

# Introduction

Horizontal wells are one of the most effective technologies to emerge in the petroleum industry, and one of the cornerstones of modern oil and gas production. Offering increased reservoir draining and access, reduced water and gas coning, and a reduced footprint (Bosio & Economides, 2011). Horizontal wells are an evolution of the simple vertical wells, and the upgraded directional wells that came later.

It was not until the 1970s and 1980s that economic feasibility and technological advancements led to widespread adoption of horizontal wells (Gallagher, 2022). A major breakthrough came with the Rospo Mare 6 well in Italy in 1982. The use of horizontal drilling, although at 2.1 times the cost of a simpler alternative, resulted in 20 times higher oil production, demonstrating the technique's potential (Bosio & Economides, 2011). Today, approximately 81% of wells are drilled directionally, reflecting the industry's shift towards more efficient and targeted reservoir access (Gallagher, 2022). There are also statistics that show that of the wells that have the highest productivity, a majority are horizontal wells, with the proportion rapidly increasing.

Despite their advantages, horizontal wells introduce new challenges, particularly regarding well intervention. Well intervention refers to any operation carried out on a well during its productive life to improve performance, monitor, repair damage, and extend its operational lifespan. (Vedvik, 2025)

Unlike vertical wells, where tools can simply be lowered, horizontal wells require complex intervention tools to overcome additional frictional forces, bending stresses, and potential buckling. Traditional intervention methods, such as slicklines, use rollers to overcome the accumulated frictional force. This report explores the design of a floating composite intervention rod. As a substitute for coiled tubing or slickline for well intervention.

The study will investigate a design approach, analyzing material properties, mechanical stability, and buoyancy control. Various failure criteria will be applied to assess performance, guiding the development of an optimized intervention solution for horizontal well applications.

## Theory

### Failure prediction

Understanding failure mechanisms is essential when working with materials subjected to any kind of loading. If and when failure occurs, it can have devastating effects on structures and human life. It is therefore crucial to have models that predict failure. One can say that it is even more crucial in the context of lightweight design where a material's capabilities are already pushed further towards the limit.

To aid in failure prediction, multiple failure criterions have been developed. A failure criterion is often a set of rules, functions, or expressions of logic that gives a clue to whether failure is bound to happen or not (Vedvik, 2025a). This is commonly a function that returns a value, normally 1, to indicate failure. Anything below this set value is not considered material failure. The function takes stress-state/loading and strength properties as parameters.

$$f(\sigma, \Psi) = 1 \Rightarrow \text{failure} \quad (1)$$

To provide a more understandable and useful to humans explanation of how stresses cause failure, the Load Proportionality Factor (LPF) is introduced. Since the failure function alone has no proportionality, meaning that if it comes out to 0.5 it does not mean it is safe to double the load. Here the LPF comes in useful, which tells us how much we can increase the loads without causing failure (Vedvik, 2024).

Loading conditions are rarely as simple as only one stress component existing while all others are zero. In real life the forces and loads are applied in different directions, and with multiple stress components present we need a way of describing how they interact. There are therefore developed criterias for different occasions, some of them described here.

## Maximum Stress Criterion

The Maximum Stress Criterion is one of the simplest failure criterions used for composite materials. It predicts failure when any of the stress components in a material exceeds its respective strength limit (Vedvik, 2024). In the case of composites, these limits include tensile and compressive strengths in the fiber and matrix directions, as well as shear strength. This approach is particularly useful for composite materials such as carbon/epoxy, where strength values in different directions vary significantly due to the orthotropic nature of the material.

For a three-dimensional stress state, the Maximum Stress Criterion evaluates six conditions:

$$-X_C < \sigma_1 < X_T \quad -Y_C < \sigma_2 < Y_T \quad -Z_C < \sigma_3 < Z_T \quad |\tau_{12}| < S_{12} \quad |\tau_{13}| < S_{13} \quad |\tau_{23}| < S_{23}$$

- $X_T, Y_T, Z_T$  are the tensile strengths in the principal material directions.
- $X_C, Y_C, Z_C$  are the compressive strengths in the principal material directions.
- $S_{12}, S_{13}, S_{23}$  are the shear strengths in their respective planes.

Failure is predicted when any of these conditions are exceeded. The criterion can also be expressed using a max function as seen in equation (2).

$$f = \max\left(-\frac{\sigma_1}{X_C}, \frac{\sigma_1}{X_T}, -\frac{\sigma_2}{Y_C}, \frac{\sigma_2}{Y_T}, -\frac{\sigma_3}{Z_C}, \frac{\sigma_3}{Z_T}, \frac{|\tau_{12}|}{S_{12}}, \frac{|\tau_{13}|}{S_{13}}, \frac{|\tau_{23}|}{S_{23}}\right) \quad (2)$$

Failure occurs when  $f \geq 1$

## Tsai-Wu

The Tsai-Wu failure criterion was developed at NASA, intended to be a versatile failure method mainly for unidirectional composites under complex loading conditions. The Tsai-Wu failure criterion improves on simpler criterions such as the maximum stress criterion, incorporating both quadratic and linear terms. Making it behave closer to the real life behavior of the material it is applied to. Tsai-Wu is widely utilized in aerospace, and other high performance applications. (Experts Of CAE Assistant Group, 2024)

The Tsai-Wu failure criterion, failure is predicted when  $f$  is greater than or equal to 1, see (3).

$$f = F_1 \sigma_1 + F_2 \sigma_2 + F_3 \sigma_3 + F_{11} \sigma_1^2 + F_{22} \sigma_2^2 + F_{33} \sigma_3^2 + F_{44} \tau_{23}^2 + F_{55} \tau_{13}^2 + F_{66} \tau_{12}^2 + 2F_{23} \sigma_2 \sigma_3 + 2F_{13} \sigma_1 \sigma_3 + 2F_{12} \sigma_1 \sigma_2 \quad (3)$$

With the strength parameters defines as follows:

$$F_1 = \frac{1}{X_T} - \frac{1}{X_c}, \quad F_2 = \frac{1}{Y_T} - \frac{1}{Y_c}, \quad F_3 = \frac{1}{Z_T} - \frac{1}{Z_c} \quad (4)$$

$$F_{11} = \frac{1}{X_T X_c}, \quad F_{22} = \frac{1}{Y_T Y_c}, \quad F_{33} = \frac{1}{Z_T Z_c} \quad (5)$$

$$F_{44} = \frac{1}{S_{23}^2}, \quad F_{55} = \frac{1}{S_{13}^2}, \quad F_{66} = \frac{1}{S_{12}^2} \quad (6)$$

$$F_{23} = f_{23} \sqrt{F_{22} F_{23}}, \quad F_{13} = f_{13} \sqrt{F_{11} F_{33}}, \quad F_{12} = f_{12} \sqrt{F_{11} F_{22}} \quad (7)$$

Experimentally determined coefficients:  $f_{12}$ ,  $f_{13}$ ,  $f_{23}$  (Vedvik, 2024b)

## Hashin criteria

The Hashin failure criteria was developed by Zvi Hashin, a mechanical engineer well respected for his groundbreaking contributions to accurate analysis of composite materials (Hashin, 2012). He proposed a criteria that, unlike the others previously presented, took into account the different failure modes of composites. Composite materials fail through various mechanisms due to their orthotropic nature, making

prediction more complex. The Hashin theory is designed to predict what type of damage that occurs under loading and thereby provides a detailed prediction of where the failure will occur first (Veidth, 2024). The two modes that are distinguished are failure in the fibers (FF) and failure in the matrix (IFF).

Failure in fiber and matrix are again divided into tensile and compressive loading, resulting in four individual relations to describe the failure mechanism.

- Fiber failure in tension

$$\left(\frac{\sigma_1}{X_T}\right)^2 + \frac{1}{S_{12}^2} (\tau_{12}^2 + \tau_{13}^2) = 1 \quad (8)$$

- Fiber failure in compression

$$-\frac{\sigma_1}{X_c} = 1 \quad (9)$$

- Matrix failure in tension

$$\frac{1}{Y_T^2} (\sigma_2 + \sigma_3)^2 + \frac{1}{S_{23}^2} (\tau_{23}^2 - \sigma_2 \sigma_3) + \frac{1}{S_{12}^2} (\tau_{12}^2 + \tau_{13}^2) = 1 \quad (10)$$

- Matrix failure in compression

$$\frac{1}{4S_{23}^2} (\sigma_2 + \sigma_3)^2 + \frac{1}{S_{23}^2} (\tau_{23}^2 - \sigma_2 \sigma_3) + \frac{1}{S_{12}^2} (\tau_{12}^2 + \tau_{13}^2) + \frac{1}{Y_c} \left( \frac{Y_c^2}{4S_{23}^2} - 1 \right) (\sigma_2 + \sigma_3) = 1 \quad (11)$$

(Vedvik, 2024b)

## Design Approach

Based on traditional horizontal well structures, the main failure mode is widely regarded as buckling from the axial load to press the pipe in the well (Vedvik, 2025). To minimize the risk of buckling, maximizing the 2nd moment of area was desired. To further reduce the failure mode of buckling, the requirement of neutral buoyancy was introduced.

