Project: Plasticity and Breaking in lumped modelled beams

Ludvig Wendelius*
Umeå University

Utvecklingsarbete i samverkan med näringslivet A, 4.5 ECTS

January 17, 2012

Examinator: Martin Servin†

*ludvig.wendelius@gmail.com
†martin.servin@physics.umu.se

Umeå University
January 17, 2012
Ludvig Wendelius

Sammanfattning

Företaget Algoryx håller på att utveckla en fysikmotor (AgX) som använder stelkroppar i sin simuleringsmiljö. Stelkroppar kan per definition inte deformeras. Detta leder till problem i modellering av matrial som under stora laster ska ge vika och deformeras. Resultatet av detta projekt är en model för att hantera *elastisk* och *plastisk* deformation i balkar diskretiserade av flera stelkropps segment. Modellen tar även upp hur brott kan hanteras.

Abstract

The company Algoryx Simulations are currently developing a physics engine (AgX) which uses rigid bodies in its simulation environment. Rigid bodies are by definition unable to deform. This leads to problems in modelling of structures that under large loads should give way and deform. The result of this project is a model that deals with *elastic* and *plastic* (permanent) deformation in beams by dividing the beam into small rigid body segments held together by torsional springs. The class can also handle *breaking* in these beams.

Contents

1 Introduction

The aim of this project was to find and implement a simple model from *solid mechanics* that explains the phenomenon of *elasticity*, *plasticity* and *breaking* in a lumped element model beam. To find this model a Literature study was performed.

A lumped element model is a body that has been discretized by multiple segments of rigid bodies held together by constraints. By setting the stiffness of these constraints you can make the body more flexible. This is known as *elastic* behaviour.

If you then manipulate the equilibrium positions of the constraints when they produce certain forces you can get permanent deformation to your lumped element body. This is phenomenon is called *plasticity* from solid mechanics. Another phenomenon that can occur under this criterion is *breaking*. Breaking can also happen after a certain amount of permanent deformation have been achieved.

This project was conducted in the course *Utvecklingsarbete i samverkan med näringslivet A, 4.5 ECTS* by the student Ludvig Wendelius. The student's assignment was to write the literature study and create a the model for the beams. The project was also conducted in collaboration with a developer from Algoryx, Tor Sterner. Who made the AgX implementation of the model that was tested. The Implemented model was tested by the student.

2 Method

Through out this report whenever you see something written in `courier` that means that it is a library, method or class related to Agx with Lua bindings.

Agx is a big and complex library. When you add an additional class to it, it is important to implement it so it fits the over all architecture of the software. In order for our `Beam` class to fit this structure we let it inherit from the class `agxModel.Tree`. `Tree` was already a class consisting of a user specified number of rigid bodies that was held together by constraints. Which is exactly what we wanted for our `Beam` class.

By inheriting from `Tree` we got access to a bunch of methods that allowed easy access to all of the constraints and bodies the beam is composed of. Some of these had to be re-implemented to fit the model as some new parameters were introduced. Then it was just matter of finding the correct way to parametrize the constraints to fit the plasticity and breaking models.

3 Theory

If you (the reader) are new to the subject of *solid mechanics* please read through the *Litterature Study* Appendix A before reading this section. It gives a short introduction to the subject that could be used to understand the rest of the report.

The proposed model for implementation is called *ideal plastic behaviour*. In this relation the stress σ is directly proportional to the strain ϵ up to the yield criterion. We define the yield criterion: "When the *stress* in the any part of a cross section becomes equal to a fixed value, known as the *yield stress* σ_{yield}, the beam will start to deform plastically."

The yield stress is the maximum stress a material can withstand without subjecting to permanent deformation. In *ideally plastic behaviour* stress can never be greater then then σ_{yield}. Increasing strain after the yield criterion is met will only result in permanent deformation.

In addition to the yield criterion, a *breaking* criterion was introduced and defined as: "the maximum permanent deformation ϵ_{break} the beam can withstand before breaking."

Figures 1 and 2 shows the stress-strain graphs for ideal plastic behaviour with and without breaking.

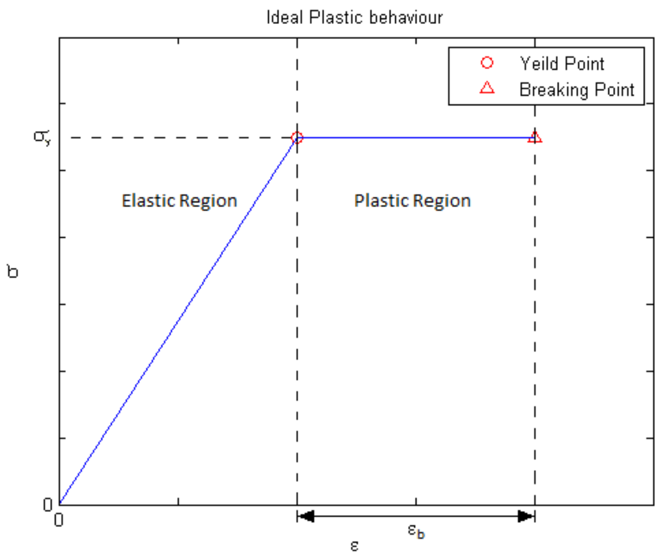

Figure 1: The stress-strain curve for *ideal plastic behaviour* with breaking.

