) Before you set out the geometry of the sketched anchorage hooks, the direction vector of the hooks need to be defined. In this case the vector is defined from the start point of each primary bar towards the center point of the bottom cover reference plane. (So the point.Z is equal for both points).

(2) Then the line is created, by the start point, vector and a defined length (see step 1).

(3) Finally the longitudinal lines and the hook lines are combined into polycurves for the rebar creation.



Step 8 – Create Rebar in Revit



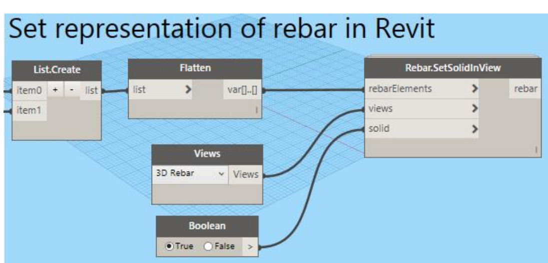
(1) The resulting curves of the stirrups are used to generate the rebar in Revit with the *Rebar.ByCurve* node (from the **Dynamo for Rebar** package). The "vector" is the direction in which the rebar would get distributed, and represents the normal vector of the plane in which the rebar sketch is created. This can be easily found with the *Curve.Normal* node. You can find out more in the chapter Face Reinforcement.

(2) With the *Rebar.SetLayoutToMaximumSpacing* a layout rule is applied to the generated stirrup. This is explained more in detail here.

(3) The *Rebar.SetDistributionType* from the **BIM4Struc.Rebar** package allows it to set the rebar set to a *Varying* Length distribution.

The created Rebar are treated as 'Rebar by Sketch' once you generate them from within Dynamo. You will see in the Rebar Shapes of the family browser, popping up new definitions in case it doesn't meet one of the standard shape. If you edit the family type you could add a Shape Image. The Rebar Shape family is created automatically from the sketch. Depending on the order in which you generate the sketch lines in Dynamo, it will assign the A, B, C, and D... values. These last ones can then be used in the Bending schedules.

Step 9 – Set solid representation of rebar in Revit



Again the representation of the rebar can be set as solid in the selected view. This time the function works on multiple rebar.

Area Reinforcement

Creates Area Reinforcement in a wall or floor according to the wall/floor sketch geometry. The exterior and interior reinforcement are the same by default.

DATASETS

REVIT

Rebar Productivity.rvt

DYNAMO

04a Area Reinforcement – Batch Create.dyn
04b Area Reinforcement - Create New.dyn

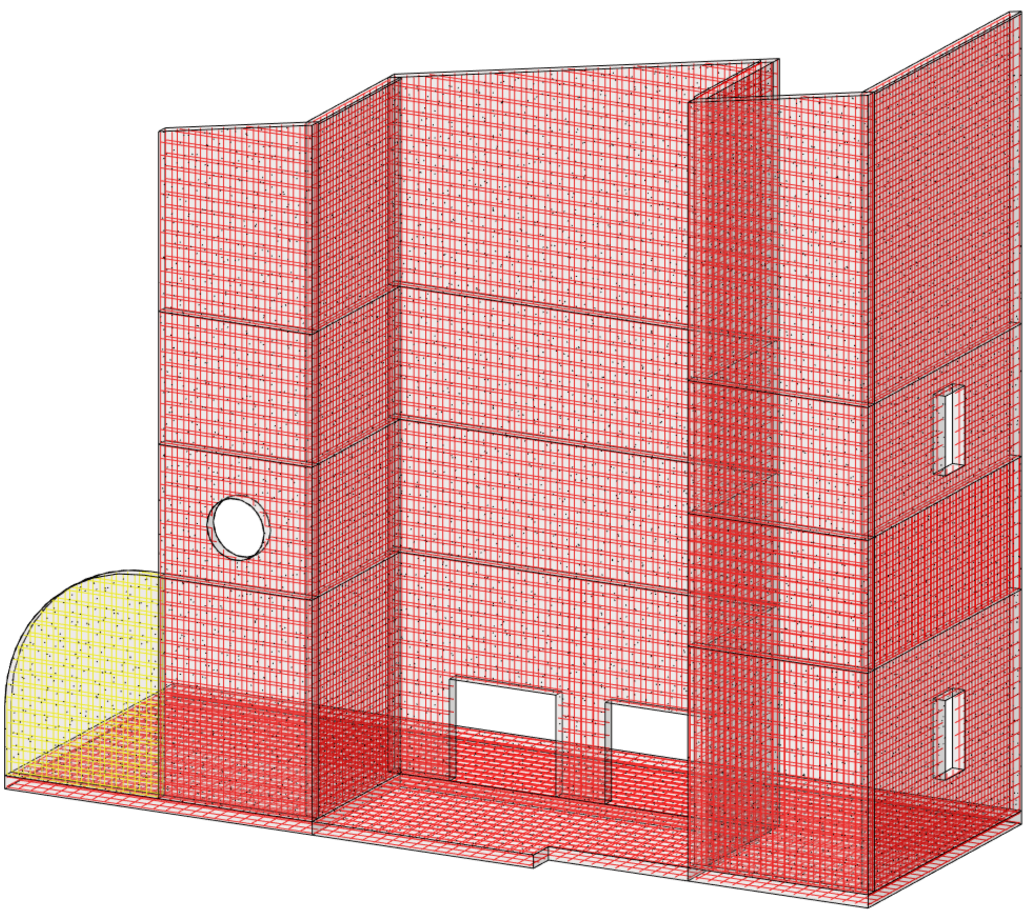
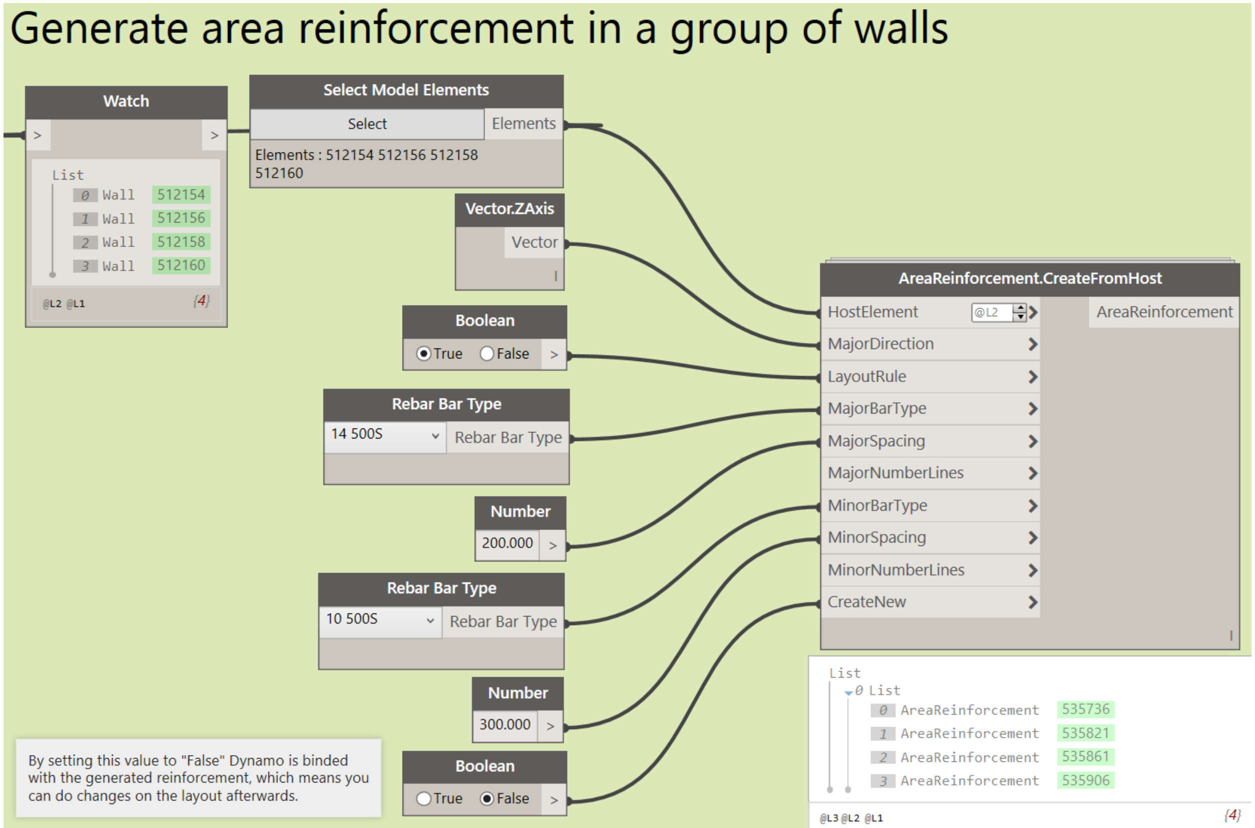


FIG. 14 - AUTOMATED AREA REINFORCEMENT IN WALLS & FLOORS



This short script uses the custom node *AreaReinforcement.CreateFromHost* (from the **BIM4Struc.Rebar** package), which creates Area Reinforcement in a wall or floor according to the wall/floor sketch geometry. The exterior and interior reinforcement are the same by default.

In the port “LayoutRule” you can choose between the layouts “Maximum Spacing” (when “true”) and “Fixed Number” (when “false”).

Depending on that choice, you need to set the values for MajorSpacing / MinorSpacing or “MajorNumberOfLines / MinorNumberOfLine”.

The bottom port “CreateNew” makes it possible to

- create a new element for each run of Dynamo (when set to “True”)
- create and update the generated element (when set to “False”), which is the default behavior with element binding.

The example *04a Area Reinforcement – Batch Create.dyn* shows how to create area reinforcement in a selected group of walls or floors, and updates the generated reinforcement when made a change to the properties.

The example 04b Area Reinforcement - Create New.dyn shows how to add area reinforcement to the selected wall or floor, for every Dynamo run. You’ll have to change the selection thus, before you run the script, to avoid double generation of reinforcement.

COMPLEX REBAR MODELLING EXAMPLES

Complex Rebar Modelling Examples

In the next few examples complex reinforcement modelling techniques are explained. In most of the examples the technique can be split up in two parts: Evaluating surfaces as a set of curves and creating rebar elements from such curvatures.

Face Reinforcement

This example shows how to create complex (skin) reinforcement along (double) curved surfaces.

DATASETS

REVIT

Complex Rebar Overview.rvt

DYNAMO

01 Face Reinforcement.dyn

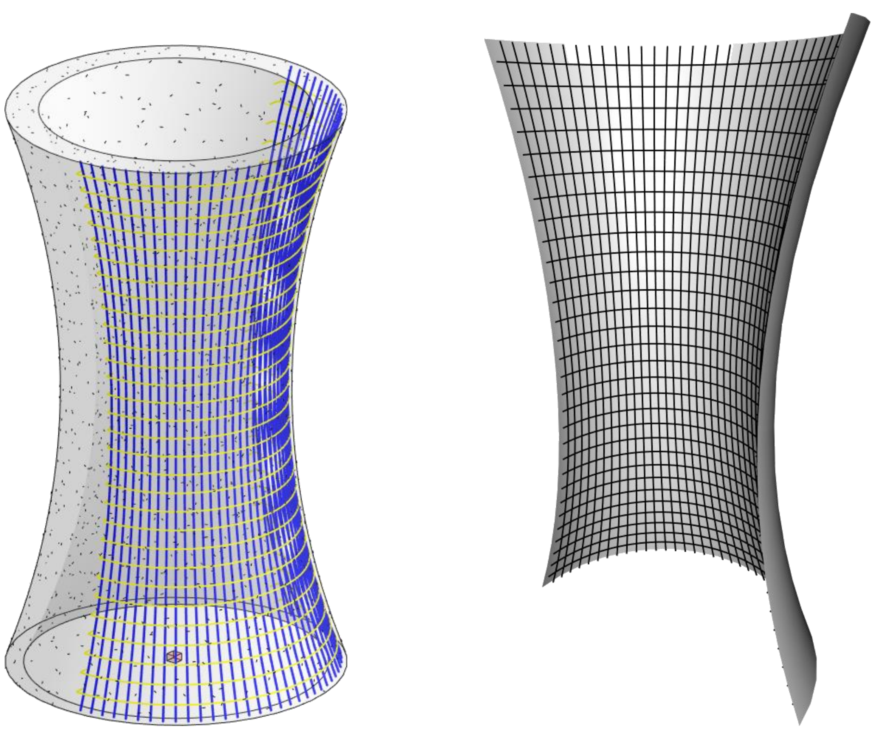
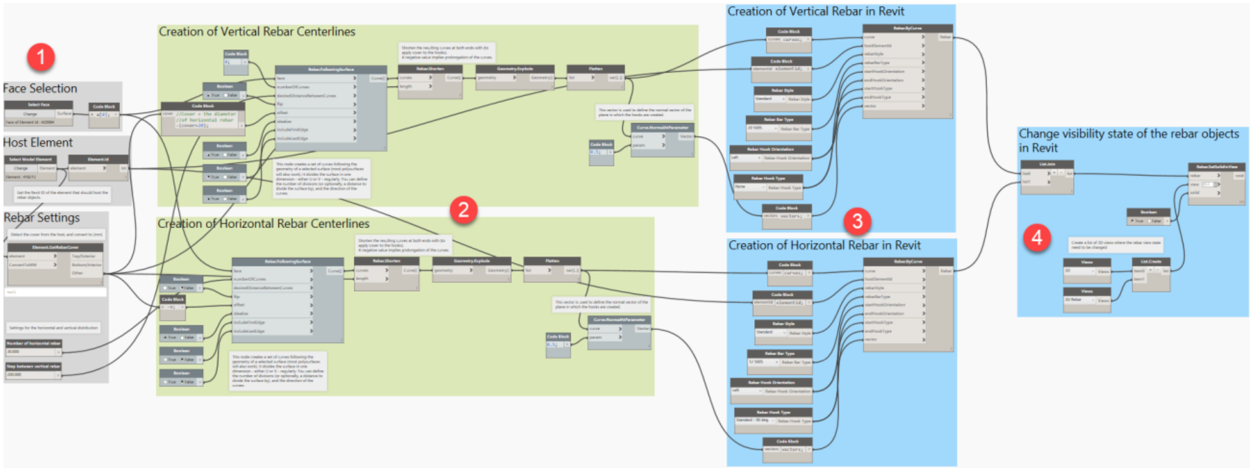
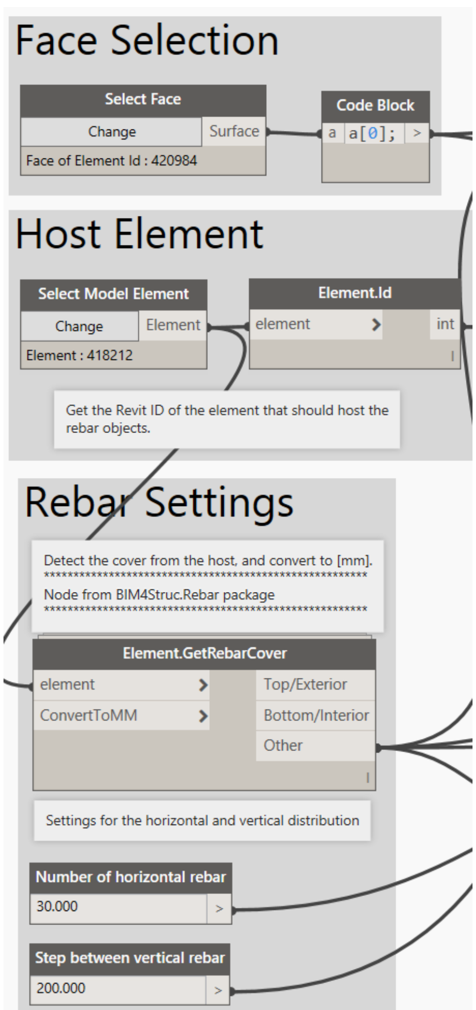


FIG. 15 - FACE REINFORCEMENT IN CURVED OBJECTS



Step 1 – Input Geometry



First, select the face along which you want to create the (skin) reinforcement.

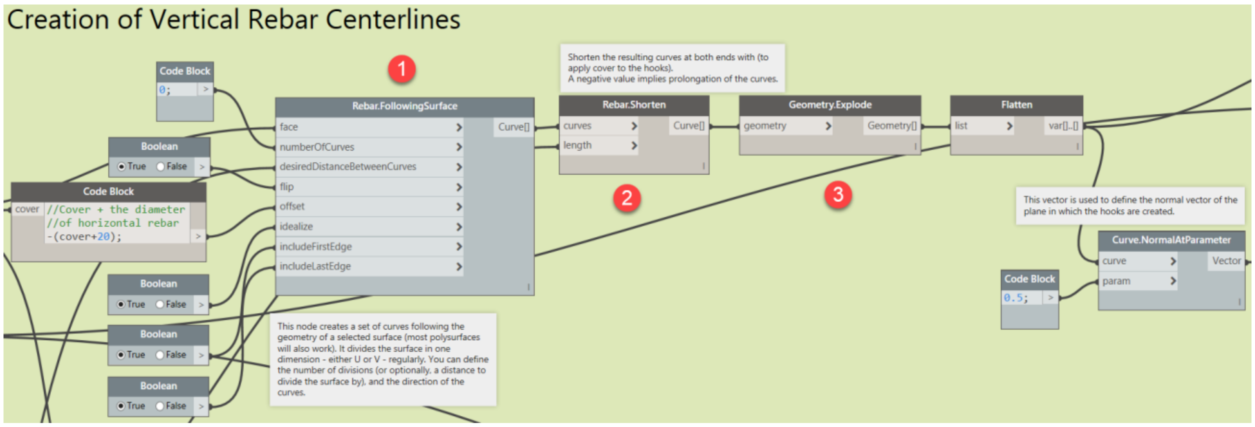
Then, select the Revit family instance that needs to host the rebar. In this case, the family instance is an element from the Mass category, which can't host rebar. To work around this, you could select a "dummy" object that is capable of hosting rebar (e.g. a Generic Model).