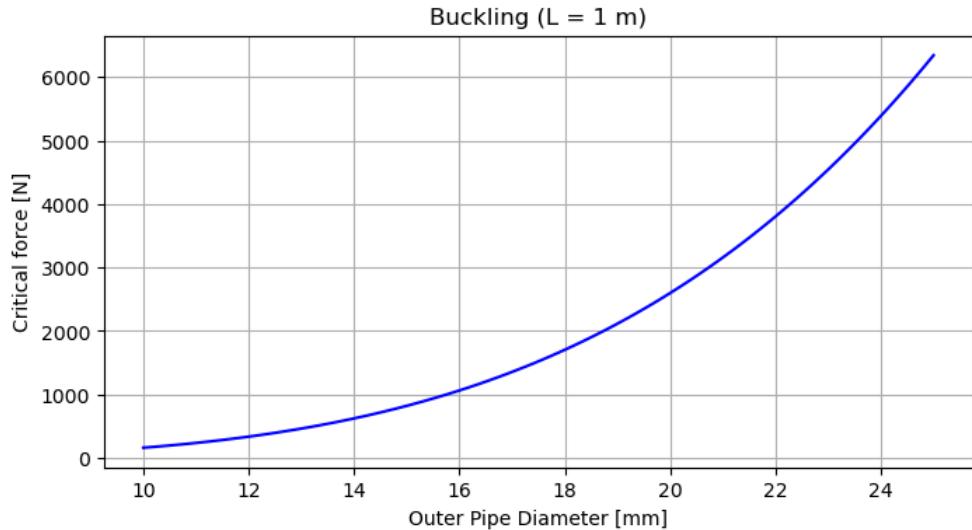
Deriving the following relationship for neutral buoyancy, assuming the density of the fluid inside to be equal to that of water, where  $\rho_{cf}$  is the density of the carbon fiber material.  $D$  and  $d$  are the outer and inner diameters of the pipe, respectively.

$$\frac{d}{D} = \sqrt{1 - \frac{\rho_{water}}{\rho_{cf}}} \quad (12)$$

Using the euler buckling equation (13), with the diameters calculated from eq (12), the critical buckling load  $P_{cr}$  was plotted in Figure 1.

$$P_{cr} = \frac{\pi^2 EI}{L_k^2} \quad (13)$$

Where  $E$  is the young's modulus of the material,  $I$  is the 2nd moment of area and  $L_k$  is the euler buckling length, which in this case was set to  $L_k = L$ .



*Figure 1, Critical force for buckling for different pipe dimensions.*

The maximum outer diameter  $D$  was set to 25mm, with the inner diameter calculated from eq (12). The shell thickness ratio was then calculated to 0.39, resulting in the plane stress assumption being too inaccurate based on the rule of thumb. The Finite Element Method was chosen to model the pipe to achieve the desired through-thickness accuracy. The stresses were then imported to a jupyter notebook file, where different failure modes were compared.

## Abaqus Modeling Documentation

### Model Dimensions

- Outer pipe radius = 12.50000 mm
- Inner pipe radius = 7.65466 mm
- Width = 2.00000 mm

### Boundary conditions (Figure 2)

- Bottom longitudinal edge  $U_2 = 0$ .
- Surfaces in the  $xy$ -plane  $U_3 = 0$ .

- One side of the cross section  $U_1 = 0$ , while the other side was constrained to uniform deformation in the x-direction.

### Material Properties

- Defined as elastic

E1	E2	E3	Nu12	Nu13	Nu23	G12	G13	G23
130000	10000	10000	0.28	0.28	0.5	4500	4500	3500

*Table 1, material properties of Carbon/Epoxy(a)*

### Directional properties

As shown in Table 1, the 1-direction is defined as the fiber direction. Local directional properties are shown in Figure 3. The outer layer has fibers aligned with the pipe's length, while the inner layer has fibers oriented in the hoop direction.

### Mesh

- Global mesh size = 0.5
- C3D8R elements

### Loads

- Distributed load of 100 MPa on the outer surface.
- Point load of 24543.69 N coupled to the cross section.

### Extracting Results

The normal stresses (using their local coordinate system as shown in Figure 3) were extracted from a path along an edge in the radial direction. Shear stresses were in the order of  $10^{-5}$  and neglected.

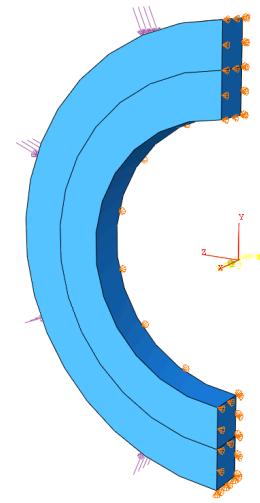


Figure 2, Loads and constraints.

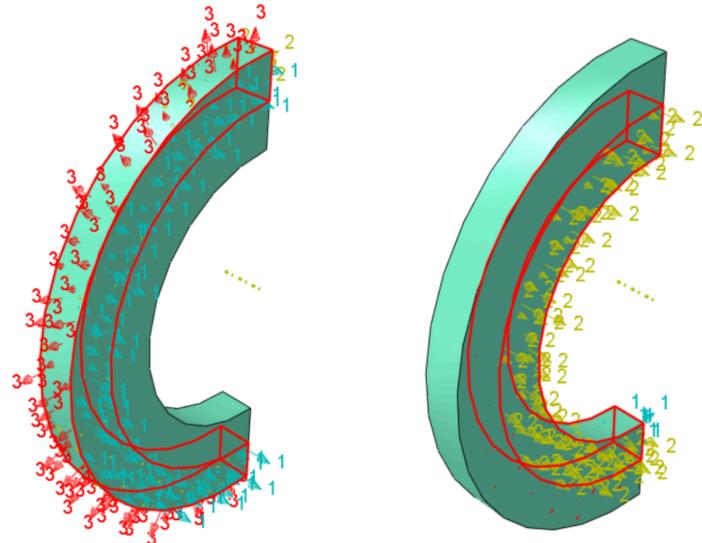


Figure 3, Fiber directions of inner and outer layer.

## Results

The specifics of the material properties of Carbon/Epoxy(a), as well as interaction parameters for the Tsai-Wu criterion, is provided in the jupyter notebook file (see Appendix).

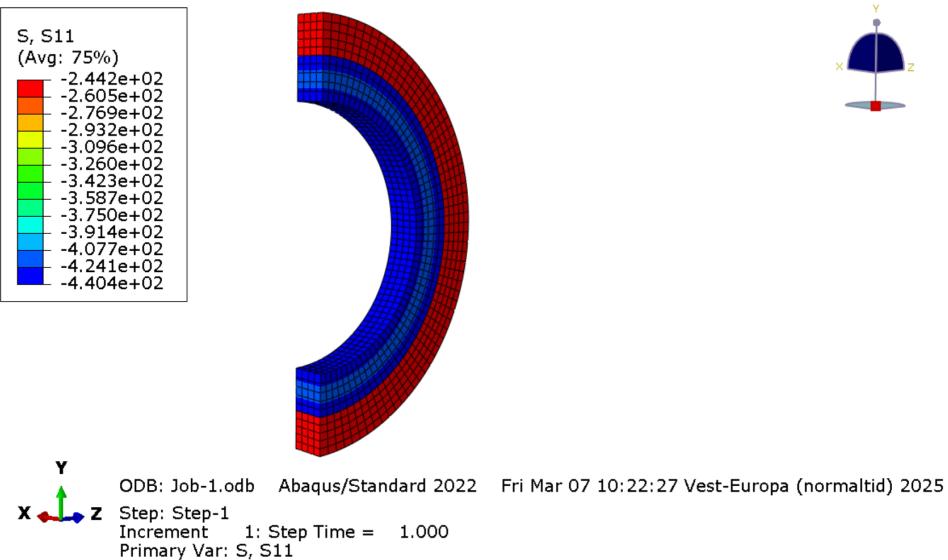


Figure 4, Normal stresses in 1-direction.

