

Introduction to Locking in Finite Element Methods

MT05.36

Bachelor final project

TU/e Faculty of Mechanical Engineering
Bachelor Final Project
Trimester 3.3 2004-2005

Student: G.A.H. van den Oord (0529435)

Supervisor: dr. ir. W.A.M. Brekelmans

Eindhoven, June 29, 2005

Contents

Preface	3
1 The Model Problem and the Basics of FEM	4
1.1 Analytical Solution of the Model Problem	4
1.2 Basics of FEM	5
1.3 FEM Analysis using Marc/Mentat	6
1.4 FEM Analysis using Paradox	8
1.5 Results of FEM Analysis of the Model Problem	8
2 Locking of the Model Problem	10
2.1 Results of FEM Analysis of the Model Problem with Varying Poisson's Ratio	10
2.2 Locking in FEM	12
3 Remedies for Locking	14
3.1 Reduced Integration and Selective Reduced Integration	14
3.2 Assumed Strain	15
3.3 Herrmann Formulation	15
3.4 Results of FEM Analysis of the Model Problem with Remedies for Locking	15
3.5 Remedies in Paradox	18
4 Conclusion	20
Literature	21
Appendix 1 Symbols used	22
Appendix 2 Analytical solution	23
Appendix 3 Matlab script for Paradox calculations	26
Appendix 4 Matlab script to generate a quarter circle mesh	29

Preface

In designing and engineering, extensive use is made of finite element methods in order to calculate heat transfer, stresses, deformation and so on. Although finite element methods can provide very good (approximations) of solutions to problems which couldn't be solved analytically, there are some situations for which problems arise in using finite element methods.

One of those situations occurs when dealing with mechanical problems where (nearly) incompressible materials are evaluated. It seems that for these materials, finite element methods produce results showing stiffness far greater than would be expected, rendering the results useless. This particular problem, which is called locking, is to be examined in this report.

To be able to learn something about results from finite element methods, first of all a model problem, for which there is an analytical solution, is introduced in the first chapter, as well as the basics of finite element methods. Finite element solutions using basic element types and various numbers of elements in Marc/Mentat are to be compared to the analytical solution to show that they indeed provide an accurate approximation. Further more; using the Paradox plug-in for Matlab, a finite element calculation is performed that is more insightful than using the Marc/Mentat calculation program.

In the second chapter, the actual problem of locking is treated. It is shown that increasing Poisson's ratio towards 0.5 (for incompressible behavior) leads to locking in both Marc/Mentat and Paradox analysis. Also a theoretical background of locking is given.

Finally, in the third chapter, a number of solutions to the problem that are available in Marc/Mentat are discussed. These solutions, however, can cause other unexpected behavior. When possible, these methods are also tested using Paradox.

1 The Model Problem and the Basics of FEM

In order to check experiments done with finite element methods (FEM), first a model problem is evaluated for which there exists an analytical solution. In this way, the results of the FEM calculations can be compared to this solution.

For the model problem, a situation is chosen in which a circular cylinder or ring of a thickness t , inner radius r_i and outer radius r_o is subjected to an internal pressure p . (Figure 1.1) The cylinder/ring is thought to be suspended in roller supports, so that a state of plane strain can be assumed. Goal of the calculations is to find the displacement u of a point on the inner radius caused by this internal pressure. Further (known) material properties are Elasticity modulus E and Poisson's ratio ν .

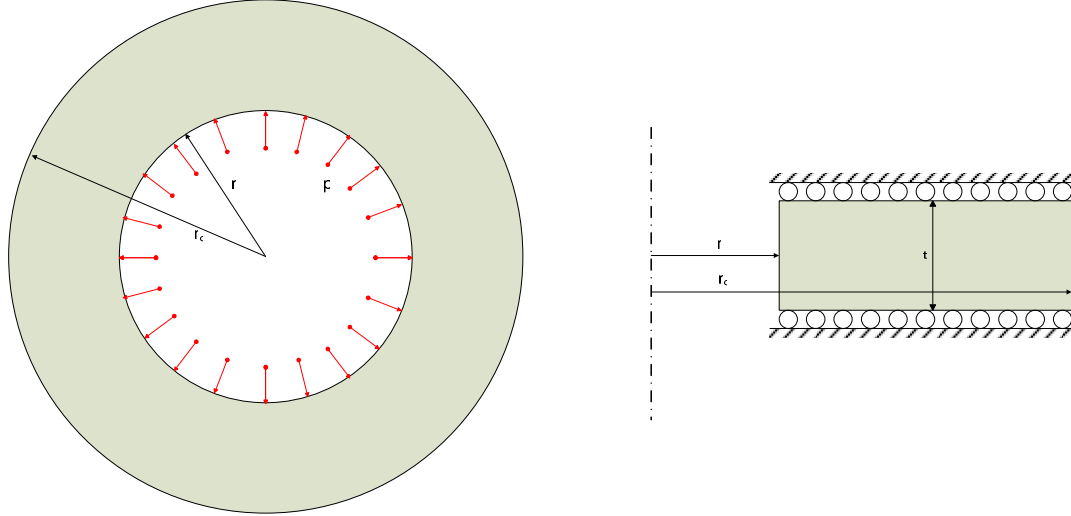


Figure 1.1 Model problem

1.1 Analytical Solution of the Model Problem

The analytical solution is calculated by means of three sets of equations [Geers]:

Equilibrium:

$$\frac{d\sigma_{rr}}{dr} + \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r} = 0 \quad (1.1)$$

Strain-displacement relations:

$$\begin{aligned} \varepsilon_{rr} &= \frac{du}{dr} \\ \varepsilon_{\theta\theta} &= \frac{u}{r} \\ \varepsilon_{zz} &= 0 \end{aligned} \quad (1.2)$$

Constitutive relations:

$$\begin{aligned} \sigma_{rr} &= \frac{E}{(1-\nu)(1-2\nu)} \left((1-\nu)\varepsilon_{rr} + \nu\varepsilon_{\theta\theta} \right) \\ \sigma_{\theta\theta} &= \frac{E}{(1-\nu)(1-2\nu)} \left(\nu\varepsilon_{rr} + (1-\nu)\varepsilon_{\theta\theta} \right) \end{aligned} \quad (1.3)$$

First of all, the strain-displacement relations are substituted in de constitutive equations, which in turn are substituted in de equilibrium equation. Further calculations, which can be found in Appendix 2, then yield the differential equation:

$$\frac{d^2u}{dr^2} + \frac{1}{r} \frac{du}{dr} - \frac{u}{r^2} = 0 \quad (1.4)$$

For this equation, a general solution of the following form exists (again, more detailed calculations can be found in Appendix 2):

$$u = C_1 r + C_2 \frac{1}{r} \quad (1.5)$$

To find the actual solution, two boundary values have to be introduced in order to compute C_1 and C_2 . In this case both the radial stresses (σ_{rr}) on the inner radius r_i and the outer radius r_o are known. On the inner radius σ_{rr} equals $-p$, on the outer radius σ_{rr} equals 0. Using these two conditions in the combined strain-displacement-constitutive equation for σ_{rr} (Appendix 2) yields the values of the integration constants:

$$\begin{aligned} C_1 &= \frac{pr_i^2(1+\nu)(-1+2\nu)}{E(r_i^2 - r_o^2)} \\ C_2 &= -\frac{pr_i^2 r_o^2(1+\nu)}{E(r_i^2 - r_o^2)} \\ u &= \frac{pr_i^2(1+\nu)(-1+2\nu)}{E(r_i^2 - r_o^2)} r - \frac{pr_i^2 r_o^2(1+\nu)}{E(r_i^2 - r_o^2)} \frac{1}{r} \end{aligned} \quad (1.6)$$

1.2 Basics of FEM

In finite element methods, as the phrase implies, the volume (3 dimensional) or the surface (2 dimensional) of a structure or part is divided into a number of elements, together called a mesh. An approximation of the solution to a differential equation and its boundary conditions is then calculated for each of these elements using a number of nodes and shape functions.

To be able to discuss some of the problems with FEM in the following, it is necessary to introduce the basics of FEM and its different matrices and procedures. This is done by means of the explanation of FEM by [Peerlings]. First of all, a model differential equation is stated:

$$\vec{\nabla} \cdot ({}^4\mathbf{C}(\vec{x}) : \vec{\nabla}\vec{u}) + \vec{f}(\vec{x}) = \vec{0} \quad (1.7)$$

In the line of the model problem from section 1.1, \mathbf{C} would be the constitutive relation, u the displacement and f would be a distributed body force, such as gravity. This equation is then multiplied by some unknown test function $\vec{\phi}(\vec{x})$ and then integrated over the domain, resulting in the following equation:

$$\int_V \left[\vec{\phi}(\vec{x}) \cdot \vec{\nabla} \cdot ({}^4\mathbf{C}(\vec{x}) : \vec{\nabla}\vec{u}) + \vec{\phi}(\vec{x}) \cdot \vec{f}(\vec{x}) \right] dV = 0 \quad (1.8)$$

If this formulation has to hold for every $\vec{\phi}(\vec{x})$, this is the same as stating equation(1.7). Using the product rule for divergence, the divergence theorem and the fact that the applied load on the boundary is defined as $\vec{q}(\vec{x}) = \vec{n}(\vec{x}) \cdot {}^4\mathbf{C}(\vec{x}) : \vec{\nabla}\vec{u}$, this formulation is transformed into the weak form of the equation. It's called the weak form because the differentiation demands on u (and its approximation) are less strict. In equation(1.7) and (1.8) u has to be differentiated two times, whereas in the weak form this only has to be performed one time. The weak form is:

$$\int_V (\vec{\nabla}\vec{\phi})^T : {}^4\mathbf{C}(\vec{x}) : \vec{\nabla}\vec{u} dV = \int_V \vec{\phi}(\vec{x}) \cdot \vec{f}(\vec{x}) dV + \int_S \vec{\phi}(\vec{x}) \cdot \vec{q}(\vec{x}) dS = 0 \quad (1.9)$$

In this, the left hand side denotes the increment in stored energy due to the applied virtual work that is on the right hand side. Next, the elements and their nodes come into play. The vector field $\vec{u}(\vec{x})$ and the test function $\vec{\phi}(\vec{x})$ are replaced by approximate fields over each element, computed by adding up each of the nodal values multiplied with the shape function:

$$\begin{aligned}\bar{u}^h(\bar{x}) &= \sum_{i=1}^n N_i(\bar{x})u_i = \underline{N}^T(\bar{x})\bar{u} \\ \bar{\phi}^h(\bar{x}) &= \sum_{i=1}^n N_i(\bar{x})\phi_i = \underline{N}^T(\bar{x})\bar{\phi}\end{aligned}\quad (1.10)$$

The shape functions thus transform the values at the nodes of an element into an approximate of the real field along the element, and make sure the approximate fields of two neighboring elements match. In the weak form, the fields are replaced by these approximates. Then, stating that this should hold for every $\bar{\phi}^h(\bar{x})$, this yields:

$$\underline{\mathbf{K}} \cdot \bar{u} = \bar{f} + \bar{q} \quad (1.11)$$

With:

$$\begin{aligned}\underline{\mathbf{K}} &= \int_{V^h} \bar{\nabla} \underline{N} \cdot {}^4 \mathbf{C}(\bar{x}) \cdot \bar{\nabla} \underline{N}^T dV \\ \bar{f} &= \int_{V^h} \underline{N}(\bar{x}) \bar{f}(\bar{x}) dV \\ \bar{q} &= \int_{S^h} \underline{N}(\bar{x}) \bar{q}(\bar{x}) dS\end{aligned}\quad (1.12)$$

The combination $\bar{\nabla} \underline{N}$ is referred to as the strain displacement tensor \mathbf{B} and $\underline{\mathbf{K}}$ is called the stiffness tensor in the case of a mechanical problem. The integrals in these expressions are calculated numerically by means of a set of Gauss integration points for each element. The contribution of each of these elements is then assembled into the total tensor and vectors, and finally, after partitioning according to the known and unknown parts of the displacement field \bar{u} and boundary condition \bar{q} , the system can be solved.

