# B-bar comparative study

*Student: Eivind Hugaas, NTNU, eivind.hugaas@ntnu.no*

## Introduction

In the subject KT8215 a student project was carried out investigating the B-bar element formulation's ability to alleviate volumetric locking in fully integrated elements. This report summarizes the work.

## Problem

Volumetric locking is the key problem to be solved by using the B-bar element formulation. Volumetric locking occurs when the interpolation within the element is not able to capture correct distribution of the strain for nearly incompressible materials due to over constraining from too many volumetric strain degrees of freedom causing too stiff behavior. Volumetric locking occurs when using fully integrated elements. This report will only focus on hexahedron elements and will not take into account other element shapes, most prominently the tetrahedron.

There are other ways to overcome volumetric locking than using the B-Bar element formulation. Using quadratically interpolated elements with reduced integration will eliminate the volumetric locking. In Abaqus this is the C3D20R element. Reduced integration also solves the issue with linearly interpolated elements, C3D8R elements in Abaqus. However, with linearly interpolated reduced integration elements, unwanted modes are easily triggered with nearly incompressible materials. These modes are known as hour glass modes due to their characteristic shape. Using quadratically interpolated elements is expensive computationally compared to linearly interpolated ones, so modifying linearly interpolated fully integrated elements so they can handle volumetric locking is advantageous to reduce the computational cost. The B-Bar element formulation is one such modification, which this report investigates the effect of using.

## Theoretical background

Understanding volumetric locking can be difficult. This section seeks to present the key points of the B-Bar element formulation and how volumetric locking can be understood on the basis of why the B-Bar formulation can alleviate it. It is assumed familiarity with multi-field element formulations.

The B-Bar element formulation is a modification of the three field element formulation. Instead of supplying separate interpolation functions for the assumed stress, assumed strain and deformation fields it couples the assumed strain field to the deformation field by a modified B-matrix. Besides putting constraints on the strain field, the assumed stress field has to be orthogonal to the difference between the strain field deducted from displacements and the assumed strain field, as shown in equation (1).

$$\int_V (\boldsymbol{\varepsilon}(\mathbf{u}) - \tilde{\boldsymbol{\varepsilon}})^T \tilde{\boldsymbol{\sigma}} dV = 0 \tag{1}$$

Equation (2) shows the Hu Washizu variational principle's weak form without body forces and with the condition for the B-Bar method stated in equation (1) acting, as it is used in this report to deduct the element stiffness matrix. As can be seen, the equation is much simplified compared to the full

Hu Washizu formulation, which is part of why the B-Bar method is considered a fairly elegant element formulation.

$$\delta\Pi_{\mathrm{HW}} = \int_V \delta\tilde{\boldsymbol{\varepsilon}}^T \boldsymbol{\sigma}(\tilde{\boldsymbol{\varepsilon}})\mathrm{dV} - \int_{S_t} \delta\boldsymbol{u}^T \bar{\mathbf{t}}\,\mathrm{dS} = 0 \tag{2}$$

To relax the constraint put by volumetric locking on a fully integrated element the assumed strain field using the B-Bar formulation can be formulated so that the volume averaged volumetric strain is the governing degree of freedom in the assumed strain field. The resulting formulation for the assumed strain field is given in equation (3). The assumed strain field is equal to the deformation deducted strain field as long as there is no volumetric change. Once volumetric strains are forced by the interpolation functions in the integration points, the assumed strain field is equal to the deformation deducted strain field plus its difference in pointwise volumetric strain with volume averaged volumetric strain. It is important to note that the average volumetric strain is averaged over the volume, meaning that it differs from just taking the average from the integration points.

$$\tilde{\varepsilon}_{ij} = \varepsilon_{ij} + \frac{1}{3}(\bar{\varepsilon}_v - \varepsilon_{kk})\delta_{ij} \tag{3}$$

It is evident from equation (3) that if the element is uniformly deformed it wont potentially lock and the assumed strain field will equal the deformation deducted strain field. However, if there is a variation in the volumetric strain over the element, locking becomes an issue and the assumed strain field will be different from the deformation deducted strain field.