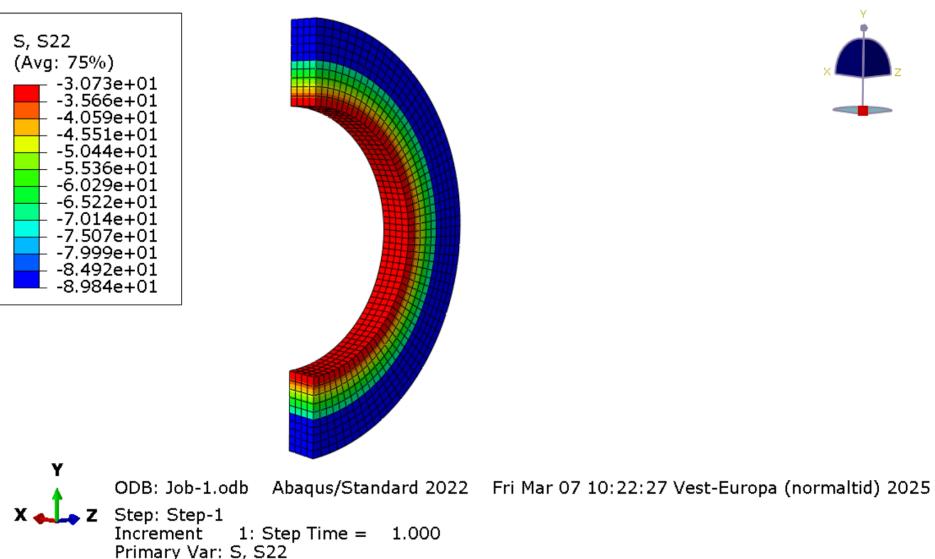


Figure 5, Normal stresses in 2-direction.

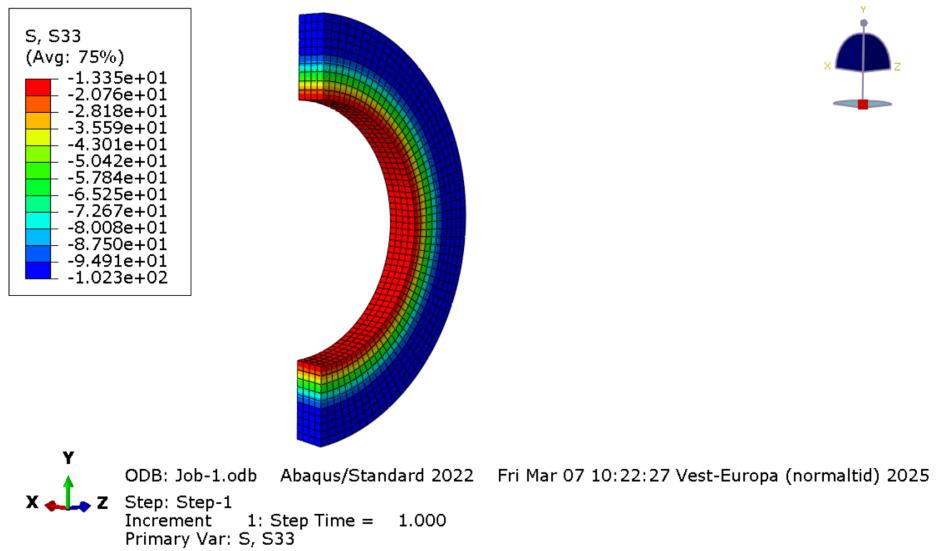
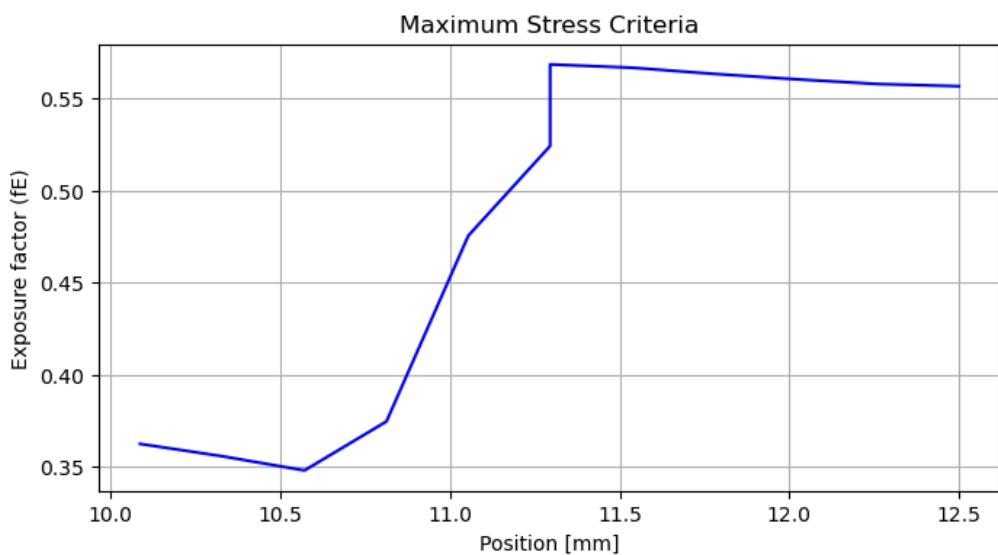


Figure 6, Normal stresses in 3-direction.

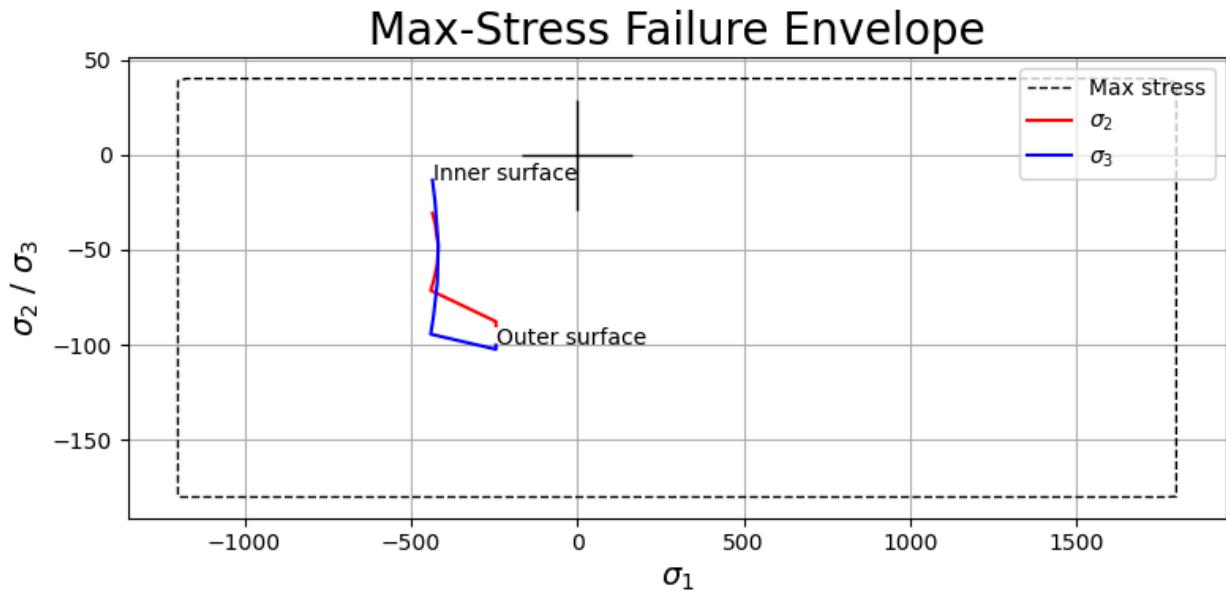
**Maximum stress criterion**

<i>Position (mm)</i>	<i>Exposure factor</i>
----------------------	------------------------

<i>Position (mm)</i>	<i>Exposure factor</i>
12.50	0.55667
12.26	0.55788
12.02	0.56046
11.78	0.56338
11.54	0.56672
11.29	0.56850
11.29	0.52423
11.05	0.47556
10.81	0.37471
10.57	0.34814
10.33	0.35584
10.09	0.36256



*Figure 7, Maximum Stress exposure factors for various positions in the pipe wall*



*Figure 8, Max stress and max strain envelopes for various positions in the pipe wall.  
Red represents 2-direction and blue represents 3-direction.*

#### **Tsai-Wu Criterion**

<i>Position (mm)</i>	<i>Exposure factor</i>
-----	
12.50	0.27393
12.26	0.27299
12.02	0.27099
11.78	0.26873
11.54	0.26616
11.29	0.26479
11.29	0.19188
11.05	0.18447
10.81	0.16913
10.57	0.15375
10.33	0.14354
10.09	0.14659