1.3 FEM Analysis using Marc/Mentat

FEM analysis can be carried out by means of a lot of different element types, differing in shapes and number of nodes. In this case, it is chosen only to examine two basic types of elements: a quadrilateral element using 4 nodes at the corners (quad4) and a quadrilateral element using 8 nodes: 4 at the corners and 4 additional nodes on the edges (quad8). Because a quad8 element has biquadratic shape functions (quadratic in both coordinates of the element) its approximation with the same number of elements will be better than that of a quad4 element, which uses bilinear shape functions to approximate the displacement field. The quad8 elements however use more computing power.

Apart from the choice in elements, there are two ways of setting up a FEM calculation of the displacement in the model problem. Because this problem is axisymmetric, it is elegant to model only a cross section of the ring, with the applied pressure perpendicular to the inner edge of the cross section. (Figure 1.2) Then, by making the calculation program using axisymmetric calculations, the problem can be calculated using much simpler geometry than would be the case for calculations in a plane strain fashion on the ring. In this way all elements can be made of the same size and shape and boundary conditions are easily placed in perpendicular coordinates.

However, for reasons made clear in the next paragraph, the model problem will also be calculated by modeling a quarter piece of the ring. (figure 1.3) Only a quarter ring is used because the full circle would require more elements and hence more computing power. Actually, even a smaller section than a quarter would be quite enough, but this would mean placing boundary conditions in awkward directions, whereas they can be placed in perpendicular coordinates when a quarter ring is used. Still, the applied pressure has to be placed along a curved edge, but this can easily be done in Marc/Mentat.

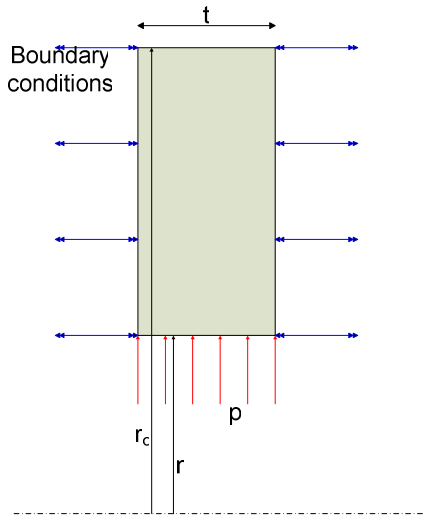


Figure 1.2 Axisymmetric geometry

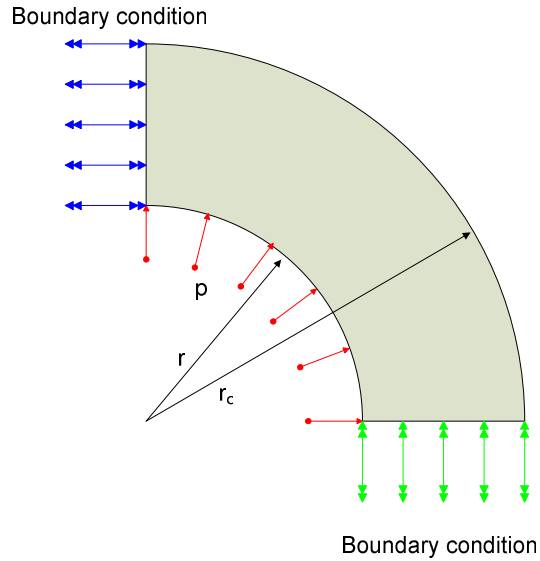


Figure 1.3 Plane strain geometry

Next, the geometries are subdivided into elements. For the axisymmetrical case, there are no applied forces in the direction parallel to the heart line (the boundary conditions are fixed displacements in this direction, equal to 0) so that in this direction only one element will suffice. The radial direction is the one where deformation takes place and the number of elements along this direction is steadily increased to show that the FEM analysis indeed converges to the analytical solution. An example of a FEM analysis output can be found in figure 1.4.

For the plane strain geometry, again the number of elements along the radial direction is increased to show convergence. However, for this geometry the number of elements in the tangential direction is also of some importance. If the number is too low, the circular edges of the geometry will be badly represented and the total surface of the mesh, and thus the amount of material, will differ from the actual geometry. So, for this case, an appropriate amount of elements in tangential direction is used (20) and a check is performed at 40 elements to make sure that the approximation doesn't converge a whole deal faster when increasing the number of elements in radial direction.

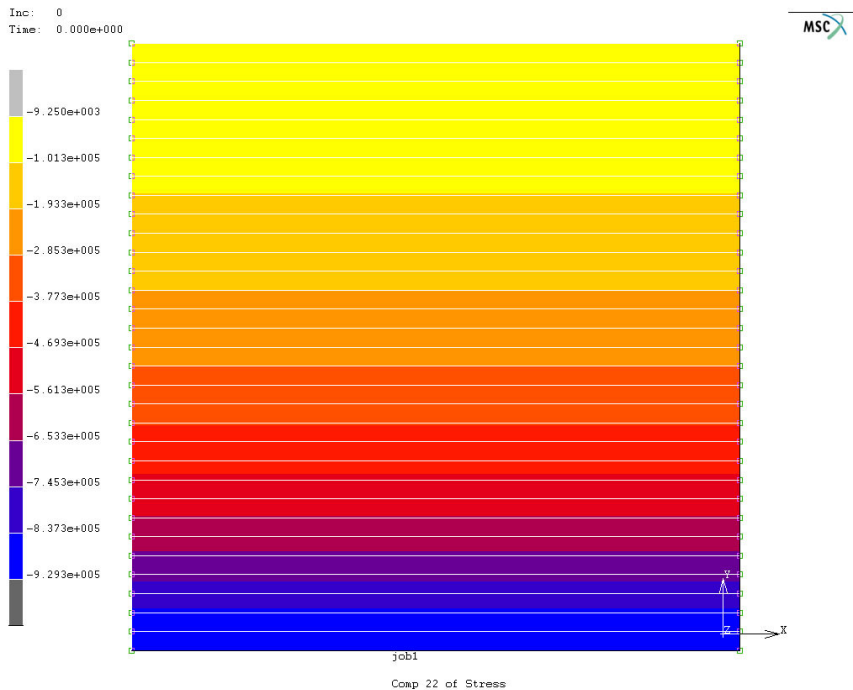


Figure 1.4 Example of FEM analysis output: de stress in radial direction for the axisymmetrical case

1.4 FEM Analysis using Paradox

Because a computer program like Marc/Mentat isn't very insightful, a FEM analysis is done by means of the Paradox plug-in in Matlab as well. This combination allows for relatively easy programming of the vector and tensor calculations needed for this kind of FEM, as well as providing a system for defining nodes, elements and meshes.

A problem occurs however when programming the axisymmetrical variant of the model problem. It seems that there exists no mathematical gradient operator in cylindrical coordinates in Paradox. This operator is needed to calculate the stiffness matrix in the axisymmetrical case and therefore the absence of it makes this a difficult procedure. Although it can be done either by comparing the Cartesian to the cylindrical situation and then adding the appropriate extra terms, or by defining the strain displacement tensor by hand, it is chosen not to do this.

This leaves only the plane strain situation to program, which is a relatively straightforward application of the known procedures of FEM. First a separate script is written to generate a quarter circle mesh of variable size using variable number of elements along both radial and tangential directions. This is done in order to be able to change the number of elements easily. Also information about the positions of the nodes on the geometry is generated: for partitioning of the calculations and applying boundary conditions, knowledge is needed about which nodes lie on which part of the boundary or the interior of the mesh.

From there on end it's just a matter of assembling the stiffness matrix with the appropriate constitutive relations, partitioning according to the boundary conditions and solving the system in a main script. It should be noted however, that the distributed load p should be converted to point loads at the nodes on the inner edge. This can introduce errors because of the calculation of the directions of these loads. Also the (under deformation) varying surface on which the distributed load p acts calls for equally varying point loads, instead of the constant forces used in the calculation.

An example of the results of a FEM analysis using Paradox is shown in figure 1.5, whereas both Matlab scripts can be found in Appendices 3 and 4.

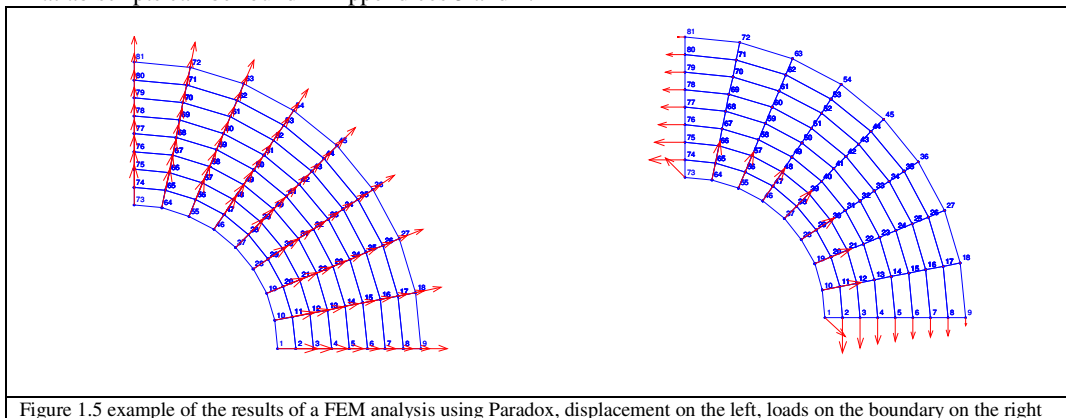


Figure 1.5 example of the results of a FEM analysis using Paradox, displacement on the left, loads on the boundary on the right

1.5 Results of FEM Analysis of the Model Problem

In order to create a dimensionless quantity in the order of 1 to represent results, the actual result of the calculations (the displacement of a point on r_i) is transformed into a dimensionless number for stiffness, S , by taking a dimensionless number for the applied load (p/E) and dividing it by a dimensionless number for the displacement (u/r_i). The resulting stiffness for varying numbers of elements in the radial direction can be found in figures 1.6 and 1.7.