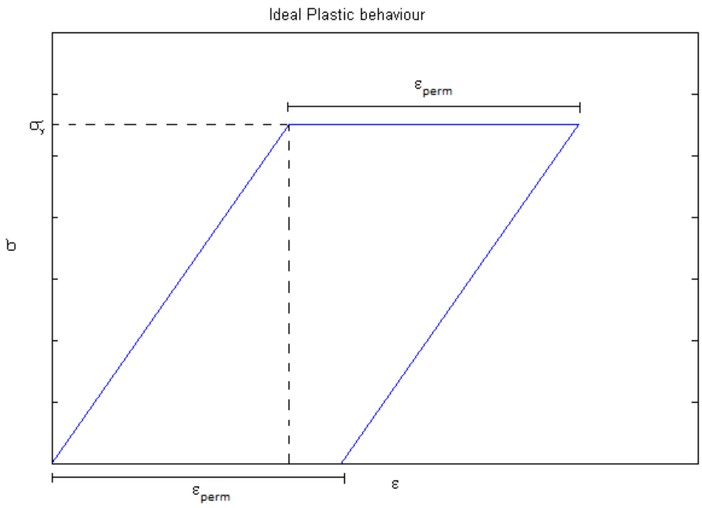

Figure 2: The stress-strain curve for *ideal plastic behaviour* with permanent deformation.

3.1 The Implemented Model

In the Agx class `agxModel.Beam` the beam is discretized over its long side by cutting the beam into smaller rigid body segments. These segments are held together by torsional 'springs' (`agx.LockJoints`). `agx.LockJoint` actually removes all translational degrees of freedom as well. But as these will not be manipulated we call them torsional springs.

In the implementation for the Agx class `agxModel.Beam` rigid body segments of equal length are held together by torsional 'springs' (`agx.LockJoints`). Figure 3 shows a beam that has been discretized by multiple rigid body segments. In Figure 4 we can see how the `LockJoints` are positioned and oriented in between the segments.

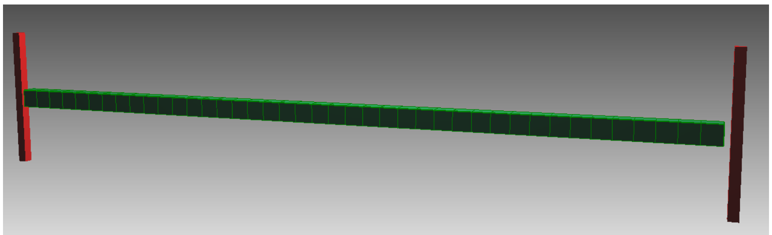

Figure 3: A Beam discretized by multiple segments of rigid bodies (outlined) held together by torsional springs. Springs not visible in figure.

The number of `LockJoints`, N_c (or segments $N = N_c + 1$) determines the resolution of the model. But the tension that allows the beam to bend is found on the cross section in between two segments. When in reality this

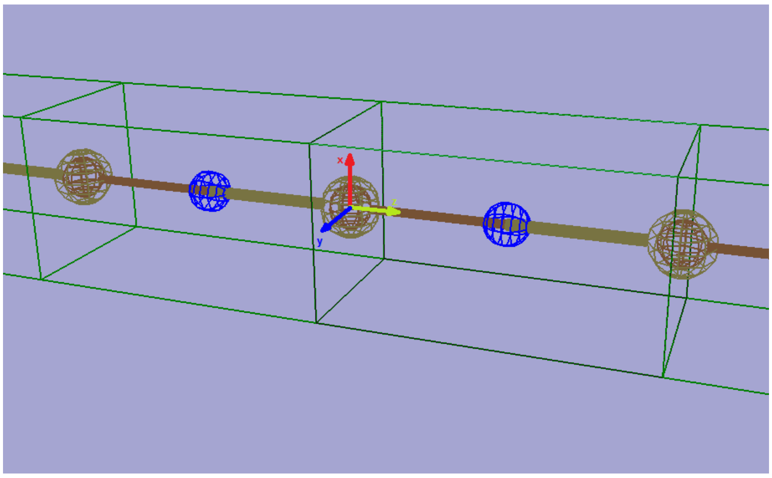

Figure 4: The position and orientation of a `LockJoint` in the `Beam` class.

would happen continuously over the entire beam. Therefore, the segment length will play a role in how the springs are set up. Introducing more segments will reduce the angularity of the beam making it more continuous.

When a discretized beam is subjected to some external force the springs between segments will respond with the torsional moments. These moments corresponds to the **bending moment** or **twisting torque** of that particular cross section! If the rotational degrees of freedom of the springs have the correct spring coefficients.

3.2 How to set up the constraints

Every `LockJoint` has three degrees of rotational freedom one around each unit axis, see Figure 4. Of course, in the case of a rectangular cross sections as seen in Figure 4 bending will occur around the x and y axes and twisting will be around the z axis.