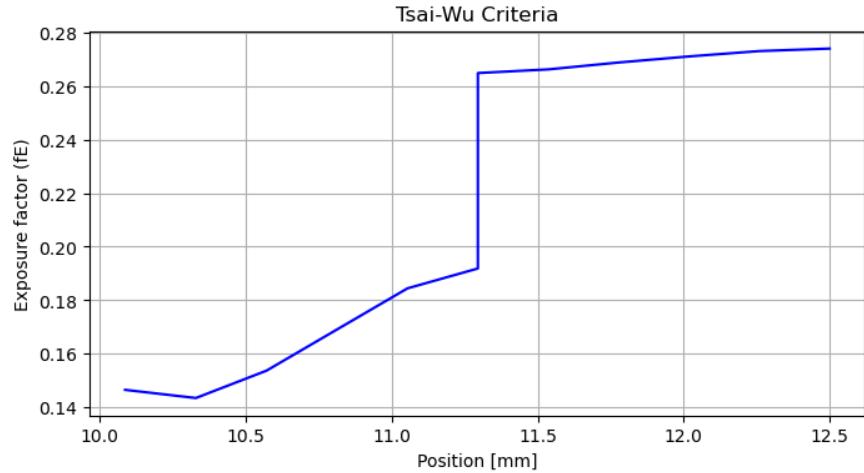


Figure 9, Tsai-Wu exposure factors for various positions in the pipe wall

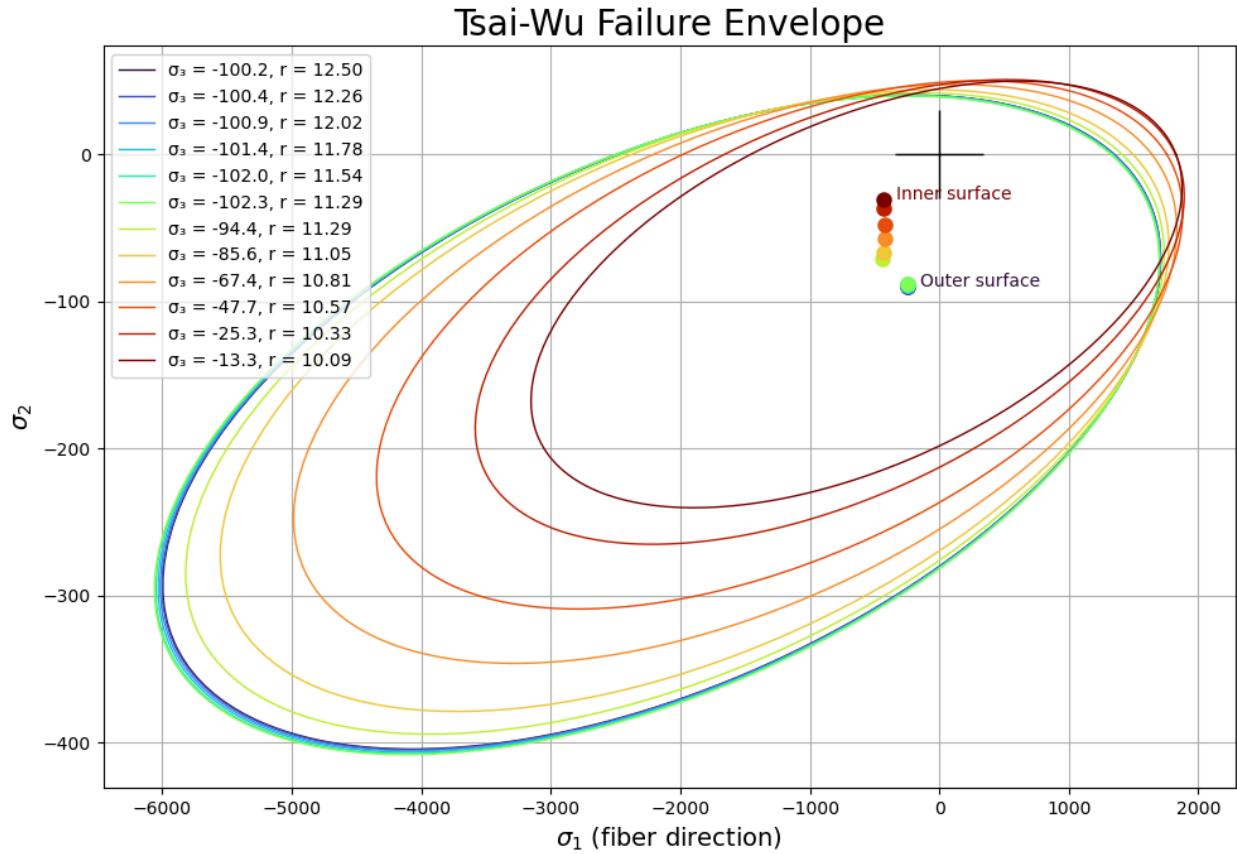
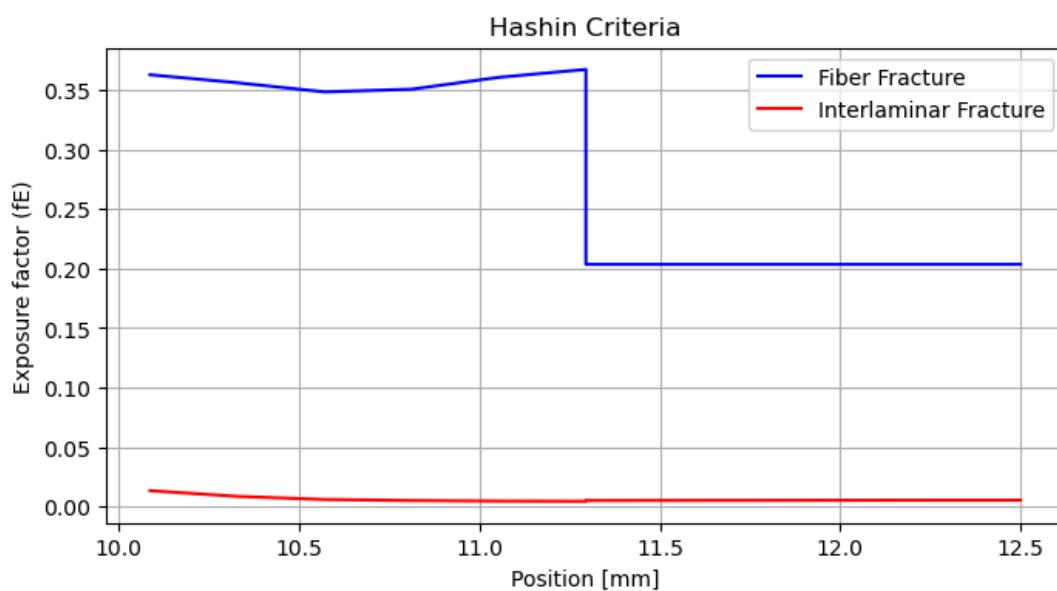


Figure 10 shows the Tsai-Wu envelope for various positions along the pipe wall. Each point on the pipe wall has its own envelope, represented by its corresponding color.

### ***Hashin Criteria***

<i><b>Position (mm)</b></i>	<i><b>Fiber Fracture</b></i>	<i><b>Inter Fiber Fracture</b></i>
12.50	0.20348	0.00562
12.26	0.20348	0.00560
12.02	0.20348	0.00554
11.78	0.20348	0.00548
11.54	0.20348	0.00541
11.29	0.20348	0.00538
11.29	0.36703	0.00474
11.05	0.36040	0.00489
10.81	0.35046	0.00532
10.57	0.34814	0.00619
10.33	0.35584	0.00882
10.09	0.36256	0.01362



*Figure 11, Hashin exposure factors for radial positions.*

## Discussion

The analysis of the composite intervention rod using Maximum Stress, Tsai, Wu and Hashin failure criteria provided insight into its structural reliability. The Maximum Stress Criterion offered a conservative baseline compared to Tsai-Wu and Hashin, flagging failure when any stress component exceeded directional strength limits. The criterion was especially conservative in this instance considering the compressive stresses.

It was observed that the exposure factor was higher in all the positions for Maximum Stress compared to Tsai Wu and Hashin. While the Maximum Stress Criterion confirmed no immediate failure under 100 MPa external pressure (all exposure factors <1), its simplicity limited its utility for composites, where multiaxial stress interactions dominated failure mechanisms.

The Tsai-Wu Criterion addressed this limitation by incorporating a quadratic stress coupling, revealing that the rod operated at 14.7-27.4 % of its failure envelope (exposure factors: 0.14659-0.27393, Fig. 9 and Fig. 10).