In these results, clearly the difference in convergence between quad4 and quad8 can be seen. Whereas quad8 already shows a great level of convergence for 4 elements in this situation, for quad4 to reach the same level takes about 16 elements along the radial direction. Also, some differences in results for the different geometries can be spotted. It seems that the axisymmetric calculations perform much better than the plane strain calculations. This is most likely caused by the fact that the cross-section used in the axisymmetric calculations is exactly the same as the actual cross-section, unlike the quarter circle which slightly differs from the actual ring. Because the nodes of the mesh lie on the actual inner radius, the mesh has some excess material compared to the actual ring, making it a bit stiffer. The results for the plane strain calculations with twice as many elements along the tangential direction shows that this error indeed shrinks when more elements are used. For future calculations however, 20 elements along the tangential direction will be used in the Paradox calculation to save on calculation

time. For Marc/Mentat, the extra elements don't cause much extra computing time, so the 40 elements model can still be used in Marc/Mentat.

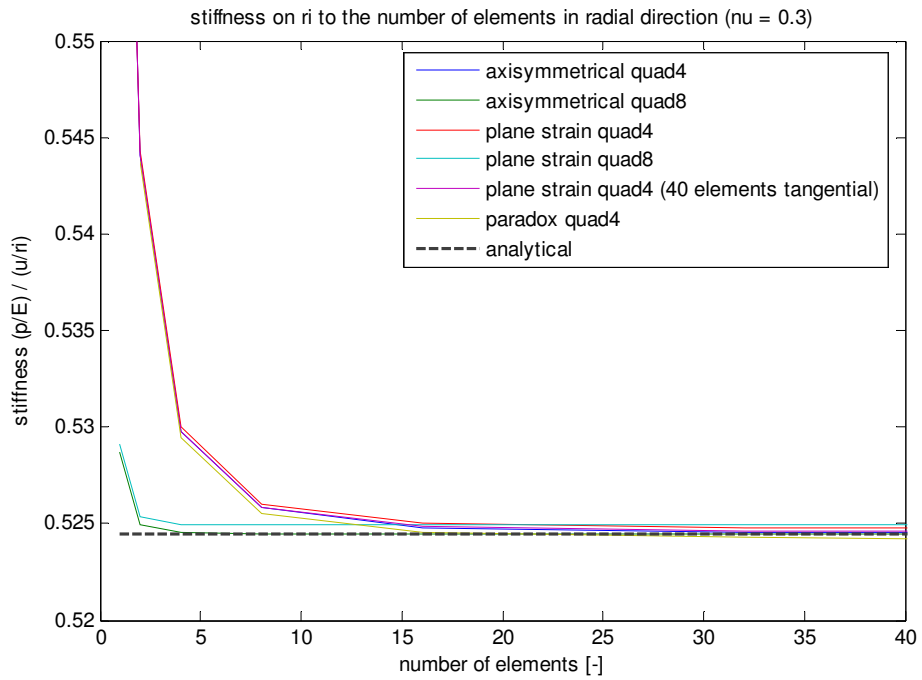


Figure 1.6 Stiffness for increasing number of elements

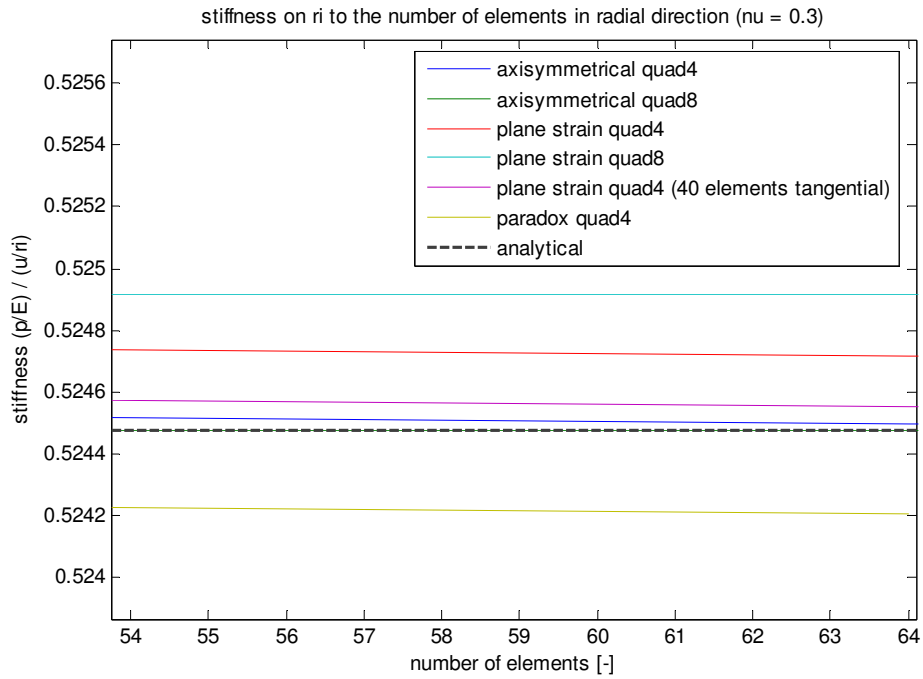


Figure 1.7 Stiffness for increasing number of elements, close up

2 Locking of the Model Problem

In the first chapter it was shown that Finite Element Methods can give a decent approximation to the simple problem that is used. However, there are situations for which the results provided by FEM are, in fact, useless. One of those situations occurs when dealing with incompressible or nearly incompressible materials, or in this context, when Poisson's ratio approaches 0.5.

A Poisson's ratio of 0.5, which means incompressible material, will immediately give erroneous results because Lamé's constant λ will become infinitely large. Therefore, only values approaching 0.5 will be tested in the following

2.1 Results of FEM Analysis of the Model Problem with Varying Poisson's Ratio

Like in the previous chapter, FEM analysis is performed on the model problem using both an axisymmetrical and a plane strain approach, using both quadrilateral elements with four and eight nodes. This time, the number of elements in radial direction is kept constant at 32 in order to keep calculation times of the Paradox analysis within reasonable limits. Again, the dimensionless parameter S is used, and plotted against Poisson's ratio. Results of these calculations can be seen in figures 2.1 to 2.3