How to set the coefficients for a `LockJoint` with two neighbouring rigid body segments with length L_a and L_b can be found in [1] to be:

$$c_{bending} = \frac{YI_A}{L_a + L_b} \tag{1}$$

$$c_{torsion} = \frac{YI_A}{2(L_a + L_b)1(1 + \sigma)} \tag{2}$$

Where Y is Young's modulus, σ is the Poisson's ratio of the material and I_A is the areas moment of inertia for a cross section along a central axis normalized by mass. Equation (1) can be a little bit misguiding as we can have bending around both the x and y axes (see Figure 4). If the beam extends further in x than in y direction then surely it must be more difficult to bend in around this axis. The difference lies in the value for I_A.

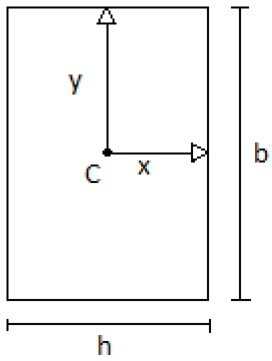

Figure 5: A rectangular cross section of a beam with width b and height h.

In the case of bending a beam with cross section like the one in Figure 5, the areas moment of inertia around the respective axes are

$$I_x = \frac{bh^3}{12}, \ I_y = \frac{b^3h}{12} \tag{3}$$

For twisting the value for I_A is the largest of I_x or I_y. The beam class will calculate the coefficients $c_{torsion}$ and $c_{bendign}$ (both of them) and set them in the `Lockjoints`, when the `setMaterial` method is called.

3.3 Dealing with Plasticity

When you use the `setMaterial` method on a `Beam` one of the parameters is the yield stress σ_{yield}. Given σ_{yield} you can easily calculate the maximum torque and bending moment that a cross section can give before turning plastic.

$$M_{x,max} = \sigma_{yield}\frac{I_x}{b} \tag{4}$$

$$M_{y,max} = \sigma_{yield}\frac{I_y}{h} \tag{5}$$

Or in the case of torsion the maximum allowed torque becomes:

$$T_{z,max} = \begin{cases} \tau_y \frac{b^2h^2}{3b+1.8h}, & b > h \\ \tau_y \frac{b^2h^2}{3h+1.8b}, & h > b \end{cases} \tag{6}$$

Where τ_y is the *shear yield stress*. In the case of torsion the stress will be parallel to the surface of the cross section unlike in bending where it perpendicular. For homogeneous materials $\tau_y = \sigma_y$ but for inhomogeneous like wood they wont be.

With Equations (4), (5) and (6) the yield criterion is now expressed by maximum moment and torque instead of stresses.

It is important to note that deformation is a local phenomenon and all constraints can and will deform independently of each other. Even though they effect each other with forces.

What `Beam` does when the twisting torque or bending moments exceed their respective maximum is **redefining** the equilibrium position for the constraints by some angle θ. The new equilibrium is set so that the torque/moment is below maximum, i.e. just below the yield criterion. We still have the maximum allowed torque/bending that the constraint allows in that direction but we also have a permanent deformation that can be seen when the external forces have been removed. Figure 6 show two beams after being permanently deformed by bending and torsion respectively.

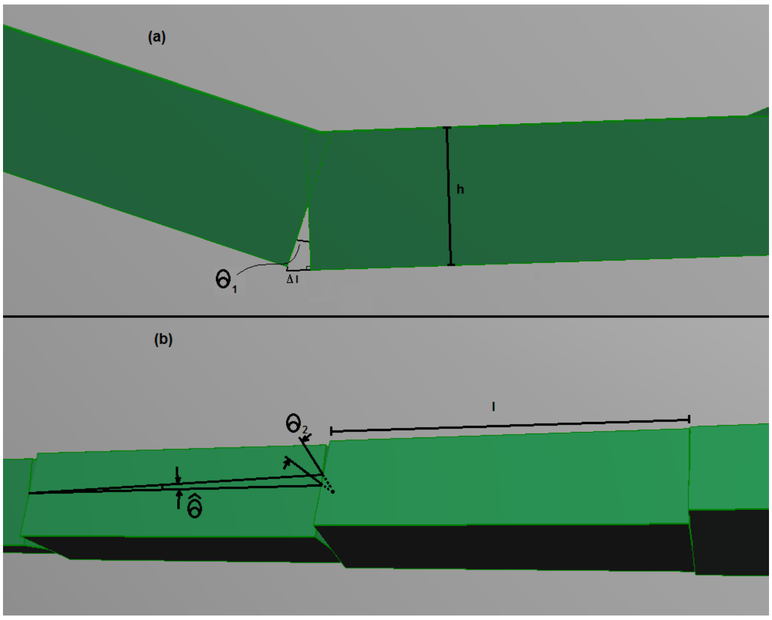

Figure 6: (a) A constraint in a beam that has been deformed by an angle θ_1 from bending. (b) A constraint in a beam that has been permanently deformed by an angle θ_2 from torsion.