Be careful with this workaround, as it may cause performance issues if you use it too much in your project, due to warnings related to "Rebar is placed outside its host").

The cover distance is read from the rebar host object with the node *Element.GetRebarCover* (from the **BIM4Struc.Rebar** package). This cover acts already as the cover-to-center distance. In other situations you might need to add the bar diameter / 2 to this value.

At the bottom, you can select the layout for the horizontal and vertical rebar that need to be created further on in the script.

Step 2 – Evaluate surfaces and create Rebar Centerlines



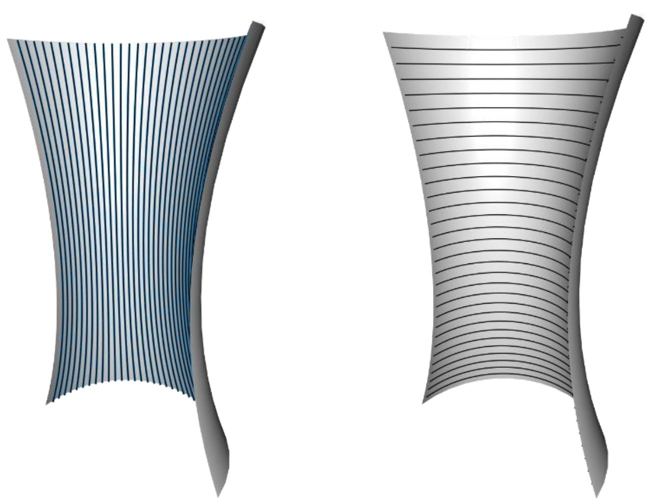
(1) The *Rebar.FollowingSurface* node (from the **Dynamo for Rebar** package) evaluates the surface curvature directly across a selected surface. It requires a number of curves to create and allows you to flip the evaluation direction from vertical to horizontal. The offset parameter offsets the curvature along the surfaces normal. Idealize toggles between a simplified and a precise surface evaluation.

With the "flip" option you can choose in which iso-direction of the surface you want these curves to be generated.

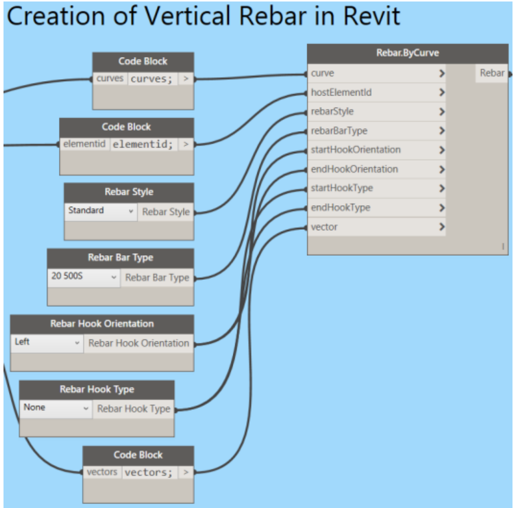
Additionally the "includeFirstEdge" and "includeLastEdge" allow to eliminate the surface edges in case they are taken into account for the curve generation.

(2) The *Rebar.Shorten* node (**Dynamo for Rebar** package) allows to shorten the resulting curves at both ends with a specific distance (e.g. to apply cover to the hooks). A negative value implies prolongation of the curves.

(3) The resulting curves need to be exploded in some cases, in order to be accepted as input for the Rebar creation in Revit (see step 4).



Step 3 – Creation of Rebar in Revit



The generated curves from the surface evaluation can now be used to create rebar objects in Revit from. There are two nodes from the **Dynamo for Rebar** package that can be used for that:

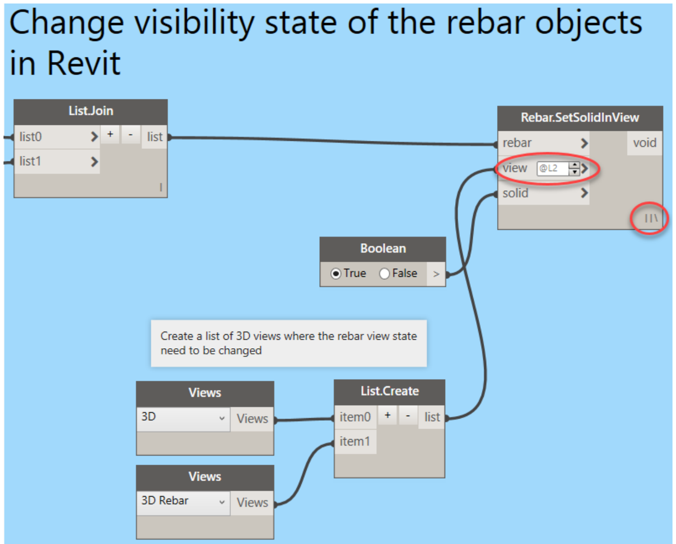
- *Rebar.ByCurve* : Creates one single rebar element in Revit from a curve (or list of curves) and a series of rebar properties.
- *RebarContainer.ByCurve* : Creates a rebar container element from a list of curves and a series of rebar properties. The use of containers is highly encouraged as Revit can get bogged down by thousands of rebar family instances in your model. Containers are like groups of rebars in a single family instance.

The properties that need to be assigned to both nodes:

- *Curve* : An array of curves that define the shape of the rebar curves. They must belong to the plane defined by the normal and origin. Bends and hooks should not be included in the array of curves.
- *hostElementId* : the Revit ID of the element hosting the Rebar. The element must support rebar hosting.
- *rebarStyle* : choose between “Standard” or “StirrupTie”. This will influence the bending radius and hook style.
- *rebarBarType* : the rebar family type representing the bar diameter
- *startHookOrientation* or *endHookOrientation* : the orientation of potential hooks (“Left” or “Right”)
- *startHookType* or *endHookType* : the Rebar Hook to apply at the ends of the bar.
- *vector* : specifies the orientation of the rebar hooks. This vector is the normal to the plane that the rebar curves lie on. This can be the surface or the curves normal.

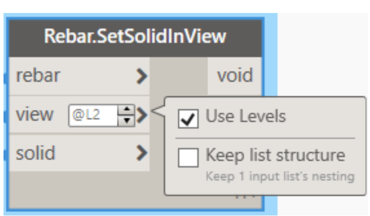
The created Rebar are treated as 'Rebar by Sketch' once you generate them from within Dynamo. They might pop up in the Revit Shape Browser as new definitions, in case they don’t meet with existing loaded shapes. Depending on the order in which you generate the sketch lines in Dynamo, it will assign the A, B, C, D... values. These last ones can then be used in the Bending schedules.

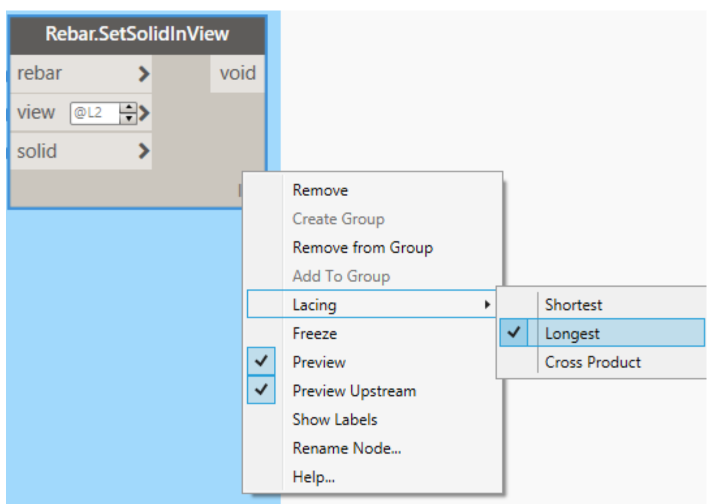
Step 4 – Set visibility state of Rebar in Revit



Use the *Rebar.SetSolidInView* node (from the **DynamoForRebar** package) to set the solid representation of the generated rebar, for several views at once.

Make sure you use the levels at the port “view”. And if you run this node on multiple rebar, then also make sure the lacing is set to “Longest”, as indicated on the image below.





Morphed Reinforcement

With this example you will learn how to create a number of rebar which are “morphed” between two boundary curves.

DATASETS

REVIT

Complex Rebar Overview.rvt

DYNAMO

02 Morphed Reinforcement.dyn

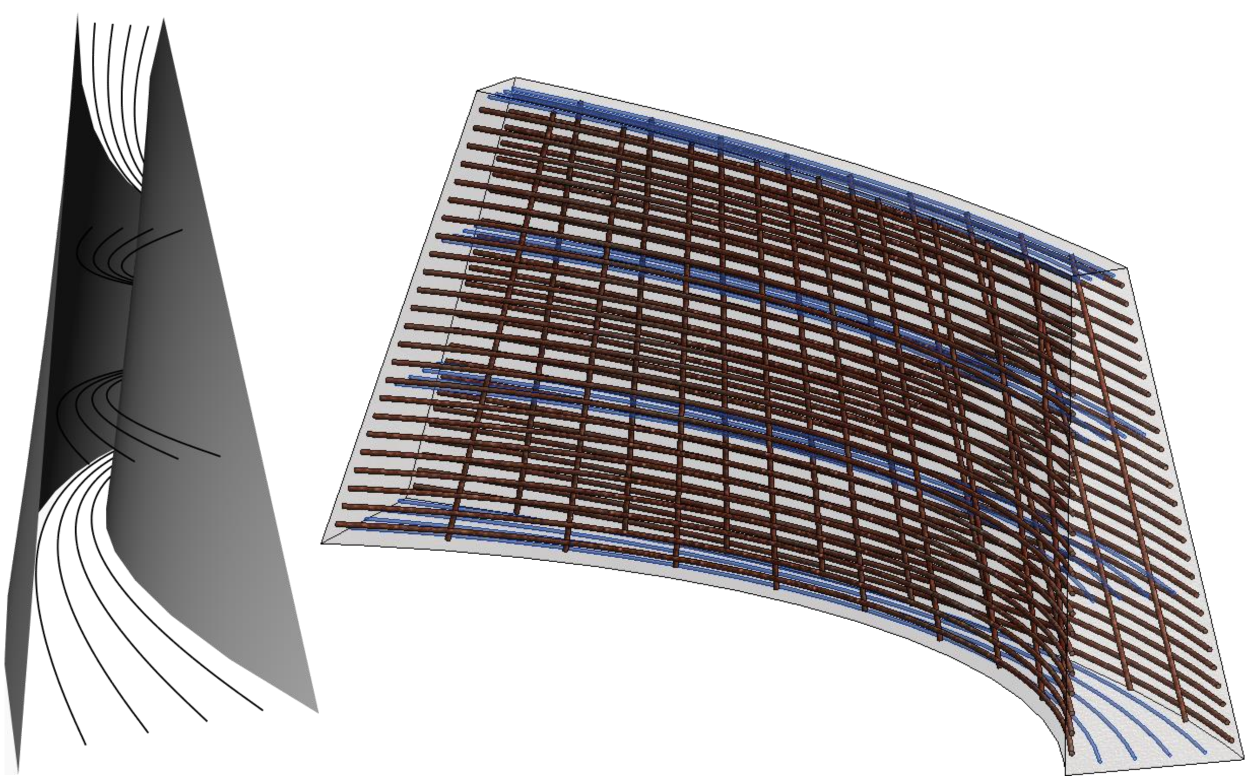
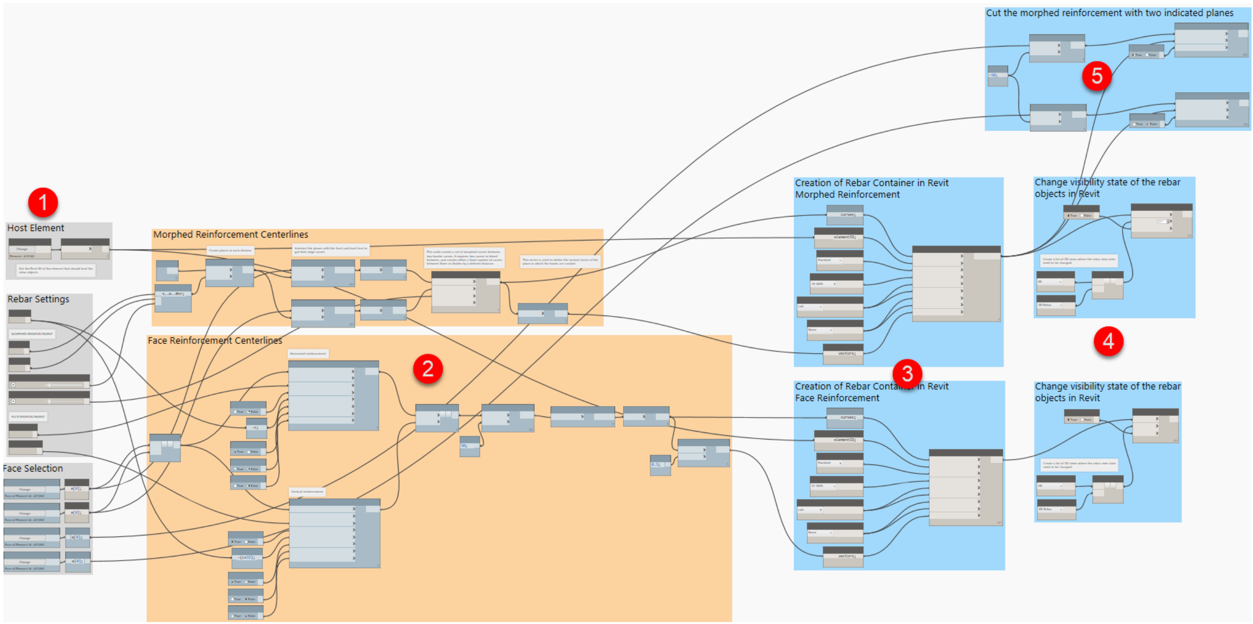
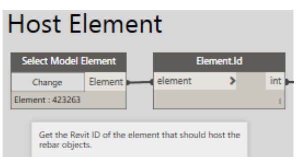


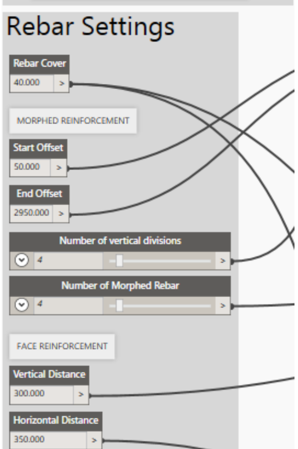
FIG. 16 - REINFORCEMENT MORPHED BETWEEN EDGES (SELECTED LINES)



Step 1 – Input Geometry

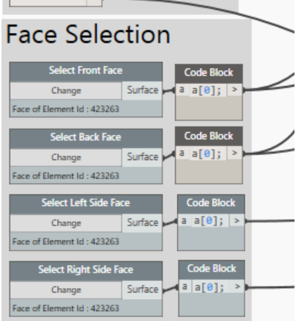


Select the Revit family instance that will host the rebar.



Indicate the desired cover-to-center for the Rebar. In this script a fixed value is used. In this example, it is shown how you could use the Rebar Cover setting from the element and the diameter of the rebar to define this.

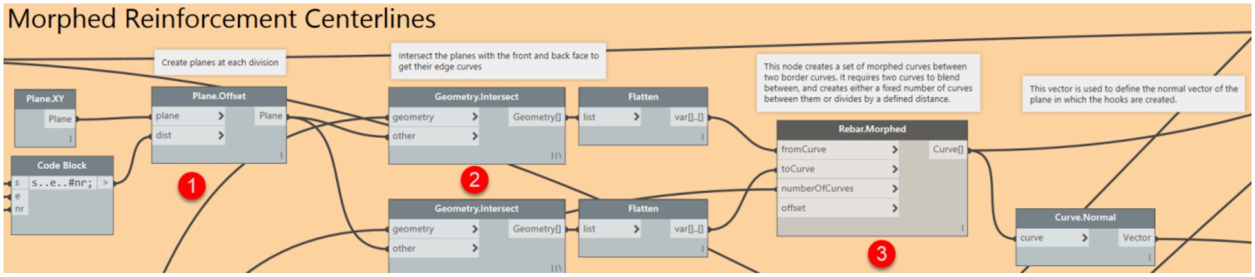
The offsets and layout of the rebar is also given here.



Select the faces as indicated. They will form the boundaries for the morphed and surface reinforcement.

Step 2 – Evaluate surfaces and Create Rebar Lines

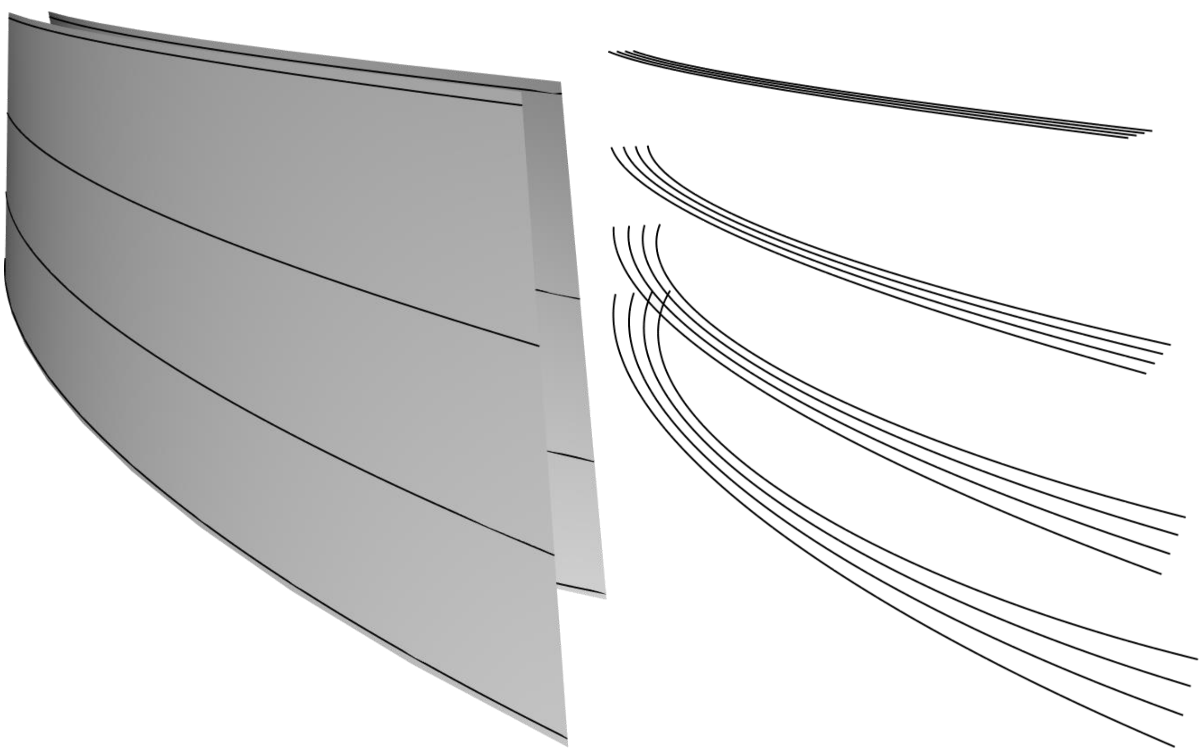
The top part will create morphed curves between two generated curves. This is not just generated between two selected edges, but along the height of the shape, which gets smaller towards the top.