An example of the B-Bar formulation’s effect is if the average volumetric strain is small or zero, but the volumetric strains in the integration points are relatively large. The assumed strain field will then be close or equal the deviatoric strain and the assumed stress field will be much relaxed compared to if the volumetric strains in the integration points had been taken into account, assuming a nearly incompressible material.

## Comparative method

Abaqus does not have the B-Bar element formulation implemented. To provide comparison, the B-Bar was coded in a Python script and compared with two other element formulations also coded in the script. Abaqus was used to benchmark the script to verify its functionality (no bugs). The two other formulations were the fully integrated linear eight node element (eight integration points) and the reduced integration linear eight node element (one integration point).

The coding for the element interpolation and numerical integration was carried out at large based on the methods suggested by C. Felippa [1], [2]. The B-Bar element formulation was coded based on the methods and equations described by O. S. Hopperstad [3], [4].

The Python script can be found in Eivind Hugaas’ GitHub repository [5], to run it and get detailed explanation on its functionality, clone the folder structure and open “Element_Formulations.py”.

## Python implementation

The basis for the Python script’s element formulations is the eight node hexahedron element as illustrated with node numbers and natural coordinate system in Figure 1.

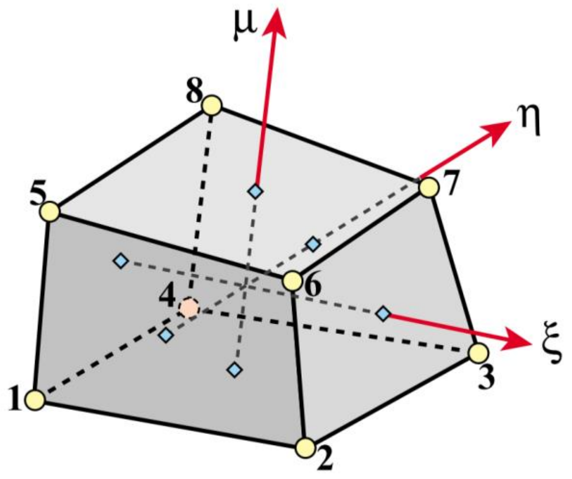
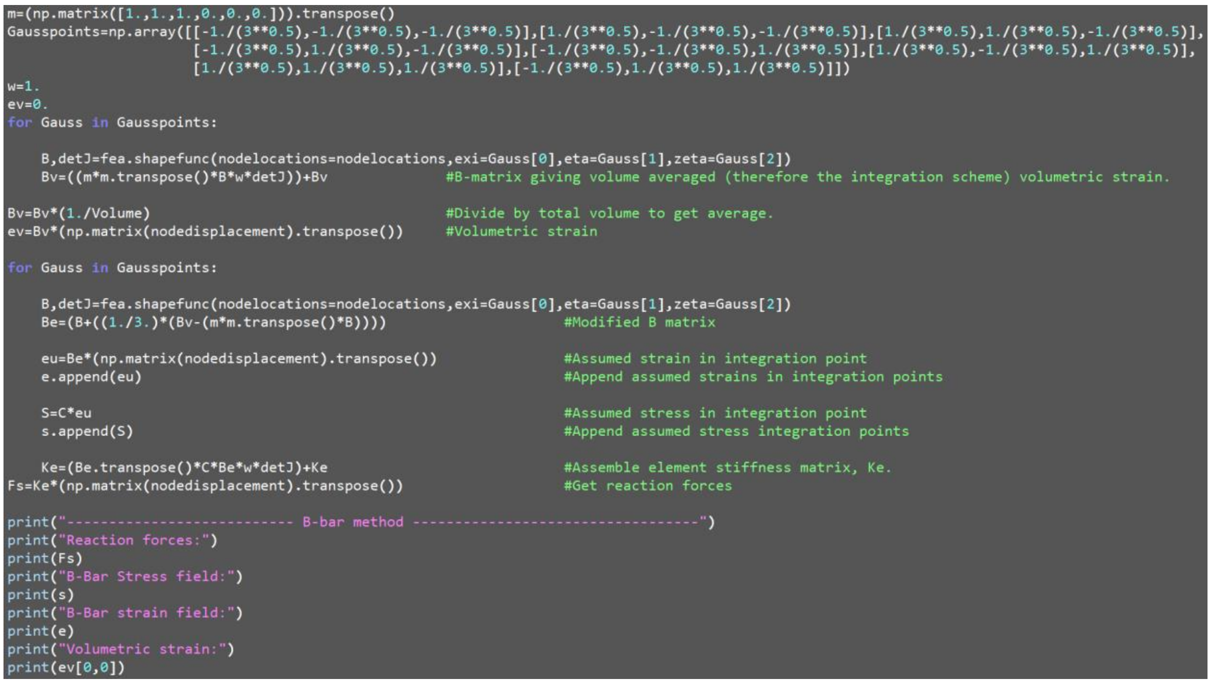