3.4 Dealing with Breaking

The maximum permanent *strain* ε_{break} and *shear strain* γ_{break} that a material can withstand before failing (breaking) is used as the breaking criterion in the model. As The deformation that the constraints suffers are in angles. The mapping between these angles and the permanent strains can be calculated for bending as

$$\varepsilon_{max,x} = \frac{\sqrt{2\left(\frac{b}{2}\right)^2(1-\cos\theta_{1,x})}}{l}, \varepsilon_{max,y} = \frac{\sqrt{2\left(\frac{h}{2}\right)^2(1-\cos\theta_{1,y})}}{l} \tag{7}$$

where $\theta_{1,x/y}$ is the permanent deformation angle around x and y see Figure 6 and 5. For torsion we have

$$\gamma_{max} = \hat{\theta} = \frac{c/2 \cdot \theta_2}{l}. \tag{8}$$

where l is the segment length and c is the which ever of h and b that is the greatest (in the case of rectangular cross section). This also applies for a beam with circular cross section but with h, b and c being replaced with the diameter of the circle. Again for homogeneous materials we have $\varepsilon_{break} = \gamma_{break}$.

Unlike the case with plasticity which works passively, breaking might require some extra implementation for the user. This is because after breaking you have two shorter beams instead of one long. If you want to keep both beams and continue collecting data from then you must implement some method for doing this. You might also want to consider the problem if two or more constraint (springs) full fill the breaking criterion in during one time step.

The steps below explains the implementation a user needs to do in AgX to enable the breaking feature. **Note** that a constructed `Beam` will not break, as default.

1. Set the break limit values with the method `agxModel.Beam:setBreaklimit` which as a parameter takes the maximum permanent deformation (strain) in x, y and z that the constraint is allowed to receive before failure.

2. Construct an `agxModel.LuaBranchEventListener` and make it *listen* to the your `agxModel.Beam`. Using the `agxModel.Beam:setBranchEventListener` method.

3. Implement the method `agxModel.BranchEventListener:onHighLoad` that allows the user to collect all segments that have met the breaking criterion.

4. Then it is up to the user to decide how to deal with the segments, e.g. a `agxSDK.LuaStepEventListener` could be implemented to use the method `agxModel.Beam:cut` on all or just a few of them.

4 Results and Discussion

Below follows a few results that can be taken from the class `agxModel.Beam`. Of course one important result is the class itself. Were beams with circular and rectangular cross sections can be constructed. Remember that the height and width (or radius) should be much smaller than the length of the beam.

4.1 Independent from number of Segments

The model is totally independent from the number of segments that the user chooses to discretize the beam with. To see of this hold a set up with beams lying at rest on two pillars with only the gravity and normal forces present, see Figure 7. We then consider the cases of a two beams one with

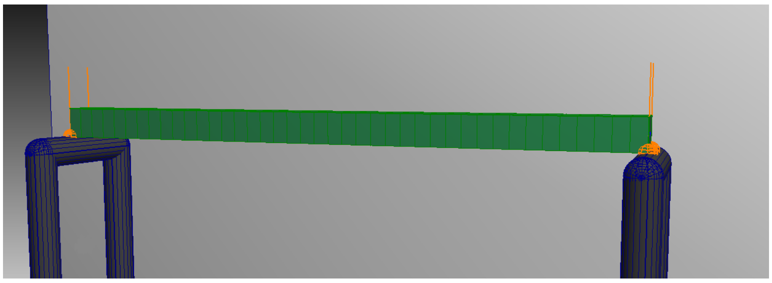

Figure 7: Set up of a beam at rest with support only on the ends.

rectangular and one with circular cross section with different number of constraints $N_c = 20, 30, 50, 100$. Then plotting the moments that each of these constraints gives under every time step results in Figure 8.

From Figure 8 we see that the model is indeed independent with respect to the number of segments a user introduces. But during testing it was found that the segment length L should atleast be smaller then both the height h and width b of the beams cross section. It is just a matter of finding a resolution of your beam that makes it look nice.

The class should also be resolution independent with respect to breaking. As the strain is calculated using the segment length. See Section 3.4.

4.2 Ideal Plastic Behaviour

The material parameters for steel was used in a beam with quadratic cross section. The beam was subjected to a twisting torque in on of the ends and the following stress-strain curve was produced, Figure 10.

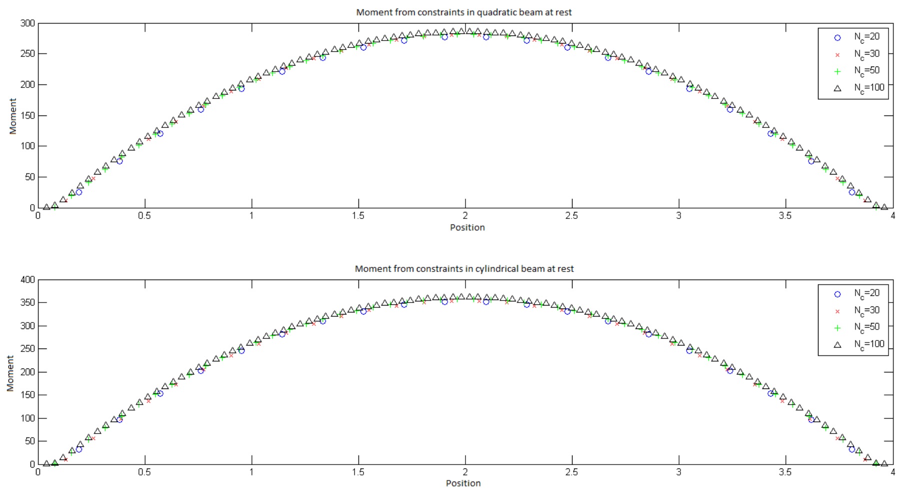

Figure 8: The moments given by the constraints in a four meter quadratic and cylindrical beam at rest.