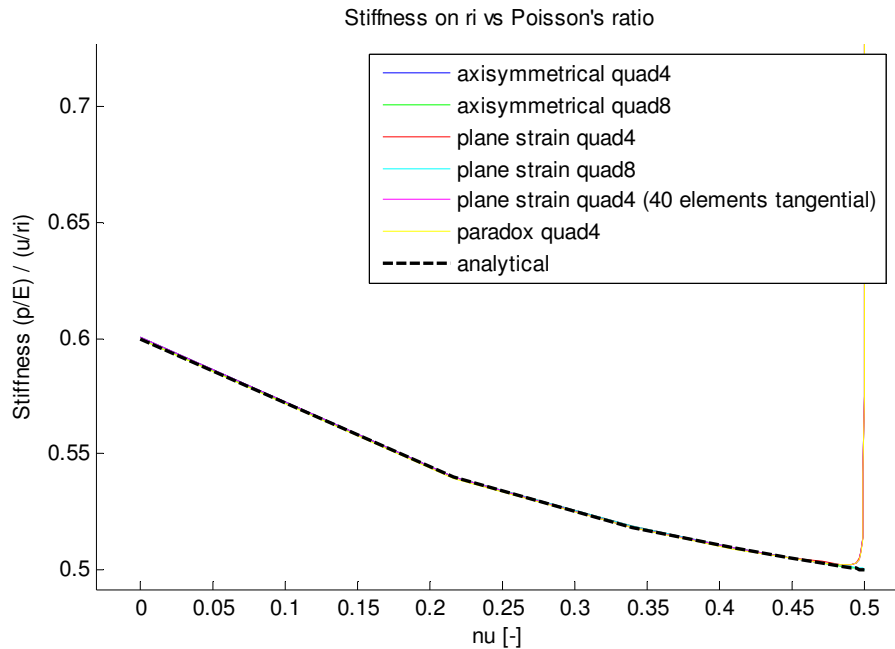


Figure 2.1 Stiffness on ri against Poisson's ratio ν

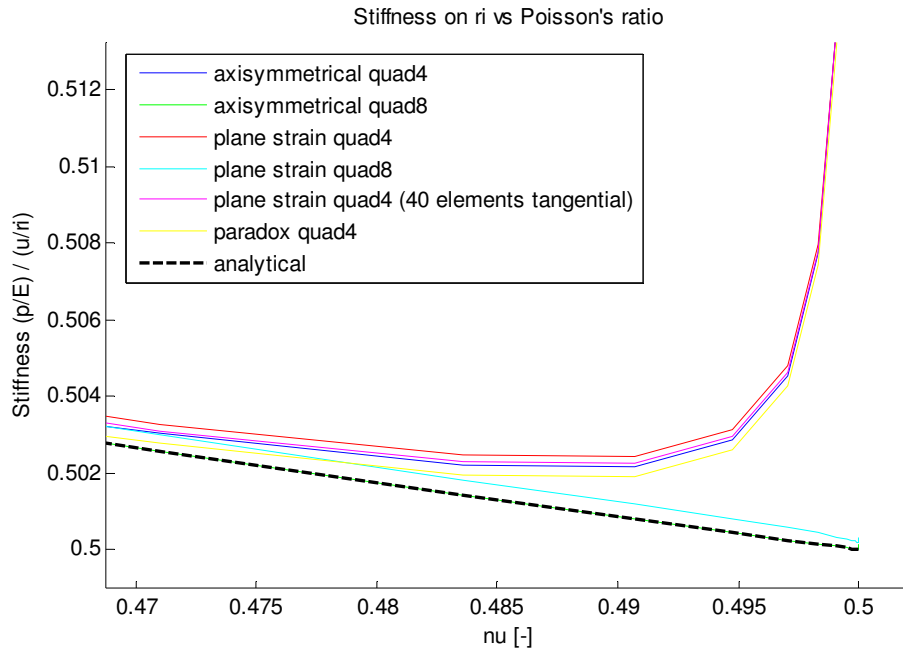


Figure 2.2 Same as 2.1, except further zoomed in on $\nu = 0.5$

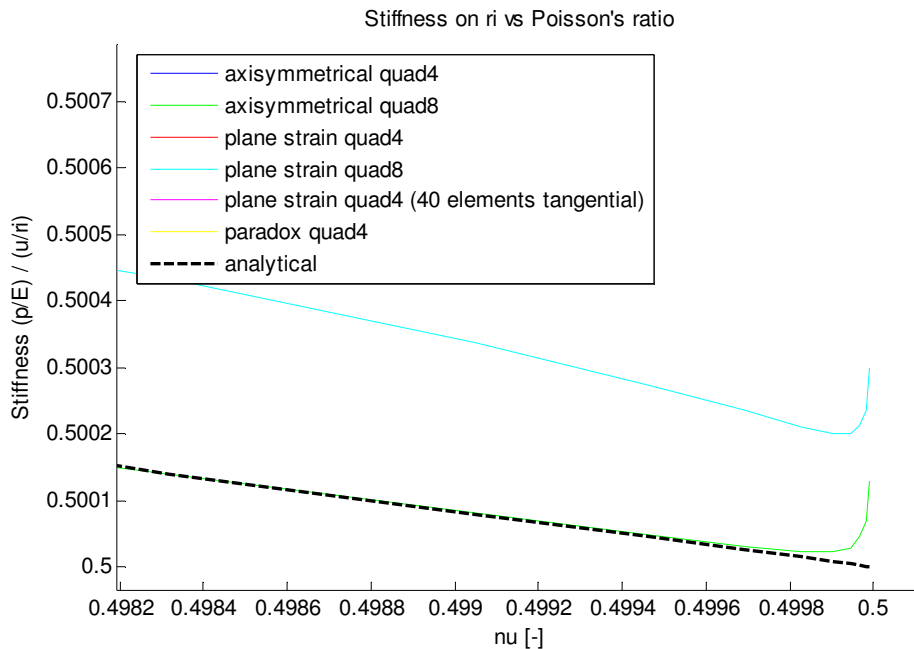


Figure 2.3 Same as 2.1 and 2.2, except further zoomed in on $\nu = 0.5$

Again, a large difference in the results for quad4 and quad8 elements exists. All the methods using standard quad4 elements show a huge increase in stiffness when approaching a Poisson's ratio of 0.5. In this case, results start to deviate from the analytical solution as early as Poisson's ratio is around 0.475 (Other calculations, not shown here, show that locking can be postponed if more elements are used in radial direction). From approximately 0.495, the error is larger than one percent and grows rapidly to become larger than 100 percent. This clearly shows the phenomenon of locking.

For the two calculations with quad8 elements, the approximations follow the trend of the analytical solutions for much higher values of Poisson's ratio, but for sufficiently high values, these two calculations also yield stiffness values that are too high. The effect however, seems to be very much smaller than for the quad4 elements and for the closest to 0.5 the calculations can get before rounding off ($\nu = 0.49999$), the error remains smaller than 0.2 percent.

Clearly, locking can cause for large errors in FEM analysis of nearly incompressible materials that render the results useless and one should be aware of this problem when dealing with these kinds of calculations. For example when performing plasticity calculations: in these calculations, elastic deformation is calculated with compressible material properties, but plastic deformation is usually approximated with (nearly) incompressible material properties. When nothing is done to prevent locking, results of these calculations will result in stresses that are too high or displacements that are too small.

2.2 Locking in FEM

Although the term ‘locking’ wasn’t used for this problem until the mid 1970’s, the problem itself was studied as far back as the 1960’s. At that time the problems that arose with locking of four node quadrilaterals and other lower order elements were combated with the development of higher order elements, such as eight node quadrilaterals. Techniques to prevent locking for lower order elements were developed from the 70’s and onwards. However, even today, when there are solutions to the problem, locking and its origins are not fully understood.

Locking can occur for a number of reasons and, for some element types, can even depend on the shape of an element. However, it seems that locking only happens when an element cannot interpolate a field property correctly with the nodal values and the element’s shape functions. [MacNeal]

MacNeal compares this interpolation failure to a similar problem in the field of data acquisition. When a sine signal is sampled with a sampling frequency that is too low, the frequency of the sine wave is measured to be lower than its actual frequency. In the output of the measurements, the representation of the sine wave at a lower frequency is called an alias. In the much the same way, a field property, in this case displacement can be wrongly represented due to an incorrect interpolation of the nodal values along the shape functions.

Take for example the 4 node quadrilateral element. Its shape functions are listed in table 2.1. When the displacement is of the form $u=\zeta^2$, its nodal values, u_i are (1,1,1,1) for nodes 1 to 4 respectively. Multiplying the nodal values with their appropriate shape functions and adding them up however results in the value of 1 for the displacement in the entire element, were ζ^2 was expected. This can be seen as the principal cause of locking, which is shown later.

In general, higher order elements have higher order shape functions and therefore can interpolate field properties of higher orders exactly, but are also limited in the same way.

Table 2.1 shape functions for the 4 node quadrilateral element

node	coordinates (ξ,η)	shape function
1	(-1,-1)	$(\xi-1)(\eta-1)/4$
2	(1,-1)	$-(\xi+1)(\eta-1)/4$
3	(1,1)	$(\xi+1)(\eta+1)/4$
4	(-1,1)	$-(\xi-1)(\eta+1)/4$

The quad4 element is troubled with two different kinds of locking. One of them is shear locking, which arises when the shear component is calculated by means of a wrongly interpolated displacement field that is prescribed to describe in plane bending using a plane stress formulation. The size of the error caused by this type of locking depends on the aspect ratio of the element (figure 2.4) and grows larger with increasing aspect ratios.

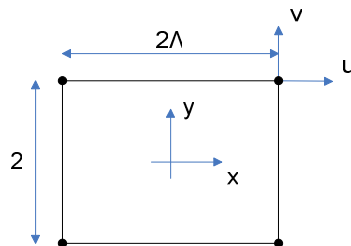


figure 2.4 aspect ratio (A)

The second type of locking, which is more important in this case, is called dilatation locking. Following the explanation of MacNeal, an example of in plane bending, plane strain, is reviewed. For this example, an assumed displacement state is stated for the element, which is stated in table 2.2. Also

in this table, the theoretical strains and in plane stresses are shown. On the right hand side however, the alias of the displacement state and its derived strains and in plane bending are shown.

table 2.2 Example of in plane bending, plane strain, showing dilatation locking

assumed displacement	alias of assumed displacement
$u = xy$ $v = -\frac{1}{2}x^2 - \frac{\nu}{2(1-\nu)}y^2$	$u^a = xy$ $v^a = -\frac{1}{2}\Lambda^2 - \frac{\nu}{2(1-\nu)}$
derived strains	derived strains
$\epsilon_x = y$ $\epsilon_y = -\frac{\nu}{1-\nu}y$ $\gamma_{xy} = 0$	$\epsilon_x = y$ $\epsilon_y = 0$ $\gamma_{xy} = x$
in plane stresses	in plane stresses
$\sigma_x = \frac{Ey}{1-\nu^2}$ $\sigma_y = 0$ $\tau_{xy} = 0$	$\sigma_x = \frac{E(1-\nu)y}{(1+\nu)(1-2\nu)}$ $\sigma_y = \frac{E\nu y}{(1+\nu)(1-2\nu)}$ $\tau_{xy} = \frac{Ex}{2(1+\nu)}$

Now, dilatation is defined as the volumetric expansion:

$$e = \epsilon_x + \epsilon_y + \epsilon_z \quad (2.1)$$

In case of the actual strain, the dilatation is $e = \epsilon_x + \epsilon_y + \epsilon_z = \left(\frac{1-2\nu}{1-\nu}\right)y$, which shrinks to zero when Poisson's ratio approaches 0.5, as would be expected from incompressible materials. However, in case of the quad4 element, the dilatation becomes $e = \epsilon_x + \epsilon_y + \epsilon_z = y$ which remains finite as Poisson's ratio approaches 0.5. When reviewing the strain energy density (i.e. a measure for the left hand side of the weak form of the differential equation, equation(1.9)) for both the theoretical and the element's case;

$$W'_s = \frac{1}{2} \int_V \{\sigma\}^T \{\epsilon\} dV = \frac{E}{2(1-\nu^2)} y^2$$

$$W_s'^a = \frac{E}{2(1-\nu^2)} \left[\left(\frac{1-\nu}{1-2\nu}\right)y^2 + \frac{x^2}{2} \right] y^2 \quad (2.2)$$

It shows that there are two sources for the error; the first is the x^2 term which arises from the wrong shear strain term. The second is the extra multiplication of the y^2 term with $(1-\nu)^2/(1-2\nu)$. This term grows very large when Poisson's ratio approaches 0.5. The source of this problem lies in the fact that the interpolation of y^2 doesn't produce the correct linear strain for ϵ_y . This also causes the dilatation to remain finite, and therefore this type of locking is defined as dilatation locking.

For the quadrilateral element with 8 nodes, it can be shown [MacNeal] that dilatational locking does not occur, but that a form of shear locking indeed is present.

3 Remedies for Locking

To be able to use FEM in fields where locking occurs with the basic element types, the past decades a lot of research was directed at finding solutions for the problem of locking. They exist in a wide variety and can have unwanted side effects of their own. In this report, which is driven by the remedies for locking used in Marc/Mentat, only those solutions will be discussed. The remedies used in Marc/Mentat represent to basic and most used remedies for locking, so that a study of them provides a basic insight in the number of solutions that have been developed over the past decades.

In Marc/Mentat, during different stages of modeling, options can be selected that provide a calculation that is not affected by locking. First, when selecting geometric properties, there exists one or two options: constant dilatation and assumed strain; where the second is not available when using the axisymmetrical geometry. And second, when choosing element types, again two options exist to prevent locking: reduced integration and Herrmann formulation. These four options will be discussed in the following sections.

Also, increasing the number of elements (while remaining the same shape in the case of shape sensitivity) can delay the effects of locking to values of Poisson's ratio closer to 0.5. [MacNeal] However, this is not a very desirable solution because it would take more computational power and would still not solve the problem completely.

3.1 Reduced Integration and Selective Reduced Integration

As was mentioned in section 2.2, locking can occur when a field property is represented by a wrong alias. This alias then gives erroneous values when numerical integration using Gauss-points is used, because the function value of the alias is different on those gauss points compared to the real field property.

Nevertheless, if the alias for in plane bending in a plane strain formulation (section 2.2) were to be integrated by means of a single Gauss point on (0, 0), the alias would indeed give the right results for this element. This reduction of used Gauss points is called reduced integration. In the case of the quad8 element, although in this case full integration requires a set of nine Gauss points and reduced would use 4, it can also be shown that reduced integration evaluates the alias for the shear stress correct, so also for quad8 elements, reduced integration can be a solution to locking.