*Figure 1 Eight node hexahedron with node numbers and natural coordinate system, taken from [1].*

The script is as simple as possible and does not include any constraint conditions, meaning all degrees of freedom has to be set. It includes no iteration scheme either, meaning that no non-linear deformations are taken into account.

To give an example of the scripts functionality, a snip of the B-Bar method integration scheme is shown in Figure 2. Looking at the Be-matrix giving the assumed strain field from the node displacements it can be seen to follow equation (3). The B-matrix and the determinant of the Jacobian is gathered from a separate script. Which script is which and how they are imported can be found in more detail in the GitHub link and is nothing but standard Python routine. The reader is assumed to have a basic understanding of Python.



```
m=(np.matrix([1.,1.,1.,0.,0.,0.])).transpose()
Gausspoints=np.array([[-1./(3**0.5),-1./(3**0.5),-1./(3**0.5)],[1./(3**0.5),-1./(3**0.5),-1./(3**0.5)],[1./(3**0.5),1./(3**0.5),-1./(3**0.5)],
                      [-1./(3**0.5),1./(3**0.5),-1./(3**0.5)],[-1./(3**0.5),-1./(3**0.5),1./(3**0.5)],[1./(3**0.5),-1./(3**0.5),1./(3**0.5)],
                      [1./(3**0.5),1./(3**0.5),1./(3**0.5)],[-1./(3**0.5),1./(3**0.5),1./(3**0.5)]])
w=1.
ev=0.
for Gauss in Gausspoints:

    B,detJ=fea.shapefunc(nodelocations=nodelocations,exi=Gauss[0],eta=Gauss[1],zeta=Gauss[2])
    Bv=((m*m.transpose()*B*w*detJ))+Bv              #B-matrix giving volume averaged (therefore the integration scheme) volumetric strain.

Bv=Bv*(1./Volume)                                   #Divide by total volume to get average.
ev=Bv*(np.matrix(nodedisplacement).transpose())     #Volumetric strain

for Gauss in Gausspoints:

    B,detJ=fea.shapefunc(nodelocations=nodelocations,exi=Gauss[0],eta=Gauss[1],zeta=Gauss[2])
    Be=(B+((1./3.)*(Bv-(m*m.transpose()*B))))                   #Modified B matrix

    eu=Be*(np.matrix(nodedisplacement).transpose())             #Assumed strain in integration point
    e.append(eu)                                                #Append assumed strains in integration points

    S=C*eu                                                      #Assumed stress in integration point
    s.append(S)                                                 #Append assumed stress integration points

    Ke=(Be.transpose()*C*Be*w*detJ)+Ke                          #Assemble element stiffness matrix, Ke.
Fs=Ke*(np.matrix(nodedisplacement).transpose())                 #Get reaction forces

print("-------------------------- B-bar method ---------------------------------")
print("Reaction forces:")
print(Fs)
print("B-Bar Stress field:")
print(s)
print("B-Bar strain field:")
print(e)
print("Volumetric strain:")
print(ev[0,0])
```