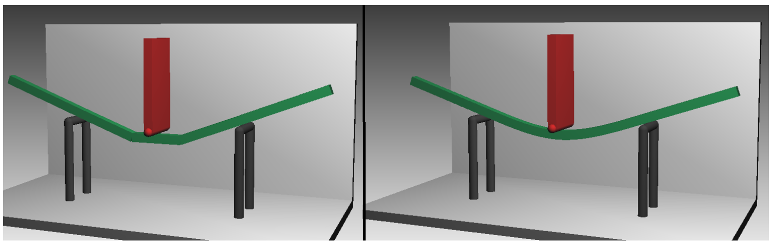

Figure 9: To the left is a beam with poor resolution and to the right is a beam with better resolution.

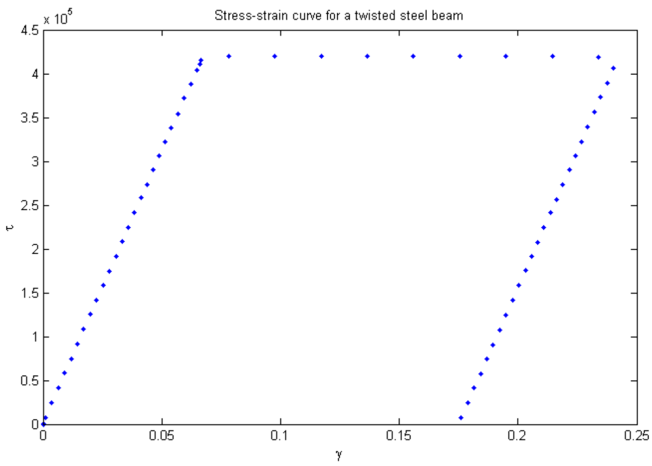

Figure 10: The shear stress-strain for a twisted steel beam.

By comparing Figures 10 and 2 you find that the implemented model follows the same stress-strain curve as the *ideal plastic behaviour* approximation.

As the constraints are set up in the same way for twisting and bending the result will be the same if a figure similar to 10 would be made for bending.

4.3 Breaking

The class `agxModel.Beam` can handle the phenomenon of breaking as explained above. However, when trying to find known values for the maximum allowed strain for some material the search came up empty! So there will be no other result for the breaking other than the fact that the class can handle it.

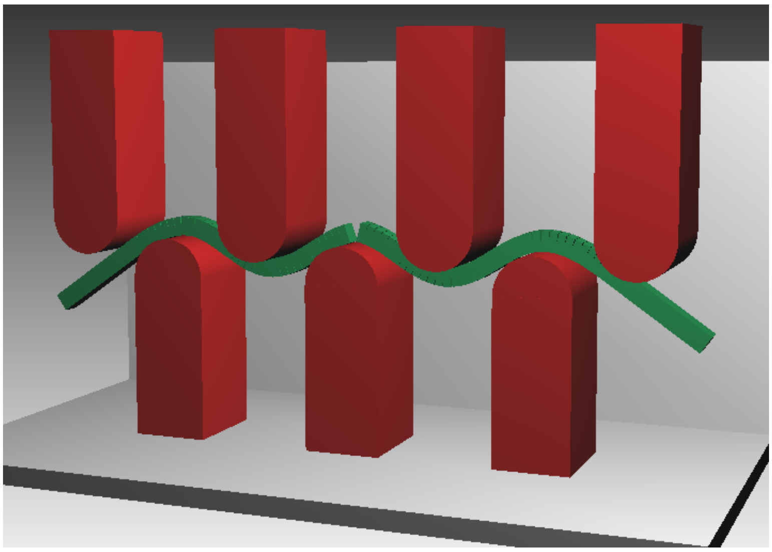

Figure 11: A beam that has broken after deformation.

The most difficult part of this project was to understand the breaking phenomenon and find the correct mapping between the angles in the constraints to the strain in the material. This left little time to produce any results this phenomenon.

As it stands above searching for values for the maximum strain before breaking has come up empty. However you can find values for a stress at wich a material breaks. This stress is larger then the yield stress though so when using the *ideal plastic behaviour* model this *breaking stress* will never be reached. This may suggest that you may want to change the model in the future, to allow for some increasing stress and some permanent deformation after the yield stress is met.

5 Validation Proposals

1. The first validation proposal that comes to mid is to compare results from simulations with the `Beam` class and some empirical studies. There are a lot data on various alloys and getting the values for yield stress, Young's modulus and poission ratio is quite easy. However finding values for the breaking strain is more difficult. Actually no value for the breaking strain was found while performing this project. Usually the breaking criterion is explained another stress value ($> \sigma_{yield}$), see SECTION 4.3.

2. A good way to validate the model could be to use known equations from elementary bending cases and compare these to what you get from using `Beam` in simulations ([4] p.384).

3. The slope in Figures 1 or 2 could also be checked that they follow what you expect from *Hooke's law* for some well known material.