Reduced integration can however cause unwanted behavior of the element. Because reduced integration reduces the rank of the total stiffness tensor, this tensor can become singular. This can for example give rise to modes that produce displacements, but require no forces. Take for example the element on the left in figure 3.1. When deformed into the element on the right, both the strain in x and y direction, as well as shear strain, would remain zero when evaluated at a single integration point at (0, 0). Hence, this structure can deform in this way without causing any stresses in the material. This kind of behavior is called a spurious mode.

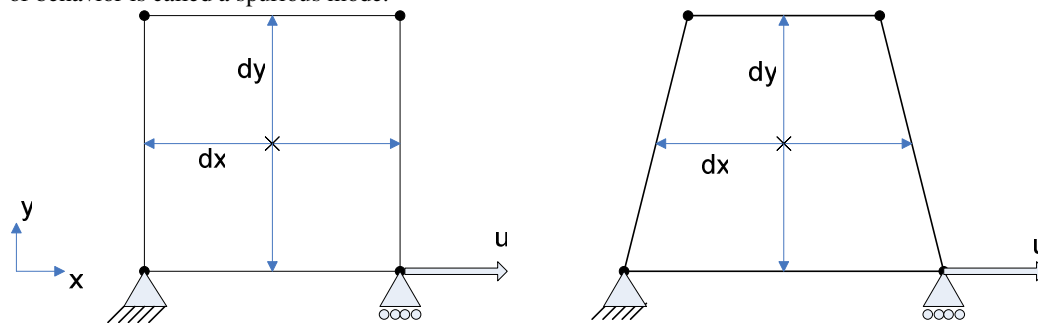


figure 3.1 spurious mode in a quad4 element using reduced integration

When this particular mode is encountered in a mesh of multiple elements, this is called hour glassing or chicken wiring, because of its shape. Because hour glassing is the most common of all spurious modes, sometimes they are all referred to as hour glassing.

Of course, spurious modes are unwanted and result in erroneous solutions of the FEM analysis. In Marc/Mentat, hour glassing in reduced integration is prevented by altering the stiffness tensor according to principles laid out by Hu-Washizu [Marc/Mentat]. Reduced integration can be selected while setting the element type in the job options menu.

Another way to prevent spurious modes from happening is not to integrate the entire constitutive relation with reduced integration, but only the parts that would otherwise cause locking. To prevent shear locking for example, only the part of the constitutive relation that calculates the shear component has to be evaluated at a single Gauss point, the rest can be dealt with by a full integration scheme. This way, the rank of the stiffness tensor remains full, preventing spurious modes. This kind of reduced integration is called selective reduced integration.

In the case of dilatation locking, also a selective reduced integration can be used. In this case a split is made between deviatoric strains and stresses, and dilatational strains and stresses. The dilatational part consists of a change in volume only, whereas the deviatoric part only changes the shape of the element. In this case, the dilatational part is evaluated with a reduced integration scheme, and the deviatoric part with a full integration scheme, so that the dilatation is correctly evaluated to be zero in the limit of $\nu = 0.5$, and constant over the element for lower values of Poisson's ratio.

In Marc/Mentat, only selective reduced integration for dilatation locking exists. It can be selected in the geometry options menu, by checking the constant dilatation box.

3.2 Assumed Strain

Another way to prevent locking is to modify the strains that are computed from the displacements before integration takes place. Normally, the strain vector for an element is calculated by taking the derivative of the displacement field calculated by means of the shape functions and nodal values of displacement: $\{\boldsymbol{\varepsilon}\} = \bar{\mathbf{V}}\mathbf{N}\{u_i\} = \mathbf{B}\{u_i\}$. In order to generate strains of an assumed form $\{\hat{\boldsymbol{\varepsilon}}\}$, \mathbf{B} is substituted with a modified version: $\{\hat{\boldsymbol{\varepsilon}}\} = \bar{\mathbf{B}}\{u_i\}$

How this $\bar{\mathbf{B}}$ is formed depends on the form of the strain field that is assumed to exist in the problem, and on which point the strains calculated with \mathbf{B} are chosen to be equal to those calculated with $\bar{\mathbf{B}}$.

To prevent dilatational locking, a split in deviatoric and dilatational parts of the problem is possible. The dilatational part of \mathbf{B} is then replaced with a modified version, generating $\bar{\mathbf{B}} = \mathbf{B}^{dev} + \bar{\mathbf{B}}^{dil}$ [Hughes]. These types of methods are called B-bar methods.

It is also possible to assume a strain field independently of the displacement field, by multiplying a row of basic functions by a set of unknown coefficients. This assumed displacement field is then equated to the strain field calculated by means of the displacement field in order to calculate the coefficients. If this displacement field was generated from the shape functions, i.e. also of an 'assumed' form, this method is called a hybrid method.

In Marc/Mentat, the option for an assumed strain field is also available at the geometry options menu, but only for the plane strain method of the model problem. In Marc/Mentat documentary only little is said about this assumed strain option, and it isn't clear if this is some sort of B-bar method or maybe a hybrid method.

3.3 Herrmann Formulation

The last option that is available in Marc/Mentat is the Herrmann formulation, which is available while setting the element type. This method involves taking the pressure on an element as an extra variable next to the displacement, creating a two-field problem [Zienkiewicz & Taylor]. This is beyond the scope of this project and will therefore not be treated, although experiments with this type of calculations have been done and their results are shown in the next section.

3.4 Results of FEM Analysis of the Model Problem with Remedies for Locking

Following the same experiment set up as in chapter 2, experiments have been made using the locking remedies available in Marc/Mentat. This time, due to the number of models, data is collected separately for the axisymmetrical case and the plane strain case, and also for quad4 and quad8. The relevant data (close to $\nu = 0.5$) is shown in figures 3.2 to 3.5

In these results, it seems that after removing the excessive dilatation locking in the quad4 computations, no further locking exists. Indeed, for the axisymmetrical case, a reduced integration

scheme results in an approximation that is virtually the same as the analytical solution. For plane strain calculation, an assumed strain method provides the same outcome as one with a reduced integration scheme, but the difference in precision between axisymmetrical and plane strain off course still remains.

For the analysis using a quad8 element, assumed strain and constant dilatation do not provide a solution for the minor locking problem, as would be expected. These two methods are, at least in Marc/Mentat, designed to cope with dilation locking, which doesn't occur in a quad8 element. Probably, this element suffers from the shear locking mentioned in chapter 2. Reduced integration is a working remedy here, as is the Herrmann formulation. However, the error induced by reduced integration and the Herrmann formulation in the plane strain approach, is greater the error produced by locking. In case of the axisymmetrical approach, they both give good results.

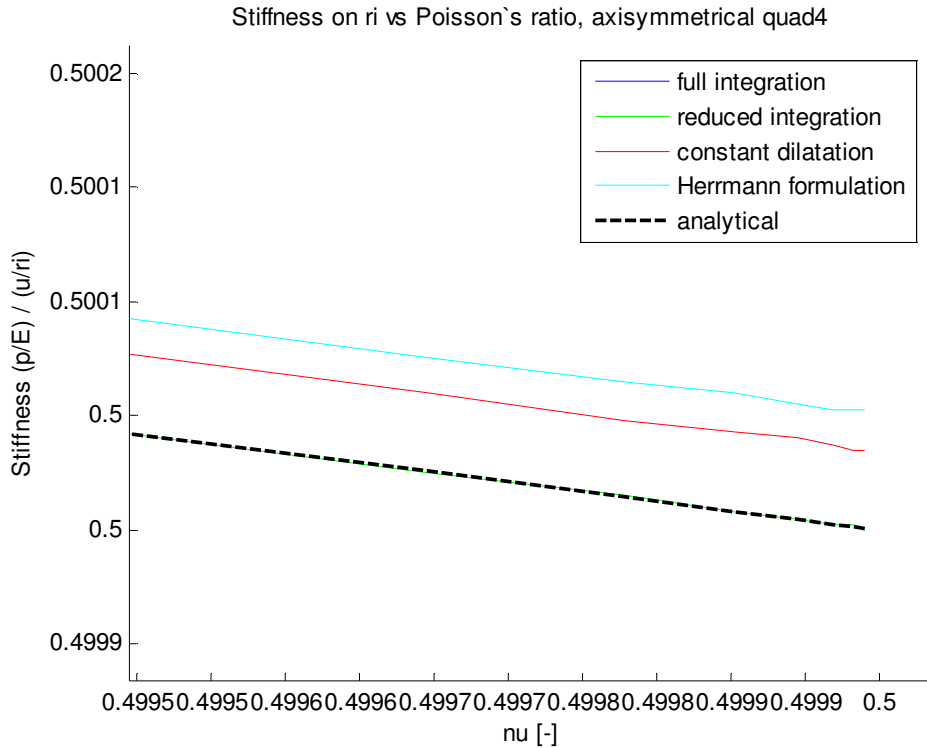


figure 3.2 FEM analysis results for axisymmetrical quad4, using locking remedies (shown is only the domain very close to $\nu = 0.5$, so the line displaying full integration falls outside this graph)

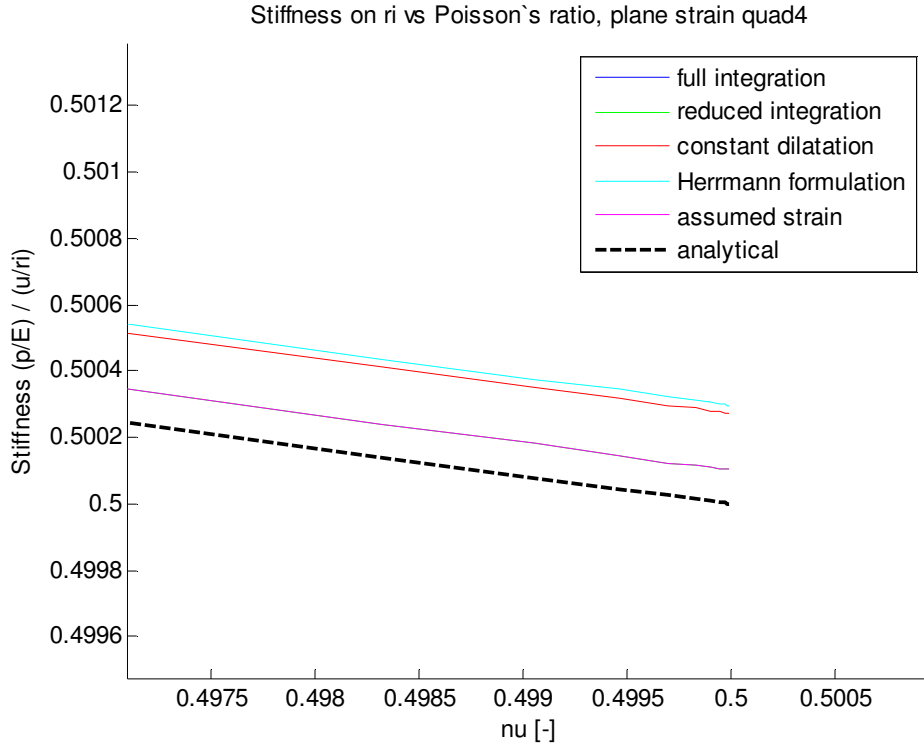


Figure 3.3 FEM analysis results for plane strain quad4, using locking remedies (shown is only the domain very close to $\nu = 0.5$, so the line displaying full integration falls outside this graph. Also, the line for reduced integration is not visible because of the line for assumed strain)

