

## A technique to avoid aspect-ratio locking in QUAD8 element for extremely large aspect-ratios

S. Rajendran\*

School of Mechanical and Aerospace Engineering, Nanyang Technological University,  
50 Nanyang Avenue, 639798, Singapore

(Received October 22, 2008, Accepted November 22, 2010)

**Abstract.** This paper investigates the aspect-ratio locking of the isoparametric 8-node quadrilateral (QUAD8) element. An important finding is that, if finite element solution is carried out with in exact arithmetic (i.e., with no truncation and round off errors), the locking tendency of the element is completely avoided even for aspect-ratios as high as 100000. The current finite element codes mostly use floating point arithmetic. Thus, they can only avoid this locking for aspect-ratios up to 100 or 1000. A novel method is proposed in the paper to avoid aspect-ratio locking in floating point computations. In this method, the offending terms of the strain-displacement matrix (i.e., **B**-matrix) are multiplied by suitable scaling factors to avoid ill-conditioning of stiffness matrix. Numerical examples are presented to demonstrate the efficacy of the method. The examples reveal that aspect-ratio locking is avoided even for aspect-ratios as high as 100000.

**Keywords:** finite element method; quadrilateral element; aspect-ratio; locking; exact arithmetic; ill-conditioning.

---

### 1. Introduction

Although finite element method (FEM) has established itself as the single most powerful numerical tool in mechanics over the last few decades, there is still some scope for further improvements in areas such as ill-conditioning of stiffness matrix due to *locking* or other causes.

*Shear locking*, *volumetric locking* and *membrane locking* have been extensively investigated by several researchers. Locking may be broadly defined as a pathological condition of a finite element leading to grossly excessive stiffness for a particular deformation mode (e.g., see MacNeal 1994). Consequently, a locking-prone element cannot represent that particular deformation mode correctly even with a very fine mesh. Under certain conditions, the locking-prone element virtually fails to respond to the applied forces, leading to extremely small displacements. In other words, the element behaves as if “locked” out of that particular deformation mode. For instance, this situation arises when modeling a thin cantilever beam subjected to a transverse load. Since the beam is thin, it is natural to expect more of a bending deformation than the shear deformation. However, locking-prone finite elements, such as the classical isoparametric bilinear quadrilateral (QUAD4) element,

---

\*Corresponding author, Associate Professor, E-mail: [msrajendran@ntu.edu.sg](mailto:msrajendran@ntu.edu.sg)

are unable to respond to the bending action correctly in view of excessive stiffness due to shear terms in the stiffness matrix. This pathological condition is termed *shear locking*. Nevertheless, the shear locking tendency in higher order elements such as QUAD8 element, is known to be very mild (e.g., see Prathap 1993). For the normal deformation modes of interest in engineering, this mild locking tendency can often be ignored.

Literature available on the locking behavior of finite elements is too vast for an exhaustive review in this paper. Over the last four decades, extensive research has been carried out on the locking tendency of finite elements and several techniques to combat locking have been proposed (e.g., see MacNeal 1994, Prathap 1993). One of the classical techniques to alleviate the locking problem is to use a lower order integration for the numerical evaluation of the stiffness matrix. This technique, often called the *reduced integration* or *under integration* in the literature, was proposed as a practical remedy to locking problems around 1970 (Doherty *et al.* 1969, Zienkiewicz 1971). Initially, this technique tended to be disregarded as an *extra-variational trick* or a *variational crime* although, subsequently, it proved to be the main source of inspiration for a spate of techniques to treat locking problems. An excellent account of locking and effect of reduced integration on the performance of QUAD8 element can be found in the literature (MacNeal 1994, Prathap 1993, Cook 1989, MacNeal and Harder 1992). Reduced integration generally tends to soften an element, which counters the excessive stiffness caused by locking. However, a naïve application of reduced integration technique for both offending and non-offending terms of the stiffness matrix has been found to introduce *spurious modes (zero energy modes)* that ruin the performance in some types of elements (such as QUAD9, the nine node quadrilateral element based on Lagrangian shape functions). This observation motivated the development of *selective reduced integration* technique wherein the reduced integration is applied selectively only to the offending terms of the stiffness matrix.

Many other techniques have also been proposed to eliminate different types of locking in finite elements. Addition of *bubble functions* (e.g., see Wilson *et al.* 1973), *assumed strain/stress formulations* (e.g., see Hughes 1980, Pian 1964), *mixed* or *hybrid* approaches are notable ones. The *field-consistency* approach to treat the locking problems, pioneered by Prathap and his co-researchers (e.g., see Chandra and Prathap 1989, Ramesh Babu 1985, Rajendran and Prathap 1999, Naganarayana 1991, Prathap 1994, 1997), has been another systematic approach to avoid locking problems in several types of elements. In this approach, the locking tendency of finite elements is viewed as a consequence of lack of *consistency* in representing the stress field. The corrective action to enforce consistency involves the use of a lower order best-fit of the stress expression corresponding to the offending mode.

Many of the techniques proposed to eliminate locking can be viewed as a rationalization or generalization of the idea of selective reduced integration technique. In *mixed formulations*, for example, the offending terms are ameliorated not simply by reduced integration but by an equivalent modification in the design of interpolation functions. Polynomial functions of different orders are used for interpolating the displacements and the strains/stresses. The relation between mixed formulation and reduced integration technique is discussed by Malkus and Hughes (1978). *Function space approach* (Sangeeta *et al.* 2005, Mukherjee and Prathap 2001) and *selective smoothed finite element* (Nguyen *et al.* 2007) are two recent approaches to understand locking in finite elements.

*Aspect-ratio locking*, sometimes known as *aspect-ratio stiffening*, is another kind of locking which is of considerable importance in modeling very thin structures. Although there have been numerous

studies on the locking tendency of finite elements, papers focusing on aspect-ratio locking are limited (Pitkäranta 2000). Whatever be the type of locking, the root cause of locking tendency is the large difference between the magnitudes of the offending and non-offending stiffness terms. In the case of shear locking, this difference is caused by the spurious shear terms, whereas in the case of aspect-ratio locking, it is caused by the large difference in dimensions between the thickness and length directions arising in slender structures like beams and plates.

In Section 2, the aspect-ratio locking of QUAD8 element is investigated for thin beam applications. A practical remedy to the aspect-ratio locking is investigated in Section 3. The computer code used for the work is discussed in Section 4. Important observations are summarized in Section 5.