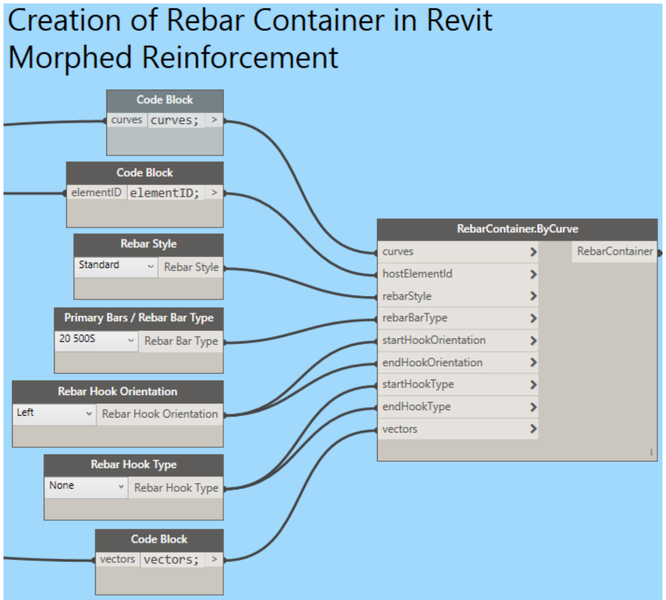
(1) Therefore planes, parallel to the XY plane, are created at indicated distances (start & end offset).

(2) When these planes cut the front and back faces of the host element, this results in curves representing the common edges of these planes and the faces.

(3) The resulting curves are now used as start (“fromCurve”) and end (“toCurve”) curves to create morphed reinforcement with the *Rebar.Morphed* node (from the **Dynamo for Rebar**) package.



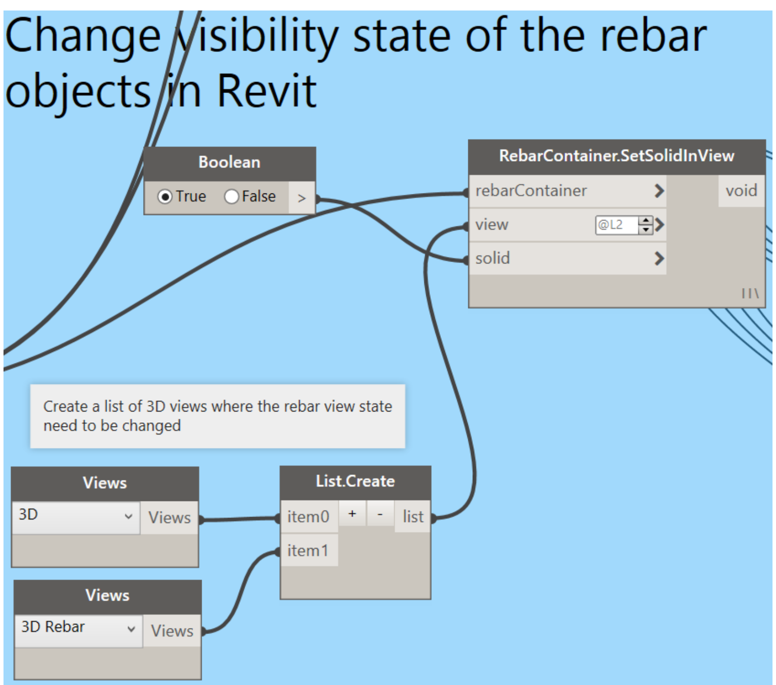
Step 3 – Creation of Rebar in Revit



In this example a *RebarContainer.ByCurve* method is used now. This is previously explained in this part of the handout.

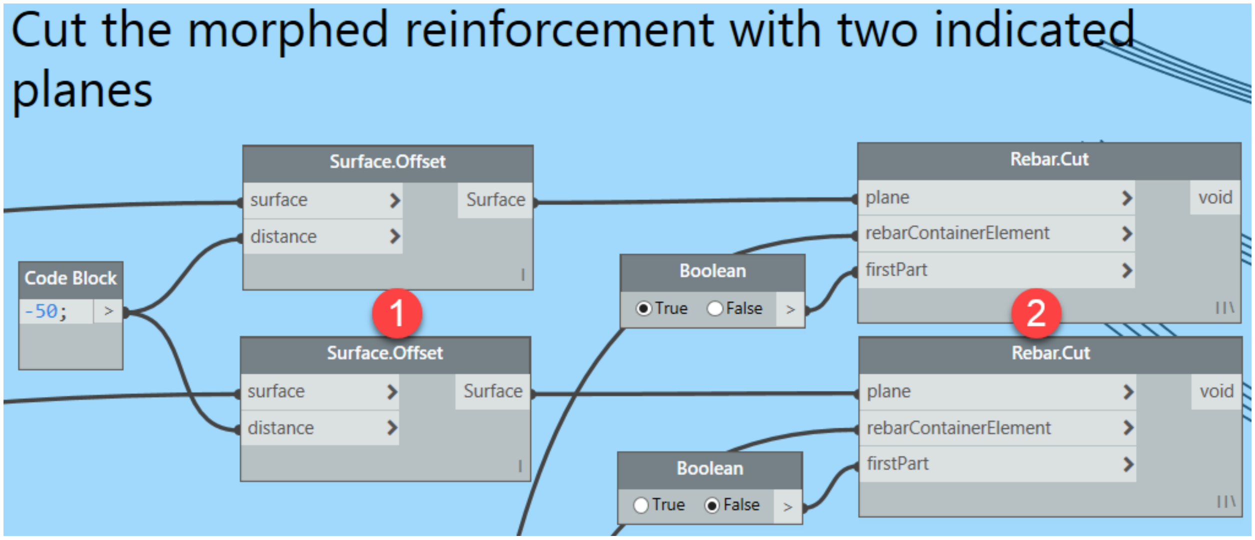
Step 4 – Set visibility state of Rebar in Revit

See previous example.



Step 5 – Cut Rebar

As the outer ends of the morphed rebar in Revit intersect the side faces of the host object, the rebar elements need to be cut, to meet the cover requirements of the element.



(1) The cutting boundaries are defined by offsetting the side surfaces with a specific value (50 mm in this case).

(2) These surfaces are used as cutting planes with the *Rebar.Cut* node (**Dynamo for Rebar** package). The result will be either the left or the right side of the division, depending on the choice for the "firstPart" port.

This operation is done straight in the Revit project and works only on RebarContainers!

Perpendicular Reinforcement

This is an example that illustrates how to create a set of linear curves normal to a surface. It requires the selection of a driving surface and a set of bounding faces to define the end of the projection.

DATASETS

REVIT

Complex Rebar Overview.rvt

DYNAMO

03 Perpendicular Reinforcement.dyn

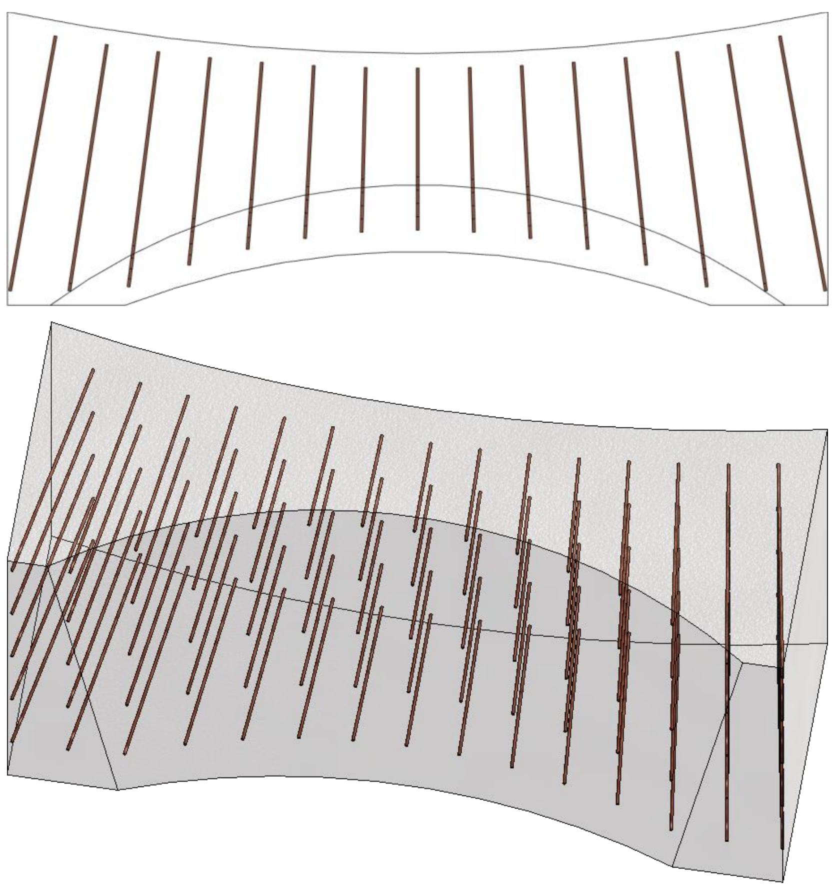
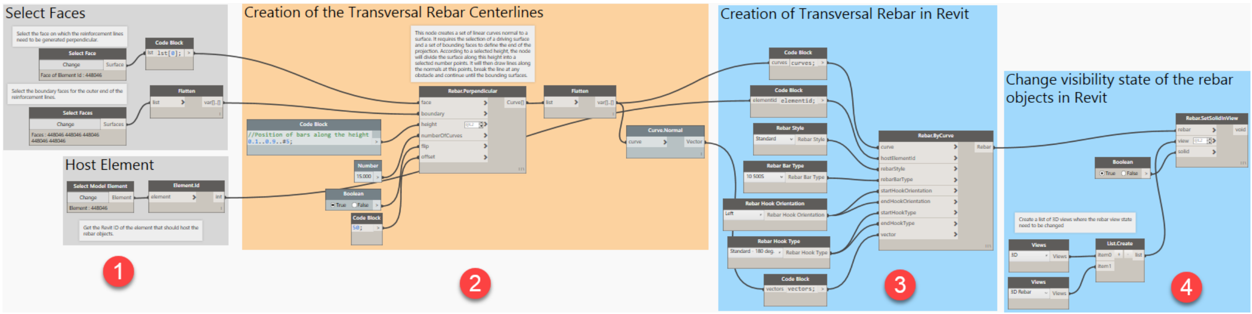
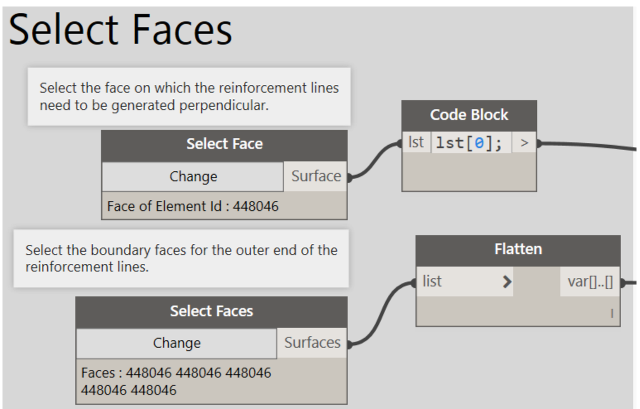


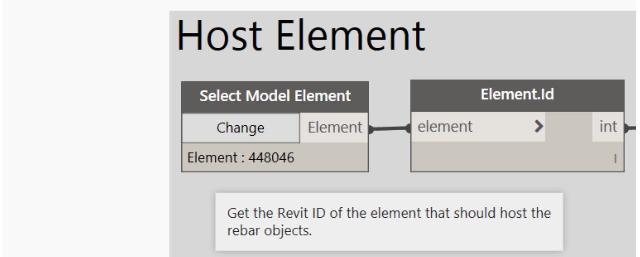
FIG. 17 - REINFORCEMENT PERPENDICULAR TO A SELECTED (CURVED) SURFACE



Step 1 – Input Geometry

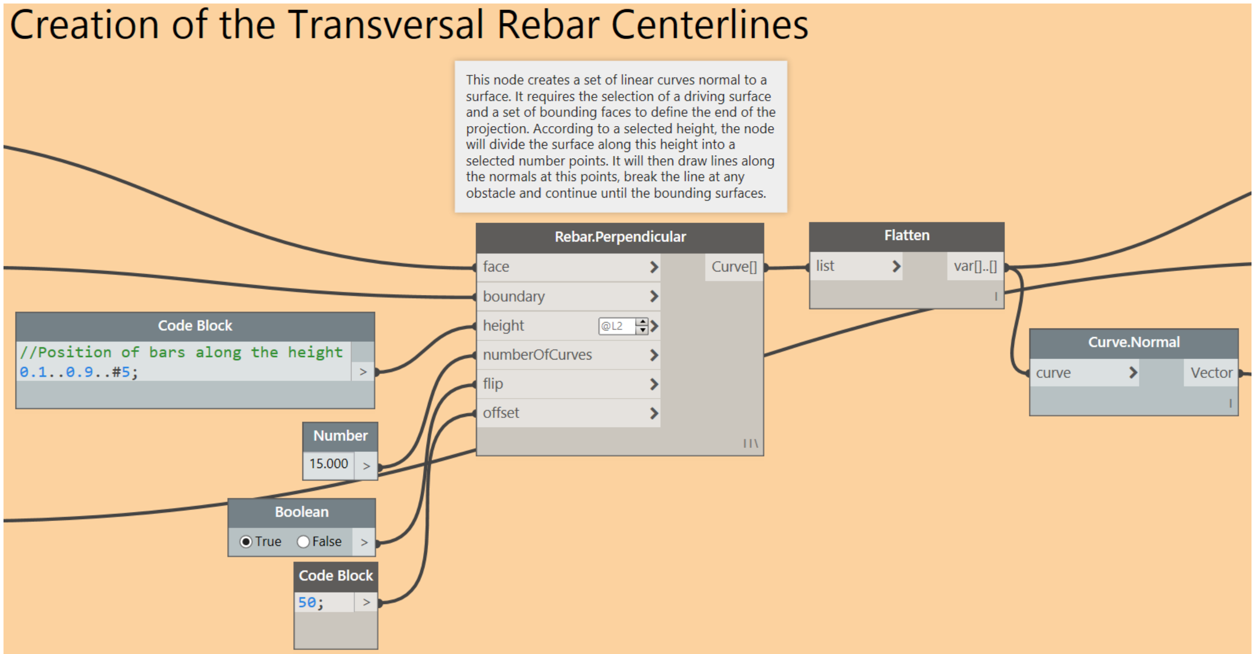


Select the face on which the reinforcement needs to be modelled perpendicularly and the boundary faces that will limit the reinforcement at the other side.



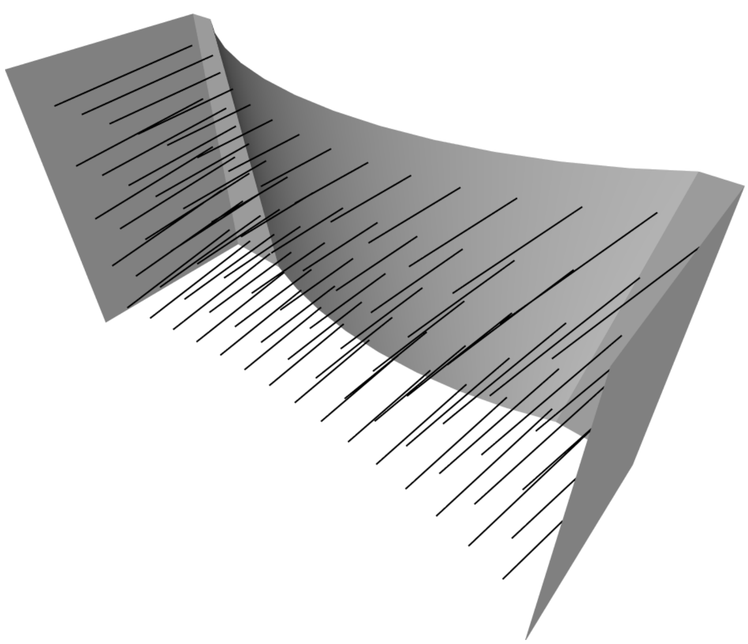
Select the element that will host the rebar.