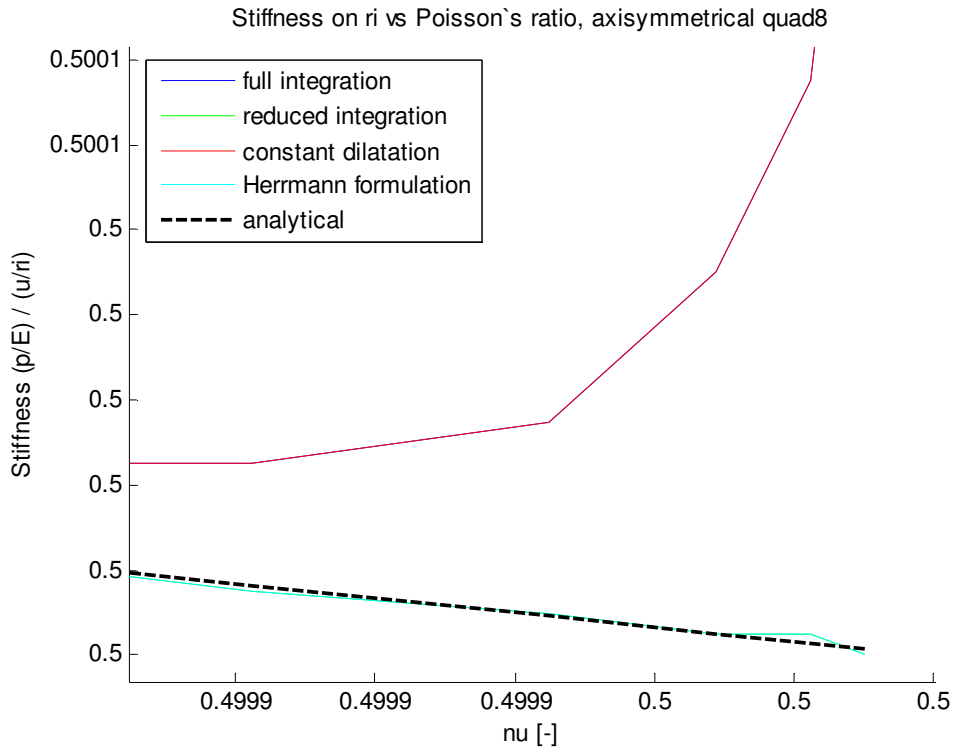


Figure 3.4 FEM analysis results for axisymmetrical quad8, using locking remedies (the line for full integration is not visible because of the line for constant dilatation and the line for reduced integration is lost behind the one for Herrmann formulation)

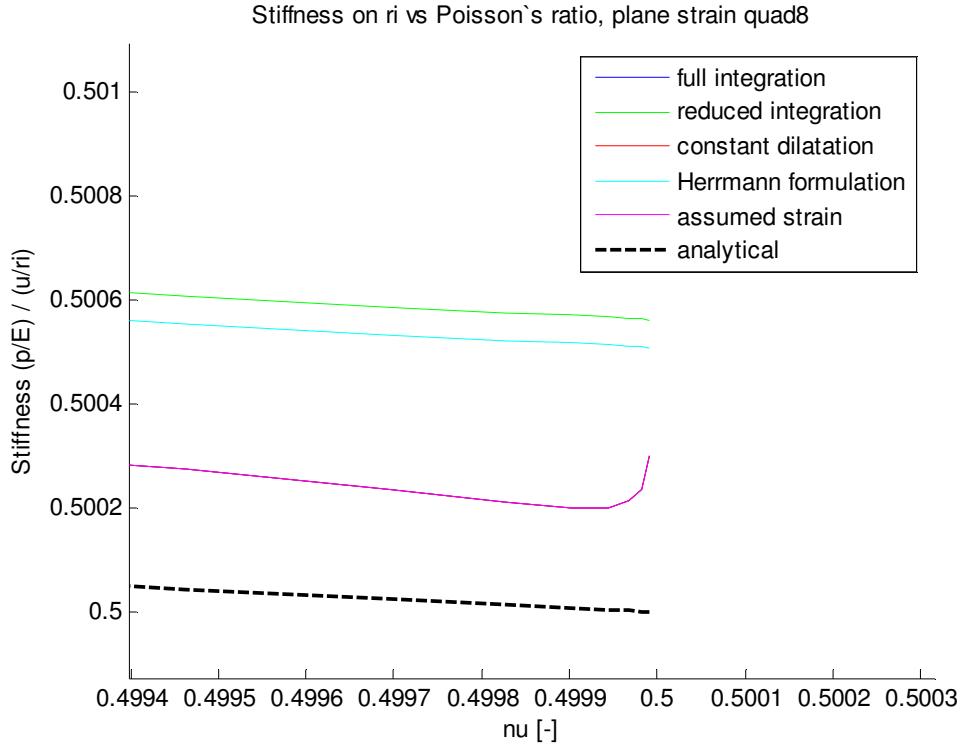


Figure 3.5 FEM analysis results for plane strain quad8, using locking remedies (The lines for full integration and constant dilatation are not visible because of the line for assumed strain)

3.5 Remedies in Paradox

The final challenge in this project is to implement these different locking remedies into the Paradox calculations. However, this is more easily said than done. Of course it is easy to use a reduced integration scheme, but without an extra adaptation to prevent spurious modes, this won't produce any results other than erratic displacements or simply an error because the stiffness tensor is singular.

However, a selective reduced integration of the form mentioned in [Hughes, 4.4] is easily implemented. Reduced integration of this form evaluates the λ part of the constitutive relation using reduced integration, and the μ part is fully integrated. The constitutive relation in the Paradox computation can be split into two parts using this division, thus creating a selective reduced integration.

As for the assumed strain method or B-bar method, it seems that implementation is not that easy because it requires formulating $\bar{\nabla} \bar{N}$ by hand, whereas the Paradox environment normally takes care of this. Because it seems that there is no way to create a tensor using symbolic variables, writing a script for these methods would take more investigating of Paradox, and more time. Time however was limited and therefore, no working Paradox script was written using these methods.

Results for the selective reduced integration are shown in figure 3.6 and 3.7. From this graph it is clear that this kind of selective reduced integration renders good results: locking is indeed prevented, while still remaining very close to the analytical solution. The error of the Paradox approximation being less stiff than the analytical solution was already present in previous results.

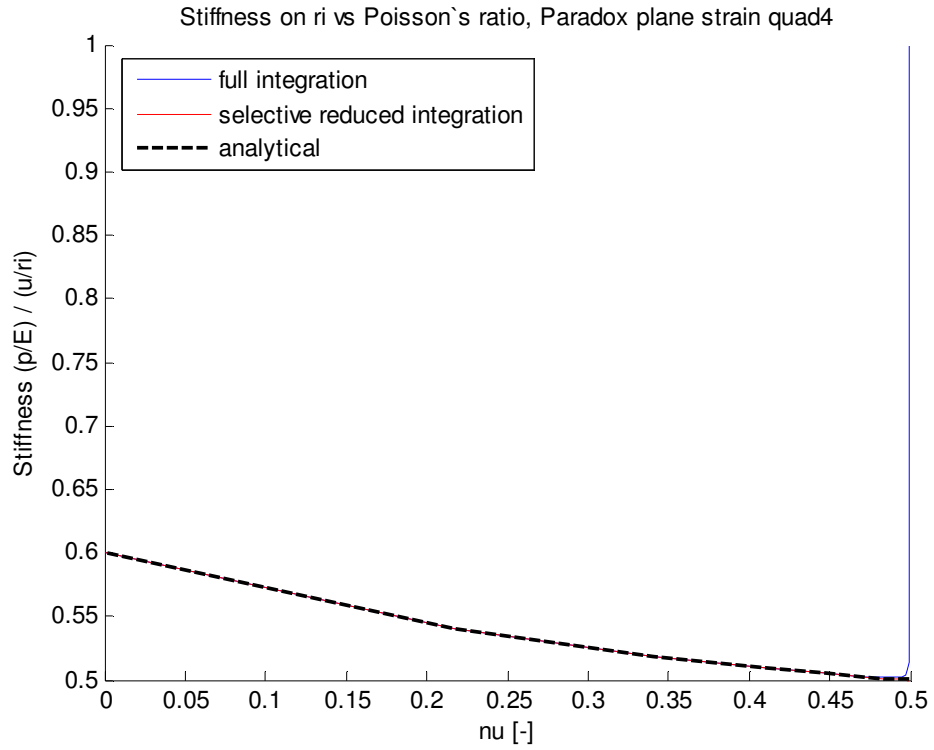


Figure 3.6 FEM analysis using selective reduced integration in Paradox

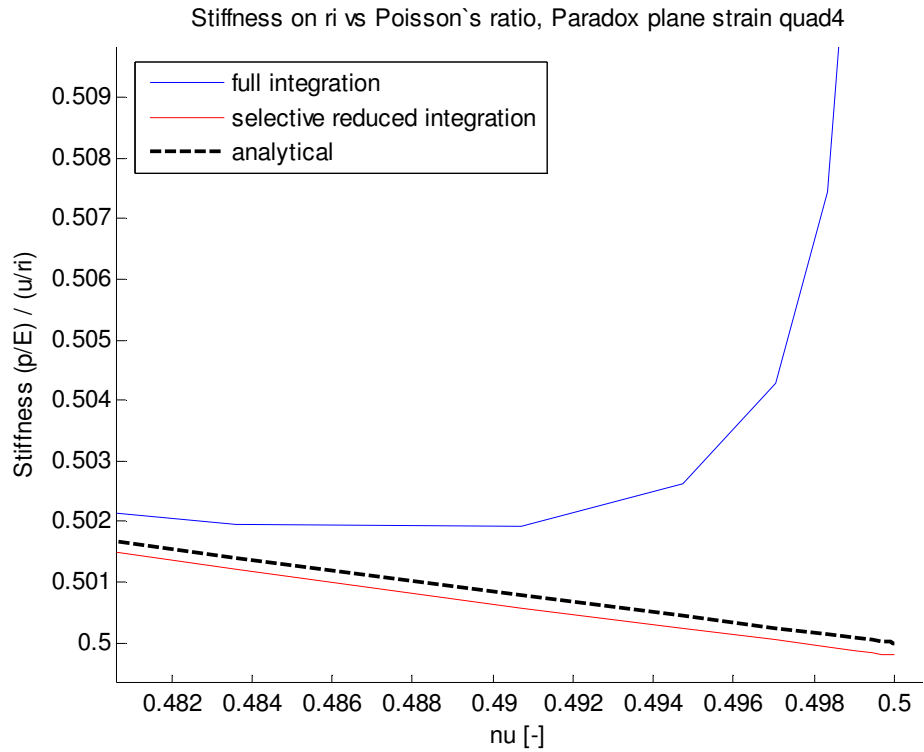


Figure 3.7 FEM analysis using selective reduced integration in Paradox, close up

4 Conclusion

In conclusion, it has become clear that interpolation failure in finite elements can lead to unexpected and unwanted behavior. In this case interpolation failure caused excessive stiffness of the simple 4 node quadrilateral and, though be it far less, 8 node quadrilateral elements when dealing with values of Poisson's ratio close to 0.5. This behavior of excessive stiffness was called locking. It was shown by means of literature that different kinds of locking exist, of which dilatation locking (present in the quad4 element) leads to the greatest increase in stiffness when approaching incompressible material properties.