## 2. Aspect-ratio locking in thin beams

Exact arithmetic calculations are necessary to illustrate an interesting feature of aspect-ratio locking. Hence, finite element computer codes written in *Mathematica*<sup>®</sup> (Wolfram 1999) are used for the purpose. The code is listed in Appendix A. The details of the code will be discussed in Section 4.

### 2.1 Cantilever beam with unit tip-moment load

A cantilever beam is subjected to a unit tip-moment is shown in Fig. 1. The depth of the beam,  $D$ , is taken as 1 unit while the length,  $L$ , is varied from 10 units to 100000 units. The thickness of the beam is taken as 1 unit. The Young's Modulus,  $Y$ , and the Poisson's ratio,  $\nu$ , are taken as 600 units and 0.25, respectively. The problem is modeled with a single serendipity type QUAD8 element. The stiffness integral is evaluated using four different integration schemes, viz., exact integration,  $3 \times 3$  Gaussian quadrature, uniform reduced integration (i.e.,  $2 \times 2$  Gaussian quadrature for both shear and other terms) and selective reduced integration (i.e.,  $2 \times 2$  Gaussian quadrature for the shear terms and  $3 \times 3$  Gaussian quadrature for the rest). The reference vertical tip displacement for the problem is calculated as  $v_{ref} = ML^2/2YI$  where  $M$  is the moment and  $I$  is the sectional area moment of inertia. This solution, which is based on strength of materials approach, is sufficiently accurate for thin beam modeling which is the focus of this paper. The computed vertical tip

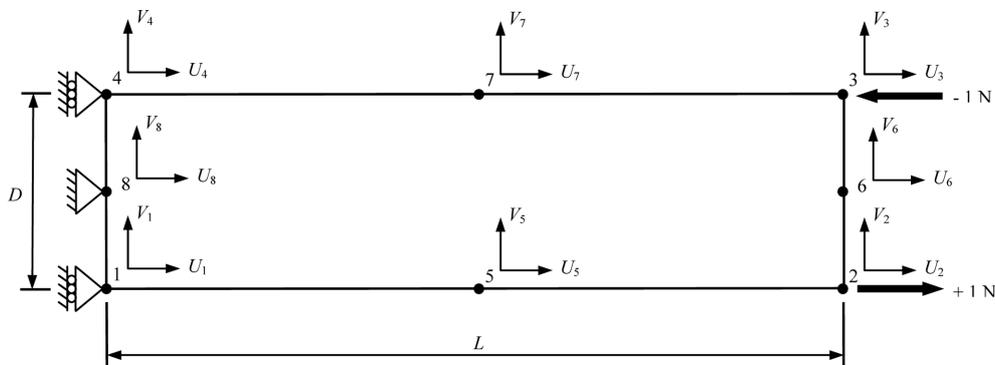


Fig. 1 A cantilever beam with a unit tip-moment load

Table 1 Vertical displacement at node 6 for tip-moment load with all the input data in decimal format

Aspect-ratio ( $L/D$ )	Exact integration	Uniform $3 \times 3$ integration	Uniform reduced ( $2 \times 2$ ) integration	Selective reduced integration	Reference solution ( $v_{ref}$ )
10	1.0000000000121245	1.0000000000052944	0.9999999999892633	0.999999999990412	1
100	100.00000311771221	100.00000348167083	99.99997508816216	100.00000194356474	100
1000	9985.976455413136	9992.93808193509	9993.32563023032	10014.440983747296	10000
10000	-912436.0024432316	-85658.65407183554	36145.92082289764	66668.72957898416	$10^6$
100000	309.6015444357179	418.9506213170821	1056.8435116900973	-2799.4250665808036	$10^8$

Table 2 Same as Table 1, but with all the input data in integer/fractional format

Aspect-ratio ( $L/D$ )	Exact integration	Uniform $3 \times 3$ integration	Uniform reduced ( $2 \times 2$ ) integration	Selective reduced integration	Reference solution ( $v_{ref}$ )
10	1	1.0000000000000002	1	1.0000000000000002	1
100	100	100.00000000000001	100	100.00000000000001	100
1000	1000	10000.000000000002	1000	10000.000000000002	10000
10000	$1 \times 10^6$	$1.0000000000000001 \times 10^6$	$1 \times 10^6$	$1.0000000000000001 \times 10^6$	$10^6$
100000	$1 \times 10^8$	$1.0000000000000001 \times 10^8$	$1 \times 10^8$	$1.0000000000000001 \times 10^8$	$10^8$

displacements (at node 6, Fig. 1) obtained using the *Mathematica*<sup>®</sup> code are listed in Table 1. The main observations from Table 1 are:

- (i) The accuracy of computed displacements deteriorates with the increase in aspect-ratio irrespective of the integration schemes employed.
- (ii) Only a slight deterioration is seen for aspect-ratios 10, 100 and 1000.
- (iii) Severe deterioration is seen for aspect-ratios 10000 and 100000.

It is important to note that, for the results reported in Table 1, the input data to the *Mathematica*<sup>®</sup> code (such a dimensions of the cantilever, loads and material properties) are in the decimal format. However, when all the input values are entered in integer/fractional format (e.g., 0.31 changed to 31/100 and 4.0 changed to 4), the computed displacements turn out to be surprisingly very accurate with no trace of 'locking' whatsoever, yielding exact displacements (see Table 2).

The reason for this intriguing accuracy has been investigated. It transpires that whenever the input data is entered in integer or fractional format, all computations done by *Mathematica*<sup>®</sup> are in exact arithmetic, whereby no approximations such as truncation and round off enter the computations. Thus, exact arithmetic computations have been responsible for the excellent accuracy of results in Table 2. It is well-known that reduced integration of stiffness matrix alleviates shear locking problem in many types of finite elements. However, this general trend is not seen in columns 4 and 5 of Table 1. Thus, the reduced integration is unable to avoid deterioration of results, but exact arithmetic is. This observation clearly suggests that the deterioration of results in the present case is not caused by shear locking, but by the aspect-ratio locking induced by the large differences in lateral and longitudinal stiffness values due to the high aspect-ratios.