The Hashin Criteria further refined the analysis by distinguishing fiber-dominated failure (exposure: 0.20348-0.36256) from matrix/interlaminar failure (exposure: 0.00562-0.1362). The increased fiber fracture exposure at the inner radius (0.367 at 11.29 mm, Fig. 11) suggested that bending stresses concentrated near the neutral axis, while the outer layer's axially aligned fibers helped mitigate these stresses. Matrix failure remained negligible (<1.4%), validating the laminate's design. Given that fiber failure was the primary concern in horizontal well applications, Hashin's approach aligned best with the real world composite behavior, it was deemed the most suitable failure criterion in this case.

When selecting a safety factor, both reliability and practicality must be considered. Key factors include material variability, load uncertainties, and manufacturing defects. Using the critical exposure factor of 0.367 (fiber fracture) from the Hashin criteria yields a

safety factor of  $\frac{1}{0.367} = 2.72$  for the fibers. Although this does not account for all factors affecting the design, it suggests that the solution could be viable.

While this assignment provides a solid foundation for evaluating the intervention rod's performance, further analysis can be done to better capture real-world conditions. One area for improvement is the inclusion of frictional forces in the simulation. Horizontal well interventions involve contact forces between the rod and the wellbore which introduce additional bending and axial loads. Simulation of different friction coefficients and well curvatures could provide a more comprehensive picture of the failure risk.

## References

- Bosio, J., & Economides, M. (2011, January 2). *A life led horizontally*. Offshore Engineer. Retrieved March 12, 2025, from <https://www.oedigital.com/news/445102-a-life-led-horizontally>
- Experts Of CAE Assistant Group. (2024, August 14). *Introduction To Tsai Hill Failure Criterion & Tsai Wu Failure Criterion*. CAE Assistant. Retrieved March 12, 2025, from <https://caeassistant.com/blog/tsai-hill-failure-criterion-tsai-wu-criterion/>
- Gallagher, J. (2022, November 22). *History of Horizontal Wells*. Ellingson-DTD. Retrieved March 12, 2025, from <https://horizontaldrill.com/history-of-horizontal-wells/>
- Hashin, Z. (2012). *Zvi Hashin*. The Franklin Institute. Retrieved March 12, 2025, from <https://fi.edu/en/awards/laureates/zvi-hashin>
- Vedvik, N. P. (2024b). *Failure Criteria*. TMM4175 Lightweight Design. Retrieved March 12, 2025, from <https://nilspv.folk.ntnu.no/TMM4175/failure-criteria.html>
- Vedvik, N. P. (2024a). *Failure Prediction*. TMM4175 Lightweight Design. Retrieved March 10, 2025, from <https://nilspv.folk.ntnu.no/TMM4175/failure-prediction.html>

Vedvik, N. P. (2025, March 3). *Composite rod for horizontal well intervention*. TMM4175

Lightweight Design. Retrieved March 12, 2025, from

<https://nilspv.folk.ntnu.no/TMM4175/2025-assignment-2>

Veidth, M. (2024, October 31). *What Is Hashin Failure Criteria? | Hashin Damage In Abaqus*.

CAE Assistant. Retrieved March 12, 2025, from

<https://caeassistant.com/blog/hashin-failure-criteria-hashin-damage-abaqus/>

## Appendix

# Assignment 2

Hvis objektet fortrenger like mye vann som sin egen vekt, vil det ligge i ro.

$$\frac{\rho_{\text{pipe}}}{\rho_{\text{water}}} = 1$$

$$m_{\text{inside}} \approx 0 \implies \frac{m_{\text{cf}}}{m_{\text{water}}} = 1$$

$$\frac{\pi(D^2 - d^2)L\rho_{\text{cf}}}{\pi D^2 L \rho_{\text{water}}} = 1$$

$$\frac{D^2 - d^2}{D^2} = \frac{\rho_{\text{water}}}{\rho_{\text{cf}}}$$

$$\frac{d}{D} = \sqrt{1 - \frac{\rho_{\text{water}}}{\rho_{\text{cf}}}}$$

```
In [5]: # Imports
import numpy as np
import matplotlib.pyplot as plt
```

```
In [6]: # Constants
MAX_DIAMETER = 25 # mm
RHO_W = 1000 # kg/m^3
EXTERNAL_PRESSURE = 100 # MPa
```

## Functions

```
In [7]: def get_inner_diameter(D, rho_w, rho_material):
    """ Calculates the inner diameter based on the outer diameter to achieve neutral buoyancy """
    return np.sqrt(1 - (rho_w/rho_material)) * D

def fE_tsaiwu(s,m):
    """ Calculates the Tsai-Wu exposure factor """
    s1,s2,s3,s23,s13,s12=s[0],s[1],s[2],s[3],s[4],s[5]
    XT,YT,ZT,XC,YC,ZC,S12,S13,S23 = m['XT'], m['YT'], m['ZT'], m['XC'], m['YC'], m['ZC'], m['S12'], m['S13'], m['S23']
    f12, f13, f23 = m['f12'], m['f13'], m['f23']
    F1, F2, F3 = (1/XT)-(1/XC), (1/YT)-(1/YC), (1/ZT)-(1/ZC)
    F11, F22, F33 = 1/(XT*XC), 1/(YT*YC), 1/(ZT*ZC)
    F44, F55, F66 = 1/(S23**2), 1/(S13**2), 1/(S12**2)
    F12 = f12*(F11*F22)**0.5
    F13 = f13*(F11*F33)**0.5
    F23 = f23*(F22*F33)**0.5
    a=F11*(s1**2) + F22*(s2**2) + F33*(s3**2)+ 2*(F12*s1*s2 + F13*s1*s3 + F23*s2*s3)+\
    F44*(s23**2) + F55*(s13**2) + F66*(s12**2)
    if a==0:
        return 0
    b=F1*s1 + F2*s2 + F3*s3
    c=-1
    R=(-b+(b**2-4*a*c)**0.5)/(2*a)
    fE=1/R
    return fE

def fE_tsaiwu_const_s3(s1, s2, s3 ,m):
    """ Modified Tsai-Wu exposure factor for constant s3. Simplified to exclude shear stresses """
    XT,YT,ZT,XC,YC,ZC, = m['XT'], m['YT'], m['ZT'], m['XC'], m['YC'], m['ZC']
    f12, f13, f23 = m['f12'], m['f13'], m['f23']
    F1, F2, F3 = (1/XT)-(1/XC), (1/YT)-(1/YC), (1/ZT)-(1/ZC)
    F11, F22, F33 = 1/(XT*XC), 1/(YT*YC), 1/(ZT*ZC)
    F12 = f12*(F11*F22)**0.5
    F13 = f13*(F11*F33)**0.5
    F23 = f23*(F22*F33)**0.5
    a=F11*(s1**2) + F22*(s2**2) + 2*F12*s1*s2

    if a==0:
        return 0
    b=F1*s1 + F2*s2 + 2* (F13*s1*s3 + F23*s2*s3)
    c=-1 + F33*(s3**2) + F3*s3
    R=(-b+(b**2-4*a*c)**0.5)/(2*a)
    fE=1/R
    return fE

def fE_maxstress(s,m):
    """ Calculates the maximum stress exposure factor """
    # Using local variables for easier coding and readability..
    s1,s2,s3,s23,s13,s12=s[0],s[1],s[2],s[3],s[4],s[5]
```

```

XT,YT,ZT,XC,YC,ZC,S12,S13,S23 = m['XT'], m['YT'], m['ZT'], m['XC'], m['YC'], m['ZC'], m['S12'], m['S13'], m['S23']
fE=max(s1/XT,-s1/XC,s2/YT,-s2/YC,s3/ZT,-s3/ZC,abs(s12/S12),abs(s13/S13),abs(s23/S23))
return fE

def fE_maxstrain(s,m):
    """ Calculates the maximum strain exposure factor """
    s1,s2,s3,s23,s13,s12=s[0],s[1],s[2],s[3],s[4],s[5]
    XT,YT,ZT,XC,YC,ZC,S12,S13,S23 = m['XT'],m['YT'],m['ZT'],m['XC'],m['YC'],m['ZC'],m['S12'],m['S13'],m['S23']
    E1,E2,E3,v12,v13,v23,G12,G13,G23=m['E1'],m['E2'],m['E3'],m['v12'],m['v13'],m['v23'],m['G12'],m['G13'],m['G23']
    e1= (1/E1)*s1 + (-v12/E1)*s2 + (-v13/E1)*s3
    e2=(-v12/E1)*s1 + (1/E2)*s2 + (-v23/E2)*s3
    e3=(-v13/E1)*s1 + (-v23/E2)*s2 + (1/E3)*s3
    e23,e13,e12 = s23/G23, s13/G13, s12/G12
    fE=max( e1/(XT/E1),-e1/(XC/E1),e2/(YT/E2),-e2/(YC/E2),e3/(ZT/E3),-e3/(ZC/E3),
            abs(e12/(S12/G12)),abs(e13/(S13/G13)),abs(e23/(S23/G23)) )
    return fE

def fE_hashin(s,m):
    """ Hashin exposure factor """
    s1,s2,s3,s23,s13,s12=s[0],s[1],s[2],s[3],s[4],s[5]
    XT,YT,ZT,XC,YC,ZC,S12,S13,S23 = m['XT'], m['YT'], m['ZT'], m['XC'], m['YC'], m['ZC'], m['S12'], m['S13'], m['S23']
    if s1>0:
        R = ( 1/( (s1/XT)**2 + (1/S12**2)*(s12**2 + s13**2) ) )**0.5
        fE_FF=1/R
    if s1<=0:
        fE_FF=-s1/XC
    if (s2+s3)>0:
        temp=( (1/YT**2)*(s2+s3)**2*(1/S23**2)*(s23**2-s2*s3)+(1/S12**2)*(s12**2+s13**2) )
        if temp==0:
            fE_IFF = 0
        else:
            R = (1/temp)**0.5
            fE_IFF = 1/R
    if (s2+s3)<0:
        b = (1/YC)*((YC/(2*S23))**2-1)*(s2*s3)
        a = (1/(4*S23**2))*(s2+s3)**2+(1/S23**2)*(s23**2-s2*s3)+(1/S12**2)*(s12**2+s13**2)
        if a==0:
            fE_IFF = 0.0
        else:
            c=-1
            R=(-b+(b**2-4*a*c)**0.5)/(2*a)
            fE_IFF = 1/R
    return fE_FF, fE_IFF

def fE_hashin_const_s3(s1,s2,s3,m):
    """ Modified Hashin exposure factor for constant s3. Excludes shear stresses and (s2 + s3) >= 0 """
    XT,YT,ZT,XC,YC,ZC,S12,S13,S23 = m['XT'], m['YT'], m['ZT'], m['XC'], m['YC'], m['ZC'], m['S12'], m['S13'], m['S23']

    if s1>0:
        R = XT/s1
        fE_FF=1/R
    if s1<=0:
        fE_FF=-s1/XC

    if (s2+s3)<0:
        a = 1/(4*S23**2) * s2**2
        b = 1/(4*S23**2) * 2*s2*s3 - 1 / S23 * s2 * s3 + 1/YC * (YC**2/(4*S23**2) - 1) * s2
        if a==0:
            fE_IFF = 0.0
        else:
            c = 1/(4*S23**2) * s3**2 + 1/YC * (YC**2/(4*S23**2) - 1) * s3 -1
            R=(-b+(b**2-4*a*c)**0.5)/(2*a)
            fE_IFF = 1/R
    return fE_FF, fE_IFF

# ----- PLOTS -----

def plot_tsaiwu_various_s3(material, sigma_1, sigma_2, sigma_3, pos):
    """plots the Tsai-Wu failure envelope for various s3"""

    num_iterations = len(sigma_1)

    fig,ax = plt.subplots(figsize=(10,7))
    plt.title("Tsai-Wu", fontsize=20)

    # Lag en fargegradIENT basert på antall iterasjoner
    colors = [plt.cm.turbo ( i / (num_iterations - 1)) for i in range(num_iterations)]

    # Making axes through the origo:
    ax.plot((0),(0),'+',color='black',markersize=50)

    for i in range(num_iterations): # itterer over alle punkter
        # if (i%2 == 0):
        #     continue

        color = colors[i % len(colors)]
        s3i = sigma_3[i]
        # empty list of normal stresses in the 1-2 plane:
        s1_TW,s2_TW=[],[] # tsai-wu

        for a in np.linspace(0, 2*np.pi, 3600):