*Figure 2 Integration for the B-Bar method*

## Load case

A thick walled pressurized pipe in plane strain is an example load case where volumetric locking will occur for elements prone to this for nearly incompressible materials. This load case was chosen for the work. This load case triggers a strain field with small volume averaged volumetric strain relative to the volumetric strains in the integration points.

A simple pressurized pipe was analyzed with Abaqus using quadratically interpolated reduced integration elements to get the correct deformation shape. This could also be done analytically, but Abaqus was chosen due to its convenient nodal displacement result extraction. Nodal displacement values were extracted for one element and applied in the script and in a single element in Abaqus for comparison of reaction forces, the chosen benchmark parameter. The material properties are reported in Table 1 and the nodal locations and displacements are reported in Table 2. Integration point number 1 in a fully integrated four node element was chosen to showcase a comparison between the strain components and is provided in

Table 3. As can be seen, there is small difference between the B-Bar strain and the deviatoric strain despite the volumetric strain being considerable in the integration point (556.4 microstrain).

Figure 3 shows the pressurized pipe with the chosen element along with the coordinate system used. For implementation into the script, cartesian and not cylindrical coordinates were used. Figure 4 shows the single element with displacement constraints.

*Table 1 Material properties.*

| **Parameter** | **Value** |
|---|---|
| Young’s modulus | 100 |
| Poisson ratio | 0.495 |

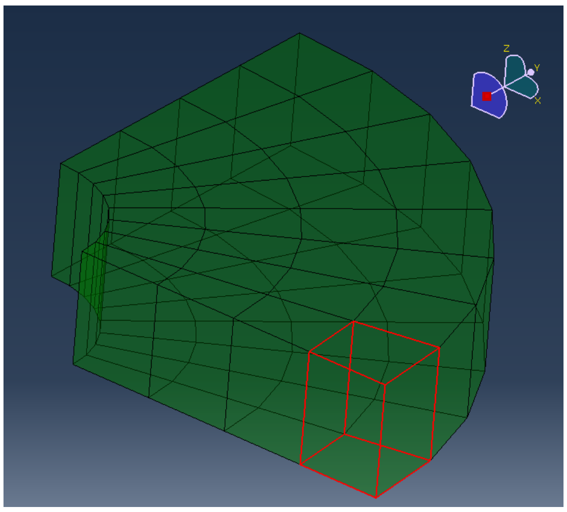


*Figure 3 Chosen element. Coordinate system is as displayed in the upper right corner.*