It is to be mentioned here that even if any one of the input values, for example the Young's

modulus, is entered in decimal format, the computations done by *Mathematica*<sup>®</sup> switch over to floating point arithmetic. Consequently, the truncation and round off errors tend to plague the accuracy of computations and hence the deterioration of results in Table 1.

2.2 Cantilever beam with unit tip-shear load

The geometry and boundary conditions are the same as in Section 2.1. However, the Young’s modulus is taken as 4000 units while the Poisson’s ratio value is the same as in Section 2.1. A tip-shear force of  $P = 1$  unit is applied as shown in Fig. 2. The reference displacement for this problem (based on strength of materials approach) is given by  $v_{ref} = PL^3/3YI$ . Here again, the computations have been carried out in exact (integer) as well as floating point arithmetic by entering all the input data in integer/fractional format or decimal format, respectively. The corresponding tip displacements computed are listed in Table 3 and 4, respectively. The following observations emerge from Table 3:

- (i) The displacements shown (in any of the columns) reveal a slight deterioration in accuracy as the aspect-ratio is increased up to 1000, but a very severe deterioration for aspect-ratios beyond 10000.
- (ii) Columns 4 and 5 show that, for aspect-ratios up to 1000, uniform reduced (2 × 2) integration

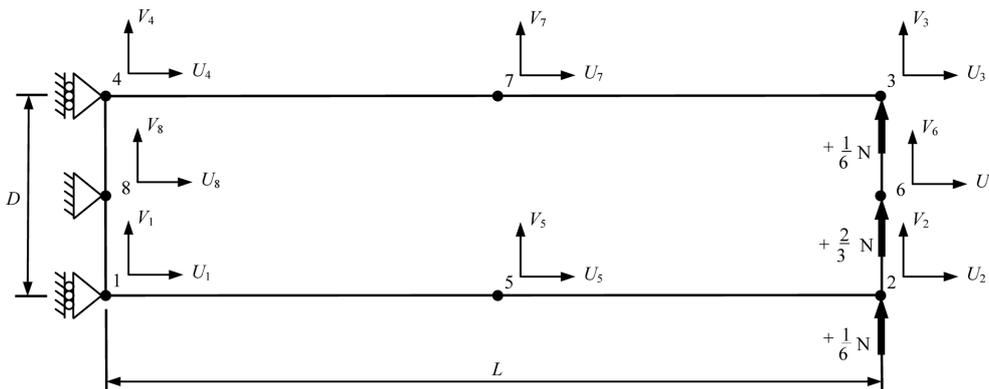


Fig. 2 A cantilever beam with a unit tip-shear load

Table 3 Vertical displacement at node 6 for tip-shear load with all the input data in decimal format

Aspect-ratio (L/D)	Exact integration	Uniform 3 × 3 integration	Uniform reduced (2 × 2) integration	Selective reduced integration	Reference solution (v <sub>ref</sub> )
10	0.784238866722784	0.7842388667658277	1.0068781250142023	1.0068781249952705	1
100	750.3760535881969	750.3762323623826	1000.0686712109683	1000.069139872639	1000
1000	749186.1856437577	749335.3105193361	998024.2914658628	1000354.1456111355	1000000
10000	-0.048438934078823 × 10 <sup>9</sup>	-0.079692701231378 × 10 <sup>9</sup>	0.045475810748153 × 10 <sup>9</sup>	-0.0389677023510543 × 10 <sup>9</sup>	10 <sup>9</sup>
100000	0.000002027627867 × 10 <sup>12</sup>	-0.00000474008002 × 10 <sup>12</sup>	0.00000885599275 × 10 <sup>12</sup>	-0.0000099949801286 × 10 <sup>12</sup>	10 <sup>12</sup>

Table 4 Same as Table 3, but with all the input data in integer/fractional format

Aspect-ratio ( $L/D$ )	Exact integration	Uniform $3 \times 3$ integration	Uniform reduced ( $2 \times 2$ ) integration	Selective reduced integration	Reference solution ( $v_{ref}$ )
10	0.7832541030573198	0.78325410305732	1.0071041666666667	1.005059949334577	1
100	750.3667975450328	750.366797545033	1000.0708354166667	1000.0500060540812	1000
1000	750003.6718359556	750003.6718359558	1000000.7083335416	1000000.500006056	1000000
10000	0.750000036718746 $\times 10^9$	0.750000036718746 $\times 10^9$	1.000000070833334 $\times 10^9$	1.000000050000001 $\times 10^9$	$10^9$
100000	0.75000000036719 $\times 10^{12}$	0.75000000036719 $\times 10^{12}$	1.000000000708334 $\times 10^{12}$	1.000000000500002 $\times 10^{12}$	$10^{12}$

and selective reduced integration are able to recover very accurate nodal displacements as compared to that of exact integration or uniform  $3 \times 3$  integration. (It should be noted that this increase in accuracy is not related to locking. The sampling points of reduced integration are in fact the points of optimal stress recovery and as a result, when the stiffness integrals are evaluated using reduced integration, the displacements turn out to more accurate.)

The following are the observations from Table 4:

- (i) The displacements listed (in any of the columns) show no sign of deterioration with aspect-ratio.
- (ii) As in Table 3, here again, uniform reduced ( $2 \times 2$ ) integration and selective reduced integration are able to recover very accurate nodal displacements as compared to exact integration or uniform  $3 \times 3$  integration.

### 2.3 Cantilever bar with unit tip-tensile load

The dimensional details and boundary conditions are the same as in Section 2.1. However, the Young's modulus is taken as 1000 units while the Poisson's ratio value is the same as in Section 2.1. A tensile force of  $P=100$  units is applied at the right end as in Fig. 3. The reference displacement for this problem is given by  $v_{ref} = PL/AY$  where  $A$  is the cross-sectional area of the bar.