Step 2 – Evaluate surfaces and Create Rebar Lines

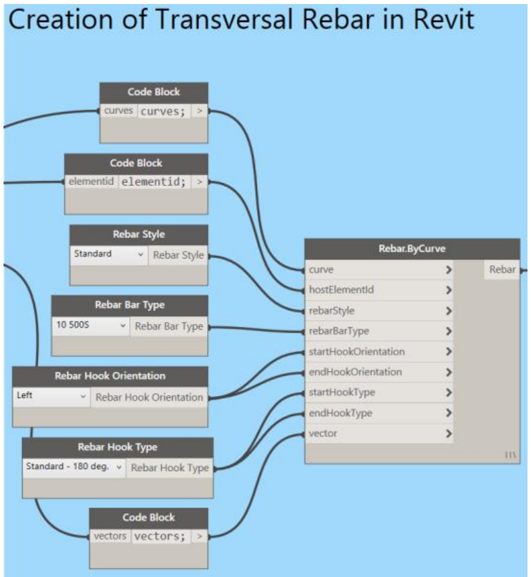


The *Rebar.Perpendicular* node (**Dynamo for Rebar** package) creates a set of linear curves normal to a surface. It requires the selection of a driving surface ("face") and a set of bounding faces ("boundary") to define the end of the projection.

The node will divide the surface along its height into a given number of points. It will then draw normal oriented lines on these points, break the line at any obstacle and continue until the bounding surfaces.

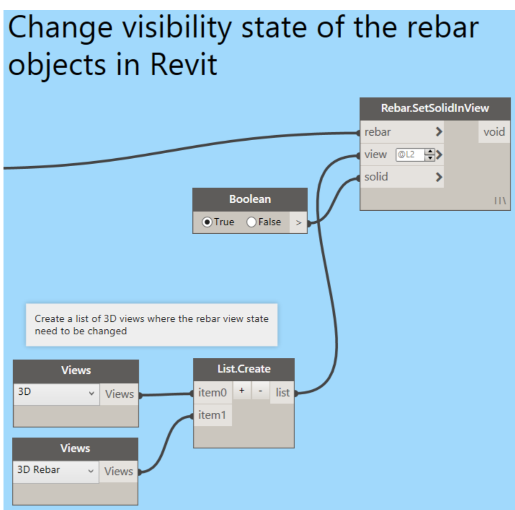


Step 3 – Creation of Rebar in Revit



This is previously explained in [this part](#) of the handout.

Step 4 – Set visibility state of Rebar in Revit



See [previous example](#).

Transversal Rebar Distribution

With these examples you will learn how to distribute transversal rebar (e.g. stirrups or ties) along a complex curved path. Before you start you need to model a basic rebar in the Revit model, which needs to be distributed.

DATASETS

REVIT

Complex Rebar Overview.rvt

DYNAMO

04a Beam Transversal Rebar Distribution.dyn
04b Wall Transversal Rebar Distribution.dyn

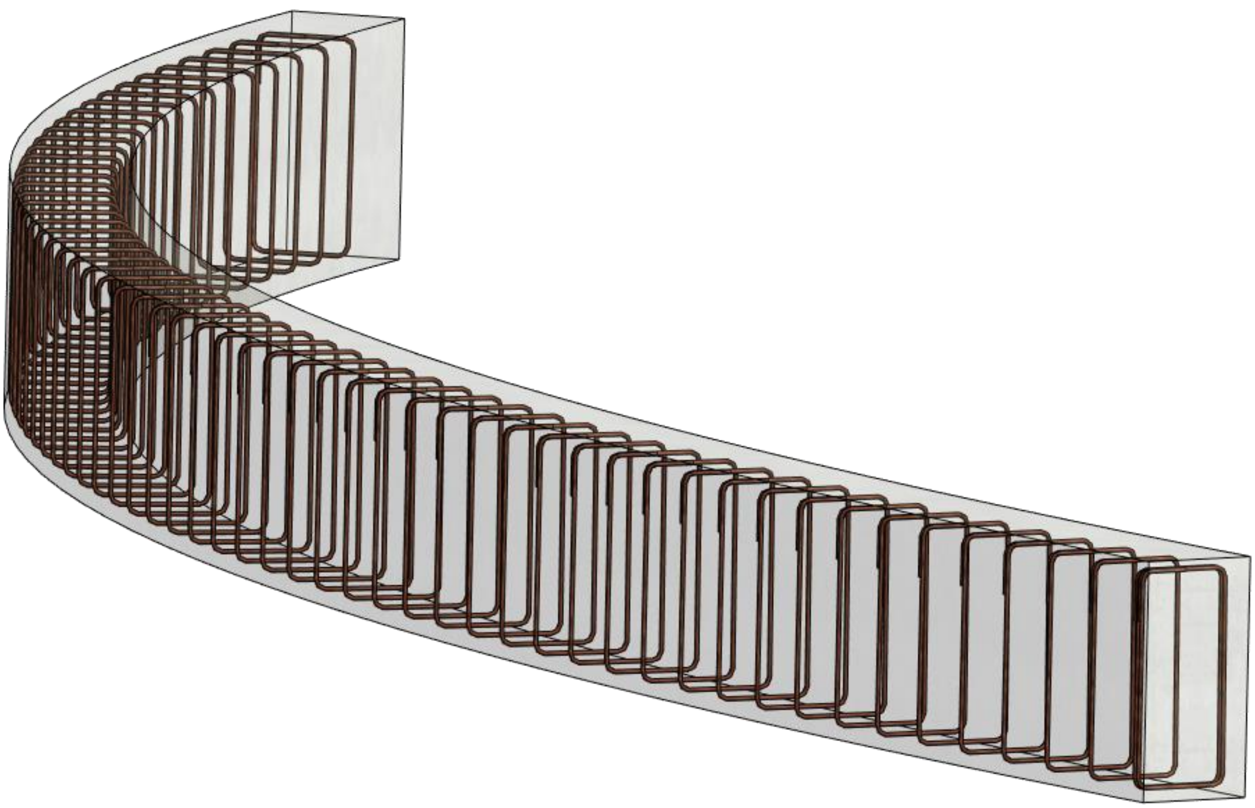
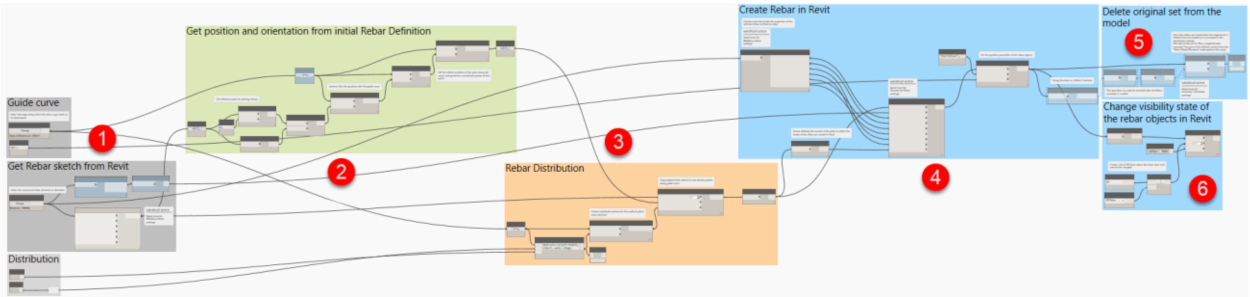


FIG. 18 - REBAR DISTRIBUTION IN A CURVED BEAM

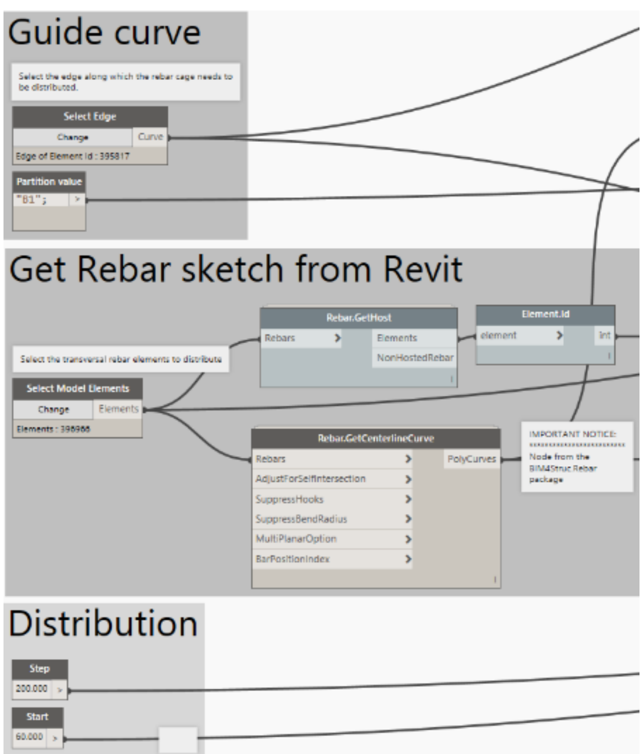


Before you start: Model a base set of reinforcement

Before you start it is necessary to have modelled a base set of transversal reinforcement in the beam or wall. This can be done with the out-of-the-box Revit tools. This base set has the right properties already. Optionally you can save the rebar objects in a Selection Set in Revit, in order to find it back easily after distribution and to delete it then

Step 1 – Input Geometry

This base set of rebar objects will be used to generate a distributed set of rebar. Therefore we need to read the original geometry of the rebar elements into Dynamo. This is done in the "Get Rebar sketch from Revit" group, by means of the *Rebar.GetCenterlineCurve* node (from the **BIM4Struc.Rebar** package). The node returns the sketch behind the 3D Rebar object in Revit. (Read the input port tooltips on the node for more information about the options).



Select the edge along which the rebar needs to be distributed = guide curve.

Select the already modelled rebar element(s) from Revit, to distribute.

With the *Rebar.GetHost* node the family instance that will host the rebar is returned.

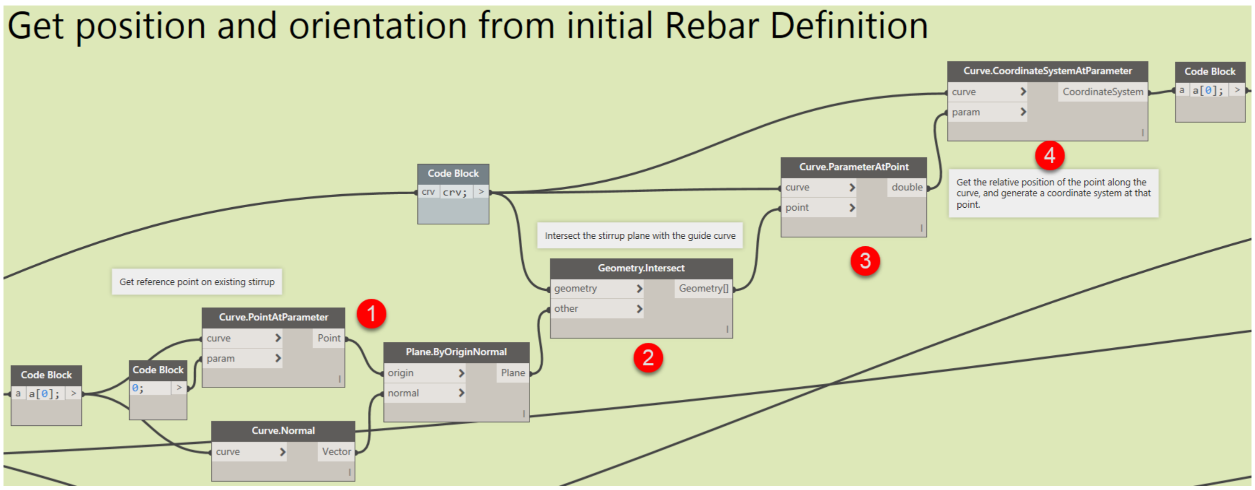
The *Rebar.GetCenterlineCurve* allows it to detect the sketch geometry of the original selected rebar, in order to recreate the new distributed rebar. See also [this chapter](#).

(both nodes from **BIM4Struc.Rebar** package)

The distribution parameters indicate the step between the rebar and the start position along the guide curve. (Expressed in mm).

Step 2 – Position & Orientation of Initial Rebar Definition

This step is needed to detect the right orientation of the base rebar set and its relative position to the selected distribution path (*Select Edge* from Step 1). This information is crucial to make the distribution working. The position and orientation is represented then by the *Curve.CoordinateSystemAtParameter* node.

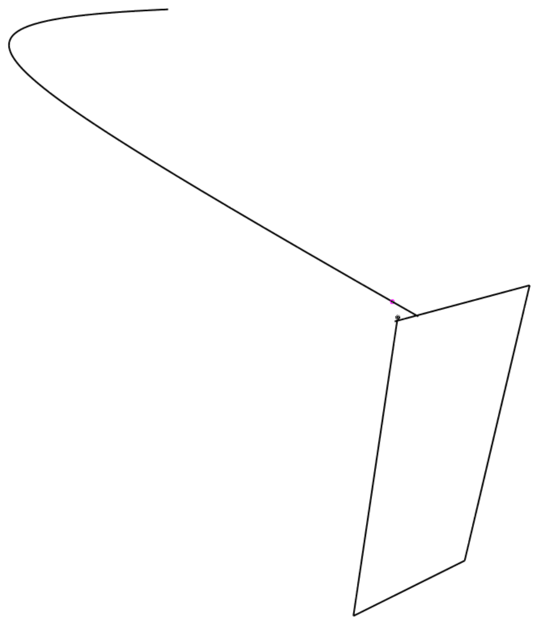


(1) Select the startpoint of the first segment of the rebar sketch from Revit, and detect its normal vector. These values are used to define the plane in which the sketch segment is created.

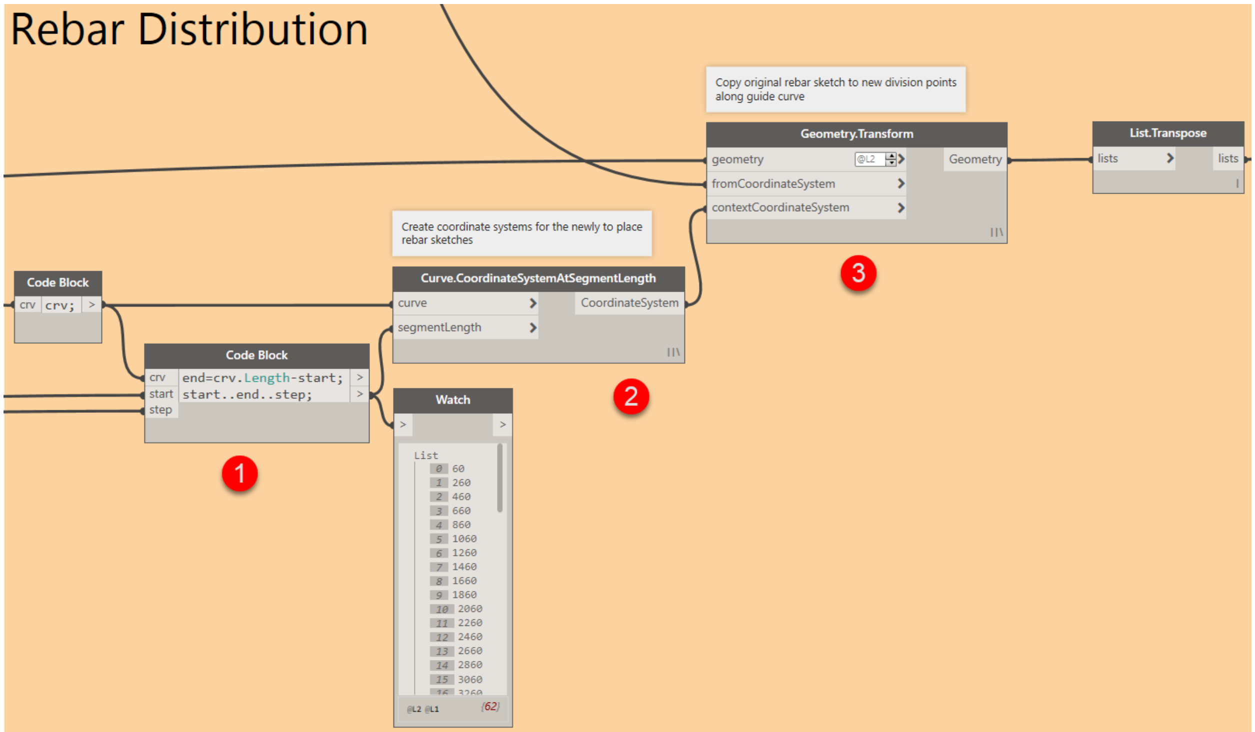
(2) When intersecting this plane with the guide curve, you'll get the point reference on this curve, representing the "origin" of the selected stirrup.

(3) Convert the point on the curve into a relative value on the guide curve.

(4) Create a coordinate system on that position on the curve. This represents then the coordinate system of the rebar sketch on the guide curve, taking its orientation into account. This information is used as origin for the distribution in the next step.