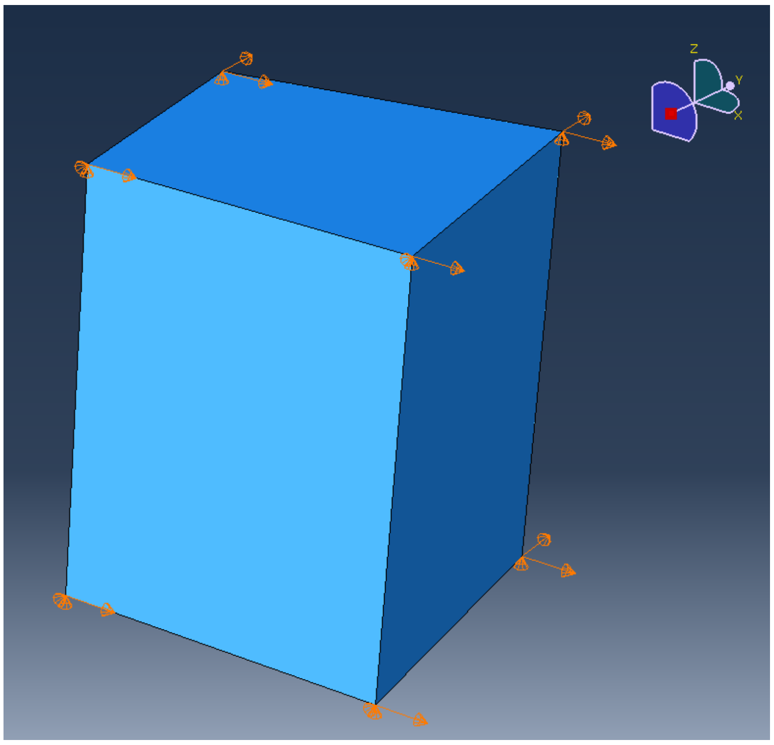


*Figure 4 Element with arrows indicating application of displacement constraints.*

*Table 2 Node Coordinates and applied plain strain displacements. The element is 1 unit long in the z axis.*

| **Nodes:** | **1,5** | | **2,6** | | **3,7** | | **4,8** | |
|---|---|---|---|---|---|---|---|---|
| **Coordinate** | **x** | **Y** | **x** | **y** | **x** | **y** | **x** | **y** |
| **Undeformed coord.** | 3.2500 | 0.0000 | 4.0000 | 0.0000 | 3.923141 | 0.780361 | 3.187552 | 0.634044 |
| **Applied displacement** | 0.01582 | 0.0000 | 0.01289 | 0.0000 | 0.012649 | 0.002516 | 0.015508 | 0.003085 |

*Table 3 Strain in integration point one using the normal B-matrix and assumed strain deducted from the B-Bar formulation compared to the deviatoric strain from the B-matrix. As can be seen, the B-Bar strain is close to the deviatoric strain in the integration point.*

| | **Strainfield from B-matrix (microstrain)** | **Deviatoric strain (microstrain)** | **Assumed B-Bar strain field (microstrain)** |
|---|---|---|---|
| $\varepsilon_{11}$ | -3868 | -4054 | -4033 |
| $\varepsilon_{22}$ | 4425 | 4239 | 4260 |
| $\varepsilon_{33}$ | 0 | -185 | -164 |
| $\varepsilon_{12}$ | -1172 | -1172 | -1151 |
| $\varepsilon_{23}$ | 0 | 0 | 21 |
| $\varepsilon_{31}$ | 0 | 0 | 21 |

## Python script verification tests

To verify the script’s functionality for its reduced integration element, the C3D8R element in Abaqus was used as benchmark. Table 4 shows the benchmarking. As can be seen, there is agreement between the script and Abaqus for the reduced integration element.

The linearly interpolated fully integrated element in Abaqus (C3D8) has selective integration to alleviate some of the shear and volumetric locking issues. The selective integration is too poorly described in the Abaqus documentation [6] to state to what degree it eliminates the volumetric locking issue. To verify the functionality of the script's fully integrated element it was benchmarked against the C3D8 element in Abaqus with a poisson ratio of 0.1 to nearly eliminate the effect of volumetric locking. Table 5 shows the benchmarking. As can be seen, there is agreement between the script and Abaqus for the fully integrated element.

*Table 4 Benchmarking of the Python script's reduced integration element with Abaqus C3D8R reduced integration element. Shaded columns contains too small values for relevant relative comparison.*

| **Node:** | **1** | | | **2** | | | **3** | | | **4** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Reaction Force component:** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** |
| Abaqus C3D8R | 0.0177 | -0.0873 | -0.0269 | -0.0003 | -0.0890 | -0.0288 | -0.0177 | 0.0872 | -0.0288 | 0.0003 | 0.0891 | -0.0269 |
| Script 8 node red. int. | 0.0177 | -0.0873 | -0.0278 | -0.0003 | -0.0890 | -0.0278 | -0.0177 | 0.0873 | -0.0278 | 0.0003 | 0.0890 | -0.0278 |
| **Script red int./C3D8R** | **1.0000** | **1.0000** | **1.0357** | **1.0000** | **1.0000** | **0.9667** | **1.0000** | **1.0005** | **0.9667** | **1.0000** | **0.9994** | **1.0357** |