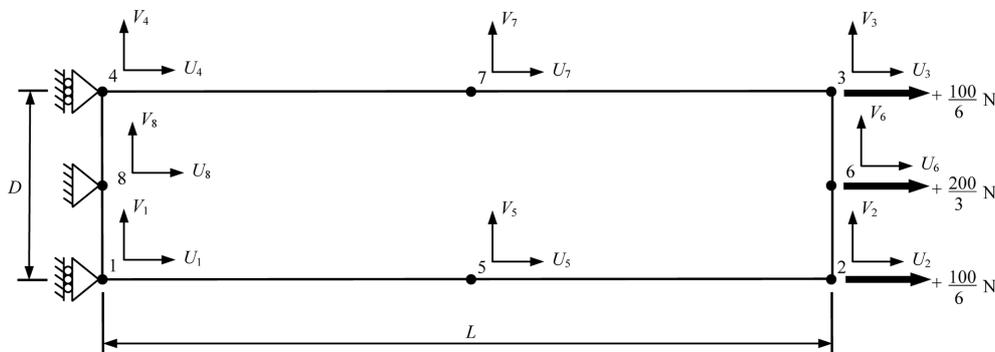


Fig. 3 A cantilever bar with a unit tip-tensile load

Table 5 Horizontal displacement at node 6 for tip-tensile load with all the input data in decimal format

Aspect-ratio ( $L/D$ )	Exact integration	Uniform $3 \times 3$ integration	Uniform reduced ( $2 \times 2$ ) integration	Selective reduced integration	Reference solution ( $v_{ref}$ )
10	1.000000000000006	0.999999999999765	1.000000000000084	0.999999999999938	1
100	9.99999999999615	10.00000000006999	9.99999999983073	10.0000000001535	10
1000	100.0000000321863	100.0000003490622	99.999999951623	99.999998431846	100
10000	999.9999913266515	999.9999851985052	999.99999461749	999.9999741747741	1000
100000	9999.987045460732	10000.020893922572	10000.004208029446	9999.946309214758	10000

Table 6 Same as Table 5, but with all the input data in integer/fractional format

Aspect-ratio ( $L/D$ )	Exact integration	Uniform $3 \times 3$ integration	Uniform reduced ( $2 \times 2$ ) integration	Selective reduced integration	Reference solution ( $v_{ref}$ )
10	1	1.000000000000002	1	1.000000000000002	1
100	10	10.000000000000002	10	10.000000000000002	10
1000	100	100.000000000000001	100	100.000000000000001	100
10000	1000	1000.000000000000001	1000	1000.000000000000001	1000
100000	10000	10000.000000000000002	10000	10000.000000000000002	10000

Table 7 Results\* of ANSYS Plane 82 element

Aspect-ratio ( $L/D$ )	Tip-moment problem	Tip-shear problem	Tip-tension problem
10	1.0000	1.0069	1.0000
100	100.00	1000.1	10.000
1000	9998.2	$0.99993 \times 10^6$	100.00
10000	$0.39803 \times 10^6$	-	1000.0
100000	3230.4	-	10000.0

\*Rounded to only 5 significant places

The computed results are shown in Tables 5 and 6. Table 5 shows that very mild ill-conditioning exists. Table 6 shows that even this mild ill-conditioning effect is completely removed by exact arithmetic.

The above three problems have also been solved using ANSYS (Kohnke 1997). The displacements for the three problems discussed in Sections 2.1-2.3 as given by PLANE82 (ANSYS) are listed in Table 7, which reveal that, here again, the aspect-ratio locking seems to be plaguing the results. For the tip-shear problem, the aspect-ratios 10000 and 100000 could not be handled by ANSYS in view of high aspect-ratios.

### 3. Ill-conditioning correction to avoid aspect-ratio locking

Although the aspect-ratio locking could easily be avoided by exact arithmetic computations, use of exact arithmetic is not a practically attractive approach to solve the locking problem. This is because most of the finite element packages available today do not employ symbolic computation, and floating point arithmetic is prevalent in all practical finite element packages. Hence, in this section a more practical technique, called *progressive ill-conditioning correction* (PIC), is introduced for avoiding the aspect-ratio locking in floating point computations.

#### 3.1 Progressive ill-conditioning correction (PIC)

The ill-conditioning of a stiffness matrix generally arises due to large differences in the entries of the stiffness matrix. In turn, the large differences in the entries of the stiffness matrix arise from the large differences in the entries of strain-displacement matrix  $[\mathbf{B}]$ .

$$[\mathbf{B}] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \dots & \frac{\partial N_8}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & \dots & 0 & \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \dots & \frac{\partial N_8}{\partial y} & \frac{\partial N_8}{\partial x} \end{bmatrix} \quad (1)$$

For beams which are thin in the  $y$ -direction, the derivative,  $\partial N_i/\partial y$ , will turn out to be very large as compared to  $\partial N_i/\partial x$ . For instance, for a beam with a length of 100000 units (in the  $x$  direction) and depth of 1 unit (in the  $y$  direction),  $\partial N_i/\partial x$  will be of the order of  $1/100000$  units whereas  $\partial N_i/\partial y$  will be of the order of 1 unit. This large difference leads to entries of very large as well as very small magnitudes in the stiffness matrix. This ruins the conditioning of the stiffness matrix and thereby the accuracy of computed solution.

The idea behind the PIC method is to multiply the offending terms in the  $[\mathbf{B}]$ -matrix by an appropriate (penalty like) correction factor that varies progressively with the aspect-ratio of the element. The objective is to adjust the condition number of the element stiffness matrix so that ill-conditioning does not affect the element performance for thin beams. For the present work, the  $\partial N_i/\partial y$  terms appearing in the second row of the  $[\mathbf{B}]$ -matrix is multiplied by a variable correction factor,  $\alpha_y$ . The modified  $[\mathbf{B}]$ -matrix becomes

$$[\mathbf{B}] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \dots & \frac{\partial N_8}{\partial x} & 0 \\ 0 & \alpha_y \frac{\partial N_1}{\partial y} & 0 & \alpha_y \frac{\partial N_2}{\partial y} & 0 & \alpha_y \frac{\partial N_3}{\partial y} & \dots & 0 & \alpha_y \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \dots & \frac{\partial N_8}{\partial y} & \frac{\partial N_8}{\partial x} \end{bmatrix} \quad (2)$$

The variable correction factor,  $\alpha_y$ , is a function of the parameter,  $s$ , which is a measure of the aspect-ratio, defined as

$$s = \frac{dy d\xi}{d\eta dx} \tag{3}$$

where  $\xi$  and  $\eta$  are the natural (parametric) coordinates. For a rectangular beam of length  $L$  (in  $x$ -direction) and depth  $D$  (in  $y$ -direction), we can show that  $s = \frac{dy d\xi}{d\eta dx} = \frac{D2}{2L} = \frac{D}{L} = \frac{1}{(L/D)}$ . Thus, for a rectangular beam, the parameter  $s$  turns out to be the inverse of the aspect-ratio,  $L/D$ .