```

```

s1i=np.cos(a)
s2i=np.sin(a)

feTW=fE_tsaiwu_const_s3(s1i,s2i,s3i,material)

# then scaling by the Load-proportionality ratio (1/fE):
s1_TW.append(s1i/feTW)
s2_TW.append(s2i/feTW)

current_r = (MAX_DIAMETER - pos[i])/2
# Plot failure envelope
ax.plot(s1_TW, s2_TW, label=f"S3 = {s3i:.1f}, r = {current_r:.2f}", linewidth=1, color=color)

# ---- Plot actual stress ----
ax.plot(sigma_1[i], sigma_2[i], color=color, marker='o', markersize=8)

# Add text for positions
ax.text((sigma_1[0] + 100), sigma_2[0], "Outer surface", fontsize=10, color=colors[0])
ax.text((sigma_1[-1] + 100), sigma_2[-1], "Inner surface", fontsize=10, color=colors[-1])

ax.legend()
ax.set_xlabel(r'$\sigma_1$ (fiber direction)', fontsize=14)
ax.set_ylabel(r'$\sigma_2$', fontsize=14)
ax.grid(True)
plt.tight_layout()

def plot_hashin_various_s3(material, sigma_1, sigma_2, sigma_3, pos): # TODO: Incomplete function (not in use)

num_iterations = len(pos)

fig,ax = plt.subplots(figsize=(8,4))
# ax.set_xlim(-3000, 2000)
# ax.set_ylim(100, -400)

# Lag en fargegradient basert på antall iterasjoner
colors = [plt.cm.turbo(i / (num_iterations - 1)) for i in range(num_iterations)]

# Making axes through the origo:
ax.plot((0),(0),'+',color='black',markersize=50)

for i in range(num_iterations): # itterer over alle punkter

    color = colors[i % len(colors)]
    s3i = sigma_3[i]
    # empty list of normal stresses in the 1-2 plane:
    s1_Hashin,s2_Hashin,[],[]

    for a in np.linspace(0, 2*np.pi, 3600):
        s1i=np.cos(a)
        s2i=np.sin(a)

        fE_FF, fE_IFF =fE_hashin_const_s3(s1i,s2i,s3i,material)

        # then scaling by the Load-proportionality ratio (1/fE):
        s1_Hashin.append(s1i/fE_IFF)
        s2_Hashin.append(s2i/fE_IFF)

    # Plot failure envelope
    ax.plot(s1_Hashin,s2_Hashin,label=f"S3 = {s3i:.1f}, r = {((MAX_DIAMETER - pos[i]):.2f)}", linewidth=1, color=color)

    # ---- Plot actual stress ----
    ax.plot(sigma_1[i], sigma_2[i], color=color, marker='o')

    # Add text for positions
    ax.text(sigma_1[0], sigma_2[0], "Outer surface", fontsize=10, color=colors[0])
    ax.text(sigma_1[-1], sigma_2[-1], "Inner surface", fontsize=10, color=colors[-1])

    # ax.legend()
    ax.set_xlabel(r'$\sigma_1$ (fiber direction)', fontsize=14)
    ax.set_ylabel(r'$\sigma_2$', fontsize=14)
    ax.grid(True)
    plt.tight_layout()

def plot_max_stress(material, sigma_1, sigma_2, sigma_3, pos, label_start="", label_end ""):
    """Plots the failure envelope for max stress"""
    fig,ax = plt.subplots(figsize=(8,4))

    # Making axes through the origo:
    ax.plot((0),(0),'+',color='black',markersize=50)

    # empty list of normal stresses in the 1-2 plane:
    s1_MS,s2_MS,[],[]
    s1_ME,s2_ME,[],[]

    for a in np.linspace(0, 2*np.pi, 3600):
        s1i=np.cos(a)
        s2i=np.sin(a)

        fe_MS = fE_maxstress((s1i,s2i,0,0,0,0), material)
        fe_ME = fE_maxstrain((s1i,s2i,0,0,0,0), material)

        # then scaling by the Load-proportionality ratio (1/fE):

```

```

s1_MS.append(s1i/fe_MS)
s2_MS.append(s2i/fe_MS)

s1_ME.append(s1i/fe_ME)
s2_ME.append(s2i/fe_ME)

# Plot failure envelope
ax.plot(s1_MS,s2_MS,color="black",linewidth=1, ls="--", label="Max stress")
ax.plot(s1_ME,s2_ME,color="m", linewidth=1, ls="--", label="Max strain")

# ---- Plot actual stresses ----

# Plot points
ax.plot(sigma_1, sigma_2, color='red', label=r"\sigma_2$")
ax.plot(sigma_1, sigma_3, color='blue', label=r"\sigma_3$")

# Add text for min and max radial pos
ax.text(sigma_1[0], sigma_3[0], label_start, fontsize=10)
ax.text(sigma_1[-1], sigma_3[-1], label_end, fontsize=10)

ax.legend()
ax.set_xlabel(r'\sigma_1$', fontsize=14)
ax.set_ylabel(r'\sigma_2 / \sigma_3$', fontsize=14)
ax.grid(True)
plt.tight_layout()