4. Something that should be validated as well is what happens when a beam is subjected to bending around the x and y (see Figure 14)axis at the same time. Will the beam still yield when it is supposed to or will it become more resilient? This should be looked at for both cylindrical and rectangular cross sections.

5. Similar to the item above a beam could be subjected to both bending and twisting at the same time and the model should be validated to hold for this, i.e. that the beam *yields* when it is supposed to. This could pose an even bigger problem if the beam is made from a non-homogeneous material where the yield stress in shear and normal is not the same.

6. When it comes to breaking the beam will not break unless the permanent deformation surpasses a certain *angle*. Which means that you can bend the beam as much as you want in any direction as long as you do not exceed this limit and then bend it back to its original position. This process can be repeated infinite amount of times right now. To solve this it might be good to have some memory variable in each constraint that keeps track of how much the constraint has been permanently deformed in total.

A Literature study

This study war performed so that the members of the Plasticity Project could gain enough knowledge about bending and twisting to find a simple but good model to implement into Agx. It also serves as an introduction for the reader if she/he is new to the subject. The outlining in this study will begin by explaining the different type of deformations for a beam and then move of to the physics of the deformations that are relevant for the project (bending and twisting).

There are three types of deformations that can occur in a beam. These are *stretching* (tangential deformations), *bending* (curvature deformations) and *torsion* (twisting deformations). Any type of arbitrary deformations can be expressed as a combination of these three. See Figure 12 (figure borrowed from [1]) for an illustration of the different deformations.

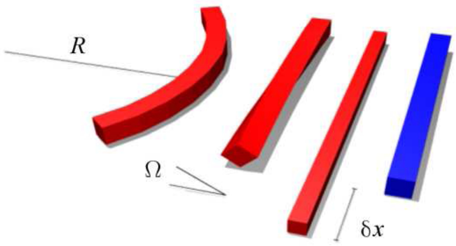

Figure 12: The three types of deformations for a beam: bending with a radius of curvature R, twisting by an angle Ω and stretching by a length δx.

A limitation in this project is that only the cases of twisting and bending are considered.

Bending

When a beam gets bent it means that the material on the inside of the curve becomes compressed and the material on the other side becomes stretched. Because of this there must exist some surface that goes along the length of the beam that neither stretches or compresses. This surface is called the *neutral* surface. For small bending in simple beams this surface will go through the center of gravity of the cross section. This is only true if the beam is not subjected to stretching or compressing which is called *pure* bending [2].

Figure 13 shows a beam subjected to *pure* bending. Observing a thin segment of the beam, Figure 13(b) shows that the material below and above

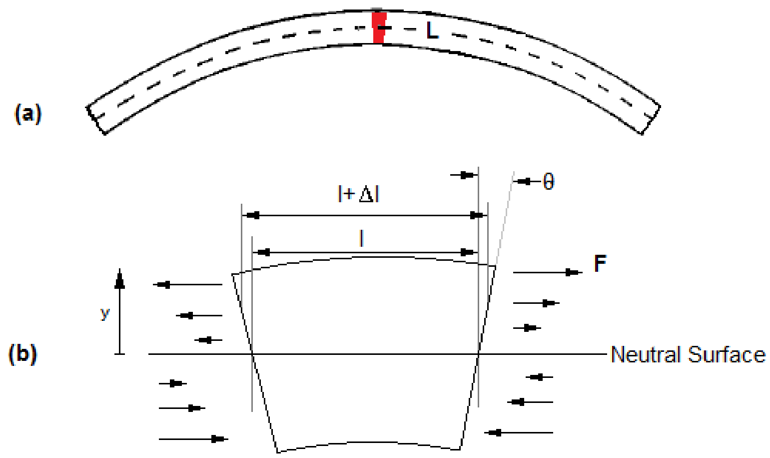

Figure 13: A beam with rest length L is bent the marked area in **a)** is a thin segment with length l. This segment is zoomed-in in **b)**. The compressional and stretching forces of the segment are also shown.

the neutral surface suffers compressional or stretching *strains* respectively. Both these strains are proportional to the distance from the neutral surface y. I.e. the longitudinal stretch Δl is proportional to the height y. The *strain* ε is defined as the ratio between the stretch and rest length.

$$\varepsilon = \frac{\Delta l}{l} \tag{9}$$

An other important physical unit in this scenario is the *stress* σ which is defined as the strain times the Young's modulus of the material Y.

$$\sigma = Y\varepsilon \tag{10}$$

The forces that produces the strain are shown in Figure 13(b). If we take a look at some arbitrary cross section of the beam the forces acting upon have different directions above and below the neutral surface. These forces give rise to a bending moment M about the neutral surface (bending moment is basicly equivalent to torque).