An appropriate choice of the function,  $\alpha_y$ , is crucial for the success of the present method. Based on a trial and error process, the following choice has been found to be satisfactory

$$\begin{aligned} \alpha_y &= s^{-1+e^{-\frac{1}{5}\log_{10}(s)}} && \text{for } 0 \leq s \leq 1 \\ &= 1 && \text{for } s > 1 \end{aligned} \tag{4}$$

A plot of  $\alpha_y$  versus  $s$  is shown in Fig. 4.

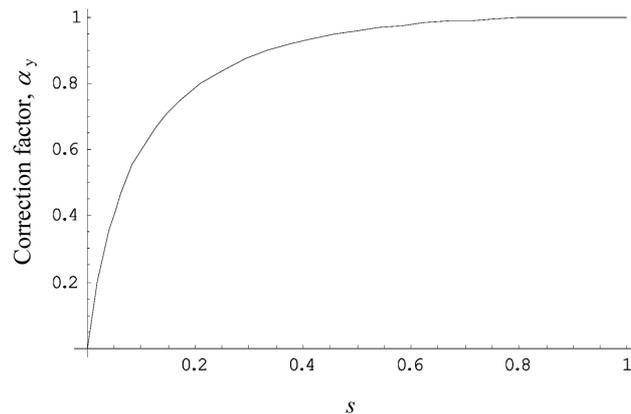


Fig. 4 Aspect-ratio dependent correction factor (see Eq. (4))

Table 8 Same as Table 1, but with PIC<sup>1</sup>

Aspect-ratio (L/D)	Exact integration	Uniform 3 × 3 integration	Uniform reduced (2 × 2) integration	Selective reduced integration	Reference solution (v <sub>ref</sub> )
10	1.000000000033253	1.000000000090163	1.000000000006848	1.000000000042324	1
100	100.00000000618486	100.00000047339321	99.99999973682857	100.00000016447505	100
1000	9999.999896102003	9999.9996472898	9999.999807618442	10000.000444540796	10000
10000	1.0000000471041086 × 10 <sup>6</sup>	0.9999999919084449 × 10 <sup>6</sup>	0.9999998483984278 × 10 <sup>6</sup>	0.9999999674771561 × 10 <sup>6</sup>	10 <sup>6</sup>
100000	9.999986747845925 × 10 <sup>8</sup>	1.0000045930621749 × 10 <sup>8</sup>	0.9999934888859335 × 10 <sup>8</sup>	1.000003437579662 × 10 <sup>8</sup>	10 <sup>8</sup>

<sup>1</sup>PIC stands for Progressive ill-conditioning correction.

Table 9 Same as Table 3, but with PIC<sup>1</sup>

Aspect-ratio ( $L/D$ )	Exact integration	Uniform $3 \times 3$ integration	Uniform reduced ( $2 \times 2$ ) integration	Selective reduced integration	Reference solution ( $v_{\text{ref}}$ )
10	0.7843821998474318	0.7843821998555516	1.0072992602376556	1.007299260235363	1
100	750.3897416391114	750.3897422027492	1000.1227194625654	1000.1227201446147	1000
1000	750008.3170000897	750008.3389638492	1000018.9042145895	1000018.9323219948	1000000
10000	0.7500125028352147 $\times 10^9$	0.7500125118896408 $\times 10^9$	1.0000499494341776 $\times 10^9$	1.0000499088701078 $\times 10^9$	$10^9$
100000	0.750610178351064 $\times 10^{12}$	0.750610224108150 $\times 10^{12}$	1.0024809681419504 $\times 10^{12}$	1.0024831587425344 $\times 10^{12}$	$10^{12}$

<sup>1</sup>PIC stands for Progressive ill-conditioning correction.

Table 10 Same as Table 5, but with PIC<sup>1</sup>

Aspect-ratio ( $L/D$ )	Exact integration	Uniform $3 \times 3$ integration	Uniform reduced ( $2 \times 2$ ) integration	Selective reduced integration	Reference solution ( $v_{\text{ref}}$ )
10	1.0000000000000006	0.999999999999977	0.999999999999963	0.999999999999943	1
100	9.999999999999615	10.00000000010692	9.99999999997128	10.00000000000256	10
1000	100.00000000321863	100.00000002970569	100.0000000273887	99.999999821905	100
10000	999.9999913266515	999.9999894833795	1000.0000010683636	999.9999760287732	1000
100000	9999.987045460732	10000.020930516754	10000.00174698653	9999.954840800394	10000

<sup>1</sup>PIC stands for Progressive ill-conditioning correction.

The displacements computed with the PIC are summarized in Tables 8-10. It is seen from these tables that the PIC method is indeed very effective in removing the locking effects even for aspect-ratios as high as 100000. The computed displacements are now very close to that of exact arithmetic (listed in Tables 2, 4 and 6).

Note that the  $y$ -derivative term,  $\partial N_1/\partial y$ , appears not only in row 2 but also in row 3 of  $[\mathbf{B}]$ -matrix (Eq. (2)). However, in this paper, the correction is applied only to row 2. Numerical experiments (not reported here) show that a similar correction applied to row 3 slightly improves the results for the tip-moment problem, but completely ruins the results for the tip-shear problem, and has practically no effect for the tip-tensile problem. Hence, for all the results reported here on, no correction has been applied to row 3.

The variable correction factor as defined in Eq. (4) applies to thin beams that are thin in  $y$ -direction. For beams that are thin in  $x$ -direction, we may similarly define another variable correction factor as

$$\alpha_x = r^{-1+e^{-\frac{1}{5}\log_{10}(r)}} \quad \text{for } 0 \leq r \leq 1$$

$$= 1 \quad \text{for } r > 1 \quad (5)$$

where

$$r = \frac{dx d\eta}{d\xi dy} \tag{6}$$

For a general case, where the beam may be thin either in the  $x$ - or  $y$ - direction, the  $[\mathbf{B}]$ -matrix is written as

$$[\mathbf{B}] = \left[ \begin{array}{cc|cc|cc|c|cc} \alpha_x \frac{\partial N_1}{\partial x} & 0 & \alpha_x \frac{\partial N_2}{\partial x} & 0 & \alpha_x \frac{\partial N_3}{\partial x} & 0 & \dots & \alpha_x \frac{\partial N_8}{\partial x} & 0 \\ 0 & \alpha_y \frac{\partial N_1}{\partial y} & 0 & \alpha_y \frac{\partial N_2}{\partial y} & 0 & \alpha_y \frac{\partial N_3}{\partial y} & \dots & 0 & \alpha_y \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \dots & \frac{\partial N_8}{\partial y} & \frac{\partial N_8}{\partial x} \end{array} \right] \tag{7}$$

For thin beams that are oriented arbitrarily (neither parallel to  $x$ - nor  $y$ -axis), the above approach cannot be applied in a straight forward manner. The correction, in such a case, needs to be applied in three steps:

- (i) Apply the ill-conditioning correction to the  $[\mathbf{B}]$ -matrix in an element-based coordinate system.
- (ii) Obtain the stiffness matrix in the element-based coordinate system.
- (iii) Transform the stiffness matrix to global coordinate system.