```

Hashin and Tsai-Wu were modified to set sigma 3 to constant. Example using Tsai-Wu:

$$\begin{aligned}
a &= F_{11}\sigma_1^2 + F_{22}\sigma_2^2 + 2F_{12}\sigma_1\sigma_2 \\
b &= F_1\sigma_1 + F_2\sigma_2 + 2(F_{13}\sigma_1\sigma_3 + F_{23}\sigma_2\sigma_3) \\
c &= -1 + F_{33}\sigma_3^2 + F_3\sigma_3 \\
R &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\
f_E &= \frac{1}{R}
\end{aligned}$$

## Material

```
In [8]: Carbon_EpoxyA = {
    "name": "Carbon/Epoxy(a)",
    "units": "MPa-mm-Mg", "type": "UD", "fiber": "Carbon",
    "Vf": 0.55,
    "rho": 1600, # kg/m^3
    "description": "Typical low modulus carbon/Epoxy from TMM4175",
    "E1": 130000,
    "E2": 10000,
    "E3": 10000,
    "v12": 0.28, "v13": 0.28, "v23": 0.5,
    "G12": 4500, "G13": 4500, "G23": 3500,
    "a1": -0.5e-06, "a2": 3.0e-05, "a3": 3.0e-05,
    "XT": 1800, "YT": 40, "ZT": 40,
    "XC": 1200, "YC": 180, "ZC": 180,
    "S12": 70, "S13": 70, "S23": 40,
    "f12": 0.5, "f13": 0.5, "f23": -0.5
}
```

## Pipe Dimensions

```
In [9]: D_end = MAX_DIAMETER
num_layers = 2

d = get_inner_diameter(D_end, RHO_W, rho_material=Carbon_EpoxyA['rho'])

ri = d/2.0
ro = D_end/2.0
t = ro - ri
h = t / num_layers

print(f"Inner radius: {ri:.5f} mm\nOuter radius: {ro:.5f} mm\nThickness: {t:.5f} mm\nLaminate thickness: {h:.5f} mm")

Inner radius: 7.65466 mm
Outer radius: 12.50000 mm
Thickness: 4.84534 mm
Laminate thickness: 2.42267 mm
```

## Calculation for Point Force in Abaqus (longitudinal pressure)

```
In [ ]: A = 0.5 * np.pi * ro**2
F = A * EXTERNAL_PRESSURE
F
```

Out[ ]: 24543.692606170258

# Results From Abaqus

In [11]: # Position (from  $r\theta$  to  $ri$ )

```
pos = [
    0,
    0.482111,
    0.96419,
    1.44623,
    1.92824,
    2.41019,
    2.41019,
    2.89234,
    3.3751,
    3.85853,
    4.34273,
    4.82781,
]

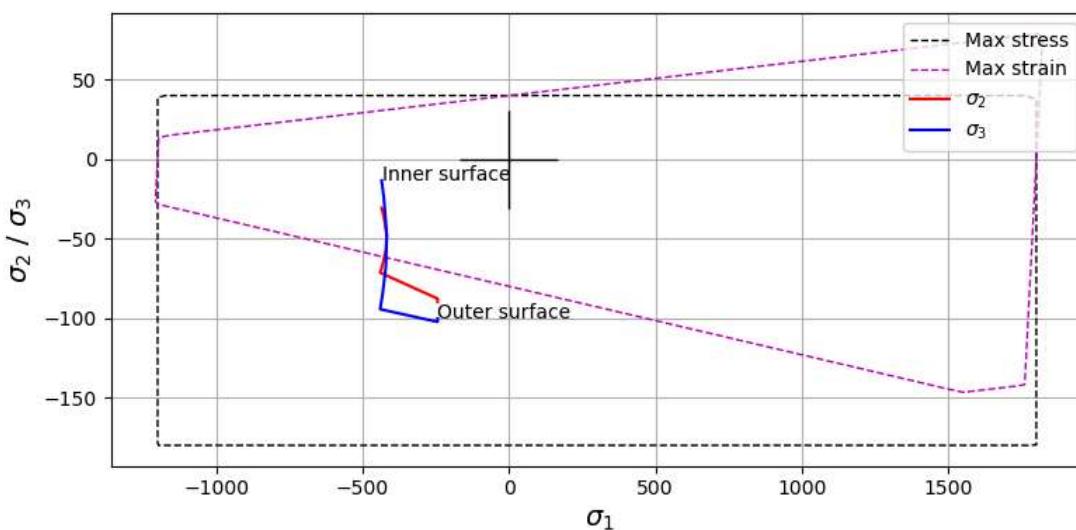
S11 = [
    -244.178,
    -244.178,
    -244.178,
    -244.178,
    -244.178,
    -244.178,
    -440.439,
    -432.477,
    -420.548,
    -417.767,
    -427.01,
    -435.068,
]

S22 = [
    -89.8445,
    -89.6269,
    -89.1631,
    -88.6367,
    -88.0361,
    -87.7157,
    -71.3565,
    -66.8048,
    -57.4717,
    -47.5171,
    -36.539,
    -30.734,
]

S33 = [
    -100.201,
    -100.418,
    -100.882,
    -101.408,
    -102.009,
    -102.33,
    -94.3609,
    -85.6005,
    -67.4482,
    -47.6588,
    -25.3043,
    -13.3474,
]
```

## Max Stress

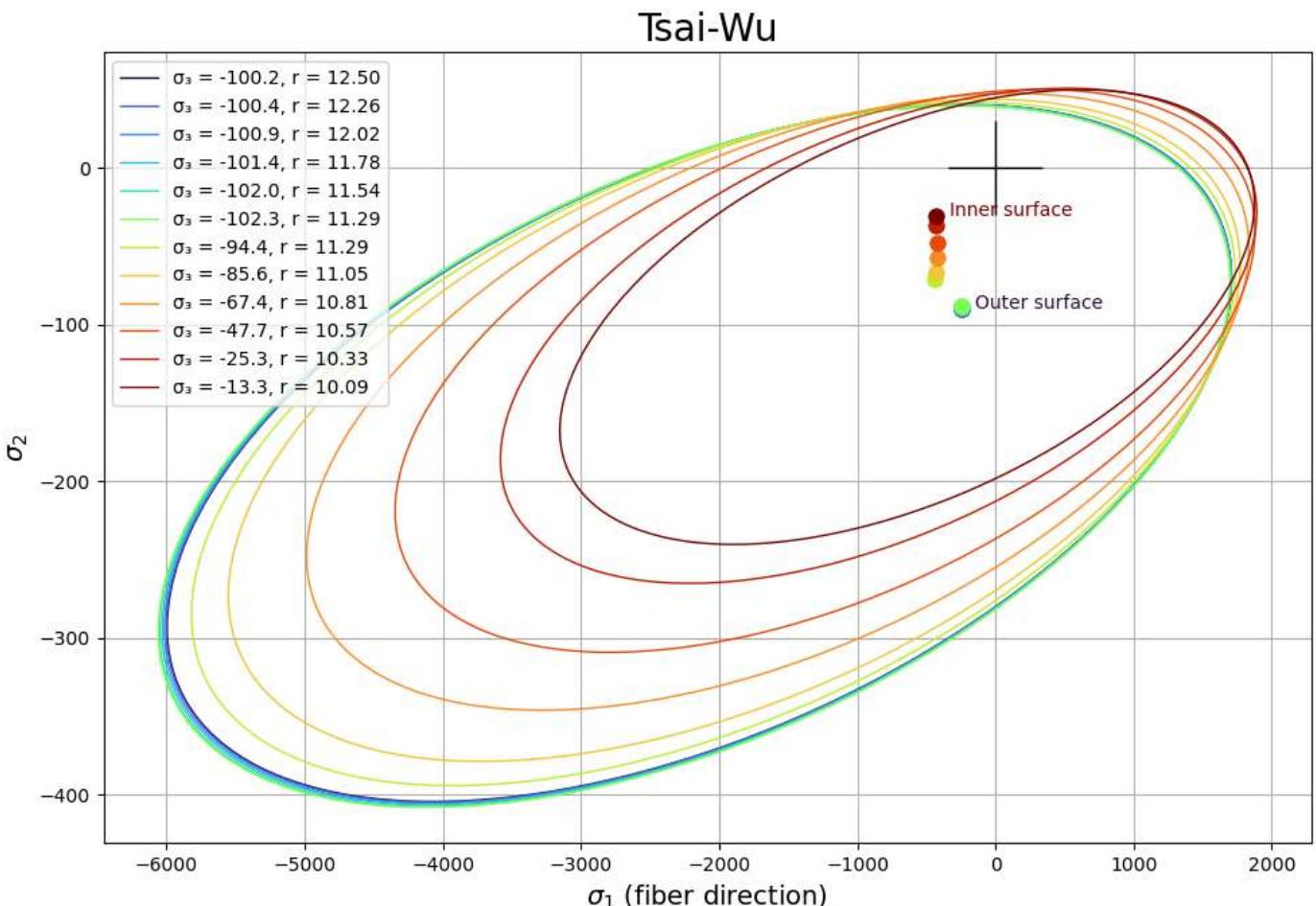
In [13]: plot\_max\_stress(Carbon\_EpoxyA, sigma\_1=S11, sigma\_2=S22, sigma\_3=S33, pos=pos, label\_start="Outer surface", label\_end="Inner surface")