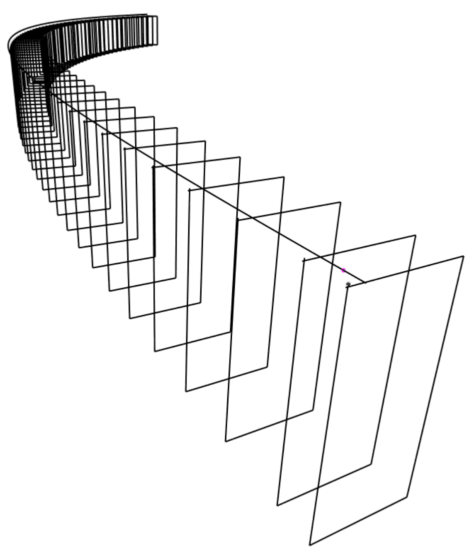
Step 3 – Rebar Distribution



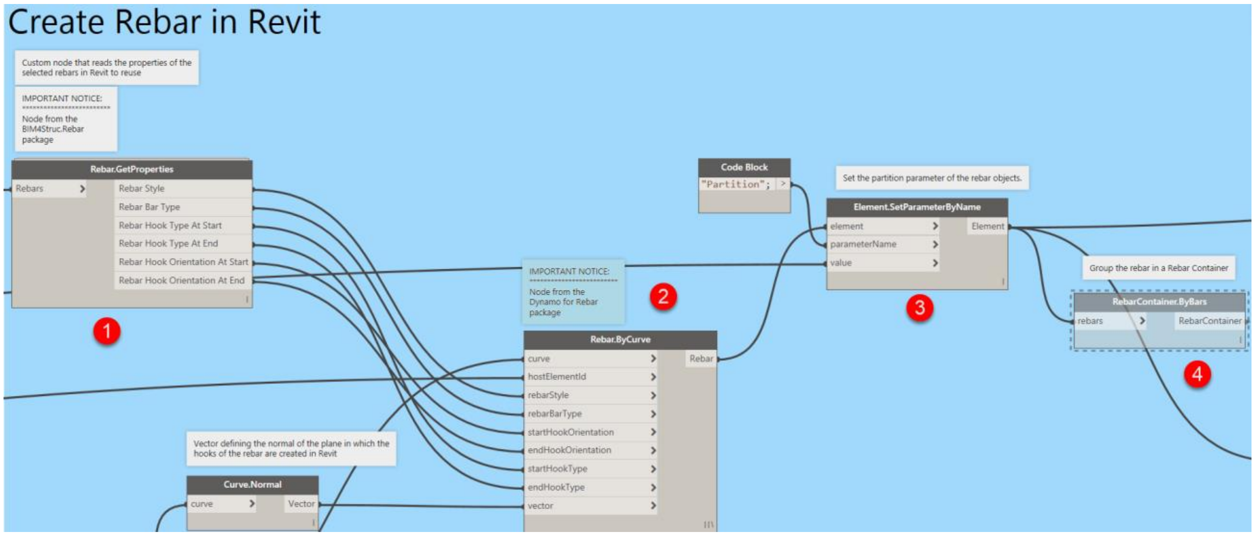
(1) Create a distribution range with the start position and step along the curve.

(2) Generate coordinate systems at each of these distribution positions along the guide curve.

(3) Copy the rebar sketch from step 1, from its origin coordinate system, defined in step 2 to the new coordinate systems. This will make sure that the rebar is oriented to the normal at that position along the curve.



Step 4 – Create Rebar in Revit



(1) The properties of the distributed rebar needs to be identical to the original selected one. The *Rebar.GetProperties* node (**BIM4Struc.Rebar** package) gets all the properties of a selected rebar.

(2) These properties together with the resulting curves from the distribution in the previous step, are used to create the single stirrups in Revit.

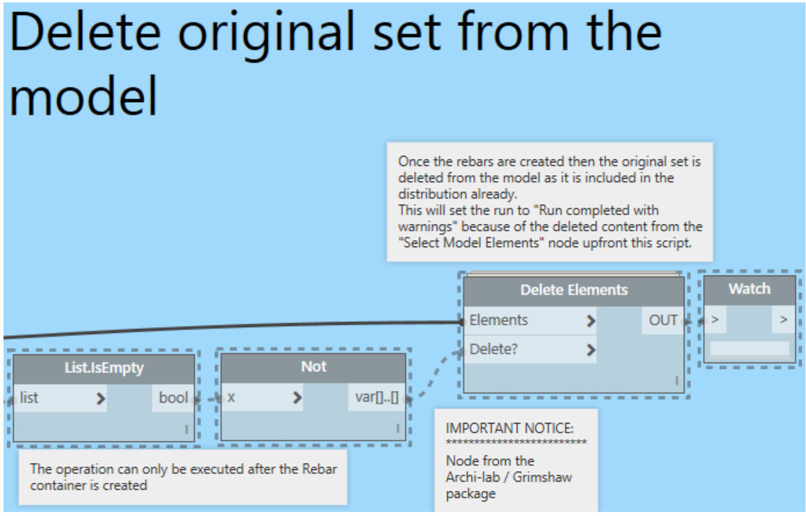
(3) The generated rebar are immediately available to add parameter values to them

(4) Optionally you can combine them in a Rebar Container with the *RebarContainer.ByBars* node (**Dynamo for Rebar** package).

Step 5 – Set visibility state of Rebar in Revit

This step is analogue to [this step](#).

Step 6 – Delete Original Rebar



Optionally the original created rebar in Revit can be deleted. Or in this case, the single rebars are deleted once the Rebar Container is created from them. This is achieved with the *Delete Elements* node from the **Archi-lab / Grimshaw** package. This node is only getting active, once the *RebarContainer.ByBars* node from Step 4, isn't empty.

Radial Reinforcement

The modelling and detailing of reinforcement in a regular, circular floor in Revit can be done very straightforward using Path Reinforcement or by means of polar arrays. Once these circular floors have a varying thickness, and thus have a complex double curved top surface, the reinforcement modelling is best supported by using Dynamo.

In this example the perpendicular and radial reinforcement is generated, taking into account the feasible stock lengths of a rebar (using split & overlap technology).

DATASETS

REVIT

Complex Rebar Overview.rvt

DYNAMO

05 Radial Reinforcement.dyn

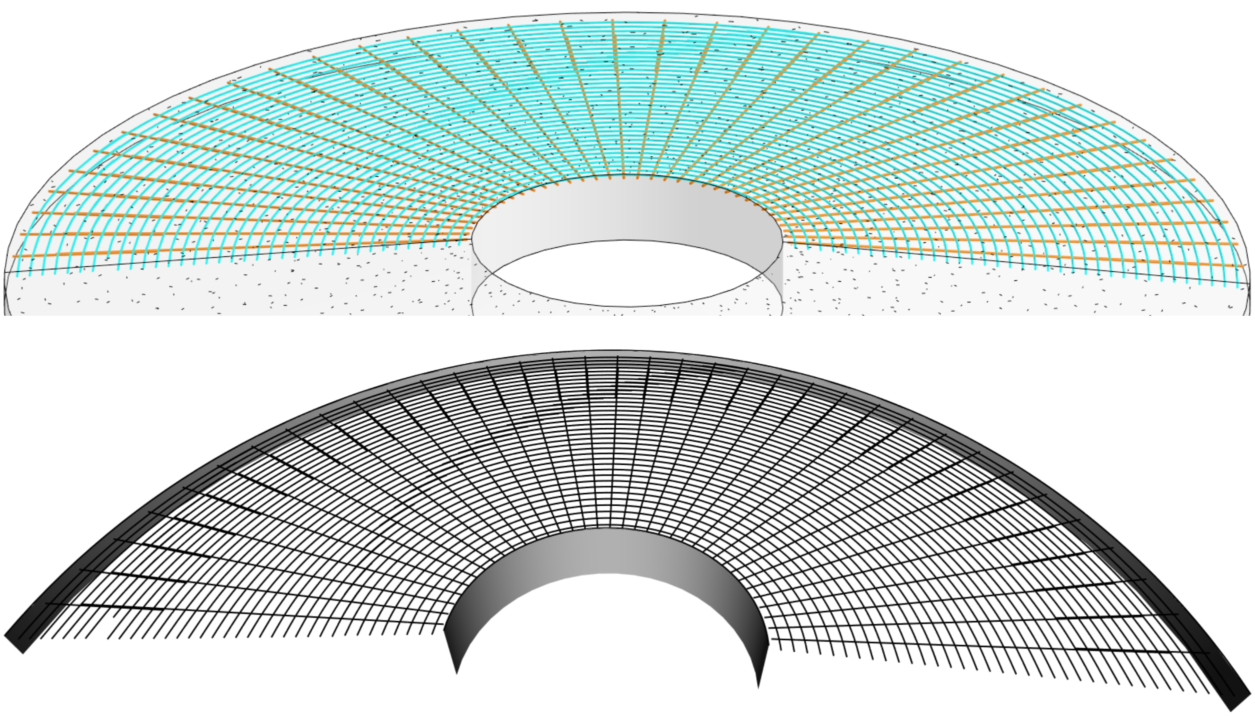
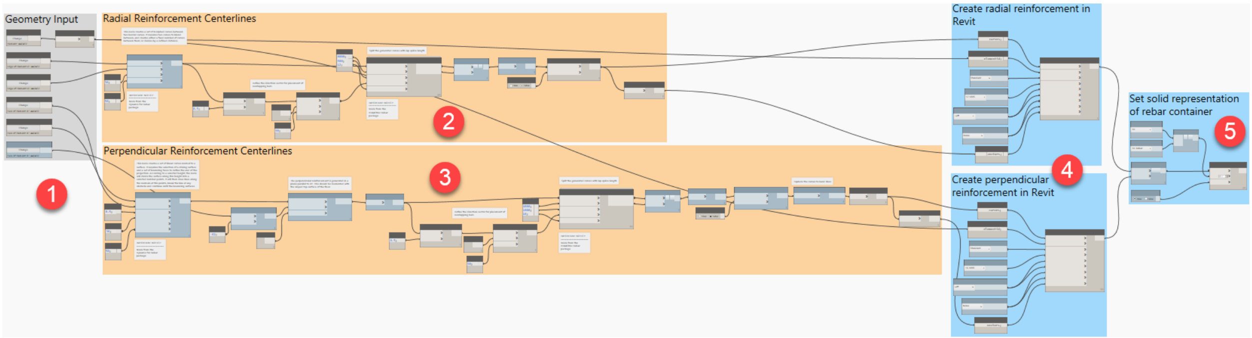
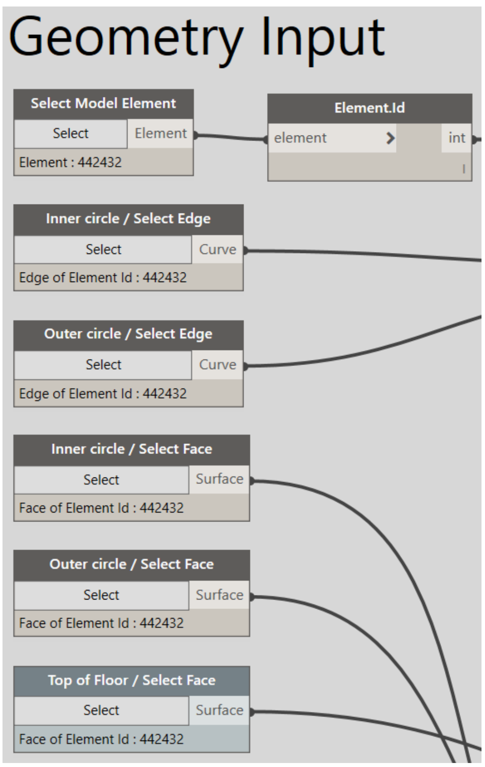


FIG. 19 - RADIAL REINFORCEMENT IN REVIT & DYNAMO

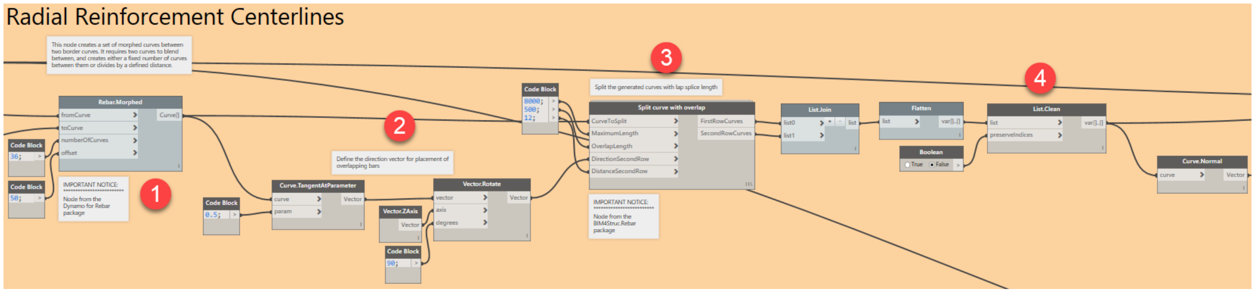


Step 1 – Input Geometry

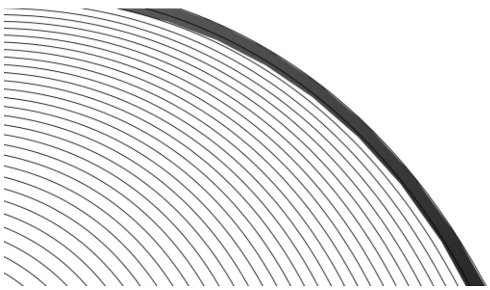
Select the indicated edges, faces and model to as geometrical references.



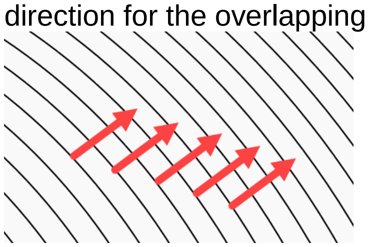
Step 2 – Radial Reinforcement Centerlines



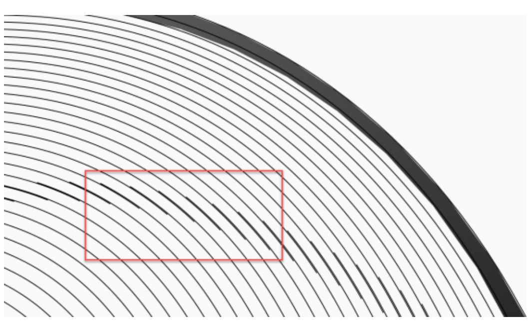
(1) With the *Rebar.Morphed* (**Dynamo for Rebar** package) a defined number of radial rebar lines between the inner and outer circle are generated. With the “offset” value you can set the cover to center distance.



(2) From each of the resulting curves a vector is defined, which will be used as translation direction for the overlapping rebar.

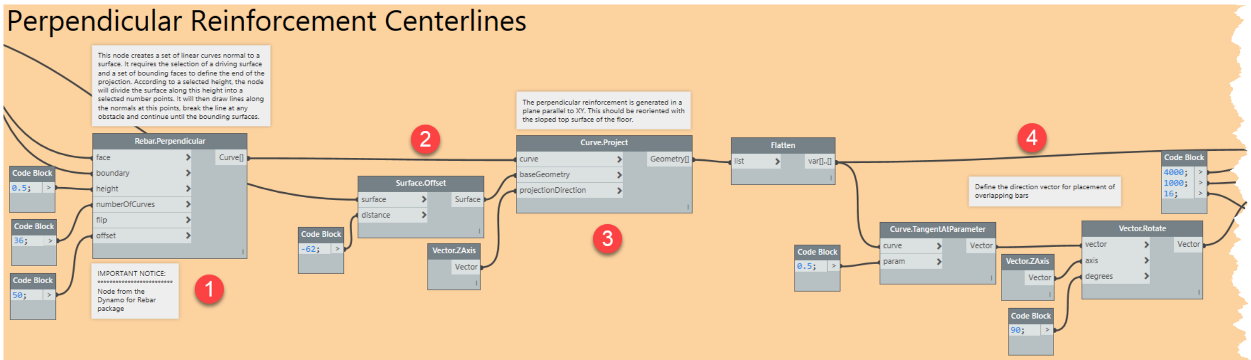


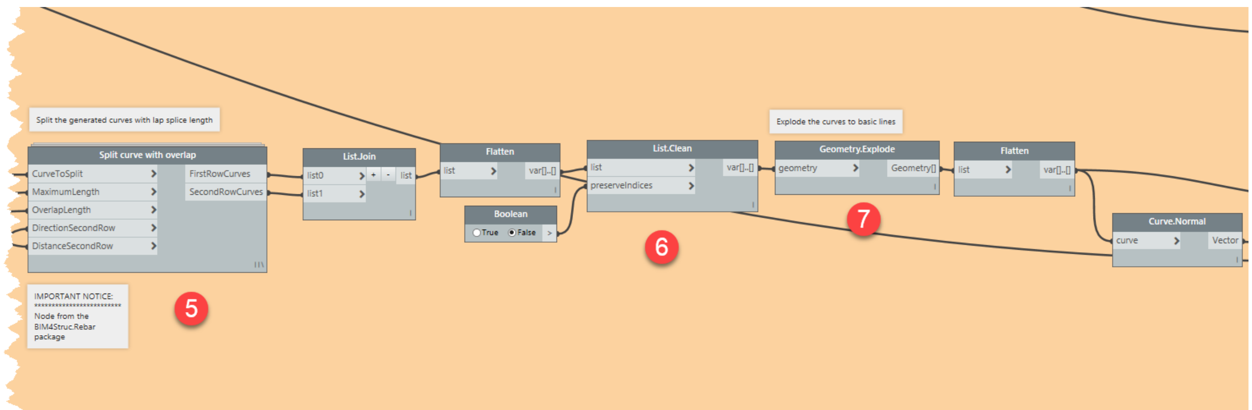
(3) The node *Split curve with overlap* (**BIM4Struc.**Rebar package) splits the original rebar with an indicated fabrication length and assign an overlap. This results in two rows of bars. The second row is positioned next to the first row, according the direction vector defined in (2).



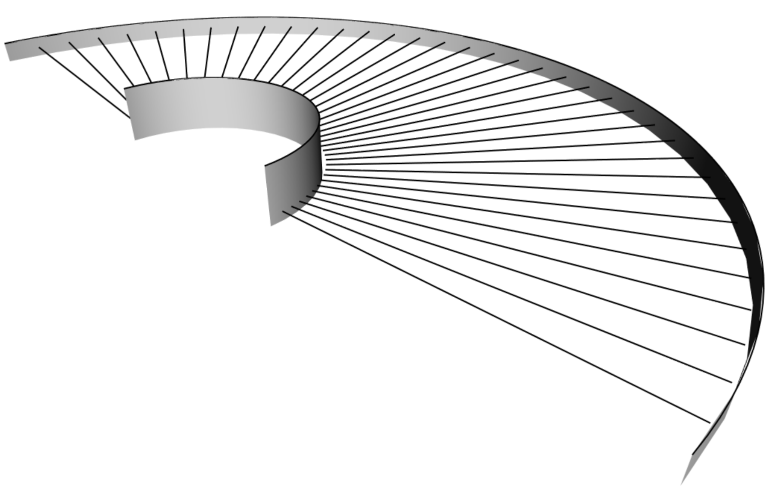
(4) Then clean the resulting array in case there are “null” entries.