Because reliable results for (nearly) incompressible results are indeed wanted, measures have been developed to prevent locking. Many different solutions have been designed over the past decades, some of which (constant dilatation, assumed strain) are specialized in preventing dilatation locking and others (reduced integration, selective reduced integration, Herrmann formulation) can also fix other types of locking.

Finally, effort was made to implement these remedies for locking in a Paradox script. It seemed however that for the most methods more understanding of both Paradox and the locking remedies was needed to be able to do this. Selective reduced integration was the only remedy that was successfully tried in Paradox, preventing the locking phenomenon and giving good approximation results.

Literature

[Geers] Geers, M.G.D. (2004). *Applied Elasticity in Engineering; Toegepaste elasticiteitsleer; Lecture notes - course 4A450*. Eindhoven, TU/e.

[Hughes] Hughes, Thomas J.R. (1987). *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. New Jersey: Prentice-Hall.

[Marc/Mentat] (2001). *MSC.Marc Volume A: Theory and User Information, version 2001 & Volume B: Element library, version 2001*. Santa Ana, MSC.Software Corporation.

[MacNeal] MacNeal, Richard H., Faulkner, L.L. (Ed.) (1994). *Finite Elements: their Design and Performance*. New York, Marcel Dekker, Inc.

[Peerlings] Peerlings, Ron. *Finite Element Method; Lecture notes for the course 'Eindige-elementenmethode' (4A700)*. Eindhoven, TU/e.

[Zienkiewicz & Taylor] Zienkiewicz, O.C., Taylor, R.L. (2000), *The Finite Element Method; Volume 1; The Basis*, fifth edition, Oxford, Butterworth-Heinemann.

Appendix 1 Symbols used

Symbol	Quantity	Unit	Abbreviation unit
η, ζ, x, y	distance	meter	m
e	dilatation	[-]	[-]
E	elasticity modulus	Newton per square meter	N/m ²
ε, γ	strain	[-]	[-]
λ, μ	Lamé's constants	Newton per square meter	N/m ²
A	aspect ratio	[-]	[-]
ν	Poisson's ratio	[-]	[-]
p	pressure	Newton per square meter	N/m ²
r	radius	meter	m
S	dimensionless stiffness	[-]	[-]
σ, τ	stress	Newton per square meter	N/m ²

Appendix 2 Analytical solution

Equilibrium:

$$\frac{d\sigma_{rr}}{dr} + \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r} = 0$$

Strain-displacement relations:

$$\varepsilon_{rr} = \frac{du}{dr}$$

$$\varepsilon_{\theta\theta} = \frac{u}{r}$$

$$\varepsilon_{zz} = 0$$

Constitutive relations:

$$\sigma_{rr} = \frac{E}{(1-\nu)(1-2\nu)} \left((1-\nu)\varepsilon_{rr} + \nu\varepsilon_{\theta\theta} \right)$$

$$\sigma_{\theta\theta} = \frac{E}{(1-\nu)(1-2\nu)} \left(\nu\varepsilon_{rr} + (1-\nu)\varepsilon_{\theta\theta} \right)$$

Substitution of strain-displacement relations in constitutive relations:

$$\sigma_{rr} = \frac{E}{(1-\nu)(1-2\nu)} \left((1-\nu)\frac{du}{dr} + \nu\frac{u}{r} \right)$$

$$\sigma_{\theta\theta} = \frac{E}{(1-\nu)(1-2\nu)} \left(\nu\frac{du}{dr} + (1-\nu)\frac{u}{r} \right)$$

Substitution in equilibrium:

$$\frac{d}{dr} \left(\frac{E}{(1-\nu)(1-2\nu)} \left((1-\nu)\frac{du}{dr} + \nu\frac{u}{r} \right) \right) + \frac{1}{r} \left(\frac{E}{(1-\nu)(1-2\nu)} \left((1-\nu)\frac{du}{dr} + \nu\frac{u}{r} \right) - \frac{E}{(1-\nu)(1-2\nu)} \left(\nu\frac{du}{dr} + (1-\nu)\frac{u}{r} \right) \right) = 0$$

$$\frac{E}{(1-\nu)(1-2\nu)} \left(\frac{d}{dr} \left((1-\nu)\frac{du}{dr} + \nu\frac{u}{r} \right) + \frac{1}{r} \left(\left((1-\nu)\frac{du}{dr} + \nu\frac{u}{r} \right) - \left(\nu\frac{du}{dr} + (1-\nu)\frac{u}{r} \right) \right) \right) = 0$$

$$(1-\nu)\frac{d^2u}{dr^2} + \nu\frac{d}{dr}\frac{u}{r} + \frac{(1-\nu)}{r}\frac{du}{dr} + \nu\frac{u}{r^2} - (1-\nu)\frac{u}{r^2} - \frac{\nu}{r}\frac{du}{dr} = 0$$

$$(1-\nu)\frac{d^2u}{dr^2} + \nu\left(\frac{1}{r}\frac{du}{dr} - \frac{u}{r^2}\right) + \frac{(1-\nu)}{r}\frac{du}{dr} + \nu\frac{u}{r^2} - (1-\nu)\frac{u}{r^2} - \frac{\nu}{r}\frac{du}{dr} = 0$$

$$(1-\nu)\frac{d^2u}{dr^2} + \frac{\nu}{r}\frac{du}{dr} - \nu\frac{u}{r^2} + \frac{(1-\nu)}{r}\frac{du}{dr} + \nu\frac{u}{r^2} - (1-\nu)\frac{u}{r^2} - \frac{\nu}{r}\frac{du}{dr} = 0$$

$$(1-\nu)\frac{d^2u}{dr^2} + \frac{(1-\nu)}{r}\frac{du}{dr} - (1-\nu)\frac{u}{r^2} = 0$$

$$\boxed{\frac{d^2u}{dr^2} + \frac{1}{r}\frac{du}{dr} - \frac{u}{r^2} = 0}$$

General solution DV:

$$\begin{aligned}
 u &= Cr^\lambda \xrightarrow{\frac{d}{dr}} \frac{du}{dr} = \lambda Cr^{\lambda-1} \xrightarrow{\frac{d}{dr}} \frac{d^2u}{dr^2} = \lambda(\lambda-1)Cr^{\lambda-2} \\
 \frac{d^2u}{dr^2} + \frac{1}{r} \frac{du}{dr} - \frac{u}{r^2} &= 0 \rightarrow \lambda(\lambda-1)Cr^{\lambda-2} + \frac{1}{r} \lambda Cr^{\lambda-1} - \frac{1}{r^2} Cr^\lambda = 0 \\
 \lambda(\lambda-1)Cr^{\lambda-2} + \lambda Cr^{\lambda-2} - Cr^{\lambda-2} &= 0 \\
 Cr^{\lambda-2} (\lambda(\lambda-1) + \lambda - 1) &= 0 \\
 \lambda(\lambda-1) + \lambda - 1 &= 0 \\
 \lambda^2 &= 1 \\
 \lambda &= \pm 1
 \end{aligned}$$

$$\boxed{u = C_1 r + C_2 \frac{1}{r}; \frac{du}{dr} = C_1 - C_2 \frac{1}{r^2}}$$

Boundary values:

$$\begin{aligned}
 r = r_i \rightarrow \sigma_{rr} &= -p \\
 \frac{E}{(1-\nu)(1-2\nu)} \left((1-\nu) \frac{du}{dr} + \nu \frac{u}{r_i} \right) &= -p \\
 \frac{E}{(1-\nu)(1-2\nu)} \left((1-\nu) \left(C_1 - C_2 \frac{1}{r_i^2} \right) + \nu \frac{\left(C_1 r + C_2 \frac{1}{r_i} \right)}{r_i} \right) &= -p \\
 -\frac{E(C_1 r_i^2 - C_2 + 2\nu C_2)}{(1+\nu)(-1+2\nu)r_i^2} &= -p \\
 C_1 &= \frac{p(1+\nu)(-1+2\nu)}{E} - \frac{C_2(-1+2\nu)}{r_i^2}
 \end{aligned}$$

$$\overline{r = r_o \rightarrow \sigma_{rr} = 0}$$

$$\frac{E}{(1-\nu)(1-2\nu)} \left((1-\nu) \frac{du}{dr} + \nu \frac{u}{r_o} \right) = 0$$

$$\frac{E}{(1-\nu)(1-2\nu)} \left((1-\nu) \left(C_1 - C_2 \frac{1}{r_o^2} \right) + \nu \frac{\left(C_1 r + C_2 \frac{1}{r_o} \right)}{r_o} \right) = 0$$