## Tsai Wu

```
In [14]: # Get D and d
D = MAX_DIAMETER
d = get_inner_diameter(D, RHO_W, rho_material=Carbon_EpoxyA['rho'])

plot_tsaiwu_various_s3(Carbon_EpoxyA, sigma_1=S11, sigma_2=S22, sigma_3=S33, pos=pos)
```



## Print and plot the exposure factors

```
In [20]: print("Tsai-Wu criterion")
print(f'{Position:>10} {Exposure factor:>20}')
print("-" * 40)

fig,ax = plt.subplots(figsize=(8,4))

fE_list, pos_list = [], []

for i in range(len(pos)):
    r_pos = (MAX_DIAMETER - pos[i]) / 2
    s1i, s2i, s3i = S11[i], S22[i], S33[i]

    fe = fE_tsaiwu_const_s3(s1i, s2i, s3i, Carbon_EpoxyA)
    fE_list.append(fe)
    pos_list.append(r_pos)
```

```

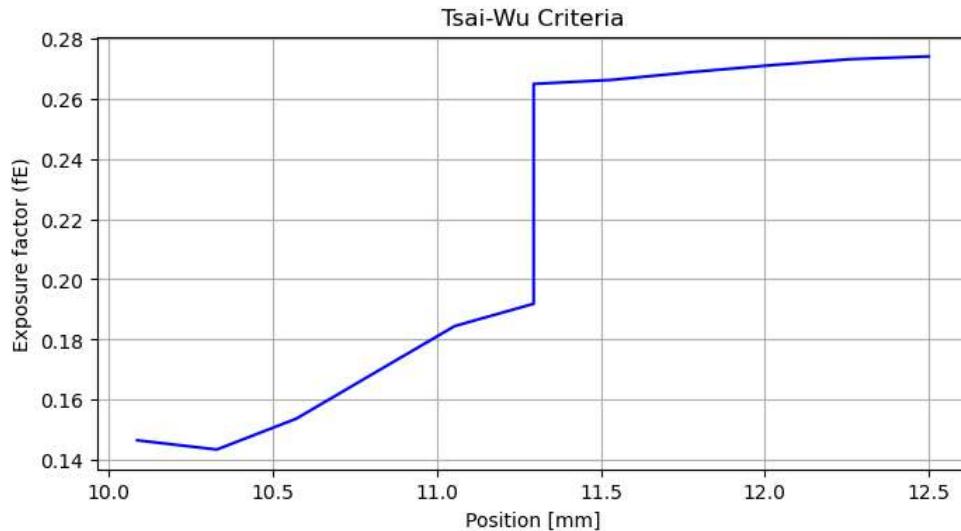
print(f"{{r_pos:10.2f} {fe:20.5f}}")

plt.plot(pos_list, fE_list, label="Fiber Fracture", color="b")
plt.xlabel("Position [mm]")
plt.ylabel("Exposure factor (fE)")
plt.grid()
plt.title("Tsai-Wu Criteria")
plt.show()

```

Tsai-Wu criterion  
Position      Exposure factor

Position	Exposure factor
12.50	0.27393
12.26	0.27299
12.02	0.27099
11.78	0.26873
11.54	0.26616
11.29	0.26479
11.29	0.19188
11.05	0.18447
10.81	0.16913
10.57	0.15375
10.33	0.14354
10.09	0.14659



## Hashin

### Print and plot the exposure factors

```

In [18]: print("Hashin Criteria")
print(f"{'Position':>10} {'Fiber Fracture':>20} {'Interlaminar Fracture':>25}")
print("-" * 60)

fig,ax = plt.subplots(figsize=(8,4))

fE_FF_list, fE_IFF_list, pos_list = [], [], []

for i in range(len(pos)):
    r_pos = (MAX_DIAMETER - pos[i]) / 2
    s1i, s2i, s3i = S11[i], S22[i], S33[i]

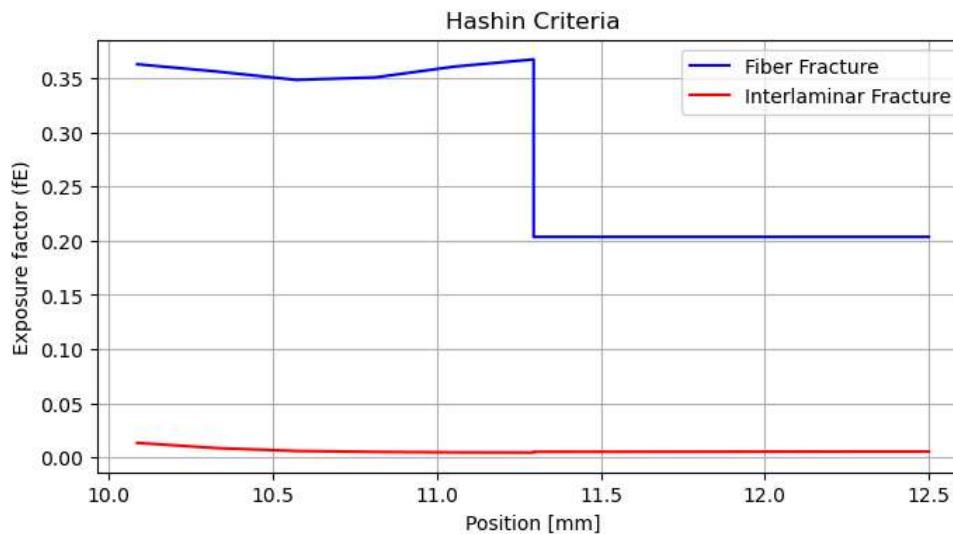
    fE_FF, fE_IFF = fE_hashin_const_s3(s1i, s2i, s3i, Carbon_EpoxyA)
    fE_FF_list.append(fE_FF)
    fE_IFF_list.append(fE_IFF)
    pos_list.append(r_pos)

    print(f"{{r_pos:10.2f} {fE_FF:20.5f} {fE_IFF:25.5f}}")

plt.plot(pos_list, fE_FF_list, label="Fiber Fracture", color="b")
plt.plot(pos_list, fE_IFF_list, label="Interlaminar Fracture", color="r")
plt.legend()
plt.xlabel("Position [mm]")
plt.ylabel("Exposure factor (fE)")
plt.grid()
plt.title("Hashin Criteria")
plt.show()

```

Hashin Criteria	Position	Fiber Fracture	Interlaminar Fracture
	12.50	0.20348	0.00562
	12.26	0.20348	0.00560
	12.02	0.20348	0.00554
	11.78	0.20348	0.00548
	11.54	0.20348	0.00541
	11.29	0.20348	0.00538
	11.29	0.36703	0.00474
	11.05	0.36040	0.00489
	10.81	0.35046	0.00532
	10.57	0.34814	0.00619
	10.33	0.35584	0.00882
	10.09	0.36256	0.01362



## Buckling calculations

```
In [ ]: # Compare different outer diameters and buckling
Outer_diamters = np.linspace(MAX_DIAMETER, MAX_DIAMETER*0.4, 100)

buckling_force = []

for D_outer in Outer_diamters:
    ro = D_outer / 2.0
    d = get_inner_diameter(D_outer, RHO_W, rho_material=Carbon_EpoxyA['rho'])
    ri = d / 2.0

    I = np.pi * (ro**4 - ri**4) / 4.0

    # E modulus from Abaqus
    E_eff = 39E3 # GPa

    # Calculate critical buckling force
    L = 1000 # mm
    P = np.pi**2 * E_eff * I / (L**2)
    buckling_force.append(P)

fig,ax = plt.subplots(figsize=(8,4))

plt.plot(Outer_diamters, buckling_force, color="b")
plt.xlabel("Outer Pipe Diameter [mm]")
plt.ylabel("Critical force [N]")
plt.grid()
plt.title("Buckling (L = 1 m)")
plt.show()
```