*Table 5 Benchmarking of the Python script's fully integrated element with Abaqus C3D8 fully integrated element at a poisson ratio of 0.1 to nearly eliminate the effect of volumetric locking which the C3D8 element handles through selectively reduced integration. Shaded columns contains too small values for relevant relative comparison.*

| **Node:** | **1** | | | **2** | | | **3** | | | **4** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Reaction Force component:** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** |
| Abaqus C3D8<br>poisson ratio =0.1 | 0.0697 | -0.0665 | 0.0010 | -0.0564 | -0.0685 | -0.0012 | -0.0686 | 0.0562 | -0.0012 | 0.0553 | 0.0788 | 0.0010 |
| Script 8 node full int.<br>poisson ratio =0.1 | 0.0700 | -0.0688 | -0.0005 | -0.0566 | -0.0667 | 0.0003 | -0.0684 | 0.0543 | 0.0003 | 0.0550 | 0.0811 | -0.0005 |
| **Script full int./C3D8** | **1.0037** | **1.0349** | **-0.5039** | **1.0037** | **0.9725** | **-0.2619** | **0.9970** | **0.9665** | **-0.2593** | **0.9954** | **1.0294** | **-0.5005** |

## Triggering of volumetric locking

Due to only analyzing single elements, the correct solution to the problem is obtained from the reduced integration element, seeing as hour glass modes are prohibited by the constraints. Table 6 shows the reduced integration element compared with the fully integrated element. The fully integrated element can be seen to have extensive volumetric locking with as much as 5 times the reduced integration element's reaction force.

*Table 6 Volumetric locking demonstrated by the fully integrated element compared to the reduced integrated element.*

| **Node:** | **1** | | | **2** | | | **3** | | | **4** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Reaction Force component:** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** |
| Script 8 node full int. | 0.0386 | -0.2770 | -0.1444 | -0.0173 | 0.0651 | 0.0917 | -0.0006 | -0.0669 | 0.0887 | -0.0206 | 0.2787 | -0.1473 |
| Script 8 node red. int. | 0.0177 | -0.0873 | -0.0278 | -0.0003 | -0.0890 | -0.0278 | -0.0177 | 0.0873 | -0.0278 | 0.0003 | 0.0890 | -0.0278 |
| **Script full int./Script red int.** | **2.1856** | **3.1740** | **5.1891** | **56.8788** | **-0.7312** | **-3.2950** | **0.0367** | **-0.7664** | **-3.1891** | **-67.7738** | **3.1307** | **5.2950** |

An example of the hour glass mode can be seen in Figure 5. The figure shows how the C3D8R has hour glassing while the C3D20R element has not when several elements are used to analyze the problem giving enough degrees of freedom for the hour glass mode to develop.