Step 3 – Perpendicular Reinforcement Centerlines

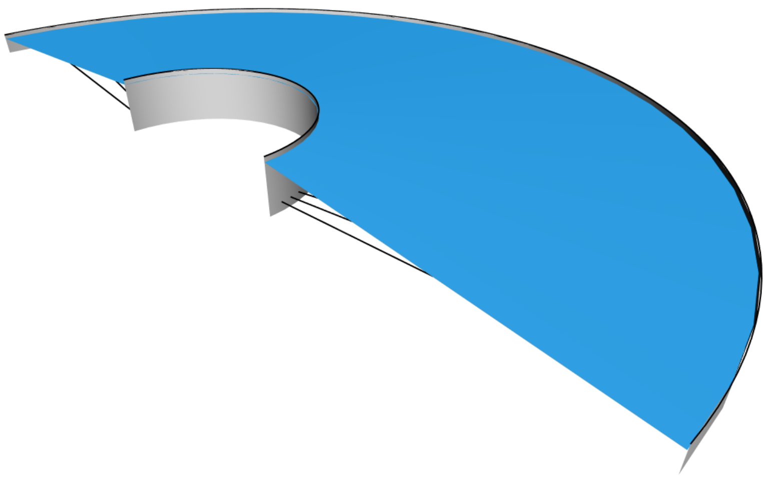




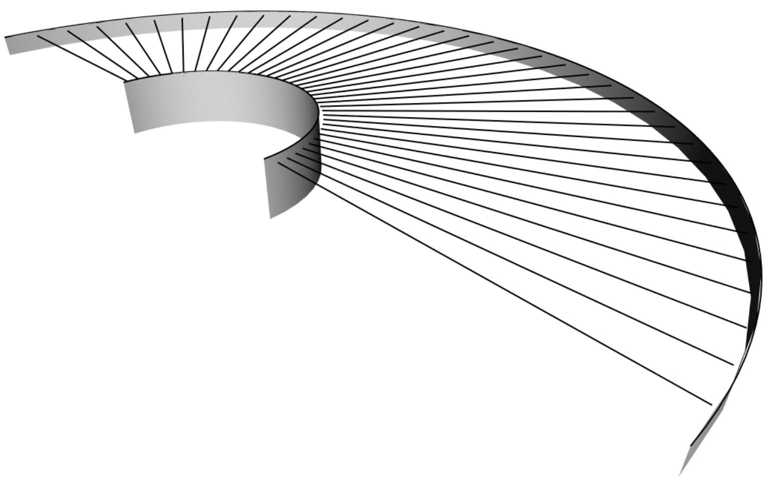
(1) The Rebar.Perpendicular node (Dynamo for Rebar package) allows to create lines perpendicular on the selected "face" (inner circle face) and continues until it reaches the "boundary" face(s) (outer circle face). A "number of curves" are generated at a relative height (e.g. 0.5). The lines start at the specified "offset" with respect to the faces (e.g. the cover distance)



(2) The sloped floor top surface is now offset towards the inside of the solid and generates a new plane that represents to cover-to-center reference plane for the rebar lines.

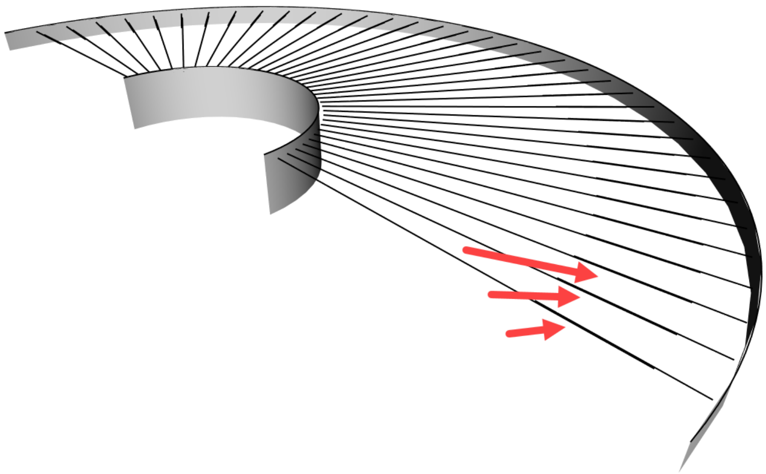


(3) The generated curves from (1) are parallel to the WCS XY plane. To make sure they follow the sloped surface of the floor, they are projected to the cover-to-center reference plane from (2) by means of the *Curve.Project* node.



(4) From each of the resulting curves a vector is defined, which will be used as translation direction for the overlapping rebar.

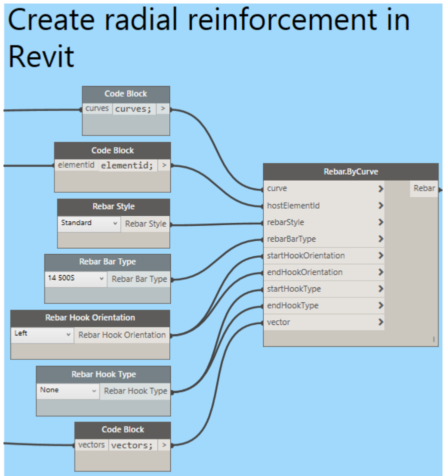
(5) The node *Split curve with overlap* (**BIM4Struc.**Rebar package) splits the original rebar with an indicated fabrication length and assign an overlap. This results in two rows of bars. The second row is positioned next to the first row, according the direction vector defined in (5).



(6) The resulting array is cleaned of "null" entries.

(7) In some cases the resulting geometry needs to be exploded before you can create Rebar objects from it. In this particular case curves with a degree were generated in (5), while they're actually straight lines. With *Geometry.Explode* you get the basis shapes again from the curves.

Step 4 - Create Rebar in Revit



This is previously explained in this part of the handout.

Step 5 – Set visibility state of Rebar in Revit

This step is analogue to this step.

Multi-Segmented Rebar

This short example show how multi-segmented rebar are handled through Dynamo. The goal is to create a bended, multi-segmented (single planar) rebar which gets smaller towards the ends of a floor plate.

DATASETS

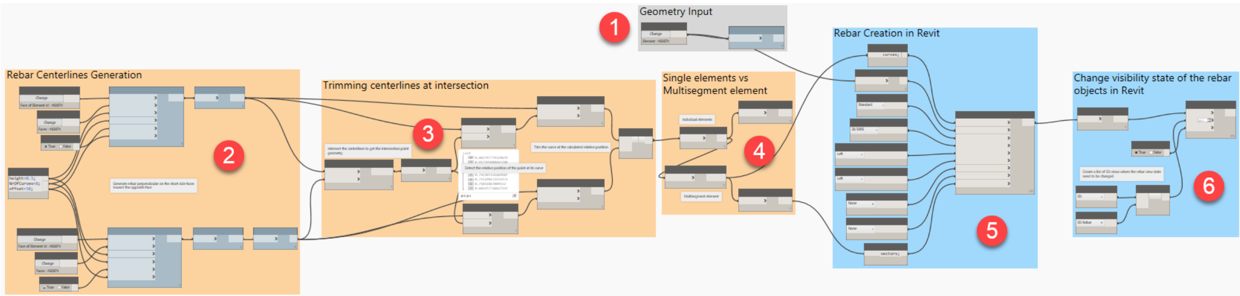
REVIT

Complex Rebar Overview.rvt

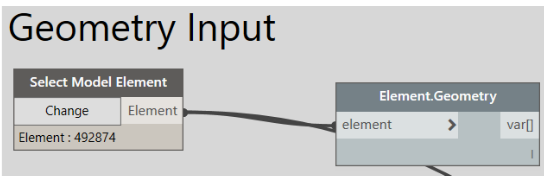
DYNAMO

06 Multi-Segmented Rebar.dyn

FIG. 20 - MULTI-SEGMENTED REBAR IN REVIT

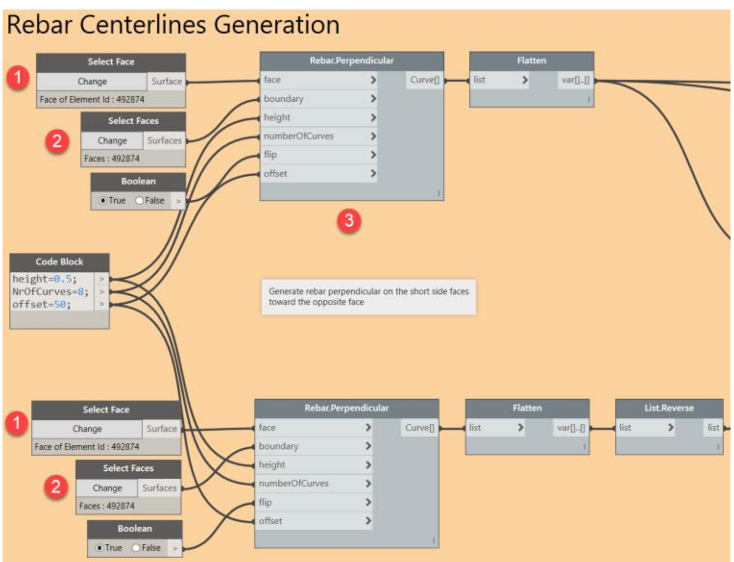


Step 1 – Input Geometry

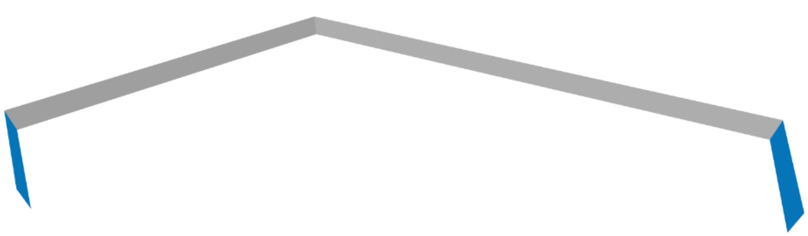


Select the element that will host the Rebar.

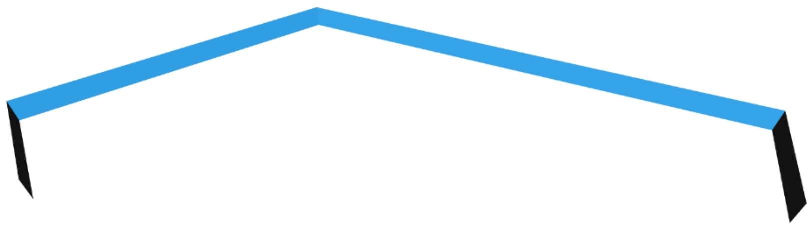
Step 2 – Evaluate surfaces and Create Rebar Lines



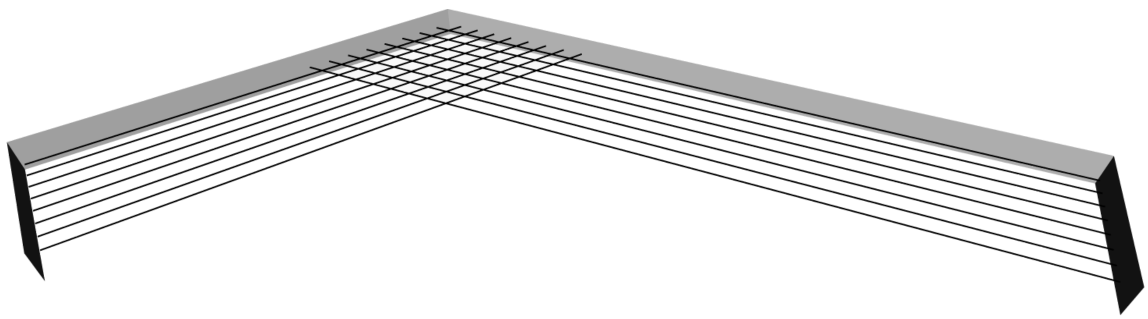
(1) Select the start/end faces of the floor, which you want the *perpendicular* reinforcement to start from.



(2) Select the opposite boundary face at the other end of the rebar line.

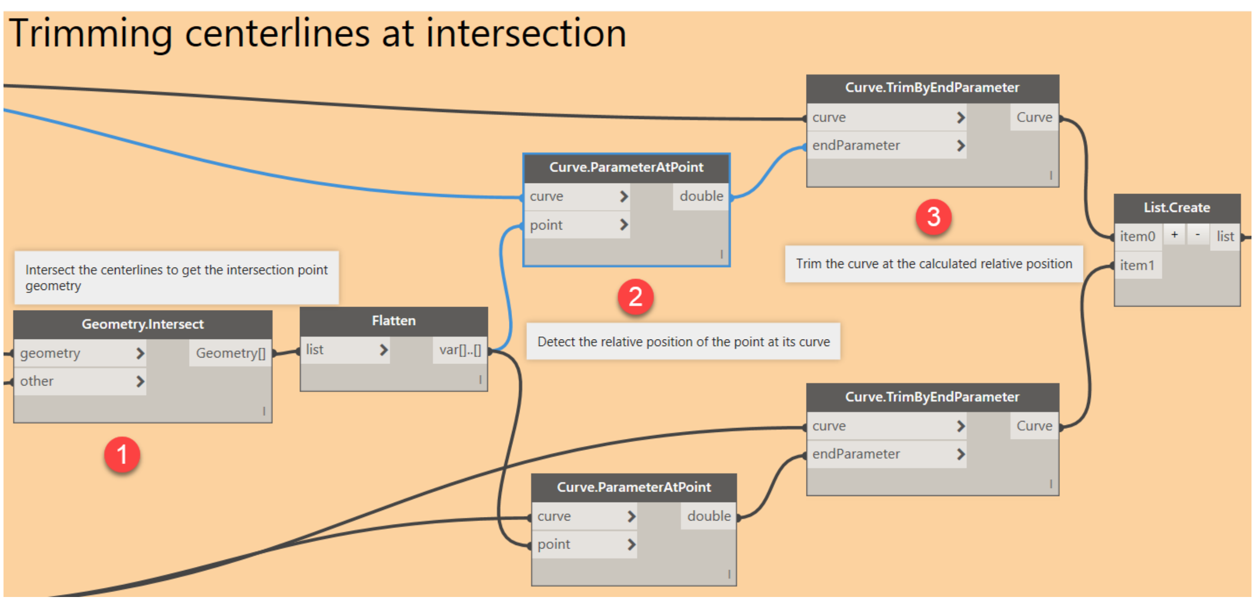


(3) Create the rebar lines with the *Rebar.Perpendicular* node (**Dynamo for Rebar** package).



These lines cross each other, so they still have to be trimmed to a corner.

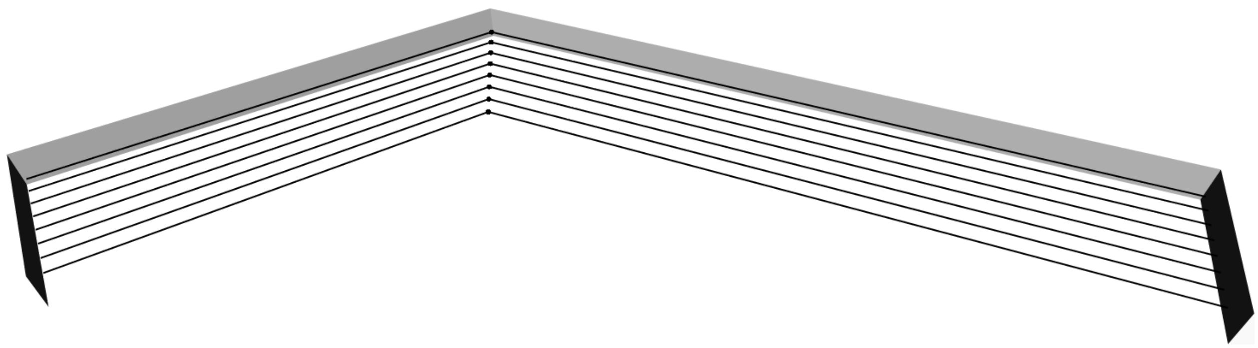
Step 3 – Trim rebar centerlines at intersection



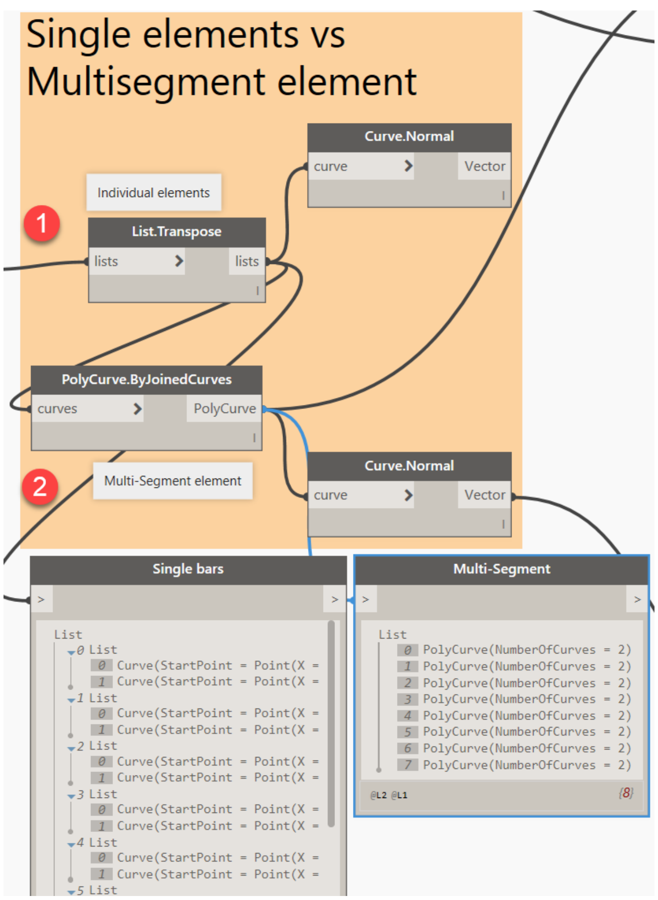
(1) First, get the point at the intersection of two lines with the *Geometry.Intersect* node.