The stress distribution for a cross section like the one in Figure 14 can be calculated with the help of the bending moment as

$$\sigma = M_x \frac{y}{I_x} \text{ where } I_x = \frac{bh^3}{12}. \tag{11}$$

Where I_x is the areas moment of inertia around x. In the same way, if the bending is around the y axis instead we have

$$\sigma = M_y \frac{x}{I_y} \text{ where } I_y = \frac{b^3 h}{12}. \tag{12}$$

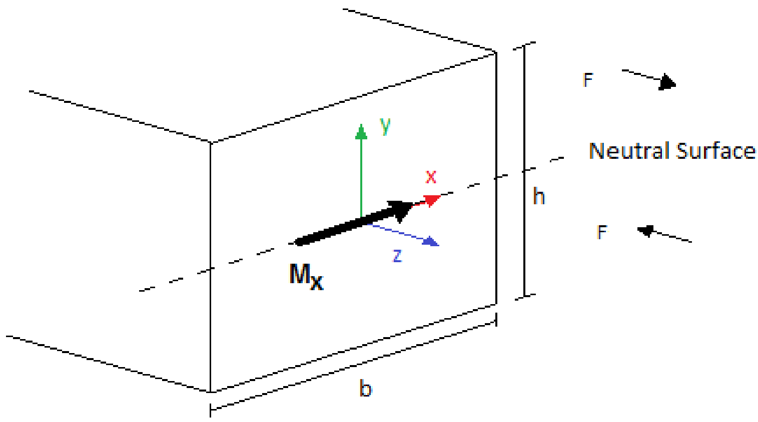

Figure 14: An arbitrary cross section for a rectangular beam. With bending moment M_x around the neutral axis corresponding to the force directions from Figure 13(b).

The reasoning above holds for any cross sections under the assumption of pure bending. As long as you use the correct area moment of inertia.

E.g. when bending a circular beam we get the following (using cylindrical coordinates).

$$\sigma = M_{\perp r} \frac{r}{I_{\perp r}} \text{ where } I_{\perp r} = \frac{\pi R^4}{4}. \tag{13}$$

Beam modelled by lumped elements

In Equation (9) Δl can be expressed as either a function of the distance from the neutral axis y or the angle θ. When discretizing a beam using rigid bodies segments the calculation of strain is has to be done with the angles as that is what you can get from the constraint linking two segments together. What we want to do is to adapt what we see in Figure 13b) to the lumped element model. Looking at Figure 13b) we see the angle θ appears twice. This could imply that we want to use the angle from two constraints when calculating the strain for one segment.

The problem with this is that the constraints will have to be paired, which is not what you want. Instead the small segment in Figure 13b) can be seen as one constraint and half of the two adjacent rigid body segments that the constraint connects. See Figure 15.

Then you can express the maximum stretch in terms of the segment height h and θ by using the law of cosines.

$$\Delta l^2 = 2\left(\frac{h}{2}\right)^2 (1 - \cos\theta) \tag{14}$$

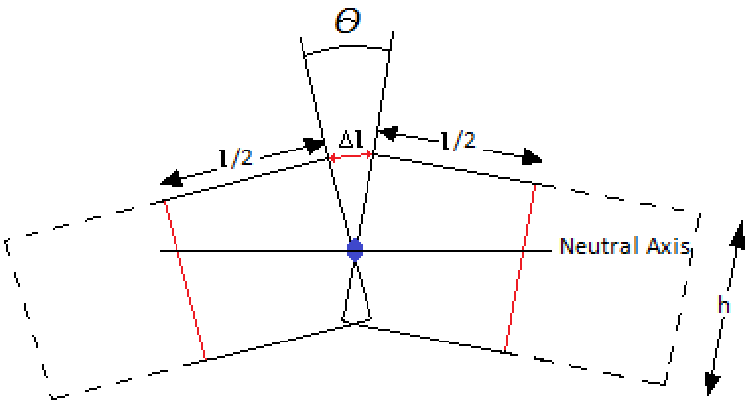

Figure 15: Figure 13b) translated into the case when a beam is discretized with multiple rigid body segments. **Note** the dashed part of the rigid body segments are not part of what we are looking at.

We get the maximum strain on the beam in as a function

$$\varepsilon_{max} = \frac{\sqrt{2\left(\frac{h}{2}\right)^2(1-\cos\theta)}}{l} \tag{15}$$

Note that the end segments of the beam might need to be half as long as the others. Or you just divide those strain equations by $3l/2$ instead of just l.

Twisting

In the case of bending (explained above) the stresses are orthogonal to the surface cross section. But if the beam is twisted by an angle Ω, like in Figure 12, the stresses will be parallel to the same surface. This is called *shear stress* τ. The *shear strain* γ thin circular cylinder like the one in

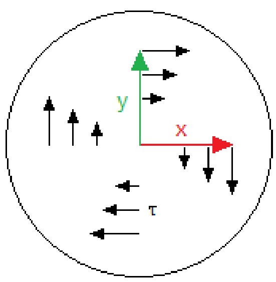

Figure 16: The stress distribution in a cross section for a cylindrical beam.

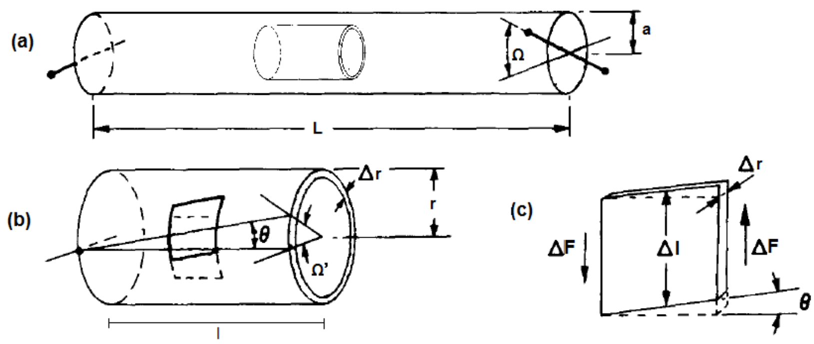

Figure 17: (a) A cylindrical beam, length L, radii a twisted an angle Ω. (b) A small shell with radii $r < a$ from the beam. (c) Shows a small piece from the shell in (b).