### 3.2 Condition number of the stiffness matrix

The condition number,  $C$ , of a matrix is defined as the ratio of its largest eigenvalue,  $\lambda_{\max}$ , to the smallest eigenvalue,  $\lambda_{\min}$ .  $C = \lambda_{\max}/\lambda_{\min}$ . The larger the condition number, the more severe is the ill-conditioning. The condition number of the element stiffness matrix for the element in Fig. 1, computed for various aspect-ratios, is listed in Table 11. The numbers are presented in the format,  $a \times 10^b$ , with the mantissa  $a$  being rounded to one decimal place. Many observations emerge from Table 11.

Firstly, we note from Table 11 that the condition number generally increases with aspect-ratio, which is rather an expected trend. However, even with PIC, we see the same trend, although we

Table 11 Condition number of the stiffness matrix ( $3 \times 3$  stiffness integration)

Aspect-ratio ( $L/D$ )	Before applying PIC <sup>1</sup>		After applying PIC	
	Exact arithmetic	Floating-point arithmetic	Exact arithmetic	Floating-point arithmetic
10	$3.1 \times 10^5$	$3.1 \times 10^5$	$1.1 \times 10^5$	$1.1 \times 10^5$
100	$3.0 \times 10^9$	$3.0 \times 10^9$	$1.0 \times 10^9$	$1.0 \times 10^9$
1000	$3.0 \times 10^{13}$	$3.0 \times 10^{13}$	$1.0 \times 10^{13}$	$1.0 \times 10^{13}$
10000	$3.0 \times 10^{17}$	$5.3 \times 10^{16}$	$1.2 \times 10^{17}$	$1.3 \times 10^{17}$
100000	$3.0 \times 10^{21}$	$2.9 \times 10^{16}$	$1.0 \times 10^{18}$	$3.6 \times 10^{18}$

<sup>1</sup>PIC stands for Progressive ill-conditioning correction.

would expect a decreasing trend. The reason for this has been traced to the fact that we have not applied PIC to row 3 of the [B]-matrix (see Section 3.1), so the corresponding terms in the stiffness matrix remain very large compared to the rest.

The next observation is that, the application of PIC actually tends to increase the condition number for aspect ratios 10000 and 100000 (compare columns 2 and 3 with columns 4 and 5, respectively) although we would expect a decrease. The reason for this can be explained as follows. For high aspect ratios without ill-conditioning correction (PIC),  $\lambda_{\min}$  is over-estimated due to stiffening effect caused by aspect-ratio locking, and therefore the condition number,  $C = \lambda_{\max}/\lambda_{\min}$ , is under-estimated. With the application of PIC,  $\lambda_{\min}$  is no more over-estimated, and hence the condition number,  $C = \lambda_{\max}/\lambda_{\min}$ , now appears to increase with the application of PIC.

Before attempting to interpret the results of Table 11, it is important to appreciate a special problem associated with the floating point computation of matrix condition number. Whenever the stiffness matrix is ill-conditioned, any computation involving the matrix entries will be affected adversely by the ill-conditioning of the matrix. Consequently, the computed condition number itself may be in error whenever the ill-conditioning seriously affects the computation of  $\lambda_{\min}$  or  $\lambda_{\max}$ . However, exact arithmetic is free from this problem, in which case the computed condition number turns to be exact.

Columns 2 and 3 of Table 11 show the condition numbers before applying PIC. It is seen that exact arithmetic and floating point arithmetic give same condition numbers for aspect ratios 10 to 1000. However, for aspect-ratios 10000 and 100000, they give condition numbers of different orders of magnitude. This large difference suggests that ill-conditioning must be seriously affecting the floating point computations. This inference is supported by the extremely poor displacements results (obtained before applying PIC) in Tables 1, 3 and 5.

Columns 4 and 5 of Table 11 show the condition numbers after applying PIC. Here, exact arithmetic and floating point arithmetic give condition numbers of the same order of magnitude for all aspect-ratios from 10 to 100000. This suggests that ill-conditioning is not seriously affecting the floating point computations any more. This conclusion is supported by the excellent displacement results in presented in Tables 8-10. Thus, as long as the condition numbers given by exact and floating point arithmetic are of comparable magnitudes, ill-conditioning does not seem to affect the results of floating point computations much.

#### 4. A discussion on the code listing

The code listing shown in Appendix A is for tip-moment loading with  $3 \times 3$  integration. Using the code as it is, the last entry of column 3 of Table 8 can be generated, which corresponds to an aspect-ratio of 100000. The vertical tip displacement (of node 6) computed by the code appears on the last line of listing. The aspect-ratio can be changed by editing line In[4]. Note that the MCC correction is applied through lines In[24] and In[25].

For generating the results of third column of Table 1,

- (i) remove line In[24], and
- (ii) replace line In[25] by In[25]:=penalty=1.