(2) Define the relative position of that point on each of the two curves, with *Curve.ParameterAtPoint*

(3) Trim the curves at the relative parameter with the *Curve.TrimByEndParameter* and create a list of the resulting curves, holding the left ones in the first sub list, and the right ones in the second sub list.

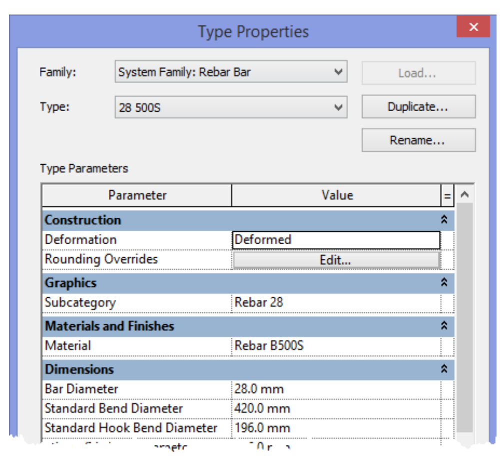


Step 4 – Create multi-segment centerline

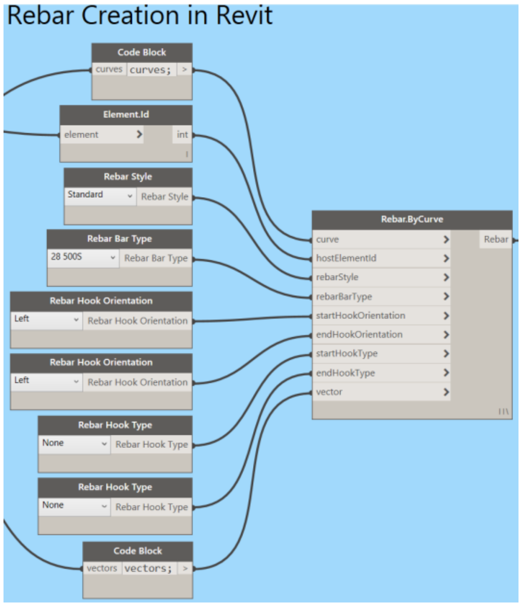


(1) The list of curves from the previous step need to be transposed so the adjacent lines are grouped in subsists. If these lines are connected with the *Rebar.ByCurve* node, than you'll get single rebar elements for each curve.

(2) When joining these curves into a polycurve, they will be treated as one single rebar, made from multiple segments. This will cause a bend radius at the intersection. The size of the bend depends on the type parameter "Standard Bend Diameter" of the Rebar Bar Type in Revit.



Step 5 - Create Rebar in Revit



This is previously explained in [this part] of the handout.

Step 6 – Set visibility state of Rebar in Revit

This step is analogue to [this step].

Practical Use Cases

Bridge Deck Reinforcement

This example will give you some insight in the possibilities that Dynamo offers to Revit when modelling complex rebar cages into a curved concrete bridge deck. The example is part of a full concrete bridge design workflow.

Before you start it is necessary to have modelled a base set of transversal reinforcement in the bridge deck. This can be done with the out-of-the-box Revit tools. This base set has the right properties already. Optionally you can save the rebar objects in a Selection Set in Revit, in order to find it back easily after distribution and to delete it then. (See 0:17 on this **video**)

DATASETS

REVIT

Maurits Project Flyover - 01 Bridge Reinforcement start.rvt

DYNAMO

01 Transversal Rebar Distribution.dyn
02 Longitudinal Rebar Distribution (bottom).dyn

The full workflow applied here is explained on these media:

Blog http://autode.sk/bridgedesign

YouTube https://www.youtube.com/watch?v=qO6QmiNm-No

Transversal Rebar Distribution

The method applied in the script *01 Transversal Rebar Distribution.dyn* is a practical use of "Transversal Rebar Distribution", which is explained here.

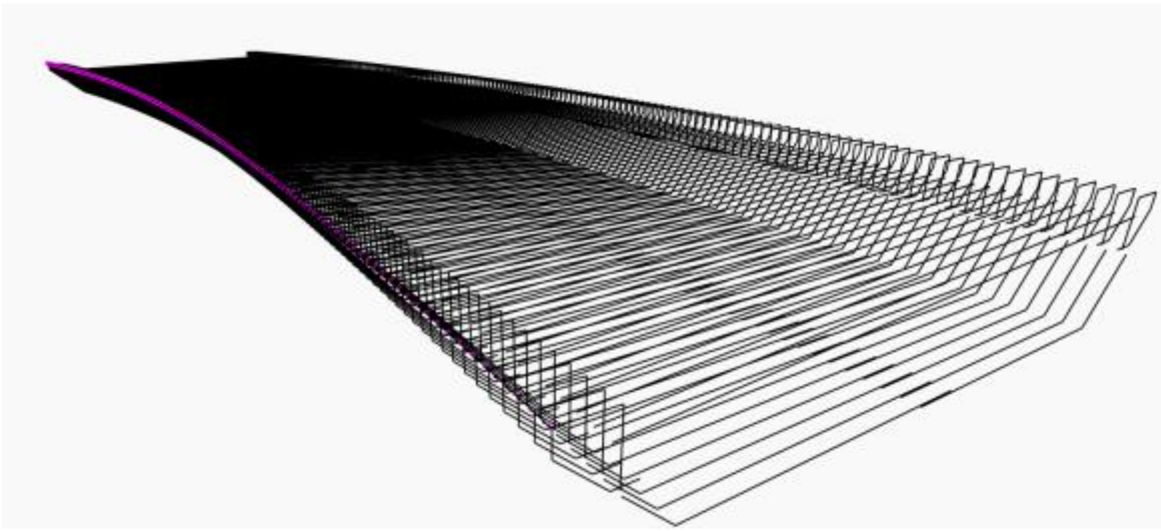


FIG. 21 – TRANSVERSAL REBAR DISTRIBUTION IN DYNAMO

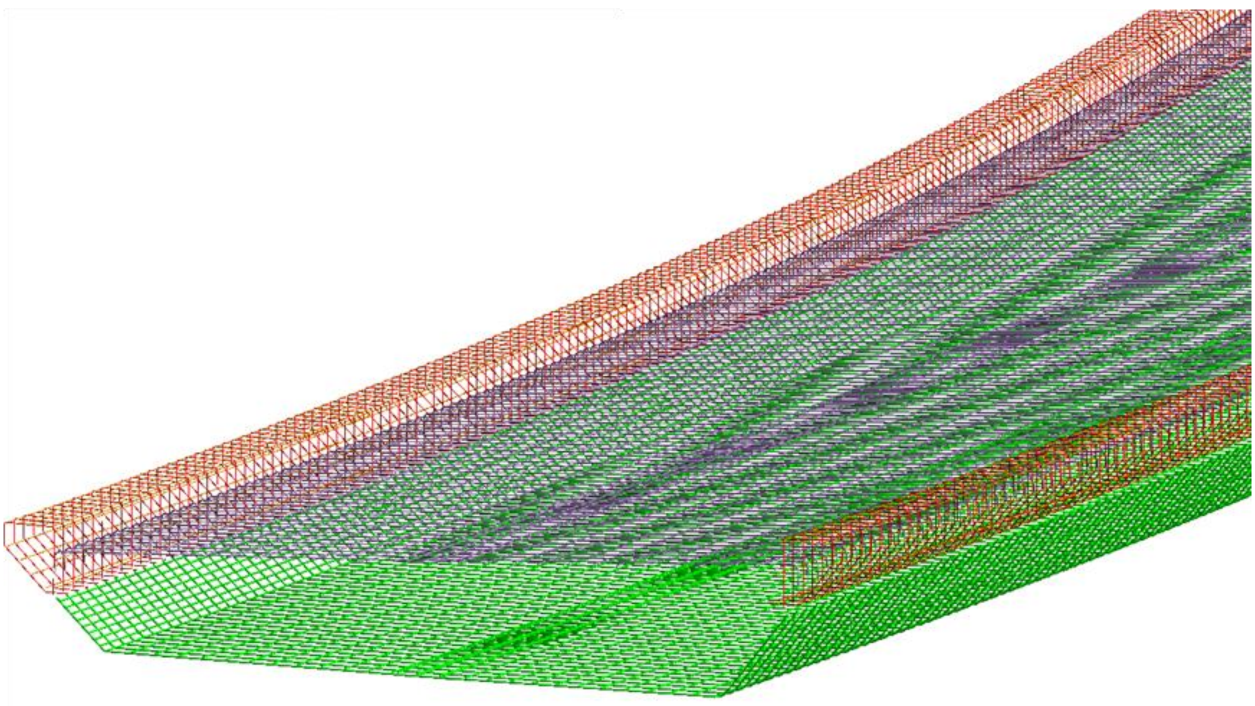
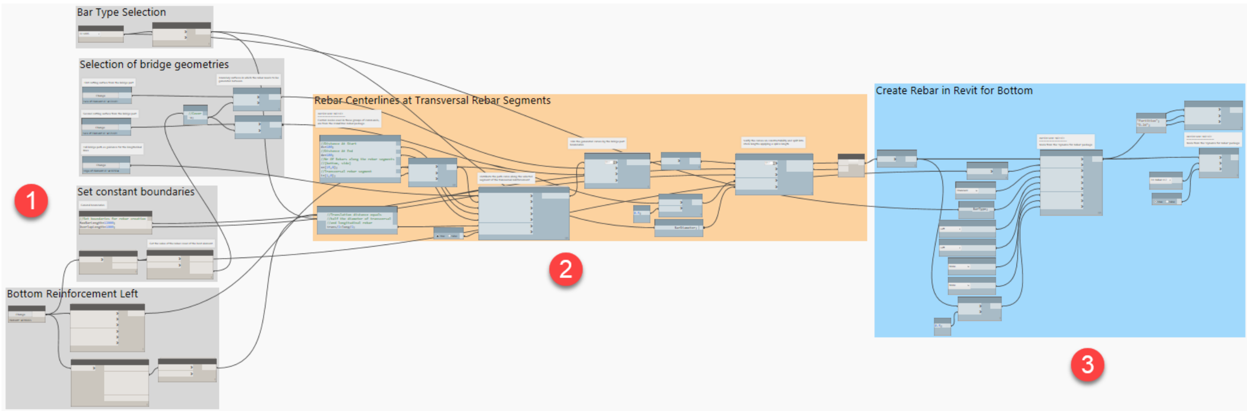


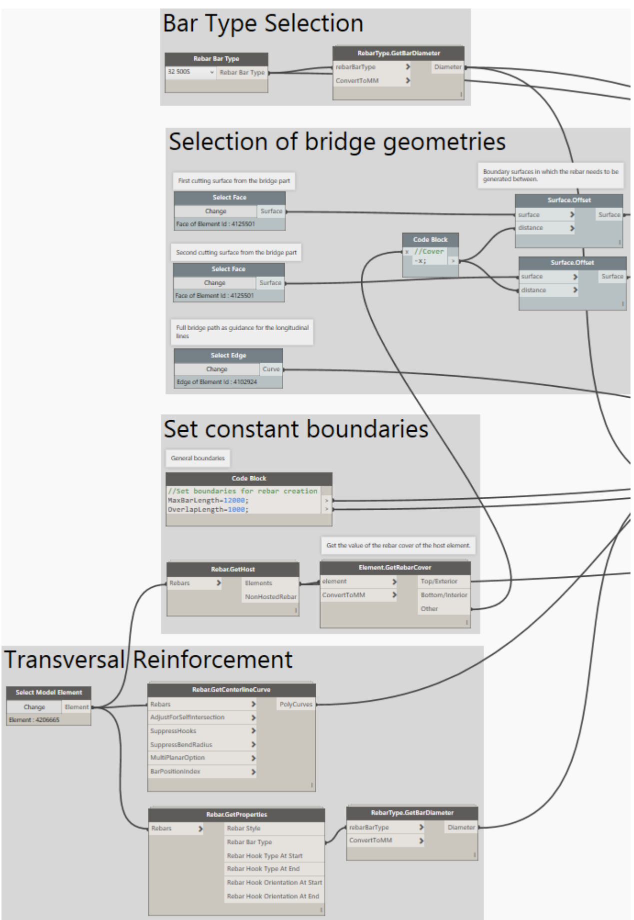
FIG. 22 - REBAR DISTRIBUTION IN CONCRETE BRIDGE DECKS

Longitudinal Rebar Distribution

The script *02 Longitudinal Rebar Distribution (bottom).dyn* makes it possible to generate longitudinal reinforcement inside of the transversal rebar cage. It is based on the geometry of the transversal reinforcement and the guide path (curve) of the bridge element.



STEP 1 – INPUT GEOMETRY



Choose the Rebar Bar Type for the longitudinal rebar.

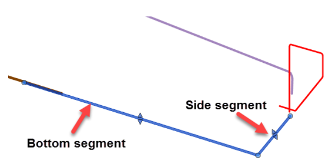
Select the end faces of the bridge deck part (vertical faces) in between which the rebar needs to be generated (green in the image below). The offset surfaces are used then to cut the longitudinal rebar lines.

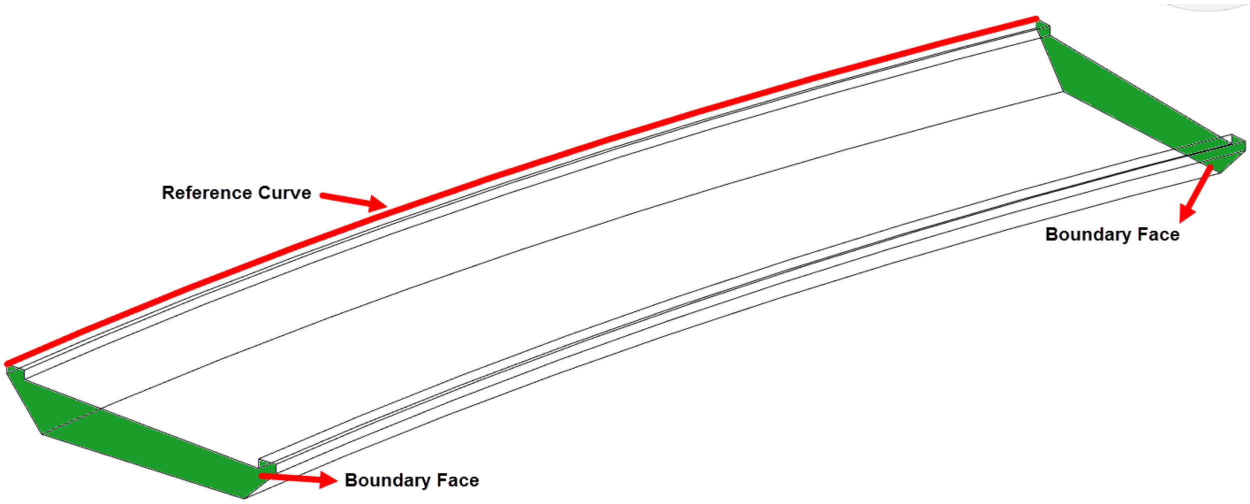
Also the guide curve that represents the reference for the longitudinal reinforcement needs to be selected here (red in the image below). This must be the full bridge length and not the edge of one part of the bridge.

Set some boundaries such as the maximal fabrication length and desired overlap length for the longitudinal rebar.

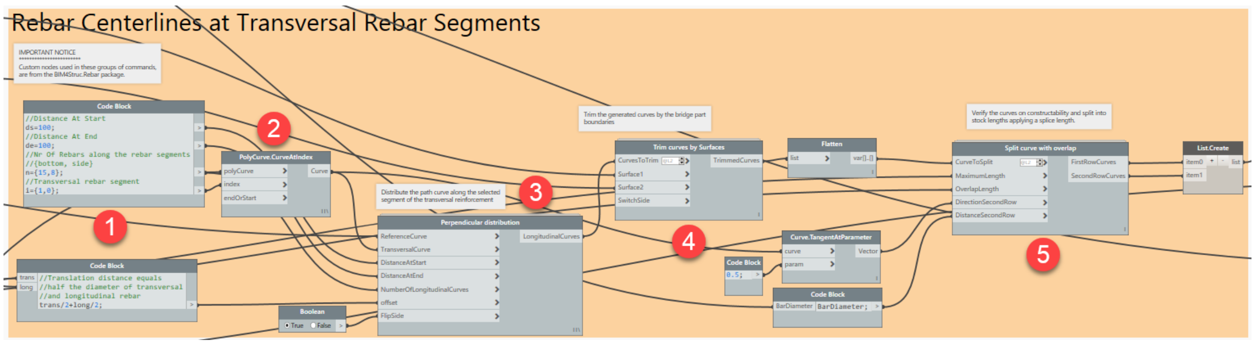
The cover is detected from the element that will host the rebar.

Select the transversal rebar element which will act as transversal curve for the perpendicular distribution. Get the polycurve which the rebar is built of.





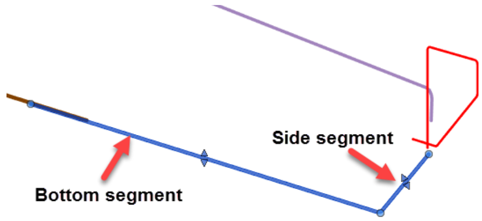
STEP 2 – REBAR CENTERLINES AT TRANSVERSAL SEGMENTS



(1) Before you start generating geometries, the dimensions need to be set.