Figure 17 can be found to be [2]:

$$\gamma = \theta = \frac{r\Omega'}{l}. \tag{16}$$

The shear stress can be calculated from the shear strain γ, Young's modulus Y and the Poissons ratio σ[4].

$$\tau = \gamma\frac{Y}{2(1+\sigma)} \tag{17}$$

Or as a function of the applied torque T at one of the ends in a cylindrical beam and the distance r from the centre of the cross section [3]

$$\tau = \frac{Tr}{J} \text{ where } J = \frac{\pi R^4}{2} \tag{18}$$

J being the beams *polar moment of inertia* and R is the radius of the cross section.

For a quadratic cross section with height h and width b the equation becomes a bit more odd and we only calculate the maximum stress that the cross section is subjected to. Which should be somewhere on its boundary.

$$\tau_{max} = \begin{cases} T\frac{3h+1.8b}{h^2b^2} & \text{if } h > b \\ T\frac{3b+1.8h}{h^2b^2} & \text{if } h < b \end{cases} \tag{19}$$

Yielding

In engineering the *yeild stress* σ_{yield} refers to the maximum amount of stress that a material can withstand before being *plastically* deformed. For stresses below σ_{yield} the material will deform *elastically*, i.e. the material will be able to return to its original form after the stresses have been removed. Once the stress surpasses this value some fraction of the sustained deformation (strain) becomes permanent.

For *inhomogeneous* materials (e.g. wood) the yield stress will be different when being subjected to *shear stresses*(e.g. from twisting) instead of normal stresses (e.g. from bending). This *yield shear stress* is denoted by τ_{yield} and $\tau_{yield} = \sigma_{yield}$ only applies to *homogeneous* materials.

What actually happens to the material in when it is inside the plastic region is dependent on the material. There is no general case that applies to all materials. There are also some phenomenons that takes place in the plastic region that applies to some materials. Like *Hardening* in metals. Unfortunately these will not be given any explanation here since the goal of the project is to find some simple model that works more generally.

As discussed above the stress increases with the distance from the *neutral surface* of the cross section. It is the largest value for the stress that has to be taken into account when checking if the material *yields*. This is because yielding is a local phenomenon and even though a material subjected to e.g. bending might be fine close to the *neutral surface* it may be plastically deformed close the boundary.

When calculating on a beam to see if it should suffer permanent deformation you should calculate the stress where it is highest. From Equations 11, 12, 13, 18 and 19 we get the following maximum stress and shear stress for a rectangular cross section to be

$$\sigma_{max} = M_y \frac{b}{2I_y} \tag{20}$$

$$\sigma_{max} = M_x \frac{h}{2I_x} \tag{21}$$

$$\tau_{max} = \begin{cases} T\frac{3h+1.8b}{h^2b^2} & \text{if } h > b \\ T\frac{3b+1.8h}{h^2b^2} & \text{if } h < b \end{cases}$$

and for a circular cross section

$$\sigma_{max} = M_{\perp r} \frac{R}{I_{\perp r}} \tag{22}$$

$$\tau_{max} = \frac{TR}{J} \tag{23}$$

Values for σ_{yield} and τ_{yield} can be looked up in various tables, e.g. [4].

Breaking

Just like every material has a maximum stress it can take before yielding it also has a maximum amount permanent deformation or permanent *strain* it can receive before it fails. Just like in the case of yielding the value for the maximum *strain* is different for shear γ_{break} and normal ε_{break} deformations.

When calculating on a beam to check if it should break you need to look at the part where the strain is at its maximum values. In Equations 9 and 16 this will we at the boundary of the beams. I.e.

$$\varepsilon_{max} = \frac{\Delta l\big|_{y_{max}}}{l} = \frac{\sqrt{2\left(\frac{h}{2}\right)^2(1-\cos\theta)}}{l} \tag{24}$$

$$\gamma_{max} = \frac{a\Omega'}{l}. \tag{25}$$

References

[1] Martin Servin, Claude Lacoursière, *Rigid Body Cable for Virtual Environments*. June 13, 2007. 5, I

[2] Richard P. Feynman, *Feynman Lectures on Physics*. Vol 2, Ch 38 - Elasticity. June, 1963. I, V

[3] Dr. D. B. Wallace, *Basic Stress Equations*. http://www.eng.uah.edu/~wallace/mae466/DOC/bas_str.pdf. June, 1963. V

[4] Carl Nordling, Jonny Österman, *Physics Handbook: for Science and Engineering*. Studentlitteratur, 8th edition, 2006. 12, V, VI

[5] Wikipedia, *Yield (Engineering)*. http://en.wikipedia.org/wiki/Yield_%28engineering%29