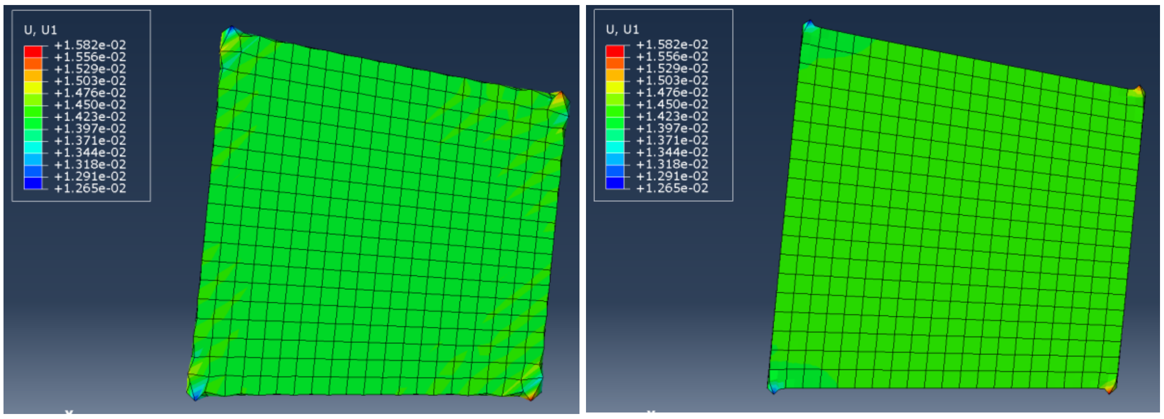


*Figure 5 Comparison of C3D8R (left) and C3D20R (right) using several elements. It can be seen that the C3D8R element has hour glass modes, while the C3D20R has not.*

## B-Bar comparison

Table 7 shows comparison of the B-Bar element formulation with the reduced integration element reaction forces. As can be seen, the B-Bar comes within 4% of the correct solution disregarding RF1 in node 2 and 4, which are too small for relevant comparison.

*Table 7 Comparison of the B-Bar element formulation with the reduced integration element. As can be seen, the B-Bar manages to aproximate the correct solution well. Shaded columns contains too small values for relevant relative comparison.*

| **Node:** | **1** | | | **2** | | | **3** | | | **4** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Reaction force** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** | **RF1** | **RF2** | **RF3** |
| **Script B-Bar** | 0.0181 | -0.0914 | -0.0261 | -0.0007 | -0.0856 | -0.0296 | -0.0173 | 0.0839 | -0.0296 | -0.0002 | 0.0932 | -0.0261 |
| **Script 8 node red. int.** | 0.0177 | -0.0873 | -0.0278 | -0.0003 | -0.0890 | -0.0278 | -0.0177 | 0.0873 | -0.0278 | 0.0003 | 0.0890 | -0.0278 |
| **Relative difference red int.** | **1.0260** | **1.0471** | **0.9705** | **2.2085** | **0.9624** | **1.0282** | **0.9789** | **0.9616** | **1.0275** | **-0.5349** | **1.0462** | **0.9698** |

## Conclusion

Volumetric locking is an important issue to take into account when analyzing nearly incompressible materials. This was proven by comparing an eight node linearly interpolated fully integrated element with the correct solution of a pressurized pipe load case. The fully integrated element had reaction forces that were as much as five times higher than the correct solution. The B-Bar element formulation was benchmarked against the correct solution and was proven to be an efficient element formulation to overcome the problem of volumetric locking in fully integrated linearly interpolated elements, having reaction forces within 4% of the correct solution.

## References

[1] C. Felippa, ASEN 6367: Advanced Finite Element Methods for Solids, Plates and Shells (AFEM) (subject with learning materials at: https://www.colorado.edu/engineering/CAS/courses.d/AFEM.d/), Boulder: University of Colorado Boulder, 2017.

[2] C. Felippa, Advanced Variational Methods in Mechanics (subject with learning materials available at https://www.colorado.edu/engineering/CAS/courses.d/AVMM.d/Home.html), Boulder: University of Colorado, 2017.

[3] O. S. Hopperstad, Energimetoder, forelesningsnotater, Trondheim: NTNU, 2015.

[4] O. S. Hopperstad, Lecture Notes, Theoretical basis for nonlinear finite element methods, Trondheim: NTNU, 2018.

[5] E. Hugaas, «Eivind Hugaas' GitHub,» 12 May 2018. [Internett]. Available: https://github.com/eivindhugaas/FEA_program. [Funnet 12 May 2018].

[6] Dassault Systèmes, SIMULIA User Assistance 2017, 2016.