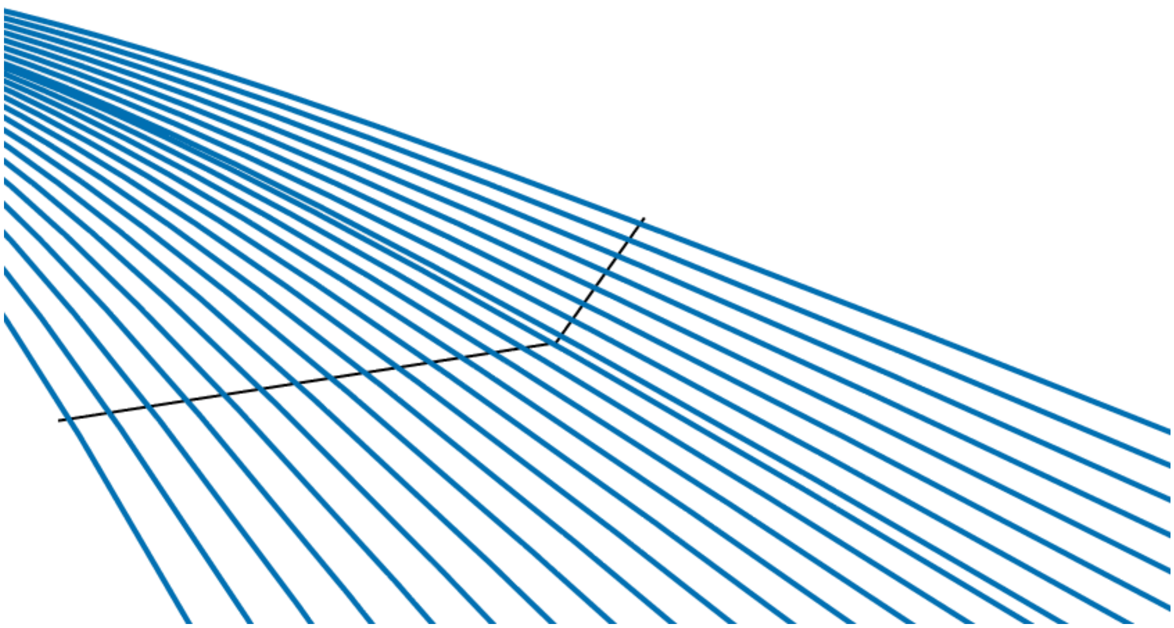
`n={15,8}` refers to the number of rebars along the specific segments of the transversal rebar. 15 along the bottom rebar segment, 8 along the side segment.

`i={1,0}` refers to the index along the Polycurve, generated by the *Rebar.GetCenterline* node in Step 1. 1 = bottom segment, 0 = side segment. This index depends on how the rebar sketch is initially created in Revit



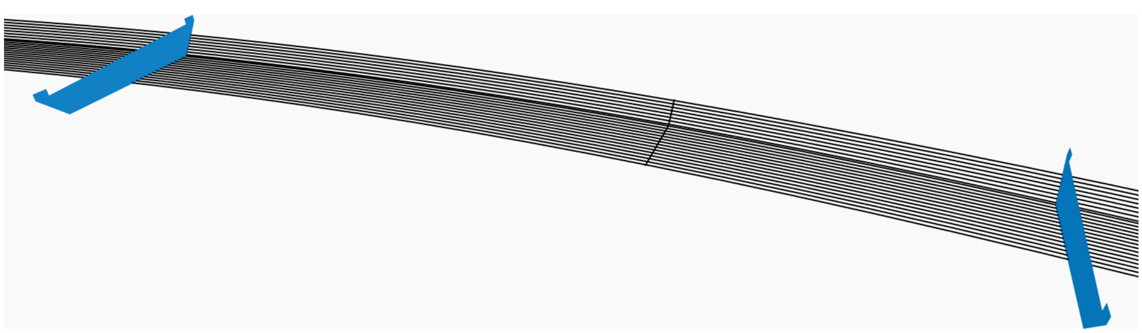
(2) Get the index of the transversal rebar polycurve with the *PolyCurve.CurveAtIndex* node.

(3) The custom node *Perpendicular distribution* (**BIM4Struc.Rebar**) package, distributes the "ReferenceCurve" along the direction of the "TransversalCurve". Distribution starts/ends at the absolute distances indicated by "DistanceAtStart" or "DistanceAtEnd".

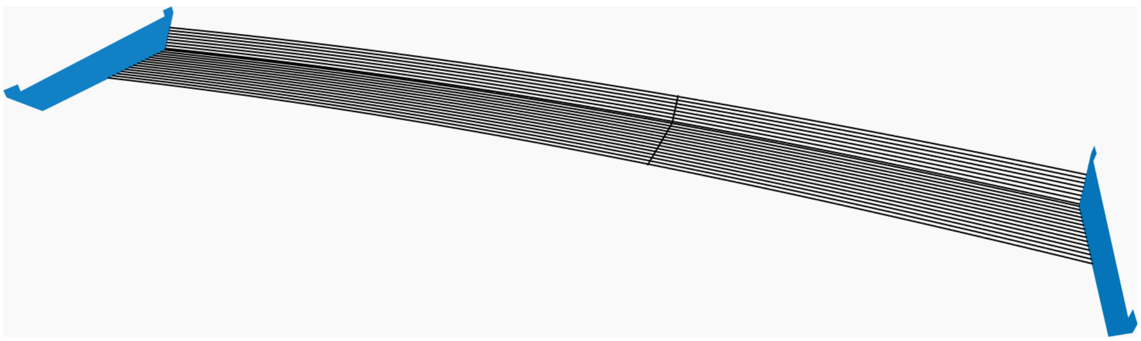


(4) The generated curves are exceeding the boundary surfaces. With the node *Trim curves by Surfaces (***BIM4Struc.Rebar** package) it's possible to trim the curves to the right boundaries.

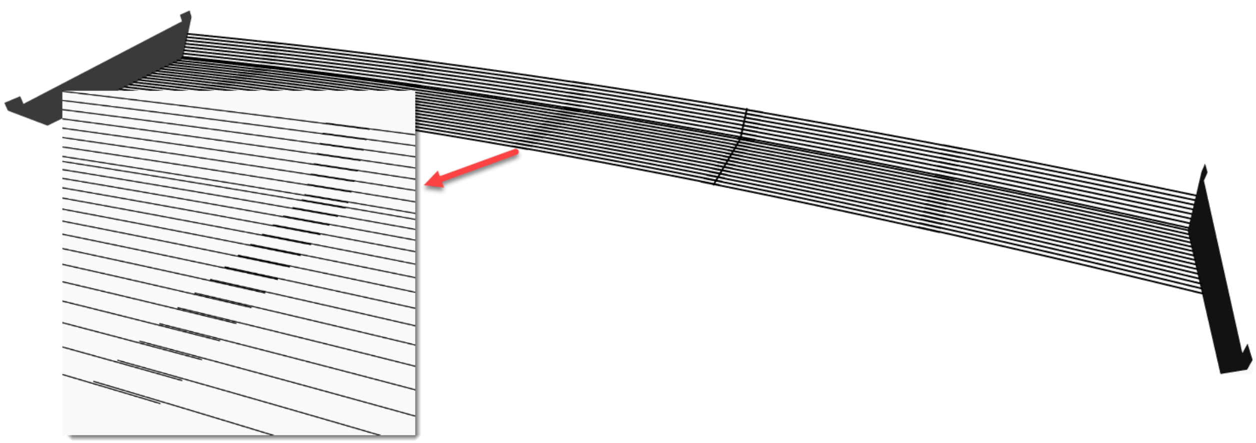
Before Trimming:



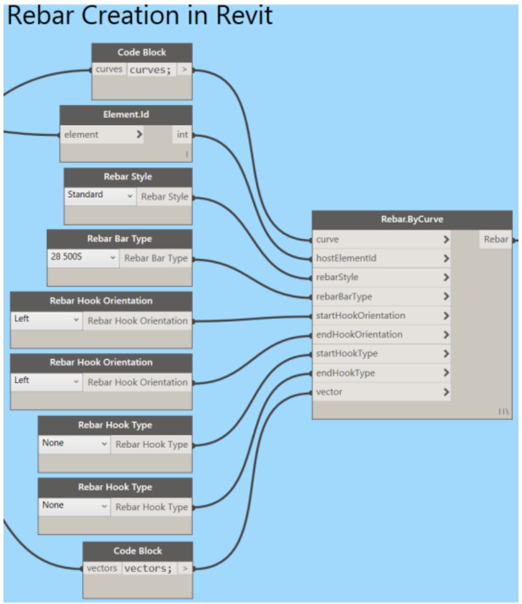
After Trimming:



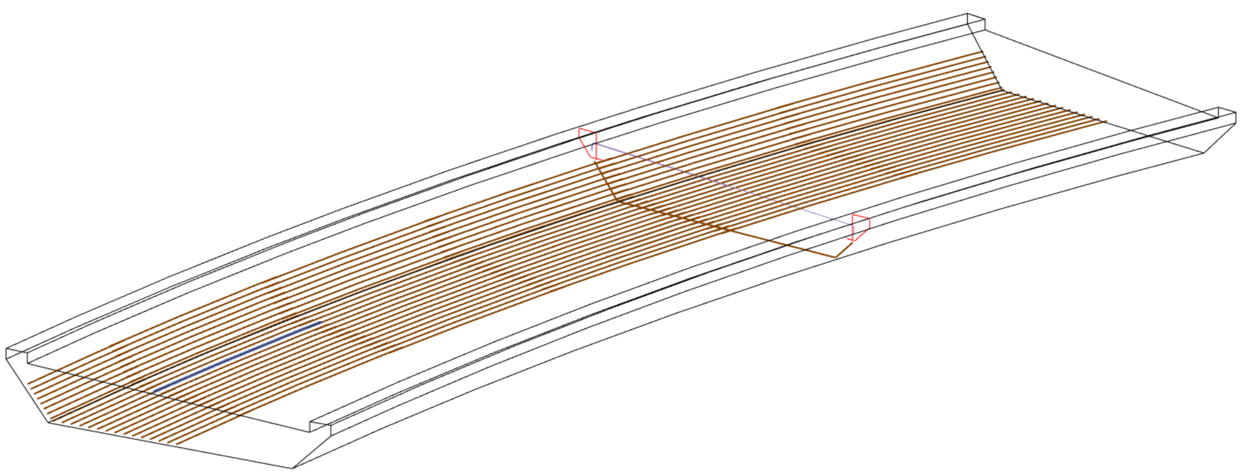
(5) Finally the rebar centerlines are too long for fabrication. With the node *Split curves with Overlap* (**BIM4Struc.Rebar** package) it's possible to split the curves, add splice length and create a new line next to the original with overlap. This is also explained in this example.



STEP 3 – CREATE REBAR IN REVIT



This is previously explained in [this part](#) of the handout.



Diabolo Pier Reinforcement

In one of my classes at Autodesk University 2015, I presented a part about how to model Structural Rebar in Revit with Dynamo, in a complex bridge pier, called the "Diabolo Pier".

The workflow I applied involved MS Excel, Revit and of course Dynamo.

DATASETS

REVIT

Diabolo elevated railway - 01 Rebar detailing start.rvt

DYNAMO

Diabolo Column Rebar Base Part.dyn
Diabolo Column Rebar Top Part.dyn
Diabolo Column Rebar Configuration.xlsx

AUTODESK UNIVERSITY 2015

You can find a detailed handout on this topic and watch a recording of this class here:

MSF11845 – Dynam(o)ite Your Design from Concept to Fabrication

Recording: starting at 00:18:39
Handout: Case 2, at page 35.

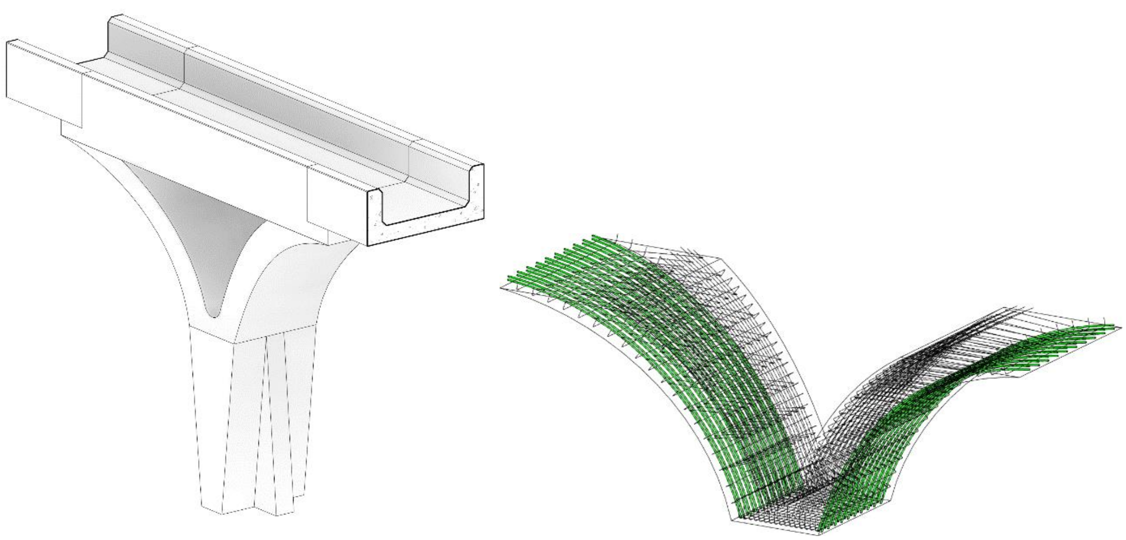


FIG. 23 - COMPLEX REBAR DESIGN OF DIABOLO PIER

BIM4Struc.Rebar node list

Below a list of nodes that are available in the package **BIM4Struc.Rebar v. 1.2.3** and where they are explained in this handout.

Node	Description	Example
AREA REINFORCEMENT		
AreaReinforcement.CreateFromHost	Create Area Reinforcement in a wall or floor according to the wall/floor sketch geometry. The exterior and interior reinforcement are the same by default.	Click here
CURVE EVALUATION		
Perpendicular Distribution	Node that distributes curves along selected transversal curves (e.g. distribution of longitudinal rebar along transversal rebar segments.)	Click here
Split curve with overlap	Splits a curve into multiple segments with a maximum length and a gap between two consecutive ones, and create a second curve next to it with similar division and overlap. This node is to prepare rebars centerlines with splice lengths.	Click here Or Click here
Trim curves by Surface	Trim curves by 2 faces and keep the internal part.	Click here
REBAR LAYOUT		
Layout.DistributionRange	Defines the distribution ranges for rebar sets based on absolute or relative values.	Click here
Rebar.SetDistributionType	Toggle to set the distribution type of a rebar set to "Varying Length" (True) or "Uniform" (False)	Click here
Rebar.SetLayoutAsFixedNumber **Rebar.SetLayoutAsMaximumSpacing** **Rebar.SetLayoutAsMinimumClearSpacing** **Rebar.SetLayoutAsNumberWithSpacing** **Rebar.SetLayoutAsSingle**	Nodes to change the “Layout Rule” property of a Rebar in Revit.	Click here
REBAR QUERY		
Rebar.GetCenterlineCurve	Get the centerline curves from a selected set of rebars	Click here
Rebar.GetFullGeometryForView	Generates full geometry for the Rebar for a specific view.	/
Rebar.GetHost	Get the family instance and ID from the host of the selected rebar.	Almost every example
Rebar.GetProperties	Get the main properties of a selected array of rebars: Style, Bar Type, Hook Type, Hook Orientation.	Click here

RebarType.GetBarDiameter	Get the diameter of a given Rebar Bar Type	Click here
REBAR AUTHORING IN REVIT		
Rebar.SetSolidInView	Sets this rebar element to be shown solidly in a 3D view.	Click here
Rebar.SetUnobscuredInView	Sets this rebar element to be shown unobscured in a view.	Click here
Rebar.CopyByVector	Copy a selected Revit object (e.g. Rebar) and copy it according to a list of vectors.	Click here
VARIOUS		
Element.GetRebarCover	Get the rebar cover dimensions for a selected Revit element.	Click here

Dynamo for Rebar node list

Below a list of nodes from the package **Dynamo for Rebar v. 1.2.0** that are used and where they are explained in this handout.

Node	Description	Example
SURFACE EVALUATION		
Rebar.FollowingSurface	This node creates a set of curves following the geometry of a selected surface (most polysurfaces will also work). It divides the surface in one dimension - either U or V - regularly. You can define the number of divisions (or optionally, a distance to divide the surface by), and the direction of the curves.	Click here
Rebar.Morphed	This node creates a set of morphed curves between two border curves. It requires two curves to blend between, and creates either a fixed number of curves between them or divides by a defined distance.	Click here
Rebar.Perpendicular	This node creates a set of linear curves normal to a surface. It requires the selection of a driving surface and a set of bounding faces to define the end of the projection. According to a selected height, the node will divide the surface along this height into a selected number points. It will then draw lines along the normals at these points, break the line at any obstacle and continue until the bounding surfaces.	Click here

REBAR AUTHORING		
Rebar.Cut	The cut rebar node cuts a selected rebar container at a selected surface. The result will be either the left or the right side of the division.	Click here
Rebar.ByCurve	Creates one single bar element in Revit from a curve and a series of rebar properties.	Click here
RebarContainer.ByCurve	Creates a rebar container element from a list of curves and a series of rebar properties. The use of containers is highly encouraged as Revit can get bogged down by thousands of rebar family instances in your model. Containers are like groups of rebars in a single family instance.	Click here
CURVE EVALUATION		
Rebar.Shorten	This node shortens a selected curve from both ends by the same distance.	Click here

Learning Resources

More learning resources on the products that have been used in this class can be found below:

Dynamo	http://au.autodesk.com/au-online/classes-on-demand/search?full-text=dynamo http://dynamobim.com/learn/ http://dynamoprimer.com/ http://dictionary.dynamobim.com/ http://dynamobim.com/forums/forum/dyn/ https://www.lynda.com/Revit-tutorials/Dynamo-Essential-Training/455724-2.html http://www.revitforum.org/dynamo-bim/24005-dynamo-learning-resources.html#post136270
Dynamo for Rebar	Autodesk University 2015: MSF11845 – Dynam(o)ite Your Design from Concept to Fabrication TT Core Studio GitBook https://core-studio.gitbooks.io/dynamoforrebar/content/ https://github.com/tt-acm/DynamoForRebar
BIM4Struc.Rebar	http://revitbeyondbim.wordpress.com

Table of Figures