For generating the results of third column of Table 2,

- (i) remove line In[24],

- (ii) replace In[25] by In[25]:=penalty=1, and
- (iii) replace the Poisson's ratio value of 0.25 in line In[7] by 1/4

For other integration schemes (exact integration,  $2 \times 2$  and selective reduced integration), the code needs to be modified appropriately. In particular, for the case of exact integration, lines In[31]-In[37] are to be replaced by

$$K = \text{Simplify} [\text{Integrate} [\text{ker}, \{\text{xi}, -1, 1\}, \{\text{et}, -1, 1\}]];$$

For tip-shear and tip-tensile problems, line In[44] must be replaced by

$$P = \text{Transpose} [\{\{0, 0, 0, 1/6, 0, 1/6, 0, 0, 0, 0, 2/3, 0, 0, 0, 0\}\}] \text{ and}$$

$$P = \text{Transpose} [\{\{0, 0, 100/6, 0, 100/6, 0, 0, 0, 0, 0, 200/3, 0, 0, 0, 0\}\}], \text{ respectively.}$$

## 5. Conclusions

The aspect-ratio locking of isoparametric QUAD8 element has been investigated. The following are some of the important observations from the present work:

1) The aspect-ratio locking effect vanishes completely whenever all the finite element computations are carried out in exact arithmetic, i.e., in terms of integer and fractional arithmetic.

2) In most finite element computer programs available today, computations are usually carried out in floating point arithmetic rather than exact arithmetic. The progressive ill-conditioning correction (PIC) method proposed in Section 3 is useful in such situations. With this approach, even aspect-ratios as high as 1000 to 100000 can now be handled by the finite element program. With the PIC method, the computed results for aspect-ratios in the range 1000 to 100000 are far superior to that of ANSYS (compare the results of Tables 8-10 with that of Table 7).

3) The PIC method, as the name suggests, is a continuous correction method. In other words, the extent of correction applied varies continuously with aspect-ratio. For thick beams i.e., for  $s \geq 1$  or  $r \geq 1$ , PIC effectively applies no correction. For extremely thin beams (for  $s \rightarrow 0$  or  $r \rightarrow 0$ ), the maximum correction is applied so as to nullify the offending terms.

4) Although the choice of suitable functions for the variable correction factors  $\alpha_x$  and  $\alpha_y$  has been rather adhoc in this paper, the results clearly demonstrate the effectiveness of the PIC method. This may motivate further research by other researchers leading to a more rational choice of functions for  $\alpha_x$  and  $\alpha_y$ .

## References

- Chandra, S. and Prathap, G. (1989), "A field-consistent formulation for the eight-node solid finite element", *Comput. Struct.*, **33**, 345-355.
- Cook, R.D. (1989), *Concepts and Applications of Finite Element Analysis*, 3rd Edition, John Wiley & Sons, Canada.
- Doherty, W.P., Wilson, E.L. and Taylor, R.L. (1969), "Stress analysis of axisymmetric solids using higher order quadrilateral finite elements", University of California, Berkley, Struct. Eng. Lab Report SESM, 69-3.
- Hughes, T.R.J. (1980), "Generalization of selective integration procedures to anisotropic and nonlinear media", *Int. J. Numer. Meth. Eng.*, **15**, 1413-1418.

- Kohnke, P.C. (1997), *ANSYS: Theory Reference Release 5.4*, ANSYS, Inc., Canonsburg, PA.
- MacNeal, R.H. and Harder, R. (1992), "Eight nodes or nine?", *Int. J. Numer. Meth. Eng.*, **33**, 1049-1058.
- MacNeal, R.H. (1994), *Finite Element: Their Design and Performance*, Marcel Dekker, New York.
- Malkus, D.S. and Hughes, T.R.J. (1978), "Mixed finite element methods – reduced and selective integration techniques: a unification concepts", *Comput. Meth. Appl. Mech. Eng.*, **15**, 63-81.
- Mukherjee, S. and Prathap, G. (2001), "Analysis of shear locking in Timoshenko beam elements using the function space approach", *Commun. Numer. Meth. Eng.*, **17**, 385-393.
- Naganarayana, B.P. (1991), "Consistency and correctness in quadratic displacement finite elements", PhD Thesis, Faculty of Engineering, Indian Institute of Science, Bangalore.
- Nguyen, T.T., Liu, G.R., Dai, K.Y. and Lam, K.Y. (2007), "Selective smoothed finite element method", *Tsinghua Sci. Technol.*, **12**, 497-508.
- Pian, T.H.H. (1964), "Derivation of stiffness matrix by assumed stress distributions", *AIAA J.*, **2**, 1333-1336.
- Pitkäranta, J. (2000), "The first locking-free plane-elastic finite element: historia mathematica", *Comput. Meth. Appl. Mech. Eng.*, **190**, 1323-1366.
- Prathap, G. (1993), *The Finite Element Method in Structural Mechanics*, Kluwer Academic Publishers, Dordrecht.
- Prathap, G. (1994), "Locking, rank and singularity of penalty linked stiffness matrix and consistency of strain field", *Comput. Struct.*, **52**, 35-39.
- Prathap, G. (1997), "A field-consistency approach to plate problems", *Struct. Eng. Mech.*, **5**, 853-865.
- Rajendran, S. and Prathap, G. (1999), "Eight-node field-consistent hexahedron element", *Struct. Eng. Mech.*, **7**, 1-10.
- Ramesh Babu, C. (1985), "Field-consistency in the finite element formulation of multi-strain-field problem in structural mechanics", PhD Thesis, Indian Institute of Technology, Madras.
- Sangeeta, K., Mukherjee, S. and Prathap, G. (2005), "A function space approach to study rank deficiency and spurious modes in finite elements", *Struct. Eng. Mech.*, **21**, 539-551.
- Wilson, E.L., Taylor, R.L., Doherty, W. and Ghasbousi, J. (1973), "Incompatible displacement models", Published in *Numerical and Computer Methods in Structural Mechanics*, (Eds. Fenves, S.J. et al.), Academic Press, New York.
- Wolfram, S. (1999), *The Mathematica Book*, 4th edition, Wolfram Media.
- Zienkiewicz, O.C., Too, J. and Taylor, R.L. (1971), "Reduced integration technique in general analysis of plates and shells", *Int. J. Numer. Meth. Eng.*, **3**, 275-290.