$$-\frac{E(C_1 r_o^2 - C_2 + 2\nu C_2)}{(1+\nu)(-1+2\nu)r_o^2} = 0$$

$$C_2 = -\frac{C_1 r_o^2}{-1+2\nu}$$

$$\boxed{C_2 = -\frac{pr_i^2 r_o^2 (1+\nu)}{E(r_i^2 - r_o^2)}$$

$$C_1 = \frac{pr_i^2 (1+\nu)(-1+2\nu)}{E(r_i^2 - r_o^2)}$$

Appendix 3 Matlab script for Paradox calculations

%This is a stand alone version of the Matlab script written to perform FEM
%using Paradox on a pressurised cilinder. Also needed is the script
%kwardcirkelmesh.m to generate a mesh of a quarter cirkel with a variable
%number of elements. Also, a function version exists where one can input
%the variables by hand.

```
clear all
close all
startparadox %load the paradox environment

% parameters
Ri = 0.1; %internal radius [m]
Ro = 0.2; %outer radius [m]
p = 1e6; %internal pressure [Pa]
po = 0; %external pressure [Pa]
E = 1e11; %modulus of elasticity [Pa]
nu = 0.3; %Poisson's ratio [-]
elementenradiaal = 8; %number of elements in radial direction [-]
elemententangential = 8; %number of elements in tangential direction [-]

% calculating forces on nodes due to pressure
Fi = p*pi/2/elemententangential*Ri;
Fo = po*pi/2/elemententangential*Ro;

% elasticity tensor
lambda = ( E * nu ) / ( (1+nu) * (1-2*nu) );
mu = E / ( 2 * (1+nu) );
c = zeros(2, 2, 2, 2);
c(1, 1, 1, 1) = lambda + 2*mu;
c(2, 2, 2, 2) = lambda + 2*mu;
c(1, 1, 2, 2) = lambda;
c(2, 2, 1, 1) = lambda;
c(1, 2, 1, 2) = mu;
c(1, 2, 2, 1) = mu;
c(2, 1, 1, 2) = mu;
c(2, 1, 2, 1) = mu;
C = Tensor42(c);

% construct mesh
[Vh, NDSinneredge, NDSouteredge, NDShorcut, NDSvercut, NDSinnerhor, NDSinnerver, NDSouterhor, NDSouterver,
NDScentre] = kwartcirkelmesh(Ri,Ro,elemententangential,elementenradiaal);
plot(Vh)
drawnow
m = numberOfElements(Vh);
n = numberOfNodes(Vh);

% master element
Q = Quad( [ Node(Point2([-1 -1]), 1)
           Node(Point2([+1 -1]), 2)
           Node(Point2([+1 +1]), 3)
           Node(Point2([-1 +1]), 4) ] );
Ne = ShapeFunctions(Q);
gradxiNe = grad(Ne);
xi = [ Point2([-1/sqrt(3) -1/sqrt(3)])
       Point2([+1/sqrt(3) -1/sqrt(3)])
       Point2([+1/sqrt(3) +1/sqrt(3)])
       Point2([-1/sqrt(3) +1/sqrt(3)]) ];
w = [ 1 1 1 1 ]';

% build stiffness matrix
K = ZeroTensor22(n, n);
for e = 1:m
    xe = posVector( nodes(element(Vh, e)) );
    J = dyadicProduct(gradxiNe', xe);
    gradNe = inv(J) * gradxiNe;

    Ke = ZeroTensor22(4, 4);
    for k = 1:4
        Ke = Ke + w(k) * feval( gradNe * C * gradNe' * det(J) , xi(k) );
    end
end
```

```

% assembly
ii = nodeTags(element(Vh, e));
K(ii, ii) = K(ii, ii) + Ke;

end

% expand system into components
K = components(K);

% partition system

%inner edge: flux in horizontal and vertical direction prescribed
for i = 1 : length( NDSinneredge )
theta = pi / 2 / elemententangential * i ;
iif_inner = sort( [ 2 * NDSinneredge 2 * NDSinneredge - 1 ] );
qf_inner( 2 * i - 1 ) = Fi * cos( theta ); %horizontal flux
qf_inner( 2 * i ) = Fi * sin( theta ); %vertical flux
end

%outer edge: flux in horizontal and vertical direction prescribed
for i = 1 : length( NDSouteredge )
theta = pi / 2 / elemententangential * i ;
iif_outer = sort( [ 2 * NDSouteredge 2 * NDSouteredge - 1 ] );
qf_outer( 2 * i - 1 ) = Fo * cos( theta ); %horizontal flux
qf_outer( 2 * i ) = Fo * sin( theta ); %vertical flux
end

%horizontal cutting plane: flux in horizontal direction prescribed (=0),
%displacement in vertical direction prescribed (=0)
iif_horcut = 2 * NDShorcut - 1 ;
qf_horcut = zeros( size( iif_horcut ) ); %horizontale flux
iig_horcut = 2 * NDShorcut ;
ug_horcut = zeros( size( iig_horcut ) ); %verticale verplaatsing

%vertical snijvlak: cutting plane: flux in vertical direction
%prescribed (=0), displacement in horizontal direction prescribed (=0)
iif_vercut = 2 * NDSvercut ;
qf_vercut = zeros( size( iif_vercut ) ); %horizontale flux
iig_vercut = 2 * NDSvercut - 1 ;
ug_vercut = zeros( size( iig_vercut ) ); %verticale verplaatsing

%central nodes: flux in horizontal and vertical direction prescribed (=0)
iif_centre = sort( [ 2 * NDScentre 2 * NDScentre - 1 ] );
qf_centre = zeros( size( iif_centre ) );

%corner nodes
%inner edge and horizontal cutting plane: horizontal flux prescribed
%(=p/2), displacement in vertical direction prescribed (=0)
iif_innhor = 2 * NDSinnhor - 1;
iig_innhor = 2 * NDSinnhor;
qf_innhor = Fi/2;
ug_innhor = 0;

%inner edge and vertical cutting plane: vertical flux prescribed
%(=p/2), displacement in horizontal direction prescribed (=0)
iif_innver = 2 * NDSinnver;
iig_innver = 2 * NDSinnver - 1;
qf_innver = Fi/2;
ug_innver = 0;

%outer edge and horizontal cutting plane: horizontal flux prescribed
%(=p/2), displacement in vertical direction prescribed (=0)
iif_outhor = 2 * NDSouthor - 1;
iig_outhor = 2 * NDSouthor;
qf_outhor = Fo/2;
ug_outhor = 0;

%inner edge and vertical cutting plane: vertical flux prescribed
%(=p/2), displacement in horizontal direction (=0)
iif_outver = 2 * NDSoutver;
iig_outver = 2 * NDSoutver - 1;
qf_outver = Fo/2;
ug_outver = 0;

```

```

iif = [ iif_inner iif_outer iif_horcut iif_vercut iif_centre iif_innhor iif_innver iif_outhor iif_outver];
iig = [ iig_horcut iig_vercut iig_innhor iig_innver iig_outhor iig_outver];
Kgg = K( iig, iig );
Kfg = K( iif, iig );
Kgf = K( iig, iif );
Kff = K( iif, iif );

ug = [ ug_horcut ug_vercut ug_innhor ug_innver ug_outhor ug_outver ];
qf = [ qf_inner qf_outer qf_horcut qf_vercut qf_centre qf_innhor qf_innver qf_outhor qf_outver];

% solve for displacement field
uf = Kff \ (qf - Kfg*ug);
u = zeros(2*n, 1);
u(iig) = ug;
u(iif) = uf;
u = Vector2(u);

% plot displacement solution
uh = VectorField(Vh);
uh(nodes(Vh)) = u;
plot(uh)
drawnow

% determine reaction forces
qg = Kgg * ug + Kgf * uf;
q = zeros(2*n, 1);
q(iig) = qg;
q(iif) = qf;
q = Vector2(q);

% plot reaction forces solution
figure
qh = VectorField(Vh);
qh(nodes(Vh)) = q;
plot(qh)
drawnow

```

Appendix 4 Matlab script to generate a quarter circle mesh

```
%Matlab script to generate a mesh of a quarter circle with variable inner
%and outer radius and variable number of elements in both radial and
%tangential direction. Returned is the mesh itself and lists of similar
%nodes

function [M, NDSinneredge, NDSouteredge, NDShorcut, NDSvercut, NDSinnhor, NDSinnver, NDSouthor,
NDSoutver, NDScentre] = kwartcirkelmesh(ri,ro,nt,nr)

%Generate cilindrical point coordinates
polairecoordinaten=[];
cartesischecoordinaten=[];
for i=1:nt+1
    thetadegrees=(i-1)*90/nt;
    theta=thetadegrees/180*pi;
    for j=1:nr+1
        r=ri+(j-1)*(ro-ri)/nr;
        polairecoordinaten=[polairecoordinaten; r theta];
    end
end

%Generate cartesian point coordinates
for i=1:length(polairecoordinaten)
    xi=polairecoordinaten(i,1)*cos(polairecoordinaten(i,2));
    yi=polairecoordinaten(i,1)*sin(polairecoordinaten(i,2));
    cartesischecoordinaten=[cartesischecoordinaten; xi yi];
end

%Generate nodes from points
for i=1:length(cartesischecoordinaten)
    NDS(i)=Node(Point2(cartesischecoordinaten(i,:),i));
end

%Generate mesh from nodes
M=mesh;
j=1;
for i=1:nt*nr
    if mod(i,nr)==1 & i>nr
        j=j+1;
    end
    Q=Quad( [
        NDS(j)
        NDS(j+1)
        NDS(j+1+(nr+1))
        NDS(j+(nr+1)) ] );
    M=insert(M,Q);
    j=j+1;
end

%Nodes on inner and outer edge
for i=0:nt
    NDSinneredgeall(i+1)=(1+i*(nr+1));
    NDSouteredgeall(i+1)=((i+1)*(nr+1));
end
NDSinneredge=NDSinneredgeall(2:length(NDSinneredgeall)-1); %Node numbers of nodes on inneredge, excluding
the corner nodes
NDSouteredge=NDSouteredgeall(2:length(NDSouteredgeall)-1); %Node numbers of nodes on outeredge, excluding
the corner nodes

%Nodes on horizontal and vertical cut plane
NDShorcutall=[1:nr+1];
NDSvercutall=[length(NDS)-nr:length(NDS)];
NDShorcut=NDShorcutall(2:length(NDShorcutall)-1); %Node numbers of nodes on horizontal cutting plane, excluding
the corner nodes
NDSvercut=NDSvercutall(2:length(NDSvercutall)-1); %Node numbers of nodes on vertical cutting plane, excluding
the corner nodes

%Nodes on both cut plane and edge (corner nodes)
NDSinnhor=NDShorcutall(1);
NDSinnver=NDSvercutall(1);
NDSouthor=NDShorcutall(length(NDShorcutall));
NDSoutver=NDSvercutall(length(NDSvercutall));
```

```
%Nodes in the centre of the mesh
NDSall=[1:length(NDS)];
NDSused=[NDSinneredge, NDSouteredge, NDSshortcut, NDSvercut, NDSinnerhor, NDSinnerver, NDSouterhor,
NDSouterver];
NDScentre=[];
for i=1:length(NDSall)
    if length(find(NDSused==NDSall(i)))==0
        NDScentre=[NDScentre NDSall(i)];
    end
end
```