## Appendix A. Listing of Mathematica® code

```

In[1]:=(* One element model for tip-moment problem using quadratic quadrilateral element and 3x3 inte-
gration *)
In[2]:=(* Legend: L=length of the element; d=depth of the element; thick=thickness of the element;
Y=Young's modulus; nu=Poisson's ratio; NN=shape function matrix; *)
In[3]:=(* Input data *)
In[4]:=aspectratio=1/100000;L=10;d=L*aspectratio;
In[5]:=x1=0;y1=0; x2=L;y2=0; x3=L;y3=d; x4=0;y4=d;
In[6]:=x5=L/2;y5=0; x6=L;y6=d/2; x7=L/2;y7=d; x8=0;y8=d/2;
In[7]:=Y=600; nu=0.25;thick=1;
In[8]:=xc=Transpose[{{x1,x2,x3,x4,x5,x6,x7,x8}}];
In[9]:=yc=Transpose[{{y1,y2,y3,y4,y5,y6,y7,y8}}];
In[10]:=(* Shape functions *)
In[11]:=NN = { { -1/4 + et^2/4 + etxi/4 - et^2xi/4 + xi^2/4 - etxi^2/4 }, { -1/4 + et^2/4 - etxi/4 + et^2xi/4 + xi^2/4 - etxi^2/4 },
{ -1/4 + et^2/4 + etxi/4 + et^2xi/4 + xi^2/4 + etxi^2/4 }, { -1/4 + et^2/4 - etxi/4 - et^2xi/4 + xi^2/4 + etxi^2/4 },
{ 1/2 - et/2 - xi^2/2 + etxi^2/2 }, { 1/2 - et^2/2 + xi/2 - et^2xi/2 }, { 1/2 + et/2 - xi^2/2 - etxi^2/2 },
{ 1/2 - et^2/2 - xi/2 + et^2xi/2 } };
In[12]:=(* Local derivatives of shape functions *)
In[13]:=dNdx=Flatten[D[NN,xi]];
In[14]:=dNdet=Flatten[D[NN,et]];
In[15]:=dNxiet={dNdx,dNdet};
In[16]:=(* Jacobian matrix *)
In[17]:=dxdxi=Simplify[dNdx.xc][[1]];dxdet=Simplify[dNdet.xc][[1]];
In[18]:=dydxi=Simplify[dNdx.yc][[1]];dydet=Simplify[dNdet.yc][[1]];
In[19]:=Jac={{dxdxi,dxdet},{dydxi,dydet}};
In[20]:=JacInv=Inverse[Jac];
In[21]:=(* Global derivatives of shape functions *)
In[22]:=dNxy=Simplify[JacInv.dNxiet];
In[23]:=(* B-matrix *)
In[24]:=s=dydet/dxdxi;
In[25]:=penalty=s^(-1+Exp[-N[Log[10,s]/5]])
Out[25]=2.56207*10^-9
In[26]:=Bmat={{dNxy[[1,1]],0,dNxy[[1,2]],0,dNxy[[1,3]],0,dNxy[[1,4]],0,dNxy[[1,5]],0,
dNxy[[1,6]],0,dNxy[[1,7]],0,dNxy[[1,8]],0},
penalty*{0,dNxy[[2,1]],0,dNxy[[2,2]],0,dNxy[[2,3]],0,dNxy[[2,4]],0,
dNxy[[2,5]],0,dNxy[[2,6]],0,dNxy[[2,7]],0,dNxy[[2,8]]},
{dNxy[[2,1]], dNxy[[1,1]],dNxy[[2,2]],dNxy[[1,2]],dNxy[[2,3]],dNxy[[1,3]],
dNxy[[2,4]],dNxy[[1,4]],dNxy[[2,5]],dNxy[[1,5]],dNxy[[2,6]],
dNxy[[1,6]],dNxy[[2,7]],dNxy[[1,7]],dNxy[[2,8]],dNxy[[1,8]]} };
In[27]:=(* D-matrix *)
In[28]:=const=Y/(1-nu*nu);
In[29]:=Dmat=const*{{1,nu,0},{nu,1,0},{0,0,(1-nu)/2}}
Out[29]={{640.,160.,0},{160.,640.,0},{0,0,240.}}

```

```

In[30]:=(* Stiffness-matrix by 3×3 integration *)
In[31]:=ker=Simplify[Transpose[Bmat].Dmat.Bmat]*thick*Det[Jac];
In[32]:=g=Sqrt[3/5];
In[33]:=K11=Simplify[ker/.xi[Rule]-g/.et[Rule]-g];K12=Simplify[ker/.xi[Rule]0/.et[Rule]-g];
      K13=Simplify[ker/.xi[Rule]g/.et[Rule]-g];
In[34]:=K21=Simplify[ker/.xi[Rule]-g/.et[Rule]0];K22=Simplify[ker/.xi[Rule]0/.et[Rule]0];
      K23=Simplify[ker/.xi[Rule]g/.et[Rule]0];
In[35]:=K31=Simplify[ker/.xi[Rule]-g/.et[Rule]g];K32=Simplify[ker/.xi[Rule]0/.et[Rule]g];
      K33=Simplify[ker/.xi[Rule]g/.et[Rule]g];
In[36]:=w1=5555555555555555/10000000000000000; w2= 8888888888888888/10000000000000000;
      w3=5555555555555555/10000000000000000;
In[37]:=K=(K11*w1*w1 + K12*w1*w2 + K13*w1*w3)+(K21*w2*w1 + K22*w2*w2 +
      K23*w2*w3)+(K31*w3*w1 + K32*w3*w2 + K33*w3*w3);
In[38]:=K=Simplify[K];
In[39]:=(* Application of boundary conditions and loads *)
In[40]:=Do[{K[[1,i]]=0,K[[i,1]]=0},{i,1,16}]; K[[1,1]]=1;
In[41]:=Do[{K[[7,i]]=0,K[[i,7]]=0},{i,1,16}]; K[[7,7]]=1;
In[42]:=Do[{K[[15,i]]=0,K[[i,15]]=0},{i,1,16}]; K[[15,15]]=1;
In[43]:=Do[{K[[16,i]]=0,K[[i,16]]=0},{i,1,16}]; K[[16,16]]=1;
In[44]:=P=Transpose[{{0,0,1,0,-1,0,0,0,0,0,0,0,0,0,0,0}}];
In[45]:=(* Solving for displacements *)
In[46]:=U=Inverse[K].P
Out[46]={{0.}, {243945}, {1000.0051241442894}, {1.0024440432498237`*8}, {-1000.01},
{1.00244×108}, {0.}, {243944}, {500.003}, {2.52441×107}, {0.000529244}, {1.×108},
{-500.002}, {2.52441×107}, {0.}, {0.}}
In[47]:=(* Verical displacement of node 6 *)
In[48]:=SetPrecision[N[U[[12]],17]
Out[48]={1.0000045930621749×108}

```